UNIVERSIDAD DE MURCIA

ESCUELA INTERNACIONAL DE DOCTORADO

Policy-based security management for SDN/NFV-aware next-generation IoT infrastructures

Gestión de seguridad dirigida por políticas para infraestructuras IoT de nueva generación basadas en SDN/NFV

D. Alejandro Molina Zarca

2020

Universidad de Murcia

Facultad de Informática

Gestión de seguridad dirigida por políticas para infraestructuras IoT de nueva generación basadas en SDN/NFV

Tesis Doctoral

Presentada por:
Alejandro Molina Zarca

Supervisada por:
Dr. Antonio Fernando Skarmeta Gómez
Dr. Jorge Bernal Bernabé

Murcia, Septiembre de 2020

University of Murcia

Faculty of Computer Science

Policy-based security management for next-generation SDN/NFV-enabled Internet of Things infrastructures.

Ph.D. Thesis

Authored by:
Alejandro Molina Zarca

Supervised by:
Dr. Antonio Fernando Skarmeta Gómez
Dr. Jorge Bernal Bernabé

Murcia, September 2020

A mi madre y a mi abuelo.
Que esta tesis sea el reflejo de la perseverancia que siempre me han enseñado.

Agradecimientos

Quiero dar las gracias a mis directores de tesis, Antonio y Jorge, por darme la oportunidad de descubrir el mundo de la investigación, así como por guiarme, motivarme y animarme a cada paso durante estos años. Gracias por confiar en mí para ser uno más en el equipo. Gracias por darme a descubrir lo que es estar en primera linea ante el fascinante mundo que son los proyectos de investigación internacionales. Gracias por vuestro tiempo, por vuestras directrices, por vuestros consejos y por vuestra amistad. Todas las publicaciones de esta tesis son el fruto de vuestra diligencia y buen hacer. Vuestro incesable esfuerzo por estar a la vanguardia de esta profesión hace de mí lo que soy hoy.

Doy gracias a mi madre, a la que se lo debo todo. Por enseñarme que caerse y levantarse ante la adversidad es parte esencial del aprendizaje, por enseñarme con su ejemplo los frutos de poner el corazón en todo lo que haces. Tu pequeño, al que compraste su primer ordenador, hoy presenta su tesis doctoral. A mi padre, siempre cuidando e intercediendo por mí desde lo alto, cuántas veces he notado tu cálido abrazo en los momentos más difíciles. A mi abuelo, modelo incansable de constancia, voluntad, firmeza y perfeccionismo que marcan el ejemplo a seguir (Don justo, metódico, preciso). Espero ser el padre que él ha sido para mi. Muy especialmente, a mi hermano, cuya fe en mi me ha llevado a alcanzar metas con las que sólo podía soñar. Te quiero hermano.

A mi esposa, que ha recorrido a mi lado cada paso durante mis andanzas en el mundo de la informática, contra viento y marea, desde la formación profesional hasta el doctorado, y lo que quede por llegar. Gracias por ser el faro de mi vida. Sin tu apoyo incondicional nada de esto hubiera sido posible. Esta tesis es tan tuya como mía.

Gracias a su familia, Juani, Pedro e Isabel. Siempre alentando mi estudio y trabajo, siempre con su mejor cara, siempre con un dulce en el plato.

También quisiera agradecer a todos mis compañeros de investigación y a los compañeros de laboratorio (Dibulibu, T3 y Gaia), su apoyo, así como todo lo que me han enseñado. Es fácil trabajar en el ambiente que generáis día a día. En especial quiero dar las gracias a Jorge y a Jordi, siempre al pie del cañón para todos nosotros, con una sonrisa de oreja a oreja, truene o llueva.

A Rafa y a Dan, por un épico trabajo en equipo, así como por todos los grandes momentos durante los viajes de proyecto (ki 200). A Sara, por su gran labor de investigación, de la que me ha hecho partícipe, y especialmente a nivel personal, por todo lo que me ha ayudado en la gran batalla que resulta el depósito de la tesis.

Finalmente, a mis profesores de la facultad, un elenco al que de verdad admiro y respeto. Especialmente a Pedro Miguel, Ginés, Oscar, Gabi, Rafa, Gregorio, Antonio Ruiz y Antonio Skármeta, cuyas clases despertaron en mi la pasión que me impulsó para llegar hasta aquí. Gracias Pedro, Antonio Ruiz y Antonio Skarmeta por apostar por mi con esas primeras becas de alumno interno, colaboración e investigación.

"Si he visto más lejos es porque estoy apoyado sobre los hombros de gigantes".
Isaac Newton

"Si un Terminator, una máquina, es capaz de aprender el valor de la vida humana, quizás nosotros también podamos".
Sarah Connor

Contents

List of Figures

List of Tables

Abbreviations

6LoWPAN	IPv6 over Low power Wireless Personal Area Networks
AAA	Authentication, Authorization and Accounting
ABAC	Attribute-based Access Control
ACL	Access Control List
ANASTACIA	Advanced Networked Agents for Security and Trust Assessment in CPS / IOT Architectures
ARP	Address Resolution Protocol
ARPANET	Advanced Research Projects Agency Network
CIM	Common Information Model
CoAP	Constrained Application Protocol
CP-ABE	Ciphertext-Policy Attribute-Based Encryption
CPS	Cyber Physical System
DCapBAC	Distributed Capability-based Access Control
DDoS	Distributed Denial of Service
DHCP	Dynamic Host Configuration Protocol
DMTF	Distributed Management Task Force
DoS	Denial of Service
DPI	Deep Packet Inspection
DTLS	Datagram Transport Layer Security
E2E	End to End
EAP	Extensible Authentication Protocol
ECA	Event-Condition-Action
ECA-P	Event-Condition-Action-Postcondition
ECDSA	Elliptic Curve Digital Signature Algorithm
E-P3P	Platform for Enterprise Privacy Practices
EPAL	Enterprise Privacy Authorization Language
ETSI	European Telecommunications Standards Institute
GPS	Global Positioning System
GTP	GPRS Tunneling Protocol
GUI	Graphic User Interface
HRBAC	Hierarchical Role-Based Access Control
HSPL	High-level Security Policy Language
HSPL-OP	High-level Security Policy Language Orchestration Policy
HVAC	Heating, ventilation and air conditioning
I2NSF	Interface to Network Security Functions
ICMP	Internet Control Message Protocol
IETF	Internet Engineering Task Force
ILP	Integer Linear Programming
IoT	Internet of Things

IPS	Intrusion Prevention System
IRTF	Internet Research Task Force
KVM	Kernel-based Virtual Machine
LXC	Linux Containers
MANETS	Mobile Ad-hoc Networks
MANO	Management and Orchestration
MILP	Mixed Integer Linear Programming
MMT	Montimage Monitoring Tool
MQTT	Message Queue Telemetry Transport
MSK	Master Session Key
MSPL	Medium-level Security Policy Language
MSPL-OP	Medium-level Security Policy Language Orchestration Policy
MUD	Manufacturer Usage Description
NB-IoT	Narrowband-IoT
NFV	Network Function Virtualization
NGAC	Next Generation Access Control
NIST	National Institute of Standards and Technology
OSCORE	Object Security for Constrained RESTful Environments
OSDF	Open Software Defined Framework
OSI	Open Systems Interconnection
OSM	Open Source Mano
OSSIM	Open Source SIEM
OWL	Web Ontology Language
PaC-EP	Pana Client - Enforcement Point
PANA	Protocol for Carrying Authentication for Network Access
PDL	Policy Description Language
PDP	Policy Decision Point
PEMK	PaC-EP Master Key
PFDL	Policy Framework Definition Language
PoC	Proof of Concept
PPDL	Preference-based Policy Description Language
QoS	Quality of Service
RBAC	Role-based Access Control
RFID	Radio Frequency Identification
SAs	Security Associations
SAM	Structural Attack Model
SDN	Software Defined Networking
SIEM	Security Information and Event Management
SOTA	State of the Art
SQL	Structured Query Language
SSM	Smart Security Mechanism
TCP	Transport Control Protocol
TIHDL	Technology Independent Honeynet Description Language
UML	Unified Modeling Language
VIM	Virtual Infrastructure Manager
VLAN	Virtual Local Area Network
VXLAN	Virtual eXtensible Local Area Network
W3C	World Wide Web Consortium
WPS	WiFi Protected Setup
WSN	Wireless Sensor Network
XACML	eXtensible Access Control Markup Language
XML	eXtensible Markup Language

Capítulo 1

Resumen

1.1. Motivación

Al principio de esta tesis (2017), distintas fuentes estimaban cerca de veintitrés mil millones de dispositivos IoT conectados a la red y los estadistas pronosticaban que esos números podrían alcanzar hasta 50 mil millones de dispositivos en pocos años[1]. Ahora (2020) esas estimaciones consideran que cada segundo, cerca de 127 nuevos dispositivos IoT son conectados a la red[2], generando unas expectativas las cuales indican que en cinco años podríamos hablar de 70 mil millones de dispositivos IoT compartiendo nuestro entorno. La figura 1.1 muestra un histórico y estimación desde 2015 hasta 2030 adquiridas de distintas fuentes (no todas las fuentes proporcionaron todos los años). Ya que cada fuente tiene sus propios puntos de vista sobre qué puede considerarse un dispositivo IoT, podemos ver que los resultados varían significativamente. Por ejemplo, algunos de ellos como la fuente *IoT Analitics* sólo tiene en cuenta nodos como IoT gateways o concentradores en lugar de considerar cada sensor o actuador. No obstante, independientemente de la precisión de la previsión, la tendencia es clara. IoT es ahora una realidad y la adopción e implantación de tecnologías relacionadas está creciendo a un ritmo vertiginoso. Si bien es cierto que el manejo de esta sobrecogedora cantidad de dispositivos y conexiones representa per se un enorme desafío, la naturaleza de los dispositivos IoT conlleva desafíos específicos que deben ser afrontados para asegurar una adecuada implantación de los dispositivos IoT, así como sus redes en las infraestructuras y dominios actuales. En primer lugar, características como su reducido tamaño, su autonomía, y su precio económico hace asequible despliegues masivos que generan una gran heterogeneidad, la cual dificulta la administración de los mismos, ya que los administradores de sistemas deben lidiar con un gran número de configuraciones dependiendo de las diversas especificaciones y fabricantes, haciendo incluso frente al denominado vendor-locking donde no es posible aplicar configuraciones requeridas en el dispositivo a causa de restricciones del fabricante. Además, los dispositivos IoT están muy limitados en aspectos tales como el consumo de energía, computación o comunicaciones, considerando que muchos dispositivos IoT son desplegados con baterías con el propósito de trabajar de forma autónoma durante largos periodos de tiempo (incluso años) sin ser reemplazados. En ese sentido, estos dispositivos necesitan consumir lo mínimo posible, pero al mismo tiempo deben proporcionar capacidades como sensorización, comunicaciones inalámbricas o incluso capacidades de actuación. Además, aparte del despliegue y las restricciones técnicas, la seguridad se ha

[1] https://www.statista.com/

[2] https://securitytoday.com/articles/2020/01/13/the-iot-rundown-for-2020.aspx

vuelto un desafío principal en los entornos IoT. Debido a la gran cantidad de dispositivos, modelos y firmwares con sus respectivas versiones, las pruebas regulares así como mantener todos los dispositivos en un estado seguro mediante actualizaciones frecuentes se vuelve cada vez más difícil. También, la combinación de su naturaleza limitada y la filosofía de *cuanto más barato mejor* durante la producción, tiende a generar problemas de seguridad desde el principio. Por ejemplo, contraseñas cortas, bien conocidas y sin mecanismos de prevención para ataques de fuerza bruta suelen ser puntos débiles de estos dispositivos. De hecho, las capacidades técnicas restringidas y las contraseñas débiles facilitan en general infecciones del tipo malware y ransomware. A menudo, entornos IoT son infectados utilizando ataques que desbordan sus recursos o ataques de fuerza bruta, propagando la infección rápidamente a lo largo de la infraestructura IoT. De esta forma, enormes cantidades de dispositivos infectados pueden ser manejados por un atacante como si de un ejército se tratase, formando botnets con propósitos maliciosos. Como prueba de ello, hace unos años una cantidad estimada de 100 mil dispositivos IoT generó un ataque botnet de denegación de servicio distribuido (DDoS)[3], consiguiendo velocidades de ataque de hasta 1.2 Tbps. Además, mas allá de la seguridad de los dispositivos en sí, la seguridad y privacidad de los datos también se ha vuelto esencial. Es importante considerar que muchos de estos dispositivos forman parte de nuestras vidas. Miles de millones de sensores están distribuidos por el mundo, recogiendo información sensible sobre las personas y su entorno cada segundo. Si bien este ecosistema fue concebido para proporcionar una significativa mejora en los servicios de usuario, ahora todos los datos obtenidos son frecuentemente el objetivo de los hackers. *El conocimiento es poder.* Si los datos no son protegidos adecuadamente, los atacantes pueden realizar operaciones que relacionan conjuntos de datos con usuarios para inferir nueva información aun más sensible. Por ejemplo, si un atacante consigue obtener un historial de las localizaciones de una persona, éste podría identificar patrones para prever los momentos mas vulnerables de la víctima a lo largo del día, o incluso podría chantajearlo basándose en localizaciones especificas visitadas de aspecto privado.

Si bien es cierto, que esta gran cantidad de dispositivos IoT con sus desafíos de seguridad actuales ya son preocupantes, el escenario se vuelve aun más complejo si también consideramos el torrente de la virtualización. La virtualización de dispositivos está creciendo realmente rápido tanto en entornos de computación, funciones de red virtuales (NFV) y servicios entre otros. Debido a este vertiginoso incremento en dispositivos físicos y virtuales, así como su heterogeneidad, el despliegue, administración y manejo de la seguridad se han vuelto un desafío fundamental que debe ser considerado. De hecho, los pilares de este desafío no son nuevos. En el campo de redes y telemática, problemas similares fueron observados para llevar a cabo un manejo eficiente de distintos tipos de elementos de red, los cuales son también masivamente desplegados por diferentes fabricantes con distintas implementaciones. Con el fin de hacer más flexible el manejo de la red en este aspecto, emergen las soluciones basadas en redes definidas por software (SDN), las cuales crecen en popularidad. Este paradigma permite añadir un alto nivel de abstracción sobre los dispositivos de red proporcionando un control común centralizado, algo que podría ser realmente interesante para el paradigma IoT.

En este sentido, las nuevas generaciones de redes tales como 5G presentan un ejemplo claro de la necesidad de afrontar escenarios dinámicos y flexibles los cuales requieren altas capacidades de automatización y procesos de mantenimiento llevados a cabo por la infraestructura y/o por los administradores. Mediante la integración de tecnologías como SDN y NFV, así como de varios elementos de orquestación, es posible cubrir esta necesidad e incorporar nuevas características como el completo manejo del ciclo de vida de todos los elementos de la infraestructura. No obstante, no solo debemos centrarnos en automatización. Los desafíos referentes a seguridad también crecen proporcionalmente y una apropiada integración de las tecnologías SDN y NFV pueden proporcionar características interesantes para dotar a la infraestructura de nuevas medidas de seguridad de forma flexible y dinámica. Considerando estas poderosas herramientas para llevar la seguridad a un nuevo nivel, ahora, la necesidad de homogeneizar los aspectos de seguridad, evitando lidiar con cada implementación y configuración para cada componente es esencial. Para ello, la seguridad basada en políticas permite incluso a usuarios no técnicos definir políticas de seguridad, utilizando términos de alto nivel, que serán

[3] https://www.theguardian.com/technology/2016/oct/26/ddos-attack-dyn-mirai-botnet

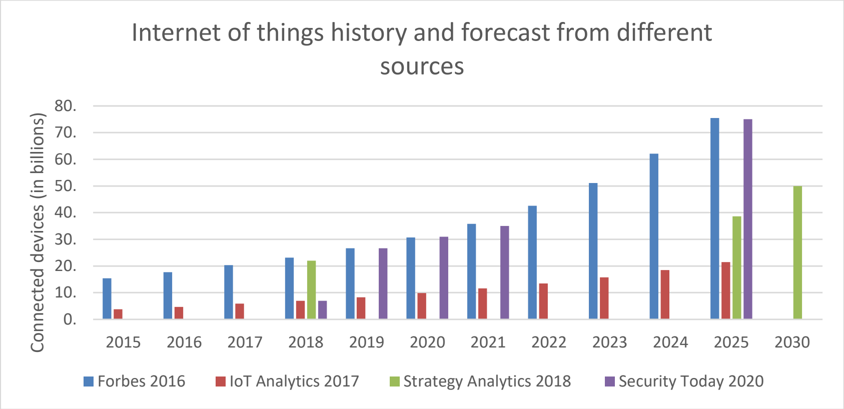

Figura 1.1: Internet de las cosas: Histórico y previsión desde distintas fuentes.

refinados en diferentes niveles de abstracción para finalmente ser aplicados sobre distintos elementos que forman las redes de nueva generación tales como controladores SDN, manejadores NFV o componentes de seguridad específicamente diseñados para ello.

Si bien es cierto que soluciones proporcionadas por la comunidad investigadora a lo largo de diversos proyectos europeos tales como DESEREC [37], SELFNET [46], SECURED [54] o en publicaciones como [72], [81] o [80] proporcionan avances significativos sobre la mitigación de algunos de los desafíos mencionados previamente, estos suelen centrarse en redes tradicionales o puntualmente en IoT para casos de uso muy concretos. Así, al comienzo de esta tesis no existían soluciones interoperables (o al menos no lo suficientemente maduras) especialmente diseñadas para IoT, las cuales automaticen la gestión de la seguridad de forma dinámica y reactiva apoyándose conjuntamente en políticas de seguridad, tecnologías SDN y NFV. En ese sentido, se requiere que las nuevas soluciones consideren las necesidades específicas de seguridad en el ámbito IoT, para que ahora sean parte esencial de las políticas de seguridad de los sistemas. También deben proporcionar complementos y componentes de seguridad necesarios para aplicar dichas políticas mediante nuevas capacidades de orquestación y auto-curación que mantengan un alto nivel de seguridad permanente de forma autónoma. Con estas propiedades, las nuevas soluciones serían capaces de lidiar con desafíos como la configuración de seguridad de despliegues masivos de dispositivos IoT, su evolución dinámica, heterogeneidad, así como su naturaleza de características limitadas. No obstante, encontramos una carencia en el estado del arte en relación a soluciones que cubran todas estas necesidades, las cuales serán el foco principal bajo los distintos objetivos de este trabajo.

La presente tesis presenta el resultado de investigación, diseño e implementación de un framework reactivo basado en políticas de seguridad para redes IoT sobre nuevas infraestructuras SDN y NFV. El framework permite a los administradores de seguridad modelar y definir de forma proactiva nuevas políticas de seguridad a distintos niveles de abstracción, dependiendo de la complejidad requerida y su nivel de conocimiento, así como reutilizar las ya existentes en un repositorio común. Las políticas de seguridad de más alto nivel de abstracción pueden ser refinadas en políticas de seguridad de nivel medio, más complejas, las cuales serán traducidas a configuraciones finales, aplicables sobre cada dispositivo final o componente de seguridad de forma automática. Por ejemplo, las políticas de redirección de trafico pueden ser traducidas a IPTABLES o a reglas SDN dependiendo del despliegue actual. El manejo de las nuevas políticas de orquestación de la seguridad ha sido también dotado con detección de conflictos y dependencias para garantizar su correcta instanciación en el sistema. A su vez, gracias al manejo de las políticas de seguridad para entornos IoT, así como la capacidad de su aplicación en distintos componentes de seguridad según el despliegue actual, el framework contribuye a la mitigación de desafíos como la heterogeneidad, los despliegues masivos o el vendor-locking. Por otro lado, la instanciación dinámica de nuevas funciones de red virtuales mediante SDN y NFV permiten desplegar componentes de seguridad bajo demanda en ubicaciones especificas, pudiendo dotar de nuevas funciones de seguridad avanzadas en diversos entornos IoT. Para proporcionar capacidades de auto-curación y auto-reparación, el framework también es capaz de generar y aplicar automáticamente políticas de orquestación de seguridad reactivas, las cuales conformarán un plan de mitigación. Esta nueva capacidad reactiva permite al framework mantener un nivel constante de seguridad en el sistema, acorde a las amenazas detectadas o a las medidas de seguridad recomendadas a lo largo del tiempo. Para validar nuestra propuesta, los nuevos modelos de políticas de seguridad, los componentes del framework, sus flujos, las nuevas VNFs y su aplicación sobre redes IoT mediante tecnologías SDN y NFV fueron validados en diversos artículos publicados en revistas de alto impacto, así como en distintos casos de uso sobre el proyecto Europeo ANASTACIA H-2020[4]. En ese sentido, el resultado de esta tesis proporciona lo que consideramos una nueva valiosa referencia en seguridad sobre entornos IoT.

La presente tesis doctoral fue llevada a cabo en el marco del programa FPI del gobierno Español (ref. PRE2018- 083731)

1.2. Objetivos y Metodología

Para contribuir a la investigación que trata los distintos desafíos expuestos en la sección previa, el objetivo de esta tesis se centra principalmente en la investigación de un framework con propiedades de auto-curación y auto-reparación mediante diversas capacidades de orquestación basadas en políticas de seguridad, el cual persigue la automatización del manejo de la seguridad de infraestructuras complejas, mitigando problemas como el vendor-locking, lidiando a su vez con la heterogeneidad, entornos de recursos limitados y despliegues masivos que caracterizan la naturaleza IoT. Este framework proporciona un ciclo completo de seguridad compuesto por la definición, orquestación y aplicación proactiva de políticas de seguridad, monitorización de la infraestructura, y reacción ante nuevas amenazas mediante la orquestación y aplicación de nuevas políticas de seguridad reactivas. Así, por un lado las políticas de seguridad proactivas permiten a los administradores de seguridad definir, refinar, traducir y aplicar políticas de seguridad a un nivel de abstracción alto/medio sin necesidad de conocer en detalle la infraestructura subyacente. Por otro lado, las políticas de seguridad reactivas proporcionan al framework capacidades como auto-curación o auto-reparación ya que es capaz de aplicar diferentes contramedidas dependiendo de las amenazas identificadas por las herramientas de monitorización y reacción. Por ejemplo, una política de seguridad reactiva podría aislar una parte de la infraestructura o incluso desplegar complejas contramedidas como una réplica virtual completa de un entorno IoT real (IoT honeynet) para obtener más información sobre el ataque en curso. Para lograr el objetivo principal de esta tesis, éste se ha dividido en los siguientes objetivos.

- Objetivo 1: Definición y extensión de nuevos **modelos de políticas** de seguridad a partir del estado del arte, proporcionando capacidades de seguridad para entornos IoT, tales como autenticación, autorización, privacidad, protección de las comunicaciones, monitorización, administración de dispositivos, replicación de entornos IoT reales en IoT honeynets o calidad de servicio.

- Objetivo 2: Diseñar la **arquitectura** del framework basado en políticas de seguridad capaz de aplicar dichas políticas de forma proactiva y reactiva según el estado actual del sistema, así como manejar su ciclo de vida completo.

- Objetivo 3: Diseño, implementación y validación de los procesos de **refinamiento y traducción de políticas** de seguridad para IoT a partir del estado del arte, con el fin de definir, modelar y transformar éstas entre distintos niveles de abstracción según sea requerido (alto/medio).

- Objetivo 4: Diseño, implementación y validación del proceso de **orquestación y aplicación de políticas de forma proactiva y reactiva**, considerando distintas prioridades, así como posibles conflictos y dependencias.

[4] http://www.anastacia-h2020.eu/

- Objetivo 5: Implementación, despliegue y validación de nuevos mecanismos o **funciones de seguridad especialmente diseñados para IoT**, basados en NFV-SDN y dirigidos por políticas, los cuales permitan a) filtrado y tratamiento de flujos de red, b) autenticación, autorización, c) protección de las comunicaciones, d) control de dispositivos IoT, e) despliegue de IoT honeynets virtuales para mitigar ciberataques.

- Objetvo 6: **Validación y evaluación del framework** propuesto para distintos casos de uso y escenarios reales sobre los que se aplican políticas de seguridad de forma proactiva y reactiva utilizando los nuevos mecanismos y funciones de seguridad especialmente diseñados para IoT, basados en SDN/NFV.

Para alcanzar estos objetivos el esfuerzo fue dividido en diferentes bloques los cuales corresponden a cada uno de ellos con el fin de satisfacer el objetivo final. De acuerdo con esto, se aplicó una metodología iterativa incremental sobre cada bloque y entre bloques. Así, sobre cada bloque se realizó un análisis de requisitos, estado del arte, diseño de la solución, implementación de prueba de concepto, configuración, despliegue, evaluación y análisis de los resultados. Estos últimos proporcionaron nuevo conocimiento para refinar las siguientes iteraciones en el mismo bloque, así como sus posibles interacciones con el resto. De esta forma, cada bloque fue refinado a lo largo del proyecto, contribuyendo a la solución final del mismo. Específicamente, analizamos el estado del arte para los modelos de políticas de seguridad, frameworks basados en políticas, así como soluciones que integran SDN/NFV. Entonces, seleccionamos un modelo de políticas que consideramos acorde a nuestros objetivos. Estos son, High-level Security Policy Language (HSPL) y Medium-level Security Policy Language (MSPL) [16]. Extendimos y actualizamos los modelos para proporcionar nuevas capacidades para entornos IoT como la administración de dispositivos IoT, despliegue de IoT honeynets virtuales, filtrado y reenvío de tráfico, autenticación, autorización, protección de las comunicaciones, así como la combinación de múltiples políticas de seguridad mediante las políticas de orquestación, considerando prioridades, conflictos y dependencias entre ellas. Cada nuevo o extendido modelo de políticas fue validado sobre nuevos componentes de seguridad, especialmente diseñados o adaptados para IoT, a lo largo de las distintas versiones del framework cuya implementación fue evolucionando durante la tesis como resultado de cada iteración.

1.3. Resultados

Durante el periodo de esta tesis, la metodología iterativa sobre los objetivos produjo múltiples resultados tales como **un capítulo de libro, un artículo de conferencia y nueve publicaciones indexadas en JCR, de las cuales cinco conforman el compendio de la tesis**. Debido a que los resultados de la misma fueron también validados durante el proyecto europeo ANASTACIA H-2020, a lo largo de éste también fueron producidos múltiples informes técnicos (más de 20 entregables de proyecto europeo). La tabla 1.1 muestra los resultados principales conseguidos durante la tesis, así como su relación entre los resultados, objetivos y publicaciones.

