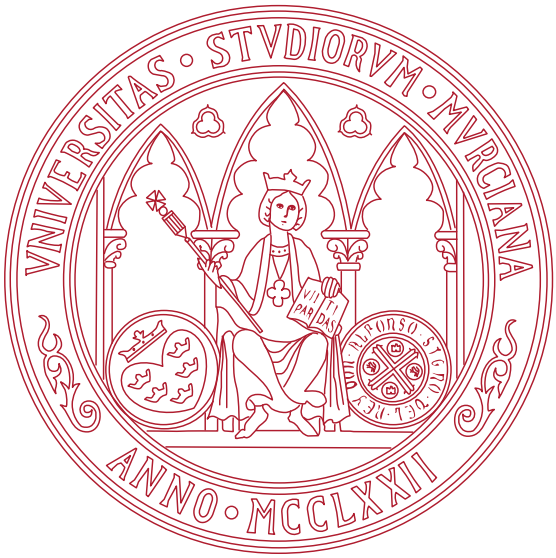
UNIVERSIDAD DE MURCIA

DEPARTAMENTO DE INGENIERÍA DE LA INFORMACIÓN Y LAS COMUNICACIONES

Gestión de la Confianza en Redes Colaborativas Centradas en el Intercambio Seguro de Alertas para la Detección de Ataques Distribuidos

D. Manuel Gil Pérez

2015



UNIVERSIDAD DE MURCIA

DEPARTAMENTO DE INGENIERÍA DE LA INFORMACIÓN Y LAS COMUNICACIONES

Trust Management in Collaborative Networks Focused on Securely Sharing Alerts for Detecting Distributed Attacks

Gestión de la Confianza en Redes Colaborativas Centradas en el Intercambio Seguro de Alertas para la Detección de Ataques Distribuidos

Author

Manuel Gil Pérez

Thesis advisors

Dr. Antonio Fernando Skarmeta Gómez

Dr. Gregorio Martínez Pérez

Murcia, October 2015

Abstract

Systems based on Information and Communication Technologies (ICT) are under a constant threat of attack attempts driven by malicious entities. Two of the potential purposes of these attacks are i) to break the security policies of such systems to access confidential information and ii) to deactivate certain services that these systems offer to customers. Numerous detection tools and techniques have recently appeared to protect and defend these systems, by reporting incidents involving malicious activities.

Until a few years ago, the attackers used to follow an "easily" recognizable pattern. They conducted point-to-point attacks, between the attacker and the victim, so just deploying an *Intrusion Detection System* (IDS) that monitors the victim was normally a quite adequate mechanism. This modus operandi, nevertheless, has evolved towards a more distributed attack paradigm, much more complicated to detect. Today, attackers leverage, amongst others, the largest number of devices connected to the network to launch their attacks from multiple sources towards a given set of victims.

As a way to tackle this new paradigm, the previous IDS-based solutions have been re-claimed as the most important tools for the detection of *distributed attacks*, although requiring certain settings. Monitoring capabilities of a single IDS only allow it to detect suspicious activities in a very limited portion of the whole detection environment to which it belongs. Usually, the computation system or network segment where the IDS is installed. Due to these limitations, alarms generated by a single IDS could be considered as isolated incidents, with a marginal significance in a distributed context. Therefore, the success in detecting distributed attacks becomes a deployment problem where a large number of IDSs must be spread out over the detection network, thereby creating a totally uncoupled system amongst multiple detection units. The use of multiple IDSs spread out amongst the security or administrative domains can offer better detection coverage in multi-domain environments, where the sources or targets of a given attack can come from or address several points within the underlying network.

Diversifying monitoring processes, by means of a strategic deployment of the IDSs in distinct security or administrative domains, forces to establish a close collaboration amongst IDSs to exchange all relevant information for detecting distributed attacks; mainly: alerts, incidents, and attacks detected by each IDS in an individual fashion. As a result, the logical union amongst IDSs forms an overlay layer for exchanging security information between peers called *collaborative alert system*. It aims at building up a collective knowledge base of alerts, acquired by sharing all isolated incidents detected by each IDS individually and autonomously, which allows having a holistic view about

what is actually happening in the system being protected. A more global perception on the system will provide a better opportunity of detecting distributed attacks.

In distributed systems like the ones outlined earlier, the growing expansion in the end users' mobility also deserves special attention, due to the new possibilities that the users have in purchasing new devices with high computing and wireless communication capabilities. Devices belonging to *mobile users* may be used by the collaborative alert system as "small" detection units from which to gather further security alerts. Adopting such alerts can help the system to enhance its coverage in detecting distributed attacks, but also to receive input (alerts) regarding certain zones where the system has deployed a pool of IDSs providing contradictory alerts in their detection behaviour. For example, a given number of IDSs is alerting the occurrence of an event altering the security policies of the system, whereas other IDSs deployed in the same detection zone do not provide similar alerts even when being configured accordingly to do so.

Malicious acts of IDSs are a consequence, almost certainly, of being compromised by a given attacker whose main purpose is to interfere in the proper operation of the system, so as to camouflage subsequent attacks without being detected. Therefore, the aim of these IDSs with a malicious attitude is, in the attacker's power, to produce errors in the detection processes of the rest of the IDSs, sending them out information and alerting them to an incident that did actually occur or deliberately obviating to inform them about facts that did. In this context, the non-detection of false (fraudulent) alerts can lead to an IDS to be fully ineffective in detecting either local or distributed attacks, where potential attackers can launch their offensive from different sources that is intended for more than one target as a their main objective.

The diversification in the monitoring processes aforementioned opens new research opportunities focused on how to manage security within distributed detection systems and what sort of information can be exchanged amongst their IDSs. This fact indicates a need for the definition and deployment of security frameworks to truly enable end-to-end secure communications, in order to preserve and ensure each communication link as well as the data information shared between stakeholders. This reflects an even greater challenge when the multi-domain scenarios are being taken into consideration, where a given number of security or administrative domains must collaborate in order to detect distributed attacks, by sending them out with each other the alerts that each of their IDSs are capable of generating and sharing internally.

Hereinafter, the main factors that have been recently confirmed as new challenges and opportunities in detecting attacks, although this time focused on a fully distributed environment, are highlighted with the ones described as follows:

- The design of a strategic *placement model* of all IDSs that allows maximising the detection coverage when identifying distributed attacks, also addressing the scalability issues associated with the retrieval and the subsequent analysis of huge amounts of distributed information (mainly alerts) in a real-time fashion.

  How to manage this wealth of information suggests that this model will have a great importance in the performance of a collaborative alert system.

- The *interoperability* amongst the different IDS-based solutions coexisting within the same collaborative alert system. All these IDSs have to achieve a common and consistent understanding when representing the same information, since each of the previous solutions can have its particular detection capabilities as well as its proprietary formats to represent the same detected incident.

- The *security in the communications* that guarantees the identity of the IDSs and the security or administrative domains participating in the collaborative alert system as legitimate bodies; the confidentiality and integrity of the information exchanged amongst all parties; and the preservation of privacy in sensitive data dissemination before being transmitted along the network, especially those parts that could be used by an attacker to find out vulnerabilities of potential victims.

- The *trust management* on the information that each entity receives from others before endorsing it as true. An ill-intentioned behaviour from one or more entities, sending out fraudulent information, can lead the system to suspect alteration attempts of its security policies, when it is not real. Because of this, the detection units should only base their decisions on analysing the information sent by entities as reliable as possible. If not, the system may launch certain response mechanisms that might change the state of security in which the system was.

Regarding the latter challenge, the trust management for modelling the behaviour of the IDSs is claimed as a must tool in assessing alerts. Furthermore, the collaboration of both the IDSs, distributed across different security or administrative domains, and the "small" detection units provided by mobile users, is faced to two well-known challenges in trust management systems. In particular, two problems appear which are related to the assignment of initial trust scores to new entities (newcomers) that want to join a collaborative system. These two problems are known in the current literature as *cold-start* and *bootstrapping*, differing respectively in whether it is the first time that the entity participates with the system or it has already done earlier in other parts of the system. Both problems are also extrapolated to any collaborative environment where entities need to join each other, at least once, for cooperation purposes. Computing the initial trust in the cold-start problem is a common issue to all entities in a collaborative system, whereas the bootstrapping issue specifically affects highly dynamic scenarios, where mobile entities cooperate with each other, or with other system infrastructure entities, along their path of travel. Because of the great interest in this new paradigm of collaboration, both problems have also been incorporated as distinct examples in the last challenge of managing trust in distributed environments.

Achieving all these challenges is part of the security management when exchanging detection information, this having different security levels according to the placement model established amongst IDSs. These levels can vary depending on whether IDSs belong to the same (*intra-domain*) or different administrative domains (*inter-domain*). This allows offering a common policy to all the IDSs of the same administrative domain, as it is managed by a single entity. In the context of this doctoral thesis, it is proposed:

- A trust management model at intra-domain level, with which the system can check the probability that the alerts shared between the IDSs of a determined security domain are classified as actual events observed in reality. That is, if these alerts can be considered as true or false positives for detection purposes.
- Another trust management model at inter-domain level, with which to prove the trustworthiness of the different security or administrative domains when assessing the different alerts exchanged between them as true or false positives.

The doctoral dissertation herein presented is elaborated on the distributed context introduced earlier, trying to satisfy the four major challenges as follows. The main goal is focused on contributing a collaborative alert system capable of detecting distributed attacks (*placement model*), by making use of a collective knowledge base built up from all the alerts generated by numerous IDSs deployed throughout the detection network (*interoperability*), also assuming both the existence of detection units with malicious attitude in behaviour (*trust management*) and adopting alerts from mobile users. The *cold-start* and *bootstrapping* problems also take on a singular prominence.

Owing to the high risk in exchanging information in the collaborative alert system, this doctoral thesis also addresses the *security in the communications* challenge. This security is mandatory for all communications, as exchanging information is required to detect attacks across public and open networks in intra- and inter-domain environments –within the same security domain and amongst security or administrative domains, respectively. The definition of a trust model based on public key cryptography has been proposed to endow the system with a mechanism that provides a given degree of security in information sharing, with which security in multi-domain scenarios is also addressed. Lastly, the collaborative alert system is strengthened with a mechanism with which to acquire a given degree of autonomy to maximise the decision-making process quality in detecting an attack. Decisions made on alerts with different degrees of confidence. This maximisation entails to gain better evidences in detecting distributed attacks, which is essential for minimising risk exposure. This will become the ability to properly assess how much trust we can place in each piece of information and what risks we incur by believing or not believing it. Blindly accepting each report as truth is certainly dangerous. In this way, the system will be able to switch the *placement model* of the infrastructure IDSs –moving them to other detection networks– or to reconfigure them with other monitoring policies in order to adapt the detection capabilities according to the behaviour they displayed when generating previous alerts.

It must be emphasised that two points listed earlier were not adopted in defining the objectives of this doctoral dissertation. First, the privacy management in exchanging information is proposed as future work, this being significant enough to warrant security amongst the components on multi-domain scenarios. Secondly, the optimisation of the detection mechanisms consuming huge amounts of information is proposed as future work. In any case, this latter challenge has been taken into consideration in the proposed collaborative alert system when designing the placement model, in order to exchange the least possible amount of information without compromising its detection rate.

A mis padres y abuelos

Índice general

1. Introducción y objetivos **1**
- 1.1. Hacia la detección de ataques distribuidos . . . 2
  - 1.1.1. Sistemas de detección de intrusiones o ataques . . . 2
  - 1.1.2. Requisitos para la detección colaborativa de ataques . . . 3
- 1.2. Seguridad en el intercambio de información . . . 5
- 1.3. Unidades de detección con actitudes maliciosas . . . 7
  - 1.3.1. Confianza en las entidades emisoras de información . . . 8
  - 1.3.2. Movilidad de las unidades de detección . . . 9
- 1.4. Calidad en procesos distribuidos de detección . . . 11
- 1.5. Objetivos y estructura de la tesis doctoral . . . 12
- 1.6. Aportaciones de esta tesis doctoral . . . 14
- 1.7. Publicaciones relacionadas y contribuciones a proyectos de investigación 17

2. Estudio de los sistemas de confianza en entornos multidominio **23**
- 2.1. Retos en la detección colaborativa de ataques . . . 24
  - 2.1.1. Reducción del número de falsas alarmas . . . 24
  - 2.1.2. Incapacidad para detectar nuevos ataques . . . 26
  - 2.1.3. Gestión de gran cantidad de información en tiempo real . . . 28
  - 2.1.4. Escalabilidad y robustez al tener que analizar grandes volúmenes de información . . . 29
  - 2.1.5. Incremento de ataques de denegación de servicios . . . 29
  - 2.1.6. Seguridad de las herramientas de detección . . . 30
  - 2.1.7. Interpretación de los datos cifrados . . . 31
  - 2.1.8. Incremento de herramientas automáticas de ataque . . . 32
  - 2.1.9. Reacción rápida ante los incidentes ocurridos . . . 32
- 2.2. Intercambio seguro de alertas de detección con Infraestructuras de Clave Pública . . . 33
  - 2.2.1. Principales modelos de certificación . . . 34
  - 2.2.2. Construcción de caminos de certificación candidatos . . . 39
  - 2.2.3. Mecanismos y protocolos de validación . . . 40
  - 2.2.4. Servicios de una PKI para entornos multidominio . . . 42
- 2.3. Sistemas para detectar ataques distribuidos . . . 44
  - 2.3.1. Criterios en el diseño de un sistema colaborativo . . . 45

2.3.2. Modelos para la detección de ataques distribuidos 49
2.4. Modelos de confianza basados en reputación 52
2.4.1. Fuentes de información para el cálculo de la reputación 53
2.4.2. Sistemas colaborativos para la detección de intrusiones basados en confianza y reputación 58
2.4.3. Confianza inicial de nuevas entidades 61
2.5. Maximización de la calidad en la detección 64
2.5.1. Reubicar los IDSs bajo otro modelo de despliegue 66
2.5.2. Reconfigurar las capacidades de detección de los IDSs 66
2.5.3. Puesta en marcha de un nuevo modelo de despliegue 67
2.6. Conclusiones del capítulo 69

3. Gestión de la confianza basada en PKI para entornos multidominio 73
3.1. Extensiones de los certificados X.509 75
3.1.1. AuthorityKeyIdentifier y SubjectKeyIdentifier 76
3.1.2. KeyUsage 76
3.1.3. CertificatePolicies 77
3.1.4. PolicyMappings 77
3.1.5. BasicConstraints 77
3.1.6. NameConstraints 77
3.1.7. CRLDistributionPoints 77
3.1.8. AuthorityInfoAccess y SubjectInfoAccess 78
3.2. Definición de un modelo de confianza para una federación de PKIs . . . 78
3.2.1. Construcción de una federación de PKIs 79
3.2.2. Diseño de un algoritmo de construcción y validación de caminos de certificación 80
3.3. Despliegue de una federación de PKIs para un escenario multidominio . 86
3.4. Pruebas de rendimiento 90
3.4.1. Cumplimiento de los requisitos en un entorno real 91
3.4.2. Validación del algoritmo en un entorno de laboratorio 96
3.4.3. Medidas de rendimiento en entornos reales 102
3.5. Conclusiones del capítulo 104

4. Confianza en redes colaborativas de detección de intrusiones 107
4.1. Diseño de un sistema colaborativo de alertas 108
4.1.1. Descripción de la arquitectura del sistema 109
4.1.2. Comunicaciones en el sistema de detección 112
4.2. Arquitectura de un componente del CIDN 114
4.2.1. Componentes de una entidad del CIDN 114
4.2.2. Normalización de alertas y reglas de detección 119
4.3. Sistema de reputación intradominio 119
4.3.1. Cálculo de la reputación de un IDS 120
4.3.2. Proceso de elección del Comité de Sabios 123

4.4. Perfil de comunicaciones intradominio 125
4.5. Resultados experimentales 131
4.5.1. Cobertura de la detección de un CIDN 132
4.5.2. Pesos en las recomendaciones sobre un IDS 135
4.6. Conclusiones del capítulo 137

5. Confianza multidominio en un sistema colaborativo de alertas 139
5.1. Sistema de reputación interdominio 140
5.1.1. Satisfacción sobre la alerta publicada por otro CIDN 142
5.1.2. Credibilidad en la opinión suministrada por otro CIDN 142
5.1.3. Modelado de los factores de contexto 143
5.2. Perfil de comunicaciones interdominio 145
5.3. Resultados experimentales 147
5.3.1. Ratio de la detección del sistema colaborativo de alertas 148
5.3.2. Cantidad aleatoria de los IDSs que componen un CIDN 150
5.4. Evaluación de nuevas unidades de detección 152
5.5. Sistema de reputación de nuevas unidades 155
5.5.1. Modelado de las capacidades de detección 156
5.5.2. Recomendaciones de fuentes de información del CAS 161
5.5.3. Niveles de garantía en los mecanismos de autenticación 163
5.5.4. Cálculo de la reputación inicial de una nueva unidad 164
5.6. Perfil de comunicaciones para nuevas entidades 171
5.7. Resultados experimentales 175
5.7.1. Evaluación de la cobertura de la detección 176
5.7.2. Valoración de los IDSs estáticos y móviles al variar sus estados de comportamiento 178
5.8. Conclusiones del capítulo 180

6. Reducir la incertidumbre en un sistema colaborativo de alertas 183
6.1. Elementos del sistema de una organización 184
6.1.1. Elementos adicionales para el CAS 184
6.1.2. Evaluación del sistema de monitorización 186
6.2. Configuración de un sistema de monitorización 187
6.2.1. Descripción del problema 188
6.2.2. Restricciones en un sistema genérico de información 190
6.3. Modelo adaptativo de la monitorización 192
6.3.1. Definición de estado como una matriz de configuración 192
6.3.2. Arquitectura para buscar y desplegar un nuevo estado 193
6.4. Reputación en la diversidad de la confianza 196
6.4.1. Cálculo de la diversidad de la confianza 197
6.4.2. Modelado del comportamiento de los IDSs 199
6.4.3. Actualización de la reputación de los IDSs 200
6.4.4. Decisión en la evolución hacia una nueva configuración 203

6.5. Resultados experimentales 205
6.5.1. Mejora en la evaluación de la confianza 205
6.5.2. Resistencia frente al mal comportamiento de los IDSs 209
6.5.3. Evaluación de los niveles de acuerdo entre los IDSs 211
6.6. Conclusiones del capítulo 212

7. Conclusions and future works **213**
7.1. Conclusions 213
7.2. Future work 218

Bibliografía **221**

Índice de figuras

2.1. Principales modelos de certificación 35
2.2. Relación de confianza mutua mediante dos certificados cruzados 37
2.3. Definición de los objetos a almacenar por un Servicio de Directorio . . 43

3.1. Algoritmo de construcción y validación de caminos de certificación . . . 83
3.2. Escenario multidominio de pruebas 88
3.3. Extensiones establecidas como requisito y no definidas por la FBCA . . 93
3.4. Los cinco procesos parciales que definen el Servicio de Validación 97
3.5. Tiempos promedios según la longitud del camino de certificación y el método de revocación: a) CRL/ARL y b) OCSP 99

4.1. Arquitectura de un sistema colaborativo de alertas con cinco CIDNs . . 111
4.2. Ejemplo de un sistema de publicación/suscripción 112
4.3. Componentes que conforman una entidad del CIDN 114
4.4. Niveles de confianza en el cálculo de la severidad de las alertas 121
4.5. Diagrama de secuencia UML para las comunicaciones intradominio . . 126
4.6. Alertas bien o mal clasificadas, generadas por un determinado porcentaje malicioso de a) HIDSs o b) NIDSs y los cuatro niveles de severidad . . 132
4.7. Alertas bien o mal clasificadas según los pesos en las recomendaciones, con un 20 % de HIDSs maliciosos 136
4.8. Alertas bien o mal clasificadas según los pesos en las recomendaciones, con un 40 % de HIDSs maliciosos 137

5.1. Diagrama de secuencia para las comunicaciones interdominio 145
5.2. Reparto de alertas según pertenencia y porcentaje de CIDNs maliciosos 149
5.3. Distribución de alertas con un número aleatorio de IDSs 150
5.4. Ejemplo de los tres tipos de unidades que pueden unirse al CAS 152
5.5. Todos los posibles caminos de confianza entre D_E y D_A 153
5.6. Proceso de negociación para obtener mejores coberturas de detección . 159
5.7. Diagrama de secuencia UML para una nueva unidad de detección . . . 171
5.8. Coberturas de la detección después de ejecutar los procesos de cold-start y bootstrapping según las capacidades de detección de los IDSs 176
5.9. Variación de la reputación al aumentar el número de IDSs maliciosos . 178

5.10. Variación de la reputación cuando todas las entidades del CAS tienen un comportamiento malicioso 179

6.1. Ejemplo de configuración de monitorización 188
6.2. Arquitectura del despliegue de una configuración de monitorización . . 193
6.3. Evolución de los estados a lo largo del tiempo 196
6.4. Variaciones en el modelado de tiempo según varios factores de olvido . 202
6.5. Ejemplo para la obtención de un modelo de despliegue más confiable . 206
6.6. Confianza en las alertas según el porcentaje de IDSs maliciosos 209
6.7. Variación en los acuerdos al aumentar el número de IDSs maliciosos . . 211

Índice de tablas

2.1. Comparación de los principales modelos de certificación 39
2.2. Ventajas e inconvenientes de los principales esquemas de despliegue . . 49
2.3. Principales sistemas para la detección de ataques distribuidos 50
2.4. Características de las diferentes fuentes de información 57

3.1. Requisitos en las extensiones de los certificados X.509 76
3.2. Número de certificados y listas de revocación obtenidos de la FBCA . . 92
3.3. Requisitos software para el Servicio de Validación y hardware utilizado para su puesta en marcha en el escenario de pruebas multidominio . . . 96
3.4. Tiempos promedios y desviaciones estándares obtenidas para el camino de certificación más largo y el método de revocación utilizado 101
3.5. Tiempos promedios y desviaciones estándares en entornos reales 102

4.1. Pesos y variables del sistema de reputación intradominio 124
4.2. Alertas generadas y clasificadas como válidas según su severidad 134

5.1. Pesos y variables del sistema de reputación interdominio 144
5.2. Variables del sistema de reputación para nuevas unidades de detección . 170

6.1. Variables para la monitorización según la diversidad de la confianza . . 204
6.2. Confianza promedio en las alertas según varias medidas/operaciones . . 207
6.3. Alertas aceptadas y descartadas según los acuerdos alcanzados 208

Capítulo 1

Introducción y objetivos

El objetivo detrás de este primer capítulo es servir como introducción y motivación sobre la problemática principal en la que se enmarca esta tesis doctoral: la gestión de la confianza en la detección de ataques en entornos distribuidos. Para ello se profundiza en dos pilares clave que afectan a la seguridad en la detección de ataques distribuidos. En primer lugar, la *seguridad en las comunicaciones* entre los componentes del sistema de detección se aborda con soluciones basadas en criptografía de clave pública, con las que poder alcanzar altos niveles de seguridad en el intercambio de información frente a entidades no autorizadas, haciéndola ininteligible a entidades maliciosas y evitando alteraciones de su contenido. En segundo lugar, la *evaluación de la confianza* de las distintas unidades de detección que necesitan intercambiar información para detectar ataques en un contexto distribuido. Esta evaluación se basa en mecanismos de confianza y reputación con los que modelar el comportamiento de las unidades de detección, la cuales pueden ser IDSs individuales dentro de un mismo dominio de seguridad, que colaboran entre sí para la detección local de ataques, o también dominios de seguridad que desean colaborar con otros dominios en la detección de ataques distribuidos. El éxito en la detección de ataques sólo es posible si la información intercambiada procede de entidades con una actitud no maliciosa en sus comportamientos.

La intención con este capítulo también es dejar clara la definición de los objetivos de esta tesis doctoral, las aportaciones propias que han permitido resolver los retos planteados, una descripción de la estructura esbozada en capítulos con las soluciones aportadas y las publicaciones asociadas a este trabajo de investigación.

La metodología tras la cual se aborda este primer capítulo se centra en desarrollar un análisis del estado actual de cómo ha evolucionado la detección de ataques hacia un contexto totalmente distribuido, tomando un especial interés la seguridad tanto en sus comunicaciones como en el intercambio de información entre sus actores principales. Con este análisis se pretenden identificar las deficiencias en este nuevo contexto, que posteriormente se tratarán en profundidad en los capítulos subsecuentes para dar una serie de soluciones que posibiliten la detección de ataques distribuidos. Nótese que a lo largo de este capítulo se hacen referencias a ciertas tecnologías y modelos a modo introductorio, que posteriormente serán analizados con mayor detalle.

1.1. Hacia la detección de ataques distribuidos

Los sistemas basados en las tecnologías de la información y las comunicaciones siempre han estado bajo la constante amenaza de sufrir ataques por terceras entidades maliciosas, la mayoría de ocasiones orientadas a romper sus políticas de seguridad y así poder, entre otros objetivos, acceder a cierta información confidencial de la víctima o deshabilitar servicios concretos que ésta ofrece [1]. El objetivo de estos atacantes puede variar, entre otros, desde personas que buscan fama o reconocimiento frente al resto de la comunidad de hackers, espionaje industrial entre organizaciones o, incluso, por ciberterrorismo. En este último caso, los atacantes pretenden causar pánico o terror a una población para influenciar, por ejemplo, en lo político o en lo religioso [2].

Como solución en la defensa de estos sistemas han aparecido en las dos últimas décadas un amplio rango de tecnologías y herramientas de detección. Sus objetivos son la protección del sistema frente a la autenticidad, confidencialidad, integridad, disponibilidad y confiabilidad tanto de los componentes de la red de detección como de la información que éstos gestionan. Una de las soluciones que más se ha utilizado es el uso de métodos de *control de acceso* que incluyen una amplia variedad de tecnologías de seguridad, desde la autenticación y la gestión de la identidad, para controlar quién tiene acceso a qué servicios, hasta las tecnologías basadas en el filtrado del tráfico de red desde y/o hacia la red de comunicaciones del sistema de información.

1.1.1. Sistemas de detección de intrusiones o ataques

Muchas de las soluciones que se han planteado, como el control de acceso introducido arriba, permiten implantar un conjunto de mecanismos para que solamente los usuarios legítimos de un sistema puedan hacer uso de sus servicios y recursos. Sin embargo, existe otra gran cantidad de amenazas que podrían llevarse a cabo y que pasarían desapercibidas para cualquiera de las tecnologías anteriores. Por ejemplo, la posibilidad de sufrir un ataque de *Denegación de Servicio* (del inglés Denial of Service, DoS) [3]. En este sentido, los *Sistemas de Detección de Intrusiones* (del inglés Intrusion Detection System, IDS) [4, 5] se han consolidado como una de las tecnologías más eficientes en la monitorización de los parámetros de seguridad con las que un sistema pueda detectar comportamientos anómalos o maliciosos. Esta protección abarca tanto a los usuarios internos de la organización como a terceras entidades externas.

Las amenazas anteriores a las que se ha hecho referencia han cambiado en los últimos años hacia un modo de comportamiento mucho más complejo y sofisticado. Hasta hace unos años, los atacantes lanzaban sus actividades malintencionadas desde una única fuente de origen del ataque hacia una víctima en concreto, por lo que la instalación de un IDS para la monitorización de esa víctima era más que suficiente para detectar ataques contra sus servicios y recursos. Sin embargo, los atacantes han modificado sus modos de actuación en estos últimos años hacia un nuevo paradigma mucho más global, donde sus ataques ya no se realizan punto a punto, entre el atacante y la víctima, sino que son ejecutados desde múltiples fuentes de origen hacia una o varias víctimas.

Debido a los cambios en el comportamiento de los atacantes hacia un paradigma de ataque mucho más distribuido, las actividades ilícitas que los IDSs pueden detectar a día de hoy se consideran como alertas aisladas, con un significado casi inapreciable si se analizan de forma individual dentro del estrecho ámbito de detección donde el IDS está instalado [6]. Por ejemplo, un ataque a gran escala del tipo *portscan* –muy utilizado en las primeras fases de un ataque para descubrir las vulnerabilidades del software– podría pasar inadvertido para un IDS si no analiza también las alertas que otros IDSs están generando en otras zonas del sistema. Igualmente, un IDS podría generar un alto número de *falsos positivos* indicando que algo sospechoso ha ocurrido en su limitada área de detección, cuando realmente no es cierto, al menos desde un punto de vista global del sistema [7]. Si no contempla en sus procesos de detección lo que está ocurriendo en otras zonas del sistema, el IDS tendría que alertar sobre todas las posibles actividades sospechosas que pudieran ocurrir, aunque no correspondan a incidentes reales asociados a ataques contra la seguridad del sistema.

Varios trabajos [8, 9] han confirmado que hasta casi un 99 % de las alertas generadas por un IDS son habitualmente falsos positivos. Tomando Snort [10] como ejemplo, al ser el software de facto más utilizado en la detección de ataques orientado al análisis del tráfico de red, en [11] se presenta un estudio donde se concluye que Snort genera hasta un 96 % de alertas aisladas que realmente corresponden a falsos positivos, mientras que menos del 1 % son alertas relacionadas con incidentes que realmente han ocurrido en la red de detección. También es importante apuntar que en este estudio se ha analizado la generación de *falsos negativos* –errores en los procesos de detección de un IDS al no informar sobre un evento que realmente ha ocurrido en la red de detección. Snort genera un porcentaje de falsos negativos bastante inferior al de los falsos positivos, sobre el 7,23 %, aunque sigue siendo suficientemente significativo para ser ignorado. Sin embargo, otros estudios [12, 13] han demostrado que esos porcentajes de error pueden reducirse en gran medida, hasta el 90 % en el caso de falsos positivos y alrededor de un 80 % en el de falsos negativos, si los IDSs desplegados en la red de detección establecen ciertos mecanismos de colaboración para, además de posibilitar la detección de ataques distribuidos mediante el intercambio de las alertas que cada IDS genera por separado, también puedan reducir considerablemente el ratio de posibles errores (principalmente, falsos positivos) en los procesos autónomos de detección de los IDSs.

1.1.2. Requisitos para la detección colaborativa de ataques

Los problemas introducidos anteriormente hacen pensar que las alertas generadas por cada IDS tienen que ser analizadas desde un punto de vista más global para que la detección de ataques distribuidos sea más efectiva. Estas alertas pueden obtenerse desde dentro de la propia organización que está siendo monitorizada, pero también pueden recibirse de otros dominios administrativos con los que se tenga algún acuerdo de confianza para compartir (parte de la) información de seguridad. En cualquier caso, es un requisito indispensable que cada IDS de la red de detección comparta con el resto de IDSs las alertas aisladas que ha detectado de forma individual, advirtiendo

de posibles incidentes que hayan ocurrido en su área restringida de monitorización. Como resultado, cada IDS por separado puede incorporar a sus procesos internos de detección, especialmente los de *agregación* y *correlación*, todas las alertas recibidas de otros IDSs y así poder comprobar si realmente está ocurriendo, o ha ocurrido, un ataque distribuido [14, 15]. Además, el sistema de colaboración resultante también permitirá reducir, como se ha comentado anteriormente, el alto número de falsas alertas (falsos positivos y negativos) que cada IDS pueda producir individualmente.

A raíz de la compartición de información entre los IDSs, pertenecientes a uno o varios dominios de seguridad y/o administrativos, se puede ofrecer una mejor cobertura de la detección de ataques distribuidos en escenarios multidominio. Para mejorar la cobertura de la detección, así como la exactitud y precisión en la detección de ataques, es necesario el despliegue e instalación de un alto número de IDSs que monitoricen cada área de detección de forma independiente. Sobre esa base, la unión de todos los IDSs desplegados a lo largo de todo el sistema de detección formarán una red lógica de unidades de detección que van a permitir tener una percepción mucho más global de lo que está ocurriendo realmente en el sistema [16].

El despliegue de los IDSs se convierte por tanto en un punto de crucial importancia en la detección de ataques distribuidos. En este sentido, es necesaria la definición de una serie de requisitos, establecidos como objetivos, que cualquier sistema colaborativo para la detección de ataques distribuidos tiene que cumplir:

- *Cobertura de la detección.* El conjunto total de las capacidades de detección de los IDSs tienen que ofrecer la mayor cobertura posible en la detección de ataques. En este sentido, es necesario que sean monitorizadas todas las áreas de la red de detección donde se podría dar un ataque distribuido.

- *Alto rendimiento en tiempo real.* La gran cantidad de información que los IDSs deben analizar, especialmente el tráfico de red, puede convertirse en un problema intratable, viéndose comprometido el análisis y las capacidades de detección de los IDSs. Aunque este análisis en profundidad se podría hacer en procesos offline, la detección tardía de ataques podría acarrear serios problemas al sistema.

- *Escalabilidad.* Es necesaria la distribución de la carga de trabajo en el análisis de grandes cantidades de información entre un número suficiente de IDSs, para que sea eficiente y no se vean comprometidos conforme la información crezca en volumen. En caso contrario, podrían haber ataques que pasasen desapercibidos sin ser detectados si no se analiza toda esa información.

- *Robustez.* Los IDSs tienen que ser resistentes frente a cualquier ataque contra sus procesos o recursos de detección, ya que también están expuestos a que sean comprometidos como cualquier otro tipo de software. Los atacantes buscan, entre otros objetivos, deshabilitar ciertos IDSs para que no informen de incidentes sobre sus próximas actividades maliciosas (falsos negativos) u obligarles a que generen grandes cantidades de falsas alertas (falsos positivos) para desviar la atención del sistema y así poder ejecutar sus actividades sin que sean detectados.

- *Precisión en la detección*. Los IDSs deben analizar toda la información posible para sus propósitos de detección, y que ésta corresponda a hechos que hayan ocurrido realmente. La incorporación de información fraudulenta puede conducir a errores en los procesos de detección de los IDSs.

- *Seguridad*. La seguridad de los IDSs es vital para que el sistema sea fiable y de confianza, por lo que tiene que ser exigida tanto a las comunicaciones entre los IDSs como a la información que intercambian. Es indispensable que se conozca la procedencia de la información compartida –alertas, información de detección o de la red– para evitar que se comparta información fraudulenta.

Los cuatro primeros requisitos están muy relacionados con el modelo de despliegue estratégico de los IDSs que el sistema colaborativo de detección tiene que implantar para que sea escalable, robusto, que toda la información sea analizada en tiempo real y que la cobertura de la detección sea máxima. Por otro lado, los dos últimos se vinculan a la gestión de la confianza, cuya finalidad es constatar que la información de seguridad intercambiada corresponde con datos reales, y que no se han generado como resultado de un comportamiento malicioso por parte de los IDSs debido a un mal funcionamiento interno o después de que hayan sido comprometidos por un atacante.

1.2. Seguridad en el intercambio de información

Todos los requisitos definidos en el apartado anterior son de obligado cumplimiento para que un sistema colaborativo orientado a la detección de ataques distribuidos tenga el éxito esperado. Sin embargo, el intercambio de información de detección entre los IDSs que forman el sistema toma un especial protagonismo en entornos multidominio, ya que los IDSs pertenecientes a varios dominios de seguridad y/o administrativos se ven abocados a intercambiar información con otras entidades con las que, a priori, no comparten ninguna relación de confianza. Incluso podrían ser entidades totalmente desconocidas entre sí. Debido a esto, la *gestión de la seguridad* en el intercambio de información, con respecto tanto a los actores que participan en las interacciones como en el contenido de la información que se transmite por la red de comunicaciones, se revela como uno de los primeros objetivos que deben ser abordados.

El objetivo principal en la gestión de la seguridad es la protección del sistema frente a ataques que intenten comprometer la *confidencialidad* e *integridad* de la información intercambiada entre los IDSs, así como la *autenticidad* de las entidades que forman el sistema colaborativo para la detección de ataques distribuidos como entidades legítimas del mismo. Las tecnologías basadas en criptografía [17], especialmente las basadas en criptografía de clave pública, se han posicionado como una de las soluciones más eficaces en la protección de los sistemas frente a las tres propiedades anteriores, y que están reguladas por la gestión de la seguridad: confidencialidad, integridad y autenticidad. A esa lista sólo falta incluir la gestión de la *disponibilidad* para alcanzar el cumplimiento de los cuatro pilares básicos de la *seguridad de la información*.

Es importante subrayar en este punto que ninguna tecnología es capaz de ofrecer, a día de hoy, protección frente a las cuatro propiedades de la seguridad sin tener que recurrir a otras soluciones complementarias. La criptografía de clave pública ofrece protección a tres de las propiedades anteriores, pero no a la disponibilidad, por lo que es necesario que la criptografía de clave pública sea complementada con algún otro mecanismo para que, entre ambos, permitan la consecución de las cuatro propiedades requeridas por la seguridad de la información. En concreto, los sistemas de detección de intrusiones, introducidos en la sección anterior, permiten monitorizar la disponibilidad de los servicios y de los recursos que una organización ofrece a sus clientes. En cuanto uno de esos servicios o recursos dejen de estar disponibles, porque por ejemplo han sido víctimas de un ataque de *Denegación Distribuida de Servicio* (del inglés Distributed Denial of Service, DDoS) [18, 19] o desactivados por la intrusión de un atacante, los IDSs distribuidos por la red de detección tendrán notificación de lo ocurrido y podrán poner en marcha algún mecanismo automático de respuesta para su mitigación [20], o notificar a los administradores del sistema para que tomen las medidas oportunas.

La integración de la criptografía de clave pública en soluciones basadas en IDS le va a permitir a estas últimas tener evidencias más confiables sobre la posible ejecución de un ataque. El sistema colaborativo orientado a la detección de ataques podrá alcanzar entonces un mayor nivel de seguridad, tanto en el intercambio de información entre los sistemas de detección (IDSs y dominios de seguridad) como en la autenticidad de que las alertas han sido generadas y enviadas por entidades legítimas del sistema, después de que éstas hayan superado satisfactoriamente un *proceso de autenticación* [21].

En el contexto de un sistema colaborativo, especialmente el destinado a la detección de ataques distribuidos, la seguridad tiene que ser gestionada a dos niveles:

- *Seguridad intradominio.* Protección de las alertas que cada IDS ha detectado de forma autónoma antes de que sean transmitidas y compartidas entre todos los IDSs de un mismo dominio de seguridad.
- *Seguridad interdominio.* Protección en el intercambio de información entre los distintos dominios que tengan un acuerdo previo de colaboración, ya sea entre los dominios de una misma organización o entre las redes de detección pertenecientes a varios dominios administrativos u organizaciones [22].

El éxito en la autenticación de entidades y en el intercambio seguro de información, haciendo uso de la criptografía de clave pública, pasa por el despliegue y gestión de una *Infraestructura de Clave Pública* (del inglés Public Key Infrastructure, PKI) [23]. Una PKI ofrece los mecanismos necesarios para la construcción de un modelo de confianza entre sus entidades, permitiendo, entre muchos otros objetivos, que puedan establecer relaciones de colaboración bajo un modelo seguro de comunicaciones. De esta manera, y bajo el marco de seguridad que ofrece una PKI, los sistemas de detección de diferentes dominios de seguridad y/o administrativos tienen a su disposición un mecanismo para el intercambio seguro de la información, después de que cada uno haya autenticado a su homónimo haciendo uso de las credenciales de este último.

Los estándares relacionados con la especificación formal de una PKI hacen uso del formato estándar X.509 para la definición de la sintaxis de una credencial, siendo el *certificado X.509* el ejemplo principal de credencial en las soluciones basadas en PKI [24, 25]. Estos certificados son emitidos por una autoridad certificadora confiable, y firmados digitalmente para asignar a las entidades con una identidad digital que les identifique de forma segura y unívoca como una de sus entidades legítimas.

El modelo de seguridad intradominio definido más arriba se consigue instalando una PKI en el dominio donde se desea gestionar su seguridad. En cambio, la construcción de un modelo de seguridad interdominio conlleva el establecimiento de relaciones de confianza más avanzadas y complejas en su ejecución, creadas entre varios dominios de seguridad pertenecientes, posiblemente, a más de una organización. Otros requisitos, como la *interoperabilidad*, tienen una especial relevancia cuando tienen que establecerse nuevos modelos de seguridad y confianza, ya que éstos tienen que garantizar cierta eficiencia en la gestión y compartición de la información en entornos multidominio.

En este contexto, cómo una determinada entidad puede validar las credenciales de cualquier otra entidad antes de establecer una comunicación segura entre ambas, ya sea entre dos IDSs o entre dos dominios de seguridad, se convierte en un elemento crítico para el éxito deseado en la correcta ejecución de la comunicación. La clave en este punto todavía se convierte en un reto de mayor complejidad si la validación de las credenciales tiene que realizarse entre entidades cuya competencia criptográfica se enmarcan entre PKIs distintas. Es decir, entre dominios administrativos que, casi con total certeza, pertenecen a organizaciones diferentes.

1.3. Unidades de detección con actitudes maliciosas

En cualquier sistema de comunicaciones, la entidad que recibe cierta información de otra entidad tiene que validarla antes de que la acepte como verdadera. Es decir, que la información corresponde a datos veraces y que no se asocian a información inventada. Cualquier información falsa tiene que ser eliminada automáticamente para evitar errores en la entidad receptora durante su tratamiento y análisis.

Dentro del contexto de un sistema colaborativo orientado a la detección de ataques, el comportamiento malintencionado de un IDS, o de un dominio de seguridad, puede hacer que el resto de entidades obtenga información de detección falsa (por ejemplo, alertas inventadas) sobre la alteración de las políticas de seguridad, cuando el hecho en cuestión no ha ocurrido en la realidad. La sospecha infundada de una alteración de la seguridad puede inducir al sistema de detección a poner en marcha ciertos mecanismos de respuesta que podrían interferir en su correcto funcionamiento de operación. Aunque la situación inversa también podría ocurrir, a pesar de que suele estar infravalorada: la denegación malintencionada de tener que informar sobre eventos que han ocurrido en la realidad. En cualquiera de los dos casos anteriores, el sistema de detección vería comprometida su percepción sobre el estado de seguridad en el que se encuentra, debido a comportamientos deshonestos de sus entidades.

1.3.1. Confianza en las entidades emisoras de información

Como solución frente a comportamientos deshonestos, es necesaria la *gestión de la confianza* sobre las entidades emisoras de información. La evaluación de esa confianza hace que las transacciones de intercambio de información sean más seguras, aceptando exclusivamente como veraz toda aquella información proveniente de una entidad lo suficientemente confiable para considerar su comportamiento como honesto.

En este sentido, se ha optado por dos mecanismos para evaluar la confianza de una entidad emisora de información. Por un lado, la autenticación mediante criptografía de clave pública se ha presentado anteriormente como una de las tecnologías más eficaces en la identificación de una entidad y, en consecuencia, en la confianza que cualquier otra entidad puede depositar en ella. Sin embargo, la autenticación de cualquier entidad ni implica que su comportamiento vaya a ser el esperado ni es un proceso obligatorio para muchas de las entidades. El sistema no puede obligar la autenticación a entidades externas como, por ejemplo, los *usuarios móviles* introducidos en la siguiente sección. Por tanto, es necesaria la implantación de otros mecanismos que ayuden a identificar comportamientos malintencionados de una entidad. En concreto, los sistemas basados en *reputación* se han instaurado como uno de los mecanismos más prometedores en la gestión de la confianza para la detección de comportamientos maliciosos [26, 27].

El modelado del comportamiento de una entidad en un entorno computacional se basa en disciplinas conocidas como las socioeconómicas y, principalmente, en relaciones personales entre los seres humanos, viéndose éstas muy bien reflejadas en las actuales redes sociales. Por ejemplo, conceptos sociales como "los amigos de mis amigos son mis amigos" se están utilizando hoy en día en la definición de los modelos de confianza basados en reputación para la representación del comportamiento de una entidad.

Para su puesta en marcha, cada entidad tiene que evaluar la confianza que tiene en el resto de entidades teniendo en cuenta diversos factores como, por ejemplo, su propia experiencia con ellas según todas las interacciones realizadas en el pasado. Estas *experiencias directas* se consideran uno de los factores más importantes en la evaluación de una entidad, ya que se basan en las propias interacciones que la entidad evaluadora ha hecho anteriormente con la entidad que está siendo evaluada. Sin embargo, existen otros factores que también podrían ayudar a obtener un mejor cálculo de la confianza entre entidades. Por ejemplo, mediante la obtención de experiencias directas que otras entidades confiables han tenido con la entidad que se está evaluando [28]. A esta última fuente de información también se le conoce como *experiencias indirectas*.

En el contexto de un sistema colaborativo para la detección de ataques, la gestión de la confianza mediante mecanismos basados en reputación puede ayudar a modelar el comportamiento de las entidades en los dos niveles comentados anteriormente:

- *Confianza intradominio*. Probabilidad de que las alertas intercambiadas entre los IDSs de un mismo dominio de seguridad representen hechos reales.

- *Confianza interdominio*. Confianza entre los distintos dominios de seguridad y/o administrativos al evaluar como cierta la información que intercambian.

Toda la información de detección que no provenga de entidades con un mínimo nivel de confianza se considera como información falsa o incorrecta, como consecuencia de un comportamiento malintencionado. La no incorporación de información fraudulenta en los procesos de detección de los IDSs va a permitir que se tenga tanto una mejor precisión en la detección de ataques como una mayor robustez en los procesos o recursos de detección, al poder identificar amenazas derivadas de comportamientos maliciosos. Indudablemente, el rendimiento del sistema también se verá favorecido al reducir la gran cantidad de información que los IDSs tienen que analizar a, exclusivamente, aquella que represente hechos acontecidos en la realidad. Además, el sistema también tendrá una mejor percepción de la auténtica cobertura de la detección, ya que sabrá perfectamente que los activos que desea proteger están siendo monitorizados por un conjunto específico de IDSs con actitudes no maliciosas en su comportamiento.

1.3.2. Movilidad de las unidades de detección

Los procesos de detección de los IDSs que el sistema colaborativo tiene desplegados pueden complementarse con nuevas alertas procedentes de otras entidades externas con capacidades de detección. La adopción de estas alertas va a permitir que se mejore la cobertura de la detección, así como la escalabilidad de los procesos de detección y toma de decisiones. Por un lado, el sistema podría ver comprometida su cobertura de la detección debido a la existencia de IDSs muy poco fiables, al tener valores de reputación demasiado bajos. En consecuencia, el sistema no tendría en cuenta la mayoría de las alertas ya que el comportamiento de los IDSs se encuentra bajo sospecha, dejando al sistema bajo un estado de seguridad ciertamente comprometido. Por otro lado, los procesos de toma de decisiones también se verían fortalecidos con nueva información que le permitirían al sistema determinar si realmente está ocurriendo un ataque, o si los IDSs de la infraestructura se están comportando adecuadamente. Debido a ello, tanto la precisión en la detección como la robustez también se verían reforzados por la inclusión de alertas proporcionadas por entidades externas de detección. Finalmente, destacar que las nuevas unidades de detección también pueden proporcionar alertas que hasta entonces no se contemplaban como posibles amenazas frente a alteraciones de la seguridad. Podrían existir vulnerabilidades que pasaban desapercibidas para el sistema al no haber IDSs configurados para su detección.

Como principal elección en el uso de estas nuevas unidades de detección, el sistema colaborativo podría aprovecharse de los *dispositivos móviles* que actualmente poseen los usuarios finales, como portátiles o teléfonos inteligentes. Estos dispositivos presentan grandes capacidades de cómputo, además de ciertas características que los hacen muy interesantes como, por ejemplo, sus capacidades de movilidad. De esta manera, los dispositivos móviles de estos usuarios pueden ser utilizados por el sistema colaborativo como pequeñas unidades de detección de las que obtener alertas adicionales a las que ya recibe de los IDSs desplegados en la infraestructura. A cambio, el sistema podría conceder ciertos beneficios a esos usuarios para incentivar su colaboración como, por ejemplo, proporcionarles un mayor ancho de banda en sus conexiones.

Una vez que se ha accedido a obtener alertas desde unidades externas de detección, el sistema colaborativo para la detección de ataques también tendría que evaluar su confianza sobre estas nuevas entidades –los dispositivos móviles de los usuarios finales– antes de admitir sus alertas como verdaderas. Por tanto, estas unidades de detección se tienen que ver sometidas al control de un sistema de gestión de la confianza para el modelado de sus comportamientos, de la misma manera que también se reclama al resto de entidades que el sistema colaborativo tiene desplegadas en su infraestructura. Sin embargo, las grandes capacidades de movilidad de los usuarios poseedores de estos dispositivos suponen un nuevo reto ya que, en muchas ocasiones, el sistema de gestión de la confianza se tendrá que enfrentar a la evaluación de nuevas entidades sobre las que todavía no posee información para un cálculo inicial de la confianza.

La falta de información sobre una nueva entidad, totalmente desconocida hasta el momento para el sistema colaborativo al no tener constancia de ninguna transacción previa (por ejemplo, el envío de alertas), hace que este hecho se convierta en un gran desafío a la hora de asignarle un valor inicial de la confianza. Este reto se conoce en la literatura como el problema *cold-start* (arranque en frío) [29]. Este problema también es extensible a cualquier otra entidad de un sistema colaborativo, independientemente de sus objetivos, ya que tienen que unirse al sistema, al menos, una primera vez. A partir de ese momento, cualquier entidad del sistema, ya sea móvil o perteneciente a la infraestructura, se verá abocada a seguir el mismo modelo de confianza basado en la reputación para el modelado de su comportamiento.

La mayor diferencia entre las entidades pertenecientes a la infraestructura y las de los usuarios con dispositivos móviles es que estos últimos pueden moverse entre los distintos dominios de seguridad y/o administrativos del sistema y poder colaborar con cada uno de ellos. La primera vez que vayan a colaborar con alguno de los dominios, el sistema colaborativo se enfrenta al problema cold-start, donde no existe información sobre esa nueva entidad. Pero, a partir de ese momento, una nueva colaboración con cualquier otro dominio se considera como el problema *bootstrapping*, ya que han dejado de ser entidades desconocidas para el sistema al haber colaborado anteriormente con alguno de sus dominios [30]. Pero el principal inconveniente con respecto a este último problema es buscar en qué otros dominios ha participado el usuario móvil para obtener información previa de su comportamiento, y así poder realizar un cálculo mucho más preciso de la confianza inicial sobre esa nueva entidad. El problema bootstrapping es muy común en escenarios altamente dinámicos, donde las entidades móviles colaboran con múltiples dominios a lo largo de su trayectoria de movimiento.

El cálculo inicial de la confianza sobre una nueva unidad de detección, especialmente para el problema cold-start, supone un gran desafío ya que el dominio que tiene que realizar ese cálculo solamente dispondrá de información que la propia entidad le pueda proporcionar. Si es un IDS de la infraestructura, instalado por un administrador, el sistema colaborativo puede presuponer una "buena voluntad" inicial de colaboración, pero no así de los usuarios móviles. El problema bootstrapping también supone un gran desafío, aunque relativamente inferior, ya que al menos se podrá obtener información de otros dominios confiables con los que el usuario móvil ya haya colaborado.

1.4. Calidad en procesos distribuidos de detección

En la sección anterior se ha argumentado que la inclusión de alertas provenientes de entidades externas (en ese caso, dispositivos móviles) permite mejorar la cobertura de la detección en un sistema colaborativo orientado a la detección de ataques. Además del uso de esas entidades móviles, el sistema también puede incrementar la cobertura de la detección ayudándose de los IDSs de la infraestructura, configurados e instalados por los propios administradores del sistema, ya que éstos tampoco son elementos estáticos que siempre deban estar desplegados en la misma zona de detección ni tienen que estar siempre configurados con las mismas políticas de detección.

El sistema colaborativo de detección puede cambiar y/o modificar en un momento dado tanto la configuración como la ubicación de cualquier IDS de la infraestructura siguiendo un determinado interés general [31]. Por ejemplo, reubicando los IDSs de la infraestructura desde su posición hacia una zona de detección donde están ocurriendo actividades "sospechosas", o configurando en determinados IDSs una serie concreta de políticas de detección debido a que el sistema no es capaz de tomar una decisión con la información proporcionada por el resto de IDSs que tienen configuradas esas mismas políticas de detección. En cualquier caso, cuanto mayor información "útil" haya sobre un incidente, y que esta información provenga de entidades lo más confiables posible, menor *incertidumbre* tendrán los IDSs en la asunción de que un hecho haya ocurrido en la realidad a partir de su propia información y la recibida de otros.

La incertidumbre sobre una información de detección recibida aumenta debido a, principalmente, tres supuestos:

- El número de unidades de detección desplegadas en un dominio es escaso, sobre todo entidades de la infraestructura. En consecuencia, el sistema recibirá muy poca información, o de muy mala calidad, sobre lo que está ocurriendo.

- Un alto número de entidades desplegadas sobre el mismo dominio pueden alcanzar la misma entropía que en el supuesto anterior, si estas entidades tienen valores muy bajos en su reputación. El sistema no podría dirimir si la información ha sido generada como consecuencia de un comportamiento honesto o malicioso.

- Se recibe información contradictoria de los procesos de monitorización sobre las mismas fuentes de información. Unas entidades alertan sobre la alteración de ciertas políticas de seguridad mientras que otras, desplegadas en la misma zona de detección, no lo hacen aun cuando también están monitorizando el cumplimiento de esas mismas políticas de seguridad.

En este sentido, es necesaria la implementación y puesta en marcha en el sistema colaborativo de algún mecanismo que le permita adaptar sus capacidades de detección de forma autónoma, sin que tenga que intervenir un administrador, teniendo en cuenta, entre otros factores, el número y reputación de todas sus unidades de detección. Este mecanismo hará que la cobertura de la detección sea la mejor posible, tanto a nivel de

dominio de seguridad y/o administrativo como a nivel del propio sistema colaborativo para la detección de ataques. Este incremento en la cobertura de la detección también tendrá asociada una mejora sustanciable en la precisión de los procesos de detección, viéndose incrementada su robustez, y en la calidad para la toma de decisiones sobre la posible detección de un ataque, o sobre un incidente en particular.

La adaptación de las capacidades de detección del sistema colaborativo se puede conseguir mediante la reconfiguración de las propiedades de detección de los IDSs y/o ejecutando un *modelo de despliegue* óptimo de esos IDSs. Ambos tienen que ayudar a minimizar la incertidumbre en la toma de decisiones sobre la veracidad de las alertas. Cuando la ejecución de un modelo de despliegue deje de ser el adecuado, el sistema tiene que evaluar todos los posibles modelos que podrían aplicarse, teniendo en cuenta fundamentalmente las reputaciones de todos los IDSs, y cambiarlo por uno nuevo que proporcione mejores competencias cuando tenga que decidirse si una información es confiable o, por el contrario, si proviene de entidades con actitudes maliciosas.

1.5. Objetivos y estructura de la tesis doctoral

Una vez analizada la problemática en la que se enmarca esta tesis doctoral, el siguiente paso es definir los subobjetivos que se pretenden alcanzar y las aportaciones por las que se han optado para conseguir el objetivo principal que se ha planteado: definición, diseño y puesta en marcha de un *Sistema Colaborativo de Alertas* (del inglés Collaborative Alert System, CAS) para la detección de ataques distribuidos mediante el intercambio seguro de la información de detección en entornos multidominio.

En este primer capítulo se ha presentado el contexto en el que se desarrolla esta tesis doctoral. Como requisito fundamental se ha impuesto la gestión de la seguridad y la confianza en el intercambio de información de detección, haciendo un gran énfasis en estos conceptos durante su gestión en entornos multidominio. La definición de estos subobjetivos se desarrollan a continuación estableciendo, en cada caso, los capítulos donde luego se exponen en mayor detalle, mientras que las soluciones aportadas para la consecución de cada subobjetivo se abordan en la siguiente sección.

En primer lugar se necesita una minuciosa revisión del estado actual de las distintas herramientas, modelos y tecnologías que se han propuesto hasta el momento asociados con la gestión de la seguridad, enfocados principalmente a entornos multidominio, y a la detección de amenazas ejecutadas bajo un paradigma de ataque distribuido. En este estudio, se hará un especial hincapié sobre la gestión de la confianza de aquellos usuarios con dispositivos móviles, capaces de generar alertas sobre un área particular de la red de detección. En este punto se analizarán las distintas soluciones propuestas sobre qué confianza inicial se le puede asignar a esas entidades cuando se unen por primera vez al CAS (problema conocido como *cold-start*) o cuando ya han colaborado anteriormente y vuelven a unirse, aunque sea en otro dominio de seguridad con el que todavía no han colaborado (problema conocido como *bootstrapping*). La revisión del estado actual se desarrolla en su totalidad en el Capítulo 2.

El Capítulo 3 se centra en el primer subobjetivo planteado sobre la protección de las comunicaciones entre los diferentes actores que pueden participar en un CAS. La finalidad detrás de este subobjetivo es garantizar la preservación de la autenticidad, confidencialidad e integridad de la información que necesita compartirse, orientada a construir una base de conocimiento de alertas que permita la detección de ataques distribuidos. Con esta finalidad en mente, este capítulo presenta una solución basada en tecnologías de PKI para la construcción de modelos avanzados de confianza en entornos multidominio, donde es necesario el establecimiento de relaciones de confianza entre los distintos dominios de seguridad y/o administrativos que forman el CAS.

Estas relaciones se crean entre las PKIs que cada dominio de seguridad gestiona de forma individual, construyendo así una *federación de PKIs* que sirve como vehículo conector de la seguridad entre los distintos dominios del CAS. En la consecución de parte de este subobjetivo también se propone una serie de requisitos que sirven como base en la construcción e interoperabilidad de una federación de PKIs. Finalmente se especifica el diseño e implementación de un *Servicio de Validación* capaz de construir y validar los caminos de certificación en cualquier modelo avanzado de confianza. Este servicio sirve como medio principal en la validación de entidades legítimas del CAS antes de realizar cualquier intercambio de información de detección.

El siguiente subobjetivo que se ha planteado es el diseño y puesta en marcha de un sistema para la gestión de la confianza con el que se pueda modelar el comportamiento de los IDSs que forman parte de una *Red Colaborativa de Detección de Intrusiones* (del inglés Collaborative Intrusion Detection Network, CIDN). Cada CIDN corresponde con la red de detección de un dominio de seguridad en particular, por lo que la unión de todos los CIDNs forma el deseado sistema colaborativo de alertas (CAS).

Según la confianza que un CIDN tenga en sus IDSs, representada por sus valores de reputación, éste se encuentra en disposición de aceptar solamente las alertas que sean de confianza. Es decir, aquellas que son generadas por IDSs con un cierto valor mínimo de reputación para el CIDN. Estas alertas formarán parte de una base local de conocimiento de alertas (una por CIDN) para la detección de ataques. Las alertas consideradas como fraudulentas, compartidas por IDSs con una baja reputación, son marcadas como tales por el CIDN para que el resto de IDSs no las incorporen a sus procesos internos de detección. Este sistema permitirá la definición de un mecanismo basado en reputación para la gestión de la confianza en un entorno intradominio, el cual se aborda en el Capítulo 4 junto con la definición y diseño de un CIDN.

El sistema anterior sobre la gestión de la confianza intradominio, ejecutado por cada CIDN de forma independiente, servirá como base para la definición de un sistema homólogo, pero orientado a entornos multidominio. Este nuevo sistema corresponde a la propuesta de un tercer subobjetivo, donde es necesaria la gestión dentro del CAS de la confianza interdominio enfocada a que cada CIDN pueda modelar el comportamiento del resto antes de aceptar como válida cualquier información enviada por estos últimos. Únicamente la información que sea generada por CIDNs con una reputación suficiente, que demuestre la honestidad en sus comportamientos, será incorporada a la base global de conocimiento construida a nivel del CAS, entre todos los CIDNs.

De esta manera, la base global de conocimiento de alertas, imprescindible para la detección de ataques distribuidos, se encontrará distribuida entre todos los CIDNs que componen el CAS, cumpliendo de esta manera dos de los requisitos más importantes en cualquier sistema distribuido de información: la escalabilidad y la robustez.

En este contexto interdominio, y dentro del mismo subobjetivo, se planteará la incorporación de alertas provenientes de usuarios finales con dispositivos móviles. La adopción de esas alertas obligará a ampliar el sistema para la gestión de la confianza con un modelo particular que permita calcular una confianza inicial de estas nuevas unidades de detección. Este cálculo tiene que abordar los problemas conocidos como *cold-start* y *bootstrapping*, dependiendo de si el CIDN donde la entidad desea unirse puede obtener información previa para realizar ese cálculo. En el Capítulo 5 se presenta el diseño de este sistema interdominio, incluyendo el cálculo inicial de la confianza sobre cualquier entidad perteneciente al CAS, haciendo un especial énfasis a aquellas unidades de detección con capacidades individuales en su movilidad.

El último subobjetivo de esta tesis doctoral, abordado en el Capítulo 6, consiste en el diseño y puesta en marcha de un esquema adaptativo le permita al CAS reubicar y/o reconfigurar dinámicamente los IDSs de la infraestructura según sus comportamientos –valores de reputación. La finalidad es maximizar la calidad de la información obtenida tanto de los IDSs independientes como de los CIDNs a nivel de agrupación. Esta calidad está orientada a la toma de decisiones sobre la detección de un ataque, o sobre un evento en particular, a partir de la información de detección obtenida en los dos niveles comentados con anterioridad: a nivel local dentro de un CIDN y globalmente a nivel del CAS. Este mecanismo de reconfiguración dinámica se obtendrá cambiando el modelo de despliegue de los IDSs de la infraestructura (por ejemplo, trasladando los IDSs de un CIDN a otro) o bien modificando sus capacidades de detección para obtener nuevas evidencias que le permitan al sistema tener una mejor evaluación de los IDSs.

Finalmente, en el Capítulo 7 se presentarán las principales conclusiones extraídas como resultado de este trabajo de investigación, y se definirán aquellas líneas de mayor interés que no se han abordado en esta tesis doctoral y que se han dejado como posibles referencias para futuras investigaciones.

1.6. Aportaciones de esta tesis doctoral

A continuación se enumera una lista con las principales aportaciones que se han realizado en esta tesis doctoral, cada una explicada en detalle en los siguientes capítulos. Esta lista se ha estructurado siguiendo los cuatro grandes bloques introducidos en las secciones anteriores, y donde cada punto de un bloque en concreto se relaciona con alguno de los subobjetivos que se acaban de presentar.

Con respecto al primero de los subobjetivos planteados, el relacionado con la gestión de la seguridad en el intercambio de información mediante el uso de criptografía de clave pública, desarrollado en el Capítulo 3, se han obtenido una serie de aportaciones [32, 33], entre las que se pueden destacar las que se indican en la siguiente lista.

- Se ha propuesto la correcta definición de las extensiones de un certificado X.509 como una serie de requisitos (obligatorio, recomendado, opcional o no aplicable) para que haya una apropiada interoperabilidad entre los dominios de seguridad de una federación de PKIs, tanto en su fase de creación como en el modelo de confianza establecido. Aunque se han definido los requisitos para las 17 posibles extensiones, esta propuesta se centra principalmente en las que están relacionadas con el correcto funcionamiento de cualquier Servicio de Validación.

- Se han propuesto los mecanismos necesarios para la creación de una federación de PKIs mediante el establecimiento de las relaciones de confianza entre los dominios de seguridad con modelos de certificación cruzada, ya sean relaciones peer-to-peer o las establecidas a través de una *Bridge CA* (BCA) neutral [34].

- Se ha diseñado un *algoritmo para la construcción y validación de caminos de certificación* con el que poder validar certificados X.509 en entornos multidominio, mediante la recuperación de todo el material criptográfico –certificados y listas de revocación o respuestas OCSP– asociado a la solicitud recibida.

- Se ha definido la funcionalidad completa de un Servicio de Validación, el cual tiene que incorporar el algoritmo diseñado anteriormente para la construcción y validación de caminos de certificación.

- Se ha presentado un estudio donde se discuten los problemas de interoperabilidad y la correcta definición de las extensiones a partir del análisis de un gran número de certificados X.509. En este estudio se ha optado por analizar los certificados de la *Federal Bridge Certification Authority* (FBCA), al ser una de las federaciones de PKI operativas más representativas en la actualidad [35].

- Se ha especificado la definición y puesta en marcha de una federación de PKIs, desplegada en un escenario multidominio, a partir de la que se puedan extraer ciertas conclusiones relacionadas con los distintos factores que pueden influir en el rendimiento de un Servicio de Validación.

  Entre esos factores se prestará una especial atención al impacto que supone la longitud de los caminos de certificación en el rendimiento global y a dos de los mecanismos principales que permiten evaluar la revocación de los certificados que componen un camino de certificación: CRL/ARL y OCSP.

La consecución del segundo subobjetivo planteado, el relacionado con el diseño de un sistema de gestión de la confianza basado en reputación, enfocado concretamente al modelado del comportamiento de los IDSs en una red colaborativa de detección de intrusiones (CIDN), se ha conseguido mediante una serie de aportaciones [36, 37], las cuales se listan a continuación. Este subobjetivo se aborda a lo largo del Capítulo 4.

- Se ha diseñado un sistema colaborativo de alertas compuesto por múltiples y heterogéneas redes colaborativas de detección de intrusiones capaces de detectar ataques en entornos distribuidos.

Este diseño sigue un modelo de despliegue de los IDSs basado en un *esquema parcialmente descentralizado*, buscando ofrecer soluciones a varios inconvenientes que presentan los modelos centralizados o los totalmente descentralizados: falta de escalabilidad y sobrecarga en las comunicaciones, respectivamente.

- Se ha definido un *sistema de confianza intradominio basado en reputación* que permite modelar el comportamiento de los IDSs de un CIDN. A través de este sistema, el CIDN puede "predecir" si las alertas generadas por un IDS son como consecuencia de un acto honesto o malicioso. Es decir, si corresponden o no a un intento real de alteración de las políticas de seguridad del CIDN, o es debido a un mal funcionamiento del IDS (por ejemplo, al ser comprometido por un atacante).
- Se han propuesto las modificaciones necesarias que se tienen que realizar en los distintos módulos que componen los procesos de detección de un IDS para que incorporen la gestión de la confianza descrita en el punto anterior.
- Se ha definido el modelo de seguridad más apropiado para que las comunicaciones entre los IDSs de un CIDN sean lo más seguras posible. Este marco de seguridad se basa en soluciones de PKI para hacer factible la detección de ataques a través del intercambio de las alertas que cada IDS genera de forma individual.

El tercero de los subobjetivos planteados, que se desarrolla en el Capítulo 5, se asocia con el diseño de un sistema de gestión de la confianza basado en reputación, al igual que en el caso anterior, pero ahora orientado al modelado en entornos multidominio del comportamiento de los CIDNs [36, 38]. Además, este subobjetivo también incluye la posibilidad de adoptar alertas que un CIDN recibe desde otras unidades externas de detección con altas capacidades en su movilidad. En concreto, de dispositivos móviles pertenecientes a los usuarios finales [39]. A continuación se enumeran las aportaciones principales que se han obtenido para la consecución de este nuevo subobjetivo.

- Se ha definido un *sistema de confianza interdominio basado en reputación* con el que poder modelar el comportamiento de los distintos CIDNs pertenecientes al CAS. De esta manera, cada CIDN puede filtrar todas aquellas alertas que hayan sido generadas por otros CIDNs sospechosos de actitudes maliciosas. Es decir, alertas provenientes de CIDNs con una baja reputación.
- Se ha propuesto el diseño de un modelo de confianza basado en reputación capaz de calcular la confianza inicial sobre una nueva unidad de detección, ya sea un IDS de la infraestructura, un CIDN que desea unirse a otro CIDN para mejorar su cobertura y precisión en la detección o una unidad de detección con altos grados de movilidad que puede aportar un usuario final. Este cálculo permite dar solución a los problemas conocidos como *cold-start* y *bootstrapping*.
- Se ha propuesto el modelo de seguridad más adecuado, basado en soluciones de PKI, para que las comunicaciones multidominio entre los distintos CIDNs se lleven a cabo bajo un marco estable en su seguridad.

En este entorno multidominio también es un requisito esencial el uso de modelos avanzados de confianza, donde los CIDNs tienen que establecer previamente sus relaciones de colaboración bajo un modelo seguro en sus comunicaciones.

Con respecto al último subobjetivo, el relacionado con la definición de un esquema adaptativo capaz de poner en funcionamiento el mejor modelo de despliegue de los IDSs de la infraestructura según sus comportamientos, se han obtenido las aportaciones que se destacan a continuación [40]. Este subobjetivo se aborda en el Capítulo 6.

- Se ha diseñado un modelo adaptativo que permite a un CIDN, o a nivel del CAS, obtener y aplicar un modelo de despliegue de sus IDSs capaz de maximizar la confianza en sus procesos de detección. La aplicación de este modelo se desarrolla en los dos niveles que define un CAS: internamente entre los IDSs de un mismo CIDN y, de forma más global, entre todos los IDSs del CAS.
- Se ha propuesto un sistema de reputación basado en la *diversidad de la confianza* capaz de obtener el mejor modelo de despliegue posible, a partir de los valores de reputación de los IDSs y ciertas restricciones que son de obligado cumplimiento. Por ejemplo, limitaciones en las capacidades de detección de un IDS que no serían útiles para un determinado CIDN. La diversidad de la confianza se define como una métrica heurística que mide la calidad de cualquier modelo de despliegue.

Todas las aportaciones que se han presentado en los puntos anteriores, definidas como objetivos de esta tesis doctoral, han sido todas implementadas y desplegadas en distintos escenarios de aplicación. Cada uno de estos escenarios, y las configuraciones pertinentes que se han llevado a cabo para la correcta puesta en marcha de los mismos, han sido utilizados posteriormente para validar estas aportaciones, a través del análisis y su correspondiente discusión sobre todas las pruebas ejecutadas.

1.7. Publicaciones relacionadas y contribuciones a proyectos de investigación

El trabajo de investigación que se ha realizado a lo largo de esta tesis doctoral ha propiciado la publicación de diversos trabajos en congresos y revistas internacionales, cinco de ellos publicados e indexados en el *Journal Citation Report* (JCR) del *Institute for Scientific Information* (ISI).

A lo largo de esta sección se define una relación con los trabajos de investigación más significativos en el contexto de esta tesis doctoral, junto con una breve descripción de las aportaciones realizadas en cada uno de ellos. Todas las publicaciones asociadas a estos trabajos de investigación se presentan estructuradas según los cuatros subobjetivos que se han definido en las secciones anteriores, intentando dejar constancia que todos los subobjetivos planteados en esta tesis doctoral han quedado bien cubiertos a nivel de publicaciones en distintas conferencias y revistas internacionales.

En la siguiente lista se presentan las publicaciones más importantes con respecto al primero de los subobjetivos planteados, el relacionado con la gestión de la seguridad en el intercambio de información mediante el uso de criptografía de clave pública.

- G. López Millán, M. Gil Pérez, G. Martínez Pérez, A. F. Gómez Skarmeta. **PKI-based trust management in inter-domain scenarios**. Revista *Computers & Security*, vol. 29, no. 2, pp. 278–290, 2010.

  Este artículo [32] define los requisitos principales en términos de certificación cruzada que las infraestructuras de seguridad deben seguir para una correcta interoperabilidad entre cada uno de sus dominios administrativos. Estos requisitos se derivan a partir del estudio de los escenarios interdominio más relevantes en la actualidad, así como de la experiencia personal adquirida en varios proyectos de investigación donde se han desplegado este tipo de infraestructuras. Este trabajo también presenta un análisis en profundidad sobre el rendimiento que ofrece el algoritmo propuesto de construcción y validación de caminos de certificación en un entorno de laboratorio, así como su extrapolación a un entorno de certificación real sobre varias de las federaciones de PKI más representativas.

- A. Ruiz Martínez, D. Sánchez Martínez, C. I. Marín López, M. Gil Pérez, A. F. Gómez Skarmeta. **An advanced certificate validation service and architecture based on XKMS**. Revista *Software: Practice and Experience*, vol. 41, no. 3, pp. 209–236, 2011.

  En este artículo [33] se propone una extensión a la especificación XKMS [41] para incluir la definición de un servicio avanzado de validación de firmas electrónicas en largos periodos de tiempo. Esta extensión también permite que, en escenarios con múltiples dominios administrativos, cada uno de ellos pueda utilizar sus propios mecanismos de validación, ya sean estándares o propietarios, permitiendo así una máxima interoperabilidad para la validación de cualquier documento electrónico. En este trabajo también se presenta el despliegue de este servicio avanzado en un escenario real, que actualmente se encuentra en producción, como es la firma de actas de las asignaturas dentro del marco de la Universidad de Murcia.

A continuación se presenta la publicación con mayor relevancia para la consecución del segundo de los subobjetivos que se han definido anteriormente, el relacionado con el diseño de un sistema de gestión de la confianza basado en reputación. De manera más específica, este subobjetivo se centra en el modelado del comportamiento que pueden tener los distintos IDSs en un entorno intradominio, por lo que el diseño de una red colaborativa orientada a la detección de intrusiones también se ha considerado como un propósito necesario a tener en cuenta.

- M. Gil Pérez, F. Gómez Mármol, G. Martínez Pérez, A. F. Skarmeta Gómez. **RepCIDN: A reputation-based collaborative intrusion detection to lessen the impact of malicious alarms**. Revista *Journal of Network and Systems Management*, vol. 21, no. 1, pp. 128–167, 2013.

En este artículo [36] se presenta el diseño de una red colaborativa de detección de intrusiones que es capaz de construir una base de conocimiento común sobre las alertas detectadas individualmente por cada una de las unidades de detección desplegadas en la red, permitiendo así la detección de ataques distribuidos que antes eran imposibles de descubrir. En este trabajo también se define un modelo de confianza basado en reputación que permite la evaluación de las entidades emisoras de las alertas antes de que éstas sean publicadas. De esta manera, el sistema eliminará todas las alertas que provengan de unidades maliciosas que intenten publicar alertas que no representen incidentes reales.

Como en el subobjetivo anterior, el tercero de los subobjetivos planteados también se asocia con el diseño de un sistema de gestión de la confianza basado en reputación, pero en este caso orientado al modelado en entornos multidominio del comportamiento de los distintos CIDNs que forman el CAS. En este tercer subobjetivo también se incluye la posibilidad de admitir alertas provenientes de unidades externas de detección con altas capacidades de movilidad. A continuación se listan las publicaciones más significativas que han propiciado la consecución de este tercer subobjetivo.

- M. Gil Pérez, F. Gómez Mármol, G. Martínez Pérez, A. F. Gómez Skarmeta. **Mobility in collaborative alert systems: Building trust through reputation**. Actas del *Workshop on Wireless Cooperative Network Security*, vol. 6827 en el Lecture Notes in Computer Science, pp. 251–262, Mayo 2011.

  Este trabajo [38] presenta un modelo de confianza y reputación interdominio capaz de fortalecer un sistema colaborativo de alertas con un mecanismo para que pueda evaluar la validez de las alertas recibidas a través del comportamiento expuesto por las unidades de detección que las generan. Este modelo se centra en un entorno multidominio, donde los dispositivos que albergan esas unidades de detección se desplazan de un dominio de seguridad a otro, posiblemente entre distintos dominios administrativos de organizaciones dispares.

- M. Gil Pérez, V. Mateos Lanchas, D. Fernández Cambronero, G. Martínez Pérez, V. A. Villagrá. **RECLAMO: Virtual and collaborative honeynets based on trust management and autonomous systems applied to intrusion management**. Actas del *7th International Conference on Complex, Intelligent, and Software Intensive Systems*, pp. 219–227, Julio 2013.

  Este trabajo [37] presenta un sistema inteligente de respuestas automáticas frente a intrusiones o ataques capaz de inferir la respuesta más adecuada para un ataque. Además del tipo de ataque, junto con otra información necesaria como el contexto, el sistema de confianza diseñado ofrece un mecanismo para el descubrimiento de comportamientos sospechosos de las unidades de detección desplegadas en la red subyacente. Con toda esta información se propone un enfoque novedoso: desviar el ataque hacia una *honeynet* construida de forma dinámica y específica para ese ataque en particular. El funcionamiento de este sistema se contextualiza en un entorno de redes colaborativas de detección de intrusiones.

- M. Gil Pérez, F. Gómez Mármol, G. Martínez Pérez, A. F. Skarmeta Gómez. **Building a reputation-based bootstrapping mechanism for newcomers in collaborative alert systems**. Revista *Journal of Computer and System Sciences*, vol. 80, no. 3, pp. 571–590, 2014.

  En este artículo [39] se presenta un mecanismo de bootstrapping basado en la reputación capaz de establecer un valor inicial de confianza a las entidades que se unen a una red colaborativa de detección de intrusiones. Este trabajo se centra en los IDSs de la infraestructura, en aquellas unidades de detección móviles que se desplazan entre dominios de seguridad y/o administrativos y entre los dominios de seguridad que desean colaborar entre sí para crear una base de conocimiento común con las alertas detectadas individualmente.

A continuación se presenta la publicación con la que se pretende aportar una posible solución al cuarto y último de los subobjetivos planteados, el relacionado con el diseño y su puesta en marcha de un esquema adaptativo que le permita al CAS reconfigurar dinámicamente los IDSs de la infraestructura según sus comportamientos.

- M. Gil Pérez, J. E. Tapiador, J. A. Clark, G. Martínez Pérez, A. F. Skarmeta Gómez. **Trustworthy placements: Improving quality and resilience in collaborative attack detection**. Revista *Computer Networks*, vol. 58, pp. 70–86, 2014.

  En este artículo [40] se presenta el diseño de un modelo adaptativo donde todos los sensores de un sistema colaborativo de alertas, como parte básica de cualquier unidad de detección, son configurados y posteriormente instalados siguiendo un modelo de despliegue óptimo. La búsqueda de cuál es la mejor configuración de monitorización se convierte entonces en un problema de optimización, donde la diversidad de la confianza entre los distintos IDSs es maximizada a fin de obtener una mejor evidencia a la hora de evaluar las alertas generadas por estos IDSs.

Además de las publicaciones que se han listado más arriba, algunos de los resultados que se han obtenido a partir de esta tesis doctoral han contribuido al desarrollo de ciertas soluciones, las cuales han sido identificadas como hitos en los proyectos de investigación que se listan a continuación. Estos proyectos de investigación se enmarcan en ámbitos tanto regionales, nacionales como europeos.

- **SEINIT**: *Security Expert INITiative* [42].

  Financiado por la Comisión Europea bajo el contrato EU-IST FP6, IST-2002-2.3.1.5. Investigador principal: Antonio F. Gómez Skarmeta.

- **DAIDALOS**: *Designing Advanced network Interfaces for the Delivery and Administration of Location independent, Optimised personal Services* [43].

  Financiado por la Comisión Europea bajo los contratos EU-IST FP6, IST-2002-506997 e IST-2005-026943. Investigador principal: Antonio F. Gómez Skarmeta.

- **DESEREC**: *DEpendability and Security by Enhanced REConfigurability* [44].

  Financiado por la Comisión Europea bajo el contrato EU-IST FP6, IST-2004-026600. Investigador principal: Antonio F. Gómez Skarmeta.

- **MISTRAL**: *Middleware de gestión de Identidades de Seguridad en TRansacciones electrónicAs basado en código Libre.*

  Financiado por la Consejería de Educación, Ciencia e Investigación y Cultura de la Región de Murcia bajo el contrato TIC-INF 07/01-0003. Investigador principal: Gregorio Martínez Pérez.

- **SEMIRAMIS**: *SEcure Management of InfoRmation Across MultIple Stakeholders.*

  Financiado por la Comisión Europea bajo el contrato EU-IST FP7, CIP-ICT PSP-2009-3. Investigadores principales: Antonio F. Gómez Skarmeta y Gregorio Martínez Pérez.

- **RECLAMO**: *Red de sistemas de Engaño virtuales y CoLaborativos basados en sistemas Autónomos de respuesta a intrusiones y Modelos de cOnfianza* [45].

  Financiado por el Ministerio de Ciencia e Innovación de España (MICINN), bajo el contrato TIN2011-28287-C02-02, y la Comisión Europea (FEDER/ERDF). Investigador principal: Gregorio Martínez Pérez.

Capítulo 2

Estudio de los sistemas de confianza en entornos multidominio

Este capítulo sirve como revisión crítica del estado del arte actual de las tecnologías y arquitecturas sobre la gestión de la seguridad y la confianza en sistemas colaborativos orientados a la detección de ataques distribuidos, intentando evidenciar sus principales deficiencias a las que se les tratará de dar solución en los capítulos subsecuentes. Este análisis se centra en los dos mecanismos introducidos en el capítulo anterior, afrontando cada uno los objetivos asociados a los cuatro grandes bloques definidos en la Sección 1.5:

- Tecnologías basadas en criptografía de clave pública que ofrecen cierta protección frente a alteraciones de información (*confidencialidad* e *integridad*) de las alertas que los IDSs de un sistema colaborativo para la detección de ataques distribuidos necesitan intercambiar de forma segura, así como a la identificación de esos IDSs como unidades legítimas de detección del sistema colaborativo (*autenticación*).

- Sistemas de gestión de la confianza basados en reputación destinados a modelar el comportamiento de una entidad, ya sean IDSs o dominios de seguridad, tomando una especial atención la identificación de actitudes maliciosas de estas entidades en entornos multidominio. Este análisis incluye el estudio de propuestas sobre el cálculo inicial de la confianza de una nueva entidad que desea unirse al sistema colaborativo para la detección de ataques, ya sean IDSs particulares o dominios de seguridad para hacer posible la detección de ataques distribuidos.

Durante la exploración del estado del arte se hará referencia a los esquemas actuales más importantes para la detección de ataques distribuidos. La definición del modelo de despliegue de los IDSs será un factor clave en la escalabilidad, robustez, cobertura de la detección y rendimiento del sistema colaborativo. A este respecto, se presentará un último análisis sobre las propuestas actuales que pretenden maximizar la cobertura de la detección, así como la calidad de la información (sobre todo alertas) que se obtiene del sistema de monitorización, utilizando mecanismos que permitan el despliegue y/o reconfiguración de los IDSs de la infraestructura.

La metodología en este capítulo se centra en revisar, en primer lugar, los principales retos ante los que se enfrentan los sistemas colaborativos orientados a la detección de ataques distribuidos. A continuación se analizan las tecnologías actuales basadas en criptografía de clave pública, centradas en la validación de las credenciales de los IDSs (sus certificados X.509) para que éstos puedan realizar un intercambio seguro de sus alertas en escenarios multidominio. Un análisis posterior se centra en revisar las propuestas presentes en la literatura actual sobre cómo distribuir los IDSs en una red de detección. Sobre la base de este modelo de despliegue, se presenta un análisis crítico de los modelos actuales sobre la gestión de la confianza basados en reputación, incluyendo un estudio concreto de las propuestas centradas en el cálculo de la confianza inicial de una nueva unidad de detección cuando ésta desea unirse a un sistema para potenciar sus procesos de detección. Finalmente, se analizan las propuestas sobre cómo poder maximizar la cobertura y precisión en la detección de los ataques distribuidos.

2.1. Retos en la detección colaborativa de ataques

Los sistemas basados en IDS necesitan de un modelo de colaboración a partir del cual puedan intercambiar, con la mayor seguridad posible, toda la información que necesitan los procesos para la detección de intrusiones o ataques [46]. En este contexto, cabe realizar una mención especial a los entornos distribuidos, donde la información de detección, principalmente las alertas de seguridad, se tiene que intercambiar entre más de un dominio de seguridad y/o administrativo. Además de tener en cuenta las amenazas y vulnerabilidades que puede sufrir un sistema desplegado bajo un modelo de colaboración, otros retos también se suman a los desafíos anteriores y que están relacionados con la seguridad: confidencialidad, integridad y autenticidad.

Esta sección presenta un minucioso análisis de los retos y amenazas ante los que se enfrenta un sistema colaborativo para la detección de ataques. Muchos ya han sido estudiados en diversos trabajos de investigación, pero existen otros en los que todavía queda trabajo hasta conseguir IDSs más confiables en sus procesos de detección.

2.1.1. Reducción del número de falsas alarmas

Uno de los retos más importantes que los IDSs actuales están tratando como más prioritario es la precisión en los procesos de detección. Esta precisión dependerá, en gran medida, de la tasa de falsas alarmas que genera cada IDS por separado, las cuales se clasifican en dos grupos distintos [47]:

- *Falso positivo*, o errores de Tipo I. Alarma que genera un IDS a consecuencia de haber detectado, presuntamente, un incidente en el sistema, cuando el hecho que lo hubiera provocado no ha ocurrido realmente. Este error suele darse por un fallo en los algoritmos de detección que implementan los IDSs, pero también si intentan alertar sobre actividades sospechosas sin tener en cuenta lo que está ocurriendo en otras áreas desde un punto de vista más global del sistema.

- *Falso negativo*, o errores de Tipo II. Corresponde con la ausencia de una alarma por parte de un IDS cuando realmente ha ocurrido un incidente en el sistema, y el IDS estaba configurado correctamente para su detección.

Ambos tipos de falsas alarmas producen fallos graves en la detección por parte de los IDSs: alertando cuando no deberían o no haciéndolo cuando sí que lo tendrían que haber hecho. Por un lado, los falsos positivos ocasionan múltiples errores a los IDSs en sus propósitos de detección. Primero, provocan serios problemas de rendimiento para un IDS, tanto de cómputo al tener que analizar un alto número de alarmas, que realmente no representan la ocurrencia de un ataque, como de sobrecarga en las comunicaciones si varios IDSs forman parte de un sistema colaborativo para la detección de ataques distribuidos. En segundo lugar, en los últimos años se está observando un aumento en la generación masiva de falsos positivos producidos deliberadamente por los atacantes. El objetivo es inyectar *ruido* en el sistema para captar la atención de los administradores mientras ejecutan sus actividades maliciosas sin ser detectados.

A este último tipo de ataque se le conoce como *ataque por camuflaje*. Como solución, en [48] se propone mantener un conjunto de *estimaciones*, modeladas según una serie de características (atributos) que pueden extraerse de cada alarma, sobre cada entidad del sistema como si fuera un atacante, en lugar de guardar el inmenso número de alarmas que son capaces de generar. Esas estimaciones se actualizan en [48], entre otras posibles aproximaciones, mediante *estadística bayesiana*, a fin de identificar probabilísticamente si las alarmas corresponden con un ataque real o, por el contrario, están enmascarando otro ataque más importante generando una gran cantidad de falsos positivos.

Por otro lado, los falsos negativos suelen aparecer por la ejecución de un *ataque por evasión* [49], donde los atacantes explotan ciertas debilidades en las tecnologías de detección que implementan los IDSs. Los ataques por evasión fueron popularizados en 1998 por Ptacek y Newsham [50], en cuya publicación presentaban varias técnicas de evasión, entre las que destacaban la fragmentación de los ataques en pequeños paquetes de red que pasaban inadvertidos para los cuatro *Sistemas de Detección de Intrusiones basados en Red* (del inglés Network-based Intrusion Detection System, NIDS) más importantes de la época. La víctima reensamblaba posteriormente todos los fragmentos y ejecutaba el contenido del ataque en su sistema local. Una solución relativamente sencilla para la detección de ataques por evasión basados en fragmentación es hacer que los IDSs sean capaces de poder reensamblar los flujos de una comunicación, al igual que hacen los destinatarios de esos fragmentos, aunque esto implique una mayor carga computacional y de memoria para los IDSs. Como ejemplo de implementación, Snort ofrece un preprocesador llamado *stream5* capaz de reensamblar los paquetes de red fragmentados para mantener el estado de las conexiones activas [51].

Además de la fragmentación, también existen otras técnicas de evasión distintas que los IDSs deben tener en cuenta para la detección de estos ataques. Se pueden destacar la ofuscación del contenido de los ataques, donde los *ataques polimórficos* –ligera mutación de los ataques– son el ejemplo más ilustrativo [52, 53], y la completa desactivación de los IDSs para deshabilitar sus funciones de monitorización y/o detección.

En el último caso, los atacantes suelen seguir dos estrategias distintas: aprovecharse de un *bug* que termine con el proceso que ejecuta el propio IDS, como la definida por la vulnerabilidad CVE-2009-3641 publicada en [54] sobre Snort, o ejecutar un *ataque DoS* para sobrecargar al IDS (por ejemplo, enviándole una gran cantidad de falsas alarmas) mientras que el atacante lleva a cabo el ataque real. Ejemplos de trabajos donde traten la detección de ataques DoS pueden verse en [3], donde se propone calcular el ratio de flujos TCP para la identificación de paquetes maliciosos, y también en [55], donde se presenta la arquitectura *Security Gateway System* (SGS) que permite ejecutar los IDSs en redes de alta velocidad, con tráfico de hasta 2 Gbps, mediante un controlador reconfigurable implementando un mecanismo de doble token.

Debido a los problemas comentados anteriormente, los administradores empiezan a no confiar en las capacidades de detección de sus IDSs, o del sistema colaborativo de detección de intrusiones en su conjunto. Como consecuencia, muchos administradores se están percatando que son necesarios el uso y despliegue en sus redes de elementos complementarios que ayuden al correcto funcionamiento de este tipo de sistemas.

En conclusión, y a raíz del análisis anterior, se puede afirmar que la identificación y eliminación de falsas alarmas, fundamentalmente los falsos positivos, es un componente crítico para la correcta detección de ataques. Muchas propuestas que existen en la literatura actual sobre la reducción del alto número de falsos positivos se centran, en su mayoría, en el postprocesamiento de las alarmas. Por ejemplo, en [56] se propone un filtro compuesto por tres componentes donde se analiza la distribución de los falsos positivos en conjuntos similares de alarmas muy cercanos en el tiempo, obteniendo una reducción del 73,98 % sobre el total de falsos positivos que realmente se habían generado en el sistema. Otros trabajos, como los que se presentan en [12] y en [57], proponen que se obtenga una descripción única y abstracta de los posibles ataques que se estén produciendo en el sistema, mediante técnicas de *clustering*, sobre las alarmas generadas por múltiples IDSs. Aplicando esta técnica, ambos trabajos obtienen, respectivamente, una reducción media de falsos positivos de hasta un 80,3 % y un 87,5 %.

Las pruebas experimentales que se llevaron a cabo en los tres últimos trabajos se realizaron utilizando la base de datos DARPA 1999 [58]. El conjunto de paquetes de red que ofrece DARPA se considera un referente de base para la evaluación en rendimiento de las propuestas basadas en IDS, el cual puede descargarse en [59].

2.1.2. Incapacidad para detectar nuevos ataques

Las implementaciones de los IDSs actuales se basan en dos técnicas distintas para la detección de intrusiones o ataques:

- *Detección de anomalías* (del inglés Anomaly Detection). Los IDSs construyen un modelo de comportamiento, habitualmente las acciones que realizan los usuarios, durante un tiempo de entrenamiento sobre qué es lo "normal" dentro del sistema que van a monitorizar posteriormente [60]. De esta manera, cualquier acción que se salga de ese conjunto normal en el comportamiento se considera como posible actividad sospechosa, lanzándose la correspondiente alerta para su análisis.

- *Detección de usos indebidos* (del inglés Misuse Detection). Esta técnica actúa en base a patrones bien conocidos para la detección, llamados *firmas de ataque*, donde se especifican las acciones maliciosas que podrían llevar a cabo los atacantes en la ejecución de sus actos ilícitos. Los IDSs compararán posteriormente todos los datos recibidos de sus fuentes de información (*sensores*) contra esas firmas de ataque para la detección de alguna anomalía conocida a priori [61].

De entre las dos técnicas anteriores, la detección de usos indebidos es la solución más precisa cuando se desean detectar ataques conocidos, aunque es ineficiente frente a ataques desconocidos, o que acaban de ser descubiertos, ya que los IDSs todavía no tendrían configurada la última actualización de las firmas de ataque. Entre estos nuevos ataques destacan los *ataques de día-cero* (del inglés Zero-day Attack) que, básicamente, son *exploits* de tan reciente aparición que todavía no se han creado y distribuido las firmas de ataque necesarias para su detección [62]. Como ejemplo, Microsoft publicó en mayo de 2013 un aviso de seguridad sobre un ataque de día-cero en Internet Explorer 8, en cualquier versión de sus sistemas operativos [63]. Esta nueva vulnerabilidad, a la que se le ha asignado el identificador CVE-2013-1347 [64], permite que la consecución de este nuevo tipo de ataque de día-cero le otorgue a una persona que no está autorizada la posibilidad de ejecutar código arbitrario de manera remota. Microsoft publicó este aviso de seguridad el 3 de mayo de 2013, días después de que la vulnerabilidad fuera descubierta mientras que un conjunto de atacantes la estaba aprovechando a su antojo. Hasta el 9 de mayo, Microsoft no publicó en su Web el primer *parche* que daba solución a esta vulnerabilidad, pero hasta el 14 de mayo no se hicieron públicos los boletines de seguridad oficiales. En todo ese tiempo de retraso, 11 días desde su detección, cualquier ordenador que tuviera instalado Internet Explorer 8 se encontraba a expensas de que un atacante pudiera ejecutar cualquier código en su sistema.

Frente a este nuevo tipo de ataque, en [65] se propone una solución para cuantificar el riesgo en la seguridad ante la ejecución de ataques de día-cero. La propuesta para el cálculo de ese riesgo se centra en una métrica de seguridad con la que medir el número de vulnerabilidades distintas que serían necesarias para comprometer un sistema. De esta manera, la comprobación de que existe un alto número de esas vulnerabilidades indicaría que el sistema es más seguro, ya que la probabilidad de sufrir más ataques de este tipo, a la misma vez y por el mismo atacante, será inferior.