Tras el análisis del estado del arte, seleccionamos un modelo de políticas de seguridad para ser extendido de acuerdo con los requisitos establecidos para alcanzar los objetivos. Diseñamos una primera arquitectura modular e implementamos una primera prueba de concepto que integraba los modelos de políticas, su transformación, tecnologías SDN e IoT siguiendo una aproximación basada en plugins y controladores para cada componente de seguridad. Así, durante la fase de diseño, el propósito fue definir un framework genérico capaz de gestionar diversos tipos de funciones de seguridad virtualizadas especialmente diseñadas para IoT, que fueran orquestadas de forma interoperable, dinámica y eficiente, a partir de políticas de seguridad, considerando también posibles conflictos durante su despliegue. A partir de esta premisa, proporcionamos soluciones a diversos problemas de gestión de seguridad sobre entornos IoT, instanciando, evolucionando y validando el framework con nuevas implementaciones específicamente diseñadas para este fin.

Tabla 1.1: Principales resultados de la tesis

Resultado	**Objetivos**	**Publicaciones**
R1. Diseño de un framework basado en políticas de seguridad, el cual permite definir, modelar y administrar dichas políticas a distintos niveles de abstracción, a partir del estado del arte.	1, 2	[122] [124] [125]
R2.Diseño de un proceso de refinamiento el cual considera la infraestructura existente para refinar políticas de un alto nivel de abstracción (HSPL) en políticas de un nivel medio de abstracción (MSPL).	1, 2, 3	[122] [124]
R3.Diseño proactivo y reactivo del proceso de traducción de políticas de un nivel medio de abstracción (MSPL), considerando la infraestructura existente para transformarlas en configuraciones especificas, aplicables sobre distintos componentes de seguridad.	1, 2, 3	[122] [124] [125] [127] [128] [129]
R4. Diseño del proceso de aplicación de las configuraciones sobre los componentes de seguridad mediante una aproximación basada en controladores.	1, 2, 3, 5	[122] [124] [125] [127] [128] [129] [131]
R5.Diseño, Implementación y validación de políticas de orquestación, así como del proceso de aplicación de múltiples políticas, considerando la detección de conflictos y dependencias entre las mismas.	1, 3, 4	[130]
R6.Implementación y validación de los procesos de refinamiento proactivo, traducción proactiva/reactiva y aplicación de políticas de seguridad basada en plugins y controladores.	3, 5, 6	[122] [124] [125] [127] [128] [129] [131]
R7.Implementación y validación de nuevos componentes de seguridad, capaces de llevar a cabo tareas de gestión de tráfico IoT mediante la (re)configuración dinámica de la red sobre distintos controladores SDN y funciones de red virtuales.	5a, 6	[122]
R8.Implementación y validación de políticas de autenticación, autorización y protección de las comunicaciones sobre entornos IoT mediante el despliegue dinámico de servicios AAA y la reconfiguración dinámica de la red SDN.	5b, 5c, 6	[124]
R9.Implementación y validación de políticas de monitorización, redirección de tráfico, filtrado y control de dispositivos IoT mediante la reconfiguración dinámica de la red SDN y el controlador IoT diseñado e implementado especialmente para este propósito.	5d, 6	[125]
R10.Implementación y validación de políticas de virtualización de redes IoT (IoT honeynet), y redirección de tráfico mediante la reconfiguración dinámica de la red SDN y el despliegue dinámico de VNFs diseñadas e implementadas para este propósito, capaces de replicar entornos completos IoT.	5e, 6	[129]

En concreto, los primeros experimentos fueron realizados para cubrir casos de uso en los cuales el administrador de seguridad aplica de forma proactiva políticas de gestión de tráfico de red IoT mediante la reconfiguración de la red SDN. También se incluyó el diseño e implementación de una función de red virtual de seguridad (vNSF) haciendo la vez de router virtual, con el fin de establecer una comparativa. Los resultados de estos experimentos proporcionaron nuestra primera publicación [122]. Basándonos en este primer paso, el diseño de la arquitectura fue extendido para manejar el acceso a la infraestructura de los dispositivos IoT mediante el uso de servicios AAA dinámicos, así como la protección de las comunicaciones a través de nuevos modelos de políticas y componentes de seguridad como agentes

PANA virtuales, PDPs basados en XACML y proxies DTLS, que pueden ser desplegados dinámicamente como VNFs. Los resultados proporcionaron un registro dinámico de dispositivos IoT, configuración proactiva de la autenticación, y autorización dinámica reactiva basada en políticas de seguridad para los dispositivos IoT autenticados. Éstos fueron proporcionados en nuestra segunda publicación [124]. Considerando los avances proporcionados por nuestra propuesta, los componentes fueron integrados con componentes de monitorización y reacción en el ámbito del proyecto europeo ANASTACIA H-2020 para validar el ciclo completo de autocuración. En este caso, definimos nuevas políticas de seguridad específicas para monitorización y administración de dispositivos IoT a distintos niveles de abstracción. Herramientas de monitorización avanzadas fueron configuradas de forma proactiva por las nuevas políticas de seguridad, con las cuales se podían detectar comportamientos anómalo de los dispositivos IoT. El sistema entonces era capaz de generar distintas políticas de seguridad reactivas para llevar a cabo acciones de filtrado de tráfico y administración de dispositivos IoT, que fueron aplicadas sobre la red SDN y nuestro controlador de dispositivos IoT respectivamente. Los resultados de esta investigación fueron publicados en nuestro tercer artículo JCR [125]. Una vez validado el ciclo de auto-curación, definimos nuevas y avanzadas contramedidas como el despliegue dinámico de IoT honeynets, las cuales permiten replicar virtualmente entornos IoT físicos. Para ello, proporcionamos nuevas políticas capaces de modelar redes de dispositivos IoT, considerando la información disponible sobre la infraestructura, adquirida durante el registro de los dispositivos gracias a nuestros resultados anteriores. También se diseñó e implementó el manejador de IoT honeynets como un nuevo componente de seguridad. Así, combinando la configuración de red dinámica y el despliegue bajo demanda de IoT honeynets, es posible utilizar esta nueva funcionalidad de forma reactiva y transparente para redirigir a un atacante a un entorno IoT virtual controlado durante un ataque en curso, con distintos fines. Los resultados de esta investigación aparecen en nuestra cuarta publicación [129]. Finalmente, considerando la necesidad de aplicar múltiples políticas de seguridad, así como establecer sus prioridades y dependencias, extendimos los modelos para proporcionar lo que denominamos políticas de orquestación. De esta forma, una política de orquestación puede representar un plan de aplicación de políticas indicando su orden, prioridades y dependencias entre políticas o eventos. Por ejemplo, en el caso de la IoT honeynet reactiva y transparente, el tráfico sólo debe ser redirigido cuando la honeynet ha sido desplegada apropiadamente. Para el manejo y la correcta instanciación de las políticas, también proporcionamos un detector de conflictos y dependencias con el fin de asegurar que las políticas de orquestación son aplicadas de forma adecuada. Éstos aspectos y los experimentos asociados componen nuestra quinta publicación [130].

El resultado final de implementación como prueba de concepto sigue evolucionando y de hecho algunas partes del mismo son consideradas para nuevos proyectos Europeos (e.j., h2020 INSPIRE 5G+). Por supuesto, la implementación es de código abierto y es almacenada en el repositorio del grupo de investigación del departamento. A continuación, el lector puede encontrar un resumen más detallado sobre cada una de las publicaciones que componen el compendio de esta tesis. Además, las publicaciones completas pueden consultarse en el capítulo 4.

1.3.1. Enhancing IoT security through network softwarization and virtual security appliances

Esta primera publicación [122] avanza un paso más la gestión de red mediante políticas de seguridad, extendiendo su aplicabilidad sobre entornos IoT mediante distintos controladores de red SDN, así como instanciando cortafuegos virtuales (vFirewalls), estableciendo también una comparativa entre los mismos. En primer lugar, proporciona un análisis de integración y aplicación de las características SDN y NFV con el fin de mejorar la seguridad en redes IoT. En concreto, explica interesantes capacidades de SDN aplicables al ámbito de la seguridad tales como el control o manipulación dinámico de flujos, el cual permite reconfigurar el comportamiento de la red de acuerdo con las especificaciones de seguridad, (e.j., aislamiento de red) así como manipular campos específicos de los paquetes. El manejo centralizado de la SDN y la monitorización de dispositivos SDN también es resaltado para analizar y verificar el estado actual de la red en términos de seguridad (e.j., detección de picos anómalos de tráfico en la red IoT). De la misma forma que para SDN, la publicación también proporciona características de NFV aplicables a la seguridad, tales como el desacoplamiento de las funciones de seguridad del hardware, evitando así el vendor-locking y facilitando la escalabilidad bajo demanda así como la movilidad. Propiedades que ahora permiten desplegar, migrar y escalar en ambos sentidos funciones de red virtuales cuando y donde sea requerido, proporcionando nuevos servicios de seguridad a dispositivos IoT. Una vez el articulo ha expuesto los beneficios de aplicar características de SDN y NFV para mejorar la seguridad en el ámbito de IoT, éste proporciona un primer diseño de la arquitectura del framework a alto nivel (R1), el cual se compone de tres planos principales. El plano de usuario, donde el administrador de seguridad introduce políticas de seguridad de alto nivel en el sistema, el plano de orquestación, enfocado en la orquestación del framework, y el plano de aplicación de políticas, donde las políticas son aplicadas utilizando distintos habilitadores de seguridad existentes (o no) sobre la infraestructura. En esta línea, se establece un primer diseño de los flujos de transformación y aplicación de políticas (R2, R3, R4), así como posibles casos de uso como un sistema de gestión de edificios o un sistema de computación en los límites de la nube (Edge).

Finalmente, la publicación proporciona una primera prueba de concepto de la implementación del framework, capaz de recibir políticas de seguridad de filtrado de alto nivel las cuales contienen valores en lenguaje tradicional (e.j., Sensor 1), que son refinadas en políticas de seguridad de nivel medio, donde los valores de alto nivel son transformados en conceptos entendibles por una máquina (e.j., IPs y puertos). Una vez las políticas de filtrado han sido refinadas, la publicación también muestra una comparación de múltiples traducciones y aplicación de políticas sobre distintos componentes de seguridad a lo largo de la infraestructura (R6, R7). Estos fueron, ONOS, Opendaylight (OpenFlow) y un router virtual (NETCONF).

1.3.2. Enabling Virtual AAA Management in SDN-Based IoT Networks

Esta segunda publicación [124] se centra en proporcionar una gestión dinámica del framework basado en políticas de seguridad para las capacidades de protección de las comunicaciones, autenticación y autorización (parte de AAA), la cual es llevada más allá del estado del arte para ser interoperable con entornos IoT. La solución contempla el proceso de unión a la red, permitiendo además su despliegue dinámico en los límites de la nube (edge), lo más cercano posible de los dispositivos finales. En concreto, proporciona distintos flujos de trabajo y operaciones para permitir que los dispositivos IoT accedan a recursos específicos (incluso a la propia red) de forma segura. Para ello, dado que se asume una política de denegación de acceso por defecto, primero se define el proceso por el cual el administrador de seguridad autoriza de forma proactiva el tráfico de autenticación en la red para los dispositivos IoT (R1, R2), así como las posibles interacciones que pudieran tener con la infraestructura en un futuro. Por ejemplo, especificando que determinados dispositivos IoT podrán almacenar valores de temperatura en el IoT broker una vez autenticados. Este tipo de autorización de tráfico, a diferencia de enfoques tradicionales, se instancia dinámicamente en la SDN mediante la modificación de las reglas de flujo, para permitir el tipo de protocolo de transporte de autenticación utilizado por los dispositivos IoT. El artículo también proporciona ejemplos de políticas de seguridad para autorizar el acceso a los recursos, así como el desvío del tráfico. Una vez se permite el tráfico para el protocolo de transporte de autenticación IoT, los dispositivos IoT son capaces de realizar el proceso de autenticación que finaliza con la generación de una Master Session Key (MSK), así como con el registro de los nuevos dispositivos IoT en el sistema a través del controlador de dispositivos IoT (IoT Controller), el cual también diseñamos e implementamos. Cuando un dispositivo IoT autenticado intenta acceder a algún recurso, primero recuperará un token de capacidad que indica su papel en la infraestructura, tal como el administrador de seguridad especificó en las políticas proactivas. Una adquisición exitosa de este token genera una nueva política de autorización (R3) que permite el tráfico SDN según los permisos específicos otorgados al dispositivo IoT.

La publicación también proporciona el proceso para refinar, traducir y hacer cumplir las políticas de protección de las comunicaciones (R3, R4, R6), específicamente, mediante conexiones DTLS entre los dispositivos IoT y los componentes de seguridad (e.j., IoT broker o DTLS proxy). En este caso, cuando se solicita la aplicación de la protección las comunicaciones, tras el refinamiento y la traducción de la política, se genera una nueva clave maestra de aplicación de políticas (e.j., clave maestra PaC-EP o PEMK en el caso del protocolo PANA), la cual se incluye en las configuraciones del componente de seguridad, que serán distribuidas utilizando un canal seguro. El componente de seguridad (generalmente, el IoT broker o el proxy) utiliza entonces la PEMK como clave compartida para preparar el canal DTLS con el dispositivo IoT, el cual también recibe la solicitud de aplicación de protección de las comunicaciones a través del controlador de IoT. Finalmente, la publicación muestra mediciones y resultados para todo el proceso (R8), incluida la autorización de redes, recursos, la protección de las comunicaciones y las interacciones de seguridad de los dispositivos IoT, combinando la potencia de las redes SDN, el despliegue virtual dinámico mediante NFV y el manejo de aspectos específicos de seguridad de IoT mediante su controlador (IoT Controller).

1.3.3. Security Management Architecture for NFV/SDN-aware IoT Systems

La tercera publicación [125] avanza el estado del arte actual sobre monitorización, detección y reacción en entornos tradicionales, incluyendo componentes de monitorización IoT (e.j., 6LowPAN) y políticas de seguridad IoT reactivas (e.j., IoT Control) sobre la infraestructura SDN/NFV. Enfoque que aun no había sido tratado en profundidad en trabajos anteriores. También proporciona la implementación de nuevas funcionalidades sobre nuestro controlador IoT para llevar a cabo las nuevas políticas. Así, este trabajo presenta una versión avanzada de la arquitectura del framework (R1, R3), integrada en la arquitectura de ANASTACIA con fines de validación, incluyendo los componentes e interfaces que componen cada plano. El diseño del plano de usuario ahora contempla no solo el editor de políticas de alto nivel para su modelado e instanciación, sino también paneles de control y notificaciones entre otros, para proporcionar a los administradores de seguridad información en tiempo real sobre la seguridad y la privacidad, permitiéndolos interactuar en caso de que sea requerido, dependiendo de la naturaleza del problema. El plano de orquestación de la seguridad se compone entre otros por el intérprete de políticas, a cargo de refinar y traducir políticas de seguridad (R3, R4) para generar las configuraciones finales de los componentes de seguridad (e.j., reglas SDN). En el mismo plano, el proveedor de componentes de seguridad está a cargo de proporcionar plugins que contienen la lógica de traducción entre políticas de nivel medio y componentes. El orquestador de seguridad, obviamente en el plano de orquestación, se hace cargo de orquestar y hacer cumplir políticas de seguridad proactivas y reactivas a lo largo de la infraestructura.

Para generar las políticas reactivas, se utilizan los módulos del plano de monitorización y reacción, los cuales fueron proporcionados por otros autores del artículo, e integrados en la arquitectura. El modulo de monitorización, adquiere información de distintos agentes de monitorización (e.j., Snort o agentes MMT [56]) que se encuentran en el plano de aplicación de políticas (distintos puntos de la infraestructura). Partiendo de esta información, éste módulo notifica cualquier tipo de incidente a un sistema de decisión y veredictos localizado en el módulo de reacción, el cual analiza y establece una correlación de la información para decidir si se requiere una contramedida. En ese caso, un servicio de mitigación genera nuevas políticas reactivas para ser aplicadas en la infraestructura. Todos los procesos mencionados, así como un ejemplo de política reactiva de filtrado son detallados en el artículo, incluyendo también nuevos flujos de trabajo para las partes de monitorización y reacción. Tras proporcionar detalles sobre el diseño de la arquitectura, la publicación muestra un conjunto de amenazas o ataques sobre infraestructuras IoT y cómo el nuevo diseño es capaz de mitigarlos. Por ejemplo, un conjunto de dispositivos IoT infectados con un malware pueden ser aislados del resto de la infraestructura mediante la administración de la red SDN, mientras al mismo tiempo se podrían realizar modificaciones al firmware a través del controlador IoT (políticas de filtrado y administración de IoT).

Con el fin de validar el nuevo diseño, siguiendo el paradigma iterativo incremental, una nueva versión de la arquitectura fue implementada y desplegada donde se realizaron simulaciones para distintos escenarios (R6, R9). Un escenario de computación móvil en el Edge y un escenario sobre un sistema de administración de un edificio. En el primer caso, varios dispositivos IoT 6LowPAN enviaban mensajes continuamente a un objetivo específico, y un inspector de tráfico (DPI) desplegado sobre la 6LowPAN detectaba y notificaba la amenaza al modulo de monitorización, el cual disparaba una reacción que filtraba el tráfico mediante la SDN. En el segundo caso, un dispositivo IoT fue manipulado para notificar valores anómalos de temperatura con el objetivo de hacer sonar la alarma de incendios. En este caso, se utilizó una herramienta de monitorización más sofisticada [121] para discernir entre valores regulares y situaciones anómalas según un aprendizaje previo del entorno. De esta manera, el modulo de monitorización notificaba el fallo al modulo de reacción, el cual generaba nuevas políticas reactivas para reiniciar o finalmente apagar el dispositivo IoT hasta que fuese comprobado físicamente. Después de que el administrador de seguridad verificase que no hay un riesgo real, la alarma antiincendios también volvía a la normalidad. Finalmente, la publicación proporciona una evaluación de rendimiento donde se enviaron distintas ráfagas de incidentes las cuales dispararon los procesos de reacción que aplicaron nuevas políticas SDN e IoT respectivamente.

1.3.4. Virtual IoT HoneyNets to Mitigate Cyberattacks in SDN/NFV-Enabled IoT Networks

A diferencia de los trabajos analizados sobre honeynets en el estado del arte, de los cuales ninguno se basa en una combinación completa de políticas de seguridad, SDN/NFV e IoT y todos despliegan las honeynets de forma estática y proactiva para simular servicios o redes tradicionales, esta cuarta publicación [129] proporciona resultados sobre la capacidad del framework para desplegar VNFs dinámicas capaces de replicar infraestructuras IoT reales a partir de un modelo actualizado del sistema. Este despliegue está gobernado por políticas IoT honeynet, así como políticas de control de red las cuales aplican redirecciones de tráfico transparente mediante la red SDN. El trabajo por tanto se enfoca en proporcionar al framework capacidades de virtualización de entornos IoT, transformándolos en IoT honeynets de alta interacción. Una honeynet de alta interacción se compone por un conjunto de honeypots de alta interacción los cuales simulan tanto como sea posible un despliegue real (e.j., una virtualización completa de un dispositivo IoT incluyendo sus recursos e interfaces). En el lado opuesto, una honeynet de baja interacción se compone por honeypots de baja interacción los cuales solo simulan ciertas partes del entorno real (e.j., solo una respuesta ICMP). Para proporcionar nuestra solución, la publicación muestra una comparativa sobre el estado del arte entre distintas soluciones, considerando características importantes como, si la solución está basada en políticas, si utilizan SDN y/o NFV, o si proporcionan capacidades dinámicas (e.j., despliegues dinámicos de la honeynet). Tras el estudio del estado del arte, nuestra nueva propuesta se centró en extender el framework para manejar nuevos modelos de políticas para IoT honeynets. Específicamente, extendimos el Technology Independent Honeynet Descrption Language (TIHDL) [26] con nuevos tipos específicos para IoT como la IoT honeynet, IoT router y IoT honeypot. De esta forma, una IoT honeynet se compone de una serie de routers, gateways y honeypots. Un IoT honeypot es capaz de representar información tal como el nivel de interacción, interfaces, firmware, software, modelo, ubicación y recursos. Los routers IoT especifican parámetros similares a los IoT honeypots, pero también proporcionan información de enrutamiento. Ésta extensión del lenguaje fue entonces homogeneizada en Medium-level Security Policy language (MSPL) y empleada como una nueva capacidad del framework.

Una vez se extendió el modelo de política, el artículo proporciona explicaciones detalladas sobre la administración basada en políticas para la nueva contramedida, esta vez proporcionando también un algoritmo para lidiar con dependencias durante el proceso de orquestación, y cómo este proceso puede ser integrado en el framework. Con esta nueva capacidad, cuando el modulo de monitorización detecta algún tipo de incidencia, el modulo de reacción puede escoger el despliegue dinámico de IoT honeynets como parte de una mitigación. Se generará entonces una nueva MSPL de IoT honeynet a partir de la información del despliegue real de IoT, disponible gracias al proceso de registro ([124] muestra el proceso de registro automatizado durante el bootstrapping). Esta nueva MSPL reactiva es traducida (R3, R4, R6) utilizando distintos plugins para los componentes de seguridad, capaces de emular entornos IoT dependiendo de la configuración del entorno real (e.j., Cooja para dispositivos

Contiki, o Mininet 6LowPAN con uPython para dispositivos uPython). Debido a que las reglas de filtrado y reenvío de tráfico para redirigir el tráfico desde/hacia la IoT honeynet deben ser aplicadas después de que la IoT honeynet haya sido desplegada completamente, el algoritmo de dependencias entre políticas registra la dependencia, con lo que la política dependiente solo será aplicada cuando su dependencia haya sido solventada (e.j., la IoT honeynet ha sido desplegada y configurada). Para validar la solución, la publicación también proporciona detalles sobre la implementación, incluyendo los nuevos modelos MSPL, así como sus traducciones y medidas de rendimiento. Específicamente, evaluamos dos entornos reales compuestos por distinto número y modelos de dispositivos IoT (R10), colocados en distintas topologías, proporcionando mediciones para el proceso completo. También se consideraron tiempos de convergencia en el enrutamiento, comparando los resultados entre escenarios, así como entre las topologías reales y un despliegue clásico en forma de malla.

1.3.5. Semantic-Aware Security Orchestration in SDN/NFV-Enabled IoT Systems

La última publicación que forma el compendio de esta tesis [130] se enfoca en los nuevos modelos de políticas de orquestación definidos, así como en la detección de conflictos, dependencias y en la optimización de la orquestación (R5). A diferencia de las soluciones hasta el momento relacionadas con frameworks de seguridad para IoT, de las cuales pocas se basan en políticas de seguridad y menos aún consideran procesos de orquestación y detección de conflictos durante su aplicación, este trabajo proporciona no solo nuevas políticas con capacidades de orquestación de la seguridad, sino además la detección de conflictos y dependencias, así como procesos de optimización de la aplicación de dichas políticas sobre la infraestructura. El trabajo muestra un estado del arte sobre la orquestación de seguridad utilizando SDN/NFV en IoT, la optimización de funciones de servicios encadenados (SFC) y el manejo de la seguridad y la red basado en la semántica, los cuales son los temas principales del artículo. Partiendo de esto, resaltamos los nuevos componentes en el plano de orquestación sobre la arquitectura de ANASTACIA, utilizada para validar la propuesta. En este punto, nuevos componentes dentro del orquestador fueron definidos para optimizar el proceso de orquestación teniendo en cuenta los datos proporcionados por el detector de conflictos, el modelo del sistema y datos de monitorización.

No obstante, nuestras principales contribuciones en este trabajo fueron la definición de las políticas de orquestación, así como los flujos completos para manejar conflictos y dependencias ente políticas y eventos durante el proceso en aplicación de políticas. En este sentido, motivamos las políticas de orquestación, apoyándonos en las interacciones de AAA donde múltiples políticas de autenticación y autorización deben ser aplicadas siguiendo un orden específico, también considerando que algunas de ellas solo pueden ser aplicadas de acuerdo a ciertos eventos del sistema. Por ejemplo, el acceso a un determinado recurso por parte de un dispositivo IoT solo puede ser autorizado cuando dicho dispositivo IoT específico ha sido apropiadamente autenticado. Además, el proceso de orquestación debe asegurar que la aplicación de nuevas políticas no generará nuevos conflictos en el sistema. Con este fin, definimos y proporcionamos distintos ejemplos de reglas para la detección de conflictos y dependencias, para considerar conflictos bien conocidos. También proporcionamos reglas para la detección de conflictos basadas en contexto, como los conflictos debido a una capacidad requerida inexistente, o a recursos insuficientes. Estos tipos de conflictos no solo consideran conflictos entre políticas sino también entre políticas y la infraestructura. Una vez definidos los conjuntos de reglas, diseñamos su integración con las políticas y motor de reglas, el cual carga el conocimiento sobre la infraestructura y las políticas como una serie de hechos. Cuando una nueva política va a ser aplicada en el sistema, se verifica contra el conjunto de reglas, las cuales consideran los hechos actuales para proporcionar un veredicto. Teniendo en cuenta los nuevos modelos y funcionalidades, las políticas de seguridad simples se transformaron en políticas de orquestación capaces de establecer orden, prioridades y dependencias entre políticas y eventos. De esta forma, proporcionamos nuevos flujos completos para integrar las políticas de orquestación en el framework, para su aplicación tanto en escenarios proactivos como reactivos. Finalmente, también proporcionamos la validación y evaluación de rendimiento para la nueva implementación según distinto numero de reglas, hechos y políticas de seguridad (R11).

1.4. Conclusiones y Trabajos Futuros

Debido a los problemas que caracterizan los entornos IoT, tales como su heterogeneidad, los despliegues masivos, el vendor-locking, así como su naturaleza de recursos limitados, aparecen nuevos desafíos que amenazan directamente la gestión de la seguridad de estos entornos, así como la seguridad en si misma. Para mitigar estos problemas entre otros, a lo largo de esta tesis se ha llevado a cabo una labor de investigación con el fin de diseñar e implementar un novedoso framework basado en políticas de seguridad, el cual ha sido validado a lo largo de múltiples publicaciones, así como durante el proyecto europeo ANASTACIA H-2020. Nuestra propuesta es capaz de manejar políticas de seguridad a un alto nivel de abstracción, las cuales son independientes a la infraestructura subyacente, desacoplando así los requisitos de seguridad de implementaciones específicas, con el fin de mitigar problemas como la heterogeneidad o el vendor-locking. La modularidad del diseño así como su apropiada integración con SDN, NFV y tecnologías de monitorización, dotan al framework de novedosas capacidades reactivas, dinámicas y flexibles, tales como la automatización de la administración de seguridad y la auto-curación o auto-reparación, como ha sido expuesto en los resultados de la tesis.

En ese sentido, se han proporcionado resultados del diseño, implementación y validación para el aislamiento de dispositivos IoT comprometidos mediante la aplicación de políticas de filtrado de tráfico de alto nivel. Dichas políticas fueron refinadas y traducidas por el framework para obtener las configuraciones de seguridad pertinentes para cada componente de seguridad. Configuraciones que fueron aplicadas durante el proceso de orquestación sobre distintos controladores SDN tales como ONOS u Opendaylight, así como sobre routers virtuales desplegados como VNFs, mostrando los beneficios de la adopción de SDN como mecanismo de seguridad integrado en entornos IoT. Este resultado es un avance importante frente al estado del arte, que no adoptaba e integraba un enfoque SDN/NFV basado en políticas de seguridad para la gestión de la seguridad en IoT.

También se han desarrollado capacidades de AAA dinámicas para entornos IoT, inexistentes hasta el momento, mediante el refinamiento, traducción y aplicación de distintas políticas de autenticación y autorización, así como de protección de las comunicaciones mediante la distribución de material criptográfico a las entidades pertinentes. Las capacidades de autenticación y autorización dinámicas también facilitaron el registro de los dispositivos IoT en el sistema, el cual mantiene la información sobre el estado de la infraestructura. Se han diseñado los flujos de integración, instanciados considerando protocolos de transporte de autenticación como PANA, características de autorización como capability tokens y protección de las comunicaciones como DTLS. En estos nuevos casos también se han desarrollado nuevos plugins y drivers que facilitan la traducción de políticas de nivel medio a configuraciones finales, así como su aplicación por parte del orquestador sobre distintos componentes de seguridad (un proxy DTLS, un PDP XACML y el IoT Controller), que también fueron validados. Así, se proporcionaron resultados del proceso completo desde que el dispositivo comienza su autenticación en el sistema hasta que realiza sus primeras operaciones a través de un canal seguro. Dichos resultados muestran la viabilidad y rendimiento de la solución, basada en una aproximación novedosa que explota SDN/NFV para gestión eficiente de la autenticación y autorización en escenarios IoT.

Los componentes e interacciones del framework también han sido validados sobre el proyecto europeo ANASTACIA, donde se han integrado con los elementos de monitorización y reacción. Así, se han analizado un conjunto de amenazas para entornos IoT y se han propuesto formas mediante las cuales el framework es capaz de reaccionar dinámicamente para mitigarlas. En este aspecto, se han definido nuevas políticas de monitorización, así como las interacciones entre los nuevos módulos sobre distintos escenarios donde múltiples dispositivos IoT fueron comprometidos. Gracias a la monitorización del sistema, la amenazas son detectadas, disparando el proceso de reacción que automáticamente genera nuevas políticas de seguridad para instanciar nuevas contramedidas reactivas, en este caso, a través de la SDN (redirección y filtrado) así como del controlador de IoT (administración de dispositivos). Los escenarios han sido implementados y validados, proporcionando el flujo completo de instanciación de políticas proactivas, monitorización y políticas reactivas.

En cuanto a la parte del framework relativa a la automatización y virtualización basada en NFV de los dispositivos IoT, se han proporcionado los primeros resultados hasta el momento sobre la instanciación dinámica transparente de redes IoT virtuales, que replican entornos IoT reales como una nueva contramedida de seguridad mediante la integración de SDN, NFV y emuladores específicos IoT. Para ello, se han extendido los modelos de políticas para representar redes de dispositivos IoT a partir de un lenguaje existente (TIHDL), capaz de representar conceptos específicos de honeynets. Sobre éste, se han añadido nuevas entidades y relaciones para dispositivos IoT, sus redes y recursos, siendo finalmente integrado como un nuevo modelo de las políticas MSPL (IoT Honeynet). A partir de este nuevo modelo, se ha proporcionado un diseño detallado del despliegue dinámico de IoT honeynets, aprovechando la información de los dispositivos IoT reales obtenida durante el proceso AAA. De esta forma, se han replicado distintos escenarios reales de IoT para distintas infraestructuras, también considerando dependencias entre políticas para asegurar que éstas son aplicadas adecuadamente (e.j., IoT honeynet ha sido desplegada y configurada antes de aplicar la redirección o el filtrado de tráfico). Esta investigación también ha concluido con la implementación y validación de nuevos plugins y procesos para la transformación de políticas de IoT honeynet a configuraciones finales, así como la instanciación dinámica de nuevos componentes de seguridad como el emulador de contiki (Cooja). Para este fin, ha sido desarrollado un agente manejador de IoT honeynets y las nuevas funciones pertinentes sobre el controlador de IoT. Durante las pruebas de validación se ha contemplado todo el proceso de despliegue reactivo para distintas infraestructuras y topologías IoT. Estos resultados muestran la viabilidad y rendimiento de la nueva solución, inexistente hasta el momento, combinando SDN, NFV y diversos entornos IoT virtuales que permiten desplegar bajo demanda IoT honeynets de forma reactiva. La nueva capacidad de reacción puede ser aplicada con distintos fines de seguridad, como redirigir a un atacante de forma transparente al entorno IoT virtualizado, mientras se realizan análisis del ataque en curso de forma segura.

Finalmente, como última contribución de esta tesis, se han diseñado las políticas de orquestación para mejorar notablemente la capacidad de mitigación del sistema. Estos nuevos modelos de políticas son capaces de albergar a su vez múltiples políticas de seguridad, indicando también características necesarias para su orquestación tales como el orden de aplicación, sus prioridades o incluso dependencias entre ellas, o entre las políticas y los eventos del sistema. De esta forma, las políticas de orquestación pueden proporcionar complejos planes de mitigación. Para asegurar que dichas políticas son aplicadas apropiadamente, también ha sido diseñado e implementado un detector de conflictos y dependencias, el cual verifica que las nuevas políticas de seguridad no presentaran conflictos con el estado actual del sistema, de acuerdo con un conjunto de reglas bien definidas. Los resultados de esta investigación fueron validados para distintos números de reglas, hechos y políticas de seguridad.

Es importante resaltar que los resultados de esta tesis, así como la implementación de sus distintos componentes, han sido y están siendo explotados y reutilizados en proyectos europeos H-2020 como ANASTACIA e INSPIRE 5G+.

Sin bien consideramos que el resultado de la tesis proporciona una referencia valiosa en el ámbito de seguridad para IoT, quedan abiertas distintas líneas de investigación que se han ido resaltando durante la evolución de esta tesis. Uno de los focos principales para un trabajo futuro reside en al plano de orquestación de la seguridad. De hecho, actualmente estamos evolucionando la orquestación en relación a cómo seleccionar el mejor componente de seguridad, teniendo en cuenta toda la información almacenada en el modelo del sistema, el cual contiene una representación de la infraestructura subyacente. Para ello, el orquestador debe ser capaz de conocer continuamente la situación actual de la infraestructura, así como sus instancias, servicios, controladores, redes, políticas y propiedades de seguridad para proporcionar esta información a los algoritmos de orquestación. De esta forma, los conflictos de políticas y sus dependencias deben ser consideradas durante la orquestación, no solo teniendo en cuenta situaciones inter o intra políticas sino también entre la propia infraestructura en términos de seguridad, disponibilidad y calidad de servicio. De hecho, existen distintas propuestas sobre algoritmos para ubicar nuevos servicios basados en distintos tipos de optimización como greedy, scored, fuzzy rules, ILP o MILP entre otros. No obstante, los resultados tienden a considerar únicamente recursos disponibles, en lugar de incluir las condiciones de seguridad en el entorno actual. Aparte de esto, también sería muy útil considerar no solo un algoritmo de orquestación, sino múltiples algoritmos para seleccionar el más apropiado de acuerdo con el estado actual del entorno (meta-orchestration algorithm). También, otra línea interesante de investigación trata de no solo programar el plano de control a lo largo de la red SDN sino también el plano de datos. Por ejemplo, nuevos habilitadores de seguridad basados en P4 podrían generar código P4 dependiendo de los dispositivos P4 existentes en la infraestructura (físicos o virtuales). De esta forma podríamos manejar el plano de datos considerando campos y opciones mas allá de la capa de transporte de la pila TCP. Además esto contribuiría significativamente a la aplicación de políticas de calidad de servicio teniendo en cuenta características como la telemetría in-band de red, combinando la información del plano de datos y del plano de control para reconfigurar el balanceo de carga de la misma.

CHAPTER 2

Abstract

2.1. Motivation

At the beginning of this PhD thesis (2017), different sources estimated around twenty-three billion of Internet of Things (hereinafter IoT) devices connected to the network, and statisticians predicted that these numbers could reach up to 50 billion of devices in few years[1]. Now (2020) those estimations consider that every second, around 127 new IoT devices are connected to the network[2], generating the expectations that in five years we could find around 70 billions of IoT devices sharing our environment. Figure 2.1 shows a history and forecast from 2015 to 2030, gathered from different sources (not all sources provided data for all years). Since each source has its own consideration regarding what an IoT device is, we can find that the results vary significantly. For instance, some of them like *IoT Analitics* source only take into account active nodes like IoT gateways or concentrators, instead of considering each sensor or actuator. Nevertheless, independently of the accuracy of the prevision, the trend is clear. IoT is now a reality and the adoption and implantation of IoT related technologies are increasing at vertiginous velocity.

Despite it is true that the proper managing of this overwhelming amount of devices and connections represents per se a huge challenge, the nature of IoT devices entails challenges which must be faced in order to ensure a proper adoption of IoT devices and IoT networks in current infrastructures and domains. In first place, characteristics like the small size, the autonomy, and the cheap price make affordable massive deployments of IoT devices but, at the same time, it generates a huge heterogeneity so the administrators have to deal with a huge number of different configurations depending on multiple specifications and manufacturers, even facing problems like vendor-locking, where it is not possible to configure the device as expected due to manufacturer restrictions. In addition, IoT devices are constrained in several aspects like computation, communications and power consumption, considering some IoT devices are deployed with the aim to work autonomously during long time periods (maybe years) without replacement. In that sense, those devices need to consume as less energy as possible but at the same time they have to provide capabilities like sensing and wireless communications or even actuation. Apart from the deployments and technical issues, security becomes a main challenge in IoT environments. Due to the huge amount of devices, models, firmwares and versions, regular testings as well as to keep all the devices in a safe state, by providing regular updates and upgrades of the firmware becomes harder. Besides, the combination of the constraint nature of IoT devices

[1] https://www.statista.com/

[2] https://securitytoday.com/articles/2020/01/13/the-iot-rundown-for-2020.aspx

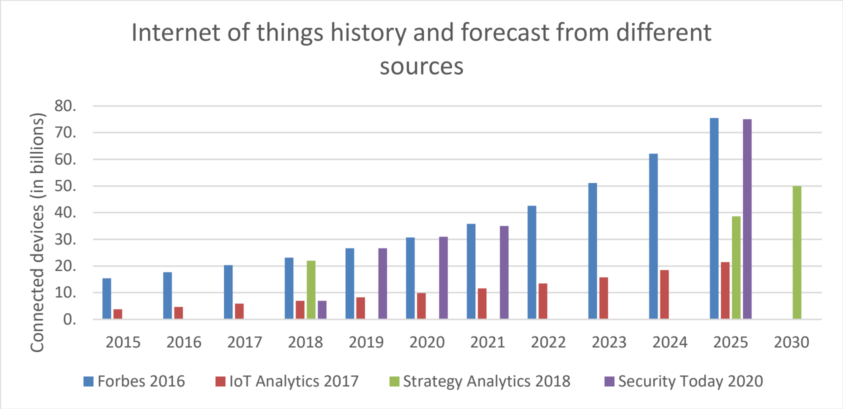

Figure 2.1: Internet of Things evolution and forecast from different sources.

and "*the cheaper the better*" production philosophy trends to generate weak security by default. For instance, well known passwords, or short-length passwords without brute-force attacks prevention mechanisms are a common weak point in this kind of devices. In fact, constraint nature and weak passwords ease in general malware and ransomware infection. Often, IoT environments are infected by performing resources overflow or brute-force attacks, and the infection is spread fast across IoT infrastructures. In this way, huge amounts of infected IoT devices can be also commanded as an army by the attacker, in form of IoT botnets with malicious purposes. Proof of that, few years ago a botnet DDoS attack[3] was largely made by an estimated amount of 100k IoT devices, achieving up to 1.2Tbps of attack speed. Moreover, beyond the security of the IoT devices themselves, data security and data privacy becomes also essential. It is important to take into account that many IoT devices are part of our lives. Billions of sensors are spread around the world, gathering every single second sensible information about people and environment. Despite this ecosystem was conceived for providing better and accuracy services to the users, the retrieved data is now the target of the attackers. "*Knowledge is power*". If data is not properly protected, attackers can perform linking operations among data as well as among users in order to infer even more sensible knowledge. For instance, if an attacker manages to retrieve the location history of a user, he/she could identify patterns in order to foresee the most vulnerable moments in the course of days, or even he/she could perform a blackmail attack based on specific locations visited by the user.

While it is true the forecast of new physical amount of devices as well as the associated security challenges are overwhelming in themselves, the scenario can be even more complex when we also take into account virtualisation. Device virtualization is growing really fast, both for computing environments, and network functions virtualisation, hereinafter NFV, as well as for its services. Due to this vertiginous increasing in both physical and virtualized devices and their heterogeneity, the deployment, administration and security management has become a fundamental challenge to be addressed. In fact, pillars of this challenge are not new. In the field of networks and telematics, similar problems have already been observed in order to carry out efficient management of different possible types of network elements, since these are also usually provided from different manufacturers with different implementations. In order to make more flexible the management of these network devices, solutions based on the use of software-defined networks, hereinafter SDN, are growing in popularity. This approach allows adding a higher level of abstraction on the final network devices, providing centralized common control, something which could be really useful in the IoT paradigm.

In that sense, new networks generation like 5G presents a clear example of the need to face dynamic and flexible scenarios which require strong automation and maintenance processes, carried out directly by the infrastructure and/or by system administrators. By integrating technologies such as SDN, NFV

[3] https://www.theguardian.com/technology/2016/oct/26/ddos-attack-dyn-mirai-botnet

and various elements of orchestration, it is possible to meet this need and to incorporate new desirable characteristics such as the management of the complete life cycle of all elements of the infrastructure. However, we should not just focus on automation. As we mentioned before, an important challenge such as security also grows proportionally, and a proper integration of SDN and NFV can provide interesting features in terms of security measures in a flexible and dynamic way [10-13]. Considering these powerful tools to apply dynamically security measures over the whole infrastructure, now, the need to homogenize security aspects, avoiding dealing with every single implementation and configuration for each component involved in the security requirements is essential. To this aim, policy-based security allows even non-technical users to define security policies at high-level terms (human-readable terms) which will be refined and translated at different levels of abstraction, for finally to be enforced over the different elements such as those that form the basis of new generation networks.

While it is true the provided solutions by the research community have provided significant advances in the aforementioned challenges during diverse EU projects such as DESEREC [37], SELFNET [46], SECURED [54] or publications like [72], [81], [80], those results are commonly focused on traditional networks, and IoT subject is often only addressed for very specific use cases. Thus, at the beginning of this PhD, there were not solutions (or not mature enough) specially designed for IoT in order to automatize IoT security management in dynamic and reactive ways, relying on security policies, SDN and NFV technologies. In that sense, it is required that new solutions consider the specific security necessities in the IoT scope, to be now an essential part of the security policies of the systems. Besides, they also have to provide new security plugins and components required to apply those new policies through new orchestration and self-healing capabilities which keep permanently a high level of security in an autonomous way. According to this, new solutions would be able to deal with challenges like massive IoT security configurations, IoT dynamic evolution, heterogeneity, as well as the IoT constraint nature. However, we found a lack in the state of art regarding solutions able to cover all these requirements, which will be the main focus of the different goals of this work.

The present PhD thesis presents the results of the research, design and implementation of a reactive policy-based security framework for IoT networks in new SDN/NFV-enabled infrastructures. The framework allows security administrators to model and define proactively new security policies at different levels of abstraction, depending on the required complexity and the level of knowledge, as well as to reuse existing ones in a common repository. High-level security policies can be refined into medium-level security policies, which in turn are translated into final specific security enablers configurations. For instance, forwarding policies can be translated into IPTABLES or SDN rules depending on the current deployment. Policy management and orchestration is also endowed with policy conflict and dependencies detection to guarantee the safe instantiation of the security policies over the infrastructure. Taking the advantage of IoT security policies management, as well as the ability to enforce them along different security enablers, depending on the current deployment, the framework contributes to the mitigation of security challenges like heterogeneity, massive deployments or the vendor-locking. Besides, dynamic instantiation of new virtual network functions through SDN and NFV allows deploying security enablers on demand in specific locations, providing new advanced security functions in diverse IoT environments. To provide self-healing and self-repairing capabilities, the framework is also able to enforce automatically reactive security policies as part of a mitigation plan (orchestration policies). This new reactive capability allows the framework to maintain a constant security level in the system, according to the detected threats or the recommended security measures over time. In order to validate the feasibility of the proposal, new security policy models, framework components, workflows, new security enablers specially designed for IoT, policy enforcements over IoT networks, SDN and NFV technologies were validated in several JCR publications. Besides, those results were also validated during ANASTACIA H-2020 EU Project[4]. The result of this thesis provides what we consider a new valuable reference for IoT security.

This PhD was supported by the Spanish Government through the FPI Program (ref. PRE2018-083731).

[4] http://www.anastacia-h2020.eu/

2.2. Goals and Methodology

In order to contribute in the research of mitigating the different challenges that have been exposed in the previous section, the objective of this thesis is mainly focused on the research of a self-healing, self-repairing and orchestration policy-based framework, which pursues the automation of security managing of complex IoT infrastructures, avoiding vendor locking issues as well as dealing with heterogeneity, constraint environments and massive deployments which characterizes IoT nature. This framework provides a full security loop composed by the proactive definition and enforcement of orchestration security policies, monitoring of the whole infrastructure, reaction according to the monitoring feedback, as well as definition and enforcement of reactive security policies. On one hand, proactive security policies allow security administrators defining, translating and enforcing high-level/medium-level of abstraction security policies, in order to apply security configurations in the system, without specific knowledge of the underlying enforcement points technologies or security enablers. On the other hand, reactive security policies provide the framework capabilities like self-healing or self-repairing in terms of it is able to apply automatically different sets of countermeasures, depending on the identified threats by the monitoring and reaction tools. For instance, a reactive security policy could isolate a part of the IoT infrastructure, or even it could deploy a complete IoT virtualized environment as an IoT honeynet, in order to retrieve more information about a potential misbehavior or an ongoing attack. To achieve the main goal of the thesis, we identified the following objectives:

- Objective 1: Definition and extension of new security **policy models** for IoT environments such as IoT management, IoT honeynets, authentication, authorization, privacy, channel protection, monitoring, or QoS.

- Objective 2: Design of Policy-based **framework architecture** able to enforce proactive and reactive security policies, according to the current status of the system, as well as managing the full security loop lyfe-clicle.

- Objective 3: Design, implementation and validation of the **policy refinement and translation** processes for IoT environments, extending current SoA with the aim to define, model and transform those security policies between different abstraction levels (high/medium/low) according to the requirements.

- Objective 4: Design, implementation and validation of the proactive/reactive policy **orchestration and enforcement** processes, considering different priorities as well as conflicts and dependencies.

- Objective 5: Implementation, deployment and validation of new **security functions specially designed for IoT**, based on SDN/NFV technologies, addressed by security policies which allow a) filtering and flows management, b) authentication and authorisation, c) channel protection, d) IoT management, e) virtual IoT honeynet deployments, for mitigating cyberattacks.

- Objective 6: **Validation and evaluation of the proposed framework** over different use cases and real scenarios, where multiple security policies are enforced in proactive and reactive ways across the new security enablers, specially designed for IoT, based on SDN/NFV technologies.

In order to achieve these objectives, the effort was divided into different blocks which correspond to each objective, to satisfy the final goal. In this term, we applied an incremental iterative methodology for each block. This is, each block pass multiple times through phases of, requirements analysis, state of art research, solution design, implementation, configuration and deployment, evaluation and analysis of results. These last phases generated new knowledge to refine subsequent iterations on the same block as well as its possible interactions with the other ones. In this way, each block is refined throughout the project and in turn shaping the overall solution. Specifically, we analyzed the state of the art for security policy models, policy-based frameworks as well as SDN/NFV and IoT integration solutions. Then, we selected the security policy models we consider more suitable for our interest. These were, High-level Security Policy Language (HSPL) and Medium-level Security Policy Language (MSPL) [16]. We extended and updated the models in order to provide and validate new security capabilities for IoT environments like IoT management (e.g. power management), IoT honeynet virtualization, traffic filtering, forwarding, Authentication, Authorisation (bootstrapping process), Channel protection (DTLS), as well as the combination of multiple security policies (orchestration policies), also taking into account priorities, conflicts and dependencies between them. Each one of the new or extended policy models was validated over new security enablers, specially designed for IoT, along different versions of the framework whose implementation was evolving during the thesis, according to the results on each iteration. In fact, the results of this thesis were also applied and validated during ANASTACIA H-2020 EU project[5].

2.3. Results

During the thesis period, the iterative methodology over the objectives produced multiple results such as **a book chapter, a conference article, and nine publications JCR indexed, of which five compose the compendium of this thesis**. Since the results were also validated during ANASTACIA H-2020 European Project, several technical reports (e.g., more than 20 European project deliverables) were also produced. Table 2.1 shows the main relevant results achieved during the thesis, as well as the relationship between the results, objectives and publications.

After the state of art analysis, we selected the security policy models to be extended according to the requirements of the established goals. Then, we designed a very first modular architecture and we implemented a proof of concept for integrating the selected policy models, as well as policy transformation tasks, for SDN, NFV and IoT domains by following a security enabler plugin-based and driver based approach.

During the design phase, the main goal was to define a framework generic enough to manage diverse kind of virtual security functions, specially thought for IoT, which are policy-based orchestrated in an interoperable, dynamic and efficient way, also considering possible conflicts or dependencies during the deployment. Through this approach, we provided solutions to different IoT security management problems, by instantiating, evolving and validating our framework with new implementations designed or adapted for IoT. Specifically, first experiments were performed for use cases where security administrator applies security policies for IoT traffic management through the reconfiguration of the SDN network. Besides, it was also designed and implemented a virtual network security function (vNSF) which includes the logic of a firewall, with the aim to establish a comparative. The results of those experiments were provided in our first publication [122]. Based on the previous results, the architecture design was extended for managing IoT bootstrapping processes by using dynamic AAA services and channel protection, through new policy models and new security enablers like virtual PANA agents, XACML-based PDPs and DTLS proxies, deployed as VNFs. The results provided dynamic IoT registration, proactive policy-based authentication configurations, and reactive policy-based authorisation for the authenticated IoT devices. They were detailed in our second publication [124].

Considering the advances provided by our proposal, the components were integrated with monitoring and reaction components in the scope of ANASTACIA H-2020 EU project, for validating the whole self-healing loop. In this case, we define new monitoring and IoT security policies at different levels. Advanced monitoring tools were proactively configured by the new monitoring security policies, which detect miss-behaviours in the IoT domain. The system then generates different reactive security policies, according to the detected threats, such as filtering and IoT management, which are enforced through SDN and our IoT Controller. These results were published in our third JCR paper [125]. Once the self-healing loop was validated, we defined advanced and novelty dynamic countermeasures like dynamic virtual IoT honeynets, which allow replicating physical IoT environments on demand. To this aim, we provided a new security policy model able to represent IoT networks from the available

[5] http://www.anastacia-h2020.eu/

Table 2.1: Main thesis results

Result	Objectives	Publications
R1. Design of a security policy-based framework, able to define, model and manage security policies at different levels of abstraction, extending the SoA.	1, 2	[122] [124] [125]
R2.Design high-level policy refinement process, which considers current infrastructure to refine high-level security policies (HSPL) into medium-level security policies (MSPL).	1,2,3	[122] [124]
R3.Design proactive/reactive medium-level security policies translation process, which considers current infrastructure to transform medium-level security policies (MSPL) into specific configurations that can be enforced in security enablers.	1,2,3	[122] [124] [125] [127] [128] [129]
R4. Design the enforcement process in order to apply the specific configurations over the required security enablers by using a driver-based approach.	1, 2, 3, 5	[122] [124] [125] [127] [128] [129] [131]
R5. Design, implementation and validation of orchestration policies as well as the enforcement of multiple security policies, considering conflict and dependencies detection.	1, 3, 4	[130]
R6.Implementation and validation of proactive policy refinement, proactive/reactive policy translation and policy enforcement process.	3, 5, 6	[122] [124] [125] [127] [128] [129] [131]
R7.Implementation and validation of security enablers to perform IoT network traffic management through dynamic network (re)configuration by using different SDN controllers and virtual network functions.	5a, 6	[122]
R8.Implementation and validation of dynamic authentication, authorization and channel protection security policies in IoT environments through dynamic deployments of AAA services as well as dynamic re(configuration) of the SDN network.	5b, 5c, 6	[124]
R9.Implementation and validation of security policies for monitoring, traffic forwarding, filtering and IoT management through dynamic re(configuration) of the SDN network as well as an IoT controller specially designed to this aim.	5d, 6	[125]
R10.Implementation and validation of IoT honeynet policies and transparent traffic forwarding through dynamic re(configuration) of SDN network and dynamic deployment of VNFs, specially designed and implemented to this aim, able to replicate whole IoT environments.	5e, 6	[129]

information of current IoT deployment, gathered during the bootstrapping process as part of our previous results. Besides, we designed and implemented the IoT honeynet manager as new security enabler. By combining reactive networking and IoT honeynet policies, this new security countermeasure can be deployed and managed on demand, transparently, to redirect an attacker to a fake replica of the IoT environment. The results of this research are the main focus of our fourth publication [129]. Finally, considering the requirements of enforcing multiple proactive and reactive security policies, as well as to establish different enforcement priorities, we extended the policy models for providing orchestration policies. In this way, the orchestration policy can represent an enforcement plan by indicating the enforcement order, priorities and dependencies between policies or events. For instance, in the reactive IoT honeynet, the traffic must be redirected only when the IoT honeynet has been properly instantiated. In this regard, we also provided a policy conflict and dependencies detector for ensuring that orchestration policies can be enforced in a safe way [130].

It is important to highlight that final Proof of Concept (PoC) results are still being evolved, and some parts of the design are in fact considered in new European Projects (e.g., INSPIRE 5G+). Of course, the implementation is open source. It is stored in the repository of the research group[6]. Next, the reader can find an extended abstract for each publication which composes the thesis compendium. Besides, full information regarding each publication can be found in chapter 4.