Por otro lado, la detección de anomalías sí que es capaz de detectar nuevos tipos de ataque, sin la ayuda externa por parte de administradores o actualizaciones de los procesos de detección de los IDSs. A pesar de ello, la detección de anomalías sufre serios problemas durante el proceso de entrenamiento en saber cuál es el comportamiento normal del sistema y de sus usuarios, además de que es necesario que esa fase se tenga que hacer en un entorno totalmente libre de ataques, difícil de contemplar en un entorno real en producción. Además, las modificaciones en el comportamiento de los usuarios también suponen un gran problema de adaptación a los IDSs basados en la detección de anomalías, lo cual supondrá la generación de grandes cantidades de falsos positivos hasta que se modelen correctamente sus nuevos patrones de comportamiento.

Debido a los problemas anteriores sobre la detección de anomalías, especialmente los relacionados con la generación de falsos positivos, han aparecido otras soluciones que intentan resolver el problema planteado. En [66], por ejemplo, se propone un enfoque de *detección de anomalías basada en especificaciones* (del inglés Specification-based Anomaly Detection) donde el comportamiento "normal" de los componentes críticos del sistema se modela manualmente, sin hacer uso de técnicas de aprendizaje –proceso de entrenamiento– como hacen los sistemas basados en la detección de anomalías.

2.1.3. Gestión de gran cantidad de información en tiempo real

Las capacidades que un IDS debe tener para monitorizar un sistema de información implica la adopción de "buenos" algoritmos de detección. Sobre todo, algoritmos con grandes y precisas capacidades de detección, y un rápido procesamiento de los datos que obtiene de las fuentes de información; principalmente, paquetes de red. Sin estas capacidades, los IDSs no podrán realizar sus funciones de monitorización y análisis en tiempo real, haciendo casi imposible la detección de los ataques.

Este problema se debe a que las redes actuales proporcionan tasas de transmisión cada vez más altas, pasando en muy pocos años de los 100 Mbps a los 10 Gbps actuales. Esto hace que la gran cantidad de información que fluye por una red sea enorme para que un IDS sea capaz de recuperar y analizar cada paquete de red. Por ejemplo, Snort funciona correctamente en redes de hasta 1 Gbps, pero comienza a sufrir problemas a partir de 1,5 Gbps cuando empieza a descartar paquetes de red por sobrecarga [67]. En concreto, la tasa de pérdida de paquetes es del 1,15 % en redes con una velocidad de transmisión de 2 Gbps. Sin embargo, hay trabajos que realizan modificaciones a Snort para alcanzar velocidades más altas sin pérdida de paquetes de red. Por ejemplo, Gnort es una solución basada en Snort donde las operaciones con un alto coste computacional, como la comprobación de coincidencia con un patrón de detección, son ejecutadas por la tarjeta gráfica en lugar de por la CPU [68]. Esta solución consigue que la adaptación de Snort funcione correctamente sobre redes con velocidades de 2,3 Gbps.

Para seguir mejorando el rendimiento en el análisis de la información, la mayoría de trabajos se centran en técnicas avanzadas de paralelización sobre tecnologías hardware. Concretamente, en [69] se presenta un estudio completo de las ventajas e inconvenientes de varias técnicas de paralelización, como las basadas en FPGA (Field Programmable Gate Array), soportando velocidades de hasta 4 Gbps sin pérdida [70], o las basadas en ASIC (Application-Specific Integrated Circuit), alcanzando velocidades cercanas a 7,2 Gbps [71]. Aunque estas soluciones proporcionan mejoras sustanciales, también son enfoques basados en hardware cuyos costes de implantación son más elevados.

Todas las soluciones anteriores van a ser insuficientes en un futuro cercano, ya que se vaticinan redes con velocidades de transmisión todavía más elevadas. En concreto, existe un proyecto en marcha llamado *Advanced Networking Initiative*, financiado por el Departamento de Energía (DOE) del gobierno de Estados Unidos, con el que pretenden alcanzar velocidades de 1 Tbps, teniendo ya en la actualidad cuatro laboratorios del DOE conectados bajo una red de comunicaciones de 100 Gbps [72].

2.1.4. Escalabilidad y robustez al tener que analizar grandes volúmenes de información

Como se ha comentado en el punto anterior, la gran cantidad de información que los IDSs deben analizar se convierte en un problema casi intratable. La solución más factible pasa por la instalación de múltiples IDSs a lo largo del sistema para que cada uno se encargue de analizar las políticas de seguridad de una "pequeña" parte de la red de detección [73]. De esta manera, además de evitar una sobrecarga de información a cada IDS de manera individual, también se ofrece un mecanismo que proporciona robustez al sistema. Es decir, si alguno de los IDSs deja de funcionar, o expone incluso un comportamiento anómalo o malicioso (por ejemplo, publicando falsas alarmas sobre incidentes que no han ocurrido en la realidad), todavía existen otros IDSs en la red de detección que seguirían monitorizando esa zona del sistema.

En [6] se puede encontrar un análisis detallado sobre cómo se tienen que desplegar los IDSs en un sistema de detección para eludir problemas en la escalabilidad y en la robustez. A este respecto, en la Sección 2.3 se presenta un estudio en profundidad sobre los distintos criterios y propuestas que plantean las actuales arquitecturas colaborativas para la detección de ataques distribuidos, las cuales identifican ambos problemas como dos de los retos más prioritarios en esta temática.

2.1.5. Incremento de ataques de denegación de servicios

El comportamiento de los atacantes está cambiando en los últimos años para evitar que sus acciones sean detectadas por los sistemas de detección actuales, pasando de un modo de actuación centrado en un único punto de origen, desde la propia máquina del atacante, a un nuevo enfoque donde el ataque se lanza desde múltiples puntos de la red. Estos atacantes suelen utilizar *lanzaderas* –sistemas comprometidos con anterioridad– para lanzar un ataque orquestado hacia un objetivo común, creando para ello una red *botnet* con cientos o miles robots (o lanzaderas) que el atacante puede controlar de forma remota [74, 75]. El objetivo principal de estos ataques es la ejecución de un ataque de *Denegación Distribuida de Servicio* (del inglés Distributed Denial of Service, DDoS), donde el atacante intenta deshabilitar algún servicio crítico del sistema objetivo inundando su red de tráfico o sobrecargando sus recursos computacionales [76].

La detección de ataques DDoS necesita un compromiso global por parte de todos los dominios que forman Internet, ya que, en caso contrario, esta tarea se convierte en una misión altamente compleja. Habrían áreas inseguras que podrían ser comprometidas y utilizadas posteriormente como lanzaderas de este tipo de ataques a gran escala. Con respecto a este compromiso, existen trabajos que proponen enfoques de filtrado de los paquetes de red sospechosos de un ataque DDoS en el propio núcleo de Internet [77], o estableciendo métricas de seguridad en las conexiones troncales de Internet (a nivel de *backbone*) que detecten posibles ataques DDoS según el volumen de tráfico [18]. Sin embargo, estas soluciones implican un compromiso a nivel de Internet en su conjunto, algo difícil de considerar en un mundo globalizado como el actual.

Las redes botnet han sido identificadas por Joseph Demarest, director de la ciberdivisión del FBI, como una de las mayores amenazas actuales de Internet. Ante un Comité del Senado de Estados Unidos, Demarest ha afirmado que "cada segundo, 18 ordenadores pasan a formar parte de una red botnet" [78], lo que se traduce en más 500 millones de equipos comprometidos por año, estimándose unas pérdidas económicas que podrían rondar los 110 000 millones de dólares a nivel mundial.

Como solución, en [79] se estudian diversas técnicas para detectar y desmantelar redes botnet, donde caben destacar las técnicas basadas en la detección de solicitudes a servidores DNS (Domain Name System), para localizar el *servidor C&C* (del inglés Command & Control), y las basadas en *minería de datos* (del inglés Data Mining), cuyo objetivo es el análisis del tráfico de red para detectar *canales C&C*. Los canales C&C, proporcionados por un servidor C&C central, son utilizados por el atacante para dirigir de forma remota todos los robots ante la proyección de su ataque orquestado.

A la lista anterior de técnicas destinadas a la detección de redes botnet, también hay que incorporar otros trabajos, como el presentado en [80], que basan esa detección en técnicas de *aprendizaje automático* (del inglés Machine Learning). En este punto, también caben destacar otros esfuerzos relacionados con la detección de canales C&C. En concreto, en [81, 82] se propone utilizar el tráfico de red saliente en la detección de ataques, en lugar del tráfico entrante como hacen los IDSs. A este nuevo enfoque lo denominan *Sistemas de Detección de Extrusiones* (del inglés Extrusion Detection System, EDS). Con respecto a la detección de redes botnet, el objetivo de los EDSs se centra en la detección de canales C&C cifrados inspeccionando todo el tráfico saliente desde las propias máquinas del sistema. Además, también detectan la construcción de redes botnet mediante el análisis de métodos de propagación que intentan comprometer nuevos sistemas a través de, principalmente, el correo electrónico.

2.1.6. Seguridad de las herramientas de detección

Como cualquier otro tipo de software, una herramienta de seguridad como un IDS también debe ser seguro en sí mismo para que sea confiable. Por un lado, los IDSs deben ser robustos frente a la aparición de bugs en su implementación, como la vulnerabilidad CVE-2009-3641 sobre Snort comentada más arriba [54], y, por otro lado, deben hacer uso de protocolos seguros que protejan el contenido de la información transmitida por la red. Este último caso es especialmente relevante en entornos colaborativos, donde los IDSs que forman una red colaborativa de detección de intrusiones, siendo éstas redes autónomas de un sistema colaborativo de alertas, necesitan intercambiar información de seguridad para la detección de ataques distribuidos. En este sentido, la mayoría de soluciones proponen el uso de criptografía de clave pública, tanto en la creación de los canales seguros como en la autenticación de los IDSs para demostrar que son entidades legítimas del sistema [83]. Sin embargo, todas estas soluciones aplican conceptos de PKI en entornos totalmente intradominio, dentro de un mismo marco organizativo, sin hacer ninguna referencia a cómo tendrían que ser implantados en sistemas de confianza complejos como, por ejemplo, en escenarios de certificación cruzada.

Además de lo expuesto anteriormente, en entornos colaborativos también es de vital importancia que la información emitida por cada uno de los IDSs (alertas) sea veraz y que corresponda a hechos que hayan ocurrido realmente. Esa veracidad, o precisión en los datos de entrada, se puede modelar mediante la evaluación de la confianza que el IDS receptor tiene en el que envió la información. Si considera que el IDS emisor no tiene un mínimo nivel de confianza, el IDS receptor descartará la información al considerarla como falsa, ya que parece que proviene de un *presunto* IDS malicioso. En la literatura existen muchos trabajos que proponen la construcción y evaluación de redes de confianza entre los participantes de un sistema colaborativo [84, 85], aunque muy pocos son los que centran sus esfuerzos en evaluar la confianza de los IDSs en una red de detección de ataques. En la Sección 2.4.2 se profundizará en aquellas soluciones que permiten evaluar la confianza de los IDSs emisores de alertas, siempre en el contexto de un sistema colaborativo para la detección de ataques distribuidos.

2.1.7. Interpretación de los datos cifrados

Gran cantidad de la información que fluye por las redes actuales aparece de forma cifrada para evitar ataques contra la confidencialidad e integridad de su contenido. De la misma manera, un atacante también podría cifrar el contenido de sus ataques para eludir que sean detectados por los NIDSs que el sistema objetivo tuviera desplegados. Los NIDSs no son capaces de procesar información cifrada, a menos que tengan la clave de descifrado correspondiente, algo totalmente inviable. Este es el caso que ocurre con los canales C&C cifrados que se han analizado anteriormente.

La instalación en cada una de las máquinas de la red de un *Sistema de Detección de Intrusiones basado en Host* (del inglés Host-based Intrusion Detection System, HIDS) puede ayudar en el análisis de este tipo de información cifrada. Los HIDSs son los únicos componentes de un sistema de detección con acceso a los datos descifrados para su evaluación, ya que están instalados en el mismo sistema donde se ejecutará el algoritmo de descifrado con la clave necesaria. Por ejemplo, en [86] se presenta la implementación de un módulo para los HIDSs que permite el análisis de datos procedentes de canales cifrados para un servidor Web Apache. Sin embargo, las soluciones basadas en HIDSs tienen una percepción totalmente local de los ataques, no pudiendo ejecutar *procesos de correlación* sobre eventos acaecidos en varios sistemas que podrían indicar la ejecución de un ataque más global [87]; por ejemplo, un ataque distribuido a gran escala.

Como solución al acotado ámbito en la detección que tienen los HIDSs, surgieron otros trabajos, como [88], donde el tráfico de red que cada sistema descifra es reenviado a un NIDS para un análisis en profundidad. De todas maneras, esta solución no impide que el sistema sea comprometido por un atacante, pudiendo nutrir al NIDS con datos falsos, o bloqueando esa comunicación, y tampoco evita la pérdida de confidencialidad de la información cifrada ya que ésta se envía en claro entre el sistema final y el NIDS. A este respecto, en [89] se propone el uso de un algoritmo criptográfico basado en el esquema de compartición de secretos de Shamir [90], sin que la confidencialidad de la información transmitida por los canales seguros se vea comprometida.

2.1.8. Incremento de herramientas automáticas de ataque

Es muy habitual encontrar en Internet herramientas automáticas de ataque que los usuarios pueden descargar y ejecutar sin ningún conocimiento previo de qué hacen y, por supuesto, cuáles son sus consecuencias reales. La empresa Imperva, dedicada a la seguridad para la protección de datos en grandes empresas, publicó un informe en abril de 2012 que revelaba que el 98 % de los ataques por inclusión remota de archivos –afecta a páginas PHP– y que el 88 % de los ataques por inyección de código SQL provenían de herramientas automáticas [91]. Las "ventajas" que tienen los administradores frente a este tipo de amenaza es que también pueden descargar esas herramientas y configurar sus sistemas para que sean capaces de detectarlas. Estas herramientas se caracterizan por generar siempre el mismo tráfico de red y tener el mismo comportamiento en las víctimas: accesos a memoria, llamadas al sistema y procesos que ejecutan.

Como solución, se pueden utilizar técnicas basadas en la detección de *escenarios de ataque*, que permiten rastrear la ejecución de los pasos realizados por esas herramientas automáticas [92]. Esta ventaja también supone, a su vez, un gran inconveniente para los administradores, como es el tener que conocer a la perfección todas las herramientas de ataque que puedan aparecer. Además, el sistema siempre será vulnerable durante un tiempo: desde que la herramienta está disponible en Internet, el administrador conoce de su existencia, si llega a conocerla, la descarga y estudia cómo puede caracterizarla para que el sistema sea capaz luego de detectarla. Como ejemplo, en febrero de 2014 surgió una herramienta de ataque que fue utilizada para comprometer un gran número (indeterminado) de IPs en China. Esta herramienta utilizaba métodos de fuerza bruta sobre el servicio SSH para acceder a cada víctima como usuario *root* [93].

2.1.9. Reacción rápida ante los incidentes ocurridos

A pesar de que las capacidades de reacción no forman parte de las soluciones basadas en IDS, esta característica está siendo ampliamente investigada para que los sistemas sean totalmente autónomos frente a los incidentes ocurridos en el sistema. Los IDSs son componentes pasivos cuya reacción más habitual es la notificación de incidentes a otros elementos como, por ejemplo, administradores que se encarguen de tomar las acciones que consideren más oportunas. Como solución, aparecieron los *Sistemas Autónomos de Respuesta a Intrusiones* (del inglés Automated Intrusion Response Systems, AIRS) con el objetivo de aplicar los mecanismos de respuesta más adecuados según los incidentes detectados por los IDSs, sin la intervención de un administrador [94, 95].

Entre los primeros AIRSs que plantean el uso de sistemas de detección y respuesta frente a intrusiones están CSM (Cooperating Security Managers) [96] y EMERALD (Event Monitoring Enabling Responses to Anomalous Live Disturbances) [97]. CSM está basado en host, con respuestas locales en el propio sistema de computación, y EMERALD en sistemas distribuidos. En esa misma línea apareció AAIRS (Adaptive Agent-based Intrusion Response System), donde se plantea el uso de agentes software en una metodología adaptativa de las respuestas frente a intrusiones [98].

En la última década, y a raíz del nuevo paradigma de ataque mucho más distribuido, las soluciones basadas en AIRS también se han adaptado debidamente a esos cambios. Entre estas soluciones, destacar la aparición de ADEPTS (Adaptive Intrusion Tolerant Systems) [99] y de RRE (Response and Recovery Engine) [100]. ADEPTS modela las intrusiones como *grafos de ataque*, junto con la respuesta más apropiada si el atacante ejecuta los pasos definidos en uno de esos grafos. Por otro lado, RRE propone el uso de *árboles de respuesta*, basados en un modelo escalable de espacio-estado, para encontrar el conjunto más óptimo de respuestas siguiendo una evaluación paso a paso.

Las nuevas capacidades de reacción que proponen las soluciones anteriores son muy deseables para los nuevos sistemas autónomos, pero a su vez también conllevan ciertas consecuencias muy graves mientras que no se pueda mejorar la precisión en los procesos de detección. Una modificación en la configuración del sistema debido a un error en los procesos de detección podría tener consecuencias muy negativas [101].

2.2. Intercambio seguro de alertas de detección con Infraestructuras de Clave Pública

De entre la lista de retos presentados en la Sección 2.1, la seguridad de los IDSs como herramientas de detección se ha pronunciado como un desafío al que se le tiene que dar solución, proporcionando mecanismos con los que proteger las alertas que los IDSs necesitan intercambiar para la detección de ataques. Éstos tienen que ofrecer un marco de seguridad bajo el que los IDSs de un sistema colaborativo para la detección de ataques distribuidos puedan intercambiar sus alertas de forma segura: autenticando a los IDSs como entidades legítimas del sistema y protegiendo sus comunicaciones frente a alteraciones de las alertas de detección durante su transmisión por la red.

Hasta hace unos años, la gestión de la seguridad en una organización era una tarea relativamente sencilla. La instalación de una Infraestructura de Clave Pública (PKI) era un requisito más que suficiente, en el que la *Autoridad de Certificación* (del inglés Certification Authority, CA) tenía un control absoluto de la seguridad sobre todas las entidades de sus unidades administrativas, incluyendo también a los IDSs como una entidad más a gestionar dentro de su dominio de seguridad. A pesar de todo ello, las organizaciones actuales requieren de cierto grado de *flexibilidad* en el establecimiento y revocación de relaciones de confianza con otras organizaciones.

En entornos colaborativos como los sistemas orientados a la detección de ataques distribuidos, es necesario el despliegue de nuevas soluciones basadas en PKI para que el intercambio de alertas entre distintos dominios de seguridad, cada uno implementando su propia infraestructura a nivel intradominio, se lleve a cabo bajo el marco de seguridad deseado en escenarios multidominio. A pesar de ello, el despliegue de nuevos modelos avanzados de certificación tiene un alto impacto, tanto en la *interoperabilidad* entre los dominios de seguridad como en el rendimiento cuando un IDS tiene que validar todo el material criptográfico (certificado X.509) de otro IDS, perteneciente a otro dominio de seguridad, antes de establecer una comunicación segura. Cómo desplegar un modelo

avanzado de certificación con el que dar soporte a la interoperabilidad, y cómo realizar la validación de certificados X.509 en un entorno multidominio, son dos desafíos ante los que las soluciones presentes en la literatura no ofrecen una solución viable.

La validación de un certificado X.509 (o simplemente *certificado*) la puede hacer el propio IDS, aunque lo habitual es que se lo delegue a una *Tercera Entidad de Confianza* (del inglés Trusted Third Party, TTP). Para ello, este TTP debe tener un *Servicio de Validación* que sea capaz de garantizar que el certificado a evaluar es confiable antes de iniciar el intercambio de alertas con el IDS poseedor del mismo. Este proceso es sencillo en modelos de certificación simples, ya que todo el material criptográfico está disponible en el mismo dominio de seguridad. En cambio, en los escenarios multidominio se hace necesaria la implantación de modelos avanzados de certificación al tener todo el material criptográfico distribuido entre múltiples dominios y/u organizaciones.

En entornos multidominio, es necesario que se tengan que procesar todos los caminos de certificación desde la fuente de validación hasta el objetivo deseado, pudiendo haber múltiples opciones de itinerario entre la PKI de origen y la de destino [102, 103]. De manera formal, un *camino de certificación* [104] es una "lista ordenada de certificados que comienza con un certificado cuya firma digital se puede verificar utilizando uno de los certificados en los que el usuario del servicio confía, y termina con el certificado que se desea validar". A estos certificados de confianza se les conoce comúnmente, del inglés, como *Trust Anchors* [105]. Nótese que los certificados de un camino de certificación en estos escenarios serán, normalmente, de PKIs de distintos dominios de seguridad.

Cualquier Servicio de Validación se compone de dos procesos independientes para la construcción y validación de caminos de certificación, siendo ambos complementarios para su ejecución [106]. Estos procesos son los siguientes:

- Construcción de uno o más caminos de certificación candidatos entre el certificado a validar y alguno de los Trust Anchors en los que el usuario del servicio confía, o en el sentido inverso entre los Trust Anchors y el certificado a validar. A este proceso se le denomina *construcción de caminos de certificación*.
- Comprobar que cada uno de los certificados en el camino de certificación es válido. Esta tarea debe examinar, entre otras muchas cosas, la correcta estructura de cada uno de los certificados, sus periodos de validez y sus estados de revocación. A este proceso se le denomina *validación de caminos de certificación*.

En las siguientes secciones se estudian en profundidad los dos procesos anteriores. Además, también se analizan los principales modelos de certificación, especialmente los aplicables a entornos multidominio, ya que son los que mejor se adaptan a un sistema colaborativo de detección de ataques, planteado como objetivo en esta tesis doctoral.

2.2.1. Principales modelos de certificación

En esta sección se analiza el concepto de seguridad en entornos intradominio a través de la implantación de alguno de los modelos de certificación simples, y cómo puede extenderse esa seguridad a nuevos escenarios multidominio mucho más complejos.

En los escenarios multidominio, como los sistemas colaborativos para la detección de ataques distribuidos, los IDSs de varios dominios de seguridad van a poder autenticarse para intercambiar alertas de forma segura más allá de sus dominios locales.

1. Modelos simple y jerárquico

El modelo simple fue el primero en aparecer ya que sólo implica instalar una única CA, configurada como punto central de confianza para todo el dominio de seguridad. Esta CA emite certificados de entidades finales para todos sus usuarios, dispositivos y procesos software (como los IDSs) que necesiten de procedimientos seguros basados en criptografía asimétrica [17]. De entre las estructuras de certificación de la Figura 2.1, la entidad *Root* CA_4 (Figura 2.1a) implementa este modelo simple. Sin embargo, su mayor problema recae en la centralización de toda la confianza en un único punto, convirtiendo la clave privada de su CA en el foco de atacantes cuyo propósito sea comprometer todo el sistema. Además, la gestión de esa CA podría hacerse cada vez más compleja ya que, por ejemplo, su *Lista de Revocación de Certificados* (del inglés Certificate Revocation List, CRL) [24] podría ser demasiado grande para que sea descargada posteriormente por usuarios que están interesados en realizar una validación offline.

El modelo de certificación jerárquico aparece como una primera solución a la falta de escalabilidad que experimenta el modelo simple. En este nuevo modelo se establecen, de forma jerárquica, una serie de relaciones de confianza entre las distintas CAs que son parte de la misma unidad organizativa. En la práctica, cada una de esas CAs gestiona la confianza dentro de un mismo dominio de seguridad. En la Figura 2.1b se muestran varios ejemplos de modelos siguiendo un esquema de certificación jerárquico.

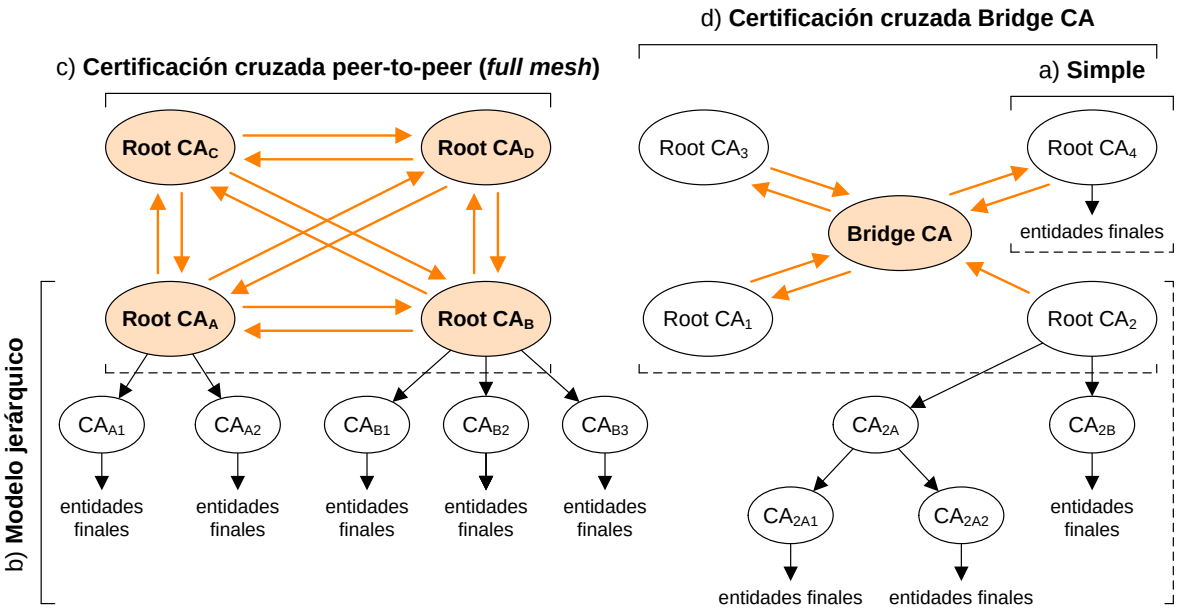


Figura 2.1: Principales modelos de certificación

En un modelo jerárquico existe una única CA raíz, cuyo certificado es autofirmado, que emite certificados a sus CAs subordinadas, creando así un *árbol de certificación*. De esta manera se establecen relaciones de confianza entre las CAs de forma unidireccional, no pudiendo haber un certificado emitido por una CA subordinada hacia la CA emisora. En este escenario, todas las CAs son capaces de emitir certificados de entidades finales, incluyendo los IDSs, aunque en la práctica solamente las CAs emisoras de los nodos hoja dentro del árbol de certificación emiten ese tipo de certificados.

Ambos modelos, tanto el simple como el jerárquico, permiten construir *sistemas de confianza intradominio*, donde todos los IDSs del mismo dominio de seguridad pueden confiar entre sí. Sin embargo, estos dos modelos no permiten establecer relaciones de confianza con otros dominios, para la creación de un *sistema de confianza interdominio*, con el objetivo de construir un sistema de confianza para que los IDSs puedan compartir sus alertas de forma segura, y sin tener que preocuparse del dominio de seguridad a los que pertenecen. En el siguiente apartado se estudian los dos modelos principales de certificación para el establecimiento de relaciones de confianza interdominio.

2. Modelos de certificación cruzada

En cualquier modelo de certificación cruzada, las relaciones de confianza entre las CAs de distintos dominios de seguridad se realizan mediante la generación de un nuevo tipo de certificado, llamado certificado cruzado. Un *certificado cruzado* sigue siendo un certificado X.509 de clave pública, pero emitido por la CA de un dominio de seguridad para otra CA perteneciente a otro dominio distinto al emisor [102], como el propósito último de establecer una relación de confianza entre ambos dominios.

Desde el punto de vista del estándar X.509, un certificado cruzado no difiere mucho de los certificados que ya se han definido en los modelos anteriores. La mayor diferencia radica en el conjunto de restricciones que deben incluir para reflejar los acuerdos de confianza establecidos entre los dominios de seguridad, definidas en las extensiones del certificado cruzado. No es necesario tener que modificar los certificados de ambas CAs para plasmar los acuerdos de confianza entre ambos dominios, siendo ésta una gran ventaja a la hora de revocar cualquier certificado cruzado para cesar esa relación de confianza. La seguridad en ambos dominios de seguridad se mantendrá por separado como se contemplaba antes de establecer esa relación de confianza.

La generación de un único certificado cruzado entre dos CAs establece una relación de confianza unidireccional, entre la CA que lo emite (por ejemplo, CA_A) y la otra CA con la que quiere relacionarse (por ejemplo, CA_B) [107]. Esta relación sería de la forma $CA_A \rightarrow CA_B$, lo cual no implica la existencia de la relación inversa $CA_A \leftarrow CA_B$. Para el establecimiento de una relación de confianza bidireccional, también conocida como *relación de confianza mutua*, se deben generar dos certificados cruzados distintos que establezcan una biyección entre las dos CAs. Por tanto, para una relación de confianza mutua $CA_A \leftrightarrow CA_B$ deben definirse tanto $CA_A \rightarrow CA_B$ como $CA_A \leftarrow CA_B$.

En la Figura 2.2 se muestra un ejemplo de modelo de certificación cruzada entre dos CAs (relación bidireccional), utilizando certificados cruzados para su implementación.

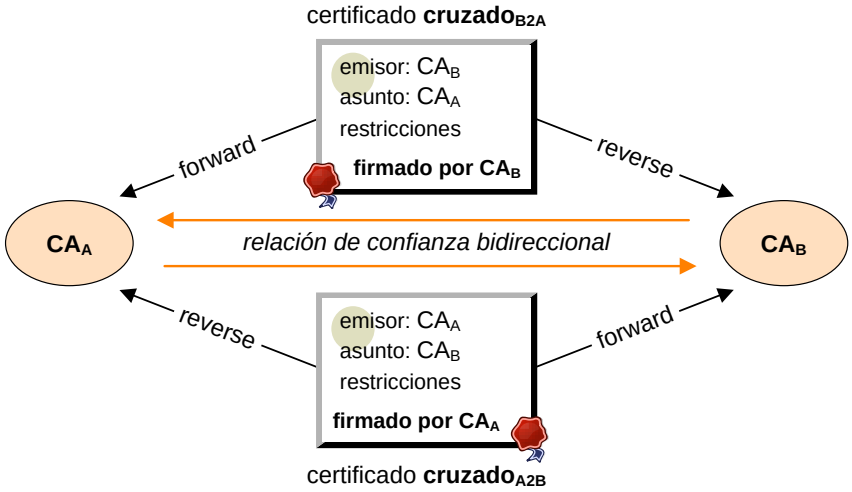


Figura 2.2: Relación de confianza mutua mediante dos certificados cruzados

Desde el punto de vista de la autoridad CA_A, existen dos certificados cruzados para el establecimiento de una relación de confianza mutua con CA_B:

- Un *certificado forward*, también conocido como `issuedToThisCA`. Representa la relación de confianza entre la CA emisora del certificado (CA_B) hacia la entidad beneficiaria del mismo (CA_A). En las extensiones de este certificado se incluirán las restricciones establecidas entre ambos dominios de seguridad para el manejo de la relación de confianza, siempre desde CA_B hacia CA_A.
- Un *certificado reverse*, también conocido como `issuedByThisCA`. Define, de forma análoga al anterior, una relación de confianza, pero ahora de CA_A hacia CA_B.

La agrupación de estos certificados cruzados permite la definición de una relación de confianza mutua entre CA_A y CA_B. Después de que el *par de certificados cruzados* sea emitido por las dos CAs, este par debe almacenarse en un *Servicio de Directorio* (por ejemplo, en un servidor LDAP [108, 109]), bajo el atributo `crossCertificatePair`, para que esta nueva relación de confianza mutua sea pública y accesible por el resto de entidades. Siguiendo el ejemplo anterior, el Servicio de Directorio de CA_A tendrá que almacenar el par de certificados cruzados $<cruzado_{B2A}, cruzado_{A2B}>$, siendo el orden <`forward,reverse`> como establece el estándar [25]. Desde el punto de vista de CA_B, la entrada `crossCertificatePair` en su Servicio de Directorio mantendrá esta misma información, pero de forma inversa: $<cruzado_{A2B}, cruzado_{B2A}>$.

Como se ha comentado antes, en una relación bidireccional la confianza entre ambas CAs es mutua bajo una serie de restricciones, las cuales definen la política bajo la cual una de las CAs confía o no en los certificados emitidos por la otra. De esta forma, cada certificado cruzado puede definir diferentes requisitos y restricciones de certificación en sus extensiones para cada sentido de la relación de confianza entre las dos CAs.

En los dos siguientes apartados se definen en detalle los dos modelos principales que existen en la actualidad sobre certificación cruzada.

Certificación cruzada peer-to-peer

El modelo de certificación cruzada peer-to-peer permite la definición de un modelo de confianza interdominio, donde se establecen relaciones de confianza entre dos CAs raíces autónomas. Estas CAs pueden implementar internamente cualquier otro modelo, como el simple o el jerárquico. En este escenario se suele desplegar una malla completa de certificados cruzados, conocido como modelo *full mesh*, donde se definen todas las posibles relaciones de confianza mutuas existentes entre todos los dominios de seguridad implicados. Un escenario típico de este modelo se puede ver en la Figura 2.1c.

El mayor inconveniente de este modelo es su problema de escalabilidad en entornos multidominio, al ser necesaria una malla completa de relaciones de confianza entre todos los dominios. Considerando un escenario con n CAs, una malla de confianza completa necesitaría $n(n-1)$ relaciones. Además, este modelo no es tan fácil de desarrollar si se compara con el jerárquico, ya que se necesitan múltiples relaciones de confianza con el resto de PKIs que integran la federación, además de que las alternativas para construir un camino de certificación se incrementarían exponencialmente.

Bridge CA

El modelo de certificación cruzada mediante una *Bridge CA* (BCA) fue propuesto para dar solución al problema de escalabilidad del modelo peer-to-peer [34]. Este nuevo modelo define las relaciones de confianza entre los dominios de seguridad a través de un punto neutral de confianza de alto nivel, la BCA, gestionada por una entidad imparcial sobre la que todos confían. Por ejemplo, la *FPKI Management Authority* administra la *Federal Bridge Certification Authority* (FBCA) para el gobierno de Estados Unidos. La FBCA comenzó a operar en 2001 [35, 110], creando un marco común de confianza transitiva entre cuatro grandes entidades gubernamentales de Estados Unidos: *National Aeronautics and Space Administration* (NASA), *Department of the Treasury* (DOT), *Department of Defense* (DoD) y *Department of Agriculture's National Finance Center* (USDA/NFC). En la actualidad, la *Higher Education Bridge Certification Authority* (HEBCA), desplegada bajo los dominios del *Research and Education Bridge Certificate Authority* (REBCA) [111], pretende establecer una relación de confianza interfederación con la FBCA. La HEBCA facilita las comunicaciones confiables entre instituciones de educación superior estadounidenses y diversas entidades gubernamentales.

Las relaciones de confianza entre cada dominio de seguridad y la BCA pueden ser unidireccionales o bidireccionales, como se muestra en la Figura 2.1d, viéndose reducido el número de posibles relaciones de confianza hasta las n CAs de la malla de confianza. Cada dominio de seguridad funciona independiente del resto, y si alguno de ellos se viera comprometido, la BCA no se vería afectada. Si por el contrario se viera comprometida la BCA, este hecho no afectaría al funcionamiento individual de las CAs, viéndose sólo comprometidas las relaciones de confianza con la BCA y los caminos de certificación en la construcción y validación. Otra ventaja de este modelo es que el número de caminos de certificación se reduce al verse decrementado el número de relaciones de confianza, por lo que los caminos de certificación serán más fáciles de construir y validar.

		Infraestructura		Caminos de certificación		
	Modelo	**Crear**	**Escalable**	**Construcción**	**Validación**	**Inconvenientes**
Intradominio	**Simple**	Fácil	Baja	Fácil	Fácil	Punto central de fallo y alta centralización de la confianza
	Jerárquico	Fácil	Baja / Media	Fácil	Fácil	Centralización de la confianza en la CA raíz
Interdominio	**Peer-to-peer**	Media / Difícil	Media / Alta	Media / Difícil	Fácil / Media (*)	Alta complejidad al construir la infraestructura de certificación
	Full mesh	Media / Difícil	Media	Difícil	Fácil / Media (*)	Complejidad en la construcción de los caminos de certificación
	Bridge CA	Media	Alta	Media	Fácil / Media (*)	Punto central de fallo en la construcción y validación de los caminos de certificación

(*) Depende de los mecanismos y protocolos de validación

Tabla 2.1: Comparación de los principales modelos de certificación

A modo de resumen, en la Tabla 2.1 se muestran las principales diferencias entre los modelos de certificación que se han analizado más arriba, haciendo una clara distinción tanto en términos de dificultad entre la creación y escalabilidad de las infraestructuras de certificación (establecimiento de relaciones de confianza) como la implicación que éstos tienen en la validación de un certificado X.509 (construcción y validación).

2.2.2. Construcción de caminos de certificación candidatos

El grupo de trabajo del IETF sobre PKIX –soluciones PKI basadas en X.509– [112] ha definido varios protocolos estándares para la validación de caminos de certificación, incluyendo un conjunto de requisitos de obligado cumplimiento. Sin embargo, ninguna institución de estandarización ha definido un protocolo estándar que guíe el desarrollo de un servicio que construya caminos de certificación. El IETF solamente ofrece una serie de guías y recomendaciones para que los desarrolladores implementen sus propios mecanismos de construcción de caminos de certificación [104, 113]. Estos protocolos, guías y recomendaciones han surgido debido a la creciente necesidad en delegar a TTPs el complejo proceso de construir y validar los caminos de certificación. Esta delegación está motivada, principalmente, para independizar a todas las entidades, entre las que se incluyen los IDSs de un sistema colaborativo para la detección de ataques distribuidos, de todos los protocolos que tendrían que implementar para realizar la validación de un simple certificado, haciendo así que sea más fácil la implementación de las aplicaciones que utilizan habitualmente, como los procesos de detección de ataques.

De esta manera, el Servicio de Validación de esas TTPs será el encargado de realizar todo (o alguna parte) del complejo proceso de construcción y validación de caminos de certificación. Con respecto al proceso de construcción, el Servicio de Validación debe tener en cuenta los modelos de certificación analizados en la Sección 2.2.1.

Las estructuras de certificación más complejas analizadas en la Sección 2.2.1, las que hacen uso de modelos de certificación cruzada, pueden dificultar la construcción de caminos de certificación, debido a que el Servicio de Validación necesitaría *atravesar* múltiples dominios de PKI, cada uno con su propia estructura de certificación interna, teniendo que profundizar por ello en entornos de certificación multidominio.

Un entorno de certificación se puede representar por un grafo dirigido, donde cada nodo simboliza un par de claves, pertenecientes a un IDS en cuestión, y cada relación entre los nodos se puede representar como un enlace dentro del grafo. De esta forma, el proceso de construcción de caminos de certificación se reduce a un *algoritmo de búsqueda* por el árbol de certificación. Este algoritmo tiene que decidir en qué dirección debe realizar la búsqueda durante la construcción de los caminos de certificación. En este sentido, existen dos direcciones para la ejecución del proceso de construcción que pueden usarse independientemente, o incluso optar por una solución híbrida:

- *Dirección reverse*, o construcción desde la raíz. La construcción comienza por un Trust Anchor hasta que se alcance el certificado a validar. Este enfoque sería el más adecuado si el usuario del servicio define pocos Trust Anchors en su solicitud.
- *Dirección forward*, o construcción desde el objetivo. El proceso de construcción se realiza en sentido inverso al mecanismo anterior: desde el certificado a ser validado hasta que se alcance uno de los Trust Anchors. Esta construcción podría ser más compleja en modelos estrictamente jerárquicos.

Qué dirección es mejor para construir caminos de certificación es una problemática ampliamente tratada en [114]. Ahí se aconseja la construcción en sentido reverse siempre y cuando se haga uso de las optimizaciones que no puedan hacerse en sentido forward, como el procesamiento de la firma digital, el estado de revocación o la comprobación de las políticas de validación. En contraposición, en el estándar presentado en [104] se aconseja la construcción en sentido forward cuando se tengan varios Trust Anchors ya que, en sentido contrario, habría que realizar una construcción diferente para cada uno de los Trust Anchors definidos hasta encontrar el más útil para la construcción.

2.2.3. Mecanismos y protocolos de validación

Una vez construido un camino de certificación candidato, el Servicio de Validación tiene que comprobar su validez teniendo en cuenta una serie de requisitos. Primero, todos los certificados de ese camino tienen que ser estructuralmente válidos, chequeando para ello la integridad de su contenido y el correcto cumplimiento de sus extensiones. Además, y según el estándar X.509 [24], el Servicio de Validación tiene que comprobar, como requisitos mínimos para la validación estructural de un camino de certificación, los siguientes puntos: el asunto de un certificado del camino de certificación tiene que ser el emisor del siguiente, el primer certificado tiene que estar emitido por uno de los Trust Anchors, el último certificado es el que se quiere validar y cada certificado debe ser válido en el momento en el que se está ejecutando el proceso de validación.

Como segundo requisito, el Servicio de Validación debe cotejar que se cumplen todas las restricciones impuestas en los certificados, definidas en sus extensiones, además de las posibles políticas de validación exigidas por el usuario del servicio en su solicitud; el IDS en un sistema colaborativo para la detección de ataques distribuidos. Por último, cada uno de los certificados incluidos en el camino de certificación tiene que ser válidos con respecto a sus estados de revocación en el momento de la validación.

En los tres siguientes apartados se analizan los principales mecanismos y protocolos estándares de validación, todos definidos por el grupo de trabajo PKIX del IETF, con los que conocer el *estado de revocación* de un certificado X.509.

1. Certificate and Authority Revocation List (CRL/ARL)

Cuando un certificado deja de ser válido antes de su periodo máximo de validez, establecido en el elemento `notAfter`, ese certificado queda revocado –deja de ser válido– debido a varias posibles razones como, entre otras, el compromiso o pérdida de la clave privada, la CA emisora ha sido comprometida, el certificado ha sido reemitido por uno nuevo o porque la entidad que identifica ese certificado ha cambiado su afiliación.

Revocar un certificado implica que debe ser incluido en una *lista de revocación*, que es firmada digitalmente por la CA que emitió el certificado, para que el resto tenga conciencia de su invalidez, aunque siga estando en vigor según su periodo de validez. Dependiendo de qué tipo de entidad representa el certificado que se está revocando, su número de serie –único dentro de la CA– se incluye en una CRL, si es un certificado de una entidad final, como un IDS, o en una *Lista de Revocación de Autoridades* (del inglés Authority Revocation List, ARL), si el certificado representa una autoridad, como una CA o una *Autoridad de OCSP* (Online Certificate Status Protocol). La estructura de ambas listas de revocación es la misma, variando exclusivamente las instancias de esas estructuras a nivel semántico sobre qué tipo de entidades son revocadas [24].

El mayor problema de estas listas de revocación es el espacio de tiempo desde que un certificado es declarado inválido hasta que se incluye en una CRL o en una ARL. Como ejemplo, la FBCA declara en sus políticas internas que el tiempo entre la generación de dos CRLs será de 4 horas, como máximo [115]. Esto significa que, un certificado revocado poco después de haber generado la última CRL, no se verá realmente revocado hasta 4 horas después, siendo todavía válido durante ese espacio de tiempo.

2. Online Certificate Status Protocol (OCSP)

OCSP es un protocolo de validación estándar para conocer el estado de revocación de un certificado en el mismo momento en que se realiza la solicitud de validación [116]. De esta manera, se evita el problema de invalidez desde que se revoca un certificado hasta su publicación en una lista de revocación. Sin embargo, una Autoridad de OCSP sólo puede dar respuestas de validación sobre certificados que pertenecen a su PKI. En cualquier caso, es altamente recomendable que OCSP lo ofrezcan todas las PKIs, ya que lo suelen utilizar otros protocolos para comprobar el estado de revocación individual de cada certificado en el camino de certificación que está siendo analizado.

En [117] se estudia qué mecanismo o protocolo es mejor utilizar en la validación de certificados, según varios criterios como la seguridad, interoperabilidad, complejidad y rendimiento. Este estudio se decanta por OCSP ya que dispone de mayor seguridad que CRL/ARL, al realizar la validación en el mismo momento de la solicitud. Sin embargo, las pruebas de evaluación que se hicieron se centran en entornos cerrados, sin considerar escenarios multidominio con múltiples infraestructuras de seguridad bajo una misma federación de PKIs, como es el caso de los sistemas colaborativos para la detección de ataques distribuidos, que han sido planteados como objetivo en esta tesis doctoral.

3. Server-based Certificate Validation Protocol (SCVP)

SCVP es un protocolo estándar del IETF que le permite a un IDS, como ejemplo de cliente final, delegar a un Servicio de Validación la construcción y, de manera opcional, la validación de caminos de certificación [118]. Ambos se llevan a cabo bajo políticas de construcción y validación donde se especifican, entre otras cosas, el conjunto de Trust Anchors a utilizar durante los procesos de construcción y validación, qué información tiene que ser devuelta como respuesta, y qué mecanismo de revocación tiene que utilizar el Servicio de Validación (por ejemplo, el uso de CRLs/ARLs u OCSP).

Además, SCVP cumple en su totalidad con todos los requisitos que han definido los estándares Delegated Path Discovery (DPD) y Delegated Path Validation (DPV). Estas dos especificaciones definen en [113] el conjunto de requisitos que cualquier Servicio de Validación debe cumplir a la hora de definir mecanismos, algoritmos o protocolos para, respectivamente, construir y validar caminos de certificación.

2.2.4. Servicios de una PKI para entornos multidominio

Uno de los servicios que cualquier organización debe ofrecer a sus entidades finales es poder conocer si el certificado de otra entidad es o no confiable. Esta decisión se basa en la existencia de un camino de certificación válido entre ese certificado y alguno de los Trust Anchors de confianza. Como ejemplo, un escenario habitual en la vida cotidiana es cuando un usuario A recibe un correo electrónico firmado por otro usuario B. Ambos podrían ser de la misma organización, pero también pertenecer a instituciones distintas. En cualquier caso, el usuario A sólo confiará en ese correo electrónico si, y solo si, existe un camino de certificación válido entre el certificado de B, cuya clave privada asociada firmó el correo electrónico, y alguno de los Trust Anchors en los que confía A.

Un ejemplo similar se puede extrapolar a los sistemas colaborativos para la detección de ataques distribuidos. Los IDSs tienen que comprobar la existencia de, al menos, un camino de certificación válido entre ellos que les garantice un cierto nivel de confianza antes de establecer canales seguros de comunicación para el intercambio de alertas.

A la vista de estas necesidades, toda PKI debe soportar los servicios que se detallan a continuación, para estar segura que, al menos, un camino de certificación se puede construir y validar en tiempo real. Su implementación es obligatoria para todos los dominios de seguridad que deseen ofrecer un correcto Servicio de Validación.

Servicio de Directorio

En este servicio se tienen que almacenar todos los certificados que gestiona una PKI, así como las listas de revocación (CRLs y ARLs) que serán utilizadas en los procesos de validación. Los Servicios de Directorio actuales suelen estar basado en soluciones LDAP, aunque también se podrían utilizar otros protocolos como DNSSEC [119]. Tomando el caso más común en las implementaciones de PKI actuales, los servidores LDAP deben seguir los esquemas que se han definido en [109], donde se establece que se tienen que implementar, de manera obligatoria, dos clases de objeto: el objeto `pkiCA`, definido para aquellas entidades que desean actuar como Autoridades de Certificación, y el objeto `pkiUser` para guardar las entidades finales que son parte de una PKI, como los IDSs. En la Figura 2.3a se especifica la definición formal de ambas clases de objeto, como se detalla en la Sección 11.1 de la recomendación ITU-T X.509 [25].

a) **Objetos obligatorios en un Servicio de Directorio**

```
pkiCA OBJECT-CLASS ::= {
  SUBCLASS OF {top}
  KIND auxiliary
  MAY CONTAIN { cACertificate | certificateRevocationList
    | authorityRevocationList | crossCertificatePair }
  ID id-oc-pkiCA }
```

```
pkiUser OBJECT-CLASS ::= {
  SUBCLASS OF {top}
  KIND auxiliary
  MAY CONTAIN {userCertificate}
  ID id-oc-pkiUser }
```

b) **Definición del atributo para almacenar un par de certificados cruzados**

```
crossCertificatePair ATTRIBUTE ::= {
  WITH SYNTAX CertificatePair
  EQUALITY MATCHING RULE certificatePairExactMatch
  ID id-at-crossCertificatePair }
```

```
CertificatePair ::= SEQUENCE {
  issuedToThisCA   [0] Certificate OPTIONAL,
  issuedByThisCA   [1] Certificate OPTIONAL
  --  at least one of the pair shall be present }
```

Figura 2.3: Definición de los objetos a almacenar por un Servicio de Directorio

La clase de objeto `pkiCA`, como se puede ver en la parte izquierda de la Figura 2.3a, podría contener uno o varios de los siguientes atributos:

- `cACertificate`. En este atributo se almacenan los certificados de cualquier tipo de Autoridad de Certificación, ya sea una CA raíz autofirmada en un modelo simple, una CA subordinada en un modelo jerárquico o una entidad de confianza neutra en un modelo de certificación cruzada basado en una BCA.

- `certificateRevocationList`. Almacena las listas de revocación de certificados que no sean una autoridad dentro de la PKI. Es decir, son CRLs con información sobre la revocación de certificados de entidades finales como, por ejemplo, los IDSs de un sistema colaborativo para la detección de ataques distribuidos. Estas CRLs son emitidas por alguna de las CAs almacenadas en `cACertificate`.

- `authorityRevocationList`. De forma análoga, este atributo también almacena listas de revocación, pero en este caso de autoridades dentro de la PKI.

- `crossCertificatePair`. Almacena la lista de certificados cruzados que alguna de las CAs de `cACertificate` ha emitido para establecer relaciones de confianza con otros dominios de seguridad. Su definición formal se puede ver en la Figura 2.3b, donde en la parte derecha se muestra el objeto `CertificatePair`, el cual define dos elementos opcionales para almacenar un par de certificados cruzados con los que establecer una relación de confianza bidireccional entre dos dominios. Ambos corresponden con los certificados *forward* (elemento `issuedToThisCA`) y *reverse* (elemento `issuedByThisCA`) explicados en la Sección 2.2.1.

Por último, la definición de la clase de objeto `pkiUser`, definida en la parte derecha de la Figura 2.3a, solamente permite incluir el atributo `userCertificate`, donde la PKI debe almacenar, obligatoriamente, los certificados de las entidades finales.

Servicio de Validación

Como especifica el estándar [113] en sus requisitos, todos los dominios de seguridad tienen que implementar un Servicio de Validación al que se le pueda delegar el complejo proceso de validación de un determinado certificado. Como parte principal de cualquier Servicio de Validación, el proceso de validación dentro de un entorno multidominio, con múltiples certificados emitidos por distintas unidades organizativas, debe estar fundado en protocolos de validación estándares. El uso de protocolos propietarios puede conducir a errores en la validación de alguno de los certificados del camino de certificación, al poder haber dominios que no soporten esos protocolos específicos.

Por tanto, el objetivo de interoperabilidad que ofrecen los protocolos estándares es un requisito indispensable para el correcto funcionamiento de los procesos de validación. En este sentido, se pueden utilizar varios protocolos para desplegar este servicio, como SCVP u OCSP (analizados en la Sección 2.2.3).

En conclusión, el Servicio de Directorio y el Servicio de Validación son críticos para cualquier PKI que desee ofrecer mecanismos de construcción y validación de caminos de certificación. Sin embargo, en la literatura actual no se pueden encontrar soluciones basadas en PKI que definan, e implementen, un Servicio de Validación con el que llevar a cabo estos dos procesos (construcción y validación) en escenarios multidominio, involucrando a más de un dominio de seguridad. Sin este servicio, los IDSs de un sistema colaborativo para la detección de ataques distribuidos no van a poder establecer canales seguros en sus comunicaciones para el intercambio de sus alertas de detección.

2.3. Sistemas para detectar ataques distribuidos

Una vez analizados los mecanismos que proveen de un marco de seguridad, para que los IDSs de un sistema colaborativo para la detección de ataques distribuidos puedan intercambiar sus alertas de forma segura, el siguiente paso se debe centrar en los propios modelos que, en sí mismos, permitan llevar a cabo esos procesos de detección.

Para llevar a cabo un análisis en profundidad sobre los modelos más relevantes que se han propuesto en la literatura actual, sobre diversos sistemas colaborativos para la detección de ataques distribuidos, en primer lugar se presenta una serie de criterios que, posteriormente, se utilizarán como parámetros a la hora de clasificar esos sistemas de detección. Dependiendo de esos criterios se podrá comprobar cuál es el objetivo de cada sistema por separado, y la manera por lo que han sido diseñados así.

2.3.1. Criterios en el diseño de un sistema colaborativo

Antes de realizar el diseño de cualquier sistema colaborativo, los administradores encargados de su desarrollo deben decidir cuál es la mejor distribución de los distintos componentes del sistema. En un contexto orientado a la detección de ataques, estos componentes serán los IDSs. A continuación se presentan cuatro criterios bien conocidos en la literatura para diseñar un sistema colaborativo, dando preferencia a los sistemas colaborativos de detección de ataques al ser objetivo de esta tesis doctoral [120].

Los administradores tienen que evaluar, como primer criterio, el *ámbito* inicial sobre cómo van a realizar el despliegue del sistema colaborativo para la detección de ataques. A este respecto, existen tres alternativas principales:

- *Local*. Todos los IDSs del sistema son altamente confiables entre sí ya que estarán desplegados bajo el mismo marco organizativo, por lo que no es tan necesario el uso de mecanismos altamente sofisticados para la gestión de su seguridad.
- *Global*. Los IDSs tienen que desplegarse entre varios dominios de seguridad y/o administrativos. Este ámbito global del sistema provoca que todas las alertas que tienen que compartirse entre los IDSs tengan que estar limitada en su contenido, a fin de evitar el envío de información sensible (por ejemplo, direcciones IP) a los IDSs que no pertenezcan al mismo dominio que el emisor.
- *Híbrida*. El sistema colaborativo se divide entre diferentes zonas de confianza, o dominios de seguridad, donde cada uno puede establecer sus propias políticas de seguridad de acuerdo a sus distintos niveles de confianza.

En cualquiera de estos ámbitos de despliegue surge un nuevo criterio para el diseño de un sistema colaborativo, el relacionado con el cumplimiento de la *privacidad* para gestionar la información que los IDSs del sistema pueden intercambiar. Su contenido se puede ver reducido, por motivos de seguridad, dependiendo del nivel de confianza que los IDSs tengan entre sí, ya sean parte o no del mismo dominio de seguridad y/o administrativo. El objetivo sobre la privacidad es no revelar información sensible a otros IDSs (por ejemplo, el envío de puertos donde estén escuchando ciertos servicios críticos) los cuales podrían estar manifestando un comportamiento malicioso en sus intenciones. Como posible solución, los IDSs podrían eliminar cierto tipo de información. Existen muchos trabajos que pretenden dar solución al problema de la privacidad en sistemas colaborativos para la detección de ataques, donde el intercambio de alertas entre los IDSs es un factor clave en sus propósitos de detección.

Por ejemplo, en [121] se presenta el concepto de jerarquía con el que poder balancear los requisitos de privacidad y la necesidad en el análisis de las alertas intercambiadas, obligatorio en la detección de ataques. Este trabajo propone el uso de la entropía a la hora de generalizar los atributos sensibles de las alertas a conceptos de más alto nivel. Otros trabajos, como [122], hacen uso de técnicas de ofuscación que garantizan altos niveles de privacidad sin comprometer los procesos de detección. Más recientemente, la gestión de la privacidad se centra en proteger las propias reglas de detección que utilizan los IDSs, ya que cualquier atacante que haya comprometido un IDS puede analizar sus reglas y aprender sobre las vulnerabilidades del sistema para lanzar su próximo ataque con éxito. En este contexto, el sistema ZIDS [123] se apoya en protocolos criptográficos para garantizar la privacidad de las firmas de ataque de un IDS.

Como tercer criterio se establece la *especialización*, o atribuciones, de un sistema colaborativo, siendo los tipos de amenazas un ejemplo en un sistema colaborativo para la detección de ataques, como ataques de denegación distribuida de servicio (DDoS) o inyección SQL [124]. Por tanto, el sistema colaborativo puede variar considerablemente según el objetivo final para el que está diseñado. Por ejemplo, la instalación de los IDSs seguirán un modelo de despliegue diferente si el sistema desea detectar ataques DDoS o correo basura. En el primer caso se necesita un despliegue estratégico de un gran número de NIDSs en todas las redes posibles, mientras que, en el segundo caso, la instalación de un pequeño número de HIDSs en los servidores de correo sería suficiente.

Finalmente, y como último criterio, se asienta la *topología* que debe seguir el sistema colaborativo. Este punto va a tener un alto impacto tanto en la escalabilidad como en la robustez del sistema colaborativo para la detección de ataques distribuidos, ya que se deben analizar grandes volúmenes de información en tiempo real [6, 125]. Este hecho se ha identificado en la Sección 2.1 como un reto al que se le tiene que dar solución. En los siguientes puntos se presentan los cuatro esquemas topológicos más importantes.

1. Esquema centralizado

Toda la información es analizada por un único IDS del sistema de forma totalmente centralizada. Es el esquema más fácil de administrar, simplificando al máximo su diseño, y el más eficiente ya que ese IDS central tiene a su disposición todas las alertas que han generado el resto de IDSs, haciendo que la información sea totalmente consistente y coherente. Las soluciones DIDS (Distributed Intrusion Detection System) [126], IDES (Intrusion Detection Expert System) [127] y NADIR (Network Anomaly Detection and Intrusion Reporter) [128] se afianzaron como las primeras propuestas más importantes en distribuir sus unidades de detección siguiendo un enfoque centralizado.

A pesar de esas ventajas, un esquema centralizado tiene unos inconvenientes que lo hacen inviable para los escenarios colaborativos. Esta centralización presenta grandes problemas de robustez y seguridad al ser el IDS central un punto único frente a fallos y ataques, no siendo aconsejable que toda la confianza recaiga sobre una única entidad. Además, también suele experimentar problemas de escalabilidad, al ser ese IDS central un posible *cuello de botella* para el rendimiento del sistema colaborativo.

2. Esquema jerárquico

La distribución de los IDSs en el sistema siguiendo un enfoque jerárquico surgió a raíz de los problemas de escalabilidad que tenían los esquemas centralizados. En este nuevo enfoque, los IDSs preprocesan de manera local todas las alertas que reciben de los IDSs inferiores en la estructura jerárquica, enviando a la siguiente capa de la jerarquía la nueva información resultante para su intercambio y posterior análisis. Los sistemas colaborativos siguiendo este esquema distribuyen sus IDSs en pequeños grupos según su localización geográfica, la complejidad de su administración, la agrupación de todos los IDSs con características similares y otros factores particulares del sistema colaborativo que se trate. Por ejemplo, otros factores como los tipos de ataques que son previstos en un sistema colaborativo para la detección de ataques –el sistema puede conocer de antemano las posibles vulnerabilidades ante las que está expuesto.

De entre los sistemas colaborativos para la detección de ataques se pueden destacar, como los ejemplos más representativos, dos de las soluciones más pioneras: el proyecto EMERALD [97] y también GrIDS (Graph-based Intrusion Detection System) [129]. La información en GrIDS es agregada de forma jerárquica en *grafos de actividades*, donde cada nodo en ese grafo representa uno o varios hosts agrupados en un pequeño grupo, y cada arista representa el tráfico de red entre ellos. Por otro lado, el proyecto EMERALD divide su sistema colaborativo en tres capas de abstracción, las cuales incluyen procesos de análisis independientes y ajustables que permiten combinar técnicas de detección de anomalías (modelando perfiles estadísticos) y detección de usos indebidos.

Este nuevo esquema topológico de distribuir los IDSs de forma jerárquica evita que se tenga un punto único frente a fallos y ataques, además de que permite gestionar el problema de la escalabilidad con una mayor solvencia; dos inconvenientes que sí tienen los sistemas centralizados. Sin embargo, las estructuras jerárquicas sufren parcialmente en las capas más altas de la jerarquía de los dos problemas comentados anteriormente. Un fallo en los IDSs superiores puede comprometer la funcionalidad de gran parte de la jerarquía y, en consecuencia, de casi todo el sistema colaborativo. Adicionalmente, una estructuración de los IDSs siguiendo un enfoque jerárquico también hace que el nivel de abstracción de la información sea mayor en las capas superiores de la jerarquía. En sistemas colaborativos para la detección de ataques, esa abstracción implicaría que la cobertura de la detección en las capas superiores fuera incluso más limitada, ya que la información sería demasiado "genérica" para detectar ataques distribuidos.

3. Esquema totalmente descentralizado

El despliegue de los IDSs bajo un esquema totalmente descentralizado apareció para dar solución a los problemas que presentaban los dos enfoques anteriores. En este nuevo esquema, los IDSs de un sistema colaborativo para la detección de ataques actúan como sistemas autónomos sin tener un punto único frente a fallos y ataques, evitando así que se tengan entidades demasiado confiables para el resto de miembros del sistema. Además, este enfoque también da solución a los problemas de robustez y escalabilidad que hacían inviables a los sistemas centralizados [130].

Estas ventajas hacen sospechar que los esquemas totalmente descentralizados son más atractivos en su implementación que los enfoques centralizados o jerárquicos. Sin embargo, la descentralización causa unos inconvenientes que podrían ser determinantes según el escenario donde se aplique. En escenarios con un alto grado de movilidad, por ejemplo, la gestión de la seguridad se hace bastante compleja, como ocurre en las *Redes Vehiculares Ad-Hoc* (del inglés Vehicular Ad-Hoc Network, VANET). Muchos trabajos sobre estas redes se basan en soluciones de PKI para crear un modelo de seguridad entre todos los vehículos del sistema totalmente descentralizado. Por ejemplo, en [131] se propone que cada vehículo posea un certificado X.509 para que pueda ser identificado como una entidad unívoca en la VANET, además de poder establecer comunicaciones seguras con el resto de vehículos vecinos [132]. El uso de tecnologías basadas en PKI hace que los modelos de certificación cruzada, analizados en la Sección 2.2.1, tengan una gran relevancia, ya que los certificados de los vehículos pueden ser de distintos dominios. Además de esta complejidad en la gestión, otro de los problemas es la gran cantidad de mensajes que los IDSs deben intercambiar para mantener la información consistente y coherente, independientemente del escenario donde se implanten.

Entre los trabajos más representativos de sistemas colaborativos para la detección de ataques, siguiendo un esquema totalmente descentralizado, se pueden encontrar Indra (Intrusion Detection and Rapid Action) [133] y Worminator [134], este último implementando un modelo de comunicaciones Peer-to-Peer (P2P). En ambos trabajos, los IDSs que participan en el sistema colaborativo intercambian información –alertas y topología de la red, entre otros– con el resto de IDSs de forma periódica.

4. Esquema parcialmente descentralizado

Este esquema surgió para darle solución a los inconvenientes implícitos en los tres enfoques anteriores, abordando los problemas de tener un punto único frente a fallos y ataques, la falta de escalabilidad y robustez, la alta sobrecarga en las comunicaciones debido al intercambio excesivo de información y la dificultad en su gestión. Los dos primeros son inherentes a los esquemas centralizados, mientras que los dos últimos son problemas innatos a los esquemas jerárquicos y totalmente descentralizados.

Los IDSs en este esquema se distribuyen entre los distintos dominios de seguridad, donde un IDS por dominio actuará como líder [135]. Estos IDSs, llamados *supernodos*, compartirán entre ellos la información indispensable para detectar ataques distribuidos. De esta manera, la estructura interna de cada dominio de seguridad seguirá un enfoque centralizado, mientras que la construcción del sistema colaborativo para la detección de ataques distribuidos va a seguir un enfoque totalmente descentralizado, permitiendo así que todos los IDSs tengan una visión común y holística de la información.

El mayor inconveniente de este enfoque es que los supernodos deben ser totalmente confiables, tanto para los IDSs de su dominio de seguridad como para los supernodos de los otros dominios. ABDIAS (Agent-based Distributed Intrusion Alert System) [136] y TRINETR [137] son las primeras soluciones que implementan un sistema colaborativo para la detección de ataques siguiendo un esquema parcialmente descentralizado.

Como resumen, la Tabla 2.2 muestra las ventajas e inconvenientes más importantes de los cuatro esquemas de despliegue analizados en los puntos anteriores.

Esquema	Ventajas	Inconvenientes
Centralizado	▪ Fácil de desplegar y administrar ▪ Simplicidad en el diseño ▪ Máxima eficiencia al disponer de toda la información posible ▪ Mantiene la información totalmente consistente y coherente	▪ Punto único frente a fallos y ataques ▪ Alta confianza en una única entidad ▪ Gran cuello de botella para el rendimiento ▪ Grandes problemas de escalabilidad ▪ Difícil alcanzar un alto rendimiento y robustez
Jerárquico	▪ No hay punto central de fallos y ataques ▪ Diseño y despliegue bastante sencillo ▪ El análisis de la información se distribuye entre todos los IDSs de la jerarquía	▪ Escalabilidad muy limitada ▪ La robustez se puede ver comprometida en cada capa de la jerarquía ▪ Las capas adyacentes en la estructura jerárquica deben ser confiables entre sí ▪ Alto nivel de abstracción de los datos
Totalmente descentralizado	▪ No hay punto único de fallos y ataques ▪ No es necesaria una entidad global totalmente confiable ▪ Alta robustez y escalabilidad	▪ Sobrecarga de mensajes por inundación ▪ Mayor vulnerabilidad a la manipulación ▪ La topología del sistema puede cambiar con cierta frecuencia
Parcialmente descentralizado	▪ Mayor facilidad en su manejo y gestión ▪ Máxima robustez y escalabilidad ▪ Reducción en el intercambio de mensajes ▪ Búsqueda eficiente de la información	▪ Los líderes tienen que ser totalmente confiables para el resto de IDSs ▪ Carga adicional de trabajo a los líderes

Tabla 2.2: Ventajas e inconvenientes de los principales esquemas de despliegue

2.3.2. Modelos para la detección de ataques distribuidos

En esta sección se analizan los sistemas colaborativos que son más significativos para la detección de ataques distribuidos. Aunque a continuación se van a describir cada uno, en la Tabla 2.3 se muestra un resumen de los mismos, ordenados cronológicamente según el momento en el que aparecieron para deducir, de forma más clara, la tendencia que han tenido con respecto a los criterios que se han presentado en la sección anterior: *topología* en el despliegue de los IDSs, *ámbito* de despliegue, *especialización* de qué tipos de amenazas son capaces de detectar y *gestión* de la seguridad y la privacidad.

Los primeros sistemas de detección de ataques distribuidos que se propusieron se centraban exclusivamente en el despliegue de los diferentes IDSs. Los administradores habían identificado que el modelo de despliegue era un elemento clave en la detección de este nuevo tipo de ataque. Al principio, la mayoría comenzaron a desplegar los IDSs siguiendo un enfoque colaborativo de forma centralizada, apareciendo soluciones como DIDS, IDES y NADIR, introducidas anteriormente, pero enseguida emergieron otras como NSTAT (Network State Transition Analysis Tool) [138], DShield [139] y CRIM (Cooperative Intrusion Detection Framework) [140].

	Topología				Ámbito			Especialización			Gestión	
Esquema	Centralizado	Jerárquico	Totalmente descentralizado	Parcialmente descentralizado	Global	Local	Híbrido	General	Malware	Spam	Seguridad	Privacidad
DIDS	✓					✓		✓				
IDES	✓				✓			✓				
NADIR	✓					✓		✓			✓	✓
CSM			✓		✓			✓				
GrIDS		✓			✓			✓				
EMERALD		✓				✓		✓				
NSTAT	✓					✓		✓				
AAFID		✓			✓			✓				
DShield	✓				✓			✓				✓
CRIM	✓					✓		✓				
Indra			✓			✓		✓			✓	
CIDS	✓				✓			✓				
DOMINO		✓	✓				✓		✓		✓	
ABDIAS				✓			✓	✓				
TRINETR				✓	✓			✓				
NetShield			✓		✓				✓		✓	
Worminator			✓		✓				✓			✓
PAID			✓		✓			✓			✓	
MIND	✓		✓				✓	✓				
Gossip	✓		✓			✓			✓			
D-SCIDS			✓				✓	✓				
DSOC		✓		✓			✓	✓			✓	
ALPACAS			✓		✓					✓		✓
MADIDF			✓			✓		✓				
CIMD				✓			✓	✓			✓	
ZIDS	✓					✓		✓			✓	✓
MIDS			✓			✓		✓			✓	
ACO-AD		✓	✓		✓			✓			✓	

Tabla 2.3: Principales sistemas para la detección de ataques distribuidos (ordenados cronológicamente por fecha de aparición)

Durante ese tiempo, se empezaron a desarrollar nuevos sistemas optando por otros esquemas que pudieran solucionar los problemas intrínsecos a los modelos centralizados. Este hecho dio lugar a que, de forma simultánea a las soluciones anteriores, empezaran a surgir los primeros sistemas que funcionaban bajo un modelo jerárquico. Entre esos sistemas se pueden destacar GrIDS, EMERALD y AAFID (Autonomous Agents For Intrusion Detection) [141, 142], aunque de forma muy temprana también surgieron CSM y, algo más posterior, Indra, que fueron los primeros sistemas en aprovecharse de las ventajas que proporcionan los esquemas totalmente descentralizados.

Analizando la Tabla 2.3, se puede ver que el sistema DOMINO (Distributed Overlay for Monitoring Internet Outbreaks) [143] supuso un punto de inflexión en los modelos de despliegue de los IDSs. Aunque no se pueda considerar que esté desarrollado bajo un modelo parcialmente descentralizado, sí fue el primero en seguir un modelo híbrido: cierta parte del sistema se gestionaba de forma jerárquica y otra de forma totalmente descentralizada. DOMINO se centra en que los componentes de un sistema distribuido intercambien información para detectar ataques, definiendo tres tipos de componentes: *axis overlay*, *satellite communities* y *terrestrial contributors*. Los axis se organizan en una red overlay P2P a través de un modelo publicación/suscripción, mientras que los satélites se asocian entre sí, o con un nodo axis en particular, mediante un modelo de despliegue jerárquico. Los nodos axis son los responsables de la detección de ataques distribuidos mediante el intercambio periódico de la información de intrusiones extraída de los nodos satélites, haciendo que ese intercambio fuera seguro con criptografía de clave pública. El último componente, los nodos terrestrial, se definieron como un modo de extensión de DOMINO para que se pudiera incluir nueva información de utilidad; por ejemplo, información de intrusiones obtenidas de entidades externas.

A partir del sistema DOMINO, empezaron a emerger nuevas soluciones desplegando los diferentes IDSs bajo un modelo parcialmente descentralizado. Por ejemplo, los dos sistemas que se han introducido anteriormente, ABDIAS y TRINETR, así como otras propuestas posteriores como DSOC (Distributed Security Operation Center) [144] y CIMD (Collaborative Intrusion and Malware Detection) [145]. Durante ese pequeño espacio de tiempo también aparecieron la gran mayoría de propuestas concebidas para la explotación de sus sistemas bajo un modelo totalmente descentralizado como, entre otros, NetShield [146], PAID (Probabilistic Agent-based Intrusion Detection) [147], MIND (Multi-dimensional Indexing System for Network Diagnosis) [148], Gossip [149] y D-SCIDS (Distributed Soft Computing-based IDS) [150].

Sistemas más modernos que los anteriores suelen inclinarse más hacia los esquemas total o parcialmente descentralizados, como, por ejemplo, los sistemas ALPACAS (A Large-scale Privacy-aware Collaborative Anti-spam System) [151], MADIDF (Mobile Agent based Peer-to-Peer Distributed Intrusion Detection Framework) [152] o MIDS (Multiagent-based Architecture for Intrusion Detection System) [153]. Aquellos que no lo hacen, los que siguen prefiriendo un esquema centralizado o jerárquico, es porque el ámbito de despliegue de sus IDSs está dirigido a la detección local de ataques dentro de su propio, y único, dominio de seguridad. Ahí consideran que los esquemas centralizados o jerárquicos proporcionan mayores ventajas que los otros esquemas.

Con respecto al ámbito en el despliegue de los IDSs, casi todas las soluciones han optado por una orientación global o local de la detección, mientras que muy pocas han preferido una perspectiva híbrida. La mayor ventaja de los sistemas que se decantaron por esta última orientación era proporcionar a cada uno de los dominios de seguridad y/o administrativos la elección de sus propias políticas internas de seguridad. De nuevo, el sistema DOMINO fue uno de los primeros en promulgar una distribución de sus IDSs siguiendo un modelo híbrido. Además, como se puede observar en la Tabla 2.3, existe una cierta relación entre la topología y el ámbito escogido para el despliegue de los IDSs: un ámbito híbrido solamente lo pueden proporcionar aquellos sistemas desplegados bajo un modelo descentralizado, o mediante una combinación entre los modelos jerárquicos y los totalmente descentralizados, como propone DOMINO.

Actualmente, múltiples líneas de investigación están posando el foco de atención en las comunicaciones entre los IDSs, donde se han abierto otras cuestiones como: ¿qué tipo de información pueden intercambiar los IDSs sin comprometer la privacidad? Y si no se puede compartir todo, ¿cómo se vería afectada la precisión en la detección de ataques? Además, ¿qué protocolos deben aplicarse para que las comunicaciones entre los IDSs sean seguras? Aunque algunos de los primeros sistemas ya lanzaron unas propuestas iniciales, han sido los más recientes los que han identificado la seguridad y la privacidad como factores primordiales en el intercambio de la información de detección.

2.4. Modelos de confianza basados en reputación

Como se ha esbozado en los retos presentados en la Sección 2.1, otro desafío de suma importancia, ante el que los sistemas colaborativos para la detección de ataques deben adoptar una solución, es la identificación de comportamientos maliciosos de los IDSs. Si, por ejemplo, uno o varios IDSs han sido comprometidos por un atacante, éstos pueden compartir falsos positivos sobre incidentes que no han ocurrido realmente, captando así la atención del sistema sobre alguna zona de monitorización para lanzar el próximo ataque sobre otra área que está siendo "desatendida" por el sistema en ese momento. Además, la identificación y eliminación de falsos positivos debido a comportamientos maliciosos, también ayudará al sistema a que tenga una mayor robustez en sus activos de detección, así como una mayor precisión a la hora de detectar ataques que realmente estén ocurriendo en el sistema. Por tanto, los IDSs deben tener un nivel de confianza suficiente en que otros IDSs no presentan actitudes maliciosas en sus comportamientos, antes de aceptar sus alertas como evidencias de hechos ocurridos en la realidad.

Los sistemas basados en reputación para la *evaluación de la confianza* son capaces de modelar el comportamiento de cualquier tipo de entidad, con los que poder identificar la existencia de un comportamiento malicioso a la hora de, por ejemplo, compartir las alertas entre varios IDSs [26]. Los mecanismos que implementan esos sistemas necesitan cierta información para realizar un cálculo de la confianza lo más preciso posible, antes de que se lleve a cabo el intercambio de alertas entre los IDSs. De dónde y cómo extraer esa información es un punto clave en el cálculo de la confianza sobre un IDS.

2.4.1. Fuentes de información para el cálculo de la reputación

El uso de una fuente de información u otra en el cálculo de la reputación, o incluso una mezcla entre ellas, recaerá principalmente en:

- *Disponibilidad de la información.* Es posible que dos IDSs que deseen intercambiar sus alertas nunca lo hayan hecho anteriormente, por lo que no se dispondrá de la información suficiente para realizar el cálculo de la reputación. Pero, también puede darse el caso que el IDS que está realizando el cálculo sí que tenga un valor de reputación sobre la otra entidad, aunque ese valor puede ser tan antiguo que no represente el comportamiento real que pudiera tener en la actualidad.
- *Conocimiento extraído de la comunidad.* El IDS que desea realizar el cálculo puede solicitar a otros IDSs de su comunidad información sobre el comportamiento que ha tenido con ellos el IDS sobre el que está calculando su reputación.
- *Precisión en los cálculos.* El IDS podría utilizar información imprecisa al realizar el cálculo de la reputación. Este hecho ocurre, principalmente, si la información procede de otros IDSs que utilizan otros mecanismos distintos en sus evaluaciones. Es probable que evalúen la reputación sobre un mismo IDS de forma desigual.

Siguiendo esas limitaciones, las fuentes de información se pueden clasificar en cuatro grupos. En los siguientes apartados se detallan cada uno de ellos, los cuales se pueden utilizar para realizar el cálculo de la confianza de un IDS a través de su reputación. En el análisis de cada grupo también se presentan los sistemas de confianza y reputación más importantes que aprovechan las particularidades de cada uno.

Experiencias directas

Estas experiencias, también conocidas como *interaction trust*, hacen referencia a los intercambios de alertas que previamente han mantenido dos IDSs (sobre el que se está calculando su reputación y el que la está realizando) [154]. Después de llevarse a cabo el intercambio de cada una de las alertas, el IDS que está calculando la reputación del otro IDS tiene que "puntuar" la satisfacción que ha obtenido de este último. Cuanto más alta, mayor será la confianza en que el IDS al que se le está calculando la reputación ha exhibido un buen comportamiento. La mayoría de trabajos representan la satisfacción en formato numérico con valores continuos, normalmente en un rango entre 0 y 1, pero también existen otros donde se usan otros tipos de representaciones, como la definición de un conjunto discreto de n elementos. En [155], por ejemplo, el modelo de reputación cognitivo Repage, para la evaluación de la confianza en sistemas multiagentes, emplea cinco etiquetas: *Very Bad*, *Bad*, *Neutral*, *Good* y *Very Good*.

Después de que el IDS que está calculando la reputación mida la satisfacción sobre el intercambio de una alerta, éste tiene que actualizar la reputación del otro IDS para plasmar su comportamiento actual. Si no ha cumplido con las expectativas esperadas, la reputación del otro IDS tendrá que ser algo menor, mientras que será incrementada si

el resultado ha sido satisfactorio. Al actualizar el valor de reputación, también deberían considerarse otros factores que podrían alterar su resultado. Por ejemplo, el tiempo es un factor a considerar, ya que los intercambios recientes de alertas suelen afectar a la confianza con un mayor impacto que las antiguas (representan mejor el comportamiento actual de un IDS). A esta función de penalización basada en el tiempo se le suele conocer en la literatura como *factor de olvido* (del inglés Forgetting Factor).

Sporas [156] y REGRET [157] son de los primeros sistemas en modelar un factor de olvido, aunque ambos hacen alusión a un valor de "frescura" como un peso en función de la media entre el tiempo actual y cuando se registró la interacción. Más recientemente, en [158], el factor de olvido se modela a través de un factor exponencial F^{t_k}, con un valor fijo $0 \leq F \leq 1$, donde t_k es el momento cuando la entidad que está realizando el cálculo tuvo constancia de la interacción. Por otro lado, en [159] se presenta FIRE, un modelo que evalúa la confianza en sistemas multiagente para seleccionar los mejores con los que interactuar, donde su factor de olvido penaliza las interacciones de manera exponencial conforme aumenta el tiempo desde que se llevaron a cabo:

$$\omega_l(r_i) = e^{-\frac{\Delta t(r_i)}{\lambda}}, \text{ siendo inicialmente } \lambda = -\frac{5}{ln(0,5)}$$

donde $\omega_l(r_i)$ es el peso asignado a la evaluación r_i y $\Delta t(r_i)$ la diferencia entre el tiempo actual y el tiempo cuando se registró la evaluación r_i.

En conclusión, la principal ventaja en usar experiencias directas es que el cálculo de la reputación se basa en la propia experiencia que un IDS tiene sobre otro IDS, aunque hay que tener en cuenta que la escala de evaluación para un IDS puede ser diferente al de otros con los que está colaborando en el mismo sistema. Además, tampoco introduce sobrecarga en las comunicaciones, al no ser necesarias las experiencias que otros IDSs externos hayan tenido con el IDS al que se le está calculando la reputación.

A pesar de todo, hace falta un número relativamente alto de intercambios de alertas para obtener un cálculo lo más preciso posible. Además, también es necesario que esos intercambios sean recientes ya que, en caso contrario, el cálculo se realizaría con unos intercambios de alertas que podrían no reflejar el comportamiento actual del IDS que se está evaluando. Este problema se acrecienta durante la *fase de bootstrapping* de una nueva entidad, como un IDS, cuando éste todavía no ha intercambiado ninguna alerta. Al no existir evidencias directas de colaboración, el cálculo de la reputación se tendría que hacer con otros mecanismos que utilizaran, por ejemplo, las experiencias que otros IDSs externos de confianza han tenido con el IDS que se está evaluando.

Experiencias indirectas (*recomendaciones* o *advertencias*)

Este tipo de información, también conocida habitualmente como *witness reputation*, la proporcionan IDSs externos de confianza que le ofrecen al IDS que está realizando el cálculo de la reputación sus propias opiniones sobre el comportamiento que han tenido en el pasado con el IDS que está siendo evaluado [160]. Estas opiniones están basadas en las experiencias directas de esos IDSs externos, siendo *recomendaciones* o *advertencias* dependiendo de si la opinión ha sido positiva o negativa, respectivamente.

Entre las diversas soluciones que hacen uso de experiencias indirectas, destacar las dos características que define PeerTrust [161]. Por un lado, introduce tres parámetros básicos de la confianza, donde se incluyen las opiniones que tiene una entidad sobre las interacciones que ha realizado otra tercera entidad con el sistema, así como dos factores de contexto: uno sobre las interacciones en sí mismas y otro sobre la comunidad donde se ejecutan dichas interacciones. Por otro lado, PeerTrust también define una métrica de confianza global para combinar todos los parámetros anteriores.

Las experiencias directas se deben a la confianza individual que gestionan los propios IDSs, mientras que las experiencias que provienen de otros IDSs externos se vinculan a la *confianza social*. Como ejemplo de experiencias indirectas, se suele considerar una red de relaciones de confianza del tipo $A \rightarrow B \rightarrow C$. La entidad A puede confiar con un cierto grado en las recomendaciones que B le envíe, según la confianza que A tenga en B, pero también podría obtener recomendaciones que le pudiera enviar C a través de B. A este fenómeno, con respecto a la transitividad de la confianza, se le denomina *propagación de la confianza* (del inglés Trust Propagation) [162].

Los trabajos que estudian cómo propagar la confianza, en redes como la del ejemplo anterior, se pueden separar en dos grupos distintos. Ambos se diferencian en el ámbito en el que las entidades calculan y asignan los valores de confianza.

- *Ámbito global*. Son técnicas que atribuyen un valor de confianza global a cada entidad, siendo este valor el mismo para todos los miembros de la red de confianza. Entonces, la composición del valor de confianza entre dos entidades se calcula según el valor mínimo de todos los enlaces para llegar desde una entidad hasta otra, no pudiendo ser el valor de confianza entre dos entidades superior al enlace más débil entre ellas. Como ejemplo, esta técnica se usa en el modelo de confianza y reputación EigenTrust para redes de comunicaciones P2P [163].

- *Ámbito local*. Son técnicas que asignan los valores de confianza entre cada par de entidades, por lo que es más realista que la anterior ya que dos entidades pueden tener opiniones distintas sobre cualquier otra entidad. En [164], se proponen varios algoritmos para esa propagación, pero son los trabajos [165] y [166] los primeros presentes en la literatura actual que ofrecen estudios comparativos sobre varios de los algoritmos propuestos en la propagación de la confianza.

Concretamente, [165] propone una *función de concatenación* según las confianzas depositadas en cada arista del camino de confianza, y una *función de agregación* final donde se combinan todas esas creencias intermedias. Utiliza la multiplicación para la función de concatenación, y propone que se utilicen tres enfoques para la de agregación: valor máximo (operación OR), valor mínimo (operación AND) y media aritmética. Por otro lado, [166] también propone un proceso de propagación similar al anterior, aunque en este caso se aplica además una penalización a cada uno de los saltos del camino de confianza. En la red de ejemplo anterior, a la segunda relación $B \rightarrow C$ se le aplica una penalización desde el punto de vista de la entidad A, ya que en esa relación se establece un indicador de confianza en la que A no está implicada de forma directa.

A modo de conclusión, las experiencias indirectas pueden ayudar a que un IDS, que está realizando el cálculo de la reputación sobre otro IDS, obtenga un cálculo más satisfactorio, al tener un mayor número de evidencias sobre su comportamiento. Sin embargo, existen varios inconvenientes que recomiendan que se utilicen las experiencias indirectas de forma complementaria a los otros tipos de fuentes de información. Uno de esos inconvenientes es que los IDSs que ofrecen esa información tienen que exponer un comportamiento altruista, lo cual no siempre es una opción posible, por lo que el concepto *colaboración* tendrá que estar muy bien motivado e incentivado [167]. Además, también habría que tener en cuenta que este enfoque necesita de una mayor capacidad de almacenamiento, ya que se deben registrar todos los intercambios de alertas junto con sus evaluaciones. Finalmente, también hay que tener en cuenta que la propagación de la confianza no escala bien en escenarios altamente dinámicos, ya que los IDSs no podrían almacenar ni gestionar toda la red (malla) de relaciones de confianza.

Reglas de confianza basadas en roles

La imposibilidad de tener experiencias directas o indirectas de, y sobre, un IDS, debido por ejemplo a que es nuevo en la comunidad, abre el abanico a incorporar nuevas fuentes de información con las que se pueda "deducir" su reputación. Un ejemplo es la definición en FIRE de un conjunto de *reglas de confianza basadas en roles* (del inglés Role-based Rules), donde la reputación de una entidad se calcula aplicando un conjunto de reglas preestablecidas [159]. Estas reglas se obtienen a partir del modelado de otras entidades muy similares en sus características a la nueva entidad, o incluso incorporando normas sociales muy específicas del dominio, como, por ejemplo, si pertenecen al mismo dominio administrativo o si la entidad a evaluar se ha autenticado o no en el sistema. Por tanto, cada entidad debe especificar y mantener su propio conjunto de reglas en una base de datos interna donde almacene esas reglas basadas en roles.

Aunque esta fuente de información se suele utilizar en casos muy particulares, se puede utilizar también de forma complementaria con las dos anteriores para mejorar la precisión en el cálculo de la reputación de cualquier IDS. Sin embargo, esta fuente está tomando bastante auge en la actualidad en el estudio de qué reputación debe tener una nueva entidad cuando nunca ha interactuado con la entidad que está calculando la reputación. O sí lo ha hecho, pero con otras entidades con las que no tiene ninguna relación de confianza. En cualquier caso, como esta problemática en el cálculo inicial de la confianza de una nueva entidad se ha enunciado como un objetivo de esta tesis doctoral, esta fuente de información se incluye en la siguiente sección bajo un apartado propio e independiente donde se analizará con mayor nivel de detalle.

En conclusión, la principal ventaja del uso de reglas basadas en roles es la posibilidad de calcular un valor inicial de la reputación a IDSs totalmente desconocidos para el dominio de seguridad donde desea colaborar. Sin embargo, aparecen inconvenientes que podrían desaconsejar su uso, como la falta de precisión en el cálculo de la reputación. La asignación a un nuevo IDS con un alto grado de reputación, cuando realmente podría ser malicioso en sus comportamientos, podría tener consecuencias significativas.

Reputación certificada

Como alternativa a los tres enfoques anteriores, la *reputación certificada* (del inglés Certified Reputation) plantea un esquema offline sobre cómo propagar la reputación, donde las evaluaciones sobre cada intercambio de alertas las gestiona el propio IDS al que se le va a calcular su reputación, y no el IDS que realiza ese cálculo como se hacía hasta ahora. Esto permite solventar tanto la carga en la gestión de la información, especialmente el almacenamiento, como la propagación de la confianza. La motivación en el uso de la reputación certificada se identifica en [168], donde se aconseja esta fuente en lugar de las otras tres, sobre todo en sistemas abiertos y altamente distribuidos como los sistemas multiagente. Las entidades en estos sistemas actúan de forma autónoma, por lo que suelen ser egoístas en compartir información con otras y tampoco tienen mucha información de su entorno por su carácter distribuido.

Esta nueva fuente de información también tiene una serie de inconvenientes. Sobre todo, el IDS al que se le va a calcular la reputación no puede conocer la satisfacción que han tenido otros IDSs con los que ha intercambiado alertas. Sino, podría descartar las que fueran negativas. Para evitar estas alteraciones, algunos trabajos proponen el uso de la criptografía asimétrica para firmarlas y cifrarlas digitalmente [169]. Ambos procesos permiten preservar la integridad de las evaluaciones, proporcionar un mecanismo de no repudio sobre la existencia de dichas evaluaciones e impedir que el IDS sólo haga uso de las evaluaciones más positivas. Por otro lado, la principal ventaja de los modelos de confianza basados en reputación certificada es que son los más precisos en calcular la reputación, aunque, como se ha comentado antes, se debe tener un rendimiento de computación medio-alto para poder hacer uso de librerías criptográficas. Debido a esto, el uso de esta fuente de información en escenarios móviles puede ser un problema insalvable, si las entidades, como los IDSs, disponen de escasos recursos.

A modo de resumen de todas las fuentes de información analizadas más arriba, la Tabla 2.4 muestra sus características principales.

Características	Experiencias directas	Experiencias indirectas	Reglas basadas en roles	Reputación certificada
Información propia	Sí	No	Sí	Posiblemente
Información externa	No	Sí	No	Sí
Número alto de interacciones	Alta	Media	Ninguna	Media/alta
Precisión en los cálculos	Alta	Media	Baja	Muy alta
Sobrecarga en comunicaciones	Ninguna	Alta	Ninguna	Media
Conocimiento del comportamiento	Sí	Sí	No	Sí
Evaluación de nuevas entidades	No	Posiblemente	Sí	Sí
Adaptación a escenarios móviles	Posiblemente	Posiblemente	Sí	Media/baja

Tabla 2.4: Características de las diferentes fuentes de información

Las experiencias directas y las reglas de confianza basadas en roles no dependen de otros IDSs externos. Es información que el propio IDS que está calculando la reputación puede gestionar internamente, favoreciendo así la precisión sobre el resultado obtenido en ese cálculo y evitando sobrecargas en las comunicaciones. Sin embargo, la inclusión de opiniones externas –experiencias indirectas– puede ayudar a mejorar el cálculo de la reputación con un mayor número de evidencias en el comportamiento pasado del IDS al que se desea evaluar. Por otro lado, en escenarios muy particulares, como los entornos móviles, las reglas de confianza basadas en roles son muy útiles cuando el IDS a evaluar es la primera vez que va a intercambiar alertas en el sistema colaborativo. Con el resto de fuentes no es posible estimar la reputación de un IDS totalmente desconocido, pero podría darse el caso que ese nuevo IDS sí que haya interactuado con otros, pudiendo existir entonces experiencias indirectas para el cálculo de su reputación inicial.

2.4.2. Sistemas colaborativos para la detección de intrusiones basados en confianza y reputación

Los sistemas colaborativos de detección de intrusiones analizados en la Sección 2.1 presuponen que todos los IDSs, así como los dominios de seguridad y/o administrativos, cooperan entre sí de forma honesta: alertando cuando ocurre una amenaza potencial contra las políticas de seguridad del sistema que están monitorizando, o no alertando si el sistema se encuentra en un estado libre de ataques. Sin embargo, esta presunción de honestidad puede conducir al sistema de detección a tener una percepción errónea sobre el estado de seguridad en el que está si sus IDSs exponen comportamientos deshonestos. Por ejemplo, un atacante puede comprometer uno o varios IDSs, haciéndoles que envíen información fraudulenta para provocar errores en los procesos de detección de los otros IDSs, y que así el sistema sea incapaz de detectar un ataque concreto.

Como solución a la identificación de comportamientos maliciosos, en esta sección se han analizado varias propuestas que permiten modelar el comportamiento de cualquier tipo de entidad haciendo uso de los mecanismos de confianza basados en reputación. Sin embargo, todas esas propuestas, como EigenTrust o PeerTrust, centran sus soluciones sobre escenarios muy genéricos que podrían ser aplicables a cualquier tipo de entorno, después de adaptarlas según sus características. En el caso de los sistemas colaborativos de detección de intrusiones, éstos tienen características bastante singulares que podrían hacer inviable el cálculo de la confianza usando esos modelos, si no pudieran gestionar, por ejemplo, las capacidades de detección de los IDSs para evaluar si una alerta es o no verdadera. En este sentido, en los últimos años han aparecido una serie de propuestas con modelos de confianza y reputación aplicables a los sistemas colaborativos para la detección de intrusiones, los cuales se analizan a continuación con más detalle.