2.3.1. Enhancing IoT security through network softwarization and virtual security appliances

This first publication [122] goes one step further in security policy-based network management, extending the policy enforcement along IoT environments through different SDN controllers, as well as virtual firewalls, also establishing a comparison between them. First, it provides an analysis of integration and application of SDN and NFV features in order to improve security in IoT networks. Specifically, it explains interesting SDN features applicable to security scope such as dynamic flow control or manipulation, which allows to reconfigure the network behaviour according to the security specifications (e.g., network isolation), as well as to manipulate specific fields of the packets. SDN centralized management and SDN devices monitoring are also highlighted in order to analyze and verify the current status of the network in terms of security (e.g., traffic peaks detection in IoT networks). In the same terms of SDN, the publication also provides NFV features applicable to security such as decoupling security functions from hardware, which avoid vendor-locking and facilitates on-demand scalability and mobility. Those properties contribute to deploy, migrate, scale up or down Virtual Security Functions (VNFs) when required for providing new dynamic security services to constraint IoT devices. Once the article has exposed the benefits of applying SDN and NFV features in order to improve the security in IoT networks, it provides a high-level first design of the framework architecture (R1), composed by three main planes; the user plane where the security administrator introduces high-level security policies into the system, security orchestration plane, focused on the framework orchestration, and the security enforcement plane, where the security policies are enforced over different security enablers. First design and workflows of policy refinement, translation and enforcement processes were also provided (R2, R3, R4), as well as relevant applicable use cases like building management system or edge computing countermeasures.

Finally, the publication provides a first Proof of Concept (PoC) implementation of the first design of the framework, able to receive high-level filtering policies, which contain human readable values (e.g., Sensor 1), and to refine them in medium-level security policies where high-level values have been refined into machine readable concepts (e.g., IPs and ports). Once filtering policies have been refined, the publication also shows a comparison of filtering policies translation and enforcement for different security enablers across the infrastructure (R6, R7), these were, ONOS, Opendaylight (OpenFlow) and a virtual firewall (NETCONF).

2.3.2. Enabling Virtual AAA Management in SDN-Based IoT Networks

This second publication [124] is focused on providing a dynamic policy-based management of the framework for channel protection, Authentication, and Authorisation (parts of AAA), which goes beyond the state of the art to be interoperable with IoT environments during the bootstrapping process, also allowing dynamic deployments in the limits of the cloud (edge), as near as possible of the IoT devices. Specifically, it provides different workflows and operations in order to allow IoT devices to access specific resources (even the network itself) in a secure way. To this aim, since it is assumed a deny by default policy, first, the process in which the security administrator authorise proactively the authentication traffic in the network for the desired IoT devices is defined (R1, R2). This process also allows indicating possible future interactions that IoT devices could have with the infrastructure (resource authorisation). For instance, by modelling that concrete IoT devices will be able to put

[6] https://ants-gitlab.inf.um.es/anastacia-framework

temperature in the IoT broker. This kind of traffic authorisation, unlike traditional approaches, is then instantiated dynamically in the SDN by modifying the flow rules in order to allow the kind of authentication protocol used by the IoT devices. Examples of resource authorisation and traffic divert policies were also provided. Once the authentication protocol for the specified IoT devices is allowed, they are able to perform the authentication process which ends with the generation of a Master Session Key (MSK), as well as the registration of the new IoT devices into the system (IoT Controller).

When an authenticated IoT device tries to access some resource, it first retrieves a capability token which indicates what it can do, as the security administrator specified at the beginning through the proactive policies. A successful acquisition of this token generates a new reactive authorisation policy enforcement (R3) for automatically allowing the traffic in the SDN for the granted specific purposes of the IoT device. In addition, the publication also provides the process for refining, translating and enforcing channel protection policies (R3, R4, R6), specifically, DTLS channel protection policies, between IoT devices and the desired endpoints (e.g., IoT broker or DTLS proxy). In this case, when a new channel protection enforcement is requested, after policy refinements and translations, a new enforcement master key (e.g., PaC-EP Master Key or PEMK in case of PANA protocol) is generated and included in the final security enabler or endpoint configurations, which are enforced through a secure channel. The endpoint or security enabler (usually, the IoT broker or the DTLS proxy) then use the PEMK as shared key in order to prepare the DTLS channel with the IoT device, that also receives a channel protection enforcement request though the IoT Controller. Finally, the publication shows results and measurements for the whole process (R8). It includes network and resource authorization, channel protection, and the required IoT security interactions by combining the power of the SDN networks, dynamic virtual deployments through NFV, and specific IoT security aspects through the IoT controller.

2.3.3. Security Management Architecture for NFV/SDN-aware IoT Systems

Third publication [69] advances current state of the art regarding monitoring, detection and reaction in traditional environments by including specific IoT monitoring components (e.g., 6LowPAN), and reactive IoT security policies (e.g., IoT control) over the SDN/NFV infrastructure. Besides, it provides the implementation of new functionalities of the IoT controller to enforce new IoT security policies. Thus, this work presents an advanced version of the framework architecture (integrated inside ANASTACIA framework for validation), including components and interfaces which compose each plane. User plane design, now contemplates not only the Policy Editor Tool component for high-level policies modeling and instantiation, but also alerting and notification dashboards among others, in order to provide security administrators real-time information about security and privacy. This feature allows them to interact with the system if required, depending on the nature of the issue. Security orchestrator plane consolidates the proposed elements by this thesis, composed by the policy interpreter, in charge of refining and translating security policies ((R3, R4)) to generate final security enablers configurations (e.g., SDN rules), the security enablers provider, in charge of providing security enablers plugins depending on the required capability of the policy (e.g., filtering), and the security orchestrator, in charge of orchestrating and enforcing proactive/reactive security policies along the whole infrastructure.

To generate reactive security policies, modules from Monitoring and reaction Plane are used. For validating the monitoring and reaction stage, different tools and components were provided by other authors of the paper, and integrated in the architecture. The monitoring module retrieves information from different monitoring agents (e.g., Snort or MMT agent) that can be located in the enforcement plane. The module notifies any kind of issue to a verdict and decision support system in the reaction module, which analyzes and correlates the information in order to decide if any kind of countermeasure is required [56]. If so, a mitigation action service generates new reactive security policies to be enforced. All these processes as well as an example of reactive filtering security policy are detailed in the new workflows provided for the monitoring and reaction parts. After showing the details about the architecture design, the publication provides a set of IoT threats or attacks, and how the framework architecture design could mitigate them. For instance, a set of IoT devices infected with a malware could be isolated from the rest of the infrastructure by using SDN management, while at the same time, modifications in the firmware can be performed through the IoT controller (filtering and IoT mgmt security policies). In order to verify the feasibility of the new design, and following the iterative and incremental methodology, a new version of the architecture was implemented and deployed, where we simulated two different scenarios (R6, R9). A mobile Edge computing scenario and a building management system scenario. In the first one, several 6LowPAN IoT devices send messages continuously to a specific target, and a Deep Packet Inspection (DPI) tool deployed in the 6LowPAN network detects and notifies the issue to the monitoring module, that triggers the reaction process. This process generates a new reactive filtering security policy, which is enforced through the SDN. In the second one, IoT devices have been manipulated for notifying unrealistic temperature values with the aim of rising the fire alarm. In this case, more sophisticated monitoring tools were used [121] in order to discern between regular and abnormal situations according on a AI learning process. Then, the monitoring module notifies a misbehaviour, which generates a new reactive security policy to reboot or turn off the compromised IoT devices until future checks. After security administrator verifies there is no real risk, the alarm is also turned off. Finally, the publication provides a performance evaluation for a burst of incidents at different frequencies, which triggers different reactive security policy enforcements through the SDN Controller and the IoT Controller as selected security enablers.

2.3.4. Virtual IoT HoneyNets to Mitigate Cyberattacks in SDN/NFV-Enabled IoT Networks

Unlike the analyzed works in the state of the art, which are not security policy-based, and only deploy honeynets in a static and proactive way to simulate services or traditional networks, this fourth publication [129] provides results about the deployment of dynamic VNFs, able to replicate real IoT infrastructures from a well-defined and updated model of the system. This deployment is managed by IoT honeynet policies as well as networking policies which enforce a transparent traffic forwarding through the SDN. The work therefore is focused on providing High-interaction IoT honeynet capabilities to the framework. A high-interaction honeynet is composed by a set of high-interaction honeypots which simulate as much as possible the real deployment (e.g., a full device virtualization, including endpoints and resources). On the opposite side, low-interaction honeynet is composed by low-interaction honeypots which only simulate certain part of the real environment (e.g., only ICMP echo response). In order to provide the solution, the publication shows a SOTA comparison between different proposals. It takes into account important features like, if the solution is policy-based, SDN/NFV-enabled or if it provides dynamic capabilities (e.g., dynamic honeynet deployments). After the SOTA, it proposes to evolve the previous framework in order to deal with new IoT honeynet security policies, which were also modeled.

Specifically, we extended the Technology Independent Honeynet Description Language (TIHDL) [26] with new types such as IoT honeynet, IoT router, and IoT honeypot. IoT honeynet model can be composed by a set of IoT routers, IoT gateways, and IoT honeypots. The IoT honeypot model is able to represent information like the level of interaction, interfaces, firmware, software, hardware model, location and resources. IoT router model specifies similar parameters to IoT honeypot model, but also including routing information. This extension is then homogenized in the Medium-level Security Policy Language (MSPL) the framework uses, as part of a new capability. Once the policy model have been extended, the publication provides a detailed explanation of the policy-based management, this time also providing an algorithm in order to deal with policy dependencies during the orchestration process, and how this process is properly integrated in the framework. Now, when the monitoring module detects some kind of issue, the reaction module can choose a dynamic IoT honeynet deployment as part of a mitigation plan, which generates an IoT honeynet MSPL by retrieving the IoT domain information from the system model ([124] shows how IoT devices are registered dynamically). This new reactive IoT honeynet MSPL will be translated (R3, R4, R6) by using different security enabler plugins for IoT emulators, depending on the configuration of the real environment (e.g., Cooja for contiki IoT devices or Mininet 6LowPAN with uPython for uPython-based IoT devices). Since filtering/forwarding rules to drop/forward the traffic from/to the IoT honeynet must be enforced after the IoT honeynet has been deployed, the orchestrator queues the dependant policies, and it only enforces them once the dependencies has been solved (e.g., the IoT honeynet has been properly launched and configured). In order to verify the feasibility of the design, the publication provides implementation details and experiments for the new research. It includes new MSPL models, translations, orchestration and performance evaluation for different IoT honeynet scenarios. Specifically, we replicated two real IoT environments composed by different topologies, amount and type of IoT devices (R10). We provided measurement values for the whole dynamic IoT honeynet instantiation process, also taking into account routing convergence times when required. Finally, we compared the results between the scenarios as well as between them and classic mesh deployments.

2.3.5. Semantic-Aware Security Orchestration in SDN/NFV-Enabled IoT Systems

The last publication [130] is focused on the new orchestration policy models, conflict and dependencies detection, and orchestration optimization (R5). Unlike current solutions for IoT, which few are policy-based, and even less consider orchestration processes and conflict detection during the enforcement, this work provides not only new security policies with orchestration capabilities, but also policy conflict and dependencies detection as well as optimization tasks during the policies enforcement process. The work shows an state of the art regarding SDN/NFV security orchestration in IoT, Service Function Chaining (SFC) optimization, Semantic-based network management and security management, which are the main subjects of the paper. After this analysis, we highlighted the new components in the orchestration plane over the integration with the ANASTACIA architecture, used to validate the proposal. At this point, co-authors of the paper defined new components inside the orchestrator to optimize the orchestration process, which now considers inputs from the conflict and dependencies detector as well as system model and monitoring data. From our side, we defined the orchestration security policies, as well as the whole workflows for managing conflicts and dependencies between policies and events during the enforcement process. We provided the motivation for orchestration policies supported by AAA use cases where multiple authentication, an authorisation policies must be enforced by following an specific order, also considering that some policies must be only enforced according to specific events generated by the system. For instance, resource access must be only authorised when the specific IoT device has been properly authenticated. Besides, the orchestration process must ensure that new enforcements will no generate new issues in the system.

In this regard, we defined and provided different examples of conflict and dependencies detection rules for covering different well-known conflicts. We also provided context-based conflict detection rules such as capability missing conflict or insufficient resources conflict. Those kind of conflicts not only consider conflicts between security policies, but also conflicts between security policies and the infrastructure. Once we defined the rule set, we presented the integration with the rule engine, which loads and keep up to date the data about the infrastructure and security policies as facts. When a new security policy is verified, it is matched against the rules which relies on the current facts for providing a verdict. Considering these new orchestration properties and functionalities, single security policies became orchestration policies able to represent order, priorities and dependencies between policies and events. In this way, we provided new complete workflows for integrating orchestration policies in the framework for both, proactive and reactive scenarios. The new approach was validated and evaluated through the enforcement of multiple orchestration policies, different number of rules, facts and already enforced security policies (R5).

2.4. Conclusions and Future Work

Due to the problems that characterize IoT environments such as heterogeneity, massive deployments, vendor-locking as well as constraint resources, new security challenges that are threatening the security management as well as the security itself are emerging. In order to mitigate these issues among others, we presented a novel policy-based IoT security framework which was validated along multiple JCR publications, as well as during ANASTACIA H-2020 European project. The framework provides a high-level of abstraction layer by using security policies which are independent to the underlying infrastructure, decoupling the security requirements of the specific implementations to deal with problems like heterogeneity and vendor-locking. The modularity of the design, as well as the proper integration with SDN, NFV and Monitoring technologies allows to evolve the framework behaviour with new capabilities in pro of security management, self-healing and self-repairing features as it have been exposed along this thesis.

In that sense, they were provided results for the design, implementation and validation for compromised IoT devices isolation through high-level filtering security policies, which were refined and translated by the framework to obtain the required security configurations. Those configurations were enforced during the orchestration and enforcement processes across different SDN controllers such as ONOS and Opendaylight, as well as virtual firewalls instantiated as VNFs, showing the benefits of the SDN approach as security mechanism integrated in IoT environments. This result is and important advance in the state of art, which did not consider policy-based SDN/NFV approach for managing security in IoT.

Dynamic AAA capabilities were also provided for IoT environments, nonexistent until that moment, through the refinement, translation and enforcement of different authentication, authorization and channel protection security policies. The solution also included the distribution of cryptography material to the required components of the architecture. Dynamic authentication and authorization capabilities also eased the register of IoT devices in the system model, which maintains the information about the current status of the architecture. Flows for integrating the new components were designed and instantiated, considering authentication transport protocols like PANA, authorization characteristics like capability tokens and channel protection technologies like DTLS. In this case, new plugins and drivers were also designed and developed to ease the translation of medium-level security policies into final configurations, as well as the enforcement from the security orchestrator across the new security enablers such as DTLS proxy, PDP XACML and IoT controller, which were also validated. Thus, results for proactive/reactive IoT policy-based processes, from IoT devices authentication, until IoT devices perform their first operations through a secure channel were provided. Those results showed the feasibility and performance of the solution, based on a novelty approach that exploits SDN/NFV for and efficient authentication, authorization and channel protection on IoT scenarios.

Components and interactions of the framework were also validated during ANASTACIA H-2020 EU project, where our results were integrated with monitoring and reaction elements. Thus, different IoT threats were analyzed, and different countermeasures were proposed to mitigate those threats by using our solution. To this aim, new security policies were defined, as well as the interactions between the new modules to consider specific threats for IoT domain. Thanks to the new monitoring activities, IoT threats can be detected, triggering then the reaction process which automatically generates new reactive security policies to instantiate new reactive countermeasures. In this case, countermeasures were enforced through the SDN network (filtering/forwarding), as well as the IoT controller (IoT management). Different scenarios were implemented and validated, providing the required workflows to instantiate proactive IoT monitoring security policies and reactive IoT countermeasures.

Regarding the NFV-based automatization and virtualization of IoT devices, we provided the first results about dynamic and transparent instantiation of virtual IoT networks, which replicate real IoT environments as a new security countermeasure through the integration of SDN/NFV and IoT specific emulation tools. To this aim, policy models were extended to represent IoT networks from an existent language (TIHDL), able to model specific honeynet concepts. From this new model, a new detailed design and implementation of the dynamic deployment of IoT honeynets, taking the advantage of the IoT information gathered at the bootstrapping stage, was provided. In this way, different real IoT scenarios for multiple IoT deployments were replicated, also considering dependencies between security policies to ensure they are enforced properly (e.g., IoT honeynet has been deployed and configured before the transparent forwarding has been instantiated). This research also concluded with the implementation and validation of new plugins and processes to translate the new IoT honeynet policies into final security enablers configurations, which were instantiated in new IoT security enablers such as the Cooja IoT emulator. An IoT honeynet manager was also developed as well as new functionalities for the IoT controller. During the validation experiments, we considered all the reactive deployment process for different IoT infrastructures and topologies. The results showed the feasibility and performance of the new solution, nonexistent until that moment, by combining SDN/NFV and different IoT virtual environments which allow deploy on demand IoT honeynets in reactive way. The provided solution can be applied with different security objectives like to redirect an attacker to the virtual IoT replica while the ongoing attack procedure is analyzed in a safe way.

Finally, as last contribution of this thesis, we designed orchestration policies to significantly improve the mitigation capabilities of the framework. This new policy model contains multiple security policies, and allows to specify enforcement properties such as order, priorities or event dependencies between policies or between policies and events. Thus, orchestration policies allow provide complex mitigation plans. To ensure the policies are enforced properly, we also provided a policy conflicts and dependencies detector, which verifies that new security policies will not present any kind of conflicts according to a specified rule set. This research was also validated for different facts, rules and security policies.

It is important to highlight that the results of this PhD thesis, as well as the implementation of the different components have been, and are being exploited and reused in H2020 EU projects such as ANASTACIA and INSPIRE 5G+.

Despite we consider the results of this thesis provide a valuable reference in the IoT security scope, there are still different research lines, which have been exposed during the evolution of this thesis. One of the main focus for future work and research, resides in the security orchestrator plane. In fact, we are currently evolving the orchestration in terms of, how to select the best security enabler, also taking into account all the information gathered in the system model. To this aim, the security orchestrator must be able to retrieve continuously, updated information regarding all the infrastructure such as current instances, services, controllers, networks, policies and security properties. This kind of information is then provided to a well-defined orchestration algorithm. Policy conflicts and dependencies must be also considered during the orchestration. Of course not only inter-policies or intra-policies conflicts and dependencies, but even conflicts and dependencies between security policies and the current status of the infrastructure in terms of availability, security and QoS must be considered.

Regarding the allocation, there exist several proposals of orchestration allocation algorithms, based on different optimization approaches like greedy, scored, fuzzy rules, ILP or MILP among others. However, the results trend to consider only available resources, instead of including security conditions of the environments. Apart from that, it could be useful to consider not only one orchestration algorithm but multiple algorithms in order to select the most appropriate one, according to the current status of the environment (meta-orchestration algorithm).

Another interesting research line is about not only programming the control plane along the SDN network, but also the data plane. For instance, by developing new P4 based security enablers, they could generate P4 code depending on the existing P4 available devices in the system (physical or virtual). In this way we could manage the data plane of the network considering fields and options beyond layer four in the TCP OSI stack. Besides, it would also allow to enforce QoS security policies considering features like in-band network telemetry, in order to combine control plane and data plane information for re-configuring load balancing capabilities.

Chapter 3

Introduction

Despite Internet of Things (IoT) term was coined by Kevin Ashton in 1999, when he tried to promote Radio Frequency Identification (RFID) technologies, several sources point that, as far as anyone knows, a Coca-cola vending machine[1] in the early 1980s was the first IoT device. David Nichols' office, in Carnegie Mellon University computer science department, was far away from the vending machine so many times the walk from the office to the machine was unsuccessful due the machine was empty or drinks were warm (recently reloaded). In order to avoid that, David and other colleges developed an IoT system. Since the vending machine had a light indicator for each drink slot, they placed an IoT device which measured the light indicator status. This IoT device was connected by a wire to a department computer, which in turn was connected to the Advanced Research Projects Agency Network (ARPANET), making the role of an IoT gateway. In this way, people could verify the status of the vending machine, as well as if the new drinks were cold or not (the system inferred it by considering the time from the last replacement). Even today, we also consider this example provides a clear vision of one of the IoT definitions (the definition has been evolved along the years), like *"Sensors and actuators embedded in physical objects that are linked through wired and wireless networks"*.

Far away from this first deployment, it was in 2008-2010 when IoT approach started to accelerate, providing a huge visibility in market around 2014. Now (2020) sources like *IoT Analytics* estimates end-users will expend globally up to 1567 billion dollars in IoT solutions. In this regard, it is normal to understand that companies and research labs are making huge investments in this area, which at the end materializes directly in IoT techniques and technologies evolution. What started measuring the light indicator of a vending machine, evolved in infinite IoT industrial solutions, and now is also totally part of our lives. In fact, statisticians[2] estimates now between 3.5 and 5 billion of active smartphone, which most part of the time carry out between 5 and 10 sensors since majority of devices provide at least an accelerometer, gyroscope, magnetometer, GPS, proximity sensor, ambient light sensor, microphone, touch screen and camera/s. Apart from smartphones, people also adopt the wearable approach (more than 400 million of wearable shipments on 2020[3]), where smart watches, wristbands and ear-wear among others are part of our clothing and routines, adding to the aforementioned sensors new ones like pedometer, barometer, heart-rate sensor, thermometer or humidity sensors. Beyond this overwhelming per-person IoT device approach, IoT devices manufacture price is getting cheaper, which also promotes huge deployments on business and industrial environments like smart buildings,

[1] https://www.ibm.com/blogs/industries/little-known-story-first-iot-device/

[2] https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/

[3] https://www.statista.com/statistics/690731/wearables-worldwide-shipments-by-product-category/

smart cities, big factories and smart agriculture. Actually, today it is quite common that a single room or corridors in a building contain at least CO2, humidity, temperature, luminosity and presence sensors, also considering in most cases not only the sensing part, but also the actuation part like HVAC consoles or card readers for access control at the doors. If it is true that these sensing and actuating capabilities provide huge advantages in terms of multiple topics like automation, safety or surveillance, IoT also entails big challenges. For instance, constraint resources make harder the implementation of common security measures, as well as they are often the focus of attackers exploit tools. Fast scalability and heterogeneity make difficult the security administration for every single firmware, from different versions and manufacturers, as well as different protocols which also impact on life-cycle management and secure commissioning. Of course, privacy is also a huge challenge, considering the huge amount of sensible data gathered by different sensors in our environment. All these challenges are more detailed on section 3.1

Apart from the huge and vertiginous IoT massive deployment adoption, we can also find vast deployments of virtualized devices, networks and whole environments. Virtualization concepts appeared in early 1970s when IBM provided solutions for time-sharing expensive computing resources, allowing users an affordable access to computation capabilities. This main concept of sharing resources in different ways evolved continuously until the most common scenario we can find today, huge farms of high-performance servers which contain incredible amounts of virtualized services, devices, networks as well as whole infrastructures. It is important to highlight that, apart from the virtual machine based virtualization, container approaches, whose first main concepts appeared in 1979 with techniques like root directory changing, have been also gaining importance due to its light nature among others. If well 2005 was a key year for virtualization due to free desktop virtualization solutions, the emergence of cloud-based solutions like Amazon Elastic Compute product in 2006, as well as OpenStack[4] open-source cloud-software in 2010, speed up the adoption of performing computing tasks and deploy virtualized environments and resources in servers that are accessed through the Internet (cloud). As technology matures, virtualization and automation processes have been improved considerably. Not a long time ago, when a customer desired to deploy a new virtual resource in a cloud, he/she had to emit a request to the cloud provider and wait for a response. When the cloud provider received the request, an human operator in charge of virtual deployments managed to launch and configure the new instances and connections manually. Then the customer was notified. Today, the most common scenario is that the cloud provider offers web forms to configure the virtualised environment specifications, and now, when the customer sends the request, after pertinent verification, virtualisation and configuration processes are performed automatically. In fact, the improvement and automation in virtualization processes make feasible having more flexible infrastructures by applying virtualisation no only for devices and machines but also for network functions.

Network Function Virtualisation (NFV) provides innovative characteristics like applying new network features as well as modify the network behaviour when required, by instantiating new Virtual Network Functions (VNFs) at strategic points of the network. For instance, through this approach it is possible to virtualise different implementations of routers, switches, firewalls or load-balancing among others, avoiding vendor-locking issues. In order to manage the VNFs deployment, NFV architecture, proposed by the European Telecommunications Standards Institute (ETSI)[5] contemplates the NFV Orchestrator entity (e.g., OSM[6]), which is aware not only of the VNFs life-cycle but also manages them through a Virtual Infrastructure Manager (VIM). Considering these kind of architectures and virtualization improvements, that significantly enhance deployment timing, NFV approach becomes a strong ally for providing on-demand resources. In fact, since compute nodes are highly distributed, EDGE/FOG computing that pursues to extend the cloud for providing resources as near as possible of users, plays key roles in terms of enhancing Quality of Service (QoS) properties, considering the huge amount of expected devices and simultaneous connections. Thus, the ability of deploying groups of VNFs in specific locations, as well as re-configuring the network in order to provide connectivity is now

[4] https://www.openstack.org/
[5] https://www.etsi.org/technologies/nfv
[6] https://osm.etsi.org/

a very important feature to consider. However, despite current virtualization techniques allow fast and dynamic virtual deployments, it is also necessary to interconnect those new virtualized environments with the rest of the infrastructure. For instance, it is required to provide a new flexible and dynamic path between new service instances, and the customers which are paying for them. In this regard Software Defined Networking (SDN) fits perfectly in this new gap.

SDN changes the way networks are managed. This approach considers three different planes such as application plane, control plane, and data plane. Application plane contains the SDN applications that implements the logic in order to program the network. Control plane is now centralized (it is not anymore in network devices such as routers) and it is in charge of managing the network in order to configure it, to obtain the behaviour expected according to the application plane. Data plane (still in network devices) is in charge to enforce the configurations received from the control plane. Even though it seems somewhat quite novel, main ideas started emerging twenty years ago, in mid-1990s like active networking, focused on providing a programming network interface as well as to ease programming new custom functionality. This main idea evolved along the years and today we can find advanced versions of SDN standard protocols like OpenFlow for managing data plane, as well as different SDN Controller implementations like ONOS, ODL or Ryu. By using a right combination of these SDN tools and NFV technologies, it is possible to deploy dynamically VNFs and connect them through the SDN network. This is, now it is possible to deploy and interconnect dynamically different kind of services at different points of the SDN network (e.g., EDGE/FOG/CLOUD). In fact, since it can be performed in a reasonable time (specially for containers), it offers support for new security capabilities like the dynamic deployment of monitoring tools, in specific segments of the network, or the reactive deployment of security countermeasures according to new issues detected in the infrastructure.

These new features suppose an important improvement for providing automation of security along the life-cycle of the infrastructure, specially in massive deployments like IoT. The dynamism and flexibility provided by NFV on-demand deployments, and SDN network reconfiguration, plays a key role for strengthen IoT security according to the IoT infrastructure evolution. For instance, this approach allows to deploy or reconfigure new IoT specific security measures to mitigate new attacks as middle-boxes, as near as possible of the IoT devices, which often are too constraint for implementing the most common security features, also managing the new required network paths through the SDN network dynamically. However, now security administrator must deal with VNFs configurations, SDN configurations, traditional network devices configuration and IoT devices configurations among others, each time he/she wants to apply new security measures for multiple layers or technologies (e.g., channel protection). In this regard, high-level of abstraction approaches like security policies also becomes fundamental. These kind of solutions allow security administrators defining security requirements at high-level terms, in an independent way of the underlying technologies.