Algunos de los sistemas colaborativos para la detección de intrusiones analizados en la Sección 2.3.2, como Indra o ABDIAS, empezaban a reconocer la importancia de gestionar la confianza en sus sistemas para detectar comportamientos maliciosos de sus unidades de detección al generar y compartir falsas alertas. Sin embargo, y a pesar de ese reconocimiento, ninguno de esos sistemas proponía una solución a dicha gestión. En

este sentido, la propuesta presentada en [170] fue una de las primeras en plantear una solución a la gestión de la confianza. En esta solución, cada IDS_i mantiene una lista de *nodos conocidos* con los que ha interactuado en una red overlay P2P creada para la detección de ataques distribuidos. El cálculo de la confianza sobre cada IDS_j, incluido como nodo conocido, se realiza comparando el número de experiencias satisfactorias s_j contra las que han sido desfavorables u_j. Es decir, si ambos IDSs (IDS_i e IDS_j) se han comportado de la misma manera al generar o no la alerta correspondiente. En (2.1) se muestra el cálculo propuesto de la confianza que IDS_i puede tener en IDS_j, según las satisfacciones que IDS_i ha tenido en las alertas recibidas de IDS_j.

$$t_{ij} = w_s \cdot \frac{s_{ij} - u_{ij}}{s_{ij} + u_{ij}}, \text{ siendo } w_s = \begin{cases} \frac{s_{ij}+u_{ij}}{n} & \text{si } n < num\_min\_alertas \\ 1 & \text{en otro caso} \end{cases} \tag{2.1}$$

No obstante, este trabajo presenta varios inconvenientes. En primer lugar, el cálculo de la confianza entre cada par de IDSs solamente utiliza experiencias directas entre ellos, obviando lo que la comunidad piense sobre el comportamiento del IDS que está siendo evaluado. Segundo, y quizá más importante para la detección de intrusiones, es que no tiene en cuenta que los IDSs puedan detectar lo mismo que el IDS que está realizando el cálculo de la confianza. Es decir, los IDSs pueden no estar monitorizando ni las mismas fuentes de información ni las mismas políticas de seguridad. El número de experiencias no satisfactorias siempre irá creciendo después de analizar cada alerta. En último lugar, tampoco hace uso de las particularidades que tienen los sistemas colaborativos para la detección de intrusiones, como las capacidades de detección de los IDSs o parámetros que incluyen las alertas que esos IDSs generan, proponiendo en su lugar una solución muy genérica que podría aplicarse a cualquier otro tipo de entorno.

En la literatura actual se pueden encontrar muchos trabajos que proponen modelos para la gestión de la confianza de forma similar a como lo hace la solución anterior. La gran mayoría se centran en el cálculo de la confianza de entidades desplegadas en *redes inalámbricas de sensores* (del inglés Wireless Sensor Networks), con el único objetivo de detectar comportamientos maliciosos en el intercambio de tablas y actualizaciones de enrutamiento. En este escenario particular, caben destacar los sistemas RADAR [171], ATMP [172] y TSRF [173], aunque, como se ha dicho anteriormente, no consideran las particularidades propias de los sistemas colaborativos para la detección de intrusiones, siendo soluciones muy genéricas que no pueden llegar a extrapolarse a este nuevo ámbito específico de aplicación. En comparación con [170], estos últimos trabajos sólo aportan, con respecto a la gestión de la confianza, el uso de las experiencias indirectas.

Por último, y como principales soluciones que intentan modelar el comportamiento de los IDSs en sistemas colaborativos para la detección de intrusiones, haciendo uso de los modelos de confianza basados en reputación, hacer una mención especial al trabajo de investigación realizado en la tesis doctoral desarrollada en [174]. Esa tesis doctoral, como resultado más destacado del trabajo realizado, se desgrana a continuación con el análisis de sus publicaciones más relevantes, ya que es el único presente en la literatura actual que aborda, aunque con algunas limitaciones, la gestión de la confianza dentro de los sistemas colaborativos para la detección de intrusiones.

En su modelo de confianza, propuesto en [175], cada IDS evalúa la confianza sobre otros IDSs según las experiencias personales con ellos y las obtenidas de otros IDSs de su comunidad. El primer inconveniente reseñable es que el modelo está centrado en el comportamiento de los HIDSs, por su aplicación en la detección de virus, dejando a un lado a los NIDSs en el análisis de todo el tráfico que circula por la red.

La evaluación de la confianza la calcula cada uno de los HIDSs sobre todos aquellos vecinos cercanos geográficamente, que almacena en una lista de conocidos (*acquaintance list*). Sin embargo, la proximidad física entre los distintos NIDSs no puede considerarse como un factor determinante, ya que la detección entre NIDSs cercanos sólo radica en tener que monitorizar las mismas fuentes de información.

La evaluación de la confianza que realiza un HIDS sobre cualquier otro HIDS i, que se propone en [158, 175], se actualiza según las opiniones que recibe de cada uno de los HIDSs de su lista de nodos conocidos. De cada HIDS j se ordenan todas sus opiniones con respecto al tiempo, desde el más reciente al más antiguo en un rango de tiempo t_k. De esta manera, en (2.2) se propone el cálculo de la confianza sobre el HIDS i que se desea evaluar, según todas las opiniones recibidas de cada HIDS j.

$$tw_i^j(n) = \frac{\sum_{k=0}^{n} S_k^{j,i} \cdot F^{t_k}}{\sum_{k=0}^{n} F^{t_k}} \tag{2.2}$$

donde $S_k^{j,i} \in [0,1]$ es la satisfacción que el HIDS evaluador tiene en la opinión k y n el número total de opiniones que ha recibido del HIDS j. Además, también se introduce un factor de olvido F $(0 \leq F \leq 1)$ con el que asignar exponencialmente un menor peso a las opiniones más antiguas frente a las más recientes.

Después de que el HIDS evaluador haya calculado las confianzas parciales sobre el HIDS i que está evaluando, utilizando para ello las opiniones de los HIDSs de su lista de nodos conocidos, el HIDS evaluador las agrega todas siguiendo un método ponderado. Está ponderación se basa en la proximidad física (distancia) entre el HIDS evaluador y cada uno de los HIDSs de los que ha obtenido, al menos, una opinión.

Según (2.2), lo más importante es calcular la satisfacción $S_k^{j,i}$ que el HIDS evaluador puede tener en las opiniones recibidas de cada HIDS j sobre el HIDS i que desea evaluar. Aunque no fue propuesto en los trabajos anteriores, el cálculo de esa satisfacción sí que se define en [176, 177] como (2.3), donde se tiene en cuenta la opinión esperada $r \in [0,1]$, la recibida $a \in [0,1]$ y la dificultad $d \in [0,1]$ sobre las alertas que el HIDS evaluador está preguntando. Esta dificultad se puede estimar por la antigüedad de las firmas de ataque para detectar las alertas que están siendo preguntadas, siendo más alta conforme las firmas sean más recientes para detectar ataques más complejos y actuales.

$$Sat(r,a,d) = \begin{cases} 1 - \left(\dfrac{a-r}{max(c_1 r, 1-r)} \right)^{d/c_2} & a > r \\ 1 - \left(\dfrac{c_1(r-a)}{max(c_1 r, 1-r)} \right)^{d/c_2} & a \leq r \end{cases} \tag{2.3}$$

2.4.3. Confianza inicial de nuevas entidades

Los sistemas que pretenden modelar el comportamiento de una entidad evaluando su confianza, como los IDSs pertenecientes a un sistema colaborativo para la detección de ataques distribuidos a través de los modelos analizados en la Sección 2.4, necesitan de cierta información con la que poder calcular sus valores de reputación, y por ende la confianza sobre cada IDS antes de valorar sus acciones con respecto al intercambio seguro de sus alertas. Esta obligatoriedad en la adquisición previa de información sobre los IDSs obliga a que los modelos de confianza basados en reputación adopten nuevos mecanismos cuando éstos tengan que evaluar la confianza, a través de su reputación, de IDSs totalmente desconocidos sobre los que el sistema carece de información.

La solución más popular en la literatura es dar un valor por defecto a las *entidades recién llegadas* (del inglés Newcomer), que son totalmente desconocidas para el sistema, siendo 0 y 0,5 las reputaciones iniciales más utilizadas cuando $Rep(newcomer) \in [0, 1]$. Un valor de 0 supone que, en el contexto de un sistema colaborativo para la detección de ataques, los nuevos IDSs necesitan una cooperación dilatada con el sistema, con un alto número de intercambios de sus alertas, hasta que se obtenga un valor de reputación lo más real posible. A pesar de ello, este hecho se aventura poco probable en entornos donde se espera la cooperación de IDSs móviles, que pueden estar poco tiempo en un dominio de seguridad. Por otro lado, una reputación de 0,5 es una solución injusta si hay IDSs honestos con una reputación muy cercana a ese valor, pudiendo aparecer ataques *whitewashing* [178], donde un IDS malicioso puede eliminar su *pésimo* comportamiento para obtener una nueva "apariencia" con una reputación más alta que la anterior. En la literatura, hay trabajos que apuestan por un valor nulo para las nuevas entidades [163, 179], y otros que se decantan por una solución neutra de 0,5 [180, 181], aunque todas son muy genéricas en su aplicación y obvian las particularidades que caracterizan a los sistemas colaborativos orientados a la detección de ataques distribuidos.

Existen otros trabajos, como los presentados en [182, 183], donde se propone que el valor inicial de la confianza de una nueva entidad se establezca después de que haya interaccionado X veces con el dominio de seguridad. Sin embargo, este enfoque se antoja poco realista en entornos altamente distribuidos, como en los sistemas colaborativos para la detección de ataques con la participación de IDSs móviles, ya que esas entidades podrían dejar el dominio sin tan siquiera realizar X intercambios de sus alertas.

Además de las soluciones que se han propuesto sobre los sistemas de computación, este problema sobre el cálculo de la confianza inicial de una nueva entidad también ha sido de gran interés en áreas centradas en aspectos sociales y culturales del ser humano. Alguno de estos trabajos puede tener cierta relevancia, con los que obtener ideas que se podrían extrapolar luego a los sistemas colaborativos para la detección de ataques. En [184], por ejemplo, se describe una serie de factores a tener en cuenta cuando nuevos usuarios desean ser aceptados por un grupo de individuos con los que quieren colaborar. Cuanto más similares sean los individuos con el visitante, mejor será la percepción que éstos tengan sobre él. De esta manera, la actitud del grupo hacia ese nuevo usuario se puede ver influenciado por los puntos que se enumeran a continuación.

- La *sinceridad* del nuevo usuario en sus actos y comportamientos, conforme a las normas que el grupo de acogida ya tiene establecidas. En este caso, la reputación de un nuevo IDS puede verse rápidamente decrementada, de forma más acelerada que a otros IDSs de la infraestructura, si se comprueba que sus alertas son falsas alarmas sobre hechos que no han ocurrido en la realidad.
- El *funcionamiento* del grupo de trabajo, ya sea a nivel individual o bien colectivo. Aquí son necesarias las capacidades de detección de los IDSs para comprobar si encajan con la cobertura de la detección que requiere el dominio de seguridad.
- La *similitud* cultural entre el grupo de acogida y el grupo de origen del usuario. Conforme vaya decrementando, las expectativas en el comportamiento del nuevo usuario serán cada vez menores. Por ejemplo, si dos dominios de seguridad desean colaborar para mejorar sus coberturas en la detección de ataques distribuidos, la similitud entre sus capacidades de detección tiene que ser lo más alta posible.
- La *inteligencia cultural* que tiene el usuario a la hora de adaptarse a los cambios que puede sufrir al cambiar de cultura y forma de trabajo. Este factor implica que el usuario debe entender y comprender la nueva cultura, a través del aprendizaje durante las interacciones que realice con los individuos del grupo de acogida.

Además, la aceptación de un usuario por parte de un grupo suele venir dada por las competencias o habilidades que éste tenga a la hora de realizar su trabajo (aceptación del grupo basada en tarea) y su posible iniciativa a la hora de establecer relaciones con el resto del grupo (aceptación del grupo basada en relaciones). Estos dos puntos pueden tener un alto interés en el cálculo de la confianza inicial de los nuevos IDSs que todavía son entidades desconocidas para el sistema colaborativo para la detección de los ataques distribuidos. Primero, las competencias o habilidades de un IDS se corresponderían con sus capacidades de detección, haciendo que los IDSs sean más útiles para el dominio de seguridad en quiere colaborar conforme sean mejores sus capacidades para detectar ataques. En segundo lugar, las relaciones de un IDS se podrían ver como la *voluntad* que tiene en colaborar con el dominio de seguridad al que desea unirse.

Con respecto a la inteligencia cultural, se pueden hacer uso de dos indicadores para el cálculo de la confianza inicial de un nuevo usuario: las experiencias transculturales, con las que habrá aprendido a ser flexible a los cambios culturales con diversos grupos, y su reputación teniendo en cuenta las interacciones anteriores durante esos encuentros transculturales. Estos dos indicadores pueden dar una idea sobre la inteligencia cultural que se puede esperar de cualquier entidad nueva para un grupo de trabajo, pudiendo ser extrapolables al contexto de los sistemas colaborativos para la detección de ataques como las experiencias indirectas que se han analizado en la Sección 2.4.1.

Además de las soluciones y propuestas que se han presentado más arriba, también se pueden encontrar en la literatura otra serie de trabajos que se basan en la *personalidad* o disposición cultural de una entidad, sin necesidad de utilizar sus experiencias previas; es decir, las experiencias directas analizadas en la Sección 2.4.1. Estos trabajos se analizan con más detenimiento en los dos siguientes apartados.

Cálculo de la confianza inicial según la categoría

Existen trabajos en la literatura actual que proponen modelos de confianza basados en *categorías*, o contextos conocidos, para poder inferir los valores iniciales de confianza y/o reputación de nuevas entidades a través de la generalización o categorización de la confianza. A estas entidades se les asigna un contexto en concreto, según los niveles de seguridad que ha establecido el dominio siguiendo sus necesidades [185], asignándoles a las nuevas entidades el valor inicial de la confianza que mejor encaje con el contexto más cercano. Como ejemplo, en [186, 187] se presenta un modelo cognitivo capaz de razonar dinámicamente la confianza de una entidad en términos de categorías, limitándolas a un conjunto de propiedades observables (*Manifesta*) mediante las que inferir propiedades ocultas y capacidades (*Krypta*) que puedan regular el comportamiento de las entidades en el entorno donde se aplica el modelo en cuestión.

En el contexto de estos modelos de confianza basados en las categorías, un trabajo reciente a destacar es el presentado en [188], donde se construye un modelo de confianza orientado a agentes según dos parámetros: i) las características específicas de cada uno de los agentes, pudiendo ser una la voluntad en la participación; y ii) las propiedades de las tareas que los agentes anteriores deben realizar según las características de éstos. Por tanto, existe una dependencia entre ambos parámetros, ya que cada tarea necesitará de un conjunto de recursos y habilidades que los agentes pueden proporcionar. El mapeo entre ambos conjuntos ofrece la base para el proceso general de inferencia a la hora de crear las diferentes categorías. Cómo de similar es un conjunto de agentes, con respecto a un agente individual que acaba de entrar a la red para participar en una tarea, es la base para poder asignar un valor inicial de la confianza a ese nuevo agente.

A pesar de las ventajas expuestas por los trabajos anteriores, el requisito inicial de un conocimiento previo sobre cualquier entidad impide inferir valores de confianza en entornos de total incertidumbre, donde no se tiene información previa sobre otras entidades en el sistema. Por tanto, los trabajos basados en categorías son incapaces de resolver cuestiones como las planteadas en el problema de arranque en frío (*cold-start*). En cambio, al problema *bootstrapping* sí que se le puede dar solución con este enfoque, ya que se puede asignar un valor inicial de la confianza a una nueva entidad a través de las experiencias que otras ya han tenido en el pasado. Es decir, las propuestas donde la inferencia se basa en categorías necesitan de experiencias indirectas para dar valores iniciales de la confianza a entidades sin experiencias previas en el sistema.

Modelos de confianza basados en estereotipos

Una nueva línea de investigación en el cálculo del valor inicial de la confianza de una nueva entidad se centra en el uso de modelos de confianza probabilísticos, en el que las experiencias pasadas con otras entidades se utilizan para generar *estereotipos* [189, 190]. La creación de estos estereotipos se pueden modelar con una función $S : F \rightarrow T$, donde F es un vector de características que define el perfil en el comportamiento que podría adoptar la nueva entidad, en comparación con otras ya conocidas para el sistema; y T la confianza estimada según el estereotipo de esa entidad [191].

En el trabajo presentado en [191], se definen relaciones de confianza entre los agentes que forman un grupo de trabajo con nuevos agentes que desean unirse al mismo. Debido al alto dinamismo en los sistemas multiagentes, los grupos ad-hoc que se forman tienen una duración muy corta en el tiempo, impidiendo que los nuevos agentes alcancen un número mínimo de experiencias con las que el resto, que ya forman parte del grupo, puedan evaluar y establecer sus valores de confianza. En este trabajo, las relaciones de confianza para los nuevos agentes están basadas en esos estereotipos, los cuales, a su vez, dependerán del contexto en el que el agente desea realizar la interacción.

Por otro lado, el sistema StereoTrust [192] se inspira en técnicas de minería de datos para construir los estereotipos mediante vectores de características, al igual que hacen los trabajos anteriores pero utilizando otras técnicas basadas en inteligencia artificial. Debido a ello, este modelo no puede solucionar el problema cold-start, ya que, en este contexto, no se tiene ninguna interacción pasada sobre la entidad a evaluar, pero sí el problema bootstrapping haciendo uso de estos estereotipos.

A pesar de los beneficios que pueda suponer el hacer uso de estereotipos, la principal desventaja de este enfoque es que se necesita información histórica del comportamiento de otras entidades que interaccionaron en el pasado con el sistema, ya sea mediante el uso de experiencias directas con el dominio de seguridad que va a evaluar dicha entidad o mediante experiencias indirectas extraídas de otros dominios de seguridad –dominios con los que mantiene una relación de confianza ya existente– sobre la bondad que la entidad ha tenido en sus interacciones con esos dominios [161, 193]. Estas evidencias externas se han identificado en [194] como una fuente de información potencial para obtener características a partir de las que poder generar estereotipos.

Por último, dejar constancia que todas las propuestas introducidas a lo largo de esta sección definen soluciones bastante genéricas que se pueden desplegar en varios/diversos ámbitos de aplicación, por lo que ninguna de ellas tiene en cuenta las particularidades que caracterizan a los sistemas colaborativos para la detección de ataques distribuidos.

2.5. Maximización de la calidad en la detección

La mayoría de trabajos presentados en las secciones anteriores proponen el diseño y despliegue de sus arquitecturas de seguridad de manera estática, llevados a cabo por un administrador del sistema utilizando, habitualmente, metodologías del análisis del riesgo. Estas metodologías se basan en suposiciones fijas sobre la topología de la red, las amenazas que podrían ejecutar los atacantes, el conjunto de vulnerabilidades de los activos a proteger y la localización y configuración de los IDSs dedicados a detectar esas amenazas, entre muchos otros [195]. Sin embargo, la intervención de un administrador hace que cualquier sistema de detección pierda sus capacidades de adaptación frente a cambios en el entorno de detección. Por ejemplo, cuando aparezcan vulnerabilidades no contempladas antes y que hacen que la configuración de los IDSs deje de ser efectiva frente a los cambios de detección que a partir de ese momento serían necesarios.

A este respecto, en [142] se analizan las características deseables que debería tener un sistema colaborativo para la detección de ataques distribuidos. Aunque este trabajo fue publicado hace tiempo, a continuación se enumeran algunos de los desafíos que se identifican en ese trabajo que, a pesar de los avances que se han conseguido en el ámbito de la detección de ataques, todavía representan cuestiones abiertas en la actualidad y sobre los que los sistemas colaborativos para la detección de ataques distribuidos deben ofrecer soluciones. En particular, un sistema de detección tiene que ser:

- *Resistente frente a la subversión*, monitorizándose a sí mismo para detectar si los procesos de monitorización o detección han sido alterados por un atacante.
- *Configurable* para que sea implementado de la manera más precisa posible con las políticas de seguridad que demanda el dominio de seguridad que va a monitorizar.
- *Adaptable* frente a cualquier cambio que pudiera surgir, tanto en el sistema como en el comportamiento de sus entidades a lo largo del tiempo.
- *Reconfiguración dinámica*, permitiendo al sistema que pueda realizar los cambios que considere más oportunos en la configuración de los IDSs de forma totalmente dinámica y autónoma, sin tener que resetear el sistema completo de detección.

Los cuatro desafíos enumerados en esta lista también hacen posible la maximización de la cobertura en la detección de los ataques, ya que el sistema colaborativo orientado a la detección de ataques distribuidos tiene la potestad de adaptar de manera dinámica la configuración de sus IDSs para hacer frente a cualquiera de los intentos de subversión de sus procesos de monitorización y/o detección. Además, este proceso en la adaptación también permite maximizar la *calidad* de los procesos de toma de decisiones para que la identificación de la ocurrencia de los ataques sea lo más precisa posible, admitiendo exclusivamente las alertas verdaderas que son producidas por los IDSs.

Para la consecución del objetivo anterior, el sistema colaborativo para la detección de ataques distribuidos tiene dos posibles mecanismos que podría implementar [196]: cambiar la localización actual de los IDSs (*reubicación*), moviéndolos de un dominio de seguridad a otro o desplegando nuevos IDSs en un dominio en particular, o configurar los IDSs con otras políticas distintas de monitorización y/o detección (*reconfiguración*). En el fondo, y como se ha comentado al inicio de este capítulo, el objetivo es la búsqueda de un modelo óptimo de despliegue de los IDSs con el que potenciar, en todo lo posible, la escalabilidad, la robustez, la cobertura de la detección y el rendimiento del sistema colaborativo para la detección de ataques distribuidos. Para ello, se puede utilizar uno de los dos mecanismos anteriores, o incluso usar los dos a la vez. En [197] se identifica esta situación claramente, el cual propone resolver el siguiente problema:

> Hallar un modelo de despliegue $PM = \{(IDS_j, Dominio_i)\}$ con el que maximizar la calidad de la monitorización $Q(PM)$ y minimizar su coste de ejecución $Cost(PM)$, siempre y cuando sea válido el despliegue de esa configuración $Valid(PM)$.

Los dos mecanismos que se han comentado más arriba, tanto la reubicación como la reconfiguración, son analizados en detalle en las dos siguientes secciones.

2.5.1. Reubicar los IDSs bajo otro modelo de despliegue

Las estrategias que se pueden encontrar en la literatura actual para la reubicación de cualquier tipo de software, como los IDSs en el contexto de un sistema colaborativo orientado a la detección de ataques distribuidos, se basa en la instalación, configuración y activación remota de cada uno de esos IDSs en los distintos dominios de seguridad donde sean necesarios. Esta reubicación viene definida según el modelo de despliegue que el sistema haya evaluado como el mejor posible, y que cumpla la maximización de $Q(PM)$, la minimización de $Cost(PM)$ y que $Valid(PM) = true$.

La tecnología existente en la actualidad más factible para llevar a cabo un proceso de reubicación es mediante el uso de *agentes móviles*, capaces de empaquetarse a sí mismos para ser desplegados entre los distintos dominios de seguridad. Como ejemplo, el sistema AAFID [142], analizado previamente en la Sección 2.3.2, se puede considerar como una de las primeras soluciones en utilizar agentes autónomos, donde los IDSs se equiparan a esos agentes autónomos, con capacidades de movilidad y que pueden ser configurados en tiempo de ejecución, con los que poder detectar ataques distribuidos. En este trabajo, se defiende que el uso de agentes autónomos permite definir un mejor sistema de detección de los ataques distribuidos, debido a que los IDSs como agentes autónomos van a ser entidades totalmente independientes que se pueden desplegar, eliminar y/o reconfigurar sin que se tengan que alterar el resto de componentes del sistema.

Otro trabajo a destacar es el que se presenta en [198], ya que hace uso de Snort como el software de facto que más se está utilizando en la actualidad para la detección de usos indebidos. En particular, en este trabajo se presenta el sistema DIDMAS (Distributed Intrusion Detection using Mobile Agents and Snort), donde se propone cómo se pueden desplegar y configurar IDSs móviles basados en el uso de Snort.

A pesar de las bondades de los mecanismos para la reubicación de los IDSs, existen otros factores no funcionales que se deben de tener en cuenta. En concreto, los cambios con respecto al despliegue actual de los IDSs, haciendo uso de agentes móviles, conlleva un alto coste en su puesta en marcha que no se puede considerar despreciable.

2.5.2. Reconfigurar las capacidades de detección de los IDSs

Con esta estrategia, al contrario que la reubicación anterior, ya no se realizan tareas de despliegue de forma remota, sino que en cada dominio de seguridad se mantiene un conjunto separado de IDSs y, en un momento en concreto, un número de esos IDSs son configurados y activados adecuadamente según las necesidades de monitorización que requiere el dominio de seguridad. Entonces, cuando es necesario el establecimiento de una nueva configuración para los procesos de monitorización y/o detección, el dominio de seguridad ejecuta las instrucciones necesarias para desplegar esa nueva configuración sobre los IDSs que ya tiene desplegados de antemano.

Esta estrategia es la más sencilla de llevar a cabo hoy en día con las tecnologías actuales. Como ejemplos, en [31, 199] se sugiere la implementación de un *Component Management Interface* (CMI) para gestionar, configurar y activar o desactivar de forma remota los sensores de cualquier tipo de software enfocado a la seguridad, como pueden ser IDSs o firewalls. La gestión de un CMI suele hacer referencia a una API que ofrece un modo de comunicación con el servicio que está gestionando el software en particular. Esa API le va a permitir al sistema tener los permisos necesarios para el acceso, gestión y control de todos esos servicios. Para el desarrollo de una API enfocada a la gestión de los servicios, se han propuesto varios estándares como *Web-Based Enterprise Management* (WBEM) [200] o *Web Services Distributed Management* (WSDM) [201].

Otra manera sencilla de lanzar el proceso de reconfiguración es la que se presenta en [202], donde se propone la definición de reglas ECA (Event, Condition, and Action) para la reconfiguración cuando los IDSs alertan sobre un evento. Como ejemplo, ese trabajo presenta el siguiente comando para reconfigurar una *blacklist*, gestionada por un firewall, con el que poder incluir en dicha lista la IP de origen (`SrcIpAddr`) de un atacante cuando se recibe una alerta generada por un IDS con esa dirección.

on IDSAlert if BlackList !Contains SrcIpAddr do AddSrcIpAddr in BlackList

A pesar de la sencillez en el uso de comandos, este tipo de soluciones carece de total dinamicidad en la reconfiguración de las capacidades de detección de los IDSs.

La principal ventaja de esta estrategia es su baja sobrecarga en términos de ancho de banda en las comunicaciones, haciendo por tanto que la reconfiguración suponga un bajo coste para su ejecución. Sin embargo, esta estrategia obliga a que cada dominio de seguridad mantenga un conjunto de IDSs disponibles, que en ciertas ocasiones podrían no ser suficientes ya que el dominio podría requerir un mayor número de IDSs con los que alcanzar el estado de monitorización y/o detección deseado.

2.5.3. Puesta en marcha de un nuevo modelo de despliegue

En la mayoría de los trabajos que se han analizado en las dos secciones anteriores, donde se han presentado varias soluciones según los dos mecanismos para la puesta en marcha de un modelo de despliegue de los IDSs, se proponen ciertos métodos sobre *cómo* se tienen que llevar a cabo la ejecución de los procesos de reubicación o reconfiguración de los IDSs. Pero, en muy pocas ocasiones, se define *cuándo* es el mejor momento en el que el sistema tiene que seleccionar un nuevo modelo de despliegue de sus IDSs, a fin de poder maximizar la calidad en los procesos de monitorización y/o detección. Es decir, no definen cuándo calcular la función $Q(PM)$ y, dependiendo de su resultado, tomar la decisión de si lanzar o no los procesos de reubicación y/o reconfiguración de los IDSs y con *qué* configuración en concreto tienen que hacerlo.

En [31], por ejemplo, se reconoce la posibilidad de usar enfoques basados en técnicas automáticas de optimización para obtener un nuevo modelo de despliegue estratégico de los IDSs, debido fundamentalmente al gran número de configuraciones que se podrían generar en un sistema colaborativo para la detección de ataques distribuidos.

Como ejemplo donde se utilizan algoritmos de optimización, en [197] se presenta una solución para evaluar en rendimiento las políticas de reconfiguración de un conjunto de servicios para *Redes Móviles Ad-Hoc* (del inglés Mobile Ad Hoc Network, MANET). El objetivo de estas políticas es definir qué, cuándo y dónde ubicar los servicios, aplicando para ello técnicas de programación genética y optimización multiobjetivo. De manera más formal, como se propone en este trabajo, un servicio s se reubicará en otro servidor $j = arg\ max_i\{f_i\}$ si, y solo si, $f_j > f_s$. Esa función $f_i = f(m_1^i, ..., m_d^i)$ se obtiene a partir de un conjunto de indicadores o métricas, denotadas como m, que periódicamente se recuperan de cada servidor en el que el servicio s se podría desplegar. De esta manera, cuanto más alto sea el valor de f_i, más adecuado será ese servidor para s.

Los mismos autores del trabajo anterior, proponen luego en [203] una solución para la configuración y el despliegue de un conjunto de NIDSs en una gran *Red de Sensores Inalámbricos* (del inglés Wireless Sensor Network, WSN). En este nuevo trabajo se sigue una metodología de búsqueda heurística, utilizando para ello un algoritmo Simulated Annealing [204], con el que encontrar el mejor despliegue posible de un conjunto de NIDSs para maximizar la precisión en los procesos de detección y minimizar el número de recursos computacionales, fundamental en las redes WSN. Este trabajo pretende dar una solución sobre dónde desplegar los NIDSs, y con qué configuración, pero en cambio no hace alusión a cuándo se tiene que lanzar ese proceso de búsqueda. Otros trabajos relacionados hacen uso de los algoritmos genéticos, como los presentados en [205, 206]. Sin embargo, siguen exponiendo los mismos problemas que los anteriores: la falta en el establecimiento de un momento en el tiempo (*cuándo*) en el que lanzar la búsqueda heurística para la obtención del mejor modelo de despliegue posible.

Más allá de los mecanismos de optimización anteriores, otros trabajos como los presentados en [207, 208] proponen el uso de grafos de ataque para descubrir el modelo óptimo de despliegue de los IDSs. El propósito es identificar todos los caminos posibles dentro del grafo de ataque que conduzcan al compromiso de los activos que se desean proteger y, como resultado, seleccionar el mejor modelo de despliegue para monitorizar cada uno de los caminos del grafo con el menor número posible de IDSs. Este enfoque sería muy interesante para desplegar un mayor número de IDSs, o el mismo número pero con mejores capacidades de detección, conforme el sistema advierta que el atacante se encuentra más cerca de su objetivo, aunque está opción no está contemplada en estos trabajos. Como principal inconveniente en el uso de grafos de ataque, se puede constatar que estas soluciones sólo se han definido para la detección de ataques conocidos, donde solamente actúan los IDSs basados en la detección de usos indebidos.

Por último, comentar que también se podrían utilizar modelos de confianza, como los basados en reputación de la Sección 2.4, con los que establecer un nuevo modelo de despliegue según el comportamiento de cada IDS, identificado por los valores de su reputación. En concreto, la diversidad en sus comportamientos se ha identificado en la literatura actual como un factor clave para optimizar la cooperación entre los miembros de un grupo. La mayoría de trabajos en esta área están relacionados con disciplinas bien conocidas como las socioeconómicas, centradas principalmente en tratar con personas que actúan como fuentes de información. En general, cada una de estas fuentes se puede

modelar por un número de características, como la edad, el género, el salario, la posición en el trabajo o el rol que tiene dicha persona en la organización. Como ejemplo de estos trabajos, en [209, 210] se presentan dos modelos para medir el impacto de la diversidad cuando se establecen grupos de personas con el propósito de desarrollar proyectos sobre sistemas de información. Naturalmente, otros factores distintos a la diversidad también se consideran en estos trabajos para establecer un grupo de trabajo, como los posibles conflictos de intereses, dificultades en las relaciones personales, la comunicación entre los miembros y las contribuciones que se esperan de cada uno de ellos.

En el contexto de los sistemas de detección de intrusiones, hay unos pocos trabajos, como los que se presentan en [211, 212], que hacen uso del concepto de la diversidad en las capacidades detección de los IDSs, a fin de incrementar la robustez en los procesos de detección. Por ejemplo, utilizando diferentes técnicas o configuraciones de detección para monitorizar el mismo conjunto de requisitos de seguridad. A pesar de ello, ninguna de estas soluciones aplica la diversidad con el objetivo de maximizar la calidad en los procesos de monitorización y/o detección de los IDSs.

2.6. Conclusiones del capítulo

Los sistemas colaborativos de detección de intrusiones siguen enfrentándose a día de hoy ante un número elocuente de retos, analizados al comienzo de este capítulo, a los que se les debe dar solución para lograr mejores mecanismos capaces de detectar ataques en tiempo real, donde se tiene que analizar gran cantidad de información. Principalmente, las alertas que cada IDS pueda generar por separado haciendo referencia a los incidentes acontecidos en sus ámbitos limitados de detección. Los procesos de detección para estos sistemas necesitan de la estrecha colaboración entre múltiples dominios de seguridad, a fin de que la detección de ataques distribuidos alcance el éxito esperado.

A pesar de los esfuerzos que se han hecho en los últimos años, y que se han analizado en este capítulo, todas las propuestas se centran, principalmente, sobre los procesos de detección, proponiendo nuevos algoritmos de detección, con los que obtener un mejor ratio y precisión en la detección de ataques, y nuevos modelos de despliegue de los IDSs, para la detección de ataques distribuidos. Sin embargo, todas las propuestas anteriores siguen desestimando la propia gestión del sistema de detección en sí mismo.

En la gestión de un sistema colaborativo para la detección de ataques distribuidos, la protección de la información que los IDSs tienen que analizar (sobre todo, las alertas) se erige como un reto indispensable al que se le debe dar solución, de ahí su adopción como uno de los objetivos principales para esta tesis doctoral. Este reto se ha identificado en este capítulo como *seguridad de las herramientas de detección*. Las alertas que los IDSs necesitan intercambiar, para la detección de ataques distribuidos, se tienen que proteger frente a amenazas que intenten alterar su contenido (*confidencialidad* e *integridad*), o que sean enviadas por IDSs no legítimos del sistema (*autenticidad*). Estos IDSs también deben estar seguros de que las alertas recibidas de otros IDSs sean hechos ocurridos en la realidad (*confianza*), y no falsas alertas debido a un comportamiento malicioso.

La protección en el intercambio de las alertas se basa, como propuesta en el contexto de esta tesis doctoral, en criptografía de clave pública. Esta tecnología se ha utilizado, y aún se sigue usando, como primera línea de defensa i) frente a alteraciones que puedan sufrir las alertas intercambiadas entre los IDSs y ii) frente a la correcta identificación de esos IDSs como actores legítimos del sistema colaborativo para la detección de ataques distribuidos. Sin embargo, las propuestas en la literatura que han sido revisadas en este capítulo, sólo se centran en mecanismos internos de seguridad para un único dominio, sin considerar que los ataques distribuidos van más allá de las fronteras del dominio que se está protegiendo. Como respuesta, el IETF ha definido un conjunto de estándares que permiten establecer relaciones de confianza entre más de un dominio de seguridad, bajo modelos avanzados de certificación cruzada, aunque ni el IETF ni ningún organismo de estandarización ha definido cómo realizar la construcción de una federación de PKIs a fin de dar soporte a este nuevo tipo de escenarios multidominio.

Durante la creación de una federación de PKIs, la definición de los certificados X.509 tienen que permitir la correcta interoperabilidad entre todos sus dominios de seguridad. Además, también es necesario que cada dominio de seguridad ofrezca soporte, a través de un Servicio de Validación, a la construcción y validación de caminos de certificación. Este servicio debe permitir que un IDS pueda validar el material criptográfico de otros IDSs (certificados X.509), dentro de un entorno multidominio, antes de que entre ellos puedan intercambiar sus alertas de detección de forma segura. A este respecto, el IETF sólo ofrece guías y recomendaciones que pueden usarse en un amplio rango de entornos de certificación, aunque están lejos de especificar cómo se tiene que definir un Servicio de Validación que pueda utilizarse en escenarios multidominio complejos.

Aun considerando obligatorio el uso de la criptografía de clave pública en los IDSs, como primer mecanismo de confianza que pudieran utilizar esas unidades de detección, su autenticación no implica que vayan a mostrar un buen comportamiento en sus actos, suponiendo siempre que van a enviar alertas que representen incidentes ocurridos en la realidad. Los IDSs desplegados en un sistema colaborativo para la detección de ataques distribuidos deben tener el total convencimiento de que las alertas que son generadas y compartidas por el resto de IDSs no sean fraudulentas, generadas como consecuencia de haberse visto comprometidos por un atacante con intenciones de alterar el correcto funcionamiento del sistema de detección. En caso contrario, si esas alertas son realmente fraudulentas, sus procesos de detección verían mermada su percepción sobre el estado de seguridad real en el que se encuentran los activos que están monitorizando. La gestión de la confianza de los IDSs emisores de alertas, a fin de identificar los comportamientos maliciosos de los IDSs y, por ende, la posible generación de alertas fraudulentas, está en total sintonía con uno de los principales retos identificados al comienzo de este capítulo: la reducción del número de falsos positivos, especialmente las alertas fraudulentas que son generadas por IDSs maliciosos en su comportamiento.

Los sistemas de gestión de la confianza basados en reputación han demostrado ser los mecanismos más precisos en modelar el comportamiento de cualquier entidad, pudiendo por ello ser usados para la identificación de los comportamientos maliciosos que pudiera tener cualquier IDS. Estos sistemas de confianza también proponen soluciones para el

cálculo de la reputación inicial que tiene que asignarse a un nuevo IDS que desea unirse por primera vez al sistema colaborativo para la detección de los ataques distribuidos. Pero, a pesar de todo ello, las soluciones propuestas en la literatura, analizadas en este capítulo, son muy genéricas en su aplicación, sin considerar las características propias que tienen los IDSs en sus comportamientos como unidades de detección. Por ejemplo, estas soluciones no contemplan la gestión de las capacidades de detección que los IDSs tienen implementados en sus procesos internos de detección, y que son obligatorios a la hora de evaluar la satisfacción de las alertas generadas por otros IDSs.

Por último, resaltar que los trabajos que se han analizado a lo largo de este capítulo no ofrecen respuesta a cuándo es el mejor momento para reajustar el sistema y poder maximizar la calidad en sus procesos de monitorización y/o detección. Este hecho se daría porque, por ejemplo, podrían haber áreas de detección que no están correctamente protegidas si existen IDSs desplegados con una baja reputación –posible indicio de un mal comportamiento. En este contexto, los sistemas colaborativos para la detección de ataques distribuidos pueden usar los modelos de confianza basados en reputación para maximizar la confianza en sus IDSs, reubicándolos o reconfigurándolos estratégicamente a fin de mejorar la cobertura y la precisión de la detección de ataques distribuidos de manera sustancial, dando una posible solución a cuándo llevar a cabo todo este proceso y qué nueva configuración en particular se tendría que desplegar.

Capítulo 3

Gestión de la confianza basada en PKI para entornos multidominio

Los sistemas desplegados de forma estratégica en una red orientada a la detección de ataques distribuidos requieren de un cierto nivel de seguridad en el intercambio de información. Sobre todo, en el intercambio de incidentes que cada *Sistema de Detección de Intrusiones* (del inglés Intrusion Detection System, IDS) puede percibir de manera individual, alertando sobre actividades maliciosas en los activos que está protegiendo. Este proceso sobre el intercambio de las alertas se ha visto forzado a tener que adaptarse a nuevos ámbitos de detección hacia ataques distribuidos. Se ha pasado en los últimos años de un escenario donde los mecanismos de seguridad intradominio –dentro de una misma organización– eran fácilmente implantados, mediante el despliegue, por ejemplo, de soluciones actuales como las *Infraestructuras de Clave Pública* (del inglés Public Key Infrastructure, PKI), a un nuevo paradigma donde las organizaciones deben compartir alertas de forma segura. Con el soporte de estas infraestructuras de seguridad, cualquier entidad de un sistema de detección puede autenticarse y compartir información (alertas) de forma segura utilizando sus certificados de clave pública [24, 25].

Sin la ayuda de infraestructuras de seguridad como aquellas basadas en soluciones de PKI, la compartición de alertas, especialmente en escenarios multidominio, se podría ver comprometida si fueran alteradas durante su transmisión. O incluso, si estas alertas fueran de alta criticidad, indicando un alto impacto sobre los activos, y fueran enviadas por IDSs no autenticados como entidades legítimas del sistema colaborativo orientado a la detección de ataques. Esta perturbación haría que la percepción de los IDSs sobre el estado de seguridad en el que se encuentra el sistema de detección se viera totalmente comprometido, y por ende todo el sistema que se está protegiendo.

Estos hechos han propiciado que las Infraestructuras de Clave Pública (PKI) hayan tenido que adaptarse con nuevas estructuras de certificación más complejas, en aras de reflejar los cambios que requieren los sistemas descentralizados como el anterior [213]. Estas nuevas estructuras tienen que basarse en modelos de confianza más avanzados, en contraposición a los modelos que existían previamente donde la gestión de la confianza siempre estaba controlada bajo un mismo dominio de administración.

Estos cambios han abierto nuevas posibilidades y retos, para un correcto despliegue de las tecnologías que subsanen los problemas que originan los escenarios más avanzados que se pueden encontrar en la actualidad. Con respecto a la adaptación de las actuales PKIs, aparecen una serie de cuestiones que deben abordarse en profundidad:

- ¿Qué modelo de confianza se tendría que desplegar en una red multidominio de comunicaciones? O más concretamente, ¿cuál se ajustaría mejor a los requisitos que se establecen en un sistema colaborativo orientado a la detección de ataques distribuidos, con una total descentralización de la información?
- ¿Cuál es la mejor opción en términos de rendimiento?
- ¿Cómo un usuario, dispositivo o proceso software como un IDS es capaz de validar que el certificado de clave pública de cualquier otra entidad (a partir de ahora, simplemente *certificado*) es confiable para el establecimiento de una comunicación segura entre ellas, sabiendo que pertenecen a organizaciones distintas?
- ¿Cuál es el impacto que tiene un proceso de validación en una infraestructura de confianza multidominio, como los basados en la detección de ataques distribuidos?

En la actualidad, no existe ningún acuerdo entre las organizaciones e instituciones de estandarización para dar soluciones a las cuestiones anteriores. Sólo sugieren una serie de recomendaciones a modo de guías prácticas. Debido a ello, el objetivo principal que se plantea en este capítulo es doble. En primer lugar, se pretende promocionar la adaptación de las PKIs actuales hacia modelos de confianza más avanzados con los que los nuevos entornos multidominio, como los sistemas colaborativos para la detección de ataques distribuidos, puedan proteger sus comunicaciones en el intercambio de alertas. Por otro lado, en este capítulo también se presenta el diseño de un Servicio de Validación capaz de construir y validar caminos de certificación en cualquier tipo de infraestructura de confianza, ya sea en entornos intradominio o interdominio, a fin de comprobar que la comunicación entre los IDSs se va a realizar de forma segura.

La metodología que se plantea en este capítulo consiste en, primero, analizar los requerimientos mínimos que sirven como base para la construcción de una federación de PKIs. Estas exigencias posibilitan una correcta interoperabilidad entre los distintos dominios de seguridad, incluyendo sus IDSs localmente desplegados, que deseen formar parte de una red de confianza bajo un entorno multidominio. Como resultado de este estudio, se propone una serie de requisitos que deben cumplir los dominios implicados en una federación de PKIs, tanto en su fase de construcción como en la definición de su modelo de confianza para conseguir la correcta interoperabilidad deseada.

Por último, se procede a describir tanto el diseño de un algoritmo de construcción y validación de caminos de certificación como el despliegue de una federación de PKIs en un escenario multidominio. Este escenario se utiliza posteriormente como entorno de pruebas para ejecutar una serie de experimentos con los que analizar el impacto que tiene el algoritmo propuesto en una infraestructura de confianza multidominio, como los sistemas colaborativos para la detección de ataques distribuidos.

3.1. Extensiones de los certificados X.509

Cada organización se regirá por sus propias políticas internas, que se tienen que ver reflejadas en las extensiones de todos los certificados que gestiona su PKI. La adecuada definición de esas extensiones es un requisito clave para el éxito en la construcción de una correcta interoperabilidad entre los distintos dominios de seguridad que conforman un sistema colaborativo para la detección de ataques distribuidos, así como en el éxito de cualquier algoritmo de construcción y validación de caminos de certificación.

Con la correcta consecución de los requisitos anteriores, los sistemas de detección van a poder comprobar si sus materiales criptográficos son confiables antes de establecer un canal de seguro para el intercambio de alertas de detección. La no inclusión o errores en la definición de alguna de las extensiones, que en el contexto de esta tesis doctoral se consideran como obligatorias, podría hacer inefectiva la interoperabilidad entre varios dominios de seguridad. Este hecho haría que, por ejemplo, el algoritmo de construcción y validación fuera incapaz de descubrir, y luego validar, un camino de certificación entre dos o más organizaciones con acuerdos de confianza válidos y en vigor.

A continuación se analizan en profundidad las 17 posibles extensiones que puede contener un certificado X.509, así como los requisitos indispensables para una correcta interoperabilidad entre los diferentes dominios de seguridad implicados. Para ello, se han estudiado estos requisitos para los cuatro tipos de certificados que pueden aparecer en escenarios multidominio: certificados cruzados, certificados de CA raíz (incluyendo la BCA), certificados de CA subordinada y certificados de entidades finales. Este análisis se ha realizado teniendo en cuenta las recomendaciones que ha proporcionado el grupo de trabajo sobre PKIX del IETF en [24, 104]; otros trabajos relacionados como [102, 103, 214]; la guía de requisitos técnicos publicada por la agencia federal *General Services Administration* (GSA), con la que identificar y resolver los problemas de compatibilidad e interoperabilidad que las nuevas CAs deben seguir para que puedan adherirse a la FBCA como un nuevo miembro del mismo [215]; y la propia experiencia a través de la implantación y experimentación de este trabajo en diversos proyectos de investigación, como los llevados a cabo en SEINIT, DAIDALOS y MISTRAL, entre otros.

La Tabla 3.1 detalla los requisitos asociados a cada una de las 17 extensiones de un certificado. Según el tipo de certificado, la extensión puede variar entre obligatorio, recomendado, opcional y no aplicable (representado por un guion). Para las extensiones obligatorias o recomendadas, su aplicación se basa en la correcta especificación según el estándar X.509 [24] o también para el correcto funcionamiento de cualquier algoritmo de construcción y validación de caminos de certificación. El resto de extensiones, aunque no sean obligatorias o recomendadas, podrían ser útiles para mecanismos de optimización que ayudasen a mejorar el rendimiento del algoritmo. En la última columna de la tabla, se especifica a qué fase del algoritmo está relacionada cada extensión: C para la fase de construcción, V para la fase de validación o $C+V$ para ambas.

A continuación se explican los requisitos para cada una de las extensiones marcadas en la Tabla 3.1 como obligatorias o recomendadas para alguno de los tipos de certificado. La descripción completa, y su forma de uso, se pueden encontrar en [24].

Extensión X.509	Certificado cruzado	CA raíz / BCA	CA subordinada	Entidad final	
AuthorityKeyIdentifier	Obligatorio	Opcional	Obligatorio	Recomendado	C
SubjectKeyIdentifier	Obligatorio	Obligatorio	Obligatorio	Opcional	C
KeyUsage	Obligatorio	Obligatorio	Obligatorio	Recomendado	C+V
CertificatePolicies	Recomendado	Recomendado	Recomendado	Recomendado	C+V
PolicyMappings	Recomendado	Opcional	Opcional	-	C+V
SubjectAlternativeName	Opcional	Opcional	Opcional	Opcional	-
IssuerAlternativeName	Opcional	Opcional	Opcional	Opcional	-
SubjectDirectoryAttributes	Opcional	Opcional	Opcional	Opcional	-
BasicConstraints	Obligatorio	Obligatorio	Obligatorio	-	C+V
NameConstraints	Opcional	Recomendado	Opcional	-	C
PolicyConstraints	Opcional	Opcional	Opcional	Opcional	C+V
ExtendedKeyUsage	-	-	-	Opcional	V
CRLDistributionPoints	Recomendado	Recomendado	Recomendado	Recomendado	V
InhibitAnyPolicy	Opcional	Opcional	Opcional	-	C
FreshestCRL	Opcional	Opcional	Opcional	Opcional	V
AuthorityInfoAccess	Obligatorio	Opcional	Obligatorio	Obligatorio	C+V
SubjectInfoAccess	Recomendado	Obligatorio	Obligatorio	Obligatorio	C+V

donde, en la última columna: *C* hace referencia a la fase de construcción y *V* a la fase de validación

Tabla 3.1: Requisitos en las extensiones de los certificados X.509

3.1.1. AuthorityKeyIdentifier y SubjectKeyIdentifier

El valor en la extensión `AuthorityKeyIdentifier` de un certificado debe contener el mismo identificador que el del certificado de su CA emisora en `SubjectKeyIdentifier`. Así se puede definir una forma de concatenar los certificados según estos identificadores, muy útil durante la fase de construcción. Ambas son idóneas también en certificados de CA cuando poseen más de una clave privada para generar y firmar otros certificados.

Las dos extensiones también son obligatorias cuando el algoritmo debe seleccionar el próximo certificado (entre varios) para el camino de certificación que está construyendo, y así poder evitar el análisis de caminos que no sean candidatos hasta los Trust Anchors. Son obligatorias para los certificados cruzados y certificados de CA, siendo sólo opcional la extensión `AuthorityKeyIdentifier` en los certificados de CA raíz o BCA, ya que no fueron emitidos por otras autoridades de nivel superior y podrían no incluirla.

3.1.2. KeyUsage

Todos los certificados cruzados y los certificados de CA deben definir esta extensión. Si no se define, o sí se hace pero sin incluir el bit `keyCertSign`, el camino de certificación candidato se podría eliminar al no ser válido. Así se evitaría la construcción de caminos de certificación inválidos con certificados con un uso indebido de esta extensión.

3.1.3. CertificatePolicies

Es altamente recomendada para todos los tipos de certificado a fin de conocer bajo qué políticas de seguridad han sido generados. No es obligatoria ya que el procesamiento de estas políticas, la mayoría de ellas definidas en [24], sólo tienen efecto en los dominios locales donde se definen. Debido a ello, esta extensión normalmente se establece al valor `anyPolicy` en entornos multidominio bajo un modelo de certificación cruzada.

3.1.4. PolicyMappings

Utilizada comúnmente en los certificados donde se definen las relaciones de confianza multidominio para formar una federación de PKIs: entre las CAs raíces independientes que quieren establecer una relación peer-to-peer y en los certificados cruzados entre una BCA y las CAs raíces implicadas. A través de esta extensión, los certificados cruzados pueden establecer la equivalencia entre las políticas de certificación que cada dominio de seguridad implementa. Por ello, esta extensión se considera muy recomendada, aunque no obligatoria, ya que podría no ser necesaria ninguna equivalencia entre ellas.

3.1.5. BasicConstraints

Los certificados de CA están obligados a incluir esta extensión. En caso contrario, o si el elemento `cA` es *false*, el camino de certificación candidato será clasificado luego como inválido. Esta extensión también se puede usar durante la fase de construcción, ya que puede incluir el elemento `pathLenConstraint` indicando el número de entidades intermedias que pueden seguir a un certificado de CA raíz. O lo que es lo mismo, cómo de "lejos" está un camino de certificación hasta alcanzar el límite de su longitud.

3.1.6. NameConstraints

Aunque no sea obligatoria, sí es aconsejable al poder ser definida en los certificados cruzados emitidos por una BCA a las CAs raíces para asegurar que esas últimas siempre emitirán certificados bajo un `Name` específico. En sentido contrario, entre las CAs raíces y la BCA, permite excluir a los dominios de seguridad en los que la CA correspondiente no confía. Esta extensión, por tanto, es muy útil durante la construcción de los caminos de certificación, ya que evita tener que analizar certificados pertenecientes a dominios de seguridad que, según esta extensión, están excluidos como confiables.

3.1.7. CRLDistributionPoints

Esta extensión identifica cómo y desde dónde obtener las listas de revocación para conocer si el certificado que incluye esta extensión está revocado o no. Dependiendo del tipo de certificado, esta comprobación se tiene que realizar a través de una CRL, para los certificados de entidades finales como un IDS, o una ARL, para los certificados que representan autoridades de confianza (CAs, BCAs o certificados cruzados).

Esta extensión es altamente recomendada, aunque no obligatoria, para asegurar el proceso de validación, ya que el cliente de un Servicio de Validación podría solicitar sólo el proceso de construcción de un camino de certificación. Además, tampoco puede considerarse como obligatoria debido a que el proceso de validación podría recurrir a otros mecanismos a las listas de revocación como, por ejemplo, OCSP.

3.1.8. AuthorityInfoAccess y SubjectInfoAccess

Ambas extensiones son obligatorias, o altamente recomendadas, en todos los tipos de certificado para que un algoritmo de construcción y validación pueda recuperar todo el material criptográfico que es necesario para su ejecución. El estándar definido en [104] también aconseja la extensión `AuthorityInfoAccess`, con la intención de proporcionar la usabilidad e interoperabilidad con muchas de las PKIs existentes.

Estas dos extensiones definen un punto de contacto donde se especifica: i) a través del método de acceso `id-ad-ocsp`, cómo y dónde hacer las consultas OCSP para todos los certificados del dominio de seguridad; y ii) la dirección del Servicio de Directorio, habitualmente la URL de un servidor LDAP, desde donde se pueden recuperar todos los certificados asociados a la entidad que define la extensión. En este último punto, se utiliza el método de acceso `id-ad-caIssuers` para la extensión `AuthorityInfoAccess` e `id-ad-caRepository` para `SubjectInfoAccess`.

Todos los objetos y atributos que puede almacenar un Servicio de Directorio se han descrito anteriormente en la Sección 2.2.4.

3.2. Definición de un modelo de confianza para una federación de PKIs

En esta sección se desarrolla el modelo de confianza diseñado y desplegado en un entorno de laboratorio, con el doble objetivo introducido en los apartados anteriores. Por un lado, la construcción de una infraestructura multidominio –federación de PKIs– bajo un modelo de certificación cruzada mediante una BCA. Esta federación de PKIs va a proporcionar el marco de seguridad necesario para que el intercambio de alertas entre los IDSs de un sistema colaborativo para la detección de ataques distribuidos se realice de forma segura: autenticándolos como entidades legítimas del sistema y protegiendo sus comunicaciones frente a posibles alteraciones de su contenido.

Como segundo objetivo, esta sección también presenta más adelante el diseño de un algoritmo de construcción y validación de caminos de certificación. Cualquier Servicio de Validación tiene que implementar este algoritmo, con el objetivo de que sus entidades puedan avalar un cierto nivel de confianza antes de que establezcan una comunicación segura con cualquier otra entidad, a través del certificado de este último. En el contexto de un sistema colaborativo para la detección de ataques distribuidos, la validación de los certificados de los IDSs va a permitir el establecimiento de canales seguros y confiables entre ellos para que el intercambio de alertas de detección tenga el éxito esperado.

3.2.1. Construcción de una federación de PKIs

En la construcción de la federación de PKIs, se ha hecho uso del software *Public Key Infrastructure with IPv6 support* (UMU-PKIv6) [216, 217, 218]. Este software ha sido desarrollado en el Departamento de Ingeniería de la Información y las Comunicaciones de la Universidad de Murcia, con el que poder obtener servicios básicos de certificación: emisión, renovación y revocación de certificados de clave pública. Entre los servicios avanzados, la UMU-PKIv6 ofrece funcionalidad de Sellado de Tiempo y la validación online de certificados con OCSP, además de dar soporte a todos sus componentes en entornos de red IPv4 e IPv6. También permite la publicación de los certificados de CA, certificados de entidad finales y listas de revocación (CRLs y ARLs) tanto en servidores LDAP como en repositorios públicos basados en DNSSEC.

La solución UMU-PKIv6 solamente soporta la definición y despliegue de modelos de certificación simple y jerárquico, analizados en detalle en la Sección 2.2.1. Debido a que este software de PKI no ofrece soporte para la certificación cruzada, obligatoria para el despliegue de modelos de certificación peer-to-peer o mediante una BCA, en el desarrollo de esta tesis doctoral se ha ampliado esta funcionalidad para incluir en la UMU-PKIv6 el soporte de este tipo de certificados, necesarios para el despliegue de los sistemas colaborativos para la detección de ataques distribuidos.

La adaptación de la UMU-PKIv6 a nuevos modelos de certificación cruzada también abre la necesidad de tener que ofrecer un mecanismo que sea capaz de construir y validar caminos de certificación dentro de los nuevos escenarios multidominio. Esta ampliación en la UMU-PKIv6 ha supuesto la creación y gestión de los certificados cruzados, para los que se han seguido los requisitos en las extensiones de los certificados presentados en la Sección 3.1. Con respecto a estas extensiones, a continuación se enumeran las que son necesarias, tanto para la gestión de certificados cruzados como para la creación de un algoritmo de construcción y validación de caminos de certificación. Algunas de esas extensiones ya las soportaba la UMU-PKIv6, pero otras se han tenido que definir para la consecución de los objetivos de esta tesis doctoral.

- Las extensiones `AuthorityKeyIdentifier`, `SubjectKeyIdentifier`, `KeyUsage` y `BasicConstraints` para ayudar al Servicio de Validación a decidir entre varias alternativas cuando existan varios caminos posibles de certificación.
- `AuthorityInfoAccess` con la que recuperar, desde los certificados cruzados y los certificados de las entidades finales, la información sobre la localización de los servicios de validación: CRL/ARL y OCSP.
- `NameConstraints` para excluir todos aquellos caminos de certificación que no son confiables y, por tanto, no válidos.

En cada una de las CAs, incluyendo la BCA como una autoridad de certificación más en la federación de PKIs, se ha desplegado una copia de la UMU-PKIv6 personalizada a su propio dominio de seguridad con los siguientes servicios. Este conjunto de servicios, además del Servicio de Directorio, se han analizado en detalle en la Sección 2.2.4.

- *Servidores OCSP*. Este servicio se ha implementado y desplegado en cada una de las CAs siguiendo su protocolo estándar, el cual fue definido por el grupo de trabajo PKIX del IETF en [116].
- *Servicio de Validación*. Este servicio lo ofrecen todas las CAs de la federación de PKIs a sus entidades para la construcción y validación de caminos de certificación. Soporta el protocolo SCVP [118], requerido para una correcta comunicación entre las entidades solicitantes (por ejemplo, los IDSs) y el Servicio de Validación.

Se ha escogido el protocolo estándar SCVP, en lugar de otros protocolos también analizados en la Sección 2.2.3, al ser el único que permite la construcción y validación de caminos de certificación en escenarios multidominio de certificación cruzada, como el que a continuación se describe de forma detallada. Esta funcionalidad es un requisito obligatorio para que el intercambio de alertas entre los IDSs de un sistema colaborativo para la detección de ataques distribuidos se lleve a cabo de forma segura.

3.2.2. Diseño de un algoritmo de construcción y validación de caminos de certificación

Todo el material que necesita el algoritmo de construcción y validación de caminos de certificación viene definido en la solicitud SCVP que recibe el Servicio de Validación de la entidad solicitante, ya sea ésta un usuario final o un proceso software como un IDS. En el Algoritmo 1 se muestran los pasos en pseudocódigo que el Servicio de Validación tiene que realizar antes, y después, de llamar al algoritmo de construcción y validación de caminos de certificación, ejecutado bajo la función `cpvAlgorithm`.

El proceso de lectura de la solicitud SCVP se muestra en las primeras líneas del Algoritmo 1, donde el Servicio de Validación tiene que obtener el certificado solicitado (línea 2), los Trust Anchors de confianza (línea 3) y, opcionalmente, un identificador que determina lo que se quiere obtener como respuesta (línea 4). Este último elemento `wantBack` es crucial para que el Servicio de Validación sepa si se desea la construcción y validación de un camino de certificación, o sólo la fase de construcción sin un estado de validación porque, por ejemplo, la validación se desea hacer de forma offline.

Como define el estándar SCVP, un Servicio de Validación está obligado a dar soporte a los dos procesos anteriores, pudiendo seleccionar uno u otro haciéndoles referencia en la solicitud con los OIDs `1.3.6.1.5.5.7.18.1` (`id-swb-pkc-best-cert-path`), sólo la fase de construcción, ó `1.3.6.1.5.5.7.18.2` (`id-swb-pkc-revocation-info`) para la de construcción y validación. Una vez extraída toda esta información, el Servicio de Validación tiene que llamar a la ejecución del algoritmo de construcción y validación de caminos de certificación (línea 8) con el nombre `cpvAlgorithm`.

El algoritmo de construcción de caminos de certificación propuesto puede verse, en líneas generales, como un algoritmo de recorrido por el *árbol de certificación*, asignando prioridades o pesos a cada una de las ramas para acelerar el propio proceso de búsqueda. De esta manera, el algoritmo de construcción puede simplificarse en un algoritmo de búsqueda basado en el *descubrimiento del mejor primer camino*.

Algoritmo 1: Algoritmo principal del Servicio de Validación

```
List<Certificate> trustAnchors                    // Lista de Trust Anchors confiables
OID wantBack                                      // Respuesta esperada por el usuario
Procedure validationService(InputStream user, SCVPRequest request)
1   | // Obtener certificado solicitado, Trust Anchors y lo que el usuario quiere como respuesta
2   | Certificate certificate ← request.getQueriedCertificates()
3   | trustAnchors ← request.getTrustAnchors()
4   | wantBack ← request.getWantBack()
5   | // Construir el primer punto de búsqueda con el certificado solicitado por el usuario
6   | Node node ← getNewNode(certificate)
7   | // Llamar al algoritmo de construcción y, opcionalmente, validación de caminos de certificación
8   | SCVPResponse response ← cpvAlgorithm(node)
9   | // Obtener respuesta SCVP de error al no poder construir y/o validar un camino de certificación
10  | if response == null then
11  |   | response ← getSCVPErrorResponse()
12  | // Enviar la respuesta SCVP al usuario
13  | sendSCVPResponse(user, response)
```

Debido a ello, el algoritmo debe gestionar de manera ordenada cada uno de los nodos (certificados) que esté analizando dentro del árbol de certificación, construyendo así los distintos caminos de certificación candidatos de forma incremental. Por tanto, la ejecución del algoritmo de construcción y validación de caminos de certificación debe gestionar internamente una lista de nodos con la siguiente estructura:

$$Node = (\text{Certificate } certificate, \text{ CRL } crl, \text{ ARL } arl, \text{ List<CertificatePair> } cross)$$

Para cada nodo, se tiene que almacenar el certificado de la entidad que representa (elemento `certificate`), sus listas de revocación (`crl` y `arl`) y su lista de pares de certificados cruzados (`cross`), siendo obligatorio sólo el elemento `certificate`, ya que pueden existir nodos en el árbol de certificación que no tengan listas de revocación, como las entidades finales, ni certificados cruzados, como las CAs subordinadas.

El primer nodo en el camino de certificación, conteniendo, al menos, el certificado enviado en la solicitud SCVP, lo tiene que obtener el Servicio de Validación (línea 6 en el Algoritmo 1) llamando al método `getNewNode`. Entonces, el Servicio de Validación lanzará el algoritmo de construcción y validación de caminos de certificación (línea 8) pasándole como parámetro a `cpvAlgorithm` ese primer nodo en el camino.

El método `getNewNode` tiene que obtener, si existieran, las listas de revocación y los pares de certificados cruzados asociados al certificado solicitado por el usuario. Toda esa información se puede obtener, como se detalla en la Sección 2.2.4, mediante una única consulta al Servicio de Directorio correspondiente y recuperando el objeto `pkiCA`. La URI de ese servicio se puede obtener de la extensión `SubjectInfoAccess` (método de acceso `id-ad-caRepository`) que tiene que estar definida en el certificado pasado por parámetro a este método (extensión analizada en la Sección 3.1.8).

Nótese que la estructura `Node` no permite almacenar respuestas OCSP, aunque el administrador del Servicio de Validación hubiera escogido ese protocolo de validación en lugar de listas de revocación (CRL/ARL). Esto se debe a que las listas de revocación se van a obtener de todas maneras durante la fase de construcción cuando se hagan las consultas al Servicio de Directorio, independientemente de si luego se realiza o no la fase de validación. En caso de haber optado por OCSP como mecanismo de revocación, las consultas para obtener los estados de revocación de cada uno de los certificados deben enviarse al servidor OCSP correspondiente una vez terminada la fase de construcción. La URI de cada servidor OCSP se puede obtener del certificado que representa el nodo (elemento `certificate`) haciendo uso de la extensión `AuthorityInfoAccess` (método de acceso `id-ad-ocsp`), como se ha analizado en la Sección 3.1.8.

La validación con OCSP se realiza al terminar la fase de construcción en lugar de validar la cadena de certificación temporal que está siendo analizada, ya que es mucho más costoso en tiempo que con listas de revocación (CRL/ARL), como se verá luego en la Sección 3.4. Por tanto, es preferible que el algoritmo de construcción y validación de caminos de certificación realice primero la fase de construcción, pudiendo incluso validar cada uno de los certificados que, incrementalmente, vaya analizando a través de listas de revocación (CRL/ARL). Esta información ya la obtiene recuperando el objeto `pkiCA` del Servicio de Directorio. Finalmente, se ejecutaría la validación haciendo uso del protocolo OCSP una vez obtenido un camino de certificación candidato.

En este punto, también hay que aclarar que el algoritmo que se propone aquí intenta descubrir los caminos de certificación desde el certificado solicitado por el usuario hasta uno de los Trust Anchors en los que confía, conocida esta forma de construcción como *dirección forward*, ya que, como se ha argumentado en la Sección 2.2.2, ofrece mayores ventajas que seguir una *dirección reverse*. En cualquier caso, esta última manera de construcción también sería posible en este algoritmo utilizando los certificados cruzados reverse, en lugar de los certificados cruzados forward que aquí se utilizan.

La Figura 3.1 muestra los bloques principales que componen el algoritmo diseñado para construir y validar caminos de certificación, llamado por el Servicio de Validación en la línea 8 del Algoritmo 1, el cual se ejecuta recursivamente con el último certificado añadido al camino de certificación candidato (`tempCertPath`). Este camino es una lista de nodos `Node` con todos los certificados que han sido recuperados hasta el momento para su análisis, y que siguen un mismo camino en el árbol de certificación.

El Algoritmo 2 presenta los pasos en pseudocódigo del algoritmo de construcción y validación de caminos de certificación, que también se muestran gráficamente en la Figura 3.1. Inicialmente, el nodo del certificado actual se añade (línea 3) al camino de certificación candidato (`tempCertPath`) y se intenta explorar la existencia de un camino de certificación válido a partir de éste a través de varios modelos de certificación:

- *Modelo jerárquico.* Se comprueba si el certificado actual es un certificado de CA raíz (línea 15). Si no es así, la búsqueda debe continuar la exploración a través de su certificado emisor: *búsqueda intradominio* (líneas 28-34). El algoritmo recupera todo el material criptográfico del nodo emisor del Servicio de Directorio que indica

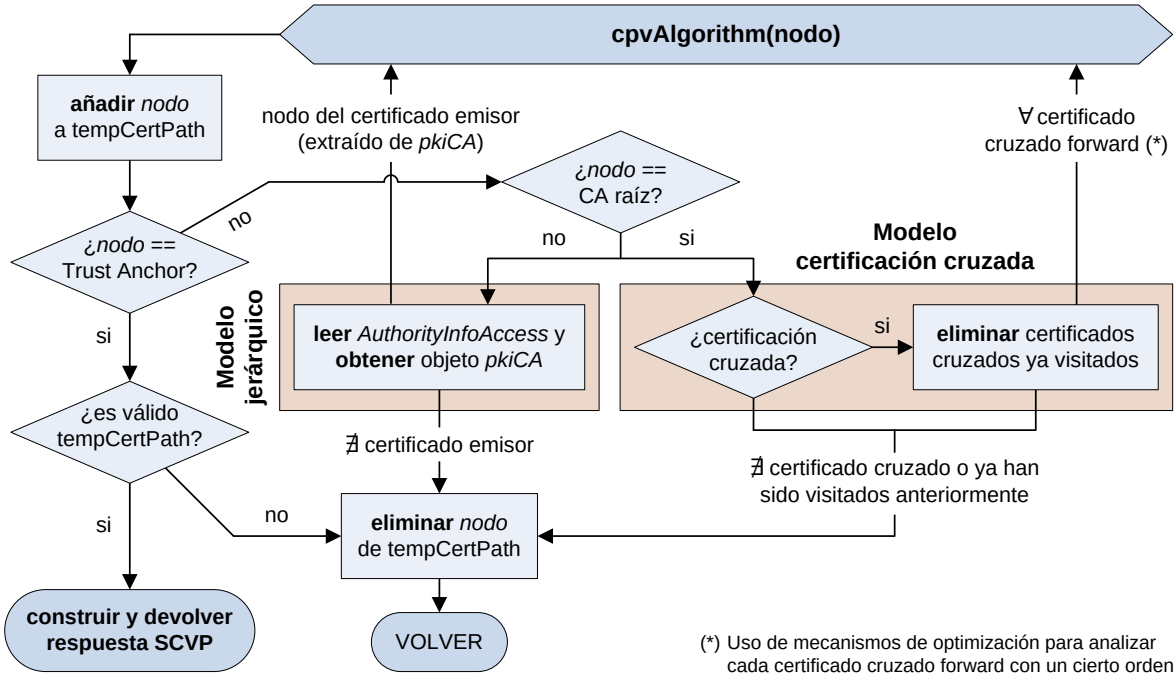


Figura 3.1: Algoritmo de construcción y validación de caminos de certificación

el certificado actual, usando el método `getAuthorityInfoAccess` (línea 29) para extraer la URI desde la extensión `AuthorityInfoAccess`, y así poder obtener su objeto `pkiCA` con el método `getPkiCAObject` (línea 32). El algoritmo se ejecuta a continuación de manera recursiva (línea 34), utilizando en cada paso el nodo que representa a la entidad emisora como nuevo certificado actual a ser analizado.

- *Modelo de certificación cruzada.* Se comprueba si el certificado actual de CA raíz atesora alguna relación de confianza con otros dominios de seguridad: *búsqueda interdominio* (líneas 16-26). Si las tiene, los nuevos certificados cruzados forward son añadidos a una cola de certificados (línea 22) que deben ser explorados uno por uno. Cada uno de estos certificados cruzados forward simboliza un posible camino de certificación, que puede terminar en alguno de los Trust Anchors definidos por el usuario. El algoritmo entonces se ejecuta recursivamente con cada uno de ellos (línea 26) para continuar la búsqueda por nuevos caminos de certificación.

En cualquier momento durante el proceso de búsqueda, el algoritmo puede encontrar que el certificado actual es uno de los Trust Anchors definidos por el usuario (línea 5 en el Algoritmo 2). Este hecho implicaría que el algoritmo ha encontrado un camino de certificación completo, desde el certificado proporcionado por el usuario hasta uno de esos Trust Anchors. Sin embargo, a pesar de encontrar un camino de certificación completo, el algoritmo todavía no tendría la certeza de que es válido hasta comprobar la correcta definición de cada uno de sus certificados, como el estándar X.509 establece en [24]: firmas digitales, períodos de validez, extensiones, etcétera.

Algoritmo 2: Algoritmo de construcción y validación de caminos de certificación

```
List<Node> tempCertPath                          // Camino de certificación temporal
List<CertificatePair> forwards                   // Certificados cruzados pendientes
Function cpvAlgorithm(Node node): SCVPResponse
1   SCVPResponse response = null
2   // Añadir el nuevo punto de búsqueda al camino de certificación temporal
3   tempCertPath.add(node)
4   // Comprobar si el certificado del punto de búsqueda es un Trust Anchor
5   if isNodeTrustAnchor(node.getCertificate(), trustAnchors) then
6       // Devolver camino de certificación y validación, según solicitud del usuario, si es válido
7       statusCode ← validateCertPath(tempCertPath, wantBack)
8       if statusCode == okay then
9           switch wantBack do
10              case id-swb-pkc-best-cert-path
11                  return getSCVPResponse(tempCertPath)
12              case id-swb-pkc-revocation-info
13                  return getSCVPResponse(tempCertPath, statusCode)
14  // Comprobar si el certificado del punto de búsqueda es una autoridad raíz de CA
15  else if isRootCACertificate(node.getCertificate()) then
16      // Extraer relaciones de confianza (modelo certificación cruzada)
17      List<CertificatePair> pairs ← node.getCrossCertificates()
18      if pairs <> null then
19          // Eliminar puntos de búsqueda que ya han sido visitados anteriormente
20          List<CertificatePair> newPairs ← newCrossCerts(forwards, pairs)
21          // Añadir los nuevos puntos de búsqueda –certificados forward– a los ya existentes
22          forwards.addAll(newPairs)
23          // Llamar recursivamente al algoritmo con cada nuevo punto de búsqueda
24          foreach newPairs do
25              Node newNode ← getNewNode(newPairs.getIssuedToThisCA())
26              response ← cpvAlgorithm(newNode)
27  else
28      // Extraer la extensión AuthorityInformationAccess (modelo jerárquico)
29      String uriAIA ← getAuthorityInfoAccess(node.getCertificate())
30      if uriAIA <> null then
31          // Recuperar el nuevo punto de búsqueda del objeto pkiCA del Servicio de Directorio
32          Node newNode ← getPkiCAObject(uriAIA, node.getIssuerDN())
33          // Llamar recursivamente al algoritmo con el nuevo punto de búsqueda
34          response ← cpvAlgorithm(newNode)
35  // Imposible continuar a partir del punto de búsqueda actual → eliminarlo del camino temporal
36  if response == null then
37      tempCertPath.removeLast()
38  return response
```

La comprobación anterior la tiene que hacer el algoritmo obligatoriamente llamando al método `validateCertPath` (línea 7), sin importar lo que el usuario haya solicitado en el elemento `wantBack`. Si el camino de certificación no es correcto en su definición, el certificado actual que está siendo analizado se debe eliminar del camino de certificación candidato (línea 37), ya que ni es correcto ni lo serán los caminos de certificación que pudieran construirse a partir de él. Al no ser válido, el algoritmo continúa entonces la búsqueda a partir del certificado anterior que se incluyó en el camino de certificación candidato construido hasta ese momento. Sin embargo, si en el punto anterior el camino de certificación sí era correcto en su definición, el algoritmo tiene que comprobar ahora lo que el usuario ha pedido en su solicitud SCVP enviada al Servicio de Validación: solamente construcción de un camino de certificación o también su validación.

Si el usuario sólo ha solicitado la construcción de un camino de certificación, usando para ello el identificador `id-swb-pkc-best-cert-path` (línea 10), el algoritmo devuelve ese camino de certificación como respuesta SCVP al Servicio de Validación llamando al método `getSCVPResponse` (línea 11). Este método tiene que incluir en la respuesta un objeto `ReplyWantBack` con el OID `1.3.6.1.5.5.7.18.1` en su campo *wb*, así como todos los certificados que constituyen el camino de certificación construido en su campo *value*, con el tipo `CertBundle` (este tipo de datos es una lista de objetos `Certificate` definido en el estándar X.509). El protocolo SCVP establece que esa lista de certificados debe seguir un cierto orden: "comenzando con el certificado solicitado por el usuario y terminando con el certificado emisor del Trust Anchor".

Si además de construir, el usuario también ha solicitado la validación del camino de certificación, identificador `id-swb-pkc-revocation-info` (línea 12 en el Algoritmo 2), éste se le pasa al proceso de validación para que coteje el estado de revocación de cada uno de sus certificados según uno de los dos mecanismos descritos en la Sección 2.2.3: listas de revocación (CRLs/ARLs) o mediante el protocolo OCSP. La respuesta SCVP, que el algoritmo le tiene que devolver al Servicio de Validación (línea 13), tiene que incluir un objeto `ReplyWantBack` con el OID `1.3.6.1.5.5.7.18.2` en su campo *wb* y una secuencia `RevInfoWantBack` en su campo *value* con: el camino de certificación construido en el campo *extraCerts* (tipo `CertBundle`), el estado de validación realizado por el Servicio de Validación bajo el objeto `ReplyCheck` con el campo `status` = 0 y, opcionalmente, los estados de revocación relacionados con cada uno de los certificados como una secuencia `RevocationInfos`. En este caso, como establece SCVP, el camino de certificación devuelto en la respuesta no tiene por qué mantener un orden.

La fase de validación la ejecuta el algoritmo mediante el método `validateCertPath` comentado anteriormente (línea 7), que, además de comprobar la correcta definición de cada uno de los certificados que forman parte del camino de certificación, también tiene que validar el estado de revocación de cada uno de esos certificados, siempre y cuando el usuario lo haya pedido en su solicitud.

En caso de que el administrador del Servicio de Validación haya optado por llevar a cabo el proceso de validación haciendo uso de listas de revocación, todas las CRLs y ARLs se recuperan de los diferentes Servicios de Directorio que han sido visitados durante la construcción del camino de certificación, mientras que, si el administrador

ha optado por OCSP, las respuestas de este protocolo se tienen que obtener realizando una solicitud OCSP para cada uno de los certificados del camino de certificación. En este último caso, las direcciones de cada servidor OCSP se pueden obtener accediendo al método de acceso `id-ad-ocsp` de la extensión `SubjectInfoAccess`, explicada en la Sección 3.1.8. Dependiendo de qué mecanismo de revocación se haya utilizado, los estados de revocación en la secuencia `RevocationInfos` pueden ser o bien listas de revocación (CRLs/ARLs), incluidos como objetos `CertificateList` según el estándar X.509 [24], o bien respuestas OCSP, incluidos como objetos `OCSPResponse` según el estándar OCSP [116]. Recalcar en este punto que, según el estándar SCVP, los usuarios no pueden solicitar el mecanismo de revocación que tiene que utilizar el Servicio de Validación, sino que es una decisión de los administradores del mismo.

Si durante los dos procesos anteriores el algoritmo termina sin encontrar un camino de certificación válido, el Servicio de Validación le devuelve al usuario una respuesta SCVP indicándole el error (líneas 10-13 en el Algoritmo 1). En este caso, la respuesta SCVP debe contener un objeto `ReplyCheck` con el campo `status` = 1, el cual indica que el Servicio de Validación no pudo construir o validar un camino de certificación desde el certificado que proporcionó en su solicitud hasta alguno de los Trust Anchors en los que confía (también definidos en su solicitud SCVP).

3.3. Despliegue de una federación de PKIs para un escenario multidominio

En esta sección se detalla en profundidad la federación de PKIs que se ha puesto en marcha para realizar diferentes pruebas de rendimiento en un escenario multidominio en particular. Sobre este escenario de pruebas se pretende, posteriormente:

1) Evaluar cómo se comportan los modelos avanzados de certificación cruzada que se han analizado en la Sección 2.2.1, ya sean mediante relaciones peer-to-peer o las establecidas a través de una BCA.

2) Analizar el impacto en el rendimiento que tiene la longitud de los caminos de certificación en este tipo de escenarios complejos.

La federación de PKIs a la que se hace alusión en esta sección es igualmente aplicable a cualquier sistema colaborativo para la detección de ataques distribuidos, como el que se ha marcado como objetivo de esta tesis doctoral. Este sistema debe tener en cuenta los dos puntos anteriores para, en primer lugar, analizar los modelos de certificación utilizados durante la construcción de la federación de PKIs de la Sección 3.2.1, con el objetivo de comprobar cuál tendría una mejor aplicación en un sistema colaborativo para la detección de ataques distribuidos. Este punto es importante ya que los distintos dominios de seguridad de este sistema van a tener que colaborar, interactuando entre ellos, para intercambiar las alertas que ocurran localmente en cada dominio, y así poder constatar si se está produciendo un ataque distribuido a nivel multidominio.

En segundo lugar, también se tiene que analizar el impacto que tiene la longitud de los caminos de certificación en el rendimiento del algoritmo de construcción y validación de caminos de certificación, explicado en la Sección 3.2. Esta longitud va a suponer un cierto impacto en el establecimiento de los canales de seguridad entre los dominios de seguridad, antes de que puedan interactuar entre sí de manera segura. Cuanto más grande sea la longitud de los caminos de certificación que el Servicio de Validación tiene que construir y, opcionalmente, validar, mayor impacto en el tiempo tendrá el sistema para poder intercambiar las alertas de forma segura y, por consiguiente, mayor tiempo se necesitará para confirmar la ocurrencia de un ataque distribuido.

En la Figura 3.2 se muestra la federación de PKIs que se ha puesto en marcha sobre un escenario multidominio de pruebas, totalmente aplicable, como se ha comentado más arriba, al sistema colaborativo para la detección de ataques distribuidos objeto de esta tesis doctoral. Los resultados que se han obtenido en las pruebas realizadas sobre este escenario se analizan más adelante en la Sección 3.4. Nótese también que el escenario de la Figura 3.2 se ha desplegado en un entorno de laboratorio cerrado y controlado, aunque posteriormente en la Sección 3.4 también se extrapolan los resultados obtenidos sobre un entorno real en producción, como es la FBCA.

Este escenario de pruebas se compone de seis CAs raíces (RCA) y una BCA neutral, que permite la unión de las seis CAs raíces bajo una misma federación de PKIs. En este escenario multidominio se pueden distinguir los dos modelos de certificación cruzada que se han analizado en la Sección 2.2.1. Entre BCA y las CAs raíces $RCA3$, $RCA5$ y $RCA6$ se ha desplegado un modelo de certificación cruzada mediante una Bridge CA; en este caso, a través de BCA. Por otro lado, también se ha desplegado un modelo de certificación cruzada peer-to-peer entre $RCA4$ y $RCA5$, y un red mallada de confianza completa –modelo *full mesh*– entre las tres CAs raíces $RCA1$, $RCA2$ y $RCA3$. Junto a cada una de las entidades que conforman el escenario de pruebas de la Figura 3.2, también se han especificado todos los certificados emitidos por, o para, cada una de esas entidades. En cada caso, se ha hecho uso de la misma nomenclatura utilizada en la Sección 2.2.4 para los atributos del Servicio de Directorio.

Como caso de uso, se considera un ejemplo típico en un sistema colaborativo para la detección de ataques distribuidos, donde dos IDSs ($IDS1$ e $IDS2$) pertenecientes a dos dominios de seguridad diferentes ($RCA4$ y $RCA1$, respectivamente) necesitan el establecimiento de un canal seguro de comunicaciones entre ambos para el intercambio de alertas. Desde el punto de vista de $IDS1$, esta entidad tiene que estar segura que existe un cierto nivel de confianza entre ella e $IDS2$, con la que necesita establecer un canal seguro por motivos de detección de ataques distribuidos. Ambas partes necesitan unas garantías evidentes sobre quién dice ser la otra, y que realmente no es cualquier otra entidad que está suplantando su identidad digital.

Para que la comprobación anterior se lleve a efecto, debe existir, al menos, un camino de certificación válido entre el certificado de $IDS2$ y alguno de los Trust Anchors en los que $IDS1$ confía. Esta validez es obligatoria para cada uno de los certificados que componen el camino de certificación, con respecto tanto a su contenido como a su estado de revocación. Para ello, este proceso se lo delega $IDS1$ al Servicio de Validación de su

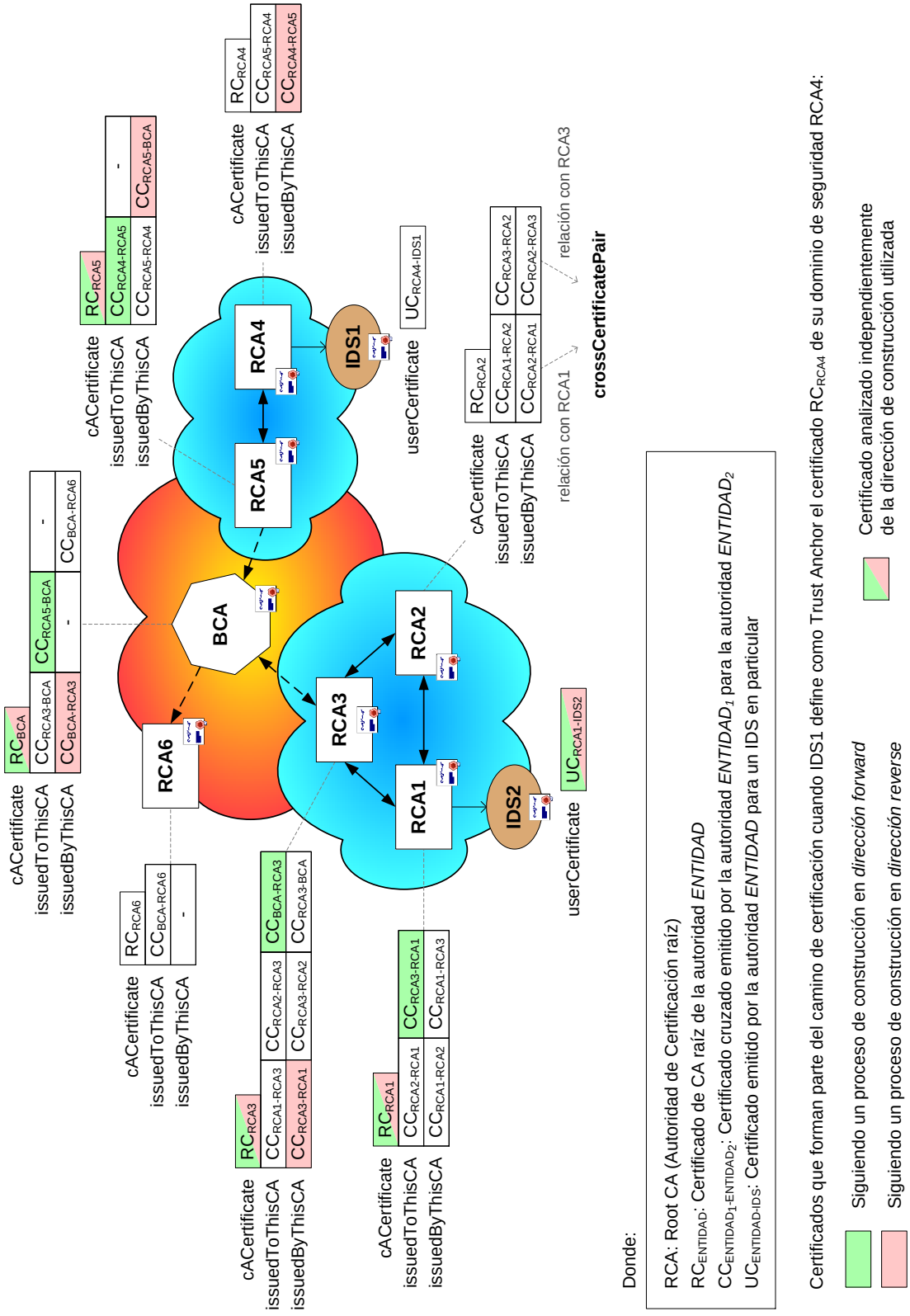


Figura 3.2: Escenario multidominio de pruebas

dominio de seguridad $RCA4$ enviándole una solicitud SCVP, para que éste construya un camino de certificación válido en su nombre con el algoritmo de construcción y validación de caminos de certificación de la Sección 3.2. Esta solicitud SCVP, que $IDS1$ le realiza al Servicio de Validación, debe incluir el certificado de $IDS2$ que desea que sea evaluado, así como los Trust Anchors en los que $IDS1$ confía.

Según los Trust Anchors que defina $IDS1$, la longitud del camino de certificación podría variar, desde un único certificado hasta nueve. En el primer caso, el camino de certificación se formaría únicamente con el certificado de $IDS2$ ($UC_{RCA1-IDS2}$) cuando $IDS1$ proporciona como Trust Anchor el certificado de $RCA1$ (RC_{RCA1}). El segundo caso corresponde con el camino de certificación más largo dentro de este escenario de pruebas, el cual se compondría de nueve certificados, cuando $IDS1$ envía como Trust Anchor el certificado de su propia CA: RC_{RCA4}. Nótese que el camino de certificación más largo realmente tendría once certificados, usando el dominio $RCA2$, aunque aquí se considera el mejor camino entre todos los más largos posibles.

El mejor *camino de certificación más largo* en este escenario de pruebas, mostrado gráficamente en la Figura 3.2, se detalla en (3.1).

$$\begin{aligned}\{CC_{RCA4-RCA5}\} \leftrightarrow RC_{RCA5} \rightarrow \{CC_{RCA5-BCA}\} \rightarrow RC_{BCA} \leftrightarrow \\ \{CC_{BCA-RCA3}\} \leftrightarrow RC_{RCA3} \leftrightarrow \{CC_{RCA3-RCA1}\} \leftrightarrow RC_{RCA1} \rightarrow UC_{RCA1-IDS2}\end{aligned} \tag{3.1}$$

donde:

- RCA simboliza una CA raíz en particular.
- BCA simboliza la autoridad neutral para la generación de la federación de PKIs.
- $RC_{ENTIDAD}$ representa el certificado de CA raíz de la autoridad cuyo nombre es $ENTIDAD$, ya sea o una RCA o la BCA.
- $\{CC_{ENTIDAD_1-ENTIDAD_2}\}$ hace referencia al certificado cruzado que la autoridad $ENTIDAD_1$ ha emitido para la autoridad $ENTIDAD_2$, a fin de establecer una relación de confianza unidireccional entre ellas. Esta relación de confianza sería de la forma $ENTIDAD_1 \rightarrow ENTIDAD_2$.
- $UC_{ENTIDAD-IDS}$ representa el certificado que es emitido por la CA raíz llamada $ENTIDAD$ para un IDS en particular.

Los nueve certificados que componen el camino de certificación más largo, para este escenario en concreto, se han resaltado en color verde en la Figura 3.2. Como se puede comprobar en el escenario, ninguno de los certificados cruzados reverse, almacenados en los elementos `issuedByThisCA`, forman parte del camino de certificación. Este hecho se debe a que el algoritmo de construcción y validación de caminos de certificación sigue un enfoque de construcción en *dirección forward*. En este caso, el algoritmo construye el camino desde el certificado a validar hasta uno de los Trust Anchors en los que $IDS1$ confía; es decir, entre $UC_{RCA1-IDS2}$ y RC_{RCA4}, respectivamente.

Si, en cambio, el administrador del Servicio de Validación hubiera optado por un enfoque de construcción en *dirección reverse*, desde el Trust Anchor definido por $IDS1$ (RC_{RCA4}) hasta el certificado de $IDS2$, el resultado del algoritmo sería el mismo camino de certificación mostrado en (3.1), pero siguiendo la dirección contraria en su proceso de construcción. La única diferencia es que, en este sentido de búsqueda en dirección reverse, se analizan los certificados almacenados en el elemento `issuedByThisCA`, como se ha resaltado en la Figura 3.2 en color rojo, en lugar de `issuedToThisCA`.

La mayor diferencia entre los dos sentidos de construcción radica en decidir el punto de inicio desde dónde el algoritmo tiene que comenzar su exploración. Si, por ejemplo, $IDS1$ establece RC_{RCA6} y RC_{RCA4} como Trust Anchors, el algoritmo siguiendo una dirección reverse podría comenzar su búsqueda con un dominio, como $RCA6$, desde el que no sería capaz de construir un camino de certificación válido, ya que no existe una relación de confianza entre $RCA6$ y BCA para alcanzar $UC_{RCA1-IDS2}$ posteriormente. El algoritmo escogería entonces el otro Trust Anchor definido por $IDS1$, el del dominio de seguridad $RCA4$, y sí que podría construir un camino de certificación válido. Sin embargo, el rendimiento final para la construcción de este camino de certificación se habría visto mermado en el tiempo, al comenzar el proceso de búsqueda desde un Trust Anchor con el que no ha conseguido, inicialmente, el éxito esperado.

3.4. Pruebas de rendimiento

El foco de atención detrás de esta sección se pone en evaluar la viabilidad de dos aspectos que, aunque parecen bien distintos, están estrechamente relacionados entre sí con respecto a los procesos de construcción y validación de caminos de certificación, en los que se estructura el algoritmo presentado en la Sección 3.2.2.

Un primer análisis, que se presenta en la Sección 3.4.1, se centra en la problemática de cómo las actuales PKIs pueden unirse a fin de construir una federación de PKIs. Cada una de esas PKIs define y gestiona sus propias políticas internas, hecho que puede hacer que un cierto dominio de seguridad no defina alguno de los requisitos y restricciones que otros dominios sí consideran como obligatorios. La incompatibilidad entre las PKIs originaría una falta de interoperabilidad, haciendo que la construcción de caminos de certificación se convirtiese en un proceso intratable dentro de una federación de PKIs con ciertas limitaciones técnicas. En el escenario multidominio de pruebas presentado en la Sección 3.3, con la integración requerida de una federación de PKIs para realizar el intercambio seguro de las alertas en un sistema colaborativo orientado a la detección de ataques distribuidos, esta incompatibilidad entre las PKIs de los distintos dominios de seguridad haría que las alertas no se pudieran intercambiar de forma segura y, como resultado, no sería factible la detección de ataques distribuidos.

Para la ejecución de este primer análisis, todos los dominios de seguridad que van a formar parte de una federación de PKIs deben seguir y cumplir el conjunto de requisitos definidos en la Sección 3.2.1 a nivel de servicios, así como los requisitos establecidos en la Sección 3.1 en la definición de sus distintos tipos de certificados.

Una vez analizada la problemática en la creación de una federación de PKIs, desde el punto de vista de la correcta implantación de sus PKIs, un segundo análisis tiene como objeto evaluar la viabilidad del algoritmo de construcción y validación de caminos de certificación, con el que poder confirmar que las comunicaciones entre las entidades de esas PKIs se vayan a realizar de forma segura. En el caso de un sistema colaborativo orientado a la detección de ataques distribuidos, este algoritmo va a poder garantizar a los IDSs un cierto nivel de seguridad en el intercambio de las alertas para ejecutar, de forma confiable, sus procesos de detección de ataques.

La ejecución y evaluación de este segundo análisis se presenta en la Sección 3.4.2, realizada en un entorno cerrado de laboratorio sobre la federación de PKIs desplegada en el escenario multidominio de la Sección 3.3. Como extrapolación del análisis anterior a un entorno real, donde se pueda comprobar el comportamiento real del algoritmo de construcción y validación de caminos de certificación propuesto en la Sección 3.2.2, en la Sección 3.4.3 se presentan los resultados reales de un conjunto de pruebas utilizando servidores LDAP y servidores OCSP públicos. La gran mayoría del gobierno federal de Estados Unidos, aunque todos pertenecientes a infraestructuras en producción.