As the other approaches, policies have also been evolved from their early adoption of access control features, to the definition of specific languages for system and security modeling, trying to cover multiple areas like infrastructure management and security. In this term, policy languages like High-level Security Policy Language (HSPL) and Medium-level Security Policy Language (MSPL) allow to model different security requirements at different levels of abstraction, based on capabilities like filtering or channel protection. Thus, security administrators don't have to deal with the configurations of multiple implementations, taking also the advantage of the formal policy modeling which eases the identification of conflicts or possible misbehaviour's in the system. However, in order to enforce policy models in the SDN/NFV/IoT infrastructures, it is required to redesign and implement different security mechanisms such as policies refinement, translation and orchestration processes. It is also required to provide new security components and virtual security functions, specially designed for IoT which properly exploit the combination of those technologies for improving security. For instance, by applying auto-generated configurations in new VNF security enablers such as virtual AAA agents for IoT, or virtual IoT honeynets.

According to the previous considerations, this PhD thesis is focused on providing specific security features suitable for IoT environments in new generation infrastructures. To this aim we have researched, designed, implemented and validated a policy-based security framework, able to enforce proactive and reactive security policies in IoT environments over SDN/NFV-enabled infrastructures.

Current chapter is organized as follows. Section 3.1 shows a summarized analysis of security challenges we must consider in order to design the policy-based approach for NFV/SDN-enabled IoT infrastructures. In section 3.2 we analyzed the state of the art of the related techniques and technologies as well as security solutions for IoT, also considering the challenges analyzed in the previous section. Section 3.3 provides the framework proposal details, including framework architecture, workflow of processes, implementation and validation aspects for different use cases. Finally, Section 3.4 shows the conclusions and lessons learned during this research period.

3.1. IoT Security Challenges

IoT security challenges have evolved according to the adoption of the IoT paradigm in the society. Recent efforts like Garcia-Morchon et al. [109] as part of the Internet Research Task Force (IRTF), provide a state of the art and challenges in the scope of IoT security. They categorized the main challenges in terms of constraints and heterogeneous communications, bootstrapping of a security domain, operational challenges (e.g., mobility), software update, end-of-life, verifying device behaviour, testing, quantum resistance, privacy, or trustworthy IoT operations. Hassija et al. [113] and Ziegler et al. [58] also recently reviewed IoT security challenges as well as potential threat sources for IoT applications at different layers. For instance, at sensing layer, they analyzed node capturing, code injection or booting attacks among others. At network layer, they identified issues like access attack or Dos/DDos attacks. In middleware layer, they highlighted attacks such as man-in-the-middle or SQL injection. Authors also remarked suitable features like machine learning or fog and edge computing in order to enhancing IoT security. Security features that our framework is able to provide.

Also considering previous challenges, but more focused on automatizing IoT security, a recent paper analyzes the actual cybersecurity challenges [57], highlighting the necessity of providing autonomic security orchestration and enforcement capabilities in softwarized and virtualized IoT/CPS systems and mobile environments. In this regard, present section summarizes the main challenges for managing security in IoT domains, considering current state of the art.

3.1.1. Constraint resources

Probably, one of the most representative characteristic of IoT devices is their constraint nature. Of course, depending on the tier of the IoT device, the meaning of the word *constraint* can vary significantly. For instance, Raspberry Pi, endowed with multi-core CPU and four GB of RAM a constraint device (despite it is almost as powerful as the laptop I am using for writing this document), if it is compared with a real compute node, or in terms of energy when it is deployed to be powered by batteries. Beyond this example, in general IoT devices are designed to be constraint in terms of size and power consumption, which in general entails also constraints at computation, storage and communication levels. For instance, Sky mote models used along this thesis, among others, are empowered by 8 Mhz, 10 KB of RAM and 48 KB of flash memory. Besides, we have to consider that IoT communication protocols have been also designed pursuing low consumption's and low-data rates (e.g., 50 bytes per frame). If it is true this constraint nature represents a challenge at implementation and management levels, it also become a risk in terms of security. For instance, denial-of-service attacks based on resources depletion becomes easier in constraint devices. In fact, a simply port mapping operation using a well-known tool like nmap[7] over a 6LoWPAN IoT network can break down the IoT infrastructure if it is not properly protected. Moreover, the small amount of CPU and memory resources make difficult the adoption of standard security features like public key cryptography, to provide for instance, authentication or channel protection security capabilities. These circumstances require new adapted security solutions for providing AAA in IoT, as proposed in this thesis, that

[7] https://nmap.org/

offloads the computation from IoT to the edge, also using pre-shared keys for cryptography operations to achieve efficient behaviour in constrained environments.

3.1.2. Scalability and Heterogeneity

Other IoT security challenges which also suppose a significant impact in security are scalability and heterogeneity. As we mentioned in section 2.1, due increasing IoT features and decreasing manufacturing prices, around 127 new IoT devices are connected to the network every second. In this regard, we have to consider that IoT deployments can be composed by huge amounts of IoT devices, provided from different manufacturers. Thus, when we need to configure them, we need to deal with specific firmwares as well as different end-points and communication protocols for each different model and manufacturer. Moreover, despite it sounds like a problem that only may occurs in fabrics or enterprises, we are not far away from facing the same kind of issues at home. Nowadays it is easy to find multiple IoT devices from different manufacturers in our homes like HVAC regulators, blind windows actuators, smart meters or virtual assistants, and it is also quite common to discover specific bugs or security issues which requires specific firmware updates. Without going further, it is frequent to discover that your home access point was configured with weak security mechanisms by default, like WPS, weak passwords, passwords generated by well-known processes, according to the model of the router, or even they have available parallel wireless zones of which you are not aware of. In fact, American Customer Institute determined that almost 83% of routers in United States are vulnerable to well-known cyber- attacks[8]. In that sense, our policy-based approach allows mitigating by-default misbehaviours by establishing well-defined proactive security policies, as well as to deal with the heterogeneity. Besides, security policies instantiation in IoT environments through SDN/NFV infrastructures also allows managing scalability in dynamic ways.

3.1.3. Commissioning

Related with the previous challenges, commissioning also requires to be properly tackled from the point of view of security. Despite recent research efforts like Manufacturer Usage Description (MUD) [77], which provides access control specifications from the beginning, allowing the system be aware regarding what should be done by the IoT device, or like available bootstrapping methods [117], currently, each IoT device or IoT device groups perform the commissioning in different ways depending on the organization, manufacturer and the environment. Beyond the concern of homogenize the commissioning behaviour, we have to take into account that sometimes important security issues are generated specifically during commissioning processes. For instance, as it was the case of popular vacuum models which send to cloud servers owners WiFi password at startup time, as part of a memory dump process. In this regard, an standard, secure and well-defined bootstrapping process will help considerably to improve security during IoT commissioning stage. In fact, the strong definition of proactive and reactive security policies can protect the infrastructure from unexpected behaviours during those processes.

3.1.4. Life-cycle management and behaviour

Another challenge which is also specially injured by scalability and heterogeneity is the life-cycle management. Once IoT devices have been properly deployed, they must be monitorized and managed during the whole life-cycle, considering that most of them must be working continuously during years. This becomes specially important in terms of security due IoT devices are prone to specific kind of attacks, and a single compromised device could generate a massive infection, able to knocking down complex infrastructures. However, a proper management during the whole life-cycle can be a nightmare if current deployment is quite heterogeneous, or if part of the deployment or depends on firmware updates from manufacturer which sometimes never come. In this regard, to keep updated a system

[8] https://www.theamericanconsumer.org/wp-content/uploads/2018/09/FINAL-Wi-Fi-Router-Vulnerabilities.pdf

model of the infrastructure can help to be aware of important deployment information like firmware version, current capabilities, well-known bugs or security issues per each model. Automatic analysis of this information can provide alerts and notifications, or even an overall IoT software security status can be inferred based on the current security status of the deployment. In fact, we consider this kind of information crucial to discern the end of life-cycle of IoT devices, not only when they are faulty or new features are required, but also when they are not able to be prepared against new security issues anymore (e.g., abandonware). An entity like our IoT Controller in combination with the system model is specifically designed to register and manage different kind of IoT devices and IoT protocols, also providing a common interface for easing the IoT life-cycle management.

3.1.5. Privacy

IoT and privacy seems to have an inverse relationship from the beginning. The more IoT devices deployed in the environment the worse privacy for the user. But in fact, this concern is today a reality. Large part of the people never abandon their smartphone, and the time we use our smartphones for accessing internet is increasing day by day[9]. This is, we are always accompanied by between 5 and 10 sensors gathering and sending information continuously/sporadically to the cloud in order to improve our user experience, even when we are not using our device. Besides, when we use our device for accessing internet, traceability is fully exploited in order to infer new information which could be useful in the future (e.g., advertisements). If it is true it sounds really helpful, and actually it is, gathered information sometimes includes sensible information which can be linked and exploited by hackers in different ways, such as identity fraud, blackmailing or even physical injuries due they could know your day by day activities. In fact, sometimes we are not sure if the amount and the nature of the gathered information is really needed by the service provider to improve the services. Indeed, users have manifested concerns several times in this regard. For instance, we can find concerns about if it is really required that a vacuum uploads up to 10 GB per month to the service provider[10] in order to enhance the cleaning results. Further, apart of our own devices, we have also to consider IoT devices deployed in the environment which we are not able to manage to, such as surveillance cameras, presence control sensors or even recreational drones, which in general board cameras whose recordings are later uploaded to different streaming platforms. According to these considerations, we think it is really important to provide privacy-by-default approaches from the beginning on each IoT deployment susceptible to gather any kind of sensible information from users. This privacy-by-default behaviour can be contemplated by modelling proactive security policies, which can be enforced along the infrastructure in different ways such us allowing only the traffic according to the defined policies, as well as providing suitable privacy encryption approaches (e.g., CP-ABE).

3.2. Related Work

This section provides the related work analyzed during this thesis for the different topics involved in the research project. These are, IoT security and privacy, SDN and NFV as security enablers, as well as research efforts regarding security policies.

3.2.1. IoT Security and privacy

To be aware of IoT security challenges evolution, the research community studies and analyzes continuously IoT security and privacy threats. As a result, we can find multiple analysis focused on IoT security and possible countermeasures. Years ago, Gao et al. [2] presented different IoT security threats for physical, network, and application layers. For each layer, they proposed different countermeasures like enhance the management, point-to-point encryption or end-to-end encryption to

[9] https://www.statista.com/statistics/319732/daily-time-spent-online-device/

[10] https://www.reddit.com/r/Xiaomi/comments/9tgyrg/any$_{r}eason_{w}hy_{m}y_{x}iaomi_{r}obot_{v}acuum_{u}ploads/$

improve authentication mechanisms and to establish a centralized security management for IoT. Later, Sicari et al. [6] presented research challenges and existing solutions for IoT security, also identifying open issues, and providing some future research lines. They categorized the solutions in terms of authentication, confidentiality, access control, privacy, trust, policy enforcement, security middlewares, and mobile security in IoT. Finally they conclude that it is necessary to design suitable solutions in order to deploy all the aforementioned security aspects, independently of the underlying technologies. In the same regard, Yu et al. [15] exposed three key requirements for the future to make feasible the massive IoT adoption. These were, providing technology abstraction through security policies, to learn from attacks and from regular profiles, and to provide context-aware enforcement capabilities. Sadeghi et al. [100] also analyzed security and privacy challenges in industrial IoT environments, summarizing existing frameworks in terms of security architectures, integrity verification or secure IoT management. Considering these works, it is important to highlight that most of the security mechanisms and countermeasures proposed have been managed and implemented in our solution.

More focused on networking, Granjal et al. [5] performed a survey of security in IoT communication layers. Specifically, they categorized security for PHY, MAC and network layers, routing, and finally also for application. For instance, they mentioned security in CoAP with an example for IEEE 802.15.4, 6LoWPAN, CoAP and DTLS, as we provided by using different security enablers in our framework.

Alaba et al. [44] provided an analysis of threats and vulnerabilities in IoT security as well as possible attacks. They compare traditional security issues with IoT security issues, highlighting problems like the exhaustion of resources. With this analysis they aim to provide a guide of existing security threats in IoT heterogeneous environments. A different approach is provided by Chen et al. [60] which considered three main aspects for IoT security challenges. Unreliable communication, hostile environment and improperly data and privilege protection. In our solution, the IoT Controller and security enablers designed for IoT data encryption and IoT data privacy face the aforementioned challenges.

In another survey, Vorakulpipat et al. [61] analyzed IoT security according to IoT generations. Specifically they identified three main generations. First generation was characterized to be used for testing environments where the most common security aspects related to traditional computer security were applied if any (e.g., confidentiality or availability). Second generation is used in real environments with more standardized protocols and centralized IoT platforms, so security aspects like authentication, device identification or authorisation becomes essential. Third generation is related to industry where authors expressed main concerns in privacy issues due to big data analysis. Of course, our platform also consider different security requirements depending on the IoT features and generations.

More recently, Ziegler et al. [58] highlighted privacy threats on IoT like the identification of the data subject, data sniffing, profiling, or spoofing, data linkage or localization. They also provided an analysis of security threats at physical, network and application layers. These previous works on security and privacy analysis were really helpful in order to model security policies like authentication, authorization, channel protection and data privacy. Finally, a new IoT security survey was provided by Ahmad et al. [59]. Specifically, they surveyed IoT security challenges and proposed countermeasures. In this case, authors categorize layers as perception, network and application, and the main identified security challenges are confidentiality, heterogeneity, integrity, lightweight solutions, authentication and availability. Those security challenges have been considered during this PhD for providing innovative security enablers for IoT in our solution.

Motivated by security and privacy issues identified during studies like the previous ones, the research community is contributing with multiple solutions to enhance different security and privacy aspects in IoT environments. In this regard, apart from the really important contributions like the earliest microthreaded operating system for sensor devices such as TinyOS [114], Dunkels et al. [106] contributed considerably to the IoT adoption and homogenization, by providing a lightweight and flexible operating system for Tiny Networked Sensors (Contiki). One of the most important feature is the ability of downloading code at run-time, in order to solve bugs or security issues. In fact, later they also provided a Cross-level sensor network simulation tool (Cooja) [25] for simulating Contiki sensor network environments. Indeed, this interesting tool is part of our solution. A Cooja agent was developed as security enabler for mapping physical environments into Cooja simulations in order to virtualize dynamically real IoT environments.

To ease the access of IoT devices to the infrastructures, Kanda et al. [105] discussed about the applicability of Protocol for Carrying Authentication for Network (PANA) for constrained environments, also recommending extensions in order to make it more suitable for these kind of domains. They also provided implementation guidelines for PANA and EAP for devices with 250KB ROM and 50KB RAM. Finally they conclude that more work was required in order to fit the solution in even more constrained devices. In this regard, Garcia-Carrillo et al. [104] provided a survey of different IoT bootstrapping techniques. They also proposed a lightweight bootstrapping service for IoT networks [4] which uses CoAP, EAP and AAA infrastructures. They performed a comparison between their solution and the most relevant related solution such as PANATIKI (PANA for Contiki IoT Operating system), showing significant improvements due to reductions in messages length. Authors also provided in [36] a low-overhead version of COAP-EAP (LO-CoAP-EAP) for Low-Rate Wireless Area networks (LP-WAN). Since our solution is policy-based and NFV/SDN-enabled, it allows specifying authentication requirements at high-level, which are deployed dynamically in different authentication agents depending on the type of authentication used by the IoT devices. In this way our framework supports different protocols for carrying authentication during the bootstrapping, being also extensible for future contributions.

In another remarkable contribution which can be also applied during bootstrapping process, Hernandez-Ramos et al. [23] designed a distributed capability-based access control (DCapBAC) for IoT. Specifically, they designed a capability token solution for CoAP resources which ensures end-to-end authentication, integrity and non-repudiation through Elliptic Curve Digital Signature Algorithm (ECDSA). From their results for authorization, they also used in [22] the bootstrapping process as a key factor in the life-cycle of smart objects. They use and extend the Protocol for Carrying Authentication for Network Access (PANA) as IoT bootstrapping protocol. In this case they assume IoT devices have x509 certificates with specific attributes, which allow an attribute-based encryption. Later, authorization method was evolved by the authors in [110] by providing Elliptic Curve Cryptography optimizations to their capability-based access control mechanisms. The results of these works were also applied on others access control systems like the proposed by Bernabe et al. [17], which provided a trust-based security mechanism which relies on the previous one. Since we also consider DCapBAC a suitable approach for IoT, it has been also applied in some scenarios of our solution, by including a capability manager which interacts with the PDP as part of resource authorisation process, according to the authorisation policies.

Also in terms of access control, Kolluru et al. [38] provided an access control solution. Specifically, they provided a Next Generation Access Control (NGAC)-based solution for service-level fine-grained access control in IoT device environments. Their solution expresses policies in terms of attributes which contemplate users, objects and operations. They also provided an use case for a heating system. However they only used certificates for DTLS connections in order to authenticate the IoT device whereas our framework allows modular and extensible authentication and authorization technologies. In the same topic, but more related on privacy, Perez et al. [108] introduced an attribute-based lightweight symmetric cryptography solution for smart building scenarios. Specifically, the solution is focused on Ciphertext-Policy Attribute-Based Encryption, in order to allow specific subjects access to specific pieces of data for privacy-preserving. This approach has been also included in our framework by configuring CP-ABE privacy on demand in IoT devices, and by deploying dynamically CP-ABE proxies as near as possible of the IoT domain, for providing data privacy to those IoT devices which are too constraint to perform this kind of operations.

For application layer protection, Selander et al. [35] proposed the standard Object Security for Contrained RESTful Environments (OSCORE) which aims to provide end-to-end application-layer protection for CoAP by using object signing and encryption techniques. We consider this proposal interesting to be integrated in future versions of our solution by enhancing our IoT RESTful components with OSCORE capabilities.

Due to the high impact of Mirai botnet attack, multiple guidelines to avoid botnets as well as different monitoring tools for IoT also emerged. Bertino, Islam [40] and Kolias et al. [101] exposed the necessity of providing scalable security solutions for tackling distributed denial-of-service attacks in IoT. They provided information regarding recent botnet (robot network of compromised machines) attacks such as Mirai botnet, as well as possible protection techniques against them such as monitoring specific ports, updating IoT devices, or making sure there are not default passwords in the infrastructure. In those aspects, our framework provides a scalable security solution able to enforce different security policies dynamically in order to mitigate attacks like this one. Linking with a policy-based approach, Polk et al. [102] from NIST provided a full guide to adopting IETF Manufacturer Usage Description (MUD), to avoid or mitigate network-based attacks on small business and home IoT environments. Through this standard each device type has associated a well-known behaviour from the beginning in terms of what is it allowed to do or not. In fact, authors in [103] also highlights the potential DDoS mitigation capabilities since the IoT devices actuation's are well defined from the beginning. We also consider this as a promising approach, indeed, our framework is able to manage and enforce MUD policies.

However, to be able to detect different issues in the infrastructure, IoT monitoring tools are also required. In this regard, Mady et al. [39] provided a hierarchical anomaly-based intrusion detection method able to detect anomalous activity. It considers two different types of IDS. Local IDS and supervisory IDS. Local IDs contributes by gathering information to the statistical model of the regular functioning. In this way, the model learns about regular activities and then, the supervisory IDS is able to correlate the data in order to detect anomalous behaviours. Related to this approach, bagaa et al [112] also provided a framework which combines monitoring agents and AI-based reactions.These promising monitoring and detection technologies were incorporated in our framework to detect and trigger mitigations for different IoT threats or attacks.

Another interesting feature, also related to monitoring capabilities are honeynets and honeypots. Oza et al. [67] performed a survey of different applications of honeypot and honeynet techniques for providing countermeasures for some kind of IoT attacks. They highlighted DoS mitigation by redirecting malicious traffic against a honeynet, Xen-based honeypot virtualization, or the use of honeynets in order to detect unknown vulnerabilities. At difference of the surveyed solutions, in our framework IoT honeynet policy models have been integrated in the security policies environment and honeynets. Our solution also allows deploying IoT honeynets in proactive and reactive ways, even replicating physical IoT environments by using different IoT virtualization agents for different IoT firmwares (e.g. contiki or uPython). However, in order to compose dynamically these advanced solutions, more powerful techniques and technologies like SDN and NFV are required.

3.2.2. SDN/NFV as Security enablers

The ability of SDN networks to reconfigure on-demand the data plane allows modifying the behaviour of vast amount of devices simultaneously from a centralized command and control point. This flexible and homogenized approach makes SDN technology suitable for tackling security issues dynamically as it has been demonstrated in different research efforts. In fact, we can find efforts employing SDN principles years ago where Suh et al. [8] extended an access router by using NetFPGA-Openflow platform to provide accountability functions in a content oriented network, for discover and mitigate DDoS attacks. For identifying main SDN security capabilities, Sandra et al. [27] provided a survey for security solutions based on SDN, as well as a discussion about new security challenges introduced by the adoption of this kind of technology. They highlighted the combination of a global overview of the network, and the ability of network programming for providing IDS and IPS capabilities. Specifically, they categorized the security solutions as SDN middle-box or Security Defined Networking (our solution implements both). First one implies there exist middle-boxes able to provide a security functionality and then, the SDN manages to re-configure flows for passing through it. The second one provides techniques like IP obfuscation or performing IDS/IPS operations by using available metrics in the SDN controller. Ali et al. [9] also presented a survey of innovative security features that can be provided by SDN networks. They categorized the security capabilities as security configuration, threat detection, threat remediation and network verification. In addition, they analyzed advanced security features like anonymization. In this regard, our solution also provides the aforementioned capabilities, including network traffic values manipulation at high level by defining specific networking security policies.

Another SDN security enabling survey is provided by Xu et al [34], which in this case highlights dynamic flow control, network-wide visibility with centralized control, network programming, simplified data plane, as well as enhancing information security processes focused on prevention, detection and response. Other surveys like [33], are focused on dealing with specific kind of attacks. In this case, it is focused on how SDN could mitigate DDoS attacks in cloud computing environments. In another approach, Rawat et al. [43] is focused on security attacks as well as SDN countermeasures to mitigate them. They also provided different approaches in order to achieve energy efficiency and security by applying SDN. For instance, by modifying dynamically the maximum speed of SDN device ports depending on the use of the link. In this regard, our framework provides reactive SDN capabilities based on monitoring values, but extending the possible countermeasures to a wide range of security policies, which can be enforced in different kind of SDN controllers or traditional networking security enablers, depending on the status of the infrastructure and the current deployment.

Considering the aforementioned features, it is comprehensible that the research community introduced SDN as part of their security frameworks, even for IoT management. Luo et al. [11] proposed to apply SDN concepts directly to Wireless Sensor Networks in a solution called SD-WSN. They used Sensor OpenFlow (SOF) as control plane protocol in order to allow the controller managing flow entries on each sensor. This protocol is an OpenFlow customization which reuses available fields in order to transport specific sensor control plane information. They also adapted the amount of control traffic in order to avoid overload the WSN. Also for managing WSN, Gallucio et al. [10] provided SDN-WISE but, unlike the previous case, they implemented their own control plane protocol, as well as controller management applications. Oliveira et al. [13] also applied this approach and they proposed TinyOS-based SDN framework which allows the WSN be managed by multiple controllers. The framework is composed by SDN-enabled sensors, end-SDN devices and controllers, where end-SDN devices are considered out of the scope of Openflow, and the flow management inside sensors is performed by TinySdnP program, which is in charge to process packet-in and packet-out messages. Despite these are really interesting approaches we decided to avoid overcharging IoT devices with SDN specific functionalities as well as providing an approach as much generic and non-disrupting as possible.

More focused on security aspects, Shin et al. [12] provided a security framework focused on design and compose, in a modular way, OpenFlow-enabled detection and mitigation software modules. Modules are focused on events, and a desired behaviour can be composed by multiple modules. For instance, one module can implement the desired monitoring matching conditions, and the next module the actions (e.g., drop). Also for validating security, Yoon et al. [50] provided different implementations in order to verify the feasibility of enabling security functions with SDN. Specifically they developed different applications in the top of Floodlight controller for different purposes like firewalls, IDS, anomaly detection like DDoS detector, and what they called advanced security functions like stateful firewalls. Flauzac et al. [45] also proposed an SDN based architecture, this time for IoT, which distributes security rules along a distributed multi-domain SDN infrastructure. Also in this regard, Choi et al. [49] defined strategies in order to endow security frameworks of software-defined approach for an efficient provision of security services for IoT. Coinciding with some aspects of previous works, our solution also provides a modular approach, but in our case, the plugin-based, driver-based and enabler-based approaches ease to managing different networking security policies through multiple SDN controllers, instead of implementing an app on the top of them. However, previous defined strategies were considered for endowing our framework with SDN features.

More focused on enhancing monitoring, Bull et al. [48] proposed a distributed SDN gateway able to detect abnormal behaviours in the traffic coming from/to the IoT devices. They remarked the utility of this technique, but also considering the impact in the installation of new rules per second due to the monitoring task process. Also in this topic, Xu et al. [31] provided a smart security mechanism (SSM) against new-flow attack, which pursues break down the SDN connectivity by exhausting the SDN resources. Specifically they reused already existing messages in the control link in order to monitorize the hit rate of the flow entries. More recently, Galeano et al. [111] provided a different monitoring approach where monitoring logic is performed by the SDN switches in order to correlate current packets with those previously received. They focused this technique in DDoS detection enforcing dynamically a mitigation, like flow modification operations, if an attack is detected. As in previous cases, our work also considers SDN metering as relevant monitoring information, but this kind of information is provided to a specific monitoring module outside the scope of the SDN Controller, which will be in charge of managing monitoring related computation tasks.

These SDN monitoring techniques can be also complemented with another techniques in order to enhance security such as honeynets and honeypots. In this regard, Fan et al. [70] provided an SDN-based transparent mechanism for TCP handover in honeypot environments, also including traffic filtering according to the alerts generated by an Snort deployment. In this way, depending on the monitoring tool, the traffic is forwarded to a valid destination or redirected against a honeypot. More recently, Lin [66] also provided a solution in order to redirect the traffic from suspicious nodes to honeypots, adding spoofing techniques in order to deceive possible attackers. They also provided an evaluation of the impact of the solution considering the spoofing of the network. In this topic, our solution also applies SDN techniques for honeynets but also providing solutions for reactive IoT honeynets, managing transparently the SDN network forwarding through IoT security policies, considering IPv6 and IoT protocols such as CoAP.

For enhancing channel protection and key distribution, Marin-Lopez et al. [30] provided an IETF initiative in terms of managing IPSec Security Associations (SAs) in SDN networks, to provide end-to-end channel protection dynamically. For instance, making the SDN controller in charge of distributing the keys through the southbound interface, among the involved end points. In this regard, our solution provides a policy-based bootstrapping process which relies on the SDN network for network authorisation, but in our case, key distribution is managed by the security orchestrator.