3.4.1. Cumplimiento de los requisitos en un entorno real

Esta sección se centra en la evaluación de los requisitos analizados en la Sección 3.1 sobre las extensiones de los certificados X.509, debido a que su adecuado cumplimiento va a permitir un correcto funcionamiento del algoritmo de construcción y validación de caminos de certificación. El incumplimiento de alguno de ellos podría conducir a que el usuario del Servicio de Validación siempre obtuviese una respuesta incorrecta, debido a que el algoritmo no sería capaz de encontrar el material criptográfico que necesita el proceso de construcción para recorrer el árbol de certificación o la información de revocación necesaria para el proceso de validación.

El análisis realizado en esta sección se ha elaborado sobre una federación de PKIs real y en producción, tomando para ello la FBCA como base de estudio, ya que es una de las federaciones de PKIs existentes más importantes y en la que se incluye un mayor número de relaciones de confianza. Ésta define una BCA compuesta de 21 relaciones de certificación cruzada a las que se han unido las agencias federales, instituciones públicas y empresas privadas más importantes de Estados Unidos. En la Sección 2.2.1 se puede encontrar una descripción de la FBCA como uno de los escenarios reales multidominio en producción más importante en el marco federativo de PKIs.

Las pruebas que se han llevado a cabo para este análisis se han ejecutado realizando una búsqueda en profundidad en los diferentes servidores LDAP de las organizaciones que conforman la FBCA. Se han extraído, y almacenados para su posterior análisis, todos los objetos `pkiCA` recuperados de cada uno de los servidores LDAP con los que se pueden deducir todos los componentes de PKI que forman parte de cada una de las unidades organizativas que constituyen la FBCA. Principalmente, aquellos relacionados con los certificados de usuario, autoridades de certificación, listas de revocación y las relaciones de confianza entre esas autoridades mediante certificación cruzada.

El punto de inicio en esta búsqueda automática se ha marcado arrancando a partir del servidor LDAP principal que gestiona la U.S. Federal PKI Authority (FPKIA), accesible en `fpkia.gsa.gov`, el cual se encuentra bajo los dominios de protección del U.S. General Services Administration (GSA). A partir de todos los elementos de PKI recuperados se van incluyendo progresivamente al proceso de búsqueda las direcciones de los nuevos servidores LDAP que se vayan descubriendo. Fundamentalmente, esas nuevas direcciones se obtienen de los certificados cruzados, ya que son los elementos de enlace entre los distintos dominios administrativos, las cuales se pueden localizar a partir de la lectura de las extensiones `CRLDistributionPoints`, `SubjectInfoAccess` y `AuthorityInfoAccess`. Estas extensiones permiten localizar, respectivamente, los puntos desde dónde recuperar las listas de revocación (CRL/ARL), el objeto `pkiCA` del certificado que contiene la extensión y el objeto `pkiCA` de su entidad emisora.

Durante la ejecución de este proceso automático de búsqueda se han obtenido un total de 96 direcciones, correspondientes a servidores LDAP que almacenan información de certificación sobre elementos de PKI pertenecientes a la FBCA. De entre todas esas direcciones, es importante destacar que solamente 43 servidores LDAP son accesibles y mantienen elementos de PKI pertenecientes a sus dominios de seguridad. Las otras 53 direcciones no contienen elementos de PKI, o tampoco son accesibles como Servicio de Directorio (por ejemplo, por restricciones de acceso a solicitudes fuera de Estados Unidos) o, incluso, algunos han dejado de existir en la actualidad. De todas maneras, el resultado de esta búsqueda ha permitido la obtención de 268 048 certificados y 2840 listas de revocación, divididos según su tipo como se muestra en Tabla 3.2.

	Certificados X.509
Certificados cruzados *forward*	126
Certificados cruzados *reverse*	101
CAs raíces / BCA	38
CAs subordinadas	253
Entidades finales	267 530
TOTAL	**268 048**

	Listas de revocación
Listas CRLs	2764
Listas ARLs	76
TOTAL	**2840**

Tabla 3.2: Número de certificados y listas de revocación obtenidos de la FBCA

A pesar de ser accesibles algo menos de la mitad de servidores LDAP, el material criptográfico recuperado a partir de ellos ilustra un número suficientemente alto para poder calificar la siguiente discusión como un análisis representativo de la problemática que puede surgir a la hora de crear o desplegar una federación de PKIs, desde el punto de vista de la correcta implantación de las PKIs implicadas en la federación.

Destacar que todo el material criptográfico referenciado en la Tabla 3.2 (certificados y listas de revocación) se encuentra correctamente almacenado por las distintas PKIs implicadas en la FBCA, teniendo en cuenta los requisitos establecidos para un Servicio de Directorio (ver Sección 2.2.4), a excepción de un certificado cruzado forward.

Ese certificado está almacenado bajo el elemento `issuedByThisCA`, cuando debería estar en `issuedToThisCA` al ser un certificado cruzado forward, pero está almacenado como si fuera uno del tipo reverse. Este error se encuentra en el servidor LDAP, accesible en `fadspublic.state.gov`, que gestiona el *Department of State* de Estados Unidos bajo el siguiente *Nombre Distintivo* (del inglés Distinguished Name, DN):

$$\text{OU} = \textit{U.S. Department of State Root CA},\ \text{OU} = \textit{Certification Authorities},$$
$$\text{OU} = \textit{FADSPKI},\ \text{OU} = \textit{Department of State},\ \text{O} = \textit{U.S. Government},\ \text{C} = \textit{US}$$

Este fallo conlleva que los usuarios nunca podrían utilizar la CA de la PKI anterior, o alguno de sus certificados subordinados de su jerarquía interna, como Trust Anchor durante el proceso de construcción de un camino de certificación válido.

Con respecto a la correcta definición que deben tener las extensiones en cada uno de los tipos de certificados X.509, siguiendo los requisitos establecidos en la Sección 3.1, en la Figura 3.3 se presenta el porcentaje de certificados, diferenciados por su tipo, que las PKIs de la FBCA no han emitido correctamente según los requisitos esperados en cada extensión. La falta en el cumplimiento de alguno de esos requisitos podría suponerle a cualquier tipo de federación de PKIs un cierto impacto en el buen funcionamiento de su algoritmo de construcción y validación de caminos de certificación.

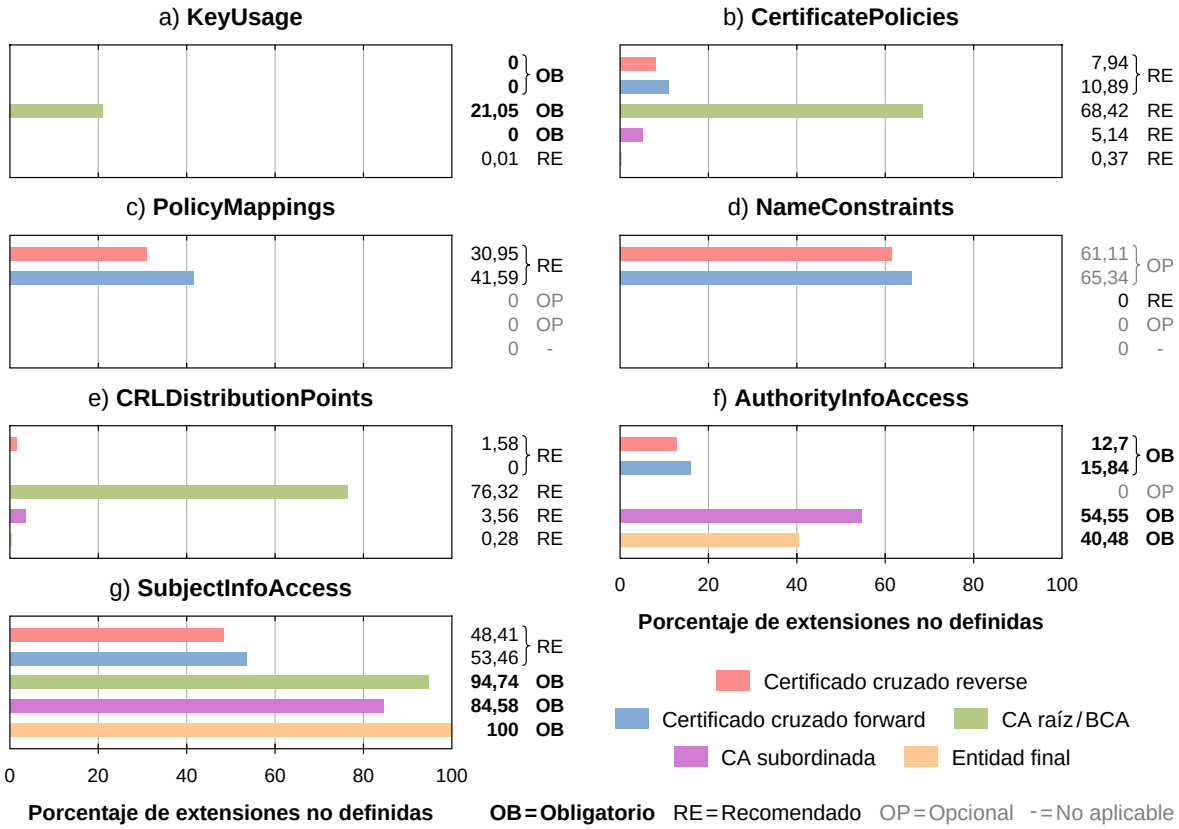


Figura 3.3: Extensiones establecidas como requisito y no definidas por la FBCA

Como en la Sección 3.1, la Figura 3.3 se ha marcado, para cada extensión y para cada tipo de certificado, con una de las cuatro opciones de requisito que tendrían que cumplir: *obligatorio* (OB), *recomendado* (RE), *opcional* (OP) y *no aplicable* (representado por un guion). Nótese que, ni se han incluido las extensiones totalmente opcionales, al no considerarse un impacto para un algoritmo de construcción y validación de caminos de certificación, ni tres de las extensiones que sí que se han definido como obligatorias: `AuthorityKeyIdentifier`, `SubjectKeyIdentifier` y `BasicConstraints`.

No se han incluido esas tres extensiones debido a que el análisis realizado revela unos resultados que no podrían percibirse con claridad en la Figura 3.3. Concretamente, la extensión `BasicConstraints` es la única que todos los certificados analizados han definido correctamente, independientemente de su tipo. Por otro lado, las extensiones `AuthorityKeyIdentifier` y `SubjectKeyIdentifier` también han sido correctamente definidas, pero solamente para todos los certificados cruzados (forward y reverse) y para los que representan CA raíces o BCAs. Sólo un 0,4 % de los certificados correspondientes a CAs subordinadas, un único certificado en realidad, no tiene bien definidas ambas extensiones, cuando es de obligado cumplimiento para este tipo de certificado.

La extensión `AuthorityKeyIdentifier` tampoco se ha definido correctamente en el 0,02 % de las entidades finales, suponiendo 53 certificados en total, mientras que la extensión `SubjectKeyIdentifier` sí está bien definida en todas esas entidades finales. A pesar de todo, ambas extensiones han sido marcadas dentro de la Sección 3.1 (ver Tabla 3.1) como extensiones o recomendadas u opcionales, respectivamente. Esta no obligatoriedad no implica un impacto sobre el correcto funcionamiento del algoritmo de construcción y validación de caminos de certificación, aunque sí el algoritmo podría tener un peor rendimiento durante la fase de construcción de los caminos de certificación candidatos en caso de no definirse esas extensiones.

Con respecto a las extensiones que sí han sido incluidas en la Figura 3.3, nótese que se han remarcado en negrita las que podrían provocar errores para las fases de construcción y validación de un camino de certificación, debido al incumplimiento en la obligatoriedad para alguna de esas extensiones, siempre siendo todo sometido al tipo de certificado al cual se hace referencia. A este respecto, véase que la discusión se puede centrar, principalmente, en tres extensiones concretas:

- `KeyUsage` (Figura 3.3a). Los certificados cruzados y los de CAs subordinadas sí definen correctamente esta extensión. Sin embargo, un total de 8 certificados (21,05 %) pertenecientes a CA raíces o BCAs no incluyen esta extensión de forma correcta. En este caso, los caminos de certificación incluyendo alguno de esos ocho certificados serían inválidos al no precisar el correspondiente uso de clave, como bien establece X.509 en su estándar [24]: "las CAs DEBEN incluir esta extensión en aquellos certificados que contengan claves públicas que sean utilizadas para validar firmas digitales de otros certificados de clave pública o CRLs".

- `AuthorityInfoAccess` (Figura 3.3f). Un número importante de certificados no definen esta extensión, aun cuando se considera crítica para la fase de construcción de caminos de certificación. Por ejemplo, un 40,48 % de certificados de entidades

finales no definen esta extensión, lo cual implica que, si alguno fuera el certificado solicitado por el usuario, el Servicio de Validación no podría obtener, a través del método de acceso `id-ad-caIssuers`, la dirección del Servicio de Directorio desde donde recuperar todo el material criptográfico de su CA emisora.

- `SubjectInfoAccess` (Figura 3.3g). La mayoría de autoridades y entidades finales de la FBCA no definen esta extensión, lo cual implica que los usuarios no pueden enviar al Servicio de Validación el certificado de estas entidades. El algoritmo sería incapaz de recuperar, por ejemplo, los certificados cruzados para continuar con el proceso de búsqueda hasta uno de los Trust Anchors. Esta extensión, considerada aquí como obligatoria, también es fuertemente aconsejable según [104, 215].

Aunque el resto de extensiones de la Figura 3.3 no supongan un cierto impacto en el funcionamiento de los procesos de construcción y validación de caminos de certificación, sí son altamente recomendadas por motivos de rendimiento en algún tipo de certificado en concreto. Por ejemplo, un algoritmo podría construir un camino de certificación en primer lugar, recuperando los certificados de CA a través del atributo `cACertificate` definido en los servidores LDAP, y después enviarlo al proceso de validación para su análisis. Para cada certificado, el proceso de validación intentaría leer cómo obtener la lista de revocación desde la extensión `CRLDistributionPoints` (Figura 3.3e), a fin de conocer si dicho certificado está revocado o no. Si esa extensión no estuviera definida en alguno de los certificados, el camino de certificación no sería válido al no disponer de su estado de revocación. Sin embargo, esta extensión no puede considerarse como obligatoria ya que es muy habitual que las CRLs y ARLs se encuentren almacenadas en el servidor LDAP bajo el objeto `pkiCA` (atributos `certificateRevocationList` y `authorityRevocationList`, respectivamente).

Otro punto importante a tener en cuenta es que hay muchas infraestructuras que no proporcionan servicios OCSP. Siguiendo con el ejemplo de la FBCA, solamente 6 de las 21 CAs raíces ofrecen este tipo de servicio. Esto conlleva un cierto revés si, durante la validación de un camino de certificación candidato, se llega a afirmar que ninguno de sus certificados está revocado, cuando alguno podría estarlo desde hace muy poco tiempo. En este caso, los usuarios de un Servicio de Validación sólo confiarán en métodos online para el chequeo del estado de revocación (por ejemplo, en respuestas OCSP), en lugar de hacerlo en mecanismos de revocación offline (por ejemplo, CRLs/ARLs o delta CRLs). Cada uno de los certificados definidos en el camino de certificación candidato debe validarse contra su servicio OCSP, proporcionando así un mayor nivel de confianza a los usuarios que deseen obtener transacciones electrónicas más seguras.

Como conclusión, todas las organizaciones pertenecientes a una federación de PKIs, como la propia FBCA, deben ofrecer servicios de validación más seguros mediante la inclusión en la extensión `AuthorityInfoAccess` dentro de sus certificados el método de acceso `id-ad-ocsp` para indicar la localización del servicio OCSP. La FBCA, por ejemplo, no es capaz de proporcionar este tipo de servicios, ya que menos de la mitad de sus infraestructuras ofrece este servicio OCSP.

3.4.2. Validación del algoritmo en un entorno de laboratorio

La evaluación que se lleva a cabo en esta sección sobre el algoritmo de construcción y validación de caminos de certificación, presentado en la Sección 3.2.2, pretende analizar el impacto en el rendimiento que se le supone al complejo proceso de construir y validar caminos de certificación. Para ello, en esta sección se detallan las medidas obtenidas sobre el rendimiento mediante la combinación de los siguientes factores:

- Evaluación del rendimiento según el *mecanismo de revocación* utilizado: listas de revocación (CRLs/ARLs) frente a respuestas OCSP. Se analizan los dos porque los administradores de cualquier dominio de seguridad, en un sistema colaborativo para la detección de ataques distribuidos, podrían decantarse por uno o por otro. Incluso podrían escoger, por ejemplo, las listas de revocación para los IDSs de su propio dominio de seguridad, e imponer OCSP para las comunicaciones seguras interdominio con los IDSs del resto de dominios de seguridad de la federación de PKIs. Esta elección podría depender también del rendimiento que tuvieran.

- Impacto en el rendimiento dependiendo de la *longitud del camino de certificación*. En esta prueba se cambia el Trust Anchor de confianza para la entidad solicitante ($IDS1$ en la Figura 3.2), haciendo variar el camino de certificación desde un único certificado, cuando el Trust Anchor es RC_{RCA1}, hasta nueve si el Trust Anchor es RC_{RCA4}, obteniendo el camino de certificación (3.1) de la Sección 3.3.

La ejecución de las pruebas se ha llevado a cabo de manera experimental sobre un entorno cerrado de laboratorio, el cual implementa la federación de PKIs presentada en la Sección 3.3. En este escenario, cada una de las CAs ha sido instalada y configurada en servidores físicos independientes, cumpliendo con las características software que se especifican en la Tabla 3.3. En dicha tabla, también se ha incluido el hardware utilizado para el despliegue de la federación de PKIs en este escenario de pruebas multidominio.

Hardware	**CPU**	Intel QuadCore Xeon E5410, 2.33 GHz, 64 bits
	Tamaño de la caché	12 MB L2
	Memoria total	4 GB
Software	**Software de PKI**	UMU-PKIv6 7.2.1 Release Candidate 2
	Repositorio	OpenLDAP 2.4.40
	Base de datos	PostgreSQL 9.4
	Contenedor de servlets	Apache Tomcat 8.0.23 (Servlet 3.1)
	JDK	Java SE 8 Update 45 + Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files 8

Tabla 3.3: Requisitos software para el Servicio de Validación y hardware utilizado para su puesta en marcha en el escenario de pruebas multidominio

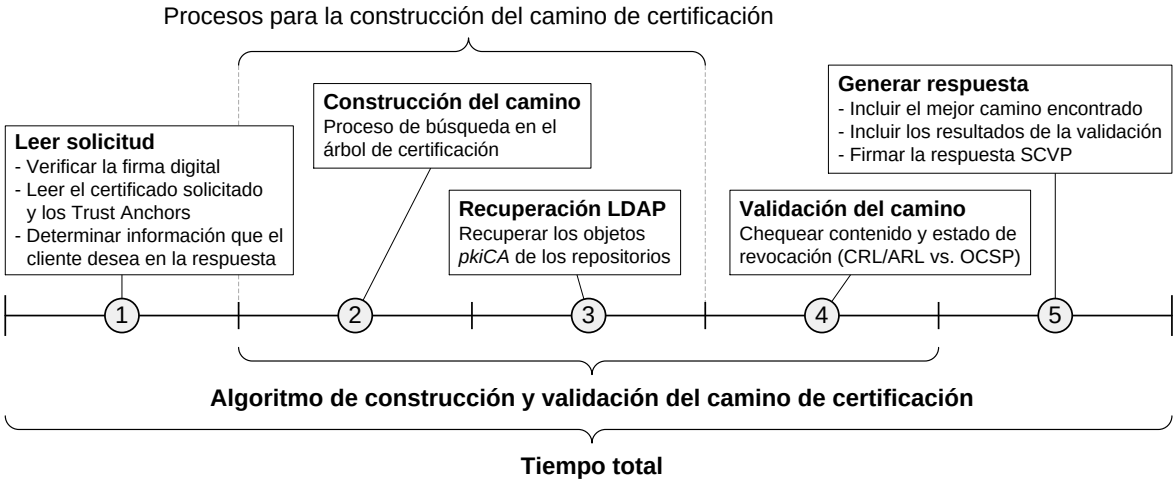


Figura 3.4: Los cinco procesos parciales que definen el Servicio de Validación

Con respecto a los requisitos software, se puede comprobar que se han hecho uso de distribuciones libres, algunas de ellas siendo de código abierto, las cuales ofrecen tanto soporte multiplataforma como un total soporte para su ejecución en redes IPv6. Por ejemplo, las versiones de OpenLDAP 2.x soportan tanto IPv4 como IPv6, mientras que PostgreSQL ofrece un soporte total para IPv6 a partir de su versión 7.

Para la evaluación de las diferentes medidas de rendimiento anteriores, el Servicio de Validación se ha dividido en cinco procesos, entre los que destaca, principalmente, el algoritmo de construcción y validación de caminos de certificación, con los que poder obtener ciertas conclusiones sobre el rendimiento de cada una de las partes de forma más atómica. En la Figura 3.4 se muestran estos cinco procesos, explicados brevemente a continuación, donde cada uno es etiquetado con un número para indicar posteriormente en el análisis de los resultados a qué proceso se está haciendo referencia.

① **Leer solicitud**. El Servicio de Validación recibe una solicitud SCVP, firmada digitalmente para autenticar a la entidad solicitante como un usuario legítimo. La solicitud incluye el certificado a validar, los Trust Anchors en los que confía y una serie de requisitos indicando qué información desea obtener como respuesta. Para estas pruebas, se considera que se solicita el mejor camino de certificación (ver Figura 3.2) y los resultados de revocación de todos sus certificados.

② **Construcción del camino**. Este proceso representa el procedimiento recursivo del algoritmo donde se explora el árbol de certificación para buscar los caminos de certificación candidatos, según los requisitos demandados por el usuario.

③ **Recuperación LDAP**. Este proceso se encarga de recuperar los objetos `pkiCA` desde los repositorios LDAP (ver Sección 2.2.4) para obtener toda la información con la que construir caminos de certificación candidatos. Estos objetos incluyen los certificados de los diferentes dominios de seguridad y las listas de revocación. Estas últimas se podrían utilizar, posteriormente, en la fase de validación.

④ **Validación del camino**. Durante este proceso, cada certificado en el camino de certificación candidato debe ser validado bajo dos enfoques distintos, como se establece en el estándar X.509. En primer lugar, se tiene que realizar un chequeo estructural de cada certificado, comprobando, entre otros, la integridad de su contenido, la correcta definición de las extensiones críticas y las firmas digitales entre todos los pares de certificados que son parte del camino de certificación. En segundo lugar, también se tiene que chequear el estado de revocación de cada certificado para comprobar si todavía es válido más allá de su vigencia. Para este último punto, el algoritmo soporta los siguientes mecanismos de revocación:

- CRL/ARL. Esta información ya ha sido recuperada en el proceso anterior.
- OCSP. Cada servidor OCSP pondrá un sello de tiempo en su respuesta, con el Protocolo de Sellado de Tiempo (del inglés Time-Stamp Protocol, TSP), que certifique cuándo se realizó la validación [219]. Todas las solicitudes y respuestas, tanto OCSP como TSP, las firman digitalmente sus autoridades para proteger la integridad de los mensajes intercambiados.

⑤ **Generar respuesta**. El Servicio de Validación genera una respuesta SCVP, firmada digitalmente, indicando el estado sobre la solicitud que recibió, el mejor camino de certificación encontrado y el correspondiente resultado de validación con los estados de revocación: CRLs/ARLs o respuestas OCSP.

Finalmente, indicar que todas las medidas de tiempo han sido tomadas en el Servicio de Validación que proporciona el dominio $RCA4$, a excepción de los tiempos obtenidos en las consultas OCSP. Estos tiempos han sido tomados directamente dentro del propio *OCSP Responder* al que se está realizando la consulta de revocación.

Tiempo promedio con respecto a la longitud del camino de certificación

Con esta primera prueba se pretende analizar el impacto que supone la construcción y validación de caminos de certificación en términos de rendimiento, teniendo en cuenta para ello la longitud de esos caminos. Con estos tiempos en la mano, se podrá analizar el comportamiento de los cinco procesos en los que se ha dividido el Servicio de Validación, desplegado sobre un escenario multidominio basado en modelos de certificación cruzada. Esta prueba también se ha ejecutado según los dos mecanismos de revocación analizados anteriormente, CRL/ARL y OCSP, con los que se tendrá una mejor visión de cuál de los dos ofrece mejores resultados en este tipo de escenarios multidominio.

Para la ejecución de esta prueba, se han enviado 125 solicitudes SCVP al Servicio de Validación de manera secuencial, para extraer así los tiempos promedios (parciales y totales) en el rendimiento de cada uno de los cinco procesos en los que se ha dividido el Servicio de Validación. Todos estos tiempos, tomados en milisegundos, se muestran en la Figura 3.5. Por un lado, la Figura 3.5a muestra el impacto que tiene el mecanismo de revocación basado en CRLs/ARLs sobre el rendimiento total del Servicio de Validación, mientras que la Figura 3.5b muestra los mismos tiempos pero utilizando OCSP.

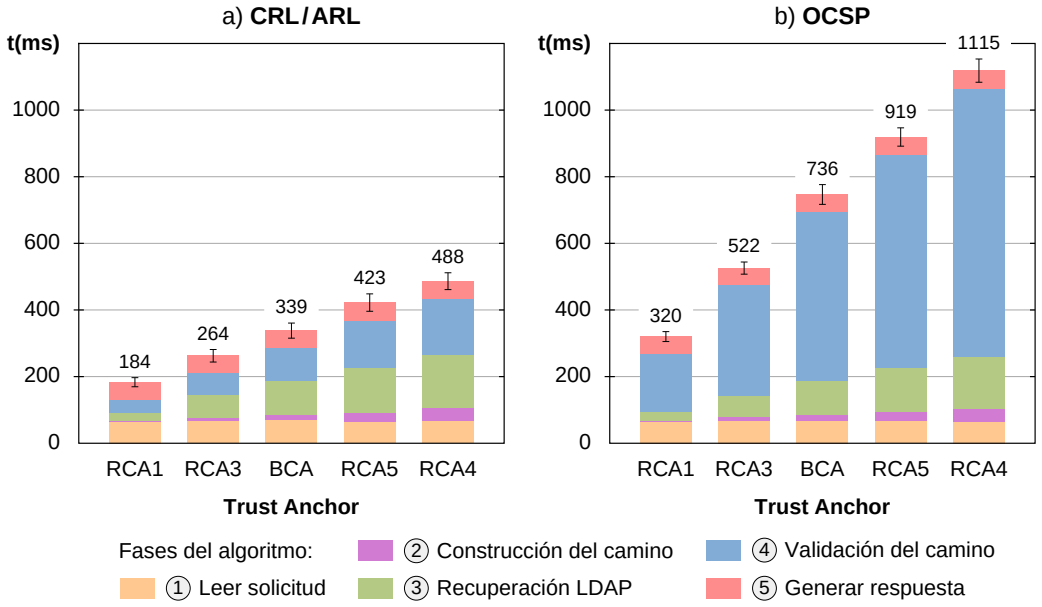


Figura 3.5: Tiempos promedios según la longitud del camino de certificación y el método de revocación: a) CRL/ARL y b) OCSP

En ambos gráficos de columnas de esta figura, el eje de abscisas representa cada uno de los dominios de seguridad que se incluyeron como Trust Anchor en la solicitud SCVP. Estos puntos de confianza serán los que el Servicio de Validación utilice durante el proceso de construcción del mejor camino de certificación.

La columna $RCA1$ representa el camino de certificación más corto posible, el cual sólo incluye el certificado que se desea validar (el certificado de $IDS2$ en el escenario de la Figura 3.2, con el nombre $UC_{RCA1-IDS2}$), mientras que la última columna representa el camino de certificación más largo (el mejor de entre todos los más largos posibles) cuando se hace uso del certificado del dominio de seguridad $RCA4$ como Trust Anchor. Este camino de certificación está compuesto por 9 certificados, los cuales se muestran en el camino (3.1) que se ha presentado en la Sección 3.3. El número que se indica en la parte superior de cada columna representa el tiempo promedio en total que tarda el Servicio de Validación en poder ejecutar todo el proceso de construcción y validación. Los pequeños segmentos verticales, superpuestos a final de cada una de estas columnas, representan las desviaciones estándares obtenidas en cada una de las ejecuciones. Por otro lado, el eje de ordenadas de ambos gráficos para esta figura muestra los tiempos parciales obtenidos en cada proceso, parte del Servicio de Validación.

Como se puede observar en la Figura 3.5, los procesos ① y ⑤ (solicitud y respuesta) tardan, respectivamente, 65 ms y 54 ms en promedio. También se puede percibir que son valores casi constantes, independientemente de la longitud del camino de certificación o del mecanismo de revocación que se esté analizando. Por tanto, estos dos procesos se pueden considerar fases independientes dentro del Servicio de Validación, ya que se ejecutan justo antes y después del propio algoritmo de construcción y validación.

Por otro lado, los procesos ②, ③ y ④ pueden considerarse como tiempos lineales, con un orden $O(n)$, ya que van creciendo de forma lineal conforme el camino de certificación también incrementa su número de certificados. Este incremento lineal también se puede observar con ambos mecanismos de revocación, por lo que el algoritmo sigue ofreciendo tiempos lineales independientemente del mecanismo utilizado.

Con respecto a la longitud que puede alcanzar el camino de certificación, el proceso ④ supone el principal factor de impacto, el cual corresponde con la validación de los caminos de certificación candidatos. Comparando los dos mecanismos de revocación, se puede constatar que, para el mecanismo basado en CRLs/ARLs, la inclusión de una nueva CA raíz en el camino de certificación –certificado de CA más certificado cruzado– supone un incremento de algo menos de 100 ms sobre el tiempo total de procesamiento. Para el caso de OCSP, este incremento supondría alrededor de 200 ms.

Siguiendo con los métodos de revocación, se puede afirmar que el mecanismo basado en CRLs/ARLs supone una sobrecarga en el tiempo total sobre un 40-55 % menor que con OCSP, aunque este valor decrementa conforme el camino de certificación aumenta. Este decremento alcanza un punto donde la validación utilizando OCSP se estabiliza alrededor del 45 % de sobrecarga adicional que con CRLs/ARLs. En cualquier caso, en entornos multidominio reales, esta afirmación podría ser ligeramente distinta, como se analiza en la Sección 3.4.3. Las solicitudes OCSP dependen mucho de la red y su tráfico en tiempo real, mientras que es menos importante para CRLs/ARLs, al recuperar todo el material de validación durante el proceso ③ dentro del algoritmo.

Finalmente, los tiempos totales de la Figura 3.5 manifiestan que, para el camino de certificación más largo, el Servicio de Validación requiere de 488 ms para el mecanismo basado en CRLs/ARLs y 1115 ms para OCSP. Ambos valores son tiempos aceptables por un usuario del Servicio de Validación, y pueden ser totalmente asumibles para los escenarios multidominio propuestos a lo largo de este capítulo. En particular, para que los IDSs de un sistema colaborativo para la detección de ataques distribuidos puedan establecer los canales seguros necesarios para el intercambio de sus alertas.

Análisis de rendimiento para el camino de certificación más largo

Siguiendo con la prueba anterior, el objetivo ahora es dar una visión más específica, a través de valores numéricos concretos, de los tiempos tomados en cada uno de los cinco procesos parciales de la Figura 3.4. Estos tiempos corresponden con la última columna de la Figura 3.5, el camino de certificación más largo con 9 certificados, siendo éste el peor de los casos cuando se utiliza el certificado de $RCA4$ como Trust Anchor.

La Tabla 3.4 recoge todos los tiempos parciales para cada uno de los procesos que conforman el Servicio de Validación, adjuntando también sus desviaciones estándares en cada uno de los casos. Es importar destacar en este punto que, para ambos mecanismos de revocación, se han extendido los tiempos de los procesos ③ y ④ en acciones mucho más atómicas para que puedan ser analizadas con un mayor nivel de detalle, al revelarse como los procesos que más impacto tienen sobre el rendimiento: *Recuperación LDAP* y la *Validación del camino*, respectivamente.

	①	②	③						④						⑤	TOTAL
			RCA1	RCA3	BCA	RCA5	RCA4	**TOTAL**	RCA1	RCA3	BCA	RCA5	RCA4	**TOTAL**		
Tiempo promedio																
CRL/ARL	65	39	33	36	34	32	26	**161**	-	-	-	-	-	**166**	55	**488**
OCSP	63	37	34	35	36	33	25	**163**	159	146	146	149	127	**807**	54	**1115**
Desviación estándar																
CRL/ARL	13	13	11	13	12	11	10	**20**	-	-	-	-	-	**15**	11	**25**
OCSP	11	13	10	11	12	10	9	**19**	25	25	22	24	18	**65**	9	**70**

Tabla 3.4: Tiempos promedios y desviaciones estándares obtenidas para el camino de certificación más largo y el método de revocación utilizado

Como se puede ver en la Tabla 3.4, todos los tiempos tomados son bastante similares sin tener en consideración el mecanismo de revocación utilizado. La principal diferencia se observa en el proceso de validación, ya que el procesamiento de las solicitudes OCSP incrementa considerable el tiempo total de este proceso si se compara con el mecanismo basado en CRLs/ARLs. De manera más concreta, OCSP tarda 807 ms en validar todos los certificados del camino de certificación, mientras que utilizando CRLs/ARLs este proceso sólo necesita 166 ms. La diferencia entre estos tiempos se debe a que, haciendo uso de CRLs y ARLs, todas las listas de revocación ya han sido obtenidas durante el proceso ③ de *Recuperación LDAP*. Por tanto, a partir de estos datos se puede constatar que OCSP tarda, alrededor de, 5 veces más que CRL/ARL.

Como se ha comentado al comienzo de esta sección, la gestión de las solicitudes y respuestas OCSP implica el tener que realizar una solicitud a un servidor TSP para obtener un sello de tiempo, que será incluido en la respuesta OCSP, con el que certificar el momento en el que la decisión de validación se llevó a cabo. Estas llamadas a ese servidor de sellado de tiempo tardan 37 ms de promedio, alrededor de una cuarta parte del tiempo total que el *OCSP Responder* tarda en procesar la solicitud OCSP. Nótese que estos tiempos no han sido incluidos en la Tabla 3.4 por motivos de espacio.

Finalmente, como última observación sobre esta prueba, se puede comprobar que la suma de los tiempos parciales para el caso de OCSP es 727 ms, y no 807 ms como se muestra en la Tabla 3.4. Esta diferencia corresponde al tiempo que el algoritmo necesita para verificar la integridad de los diferentes certificados del camino, principalmente sus firmas digitales, y el tiempo en enviar las solicitudes OCSP a través de la red.

Como conclusión de este experimento en un entorno cerrado de laboratorio, se puede confirmar que los resultados obtenidos sobre el rendimiento del Servicio de Validación, propuesto en la Sección 3.2, demuestran que el tiempo total depende en gran medida de la fase de validación de cada certificado del camino de certificación construido.

El hecho anterior manifiesta que la longitud del camino de certificación es el factor más importante a tener en cuenta en una infraestructura de certificación. Sin embargo, esta afirmación debe ser contrastada a partir de una experimentación sobre los mismos procesos anteriores, pero ahora ejecutados sobre un escenario real y en producción.

3.4.3. Medidas de rendimiento en entornos reales

Después de realizar diferentes pruebas sobre un escenario controlado de laboratorio, es inevitable preguntarse cómo serían los tiempos en rendimiento si se ejecutaran sobre un escenario real, donde otros tipos de factores, como los retrasos en la red, podrían hacer inviable el uso del Servicio de Validación propuesto. Como se ha discutido antes, solamente los procesos ③ y ④ dependen del tráfico de la red y sus retrasos. Es decir, la *Recuperación LDAP* y la *Validación del camino*, respectivamente, cuando el mecanismo de revocación es OCSP. El resto de procesos se ejecutan localmente en el propio Servicio de Validación, por lo que son independientes del escenario donde se desplieguen.

Los tiempos obtenidos en esta prueba se han tomado después de enviar múltiples consultas a servidores LDAP y OCSP públicos, todos de infraestructuras en producción. Después de analizar más de 30 infraestructuras, algunas de ellas bajo los dominios de la FBCA, la Tabla 3.5 muestra los tiempos promedios y desviaciones estándares para las infraestructuras que ofrecen un libre acceso a sus servidores LDAP y OCSP.

	Certificados cruzados		**LDAP**		**OCSP**	
Dominio	Forward	Reverse	Tiempo promedio	Desviación estándar	Tiempo promedio	Desviación estándar
DoD Root CA[a)]	15	1	1446	95	184	14
DoD Interoperability Root CA[a)]	3	1	1186	87	190	7
ORC Government ROOT[b)]	4	1	821	74	280	26
ORC ROOT[b)]	3	2	1024	92	268	21
SAFE-Biopharma Association	8	8	1519	79	309	35
Digital Signature Trust (DST)	1	1	762	67	446	58
EuroPKI	0	0	-	-	168	9
	TOTAL (promedio)		**1126**	82	**264**	24

CAs raíces de la federación de PKIs pertenecientes a la:

a) *Defense Information Systems Agency* del Departamento de Defensa (DoD) de Estados Unidos

b) *Operational Research Consultants* (ORC) del gobierno federal de Estados Unidos

Tabla 3.5: Tiempos promedios y desviaciones estándares en entornos reales

El resto de las infraestructuras no ofrecen acceso público a alguno de sus servidores, quizá debido a restricciones en sus políticas de seguridad. Por ejemplo, la EuroPKI [220] sólo ofrece acceso a su servidor OCSP, pero no a su servidor LDAP.

Las dos primeras CAs raíces de la Tabla 3.5 pertenecen a la *Agencia de Sistemas de Información de Defensa* (del inglés Defense Information Systems Agency, DISA) de Estados Unidos [221], mientras que las dos siguientes son dos CAs raíces de los *Consultores en Investigación Operativa* (del inglés Operational Research Consultants, ORC) del gobierno federal de Estados Unidos [222]. En estas pruebas también se han utilizado las CAs raíces de Digital Signature Trust (DST) [223], la de la EuroPKI [220], y la de SAFE-Biopharma Bridge Certificate Authority (SBCA) [224], la cual establece relaciones de confianza entre todos los miembros de Biopharma Association –grandes empresas farmacéuticas– y varias agencias gubernamentales de Estados Unidos.

Las pruebas que han permitido recoger los resultados de la Tabla 3.5 se han realizado enviando 25 solicitudes LDAP y 25 solicitudes OCSP, todas secuenciales, y calculando posteriormente los tiempos promedios y las correspondientes desviaciones estándares en milisegundos. Como se puede ver en esa tabla, todas las CAs utilizadas, a excepción de la EuroPKI, pertenecen o tienen una relación de confianza basada en un modelo de certificación cruzada con la FBCA. Esas relaciones vienen representadas en número a través de los certificados cruzados que gestiona cada CA, lo cual supone un efecto significativo en los tiempos obtenidos por las recuperaciones LDAP. A mayor número de certificados cruzados, mayor será el tamaño del objeto `pkiCA` a recuperar.

Las recuperaciones LDAP, al necesitar 1126 ms de promedio, suponen el factor más crítico dentro del Servicio de Validación. Como ejemplo, si se mezclasen estos tiempos con los extraídos durante las pruebas de la Sección 3.4.2, para el camino de certificación más largo, el proceso completo para la construcción y validación tomaría casi 6 segundos en un escenario real, si se utilizaran CRLs/ARLs como mecanismo de revocación. Es decir, las cinco recuperaciones LDAP tardarían 5630 ms (1126 ms por cada una), más los 325 ms que el Servicio de Validación necesita internamente para ejecutar el resto de procesos del algoritmo. Este último valor corresponde a la suma de los cinco procesos, a excepción del tiempo del proceso ③ que representa las recuperaciones LDAP, el cual es reemplazado por este nuevo tiempo tomado desde el escenario real. Para el caso de OCSP, el proceso completo tomaría un poco más de 7 segundos (concretamente, 7184 ms): 5630 ms de las cinco recuperaciones LDAP, 1320 ms para las solicitudes OCSP (264 ms por cada una) más 234 ms para ejecutar el resto de los procesos. Como en el caso anterior, este último tiempo corresponde a la suma de todos los procesos que son parte del Servicio de Validación, menos las recuperaciones LDAP y la gestión de las solicitudes OCSP que ya han sido reemplazadas por los nuevos valores reales.

Como conclusión, los resultados analizados más arriba sobre el rendimiento de un Servicio de Validación demuestran que, en un entorno cerrado de laboratorio, el tiempo total de procesamiento depende en gran medida de la fase de validación de los caminos de certificación, siendo por tanto la longitud del camino el factor con mayor impacto. Sin embargo, la extrapolación de las pruebas a un entorno con infraestructuras reales y en producción ha demostrado que otros factores, como los retrasos en la red, decanten los factores de impacto sobre el proceso de construcción, teniendo las recuperaciones de los servidores LDAP el mayor peso en términos de rendimiento.

3.5. Conclusiones del capítulo

Las infraestructuras de clave pública (PKI) han pasado a ser uno de los componentes principales para el despliegue de modelos de confianza y de seguridad en el seno interno de las organizaciones que quieran proteger, entre otros, sus canales de comunicación, otorgando un cierto grado de seguridad en el intercambio de información entre entidades que, a priori, no tienen por qué conocerse. En el contexto de un sistema colaborativo para la detección de ataques, el uso de las PKIs se convierte en un requisito obligatorio para que el intercambio de las alertas de detección entre los distintos IDSs se realice de manera segura, protegiendo que esas alertas ni sean alteradas durante su transmisión por los canales públicos de comunicación ni sean enviadas por IDSs considerados como entidades no legítimas para el sistema global de detección.

En un entorno tan distribuido como los sistemas colaborativos para la detección de ataques, donde los IDSs pertenecen a múltiples dominios de seguridad, estos dominios tienen que dar soporte a que se puedan establecer relaciones de confianza entre todos los actores implicados en los procesos de detección. Pero no exclusivamente en el marco interno de un dominio de seguridad, sino también con otros dominios con los que tenga algún acuerdo previo de colaboración para compartir algún tipo de información, como las alertas de detección, que precise la preservación de su seguridad. En este escenario, las soluciones basadas en tecnologías de PKI deben ser lo suficientemente flexibles para adaptarse a las nuevas exigencias, que en algunos casos pueden incluir el establecimiento de relaciones de certificación cruzada entre todas sus CAs, enmarcadas como las fuentes principales de confianza de cada dominio de seguridad.

Con todas esas premisas en mente, en este capítulo se ha presentado el diseño de los servicios que necesita una PKI para ofrecer soporte a cualquier escenario multidominio, incluido, por supuesto, el relacionado con los sistemas colaborativos para la detección de ataques distribuidos. Para la creación de las relaciones de confianza que permitan una correcta interoperabilidad entre todos los dominios de una federación de PKIs, en este capítulo se han analizado y propuesto una serie de requisitos de obligado cumplimiento en la definición de cada una de las extensiones de un certificado X.509, que son marcadas como obligatorias, recomendadas, opcionales o no aplicables en cada tipo de certificado que puede ser emitido en un escenario multidominio: certificados cruzados, de CA raíz (incluyendo la BCA), de CA subordinada y certificados de entidades finales. Sobre esta base, también se ha presentado el diseño de un Servicio de Validación que, apoyándose en el algoritmo de construcción y validación de los caminos de certificación propuesto en este capítulo, es capaz de validar las credenciales de un IDS (es decir, su certificado) bajo cualquier modelo avanzado de confianza. Este servicio es el principal soporte para la validación de IDSs legítimos de un sistema colaborativo para la detección de ataques distribuidos, antes de llevar a cabo un intercambio de alertas con alguno de ellos.

El diseño de las dos propuestas que se han presentado, sobre el conjunto de requisitos para la correcta emisión de certificados en escenarios multidominio y el algoritmo para la construcción y validación de caminos de certificación, pretenden subsanar la falta de estándares con los que, respectivamente: realizar la construcción de federaciones de

PKI totalmente interoperables y desarrollar servicios con los que construir y validar los caminos de certificación en escenarios multidominio. Los organismos de estandarización, concretamente el IETF, sólo ha proporcionado una serie de guías de y recomendaciones enfocadas a un amplio rango de entornos de certificación, con los que los desarrolladores de cualquier PKI puedan implementar sus propios mecanismos.

Ambas propuestas se han implementado para su evaluación, tomando como punto de partida para ello una PKI ya desarrollada por la Universidad de Murcia, denominada UMU-PKIv6, pero en la que los entornos de certificación multidominio no se incluían como característica obligatoria para las estructuras actuales basadas en nuevos modelos de confianza mucho más complejos en su definición. Las pruebas de evaluación se han realizado en un entorno cerrado de laboratorio, aunque el algoritmo para la construcción y validación de caminos de certificación se ha traslado su evaluación, posteriormente, a un entorno real en producción, donde se ha verificado que los tiempos en el rendimiento son aceptables por un usuario del Servicio de Validación, y que pueden ser totalmente asumibles en cualquier entorno de certificación multidominio.

Los nuevos servicios que aquí se han diseñado, implementado y validado se pueden poner en marcha en cualquier escenario multidominio, sobre el que se puede desplegar una federación de PKIs con un nuevo sistema de gestión de la confianza multidominio que es capaz de aportar ciertos niveles de seguridad para el intercambio de información entre sus miembros. Los siguientes capítulos se centran sobre un escenario multidominio en particular, donde el despliegue y la puesta en marcha de una federación de PKIs es un requisito indispensable para que la detección de ataques distribuidos tenga el éxito esperado, haciendo que el intercambio de la información de detección entre los distintos dominios de seguridad y/o administrativos, principalmente alertas generadas por cada IDS de forma individual, se lleve a cabo de forma segura entre todas las partes.

Capítulo 4

Confianza en redes colaborativas de detección de intrusiones

Éste y el siguiente capítulo se centran en dos objetivos principales, con una meta en común: la detección de ataques distribuidos en escenarios multidominio. El propósito detrás del primer objetivo es definir los mecanismos de colaboración necesarios entre un conjunto de *Sistemas de Detección de Intrusiones* (del inglés Intrusion Detection System, IDS), que posibiliten la detección de este tipo de ataques. La unión de todos cimenta la idea de un *Sistema Colaborativo de Alertas* (del inglés Collaborative Alert System, CAS) con el que poder construir una base común de *conocimiento interdominio* a partir del intercambio de las alertas aisladas que cada uno detecta individualmente. Cada dominio administrativo del CAS puede desplegar sus IDSs en sistemas autónomos internos a fin de maximizar la precisión en sus procesos de detección, formando cada uno una *Red Colaborativa de Detección de Intrusiones* (del inglés Collaborative Intrusion Detection Network, CIDN) capaz de construir una base de *conocimiento intradominio* local, con las alertas detectadas internamente en dicho dominio de seguridad.

La seguridad en el intercambio de alertas en redes abiertas necesita adoptar ciertos mecanismos de protección frente a las amenazas que puedan comprometer el contenido enviado por esas redes (*confidencialidad* e *integridad*), o el envío desde entidades que no legítimas en el CIDN o en el CAS (*autenticidad*). Estas tres propiedades han sido el foco de atención del capítulo anterior, cuyos resultados con tecnologías basadas en criptografía de clave pública le permiten al sistema colaborativo para la detección de ataques distribuidos que se vea amparado bajo un marco de seguridad multidominio, con el que conseguir evidencias más confiables sobre la posible ejecución de un ataque. A pesar de ello, la autenticidad de cualquier sistema de detección no implica que sus alertas representen incidentes que hayan ocurrido en la realidad. El envío de alertas fraudulentas o falsas a raíz de un comportamiento malicioso, al verse comprometido, por ejemplo, como consecuencia de un ataque de seguridad, pueden acarrear un compromiso grave de todo el sistema colaborativo de detección, haciendo que se vea decrementada su precisión a la hora de detectar ataques, especialmente los distribuidos, creyendo que se ha producido una amenaza cuando realmente no es cierto.

El segundo objetivo se centra entonces en hacer frente a la identificación de alertas fraudulentas, para que éstas sean automáticamente eliminadas y no provoquen errores dentro de los mecanismos de detección del resto de entidades del sistema colaborativo de detección. La intención detrás de este objetivo es desarrollar un modelo de gestión de la confianza con el que se pueda determinar la *bondad* de los emisores de las alertas antes de que éstas sean distribuidas al resto de miembros del sistema. Concretamente, el interés es hacer uso de modelos de confianza basados en reputación que sean capaces de identificar los comportamientos deshonestos –generación de alertas fraudulentas por sistemas de detección maliciosos– a partir de las interacciones que esos emisores hayan tenido en el pasado con el sistema. Este proceso de evaluación en el comportamiento de los sistemas de detección, antes de que sus alertas sean compartidas, procura mejorar la precisión en la detección de ataques, excluyendo las alertas fraudulentas recibidas desde sistemas de detección con comportamientos malintencionados.

La metodología que se desarrolla en este capítulo consiste en, primero, presentar el diseño de un sistema colaborativo de alertas (CAS) para la detección de ataques distribuidos, desacoplando su diseño entre distintos sistemas autónomos independientes como redes colaborativas de detección de intrusiones (CIDN). En un segundo punto, se describen en detalle los módulos que deben implementar cada entidad de un CIDN para construir las dos bases de conocimiento que son necesarias para la detección de ataques distribuidos: una base de conocimiento intradominio con las alertas generadas a nivel local en un CIDN, marcada como primer objetivo de este capítulo, y una base de conocimiento global definida a nivel interdominio que es compartida entre todos los CIDNs del CAS, cuya descripción completa se aborda en el siguiente capítulo.

Posteriormente, se presenta el diseño de un mecanismo de *confianza intradominio* basado en reputación, orientado a mejorar la cobertura de detección local de un CIDN con el que poder eliminar alertas fraudulentas generadas por IDSs maliciosos con un mal comportamiento. El siguiente capítulo se centra en diseñar un mecanismo de *confianza interdominio* basado en reputación, tomando como base el presentado en este capítulo, el cual va a ser capaz de identificar CIDNs maliciosos antes de compartir información de detección fraudulenta a nivel global del CAS. Finalmente, se utiliza un escenario de pruebas donde comprobar, a través de una serie de experimentos, cómo los IDSs son gradualmente aislados conforme sus comportamientos empeoran en el tiempo.

4.1. Diseño de un sistema colaborativo de alertas

En esta sección se presenta en detalle el diseño de un sistema colaborativo de alertas (CAS) para la detección de ataques distribuidos, el cual se compone de un conjunto de sistemas autónomos con capacidades de detección de actividades sospechosas acaecidas dentro de sus ámbitos locales. Cada uno de estos sistemas autónomos, principal foco de atención de este capítulo, son conocidos como redes colaborativas de detección de intrusiones (CIDN), cuya arquitectura interna es posteriormente detallada para cada uno de los sistemas de detección que conforman uno de esos CIDNs.

4.1.1. Descripción de la arquitectura del sistema

En la Sección 2.3.1 se presentaba un análisis sobre los criterios que se deben tener en cuenta al diseñar cualquier tipo de sistema colaborativo, independientemente del escenario final de aplicación. En esa sección se argumenta que la topología a desplegar es uno de los puntos clave de mayor relevancia, al suponer un alto impacto en términos de robustez y de escalabilidad. De entre todas las topologías analizadas, la opción de desplegar estratégicamente los IDSs siguiendo un *esquema parcialmente descentralizado* es la que mejor se ajusta a los requisitos introducidos al comienzo de este capítulo.

A continuación se define el diseño de un CAS, a modo de formalización mediante una serie de definiciones, siguiendo un modelo de despliegue parcialmente descentralizado. Este esquema topológico permite dar respuesta a los inconvenientes que presentan los otros tres esquemas, entre los que se incluyen: punto único frente a fallos y ataques, gran cuello de botella para el rendimiento y la falta de escalabilidad, todos inherentes en los esquemas centralizados; escalabilidad limitada y grandes problemas de robustez, innatos en los esquemas jerárquicos; y la elevada sobrecarga en las comunicaciones de mensajes por inundación de los esquemas totalmente descentralizados.

Definición 1. El *Sistema Colaborativo de Alertas* (CAS) se estructura en un conjunto $CAS = \{CIDN_1, CIDN_2, ..., CIDN_l\}$, donde l es el número de redes colaborativas de detección de intrusiones (CIDN) que lo componen.

La distribución de los distintos componentes que definen un CAS, entre múltiples CIDNs, permite que estos últimos sean capaces de aportar un mecanismo colaborativo para construir y compartir un conocimiento colectivo, a nivel interdominio, con el que poder detectar ataques distribuidos. La detección de este tipo de ataques la realizan los distintos IDSs que cada CIDN tiene estratégicamente desplegados en su red interna de detección, intercambiando entre los CIDNs del CAS solamente aquellas alertas que tengan un significado relevante para la detección de ataques distribuidos.

Definición 2. Cada *Red Colaborativa de Detección de Intrusiones* (CIDN) que forma parte del CAS, $CIDN_i \in CAS$ con $1 \leq i \leq l$, se define formalmente como un conjunto $CIDN_i = \{PS_i, IDS_1, IDS_2, ..., IDS_p\}$, donde PS_i es el Servicio de Publicación de $CIDN_i$ y p el número de IDSs con capacidades de detección en $CIDN_i$. Cada IDS_j, con $1 \leq j \leq p$, representa uno de los m HIDSs o n NIDSs, siendo habitualmente $n \uparrow\uparrow$ a fin de tener una gran variedad de puntos de monitorización de la red de detección.

Cuando un IDS –un HIDS o un NIDS– genera una alerta dentro de su CIDN, éste la envía al *Servicio de Publicación* (del inglés Publication Service, PS) que proporciona su CIDN para que sea compartida con el resto de IDSs. (El modelo de comunicaciones utilizando un Servicio de Publicación se explica en detalle en la Sección 4.1.2.) Antes de compartir cualquier alerta con el resto de IDSs de su CIDN, en esta propuesta de tesis doctoral se establece que el PS debe obtener el consentimiento de un grupo de expertos, llamado *Comité de Sabios* (del inglés Wise Committee, WC), encargado de evaluar las alertas y tomar una decisión de publicación dependiendo de quién las emitió.

Si la alerta proviene de una entidad benévola, el WC la acepta como válida (después de comprobarlo con el modelo de reputación intradominio definido en la Sección 4.3) y se lo notifica al PS para que, finalmente, la publique al resto de miembros del CIDN. En caso contrario, se deniega su publicación al considerar el WC que es una información fraudulenta proveniente de una entidad maliciosa. Como consecuencia de este proceso, ninguna alerta es publicada sin el consentimiento previo del WC.

Definición 3. El *Comité de Sabios* (WC) de un $CIDN_i \in CAS$, siendo $1 \leq i \leq l$, se define como un conjunto $WC_i = \{NIDS_1, NIDS_2, ..., NIDS_q\} \in CIDN_i$, donde q $(1 \leq q \leq n)$ es el número de NIDSs escogidos dentro de $CIDN_i$ como representantes al ser los NIDSs con mayor reputación. El proceso de elección de qué NIDSs se convierten en miembros de un WC en particular se explica en la Sección 4.3.2.

El WC de un CIDN se ha diseñado como un panel de expertos para conceder o bien denegar tanto la publicación de las alertas generadas por los IDSs como, opcionalmente, compartir también las *reglas de detección* que se han disparado en esos IDSs durante el proceso de detección. Los miembros del WC evaluarán si las alertas generadas por una entidad del CIDN son maliciosas o no, según la confianza depositada en el emisor de las mismas. Debido a que el WC es un elemento crucial para el éxito de un CIDN, los miembros de un WC serán seleccionados entre los NIDSs más reputables y confiables del CIDN. Los HIDSs nunca formarán parte de este comité ya que, según nuestro diseño, suelen pertenecer a usuarios finales que deciden, voluntariamente, unirse y colaborar con el CIDN de su mismo dominio de seguridad, por lo que el sistema (inicialmente) no los puede considerar como altamente confiables. En cambio, los NIDSs son unidades de detección bien conocidas para el sistema ya que son instaladas y gestionadas por los administradores del dominio de seguridad, expertos en el campo de la seguridad.

De entre los distintos NIDSs que conforman el WC, en un momento concreto, se ha seleccionado uno de ellos como el máximo representante o líder de su comité.

Definición 4. El *Líder del Comité de Sabios* (WCL) de un $WC_i \in CIDN_i$, siendo $1 \leq i \leq l$, se define como $WCL_i = \{NIDS_r \mid NIDS_r \in WC_i,\ Rep_i(NIDS_r) > Rep_i(NIDS_s)\ \forall NIDS_{s \neq r} \in WC_i\}$, donde $NIDS_r$ se selecciona como representante de WC_i, y en consecuencia de todo $CIDN_i$, al ser la entidad más confiable y reputable de entre todos los q NIDSs que forman el i-ésimo WC.

El WCL tiene como función principal, entre otras, agregar las recomendaciones de todos los miembros del CIDN con sus opiniones sobre un IDS en concreto, calcular el valor de reputación del IDS emisor de una alerta y decidir si publicarla, o no, según los dos niveles comentados anteriormente: internamente al resto de IDSs del CIDN a través del PS (*modelo intradominio*) y al resto de CIDNs participantes del CAS (*modelo interdominio*). Debido a este último tipo de publicación, el WCL también se encarga de compartir el conocimiento interno sobre alertas de su CIDN con el resto de dominios de seguridad. De esta manera, a través de los WCLs, el sistema de alertas cooperativo tendrá una visión más global de lo que está ocurriendo en su totalidad.

Como ejemplo de un sistema colaborativo para la detección de ataques distribuidos, la Figura 4.1 muestra los CIDNs de cinco dominios de seguridad, cada uno incluyendo un número indeterminado de NIDSs y HIDSs.

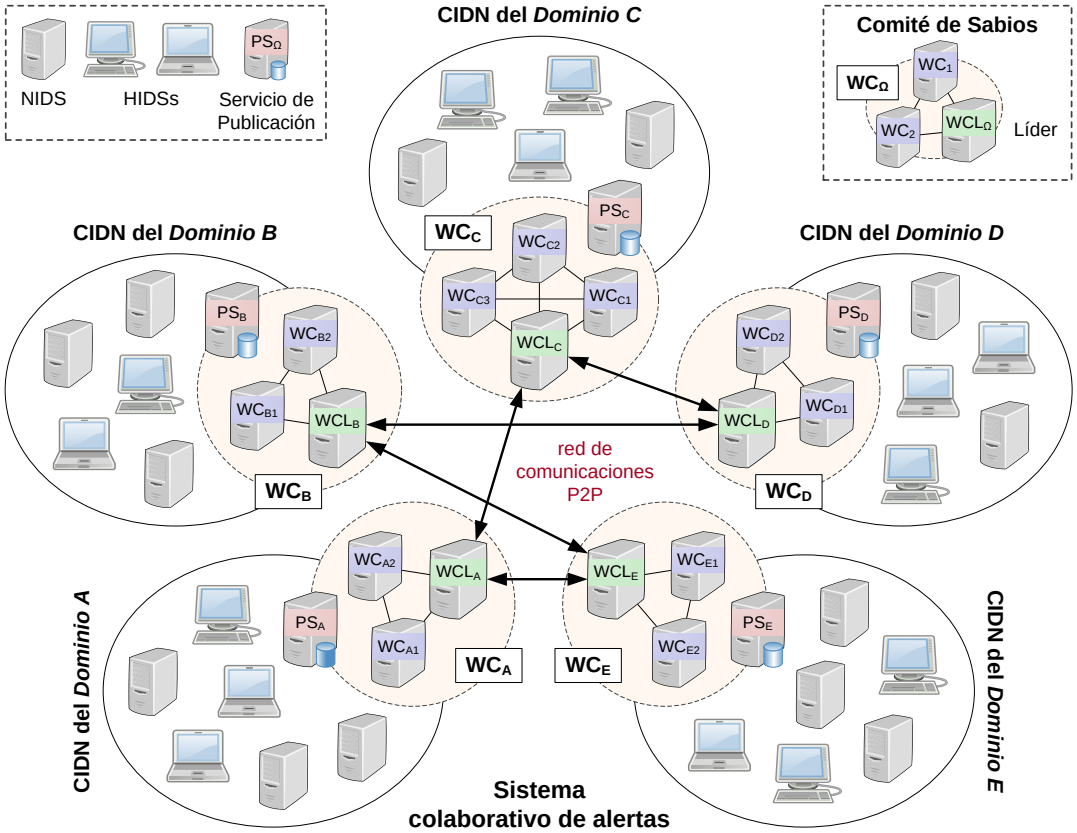


Figura 4.1: Arquitectura de un sistema colaborativo de alertas con cinco CIDNs

El modelo intradominio se basa en que todos los IDSs de un mismo CIDN tienen que trabajar de forma conjunta a nivel local, como una gran comunidad con un mismo objetivo en común, compartiendo todas las alertas que cada uno de esos IDSs detecta individualmente. Además de todas esas alertas, los IDSs de un CIDN también van a ser capaces de intercambiar valores de recomendación y reglas de detección.

Un valor de *recomendación* representa la confianza que un IDS tiene con respecto a otro, según las interacciones que los dos hayan realizado en el pasado. Estos valores son la base para la definición de un modelo de reputación intradominio, que se presenta después en la Sección 4.3, con el cual determinar si un IDS en particular exhibe algún tipo de comportamiento malicioso. Además, los IDSs de un CIDN también van a poder intercambiar internamente las *reglas de detección* que han sido "disparadas" durante el proceso de detección de cualquier alerta. Este intercambio es una manera de mejorar la cobertura de la detección compartiendo nuevo conocimiento con el que adquirir formas de detección de nuevos ataques desconocidos hasta el momento, sobre todo para IDSs que no tengan configurada la última actualización de las firmas de ataque.

Por otro lado, en el modelo interdominio para la construcción de una base global de conocimiento a nivel del CAS, se hace necesario que todos los dominios de seguridad compartan las alertas que cada uno genera internamente dentro de su propio CIDN, intercambiando únicamente aquellas alertas que cada CIDN considere relevantes para la detección de un ataque distribuido. En este caso, no es necesario el intercambio de las reglas de detección, ya que las utilizadas en un dominio podrían tener un significado muy distinto para otro. Las reglas de detección están configuradas según la estructura interna y los servicios que cada dominio de seguridad ofrece.

Como en el caso intradominio, además de las alertas también se comparten valores de recomendación, indicando la confianza que un CIDN tiene con respecto a otro para el cálculo de su reputación, teniéndose de nuevo en cuenta las interacciones que ambos CIDNs han tenido en el pasado. Estos valores de recomendación son, por tanto, la base para la definición del modelo de reputación interdominio, que se presenta en el siguiente capítulo, con el que un CIDN será capaz de determinar si la alerta compartida por otro CIDN es o no fraudulenta según el valor de reputación que tenga este último.

4.1.2. Comunicaciones en el sistema de detección

El enrutamiento de las alertas que un IDS genera de forma individual, y que desea compartir con el resto de IDSs del CIDN, junto con toda la información necesaria para su correcto funcionamiento, se realiza mediante un *modelo de publicación/suscripción*. Como se ha comentado en el punto anterior, el Servicio de Publicación (PS) actúa como una entidad intermediaria en la red de comunicaciones entre los distintos componentes de un CIDN: entre los IDSs y los miembros del WC, encargados de tomar una decisión de si publicar o no las alertas generadas por esos IDSs. De esta manera, PS_Ω ofrece un mecanismo asíncrono de las comunicaciones a nivel intradominio para el intercambio de mensajes entre todos los IDSs del dominio genérico Ω [225].

Como ejemplo de un Servicio de Publicación, la Figura 4.2 muestra cómo el líder del WC de un dominio de seguridad genérico Ω, denotado por WCL_Ω, se puede comunicar a través de este servicio con todos los IDSs de su CIDN.

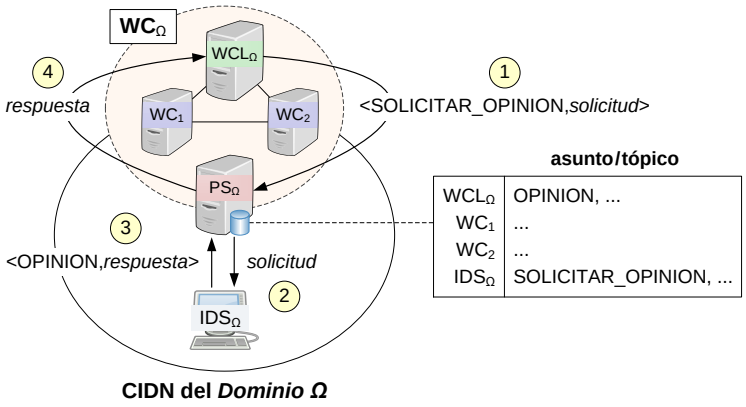


Figura 4.2: Ejemplo de un sistema de publicación/suscripción

En un modelo de publicación/suscripción, como el que se muestra en la Figura 4.2 a modo de ejemplo, los IDSs se pueden suscribir a los *asuntos* o tópicos predefinidos por los administradores, según las funcionalidades que tengan en el CIDN, para recibir todas las notificaciones acerca de los asuntos en los que estén suscritos. Cuando el PS reciba un nuevo mensaje, una copia será reenviada a todos los IDSs suscritos bajo el mismo asunto con el que el mensaje haya sido publicado. Este mecanismo es necesario para los CIDNs de todos los dominios de seguridad, siendo obligatorio el despliegue y mantenimiento de su propio PS a fin de habilitar el intercambio de información entre todos sus componentes. En [226], se presentan los procesos de cómo los IDSs se pueden dar de alta –suscripción– o de baja en un servicio de publicación/suscripción.

En el ejemplo de la Figura 4.2 se muestra cómo WCL_{Ω} puede solicitar a los IDSs de su CIDN algún tipo de información en el que esté interesado, como recomendaciones sobre cualquier otro IDS del CIDN (proceso explicado en la Sección 4.4) para el cálculo de la confianza de este último a través de su reputación. En este caso de ejemplo, WCL_{Ω} envía un mensaje a PS_{Ω} con la solicitud para obtener las recomendaciones de sus IDSs, usando para ello el asunto `SOLICITAR_OPINION`. PS_{Ω} recupera la lista de IDSs suscritos en ese asunto, y les reenvía el mensaje original de WCL_{Ω}. Como respuesta, esos IDSs devuelven sus recomendaciones usando el asunto `OPINION`.

Por otro lado, los WCLs de cada CIDN también pueden intercambiar las alertas, además de las recomendaciones que tienen sobre otros dominios de seguridad, para la construcción de la base global de conocimiento que necesita el CAS para sus propósitos de detección de ataques distribuidos. Ese intercambio se realiza a través de un modelo de enrutamiento basado en una red segura Peer-to-Peer (P2P), como se muestra en la Figura 4.1 [227]. Esta forma de comunicación permite un mecanismo de comunicaciones a nivel interdominio entre los CIDNs de todos los dominios de seguridad implicados en el CAS, siendo esta gestión interdominio el foco principal del siguiente capítulo.

Para que el intercambio de información de detección tenga el éxito esperado, tanto a nivel intradominio a través del PS como a nivel interdominio mediante la red P2P, es necesario el uso de ciertos mecanismos con los que poder llevar a cabo ese intercambio de la manera más segura y confiable posible. El uso de esos mecanismos de seguridad son obligatorios: entre los líderes de cada WC, a fin de construir el conocimiento colectivo de alertas a nivel del CAS; y a nivel intradominio dentro de un CIDN, entre los NIDSs del WC y entre el PS y el resto de los NIDSs del CIDN. En cambio, el soporte de estos mecanismos de seguridad no es obligatorio para los HIDSs, al ser entidades de detección que suelen pertenecer a usuarios finales que no forman parte de la infraestructura.

Como mecanismo para el intercambio seguro de la información de detección, tanto de las alertas generadas por los IDSs como de las recomendaciones que son necesarias para el correcto funcionamiento de los sistemas de confianza y reputación, se propone el uso de soluciones basadas en criptografía de clave pública mediante certificados X.509, como el propuesto en el Capítulo 3. Para este caso, es necesario el uso del Servicio de Validación presentado en la Sección 3.2.2, el cual posibilita que se puedan validar las credenciales criptográficas de las unidades de detección antes de que establezcan un canal seguro entre ellas para el intercambio de su información de detección.

4.2. Arquitectura de un componente del CIDN

Esta sección presenta la arquitectura que, según nuestro diseño, cualquier entidad de un CIDN tiene que adaptar para maximizar la precisión en la detección de los ataques distribuidos. También se define cómo se integra el modelo de reputación intradominio, presentado en la siguiente sección, a fin de detectar alertas fraudulentas.

4.2.1. Componentes de una entidad del CIDN

En la Figura 4.3 se muestran los módulos que cada miembro del CIDN debe soportar internamente, los cuales han sido agrupados por cinco bloques funcionales. Los NIDSs del CIDN tienen que implementar obligatoriamente esta arquitectura, pero no así los HIDSs, ya que suelen pertenecer a usuarios finales que podrían definir las características de detección que consideren más apropiadas. En el caso de los HIDSs, sólo es obligatorio el submódulo *Publicación/Suscripción*, para enviar y recibir información hacia y desde el CIDN. En los siguientes apartados se desglosan en detalle cada uno de los cinco bloques funcionales en los que está compuesta una entidad cualquiera del CIDN.

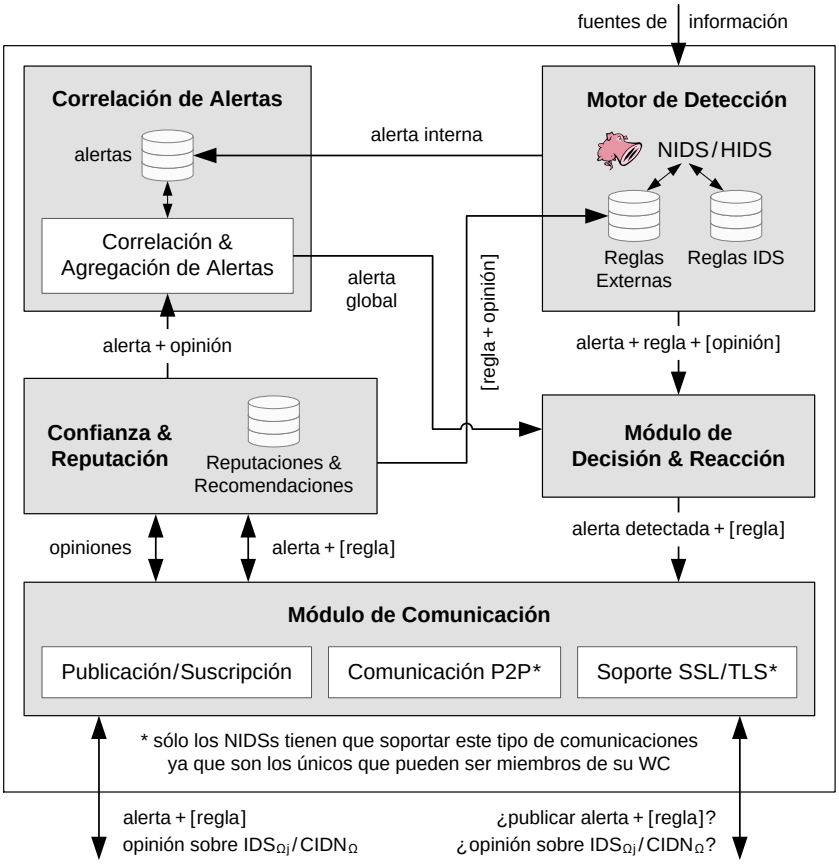


Figura 4.3: Componentes que conforman una entidad del CIDN

Módulo de Comunicación

Este bloque funcional se encarga de compartir la información relativa a las alertas generadas, opcionalmente las reglas de detección y las recomendaciones que mantiene este IDS con respecto a otros IDSs del CIDN, o con otros CIDNs si la entidad es el WCL. Este módulo se divide, a su vez, en tres submódulos, según el tipo de comunicación que el IDS necesita realizar. El submódulo *Publicación/Suscripción* se encarga de publicar y recibir información interna relacionada con el CIDN. Esta información se intercambia a través del Servicio de Publicación (PS), y su contenido puede variar dependiendo de la funcionalidad requerida en un momento dado:

- La publicación de una nueva alerta producida internamente por este IDS, junto con, posiblemente, la regla de detección causante de haberla generado.
- Una solicitud desde el WC preguntando sobre la opinión, o recomendación, que este IDS tiene sobre cualquier otro IDS concreto del CIDN.
- La respuesta de recomendación por parte de este IDS asociada con la solicitud anterior, la cual será reenviada al WC para su evaluación.
- La publicación final de una alerta, junto con su regla de detección relacionada, si ésta fue enviada, el IDS que la generó y la decisión tomada por el WC indicando si es o no fraudulenta a raíz de un comportamiento malicioso. Esta información tiene que ser normalizada según el software de detección instalado en cada uno de los IDSs, como se detalla en la Sección 4.2.2.

Los otros dos submódulos que forman parte del *Módulo de Comunicación* solamente son obligatorios y utilizados por los miembros candidatos a ser parte del WC; es decir, por todos los NIDSs del CIDN. El submódulo *Soporte SSL/TLS* incluye características de seguridad a las comunicaciones para protegerlas frente a muchas de las amenazas que intentan comprometer la confidencialidad y la integridad, fundamentalmente. Esta seguridad se proporciona mediante, por ejemplo, el uso de criptografía de clave pública, con las técnicas basadas en certificados X.509 definidas en el capítulo anterior.

El submódulo *Soporte SSL/TLS* se utiliza en dos tipos de comunicaciones distintas. Primero, entre los propios NIDSs del WC, así como entre el PS y esos NIDSs anteriores del WC con el objetivo de intercambiar las distintas recomendaciones que cada uno de ellos almacena para realizar el cálculo del valor de reputación de un IDS en particular. Es decir, en este primer punto se establecen los requisitos de seguridad que posibilitan la ejecución de las comunicaciones intradominio. Segundo, este submódulo también se utiliza para proporcionar los mecanismos de seguridad necesarios en las comunicaciones interdominio; es decir, entre más de un CIDN. Por tanto, el submódulo *Comunicación P2P* hará uso del anterior (*Soporte SSL/TLS*) para intercambiar de forma segura todas las alertas generadas entre los distintos CIDNs. Este submódulo también se utiliza para llevar a cabo el intercambio de la información de reputación que se necesita para decidir si el CIDN que envió la alerta es suficientemente confiable, y así poder considerar esa alerta como válida –no fraudulenta por un comportamiento malicioso.

Confianza y Reputación

El módulo *Confianza y Reputación* se encarga de calcular y compartir los valores de reputación y recomendación que el IDS tiene con respecto a otros; valores que almacena internamente en una base de datos conocida como *Reputaciones & Recomendaciones*. Todos los IDSs tendrán una recomendación, u opinión, acerca del resto de IDSs de su mismo CIDN, siempre y cuando estos últimos hayan publicado alertas, aunque sólo los NIDSs del WC tienen la competencia de calcular los valores finales de reputación.

Como datos de entrada, este módulo puede recibir dos mensajes diferentes:

- Solicitudes desde el WC preguntando por la recomendación que este IDS mantiene con respecto a otros IDSs de su CIDN. Para este mensaje, el módulo *Confianza y Reputación* recibe o una solicitud de recomendación (*¿cuál es tu opinión sobre el IDS j?*) o una respuesta sobre cuál ha sido la opinión que otro IDS distinto tiene con respecto a otro tercer IDS del CIDN. La solicitud para esta última respuesta tiene que haber sido, obviamente, enviada previamente por el WC.

- Publicación de nuevas alertas, posiblemente junto con la regla de detección que lanzó cada una. En este caso, el módulo *Confianza y Reputación* recupera desde la base de datos *Reputaciones & Recomendaciones* la opinión almacenada para el IDS que envió la alerta y, opcionalmente, la regla de detección. La respuesta de recomendación se adjunta a los datos de entrada y se reenvían internamente en dos partes separadas: la dupla <`alerta,recomendación`> se envía al módulo *Correlación de Alertas* y la dupla <`regla,recomendación`> al *Motor de Detección*, si, y solo si, la regla de detección se incluyó en los datos de entrada.

El modelo concreto de reputación que se ha diseñado, a nivel intradominio, se detalla posteriormente en la Sección 4.3.

Motor de Detección

El *Motor de Detección* es el módulo que habilita a cualquier IDS con capacidades de detección de actividades sospechosas dentro de su ámbito local. Este módulo hace uso de un modelo de detección de usos indebidos (*misuse detection*) para detectar las actividades sospechosas dentro del host o en la red de comunicaciones, dependiendo de si la entidad que implementa este módulo es un HIDS o un NIDS, respectivamente. En cualquier caso, estos dos tipos de entidades hacen uso de técnicas de detección basadas en firmas o patrones, donde las fuentes de información se comparan contra una base de datos de firmas de ataques conocidos para descubrir cualquier anomalía en el sistema que están monitorizando. Esta base de datos de firmas de ataque la gestiona el *Motor de Detección* internamente bajo el nombre *Reglas IDS* (ver Figura 4.3).

Como soluciones que actualmente pueden encontrarse en el mercado, el propietario de la entidad (el usuario final del HIDS o el administrador del NIDS) puede utilizar soluciones de IDS existentes como OSSEC [228], si la entidad es un HIDS, o soluciones bien conocidas como Snort [10], si la entidad es un NIDS.

En cualquiera de los casos anteriores, el IDS se alimenta de las firmas de ataque que el vendedor crea para su software en cuestión, que suelen obtenerse desde su página Web. Esas actualizaciones nutren al IDS de nuevo conocimiento para detectar nuevas amenazas imprevistas hasta el momento. Sin embargo, la actualización de estas bases de datos sufren de dos inconvenientes: i) el propietario de la entidad puede instalarlas demasiado tarde en su IDS, cuando el ataque ya haya causado un daño inesperado; y ii) algunas soluciones proporcionan las actualizaciones de sus bases de datos de firmas con mayor celeridad que otras, para prevenir, por ejemplo, ataques de día cero.

Para mantener una copia de la base de datos de firmas de ataque lo más actualizada posible, incluyendo las firmas de otras soluciones de IDS con el objetivo de maximizar el ratio o la cobertura de la detección, se ha extendido la funcionalidad del *Motor de Detección* con una nueva base de datos de firmas de ataque, llamada *Reglas Externas* en la Figura 4.3. En esta nueva base de conocimiento, se almacenarán todas las firmas de ataque publicadas por otros IDSs confiables del CIDN, difundidas conjuntamente con la alerta que cada regla lanzó durante su proceso de detección.

Como la entidad no mantiene información sobre la satisfacción de ninguna regla de detección, la base de datos *Reglas Externas* también almacenará, junto a cada regla, la opinión actual sobre el IDS que la publicó. Este valor de recomendación lo tiene que haber calculado antes el módulo *Confianza y Reputación* con la información interna que gestiona sobre ese IDS. De esta manera, la recomendación ofrecerá al *Motor de Detección* una idea sobre cómo de buena y confiable es la firma de ataques para detectar nuevo conocimiento sobre amenazas desconocidas hasta el momento.

Cuando el IDS detecte un nuevo incidente, la alerta correspondiente se la enviará al *Módulo de Decisión y Reacción* junto con la firma que se disparó. Si ésta se encuentra en la base de datos *Reglas Externas*, el *Motor de Detección* adjuntará la recomendación que este IDS tiene en el IDS emisor de la misma. En caso contrario, es de la base de datos *Reglas IDS*, no se adjuntará una recomendación ya que fue instalada por el propietario del IDS y, debido a ello, debe de tener una total confianza para el *Motor de Detección*. Por tanto, las firmas almacenadas en *Reglas IDS* siempre tendrán preferencia frente a las de *Reglas Externas*, evitando así la existencia de firmas con un cierto solapamiento que hagan que el IDS pueda generar más de una alerta notificando de la ocurrencia de un mismo incidente. Las firmas en la base de datos *Reglas Externas* podrán trasladarse con el tiempo a la de *Reglas IDS*, bajo las dos siguientes condiciones:

- La entidad que gestiona este módulo tiene la suficiente confianza en el IDS original que emitió la regla de detección.
- La regla de detección ha demostrado una gran efectividad en la identificación de nuevas amenazas, que eran indetectables anteriormente por el IDS actual.

El proceso de propagación de firmas de ataque le proporciona al *Motor de Detección* una forma de difundir un conocimiento bien conocido por los IDSs del CIDN. Con ese conocimiento, los IDSs estarán en disposición de detectar amenazas que, posiblemente, no podrían haber hecho sin ese soporte y sin la colaboración de los distintos IDSs, por supuesto, para que esa detección pueda llevarse a cabo en entornos distribuidos.

Correlación de Alertas

Este módulo agrega y correlaciona la enorme cantidad de alertas que generan los IDSs, creando grupos representativos bajo un mismo significado semántico a través de relaciones lógicas, con las que dar una visión más global de los ataques que pueden estar ocurriendo. Evitar que se genere un alto número de alertas con un mismo significado permite que se reduzca, en todo lo posible, el número de falsos positivos que los IDSs pueden generar por separado, y así propiciar la detección de ataques de más alto nivel. Este proceso de agregación y correlación recibe como entrada las alertas almacenadas en una base de datos interna, donde el submódulo *Correlación & Agregación de Alertas* va almacenando todas las alertas que el *Motor de Detección* ha generado internamente, así como las alertas externas que recibe desde otros IDSs del CIDN.

La implementación del submódulo *Correlación & Agregación de Alertas* puede hacer uso de un amplio abanico de técnicas existentes, como las que ofrecen la detección de escenarios de ataques predefinidos [229], o bien el agrupamiento de las alertas en grupos (*clusters*) según las características como, por ejemplo, la procedencia del ataque [230]. Independientemente de la técnica, la salida de este módulo, como resultado del proceso de agregación y correlación, es una única alerta de alto nivel que sintetiza la anomalía detectada. Esta alerta es enviada entonces al *Módulo de Decisión y Reacción* para que sea evaluada, antes de ser publicada y compartida con el resto de IDSs del CIDN.

Módulo de Decisión y Reacción

El *Módulo de Decisión y Reacción* recibe todas las alertas que han sido detectadas desde dos fuentes internas distintas: las alertas aisladas detectadas por el IDS instalado en el *Motor de Detección* y las alertas de alto nivel generadas por el módulo *Correlación de Alertas*. Ambos tipos de alertas son evaluadas por el *Módulo de Decisión y Reacción* para determinar si son suficientemente confiables en que hayan detectado una amenaza real antes de que sean compartidas con el resto de IDSs del CIDN.

Como regla general, las alertas de alto nivel generadas por el módulo *Correlación de Alertas* siempre serán publicadas, ya que representan potenciales anomalías críticas para el sistema. Por ejemplo, la detección de un ataque polimórfico complejo. Por otro lado, las alertas aisladas que han sido generadas por el *Motor de Detección* podrían no representar amenazas reales en su ámbito de detección, por lo que la entidad tiene que estar segura de su validez, en todo lo posible, antes de que sean compartidas con el resto del CIDN. Si alguna de esas alertas se publicase, aun no siendo válidas, el WC siempre tiene la última potestad durante su evaluación de no publicarla finalmente. Para ello, las alertas se envían al *Módulo de Comunicación*, y el submódulo *Publicación/Suscripción* es el encargado de reenviar las alertas al WC a través del PS que el dominio de seguridad tiene desplegado. La entidad actual también podría adjuntar la regla de detección que lanzó la alerta, siempre y cuando la haya detectado el *Motor de Detección*. Además, si la entidad es el WCL, la alerta detectada en su CIDN se tiene que propagar al resto de dominios de seguridad a través del submódulo *Comunicación P2P*, siempre y cuando la alerta tenga un significado relevante en la detección de ataques distribuidos.

4.2.2. Normalización de alertas y reglas de detección

Uno de los principales desafíos en el intercambio de información entre los miembros de un CIDN es su falta de interoperabilidad, al coexistir múltiples formatos estándares y propietarios para codificar la salida de los IDSs –alertas y reglas de detección. Por tanto, es necesario desplegar un *Servicio de Traducción* que normalice esta información a un formato común lo más estándar posible. Como un CIDN puede estar compuesto por un gran número de entidades heterogéneas, se ha desacoplado la función de normalización de esas entidades desplegando dicho servicio a nivel organizacional, internamente dentro del PS como un submódulo denominado *Módulo de Traducción*. Esto hace que no sean necesarios mecanismos de seguridad para proteger sus comunicaciones.

A continuación, se describen los posibles formatos estándares con los que se puede alcanzar una correcta interoperabilidad entre los miembros de un CIDN, dependiendo de si el formato está orientado a las alertas o a las reglas de detección. Una descripción completa de estos formatos se puede encontrar en [231], así como el trabajo presentado en [232] donde se puede revisar un análisis comparativo entre algunos de ellos.

- *Definición de alertas*. Las alertas se pueden normalizar a uno de los dos formatos estándar como IDMEF (*Intrusion Detection Message Exchange Format*) [233] o IODEF (*Incident Object Description Exchange Format*) [234]. Aunque aquí se ha optado por IDMEF, al existir herramientas de código abierto que lo implementan para dar la salida de los IDSs en ese formato, como el plugin Snort IDMEF [235], el uso de IDMEF se podría extrapolar en las soluciones aportadas por esta tesis doctoral a IODEF sin cambios sustanciales. Nótese que estas soluciones también podrían usar formatos más recientes como STIX (*Structured Threat Information eXpression*) [236] o CybOX (*Cyber Observable eXpression*) [237], aunque ambos formatos todavía se encuentran en procesos de estandarización.

- *Definición de reglas de detección*. Se ha optado por *Common Intrusion Detection Signatures Standard* (CIDSS) [238], ya que es el único formato común y estándar con el que poder transportar reglas entre los diferentes IDSs. Con él, se pueden empaquetar elementos comunes a todos los tipos de firmas que existen: direcciones de origen y de destino, protocolos, patrones, etcétera. Además, también existe una implementación de código libre para gestionar reglas de detección en CIDSS [239].

4.3. Sistema de reputación intradominio

En esta sección se presenta, de manera formal, el mecanismo intradominio de gestión de la confianza basado en reputación, destinado a que un determinado CIDN sea capaz de detectar y, por tanto, evitar las alertas fraudulentas enviadas por alguno de sus IDSs. Estas alertas podrían haber sido causadas por IDSs sin malas intenciones, debido por ejemplo a una mala configuración, pero también las podría haber generado algún IDS con una explícita intención maliciosa al verse comprometido por un atacante, cuyo fin es alterar la visión del sistema sobre los activos que está protegiendo.

4.3.1. Cálculo de la reputación de un IDS

Los NIDSs del Comité de Sabios (WC) son los responsables de decidir si las alertas publicadas por los IDSs del CIDN son resultado de un comportamiento malicioso, para que el resto de IDSs las tenga o no en cuenta. Esta decisión depende de la reputación del IDS emisor dentro su dominio. Es decir, lo confiable o bueno que es para que sus alertas se consideren como válidas o benevolentes. Para calcular la reputación del j-ésimo IDS de un CIDN genérico Ω, denotado como $Rep_\Omega(j)$, se considera que éste intenta publicar una alerta en $CIDN_\Omega$. Para conceder o denegar esa publicación, WCL_Ω comprueba primero si la reputación de IDS_j ha sido calculada hace poco para que su valor se pueda considerar todavía como válido. Sino, nunca se ha calculado o el que hay es demasiado antiguo, WCL_Ω pregunta a todos los HIDSs y NIDSs de $CIDN_\Omega$, a excepción de IDS_j y a los miembros de WC_Ω, cuáles son sus recomendaciones sobre IDS_j.

Cuando WCL_Ω reciba todas esas recomendaciones, las agrega bajo un único valor que comparte con el resto de miembros de WC_Ω. Con ese valor, el i-ésimo miembro de WC_Ω evaluará su propia confianza sobre IDS_j, denotada como $T_{WC_{ij}}$, a través de (4.1).

$$T_{WC_{ij}} = \alpha_i \left(\bigoplus_{k=1,k\neq i}^{|WC_\Omega|} Rec_{WC_{kj}} \right) + \beta_i \left(\bigoplus_{k=1}^{n-q} Rec_{NIDS_{kj}} \right) + \gamma_i \left(\bigoplus_{k=1}^{m} Rec_{HIDS_{kj}} \right) \quad (4.1)$$

donde $Rec_{kj} \in [0,1]$ es la recomendación que tiene k sobre IDS_j; $\oplus$ una operación de agregación; $|WC_\Omega| = q$ el número de miembros de WC_Ω; n el número de NIDSs que no son de WC_Ω; m el número de HIDSs; y $\alpha_i, \beta_i, \gamma_i \in [0,1]$ cada uno de los pesos de las recomendaciones obtenidas desde, respectivamente, cada miembro de WC_Ω, los NIDSs y los HIDSs. Estos pesos deben cumplir que $\alpha_i + \beta_i + \gamma_i = 1$ y $\alpha_i \geq \beta_i \geq \gamma_i$, debiéndose este orden a que los miembros del WC son los más reputables y a que los NIDSs suelen ser más confiables que los HIDSs, al estar gestionados por administradores del CIDN.

Cada miembro de WC_Ω administra sus propios pesos (α_i, β_i y γ_i $\forall i \in [1, |WC_\Omega|]$) sobre las recomendaciones recibidas de los NIDSs de WC_Ω y de los IDSs de $CIDN_\Omega$. Deben ser diferentes ya que, sino, (4.1) siempre daría el mismo valor, dejando a relucir que todos los miembros de WC_Ω tendrían (injustamente) la misma confianza en IDS_j.

Para mejorar el rendimiento del sistema, reduciendo al máximo la sobrecarga en la red de comunicaciones, solamente WCL_Ω pregunta las recomendaciones sobre IDS_j a los miembros de $CIDN_\Omega$, en lugar de que cada miembro de WC_Ω lo haga por separado. De esta manera, la recomendación u opinión que la entidad k tiene sobre IDS_j, en la interacción o transacción t, se calcula mediante (4.2).

$$Rec_{kj}^{(t)} = \omega_k \cdot Rec_{kj}^{(t-1)} + (1 - \omega_k) \cdot Sat_{kj}^{(t-1)} \quad (4.2)$$

donde $\omega_k \in [0,1]$ es el peso de la interacción (opinión) anterior y $Sat_{kj}^{(t-1)} \in [0,1]$ la satisfacción de la entidad k con la última alerta enviada por IDS_j.

Después de que cada miembro de WC_Ω, WC_i $\forall i \in [1, |WC_\Omega|]$, haya calculado $T_{WC_{ij}}$ de acuerdo a (4.1), estos valores deben ser agregados para obtener la reputación global, denotada como $Rep_\Omega(j)$, que WC_Ω tiene sobre IDS_j dentro del CIDN Ω.

Esa operación de agregación la realiza WCL_Ω a través de (4.3).

$$\bigoplus_{i=1}^{|WC_\Omega|} T_{WC_{ij}} = \frac{\sum_{i=1}^{|WC_\Omega|} T_{WC_{ij}}}{|WC_\Omega|} = Rep_\Omega(j) \tag{4.3}$$

Para comprobar si miembros de WC_Ω son maliciosos, el valor promedio de todos los $Rep_\Omega(j)$, calculados con (4.3), sólo se considerará válido si la desviación estándar $\sigma_{\Omega j}$ de todas las recomendaciones dadas por los miembros de WC_Ω, calculado en (4.1) como $T_{WC_{ij}}$, es menor que un valor umbral establecido de antemano por el administrador del dominio Ω. Sino, hay un desacuerdo entre los distintos miembros de WC_Ω al calcular sus recomendaciones sobre el mismo IDS, indicando que alguno puede estar comprometido o que ha dejado de funcionar correctamente. En este caso, se le notifica al administrador de $CIDN_\Omega$ de este hecho, WC_Ω se disuelve y uno nuevo es elegido.

Si en cambio WC_Ω ha conseguido llegar a un alto acuerdo entre todos sus miembros, es decir, si $\sigma_{\Omega j} \leq \sigma_{\Omega,umbral}$, es necesario decidir en un último paso si la alerta recibida de IDS_j puede ser considerada como válida –verdadero positivo– o, en cambio, puede representar una alerta fraudulenta como consecuencia de un comportamiento malicioso. Este último paso lo realiza el propio WCL_Ω, y consiste en precisar el *nivel de confianza* de IDS_j según el valor actual de su reputación; es decir, según $Rep_\Omega(j)$.

La confianza se define como conjuntos difusos [240], donde hay una correspondencia directa entre cada nivel de confianza y los *niveles de severidad* de cada alerta –impacto del incidente sobre el activo de $CIDN_\Omega$. El estándar IDMEF clasifica cada alerta con un valor *Info* (función informativa), *Low*, *Medium* o *High*. Como ejemplo, en la Figura 4.4 se muestra que la probabilidad de IDS_j, según su reputación $Rep_\Omega(j)$, es directamente proporcional al valor de pertenencia de IDS_j en cada nivel de confianza i, denotado por ε_i. Según esa probabilidad, IDS_j puede emplazarse en un nivel u otro de confianza.

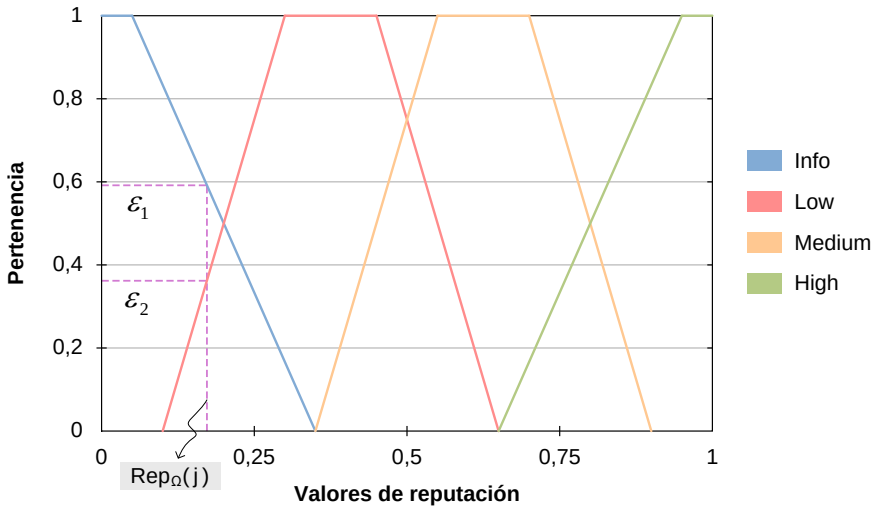


Figura 4.4: Niveles de confianza en el cálculo de la severidad de las alertas

Usando los conjuntos difusos anteriores, IDS_j podrá alertar de incidentes con mayor riesgo de impacto para el dominio de seguridad –alertas de mayor severidad– cuanto más confiable sea IDS_j (mayor reputación) para su CIDN. En el ejemplo de la Figura 4.4, si a IDS_j se le calcula el valor de reputación $Rep_\Omega(j)$ que ahí aparece, IDS_j estará en disposición de publicar alertas con severidad *Info* o *Low*, pero así *Medium* o *High*.

Finalmente, la satisfacción de la entidad k sobre la alerta generada y enviada por IDS_j, denotada en (4.2) como $Sat_{kj}^{(t)}$, depende en gran medida de dos factores:

- Una alerta con una severidad alta (o baja), generada por un IDS benevolente, se convertirá en una satisfacción alta (o baja) para la entidad receptora.
- Una alerta con una severidad alta (o baja), generada por un IDS malicioso, se convertirá en una alta (o baja) "desatisfacción" para la entidad receptora.

La combinación de ambos factores permite calcular el nivel de satisfacción, de la entidad k sobre IDS_j, a través de (4.4).

$$Sat_{kj}^{(t)} = 0{,}5 + Fancy_{kj}^{(t)} \cdot \frac{\delta_j}{2} \tag{4.4}$$

donde $\delta_j \in [0, 1]$ define el valor defuzzificado del conjunto difuso relacionado con la severidad de la alerta emitida por IDS_j y $Fancy_{kj}^{(t)} \in [-1, 1]$ la suposición de la entidad k en que la alerta emitida por IDS_j en la transacción t no es fraudulenta –el incidente ha ocurrido realmente, por lo que la alerta no ha sido generada maliciosamente.

Este último *valor de suposición* podría variar según varios factores, dependiendo de si la entidad k tiene alguna evidencia sobre la alerta emitida por IDS_j o, por el contrario, es una alerta de la cual no tiene ningún indicio. Por ejemplo, si la entidad k también ha detectado una alerta similar a la emitida por IDS_j, posiblemente del mismo escenario de ataque, este valor de suposición sería el mejor posible ya que la entidad k tendría una evidencia segura de que dicha alerta es realmente cierta. En caso contrario, si la entidad k no tiene ninguna evidencia sobre dicha alerta, esta suposición podría llegar a deducirse a partir de la publicación de otras alertas similares a la emitida por IDS_j, y que hayan sido generadas por otras entidades altamente confiables para k.

De esta manera, $Fancy_{kj}^{(t)}$ se puede calcular como se define en (4.5), según el número y valores de reputación, denotados como $Rep_\Omega(i \neq j)$, para todos aquellos IDSs i que i) hayan generado la alerta, agrupados en un conjunto G, y ii) los que no lo han hecho aun estando configurados para ello, agrupados en otro conjunto R; con $A = G \cup R$.

$$Fancy_{kj}^{(t)} = \frac{max\{\vartheta_G\} \cdot \overline{\vartheta_G} \cdot |\vartheta_G| - max\{\vartheta_R\} \cdot \overline{\vartheta_R} \cdot |\vartheta_R|}{\overline{\vartheta_A} \cdot |\vartheta_A|} \tag{4.5}$$

donde ϑ_S representa la lista de valores de reputación de todos los IDSs en el conjunto S (G, R o A) implicados en la detección de la alerta, es decir, $\vartheta = \{Rep_\Omega(i)\}\ \forall i \in S$; $max\{\vartheta_S\}$ la reputación más alta entre todos los IDSs del conjunto S; $\overline{\vartheta_S}$ la reputación promedio de los IDSs en S; y $|\vartheta_S|$ el número total de IDSs en S.

Nótese que se ha definido la máxima reputación de los IDSs dentro de un conjunto S dado, $max\{\vartheta_S\}$, con el objetivo final de limitar la confianza sobre la alerta generada al valor del IDS con la reputación más alta, ya que $Fancy_{kj}^{(t)}$ no puede exceder, en ningún caso, el valor de reputación del IDS más confiable.

En cualquier otro caso, e independientemente de si existen evidencias sobre la alerta emitida por IDS_j, la entidad k tendrá que incorporar esta nueva alerta a su motor de *Correlación & Agregación de Alertas*, descrito en la Sección 4.2.1, a fin de descubrir relaciones lógicas entre todas las alertas recibidas hasta entonces, incluyendo esta nueva. Este proceso de correlación puede revelar, por ejemplo, que la nueva alerta emitida por IDS_j era un evento esperado por el sistema de detección, ya que encaja en uno de los escenarios de ataques predefinidos por la entidad k.

4.3.2. Proceso de elección del Comité de Sabios

Como se ha comentado anteriormente, el Comité de Sabios (WC) de un dominio de seguridad en concreto se compone de varios de los NIDSs más reputables y confiables de su CIDN. Formalmente, en (4.6) se muestra el conjunto de NIDSs candidatos electos para convertirse en los distintos miembros del WC de un CIDN genérico Ω, enmarcado dentro de un dominio de seguridad en particular.

$$WCC_\Omega = \{NIDS_i \mid NIDS_i \in CIDN_\Omega,\ Rep_\Omega(NIDS_i) > \zeta_\Omega\} \tag{4.6}$$

donde ζ_Ω es un valor umbral, definido de antemano por el administrador del dominio de seguridad donde se encuentra desplegado $CIDN_\Omega$, que determina el nivel mínimo de reputación que necesita $NIDS_i$ para convertirse en un miembro de WC_Ω.

De forma adicional, el administrador de cada dominio de seguridad también puede decidir de forma individual el número mínimo y máximo de NIDSs que formarán parte del WC dentro de $CIDN_\Omega$; es decir, $|WC_\Omega| \in [min_\Omega, max_\Omega]$. De esta manera, todos los NIDSs candidatos constituirán automáticamente WC_Ω si $|WCC_\Omega| \in [min_\Omega, max_\Omega]$. Sin embargo, si $|WCC_\Omega| > max_\Omega$, la probabilidad de que un NIDS candidato se convierta en un miembro electo de su WC dependerá de su reputación global dentro del dominio de seguridad, como se puede ver en (4.7).

$$\mathbb{P}(WCC_{\Omega i} \in WC_\Omega) = Rep_\Omega(WCC_{\Omega i}),\ \forall NIDS_i \in WCC_\Omega \tag{4.7}$$

ζ_Ω se debe ir reduciendo constantemente, con pequeños decrementos, hasta alcanzar el número mínimo de candidatos hasta cumplir que $|WCC_\Omega| \leq max_\Omega$. Este caso suele producirse durante las fases iniciales del establecimiento y creación del CIDN, cuando existen muchos NIDSs candidatos para constituir WC_Ω.

En la Tabla 4.1 se muestra un extenso resumen de todos los pesos, variables y otros parámetros utilizados a lo largo de esta sección para el establecimiento del sistema de reputación intradominio. En esta tabla se incluye una breve descripción de cada uno de ellos, así como sus valores de inicialización y cálculo dependiendo de qué entidad es la encargada de su mantenimiento y configuración.

Variable	Descripción	Inicialización/cálculo
	Ecuación (4.1)	
$T_{WC_{ij}}$	Confianza de WC_i sobre IDS_j	Calculada en (4.1) y utilizada en (4.3)
α_i	Peso de las recomendaciones dadas por los miembros de WC_i	Administrador de WC_i
β_i	Peso de las opiniones de los $n-q$ NIDSs	Administrador de WC_i
γ_i	Peso de las opiniones de los m HIDSs	Administrador de WC_i
	Ecuación (4.2)	
$Rec_{kj}^{(t)}$	Recomendación de la entidad k en IDS_j en la transacción t	Calculada en (4.2) y utilizada en (4.1)
ω_k	Peso sobre la recomendación calculada en la interacción anterior	Definido por cada entidad k
	Ecuación (4.3)	
$Rep_\Omega(j)$	Reputación dada por $CIDN_\Omega$ a IDS_j	Calculada en (4.3)
$\sigma_{\Omega j}$	Desviación estándar al calcular el valor de reputación $Rep_\Omega(j)$	Utilizada como parte de (4.3)
$\sigma_{\Omega,umbral}$	Umbral máximo para considerar válido el cálculo de reputación $Rep_\Omega(j)$	Administrador de $CIDN_\Omega$
ε_i	Grado de pertenencia de $Rep_\Omega(j)$ para el nivel de confianza i	Calculado como se muestra en la Figura 4.4
	Ecuación (4.4)	
$Sat_{kj}^{(t)}$	Satisfacción de la entidad k sobre IDS_j en la transacción t	Calculada en (4.4) y utilizada en (4.2)
δ_j	Valor defuzzificado del conjunto difuso relacionado con la severidad de la alerta generada por IDS_j	Valor calculado defuzzificando el conjunto difuso (Figura 4.4)
	Ecuación (4.5)	
$Fancy_{kj}^{(t)}$	Veracidad para k sobre la alerta emitida por IDS_j en la transacción t	Calculada en (4.5) y utilizada en (4.4)
ϑ_S	Reputación de los IDSs dentro de S	Según capacidades de los IDSs
	Ecuación (4.6) y (4.7)	
WCC_Ω	NIDSs candidatos electos a WC_Ω	Calculado en (4.6)
ζ_Ω	Mínima reputación para convertir a un NIDS en miembro de WC_Ω	Administrador de $CIDN_\Omega$
min_Ω	Número mínimo de miembros de WC_Ω	Administrador de $CIDN_\Omega$
max_Ω	Número máximo de miembros de WC_Ω	Administrador de $CIDN_\Omega$

Tabla 4.1: Pesos y variables del sistema de reputación intradominio

4.4. Perfil de comunicaciones intradominio

Para una mejor comprensión del funcionamiento de la arquitectura presentada en la Sección 4.1, junto con la descripción de los distintos componentes y mecanismos de comunicación propuestos en la Sección 4.2, en esta sección se explica en detalle el perfil de comunicaciones en el comportamiento del sistema de detección a nivel intradominio. Es decir, en un único CIDN asociado a un dominio de seguridad en concreto.

Este perfil ofrece un mejor alcance y entendimiento de cómo los componentes de un CIDN genérico, denotado como $CIDN_\Omega$, interactúan entre sí para compartir y, como resultado, construir el conocimiento colectivo intradominio de alertas codiciado. Este conocimiento común permitirá afrontar el doble objetivo establecido como requisito al comienzo de este capítulo: i) detectar las amenazas distribuidas que pueden producirse localmente en la propia red y ii) identificar a aquellos IDSs que están manifestando un comportamiento malicioso en la red de detección y que, posiblemente, deseen publicar alertas fraudulentas. Con respecto a este último objetivo, esta sección también detalla cómo se ha integrado el sistema de reputación intradominio, presentado en la sección anterior, para la identificación de IDSs maliciosos en $CIDN_\Omega$.

En la Figura 4.5 se muestra un diagrama de secuencia UML con las interacciones intradominio que se producen entre todos los componentes de $CIDN_\Omega$. Nótese que el elemento $CIDN_\Omega$ que aparece en ese diagrama hace referencia a los m HIDSs y a los $n - q$ NIDSs que son parte de $CIDN_\Omega$ como red de detección, sin los q NIDSs que ya forman parte de WC_Ω (identificados en el actor contiguo). El resto de esta sección detalla cada una de esas interacciones, etiquetadas con un número para indicar posteriormente en la explicación a qué interacción se está haciendo referencia.

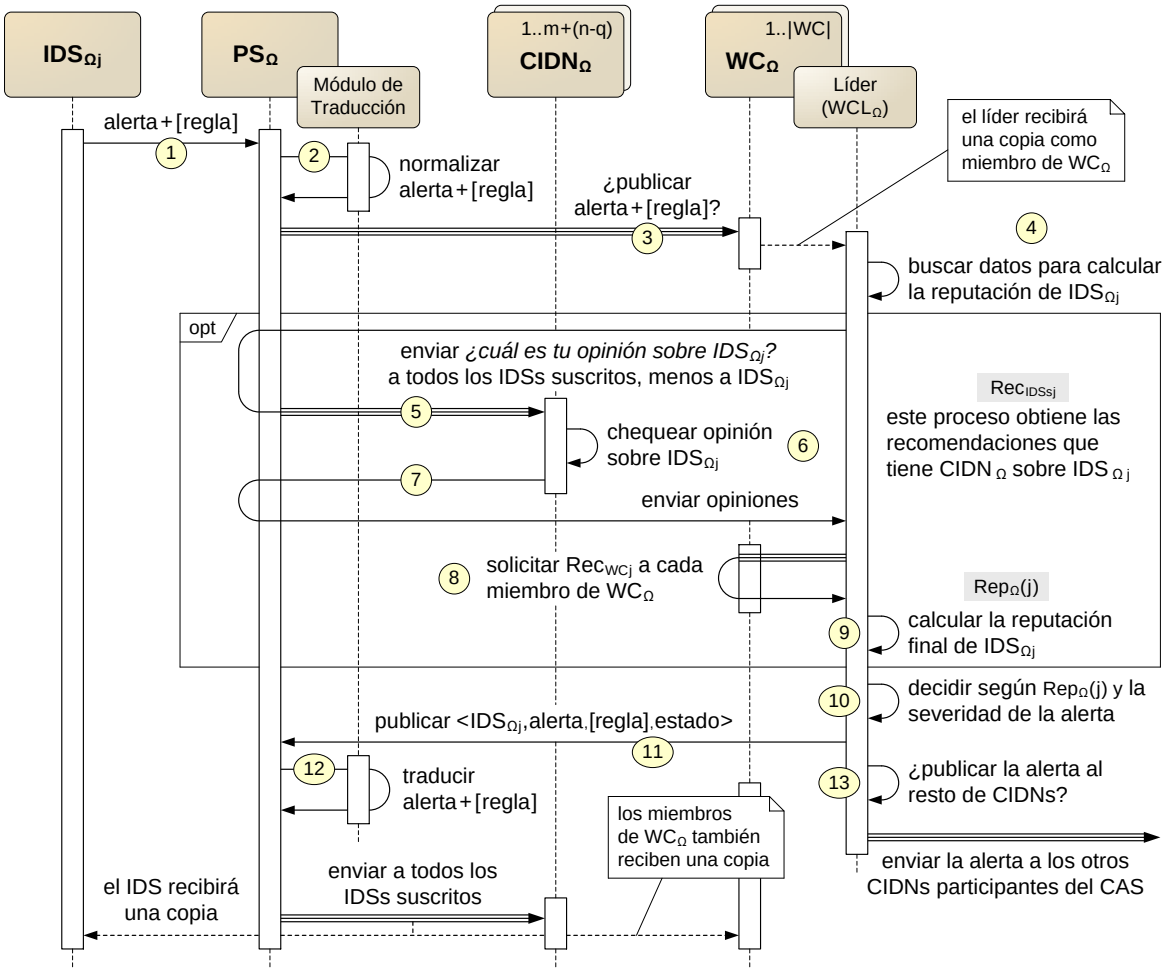
Paso 1: $IDS_{\Omega j}$ publica una nueva alerta

Se considera que un IDS de un dominio de seguridad genérico Ω, denotado $IDS_{\Omega j}$, publica una alerta a través de PS_Ω, proporcionado por el CIDN de su dominio: $CIDN_\Omega$. Nótese que la alerta puede ser publicada por un HIDS o por un NIDS, incluyendo dentro de esta última categoría cualquiera de los NIDS de WC_Ω. Si $IDS_{\Omega j}$ no estuviera suscrito en PS_Ω como miembro de $CIDN_\Omega$, la alerta sería automáticamente descartada sin que sea evaluada. Esta comprobación de pertenencia sobre $IDS_{\Omega j}$ la realiza el propio PS_Ω, ya que es el que gestiona la lista de entidades suscritas en su CIDN.

$IDS_{\Omega j}$ también puede adjuntar en esta publicación, junto con la alerta generada, la regla de detección que se disparó en su *Motor de Detección*. De esta manera, los IDSs de $CIDN_\Omega$ podrán compartir y crear un modelo de conocimiento común para ampliar, si fuera necesario, la cobertura de la detección que cada uno posee individualmente. Para esta publicación, $IDS_{\Omega j}$ envía un mensaje a PS_Ω dentro de su dominio de seguridad, donde se encuentra suscrito bajo los dos siguientes asuntos o tópicos:

- `ALERTA_GENERADA`: publicar exclusivamente la alerta.
- `ALERTA_REGLA_GENERADA`: publicar tanto la alerta como la regla de detección.

La distinción entre los dos asuntos anteriores hace que PS_Ω pueda distinguir entre ambos tipos de mensajes y sus posibles formatos internos.



IDS$_{\Omega j}$: Nodo *j*, perteneciente al CIDN del dominio de seguridad Ω, que desea publicar una nueva alerta
PS$_\Omega$: Servicio de Publicación del CIDN del dominio de seguridad Ω
CIDN$_\Omega$: Red colaborativa de detección de intrusiones con *m* HIDSs y (*n*-*q*) NIDSs, sin los *q* NIDSs de WC$_\Omega$
WC$_\Omega$: Comité de Sabios del CIDN del dominio de seguridad Ω
WCL$_\Omega$: Líder del Comité de Sabios del CIDN del dominio de seguridad Ω

Figura 4.5: Diagrama de secuencia UML para las comunicaciones intradominio

Para evitar que cualquier otro IDS dentro $CIDN_\Omega$ pueda modificar la información contenida en esta publicación, $IDS_{\Omega j}$ tiene la opción de enviar esta información a PS_Ω firmada digitalmente, o incluso cifrada para que, únicamente, PS_Ω y, posteriormente, WC_Ω tengan constancia acerca de la interacción. En este caso, la publicación se realiza usando tecnologías basadas en criptografía de clave pública mediante certificados X.509, como se ha definido en el capítulo anterior, con el objetivo de establecer canales seguros SSL/TLS entre $IDS_{\Omega j}$ y PS_Ω. Estos canales SSL/TLS se pueden establecer utilizando dos mecanismos de seguridad distintos, aunque complementarios:

- *Autenticación de servidor.* $IDS_{\Omega j}$ utiliza el certificado de PS_{Ω}, obtenido durante su proceso de suscripción en $CIDN_{\Omega}$, con el que cifrar la información contenida en cualquier publicación mediante la clave pública de PS_{Ω}.
- *Autenticación de cliente.* De forma adicional, $IDS_{\Omega j}$ también puede autenticarse frente a PS_{Ω} como una entidad legítima de $CIDN_{\Omega}$ con sus propias credenciales criptográficas, concretamente su clave privada, con la que, además, $IDS_{\Omega j}$ podría firmar digitalmente el contenido de cualquier publicación.

Todos los NIDSs están obligados a autenticarse frente a PS_{Ω} durante su suscripción (*registro*) para ser parte de $CIDN_{\Omega}$. En cambio, este requisito no es obligatorio para los HIDSs, ya que suelen ser de usuarios finales que se unen a $CIDN_{\Omega}$ voluntariamente, como se ha comentado en la Sección 4.1.1. Debido a ello, la publicación por parte de un HIDS puede llevarse a cabo sin utilizar ningún mecanismo de seguridad.

Paso 2: normalizar la alerta y, opcionalmente, la regla de detección

PS_{Ω} comprueba que $IDS_{\Omega j}$ está suscrito en alguno de los asuntos que le identifique como miembro de $CIDN_{\Omega}$. Si, además, se autenticó porque es un NIDS, o lo que desea publicar está firmado digitalmente, PS_{Ω} verifica la validez de su certificado delegando este proceso al Servicio de Validación de la Sección 3.2, y que el dominio administrativo de $CIDN_{\Omega}$ debe tener desplegado en su red interna de detección.

Si la comprobación de pertenencia de $IDS_{\Omega j}$ ha sido correcta, PS_{Ω} entonces reenvía la información recibida al *Módulo de Traducción*, para que la normalice a un formato común y estándar entendible por todos los IDSs de $CIDN_{\Omega}$ (proceso explicado en la Sección 4.2.2). La normalización se realiza sólo para la alerta, o también para la regla de detección, dependiendo del asunto utilizado por $IDS_{\Omega j}$ en el Paso 1.

La información normalizada obtenida desde el *Módulo de Traducción* se almacena internamente en PS_{Ω}, como un repositorio de históricos de interacciones pasadas. Esta información se almacena como tripletas en formato <$\texttt{IDS}_{\Omega j}$,`alerta`,`[regla]`> para que cualquier miembro de WC_{Ω} pueda recuperarla posteriormente. Esta información podría ser solicitada, por ejemplo, por un nuevo NIDS que acaba de unirse a WC_{Ω}, y que no tiene todavía estos datos para el cálculo de sus recomendaciones.

Paso 3: la tripleta <$\texttt{IDS}_{\Omega j}$,`alerta`,`[regla]`> se envía al WC_{Ω} para su evaluación

PS_{Ω} envía la tripleta normalizada a todos los miembros de WC_{Ω} con la información asociada a $IDS_{\Omega j}$ para el cálculo de su reputación, la alerta que intenta publicar y, de manera opcional, la regla de detección que disparó su *Motor de Detección*. Aunque esta información sólo la evalúa WCL_{Ω}, por motivos de rendimiento como se argumenta en la Sección 4.3.1, todos los miembros de WC_{Ω} también reciben una copia para que tengan constancia de la posible publicación. De esta manera, los NIDSs de WC_{Ω} serán capaces de detectar si WCL_{Ω} presenta síntomas maliciosos en su comportamiento, como, por ejemplo, el no tratamiento de esta publicación por parte de WCL_{Ω}.

Las comunicaciones entre PS_Ω y los miembros de WC_Ω se realizan de forma segura mediante conexiones SSL/TLS, usando para ello los certificados que PS_Ω almacena de cada uno. Además de ofrecer un mecanismo de seguridad, el uso de la criptografía de clave pública también posibilita un mecanismo de no repudio, tanto de origen como de destino, donde los IDSs no pueden negar la realización de estas comunicaciones.

Paso 4: ¿buscar datos para calcular $Rep_\Omega(j)$, reputación de $IDS_{\Omega j}$?

Como se ha comentado en la Sección 4.3.1, WCL_Ω comprueba si $Rep_\Omega(j)$ ha sido calculado recientemente. Si es así, el valor de reputación todavía se puede considerar como válido y no es necesario volver a realizar su cálculo. Esta comprobación permite al CIDN, y más concretamente a WCL_Ω, evitar peticiones de recomendaciones a todos los miembros de $CIDN_\Omega$, incluidos el resto de miembros de WC_Ω, y pasar directamente al Paso 10. De esta manera, se evita una sobrecarga de tráfico en la red de una información que todavía es válida, así como la computación en cada uno de los IDSs.

Paso 5: solicitar opiniones a los miembros de $CIDN_\Omega$ sobre $IDS_{\Omega j}$

Si WCL_Ω no guarda recomendaciones sobre $IDS_{\Omega j}$, o son demasiado antiguas para ser consideradas como válidas, pregunta a todos los miembros de $CIDN_\Omega$ sus opiniones sobre este IDS. Debido a ello, WCL_Ω envía un mensaje de *solicitud de opinión* a través de PS_Ω, con el asunto `SOLICITAR_OPINION` (ver Figura 4.2), el cual lo reenviará a todos los IDSs suscritos en $CIDN_\Omega$, a excepción de $IDS_{\Omega j}$. PS_Ω incluirá en este mensaje un identificador único, creado aleatoriamente, para que PS_Ω pueda identificar a posteriori a qué mensaje pertenece cada una de las respuestas que recibe.

De manera similar al Paso 1, las comunicaciones entre PS_Ω y cada uno de los IDSs de $CIDN_\Omega$ se realizan a través de conexiones SSL/TLS, si fuera posible, utilizando los certificados que PS_Ω mantiene de cada uno de ellos.

Paso 6: comprobar las recomendaciones sobre $IDS_{\Omega j}$

Cada miembro de $CIDN_\Omega$, después de validar la información criptográfica recibida de PS_Ω, recupera la recomendación que tiene guardada sobre $IDS_{\Omega j}$ y se la devuelve a WCL_Ω a través de PS_Ω, con el asunto `OPINION`, incluyendo el identificador del mensaje que obtuvo de PS_Ω. De nuevo, cada miembro tiene que establecer, si fuera posible, un canal seguro SSL/TLS con PS_Ω. Nótese que la información obtenida al final de este paso corresponde con las dos últimas operaciones de agregación de (4.1).

Paso 7: enviar las recomendaciones al WC_Ω

Después de que PS_Ω reciba todas las recomendaciones sobre $IDS_{\Omega j}$, y haya podido comprobar la validez digital de cada una con el certificado del emisor, son empaquetadas por PS_Ω en un único mensaje de respuesta, que es enviado de manera directa a WCL_Ω a través del canal SSL/TLS creado previamente entre ambos.

En este punto puede surgir la duda de hasta cuándo PS_Ω debe esperar hasta recibir todas las respuestas de cada miembro de $CIDN_\Omega$. Obviamente, podrían haber IDSs, principalmente HIDSs, que hayan abandonado $CIDN_\Omega$ sin darse de baja en PS_Ω, otros que por problemas de rendimiento podrían enviar su recomendación con mucho retraso o, incluso, IDSs que han sufrido algún tipo de ataque y les hayan intervenido su normal comportamiento. La alternativa adquirida en cada CIDN puede variar de uno a otro, aunque lo más habitual es utilizar una pequeña *ventana de tiempo* de, por ejemplo, dos o tres segundos, durante el que PS_Ω estará esperando recomendaciones.

Paso 8: solicitar al resto de NIDSs de WC_Ω sus opiniones sobre $IDS_{\Omega j}$

WCL_Ω solicita al resto de k miembros de WC_Ω sus recomendaciones sobre $IDS_{\Omega j}$, denotadas como $Rec_{WC_{kj}}$, usando una conexión segura SSL/TLS con cada uno de ellos. Como en el paso anterior, WCL_Ω tiene que definir cuánto tiempo debe esperar hasta recibir todas las recomendaciones, y si alguno de ellos no responde después de varios intentos fallidos para obtener su recomendación, WCL_Ω arranca el proceso descrito en la Sección 4.3.2 para que escoger uno nuevo. Nótese que las recomendaciones que WCL_Ω recibe corresponden con la primera operación de agregación de (4.1).

En este punto, cabe destacar que, por motivos de rendimiento, este paso se puede ejecutar de forma paralela a los pasos 5, 6 y 7 vistos anteriormente.

Paso 9: calcular la reputación final de $IDS_{\Omega j}$

Con todas las recomendaciones recibidas, WCL_Ω comprueba que no existe ninguna discrepancia entre los miembros de WC_Ω a la hora de evaluar un mismo IDS. Como se ha comentado en la Sección 4.3.1, existe un acuerdo cuando la desviación estándar de todas las respuestas, denotada como $\sigma_{\Omega j}$, es menor de un cierto umbral preestablecido, denotado como $\sigma_{\Omega,umbral}$. Si se confirma que $\sigma_{\Omega j} \leq \sigma_{\Omega,umbral}$, indicando que todos los miembros de WC_Ω han alcanzado un alto acuerdo, WCL_Ω calcula el valor de confianza que cada miembro i de WC_Ω tiene sobre $IDS_{\Omega j}$; es decir, calcula $T_{WC_{ij}}$ según (4.1). Si en cambio no se ha alcanzado un acuerdo, $\sigma_{\Omega j} > \sigma_{\Omega,umbral}$, este hecho se le notifica al administrador del dominio de seguridad, WC_Ω se disuelve y se elige uno nuevo.

Finalmente, WCL_Ω calcula la reputación global que tiene $IDS_{\Omega j}$ dentro de $CIDN_\Omega$, denotada como $Rep_\Omega(j)$, de acuerdo a (4.3).

Paso 10: decidir según la reputación de $IDS_{\Omega j}$ y la severidad de la alerta

WCL_Ω determina el nivel de confianza que tiene $IDS_{\Omega j}$ según su valor de reputación calculado en el paso anterior, como se ha definido en la Sección 4.3.1. Dependiendo de ese nivel de confianza, y también del valor defuzzificado del conjunto difuso asociado con la severidad de la alerta, WCL_Ω verifica si $IDS_{\Omega j}$ tiene la confianza necesaria para considerar su alerta como información no fraudulenta. Según esta decisión, el resto de IDSs van a recibir o no esta alerta, para poder incorporarla a sus procesos internos de detección como una alerta que representa hechos ocurridos en la realidad.

Paso 11: publicar la cuádrupla <IDS$_{\Omega j}$,alerta,[regla],estado>

WCL_{Ω} hace disponible a los miembros de $CIDN_{\Omega}$ la nueva alerta detectada, cuyo emisor $IDS_{\Omega j}$ ha demostrado una confianza suficiente para que sea considerado como una entidad benevolente. La cuádrupla <IDS$_{\Omega j}$,alerta,[regla],estado> la publica WCL_{Ω} a través de PS_{Ω} utilizando los asuntos `NUEVA_ALERTA` o `NUEVA_ALERTA_REGLA`, dependiendo de si la publicación solamente contiene la alerta o también incluye la regla de detección que disparó el *Motor de Detección* de $IDS_{\Omega j}$. El elemento `estado` de la cuádrupla corresponde con la decisión final que ha tomado WCL_{Ω}, en nombre de todos los miembros de WC_{Ω}, indicando si la alerta puede ser tomada o no como válida.

Paso 12: traducir la alerta y, opcionalmente, la regla de detección y enviar la cuádrupla a todos los miembros de $CIDN_{\Omega}$

Con la cuádrupla anterior enviada por WCL_{Ω}, PS_{Ω} modifica su base de datos para actualizar el estado de validación de la alerta con la decisión final tomada por WC_{Ω}. Antes de que la publicación se haga efectiva, PS_{Ω} debe enviar la cuádrupla al *Módulo de Traducción* para que la traduzca según el software de detección que tengan instalados los IDSs de $CIDN_{\Omega}$, y que estén suscritos en el asunto correspondiente para recibir esa cuádrupla traducida. (El software de detección lo tienen que haber proporcionado los IDSs durante su proceso de suscripción en $CIDN_{\Omega}$ a través de PS_{Ω}.)

Finalmente, PS_{Ω} entregará una copia traducida de la cuádrupla a cada uno de los IDSs suscritos en los asuntos `NUEVA_ALERTA` o `NUEVA_ALERTA_REGLA`. Al igual que en alguno de los pasos anteriores, PS_{Ω} envía la cuádrupla bajo canales seguros SSL/TLS, haciendo uso de los certificados que PS_{Ω} almacena de cada uno de los IDSs.

Paso 13: tomar una decisión si publicar o no la alerta al resto de CIDNs

Como último paso, WCL_{Ω} tiene que decidir si la alerta debe ser enviada al resto de CIDNs con los que existe una relación de confianza interdominio. La estrategia de cuándo propagar una alerta se basa en compartir sólo las que tienen una severidad *High*, al suponer un alto riesgo de impacto sobre el CAS. Como se define en la Sección 4.3.1, WCL_{Ω} tiene que determinar el nivel de confianza de $IDS_{\Omega j}$ según $Rep_{\Omega}(j)$, haciendo que la publicación interdominio sólo se lleve a cabo si el nivel de confianza de $IDS_{\Omega j}$ queda emplazado en el nivel *High*. Otra alternativa es hacer que WCL_{Ω} base su decisión según el número de alertas del mismo tipo que recibe en una pequeña *ventana de tiempo*. Esto podría significar que un atacante está ejecutando un ataque distribuido. Muchos trabajos de investigación han utilizado este concepto de ventana de tiempo tomando, por ejemplo, todas las alertas agregadas durante los últimos dos segundos [7].

Por último, es destacable comentar que las alertas que deben compartirse entre los CIDNs, en aras de poder detectar ataques distribuidos, no deberían revelar información crítica de la infraestructura interna de cada CIDN. Y, mucho menos, si esas alertas se van a intercambiar con CIDNs pertenecientes a otros dominios administrativos, donde la privacidad sobre dicha información tendría que ser todavía mayor.

4.5. Resultados experimentales

En esta sección se presenta la evaluación de la arquitectura basada en reputación, presentada a lo largo de este capítulo, a través de varios experimentos. Esta evaluación se ha realizado a través de una serie de resultados experimentales, obtenidos sobre un entorno intradominio de simulación, donde el objetivo principal es comprobar cómo se comporta la arquitectura cuando crece el número de IDSs maliciosos.

En este entorno de simulación se ha desplegado y configurado un CIDN genérico Ω, denotado como $CIDN_{\Omega}$, compuesto por 300 HIDSs y 200 NIDSs. Más en concreto, un ratio del 60/40 sobre el total de IDSs. Dentro del proceso de elección de WC_{Ω}, se ha establecido un rango entre 10 y 20 NIDSs para que puedan ser parte de WC_{Ω}; es decir, $|WC_{\Omega}| \in [10, 20]$. También se ha establecido un valor umbral de pertenencia de $0,8$ ($\xi_{\Omega} = 0,8$), el cual establece la reputación mínima que necesita cualquier NIDS para convertirse en miembro de WC_{Ω}. La implementación de PS_{Ω} se ha realizado adaptando la solución *Open WS-Eventing* [241], para el intercambio seguro de los mensajes entre los NIDSs de WC_{Ω} y el resto de IDSs de $CIDN_{\Omega}$. Este servicio es una implementación de código abierto, desarrollado por la Universidad de Murcia, según la recomendación estándar del W3C *Web Services Eventing* (WS-Eventing) [242].

Para la ejecución de estas pruebas, se considera que las alertas ya han sido generadas por los IDSs. Es decir, no se ha hecho uso de ninguna herramienta o método de detección en particular, debido a que la arquitectura ha sido implementada para su ejecución en un escenario de laboratorio bajo una gran cantidad de IDSs, con un total de 500 IDSs. En su lugar, se han inyectado de manera aleatoria 9000 alertas de acuerdo a las cuatro categorías de niveles de severidad definidas en el estándar IDMEF: *Info*, *Low*, *Medium* y *High*. Nótese que en esta generación aleatoria de alertas, se han simulado el doble de alertas del tipo *Low* y *Medium*, ya que en un entorno real la distribución de alertas no es totalmente equitativa en estas cuatro categorías. Las alertas de estos dos tipos son más habituales que, por ejemplo, las alertas del tipo *High*.

Sobre la proporción de IDSs maliciosos desplegados en $CIDN_{\Omega}$, se han hecho cuatro pruebas en cada uno de los experimentos variando el número de IDSs maliciosos entre el 20, 40, 60 y el 80 %. Estos valores se han escogido como porcentajes representativos para comprobar el comportamiento del sistema propuesto conforme aumenta el número de IDSs maliciosos, con los que obtener algunos indicios sobre:

- Cómo el modelo de reputación intradominio, presentado en la Sección 4.3, puede ayudar a que se alcancen mejores ratios en la detección mediante la identificación de las alertas fraudulentas generadas por IDSs maliciosos, haciendo que no sean incorporadas a los procesos internos de detección del resto de IDSs del CIDN.

- Cómo los pesos en las recomendaciones de los distintos componentes de $CIDN_{\Omega}$ –miembros de WC_{Ω} (α), NIDSs (β) y HIDSs (γ)– pueden afectar al cálculo de la reputación de un IDS del dominio de seguridad, según (4.1).

Todos los resultados obtenidos de estos experimentos son analizados y discutidos en profundidad en los siguientes apartados.

4.5.1. Cobertura de la detección de un CIDN

El objetivo principal de este primer experimento es poder comprobar qué ratio de detección puede alcanzar el modelo de reputación intradominio propuesto. La mejora de este ratio de detección se conseguirá mediante la identificación de las falsas alertas (fraudulentas) producidas por aquellos IDSs que son maliciosos en $CIDN_{\Omega}$. Las alertas que procedan de esos IDSs no serán tomadas en cuenta por el resto de miembros de $CIDN_{\Omega}$, ampliando así su ratio de detección cuando solamente las alertas procedentes de IDSs benevolentes sean tomadas en consideración.

En este experimento, se han fijado los pesos en las recomendaciones que tiene cada miembro de $CIDN_{\Omega}$ con los siguientes porcentajes: $\alpha = 0,4$ para los NIDSs de WC_{Ω}, $\beta = 0,35$ para las recomendaciones recibidas de los NIDSs y $\gamma = 0,25$ para las recibidas de los HIDSs. Recordar que todos estos pesos se han utilizado en (4.1) para el cálculo de la reputación que cada uno de esos IDSs deposita en otra entidad en particular. En este caso, el IDS que publica la alerta que está siendo evaluada.

La Figura 4.6 muestra gráficamente los resultados de las cuatro pruebas realizadas en este experimento, una por cada porcentaje de los IDSs maliciosos (HIDSs o NIDSs) desplegados en $CIDN_{\Omega}$, y teniendo en cuenta la severidad de las alertas emitidas.

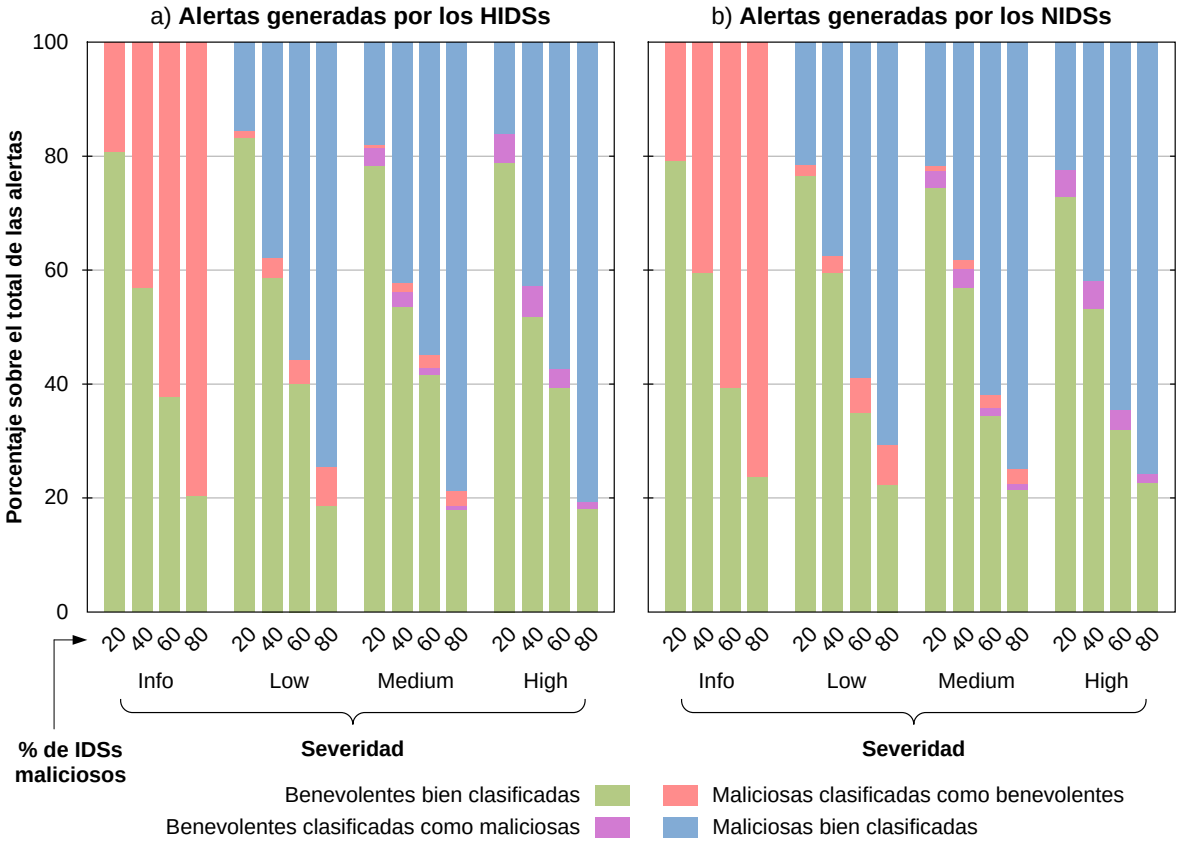


Figura 4.6: Alertas bien o mal clasificadas, generadas por un determinado porcentaje malicioso de a) HIDSs o b) NIDSs y los cuatro niveles de severidad

En los dos gráficos, se presentan las alertas que han sido generadas y publicadas por todos los IDSs, ya sean benevolentes o maliciosos, de acuerdo a las cuatro categorías de severidad de las alertas. También se incluyen las alertas que no han sido publicadas, porque WC_{Ω} las ha considerado como fraudulentas (maliciosas) y, por tanto, se tienen que eliminar para no generar confusión en el resto de miembros de $CIDN_{\Omega}$. En estos dos gráficos de columnas, el eje de abscisas representa las cuatro categorías de severidad de las alertas conforme el porcentaje de IDSs maliciosos va en aumento, desde el 20 al 80 %. Por otro lado, el eje de ordenadas representa el total en porcentaje de las alertas publicadas, o no publicadas, por los IDSs benevolentes o maliciosos.

Analizando las pruebas realizadas, cuyos resultados se pueden ver en la Figura 4.6, se puede observar que todas las alertas con severidad *Info* se publican en todos los casos. Éstas son alertas que no representa información crítica para el correcto funcionamiento del sistema, sino que exponen información extra sobre la actividad del sistema que los IDSs podrían hacer uso en futuras detecciones. La credibilidad en este tipo de alertas se deja a la elección de cada IDS, para que decida si utilizarlas o no durante sus propósitos futuros de detección. Debido a ello, las alertas con severidad *Info* nunca son analizadas por WC_{Ω} y son siempre publicadas de forma automática.

Merece la pena subrayar, como ya se ha comentado anteriormente, que las alertas se han generado de manera aleatoria, por lo que este número nunca coincidirá exactamente con el mismo número de IDSs maliciosos que las han generado. Por ejemplo, la primera barra en la Figura 4.6a muestra un ratio de publicación del 80,75 %, cuando realmente hay un 80 % de HIDSs benevolentes. Esta diferencia es incluso más alta en la segunda barra, donde la publicación de alertas por parte de HIDSs benevolentes alcanza el 57 %, cuando realmente hay un 60 % de IDSs con este comportamiento.

Con respecto al resto de severidades de las alertas (*Low*, *Medium* y *High*) se puede comprobar cómo su publicación por parte de IDSs maliciosos es incrementalmente más pequeña conforme aumenta la severidad, alcanzando un 0 % cuando es *High*. En este caso, solamente el 8,85 % de las alertas publicadas por HIDSs maliciosos con severidad *Low*, el 1,5 % del total de las alertas, no han sido bien clasificadas y han sido finalmente publicadas como alertas benévolas. Este porcentaje se incrementa ligeramente hasta el 2 % sobre el total de las alertas cuando son publicadas por los NIDSs, siendo el 8,63 % de las alertas publicadas por NIDSs maliciosos. En el peor de los casos, cuando existe un 80 % de IDSs maliciosos, el porcentaje de las alertas maliciosas crece hasta el 8,83 % de promedio (7 % del total de alertas) para ambos tipos de IDSs.

En vista de los resultados anteriores, se puede afirmar como resultado que el modelo de reputación intradominio diseñado ofrece un ratio de detección de alertas maliciosas con severidad *Low* del 96 %, cuando el porcentaje de IDSs maliciosos es menor del 50 %, mientras que este ratio decrece ligeramente hasta el 93 % cuando el porcentaje de estos IDSs maliciosos alcanza el 80 %. Para las alertas con severidad *Medium*, la publicación desde HIDSs maliciosos se ve reducida desde el anterior 8,85 % al 2,96 %, mientras que el decremento para los NIDSs maliciosos se mueve desde el anterior 8,83 % al 4,5 %. En el peor de los casos, cuando existe un 80 % de IDSs maliciosos, el porcentaje disminuye en promedio desde el 8,83 % al 3,41 % para los dos tipos de IDSs. En este caso, como

regla general, el ratio de detección del modelo de reputación propuesto es ahora incluso mejor que en el caso anterior, para las alertas con severidad *Low*, llegando a alcanzar un ratio máximo sobre el 99,5 % cuando existe un 20 % de IDSs maliciosos en el sistema. En el peor caso, con un 80 % de IDSs maliciosos, este ratio de detección todavía ofrece resultados bastante prometedores, rondando el 97,24 % de éxito.

En la Tabla 4.2 se muestran algunas de las pruebas presentadas en la Figura 4.6, pero en esta ocasión de forma numérica. Para cada una de las cuatro posibles categorías de niveles de severidad, la Tabla 4.2 indica el número total de alertas generadas en cada caso y las alertas que son realmente aceptadas y publicadas por WC_Ω. La diferencia numérica entre las alertas generadas y las válidas son aquellas que WC_Ω ha clasificado como no válidas, al ser consideradas como maliciosas (fraudulentas).

		HIDSs				NIDSs			
		Benevolente		Malicioso		Benevolente		Malicioso	
Info	**Generadas**	705	183	168	714	486	147	127	473
	Válidas	705	183	168	714	486	147	127	473
Low	**Generadas**	1497	344	305	1514	909	266	278	920
	Válidas	1497	344	27	131	909	266	24	83
Medium	**Generadas**	1487	330	337	1445	912	271	266	930
	Válidas	1427	315	10	49	878	258	12	32
High	**Generadas**	757	169	144	701	483	144	139	449
	Válidas	710	156	0	0	452	134	0	0
		80%	20%	20%	80%	80%	20%	20%	80%

Tabla 4.2: Alertas generadas y clasificadas como válidas según su severidad

Cada tipo de IDS en la Tabla 4.2, ya sea un HIDS o un NIDS, se ha dividido en dos columnas según el porcentaje de IDSs benévolos y maliciosos desplegados en $CIDN_\Omega$. Para ello, la Tabla 4.2 representa dos de las cuatro pruebas realizadas, y que también han sido mostradas, pero gráficamente, en la Figura 4.6: la primera columna en ambos tipos de IDSs representa un 20 % de IDSs maliciosos, 80 % de benévolos, mientras que la segunda columna supone un 80 % de IDSs maliciosos, 20 % de benévolos.

La Tabla 4.2 permite extraer una serie de conclusiones con respecto a la severidad de las alertas y el comportamiento malicioso de sus emisores. Ahí se puede observar que se generan 305 alertas con severidad *Low*, cuando existe un 20 % de HIDSs maliciosos, clasificando de manera errónea 27 de ellas como alertas válidas. Sin embargo, se puede constatar una ligera mejoría cuando los IDSs maliciosos generan alertas con severidad *Medium*, donde WC_Ω sólo yerra en 10 de las 337 alertas generadas. Con respecto a las alertas con severidad *High*, se puede afirmar que el modelo de reputación intradominio predice con éxito todos los casos, no publicando alertas de IDSs maliciosos.

De forma similar al análisis anterior, pero ahora poniendo el foco de atención sobre los IDSs con un comportamiento benévolo, se puede observar que la no publicación de alertas válidas, si han sido generadas por IDSs benevolentes, es cada vez mayor según vaya aumentando la severidad de las distintas alertas.

Todas las alertas con severidad *Low* son publicadas al clasificarse correctamente. El primer inconveniente comienza con las alertas de severidad *Medium*, donde un 4 % de las válidas no se publican para los HIDSs y el 3,72 % para los NIDSs, cuando existe un 20 % de IDSs maliciosos. En el peor de los casos, con un 80 % de los IDSs maliciosos, se alcanza el 4,67 % de promedio para ambos tipos de IDSs. Con respecto a las alertas con severidad *High*, comentar que la clasificación es bastante similar al análisis anterior.

Como principal conclusión, después de analizar este primer experimento, se puede afirmar que el modelo de confianza intradominio basado en reputación propuesto en este capítulo es suficientemente robusto para este tipo de entornos de simulación, alcanzando un ratio para la detección de IDSs maliciosos superiores al 93 % en el peor de los casos, cuando el porcentaje de IDSs maliciosos alcanza valores hasta del 80 %.

4.5.2. Pesos en las recomendaciones sobre un IDS

El interés principal detrás de este segundo experimento es analizar y determinar los valores más apropiados a la hora de calcular la reputación de un IDS en particular. Es decir, cómo los pesos en las recomendaciones utilizados en (4.1) pueden afectar a dicho cálculo, los cuales corresponden a las recomendaciones de los miembros de WC_Ω (α), las obtenidas desde los NIDSs (β) y las recibidas desde los HIDSs (γ).

Para las simulaciones de este experimento, se han utilizado los mismos parámetros que en el experimento anterior: los dos tipos de IDSs utilizados (HIDSs o NIDSs), las cuatro categorías de niveles de severidad (*Info*, *Low*, *Medium* y *High*) y los porcentajes de IDSs maliciosos desplegados en $CIDN_\Omega$ (20, 40, 60 y 80 %). La diferencia con este nuevo experimento es variar los pesos en las recomendaciones en lugar de utilizar valores fijos como se hizo en el primer experimento. La única restricción, como se define en (4.1), es que los diferentes pesos deben cumplir que $\alpha + \beta + \gamma = 1$ y que $\alpha \geq \beta \geq \gamma$. Por tanto, se deben alterar los valores de esos pesos siempre y cuando se cumplan esas dos restricciones. Para ello se han llevado a cabo 50 pruebas, comenzando con $\alpha = \frac{1}{3}$, $\beta = \frac{1}{3}$ y $\gamma = \frac{1}{3}$, e incrementando $\frac{1-1/3}{50}$ al peso de α en cada una de las pruebas hasta que se alcance el máximo valor permitido, con α=1. Por su lado, β y γ se irán decrementando desde sus valores iniciales, aunque dándole mayor importancia a las recomendaciones dadas por los NIDSs que las proporcionadas por los HIDSs.

Entre los 32 resultados obtenidos se han seleccionado los dos más ilustrativos, por claridad, para determinar los mejores valores para los pesos de esas recomendaciones. Estos dos resultados se muestran en la Figura 4.7, con un 20 % de entidades maliciosas, y en la Figura 4.8 donde ese porcentaje asciende al 40 %. En ambos casos, los emisores son HIDSs que intentan publicar alertas con severidad *High*.

En el eje de ordenadas de ambos gráficos, se representa el total en porcentaje de las alertas publicadas, o no publicadas, por los HIDSs benevolentes o maliciosos. Por otro lado, el eje de abscisas representa los distintos pesos sobre las recomendaciones que se quieren evaluar en este experimento, desde $\alpha = \frac{1}{3}$, $\beta = \frac{1}{3}$ y $\gamma = \frac{1}{3}$ hasta $\alpha = 1$, $\beta = 0$ y $\gamma = 0$, con el incremento/decremento comentado anteriormente.

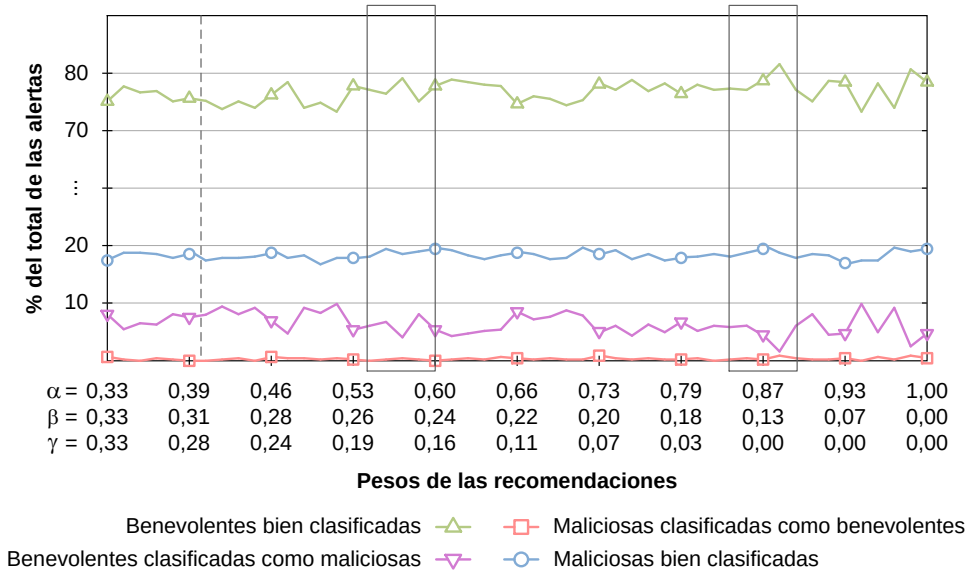


Figura 4.7: Alertas bien o mal clasificadas según los pesos en las recomendaciones, con un 20 % de HIDSs maliciosos

Como ejemplo puntual, comentar que la línea vertical discontinua hace referencia a los resultados obtenidos en el primer experimento, cuando se utilizaron pesos fijos en las recomendaciones: $\alpha = 0,4$, $\beta = 0,35$ y $\gamma = 0,25$. Esos resultados corresponden con los porcentajes de alertas mostrados en la Figura 4.6a, con un nivel de severidad *High* y un 20 % (Figura 4.7) y un 40 % (Figura 4.8) de HIDSs maliciosos.

Con respecto a las alertas publicadas por los HIDSs benevolentes, el objetivo detrás de este experimento es encontrar los puntos óptimos que maximicen el porcentaje de alertas publicadas y, por el contrario, minimicen en todo lo posible las alertas que no son publicadas por esos IDSs benévolos. Como resultado, tanto en la Figura 4.7 como en la Figura 4.8, se han resaltado mediante dos rectángulos verticales las dos zonas con los resultados más prometedores donde se optimizan los pesos sobre las recomendaciones: $<\alpha = 0,59, \beta = 0,25, \gamma = 0,16>$ y $<\alpha = 0,88, \beta = 0,12, \gamma = 0>$. Aunque los resultados puntuales podrían parecer los más prometedores para el último de los dos conjuntos de pesos anteriores, también hay que considerar aquellos datos obtenidos alrededor de esos pesos. Como se puede comprobar en ambos gráficos, los datos alrededor del último conjunto muestran resultados más dispares que los obtenidos con el primer conjunto. Debido a ello, se puede afirmar que el conjunto de pesos $<\alpha = 0,59, \beta = 0,25, \gamma = 0,16>$ revela mejores resultados si se consideran ambas pruebas de forma conjunta.

Como conclusión según este segundo experimento, se puede confirmar que los valores más apropiados para los pesos en las recomendaciones a la hora de calcular la reputación de un IDS, utilizados en (4.1), es el conjunto $<\alpha = 0,59, \beta = 0,25, \gamma = 0,16>$. Además, también se puede comprobar que con estos pesos se obtienen mejores resultados que los obtenidos en el primer experimento, con una mejoría sobre el 5 %.

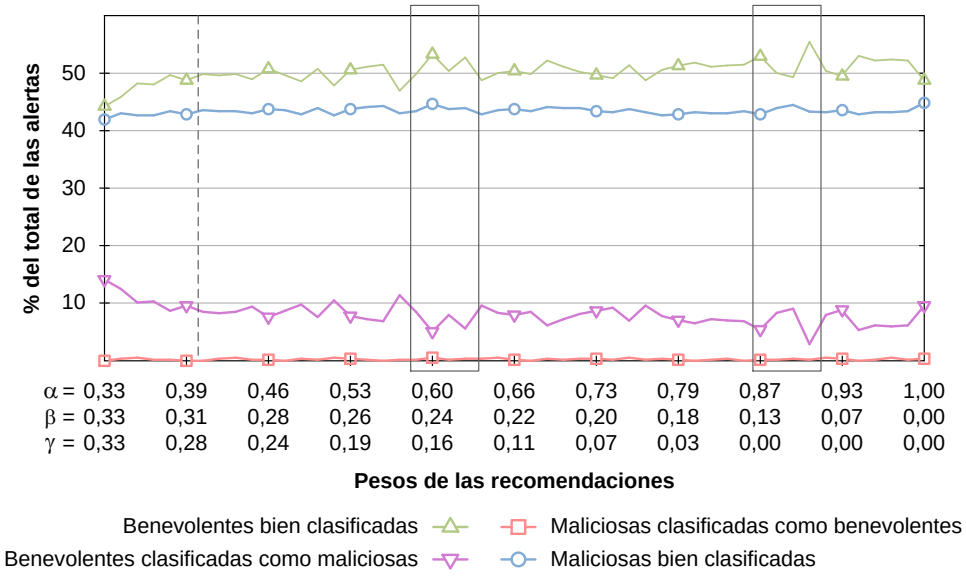


Figura 4.8: Alertas bien o mal clasificadas según los pesos en las recomendaciones, con un 40 % de HIDSs maliciosos

Los resultados que se acaban de comentar hacen pensar que la elección de los pesos sobre las recomendaciones no ejerce una gran influencia para el modelo de reputación intradominio, presentado en la Sección 4.3, habiendo una variación máxima cercana al 7,05 % en la clasificación de las alertas con un 40 % de HIDSs maliciosos.

4.6. Conclusiones del capítulo

Los pautas actuales de actuación de los atacantes han conducido a los mecanismos de detección a que deban adoptar nuevas estructuras de despliegue de sus componentes para la detección de nuevos tipos de intrusiones o ataques, especialmente los llevados a cabo en entornos distribuidos. En este contexto, este capítulo ha presentado el diseño de un sistema colaborativo de alertas (CAS), siguiendo una distribución estratégica de los IDSs con un esquema parcialmente descentralizado, capaz de construir un conocimiento colectivo de alertas con las que adquirir una visión global y holística de los parámetros de seguridad vinculados a la protección de los activos que se desean custodiar.

Con el propósito final de maximizar tanto la cobertura de la detección de cada red de monitorización como la escalabilidad en el análisis de grandes cantidades de información de detección en tiempo real, el CAS se ha desacoplado en múltiples sistemas autónomos, designados como redes colaborativas de detección de intrusiones (CIDN). Gracias a ese desacoplamiento en las funciones de un CAS, el sistema propuesto permite la definición de i) un modelo intradominio, con el conocimiento local de lo que está ocurriendo dentro de un CIDN, y ii) un modelo interdominio con un conocimiento global de la seguridad del sistema, construido entre todos los CIDNs participantes del CAS.

La construcción de una base de conocimiento colectivo a nivel del CAS se alcanza a partir del conocimiento local erigido en cada CIDN, con las alertas que cada uno de sus IDSs genera de manera autónoma en su acotado ámbito de detección, y que comparte con el resto de los IDSs de su CIDN para detectar ataques en su dominio de acción. Sin embargo, la aceptación de cualquier alerta podrían acarrear consecuencias significativas si alguno de los IDSs comparte alertas fraudulentas (falsos positivos) con el resto de su mancomunidad, pudiendo provocar errores en los procesos de detección.

Como solución frente a la publicación de alertas fraudulentas, a fin de descartarlas de los procesos de detección, en este capítulo se ha propuesto el diseño de un mecanismo con el que reforzar esos procesos de detección a nivel intradominio –internamente para un CIDN– mediante la definición de un modelo de confianza basado en reputación. Este modelo permite identificar comportamientos maliciosos de cualquiera de sus IDSs, antes de que intenten publicar alertas que no evidencien hechos que hayan ocurrido realmente. Cada CIDN delega el proceso de evaluación a una serie de NIDSs seleccionados como un grupo de expertos, o Comité de Sabios (WC), al ser las entidades que, en ese momento, tienen la mayor reputación posible de entre todas las disponibles.

Para el cálculo del valor de reputación de un IDS, en este capítulo se ha propuesto hacerlo mediante las recomendaciones que todo el CIDN tiene sobre ese IDS, según el comportamiento que ha manifestado en el pasado publicando sus alertas. La solicitud de estas recomendaciones, para el cálculo final de la reputación del IDS emisor de una alerta, la realiza el NIDS más confiable dentro del WC, identificado como tal al tener la reputación más alta de entre todos los que forman el WC. Este NIDS, como líder del WC, decide si la alerta se puede publicar o no, dependiendo de si el IDS tiene para ello la suficiente confianza –valor de reputación. Esta decisión se toma mediante un modelo basado en conjuntos difusos, que define una correspondencia entre la confianza del IDS y el impacto de la alerta sobre el activo del CIDN que se está protegiendo.

Las pruebas experimentales que se han realizado en este capítulo, sobre un entorno de simulación con un único CIDN, demuestran que el modelo basado en reputación que se ha propuesto para la gestión de la confianza en un escenario intradominio es robusto y preciso en la detección de alertas fraudulentas generadas por IDSs maliciosos, llegando a alcanzar, en promedio, la identificación del 95 % de esos falsos positivos.

El diseño del sistema de confianza basado en reputación propuesto en este capítulo tiene un ámbito local de aplicación, a nivel intradominio dentro de un CIDN, mientras que el siguiente capítulo se centra en el modelo de confianza a nivel interdominio para la construcción de la base global de conocimiento de alertas dentro del CAS. Este nuevo modelo se pondrá en marcha cuando el líder de un WC, como máximo estandarte de un CIDN, envía una alerta a sus homólogos, con los que tiene una relación de confianza, haciéndoles notar la posibilidad de un ataque a nivel distribuido.

Por último, reseñar que los intercambios de información (alertas y recomendaciones) entre los actores de un CIDN se realizan a través de canales seguros, haciendo uso de las soluciones basadas en criptografía de clave pública del Capítulo 3. Esta securización se utiliza también en el siguiente capítulo, donde el intercambio seguro entre CIDNs del CAS, a nivel multidominio, todavía supone mayor impacto sobre la seguridad.

Capítulo 5

Confianza multidominio en un sistema colaborativo de alertas

La construcción a nivel interdominio de una gran base global de conocimiento por parte del *Sistema Colaborativo de Alertas* (del inglés Collaborative Alert System, CAS), también se haya supeditada a que el intercambio de las alertas detectadas en cada *Red Colaborativa de Detección de Intrusiones* (del inglés Collaborative Intrusion Detection Network, CIDN) no represente información fraudulenta. El envío de alertas desde un CIDN con un comportamiento poco acreditado, según su histórico de las publicaciones, puede conducir a un grave compromiso del CAS que, como respuesta, podría emprender ciertos mecanismos de respuesta que alterasen el correcto estado de seguridad en el que se encontraba. Como objetivo, se plantea desarrollar un sistema interdominio de gestión de la confianza con el que modelar el comportamiento (*bondad*) de los CIDNs, a fin de identificar a nivel del CAS las alertas fraudulentas generadas por CIDNs con actitudes (presuntamente) maliciosas en su comportamiento. Con este sistema, solamente se va a distribuir en el CAS información verídica sobre el estado real de su seguridad.

Dentro de este tipo de escenarios distribuidos, se pretenden adoptar las alertas de detección que cualquier usuario final pudiera facilitar con sus dispositivos móviles, ya sean éstos teléfonos inteligentes u ordenadores portátiles. La adquisición de estas alertas puede proporcionar una mejor cobertura de la detección de los ataques, posibilitando la monitorización de ciertas áreas de la red de detección sobre las que el sistema tiene una elevada incertidumbre sobre la veracidad de las alertas que recibe. Por ejemplo, de IDSs que se encuentran ahí desplegados con una reputación demasiado baja para que se consideren sus alertas como verdaderas. La adopción de alertas de los *usuarios móviles* hace tener que dar respuesta a qué valor de confianza inicial se les tiene que conceder para que el sistema pueda comenzar a modelar sus comportamientos. Problemas bien conocidos en la literatura como *cold-start* y *bootstrapping*. Ambos problemas también se pueden extrapolar tanto a los IDSs de la infraestructura como a los CIDNs de los dominios de seguridad, aunque en entornos altamente dinámicos, como puede llegar a ser un CAS, estos dos tipos de entidades no se van a enfrentar a estos problemas con tanta asiduidad como sí lo harían los usuarios móviles.

La metodología de la que se va a hacer uso en este capítulo consiste en, primero, presentar el diseño de un mecanismo de *confianza interdominio* basado en reputación con el que detectar, y por ende eliminar, alertas fraudulentas generadas por CIDNs con un mal comportamiento. Véase que este diseño toma como base el sistema de reputación intradominio presentado en el Capítulo 4. Posteriormente, se presenta un mecanismo basado en reputación con el que se pueda calcular la confianza inicial que se le podría asignar a una nueva unidad de detección, que desea unirse al CIDN de un dominio de seguridad por motivos de colaboración. Este modelo de la confianza enuncia diferentes soluciones para el cálculo inicial de la reputación, dependiendo de si el tipo de la nueva unidad de detección es un IDS de la infraestructura instalado por un administrador, un IDS instalado en el dispositivo móvil de un usuario final o un CIDN que desea unirse al CAS para mejorar su precisión en la detección de ataques distribuidos. Cualquiera de estos tres actores son entidades recién llegadas, conocidas como *newcomer*, que, o son unidades totalmente desconocidas para el sistema (problema cold-start), o son unidades que ya han cooperado con otras entidades en el pasado (problema bootstrapping).

5.1. Sistema de reputación interdominio

En el capítulo anterior, concretamente en la Sección 4.3, se detalla el mecanismo de reputación diseñado para determinar si el IDS emisor de una alerta, dentro de un único CIDN, se puede considerar con la suficiente confianza para difundir esa alerta al resto de IDSs de su dominio de seguridad. Esta sección va un paso más allá, para establecer un nuevo modelo de reputación que sea capaz de establecer relaciones de confianza a un nivel interdominio, donde más de un CIDN se unen entre sí para establecer una red cooperativa de alertas denominada CAS. Cuando el WCL de un CIDN en particular decide propagar una alerta detectada localmente, esa alerta se distribuye a los WCLs de otros dominios de seguridad con los que el CIDN de origen mantiene una relación de confianza. Un ejemplo ilustrativo de esta propagación interdominio se muestra en la Figura 4.1, donde se puede considerar que el líder de WC_A (WCL_A) envía una alerta detectada por uno de sus IDSs a sus vecinos homólogos: WCL_C y WCL_E.

En este contexto interdominio, el WC del CIDN que recibe la alerta (WC_C y WC_E en el ejemplo anterior) tiene que evaluar la reputación del CIDN del dominio de origen (WC_A) para decidir si aceptar o no su alerta como confiable. En caso de ser un dominio de confianza, la alerta se distribuye para todos los IDSs del CIDN de destino. Sino, la alerta es descartada sin tenerla en consideración como evidencia de un posible ataque distribuido. Con este requisito en mente, se propone la adaptación y aplicación de uno de los modelos de confianza más extendidos y conocidos, como es PeerTrust.

PeerTrust [161] es un sistema de confianza basado en la reputación de sus entidades participantes. Este sistema proporciona un modelo de confianza adaptativo y coherente capaz de cuantificar y comparar cómo de buenas son esas entidades con respecto a sus comportamientos. Con ese fin, PeerTrust se basa en un sistema de opiniones según las interacciones anteriores que han tenido esas entidades con el sistema.

El modelo interdominio PeerTrust tiene dos características principales. Por un lado, introduce tres *elementos básicos de la confianza* y dos *factores de contexto*, adaptados para calcular la bondad de las entidades participantes. Por el otro lado, también define una *métrica de confianza* global con la que poder combinar esos elementos y factores según el escenario de aplicación. Esos elementos de confianza se basan en las opiniones que una entidad recibe de otras sobre el número total de interacciones que otra tercera ha realizado, mientras que la credibilidad se basa en las fuentes de esas opiniones. Sobre los factores de contexto, éstos se dividen en uno sobre la transacción (o interacción) y otro sobre la comunidad de todas las entidades donde se ejecuta esa interacción.

Todos los elementos y factores de contexto que necesita PeerTrust para su correcto funcionamiento, así como otras variables que se utilizan más adelante en las diferentes ecuaciones, se encuentran resumidos a continuación.

- $I(\Omega_u, \Omega_v)$ simboliza el número total de interacciones –alertas intercambiadas– que han sido realizadas por el dominio Ω_u con el dominio Ω_v.
- $I(\Omega_u)$ es el número total de interacciones realizadas por el dominio Ω_u –las alertas intercambiadas en nuestro escenario de detección– con el resto de dominios.
- $p(\Omega_u, i)$ indica el resto de los dominios participantes en la interacción i-ésima con Ω_u. En concreto, los CIDNs con una relación de confianza, sin contar a Ω_u.
- $S(\Omega_u, i)$ define el valor normalizado de *satisfacción* que el dominio Ω_u obtiene de $p(\Omega_u, i)$ en su interacción i-ésima.
- $Cr(\Omega_v)$ hace referencia a la *credibilidad* en la opinión dada por el dominio Ω_v.
- $TF(\Omega_u, i)$ representa el factor de contexto adaptado sobre la i-ésima *transacción* en la que el dominio Ω_u ha participado.
- y $CF(\Omega_u)$ indica el factor de contexto adaptado sobre la *comunidad* de entidades a la que pertenece el dominio Ω_u.

El valor de confianza en el dominio Ω_u lo define PeerTrust con (5.1), donde α y β establecen pesos normalizados definidos por los administradores de cada dominio según, respectivamente, la evaluación colectiva y la comunidad de entidades.

$$T(\Omega_u) = \alpha \left(\sum_{i=1}^{I(\Omega_u)} S(\Omega_u, i) \cdot Cr(p(\Omega_u, i)) \cdot TF(\Omega_u, i) \right) + \beta \cdot CF(\Omega_u) \qquad (5.1)$$

Aceptar una interacción dependerá entonces de si $T(\Omega_u)$ alcanza un mínimo umbral de confianza, definido por los administradores de cada dominio de seguridad.

Hasta este punto, se ha introducido la métrica que propone PeerTrust como modelo interdominio, dejando la definición de sus elementos al escenario final de aplicación. A continuación se presenta el cálculo de esos elementos para el diseño del sistema de confianza interdominio con el que modelar el comportamiento de los CIDNs, siendo ésta la adaptación propuesta en el contexto interdominio del CAS.

5.1.1. Satisfacción sobre la alerta publicada por otro CIDN

La satisfacción que puede obtener un CIDN de una alerta externa, que otro CIDN de confianza haya generado en sus dominios internos, está muy supeditada a que esa alerta realmente represente un ataque distribuido, y que el primero haya sido, o esté siendo, víctima del mismo. En ese caso, solamente se podrán obtener evidencias (a priori) reales a través de las alertas que hayan compartido sus propios IDSs. En caso contrario, donde el CIDN receptor de una alerta externa no tenga una experiencia directa del incidente ocurrido, este CIDN solamente podrá solicitar recomendaciones a otros terceros CIDNs de confianza que les pueda informar, o poner en preaviso, del comportamiento que haya tenido con ellos el CIDN emisor para aceptar sus alertas como verdaderas.

La obtención de experiencias directas sobre un evento se ha definido en (4.5), dentro de la Sección 4.3.1 para el cálculo de la reputación de un IDS en particular, donde $|\vartheta_G|$ indica el número total de IDSs que han detectado la alerta internamente en un CIDN. Si $|\vartheta_G| = 0$, este hecho indicaría que no existen evidencias sobre la alerta recibida desde otro CIDN, y que se tendrían que solicitar opiniones a otros terceros a fin de comprobar si dicha alerta tiene alguna repercusión a nivel del CAS.

Para el cálculo de la satisfacción en la alerta i publicada por el CIDN de un dominio genérico Ω_j, generada internamente por uno o varios de sus IDSs, se considera que dicho cálculo lo lleva a cabo el dominio Ω_u, como uno de los dominios de confianza en el que Ω_j mantiene un acuerdo de colaboración. Nótese que el resto de los CIDNs, en la misma red de confianza, también realizarán los mismos cálculos al recibir la misma alerta que Ω_u. Este cálculo de la satisfacción, denotado como $S_{\Omega_u}(\Omega_j, i)$, se realiza según (5.2).

$$S_{\Omega_u}(\Omega_j, i) = \begin{cases} Rep_{\Omega_u}(j) \times \bigoplus_{k=1}^{|p(\Omega_u,i)|} Rec_{\Omega_k,\Omega_j} \cdot T_{\Omega_k}^{(t-1)}, \forall j \in \vartheta_S & \text{si } |\vartheta_G| = 0 \\ \bigoplus_{k=1}^{|p(\Omega_u,i)|} Rec_{\Omega_k,\Omega_j} \cdot T_{\Omega_k}^{(t-1)} & \text{en otro caso} \end{cases} \tag{5.2}$$

donde $|p(\Omega_u, i)|$ indica el número de dominios participantes en la alerta i; $T_{\Omega_k}^{(t-1)}$ el último valor sobre la confianza en Ω_k; Rec_{Ω_k,Ω_j} la recomendación del k-ésimo dominio con el que Ω_u mantiene una relación de confianza; y $Rep_{\Omega_u}(j), \forall j \in \vartheta_S$, la reputación en promedio de los IDSs en Ω_u capaces de detectar la misma alerta i, calculada en (4.3).

5.1.2. Credibilidad en la opinión suministrada por otro CIDN

Dado que la credibilidad es para un dominio $\Omega_w \in p(\Omega_u, i)$ en particular, denotada como $Cr(\Omega_w)$, su cálculo se basa en una *medida de similitud* que es personalizada entre ese dominio y cualquier otro dominio Ω_v, con el que valorar la opinión dada por Ω_v con respecto a otros. Esta medida se define en (5.3), donde $I(\Omega_u)$ representa el conjunto de dominios que han interaccionado anteriormente con el dominio Ω_u.

$$Cr(p(\Omega_u, i)) = \frac{Sim(p(\Omega_u, i), \Omega_w)}{\sum_{j=1}^{I(\Omega_u)} Sim(p(\Omega_u, j), \Omega_w)} \tag{5.3}$$

definiéndose la función de similitud $Sim(\Omega_v, \Omega_w)$ como se muestra en (5.4), donde $I(\Omega_v) \cap I(\Omega_w)$ es el conjunto común de dominios que han interaccionado tanto con Ω_v como con Ω_w, el cual viene denotado por $IJS(\Omega_v, \Omega_w)$. El objetivo es comparar cuál ha sido la satisfacción que han tenido los otros dominios que también han recibido la alerta i. Es decir, cómo de parecidos son los dominios a la hora de evaluar lo mismo.

$$Sim(\Omega_v, \Omega_w) = 1 - \sqrt{\frac{\sum\limits_{\Omega_x \in IJS(\Omega_v, \Omega_w)} \left(\frac{\sum_{i=1}^{I(\Omega_x, \Omega_v)} S(\Omega_x, i)}{I(\Omega_x, \Omega_v)} - \frac{\sum_{i=1}^{I(\Omega_x, \Omega_w)} S(\Omega_x, i)}{I(\Omega_x, \Omega_w)} \right)^2}{|IJS(\Omega_v, \Omega_w)|}} \quad (5.4)$$

5.1.3. Modelado de los factores de contexto

El factor de contexto de las transacciones, denotado por $TF(\Omega_u, i)$, puede incorporar varios contextos para que a las opiniones que sean más importantes sobre una alerta se les pueda asignar un peso mayor que a otras menos importantes. Entre estos contextos de transacciones, se pueden destacar, por ejemplo, el tipo de emisor de la alerta (NIDS o HIDS), la severidad de la alerta o la fecha y hora de envío.

Dentro del contexto de un CAS, se ha optado por definir una métrica donde tener en cuenta tanto el *nivel de pertenencia* del CIDN emisor –grado de pertenencia a un nivel de confianza (ver Figura 4.4)– como el tipo de emisor de la alerta, ya que esas alertas generadas por el IDS de un administrador se deberían de considerar más importantes y confiables que las generadas por los dispositivos móviles de los usuarios finales. De esta manera, el factor de contexto de las transacciones se define según (5.5).

$$TF(\Omega_u, i) = \varepsilon_{\Omega_u}(i) + (\kappa_{\Omega_u} \cdot pNIDS + (1 - \kappa_{\Omega_u}) \cdot pHIDS) \quad (5.5)$$

donde $\varepsilon_{\Omega_u}(i) \in [0, 1]$ es el nivel de pertenencia en Ω_u del emisor de i a un conjunto difuso, según la Figura 4.4; $pNIDS$ y $pHIDS$ la proporción, respectivamente, de NIDSs y HIDSs que han detectado la alerta; y $\kappa_{\Omega_u} \in [0, 1]$ el peso definido por el administrador de Ω_u para balancear la importancia de la alerta i según el tipo de emisor.

Por otro lado, el problema de incentivar o recompensar a las entidades del sistema para que participen y proporcionen valoraciones sobre otras, siendo éste un problema clásico que sufren los sistemas de reputación actuales, lo define PeerTrust en su propia métrica de confianza, a través del factor de contexto de la comunidad. En el contexto de esta tesis doctoral, el factor de contexto se calcula mediante (5.6), donde $F(\Omega_u)$ indica el número total de opiniones que el dominio Ω_u proporciona al resto de dominios.

$$CF(\Omega_u) = \frac{F(\Omega_u)}{I(\Omega_u)} \quad (5.6)$$

Finalmente, como se hizo en el sistema de reputación intradominio presentado en la Sección 4.3, la Tabla 5.1 muestra un resumen de todos los pesos, variables y otros parámetros utilizados en esta sección para el establecimiento del sistema de reputación interdominio. Como en el caso anterior, en esta tabla se incluye una breve descripción de cada una de esas variables, así como sus valores de inicialización y cálculo dependiendo de qué entidad es la encargada de su mantenimiento y configuración.

Variable	Descripción	Inicialización/cálculo
	Ecuación (5.1)	
$T(\Omega_u)$	Valor de confianza del dominio Ω_u	Calculada en (5.1) y utilizada en (5.2)
$I(\Omega_u)$	Total de interacciones realizadas por Ω_u con el resto de dominios	Almacenado en Ω_u y también utilizado en (5.3) y (5.6)
$p(\Omega_u, i)$	Dominios participantes en la i-ésima interacción con Ω_u	Valor almacenado en Ω_u
α_{Ω_u}	Peso de evaluación colectiva	Administrador de Ω_u
β_{Ω_u}	Peso del contexto de la comunidad	Administrador de Ω_u
	Ecuación (5.2)	
$S_{\Omega_u}(\Omega_j, i)$	Satisfacción de Ω_u sobre la alerta i incluyendo la obtenida de $p(\Omega_u, i)$	Calculada en (5.2) y utilizada en (5.1) y (5.4)
ϑ_G	Total de IDSs que han detectado la alerta en un CIDN	Utilizada en (4.5)
$Rep_{\Omega_u}(j)$	Reputación de los IDSs en Ω_u	Calculada en (4.3), $\forall j \in \vartheta_S$
Rec_{Ω_k,Ω_j}	Recomendación del k-ésimo dominio que es confiable para Ω_u	Suministrada por Ω_k y que se puede calcular como en (4.2)
	Ecuación (5.3) y (5.4)	
$Cr(\Omega_v)$	Credibilidad en toda la información suministrada por cada $\Omega_v \in p(\Omega_u, i)$	Calculada en (5.3) y utilizada en (5.1)
$Sim(\Omega_v, \Omega_w)$	Similitud con cada $\Omega_v \in p(\Omega_u, i)$	Evaluar dominios en común
$IJS(\Omega_v, \Omega_w)$	Conjunto común de los dominios que han interaccionado con Ω_v y Ω_w	Almacenado en Ω_v y Ω_w
$I(\Omega_x, \Omega_v)$	Interacciones hechas por Ω_x con Ω_v	Almacenado en Ω_x
	Ecuación (5.5)	
$TF(\Omega_u, i)$	Factor de contexto de la transacción i-ésima para el dominio Ω_u	Calculada en (5.5) y utilizada en (5.1)
$\varepsilon_{\Omega_u}(i)$	Nivel de pertenencia en Ω_u del CIDN emisor de la alerta i	Según el conjunto difuso, como se muestra en la Figura 4.4
κ_{Ω_u}	Peso en alertas según tipo de emisor	Administrador de Ω_u
$pNIDS$	Proporción de NIDSs al detectar i	Sujeto a sus capacidades
$pHIDS$	Proporción de HIDSs al detectar i	Sujeto a sus capacidades
	Ecuación (5.6)	
$CF(\Omega_u)$	Factor de contexto de la comunidad de CIDNs para Ω_u	Calculada en (5.6) y utilizada en (5.1)
$F(\Omega_u)$	Total de opiniones enviadas por Ω_u	Almacenado en Ω_u

Tabla 5.1: Pesos y variables del sistema de reputación interdominio

5.2. Perfil de comunicaciones interdominio

Al igual que con el perfil presentado en la Sección 4.4, sobre el comportamiento del sistema de detección a nivel intradominio, esta nueva sección se adentra en los mismos términos que entonces pero, en esta ocasión, enfocada en las interacciones interdominio. Es decir, entre los CIDNs de varios dominios de seguridad que interactúan entre sí a fin de compartir y, en consecuencia, construir un conocimiento colectivo interdominio de alertas. Este conocimiento permitirá la detección de ataques distribuidos que pudieran darse en cualquier área administrativa del CAS, así como la identificación de los CIDNs con un comportamiento malicioso publicando información fraudulenta.

En la Figura 5.1, se muestra un diagrama de secuencia con las distintas interacciones interdominio que se producen entre los CIDNs de un CAS de ejemplo, cuya distribución de la arquitectura de red es la misma que la presentada en la Figura 4.1.

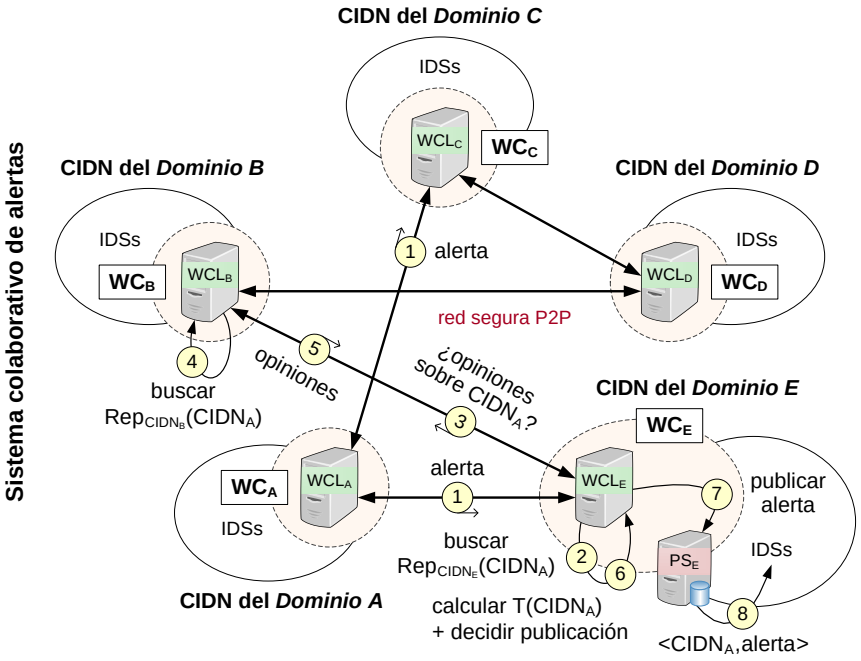


Figura 5.1: Diagrama de secuencia para las comunicaciones interdominio

A continuación se detallan cada una de esas interacciones, que se han etiquetado con un número para identificar claramente a cuál se está haciendo referencia.

Paso 1: $CIDN_A$ publica una nueva alerta

Se considera que el CIDN de un dominio de seguridad, llamado $CIDN_A$, publica a través de su líder (WCL_A) una alerta detectada por uno o varios de sus IDSs al resto de CIDNs, con los que mantiene una relación de confianza. Nótese que este perfil arranca una vez terminado el Paso 13 del perfil de comunicaciones intradominio, presentado en la Sección 4.4, después de que WCL_A haya decidido compartir dicha alerta.

Observando la Figura 5.1, los líderes de WC_C y WC_E recibirán una copia de la alerta por la red segura de comunicaciones P2P. Ambos canales deben estar configurados con SSL/TLS como mecanismo de seguridad, usando los certificados que han intercambiado con WC_A al establecer las relaciones de confianza entre sus CIDNs. La alerta enviada por WCL_A a los dos líderes WCL_C y WCL_E, sólo la aceptarán como válida si $CIDN_A$ tiene un mínimo nivel de confianza para ambos dominios de seguridad.

Esta sección detalla los pasos que tiene que seguir $CIDN_E$ para evaluar la confianza sobre $CIDN_A$, y así poder aceptar o no su alerta como válida –no fraudulenta. $CIDN_C$ también debe hacer esos mismos pasos, aunque cada CIDN utilizará sus propios valores de reputación, por lo que se obvia su explicación por duplicidad en sus funciones.

Paso 2: ¿buscar datos para calcular $T(CIDN_A)$, confianza sobre $CIDN_A$?

Después de que WCL_E reciba la alerta desde WCL_A, éste tiene que comprobar si el valor de reputación almacenado para ese emisor, $Rep_{CIDN_E}(CIDN_A)$, todavía se puede considerar como válido. Este hecho se puede deber a que ese valor ha sido calculado recientemente, y no es necesario volver a solicitar a otros dominios sus opiniones para realizar de nuevo el mismo cálculo. De esta manera, se evita la sobrecarga de tráfico y cómputo en la red para calcular un valor que todavía se puede considerar correcto. En ese caso, WCL_E estaría en disposición de pasar directamente al Paso 6.

Paso 3: solicitar opiniones a los dominios confiables del CAS sobre $CIDN_A$

WCL_E, al no mantener un valor actual válido sobre $CIDN_A$, tendrá que solicitar al resto de dominios de seguridad con los que mantiene una relación de confianza sus opiniones o recomendaciones acerca de $CIDN_A$. Esta solicitud solamente la enviará a $CIDN_B$, a través de su líder WCL_B, ya que es el único CIDN en el que confía.

Este mensaje de opinión se envía por la red segura P2P establecida anteriormente entre los dos dominios, entre WCL_E y WCL_B. Como se puede observar en la Figura 5.1, $CIDN_E$ también mantiene otra relación de confianza con $CIDN_A$ pero, como es obvio, no se le enviará una solicitud de opinión al dominio objeto de evaluación.

Paso 4: $CIDN_B$ recupera sus recomendaciones sobre $CIDN_A$

El líder de $CIDN_B$, WCL_B, recupera de su base de datos interna el último valor de recomendación –satisfacción normalizada más actual– que mantiene acerca de $CIDN_A$. Es decir, el parámetro $S(CIDN_A, I(CIDN_A))$ definido por PeerTrust en (5.1), y cuyo cálculo se presenta en (5.2) como base de la reputación de $CIDN_A$.

Paso 5: enviar las opiniones sobre $CIDN_A$ al dominio de seguridad solicitante

WCL_B le envía a WCL_E el valor normalizado de la satisfacción que tiene $CIDN_B$ en las interacciones realizadas en el pasado con $CIDN_A$. Esta comunicación se lleva a cabo por la red segura P2P, creada previamente durante la solicitud de opinión.

Paso 6: calcular $T(CIDN_A)$ y tomar una decisión de publicación

Después de que WCL_E haya recibido la opinión que $CIDN_B$ tiene sobre $CIDN_A$, este líder calcula la credibilidad sobre la fuente de información que le ha suministrado esa opinión. Es decir, WCL_E calcula $Cr(CIDN_B)$ mediante (5.3). Por otro lado, tanto el factor de la transacción como el factor de la comunidad, denotados respectivamente como $TF(CIDN_A, I(CIDN_A))$ y $CF(CIDN_A)$, también son calculados a fin de poder obtener la evaluación final en la confianza depositada en $CIDN_A$. En concreto, todos estos parámetros son la base para el cálculo de $T(CIDN_A)$, según (5.1).

Paso 7: publicar la dupla <CIDN$_A$,alerta>

En el caso de que WCL_E haya aceptado como válida la alerta recibida por $CIDN_A$, WCL_E publicará la dupla <CIDN$_A$,alerta> a todos los miembros (IDSs) de su CIDN a través de su Servicio de Publicación, denominado PS_E.

De manera similar a los pasos realizados por WCL_E, WCL_C también decidirá si publicar o no la misma alerta entre los distintos IDSs de su CIDN, teniendo en cuenta sus propios datos sobre $CIDN_A$ y la posible opinión suministrada por $CIDN_D$.

Paso 8: traducir la alerta y enviar la dupla a todos los miembros del CIDN

Siguiendo el Paso 12 del modelo intradominio, presentado en la Sección 4.4, PS_E reenvía la dupla anterior a su *Módulo de Traducción* para que sea traducida, según el software de detección que tenga instalado cada uno de los IDSs desplegados en $CIDN_E$. Una vez realizada esa traducción, PS_E enviará una copia traducida de la dupla a cada uno de los IDSs utilizando el asunto `NUEVA_ALERTA_DISTRIBUIDA`.

5.3. Resultados experimentales

Al igual que la Sección 4.5 con el sistema de reputación intradominio, a continuación se presentan varios experimentos, junto con una detallada discusión de sus resultados, para la evaluación del sistema de reputación interdominio presentado en este capítulo. Estas pruebas se han realizado sobre un entorno multidominio de simulación, con el objetivo principal de analizar el porcentaje de las alertas que este sistema de reputación interdominio es capaz de clasificar entre verdaderas o falsas. Es decir, si provienen de CIDNs con un comportamiento benevolente o malicioso, respectivamente.

En este entorno de simulación se ha configurado un CAS compuesto por 50 CIDNs, con 20 HIDSs y 10 NIDSs para cada uno de ellos. Es decir, se ha desplegado un total de 1000 HIDSs y 500 NIDSs en todo el CAS. El resto de parámetros de configuración se detallan en cada una de las pruebas experimentales, ya que varían de una a otra para así poder obtener diferentes conclusiones según el objetivo de evaluación en particular. Con respecto a la proporción de CIDNs maliciosos que se han desplegado en el sistema colaborativo de alertas, se han realizado cuatro pruebas en cada uno de los experimentos variando el número de CIDNs maliciosos entre el 20, 40, 60 y el 80 %.

Destacar que, en estas pruebas, se han usado los mismos pesos que en la Sección 4.5.1 para el cálculo en un CIDN genérico Ω de la reputación sobre sus IDSs, a fin de mantener esas mismas condiciones experimentales para analizar los nuevos resultados: $\alpha_i = 0,4$, $\beta_i = 0,35$ y $\gamma_i = 0,25$, para un $WC_i \in CIDN_\Omega$ en (4.1). Sobre los pesos del sistema de reputación interdominio en la evaluación colectiva y en el contexto de la comunidad, definidas por PeerTrust en (5.1), se han fijado $\alpha_\Omega = 0,75$ y $\beta_\Omega = 0,25$ para dar mayor peso a las experiencias directas de $CIDN_\Omega$ que a las obtenidas de su comunidad. Por último, y como en la Sección 4.5, se han inyectado aleatoriamente 9000 alertas, aunque aquí sólo se han simulado alertas con severidad *High* al ser las únicas, según la propuesta realizada en esta tesis doctoral, que se gestionan en entornos multidominios.

Con la variación anterior en el número de CIDNs maliciosos, junto con los distintos parámetros específicos que se van a definir para cada una de las pruebas, se pretenden evaluar y dar respuesta a los dos siguientes puntos:

- Cómo el modelo de reputación interdominio propuesto puede mejorar el ratio de detección mediante la eliminación de las alertas fraudulentas generadas por los distintos CIDNs con un comportamiento malicioso.
- Qué impacto tiene en el modelo de reputación interdominio el número de los IDSs generando la misma alerta en un CIDN, y que luego se comparte a nivel del CAS.

Todos los resultados obtenidos de estos dos experimentos son analizados y discutidos en profundidad en las siguientes secciones. Nótese que, durante la ejecución de todas las pruebas, se considera que un CIDN genérico Ω expone un comportamiento malicioso cuando su confianza $T(CIDN_\Omega) \leq 0,25$.

5.3.1. Ratio de la detección del sistema colaborativo de alertas

El objetivo principal detrás de este primer experimento se centra en poder analizar el porcentaje (ratio) de aciertos que tiene un determinado CIDN cuando tiene que decidir qué alertas, provenientes de CIDNs de otros dominios de seguridad, son aceptadas y, por tanto, distribuidas a todos los IDSs dentro su propio CIDN.

Como parámetros dentro de este experimento, se han llevado a cabo cuatro pruebas con diferentes valores de pertenencia, ε en (5.5), comprendidos entre 0,25 y 1 con un intervalo de 0,25, donde para cada valor de pertenencia se ha variado el porcentaje de CIDNs maliciosos entre el 20, 40, 60, y el 80 %. Los resultados de las cuatros pruebas realizadas en este experimento se muestran gráficamente en la Figura 5.2a. Nótese que, para el cálculo del factor de contexto de las transacciones definido en (5.5), se considera que la importancia sobre las alertas generadas dentro de un CIDN por los NIDSs es el doble que si hubieran sido generadas por los HIDSs, por lo que $\kappa = \frac{2}{3}$.