To provide more flexibility to the SDN configuration and management, Comer et al. [75] purposed an intent-based SDN Open Software Defined Framework (OSDF), to avoid dealing with specific configurations. It allows administrators to provide high-level network requirements like monitoring or QoS, and those high-level terms are translated into final configurations after resolving intent conflicts. Hamza et al. [72] also relies on high-level policies for managing the SDN network but in this case by using IEEE Manufacture Usage Description (MUD) policies. Then, they translate those policies in final SDN rules which are enforced in proactive and reactive ways along the SDN switches. Following the same approach, Ranganathan et al [74] presented an implementation of MUD standard for OpenFlow-enabled devices, where authors relied on DHCP in order to detect IoT interactions, as well as for installing MUD profiles in devices. For instance, to open specific ports in a specific devices. Hassan et al. [73] also provided a SDN-based MUD-compliant infrastructure, which takes advantage of security SDN features for detecting abnormal behaviours in the MUD-compliant network. Unlike previous cases, our solution allows defining security policies at multiple levels of abstraction (including MUD), which can be translated in proactive/reactive ways into multiple SDN controllers or security enablers (e.g., firewalls). Besides, our approach is not only focused on filtering or forwarding. It is able to enforce more capabilities like mirroring, datagrams manipulation or network slicing.

Beyond all the security advantages provided by SDN, Network Function Virtualization (NFV) approach also provides a huge added value in terms of security, and these security improvements increase when both technologies are combined. For instance, by deploying dynamically multiple VNFs and redirecting the traffic properly in order to create security chains. From the NFV side, European Telecommunications Standards Institute (ETSI) defined a NFV framework architecture [14] as well as its main philosophy, and required features of the supporting infrastructure. This design provided a starting point for research community which also integrated SDN in the solution. In this way, Li et al. [28] provided a survey for deployments which combine NFV and SDN approaches. They showed an state of the art and principles of NFV infrastructures as well as the relationship between them and the SDN architecture. They also analyzed different middle-box and service chaining solutions like the ones we apply in our framework. Finally, they provided e list of challenges and possible solutions. Yang et al. [42] also provided a survey on security in NFV. They highlighted the challenges and opportunities that NFV provided in terms of security. They also analyzed different existing NFV-based security solutions, highlighting the introduction of policy managers, or trusted virtual domains as key points of security. From other point of view, Lal et al. [7] analyzed risks and vulnerabilities that can be introduced by using NFV approaches. However, they also provided available mitigations and best practices in order to ensure the NFV application in a secure manner. Farris et al. [18] also provided a survey on emerging SDN and NFV security mechanisms for IoT systems. They analyzed the security features that can be provided by this approach in order to perform monitoring and reaction tasks against IoT threats. They also compared IoT security threats with conventional ones, and finally they analyzed new challenges related to the application of these technologies. Those challenges have been considered during this PhD for providing our solution.

Due optimized resource allocation in NFV-based infrastructures is still a current challenge, Gil et al. [21] provided a survey of resource allocation techniques and processes like allocation graphs, chain composition, forwarding graphs, scheduling, as well as optimization strategies. In this regard, but mostly related on deployment locations, Vaquero et al. [20] analyzed challenges associated to complex NFV orchestration processes, considering multiple deployment domains like edge, fog or cloud. They also provided different orchestration techniques in order to deal with real world scenarios, like orchestration machine learning techniques, probabilistic orchestration or hierarchical delegation among others. Boudi et al. [76] also focused on provisioning virtualised appliances, this time on the EDGE, and specifically in constraint nodes. To this aim, they resorted to light-way virtualisation techniques and technologies like containers in raspberry pi nodes, located at the EDGE. Finally they compared the results to deploy monitoring tools in cloud and in the constrained edge, highlighting the benefits of the second approach. Since there are multiple solutions focused on resources allocation, our security orchestrator will contemplate multiple algorithms depending on the deployment requirements.

Considering the security properties of aforementioned technologies, new SDN/NFV security frameworks started to emerge. For instance, Basile et al. [3] presented a first approach for integrating policy management and NFV features in order to allow specifying high-level security requirements, which are refined into technical configurations when needed. They also provided the main components of the architecture like the policy manager as well as main processes like high-level policies refinement. However, unlike our solution, they focus only in a theoretical VNF configuration according to the policies, besides, the solution did not rely on SDN and IoT was not contemplated. Santos et al. [46] also presented a novel framework (SELFNET) which introduced reactive capabilities like self-healing by integrating NFV and SDN paradigms. They aim to improve security and QoS as well as to reduce the operational expenditure with an autonomic management. We consider this a really interesting contribution and some concepts have been adopted in our solution. Nevertheless the framework is not policy-based and it does not provide specific IoT capabilities. Another SDN/NFV framework was provided by Al-Kaseem et al. [41]. They presented a proof of concept framework which integrates 6LoWPAN NFV and SDN technologies inside a 6LoWPAN gateway, which also plays NFV and SDN controller roles for prolong the lifetime of the network in terms of energy saving. However this solution was more focused on providing one specific use case implementation rather than an SDN/NFV and IoT framework. Besides the solution does not provided the advantages of a policy-based approach. Unlike previous case, Ziegler et al. [32] proposed a very beginning and theoretical global overview of a policy-based NFV/SDN-enabled IoT security framework for the Advanced Networked Agents for Security and Trust Assessment in CPS IoT Architectures (ANATASTACIA) EU project. Later, Farris et al. [29] extended those concepts, this time focused on the SDN/NFV capabilities as security enablers for IoT systems. However, they only provided different use cases from a theoretical point of view, where the framework could provide proactive or reactive security improvements.

Also in a policy-based approach, but only in access-control domain, Welch et al. [107] proposed a SDN-based framework able to enforce dynamically network IoT access control policies. They also provided a VNF which runs an IPv4 ARP server to mitigate ARP spoofing attacks, for instance, by providing ARP replies from a trusted entity or by deleting ARP broadcast messages. However, besides our framework is not only focused on access control, it can provide similar mitigations directly through the SDN controller, as well as instantiating dynamically a new VNF if required, also considering IPv6, which is more compliant with IoT approaches.

For multi-tenancy, Salva-Garcia et al. [19] provided a security framework based on partial results of our design in order to manage networking policies in multi-tenant 5G environments. Specifically, they focused on 5G traffic filtering process considering specific mobility headers like GPRS Transport Protocol (GTP). Finally they validated the solution by providing a testbed implementation. Also considering multi-tenancy, Do et al. [62] proposed an SDN/NFV architecture able to provide multi-tenant network slicing over a shared physical infrastructure. They also implemented monitoring tools in control plane (ONOS) and data plane (P4), as well as in-band network telemetry. Network slicing was provided by applying VXLAN protocol. In [65] they extended previous work by including different IoT gateways to the deployment such as 6LoWPAN and NB-IoT. They also provided experiments in order to provide QoS monitoring among others to the IoT network. In this regard, our framework provides network slicing security policies as well as monitoring policies which can be enforced in different security enablers. Also for IoT, Shin et al. [64] focused on SDN and NFV features for providing IoT recovery applications, in terms of redirecting traffic to a new device when the previous one fails, or performing end-to-end network slicing by managing VLAN tags. In the same topic, Caraguay et al. [63] presented an SDN/NFV-based architecture for IoT environments, for dynamically re-configure the network in order to modify QoS flows in real time by using an SDN application. Those solutions provided significant advances of which, part of those concepts were applied and evolved during this thesis. However, those solutions are not policy-based and only are able to re-configure network dynamically but not deploying and configuring VNFs in reactive way according to the status of the infrastructure.

For enhancing advanced security features like honeynets and honeypots, Fan et al. [71] proposed a versatile virtual honeynet management tool in order to deploy heterogeneous honeypots. Their approach also allows to configure dynamically the honeynets according to the network environments. Specifically, they use a high-level honeynet description model which can be translated into different honeynet implementations. The proposal was validated against Honeyd[11], LXC-based[12] containers and KVM-based[13] honeypots. Although this is not NFV as such, we consider these kind of dynamic virtualization uses part of NFV philosophy. Also in honeypots virtualization topic, Guerra [47] analyzed different IoT threats that can be mitigated by using honeypots and honeynets virtualization. He also implemented a new tool (HoneyIo4) which simulates four IoT devices such as a camera, a printer, video game console and a cash registering machine. In [68] Banerjee et al. also analyzed different honeynet and honeypot solutions, in this case, providing different monitoring techniques for IoT botnets. They virtualized three different honeypots with an IDS to attract IoT attacks and analyze them. Even though previous researches are really promising, we consider these kind of scenarios could be significantly enhanced by a policy-based SDN and NFV integration. In fact, our work takes the advantages of those technologies for allowing deploy dynamically policy-based IoT honeynets for different targets (e.g. Cooja or mininet-6LoWPAN), also considering current status of the IoT infrastructure for replicating real IoT environments. To this aim, specific security policies for IoT domain where defined and included as part of the available security policy models.

3.2.3. Security Policies

The use of policies allows providing different layers of abstraction between users and technologies, as well as to formalize user requirements in pro of important features like homogenization and consistency. Policies design and specification have evolved along the years in order to ease system deployments and configurations for different scopes. For instance, Strassner et al. [89] provided years ago a draft for the Policy Framework Definition Language (PFDL) in order to provide a common policy language that can be managed by the different components of a framework. In this way, vendors could implement and integrate frameworks and devices components by ensuring they are PFDL-complaint. Later, Common Information Model (CIM) [1], in the Distributed Management Task Force (DMTF), formalized an

[11] http://www.honeyd.org/

[12] https://linuxcontainers.org/

[13] https://www.linux-kvm.org/page/Main$_{page}$

extensible model to represent and managing whole systems, including networks and applications. In order to include security concepts, Damianou et al. defined Ponder [96], a language for specifying security and management policies for distributed systems, which considered pillars of security policies such as authorisation policies or event-triggered obligation policies like Event-Action-Condition (ECA)-based. They also defined policy groups and policy roles for related policies. In the same topic, the proposed standard [84], inside the Policy Framework Work Group, also extended CIM for modelling policies as such, by including new classes in order to represent policies control and information, as well as indicating their relationship. For instance, relationship between a policy rule, actions and conditions. Afterwards, [91] provided a CIM Simplified Policy Language (CIM-SPL) for modeling condition-then-action style policy rules to manage environments already defined in CIM.

Focused on authorization, OASIS provided eXtensible Access Control Markup Language (XACML) [83] which allows modeling authorization security policies considering main elements such as subjects, attributes, resources and actions, as well as authorization decisions. Privacy was also relevant in an early stage, when Hada et al. [94] defined the Platform for Enterprise Privacy Practices (E-P3P), which provided fine-grained privacy policy modeling. For instance, it allowed to formalize how data collections must be handled by a each department of an organization. In later works, [92] they also presented the Enterprise Privacy Authorization Language (EPAL) which allows formalizing privacy requirements into privacy policies, which can be used during the access control decision. In a more generic approach, Web Ontology Language (OWL) [99] was designed by World Wide Web Consortium (W3C) to represent and formalize knowledge about things and their relationship. The result was used by Uszok et al. [87] which presented KAoS, a policy and contract management for semantic web services. Specifically, the framework allowed to manage, analyze and enforce OWL policies. Authors also provided a GUI for OWL policy modeling as well as verification for semantic web services composition. In terms of virtualization, Bleikertz et al. [93] provided a Virtualization Assurance Language for Isolation and Deployment (VALID). A formal policy language focused on security requirements on virtualized infrastructures (e.g., zones isolation). The language also supports to model current status of the infrastructure in order to compare them with desired ones.

Based on existing approaches, Bertino et al. [90] extended the event-condition-action Policy Description Language (PDL) provided by Bell-Labs, in order to provide also Preferences (PPDL). In this way, the new language allows to include user-defined preferences in the workflow that manages how the policies must be processed. They also provided examples of the preferences application for different use cases (e.g., dealing with separation of duties). Shankar et al. [51] also extended Event-Condition-Action (ECA) approach, but this time for providing a Post-Condition (ECA-P). This new step allowed authors to verify if actions presented conflicts with post conditions, as well as to verify if actions have been performed properly. Based on CIM, Bernabé et al. [24] provided a security policy specification by extending CIM models. Specifically, they provided an CIM-based Security Policy Language (xCIM-SPL) and a System Description Language (xCIM-SDL), as well as the refinement process between them. They also provided tools in order to assist the policy definition. Authors applied the provided approach on a policy-based framework they developed during DESEREC project [37], endowed with policy modeling, detection and response features. Fang et al. also defined a CIM-based language called Technology Independent Honeynet Description Language (TIHDL) [26], which allows modeling honeypots and honeynets at a high-level of abstraction. They also validated the approach by translating the model into different honeypot tool configurations.

Focused on reactive policies, Cheng et al. [98] presented Stitch, a language which allows representing repair strategies in context of self-repairing or self-healing frameworks. Specifically, it allows modeling different repair decision trees, as well as to represent business objectives. At a high-level of abstraction, Kumar et al. specified a language for security goals (LOCKS) [97]. It allows modeling the most common security features and authors also showed how existing informal security goals can be properly modeled in their approach. They also validate the model against a generic attack model such as structural attack model (SAM). Focused on current vulnerabilities, Moshin et al. [95] proposed UML-SR, a new security specification language. This language focus on be able to represent security requirements in order to avoid important vulnerabilities in the system. Specifically, it extends UML with security requirements such as authentication, ports management, checking privileges or password aging verification.

Continuing the multi-level approach, Shaw et al. [55] extended previous multi-level security policies efforts by defining High-level Security Policy Language (HSPL), though for non-technical users, and Medium-level Security Policy Language (MSPL), more technical but still independent on the underlying implementations. In [54] authors provided detailed explanations regarding policy transformation processes, from HSPL to MSPL, and from MSPL to low-level configurations for different technologies. In fact, Valenza et al. provided new detailed information for those languages in [88]. We consider those results as an important advance in security policy modeling. Thus, this thesis extends them for providing new security capabilities and functionalities. Also related to a high-level specification, recently, Lear et al. [77] provided the Manufacturer Usage Description Specification (MUD) which indicates the requirements of each device in terms of access to resources. In fact, the initial efforts are focused on access control. Jethanandani et al. [86] also proposed a new standard in order to use YANG Data Model for Network Access Control Lists (ACLs). It is important to highlight that our work is also compliant with MUD models, allowing enforcing them along the infrastructure.

According to policy modeling evolution, research community were providing different policy-based solutions by adopting the already defined policy languages and concepts. Rensing et al. [52] surveyed policy-based AAA solutions and they also provided a generic solution for a policy-based approach by specifying a set of preconditions like service separation, policy paradigm or the interaction between modules. Hadjiantonis et al. [53] proposed a policy-based network management with context awareness for managing Mobile Ad-hoc Networks (MANETs), in order to handle specific configurations of this kind of networks. They also evaluated the proposed framework in terms of scalability and performance. Bernabe et al. [78] defined a new authorization model, which considers advanced access control paradigms such as Role-Based Access Control (RBAC), hierarchical (HRBAC) or conditional (cRBAC). They used Semantic Web technologies also taking into account trust-based multi-tenancy in their approach. Sicari eta al. [79] the present a framework able to provide a flexible IoT policy enforcements middleware in order to handle large amount of IoT data streams. Authors also provide an Attribute-based Access Control (ABAC) examples applied during the framework validation. Canavese et al. [82] provided a formal model for detecting interference's between different VNFs in the same network. Authors also provided in [81] a novel approach for validating network policies in order to verify and detect possible misbehaviours, due to misconfigurations or attacks.

A capability-based approach and the integration between the defined security policies for managing NFV environments were defined in [80]. In fact, part of our work extends those advances. Barrera et al. [85] also proposed a solution in order to automatize the security policies enforcement, avoiding to modify IoT devices or cloud infrastructures. Specifically, they proposed white lists that can be provided by manufacturer or even automatically according to the network behaviour. At difference of our solution which were validated across multiple security enablers, they validate the experiments by translating white lists only on IPTABLES. A similar approach was then formalized by Lear et al. in the Manufacturer Usage Description (MUD) [77] proposed standard in order to provide access control specifications to IoT devices from the beginning. However, most of the previous solutions do not provide specific security policy models, as well as IoT security policy models. Besides, validations are often only provided for limited scopes (e.g., filtering), and the deployments do not exploit the combination of SDN and NFV for mitigating IoT security threats, specially in terms of reactive countermeasures.

3.2.4. Gap Analysis

After analyzing IoT security challenges and reviewing current efforts performed by the research community, we identified there are still open gaps at different levels in order to enhance security in IoT environments. Several contributions are adopting policy-based approaches to ease the management, also providing formal verification capabilities in order to detect issues during the policy enforcement. In general, current solutions inherit from previous works (languages or concepts) but only few approaches are focused on security policies as such, and even less for IoT. Besides, provided solutions trend to be focused or to be implemented directly for one or two security domains (e.g., authentication and access control). In fact, sometimes the solutions are thought for specific technologies (e.g., IPTABLES). In this regard, after analyzing research efforts in security policy languages, we considered HSPL and MSPL languages suitable for our proposal, due their policy models are able to represent specific security aspects on several security domains (e.g., filtering, traffic inspection, authorization or channel protection). Besides models and policy transformation processes did not reinvent the wheel, this is, they inherit from previous efforts that were validated in different projects, also taking into account capability-based models, that are currently being considered and evolved by IETF working groups like Interface to Network Security Functions (I2NSF) [115] [116]. With this in mind, we extended HSPL and MSPL policy models by defining new models and capabilities as well as extending existing ones in order to enhance the security policy languages with new features like IoT command and control, IoT channel protection, IoT honeynets, authentication, traffic divert, privacy, Quality of Service, data aggregation or network slicing. Besides, despite some works consider conflicts detection during policies enforcement, dynamic deployments now require also considering specific system and security information as well as dynamic system events related to the new allocation points. Moreover, in most common scenarios it is not only required the application of one security policy but policy chains. In this regard we also extended models in order to represent HSPL Orchestration Policies (HSPL-OP) and MSPL Orchestration Policies (MSPL-OP) which now are able to include priorities and dependencies between policies and/or events (e.g., Authorization event). It is important to highlight that our solution allows also providing other kind of security policies as input. For instance, MUD standard access control policies can be also provided as high-level policies.

Regarding solutions at infrastructure level, on the one hand, research community provided multiple SDN-based solutions mostly focused on monitoring, filtering or traffic redirecting capabilities. More advanced works used SDN controller as part of key provisioning for channel protection, and also few works integrated access control policies with SDN environments. On the other hand, previous works which only relies on VNFs management and dynamic configuration lacks on the dynamic network configuration capabilities in order to provide solutions like middle-boxes. In that sense, security solutions such as security frameworks can be enhanced by applying both, SDN and NFV principles. In fact, several works provided SDN/NFV frameworks but in general, existing solutions are focused on specific implementations they are not policy-based or they do not contemplate multiple security domains. Moreover, in some of them, NFV potential is not properly harnessed since they do not consider valuable aspects like dynamic deployments and on-demand VNFs reconfiguration. Besides, only few solutions contemplate a whole security loop, but they do not take the advantage of policy management as well as they are not focused on IoT environments.

To fill in those gaps, we provide an orchestration policy-based SDN/NFV-enabled security framework for IoT, able to enforce proactive (policy editor tool) and reactive (automatically) security orchestration policies over different security enabler implementations through a plugin/driver-modular design, also considering conflicts and dependencies. At difference of previous works, this approach allows, for instance, to deploy on demand different authentication agents dynamically, according to the authentication protocols used by the IoT devices (e.g., PANA or CoAP-EAP). Moreover, since the framework is endowed with different monitoring and reaction techniques and tools (e.g., monitoring at 6LowPAN level), different regular and IoT specific threats can be detected, generating new security policies as automatic countermeasures, that are enforced dynamically through different security enablers like SDN Controllers (e.g., ONOS or ODL), NFV-MANOs (e.g, OSM or kubernetes) and IoT domain components, depending on the underlying infrastructure. In fact, the solution currently provides more than 10 different security enablers for mitigating multiple kind of cyber-attacks. Besides, in order to mitigate IoT specific management issues like heterogeneity and scalability, our work also includes the design and implementation of an extensible IoT Controller which provides IoT operations to upper layers through a Northbound API, as well as providing the communication with the IoT device by implementing different southbound IoT protocols such as MQTT or CoAP.

Following section provides a detailed explanation of the proposed framework whereas chapter 4 shows the main results obtained, including specific experiments and validations in terms of publications during this thesis period.

3.3. Policy-based security framework in new generation SDN / NFV-enabled IoT infrastructures

During this PhD thesis we designed, developed and validated a policy-based security framework for SDN/NFV-enabled IoT infrastructures. The results were validated in multiple publications, as well as during the Advanced Networked Agents for Security and Trust Assessment in CPS/IoT Architectures (ANASTACIA-H2020) EU project. The framework provides a whole IoT security loop which includes proactive security policies definition, translation, enforcement, infrastructure monitoring, and reaction according to the detected threats, relying on new security enablers specially designed for IoT, as well as a flexible and dynamic management of NFV, SDN and IoT domains to endow IoT environments with self-healing and self-repairing capabilities. In this section we detail the design, implementation and validation of the security framework, highlighting our main contributions. First, the framework architecture is exposed. Then, we provide policy models, policy transformation and conflict detection, as well as proactive/reactive policy enforcement across multiple novelty security enablers. An overview of the required workflows for each process is also provided. The section concludes with one of the complex use cases where the framework was validated, by mitigating different threats in a multi-attack scenario over an Smart Building. Results in terms of publications are provided as part of the compendium in chapter 4.

3.3.1. Framework architecture

Figure 3.1 shows the framework architecture design, which is composed by multiple planes with different responsibilities and functionalities. Those planes are, user plane, security orchestration plane, security enforcement plane, data plane and monitoring and reaction plane. An overview of the framework architecture and its evolution was provided in our publications for contextualizing the solution. During this PhD, multiple components of the proposed architecture were instantiated in the ANASTACIA framework architecture [125] for providing proactive and reactive policies enforcement and security management capabilities in IoT environments.

User plane

User plane provides different interfaces and tools for easing security management from the user point of view. Specifically, it provides dashboards for supervising the network status (Networking GUI) as well as managing alerts and reactions (SIEM GUI) and modelling security policies (Policy Editor Tool). Network dashboard allows administrators to verify current status of the SDN network devices as well as different metrics for each on of them (e.g., amount of traffic). Alerting and reaction dashboard allows to verify alerts, the automatic reactions chosen, as well as defining new reaction plans in order to provide new countermeasures according to available capabilities in the infrastructure. Finally, Policy Editor Tool allows security administrators modeling different high-level security orchestration policies according to the available policy models, including priorities and dependencies. Once security policies have been modelled, the tool also allows to refine them in medium-level security orchestration policies, as well as to request the policy enforcement, to analyze conflicts, and verify the current status of security policies in the system (e.g., pending/enforced/removed).

Security Orchestration Plane

Security Orchestration plane is in charge of managing orchestration policies, as well as deciding the appropriate policy enforcement point that provides the required security functions, from now on, security enabler. This plane is mainly composed by the Policy Manager, the Security Orchestrator, the Security Enablers Provider, as well as the required database models like the System Model to support policy management and orchestration operations. Policy Manager contains on the one hand the Policy Interpreter module, in charge of refining high-level orchestration policies into medium-level

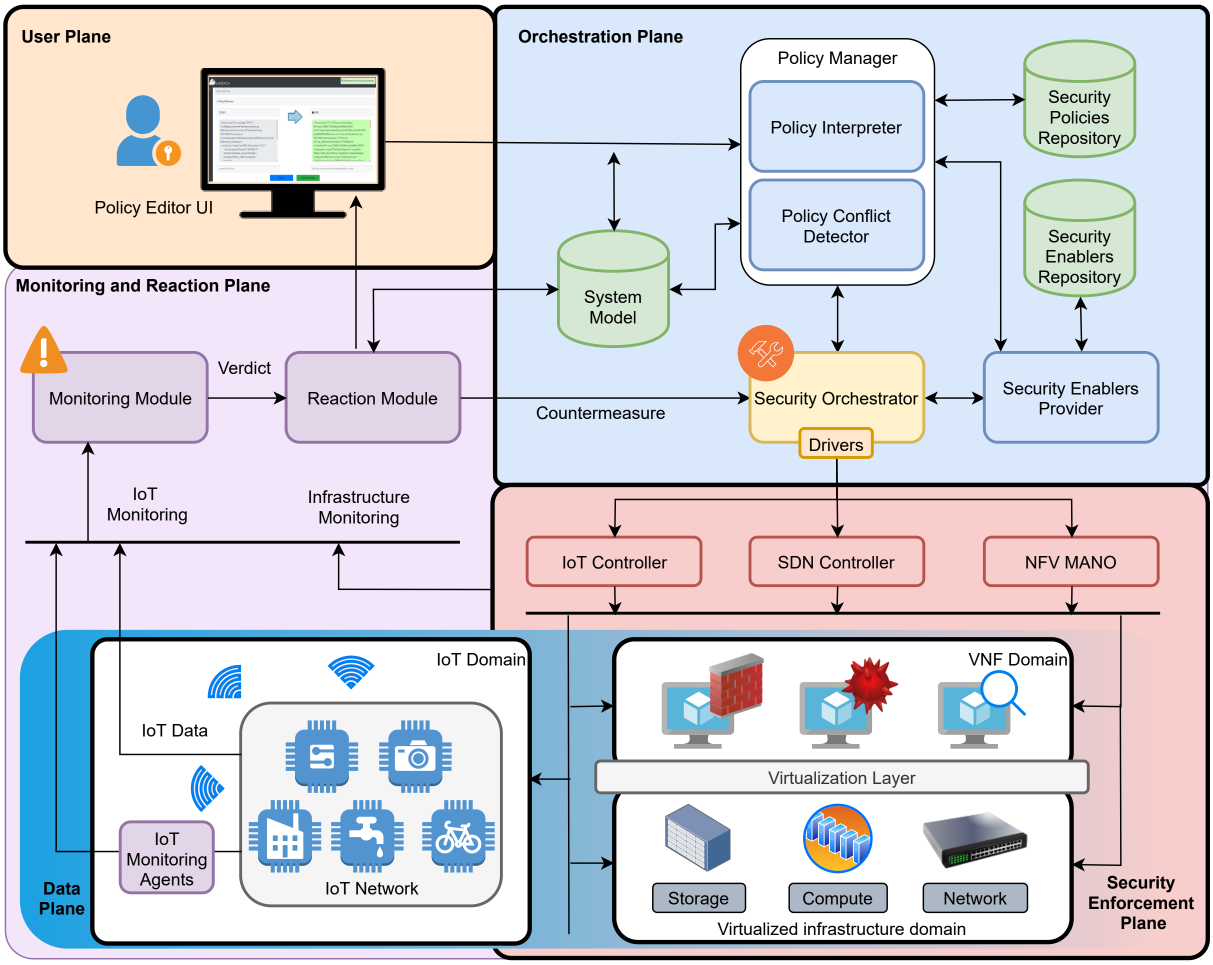

Figure 3.1: Proposed framework architecture design

orchestration policies, as well as translating medium-level orchestration policies into final security enablers configurations. In order to provide a modular and extensible approach, policy translation follows a plugin-based approach which allows that the same security capabilities can be implemented for different security enablers. On the other hand, the Policy Manager also implements the Policy Conflict Detector, in charge of detecting conflicts and dependencies between policies, as well as between policies and the infrastructure. This component also manages the security policies repository, which stores policy templates and instances.