Analizando las pruebas realizadas, se puede observar que prácticamente el 100 % de las alertas enviadas por otros dominios son descartadas cuando el valor de pertenencia es de 0,25, a excepción de un pequeño intervalo situado en la franja del 20 % de CIDNs maliciosos, que se analizará posteriormente con mayor detalle.

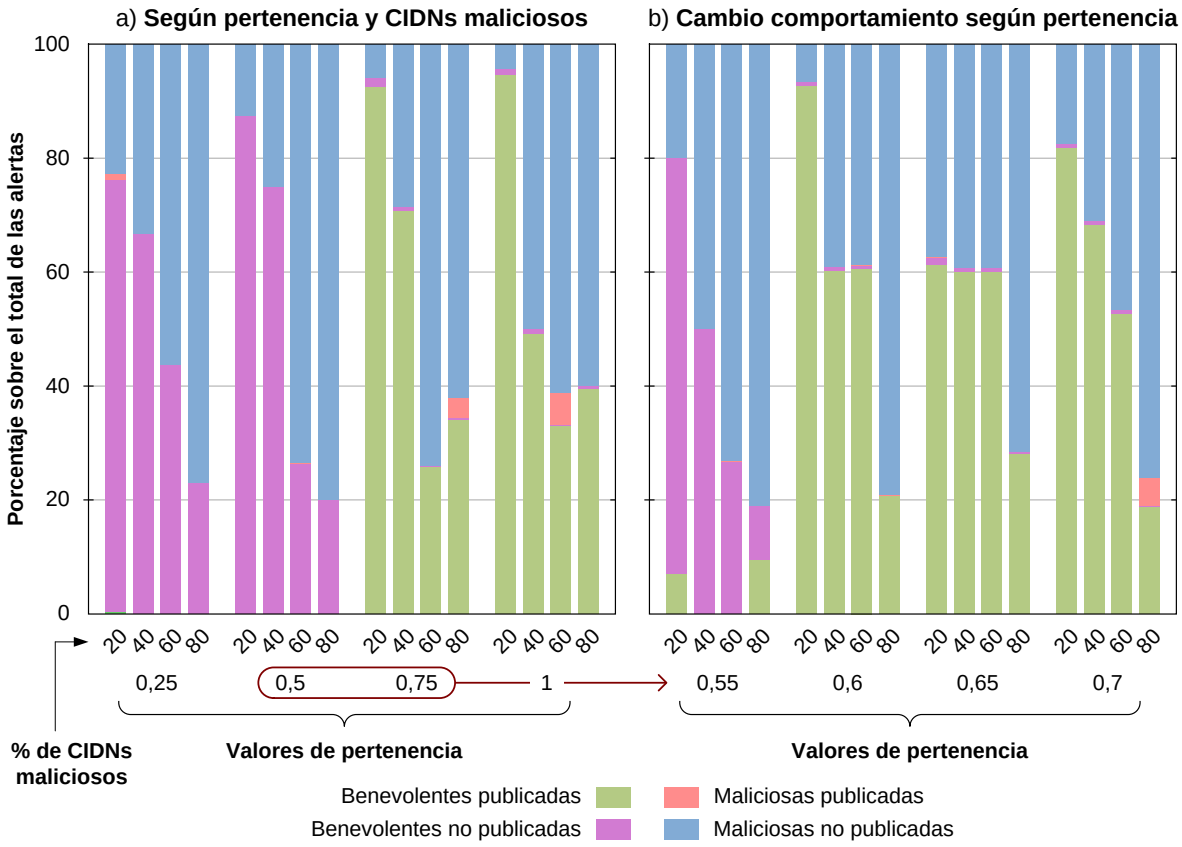


Figura 5.2: Reparto de alertas según pertenencia y porcentaje de CIDNs maliciosos

La causa principal en el rechazo de alertas se debe a que el grado de pertenencia del emisor al nivel *High* es demasiado bajo, implicando un valor de reputación que, aunque sea alto, no es suficiente para aceptar sus alertas como (posibles) hechos acontecidos en la realidad. Esa falta de reputación del CIDN que generó la alerta afecta, en gran medida, al cálculo del valor de la confianza para el emisor de dicha alerta.

Con respecto al 1,21 % de alertas fraudulentas que han sido aceptadas con un 20 % de CIDNs maliciosos, cuando se deberían haber descartado, se debe fundamentalmente al dinamismo que ofrece el factor de contexto de la comunidad (CF), definido en (5.6). Ese factor se encuentra altamente relacionado con las interacciones que los dominios de seguridad realizan entre sí, teniendo un peso muy significativo en la confianza global definida por PeerTrust, como se puede comprobar en (5.1). Este hecho puede conducir a que un CIDN consiga aumentar la confianza final sobre una de sus alertas si alcanza un valor superior en el cálculo del factor de la comunidad. Por ejemplo, siendo el emisor de todas las alertas para un dominio de seguridad en concreto.

En referencia al valor 0,5 sobre la pertenencia, los resultados son parecidos al caso anterior cuando $\varepsilon = 0,25$. El grado de pertenencia sigue siendo bajo, teniendo en cuenta los conjuntos difusos definidos en la Figura 4.4, lo cual afecta a la confianza global del CIDN emisor para que cualquier otro CIDN acepte sus alertas como verdaderas.

Los resultados de las dos últimas pruebas, cuando $\varepsilon = 0,75$ y $\varepsilon = 1$, sí arrojan una serie de conclusiones totalmente distintas a las analizadas en los dos casos anteriores. Valores de pertenencia más altos no afectan de forma tan negativa a la confianza global sobre un CIDN emisor, haciendo que una mayor cantidad de las alertas sean clasificadas correctamente: emitiendo y/o descartando la gran mayoría de alertas benevolentes o maliciosas dependiendo del porcentaje de CIDNs maliciosos en el CAS.

Como se puede observar en la Figura 5.2a, se produce un cambio significativo en el comportamiento del sistema cuando el valor de pertenencia pasa de 0,5 a 0,75. En la Figura 5.2b se muestra más detalladamente cómo se comporta el sistema de reputación interdominio en ese rango de valores de la pertenencia. Ahí se puede comprobar que el cambio se produce entre los valores 0,55 a 0,6 para la pertenencia, pasando de aceptar sobre un 10 % de las alertas benevolentes, con un 20 % de CIDNs maliciosos, a casi la totalidad de las mismas, y descartando con éxito todas las alertas maliciosas.

5.3.2. Cantidad aleatoria de los IDSs que componen un CIDN

En el experimento anterior, el número de IDSs en cada CIDN detectando las alertas se suponía que era fijo, por lo que en este nuevo experimento se pretende variar dicho número para que cada alerta sea detectada por un número aleatorio de IDSs. El resto de características en este experimento son las mismas que en el anterior, manteniendo el mismo número de CIDNs, proporción de IDSs y valores de pertenencia.

Los resultados se muestran gráficamente en la Figura 5.3.

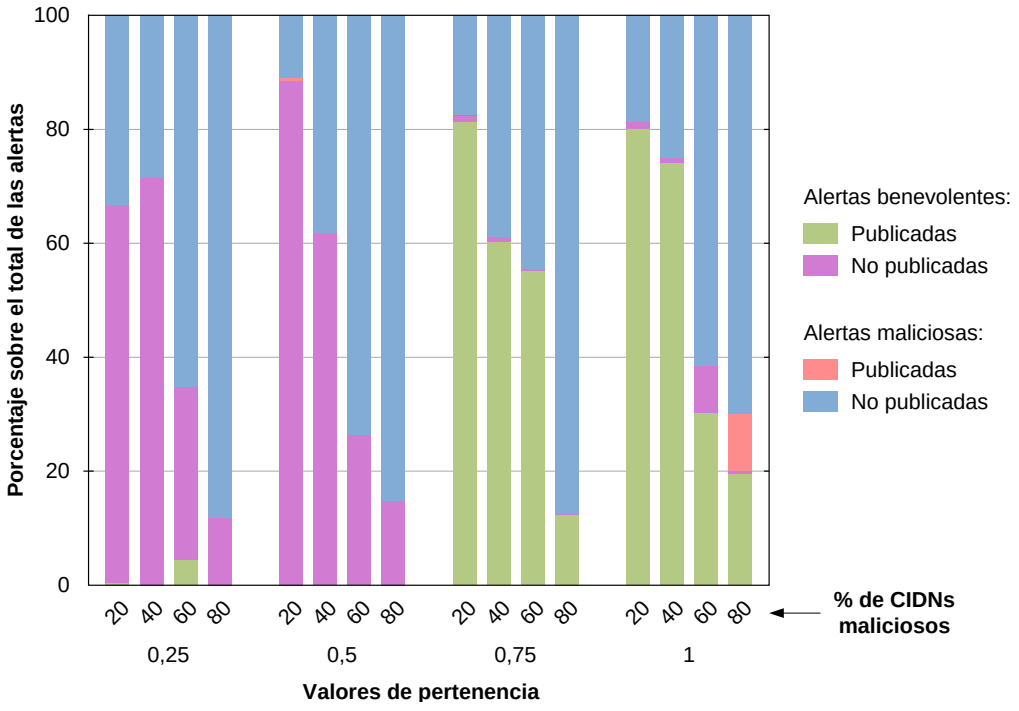


Figura 5.3: Distribución de alertas con un número aleatorio de IDSs

La aleatoriedad de los IDSs dentro de un CIDN, en la detección de una alerta, va a influir con cierta notoriedad en la posible aceptación de las alertas, ya que tanto la satisfacción como el factor de contexto de las transacciones, definidos en (5.2) y (5.5), respectivamente, suponen un peso importante a la hora de calcular la confianza global sobre cualquier CIDN, y por ende, en la aceptación o rechazo de sus alertas.

Como se puede observar viendo la Figura 5.3, casi la totalidad de las alertas son descartadas con un valor de 0,25 en la pertenencia, notándose una pequeña variación cuando el porcentaje de CIDNs maliciosos es del 60 %. Al igual que en el experimento anterior, para valores similares, la publicación de alertas fraudulentas se establece en un 4,35 %, debiéndose, en gran medida, al factor de contexto de la comunidad.

En cambio, cuando la pertenencia aumenta a 0,5, sí que son descartadas casi todas las alertas, ya sean verdaderas o fraudulentas, independientemente del porcentaje de CIDNs maliciosos presentes en el CAS. El cambio de comportamiento vuelve a repetirse como en el experimento anterior, poco antes de establecer la pertenencia al valor 0,75. En este último caso, la gran mayoría de alertas benevolentes son entonces aceptadas, y por tanto difundidas a todos los IDSs del CIDN como alertas verdaderas, descartándose casi en su totalidad el resto de alertas al considerarlas como maliciosas.

Por último, con un valor máximo de 1 en la pertenencia, se intuye de antemano que se tendrían que haber clasificado correctamente todas las alertas. A pesar de ello, este hecho no es del todo cierto si el CAS tiene un porcentaje de CIDNs maliciosos mayor del 60 %. Como ejemplo, en la Figura 5.3 se puede ver que se descartan un 8,16 % de las alertas benevolentes con un 60 % de los CIDNs maliciosos y, con un 80 % de CIDNs maliciosos, se publican sobre un 10 % de las fraudulentas. Sin embargo, los resultados anteriores muestran un ratio de error que podrían llegar a asumirse como porcentajes normales en un entorno de experimentación, debidos casi con total seguridad al factor de aleatoriedad en los IDSs que generan todas las alertas anteriores.

Debido al componente de aleatoriedad que se ha introducido para la ejecución de este experimento, la satisfacción definida en (5.2) se verá disminuida si las alertas son detectadas por un pequeño número de IDSs, haciendo que la confianza global del CIDN emisor, definida en (5.1), también se vea decrementada en consecuencia.

A modo de conclusión, y una vez analizados los resultados de los dos experimentos anteriores, se puede afirmar que el sistema de reputación interdominio, propuesto al comienzo de este capítulo, es capaz de clasificar correctamente, en promedio, hasta un 95,2 % de las alertas inyectadas en un entorno multidominio de simulación, cuando es superior a 0,6 la pertenencia del CIDN emisor. Además, también se puede confirmar que este modelo de confianza interdominio basado en reputación es suficientemente robusto cuando el porcentaje de CIDNs maliciosos en el CAS alcanza valores hasta del 80 %, llegando a clasificar correctamente hasta un 89,7 % de las alertas difundidas por la red. Comparando estos resultados con los obtenidos en el capítulo anterior, sobre el sistema de reputación intradominio, se puede constatar que el modelo interdominio presenta un ratio de error ligeramente superior, lo cual puede considerarse algo obvio al tener las alertas un significado distinto entre áreas de detección distantes.

5.4. Evaluación de nuevas unidades de detección

En todas las definiciones y experimentos que se han presentado hasta el momento, se ha supuesto que el sistema de confianza tenía toda la información necesaria para el cálculo de la reputación de cualquier unidad de detección, ya sea mediante experiencias directas o indirectas. Sin embargo, esta asunción no siempre es correcta. En ocasiones, el sistema de reputación va a tener que efectuar el cálculo de la confianza de una unidad de detección, al menos una vez, sin poder recurrir a información sobre el comportamiento que ha tenido esa unidad en el pasado. Este caso se conoce en la literatura actual como el problema *cold-start*, cuando la unidad de detección es la primera vez que se une al sistema para colaborar. Estas unidades pueden ser *IDSs estáticos* de la infraestructura o un *dominio de seguridad* que desea unirse con otros para detectar ataques distribuidos. En este escenario, también se pueden incluir las "pequeñas" unidades de detección que ofrecen los usuarios móviles. Al igual que con las unidades anteriores, los *IDSs móviles* también van a encarar el problema cold-start al menos en una ocasión, cuando vayan a colaborar con el CAS por primera vez. A partir de ese momento, los IDSs móviles se enfrentan a otro problema conocido como *bootstrapping*, cuando se vuelvan a unir al CAS aunque sea en otro CIDN distinto al que habían cooperado antes. En este caso, el CAS ya tendrá información sobre el comportamiento previo del nuevo IDS móvil. En ambos problemas, los sistemas de reputación se enfrentan a un mismo reto: ¿qué valor inicial de la confianza se le puede asignar a una nueva unidad de detección?

Como ejemplo de cómo evaluar esas tres nuevas unidades de detección, se considera el CAS de la Figura 5.4, con los CIDNs de nueve dominios de seguridad de dos dominios administrativos distintos: AD_1 y AD_2. Nótese que AD_1 sigue la misma distribución de la arquitectura de red utilizada tanto en la Figura 4.1 como en la Figura 5.1.

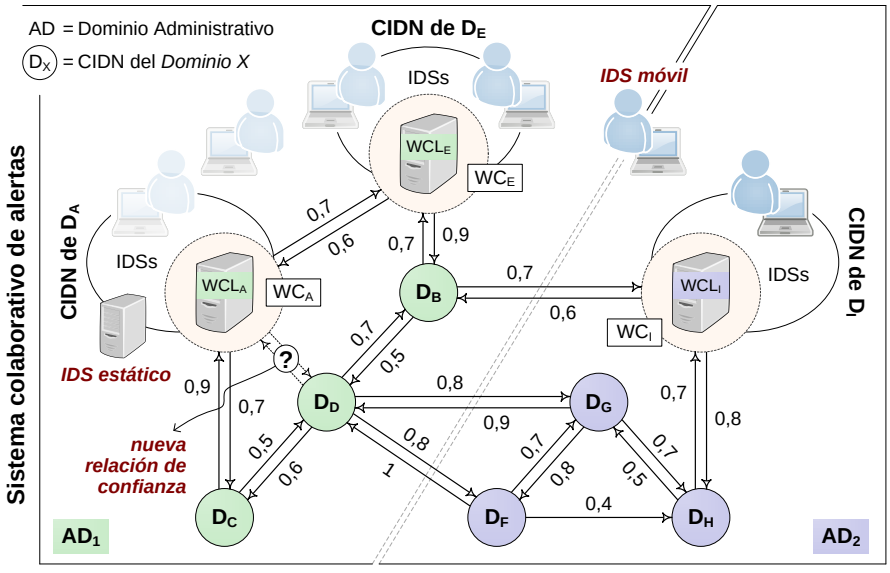


Figura 5.4: Ejemplo de los tres tipos de unidades que pueden unirse al CAS

Aunque en esa figura solamente se muestre, por claridad, la estructura interna de tres dominios de seguridad, el resto también se organiza de la misma manera. Además, en este ejemplo también se pueden apreciar los tres tipos de unidades de detección en cuestión: IDSs estáticos, IDSs móviles y dominios de seguridad que desean establecer una *nueva relación de confianza* con el CIDN de otro dominio de seguridad.

Siguiendo el ejemplo de la Figura 5.4, el CIDN donde un nuevo IDS estático desea unirse sólo puede contar con la información que éste le proporcione para el cálculo de su valor inicial de confianza, como sus capacidades de detección, por ejemplo. Sin embargo, en este mismo cálculo, pero entre dos dominios de seguridad, también se pueden hacer uso de las recomendaciones que otros dominios puedan ofrecer. Por ejemplo, D_A puede incorporar las recomendaciones que le envíe D_C sobre el nuevo dominio D_D a la hora de calcular su reputación, como medio sustituto de la confianza.

Por otro lado, el IDS móvil se traslada de un dominio de seguridad a otro con deseos de colaborar en el envío de las alertas que pueda detectar, saltando incluso entre dos dominios administrativos. Ese IDS móvil se une inicialmente al CIDN de D_A con el que nunca ha colaborado, situación que refleja claramente el problema cold-start ya que ni el CIDN de D_A ni ningún otro dominio de seguridad en el CAS tiene información histórica sobre ese IDS móvil (es un completo desconocido para el CAS). Al no poder disponer de experiencias pasadas con el IDS móvil, ya sean directas o indirectas, D_A solamente puede basar el cálculo de la confianza inicial del IDS móvil en la información que este mismo le pueda facilitar. Como segunda etapa en la trayectoria de desplazamiento del IDS móvil, éste se mueve de D_A al CIDN de D_E. Este nuevo CIDN no tiene información previa sobre el IDS móvil, asumiendo que es un total desconocido desde la perspectiva de D_E, aunque no se puede considerar como tal desde una visión global a nivel del CAS. Ya ha colaborado previamente con otro dominio con el que D_E tiene una relación directa de confianza, así como de otras relaciones indirectas por medio de terceros dominios de confianza. Este último ejemplo refleja el problema bootstrapping.

La Figura 5.5 muestra los posibles caminos de confianza entre D_E y D_A, denotados como $tp_i(D_E, D_A)$, a través de los que propagar las solicitudes de opinión preguntando sobre el comportamiento que el IDS móvil ha tenido en el CIDN de D_A.

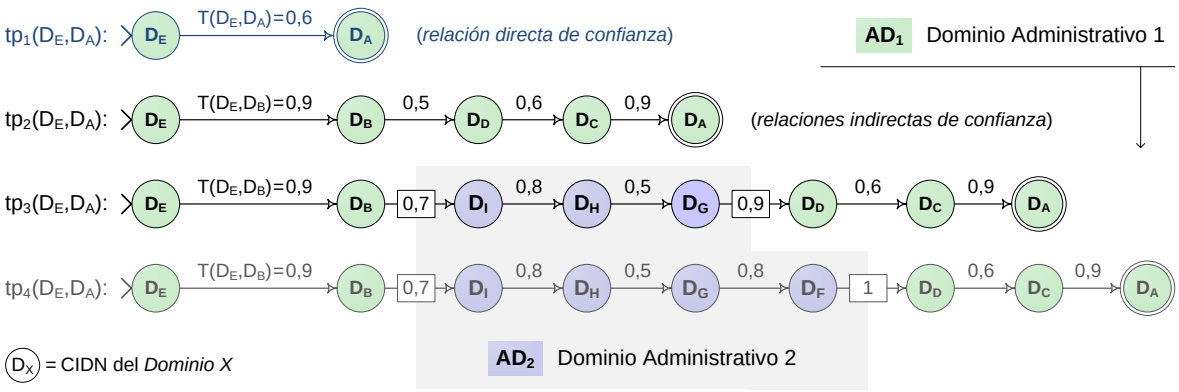


Figura 5.5: Todos los posibles caminos de confianza entre D_E y D_A

El camino de confianza más corto entre D_E y D_A, para que D_E pueda solicitar a D_A su recomendación sobre el IDS móvil, es la relación directa de confianza existente entre ambos: tp_1 con un valor de confianza $T_{tp_1}(D_E, D_A) = 0,6$. Aunque es el camino de confianza más corto, también existen otras alternativas para inferir la confianza entre ambos dominios de seguridad a través de relaciones indirectas (transitivas) de confianza. A primera vista, podría ser evidente que los caminos de confianza más cortos son más confiables o "fuertes" que los largos, como se sugiere en [243, 244]. Sin embargo, en [245] se discute en profundidad que la confianza inferida de caminos largos con altos niveles de confianza pueden llegar a ser más fuertes (confiables) que los inferidos a partir de caminos más cortos con bajos niveles de confianza. En este sentido, D_E tendría que calcular su confianza en D_A de la siguiente manera, utilizando para ello tp_2 y tp_3:

$$T_{tp_2}(D_E, D_A) = \begin{cases} 0,725 & \text{media} \\ 0,243 & \bigotimes \end{cases} \qquad T_{tp_3}(D_E, D_A) = \begin{cases} 0,757 & \text{media} \\ 0,122 & \bigotimes \end{cases}$$

Se han usado dos funciones típicas en el cálculo de la confianza sobre cada camino de confianza: la media aritmética y una función de agregación multiplicativa, denotada por $\otimes$. Ambas relaciones indirectas de confianza parecen ofrecer, inicialmente, unos resultados más altos con la media aritmética que aquellos proporcionados por la relación directa de confianza entre D_E y D_A (tp_1). Entre estos caminos de confianza, tp_3 tiene incluso un valor de confianza ligeramente superior a los obtenidos a través de tp_2. De esta manera, D_E tendría que escoger tp_3 como el mejor camino de confianza a través del cual solicitar a D_A las recomendaciones sobre el IDS móvil, al ser éste el camino de confianza más confiable entre los tres primeros caminos de la Figura 5.5.

A pesar de todo ello, también se tendrían que tener en cuenta otros factores en el cálculo de la confianza de estos caminos. Por ejemplo, penalizando progresivamente los caminos cuando sean cada vez más largos. Conforme la longitud se vaya incrementando, la confianza que el dominio solicitante (D_E en el ejemplo anterior) puede tener en estos dominios intermedios será cada vez más baja, teniéndose en cuenta para ello una función de penalización. Otro de los factores que podrían ser importantes, y que se tendría que tener en cuenta, es el posible cambio entre los distintos dominios administrativos. Este valor de confianza se tendría que ver decrementado entonces de acuerdo a la confianza que el dominio administrativo solicitante tiene depositada en el resto. Retomando el ejemplo anterior, esta penalización haría cambiar la toma de decisión a optar por tp_2, en lugar de utilizar tp_3, ya que este último camino necesita un cambio entre dos dominios administrativos. Con respecto al último camino de confianza, entre todos los mostrados en la Figura 5.5, tp_4 ni tan siquiera se llega a construir. Este camino es demasiado largo y excede la longitud máxima, superando el umbral que en las redes P2P normalmente se establece en siete saltos en este tipo de redes [246].

Finalmente, el IDS móvil se mueve de D_E a D_I en una última etapa, cambiando a un nuevo dominio administrativo. Esta etapa presenta el mismo caso que el anterior, ya que el IDS móvil es totalmente desconocido para D_I pero no así para el CAS, con la única salvedad de que D_I tendrá que centrar su búsqueda por relaciones indirectas de confianza hasta alcanzar D_A y D_E, con los que el IDS móvil ya ha participado.

5.5. Sistema de reputación de nuevas unidades

En esta sección, se presenta de forma detallada el modelo de gestión de la confianza basado en reputación con el que poder calcular la reputación inicial, como sustituto de la confianza, de una nueva unidad de detección (*newcomer*) antes de que ésta se una al sistema colaborativo para la detección de ataques distribuidos.

El sistema de confianza y reputación propuesto en esta sección se estructura en tres grandes bloques principales, todos detallados en la Sección 5.5.4, según cada uno de los tres tipos de unidades de detección siguientes:

- Un *IDS estático*, instalado por uno de los administradores dentro del dominio de seguridad y que lo ha desplegado de forma permanente en un área de detección concreta donde monitorizar sus servicios y recursos locales.
- Un *IDS móvil*, perteneciente a un usuario final, con el que desea colaborar en el CIDN de cada dominio de seguridad a lo largo de su trayectoria de desplazamiento proporcionándole las alertas que su dispositivo móvil pueda generar.
- Dos *dominios de seguridad*, que desean unirse entre sí para mejorar su precisión y cobertura de la detección en el descubrimiento de ataques distribuidos.

En este cálculo de la reputación se van a incorporar tres factores con los que dar soporte a dicho cálculo según el tipo de la nueva unidad de detección, además de poder disponer siempre de las recomendaciones que puedan ofrecer otros terceros dominios de confianza. Estos factores, que son definidos a continuación en la Sección 5.5.1, se pueden resumir como las *capacidades de detección* que la nueva unidad detección puede ofrecer al CIDN del dominio de seguridad donde desea colaborar. En este sentido, un CIDN va a ser capaz de asignar un valor de reputación más alto a aquellos IDSs que proporcionen alertas i) sobre ataques desconocidos hasta el momento para el resto de IDSs o ii) sobre ciertas zonas de detección monitorizadas por IDSs (presuntamente) sospechosos al tener una reputación demasiado baja. Por otro lado, el cálculo de la reputación entre dos dominios de seguridad también va a estar supeditado a sus capacidades de detección, comprobando qué similitud tienen a la hora de detectar ataques en los que ambos están interesados. Cuanto más alta sea la similitud entre las capacidades de detección, mayor será la reputación (confianza) inicial entre ambos dominios de seguridad.

A pesar de todo lo expuesto más arriba, las capacidades de detección que declare poseer un IDS, o un dominio de seguridad en particular, podrían no ser válidas si esa unidad, como un intento malicioso, desea conseguir una reputación inicial más alta que la que se merecería obtener en la realidad. Debido a ello, el sistema de reputación debe estar capacitado para poder actualizar la reputación, a lo largo del tiempo, de distinta manera según la bondad mostrada por la unidad de detección a la hora de declarar sus capacidades de detección. En consecuencia, la reputación de una unidad de detección se verá incrementada o decrementada más rápidamente según las capacidades de detección que la unidad de detección proporcionó durante su proceso de unión a un CIDN. Esta fase también se le suele conocer como bootstrapping.

5.5.1. Modelado de las capacidades de detección

Las capacidades de detección se estructuran en tres modelos distintos, que pretenden proveer al sistema de reputación de información sobre la *utilidad* de la nueva unidad de detección para el CIDN y su *voluntad* en colaborar con dicho CIDN, pidiéndole ajustar sus capacidades de detección para que el CIDN reciba alertas en las que esté interesado.

Definición 5. El espectro completo de capacidades de detección (*detection skill*), que necesita un CAS para la detección de ataques distribuidos, se define como un conjunto $DS = \{ds_1, ds_2, ..., ds_l\}$, donde l es el número total de capacidades de detección y ds_i, con $1 \leq i \leq l$, una capacidad específica para detectar un determinado tipo de ataque.

Debido a que un CAS está compuesto de un número concreto de CIDNs, cada uno de ellos va a requerir en sus fueros internos un subconjunto de las capacidades globales de detección que el CAS tiene definido como agrupación.

Definición 6. El conjunto de las $l_x \leq l$ capacidades de detección, necesarias para el CIDN de un dominio de seguridad $D_x \in CAS$ en particular, se define como un conjunto $DS_x = \{ds_{i_1}, ds_{i_2}, ..., ds_{i_{l_x}}\}$, cumpliéndose que $DS_x \subseteq DS$.

El conjunto DS_x anterior lo especifican los diferentes administradores de D_x según las vulnerabilidades presentes en todos los componentes y servicios que se encuentran desplegados en D_x, siendo de esta manera el conjunto DS la fusión completa de todas las capacidades de detección que cada CIDN del CAS necesita de forma individual.

Cada una de las capacidades de detección que van a necesitar los CIDNs, y por tanto el CAS como fusión de ellas, las proporcionan las distintas unidades de detección al tenerlas instaladas y configuradas en su software de detección interno.

Definición 7. La i-ésima capacidad de detección ds_i de una unidad de detección se concreta a través de una tripleta ds_i=<`nombre,num_politicas_seguridad,prioridad`>, que está compuesta por el nombre de la capacidad de detección, el número de políticas de seguridad empleadas en la detección de un ataque y un valor de prioridad indicando el nivel de severidad sobre cada una de esas reglas de detección.

Como ejemplo, se pueden considerar los 34 tipos de ataques que es capaz de detectar Snort, los cuales se clasifican según cuatro prioridades: *High*, *Medium*, *Low* y *Very Low*. Cuando la prioridad es más baja, más alto es el nivel de severidad (o riesgo) de la alerta. Una prioridad con un valor 1 representa la severidad más alta (*High*), mientras que 4 es el menos severo (*Very Low*), como define Snort en [10]. De manera similar, *Traffic Light Protocol* (TLP) [247] define cuatro grados de sensibilidad para cualquier intercambio de información, utilizando para ello la gama de colores *Red*, *Amber*, *Green* y *White*.

Definición 8. Una unidad de detección j (*detection unit*) se modela dependiendo de sus capacidades de detección como $DU_j = (du_1, du_2, ..., du_l)$, siendo cada $du_i \in [0, 1]$, con $1 \leq i \leq l$, el porcentaje de las políticas de seguridad que se necesita para alcanzar la cobertura definida por ds_i en sus propósitos de detección. Este porcentaje de políticas se encuentra definido para cada ds_i en el elemento `num_politicas_seguridad`.

Por tanto, la utilidad que atesora DU_j para cualquiera de los CIDNs del CAS va a depender de las políticas de seguridad que es capaz de proporcionarle a dicho CIDN con el que está colaborando, o pretende hacerlo ya que DU_j acaba de lanzar su proceso de bootstrapping. Nótese también que si $du_i = 0$, eso significa que la unidad de detección no soporta en absoluto la capacidad de detección que se define en ds_i, mientras que si $du_i = 1$, entonces indica que sí es capaz de soportar todas las políticas de seguridad necesarias para la detección del correspondiente tipo de ataque.

Definición 9. Un *vector acumulador* es definido como $\overline{DS} = \{\overline{ds}_1, \overline{ds}_2, ..., \overline{ds}_l\}$, donde cada $\overline{ds}_i \in \mathbb{N}$ se asocia a la i-ésima capacidad de detección $ds_i \in DS$, con $1 \leq i \leq l$, que representa el número de unidades de detección en el CAS que han reconocido durante su fase de bootstrapping que soportan ds_i. Al igual que antes, $\overline{DS}_x = \{\overline{ds}_{i_1}, \overline{ds}_{i_2}, ..., \overline{ds}_{i_{l_x}}\}$ es el vector acumulador para el CIDN del dominio de seguridad $D_x \in CAS$.

Como ejemplo de vector acumulador, $\overline{DS}_x = \{5, 0\}$ denotaría que existen 5 unidades de detección que proporcionan ds_{i_1} en la detección de su correspondiente tipo de ataque, mientras que el elemento "0" declararía que el segundo tipo de ataque, que podría ser detectado haciendo uso de ds_{i_2}, no está cubierto en DS_x por ninguna de las unidades que actualmente están colaborando con ese dominio de seguridad.

Modelo de utilidad de una nueva unidad de detección como IDS

La utilidad que cualquier IDS puede tener para el CIDN de un dominio de seguridad, al ser una unidad de detección, representa el beneficio o interés que tiene el dominio en admitir las alertas que ese IDS le pueda proporcionar como valiosas. El objetivo detrás de este interés, por parte de un determinado dominio de seguridad, es poder mejorar sus capacidades de detección en las que está interesado, así como que le permita obtener más información de seguridad sobre ciertas capacidades de detección sobre las que no tiene desplegados los suficientes IDSs para monitorizarlos, o sí los tiene, pero mediante ciertos IDSs que no le ofrecen la suficiente confianza –valor de reputación relativamente bajo– como para admitir cualquiera de sus alertas como verdadera.

La función de utilidad de una determinada unidad de detección DU_j para el CIDN de un dominio de seguridad D_x, denotado como $\Phi_x(DU_j) \in [0, 1]$, se define según (5.7), donde $du_{i_k} \in DU_j$ y $\overline{ds}_{i_k} \in \overline{DS}_x$, $\forall k \in [1, n_k]$.

$$\Phi_x(DU_j) = \sum_{k=1}^{n_x} \frac{du_{i_k}}{n_x + \overline{ds}_{i_k}} \tag{5.7}$$

Analizando (5.7) con mayor profundidad, se puede comprobar que:

- Si $du_{i_k} = 0$, $\forall k$, entonces $\Phi_x(DU_j) = 0$. Este es el resultado deseado, ya que DU_j no ofrece ninguna capacidad de detección en las que D_x está interesado.
- Si $du_{i_k} = 1$ y $\overline{ds}_{i_k} = 0$, $\forall k$, entonces $\Phi_x(DU_j) = \sum_{i=1}^{n_x} \frac{1}{n_x} = 1$.

En resumen, cada uno de los ds_{i_k} de DU_j puede contribuir, en el mejor de los casos, un total de $\frac{1}{n_x}$ al valor final de $\Phi_x(DU_j)$; es decir, cuando $DU_{i_k} = 1$ y $\overline{ds}_{i_k} = 0$. Más allá de los casos extremos, el IDS no mejorará las capacidades de detección del dominio D_x cuando $du_{i_k} = 0$, mientras que sí que contribuirá con $\frac{1}{n_x + \overline{ds}_{i_k}}$ cuando $du_{i_k} = 1$.

Por tanto, conforme $\overline{ds}_{i_k}$ sea más grande, evidenciando que hay más IDSs capaces de detectar ds_{i_k}, más baja será la contribución de ds_{i_k} a $\Phi_x(DU_j)$. En consecuencia, (5.7) es reformulada por (5.8) para que se considere la reputación de los IDSs, representados por $\psi_x(ds_{i_k}) \in [0, 1]$, además del número de IDSs que ofrecen ds_{i_k} $(\overline{ds}_{i_k})$.

$$\Phi_x(DU_j) = \sum_{k=1}^{n_x} \frac{du_{i_k}}{n_x + (\overline{ds}_{i_k} \cdot \psi_x(ds_{i_k}))} \tag{5.8}$$

Si existen muchos IDSs que facilitan ds_{i_k}, pero $\psi_x(ds_{i_k}) \to 0$ y $du_{i_k} = 1$, entonces ese ds_{i_k} debería producir una contribución para $\Phi_x(DU_j)$ cercana a $\frac{1}{n_x}$. Sin embargo, un promedio simple de la reputación podría llevar a producir valores *pobres* de $\psi_x(ds_{i_k})$. Por ejemplo, se puede considerar $\overline{ds}_{i_k} = 100$ y tener las dos siguientes posibilidades:

1) Hay 90 IDSs con una reputación demasiado baja, por ejemplo, con un valor de 0,1 en promedio, pero existen otros 10 IDSs con una reputación máxima. Por tanto, la reputación de todos esos IDSs sería $\psi_x(ds_{i_k}) = 0,19$ en promedio, aunque ds_{i_k} estaría perfectamente cubierta por esos diez últimos IDSs.

2) La mitad de IDSs tienen una reputación de 0,4 en promedio, y la otra mitad una reputación de 0,6. En este caso, $\psi_x(ds_{i_k}) = 0,5$ (superior al caso anterior).

Entre las dos posibilidades anteriores, la primera debería ofrecer mejores resultados que la segunda, ya que ds_{i_k} estaría cubierta por, al menos, 10 IDSs con un mayor nivel de reputación. Sin embargo, la reputación global en promedio para este ejemplo indica justo lo contrario. Para evitar este tipo de inconsistencias, la función $\psi_x(ds_{i_k})$, definida en (5.9), tiene en cuenta tanto la dispersión como la distribución de todos los valores de reputación de los IDSs mediante una medida de dispersión.

$$\psi_x(ds_{i_k}) = max\{Rep_x(DU_{ds_{i_k}})\} - \varphi(\{Rep_x(DU_{ds_{i_k}})\}),\ \forall DU \in D_x \tag{5.9}$$

donde $DU_{ds_{i_k}}$ son cada una de las unidades de detección $DU \in D_x$ que proporcionan ds_{i_k}; $max\{Rep_x(DU_{ds_{i_k}})\}$ la reputación más alta de todas las unidades de detección en el conjunto $DU_{ds_{i_k}}$; y $\varphi(\{Rep_x(DU_{ds_{i_k}})\})$ una medida de dispersión con la que se valora la distribución de la reputación de los IDSs que proporcionan ds_{i_k}. Entre las posibles medidas de dispersión, se puede usar la diferencia absoluta o rango, la diferencia media o la desviación estándar, siendo esta decisión de los administradores de D_x.

Como ejemplo, si se selecciona la diferencia media como medida de dispersión, esta diferencia se calcularía según (5.10).

$$\varphi(\{Rep_x(DU_{ds_{i_k}})\}) = \frac{\sum\limits_{DU_l \in DU_{ds_{i_k}}} \sum\limits_{DU_j \in DU_{ds_{i_k}}} |Rep_x(DU_l) - Rep_x(DU_j)|}{|DU_{ds_{i_k}}|^2} \tag{5.10}$$

Volviendo al ejemplo anterior, los resultados de $\psi_x(ds_{i_k})$ se tendrían que recalcular utilizando la diferencia media definida en (5.10). Los nuevos resultados serían:

1) $\psi_x(ds_{i_k}) = 1 - 0,164 = 0,836$

2) $\psi_x(ds_{i_k}) = 0,6 - 0,101 = 0,499$

Analizando estos resultados, se puede afirmar que la aplicación de cualquier medida de dispersión proporciona unos resultados bastante más realistas que cuando se calcula la reputación en promedio de un conjunto de IDSs.

Modelo de voluntad de una nueva unidad de detección ante la colaboración

Otro de los factores a tener en cuenta durante el cálculo de la reputación inicial de cualquier nueva unidad de detección, especialmente para el caso de los IDSs móviles, es la voluntad que éstos tienen en colaborar, de cara a poder negociar un conjunto distinto de las capacidades de detección que tiene en ese momento de unirse a un CIDN.

Durante ese proceso, el líder del CIDN donde desea colaborar el IDS móvil le solicita primero sus capacidades de detección para poder evaluar su utilidad para ese dominio de seguridad. Ese líder tendrá entonces la oportunidad de negociar un conjunto distinto de sus capacidades de detección, incluyendo ciertas variaciones que sean más interesantes desde el punto de vista de su CIDN. En ese caso, el líder le sugiere una configuración alternativa, lo más parecida posible a la que inicialmente envió el IDS móvil, consistente en activar, desactivar o mejorar algunas de sus capacidades de detección. Esa propuesta, por parte del líder, se representa como $\widehat{DU}_j = (\widehat{du}_1, \widehat{du}_2, ..., \widehat{du}_n)$, con $\widehat{du}_i \in [du_i, 1]$, $\forall i$. Como punto final, el IDS móvil respondería con un conjunto final de capacidades de detección a partir de las que el líder pueda construir el nuevo $\widetilde{DU}_j = (\widetilde{du}_1, \widetilde{du}_2, ..., \widetilde{du}_n)$.

En la Figura 5.6 se muestra el proceso completo para negociar el mejor conjunto de capacidades de detección entre WCL_x (líder del dominio D_x) y el nuevo IDS móvil.

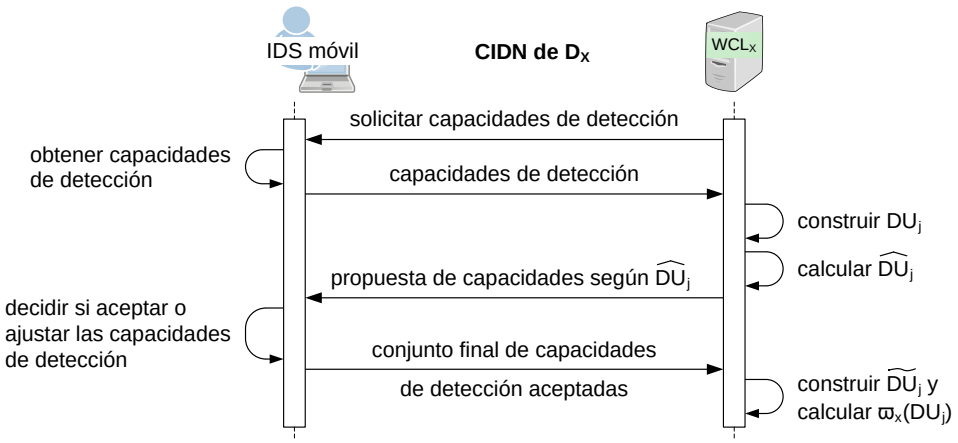


Figura 5.6: Proceso de negociación para obtener mejores coberturas de detección

Como resultado, el valor final de $\widetilde{du_i}$ dependerá de la voluntad que tiene el nuevo IDS móvil a colaborar con el CIDN del dominio de seguridad. El modelado de la voluntad ante la colaboración, denotado como $\varpi_x(DU_j)$, se define según (5.11).

$$\varpi_x(DU_j) = \sum_{i=1}^{n} \frac{\rho(du_i, \widetilde{du_i}, \widehat{du_i})}{n} \tag{5.11}$$

donde $\rho(du_i, \widetilde{du_i}, \widehat{du_i})$ se define formalmente como:

$$\rho(du_i, \widetilde{du_i}, \widehat{du_i}) = \begin{cases} 0 & \text{si } \widetilde{du_i} \leq du_i \\ \dfrac{\widetilde{du_i} - du_i}{\widehat{du_i} - du_i} & \text{si } du_i < \widetilde{du_i} < \widehat{du_i} \\ 1 & \text{si } \widetilde{du_i} \geq \widehat{du_i} \end{cases}$$

Un valor $\widetilde{du_i} \geq \widehat{du_i}$ significaría que DU_j está dispuesto a mejorar ds_i, incluso más allá del nivel esperado por el líder del CIDN, mientras que $\widetilde{du_i} \leq du_i$ significa que DU_j va a ofrecer un nivel peor o igual al proporcionado en la primera fase de negociación (no hay voluntad de colaboración). Por último, $\frac{\widetilde{du_i} - du_i}{\widehat{du_i} - du_i} \in [0, 1]$ evalúa el grado de mejoría que proporciona DU_j con respecto al cumplimiento inicial sobre ds_i.

Modelo de similitud entre dominios de seguridad

Cuando los CIDNs de dos dominios de seguridad del CAS desean colaborar entre sí, independientemente de que sean del mismo o de diferentes dominios administrativos, ambos dominios de seguridad tienen que establecer una relación segura y de confianza para intercambiar todo el material necesario para la detección de ataques distribuidos. Sobre todo, las alertas que los dos puedan generar en sus ámbitos locales de detección por separado. Como ejemplo, el establecimiento de una nueva relación de confianza se puede ver representado en la Figura 5.4 entre los dominios D_A y D_D.

El proceso de bootstrapping a este nivel, entre dominios de seguridad, implica que cada uno de esos dominios tiene que calcular un valor de reputación inicial sobre el otro en dos procesos separados (uno por cada dominio) para un establecimiento bidireccional de la relación de confianza. En este proceso, se hace uso de un modelo de similitud entre dominios de seguridad, el cual determina la afinidad existente entre ambos con respecto a las capacidades de detección en las que cada uno está interesado.

Por lo tanto, la similitud entre dos dominios de seguridad cualesquiera, D_x y D_y, denotado como $\lambda(D_x, D_y) \in [0, 1]$, se calcula mediante (5.12).

$$\lambda(D_x, D_y) = \frac{|DS_x \cap DS_y|}{|DS_x \cup DS_y|} \tag{5.12}$$

Por ejemplo, supongamos $DS_x = \{ds_{i_1}, ds_{i_2}, ds_{i_3}\}$ y $DS_y = \{ds_{i_2}, ds_{i_3}, ds_{i_4}\}$, por lo que la similitud en las capacidades de detección entre D_x y D_y se asemejaría a:

$$\lambda(D_x, D_y) = \frac{|\{ds_{i_1}, ds_{i_2}, ds_{i_3}\} \cap \{ds_{i_2}, ds_{i_3}, ds_{i_4}\}|}{|\{ds_{i_1}, ds_{i_2}, ds_{i_3}\} \cup \{ds_{i_2}, ds_{i_3}, ds_{i_4}\}|} = \frac{2}{4} = 0,5$$

5.5.2. Recomendaciones de fuentes de información del CAS

Como se ha comentado en la Sección 5.4, cuando una nueva unidad de detección DU_j se intenta unir a colaborar con un determinado dominio D_x, ya sea un IDS móvil o un dominio de seguridad, D_x tiene la opción de solicitar a otros terceros dominios de confianza dentro del CAS sus propias recomendaciones sobre DU_j. El objetivo detrás de esta búsqueda es encontrar el camino más reputable hasta el dominio de seguridad más confiable que tenga recomendaciones sobre el comportamiento que ha tenido DU_j, en sus procesos de compartir las alertas que ha detectado en dicho dominio.

Para este proceso de búsqueda, D_x le solicita a cada uno de sus dominios vecinos de confianza D_y si tiene alguna información de comportamiento –recomendaciones o advertencias– acerca de la nueva unidad de detección DU_j, denotada como $Rec_y(DU_j)$. De entre todas las respuestas que obtenga de esos dominios vecinos, D_x se quedará con aquél valor que represente el camino con el nivel de confianza más alto, después de que sea ponderado con la confianza directa existente entre D_x y cada uno de esos dominios D_y; a saber, $T(D_x, D_y)$. En el caso de que un dominio vecino de confianza D_y no tenga información de reputación sobre DU_j, $\nexists Rec_y(DU_j)$, o incluso dicho valor no se puede considerar como válido al no ser reciente, este proceso de búsqueda se propagaría por todo el CAS de forma recursiva entre todos los vecinos de D_y, mientras no se alcance el *Tiempo de Vida Máximo* (del inglés Time to Live, TTL) que establece el número de saltos entre dominios durante la construcción de los caminos de confianza.

En este proceso, hay que tener en cuenta que solamente se propaga de retorno una única respuesta de recomendación. Es decir, la recomendación sobre DU_j a través del camino de confianza más reputable, actuando cada dominio intermedio como un nodo receptor en el proceso de consulta. La principal ventaja de este enfoque es doble. Por un lado, todas las relaciones directas de confianza entre los dominios no son reveladas al resto, preservando así la privacidad de los dominios de seguridad con respecto a sus relaciones de confianza. Por otro lado, la sobrecarga en las comunicaciones que pudiera introducir este mecanismo de consulta también se vería reducida, ya que la cantidad de mensajes transmitidos por la red se reducen al máximo.

Por tanto, el mejor camino de confianza construido, que maximice la confianza que D_x puede depositar en el dominio D_y con $Rec_y(DU_j)$, se calcula según (5.13). Este valor $Rec_y(DU_j)$, también denotado como $Rec_{CAS}(DU_j)$ a partir de ahora, será propagado de vuelta hasta D_x, junto con el camino de confianza tp_i más reputable cuyo $T_{tp_i}(D_x, D_y)$ sea máximo desde la perspectiva de D_x.

$$T_{tp_i}(D_x, D_y) = \frac{1}{|tp_i|} \cdot \sum_{D_j, D_k \in tp_i}^{|tp_i|} \frac{T(D_j, D_k) \cdot \left(\frac{1}{\Delta t + 1}\right) \cdot \upsilon(D_j, D_k)}{|tp_i|} \tag{5.13}$$

donde (D_j, D_k) representa cada uno de los pares consecutivos de dominios del CAS que componen el camino de confianza tp_i; $|tp_i|$ la longitud del mismo hasta D_x; Δt el tiempo transcurrido desde la última interacción entre D_j y D_k; y $\upsilon(D_j, D_k)$ indicando si D_j y D_k están o no emplazados en el mismo dominio administrativo.

La última función $\upsilon(D_j, D_k)$, utilizada en (5.13), con la que se pueda valorar si D_j y D_k pertenecen al mismo dominio administrativo AD_z, se define como:

$$\upsilon(D_j, D_k) = \begin{cases} 1 & \text{si } D_j, D_k \in DA_z \\ \varrho_j \in [0,1] & \text{en otro caso} \end{cases}$$

Es decir, $\upsilon(D_j, D_k) = 1$ si D_j y D_k pertenecen al mismo dominio administrativo, o, en caso contrario, un valor constante $\varrho_j \in [0,1]$ que lo hayan definido de antemano los administradores del dominio administrativo donde se encuentre desplegado D_j.

Analizando (5.13) con mayor detenimiento, el primer término $|tp_i|^{-1}$ se utiliza para penalizar a los caminos de confianza que sean más largos, en igualdad de condiciones. Como se ha comentado antes en la Sección 5.4, cuanto más largo sea un camino, menor será la confianza en los dominios que componen ese camino conforme sea cada vez más largo en longitud. En otras palabras, conforme "más alejado" se encuentre el dominio que pueda ofrecer la información de reputación sobre DU_j, menos fiable será el camino de confianza que conduzca hasta esa fuente de información. Además de la función de penalización anterior, lo que (5.13) pretende calcular es, en esencia, un valor promedio ponderado de los valores directos de confianza entre cada par de dominios de seguridad a lo largo del camino de confianza. Todos esos pesos corresponden, por un lado, con la mencionada función $\upsilon(D_j, D_k)$ y, por otro lado, con el tiempo Δt transcurrido desde que se llevó a cabo la última interacción entre D_j y D_k. De esta manera, cuanto más grande sea Δt entre esos dos dominios D_j y D_k, más pequeño será el peso para el valor de confianza directa $T(D_j, D_k)$ entre ambos dominios.

Por último, para el caso particular donde DU_j sea un IDS móvil intentando unirse al CIDN del dominio de seguridad D_x, denotado por $mIDS_j$, también se le puede solicitar a otros IDSs móviles que estén colaborando con D_x, $mIDS_k \in D_x$ en ese momento, si alguno de ellos tiene alguna recomendación sobre $mIDS_j$ al haber coincidido en algún otro dominio de seguridad. Recomendación denotada como $Rec_{mIDS_k}(mIDS_j)$.

El valor final de la recomendación sobre $mIDS_j$, desde la perspectiva del dominio D_x, y denotada como $Rec_{mIDS}(D_x, mIDS_j)$, se calcula a través de (5.14).

$$Rec_{mIDS}(D_x, mIDS_j) = \sum_{mIDS_k \in D_x} \frac{Rec_{mIDS_k}(mIDS_j) \cdot \left(\dfrac{1}{\Delta t + 1}\right)}{|\{mIDS_k \in D_x\}|} \tag{5.14}$$

De nuevo, la agregación de todas las recomendaciones obtenidas de los IDSs móviles colaborando con D_x, $mIDS_k \in D_x$, los cuales proporcionan datos de recomendación sobre $mIDS_j$, $Rec_{mIDS_k}(mIDS_j)$, se calcula a través de la media ponderada teniendo en cuenta lo reciente en el tiempo de cada una de esas recomendaciones. La confianza sobre estas recomendaciones, denotada como $T_{mIDS}(D_x, mIDS_j)$, se define según (5.15) que, como anteriormente, también está sujeta a la confianza que el dominio de seguridad D_x puede depositar en el IDS móvil que actualmente está colaborando con D_x.

$$T_{mIDS}(D_x, mIDS_j) = \sum_{mIDS_k \in D_x} \frac{Rep_{D_x}(mIDS_k)}{|\{mIDS_k \in D_x\}|} \tag{5.15}$$

Finalmente, mediante (5.16) se calcula un valor de recomendación único teniendo en cuenta la información recibida de todas las fuentes que tengan alguna recomendación acerca del comportamiento de la nueva unidad de detección DU_j: las recomendaciones provenientes de otros dominios de seguridad dentro del CAS y las proporcionadas por los IDSs móviles que ya están colaborando con el dominio de seguridad D_x.

Los dos parámetros anteriores corresponden, respectivamente, con $Rec_{CAS}(DU_j)$ y $Rec_{mIDS}(D_x, DU_j)$. Es decir, con las dos funciones que han sido definidas anteriormente tanto en (5.13) como en (5.14). Con ambas funciones, destacar también que todas estas recomendaciones se ponderan según la confianza que D_x tenga en esas fuentes externas de información que proporcionan los distintos valores de recomendación.

$$\begin{aligned} Rec_{CAS,mIDS}(D_x, DU_j) = & \frac{T_{tp_i}(D_x, DU_j)}{T_{tp_i}(D_x, DU_j) + T_{mIDS}(D_x, DU_j)} \cdot Rec_{CAS}(DU_j) + \\ & + \frac{T_{mIDS}(D_x, DU_j)}{T_{tp_i}(D_x, DU_j) + T_{mIDS}(D_x, DU_j)} \cdot Rec_{mIDS}(D_x, DU_j) \end{aligned} \tag{5.16}$$

5.5.3. Niveles de garantía en los mecanismos de autenticación

Otro de los factores que el CIDN de un dominio de seguridad debe tener en cuenta, es la "fortaleza" de los mecanismos de seguridad que las nuevas unidades de detección pueden utilizar a la hora de proteger los procesos de comunicación y autenticación frente a las amenazas de seguridad bien conocidas: confidencialidad, integridad y autenticidad. Esos mecanismos de seguridad corresponden a las primitivas criptográficas asimétricas, mediante el uso de certificados X.509, que se han presentado en el Capítulo 3.

De manera adicional, estos mecanismos basados en criptografía asimétrica también permiten identificar rápidamente la ejecución de ataques tipo Sybil [248]. Este tipo de ataque corresponde con una de las amenazas de seguridad bastante común en cualquier tipo de sistema de confianza, donde las entidades maliciosas son capaces de instanciar grandes cantidades de entidades falsas, con el objetivo de prestar malos servicios –envío de alertas fraudulentas–, o también enviar buenas recomendaciones entre esas entidades para alcanzar mejores, pero inmerecidos, niveles de confianza.

La fortaleza de los mecanismos de seguridad se puede clasificar según el *Nivel de Garantía* (del inglés Level of Assurance, LoA) durante el proceso de autenticación de la nueva unidad de detección. Este nivel de garantía mide el riesgo que cualquier dominio de seguridad adopta cuando la nueva unidad de detección hace uso de mecanismos más o menos seguros, donde un mayor valor LoA mitigaría un riesgo más alto.

Una primera definición de los niveles LoA fueron propuestos por el Departamento de Defensa (DoD) de Estados Unidos en 2003, definiendo para ello cuatro niveles según el riesgo potencial de un posible error por los procesos de autenticación [249]; a saber: *minimal*, *moderate*, *substantial* y *high assurance*. A partir de esta especificación, el National Institute of Standards and Technology (NIST) propuso un conjunto adicional de guías donde se especifican todos los requisitos técnicos para cada uno de esos cuatro niveles LoA [250]. Cada uno de esos niveles está relacionado con las distintas fases que componen la gestión del ciclo de vida de un certificado X.509.

Los cuatro niveles LoA que han sido propuestos por el NIST, acerca de los niveles de garantía para gestionar los certificados X.509, se han adaptado con un nuevo sentido para el contexto de un sistema colaborativo para la detección de ataques con el siguiente significado, siendo LoA 1 el nivel más bajo y LoA 4 el más alto:

- LoA 1 (*minimal*), donde se utilizan algoritmos criptográficos "débiles", con claves RSA de 512 bits. Este mecanismo de protección se suele aplicar en más del 95 % de los servicios seguros desplegados en Internet.
- LoA 2 (*moderate*). Basado en el nivel anterior, se hacen uso de implementaciones criptográficas con una fortaleza "media", utilizando claves RSA $\geq$ 1024 bits.
- LoA 3 (*substantial*), donde se utilizan implementaciones criptográficas "fuertes". Por ejemplo, a través de *Criptografía de Curva Elíptica* (del inglés Elliptic Curve Cryptography, ECC) con claves $\geq$ 256 bits.
- LoA 4 (*high assurance*). Se utilizan dispositivos hardware criptográficos "duros", como pueden ser las tarjetas inteligentes, donde las credenciales (certificado X.509 y su clave privada) se almacenan de forma segura, necesitando el conocimiento de un secreto, o PIN, para desbloquear el acceso a esas credenciales.

Estos niveles LoA, definidos según (5.17), se utilizan en la siguiente sección durante el cálculo inicial de una nueva unidad de detección DU_j, donde $\tau(DU_j)$ modela el nivel de garantía en la fortaleza de los mecanismos de seguridad de DU_j.

$$\tau(DU_j) = \frac{5 - LoA}{4} \tag{5.17}$$

5.5.4. Cálculo de la reputación inicial de una nueva unidad

Esta sección constituye el núcleo principal donde se presenta, de forma detallada, el cálculo de la reputación inicial para una nueva unidad de detección que desea unirse al CIDN de un dominio de seguridad, con una clara distinción en apartados según el tipo de la unidad de detección: IDS estático, IDS móvil o una nueva relación de confianza con el CIDN de otro dominio de seguridad (todas ellas explicadas en la Sección 5.5.4). En cada uno de esos tres escenarios, se podrá comprobar cómo cada uno de los modelos presentados en las secciones anteriores juegan un papel importante en el proceso del cálculo de la reputación inicial de las nuevas entidades que desean entablar una relación por motivos de colaboración: i) modelo de utilidad para una nueva unidad de detección como IDS, ii) modelo de voluntad de una nueva unidad de detección ante la colaboración y, por último, iii) modelo de similitud entre dos dominios de seguridad, además de las recomendaciones que otros dominios de confianza del CAS puedan ofrecer.

Además de toda la información anterior, destacar que cada modelo también tiene en cuenta la posibilidad de hacer uso de información de comportamiento –reputación– acerca de la nueva unidad de detección en el CIDN donde desea unirse. La "frescura" de esa información será evaluada para determinar si el CIDN se enfrenta, con esa unidad de detección, ante un proceso de *bootstrapping* o de *cold-start*.

Proceso de bootstrapping de un nuevo IDS estático

A la hora de determinar la reputación inicial de un nuevo IDS estático, instalado por uno de los administradores del dominio de seguridad donde se va a desplegar, se consideran los siguientes elementos: la utilidad del IDS estático dentro del contexto del dominio de seguridad donde se está desplegando y el nivel LoA proporcionado por ese IDS estático. La función f_e definida en (5.18) aúna todos los elementos anteriores para el cálculo, en el instante t, de la reputación inicial de un nuevo IDS estático, denotado como $eIDS_j$, cuando desea unirse al CIDN de un dominio de seguridad D_x.

$$Rep_{D_x}^{(t)}(eIDS_j) = f_e(Rep_{D_x}^{\Delta t}(eIDS_j), \Delta t, \Phi_{D_x}(eIDS_j), \tau(eIDS_j)) \tag{5.18}$$

donde Δt representa el tiempo que ha transcurrido desde la última vez que $eIDS_j$ participó en D_x y $Rep_{D_x}^{\Delta t}(eIDS_j)$ el último valor de reputación que tiene D_x acerca de $eIDS_j$. Nótese también que $\Phi_{D_x}(eIDS_j)$ es la utilidad de $eIDS_j$ desde la perspectiva de D_x, definido en (5.8), mientras que $\tau(eIDS_j)$ representa el nivel LoA mostrado por $eIDS_j$ durante su proceso de autenticación en D_x, definido en (5.17).

Antes de presentar la definición completa de f_e, es necesario ver algunas condiciones que debe satisfacer esta función. Esta lista se enumera a continuación.

- Si $\exists Rep^{\Delta t} \wedge \Delta t \to 0$ entonces $f_e(Rep^{\Delta t}, \Delta t, \Phi, \tau) \to Rep^{\Delta t}$

  Si D_x tiene información de reputación sobre $eIDS_j$, $Rep_{D_x}^{\Delta t}(eIDS_j)$, y es bastante reciente en el tiempo ($\Delta t \to 0$), f_e tendría que ser muy similar a esa reputación.

- Si $\nexists Rep^{\Delta t} \vee \Delta t \to \infty$ entonces $f_e(Rep^{\Delta t}, \Delta t, \Phi, \tau) = f'_e(\Phi, \tau)$

  Si D_x no tiene información previa de reputación acerca de $eIDS_j$, o la tiene pero no puede admitirla como válida al ser bastante antigua, f_e se reduce a (f'_e) depender únicamente de la utilidad y nivel LoA de la nueva unidad de detección.

- $f_e(Rep^{\Delta t}, \Delta t, \Phi, \tau) \propto Rep^{\Delta t}$

  La salida de la función f_e debería ser proporcional a la reputación de $eIDS_j$ en D_x; es decir, proporcional a $Rep_{D_x}^{\Delta t}(eIDS_j)$.

- $f'_e(\Phi, \tau) \propto \Phi$

  La salida de la función f'_e debería ser proporcional a la utilidad que puede tener $eIDS_j$ para D_x; es decir, proporcional a $\Phi_{D_x}(eIDS_j)$.

- $f'_e(\Phi, \tau) \propto \tau$

  La salida de la función f'_e debería ser proporcional al nivel LoA que ha presentado $eIDS_j$ al autenticarse en D_x; es decir, proporcional a $\tau(eIDS_j)$.

Después de analizar todas las condiciones para f_e, (5.18) es redefinida por (5.19) para que f_e consiga ajustarse a todos los requisitos analizados más arriba.

$$Rep_{D_x}^{(t)}(eIDS_j) = \left(\frac{1}{\Delta t + 1}\right) \cdot Rep_{D_x}^{\Delta t}(eIDS_j) + \left(\frac{\Delta t}{\Delta t + 1}\right) \cdot \Phi_{D_x}(eIDS_j)^{\tau(eIDS_j)} \tag{5.19}$$

Proceso de bootstrapping de un nuevo IDS móvil

En el caso de los IDSs móviles, pertenecientes a los usuarios finales, se van a requerir (posiblemente) múltiples procesos de bootstrapping para cada uno de los dominios de seguridad a lo largo de su trayectoria de desplazamiento, como se puede contemplar en la Figura 5.4 donde el IDS móvil se mueve entre tres dominios de seguridad diferentes. Pero, al menos en una ocasión inicial, los IDSs móviles van a unirse por primera vez al CIDN de uno de esos dominios, por lo que su líder solamente va a tener a su disposición aquella información que el propio IDS móvil le pueda proporcionar.

Se deben de tener en cuenta los siguientes elementos para el cálculo de la reputación inicial de un nuevo IDS móvil, más allá de la posible existencia de registros que tenga el CIDN del dominio de seguridad con el que se quiere unir: la utilidad del IDS móvil para el dominio de seguridad, su voluntad a colaborar negociando un nuevo conjunto de capacidades de detección y el nivel LoA proporcionado por ese IDS móvil, además de las recomendaciones de otros dominios de seguridad donde ya haya participado y de otros IDSs móviles con los que haya coincidido en algún otro dominio.

La función f_m definida en (5.20) considera todos los elementos anteriores para el cálculo, en el instante t, de la reputación inicial de un nuevo IDS móvil, denotado como $mIDS_j$, cuando desea unirse al CIDN de un dominio de seguridad D_x.

$$\begin{aligned} Rep_{D_x}^{(t)}(mIDS_j) = f_m(Rep_{D_x}^{\Delta t}(mIDS_j), \Delta t, \Phi_{D_x}(mIDS_j), \\ \varpi_{D_x}(mIDS_j), \tau(mIDS_j), Rec_{CAS,mIDS}(D_x, mIDS_j)) \end{aligned} \tag{5.20}$$

donde Δt representa el tiempo transcurrido desde que $mIDS_j$ participó por última vez en D_x; $Rep_{D_x}^{\Delta t}(mIDS_j)$ el último valor de reputación que mantiene D_x sobre $mIDS_j$; $\Phi_{D_x}(mIDS_j)$ la utilidad que tiene $mIDS_j$ desde la perspectiva de D_x, definido en (5.8); y $\tau(mIDS_j)$ el nivel LoA mostrado por $mIDS_j$ durante su proceso de autenticación en D_x, definido en (5.17). Finalmente, $\varpi_{D_x}(mIDS_j)$ es la voluntad a colaborar que tiene $mIDS_j$ para cooperar con D_x, definido en (5.11), y $Rec_{CAS,mIDS}(D_x, mIDS_j)$ el valor de recomendación final sobre $mIDS_j$, proporcionado por otros dominios de seguridad del CAS y por los IDSs móviles colaborando con D_x, como se ha definido en (5.16).

Una vez más, antes de presentar la definición completa de f_m, es necesario analizar algunas condiciones que debe satisfacer esta nueva función; a saber:

- Si $\exists Rep^{\Delta t} \wedge \Delta t \to 0$ entonces $f_m(Rep^{\Delta t}, \Delta t, \Phi, \varpi, \tau, Rec_{CAS,mIDS}) \to Rep^{\Delta t}$

  Si D_x tiene información de reputación sobre $mIDS_j$, $Rep_{D_x}^{\Delta t}(mIDS_j)$, y es reciente en el tiempo ($\Delta t \to 0$), f_m tendría que ser muy similar a esa reputación.

- Si $\nexists Rep^{\Delta t} \vee \Delta t \to \infty$ entonces $f_m(Rep^{\Delta t}, \Delta t, \Phi, \varpi, \tau, Rec_{CAS,mIDS}) = f'_m(\Phi, \varpi, \tau, Rec_{CAS,mIDS})$

  Si D_x no tiene información previa de reputación sobre $mIDS_j$, o la tiene pero no puede admitirla como válida al ser muy antigua, f_m se reduce a (f'_m) depender únicamente de la utilidad, la voluntad ante la colaboración y el nivel LoA de la nueva unidad de detección, así como de las recomendaciones externas que pudiera obtener tanto de otros dominios de seguridad como de otros IDSs móviles.

- $f_m(Rep^{\Delta t}, \Delta t, \Phi, \varpi, \tau, Rec_{CAS,mIDS}) \propto Rep^{\Delta t}$

  La salida de la función f_m debería ser proporcional a la reputación de $mIDS_j$ en D_x; es decir, proporcional a $Rep^{\Delta t}_{D_x}(mIDS_j)$.

- Si $\nexists Rec_{CAS,mIDS}$ entonces $f'_m(\Phi, \varpi, \tau, Rec_{CAS,mIDS}) = f''_m(\Phi, \varpi, \tau)$

  Si no existen recomendaciones sobre la nueva unidad de detección, ni tampoco hay de otros dominios de seguridad o de otros IDSs móviles, su reputación inicial dependerá únicamente de su utilidad, su voluntad ante la colaboración y su nivel LoA. En este caso, D_x se enfrenta a un proceso cold-start.

- $f'_m(\Phi, \varpi, \tau, Rec_{CAS,mIDS}) \propto Rec_{CAS,mIDS}$

  La salida de la función f'_m debería ser proporcional a todas las recomendaciones obtenidas, tanto de otros dominios de seguridad del CAS y de los IDSs móviles que en ese momento están colaborando con el dominio D_x; es decir, tendría que ser proporcional a $Rec_{CAS,mIDS}(D_x, mIDS_j)$.

- Si $\Phi \Uparrow \vee (\Phi \Downarrow \wedge \varpi \Uparrow)$ entonces $f''_m(\Phi, \varpi, \tau) \Uparrow$

  Si la utilidad de $mIDS_j$ es alta, o muy alta, f''_m debería ser proporcionalmente alta también. Sin embargo, si la utilidad inicial de la nueva unidad de detección es baja, o muy baja, pero al mismo tiempo presenta una buena, o muy buena, voluntad a ajustar sus capacidades de detección para mejorar las del CIDN, f''_m también debería ser proporcionalmente alta.

- Si $\Phi \Downarrow \wedge \varpi \Downarrow$ entonces $f''_m(\Phi, \varpi, \tau) \Downarrow$

  Si la utilidad de $mIDS_j$ es baja, o tal vez muy baja, pero además presenta una baja, o muy baja, voluntad a querer mejorar sus propias capacidades internas de detección, f''_m también debería ser proporcionalmente baja.

- $f''_m(\Phi, \varpi, \tau) \propto \tau$

  La salida de la función f''_m debería ser proporcional al nivel de seguridad (LoA) que $mIDS_j$ ha presentado durante su proceso de autenticación en el dominio D_x; es decir, proporcional a $\tau(mIDS_j)$.

Después de analizar las diferentes condiciones para la función f_m, (5.20) es redefinida por (5.21) para que f_m se ajuste en su totalidad a los requisitos que se han enumerado más arriba, dependiendo de si existen recomendaciones del CAS y otros IDSs móviles.

$$Rep^{(t)}_{D_x}(mIDS_j) = \left(\frac{1}{\Delta t + 1}\right) \cdot Rep^{\Delta t}_{D_x}(mIDS_j) + \left(\frac{\Delta t}{\Delta t + 1}\right) \cdot f'_m(\Phi, \varpi, \tau, Rec_{CAS,mIDS}) \tag{5.21}$$

donde $f'_m(\Phi, \varpi, \tau, Rec_{CAS,mIDS})$ se define formalmente como:

$$f'_m = \begin{cases} \Phi_{D_x}(mIDS_j)^{\varpi_{D_x}(mIDS_j)\cdot\tau(mIDS_j)} & \text{si } \nexists Rec_{CAS,mIDS} \\ Rec_{CAS,mIDS}(D_x, mIDS_j)^{1-\Phi_{D_x}(mIDS_j)\cdot\varpi_{D_x}(mIDS_j)\cdot\tau(mIDS_j)} & \text{en otro caso} \end{cases}$$

Proceso de bootstrapping de un nuevo dominio de seguridad

Cuando un dominio de seguridad D_y desea unirse a otro dominio D_x ya existente, D_x tiene que calcular un valor de reputación inicial de D_y antes de realizar la primera interacción. Es decir, tiene que calcular $Rep_{D_x}^{(t)}(D_y)$. El proceso inverso también tendría que ejecutarse de forma paralela, siempre y cuando ambos dominios de seguridad deseen establecer una relación de confianza mutua y bidireccional entre sí.

Para este cálculo, D_x debe tener en cuenta los siguientes elementos, más allá de los posibles valores previos de reputación que D_x tenga almacenados sobre D_y: la similitud en los intereses de detección entre D_x y D_y, el nivel LoA de la nueva entidad y todas las recomendaciones sobre D_y proporcionadas por otros dominios de seguridad del CAS. En (5.22) se muestra la función f_d, que engloba todos los elementos anteriores.

$$Rep_{D_x}^{(t)}(D_y) = f_d(Rep_{D_x}^{\Delta t}(D_y), \Delta t, \lambda(D_x, D_y), \tau(D_y), Rec_{CAS}(D_x, D_y)) \tag{5.22}$$

donde Δt define el tiempo transcurrido desde la última vez que interactuaron D_x y D_y, y $Rep_{D_x}^{\Delta t}(D_y)$ el último valor de reputación que tiene D_x sobre D_y. Por otro lado, $\lambda(D_x, D_y)$ define la similitud existente entre D_x y D_y, definido en (5.12); $\tau(D_y)$ el nivel LoA presentado por D_y durante su proceso de autenticación en D_x, definido en (5.17); y $Rec_{CAS}(D_x, D_y)$ el valor de recomendación acerca de D_y obtenido por D_x a través del mejor camino de confianza por todo el CAS, definido según (5.13).

Al igual que en los casos anteriores, a continuación se analizan todos los requisitos que debe satisfacer f_d antes de presentar su definición completa.

- Si $\exists Rep^{\Delta t} \wedge \Delta t \to 0$ entonces $f_d(Rep^{\Delta t}, \Delta t, \lambda, \tau, Rec_{CAS}) \to Rep^{\Delta t}$

  Si D_x tiene guardada información interna sobre la reputación de D_y, $Rep_{D_x}^{\Delta t}(D_y)$, y esa información es los suficientemente reciente en el tiempo ($\Delta t \to 0$), f_d tendría que ser muy similar a esa reputación.

- Si $\nexists Rep^{\Delta t} \vee \Delta t \to \infty$ entonces $f_d(Rep^{\Delta t}, \Delta t, \lambda, \tau, Rec_{CAS}) = f'_d(\lambda, \tau, Rec_{CAS})$

  Si D_x no guarda en sus registros internos información previa de reputación sobre D_y, o en caso de tenerla no la puede admitir como válida al ser muy antigua, f_d se reduce a (f'_d) depender únicamente de la similitud entre ambos dominios de seguridad y el nivel LoA de la nueva entidad D_y, así como de las recomendaciones obtenidas de otros dominios de seguridad con respecto a D_y.

- $f_d(Rep^{\Delta t}, \Delta t, \lambda, \tau, Rec_{CAS}) \propto Rep^{\Delta t}$

  La salida de la función f_d debería ser proporcional a la reputación de D_y, desde la perspectiva de D_x; es decir, proporcional a $Rep_{D_x}^{\Delta t}(D_y)$.

- Si $\nexists Rec_{CAS}$ entonces $f'_d(\lambda, \tau, Rec_{CAS}) = f''_d(\lambda, \tau)$

  En caso de no existir recomendaciones por parte de otros dominios de seguridad acerca de D_y, la reputación inicial de esta nueva unidad de detección dependerá en exclusiva de la similitud que tenga en las capacidades de detección con D_x y de su nivel LoA. En este caso, D_x se enfrenta a un proceso cold-start.

- $f'_d(\lambda, \tau, Rec_{CAS}) \propto Rec_{CAS}$

  La salida de la función f'_d debería ser proporcional a las recomendaciones recibidas por parte de otros dominios de seguridad del CAS sobre la nueva entidad D_y; es decir, proporcional a $Rec_{CAS}(D_x, D_y)$.

- $f''_d(\lambda, \tau) \propto \lambda$

  La salida de la función f''_d debería ser proporcional a la similitud existente entre los dos dominios de seguridad, D_x y D_y; es decir, proporcional a $\lambda(D_x, D_y)$.

- $f''_d(\lambda, \tau) \propto \tau$

  La salida de la función f''_d debería ser proporcional al nivel LoA que presenta D_y al autenticarse frente a D_x; es decir, proporcional a $\tau(D_y)$.

Al igual que en el caso anterior, una vez analizados todos los requisitos que debe satisfacer f_d, (5.22) es redefinida por (5.23) para que f_d consiga ajustarse en su totalidad a todos los requisitos mencionados anteriormente.

$$Rep^{(t)}_{D_x}(D_y) = \left(\frac{1}{\Delta t + 1}\right) \cdot Rep^{\Delta t}_{D_x}(D_y) + \left(\frac{\Delta t}{\Delta t + 1}\right) \cdot f'_d(\lambda, \tau, Rec_{CAS}) \tag{5.23}$$

donde f'_d se define formalmente como:

$$f'_d(\lambda, \tau, Rec_{CAS}) = (1 - T_{tp_i}(D_x, D_y)) \cdot \lambda(D_x, D_y)^{\tau(D_y)} +$$
$$+ T_{tp_i}(D_x, D_y) \cdot \left(Rec_{CAS} \cdot \lambda(D_x, D_y)^{\tau(D_y)} + (1 - Rec_{CAS}) \cdot \lambda(D_x, D_y)^{\frac{\tau(D_y)}{Rec_{CAS}}}\right)$$

La función f'_d abarca los dos posibles casos cuando un nuevo dominio de seguridad D_y intenta unirse a otro dominio D_x. Por un lado, D_x se enfrenta a un proceso cold-start cuando no hay recomendaciones del CAS sobre D_y, o la confianza en la recomendación recibida es nula; es decir, $T_{tp_i}(D_x, D_y) = 0$. D_x solamente puede calcular la reputación inicial de D_y, para ese caso, basándose únicamente en la información que le proporciona D_y, como la similitud en los intereses de detección de ambos dominios y el nivel de seguridad (LoA) soportado por D_y. Debido a ello, solamente se tiene en consideración la primera parte de la función f'_d. Por otro lado, D_x se enfrenta a un proceso bootstrapping cuando éste recibe recomendaciones acerca de D_y que provienen de otros dominios de confianza del CAS; es decir, cuando $T_{tp_i}(D_x, D_y) > 0$. En este nuevo caso, la reputación inicial de D_y solamente dependerá de la confianza que D_x tiene en el camino construido hasta el dominio de seguridad que le proporciona la recomendación sobre D_y y el valor final sobre esa recomendación, además de tener en cuenta la misma información anterior proporcionada por D_y como si fuera un proceso cold-start.

Finalmente, y como se ha hecho con los otros sistemas de reputación, en la Tabla 5.2 se muestran todas las variables utilizadas a lo largo de las dos secciones anteriores para el cálculo de la reputación inicial de una nueva unidad de detección. Ahí se incluye una breve descripción de cada una, así como sus valores de inicialización y el cálculo según qué entidad es la encargada de su mantenimiento y configuración.

Variable	Descripción	Inicialización/cálculo
Ecuación (5.8)		
$\Phi_x(DU_j)$	Valor de utilidad de DU_j para el dominio de seguridad D_x	Calculada en (5.8) y usada en (5.19) y (5.21)
Ecuación (5.9)		
$\psi_x(ds_{i_k})$	Dispersión y distribución de la reputación de los IDSs	Calculada en (5.9) y usada en (5.8)
$Rep_x(DU_{ds_{i_k}})$	Reputación de IDSs con ds_{i_k}	También usada en (5.10)
Ecuación (5.10)		
$\varphi(\{Rep_x(DU_{ds_{i_k}})\})$	Medida de dispersión para la distribución de $Rep_x(DU_{ds_{i_k}})$	Calculada en (5.10) y usada en (5.9)
Ecuación (5.11)		
$\varpi_x(DU_j)$	Voluntad que tiene DU_j a la hora de colaborar en D_x	Calculada en (5.11) y usada en (5.21)
Ecuación (5.12)		
$\lambda(D_x, D_y)$	Similitud en las capacidades de detección de D_x y D_y	Calculada en (5.12) y usada en (5.23)
Ecuación (5.13)		
$T_{tp_i}(D_x, D_y)$	Confianza en tp_i para D_x con recomendaciones de D_y	Calculada en (5.13) y usada en (5.16)
ϱ_j	Penalización en el salto entre dominios administrativos	Administrador del dominio administrativo con D_j
Ecuación (5.14)		
$Rec_{mIDS}(D_x, mIDS_j)$	Recomendación de los IDSs móviles en D_x sobre $mIDS_j$	Calculada en (5.14) y usada en (5.16)
Ecuación (5.15)		
$T_{mIDS}(D_x, mIDS_j)$	Valor de confianza sobre las recomendaciones de $mIDS_j$	Calculada en (5.15) y usada en (5.16)
Ecuación (5.16)		
$Rec_{CAS,mIDS}(D_x, DU_j)$	Recomendación de DU_j para D_x del CAS e IDSs móviles	Calculada en (5.16) y usada en (5.21)
Ecuación (5.17)		
$\tau(DU_j)$	Nivel de seguridad (LoA) de DU_j en su autenticación	Calculada en (5.17) y usada en (5.19), (5.21) y (5.23)
Ecuación (5.19), (5.21) y (5.23)		
$Rep_{D_x}^{(t)}(DU_j)$	Reputación inicial de DU_j en D_x (nueva unidad detección)	Según tipo de unidad: IDS estático o móvil, o dominio

Tabla 5.2: Variables del sistema de reputación para nuevas unidades de detección

5.6. Perfil de comunicaciones para nuevas entidades

Esta sección desarrolla el perfil de comunicaciones para un correcto funcionamiento del sistema de confianza basado en reputación presentado en las secciones anteriores, con el objetivo de calcular la reputación inicial de una nueva unidad de detección que quiera unirse al CIDN de un dominio de seguridad, para colaborar con él a través del envío de las alertas que puede detectar en su área de detección. En este perfil, se hace hincapié en las interacciones realizadas por la nueva unidad de detección según el tipo que represente: IDS estático, IDS móvil o dominio de seguridad. Apuntar que sólo los IDSs móviles suelen tener interacciones de corto plazo con un CIDN, mientras que el resto van, normalmente, a participar por un periodo de tiempo más largo.

La Figura 5.7 muestra un diagrama de secuencia UML con todas las interacciones que la j-ésima nueva entidad de detección, denotada como DU_j, tiene que hacer cuando desea unirse al CIDN del dominio de seguridad D_x. El resto de esta sección presenta en detalle cada una de esas interacciones, donde la distinción en el tipo de DU_j se realiza durante su proceso de autenticación en el Paso 1 frente al líder del Comité de Sabios (WC), denotado como WCL_x, como máximo representante del CIDN de D_x.

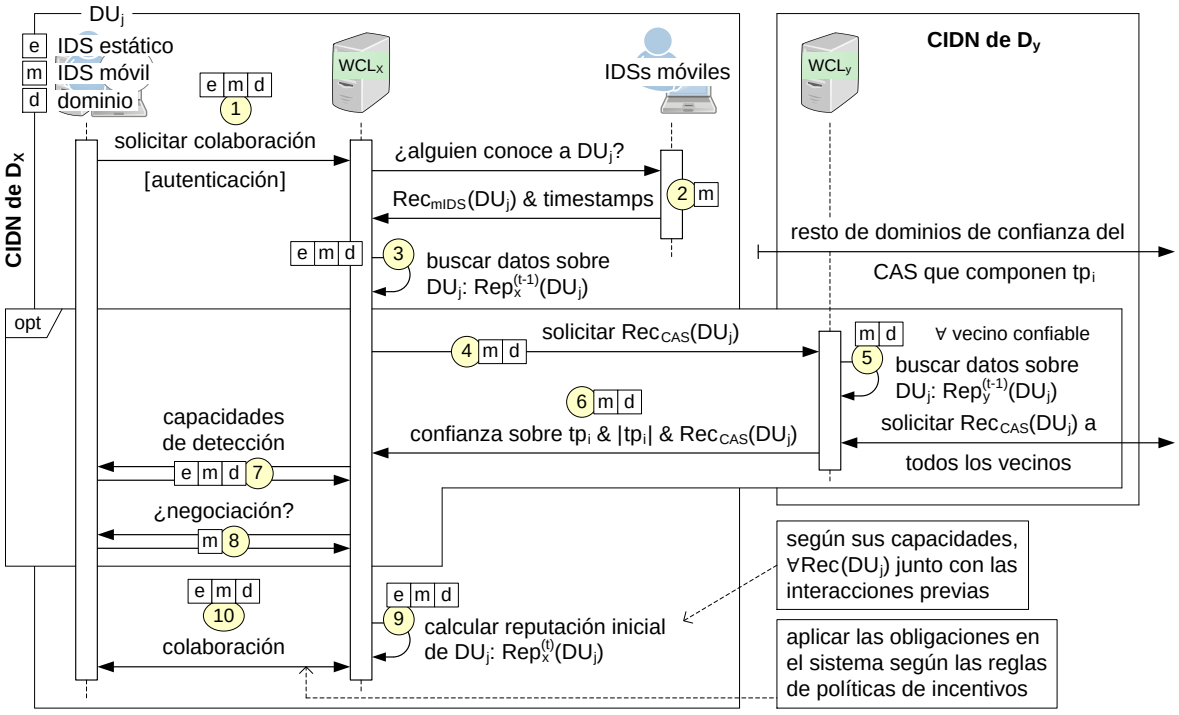


Figura 5.7: Diagrama de secuencia UML para una nueva unidad de detección

Notar que en este perfil de comunicaciones, $Rec(DU_j)$ representa a todas aquellas recomendaciones que otras entidades le pueden proporcionar a WCL_x, con el fin último de poder calcular el valor de reputación inicial de DU_j.

Paso 1: solicitar colaboración y, opcionalmente, autenticar a DU_j

Bajo el primer punto, DU_j se pone en contacto con WCL_x (líder del dominio de seguridad al cuál quiere unirse) para autenticarse como unidad de detección legítima. Este proceso de autenticación es opcional para todos los IDSs móviles, mientras que es obligatorio tanto para los IDSs estáticos como para los dominios de seguridad.

La identificación del tipo de unidad de detección se basa en si DU_j se autentica o no en el dominio de seguridad. Si no se autentica, WCL_x considera automáticamente que DU_j es un IDS móvil. En otro caso, esta decisión se basa en las credenciales que DU_j presenta durante su autenticación. En este caso, DU_j se considera un IDS estático si sus credenciales pertenecen a un administrador de D_x, o bien el líder del CIDN de otro dominio de seguridad cuando sus credenciales no pertenecen a la PKI desplegada por D_x, donde se está autenticando, y sí que pertenecen a otro dominio de seguridad con el que D_x quiere establecer una relación de colaboración. Por último, DU_j se considera que es un IDS móvil al no encajar sus credenciales en ninguno de los casos anteriores. Merece la pena comentar que los IDSs móviles se pueden autenticar en D_x, aunque sea opcional, lo cual incluso les puede proporcionar una reputación inicial más alta como se define en (5.21), según el parámetro $\tau(DU_j)$ descrito en (5.17).

La validación de las credenciales que presente DU_j (su certificado X.509) la puede obtener WCL_x delegando este proceso al Servicio de Validación de la Sección 3.2, y que D_x debe tener desplegado en su infraestructura interna de certificación.

Paso 2: solicitar opiniones a todos los IDSs móviles sobre DU_j

WCL_x requiere a los IDSs móviles que están actualmente participando con su CIDN si alguno de ellos tiene información sobre DU_j, al haber coincidido con él en algún otro dominio de seguridad, por ejemplo. Si es así, esos IDSs móviles pueden ofrecer a WCL_x recomendaciones procedentes de experiencias previas con otros dominios inalcanzables para D_x, definidas en (5.14) como $Rec_{mIDS}(D_x, mIDS_j)$, teniendo también en cuenta la confianza sobre esas recomendación, según (5.15), para aceptarlas como válidas. Es obvio que este paso solamente tiene sentido cuando DU_j es un IDS móvil, ya que sólo este tipo de entidad pueden coincidir en otros dominios externos al actual.

Cada IDS móvil que pueda ofrecer esta información, también debe proporcionar un sello de tiempo que indique cuándo ocurrió la última interacción realizada por DU_j con el dominio de seguridad externo. Es decir, cuándo publicó la última alerta. Este valor de tiempo ayudará a WCL_x a decidir la validez de esta opinión en el tiempo.

Paso 3: ¿buscar datos para calcular $Rep_x(DU_j)$, reputación inicial de DU_j?

De forma paralela al paso anterior, WCL_x comprueba si DU_j ha colaborado con el CIDN de D_x anteriormente. Si es así, WCL_x mantendrá el último valor de reputación de DU_j, es decir, $Rep_x^{(t-1)}(DU_j)$, al ser una entidad conocida para D_x. En caso de existir un registro sobre DU_j, WCL_x comprueba si $Rep_x^{(t-1)}(DU_j)$ todavía puede considerarse como un valor de reputación válido para el dominio donde desea unirse.

Si el valor de reputación que tiene WCL_x sobre DU_j todavía se puede considerar válido, WCL_x pasará directamente o al Paso 7, para solicitar a DU_j sus capacidades de detección, o al Paso 10 para comenzar la colaboración entre la unidad de detección recién llegada y el dominio de seguridad al que desea unirse. En caso contrario, o DU_j es una entidad totalmente desconocida para D_x o, aun siendo una entidad conocida para él, su valor de reputación almacenado es demasiado antiguo para ser considerado como válido. Para ambos casos, WCL_x continúa con el Paso 4 para obtener posibles recomendaciones que otros dominios de seguridad pudieran tener sobre DU_j.

El mecanismo presentado en este paso permite reducir al máximo la sobrecarga de tráfico en la red que supondría la petición de todas las posibles recomendaciones que el resto de dominios de seguridad del CAS tienen sobre DU_j. En cualquier caso, este paso es obligatorio independientemente del tipo de unidad de detección, ya que cualquiera podría dejar y volver a unirse al CIDN en un breve espacio de tiempo.

Paso 4: solicitar opiniones a los dominios vecinos de confianza sobre DU_j

Si WCL_x no tiene en sus registros opiniones sobre DU_j, porque no ha participado hasta ahora con D_x o porque esa información es demasiado antigua para considerarla válida, éste envía una *solicitud de opinión* a los líderes de sus dominios de seguridad vecinos con los que D_x mantiene una relación directa de confianza. En el ejemplo de la Figura 5.4, WCL_A enviará esta solicitud tanto a WCL_C como a WCL_E, mientras que WCL_I lo hará a WCL_B, perteneciente a otro dominio administrativo, y localmente a WCL_H como miembro de su misma unidad organizativa. Este proceso se propaga por todo el CAS hasta alcanzar el TTL establecido por sus administradores.

Paso 5: ¿buscar datos sobre DU_j y pedir opiniones a otros dominios?

Cada líder de los CIDNs vecinos revisa si DU_j ha interactuado con su dominio de seguridad en el pasado a fin de recuperar su último valor de reputación $Rep_y^{(t-1)}(DU_j)$. Esta comprobación coincide con el Paso 3, aunque aquí cada dominio hará uso de su propia base de conocimiento. Nótese que en la Figura 5.7 solamente se detalla WCL_y por motivos de espacio, aunque igualmente se tiene que repetir en el resto de dominios vecinos de D_x. Como en el paso anterior, la solicitud de opinión se propaga al resto de dominios con los que D_y mantenga una relación de confianza directa.

Paso 6: calcular $Rec_{CAS}(DU_j)$ agregando las opiniones recibidas sobre DU_j

Cada líder en el camino de confianza agregará las respuestas de recomendación, si hay alguna, recibidas de todos sus dominios de confianza vecinos. En este proceso de agregación, se debe tener en cuenta la confianza que el dominio solicitante tiene en los dominios vecinos que le envían sus recomendaciones, mediante el cálculo de $T_{tp_i}(D_x, D_y)$ definido en (5.13). Este valor de confianza permite conocer qué nivel de veracidad tiene el dominio solicitante en la recomendación recibida del resto de dominios. Cuanto mayor sea, mayor será la confianza en que el valor $Rec_{CAS}(DU_j)$ es cierto.

La respuesta de recomendación agregada se le envía al dominio que hizo la solicitud, hasta que se alcance el dominio origen que realizó la solicitud de opinión original. En el diagrama de la Figura 5.7, las respuestas de recomendación serán agregadas y devueltas hasta que lleguen a WCL_x. De esta manera, se preserva la privacidad en la confianza que cada dominio obtiene de sus vecinos. Es decir, los dominios a lo largo del camino de confianza obtendrán respuestas agregadas de recomendación sin conocer qué opiniones han obtenido de sus dominios de seguridad vecinos.

El bloque anterior de pasos (4, 5 y 6) se ejecuta cuando DU_j es un IDS móvil o un dominio de seguridad, no siendo obligatorio para IDSs estáticos ya que ningún otro dominio va a tener información sobre él. Destacar que el Paso 2 y el Paso 3 se pueden llevar a cabo junto con el 4, 5 y 6 a fin de maximizar el rendimiento total.

Paso 7: obtener las capacidades de detección de DU_j

Además de las recomendaciones que WCL_x pudiera recibir desde otras entidades, el valor de reputación inicial de DU_j dependerá de las capacidades de detección que éste le pueda ofrecer a su CIDN con las que mejorar la precisión en los procesos de detección. Con esas capacidades de detección, WCL_x será capaz de calcular o la *utilidad* de DU_j si es un IDS estático o móvil, definido en (5.7) como $\Phi_x(DU_j)$, o la *similitud* entre el dominio de WCL_x, D_x, y el nuevo dominio de seguridad con el que desea establecer una relación directa de confianza, definido en (5.12) como $\lambda(D_x, D_y)$.

Paso 8: ¿negociar con DU_j ciertos ajustes en sus capacidades de detección?

Este paso sólo se ejecuta si DU_j es un IDS móvil, ya que para el resto de entidades no tendría sentido alguno. Dependiendo de $\Phi_x(DU_j)$, calculado en el Paso 7, WCL_x podría proponer a DU_j algunos pequeños cambios en sus capacidades de detección, como se ha detallado en la Sección 5.5.1 como modelo de voluntad frente a la colaboración para el cálculo de $\varpi_x(DU_j)$, definido en (5.11). La propuesta de estos cambios, mediante la activación o desactivación de alguna de sus capacidades de detección, la realiza WCL_x según las amenazas en las que su CIDN esté interesado en detectar.

Paso 9: calcular $Rep_x(DU_j)$, reputación inicial de DU_j

En el cálculo de la reputación inicial de DU_j, denotado como $Rep_x(DU_j)$, se utiliza toda la información obtenida en los pasos anteriores, y que es necesaria para los cálculos definidos en (5.19), (5.21) o (5.23) según el tipo de entidad que representa DU_j: si es un IDS estático, un IDS móvil o un dominio de seguridad, respectivamente.

Paso 10: comenzar la colaboración después de aplicar ciertas obligaciones

WCL_x puede informar a los componentes de su red (por ejemplo, firewalls o puntos de acceso) que apliquen una serie de políticas de incentivos para DU_j como, por ejemplo, reglas de acceso a la red personalizadas para esa nueva unidad de detección.

5.7. Resultados experimentales

En esta sección, se presentan una serie de experimentos con los que poder demostrar los beneficios que confiere el cálculo de la reputación inicial sobre nuevas unidades de detección para la precisión y la cobertura en la detección de ataques distribuidos. En concreto, el objetivo principal detrás de estos experimentos es doble:

1) Confirmar los beneficios en la incorporación de las alertas generadas por los IDSs móviles en los procesos de detección de un CIDN. Con esta nueva información, el CIDN vería incrementada su cobertura en la detección de ataques.

2) Analizar cómo las recomendaciones ofrecidas por terceros dominios de confianza pueden ayudar a identificar unidades de detección con actitudes maliciosas antes de que interactúen con un CIDN, aunque éste no posea información sobre ellas.

Los experimentos que a continuación se analizan se han realizado sobre un entorno multidominio de simulación, donde se ha desplegado la implementación de un sistema colaborativo de alertas (CAS), con todos los componentes presentados en la Sección 5.4, así como el sistema de reputación propuesto en la Sección 5.5 con el que poder evaluar la reputación inicial de una nueva unidad de detección. En concreto, se ha implementado un CAS con tres dominios administrativos donde se han desplegado de forma equitativa: 15 dominios de seguridad con sus correspondientes CIDNs, 10 IDSs estáticos por CIDN (teniendo 150 en total), 300 IDSs móviles que pueden colaborar con los distintos CIDNs a lo largo de su trayectoria de movimiento y 10 capacidades de detección.

A fin de poder medir la satisfacción sobre cualquier alerta publicada por una unidad de detección, necesaria para determinar su comportamiento como honesto o malicioso, a continuación se define la satisfacción tenida en cuenta en los experimentos analizados en los siguientes apartados. En concreto, el nivel de satisfacción definido en (4.4), para el modelo de reputación intradominio, se ha reformulado para este caso por el mostrado en (5.24), con el que calcular la satisfacción que un dominio D_x puede depositar sobre una alerta publicada, en el instante t, por una unidad de detección DU_j.

$$Sat_{xj}^{(t)} = 0{,}5 + Fancy_x^{(t)} \cdot \frac{ds_{i_k}(prioridad)}{2} \tag{5.24}$$

donde $Fancy_x^{(t)} \in [-1, 1]$ es la suposición que tiene el dominio D_x en la veracidad de una alerta generada por DU_j y $ds_{i_k}(prioridad) \in [0, 1]$ la prioridad de la capacidad de detección k que necesita D_x, la cual está relacionada con el tipo de ataque causante de haber generado dicha alerta. Para el cálculo de $Fancy_x^{(t)}$, se ha adoptado un esquema de votación basado en la mayoría como el presentado en (4.5), donde se tiene en cuenta la reputación de todas las unidades de detección que intervienen en este proceso.

Comparando ambas ecuaciones, (4.4) y (5.24), se puede comprobar la similitud entre las dos, pero adaptando, en el último caso, la nomenclatura de conceptos utilizado en este capítulo. La principal diferencia es que en (4.4) se utiliza la severidad de la alerta emitida por DU_j, mientras que en (5.24) se hace uso de la prioridad que tiene la regla de detección configurada en la capacidad de detección ds_{i_k} (Definición 7).

5.7.1. Evaluación de la cobertura de la detección

El objetivo para este primer experimento es evaluar la variación que puede sufrir la cobertura de la detección de cualquier CIDN, considerando, en primer lugar, solamente las alertas que proporcionan sus IDS estáticos y, posteriormente, incluyendo también las diferentes alertas que pueden proporcionar los distintos IDSs móviles. En la Figura 5.8 se muestran todos los resultados para este experimento, los cuales se han adquirido en determinados momentos de la simulación después de que cada uno de los componentes en el CAS funcionara aleatoriamente durante un tiempo aceptable, capturando en esta figura la cobertura de la detección que en ese momento presentaba el CAS.

Cada fila en las imágenes de la Figura 5.8 hace referencia a una de las capacidades de detección demandadas por el CAS, y cada columna con un determinado CIDN. Cada celda (i, j) muestra una intensidad de color proporcional a la cantidad de políticas de seguridad de la capacidad de detección i cubierta por todos los IDSs colaborando con el CIDN j, ponderado con los valores de reputación de cada uno de esos IDSs. Escalas de grises más oscuros indican una alta cobertura de la detección, y viceversa, mientras que las celdas que están marcadas con un símbolo $\times$ hacen notar que el CIDN no necesita protección frente a las amenazas que representa esa capacidad de detección.

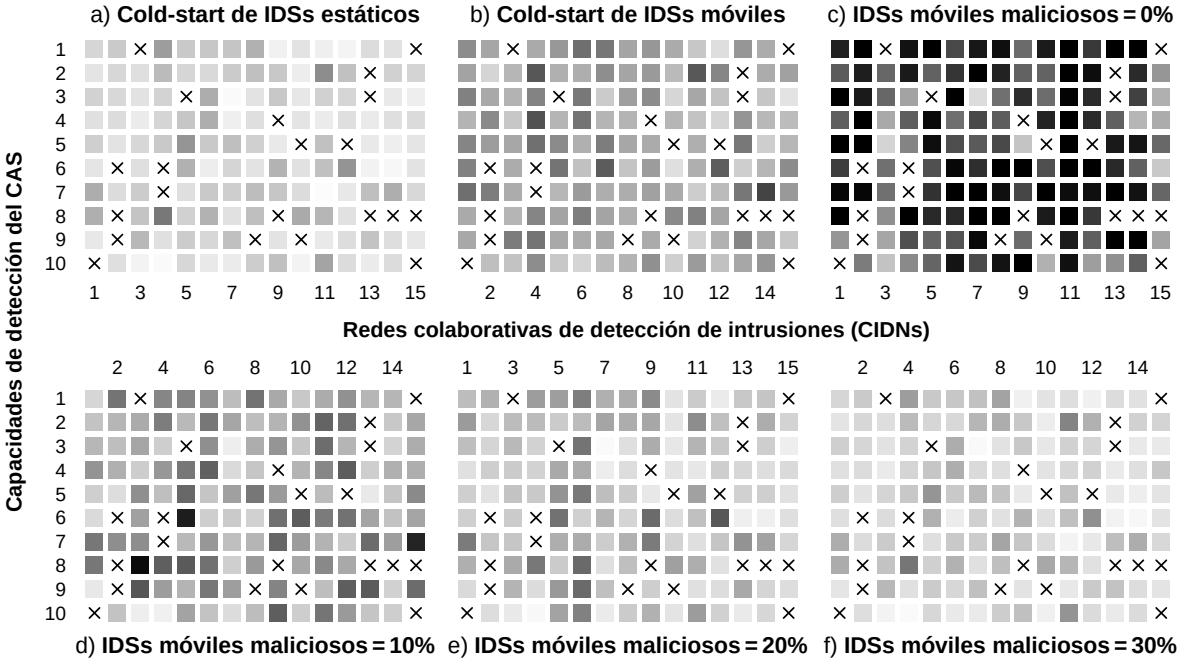


Figura 5.8: Coberturas de la detección después de ejecutar los procesos de cold-start y bootstrapping según las capacidades de detección de los IDSs

Los resultados en la Figura 5.8a atañen a la cobertura de la detección, acto seguido de desplegar en el CAS los IDSs estáticos por primera vez. Eso significa que cada CIDN se ha enfrentado al problema cold-start mediante (5.19). Estos resultados alcanzan una cobertura global de la detección del 17,46 % a nivel del CAS.

Sobre la configuración del CAS, se han desplegado de forma aleatoria todos los IDSs móviles entre los CIDNs, obteniendo una nueva y mejor cobertura de la detección (ver Figura 5.8b). Esos resultados se han obtenido después de que cada CIDN se enfrente al problema cold-start, similar al anterior, pero ahora para los IDSs móviles usando (5.21). Con esta nueva configuración se logra una mejor cobertura de la detección del 36,38 %, algo más del doble que cuando sólo habían IDSs estáticos. Por tanto, se puede observar que los IDSs móviles son capaces de, inicialmente, mejorar las capacidades de detección que necesita cualquier CIDN en su afán de detectar ataques distribuidos.

Las siguientes pruebas en este experimento se centran en determinar cómo funciona el CAS con actitudes benévolas para el comportamiento de sus unidades de detección. Es decir, con los IDSs estáticos generando alertas sobre hechos ocurridos en la realidad y los IDSs móviles colaborando en las tareas de detección mientras se desplazan entre los dominios de seguridad. Además, también se pretende analizar cómo de robusto es el sistema de reputación frente al problema bootstrapping, con un porcentaje determinado de IDSs móviles que se comportan de forma maliciosa. En la Figura 5.8c-f se muestran los resultados sobre la cobertura de la detección, variando el porcentaje de todos esos IDSs móviles maliciosos hasta un 30 % del total. Los resultados para cada prueba se han obtenido después de inyectar 50 tipos de ataques, provocando así que se vayan a generar sus correspondientes alertas, y simulando, al mismo tiempo, que esos IDSs móviles se desplazan de un dominio de seguridad a otro. Estos movimientos provocan que se lance la ejecución del proceso necesario para la gestión del problema bootstrapping en cada uno de los CIDNs, donde desean unirse a colaborar esos IDSs móviles.

Analizando como primer punto los resultados de la Figura 5.8c, la cobertura de la detección alcanza un 75,74 %, en promedio, cuando sólo se tienen en cuenta los IDSs móviles con una total honestidad en la generación de sus alertas. Esta mejoría en el CAS se debe a que todos los IDSs móviles han ido mejorando sus valores de reputación al colaborar de manera honesta con su CIDN, siendo "recompensados" por ello a nivel de reputación. Sin embargo, esos valores de reputación irán siendo cada vez menores conforme vayan apareciendo actitudes cada vez más maliciosas en sus comportamientos, decayendo progresivamente la cobertura global de la detección al unísono.

Conforme los IDSs móviles van teniendo actitudes cada vez más maliciosas, existe un punto donde la cobertura de la detección se parece a la primera configuración de la Figura 5.8a, cuando sólo habían IDSs estáticos en el CAS. Este caso se puede ver en la Figura 5.8f, con un 30 % de los IDSs móviles maliciosos. Este hecho denota que los IDSs móviles son aislados gradualmente conforme sus comportamientos van empeorando en el tiempo, llegando a un punto en que el CAS "solamente" opera con las alertas enviadas por los IDSs estáticos y por otros dominios de confianza. Por tanto, se puede afirmar que el sistema de reputación de la Sección 5.5 puede soportar sobre un 20 % de IDSs móviles con comportamientos maliciosos, antes de descartarlos como útiles en la detección de ataques. También se ha comprobado cómo la cobertura de la detección crece del 17,46 % (sin IDS móviles) al 36,38 % cuando se admiten alertas de ese tipo de IDSs, llegando al 75,74 % cuando tanto los IDSs móviles como los estáticos manifiestan una total honestidad (Figura 5.8c) al generar sus alertas.

5.7.2. Valoración de los IDSs estáticos y móviles al variar sus estados de comportamiento

La pretensión de este segundo experimento es analizar cómo varía la reputación de los IDSs, ya sean estáticos o móviles, conforme sus comportamientos van decayendo en el tiempo. Estos dos tipos de IDSs se enfrentan ante los procesos de bootstrapping de la Sección 5.5.4, antes de comenzar la colaboración con alguno de los CIDNs, teniendo en cuenta para ello distintos porcentajes de IDSs con comportamientos maliciosos.

Después de los procesos iniciales de bootstrapping, todos los IDSs generan las alertas correspondientes a la inyección sobre el CAS de 50 tipos de ataques. Destacar también que los IDSs móviles serán reubicados entre diferentes dominios de seguridad de una prueba a otra, simulando trayectorias de movimiento con el paso del tiempo.

La Figura 5.9 muestra los resultados de este experimento.

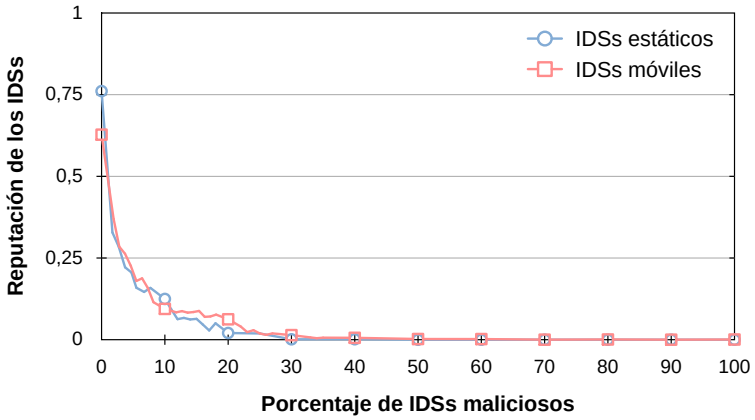


Figura 5.9: Variación de la reputación al aumentar el número de IDSs maliciosos

Como se puede observar en esta figura, los valores de reputación de los IDSs decrecen rápidamente en las primeras etapas cuando hay menos de un 5 % de los IDSs maliciosos. Este hecho es interesante desde la perspectiva de los IDSs móviles, ya que éstos siguen un patrón bastante similar al de los IDSs estáticos aunque sus valores de reputación los calculan los CIDNs para cada uno de los movimientos entre sus dominios de seguridad. Gracias al uso de las recomendaciones proporcionadas por otros dominios de confianza del CAS, los líderes de sus CIDNs siempre van a tener una visión corporativa sobre los IDSs móviles, aunque nunca hayan colaborado con el CIDN al que desean unirse.

Merece la pena mencionar en este punto que los experimentos que se han analizado anteriormente han mantenido siempre, de forma constante, la misma confianza entre los distintos dominios de seguridad, así como sus comportamientos a la hora de compartir alertas entre ellos. Por tanto, en una última prueba, cuyos resultados se muestran en la Figura 5.10, se pretende variar el porcentaje de CIDNs maliciosos entre el 5, 10, 15 y el 20 %, siguiendo la misma configuración y condiciones experimentales que en la prueba anterior de este experimento. Véase entonces que la Figura 5.9 corresponde, según esta última prueba, al caso cuando no existen CIDNs maliciosos en el CAS.

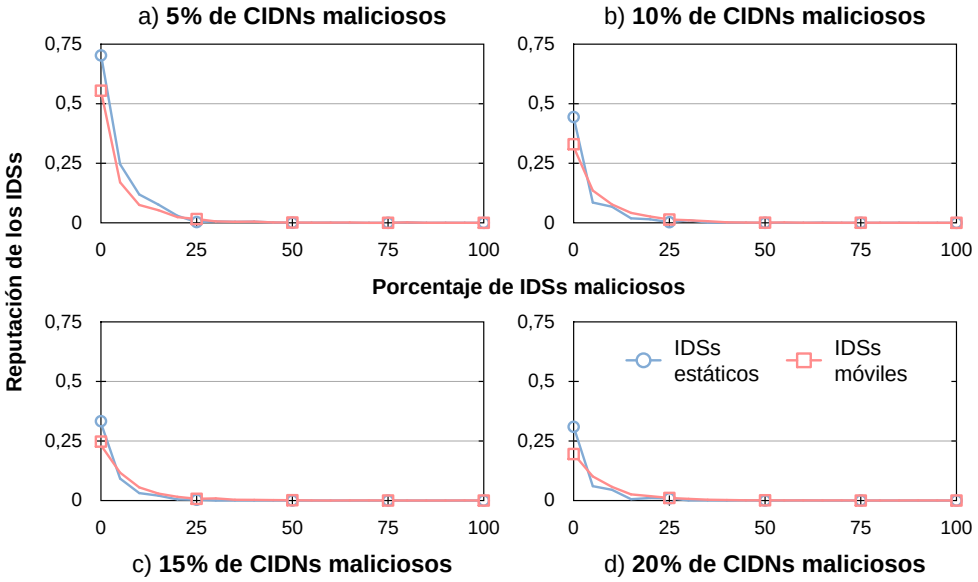


Figura 5.10: Variación de la reputación cuando todas las entidades del CAS tienen un comportamiento malicioso

Los resultados muestran una tendencia similar que en las pruebas anteriores de este experimento, aunque ahora es más acentuada la pérdida de los valores de reputación de manera más rápida. No obstante, el sistema de reputación propuesto frente al problema bootstrapping sigue siendo suficientemente robusto hasta alcanzar alrededor de un 20 % de IDSs maliciosos, sin perder precisión en los procesos de detección.

Como principal conclusión después de considerar los dos experimentos anteriores, se puede afirmar que la inclusión de alertas provenientes de IDSs móviles puede mejorar, sustancialmente, la cobertura de la detección que tienen los CIDNs sobre los activos que están protegiendo usando esas unidades de detección. Además, los resultados de la experimentación anterior también han demostrado que el comportamiento de los IDSs móviles no afecta negativamente al correcto funcionamiento del CAS, pudiendo llegar a soportar actitudes maliciosas por parte de los IDSs móviles de hasta un 20 % sin que se vea comprometida la detección del resto de componentes del CAS.

En escenarios con una peor configuración, cuando los CIDNs del CAS albergan más del 20 % de los IDSs móviles con actitudes maliciosas, se ha podido demostrar que en la cobertura de la detección, en ese caso, solamente influyen los IDSs estáticos desplegados en la propia infraestructura, instalados por sus administradores. Los IDSs móviles son aislados de los procesos de detección sin que los CIDNs del CAS tengan en consideración las alertas que llegaran a generar. A partir de ese momento, el CIDN de cada dominio de seguridad en el CAS incorporará, en los procesos de detección de sus componentes, las alertas que puedan proporcionar sus propios IDSs estáticos y las que provengan de otros dominios de seguridad externos con los que mantenga una relación de confianza a nivel multidominio, ya sean relaciones de confianza directas o indirectas.

5.8. Conclusiones del capítulo

La gestión de la confianza dentro de un sistema colaborativo de alertas (CAS), con el objetivo principal de identificar comportamientos maliciosos en la publicación de alertas por parte de sus distintas redes colaborativas de detección de intrusiones (CIDN), se ha modelado al comienzo de este capítulo tomando como base el mecanismo de confianza basado en reputación presentado en el capítulo anterior. Pero, en esta ocasión, teniendo en mente que se trata de un modelo de confianza con un ámbito global de aplicación. A través de este mecanismo, el CAS es capaz de construir una base global de conocimiento con las alertas generadas por los IDSs desplegados de forma estratégica en cada CIDN, pero adoptando exclusivamente aquellas alertas con un suficiente nivel de confianza que demuestre la veracidad del hecho acontecido en la red de detección, según la reputación que han tenido esos CIDNs con respecto a sus actos en el pasado.

Con la base global de conocimiento construida entre los distintos CIDNs, y una vez minimizado el impacto de las alertas fraudulentas con el sistema interdominio basado en reputación presentado en este capítulo, el CAS va a ser capaz de detectar ataques con una aplicación distribuida en sus procesos de ejecución. Como resultado final, cualquier CAS dispone de un mecanismo de confianza basado en reputación con el que mejorar la precisión, exactitud y cobertura en la detección de los ataques distribuidos.

Para el diseño del sistema de reputación interdominio, se ha tomado como base un modelo de confianza adaptativo bien conocido como es PeerTrust, pero definiendo los parámetros específicos de la métrica genérica que define PeerTrust para tener en cuenta cualquier sistema colaborativo para la detección de ataques distribuidos. En este caso, en el cálculo de esos parámetros para modelar el comportamiento de los CIDNs se han definido, formalmente, los conceptos de satisfacción y credibilidad sobre las alertas que son publicadas por otro CIDN. En ambos conceptos se consideran las opiniones que tienen el resto de CIDNs que también han recibido las alertas, sus valores de confianza, así como el valor promedio de reputación de todos los IDSs que también han detectado la alerta en el CIDN que está realizando el cálculo. Además, también se han definido los dos parámetros que representan los factores de contexto presentados en PeerTrust, según la importancia de las transacciones y el contexto de la comunidad. En las pruebas experimentales realizadas para la validación del sistema de reputación interdominio, en un entorno de simulación de un CAS con un alto número de CIDNs, se ha demostrado que el ratio de detección de alertas fraudulentas en un escenario multidominio es capaz de alcanzar sobre un 90 % de precisión en su clasificación.

En un segundo bloque posterior dentro de este capítulo, y siempre focalizado en un entorno multidominio, también se ha presentado un mecanismo basado en reputación con el que un CIDN, o el CAS a nivel global, pueda calcular la confianza inicial sobre una nueva entidad con capacidades de detección, la cual desea unirse a colaborar con el sistema proporcionando las alertas que es capaz de generar. Entidades que pueden ser *IDSs estáticos* de la infraestructura, instalados por los administradores; *IDSs móviles* de los usuarios finales, con altas capacidades de movilidad; o *dominios de seguridad*, que pretenden colaborar entre sí para habilitar la detección de ataques distribuidos.

El modelo de reputación propuesto permite realizar el cálculo inicial de la confianza de cualquiera de las tres unidades de detección anteriores, dando solución a dos desafíos bien conocidos en cualquier tipo de sistema colaborativo: el problema *cold-start*, donde las unidades de detección son totalmente desconocidas, al ser la primera vez que se van a unir al CAS, y el problema *bootstrapping*, donde estas unidades de detección ya han colaborado previamente en alguno de los CIDNs del CAS, por lo que ya no son del todo desconocidos. Para este cálculo, se han propuesto tres grandes fuentes de información a considerar: las capacidades de detección de la nueva unidad de detección, los niveles de garantía (fortaleza) de sus mecanismos de autenticación y las posibles recomendaciones que otros CIDNs pudieran proporcionar acerca de esa nueva unidad. Estas tres fuentes de información tendrán un significado bien distinto dependiendo de qué tipo de unidad de detección se esté evaluando, de entre las tres anteriores, y ante qué problema de los dos anteriores (cold-start o bootstrapping) se esté haciendo referencia.

Además de las alertas que son generadas por los IDSs estáticos de la infraestructura, y el intercambio de las mismas a nivel interdominio, se ha demostrado a través de una serie de resultados experimentales que la adopción de las alertas generadas por todos los IDSs móviles permiten obtener una mejor cobertura sobre la detección. Esa cobertura, como mínimo, será la misma que ofrecen los IDSs estáticos de la infraestructura aun cuando el número de IDSs móviles con comportamientos maliciosos sea cada vez mayor. Las alertas que pudieran proporcionar esos IDSs maliciosos se irían descartando de los procesos de detección del resto de las unidades de detección, llegando a un punto en el que no se tomarían en cuenta sus alertas fraudulentas y la cobertura en la detección de ataques solamente estaría cubierta por los IDSs estáticos.

Capítulo 6

Reducir la incertidumbre en un sistema colaborativo de alertas

La actualización sobre la reputación de los IDSs de la infraestructura en los capítulos anteriores, denominados IDSs estáticos, puede conducir a que existan zonas en la red de detección que las estén monitorizando IDSs con una reputación más que dudosa en sus comportamientos. La toma de decisiones sobre la detección de ataques no puede estar basada haciendo caso de IDSs poco confiables para su red colaborativa de detección de intrusiones (CIDN) o, en su defecto, para el sistema colaborativo de alertas (CAS).

Con esa premisa, este capítulo presenta un mecanismo adaptativo que maximice en cada momento la calidad de las alertas generadas por los IDSs de la infraestructura para aumentar la robustez del CAS frente a comportamientos maliciosos. A este respecto se introduce el concepto *Diversidad de la Confianza* (del inglés Trust Diversity, TD): IDSs con reputaciones dispares trabajando conjuntamente en detectar ataques. Con una baja diversidad de la confianza en los IDSs –valores de reputación muy parecidos– se propone cambiar su *modelo de despliegue*, configurándolos dónde y cuándo sean necesarios para adaptar las capacidades globales de detección dependiendo del comportamiento actual de los IDSs, y así reducir la incertidumbre del CAS sobre lo que está pasando.

Los cambios sobre el modelo de despliegue podrían ser consecuencia, por ejemplo, de que el sistema esté recibiendo alertas contradictorias sobre la misma zona de detección. Si los IDSs que están produciendo esas alertas tienen una reputación similar, la certeza del CAS sobre lo que está ocurriendo en esa red de detección no sería manejable. Entre otras posibles respuestas, el despliegue de IDSs adicionales sobre esa zona de detección podría ayudar a dar solución a esa incertidumbre, e indirectamente a volver a evaluar la confianza sobre los IDSs que actualmente están desplegados en esa zona de detección. Además de poder cambiar el modelo de despliegue, otra opción sería la reconfiguración de las capacidades de detección de los IDSs, sin tener que reubicarlos entre las distintas zonas de detección. Estos cambios se tendrían que llevar a cabo, como se propone en este capítulo, bajo un proceso totalmente automático con el que los CIDNs, o por parte del CAS a nivel global, pudieran reaccionar ante la incertidumbre actual y buscar nuevas maneras de mejorar la precisión en la detección de ataques.

La metodología tras la cual se aborda este capítulo consiste en, primer lugar, definir aquellos elementos que todavía no se han presentado en los capítulos anteriores, y que son necesarios dentro del modelado de un sistema colaborativo de alertas (CAS) para una correcta monitorización de los activos que se están protegiendo, indispensables en la detección de ataques. A continuación se presenta el mecanismo adaptativo con el que buscar y desplegar una nueva configuración de monitorización que maximice la calidad en la toma de decisiones sobre la detección de esos ataques.

6.1. Elementos del sistema de una organización

Esta sección introduce los elementos que hasta ahora no se han definido, y que son obligatorios en la definición de un gobierno de control –sistema de monitorización– con el que supervisar el correcto funcionamiento de los servicios (o activos) desplegados en cualquier organización. Estos elementos son necesarios para el mecanismo adaptativo que se presenta a continuación, cuyo objetivo es maximizar en cada momento la calidad de las alertas generadas por los IDSs que se encuentran desplegados en el CAS.

6.1.1. Elementos adicionales para el CAS

Cada dominio de seguridad del CAS impone un conjunto de requisitos u obligaciones para el correcto funcionamiento de sus servicios. Como estos requisitos vienen dados por las necesidades que tienen sus servicios, cada dominio de seguridad puede extraer automáticamente el conjunto de requisitos que necesita monitorizar a partir de la unión de todos los que sus servicios necesitan individualmente.

Definición 10. Las obligaciones o *requisitos* demandados por el CAS se definen por un conjunto $R = \{R_1, R_2, ..., R_u\}$, donde u es el número total de esos requisitos.

Como la importancia de los requisitos suele ser distinta, la monitorización de alguno de ellos puede ser más importante que la de otro dependiendo del impacto que supondría su incumplimiento. En este caso, los administradores de un CIDN tienen que establecer un peso para cada requisito, $Impacto(R_k) \in [0, 1]$ con $R_k \in R$ $(1 \leq k \leq u)$, indicando el impacto o importancia si R_k se viera comprometido.

Definición 11. Cada dominio de seguridad $D_i \in CAS$ se define según sus necesidades como $R(D_i) = \{R_1, R_2, ..., R_x\}$, con $R_k \in R$ $(1 \leq k \leq x$, y $x \leq u)$, donde x es el número de requisitos necesarios para el buen funcionamiento de los servicios de D_i.

Al ser los requisitos exigidos por cada dominio un subconjunto del total que necesita el CAS según sus propósitos globales de detección, es decir, $x \leq u$, cada dominio de seguridad podría requerir un número x distinto de requisitos. Nótese que esta definición también se podría modelar como una función basada en la lógica proposicional, donde $R(D_i) = R_1 \wedge R_2 \wedge ... \wedge R_x$ y, en extensión, el conjunto total de requisitos demandados por el CAS se modelaría como $R(CAS) = R(D_1) \wedge R(D_2) \wedge ... \wedge R(D_m)$.

Cada uno de los IDSs desplegados en un determinado CIDN es capaz de monitorizar uno o varios de los requisitos demandados por el dominio de seguridad donde permanece realizando sus funciones de detección. En este contexto, cada requisito se asimilaría a un tipo de ataque en particular contra alguno de los servicios ofrecidos por su dominio de seguridad como, por ejemplo, una escalada de privilegios o un intento de ataque DoS. Cada IDS se caracteriza entonces por un conjunto de propiedades, o habilidades, con las que monitorizar alguno de los requisitos demandados por el CIDN. Estas propiedades serían las opciones de configuración de un IDS necesarias para detectar tipos de ataque que han sido establecidos como requisitos. Un conjunto de esas propiedades –reglas de detección– son capaces de monitorizar alguno de los tipos de ataque demandados, a fin de proporcionar alertas sobre el incumplimiento de uno o más requisitos.

Definición 12. El conjunto total de *propiedades* soportadas por los IDSs se define por $P = \{P_1, P_2, ..., P_v\}$, donde v es el número total de propiedades que los distintos IDSs pueden proporcionar al sistema para sus procesos de monitorización.

De esta forma, cada uno de los IDSs se puede modelar como se define a continuación, según las propiedades que puede soportar en un momento dado.

Definición 13. Cada IDS j (IDS_j) se define teniendo en cuenta sus propiedades como $P(IDS_j) = \{P_1, P_2, ..., P_y\}$, con $P_l \in P$ ($1 \leq l \leq y$, e $y \leq v$), y donde y es el número total de propiedades que IDS_j puede ofrecer para monitorizar alguno de los requisitos (o partes de ellos) exigidos por un CIDN, o por el CAS a nivel global.

Además, cada IDS_j desplegado en un CIDN genérico Ω también tiene asociado un valor de reputación, definido como $Rep_\Omega(IDS_j) \in [0, 1]$ ($1 \leq j \leq n$), que modela su comportamiento según la evaluación de toda la información que ese IDS ha enviado en el pasado. Esa evaluación se define en el sistema de reputación de la Sección 4.3, siendo más alta la creencia sobre que las alertas que proporciona son ciertas cuanto mayor sea su valor de reputación. El resto de componentes de cada CIDN no deberían variar en el tiempo sustancialmente, a excepción de los IDSs móviles que podrían colaborar con cualquier CIDN del CAS. Sólo cambiarán si se incluye un nuevo requisito (por ejemplo, instalando un servicio que necesita un requisito que no estaba definido), si se instala un nuevo IDS de la infraestructura o si se actualiza alguno de los existentes con mejores propiedades para obtener un proceso de monitorización más efectivo.

Analizando las definiciones anteriores, es obvio que existe una relación directa entre cada una de las propiedades que un IDS puede ofrecer y los requisitos que son necesarios para su CIDN, o para el CAS, con el objetivo de tener una correcta monitorización.

Definición 14. La relación entre el conjunto de propiedades que puede monitorizar el cumplimiento de un requisito, también llamado *Mapeo entre Requisitos y Propiedades* (del inglés Mapping amongst Requirements and Properties, MRP), se define como:

$$MRP = \{(R_k, P_w) \mid R_k \leftrightarrow P_w,\ P_w = \{P_1, P_2, ..., P_l\}\}$$

donde l simboliza el conjunto necesario de propiedades (una o varias) para satisfacer el k-ésimo requisito, con $R_k \in R$ ($1 \leq k \leq u$) y $P_w \subseteq P$ ($1 \leq l \leq v$).

6.1.2. Evaluación del sistema de monitorización

El sistema de una organización, a nivel local dentro de un CIDN o globalmente en el CAS, puede configurar a su voluntad, guiado según sus necesidades de monitorización, cada uno de los IDSs y desplegarlos luego entre los distintos dominios de seguridad para alcanzar sus objetivos establecidos de detección. Esto implica la necesidad de un *Modelo de Despliegue* (del inglés Placement Model, PM) donde cada IDS será desplegado en uno de los dominios de seguridad, con una configuración concreta con la que maximizar la ganancia de información que le puede ofrecer a su CIDN, o al CAS a nivel global.

Definición 15. El *modelo de despliegue* (PM), o de asignación, se define mediante el siguiente mapeo entre los IDSs y los dominios de seguridad:

$$PM = \{(IDS_j, D_i) \mid IDS_j \rightarrow D_i\}$$

donde el j-ésimo IDS, $IDS_j \in IDS$ $(1 \leq j \leq n)$, está (o tiene que estar) desplegado en el i-ésimo dominio, $D_i \in D$ $(1 \leq i \leq m)$.

Con este mapeo del modelo anterior, se define que uno o varios IDSs están (o serán) desplegados en un dominio de seguridad en concreto, junto con aquellas propiedades que tienen (o tendrán) activadas. Obviamente, cada uno de los IDSs solamente podrá tener activadas las propiedades que soporte, según la Definición 13.

En una instancia concreta del modelo de despliegue, para un determinado momento en el tiempo, también se tienen que almacenar dos valores. Primero, un valor donde se defina la diversidad de la confianza entre los diferentes IDSs que están desplegados en el mismo dominio de seguridad y, segundo, un valor de la diversidad de la confianza entre los IDSs que están supervisando el mismo requisito dentro de un dominio concreto. La maximización de estos valores sobre la diversidad de la confianza reforzará al sistema de monitorización con un mecanismo capaz de evaluar las alertas que intenten publicar los IDSs a los que se les suponga un comportamiento sospechoso, al tener un valor bajo de reputación en ese preciso momento, sean analizados y evaluados de manera conjunta con otros IDSs con un alto valor de reputación. La diversidad en sus comportamientos tendrá que verse reflejada, posteriormente, en el nivel de desacuerdos entre los IDSs a la hora de monitorizar las mismas fuentes de información. A través de esa discrepancia, el sistema de monitorización puede valorar los comportamientos de los IDSs desplegados, variando sus reputaciones para premiar o castigar dichos comportamientos.

Por otro lado, la gestión de la diversidad de la confianza a nivel de requisito, también es imprescindible para averiguar la validez del modelo de despliegue. Cuanto menor sea esa diversidad para un requisito concreto, peor será la calidad en la toma de decisiones sobre las alertas proporcionadas por los diferentes IDSs con respecto a ese requisito. El sistema de monitorización no podrá diagnosticar la veracidad de esas alertas, ya que el comportamiento de los IDSs, modelado a través de sus valores de reputación, son muy parecidos. Es decir, el sistema no podrá obtener información suficiente para evaluar los posibles IDSs maliciosos, y así poder aislarlos del resto. En la Sección 6.4.1 se presenta el cálculo de ambos valores de la diversidad de la confianza.

Después de analizar los elementos presentados anteriormente, se puede precisar que el sistema de monitorización estará definido, en un momento concreto, por el conjunto de relaciones entre requisitos y propiedades (MRP) y por el modelo de despliegue (PM), que, a su vez, incluye el establecimiento de qué propiedades $P(IDS_j)$ son utilizadas en IDS j para monitorizar alguno de los requisitos que son demandados por el dominio de seguridad donde está desplegado. Este último conjunto define una posible *configuración de monitorización*, también denominado *estado*, en el que el sistema de monitorización se podría encontrar. Este sistema se compone, por tanto, de los diferentes estados que ha tenido desplegado anteriormente, además del actual, donde en cada uno también se guarda un valor que simboliza la diversidad de la confianza, de la misma forma que se ha definido anteriormente, pero en este caso según los comportamientos –reputación– que presentan todos los IDSs desplegados en el CAS. En cada una de esas configuraciones de monitorización, también se almacenan todas las alertas generadas y enviadas por los IDSs, como resultado de sus procesos de monitorización.

Durante el funcionamiento del sistema de monitorización, en un preciso momento se podría cambiar de configuración, intercambiando el modelo de despliegue por uno nuevo (*reallocation*) y/o activando o desactivando algunas de las propiedades (*reconfiguration*) que tienen configurados los IDSs que, en ese momento, ya están desplegados. El objetivo detrás de estos cambios es poder obtener un conjunto distinto de evidencias que reduzca la incertidumbre actual sobre las alertas que se están recibiendo. Esta incertidumbre se puede deber a que, por ejemplo, varios IDSs están proporcionando alertas dispares, aun estando configurados para alertar sobre el mismo tipo de eventos. Alguno de estos IDSs podría estar comprometido, manifestando por ello un comportamiento (presuntamente) malicioso. Otro de los posibles cambios de configuración podría deberse a que el sistema de monitorización está demasiado tiempo sin recibir notificación de una alerta, algo que se podría considerar como no habitual. En estos cambios de configuración, se tendría que tener en cuenta la diversidad de la confianza de los IDSs, ya sea a nivel de requisito, dominio de seguridad o globalmente en todo el CAS.

En la Sección 6.3 se presenta el modelo adaptativo propuesto, donde formalmente se detalla cómo se define una configuración de monitorización, o estado, qué motivaciones hay para que el sistema de monitorización cambie de un estado a otro y cómo se puede llegar a decidir hacia qué nuevo estado tiene que evolucionar.

6.2. Configuración de un sistema de monitorización

Para una mejor comprensión sobre la motivación existente en definir un mecanismo adaptativo en el tiempo, con el que maximizar la calidad en la toma de decisiones sobre la detección de un potencial ataque, en esta sección se introduce un ejemplo sobre cómo influye una configuración de monitorización en la calidad de las alertas obtenidas de los IDSs desplegados en un CAS genérico. Además, también se presentan los elementos que se han definido en la sección anterior, así como un conjunto de restricciones sobre cómo abordar la definición, configuración y despliegue de todos esos elementos.

6.2.1. Descripción del problema

En la Figura 6.1 se muestra una configuración de monitorización en particular sobre un CAS genérico, como ejemplo del modelo de elementos definidos anteriormente.

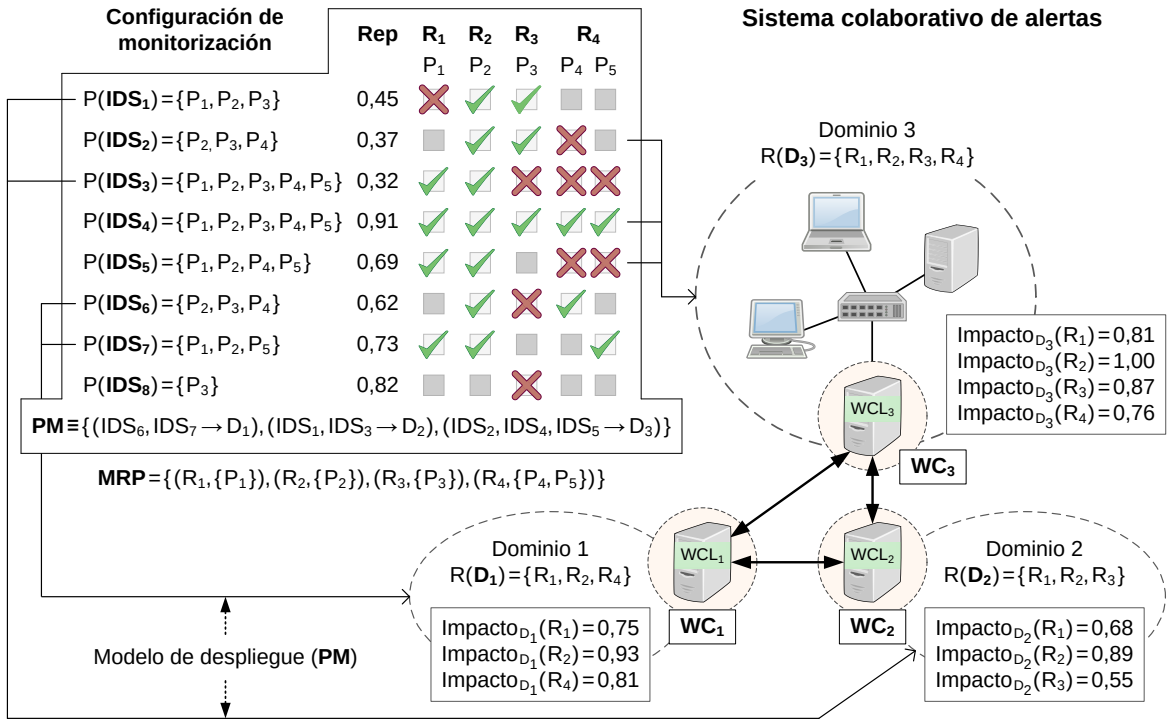


Figura 6.1: Ejemplo de configuración de monitorización

El CAS de este ejemplo muestra la distribución de todos los IDSs en tres dominios de seguridad: $CAS = \{D_1, D_2, D_3\}$. Los administradores del CAS han establecido cuatro requisitos, $R = \{R_1, R_2, R_3, R_4\}$, los cuales tienen que ser monitorizados para confirmar que el sistema está funcionando correctamente. Es decir, una serie de requisitos con los que comprobar que el sistema se encuentra en un estado seguro, libre de ataques.

En este CAS de ejemplo, también se ha estipulado una distribución de los requisitos entre dominios como: $D_1 = \{R_1, R_2, R_4\}$, $D_2 = \{R_1, R_2, R_3\}$, y $D_3 = \{R_1, R_2, R_3, R_4\}$. Nótese que solamente el dominio D_3 está configurado con todos los requisitos que se han definido en el sistema. En el resto de dominios, D_1 y D_2, alguno de los requisitos no son obligatorios ya que los servicios que tienen desplegados no lo necesitan. De esta manera, el conjunto de requisitos que necesita el CAS en sus procesos de monitorización se pueden modelar a través de una función basada en la lógica proposicional, como sería $R(CAS) = R(D_1) \wedge R(D_2) \wedge R(D_3)$. Cada dominio también especifica los riesgos en el impacto que supone el incumplimiento de esos requisitos.

Para la monitorización de los requisitos definidos en este ejemplo, se han concretado ocho IDSs junto con la definición de sus propiedades correspondientes y, también, la reputación que cada uno tiene en el momento (estado) actual para el CAS.

Finalmente, los administradores también han definido la relación entre requisitos y propiedades (ver Definición 14) según el siguiente mapeo:

$$MRP = \{(R_1, \{P_1\}), (R_2, \{P_2\}), (R_3, \{P_3\}), (R_4, \{P_4, P_5\})\}$$

Por motivos de simplicidad, se han definido tres relaciones distintas de un requisito a una única propiedad, aunque por razones de completitud para el ejemplo también se establece que R_4 precisa de dos propiedades para su monitorización: P_4 y P_5. Como se puede ver en la Figura 6.1, la unión individual de IDS_6 e IDS_7 puede satisfacer R_4.

La configuración de monitorización decretada en ese momento (estado actual) define las siguientes asignaciones en el despliegue de los ocho IDSs entre los tres dominios de seguridad: $\{IDS_6, IDS_7\} \rightarrow D_1$, $\{IDS_1, IDS_3\} \rightarrow D_2$, y $\{IDS_2, IDS_4, IDS_5\} \rightarrow D_3$. Formalmente, según la Definición 14, el modelo de despliegue de este ejemplo sería:

$$PM = \{(IDS_1, D_2), (IDS_2, D_3), (IDS_3, D_2),$$
$$(IDS_4, D_3), (IDS_5, D_3), (IDS_6, D_1), (IDS_7, D_1)\}$$

Como se puede observar, no todos los IDSs disponibles deberían estar desplegados en un dominio de seguridad. IDS_8 ni está configurado ni desplegado. Este hecho puede estar motivado a que no es necesario en ese momento al considerar el CAS que está en un estado estable, aunque IDS_8 está disponible para ser utilizado cuando sea necesario. El resto de IDSs sí que son configurados por el sistema de monitorización con respecto a sus propiedades. Como se puede ver en la Figura 6.1, esta configuración se realiza activando (representado en color verde) o desactivando (color rojo) las distintas propiedades que el sistema de monitorización considere necesarias, siempre, por supuesto, entre el conjunto disponible de propiedades que soporta cada uno de los IDSs.

Como ejemplo de funcionamiento para un momento concreto de tiempo, durante la ejecución de la configuración de monitorización mostrada en la Figura 6.1, se considera que el sistema de monitorización recibe una alerta sobre el incumplimiento de R_2. Todos los IDSs desplegados en el CAS tienen activadas la propiedad correspondiente P_2 para monitorizar R_2. Si, a la hora de informar sobre esa alerta, no existe ninguna discrepancia entre los IDSs, es decir, todos informan sobre el problema ocurrido, esa alerta se puede considerar como cierta: es un *Verdadero Positivo* y no un *Falso Positivo Malicioso* (del inglés Malicious False Positive, MFP). En caso contrario, si existe discrepancia entre los IDSs, ya que unos informan publicando una alerta mientras el resto no lo hace, se pueden dar las dos siguientes situaciones a la hora de evaluar la alerta recibida:

- La alerta se considera un Verdadero Positivo desde la perspectiva de los emisores. Esto implica que el resto de los IDSs han mostrado un mal comportamiento y han obviado informar sobre el incidente. Esta ausencia de información se clasificaría como un *Falso Negativo Malicioso* (del inglés Malicious False Negative, MFN).

- La alerta publicada por los IDSs no representa un hecho ocurrido en la realidad. Han tenido un mal comportamiento compartiendo un MFP, mientras que el resto sí que han presentado un buen comportamiento al no informar sobre un problema fingido, lo cual se clasificaría como un *verdadero negativo*.

Cualquiera de las dos situaciones anteriores, por separado, podrían ser ciertas. Esta decisión sobre qué situación sería la más acertada podría estar basada, por ejemplo, en el valor de la reputación de cada uno de los IDSs implicados en monitorizar R_2, aunque otros factores, como aquí se proponen, también se tendrían que tener en cuenta, como la diversidad de la confianza y el impacto en el incumplimiento de los requisitos. Como ejemplo, se puede considerar que la alerta anterior proviene de D_3 y ha sido enviada por uno, o dos, de los tres IDSs desplegados ahí. Si sólo IDS_4 ha enviado la alerta, el sistema podría considerar la alerta como cierta ya que el IDS posee un alto nivel de reputación, $Rep(IDS_4) = 0,91$. En cambio, la alerta podría ser un MFP si solamente es publicada por IDS_2, ya que su valor de reputación es bastante bajo, $Rep(IDS_2) = 0,37$.

Por otro lado, si dos de esos tres IDSs envían la alerta, la decisión puede ser tomada según el valor de la agregación (por ejemplo, a través de su media aritmética) obtenido a partir de los valores de reputación de ambos IDSs. En este caso, si la alerta ha sido publicada por IDS_4 e IDS_5, la confianza asignada a la alerta sería $T_{IDS_4,IDS_5}(A) = 0,8$ considerando que A es la alerta recibida. Eso significa que la alerta se podría considerar como verdadera, ya que $T_{IDS_4,IDS_5}(A) \geq 0,5$ si se asume que el umbral para considerar una alerta como verdadera es que el valor de agregación de los IDSs sea 0,5 o superior. En este caso, los dos IDSs han expuesto un comportamiento honesto, mientras que la ausencia de información por parte de IDS_2 se clasificaría como un MFN.

Si, en cambio, la alerta ha sido generada por IDS_2 e IDS_5, la confianza sobre esa alerta pasaría a ser $T_{IDS_2,IDS_5}(A) = 0,53$. Al igual que en el caso anterior, la alerta se aceptaría como verdadera, ya que $T_{IDS_2,IDS_5}(A) \geq 0,5$, a pesar de existir otros IDSs, que están desplegados en el mismo dominio de seguridad, que no han enviado esa alerta y que tienen un valor de reputación más alto que los dos IDSs anteriores. Por lo tanto, hay otros parámetros que el sistema de reputación debe de tener en cuenta al evaluar la veracidad de una alerta. Entre ellos, la diversidad de la confianza (TD) puede ser un valor de dispersión bastante importante para realizar esa evaluación.

Por otro lado, si la alerta proviene de D_1, es necesario que los dos IDSs desplegados ahí envíen la alerta, al ser demasiado baja la diversidad de la confianza en ese dominio de seguridad; concretamente, $TD(D_1) = 0,11$ (considerando la diferencia absoluta). Si solamente uno de los dos IDSs desplegados en D_1 informa sobre la alerta, el sistema no podría discernir la veracidad sobre la misma. Aunque la reputación del emisor tiene un valor respetable –$Rep(IDS_6) = 0,64$ o $Rep(IDS_7) = 0,73$–, el otro IDS que no informa también tiene una reputación similar para clasificar su comportamiento como malicioso. Una posible reacción frente a la incertidumbre que supone no poder clasificar una alerta como verdadera o fraudulenta es reconfigurar el sistema con una nueva configuración de monitorización que maximice la diversidad de la confianza.

6.2.2. Restricciones en un sistema genérico de información

Para llevar a cabo el proceso de monitorización descrito más arriba, el sistema debe asegurar el siguiente conjunto de restricciones a la hora de poder abordar la definición, configuración, y despliegue de cada uno de sus elementos.

1) Los administradores del CAS tienen que definir, al menos, una instancia por cada elemento para que el sistema de monitorización tenga una mínima funcionalidad: un dominio, un requisito, un IDS y una propiedad.

2) La unión de todas las propiedades que pueden soportar los IDSs deben satisfacer todos los posibles requisitos que el sistema pueda demandar.

3) Pueden existir dominios que no necesiten cubrir alguno de los requisitos definidos por el sistema de monitorización, aunque sí es obligatorio que todos los dominios requieran, al menos, uno de los requisitos definidos.

4) Cada uno de los requisitos establecidos por los dominios tiene que tener, al menos, una correspondencia directa con la propiedad o propiedades que lo satisfagan.

5) No puede desplegarse un IDS en el sistema de monitorización sin que previamente haya sido configurado con alguna de las propiedades que puede soportar.

6) Sólo se puede configurar un IDS con las propiedades que puede soportar, aunque pueden haber propiedades de un IDS que no se utilicen en un momento dado.

7) Un IDS sólo puede ser desplegado en un único dominio, aunque no es obligatorio que todos los IDSs tengan que ser desplegados siempre y cuando se cumplan los requisitos que cada dominio necesita monitorizar.

8) Al menos un IDS por dominio tiene que ser desplegado, a fin de satisfacer todos los requisitos demandados por esos dominios de seguridad.

9) Los IDSs desplegados en un dominio tienen que tener configuradas las propiedades necesarias para cubrir todo el espectro de requisitos obligatorios para ese dominio.

10) No pueden haber IDSs configurados con más propiedades de las que son necesarias para monitorizar los requisitos de un dominio, a fin de evitar "ruido" innecesario.

Destacar que las últimas seis restricciones obedecen al modelo de despliegue aplicado en cada momento, mientras que las cuatro primeras se vinculan a los componentes del sistema que no deberían variar habitualmente en el tiempo.

En cualquier sistema de monitorización, o CAS, se deben cumplir, obligatoriamente, todas las restricciones listadas en la enumeración anterior. Sin embargo, pueden haber situaciones donde alguna de esas restricciones se tenga que "relajar", a fin de tener un sistema de monitorización lo más seguro y fiable posible. Esta incompletitud puede ser causada por la imposibilidad de tener un amplio sistema de monitorización debido, por ejemplo, a los pocos recursos que el proceso de monitorización pueda utilizar del sistema de cómputo subyacente. De entre las restricciones anteriores, la octava, por ejemplo, se podría incumplir de manera temporal a fin de mejorar la confianza depositada sobre el sistema de monitorización, aunque existiera un dominio o un requisito sin monitorizar. Podría ser más importante en un momento dado el tener gran parte, o la totalidad, del sistema de monitorización desplegado en otro dominio mucho más crítico, teniendo en cuenta el riesgo en el impacto que reflejan sus requisitos.

6.3. Modelo adaptativo de la monitorización

En esta sección se describe en detalle el modelo adaptativo que se ha diseñado para que los CIDNs, o el mismo CAS de manera global, sea capaz de reducir la incertidumbre actual sobre la alertas que recibe desde el sistema de monitorización.

El objetivo en reducir la incertidumbre descansa en el control que tiene que tener el sistema sobre el comportamiento que los IDSs pueden exhibir a la hora de proporcionar alertas sobre el correcto funcionamiento de los activos que están siendo protegidos. Para tal fin, el sistema para la detección colaborativa de ataques va a ser capaz de modificar la configuración de monitorización actual por otra nueva con la que poder conseguir una mayor ganancia, o calidad, sobre la información. A partir de esa nueva configuración, se podrá decidir cuándo un IDS está mostrando un comportamiento honesto o malicioso. A una configuración de monitorización también se le denomina como *estado*.

6.3.1. Definición de estado como una matriz de configuración

Cada uno de los posibles estados, en el que se puede encontrar el sistema subyacente, corresponde con la configuración de monitorización desplegada en cada momento. Por lo tanto, la definición de un estado vendrá dada según el modelo de despliegue –qué IDSs están desplegados en qué dominio– y el conjunto de propiedades que se están utilizando de cada IDS para satisfacer los requisitos demandados por cada dominio.

Definición 16. Un *estado* se define a través de una matriz $m \times u$ de configuraciones de monitorización, donde las filas hacen referencia a los m dominios en los que está dividido el sistema, $\forall D_i \in CAS$ $(1 \leq i \leq m)$, y las columnas por los distintos requisitos que son demandados por esos dominios y que deben ser monitorizados para certificar su correcto cumplimiento frente a cualquier tipo de alteración, $\forall R_k \in R$ $(1 \leq k \leq u)$.

La celda (i, k) dentro de esa matriz sobre un estado define una serie de propiedades que satisface R_k, según lo demandado por D_i. Estas propiedades serán proporcionadas por los distintos IDSs que han sido desplegados en el dominio D_i.

Definición 17. Cada celda (i, k) de la matriz que representa un estado, con $1 \leq i \leq m$ y $1 \leq k \leq u$, se define formalmente como:

$$Estado(D_i, R_k) = \{(IDS_j, P_l)\} = \{IP_{jl}\}$$

donde cada celda dentro de la matriz es un conjunto de pares <`IDS,Propiedad`>, o $\{IDS_j, P_l\} = IP_{jl}$. Ese índice j corresponde al j-ésimo IDS que estará configurado con la propiedad l, con $IDS_j \in D_i$ y $P_l \in P$.

En (6.1), se muestra una matriz con el estado actual que se encuentra desplegado en el ejemplo presentado en la Sección 6.2.1, y mostrado gráficamente en la Figura 6.1.

$$\begin{pmatrix} \{IP_{71}\} & \{IP_{62}, IP_{72}\} & \{\} & \{IP_{64}, IP_{75}\} \\ \{IP_{31}\} & \{IP_{12}, IP_{32}\} & \{IP_{13}\} & \{\} \\ \{IP_{41}, IP_{51}\} & \{IP_{22}, IP_{42}, IP_{52}\} & \{IP_{23}, IP_{43}\} & \{IP_{44}, IP_{45}\} \end{pmatrix} \quad (6.1)$$

A partir de un estado, el sistema de monitorización se podría mover a uno nuevo de entre los $|IDS| \cdot |CAS| \cdot 2^{|P|} - 1$ posibles estados. Para el ejemplo de la Sección 6.2.1, el número total de posibles estados sería $24 \cdot 2^{27} - 1$ (más de 3 mil millones). Sin embargo, ese enorme número se vería drásticamente reducido después de aplicar las restricciones definidas en la Sección 6.2.2, quedándose al final en 298 407 posibles estados.

6.3.2. Arquitectura para buscar y desplegar un nuevo estado

La selección de un nuevo estado, o la primera configuración de monitorización que debe ser ejecutada, implica la búsqueda del mejor modelo de despliegue que maximice la ganancia de información a la hora de identificar ataques potenciales sobre los activos. La Figura 6.2 muestra la arquitectura propuesta para escoger, de manera automática, el mejor próximo estado al que moverse a partir de la evaluación de toda la información histórica, si la hubiera, obtenida hasta el momento.

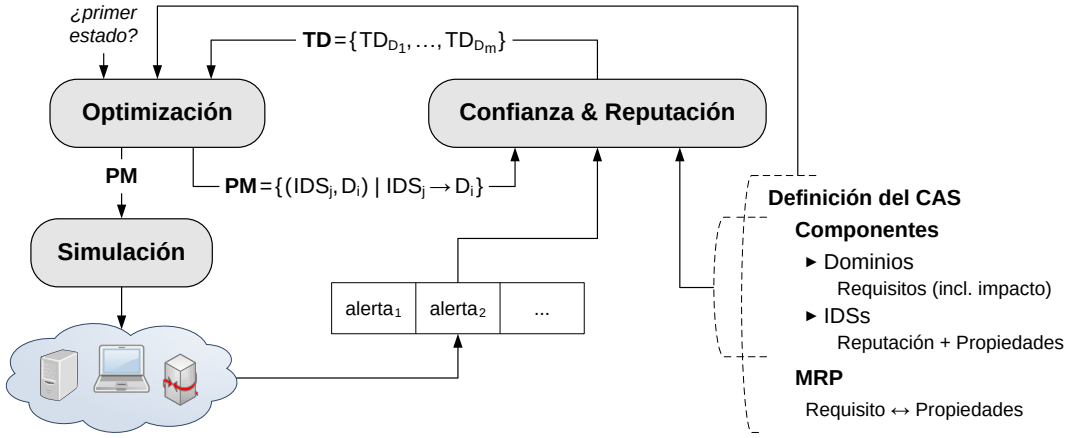


Figura 6.2: Arquitectura del despliegue de una configuración de monitorización

El módulo *Optimización* es el encargado de encontrar el mejor modelo de despliegue posible (PM) a partir de la configuración de monitorización que está en ese momento en ejecución. La búsqueda de este mejor modelo de despliegue se convierte en un problema de optimización debido al gran número de (posibles) estados que se tienen que evaluar, como se ha podido constatar con el ejemplo anterior, y también a la necesidad de tener que escoger uno que conlleve los menores cambios posibles a partir de la configuración actual. La reconfiguración suele implicar algunos costes que no deben ser despreciados. La minimización de estos cambios permitirá que el proceso de configuración y despliegue de los IDSs del nuevo estado sea lo más rápido posible.

Este módulo tiene que utilizar entonces distintas técnicas de optimización, teniendo en cuenta todos los componentes instalados en el sistema, como pueden ser el cambio aleatorio de algunas características –despliegue de los IDSs o activación/desactivación de propiedades– o el intercambio de características entre sí [197]. El diseño del algoritmo de optimización desarrollado se detalla a continuación en el siguiente apartado.

La elección particular de una u otra técnica de optimización es relativamente poco significativa, mientras se obtenga un modelo de despliegue lo suficientemente aceptable. Independientemente de la técnica que se escoja, el factor clave que debe ser maximizado es la diversidad sobre la confianza. Además, también hay que apuntar que esta búsqueda se tiene que llevar a cabo sobre los estados válidos (aquellos que cumplan los requisitos demandados por los dominios de seguridad). Esta restricción se ha tenido en cuenta en la arquitectura de optimización mostrada en la Figura 6.2.

La salida del módulo *Optimización* es un modelo de despliegue en particular, que es enviado al módulo *Confianza & Reputación* para su evaluación. Este último módulo, le devolverá al primero la diversidad de la confianza para ese modelo concreto, utilizando para ello todos los valores de reputación de cada uno de los IDSs. Además, nótese que el módulo *Confianza & Reputación* gestiona los cálculos que el sistema de monitorización tiene que realizar para evaluar la reputación de cada IDS, teniendo acceso para ese fin a todas las alertas que han sido generadas por los IDSs. Todo el sistema para la gestión de la confianza, basado en reputación, se explica en detalle en la Sección 6.4.

Cuando el módulo de *Optimización* encuentre el mejor modelo de despliegue posible, teniendo en cuenta la diversidad de la confianza calculada por *Confianza & Reputación*, esta nueva configuración de monitorización la recibe el módulo *Simulación* para ponerla en marcha en el sistema de cómputo subyacente. Por último, señalar que la decisión que seguramente tendrá una mayor consecuencia de cara al rendimiento final del sistema es poder determinar cuándo arrancar todo el proceso que se ha descrito hasta ahora, con el objetivo de buscar y aplicar un nuevo estado que sea mejor que el actual. Este punto se trata posteriormente en profundidad en la Sección 6.4.4.

Descripción del algoritmo de optimización diseñado

En esta sección, se presenta el algoritmo de optimización diseñado con el que buscar nuevos, y mejores, modelos de despliegue. Aquí se explica por completitud, aunque el objetivo principal se centra en el modelo de confianza y reputación y no en el desarrollo de nuevos algoritmos de optimización. Sin embargo, el conocimiento de este algoritmo es necesario ya que se utiliza luego en los experimentos de la Sección 6.5, con los cuales se van a poder analizar los beneficios al hacer uso del mecanismo de reconfiguración que se está proponiendo. Por lo tanto, se podrían utilizar otros algoritmos de optimización, siempre y cuando se pueda mejorar la diversidad de la confianza.

En las últimas décadas, han aparecido multitud de procedimientos de optimización heurísticos con una estructura bastante similar a la que aquí se presenta, con resultados muy prometedores en algunos casos [251]. En esta tesis doctoral se utiliza una variante del algoritmo *Simulated Annealing* que, básicamente, se puede ver como un problema básico de *Hill Climbing* [204]. En esta variante, se incluye una aceptación probabilística de no mejora de soluciones para que la búsqueda no sea infinita, aplicando un criterio de parada basado en un número máximo de movimientos fallidos, sobre la que aquí se aplica el concepto de diversidad de la confianza con el que obtener mejores modelos de despliegue dentro del espacio completo de soluciones.

El Algoritmo 3 muestra, esquemáticamente, los pasos en pseudocódigo del algoritmo de optimización diseñado basado en el de Simulated Annealing.

Algoritmo 3: Esquema de optimización basado en el algoritmo Simulated Annealing

```
1 S ← S_0
2 T ← T_0
3 while criterio no cumplido do
4 |   for MIL movimientos do
5 |   |   Escoger C ∈ N(S) con una probabilidad uniforme
6 |   |   Escoger U ∈ (0, 1) con una probabilidad uniforme
7 |   |   if F(C) > F(S) + T ln U then
8 |   |   |   S ← C
9 |   T ← ιT
```

Esta búsqueda comienza a partir de una solución inicial $S_0 \in \mathbb{S}$, siendo $\mathbb{S}$ el espacio completo de soluciones. El algoritmo se vale de un parámetro de control $T \in \mathbb{R}^+$ que es conocido como la *temperatura*, el cual se inicia con un valor positivo cualquiera T_0 que es decrementado de manera gradual en cada iteración, normalmente por un "enfriamiento" geométrico: $T_{i+1} = \iota T_i$, con $\iota \in (0, 1)$. En este punto se intenta generar un número MIL (*Moves in Inner Loop*) de estados vecinos para cada valor de temperatura. Entre esos estados, se selecciona uno como candidato C dentro de su vecindad $N(S_i) \in S_i$ y se le aplica alguna función de movimiento a S_i. En concreto, esta función consiste en decidir de forma aleatoria si intercambiar el despliegue de dos IDSs entre sí o tomar un IDS al azar y desplegarlo en otro dominio distinto, garantizando, obviamente, los requisitos de detección establecidos en la Sección 6.2.2. La nueva solución se acepta si es mejor que S_i, según lo establecido por una función *fitness* $F : \mathbb{S} \to \mathbb{R}$. Esta función F corresponde con la diversidad de la confianza del sistema que está siendo monitorizado.

Como cualquier algoritmo de optimización, éste también padece de un inconveniente común y bien conocido: la existencia de *óptimos locales*, dando la impresión de que se ha alcanzado un óptimo global dentro del espacio de soluciones. Como posible solución, se propone aceptar estados candidatos que son ligeramente peores que S_i, siempre que las funciones fitness entre S_i y el nuevo estado candidato C no sea superior de $|T \ln U|$, siendo $U \in (0, 1)$ una variable aleatoria uniforme. De esa manera, este término tenderá a 0 conforme T sea más pequeño, por lo que cada vez será mucho más complicado el poder aceptar movimientos peores conforme vaya decreciendo la temperatura. Por último, el algoritmo termina al alcanzar un criterio establecido de parada, siendo lo más habitual un número fijo de MaxIL iteraciones o, incluso, después de alcanzar un número máximo de iteraciones consecutivas sin obtener mejoras en el espacio de soluciones.

Este algoritmo de optimización propuesto está basado en el de Simulated Annealing, ya que ofrece un buen equilibrio entre la simplicidad en el proceso de optimización y la calidad de las soluciones encontradas. A pesar de ello, también se podrían utilizar otros algoritmos de optimización más sofisticados para encontrar mejores soluciones, ya que éste en concreto no garantiza la obtención de una solución óptima.

6.4. Reputación en la diversidad de la confianza

A continuación, se presenta en detalle el sistema de gestión de la confianza basado en la reputación, que adapta el modelo presentado en los capítulos anteriores, para incluir la diversidad de la confianza con el objetivo siempre de modelar el comportamiento de los IDSs mientras monitorizan los requisitos demandados por el CAS.

La confianza en los IDSs representa un factor de suma importancia, con el que poder determinar cómo se tienen que desplegar los IDSs de la infraestructura en el sistema de monitorización para maximizar la calidad en los procesos globales de detección. La idea tras este concepto es garantizar, cuando sea posible, un despliegue de todos los IDSs de forma conjunta con una alta dispersidad en sus valores de reputación. Esto aseguraría, por ejemplo, que IDSs poco fiables –dudosa reputación en sus comportamientos– nunca estuvieran desplegados sin que hubieran otros en el mismo dominio en los que el sistema tuviera una alta confianza en sus actos. Además, la diversidad de la confianza también ayuda a identificar rápidamente comportamientos maliciosos de los IDSs.

La diversidad de la confianza se va a utilizar, entonces, como guía en la evolución del sistema de monitorización entre los diferentes estados de configuración. En la Figura 6.3 se muestra un esquema de ejemplo sobre cómo los estados van evolucionando a lo largo del tiempo hasta el actual, que está ejecutándose en un momento dado t.

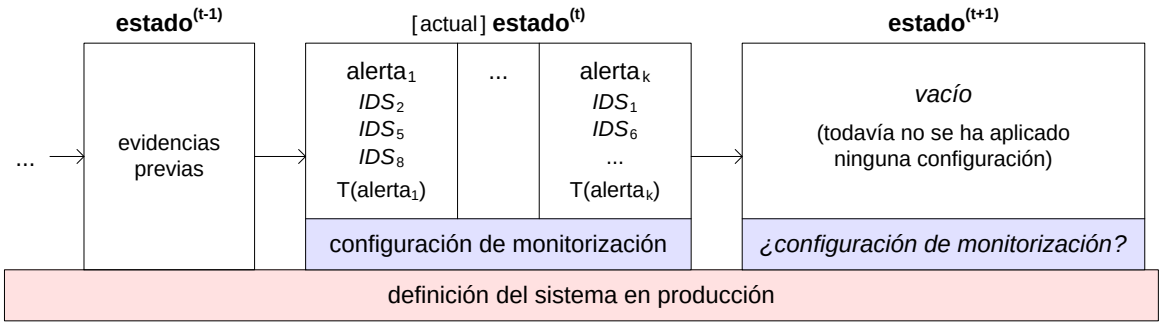


Figura 6.3: Evolución de los estados a lo largo del tiempo

En cada estado, como se puede ver en la Figura 6.3, se mantienen las distintas alertas que han sido generadas por los diferentes IDSs según la configuración de monitorización establecida en cada instante. En el caso del estado actual, denotado por $estado^{(t)}$, se han recibido k alertas hasta ese momento. Junto a ellas, también se almacenan los distintos IDSs que han generado y publicado cada una de esas alertas.

Todos los datos anteriores, alertas e IDSs, se guardan en cada uno de los estados para así poder hacer uso, en cualquier momento, de todo el historial sobre el comportamiento de cada uno de los IDSs desplegados bajo una configuración concreta de monitorización Además, en cada uno de esos estados, también se guarda el valor de confianza, denotado por $T(alerta_k)$, que el sistema de monitorización ha calculado sobre la alerta recibida.

El cálculo de la confianza sobre cada alerta, teniendo en cuenta la diversidad de la confianza, se explica a continuación en la Sección 6.4.2.

6.4.1. Cálculo de la diversidad de la confianza

En este cálculo, se pueden utilizar diversas medidas de dispersión [252] en el afán de maximizar las valoraciones de las alertas recibidas según los comportamientos de cada IDS durante sus procesos de detección. Sea cual fuere la medida de dispersión escogida, todas harán uso de los valores de reputación actuales que tienen los IDSs desplegados en los distintos dominios. Entre las posibles medidas de dispersión, se pueden destacar: la diferencia absoluta (o rango), extrayendo los IDSs con el máximo y el mínimo valor de sus reputaciones; la media aritmética de todos los valores de reputación; o la desviación estándar, varianza o covarianza de los diferentes valores de reputación de los IDSs.

En la Sección 6.5, se analizan esas medidas de dispersión para analizar cuál de ellas supone la mejor elección en el cálculo de la diversidad de la confianza. Como se ha dicho antes, el cálculo de esta diversidad se tiene que realizar a tres niveles:

- A *nivel de requisito*, para garantizar la veracidad de las alertas que ofrecerán los diferentes IDSs que tienen asignados su monitorización.
- A *nivel de dominio*, para comprobar que todos sus requisitos están monitorizados por un conjunto de IDSs que puedan avalar la veracidad de las alertas generadas.
- A *nivel global del CAS*, para confirmar que el modelo de despliegue de los IDSs es el mejor posible, y así poder reducir la incertidumbre que éstos puedan producir cuando publican alertas que se han producido en el sistema de detección.

En los siguientes apartados se presenta en detalle el cálculo de la diversidad de la confianza según cada uno de los tres niveles anteriores.

Diversidad de la confianza a nivel de requisito

En el cálculo de la diversidad de la confianza de un requisito en concreto, además del uso de una medida de dispersión, también es importante tener en cuenta la importancia de los IDSs dependiendo del impacto que supondría el incumplimiento del requisito que monitorizan. Por todo ello, el cálculo sobre la diversidad de la confianza de un requisito R_k demandado por un dominio genérico Ω, el cual se denota como $TD_\Omega(R_k) \in [0,1]$, se obtiene como resultado según (6.2). Este valor se tiene que almacenar en el modelo de despliegue (PM) actual que está en ejecución, o el enviado por el módulo *Optimización* para su evaluación como se ha presentado en la Sección 6.3.2.

$$TD_\Omega(R_k) = max\{Rep_\Omega(IDS_{j,R_k})\} \cdot \varphi(Rep_\Omega(IDS_{j,R_k}) \cdot \mu_\Omega(R_{k,IDS_j})) \tag{6.2}$$

donde $max\{Rep_\Omega(IDS_{j,R_k})\}$ representa el valor de reputación más alto de entre los IDSs desplegados en el dominio Ω, $\forall IDS_j \in PM(\Omega)$, los cuales monitorizan el requisito R_k; φ la medida de dispersión seleccionada con la que poder maximizar la diversidad de la confianza; $Rep_\Omega(IDS_{j,R_k})$ el valor de reputación del j-ésimo IDS desplegado en el dominio Ω, y que tiene asignada la monitorización de R_k; y $\mu_\Omega(R_{k,IDS_j})$ el riesgo en el incumplimiento de R_k, según los IDSs que lo están monitorizando.

El primer parámetro $max\{Rep_{\Omega}(IDS_{j,R_k})\}$ se ha definido en (6.2) al ser obligatorio para que la dispersión en un modelo de despliegue sea más alta conforme la reputación de los IDSs sea cada vez mayor. Como ejemplo, se pueden considerar dos configuraciones de monitorización con dos IDSs que monitorizan el mismo requisito, tomando 0,1 y 0,4 como valores de reputación para la primera configuración, mientras que para la segunda serían valores de 0,6 y 0,9. En cualquiera de esos dos casos, la dispersión sería la misma si se escogiera la diferencia absoluta como medida de dispersión: $\varphi = 0,3$. Sin embargo, la diversidad de la confianza de la primera configuración de monitorización tendría que ser peor que la segunda, al incluir IDSs con una reputación excesivamente baja.

En referencia al último parámetro que se ha incluido en (6.2), éste se podría definir teniendo en cuenta tanto el impacto asociado a si se incumpliera el requisito R_k como el número de propiedades que son necesarias para la monitorización de ese requisito, y que IDS_j tiene configuradas para ello. De esta manera, mediante (6.3) se puede calcular el riesgo existente al incumplimiento de R_k, denotado como $\mu_{\Omega}(R_{k,IDS_j}) \in [0,1]$.

$$\mu_{\Omega}(R_{k,IDS_j}) = \frac{Impacto_{\Omega}(R_k) \cdot |P_{\Omega}(R_{k,IDS_j})|}{|P_{\Omega}(R_k)|} \tag{6.3}$$

donde $Impacto_{\Omega}(R_k) \in [0,1]$ representa el impacto que supondría para el dominio Ω si se incumpliera el requisito $R_k \in R$; $|P_{\Omega}(R_{k,IDS_j})|$ el número de propiedades que están configuradas en IDS_j para monitorizar R_k; y $|P_{\Omega}(R_k)|$ el número total de propiedades que necesita el dominio Ω para la monitorización completa de R_k.

Tomando como ejemplo la configuración de monitorización presentada gráficamente en la Figura 6.1, la diversidad de la confianza (TD) de los requisitos demandado por D_3, usando la diferencia absoluta como medida de dispersión, serían: $TD_{D_3}(R_1) = 0,1622$, $TD_{D_3}(R_2) = 0,4914$, $TD_{D_3}(R_3) = 0,4275$ y $TD_{D_3}(R_4) = 0,6294$. Estos valores podrían ser determinantes para que D_3 tuviera la certeza en que, posiblemente, R_4 esté siendo bien monitorizado por sus IDSs, pero no así con el resto de requisitos, por lo que podría ser necesario el despliegue de una nueva configuración de monitorización que mejorara la calidad en la futura toma de decisiones sobre la detección de ataques.

Diversidad de la confianza a nivel de dominio

Una vez calculada la diversidad de la confianza de cada requisito con (6.2), el cálculo de la diversidad para un dominio genérico cualquiera Ω, denotado como $TD_{\Omega} \in [0,1]$, se puede obtener a través de una sencilla función de agregación como la definida en (6.4). De forma similar al caso anterior, este valor también se almacena para cada uno de los dominios definidos en el modelo de despliegue que está siendo evaluado.

$$TD_{\Omega} = \bigoplus_{k=1}^{\theta_{\Omega}(R)} TD_{\Omega}(R_k) \tag{6.4}$$

donde $\oplus$ simboliza una operación de agregación; $\theta_{\Omega}(R)$ el número total de requisitos demandados por el dominio Ω; y $TD_{\Omega}(R_k)$ la diversidad de la confianza para el k-ésimo requisito exigido por el dominio Ω, calculada en (6.2).

Siguiendo con el ejemplo anterior, la diversidad de la confianza del dominio D_3 sería $TD_{D_3} = 0,4432$ si se utiliza la media aritmética como operación de agregación, o bien $TD_{D_3} = 0,3337$ si se hace uso de la media armónica. Estos resultados, sobre los mismos valores para los cálculos, evidencian que la operación de agregación escogida puede ser determinante en la evaluación de las alertas que se producen en un dominio.

Diversidad de la confianza a nivel global del CAS

Finalmente, el cálculo de la diversidad de la confianza sobre todo el CAS se realiza de forma similar al anterior, pero, en este caso, en base a todos los valores de diversidad calculados para cada dominio de manera individual. Este cálculo global de la diversidad, denotado por $TD_{CAS} \in [0, 1]$, se obtiene con (6.5). Como los casos anteriores, este valor también se almacena en el modelo de despliegue que se está evaluando.

$$TD_{CAS} = \bigoplus_{i=1}^{\theta_{CAS}(D)} TD_{D_i} \tag{6.5}$$

donde $\oplus$ es una operación de agregación, como en el caso anterior en (6.4); $\theta_{CAS}(D)$ el número total de dominios que componen el CAS; y TD_{D_i} la diversidad de la confianza de cada uno de los dominios, con $D_i \in CAS$, calculada previamente según (6.4).

La diversidad global en la confianza puede variar significativamente según la medida de dispersión utilizada en (6.2), y las dos operaciones de agregación escogidas para (6.4) y (6.5). La decisión final sobre qué medidas y operaciones se deben escoger es labor de los administradores del CAS, aunque este punto es analizado en la Sección 6.5 donde se termina argumentando cuáles son las que consiguen mejores resultados.

Además de evaluar cuál sería la mejor configuración de monitorización posible hacia la que el sistema podría evolucionar, los tres valores anteriores sobre la diversidad de la confianza también se pueden utilizar para decidir cuándo el sistema tiene que cambiar el estado actual por uno nuevo con el que mejorar el sistema de monitorización subyacente. Esta decisión se analiza con total profundidad en la Sección 6.4.4.

6.4.2. Modelado del comportamiento de los IDSs

Cuando el sistema de monitorización recibe una alerta, proveniente de los IDSs que tiene bajo su cargo, ésta tiene que ser evaluada según la reputación de aquellos IDSs que tienen configurada la propiedad de monitorización causante de la violación del requisito que hizo disparar la alerta. Posteriormente, y según el comportamiento mostrado por esos IDSs, el sistema de monitorización actualiza los valores de reputación de los IDSs que se han visto implicados en la monitorización del requisito en cuestión.

En el cálculo de la confianza que el sistema de monitorización puede depositar sobre cualquier alerta, se debe tener en cuenta el grado de acuerdo alcanzado entre todos los IDSs configurados para la monitorización del evento acontecido, el número de dominios donde se ha producido y la diversidad de la confianza en esos dominios.

Con respecto a los dos últimos factores, es importante percatarse si la misma alerta ha sido generada en más de un dominio, como resultado de una violación de requisitos de manera distribuida. Teniendo en cuenta estos factores, la confianza que el sistema de monitorización puede depositar sobre una alerta A al estar monitorizando el requisito k, denotada como $T(A_{R_k}) \in [0, 1]$, se puede calcular mediante (6.6).

$$T(A_{R_k}) = \bigoplus_{i=1}^{\theta_D(A_{R_k})} |Fancy_{D_i}(A_{R_k})| \cdot TD_{D_i}(R_k) \tag{6.6}$$

donde $\oplus$ representa cierta operación de agregación, como las que se han comentado en el apartado anterior; $\theta_D(A_{R_k})$ el número de los dominios que han generado la alerta; $Fancy_{D_i}(A_{R_k}) \in [-1, 1]$ el grado de acuerdo logrado por los IDSs al generar la alerta A, que viola el requisito R_k, dentro de un dominio particular D_i; y TD_{D_i} la diversidad de la confianza del i-ésimo dominio donde se ha generado A_{R_k}, según (6.4).

Hay que puntualizar en este punto que la función $Fancy_{D_i}(A_{R_k})$ ha sido definida en la Sección 4.3.1, y cuyo cálculo se puede realizar a nivel intradominio a través de (4.5). En esta función, para obtener el grado de acuerdo alcanzado entre los IDSs, se tienen en cuenta tanto las reputaciones de los IDSs que han enviado el evento como la de aquellos que no lo han hecho, aunque estén configurados para su monitorización tendrían que haber informado sobre el evento ocurrido mediante la alerta correspondiente.

Una vez calculada la confianza en la alerta $T(A_{R_k})$, el CAS sería capaz de decidir si la alerta ha sido producida como consecuencia de un acto honesto o malicioso por parte de los IDSs (si es un Verdadero Positivo o un MFP, respectivamente). En consecuencia, la alerta se consideraría como verdadera –comportamiento honesto– si $T(A_{R_k}) > T_{umbral}$, siendo establecido ese umbral previamente por los administradores del CAS.

6.4.3. Actualización de la reputación de los IDSs

Durante la ejecución de una configuración de monitorización, los IDSs pueden enviar distintas alertas con las que notificar la violación de algún requisito para los que tienen configurada su monitorización. Una vez evaluada cada alerta por separado usando (6.6), el sistema de monitorización tendría que calcular, y luego actualizar, el comportamiento que ha presentado cada uno de los IDSs implicados en la monitorización del requisito en cuestión –comportamiento definido mediante un valor de reputación. Un IDS podrá ser entonces recompensado, aumentando su reputación, o bien penalizado, decrementando su valor, según su comportamiento en la generación y envío de alertas.

El nuevo valor de reputación de cada IDS, pero únicamente de aquellos relacionados con la alerta que se acaba de evaluar, se actualizará reemplazando su valor actual por el nuevo valor de reputación calculado. El cálculo del nuevo valor de reputación de los IDSs se puede llevar a cabo en distintos momentos en el tiempo, definido de antemano por los administradores del CAS. Por un lado, este cálculo se puede realizar después de evaluar la recepción de cada una de las alertas o, también, por otro lado, justo antes de cambiar la configuración de monitorización por otra nueva.

La primera opción podría conllevar una elevada carga de trabajo cuando el sistema recibe gran cantidad de alertas en un corto espacio de tiempo, mientras que la segunda podría considerar el mantenimiento de un valor de la reputación desactualizado o irreal, sobre todo, cuando la distancia de tiempo es muy alta entre las últimas y las primeras alertas recibidas. En cualquier caso, el cálculo de los valores de reputación de todos los IDSs se realiza de la misma manera, independientemente de cuándo se lleve a cabo.

El cálculo de la reputación para cada IDS debe tener en cuenta un par de factores: i) la evaluación calculada sobre cada alerta, según (6.6); y ii) la acción llevada a cabo (envío o denegación) por cada uno de los IDSs que se están evaluando. Además, también es necesario reajustar el comportamiento expuesto por los IDSs con respecto al tiempo, desde que se vieron implicados en las alertas hasta el momento actual.

A continuación, en (6.7) se define el cálculo del nuevo valor de reputación, denotado como $Rep(j)^{(t)}$, del j-ésimo IDS (IDS_j) en el tiempo actual t.

$$Rep(j)^{(t)} = \varsigma \cdot Rep(j)^{(t-1)} + (1-\varsigma) \cdot \frac{\sum_{k=1}^{\theta_j(A)} Sat_j(A_k) \cdot \mu(R_{A_k,j}) \cdot \xi(A_k)}{\theta_j(A)} \tag{6.7}$$

donde $Rep(j)^{(t-1)} \in [0,1]$ representa el último valor de la reputación de IDS_j hasta realizar este nuevo cálculo; $\theta_j(A)$ el número total de alertas en las que IDS_j ha estado implicado, ya sea porque ha enviado la alerta o ha obviado hacerlo, aun cuando tendría que haberlo hecho; $Sat_j(A_k) \in [0,1]$ la satisfacción sobre IDS_j en su comportamiento mostrado en la publicación de la k-ésima alerta; $\mu(R_{A_k,j})$ el riesgo asociado al requisito que se encuentra afectado por la alerta A_k, calculado en (6.3); y $\xi(A_k) \in [0,1]$ el factor de olvido según el tiempo desde la publicación de la k-ésima alerta hasta el momento actual t. Las dos partes de (6.7) se ponderan mediante un peso $\varsigma \in [0,1]$, establecido por un administrador, con el que fijar una importancia mayor a la reputación previa de IDS_j o al comportamiento en sus procesos de monitorización a lo largo del tiempo.

La satisfacción que el sistema puede obtener sobre el comportamiento de IDS_j, a la hora de informar sobre una alerta A_k en particular, denotado como $Sat_j(A_k)$, va a depender de la confianza depositada en dicha alerta durante su evaluación, según (6.6), y la acción que IDS_j realizó con respecto a esa alerta: envío o denegación. El modelado para esta satisfacción se calcula a través de (6.8). Nótese que esta ecuación difiere de la presentada en (4.4) en que, en este nuevo caso, se incluye la diversidad de la confianza en (6.6) durante la evaluación de la alerta recibida.

$$Sat_j(A_k) = \begin{cases} |Fancy(A_k)| & \text{si } (T(A_k) > T_{umbral} \wedge IDS_j \subseteq \vartheta_G(A_k)) \\ & \vee\ (T(A_k) \leq T_{umbral} \wedge IDS_j \nsubseteq \vartheta_G(A_k)) \\ -|Fancy(A_k)| & \text{en otro caso} \end{cases} \tag{6.8}$$

donde $T(A_k) \in [0,1]$ simboliza la confianza que el sistema deposita en la alerta A_k, calculada en (6.6); T_{umbral} la confianza umbral establecida por un administrador para decidir si A_k ha sido enviada por los IDSs en el que IDS_j estuvo envuelto; y $\vartheta_G(A_k)$ el conjunto de IDSs que han generado y enviado A_k.

Si la alerta se clasifica como verdadera y ha sido enviada por el IDS, o no lo es y no la ha enviado (no ha formado parte de ese MFP), ese IDS es recompensado incrementando su reputación debido a su comportamiento honesto. En el caso contrario, si la alerta es verdadera y el IDS no la ha enviado (es partícipe de un MFN), o no lo es y sí que la ha enviado (forma parte de ese MFP), este IDS tiene que ser penalizado decrementando su reputación debido a un comportamiento malicioso. Por tanto, la satisfacción calculada en (6.8) representa la recompensa o penalización en términos de reputación.

Por último, es vital disponer de una buena función del tiempo para modelar el peso de una alerta generada en el pasado con respecto a otras: alertas más recientes deberían tener una mayor importancia que las más antiguas. Ese factor de olvido se suele modelar en la literatura actual como una función lineal, mostrando una proporción constante en el tiempo. Sin embargo, este modelado también podría estar basado de manera análoga a cómo el ser humano percibe la diferencia de tiempo entre dos ocurrencias: dos alertas recientes son más significativas que dos alertas mucho más antiguas. En el contexto de esta tesis doctoral, el modelado del factor de olvido se define según (6.9).

$$\xi(A_k) = e^{-\Delta t(A_k)^{\nu}} \tag{6.9}$$

donde $\Delta t(A_k)$ define la diferencia entre el tiempo actual y cuando fue generada A_k y ν la variación del factor de olvido dando valores más o menos altos a las alertas más actuales. Dependiendo del factor ν, la importancia de una alerta puede variar a lo largo del tiempo considerablemente, como se puede contemplar en la Figura 6.4.

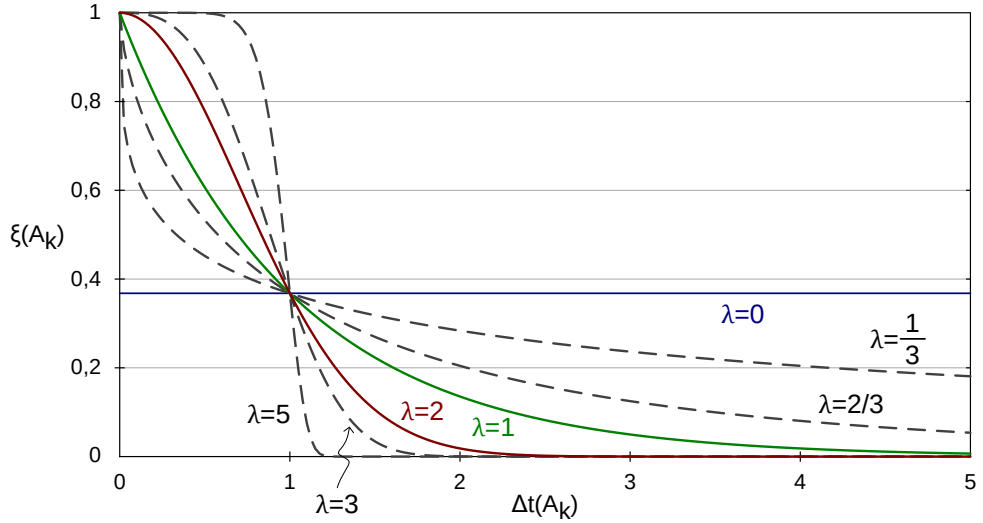


Figura 6.4: Variaciones en el modelado de tiempo según varios factores de olvido

En el caso particular cuando $\nu = \frac{1}{3}$, las alertas son "olvidadas" de forma más rápida y, a partir de un cierto momento, la diferencia entre dos alertas generadas en tiempos distintos tendría una importancia bastante similar. En cambio, si se considera $\nu = 5$, una alerta reciente mantiene una alta importancia, pero decae drásticamente hasta que pierde toda su importancia, por completo, en muy poco tiempo.

6.4.4. Decisión en la evolución hacia una nueva configuración

Existen diversas razones que motivan cambiar una configuración de monitorización por otra nueva con mejores prestaciones al monitorizar los requisitos, con el objetivo de detectar y eliminar los MFPs y MFNs generados u obviados, respectivamente, por IDSs con un comportamiento malicioso. Entre esas posibles razones, se encuentra el uso de la diversidad de la confianza. Cuando es demasiado baja, la evaluación de los IDSs en sus comportamientos deja de ser útil. El sistema de monitorización será incapaz de clasificar las alertas como verdaderas y, en consecuencia, no podría identificar comportamientos deshonestos para aislar los IDSs maliciosos de los procesos de detección.

La diversidad de la confianza se puede implantar a tres niveles, como se ha visto en la Sección 6.4.1. Primero, comprobando que la diversidad de los requisitos de un dominio Ω, calculada en (6.2), se encuentra por debajo de un umbral: $TD_{\Omega}(R_k) < TD_{\Omega,umbral}$. Si alguno lo incumple, el sistema tendría que cambiar el estado actual por uno nuevo que le proporcione esa protección. Destacar que se podrían definir diferentes umbrales para cada requisito, y distintos entre los dominios. En segundo lugar, se podría analizar si la diversidad global en el CAS, calculada con (6.5), ha bajado de otro umbral establecido para todo el CAS. Es decir, el despliegue de una nueva configuración de monitorización, presentado en la Sección 6.3.2, se lanzaría cuando $TD_{CAS} < TD_{CAS,umbral}$. Por último, sería interesante que todo este proceso se llevara a cabo si la diversidad para un dominio, calculada mediante (6.4), cayera por debajo de otro umbral; cuando $TD_{\Omega} < TD_{\Omega,umbral}$. Al igual que en el primer caso, este umbral podría ser diferente entre los dominios según los niveles de impacto establecidos para sus requisitos. (Todos los umbrales anteriores tienen que ser establecidos por los administradores del CAS.)

Además de la diversidad de la confianza, el cambio de una configuración por otra se podría deber al número de alertas recibidas en una determinada ventana de tiempo. Si durante ese espacio de tiempo, el sistema recibe un gran número de alertas con respecto al incumplimiento de algún requisito en particular, se podría modificar la configuración para prestar mayor atención sobre la monitorización de ese requisito, asignando, como ejemplo, IDSs muy diversos en su reputación. Este cambio en la configuración también debería tener en cuenta la ausencia de alertas, a fin de detectar MFNs. Si el sistema no recibe alertas durante un tiempo establecido, diferente a la ventana de tiempo anterior, se podría considerar que la probabilidad de MFNs es muy alta. En este caso, esa ventana de tiempo sin recibir un cierto número de alertas se podría definir tanto a nivel global en el sistema de monitorización como a nivel de requisitos en particular, pudiendo dar mayor importancia a los requisitos con mayor impacto en su incumplimiento.

Como último punto, también se puede tener en cuenta el número de IDSs (junto con sus reputaciones) que tienen asignada la monitorización de un requisito. Cuanto mayor sea ese número, menor será la probabilidad de incertidumbre al evaluar sus alertas.

En la Tabla 6.1 se incluye un resumen con las distintas variables utilizadas a lo largo de esta sección, para el cálculo de la diversidad de la confianza. También se aporta una breve descripción de cada una, así como sus valores de inicialización y cálculo.

Variable	Descripción	Inicialización/cálculo
Ecuación (6.2)		
$TD_{\Omega}(R_k)$	Diversidad de la confianza en un cierto requisito R_k exigido por el dominio Ω	Calculada en (6.2) y usada en (6.4) y (6.6)
φ	Medida de dispersión de reputación de los IDSs y riesgo de los requisitos	Elegida por administrador de cada dominio
Ecuación (6.3)		
$\mu_{\Omega}(R_{k,IDS_j})$	Riesgo para Ω si se incumple R_k según los IDSs que lo están monitorizando	Calculada en (6.3) y usada en (6.2) y (6.7)
$Impacto_{\Omega}(R_k)$	Impacto para Ω si se incumpliera R_k	Administradores de Ω
$P_{\Omega}(R_k)$	Propiedades para monitorizar R_k en Ω	Según cada requisito en Ω
Ecuación (6.4)		
TD_{Ω}	Diversidad de la confianza en los IDSs desplegados en el dominio Ω	Calculada en (6.4) y usada en (6.5)
$\theta_A(B)$	Número de elementos de B sobre A	Usada en (6.5), (6.6) y (6.7)
Ecuación (6.5)		
TD_{CAS}	Diversidad de la confianza en el CAS	Calculada en (6.5)
Ecuación (6.6)		
$T(A_{R_k})$	Confianza en la alerta A debido a que se está monitorizando el requisito R_k	Calculada en (6.6)
$Fancy_{\Omega}(A_{R_k})$	Acuerdo de los IDSs de Ω sobre A que informa el incumplimiento de R_k	Calculada en (4.5) y usada también en (6.8)
Ecuación (6.7)		
$Rep(j)^{(t)}$	Valor de reputación de IDS_j sobre su comportamiento en el instante t	Calculada en (6.7) y usada en (6.2)
ς	Peso de la reputación previa de un IDS	Administradores del CAS
Ecuación (6.8)		
$Sat_j(A_k)$	Satisfacción en el comportamiento de IDS_j al publicar la alerta A_k	Calculada en (6.8) y usada en (6.7)
$T(A_k)$	Confianza en la alerta A_k	Conocer si A_k es verdadera
T_{umbral}	Umbral de confianza	Administradores del CAS
$\vartheta_G(A_k)$	Lista de IDSs que han generado A_k	Usada en (4.5) y (5.2)
Ecuación (6.9)		
$\xi(A_k)$	Factor de olvido considerando tiempo transcurrido desde la generación de A_k	Calculada en (6.9) y usada en (6.7)
$\Delta t(A_k)$	Tiempo desde que fue generada A_k	Según recepción de A_k
ν	Importancia de alertas en su recepción	Administradores del CAS

Tabla 6.1: Variables para la monitorización según la diversidad de la confianza

6.5. Resultados experimentales

En esta sección, se presentan y analizan los resultados experimentales adquiridos a partir de la implementación de un prototipo del modelo adaptativo de monitorización presentado en la Sección 6.3, el cual incluye el sistema de gestión de la confianza basado en reputación de la Sección 6.4, objetivo último de este capítulo. Nótese que también se ha implementado en este prototipo el algoritmo de optimización de la Sección 6.3.2. El objetivo principal tras estos experimentos es demostrar cómo las operaciones conjuntas de reputación, diversidad de la confianza y reconfiguración de los distintos IDSs mejoran y refuerzan la calidad de la monitorización, permitiendo i) el incremento de la calidad en las evaluaciones de las alertas recibidas de los IDSs como ii) la adaptación del sistema de monitorización de acuerdo a los valores de esas funciones de confianza.