Regarding the Security Orchestrator component, it oversees the orchestration process by analyzing current infrastructure and deciding the best security enabler and location in order to enforce the orchestration security policies. To this aim, the Security Orchestrator component is composed by different modules which cover different functionalities. Specifically, System Model Service provide all the information regarding current deployments and the underlying infrastructure. Enabler and Allocation Managers are focused on the security enabler selection and allocation decisions. Finally, in order to enforce the security enabler configurations, each security enabler can be configured by using its own driver implementation. In this way, for each security enabler we provided an specific translation process through the plugin-based approach and how the security enabler is accessed to through the driver-based approach.

Also in this plane, the Security Enablers Provider manages the Security Enablers Repository. Since security policies are capability-based, each security enabler must be able to cope at least one capability. For instance, ONOS SDN Controller provides filtering and forwarding capabilities among others. In this regard, this component is able to provide those available security enablers which are able to manage the capability required by the security policy. It also stores the security enabler plugin implementation for each available security enabler. Finally, the System Model is the main database of the system, which stays updated with all the information about the current deployment. It models concepts like hardware, software, networks and their relationship, which are instantiated according to the evolution of the infrastructure. This knowledge plays key role during the whole life cycle of the infrastructure.

Security Enforcement Plane

Security Enforcement Plane allows enforcing security policies in different domains. It is compounded by different enforcement points along the whole infrastructure such as IoT Controllers, SDN Controllers, and the NFV-MANOs, as well as other security enablers. IoT Controller component has been designed and implemented to allow enforcing IoT specific configurations for IoT security policies. It abstracts IoT command and control communications with final IoT devices by providing a common northbound management API which communicates with different implementations of IoT specific southbound APIs such as MQTT or CoAP. In fact, the design was based on SDN Controller philosophy, which allows managing the SDN network from a common northbound interface, that transforms the requests into specific southbound network technologies depending on the deployed infrastructure, such as Openflow or NETCONF. In this way, configurations generated by the translation of networking security policies can be enforced through the northbound API of the involved SDN Controller. Finally, NFV-MANO allows deploying and configuring new VNFs on demand over the virtualized infrastructure, as well as re-configuring existing ones. These new deployments and configurations are generated according to the requirements specified in the security policies.

Data Plane

Data Plane provides connectivity between IoT devices and the infrastructure. For instance, between IoT devices and the IoT application (IoT Broker) which receives different IoT measurements such as temperature, CO2, light or humidity. This plane also contains SDN switches which are managed by SDN Controllers in order to modify the data plane behaviour of the network. In fact, it will be configured dynamically for providing new paths that connect new VNFs as part of the security policy requirements. Different agents with monitoring and sensing capabilities can be also connected to the data plane for enhancing capabilities of the Monitoring and Reaction Plane. In the same way, all IoT traffic which passes through SDN switches can be monitored and managed by the SDN network. Besides, some tools are able to monitoring directly IoT wireless network like the MMT 6LowPAN solution [56].

Monitoring and Reaction Plane

Monitoring and reaction Plane oversees monitoring capabilities for the whole infrastructure, as well as identify threats or attacks, also providing specific reaction plans according on the desired countermeasures. It is composed by the Monitoring module, the Reaction module and different sensors and monitoring agents, deployed along the data plane. On the one hand, Monitoring module contains a broker for pre-processing the huge amount of monitoring data. Novelty data analysis components, focused on searching abnormal IoT behaviours are also deployed [121]. Other monitoring tools as well as the SDN itself also provide resources and QoS monitoring, which feeds the system for considering this relevant information during the reaction stage.

On the other hand, the Reaction module contains a Security Information and Event Management (SIEM) module like XL-SIEM[14] or OSSIM[15] which analyzes the notifications provided by the monitoring

[14] https://booklet.atosresearch.eu/xl-siem

[15] https://cybersecurity.att.com/products/ossim

module and selects a reaction plan using an score-based approach, in terms of the suitability of the solution for mitigating the detected threat. These process considers a set of available mitigations, taking into account the available security policies and security enablers. Finally, the Reaction module builds a reactive orchestration policy by retrieving the required templates from the policy repository, according to the selected capabilities. The new reactive orchestration policy is sent to the security orchestrator in order to mitigate dynamically the threat. Alerts and countermeasures associated to the current threat are also notified to security administrators through the User Plane.

3.3.2. Policy Models

The framework proposed by this PhD thesis manages security policy models that extends previous research efforts and solutions, such us the ones provided by I2NSF IETF group as well as some EU project solutions. Specifically, figure 3.2 shows the extension/adaption flow. The proposed HSPL/MSPL Orchestration Policies extend HSPL and MSPL languages defined during SECURED EU project, which also extended concepts like multi-level of abstraction policies from Positif EU project[16], as well as capabilities from the IETF working group. Regarding the amount and type of security policies, table 3.1 shows policy models covered by other projects and solutions. As it can be seen, our solution consider previous works for extending and unifying a wide range of security policies under a single security policy-based solution. Thus, it take the advantage of using well-defined security policy languages, a capability-based and Event-Action-Condition approaches, to model and implement new security and orchestration features. With this in mind, HSPL and MSPL language schemes were extended to enrich current models and fields, also including new capabilities and new models from scratch.

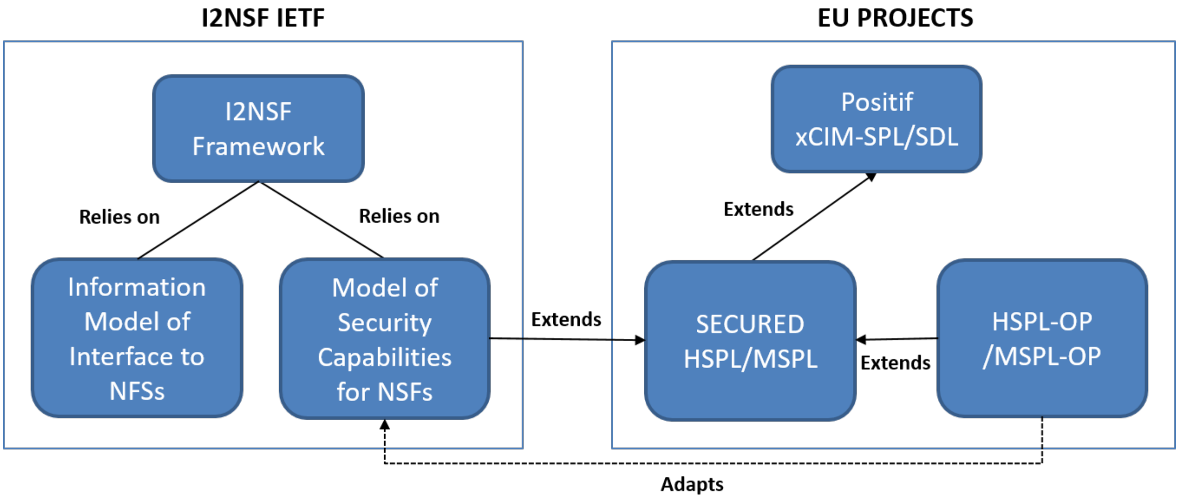

Figure 3.2: Policy models relationship

High-level Security Policy Language Orchestration Policy (HSPL-OP)

High-level Security Policy Language (HSPL) allows even non technical users defining security policies by modelling different high-level security requirements such as *"Alice is authorized to access Internet traffic"* or *"Protect Bob's all traffic confidentiality"*. The scheme provides different combinations for modelling security policies for authorization, channel protection, operation and some other capabilities like reducing bandwidth or checking resources. Regarding the syntax, an HSPL policy element (codified in XML) is mainly composed by an *action (e.g., authorise access)* that must be performed for an

[16] https://cordis.europa.eu/project/id/002314/es

Solution	**Policy Models**
WS-Policy Framework	Specific policy models for web services
Ponder2	Obligation (ECA), Authorization
OASIS (XACML)	Authorization
E-P3P	Privacy
Positif (xCIM)	Authentication, Authorization, Filtering, Channel Protection, Operation
SECURED	Authentication, Authorization, Filtering, Channel Protection, Operation, (other concepts pending to be extended)
Proposed models	Authentication, Authorization, Filtering, Traffic Divert, Channel Protection (also for IoT), Operation, Monitoring, Anonymity, IoT management, IoT Honeynet, QoS, Privacy, Data Aggregation, Orchestration policies.

Table 3.1: Policy models and solutions.

specific *subject (e.g., Alice)* and for an specific *object (e.g., Internet traffic).* It also models conditional fields in order to provide customization parameters to the high-level policy (e.g., time period). Notice that subject can be optional for operational policies (e.g., enable object).

```
[ subject ] action object [ extra_fields ]
```

To provide new features, HSPL language was extended during this PhD thesis by introducing new actions, objects and fields which allows modeling new high-level policies.Besides, new elements and attributes were defined for providing orchestration capabilities to generate HSPL Orchestration Policies. HSPL Orchestration Policies (HSPL-OP) extend HSPL scheme and allows modeling security policies at a high-level of abstraction by adding new capabilities and orchestration possibilities. Thus, while by modelling an HSPL policy it is possible to specify requirements such as an specific *action* that must be performed by a *subject* over a *resource* considering custom options *fields*, now it is also possible specifying priorities between policies as well as whether a high-level policy depends on another policies, or even on system triggered events. In this way, by modelling multiple extended HSPL models, also including their priorities and dependencies, it is possible to compose an HSPL Orchestration Policy. Besides, orchestration policies can also depend on other orchestration policies.

```
[ subject ] action object [ extra_fields ] [ dependencies ] priority
```

Regarding the model extension, from action to extra fields, the meaning is still the same that in the previous case, *action* element represents the kind of action to be performed, related with the subject and the object (e.g., authorise access). It is defined as an enumeration-based element which have been extended with new values for considering also authentication, privacy, monitoring, QoS, data aggregation and network anonymity features. *Object* element values are also provided as enumerated-based values that represents conceptual objects or targets (e.g., Internet traffic). They have also been extended with new values like IoT traffic or IoT authentication traffic. *Extra_fields* is a complex element able to slightly customize the high-level policy by indicating values such as *time_period*, *target*, *purpose* or *resource*. Time period represents the amount of time the security policy must be enforced in the system. Target, allows specifying a target for the policy (e.g., IoT broker). Finally, purpose and resource fields allow specifying additional information about the purpose and the resource involved in the policy (e.g., update specific resource). In fact, purpose and resource elements have also been extended in order to model multiple *properties*, composed by sets of key/value pairs instead of a single string value. Regarding the orchestration features, on the one hand, *Priority* element was defined in order to provide a priority rank during the policy orchestration. In this way, the security orchestrator will consider first those policies with high priority values. On the other hand, *dependencies* element provides a list of dependencies that must be satisfied before processing the security policy. This field considers dependencies between security policies as well as dependencies between security policies and system events (Event-Condition-Action). New attributes were also provided in order to identify the security policy as well as to identify the orchestration policy to which it belongs. Finally, a bi-directional attribute was also provided in order to ease the auto-generation of multiple medium-level security policies during the refinement process, in case the same policy must be enforced in a bi-directional way. More detailed information about the extension and their applicability can be consulted in publication results 4 as well as in other results such as technical reports used for validation [119].

Medium-level Security Policy Language Orchestration Policy (MSPL-OP)

Medium-level Security Policy Language (MSPL) allows modeling medium-level security policies, which contains more technical information but still independent of the underlying infrastructure. Thus, it allows representing information like IP addresses or protocols without representing final configurations for specific security enablers. In the same way that for HSPL, MSPL was also extended for providing new capabilities as well as orchestration features.

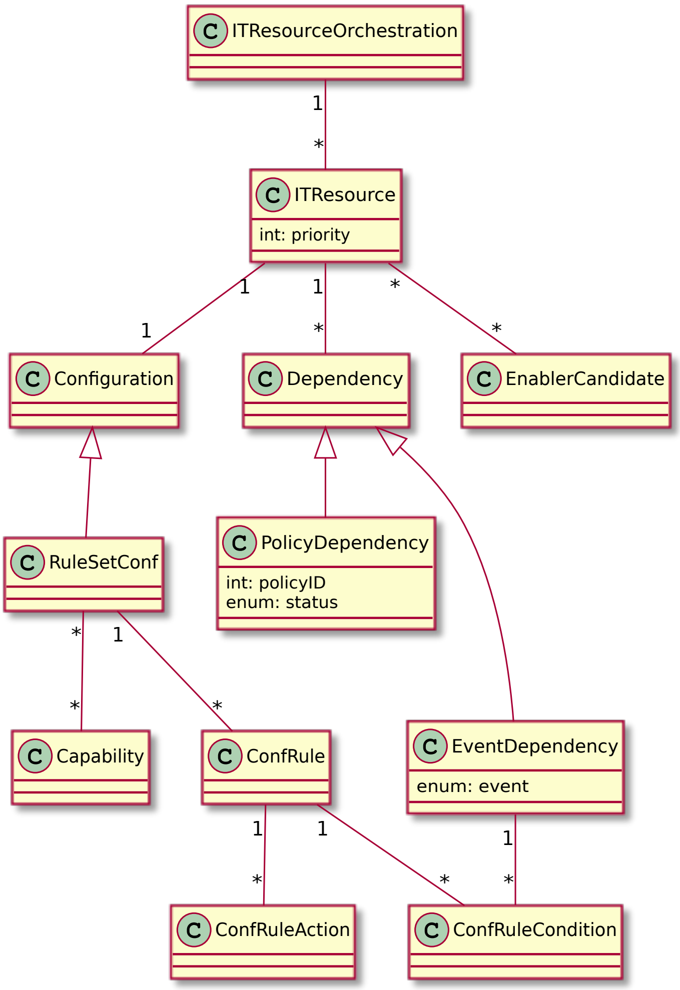

Figure 3.3: MSPL Orchestration policy elements

Figure 3.3 shows a simplification of the main components which compose an MSPL Orchestration Policy. An *ITResourceOrchestration* (MSPL-OP) element can be compounded by multiple *ITResources*

(MSPL) elements which in turn contain a *Configuration*, *Dependencies* and *EnablerCandidates*. *ITResources* also specify the enforcement *priority* as an attribute. *Configuration* element is extended by a *RuleSetConfiguration*, able to represent multiple *Capabilities* as well as multiple *ConfigurationRule* elements. On one hand, *Capability* elements represent main security functionalities such as resource authorisation, filtering or channel protection. In this way, capabilities play a key role as the first step for deciding a suitable security enabler. To enforce a specific security policy, the enabler must implement the required capability. For instance, filtering capability could be enforced through those security enablers which implement in somehow traffic filtering features such as IPTABLES filtering or SDN flows management. On the other hand, each *ConfigurationRule* is able to model different *ConfigurationRuleAction* elements and *ConfigurationRuleCondition* elements. In this way, by extending the model with new actions and conditions it is possible to provide new security policy models in order to cope new capabilities. In fact, multiple models have been extended and designed during this PhD thesis. Figure 3.4 shows the main security policies considered during this thesis, for different security topics. Blue boxes represent an MSPL extension of existing features, whereas green boxes are new security policy designs. Specifically, *Authorization* model was extended with a new definition of authorization action and condition types, as well as it was also extended for considering IoT resources by defining new *IoTApplicationLayerCondition*, which allows modeling IoT protocols or access methods. *Filtering* policy was also extended by including new fields like MAC addresses, QoS conditions or new application layers. *ChannelProtection* model was extended to represent common channel protection parameters used in IoT environments (e.g., DTLS specifications). *NetworkTrafficAnalysis* now allows modeling *MonitoringConfigurationConditions* by specifying monitoring requirements or signature lists, as well as *MonitoringActions* which provide multiple options such as generating alerts, reports, stats or even triggering behavioral analyses (*DataAnalysis*).

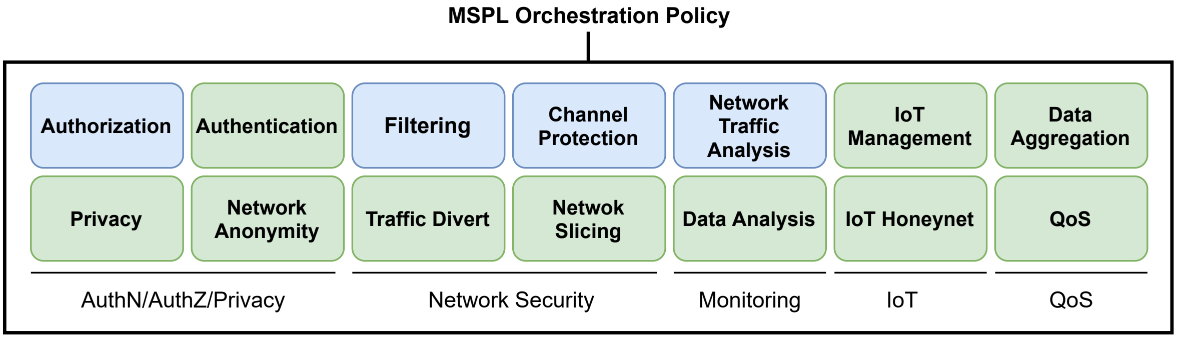

Figure 3.4: MSPL new models (green) and extensions (blue)

About new policy designs, *Authentication* policies allow specifying different action parameters for configuring authentication, such as authentication targets or methods. *Privacy* policies define a new action and condition types for specifying privacy measures, also indicating the required privacy method (e.g., Attribute-based). *NetworkAnonymity* policies allows modeling network anonymity concepts for different techniques like onion routing or traffic mixing by defining new specific technology parameters as part of an anonymity action. *TrafficDivert* defines new actions and conditions for modeling different traffic divert operations such as mirroring or forwarding features. *NetworkSlicing* policies also defines new actions (e.g., quarantine or total disconnection), and conditions for managing network slices, for instance, by specifying the slice id. *IoTManagement* defines new IoT specific actions for IoT command and control, such as power management (turn off, reboot) or IoT resources management (e.g., activate resource). *IoTHoneynet* policy model introduces IoT honeynet capabilities by extending TIHDL [26] language for IoT, as well as including it into the MSPL scheme. In this way, IoT honeynets can be completely modeled as part of an MSPL action. *DataAggregation* policy model allows defining aggregation types for configuring data aggregation according to specific subsets of traffic. MSPLs bandwidth reduction concept was improved in a *QoS* policy model, which defines new condition and action types for specifying QoS parameters such as throughput, delay, priorities or error rates among others. It is important to highlight that in pro of reusability, almost all policy models extend *FilteringConfigurationCondition* elements for represent traffic information.

Regarding dependencies, *Dependency* element was defined and extended with *PolicyDependency* or *EventDependency* elements depending on the nature of the required dependency. Whereas *PolicyDependency* indicates that a security policy depends on a specific status of another security policy (e.g., a security policy requires that another security policy must be enforced before it can be processed), an *EventDependency* specifies that the security policy depends on a specific event triggered by the system (e.g., an authorisation security policy may depend on an authentication success event). Finally, each extended MSPL which composes the orchestration policy can include a *EnablerCandidate* list, which indicates the possible security enabler candidates that should be considered for enforcing the security policy.

During this PhD, the policy modeling evolution and results were were also published in different journals [122], [124], [125], [129], [130]. Other detailed technical reports were also produced for validation during the ANASTACIA EU project [118], [120] where the PhD student is the main author.

3.3.3. Policy Transformation

In order to convert HSPL/MSPL policies in final configurations that can be enforced along the infrastructure through different security enablers, policy transformation processes are required. To this aim, we considered main concepts from literature for defining policy refinement and policy translation processes to manage policy transformations between different levels of abstraction.

Policy Refinement

Policy refinement process allows refining high-level security orchestration policies into medium-level security orchestration policies. Figure 3.5 shows a simplified workflow for HSPL-OP refinement operation (detailed workflow can be found in chapter 5.3). This process is triggered by a security administrator, who instantiates new HSPL orchestration policies by composing multiple extended HSPL policies through friendly forms offered by our Policy Editor Tool. Of course HSPL-OP can be also provided by codifying directly XML models. Once HSPL-OP has been modeled, security administrator requests the policy refinement to the Policy Interpreter. Policy Interpreter analyzes each HSPL which composes the HSPL-OP, mapping high-level terms into capabilities. For instance, the combination of *no_authorise_access* action and the *coap_traffic* object is associated to *Filtering* capability whereas the combination of *authorise_access* action and *resource* object is associated to *Authorisation* capability. When the capabilities have been identified, Policy Interpreter performs a first enforceability analysis. To this aim, it requests to the Security Enablers Provider a list of available security enabler plugin implementations, for ensuring all capabilities can be enforced in the system for at least one security enabler. Otherwise the Policy Interpreter returns a non-enforceability report. In case there are security enabler plugin implementations for covering all the identified capabilities, Policy Interpreter continues processing each HSPL policy by analyzing high-level terms and gathering information for refining them by using the system model. For instance, high-level term *coap_traffic* is refined as UDP protocol and 5684 port if so was established in the system model. When this information is gathered, Policy Interpreter builds MSPL policies according to the identified capabilities, system model information, dependencies (if any) and the list of security enabler candidates for finally, composing the MSPL orchestration policy which is provided to the security administrator as result of the policy refinement process.

Policy Translation

Translation process allows security administrators and system components to translate MSPL-OP into final security enablers configurations according to the selected security enabler, for each MSPL

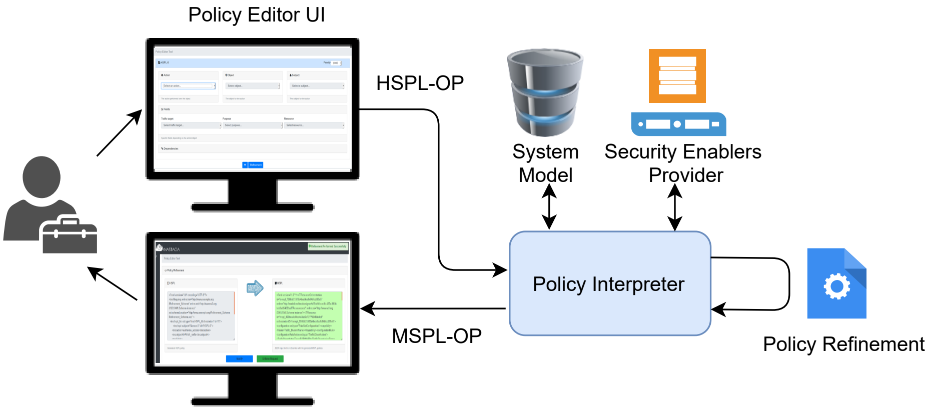

Figure 3.5: HSPL-OP refinement process

policy which compose the MSPL orchestration policy. Figure 3.6 shows a simplified workflow for policy translation operations (detailed workflow can be found in chapter 5.3). This process can be triggered by the security administrator or by the security orchestrator. In the first case, security administrator specifies manually the ID of the security enablers that will be used for each MSPL composing the MSPL orchestration policy. In the second case, the Security Orchestrator decides automatically the best security enabler according to available infrastructure information in the system model, also considering results from conflict detection process, which is explained in next section. When the policy interpreter receives the MSPL-OP, it analyzes each MSPL, requests the required plugin to the Security Enablers Provider, and loads it dynamically for performing the policy translation. During the translation process, the system model is also required for filling specific infrastructure information, which is not available in the security policy. For instance, information like the switch port where an IoT gateway is connected to. When all MSPL policies which composes the MSPL orchestration policy have been translated, the Policy Interpreter returns a data structure that includes the correspondence between each MSPL and the security enabler configuration.

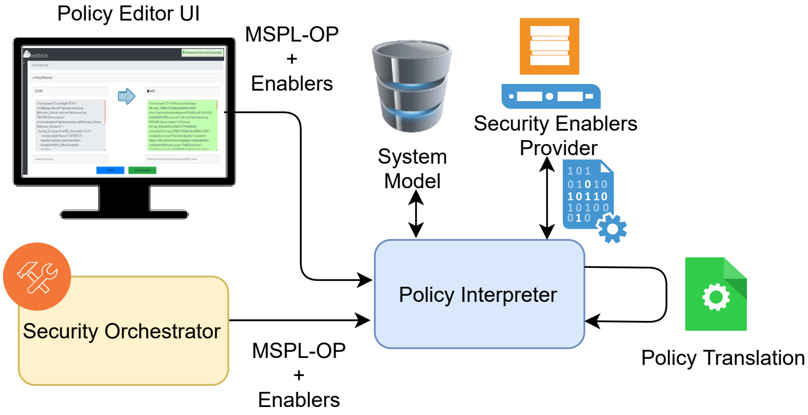

Figure 3.6: MSPL-OP translation process

The evolution of policy translation and refinement processes for providing new security features were provided in different journal publications, culminating in [130]. In fact, in [127] [128] also Manufacturer Usage Description (MUD) profiles management was integrated in the policy management process.

Policy translation were also validated in ANASTACIA EU project technical reports [120] where the PhD student is the main author.

3.3.4. Policy Conflict and dependencies detection

Policy conflict detection allows identify conflicts and dependencies between; (i) security policies which compose the orchestration policy (intra-policy), (ii) between those policies and the ones that have been already enforced in the system (inter-policy), and (iii) between the orchestration policy and the infrastructure by considering the system model information. Figure 3.7 shows a simplified interaction to perform the conflict and dependencies detection process. At startup time, the rule engine loads infrastructure information, as well as current policies enforcement information from system model and policy repository respectively. It also compiles conflict detection and dependency rules, provided by the security administrator. Since this information is modeled and introduced in the knowledge base as a set of facts, this process requires some time depending on the magnitude of the infrastructure. However, this process is only performed once. After startup process, conflict detector is subscribed to the system model and policy repository events for updating current facts dynamically. After the system is ready, Security Orchestrator, or security administrator can request orchestration policy enforcements which include (or not) the selected security enabler for each MSPL, depending on the stage of the enforcement process (proactive/reactive).