Durante la configuración del prototipo, para todos los experimentos que se presentan a continuación, se ha definido un sistema compuesto por 20 dominios, 10 requisitos y 500 IDSs. Dentro de este entorno, solamente se ha tenido en cuenta, por simplicidad, una propiedad por cada uno de los 10 requisitos definidos. Finalmente, destacar que la reputación inicial de cada IDS se ha asignado aleatoriamente en el rango [0,1].

Cada una de las simulaciones realizadas contienen las especificaciones del sistema de monitorización y una secuencia de eventos que indican dónde y cuándo deben ocurrir una serie de intentos de ataque, los cuales son inyectados en el sistema cada X tiempo. Los IDSs, como respuesta de sus procesos de detección, responderán dependiendo de sus valores de reputación: enviando las correspondientes alertas asociadas a esos eventos u obviando su ocurrencia. El sistema de monitorización evaluará la confianza sobre las alertas recibidas, actualizará la reputación de los IDSs implicados en la monitorización de los eventos y decidirá, si fuera indispensable, la búsqueda y puesta en marcha de una nueva configuración de monitorización: variando el modelo de despliegue (reallocation) y/o activando o desactivando propiedades de los IDSs (reconfiguration).

6.5.1. Mejora en la evaluación de la confianza

Con este primer experimento se pretende analizar cómo la evaluación de la confianza en todas las alertas recibidas, denotada como $T(A_k)$ y calculada en (6.6), varía cuando se tiene en cuenta la diversidad de la confianza existente entre los IDSs de un modelo de despliegue en concreto. Esto demostraría que los niveles de acuerdo alcanzados entre los IDSs se podrían considerar un factor clave para decidir si las alertas son verdaderas o fraudulentas (si han sido generadas por IDSs honestos o maliciosos, respectivamente). Todas las alertas clasificadas como maliciosas, o fraudulentas, son eliminadas de manera automática para alcanzar una mejor cobertura en la detección de ataques.

Como partida, se aplica la configuración inicial de monitorización de los IDSs de la Figura 6.5a, donde cada fila representa un dominio y cada columna un requisito. Cada celda (i, j) está proporcionalmente coloreada en una escala de grises con la diversidad de la confianza del requisito j demandado por el dominio i, calculada en (6.2), adquiriendo colores grises más oscuros para indicar que la diversidad es más baja.

Todas aquellas celdas que en la Figura 6.5 se encuentran marcadas con un símbolo $\times$ simbolizan que el dominio no necesita protección frente al requisito correspondiente. Para el cálculo de todas esas diversidades de la confianza se ha hecho uso de una función IQR (rango intercuartil) como medida de dispersión, la cual es utilizada en (6.2) bajo el parámetro φ. También se muestran en la Figura 6.5 los resultados tanto de la diversidad de la confianza en cada uno de los dominios, calculada según (6.4), como de la diversidad global en la confianza de todo el CAS, haciendo uso de (6.5).

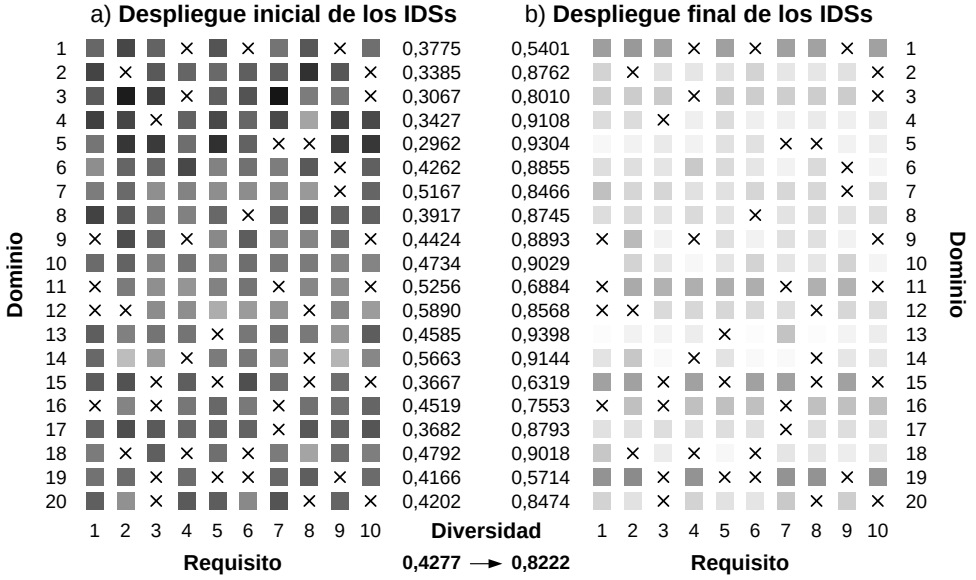


Figura 6.5: Ejemplo para la obtención de un modelo de despliegue más confiable

El modelo de despliegue mostrado en la Figura 6.5a, corresponde a uno ya existente en un momento concreto de la simulación, mientras que la Figura 6.5b expone la mejor configuración de monitorización posible, sugerida por el módulo *Optimización*, después de ejecutar el algoritmo de optimización de la Sección 6.3.2. A raíz de los dos gráficos, se puede constatar que la diversidad global de la confianza crece considerablemente en todos los requisitos y dominios, exhibiendo un incremento global del 92,24 %, pasando de 0,4277 a 0,8222 entre ambos modelos de despliegue de los IDSs.

La siguiente simulación, dentro de este experimento, se focaliza en comprobar cómo la confianza sobre las alertas varía entre las dos configuraciones de monitorización que se muestran en la Figura 6.5, con una diversidad de la confianza bastante dispar. Para su ejecución, se inyectan por separado sobre esas dos configuraciones el mismo conjunto de eventos, los cuales causan la generación de 2000 alertas por parte de los IDSs: 1000 verdaderos positivos y 1000 MFPs. Cada una de las alertas en este conjunto es evaluada, posteriormente, para calcular su confianza global según (6.6).

Destacar en este punto que los IDSs conservan en ambas configuraciones los mismos valores de reputación, a fin de mantener las mismas condiciones experimentales durante las dos simulaciones. Solamente se ha alterado el modelo de despliegue de los IDSs entre los dominios para maximizar la diversidad de la confianza en todos los niveles.

La Tabla 6.2 muestra los resultados en promedio obtenidos de ambas simulaciones. Nótese que se han escogido cuatro medidas de dispersión para calcular la diversidad de la confianza sobre los requisitos que son incumplidos por los eventos inyectados, según el resultado de (6.2), y cuatro operaciones de agregación para el cálculo de la confianza sobre cada una de las alertas, el cual se realiza a través de (6.6).

		Medida de dispersión			
	Operación de agregación	**Diferencia absoluta**	**Desviación estándar**	**Diferencia media**	**Rango intercuartil**
Despliegue inicial IDSs	**Valor mínimo**	0,0663	0,0233	0,0189	0,0273
	Valor máximo	**0,4698**	0,1541	0,1682	0,2892
	Media aritmética	0,1862	0,0576	0,0620	0,0991
	Media armónica	0,1387	0,0378	0,0458	0,0646
Despliegue final IDSs	**Valor mínimo**	0,1129	0,1086	0,0751	0,0973
	Valor máximo	**0,4837**	**0,4748**	**0,4308**	**0,9062**
	Media aritmética	0,2019	**0,3358**	0,2788	**0,4209**
	Media armónica	0,1646	0,1857	0,1594	0,2320

Tabla 6.2: Confianza promedio en las alertas según varias medidas/operaciones

Como se puede apreciar en la Tabla 6.2, la confianza en promedio que el sistema de monitorización obtiene de las mismas alertas varía, sustancialmente, entre el despliegue de los IDSs inicial (Figura 6.5a) y el final (Figura 6.5b), donde se despliega una nueva configuración de monitorización con una diversidad de la confianza más alta, utilizando los mismos IDSs bajo las mismas condiciones experimentales. Estos resultados también varían según la elección particular de las funciones utilizadas en (6.2) y en (6.6).

Una vez calculada la confianza sobre las alertas recibidas, el siguiente punto a tener en cuenta es el umbral de confianza que los administradores del CAS deben establecer para aceptar, o bien rechazar, esas alertas. La definición de este umbral, denotado en la Sección 6.4.2 como T_{umbral}, se debería abordar de forma distinta según las operaciones de agregación y medidas de dispersión escogidas. Como ejemplo, en la Tabla 6.2 se han puesto en negrita las confianzas en las alertas que superen un cierto umbral, establecido como $T_{umbral} = \frac{1}{3}$. Todas las alertas en el despliegue inicial de los IDSs para ese umbral serían rechazadas, clasificadas como FPMs, ya que no son lo suficientemente confiables al tomar la media aritmética como operación de agregación y la desviación estándar e IQR como medidas de dispersión. Se obtendrían mejores resultados con otras funciones como la diferencia absoluta y el valor máximo, a pesar de que éstas no tengan en cuenta la distribución de los datos. Por ejemplo, solamente son necesarios dos IDSs, uno con la reputación más alta y otro con la más baja, para tener una diversidad de la confianza máxima. En cambio, la desviación estándar, diferencia media e IQR suelen proporcionar mejores resultados ya que, además de la dispersión en los datos, también consideran la distribución de los mismos. Con este ejemplo se demuestra la importancia en la elección de estas funciones estadísticas por parte de los administradores.

Por otro lado, los resultados obtenidos con el despliegue final de los IDSs permiten que se consigan resultados más prometedores que los obtenidos con el otro modelo de despliegue, el cual presenta una peor diversidad de la confianza. Este incremento en la evaluación de la confianza sobre las alertas, se traduce en obtener un mejor mecanismo con el que poder discriminar las alertas entre benevolentes o maliciosas. Nótese que ese incremento se encuentra directamente relacionado con los niveles de acuerdo alcanzados por todos aquellos IDSs implicados en la generación de cada alerta.

Los niveles de acuerdo en promedio, definidos en la Sección 6.4.2, entre los distintos IDSs enmarcados en el despliegue inicial de la Figura 6.5a, muestran como resultado los valores 0,3139 y -0,3102 si se intentan evaluar por separado, respectivamente, las alertas benevolentes y las alertas maliciosas. Estos dos resultados no alcanzan niveles mínimos de acuerdo para poder aceptarlos como satisfactorios, más allá de que las confianzas en promedio sobre todas las alertas mostradas en la Tabla 6.2 sean demasiado bajas. Como consecuencia por los resultados de estas evaluaciones, se puede afirmar que existe una alta incertidumbre en saber qué es lo que realmente está ocurriendo en el CAS. Si en su lugar se hubiera tomado el despliegue final de los IDSs, mostrado en la Figura 6.5b, los niveles de acuerdo en promedio se incrementan hasta 0,7253 y -0,7098, para el mismo conjunto anterior de alertas benevolentes y maliciosas, respectivamente. En este nuevo caso, los acuerdos entre los IDSs y la confianza en las alertas sí que ofrecen las garantías suficientes para discriminarlas entre benevolentes o maliciosas.

Las mejoras anteriores, con el despliegue final de los IDSs, son un aumento notable de la calidad en los procesos de detección. Usando la misma simulación anterior, con la inyección de 1000 verdaderos positivos y 1000 MFPs para cada despliegue, la Tabla 6.3 muestra el número de alertas aceptadas o descartadas, cuando $T_{umbral} = 0,5$ para que se considere satisfactorio el nivel de acuerdo entre los IDSs. Es decir, es necesario que $Sat_j(A_{R_k}) \geq 0,5$ en (6.8) para clasificar una alerta como benevolente o maliciosa.

	Alerta benevolente		**Alerta maliciosa**	
	Aceptadas	**Descartadas**	**Aceptadas**	**Descartadas**
Despliegue inicial de IDSs	61	939	942	58
Despliegue final de IDSs	843	157	159	841
	Verdadero Positivo	Falso Negativo Malicioso (MFN)	Falso Positivo Malicioso (MFP)	Falso Negativo

Tabla 6.3: Alertas aceptadas y descartadas según los acuerdos alcanzados

Centrando la atención únicamente en las alertas benevolentes, ya que las maliciosas exhiben unos resultados bastantes similares, se puede observar que de las 1000 alertas recibidas en el despliegue inicial de los IDSs sólo 61 son aceptadas (un 6,1 % sobre el total), mientras que el 93,9 % restante son erróneamente descartadas ya que son alertas benevolentes. Sin embargo, esta mejora es claramente visible si se utiliza el despliegue final de los IDSs. Del 6,1 % anterior se pasa a un 84,3 % de alertas aceptadas, dejando un margen de error del 15,7 % para esta configuración en particular.

6.5.2. Resistencia frente al mal comportamiento de los IDSs

Los experimentos realizados en el apartado anterior han revelado los beneficios en la calidad de la detección cuando se incrementa la diversidad de la confianza. La intención, por tanto, para el siguiente experimento, se focaliza en cuantificar cómo los modelos de despliegue con diversidades de la confianza más altas pueden ser más resistentes frente a la presencia de IDSs maliciosos. Para esas nuevas simulaciones, se han vuelto a tomar las dos configuraciones de monitorización presentadas en la Figura 6.5, las cuales están considerablemente diferenciadas en sus diversidades de la confianza.

En este nuevo experimento, se han llevado a cabo varias simulaciones, incrementado progresivamente el porcentaje de IDSs maliciosos desde el 0 al 100 %. En consecuencia, los IDSs irán produciendo más falsas alertas o maliciosas –MFPs– de manera paulatina. Para cada simulación, se han inyectado los mismos 2000 eventos que en el experimento anterior: la mitad generando 1000 alertas como verdaderos positivos y otras 1000 alertas como MFPs. Una vez recibidas todas estas alertas, el sistema de monitorización calcula la confianza global sobre cada conjunto de alertas a través de (6.6). Al igual que con las simulaciones anteriores, se vuelven a utilizar la media aritmética y el rango intercuartil (IQR) como funciones de agregación y dispersión, respectivamente.

La Figura 6.6 muestra los resultados como *box plots* con la distribución global de la confianza sobre las alertas recibidas, incluyendo los valores promedios de la confianza –líneas conectando los box plots– para las dos configuraciones de monitorización. Los resultados en azul corresponden al despliegue inicial de los IDSs, con un índice bajo de la diversidad de la confianza, mientras que en rojo se representan los resultados para el despliegue final, con una diversidad considerablemente más alta.

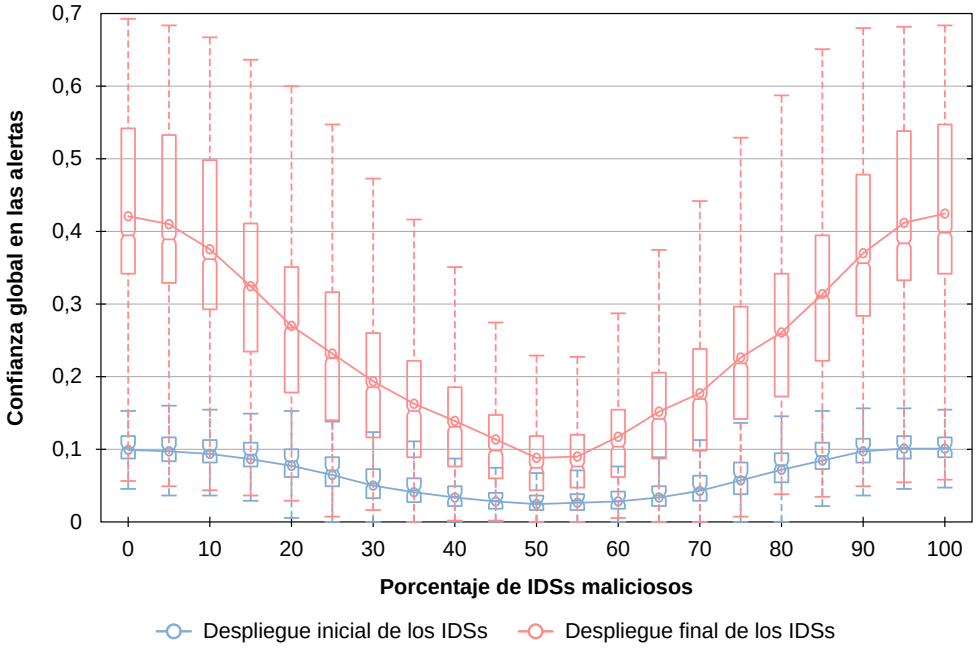


Figura 6.6: Confianza en las alertas según el porcentaje de IDSs maliciosos

Los resultados muestran una conducta común para los dos modelos de despliegue: la confianza sobre las alertas decrece conforme aumenta el número de IDSs maliciosos, alcanzando un valor mínimo cuando los IDSs maliciosos son alrededor del 50-55 %. La reputación de los IDSs maliciosos se equipara a la de los honestos, alcanzando una incertidumbre casi total. A partir de ahí, la actitud maliciosa de los IDSs prevalece sobre la de los honestos, y éstos comienzan a ser injustamente acusados (por la mayoría) de una mala conducta. Inevitablemente, este cambio en la mayoría afecta a la reputación de todos los IDSs, viéndose aumentada la de los maliciosos y reducida la de los honestos. En consecuencia, la confianza en las alertas vuelve a incrementar poco a poco debido a este cambio. Los IDSs maliciosos empiezan a engañar al sistema de monitorización en su percepción de lo que está pasando, descartando alertas benevolentes y aceptando las maliciosas como verdaderas. Se podría afirmar que los IDSs maliciosos han pasado a "tomar el control" de los procesos de decisión sobre las alertas recibidas.

Comparando los resultados de la Figura 6.6, se pueden observar los beneficios de la diversidad de la confianza. Primero, la confianza en las alertas aumenta notablemente utilizando una configuración de monitorización más óptima en términos de diversidad, como también se ha comprobado con el experimento anterior. En segundo lugar, las variaciones en la confianza entre los dos modelos de despliegue son mucho más abultadas conforme el número de IDSs maliciosos va en aumento. El descenso, o bien el ascenso, en la curvatura de la confianza es bastante más pronunciada entre un porcentaje de IDSs maliciosos y sus colindantes. Por ejemplo, si se considera el cambio de IDSs maliciosos entre el 10 y el 15 % en la Figura 6.6, la confianza en promedio en las alertas pasa desde 0,0936 a 0,0864 en el despliegue inicial de los IDSs, mientras que esta variación para la confianza cae de 0,3755 a 0,3245 con el despliegue final de los IDSs; una diferencia en la confianza de 0,0072 en el primer caso y de 0,051 en el segundo.

Esta última diferencia parece muy pequeña en cantidad, aunque lo suficientemente abultada –un 4,38 % de mejora en la confianza entre los dos modelos de despliegue– si se considera que la variación en el porcentaje de los IDSs maliciosos es de tan sólo un 5 %. En términos numéricos, ese 5 % de IDSs maliciosos correspondería a pasar de 450 IDSs con un comportamiento malicioso a 425, entre los 500 que hay desplegados en total. Sin embargo, si se considera un rango de IDSs maliciosos más alto, por ejemplo entre un 0 y un 25 % (de 0 a 125 IDSs maliciosos), las diferencias son todavía más significativas. En este caso particular, la confianza variaría de 0,0991 a 0,0645 (diferencia de 0,0346) para el despliegue inicial de IDSs, mientras que esta confianza en promedio caería desde 0,4209 a 0,2318 (diferencia de 0,1891) con el modelo de despliegue final.

Analizando los resultados, el despliegue final de los IDSs puede jugar un importante papel en la detección de IDSs maliciosos. El sistema podría percibir que una parte de la red de detección está siendo comprometida poco a poco, viendo que la confianza en las alertas disminuye progresivamente. Este hecho es más difícil detectar con el despliegue inicial de los IDSs, haciendo que un atacante pueda comprometer los IDSs uno a uno sin cambios notables en la confianza global. Por tanto, el modelo adaptativo basado en la diversidad de la confianza presentado en este capítulo se erige como un mecanismo de detección de IDSs maliciosos, mediante la inspección de sus comportamientos.

6.5.3. Evaluación de los niveles de acuerdo entre los IDSs

A través de este último experimento, se plantea analizar cómo influye la diversidad de la confianza en los niveles de acuerdo alcanzados por los IDSs en la evaluación de las alertas, sabiendo que sus comportamientos pueden variar bruscamente en el tiempo. En estas nuevas simulaciones, se vuelven a utilizar los dos modelos de despliegue usados en los experimentos anteriores, ilustrados gráficamente en la Figura 6.5. Destacar una vez más, que los valores de reputación de los IDSs son los mismos para ambos modelos de despliegue, a fin de conservar las mismas condiciones experimentales.

En este experimento se han ejecutado varias simulaciones, aumentando el porcentaje de IDSs maliciosos progresivamente e inyectando, para cada caso, los eventos necesarios para que se generen 1000 alertas benevolentes –verdaderos positivos. Nótese que estos eventos son los mismos que se han utilizado en los experimentos anteriores, aunque en estas simulaciones no se han inyectado aquellos eventos que generan alertas maliciosas, ya que siguen un patrón muy similar a las benevolentes. En un gráfico de líneas, como el presentado más abajo, se vería como un volteo vertical de los resultados.

La Figura 6.7 muestra los resultados para ambas configuraciones de monitorización.

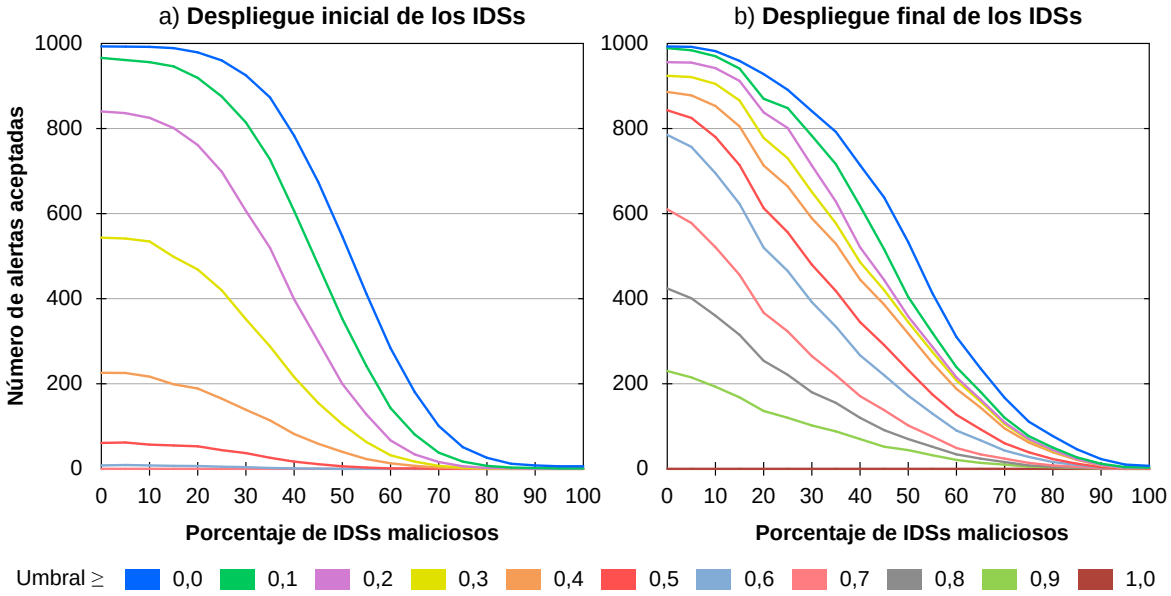


Figura 6.7: Variación en los acuerdos al aumentar el número de IDSs maliciosos

En ambos gráficos, se muestra la evolución en las alertas aceptadas como confiables, y que el sistema adopta como benevolentes, conforme el porcentaje de IDSs maliciosos aumenta hasta el 100 %. Cada línea refleja el número de alertas que son aceptadas como benevolentes según distintos valores umbral en la confianza: incrementando T_{umbral} poco a poco para obligar a que los acuerdos entre los IDSs tengan que ser más mayoritarios para poder aceptar sus alertas como confiables. Por motivos de claridad, en la Figura 6.7 no se incluyen las alertas descartadas, ya que su número es el complementario de las aceptadas. Sus resultados se dibujarían simplemente como un volteo horizontal.

Como principal conclusión, una vez analizados los resultados experimentales que se han presentado anteriormente, se puede afirmar que la diversidad de la confianza, como una métrica heurística que mide la calidad de cualquier modelo de despliegue, permite obtener en cada momento el mejor modelo de despliegue estratégico de los IDSs con el que obtener una mejor evidencia de las alertas que éstos producen y, en consecuencia, mejorar los procesos de detección de ataques. De forma adicional, el uso de la diversidad de la confianza también permite identificar, con mayor claridad y precisión, cuándo los IDSs comienzan a exponer comportamientos maliciosos en sus actos de detección.

6.6. Conclusiones del capítulo

La gestión del alto número de alertas que pueden ser generadas por múltiples IDSs, desplegados a lo largo de una red de monitorización, está lejos de ser una tarea sencilla de afrontar. Esta situación todavía supone un problema de mayor calado si las alertas, como medio para detectar ataques, provienen de diferentes organizaciones que deciden compartir sus alertas con un afán común en la detección de ataques distribuidos, pero que no son suficientemente confiables entre sí para aceptar las alertas de sus homólogos como información de detección verdadera. Es necesario, por tanto, el incremento de la calidad en las evaluaciones sobre esas alertas con el que, a la misma vez, proporcionar la base para que la toma de decisiones en la detección sea resistente frente al envío de alertas fraudulentas por parte de IDSs con un comportamiento malicioso.

El concepto en la diversidad de la confianza, que se ha introducido en este capítulo, ha demostrado ser una métrica heurística esencial para la obtención de configuraciones óptimas de monitorización. El sistema colaborativo de alertas (CAS), haciendo uso de la diversidad de la confianza, permite determinar si cualquier alerta es o no confiable (si representa hechos verdaderos acontecidos en la red de detección). A la misma vez, este concepto también se puede aprovechar para revaluar la percepción que tiene el sistema de monitorización sobre la honestidad de los IDSs. El objetivo se centra en poder tomar mejores decisiones en el futuro sobre la detección de ataques a través de la elección, y su posterior puesta en marcha, de mejores configuraciones de monitorización aplicando modelos óptimos de despliegue de los IDSs. Para la aplicación de estos modelos, se han propuesto dos enfoques: reubicar los IDSs de la infraestructura desde su posición actual a otra nueva y/o reconfigurar sus capacidades de detección por otras distintas.

Las pruebas hechas con simulaciones experimentales, en un entorno multidominio de simulación con un alto número de dominios de seguridad o CIDNs, han demostrado que el uso de la diversidad de la confianza puede desempeñar un papel muy importante en la mejora de la calidad en la detección colaborativa de ataques. En estos experimentos, se han comparado dos modelos de despliegue de los IDSs, uno inicial ya existente en un momento determinado de la simulación y otro final más óptimo con una diversidad de la confianza entre los valores de reputación de los IDSs mucho más alta. La diferencia de diversidad entre ambos modelos permite que alertas verdaderas que eran descartadas en el modelo inicial, suponiendo que eran fraudulentas, pasen a ser aceptadas.

Capítulo 7

Conclusions and future works

The objective behind this last chapter is to uncover the most important outcomes achieved by the research work developed and presented along this doctoral dissertation. Also, as an objective of the chapter at hand, a last section is included to define some possible future works that are currently emerging with growing interest, in order to continue and/or complement the research work herein presented.

The methodology under which this chapter focuses on is, primarily, performing an in depth revision related to all the problems that have arisen in detection, as well as a series of solutions to build a collaborative alert system capable of detecting distributed attacks. Subsequently, a set of research directions as future works are promoted, which have not been part of the definition of the objectives set in this thesis dissertation. Yet, these research lines, which are stated below, have a significant relevance in the context of a collaborative alert system, so as to reach an even more reliable and accurate system in detecting potential distributed attacks.

7.1. Conclusions

Despite the massive effort made in the last decades to detect intrusions or attacks, especially following the emergence of new potential attacks with a wider scope (more distributed) in their purpose, a number of research lines have been neglected with which the detection systems are able to achieve a minimum level of security. In this context, the main objective of the thesis dissertation at hand is to define, design and start-up a collaborative alert system capable of detecting distributed attacks through securely sharing detection information in multi-domain environments. This security is related to two basic pillars for a collaborative system that aims to detect distributed attacks. On the one hand, the exchange of alerts that Intrusion Detection Systems (IDS) are able to produce, which need to be exchanged in an effort to detect distributed attacks. On the other hand, the content of the alerts itself, as IDSs could create alerts from scratch notifying security policy breaches, when the facts being reported do not correspond to reality. These fraudulent alerts may be due to malicious acts, possibly as a consequence that the IDSs reporting such alerts have been compromised by an attacker.

Regarding the first pillar, related to the support of a Collaborative Alert System (CAS) to detect distributed attacks, this relies on the sharing of detection alerts. The protection of the communication channels is a mandatory requirement, in order the Collaborative Intrusion Detection Networks (CIDN) know for sure that the alerts being exchanged for detecting attacks have not been altered in content during transmission and have not been sent by non-legitimate IDSs of their CIDN. That is to say, the main objective in this protection is to ensure the preservation of authenticity, confidentiality and integrity of the alerts sent through transmission channels. Building this security framework in an intra-domain environment, for a single CIDN, has been seen in recent years as a challenge with a fairly simple solution, through the deployment and start-up of a Public Key Infrastructure (PKI). Its certification authority allows managing of all the security within the same CIDN, thereby enabling the required (mandatory) protection in security for the different communications between the IDSs.

Despite the number of works that have been proposed so far on security about the communications at intra-domain level, the trust models which are referred to are no longer feasible in new advanced certification environments, where communications in an inter-domain level take on new significance. The CIDNs of the different security and/or administrative domains forming the CAS need to be protected under the same security framework commented earlier, but now aimed at reaching the same protection of the communication channels at multi-domain level.

With that premise in mind, a solution based on PKI technology has been designed with which to build new advanced trust models in multi-domain environments. Its main objective is to securely exchange the knowledge base about the alerts internally generated by each of the IDSs. These IDSs will produce such alerts within their own CIDN. This information sharing is the key factor to generate the global knowledge base at multi-domain level, with the aim of detecting distributed attacks. The corresponding findings and the outcomes from the research work made regarding this point have led to the publication of the papers presented in [32, 33].

The design and development of the solution herein proposed is able to establish the required trust relationships between the different CIDNs, this being the necessary basis for building a PKI federation between all parties involved in the CAS. Thereby, the proper definition of all the X.509 certificate extensions of each root CA forming the PKI, each managing the security in a given CIDN, is a mandatory requirement in terms of interoperability for its proper operation. In this context, a certification path building and validation algorithm has been designed and proposed, which is the main basis for the implementation and start-up of a Validation Service. Using this service, the IDSs are able to validate the cryptographic credentials of any other IDS, so as to accept that IDS as a legitimate entity belonging to the CAS. This process must be performed before establishing a secure channel of communication between them in order to share any type of detection information, such as the alerts generated by each of them.

The main conclusions that were drawn from the two solutions summarised earlier, which are based on PKI technology, are the ones presented below. Such conclusions are presented in summary form.

- The definition that has been proposed for building any PKI federation allows a proper interoperability between all the security domains which shape the CAS, by establishing trust relationships between each one of the domains representing a given CIDN. These relationships are based on advanced cross-certification models, so allowing the creation of each of these relationships through peer-to-peer links or by means of a neutral Bridge CA (BCA).

  This interoperability has been achieved by the proper definition of each one of the X.509 certificate extensions, under certain requirements. Each of the 17 possible extensions that can be defined in a certificate has been marked as a mandatory, recommended, optional or not applicable requirement. An incorrect definition of any extension would imply that neither the suitable building of the PKI federation would not be possible nor the appropriate operation of the Validation Service.

- The design of the Validation Service, including the certification path building and validation algorithm that has been proposed, allows the validation of any X.509 certificate in a multi-domain environment. Thus successfully achieving that the IDSs can validate the legitimacy of other IDSs before establishing the expected secure communication channels between them for exchanging detection alerts.

  The certification path building process has been simplified to a search algorithm through the certification tree that shapes the PKI federation, where the candidate certification paths are built in an incremental way as the algorithm finds out new security domains along its search. All the cryptographic material –certificates and revocation lists– associated with each domain will be gathered depending on whether the algorithm is searching within a hierarchical model or in one based on cross-certification (peer-to-peer or BCA).

- The validation of the two solutions enumerated earlier, which are based on PKI technology, have been carried out through intensive simulations in a controlled lab environment. All these experiments have been transferred subsequently to a multi-domain scenario conducted on the FBCA, a real certification environment in production. It is worthy to emphasise that the use of the two solutions has also been validated by means of their deployment and their start-up on each (regional, national and European) of the research projects commented in Section 1.7, since the proposed solutions are a key requirement for all the communications in any research project aligned with security concepts.

The UMU-PKIv6 software has been used for both building the PKI federation and starting-up subsequently the Validation Service designed. Its functionality on managing security in any intra-domain environment, which it was already being provided by the UMU-PKIv6, has been updated towards a new solution with which to manage security at a multi-domain level. This new functionality is the indispensable basis for the design of a collaborative alert system, which is capable of detecting distributed attacks under the required security framework. This is the main objective for the information sharing between all the actors being carried out safely and successfully.

Once assured that the communication channels are protected for sharing alerts, how to deploy the IDSs is a must to guarantee a scalable CAS for detecting distributed attacks. It has been proposed that the detection processes distribution is accomplished by a partially-decentralised placement model, in order to overcome the scalability and overhead in communication drawbacks innate in the most common models used today: centralised or fully decentralised. The CAS design has been performed by decomposing it in several heterogeneous CIDNs, which correspond to the autonomous systems with monitoring capabilities to detect breaches of the security policies defined by the CIDN, depending on the assets being protected in its narrowed area of operation.

Each CIDN is in charge of building an internal knowledge base from the alerts that each of its IDSs is capable of producing and sharing. Merging of all that self-knowledge is the source for building up the desired global knowledge base at the CAS level, with which to achieve the detection of distributed attacks. However, as defined earlier in the second basic pillar on security management, the fraudulent alerts (false positives) have to be discarded for a proper decision-making process about a possible attack, whether local or distributed. To this end, the designed trust management systems based on reputation allow identifying ill-intentioned attitudes when exchanging fraudulent alerts, under local or distributed schemes: between all the IDSs within a CIDN (intra-domain model) and between the CIDNS shaping the CAS (inter-domain model). As findings and outcomes from the research work uncovered regarding this point, several papers have been published, which can be found in [36, 37, 38, 39].

The main conclusions that were drawn from the two trust models, using reputation as a proxy for trust and defining one at intra-domain and another at inter-domain level, are the ones given below in summary form.

- The definition and design of the proposed intra-domain trust model based on reputation is capable of modelling the behaviour of any of the IDSs deployed in a CIDN, in order to discard their alerts when the IDS does not prove (through its reputation score) the required trust for considering the alerts as actual events observed in reality. When the IDS does not own a sufficient reputation, the system considers that this IDS is susceptible to be malicious in its actions because it has been compromised by an attacker. This attacker attempts to disturb the detection processes of the CIDN by sending out false positives.

  The recommendations (opinions) provided by the IDSs on a given IDS, internally in a CIDN, are essential information to calculate its reputation score. All these recommendations will correspond to the satisfaction of any IDS on the alerts that the IDS being assessed has generated and published in the past.

- Regarding the design of a CIDN, a set of IDSs have been chosen as the maximum representatives within that CIDN, elected for having the highest reputation score in accordance with all the IDSs deployed there. Therefore, these IDSs will be the most trustworthy entities of their CIDN. This group of experts is called as a Wise Committee (WC), whose main purpose is to give consent to the publication of the alerts once the IDS which reported them has been assessed.

Amongst all the IDSs belonging to a WC, one of them has been chosen for being the head or leader of the WC and, consequently, of the entire CIDN. This election will be given by the IDS with the highest reputation score within the WC. This WC leader is in charge of taking the final decision about which reputation score should be assigned to each IDS reporting alerts, and therefore whether its alerts can be considered as true or false. Furthermore, the WC leader is also in charge of sharing the local knowledge base built in its CIDN with the homologous leaders of other CIDNs, with which it already has a trust relationship. This information sharing is the root for building the desire global knowledge base, required for the detection of distributed attacks at multi-domain level.

- Similarly to the previous case, an inter-domain trust model based on reputation has been defined to model the behaviour of any CIDN shaping the CAS. Using this system, each CIDN is capable of discarding the fraudulent alerts of its detection processes, which have been gathered from suspicious CIDNs in behaviour. That is, coming from CIDNs with a low reputation from the perspective of the receiver. Recommendations requested to other trusted CIDNs are also taken into account, in order to compute the reputation score of a given CIDN.

  Within the inter-domain context, a reputation-based trust model has also been proposed aimed at assigning an initial reputation score to any type of detection unit. These units can be IDSs belonging to the infrastructure, CIDNs expecting to collaborate with each other to detect distributed attacks or IDSs provided by end users. This trust model proposes a solution for solving two common problems of any collaborative system using reputation: cold-start and bootstrapping.

- The two proposed trust models based on reputation have been validated through intense simulations and experimental tests, discussing their results in each case.

For all the previous reputation systems, the most appropriate security models have been defined for exchanging the required detection information (recommendations and alerts) in the most secure possible way.

The trust management through reputation has been confirmed as a cross procedure for accepting the alerts generated by the different IDSs within their detection networks. Additionally, reputation can also be used to provide the CAS a certain dynamism for improving its detection processes. The proposed solution allows the collaborative alert system to maximise the decision-making process quality in detecting a potential attack, which is based on reconfiguring the infrastructure IDSs, switching the placement model for a new one, or establishing other monitoring policies on their detection capabilities. Both mechanisms are subjected to the reputation scores of the IDSs forming the CAS. The results related to this adaptive system, with which to dynamically reconfigure the IDSs, can be found as a research paper published in [40].

The main conclusions in relation to maximising the detection information quality are presented below, aimed at ensuring a proper decision-making process in detecting attacks by taking the alerts generated by the IDSs as an input.

- The designed adaptive model has demonstrated that trust diversity, based on the IDSs' reputation scores, is an important factor that should be taken into account to maximise the confidence on the detection processes of the IDSs. This entails the constant deployment of the most optimum placement model. The application of this model has been designed at the two levels that a given CAS defines: at intra-domain level with the IDSs of a same CIDN and at global level, considering all the IDSs from the perspective of the CAS.

  The diversity in the reputation scores of the IDSs has been proposed as a heuristic metric, with which the quality on the detection processes of any placement model might be checked. Additionally, but by no means least, trust diversity can also be used for deciding at what time (when) a new configuration has to be deployed with higher levels of trust on the alerts that are being produced.

- The proposed model has been largely validated by experimental tests, which have shown that trust diversity helps in deriving optimal monitoring configurations to improve the quality of collaborative attack detection.

As a final conclusion, we can confirm that the research work herein presented shows a significant advance on current collaborative systems for detecting distributed attacks. This enables the proposed collaborative alert system to improve the problems related to its scalability, robustness and reliability. Firstly, the scalability issues for gathering and analysing huge amounts of alerts (in a real-time fashion) have been achieved by deploying the IDSs with a placement model based on a partially-decentralised scheme. In this context, the best placement model is deployed at any time by using the proposed adaptive system, which is able to maximise the alerts quality with the deployment of the best possible monitoring configuration.

On the other hand, regarding the robustness and reliability issues, the reputation-based trust management system proposed for modelling the IDSs' behaviour is claimed as a must mechanism in assessing security alerts. Using this model, the alerts exchanged between the IDSs can be assessed before endorsing them as true, depending always on the reputation score of the IDSs reporting such alerts. Therefore, fraudulent alerts sent out by ill-intended IDSs will be discarded from the IDSs' detection processes to avoid confusion when a potential distributed attack is being exploited.

We can finally bear out that the proposed Collaborative Alert System for detecting distributed attacks, with the support of the secure exchange of the alerts produced by reliable IDSs in multi-domain environments, has achieved the expected success.

7.2. Future work

As with any research work, the proposals made in this doctoral dissertation open new research fields that may be addressed as future work. On this matter, and following the structure proposed in different chapters, possible research lines are drawn up below for each one, but separated depending on the specific objectives they attempt to solve.

With respect to the protection of the communications between the components of a CAS, or within a CIDN, by using the solutions based on public key cryptography, a new research area is opened to improve the performance when building certification paths. Specifically, the use of optimisation mechanisms is required with which the Service Validation can choose the best certification path when the building process uncovers several alternatives. For instance, when the Validation Service retrieves several cross-certificates and the algorithm must decide the branch from which to continue the search. Another classic optimisation approach is the use of a local cache, where the Validation Service stores all the needed information to speed up the building process in future requests. Yet, this optimisation has a serious drawback. The longer the information is not updated, the more likely invalid certification paths will be returned. Despite all the optimisations that the Validation Service could support, it is also important to deploy redundancy and load balancing mechanisms in organisations of medium/large size, in order not to overload the service with multiple concurrent requests.

Regarding the trust management models based on reputation, some research efforts are required for examination and treatment. First, a proper modelling of the differences in the recommendations that can be gathered to assess the same IDS or the same CIDN is necessary. It may be the case that several IDSs, with the same behaviour and same detection capabilities, provide disparate recommendations in assessing the same source. This fact is recognised in literature as *subjectivity difference*. Secondly, it is also required to incorporate the detection capabilities definition in assessing the satisfaction on any alert. This satisfaction assessment can considerably reduce the differences in the above recommendations. On the other hand, it is also considered desirable to provide a better model in the WC election, which does not allow very frequent changes of its members due to slight changes in reputation. As a following step, regarding the trust systems based on reputation, the definition of incentive-based policies has been left as future work. These policies are necessary to encourage the collaboration of the mobile IDSs, belonging to roaming users. Without these incentives, users with mobile IDSs will not find the need to share their alerts. Finally, the definition on how to carry out the trust relationships between the CIDNs has also been left as future line of research.

As another important piece of research, it is also proposed that the management of the privacy concerns is raised when sharing the different alerts for identifying attacks. This is a quite relevant matter to achieve the required security framework between all actors of the CAS. However, this issue is still even more important in any multi-domain scenario, such as the collaborative systems aimed at detecting the distributed attacks. Organisations have to control which detection information can, or do not want to, share with others, because it can be considered as sensitive information for being shared. This information could be susceptible of being found out by a given attacker, e.g. the assets' vulnerabilities, which could be used in future attacks against the organisation. In this sense, future works should provide useful solutions to maintain organisations' privacy, but bearing in mind that the detection ratio is not diminished.

Finally, it is worth to mention that the research work herein presented is totally aligned with the challenges in the field of cyber security. The proliferation of new cyber

attacks, thefts, threats and other potential cyber crimes are doing that are considered new research lines, which need to be tackled in order to increase trustworthiness with respect to the ICT systems. All these challenges deal with cyber security threats that appear everyday incessantly, which can be grouped and analysed into four main topics, namely: dynamic risk management, attack and defence graphs, incidents correlation and information sharing. The main idea behind these topics would provide a system with dynamic risk management over large systems, using adaptive defence graphs with privacy-preserving incidents correlation and encouraging information sharing.

Bibliografía

[1] D.E. Denning. *Information warfare and security*. Addison-Wesley Professional, 1998.

[2] J.M. Kizza. Cyber crimes and hackers. In *A Guide to Computer Network Security*, Computer Communications and Networks, pp. 107–131, 2009.

[3] B. Kashyap and S.K. Jena. DDoS attack detection and attacker identification. *International Journal of Computer Applications*, 42(1):27–33, 2012.

[4] J. McHugh, A. Christie, and J. Allen. Defending yourself: The role of intrusion detection systems. *IEEE Software*, 17(5):42–51, 2000.

[5] T. Verwoerd and R. Hunt. Intrusion detection techniques and approaches. *Computer Communications*, 25(15):1356–1365, 2002.

[6] C.V. Zhou, C. Leckie, and S. Karunasekera. A survey of coordinated attacks and collaborative intrusion detection. *Computers & Security*, 29(1):124–140, 2010.

[7] S.X. Wu and W. Banzhaf. The use of computational intelligence in intrusion detection systems: A review. *Applied Soft Computing*, 10(1):1–35, 2010.

[8] S. Axelsson. The base-rate fallacy and the difficulty of intrusion detection. *ACM Transactions on Information and System Security*, 3(3):186–205, 2000.

[9] K. Julisch. Clustering intrusion detection alarms to support root cause analysis. *ACM Transactions on Information and System Security*, 6(4):443–471, 2003.

[10] Sourcefire, Inc. Snort: An open source network intrusion prevention and detection system. `http://snort.org`.

[11] G.C. Tjhai, M. Papadaki, S. Furnell, and N.L. Clarke. Investigating the problem of IDS false alarms: An experimental study using Snort. In *SEC'08: Proceedings of the IFIP TC-11 23rd International Information Security Conference*, pp. 253–267, September 2008.

[12] R. Perdisci, G. Giacinto, and F. Roli. Alarm clustering for intrusion detection systems in computer networks. *Engineering Applications of Artificial Intelligence*, 19(4):429–438, 2006.

[13] E. Vasilomanolakis, S. Karuppayah, M. Mühlhäuser, and M. Fischer. Taxonomy and survey of collaborative intrusion detection. *ACM Computing Surveys*, 47(4):55:1–55:33, 2015.

[14] D. Xu and P. Ning. Correlation analysis of intrusion alerts. In *Intrusion Detection Systems*, volume 38 of *Advances in Information Security*, pp. 65–92, January 2008.

[15] S. Salah, G. Maciá-Fernández, and J.E. Díaz-Verdejo. A model-based survey of alert correlation techniques. *Computer Networks*, 57(5):1289–1317, 2013.

[16] M.E. Locasto, J.J. Parekh, S. Stolfo, A.D. Keromytis, T. Malkin, and V. Misra. Collaborative distributed intrusion detection. Technical Report CUCS-012-04, Department of Computer Science, Columbia University, 2004.

[17] A. Menezes, P. van Oorschot, and S. Vanstone. *Handbook of applied cryptography*. CRC Press, 1997.

[18] R.K.C. Chang. Defending against flooding-based distributed denial-of-service attacks: A tutorial. *IEEE Communications Magazine*, 40(10):42–51, 2002.

[19] H. Beitollahi and G. Deconinck. Analyzing well-known countermeasures against distributed denial of service attacks. *Computer Communications*, 35(11):1312–1332, 2012.

[20] V. Mateos, V.A. Villagrá, F. Romero, and J. Berrocal. Definition of response metrics for an ontology-based automated intrusion response systems. *Computers & Electrical Engineering*, 38(5):1102–1114, 2012.

[21] A. Stavrou and A.D. Keromytis. Systems and methods for inhibiting attacks with a network. United States Patent 8631484, January 2014.

[22] G. Martínez Pérez, F.J. García Clemente, M. Gil Pérez, and A.F. Gómez Skarmeta. Secure overlay networks for federated service provision and management. *Computers & Electrical Engineering*, 34(3):173–191, 2008.

[23] C. Adams and S. Lloyd. *Understanding PKI: Concepts, standards, and deployment considerations (2nd Edition)*. Addison-Wesley Longman Publishing Co., Inc., 2002.

[24] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W.T. Polk. Internet X.509 public key infrastructure certificate and certificate revocation list (CRL) profile. IETF Request for Comments 5280, May 2008.

[25] International Telecommunication Union (ITU). The directory: Public-key and attribute certificate frameworks. International Standard ISO/IEC 9594-8, ITU-T Recommendation X.509, August 2008.

[26] J. Sabater and C. Sierra. Review on computational trust and reputation models. *Artificial Intelligence Review*, 24(1):33–60, 2005.

[27] S. Marti. Trust and reputation in peer-to-peer networks. PhD dissertation, Stanford University, USA, 2005.

[28] J. Masthoff. Computationally modelling trust: An exploration. In *SociUM'07: Proceedings of the SociUM Workshop associated with the User Modeling Conference*, June 2007.

[29] C.C. Chen, Y.-H. Wan, M.-C. Chung, and Y.-C. Sun. An effective recommendation method for cold start new users using trust and distrust networks. *Information Sciences*, 224:19–36, 2013.

[30] F. Skopik, D. Schall, and S. Dustdar. Start trusting strangers? Bootstrapping and prediction of trust. In *WISE'09: Proceedings of the 10th International Conference on Web Information Systems Engineering*, volume 5802 of *Lecture Notes in Computer Science*, pp. 275–289, October 2009.

[31] H. Chen. Self-configuration framework for networked systems and applications. PhD dissertation, The University of Arizona, USA, 2008.

[32] G. López Millán, M. Gil Pérez, G. Martínez Pérez, and A.F. Gómez Skarmeta. PKI-based trust management in inter-domain scenarios. *Computers & Security*, 29(2):278–290, 2010.

[33] A. Ruiz Martínez, D. Sánchez Martínez, C.I. Marín López, M. Gil Pérez, and A.F. Gómez Skarmeta. An advanced certificate validation service and architecture based on XKMS. *Software: Practice and Experience*, 41(3):209–236, 2011.

[34] W.E. Burr. Public Key Infrastructure (PKI). Technical specifications: Part A - Technical concept of operations. Federal Public Key Infrastructure Technical Working Group, NIST Working Draft TWG-98-59, September 1998.

[35] IDABC European eGovernment Services. European federated validation service study: Solution profile - U.S. Federal Bridge Certification Authority (FBCA), July 2009.

[36] M. Gil Pérez, F. Gómez Mármol, G. Martínez Pérez, and A.F. Skarmeta Gómez. RepCIDN: A reputation-based collaborative intrusion detection network to lessen the impact of malicious alarms. *Journal of Network and Systems Management*, 21(1):128–167, 2013.

[37] M. Gil Pérez, V. Mateos Lanchas, D. Fernández Cambronero, G. Martínez Pérez, and V.A. Villagrá. RECLAMO: Virtual and collaborative honeynets based on trust management and autonomous systems applied to intrusion management.

In *CISIS'13: Proceedings of the 7th International Conference on Complex, Intelligent, and Software Intensive Systems*, pp. 219–227, July 2013.

[38] M. Gil Pérez, F. Gómez Mármol, G. Martínez Pérez, and A.F. Gómez Skarmeta. Mobility in collaborative alert systems: Building trust through reputation. In *WCNS'11: Workshop on Wireless Cooperative Network Security*, volume 6827 of *Lecture Notes in Computer Science*, pp. 251–262, May 2011.

[39] M. Gil Pérez, F. Gómez Mármol, G. Martínez Pérez, and A.F. Skarmeta Gómez. Building a reputation-based bootstrapping mechanism for newcomers in collaborative alert systems. *Journal of Computer and System Sciences, Special Issue on Wireless Networks Intrusion*, 80(3):571–590, 2014.

[40] M. Gil Pérez, J.E. Tapiador, J.A. Clark, G. Martínez Pérez, and A.F. Skarmeta Gómez. Trustworthy placements: Improving quality and resilience in collaborative attack detection. *Computer Networks*, 58:70–86, 2014.

[41] P. Hallam-Baker and S.H. Mysore (editors). XML key management specification (XKMS 2.0). W3C Recommendation, June 2005.

[42] EU-IST FP6 SEINIT project (*Security Expert INITiative*). `http://isoc.org/seinit/portal`.

[43] EU-IST FP6 DAIDALOS project (*Designing Advanced network Interfaces for the Delivery and Administration of Location independent, Optimised personal Services*). `http://ist-daidalos.org`.

[44] EU-IST FP6 DESEREC project (*DEpendability and Security by Enhanced REConfigurability*). `http://deserec.eu`.

[45] The RECLAMO project (*Virtual and Collaborative Honeynets based on Trust Management and Autonomous Systems applied to Intrusion Management*). `http://reclamo.inf.um.es`.

[46] J.L. Hernandez-Ardieta, J.E. Tapiador, and G. Suarez-Tangil. Information sharing models for cooperative cyber defence. In *CyCon'13: Proceedings of the 5th International Conference on Cyber Conflict*, pp. 1–28, June 2013.

[47] N. Hubballi and V. Suryanarayanan. False alarm minimization techniques in signature-based intrusion detection systems: A survey. *Computer Communications*, 49:1–17, 2014.

[48] H. Chivers, J.A. Clark, P. Nobles, S.A. Shaikh, and H. Chen. Knowing who to watch: Identifying attackers whose actions are hidden within false alarms and background noise. *Information Systems Frontiers*, 15(1):17–34, 2013.

[49] The VeriSign iDefense Intelligence Team. Intrusion detection system (IDS) evasion. Technical report, An iDefense Security Report, May 2006.

[50] T.H. Ptacek and T.N. Newsham. Insertion, evasion, and denial of service: Eluding network intrusion detection. Technical Report ADA391565, Secure Networks Inc., January 1998.

[51] J. Esler. Stream5: A TCP and UDP reassembly module for Snort. December 2012. `http://manual.snort.org/node66.html`.

[52] J.J. Martínez Molina, M.A. Hernández Ruiz, M. Gil Pérez, G. Martínez Pérez, and A.F. Gómez Skarmeta. Event-driven architecture based on patterns for detecting complex attacks. *International Journal of Critical Computer-based Systems*, 1(4):283–309, 2010.

[53] H. Du. Probabilistic modeling and inference for obfuscated network attack sequences. PhD dissertation, Rochester Institute of Technology, USA, 2014.

[54] The Mitre Corporation. Common vulnerabilities and exposures CVE-2009-3641. October 2009. `http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2009-3641`.

[55] J.-T. Oh, S.-K. Park, J.-S. Jang, and Y.-H. Jeon. Detection of DDoS and IDS evasion attacks in a high-speed networks environment. *International Journal of Computer Science and Network Security*, 7(6):124–131, 2007.

[56] G.P. Spathoulas and S.K. Katsikas. Reducing false positives in intrusion detection systems. *Computers & Security*, 29(1):35–44, 2010.

[57] G.C. Tjhai, S.M. Furnell, M. Papadaki, and N.L. Clarke. A preliminary two-stage alarm correlation and filtering system using SOM neural network and K-means algorithm. *Computers & Security*, 29(6):712–723, 2010.

[58] R. Lippmann, J.W. Haines, D.J. Fried, J. Korba, and K. Das. The 1999 DARPA off-line intrusion detection evaluation. *Computer Networks*, 34(4):579–595, 2000.

[59] Lincoln Laboratory, Massachusetts Institute of Technology. DARPA intrusion detection data sets.
`http://www.ll.mit.edu/mission/communications/cyber/CSTcorpora/ideval/data`.

[60] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez. Anomaly-based network intrusion detection: Techniques, systems and challenges. *Computers & Security*, 28(1-2):18–28, 2009.

[61] B. Coskun. Algorithms for network-based misuse detection. PhD dissertation, Polytechnic Institute of New York University, USA, 2010.

[62] L. Bilge and T. Dumitras. Before we knew it: An empirical study of zero-day attacks in the real world. In *CCS'12: Proceedings of the 19th ACM Conference on Computer and Communications Security*, pp. 833–844, October 2012.

[63] Microsoft Security TechCenter. Microsoft security advisory (2847140). May 2013. `http://technet.microsoft.com/en-us/security/advisory/2847140`.

[64] The Mitre Corporation. Common vulnerabilities and exposures CVE-2013-1347. May 2013. `http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2013-1347`.

[65] L. Wang, S. Jajodia, A. Singhal, and S. Noel. k-zero day safety: Measuring the security risk of networks against unknown attacks. In *ESORICS'10: Proceedings of the 15th European Symposium on Research in Computer Security*, volume 6345 of *Lecture Notes in Computer Science*, pp. 573–587, September 2010.

[66] R. Sekar, A. Gupta, J. Frullo, T. Shanbhag, A. Tiwari, H. Yang, and S. Zhou. Specification-based anomaly detection: A new approach for detecting network intrusions. In *CCS'02: Proceedings of the 9th ACM Conference on Computer and Communications Security*, pp. 265–274, November 2002.

[67] V. Richariya, U.P. Singh, and R. Mishra. Distributed approach of intrusion detection system: Survey. *International Journal of Advanced Computer Research*, 2(6):358–363, 2012.

[68] G. Vasiliadis, S. Antonatos, M. Polychronakis, E.P. Markatos, and S. Ioannidis. Gnort: High performance network intrusion detection using graphics processors. In *RAID'08: Proceedings of the 11th International Symposium on Recent Advances in Intrusion Detection*, volume 5230 of *Lecture Notes in Computer Science*, pp. 116–134, September 2008.

[69] W. Jiang, Y.E. Yang, and V.K. Prasanna. Scalable multi-pipeline architecture for high performance multi-pattern string matching. In *IPDPS'10: Proceedings of the 2010 IEEE International Symposium on Parallel & Distributed Processing*, pp. 1–12, April 2010.

[70] J. Yu, B. Yang, R. Sun, and Y. Chen. FPGA-based parallel pattern matching algorithm for network intrusion detection system. In *MINES'09: Proceedings of the 2009 International Conference on Multimedia Information Networking and Security*, pp. 458–461, November 2009.

[71] Y.-M. Hsiao, M.-J. Chen, Y.-S. Chu, and C.-H. Huang. High-throughput intrusion detection system with parallel pattern matching. *IEICE Electronics Express*, 9(18):1467–1472, 2012.

[72] Office of Science, U.S. Department of Energy. 100 Gbps science network. `http://science.energy.gov/ascr/news-and-resources/100gbpsnetwork`.

[73] H.T. Elshoush and I.M. Osman. Reducing false positives through fuzzy alert correlation in collaborative intelligent intrusion detection systems - A review. In *FUZZ-IEEE'10: Proceedings of the 19th IEEE International Conference on Fuzzy Systems*, pp. 1–8, July 2010.

[74] S.S.C. Silva, R.M.P. Silva, R.C.G. Pinto, and R.M. Salles. Botnets: A survey. *Computer Networks*, 57(2):378–403, 2013.

[75] Q. Han, W. Yu, Y. Zhang, and Z. Zhao. Modeling and evaluating of typical advanced peer-to-peer botnet. *Performance Evaluation*, 72:1–15, 2014.

[76] V.L. Thing, M. Sloman, and N. Dulay. A survey of bots used for distributed denial of service attacks. In *SEC'07: Proceedings of the IFIP TC-11 22nd International Information Security Conference*, pp. 229–240, May 2007.

[77] J.G. Rooney, C.J. Giblin, M. Waldvogel, and P.T. Hurley. Identifying a distributed denial of service (DDoS) attack within a network and defending against such an attack. United States Patent 7921462, May 2011.

[78] J. Demarest, Assistant Director in the Cyber Division of the Federal Bureau of Investigation (FBI). A hearing entitled "Taking down botnets: Public and private efforts to disrupt and dismantle cybercriminal networks". July 2014. `http://judiciary.senate.gov/imo/media/doc/07-15-14DemarestTestimony.pdf`.

[79] M. Feily, A. Shahrestani, and S. Ramadass. A survey of botnet and botnet detection. In *SECURWARE'09: Proceedings of the 3rd International Conference on Emerging Security Information, Systems and Technologies*, pp. 268–273, June 2009.

[80] A.K. Seewald and W.N. Gansterer. On the detection and identification of botnets. *Computers & Security*, 29(1):45–58, 2010.

[81] R. Bejtlich. *Extrusion detection: Security monitoring for internal intrusions*. Addison-Wesley Professional, 2005.

[82] C. Lüssi. Signature-based extrusion detection. Master's thesis, Swiss Federal Institute of Technology, Zurich, August 2008.

[83] C.V. Zhou, S. Karunasekera, and C. Leckie. Evaluation of a decentralized architecture for large scale collaborative intrusion detection. In *IM'07: Proceedings of the 10th IFIP/IEEE International Symposium on Integrated Network Management*, pp. 80–89, May 2007.

[84] M. Kinateder and K. Rothermel. Architecture and algorithms for a distributed reputation system. In *iTrust'03: Proceedings of the 1st International Conference on Trust Management*, volume 2692 of *Lecture Notes in Computer Science*, pp. 1–16, May 2003.

[85] M. Tavakolifard. On some challenges for online trust and reputation systems. PhD dissertation, Norwegian University of Science and Technology, Norway, 2012.

[86] M. Almgren and U. Lindqvist. Application-integrated data collection for security monitoring. In *RAID'01: Proceedings of the 4th International Symposium on Recent Advances in Intrusion Detection*, volume 2212 of *Lecture Notes in Computer Science*, pp. 22–36, October 2001.

[87] H.T. Elshoush and I.M. Osman. Intrusion alert correlation framework: An innovative approach. In *IAENG Transactions on Engineering Technologies*, volume 229 of *Lecture Notes in Electrical Engineering*, pp. 405–420, 2013.

[88] A.A. Abimbola, J.M. Munoz, and W.J. Buchanan. NetHost-Sensor: Investigating the capture of end-to-end encrypted intrusive data. *Computers & Security*, 25(6):445–451, 2006.

[89] V.T. Goh, J. Zimmermann, and M. Looi. Experimenting with an intrusion detection system for encrypted networks. *International Journal of Business Intelligence and Data Mining*, 5(2):172–191, 2010.

[90] A. Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[91] Imperva. Automation of attacks. Monthly Trend Report #9, Hacker Intelligence Initiative, April 2012. `http://imperva.com/docs/HII_Automation_of_Attacks.pdf`.

[92] S. Mandujano. A multiagent approach to outbound intrusion detection. PhD dissertation, Instituto Tecnológico y de Estudios Superiores de Monterrey, Mexico, 2004.

[93] Regit, To Linux and beyond! Investigation on an attack tool used in China. February 2014. `http://home.regit.org/2014/02/chinese-scanner`.

[94] N.B. Anuar, M. Papadaki, S. Furnell, and N. Clarke. An investigation and survey of response options for Intrusion Response Systems (IRSs). In *ISSA'10: Proceedings of the 2010 Information Security for South Africa Conference*, pp. 1–8, August 2010.

[95] A. Shameli-Sendi, M. Cheriet, and A. Hamou-Lhadj. Taxonomy of intrusion risk assessment and response system. *Computers & Security*, 45:1–16, 2014.

[96] G.B. White, E.A. Fisch, and U.W. Pooch. Cooperating security managers: A peer-based intrusion detection system. *IEEE Network*, 10(1):20–23, 1996.

[97] P.A. Porras and P.G. Neumann. EMERALD: Event monitoring enabling responses to anomalous live disturbances. In *NISS'97: Proceedings of the 20th National Information Systems Security Conference*, October 1997.

[98] C.A. Carver. Adaptive agent-based intrusion response. PhD dissertation, Texas A&M University, USA, 2001.

[99] B. Foo, Y.-S. Wu, Y.-C. Mao, S. Bagchi, and E. Spafford. ADEPTS: Adaptive intrusion response using attack graphs in an e-commerce environment. In *DSN'05: Proceedings of the 2005 International Conference on Dependable Systems and Networks*, pp. 508–517, June 2005.

[100] S.A. Zonouz, H. Khurana, W.H. Sanders, and T.M. Yardley. RRE: A game-theoretic intrusion response and recovery engine. *IEEE Transactions on Parallel and Distributed Systems*, 25(2):395–406, 2014.

[101] T.S. Sobh. Wired and wireless intrusion detection system: Classifications, good characteristics and state-of-the-art. *Computer Standards & Interfaces*, 28(6):670–694, 2006.

[102] P.M. Hesse and D.P. Lemire. Managing interoperability in non-hierarchical public key infrastructures. In *NDSS'02: Proceedings of Network and Distributed Security Symposium*, February 2002.

[103] National Security Authority. Certificate path validation version 1.4. Technical Report 1891/2006/IBEP-011, Department of Information Security and Electronic Signature, Slovakian National Security Authority, November 2006.

[104] M. Cooper, Y. Dzambasow, P. Hesse, S. Joseph, and R. Nicholas. Internet X.509 public key infrastructure: Certification path building. IETF Request for Comments 4158, September 2005.

[105] R. Housley, S. Ashmore, and C. Wallace. Trust anchor management protocol (TAMP). IETF Request for Comments 5934, August 2010.

[106] S. Lloyd. Understanding certification path construction. PKI Forum White Paper, Technology Working Group (TWG), September 2002.

[107] S. Lloyd (editor). CA-CA interoperability. PKI Forum White Paper, Technology Working Group (TWG), March 2001.

[108] W.E. Burr, N.A. Nazario, and W.T. Polk. A proposed federal PKI using X.509 V3 certificates. In *NISSC'96: Proceedings of the 19th National Information Systems Security Conference*, pp. 452–459, October 1996.

[109] K.D. Zeilenga. Lightweight directory access protocol (LDAP) schema definitions for X.509 certificates. IETF Request for Comments 4523, June 2006.

[110] Federal Public Key Infrastructure (FPKI). `http://idmanagement.gov/federal-public-key-infrastructure`.

[111] The Research and Education Bridge Certificate Authority (REBCA). http://rebca.org.

[112] S. Kent, S. Santesson (chairs), and W.T. Polk (area director). IETF-PKIX public key infrastructure X.509 working group charter. http://ietf.org/html.charters/pkix-charter.html.

[113] D. Pinkas and R. Housley. Delegated path validation and delegated path discovery protocol requirements. IETF Request for Comments 3379, September 2002.

[114] Y. Elley, A. Anderson, S. Hanna, S. Mullan, R. Perlman, and S. Proctor. Building certification paths: Forward vs. reverse. In *NDSS'01: Proceedings of Network and Distributed Security Symposium*, February 2001.

[115] D. Gallagher (chair of the Federal Public Key Infrastructure Policy Authority). X.509 certificate policy for the Federal Bridge Certification Authority (FBCA) version 2.27, December 2013.

[116] S. Santesson, M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams. X.509 Internet public key infrastructure online certificate status protocol - OCSP. IETF Request for Comments 6960, June 2013.

[117] T.P. Hormann, K. Wrona, and S. Holtmanns. Evaluation of certificate validation mechanisms. *Computer Communications*, 29(3):291–305, 2006.

[118] T. Freeman, R. Housley, A. Malpani, D. Cooper, and W.T. Polk. Server-based certificate validation protocol (SCVP). IETF Request for Comments 5055, December 2007.

[119] O. Gudmundsson, A. Sullivan (chairs), and R. Droms (area director). IETF-DNSSEC DNS extensions working group charter. http://ietf.org/html.charters/dnsext-charter.html.

[120] C. Fung. Collaborative intrusion detection networks and insider attacks. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, 2(1):63–74, 2011.

[121] D. Xu and P. Ning. Privacy-preserving alert correlation: A concept hierarchy based approach. In *ACSAC'05: Proceedings of the 21st Annual Computer Security Applications Conference*, pp. 537–546, December 2005.

[122] H. Qusa, R. Baldoni, and R. Beraldi. A privacy preserving scalable architecture for collaborative event correlation. In *TRUSTCOM'12: Proceedings of the 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*, pp. 837–843, June 2012.

[123] S. Niksefat, B. Sadeghiyan, P. Mohassel, and S. Sadeghian. ZIDS: A privacy-preserving intrusion detection system using secure two-party computation protocols. *The Computer Journal*, 2013.

[124] H.S. Acharya, S.R. Dutta, and R. Bhoi. Network and information security issues. *International Journal of Innovative Research & Development*, 2(2):400–406, 2013.

[125] L. Mekouar, Y. Iraqi, and R. Boutaba. Reputation-based trust management in peer-to-peer systems: Taxonomy and anatomy. In *Handbook of Peer-to-Peer Networking*, pp. 689–732, 2010.

[126] S.R. Snapp, J. Brentano, G.V. Dias, T.L. Goan, L.T. Heberlein, C.-L. Ho, K.N. Levitt, B. Mukherjee, S.E. Smaha, T. Grance, D.M. Teal, and D. Mansur. DIDS (Distributed Intrusion Detection System) - Motivation, architecture, and an early prototype. In *NCSC'91: Proceedings of the 14th National Computer Security Conference*, pp. 167–176, October 1991.

[127] SRI International. A real-time intrusion-detection expert system (IDES). Technical Report SRI Project 6784, Computer Science Laboratory, California, USA, February 1992.

[128] J. Hochberg, K. Jackson, C. Stallings, J.F. McClary, D. DuBois, and J. Ford. NADIR: An automated system for detecting network intrusion and misuse. *Computers & Security*, 12(3):235–248, 1993.

[129] S. Staniford-Chen, S. Cheung, R. Crawford, M. Dilger, J. Frank, J. Hoagland, K. Levitt, C. Wee, R. Yip, and D. Zerkle. Grids-a graph based intrusion detection system for large networks. In *NISS'96: Proceedings of the 19th National Information Systems Security Conference*, pp. 361–370, October 1996.

[130] C.V. Zhou, C. Leckie, and S. Karunasekera. Decentralized multi-dimensional alert correlation for collaborative intrusion detection. *Journal of Network and Computer Applications*, 32(5):1106–1123, 2009.

[131] S. Eichler. A security architecture concept for vehicular network nodes. In *ICICS'07: Proceedings of the 6th International Conference on Information, Communications & Signal Processing*, pp. 1–5, December 2007.

[132] J. Zhang. A survey on trust management for VANETs. In *AINA'11: Proceedings of the 25th International Conference on Advanced Information Networking and Applications*, pp. 105–112, March 2011.

[133] R. Janakiraman, M. Waldvogel, and Q. Zhang. Indra: A peer-to-peer approach to network intrusion detection and prevention. In *WETICE'03: Proceedings of the 12th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, pp. 226–231, June 2003.

[134] M.E. Locasto, J.J. Parekh, A.D. Keromytis, and S.J. Stolfo. Towards collaborative security and P2P intrusion detection. In *IAW'05: Proceedings of the 6th Annual IEEE SMC on Information Assurance Workshop*, pp. 333–339, June 2005.

[135] T. Dimitriou, G. Karame, and I.T. Christou. SuperTrust - A secure and efficient framework for handling trust in super peer networks. In *ICDCN'08: Proceedings of the 9th International Conference on Distributed Computing and Networking*, volume 4904 of *Lecture Notes in Computer Science*, pp. 350–362, January 2008.

[136] A. Ghosh and S. Sen. Agent-based distributed intrusion alert system. In *IWDC'04: Proceedings of the 6th International Workshop on Distributed Computing*, volume 3326 of *Lecture Notes in Computer Science*, pp. 240–251, November 2004.

[137] J. Yu, Y.V. Ramana Reddy, S. Selliah, S. Reddy, V. Bharadwaj, and S. Kankanahalli. TRINETR: An architecture for collaborative intrusion detection and knowledge-based alert evaluation. *Advanced Engineering Informatics*, 19(2):93–101, 2005.

[138] R.A. Kemmerer. NSTAT: A model-based real-time network intrusion detection system. Technical report, University of California, USA, 1998.

[139] SANS Technology Institute. DShield: A distributed intrusion detection system. `http://dshield.org`.

[140] F. Cuppens and A. Miège. Alert correlation in a cooperative intrusion detection framework. In *S&P'02: Proceedings of the 2002 IEEE Symposium on Security and Privacy*, pp. 202–215, May 2002.

[141] J.S. Balasubramaniyan, J.O. Garcia-Fernandez, D. Isacoff, E. Spafford, and D. Zamboni. An architecture for intrusion detection using autonomous agents. In *CSAC'98: Proceedings of the 14th Annual on Computer Security Applications Conference*, pp. 13–24, December 1998.

[142] E.H. Spafford and D. Zamboni. Intrusion detection using autonomous agents. *Computer Networks*, 34(4):547–570, 2000.

[143] V. Yegneswaran, P. Barford, and S. Jha. Global intrusion detection in the DOMINO overlay system. In *NDSS'04: Proceedings of Network and Distributed Security Symposium*, February 2004.

[144] A.K. Ganame, J. Bourgeois, R. Bidou, and F. Spies. A global security architecture for intrusion detection on computer networks. *Computers & Security*, 27(1-2):30–47, 2008.

[145] R. Bye and S. Albayrak. CIMD-collaborative intrusion and malware detection. Technical Report TUB-DAI 08/08-01, Technische Universität Berlin, Germany, August 2008.

[146] M. Cai, K. Hwang, Y.-K. Kwok, S. Song, and Y. Chen. Collaborative Internet worm containment. *IEEE Security & Privacy*, 3(3):25–33, 2005.

[147] V. Gowadia, C. Farkas, and M. Valtorta. PAID: A probabilistic agent-based intrusion detection system. *Computers & Security*, 24(7):529–545, 2005.

[148] X. Li, F. Bian, H. Zhang, C. Diot, R. Govindan, W. Hong, and G. Iannaccone. Mind: A distributed multi-dimensional indexing system for network diagnosis. In *INFOCOM'06: Proceedings of the 25th IEEE International Conference on Computer Communications*, pp. 1–12, April 2006.

[149] D. Dash, B. Kveton, J.M. Agosta, E. Schooler, J. Chandrashekar, A. Bachrach, and A. Newman. When gossip is good: Distributed probabilistic inference for detection of slow network intrusions. In *AAAI'06: Proceedings of the 21st National Conference on Artificial Intelligence*, pp. 1115–1122, July 2006.

[150] A. Abraham, R. Jain, J. Thomas, and S.Y. Han. D-SCIDS: Distributed soft computing intrusion detection system. *Journal of Network and Computer Applications*, 30(1):81–98, 2007.

[151] Z. Zhong, L. Ramaswamy, and K. Li. ALPACAS: A large-scale privacy-aware collaborative anti-spam system. In *INFOCOM'08: Proceedings of the 27th IEEE Conference on Computer Communications*, pp. 556–564, April 2008.

[152] D. Ye, Q. Bai, M. Zhang, and Z. Ye. P2P distributed intrusion detections by using mobile agents. In *ICIS'08: Proceedings of the 7th IEEE/ACIS International Conference on Computer and Information Science*, pp. 259–265, May 2008.

[153] O.S. Adebukola, A.O. Bamidele, and A.A. Taofik. A simulated multiagent-based architecture for intrusion detection system. *International Journal of Advanced Research in Artificial Intelligence*, 2(4):29–38, 2013.

[154] Y. Wang and V. Varadharajan. Interaction trust evaluation in decentralized environments. In *EC-Web'04: Proceedings of the 5th International Conference on E-Commerce and Web Technologies*, volume 3182 of *Lecture Notes in Computer Science*, pp. 144–153, August-September 2004.

[155] I. Pinyol, J. Sabater-Mir, P. Dellunde, and M. Paolucci. Reputation-based decisions for logic-based cognitive agents. *Autonomous Agents and Multi-Agent Systems*, 24(1):175–216, 2012.

[156] G. Zacharia. Collaborative reputation mechanisms for online communities. Master's thesis, Massachusetts Institute of Technology, USA, September 1999.

[157] J. Sabater and C. Sierra. REGRET: Reputation in gregarious societies. In *AGENTS'01: Proceedings of the 5th International Conference on Autonomous Agents*, pp. 194–195, June 2001.

[158] C. Fung, J. Zhang, I. Aib, and R. Boutaba. Trust management and admission control for host-based collaborative intrusion detection. *Journal of Network and Systems Management*, 19(2):257–277, 2011.

[159] T.D. Huynh. Trust and reputation in open multi-agent systems. PhD dissertation, University of Southampton, United Kingdom, 2006.

[160] L. You and R. Sikora. An adaptive evaluation mechanism for online traders. *European Journal of Operational Research*, 214(3):739–748, 2011.

[161] L. Xiong and L. Liu. PeerTrust: Supporting reputation-based trust for peer-to-peer electronic communities. *IEEE Transactions on Knowledge and Data Engineering*, 16(7):843–857, 2004.

[162] B. Yu and M.P. Singh. A social mechanism of reputation management in electronic communities. In *CIA'00: Proceedings of the 4th International Workshop on Cooperative Information Agents IV, The Future of Information Agents in Cyberspace*, volume 1860 of *Lecture Notes in Computer Science*, pp. 154–165, July 2000.

[163] S.D. Kamvar, M.T. Schlosser, and H. Garcia-Molina. The EigenTrust algorithm for reputation management in P2P networks. In *WWW'03: Proceedings of the 12th International Conference on World Wide Web*, pp. 640–651, May 2003.

[164] J. Golbeck, B. Parsia, and J. Hendler. Trust networks on the semantic Web. In *CIA'03: Proceedings of the 7th International Workshop on Cooperative Information Agents VII, Trust Networks on the Semantic Web*, volume 2782 of *Lecture Notes in Computer Science*, pp. 238–249, August 2003.

[165] M. Richardson, R. Agrawal, and P. Domingos. Trust management for the semantic Web. In *ISWC'03: Proceedings of the 2nd International Semantic Web Conference*, volume 2870 of *Lecture Notes in Computer Science*, pp. 351–368, October 2003.

[166] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *WWW'04: Proceedings of the 13th International Conference on World Wide Web*, pp. 403–412, May 2004.

[167] W. Li, J. Parker, and A. Joshi. Security through collaboration and trust in MANETs. *Mobile Networks and Applications*, 17(3):342–352, 2012.

[168] T.D. Huynh, N.R. Jennings, and N.R. Shadbolt. Certified reputation: How an agent can trust a stranger. In *AAMAS'06: Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems*, pp. 1217–1224, May 2006.

[169] V. Botêlho, F. Enembreck, B. Ávila, H. de Azevedo, and E. Scalabrin. Using asymmetric keys in a certified trust model for multiagent systems. *Expert Systems with Applications*, 38(2):1233–1240, 2011.

[170] C. Duma, M. Karresand, N. Shahmehri, and G. Caronni. A trust-aware, P2P-based overlay for intrusion detection. In *DEXA'06: Proceedings of the 17th International Workshop on Database and Expert Systems Applications*, pp. 692–697, September 2006.

[171] Z. Zhang, P.H. Ho, and F. Naït-Abdesselam. RADAR: A reputation-driven anomaly detection system for wireless mesh networks. *Wireless Networks*, 16(8):2221–2236, 2010.

[172] L. Gheorghe, R. Rughinis, and R. Tataroiu. Adaptive trust management protocol based on intrusion detection for wireless sensor networks. In *RoEduNet'13: Proceedings of the 12th International Conference on Networking in Education and Research*, pp. 1–7, September 2013.

[173] J. Duan, D. Yang, H. Zhu, S. Zhang, and J. Zhao. TSRF: A trust-aware secure routing framework in wireless sensor networks. *International Journal of Distributed Sensor Networks*, vol. 2014, Article ID 209436, 14 pages, 2014.

[174] C. Fung. Design and management of collaborative intrusion detection networks. PhD dissertation, University of Waterloo, Canada, 2013.

[175] C.J. Fung, O. Baysal, J. Zhang, I. Aib, and R. Boutaba. Trust management for host-based collaborative intrusion detection. In *DSOM'08: Proceedings of the 19th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management*, volume 5273 of *Lecture Notes in Computer Science*, pp. 109–122, September 2008.

[176] C.J. Fung, J. Zhang, I. Aib, and R. Boutaba. Robust and scalable trust management for collaborative intrusion detection. In *IM'09: Proceedings of the IFIP/IEEE International Symposium on Integrated Network Management*, pp. 33–40, June 2009.

[177] C.J. Fung, J. Zhang, I. Aib, and R. Boutaba. Dirichlet-based trust management for effective collaborative intrusion detection networks. *IEEE Transactions on Network and Service Management*, 8(2):79–91, 2011.

[178] K. Hoffman, D. Zage, and C. Nita-Rotaru. A survey of attack and defense techniques for reputation systems. *ACM Computing Surveys*, 42(1):40–47, 2009.

[179] P. Narula, S.K. Dhurandher, S. Misra, and I. Woungang. Security in mobile ad-hoc networks using soft encryption and trust-based multi-path routing. *Computer Communications*, 31(4):760–769, 2008.

[180] L. Su, W. Wang, and A. Niu. Reputation service and reputation based access control. In *ICEEE'10: Proceedings of the 2010 International Conference on E-Product E-Service and E-Entertainment*, pp. 1–4, November 2010.

[181] X. Wang, K. Govindan, and P. Mohapatra. Provenance-based information trustworthiness evaluation in multi-hop networks. In *GLOBECOM'10: Proceedings of the 2010 IEEE Global Telecommunications Conference*, pp. 1–5, December 2010.

[182] M. Feldman and J. Chuang. The evolution of cooperation under cheap pseudonyms. In *CEC'05: Proceedings of the 7th IEEE International Conference on E-Commerce Technology*, pp. 284–291, July 2005.

[183] Z. Malik and A. Bouguettaya. Reputation bootstrapping for trust establishment among Web services. *IEEE Internet Computing*, 13(1):40–47, 2009.

[184] A. Joardar, T. Kostova, and E.C. Ravlin. An experimental study of the acceptance of a foreign newcomer into a workgroup. *Journal of International Management*, 13(4):513–537, 2007.

[185] S.I. Ahamed, E. Hoque, F. Rahman, and M. Zulkernine. Towards secure trust bootstrapping in pervasive computing environment. In *HASE'08: Proceedings of the 11th IEEE High Assurance Systems Engineering Symposium*, pp. 89–96, December 2008.

[186] M. Venanzi, M. Piunti, R. Falcone, and C. Castelfranchi. Reasoning with categories for trusting strangers: A cognitive architecture. In *TRUST'11: Proceedings of the 14th International Workshop on Trust in Agent Societies*, pp. 109–124, May 2011.

[187] R. Falcone, M. Piunti, M. Venanzi, and C. Castelfranchi. From Manifesta to Krypta: The relevance of categories for trusting others. *ACM Transactions on Intelligent Systems and Technology*, 4(2):27:1–27:24, 2013.

[188] R. Falcone, A. Sapienza, and C. Castelfranchi. Trusting information sources through their categories. In *PAAMS'15: Proceedings of the 13th International Conference on Advances in Practical Applications of Agents, Multi-Agent Systems, and Sustainability*, volume 9086 of *Lecture Notes in Computer Science*, pp. 80–92, May 2015.

[189] C. Burnett, T.J. Norman, and K. Sycara. Stereotypical trust and bias in dynamic multiagent systems. *ACM Transactions on Intelligent Systems and Technology*, 4(2):26:1–26:22, 2013.

[190] M. Şensoy, B. Yilmaz, and T.J. Norman. STAGE: Stereotypical trust assessment through graph extraction. *Computational Intelligence*, 2014.

[191] C. Burnett, T.J. Norman, and K. Sycara. Bootstrapping trust evaluations through stereotypes. In *AAMAS'10: Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, pp. 241–248, May 2010.

[192] X. Liu, A. Datta, and K. Rzadca. Trust beyond reputation: A computational trust model based on stereotypes. *Electronic Commerce Research and Applications*, 12(1):24–39, 2013.

[193] C. Huang, H. Hu, and Z. Wang. A dynamic trust model based on feedback control mechanism for P2P applications. In *ATC'06: Proceedings of the 3rd International Conference on Autonomic and Trusted Computing*, volume 4158 of *Lecture Notes in Computer Science*, pp. 312–321, September 2006.

[194] C. Burnett, T.J. Norman, and K. Sycara. Sources of stereotypical trust in multi-agent systems. In *TRUST'11: Proceedings of the 14th International Workshop on Trust in Agent Societies*, pp. 25–39, May 2011.

[195] R. Labassi, M. Hamdi, and T.-H. Kim. Quantitative risk management: A survey of adaptive approaches to risk management for information and communication systems. *International Journal of u- and e- Service, Science and Technology*, 8(5):105–128, 2015.

[196] M. Gen, R. Cheng, and L. Lin. *Network models and optimization: Multiobjective genetic algorithm approach*. Springer London, 2008.

[197] J.E. Tapiador and J.A. Clark. Learning autonomic security reconfiguration policies. In *CIT'10: Proceedings of the 10th IEEE International Conference on Computer and Information Technology*, pp. 902–909, June 2010.

[198] I. Brahmi, S.B. Yahia, and P. Poncelet. A Snort-based mobile agent for a distributed intrusion detection system. In *SECRYPT'11: Proceedings of the 2011 International Conference on Security and Cryptography*, pp. 198–207, July 2011.

[199] V. Stankovic and L. Strigini. A survey on online monitoring approaches of computer-based systems. Technical Report 531, Centre for Software Reliability, City University London, June 2009.

[200] Distributed Management Task Force, Inc. Web-based enterprise management (WBEM). `http://dmtf.org/standards/wbem`.

[201] OASIS Technical Committee. Web services distributed management (WSDM). `http://oasis-open.org/committees/wsdm`.

[202] A.S. Teles, J.P.M. Mendes, and Z. Abdelouahab. Autonomic computing applied to network security: A survey. *Journal of Selected Areas in Telecommunications*, pp. 7–14, 2011.

[203] J.E. Tapiador and J.A. Clark. The placement-configuration problem for intrusion detection nodes in wireless sensor networks. *Computers & Electrical Engineering*, 39(7):2306–2317, 2013.

[204] S. Kirkpatrick, C.D. Gelatt Jr., and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[205] H. Chen, J.A. Clark, S.A. Shaikh, H. Chivers, and P. Nobles. Optimising IDS sensor placement. In *ARES'10: Proceedings of the 5th International Conference on Availability, Reliability, and Security*, pp. 315–320, February 2010.

[206] R. Mueller-Bady, R. Gad, M. Kappes, and I. Medina-Bulo. Using genetic algorithms for deadline-constrained monitor selection in dynamic computer networks. In *GECCO'15: Proceedings of the 2015 Genetic and Evolutionary Computation Conference*, pp. 867–874, July 2015.

[207] S. Noel and S. Jajodia. Optimal ids sensor placement and alert prioritization using attack graphs. *Journal of Network and Systems Management*, 16(3):259–275, 2008.

[208] V.D. Desai. Placement of sensor using attack graph. *Journal of Emerging Technologies and Innovative Research*, 2(1):13–21, 2015.

[209] G. Garrison, R.L. Wakefield, X. Xu, and S.H. Kim. Globally distributed teams: The effect of diversity on trust, cohesion and individual performance. *ACM SIGMIS Database*, 41(3):27–48, 2010.

[210] T.-P. Liang, J.C.-H. Wu, J.J. Jiang, and G. Klein. The impact of value diversity on information system development projects. *International Journal of Project Management*, 30(6):731–739, 2012.

[211] K. Bartos and M. Rehak. Trust-based solution for robust self-configuration of distributed intrusion detection systems. In *ECAI'12: Proceedings of the 20th European Conference on Artificial Intelligence*, pp. 121–126, August 2012.

[212] L. Wang, M. Zhang, S. Jajodia, A. Singhal, and M. Albanese. Modeling network diversity for evaluating the robustness of networks against zero-day attacks. In *ESORICS'14: Proceedings of the 19th European Symposium on Research in Computer Security*, volume 8713 of *Lecture Notes in Computer Science*, pp. 494–511, September 2014.

[213] U. Maurer. Modelling a public-key infrastructure. In *ESORICS'96: Proceedings of the 4th European Symposium on Research in Computer Security*, volume 1146 of *Lecture Notes in Computer Science*, pp. 325–350, September 1996.

[214] Japan PKI Forum, Korea PKI Forum, PKI Forum Singapore, and Chinese Taipei PKI Forum. Achieving PKI interoperability 2003: Results of the JKST-IWG interoperability project, July 2007.

[215] General Services Administration. FBCA and C4CA cross-certification. Technical Guide, March 2007.

[216] Departamento de Ingeniería de la Información y las Comunicaciones, Universidad de Murcia, Spain. UMU-PKIv6: Public Key Infrastructure with IPv6 support. `http://pki.inf.um.es`.

[217] G. Martínez Pérez. Propuesta de un entorno de seguridad para la gestión de políticas en redes IP. PhD dissertation, Universidad de Murcia, Spain, 2003.

[218] A.F. Gómez Skarmeta, G. Martínez Pérez, O. Cánovas Reverte, and G. López Millán. PKI services for IPv6. *IEEE Internet Computing*, 7(3):36–42, 2003.

[219] C. Adams, P. Cain, D. Pinkas, and R. Zuccherato. Internet X.509 public key infrastructure time-stamp protocol (TSP). IETF Request for Comments 3161, August 2001.

[220] The EuroPKI Web site. `http://www.europki.org`.

[221] U.S. Department of Defense. DoD PKI. `http://iase.disa.mil/pki`.

[222] U.S. Government. External Certificate Authority (ECA). `http://eca.orc.com`.

[223] IdenTrust Inc. Digital Signature Trust (DST). `http://identrust.com`.

[224] SAFE-Biopharma Association. SAFE-Biopharma Bridge Certificate Authority (SBCA). `http://safe-biopharma.org`.

[225] J. Garcia-Alfaro, M.A. Jaeger, G. Mühl, I. Barrera, and J. Borrell. Distributed exchange of alerts for the detection of coordinated attacks. In *CNSR'08: Proceedings of the Communication Networks and Services Research Conference*, pp. 96–103, May 2008.

[226] R.W. Maule. *Modeling and simulation support for system of systems engineering applications*, chapter Model methodology for a Department of Defense architecture design, pp. 145–183. John Wiley & Sons, Inc., 2015.

[227] E.K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim. A survey and comparison of peer-to-peer overlay network schemes. *IEEE Communications Surveys and Tutorials*, 7(2):72–93, 2005.

[228] Trend Micro, Inc. OSSEC: An open source host intrusion detection system. http://ossec.net.

[229] A.A. Ramaki, M. Amini, and R.E. Atani. RTECA: Real time episode correlation algorithm for multi-step attack scenarios detection. *Computers & Security*, 49:206–219, 2015.

[230] T. Ahmed, M.M. Siraj, A. Zainal, and M. Mat Din. A taxonomy on intrusion alert aggregation techniques. In *ISBAST'14: Proceedings of the International Symposium on Biometrics and Security Technologies*, pp. 244–249, August 2014.

[231] European Union Agency for Network and Information Security. Standards and tools for exchange and processing of actionable information. November 2014. http://enisa.europa.eu/activities/cert/support/actionable-information/standards-and-tools-for-exchange-and-processing-of-actionable-information.

[232] R. Bye, A. Camtepe, and S. Albayrak. *Collaborative computer security and trust management*, chapter Teamworking for security: The collaborative approach, pp. 12–33. IGI Global, 2009.

[233] H. Debar, D.A. Curry, and B.S. Feinstein. The intrusion detection message exchange format (IDMEF). IETF Request for Comments 4765, March 2007.

[234] R. Danyliw, J. Meijer, and Y. Demchenko. The incident object description exchange format. IETF Request for Comments 5070, December 2007.

[235] Geeknet, Inc. Snort IDMEF: An IDMEF XML plugin for Snort. October 2009. http://snort-idmef.sourceforge.net.

[236] U.S. Department of Homeland Security. STIX: Structured threat information expression. http://stix.mitre.org.

[237] U.S. Department of Homeland Security. CybOX: Cyber observable expression. http://cybox.mitre.org.

[238] A. Wierzbicki, J. Kalinski, and T. Kruszona. Common intrusion detection signatures standard (CIDSS). IETF Internet Draft 05, September 2008.

[239] CIDSS Development Group. CIDSS: Common intrusion detection signatures standard. September 2008. http://cidss.sourceforge.net.

[240] F. Gómez Mármol, J. Girao, and G. Martínez Pérez. TRIMS, a privacy-aware trust and reputation model for identity management systems. *Computer Networks*, 54(16):2899–2912, 2010.

[241] G. Martínez Pérez and J.M. Alcaraz Calero. Open WS-Eventing. http://sourceforge.net/projects/wse.

[242] D. Davis, A. Malhotra, K. Warr, and W. Chou (editors). Web services eventing (WS-Eventing). W3C Recommendation, December 2011.

[243] J.A. Golbeck. Computing and applying trust in Web-based social networks. PhD dissertation, University of Maryland, USA, 2005.

[244] X. Liu, A. Datta, K. Rzadca, and E.-P. Lim. StereoTrust: A group based personalized trust model. In *CIKM'09: Proceeding of the 18th ACM Conference on Information and Knowledge Management*, pp. 7–16, November 2009.

[245] M. Lesani and N. Montazeri. Fuzzy trust aggregation and personalized trust inference in virtual social networks. *Computational Intelligence*, 25(2):51–83, 2009.

[246] Emerson R. de Mello, Aad van Moorsel, and Joni da Silva Fraga. Evaluation of P2P search algorithms for discovering trust paths. In *EPEW'07: Proceedings of the 4th European Performance Engineering Workshop on Formal Methods and Stochastic Models for Performance Evaluation*, volume 4748 of *Lecture Notes in Computer Science*, pp. 112–124, September 2007.

[247] United States Computer Emergency Readiness Team, U.S. Department of Homeland Security. Traffic Light Protocol (TLP). `http://us-cert.gov/tlp`.

[248] J. Douceur. The Sybil attack. In *IPTPS'02: Proceedings of the 1st International Workshop on Peer-to-Peer Systems*, volume 2429 of *Lecture Notes in Computer Science*, pp. 251–260, March 2002.

[249] J.B. Bolten. E-authentication guidance for federal agencies, Memorandum M-04-04 to the Heads of all Departments and Agencies, Executive Office of the President, White House. December 2003.
`http://whitehouse.gov/omb/memoranda/fy04/m04-04.pdf`.

[250] W.E. Burr, D.F. Dodson, E.M. Newton, R.A. Perlner, W.T. Polk, S. Gupta, and E.A. Nabbus. Electronic authentication guideline. Technical Report NIST Special Publication 800-63-2, Computer Security Division, National Institute of Standards and Technology, August 2013.

[251] J.E. Tapiador, J.A. Clark, and J.C. Hernández-Castro. Non-linear cryptanalysis revisited: Heuristic search for approximations to S-boxes. In *IMACC'07: Proceedings of the 11th IMA International Conference on Cryptography and Coding*, volume 4887 of *Lecture Notes in Computer Science*, pp. 99–117, December 2007.

[252] D. De Catanzaro and J.C. Taylor. The scaling of dispersion and correlation: A comparison of least-squares and absolute-deviation statistics. *British Journal of Mathematical and Statistical Psychology*, 49(1):171–188, 1996.