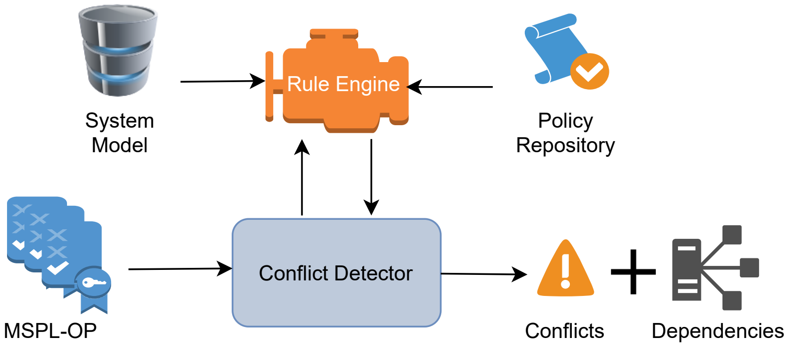

Figure 3.7: MSPL-OP conflict and dependencies detection

When conflict detector receives the orchestration policy, it executes the rule engine to verify if the new facts (MSPLs) present any kind of conflict or dependency, matching them against the predefined rules which takes into account both, semantic and context-based conflicts. Different rules for detecting conflict and dependencies were considered for designing, implement and validate the current proposal. For instance, `Same behaviour conflict`, which verifies if two security policies will provide the same security behaviour. `Priority dependency conflict` checks if a security policy (A) depends on the enforcement of another one (B) whose priority is lower than A priority. `Duties Conflict Across Policies` occurs when a security policy presents incompatibilities in terms of functionality with security policies that were already enforced (e.g., Channel protection vs Deep packet inspection). `Managers Conflict` appears when a security policy directly contradicts a previous one for the same involved subjects and parameters (e.g, Allow vs Deny). `Override conflicts` verifies if the security policy is overriding in somehow other security policies that were previously enforced (e.g., Allow CoAP traffic, vs Allow all traffic). Regarding dependencies, Event and Policy Dependencies verify if the security policy depends on any kind of event or on another security policy. Apart from those semantic conflicts, also context-based conflicts were considered such as Capability Missing, that verifies if the involved device implements the required capability (e.g. IoT device with DTLS server implementation) and Insufficient resource conflict, which verifies if the involved device disposes of enough resources for managing the policy enforcement. For instance, to verify if the IoT device reached the maximum number of DTLS connections.

Results of policy conflict and dependencies detection were provided in a journal publication [130], as well as it was validated in ANASTACIA EU project technical report [119] where the PhD student is the main author.

3.3.5. Proactive/Reactive Policy Enforcement

Policy enforcement process allows enforcing HSPL/MSPL orchestration policies across the whole infrastructure. Depending on the security requirements and the situation, policy enforcement process can be triggered by the security administrator in a proactive way as part of security measures, but it can be also triggered by the reaction module as part of dynamic countermeasures. Figure 3.8 shows a simplified workflow for MSPL-OP enforcement. When the security orchestrator receives an MSPL-OP enforcement request, it asks for a preliminary analysis to the Conflict Detector in order to identify conflicts or dependencies between security policies, as well as between security policies and the system. If any security policy presents any conflict, the process is aborted until conflicts have been solved. Otherwise, the process continues analyzing dependencies. Any security policy with any kind of dependency is queued until the dependency has been solved. Those security policies without dependencies are provided to the orchestration process which decides the best security enabler to perform the enforcement. To this aim, the orchestration algorithm retrieves the security enabler candidates from the Security Enablers Provider, and starts an allocation optimization process which decides a suitable policy enforcement point according to the available security enabler plugins, the security policy information, system model information, and the allocation algorithm. The later is also considered since different allocation algorithms can be provided for covering different approaches. For instance, if the infrastructure does not have NFV-MANO features, a *ConfOnlyAllocation* algorithm, which only looks for enforcing policies in the most suitable allocation place among the already deployed security enablers can be selected, otherwise, if the infrastructure allows dynamic deployments, advanced optimization algorithms such as weight-based, scored, greedy, Mixed-Integer Linear Programming, ant colony-based or deep-learning-assisted are required. It is important to highlight that during the optimization process, multiple conflict verifications can be performed for ensuring that selected security enablers and allocation resources are fully compliant with the security policy requirements. Once the orchestration process decided the most suitable security enablers and locations for each MSPL policy, it requests the policy translation to the Policy Interpreter, that performs the translation process as it was explained in subsection 3.3.3, returning the final security enabler configurations to the orchestrator. The Security Orchestrator then starts the enforcement process, which triggers the configuration enforcement through the specific driver implementations for each security enabler.

Since the Security Orchestrator is subscribed to system events and security policies updates, as security policies change their status or new events are generated in the system (e.g., Authentication success event), the Security Orchestrator receives a notification. Then, it analyzes if the new status solves any pending dependency. If so, a new security policy enforcement process is unchained, which after ensuring the policy enforcement over the proposed security enablers, releases the policy from the dependencies queue.

3.3.6. Proposed virtual IoT Security Enablers

In order to enhance proactive and reactive security and privacy features in our framework, different virtual security enablers devised for IoT were proposed, designed, implemented and validated during this PhD thesis. For each one of them, translator plugins and enforcement drivers were also designed and implemented. Figure 3.8 shows the security enablers we instantiated during this thesis for endowing the framework with a wide range of security capabilities. Following we provide an overview of the main proposed security enablers.

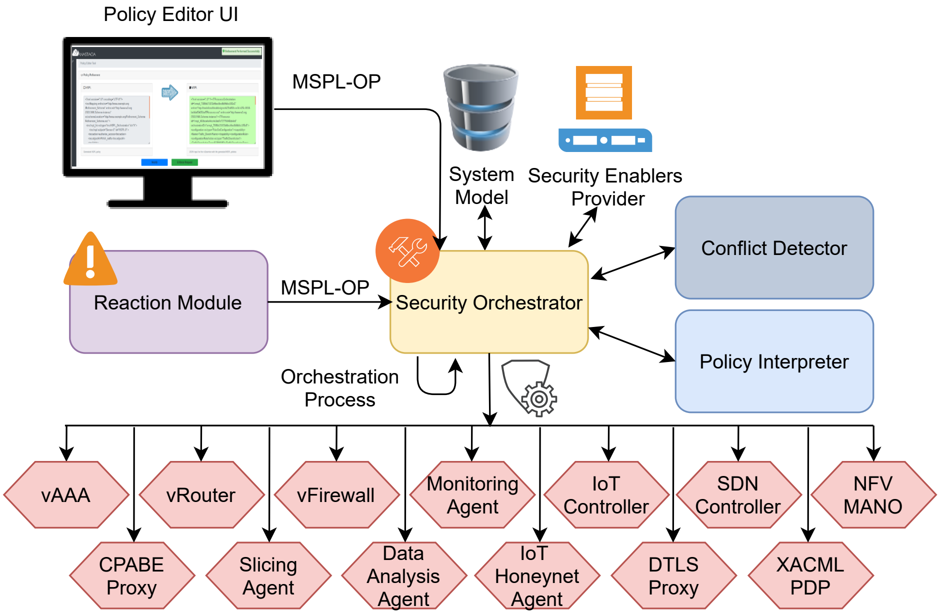

Figure 3.8: Orchestration and MSPL-OP enforcement along different security enablers

vAAA

Virtual Authentication, Authorization and Accounting (vAAA) VNF security enabler allows deploy, configure and reconfigure part of AAA infrastructure on demand by following the NFV approach. If we think in massive IoT devices trying to authenticate to the system, the ability to deploy on demand new authentication agents as near as possible of those devices contributes to flexibility and scalability properties of the framework. Besides, in combination with SDN paradigm, this approach becomes even more flexible. For instance, SDN controller detects when an IoT device is asking for an authentication process (e.g., considering destination port) so it notifies to the system in order to deploy a new orchestration policy, composed by different security policies such as authentication, resource authorisation and traffic divert. Authentication policy will generate the re-configuration or deployment and configuration of a new authentication agent, which implements the required protocol for carrying authentication according to the authentication request (e.g., PANA or COAP-EAP). The authorization policy, which depends on the authentication success event, will configure the authorization in the Policy Decision Point (e.g., XACML PDP) depending on the IoT device specifications, as well as previous information provided by the security administrator. Finally, traffic divert security policy redirects authentication traffic against the re-configured or newly deployed authentication agent.

Filtering / Traffic Divert / Network Slicing

For enhancing security through dynamic network management, different security enablers have been provided. The framework provides compatibility with different SDN Controllers such as ONOS and OpenDaylight to manage the SDN network by enforcing security policies. Moreover, the plugin/driver-based approach makes easy to include new controllers if required. Regarding VNF-based networking security enablers, on the one hand, the virtual router (vRouter) security enabler allows enforcing dynamically L3/4 networking security policies over a virtual router instance, to manage a more traditional approach of networking directives such as IPTABLES. On the other hand, vFirewall security enabler provides a virtual SDN switch, which allows managing L2-L4 networking policies on demand. By using these kind of security enablers it is possible to modify the network behaviour dynamically depending on the security policies requirements. For instance, they allow performing networking operations such as filtering, forwarding, traffic mirroring or data rewriting if required. Network slicing policies were also validated over a 5G network slicing agent provided by Ericsson, which is in charge to isolate 5G traffic in different network slices according to the network slice security policy configuration.

Channel Protection

Nowadays, providing security in communications is essential and in most cases is the by-default behaviour as it should be. However, not all IoT devices are able to provide such capabilities due their constraint nature. In this regard, a security enabler such as the Datagram Transport Layer Security (DTLS) proxy, allows deploying dynamically a proxy which provides channel protection capabilities to those devices which does not support channel protection features, or whose channel protection features are considered weak. By locating the new VNF as near as possible of the affected IoT devices it is possible to reduce considerably the amount of time and hops that the traffic travels unprotected. Moreover, depending on the required channel protection capabilities, different channel protection proxies could be dynamically deployed (e.g., DTLS, TLS, IPSec). Regarding crypto-key management, whereas in our work cryptography material is generated during the bootstrapping process, and it is managed and provided by the security orchestrator to the involved entities, it can also be also performed by the SDN controller like authors used for IPSec in [30]

Data Privacy

In the same way that for channel protection, privacy becomes fundamental in IoT, specifically when we talk about devices which retrieve sensible data day by day. Thus, data privacy must be provided for ensuring that sensible data is only accessed by authorized entities. In this regard, E2E data privacy can be achieved by dynamically managing E2E data-level encryption for specific resources (e.g., only specific user can decipher their data), by enforcing data privacy security policies. Unfortunately, as in the previous case, the most constrained devices are not able to perform the required cryptography operations to ensure privacy at data level. In this regard, data privacy policies can be enforced by deploying dynamically different data privacy proxies which implement different data privacy techniques, such as the Ciphertext-Policy Attribute-Based Encryption (CP-ABE). In this way, the solution allows providing data privacy features to those devices which are not able to provide them by default. As in the channel protection case, it can be deployed as a middle-box as near as possible of the required device, for ensuring data privacy during the most part of the communication path.

Monitoring/Data Analysis

Monitoring and data analysis capabilities play a key role in security, and this combination becomes even more important (if possible) when countermeasures are provided automatically as is the case. The framework provides different VNF monitoring agents in different scopes that can be dynamically re-configured or deployed and configured on demand. For monitoring network traffic, security enablers such as Snort, Suricata or MMT tool [56] allow monitoring several parts of the network by analyzing traffic information at different levels and technologies. For instance, the latter allows monitoring directly IoT networks like 6LoWPAN. SDN-enabled components such as SDN switches also allows retrieving valuable information that can be used for detecting threats as well as ensuring QoS. For instance, a monitoring app deployed in the top of the SDN controller can be dynamically configured for notifying the system according to specified thresholds. In fact, other security enablers can be dynamically configured for feeding the monitoring system (e.g., vAAA for authentication errors, XACML PDP for authorization errors). Besides, apart from monitoring security enablers, different data analysis agents like UTRC agent [121] can be also deployed in proactive or reactive way, for analyzing possible threats according to the behaviour of the system. These kind of solutions train the system to discern abnormal behaviours from regular behaviours of the infrastructure (e.g., data manipulation vs real measurements). Finally, all the information generated by the different monitoring and analysis tools is correlated in a SIEM engine for providing reaction alternatives according to the identified threats.

IoT management / IoT-Honeynet

Due IoT features such as heterogeneity and massive deployments, IoT management as well as IoT security management becomes a huge challenge. In this regard, our IoT Controller implementation provides a common interface which allows interacting with different IoT environments and technologies. By following the SDN controller approach, it provides a northbound API which translates the requests into different IoT specific southbound interfaces (e.g., CoAP or MQTT). Besides, due the modular implementation, new southbound interfaces can be easily included by implementing few methods. Regarding the IoT honeynet, our design and implementation of the IoT honeynet agent allows to deploy on-demand new reactive VNFs, which replicate physical IoT environments for different IoT technologies (e.g., Contiki, uPython). To this aim, the translation process that obtains the configuration of this security enabler relies on the information of the real IoT infrastructure, stored in the system model (e.g., gathered during the IoT bootstrapping process). Of course, this information must be continuously updated. In this way, specific IoT honeynets can be deployed dynamically as a countermeasure to mitigate an ongoing IoT attack. When an IoT threat is detected, a reactive orchestration composed by an IoT honeynet policy, that models the affected IoT environment, and different traffic divert policies is enforced. On one hand, the IoT honeynet policy enforcement will replicate the specified IoT environment through the IoT honeynet agent. On the other hand, the traffic will be redirected transparently between the attacker and the IoT honeynet environment, once the solution has been properly deployed. Besides, the VNFs are also endowed with monitoring properties for analyzing the attacker's behaviour.

3.3.7. Use Case: Mitigating a multi-attack on Smart Building

The proposed framework design, implementation, the security policy models, and the proposed security enablers were validated along multiple use cases in different publications, as well as in real environments. Figure 3.9 shows one of the most relevant use case we used for validating our approach. In this scenario all traffic is denied by default at the beginning (excepting discovering protocols such as arp or ndp) for the sake of security. Thus, at startup time, the security administrator composes an HSPL Orchestration Policy to authorize authentication traffic for specific locations (e.g., floors, rooms, labs). He/She also specifies future authorizations by indicating policy dependencies, which will be enforced once the IoT devices are properly authenticated. Secure traffic between IoT devices and IoT controllers are also allowed in order to perform IoT command and control operations. Finally, also monitoring policies are provided in order to detect traffic that does not matches with the authorised ones.

Once the authentication traffic has been allowed, IoT devices perform the authentication process which also unchains authorization processes, according to the orchestration policies defined by the security administrator. In this scenario, IoT devices were allowed to put sensing information in an IoT broker, once they were properly authenticated. At this point, an internal attacker manipulates physically an IoT device with the aim to start a complex attack over the infrastructure. As first step, the compromised IoT device performs an scouting process to reach other devices. Since monitoring tools (including wireless IoT monitoring tools) were properly configured from the beginning, the scouting attempting is detected by wireless/wired IDSs agents, who notify the issue to the monitoring module, which in turn feeds the SIEM. Then the SIEM generates a set of suitable countermeasures in terms of capabilities. In this case, it selects IoT power management (reboot), IoT isolation (filtering) and proactive IoT data analysis capabilities. Those capabilities are communicated to the reaction module which retrieves system model and policy repository information, for generating multiple MSPL policies that compose the reactive MSPL orchestration policy. This MSPL-OP is then provided to the security orchestrator to be enforced into the infrastructure. The security orchestrator executes the orchestration algorithm who selects the most suitable security enablers for enforcing each MSPL composing the orchestration policy.

This orchestration process also performs policy conflict detection operations in order to ensure

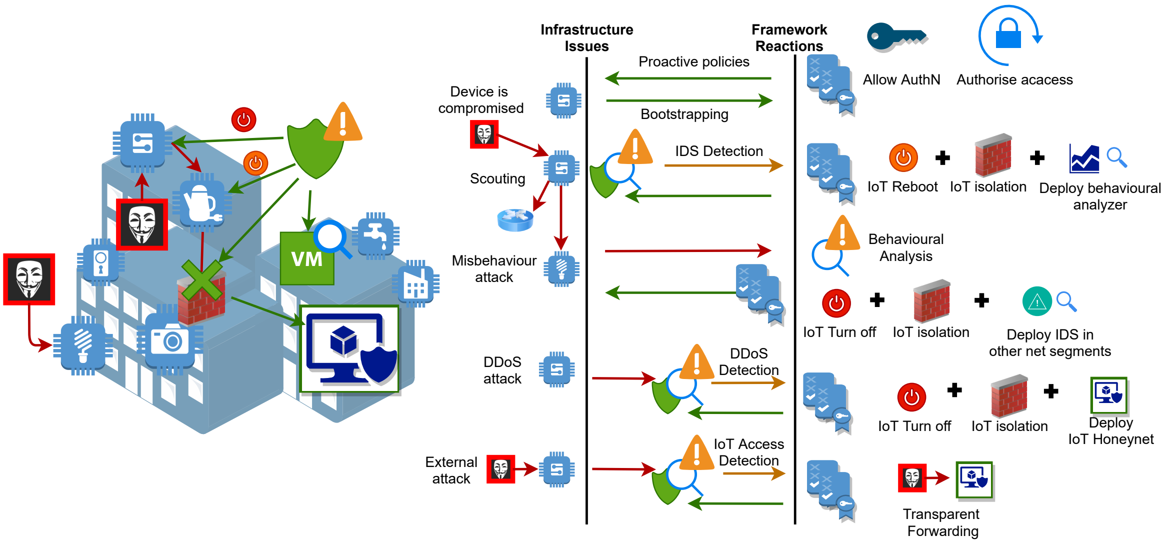

Figure 3.9: Applying our proposal for mitigating Multi-attack on Smart Building

that the mitigation will not introduce any kind of issue in current deployment. In order to reboot the IoT device, the security orchestrator selects the IoT controller, who manages the affected IoT device. To enforce the filtering policy it selects the closest connection point to the IoT device, which in this case is the first SDN switch where the affected IoT device traffic pass through. For the IoT behaviour analysis, a new VNF will be instantiated for analyzing the IoT data received for the suspect segment of the IoT network. Once the suitable security enablers and locations have been selected, the security orchestrator requests the policy translation for each security policy and security enabler to the policy interpreter. The policy interpreter uses the proper plugins for the specified security enablers to perform the policy translation process. When low-level configurations have been generated, the security orchestrator enforces them by using the proper driver implementation for each security enabler. In this case, IoT command and control is performed through the northbound API of the IoT controller, filtering configuration is applied through the northbound API of the SDN controller, and the new VNF is instantiated through the NFV-MANO, whereas the configuration is provided trough the correspondent driver for the data analyzer agent. In this way, the compromised IoT device has been rebooted and isolated from the SDN network and more advanced techniques of IoT threat detection have been deployed.

Unfortunately, the compromised IoT device propagated the infection through the 6lowPAN network, so other IoT devices in the wireless range were also infected. As part of the multi attack, new infected devices start sending abnormally high temperature values, as if the building was in fire. The fire alarm is triggered, but the new data analysis agent detects the misbehavior, and notifies it to the security administrator and the SIEM. The security administrator verifies there is no fire so he/she turn off the fire alarm. Meanwhile, the SIEM identified the new threat, and generated a new set of countermeasures, also considering the previous threat. In this case, all affected IoT devices will be turned off and isolated, as well as those IoT devices which could reach other rooms by wireless, since it seems that the infection is being spread through the wireless connection. Besides, new advanced monitoring tools (VNFs) are deployed in other segments of the network for verifying if the infection reached them. In fact, a more sophisticated attack, an slowDoS attack, starts from a nearby room against the Building Management System as part of the attacker plan. This kind of attack pursues to exhaust system resources by using techniques like long-lived connections. However, the new IDSs deployed were also trained to detect the SlowDoS attack which is reported to the SIEM that in this case, apart of turn off the affected devices, it also isolates the affected rooms in order to apply a hard quarantine. However, it deploys first IoT honeynets for emulating the affected rooms, since our solution is able to replicate physical IoT devices, firmwares, IoT services as well as physical IoT wireless networks. Finally, when the attacker tries to access any of the compromised IoT devices, the traffic is redirected transparently from/to the IoT honeynet, which allows to analyze the behaviour of the attacker, as well as retrieving and study new infections and compromised firmwares in a safe environment. It is important to highlight that the framework provides full traceability of the processes and decisions so security administrators can verify in real time detected threats, countermeasures as well as the current security and privacy level of the system according to the status of the infrastructure.

This multi-attack scenario served for evaluating our framework against a real complex use case in a real deployment. This evaluation has allowed to validate the feasibility and performance of the framework proposed in this thesis, and its holistic security management capabilities for IoT, encompassing; proactive and reactive security policy handling (including translation, refinement, conflict detection) and the autonomic enforcement of our (given attacks detectors) implemented virtual security functions (vAAA, vFirewall, vChannelProtection, vIoTHoneynet) to mitigate cyber-attacks. Results and measurements can be found across our different publications in section 5.3.

3.4. Lessons Learned and Conclusions

IoT devices are now part of our lives in both, home and industrial environments, and statisticians expect a continuous increasing torrent according to the current trend. Despite IoT paradigm provides awesome features and improvements in a wide range of domains such as wearable, smart city, smart grid, health or smart farming, this paradigm also entails different challenges, specifically in terms of security and privacy. In fact, often the constraint computation nature of IoT devices makes difficult to execute different ciphering functions in order to ensure essential security properties such as channel protection. Besides, due to the small size, as well as the low cost of its components (or the single component), huge deployments of IoT devices are incorporated massively to current infrastructures day by day. Further, apart from the basic heterogeneity between the different IoT devices required to cope with multiple functions, IoT deployments are thought to be deployed during years, so it is quite common that new IoT devices or replacements execute different firmwares in different models from different manufacturers. This heterogeneity also impacts directly in security. For instance, customized behaviours at commissioning time make difficult to ensure, in an unified way, some security basics from the beginning. Actually, a lot of devices perform operations from the beginning that the user is not aware of, such as sending information continuously to manufacturer “only for statistics purposes” which in general also entails privacy issues, especially when we talk about sensible data.

In order to cope with those challenges, this PhD thesis proposed, designed, implemented and validated a policy-based framework for security management in next-generation SDN/NFV-enabled Internet of Things infrastructures. The solution allows defining and enforcing capability-based proactive and reactive orchestration security policies, considering the current information of the infrastructure. To this aim, we defined High-level Security Policy Language Orchestration Policy (HSPL-OP) and Medium-level Security Policy Language Orchestration Policy (MSPL-OP) by extending existing policy models in the state of art. Our extension endows the framework with a wide range of security capabilities such as authentication, authorisation, network management, IoT management, IoT honeynet, channel protection, privacy and network slicing and orchestration capabilities among others. We designed and implemented the required components and workflows for modeling, refining and translating those security policy models, as well as for orchestrating and enforcing them across the SDN/NFV/IoT infrastructure, through multiple security enablers specially designed for IoT. The implementation followed a plugin/driver approach to ease including new translation/enforcement logic for new security policies or security enablers.

In that sense, we designed, implemented and validated workflows, capabilities, translator plugins and enforcement drivers to refine, translate and enforce networking security policies for isolating IoT compromised devices through several SDN controllers and virtual firewalls. Final configurations were enforced through multiple network technologies (e.g., OpenFlow and NETCONF). In this regard, we provided a comparison of the whole mitigation process for different networking security enablers and policies. The results showed the benefits of the SDN approach as security mechanism integrated in IoT environments.

For managing authentication and authorization policies, we designed, implemented and validated a novel dynamic AAA agent able to be deployed on demand as near as possible of the IoT devices. Besides, different authentication agents can be configured and deployed depending on the authentication and authorization requirements of the IoT devices (e.g., PANA, CoAP-EAP). We also provided policy-based IoT channel protection by default (e.g., DTLS) as part of the bootstrapping process, as well as capability-based authorization (e.g., DCapBAC). The validation of this approach was performed by managing dynamic IoT bootstrapping processes, including authentication, authorization, and channel protection for different amount of IoT devices and policies. Those results showed the feasibility and performance of the softwarized, centralized and dynamic solution, based on a novelty approach driven by the security orchestrator, that exploits SDN/NFV for and efficient authentication, authorization and channel protection on IoT scenarios.

To validate monitoring security policies and reaction capabilities, our framework was integrated in the ANASTACIA EU project architecture, where co-authors provided monitoring and reaction tools. By using those tools, different IoT threats were detected and new reactive security policies were instantiated. Specifically, the reactive security policies were dynamically enforced though the SDN network, as well as through our implementation of the IoT Controller, who managed to reconfigure the affected IoT devices by using specific IoT protocols (e.g. CoAP, MQTT). This contribution was validated by generating bursts of monitoring events at different frequencies, and enforcing reactive security policies accordingly. The results showed the feasibility of the reactive policies enforcement across different security enablers.

Regarding the NFV-based automatization and virtualization of IoT devices, we designed, implemented and validated the first approach for deploying reactive policy-based IoT honeynets, able to replicate real IoT infrastructures. To this aim, we extended a current state of art honeynet policy model (TIHDL) with IoT concepts, and we homogenized the new model to be part of the MSPL policies, as a new capability. Thus, the reaction module is able to request a reactive IoT honeynet deployment by modeling an IoT honeynet policy as well as different forwarding policies. Of course, accuracy information about the real IoT deployment must be continuously updated in the system model. To properly enforce the new policies, we also defined a first dependencies workflow in order to deal with restrictions such as those that occur when the forwarding policies enforcement depends on previous enforcement of the IoT honeynet policy. This research was validated by replicating dynamically multiple IoT topologies for different models and firmwares according to the real IoT infrastructure. The results showed the feasibility and performance of the new solution, nonexistent until that moment, by combining SDN/NFV and different IoT virtual environments which allow deploy on demand IoT honeynets in reactive way.

To provide complex reactive capabilities, we defined, implemented and validated orchestration policies, as well as policies conflict detection and dependencies management. Orchestration policies contain multiple security policies with different priorities, which can also depend on other security policies, or on events triggered by the system. To manage them, as well as to manage possible conflicts between policies or between policies and the infrastructure, we also designed and implemented the policy conflict and dependencies detection process. This process loads the information regarding the infrastructure and the security policies as facts. Those facts will be used to verify possible issues at policy enforcement stage. This research was also validated for different rules, facts and security policies.

We also jointly validated the proactive/reactive security orchestration capabilities of the framework against a multi-attack scenario over a real smart building. An internal attacker compromised an IoT device which unchains different kind off attacks and infections (scouting, misbehaviours, slow DoS, external attack). Those threats were detected and dynamically mitigated by different reactive orchestration policies through different security enablers, specially designed for IoT. The results of this experiment showed the feasibility of the framework to deal with multiple attacks at real time over a real smart building scenario.

According to the results, we consider the framework a valuable reference for enhancing security in IoT environments. The policy-based approach, the new policy models, proactive and reactive enforcements, security orchestration, conflicts and dependencies management, and the plugin/driver-based solution, contributes to mitigate multiple IoT challenges such as the ones exposed during the thesis. However, we think the security orchestration *intelligence* must be enhanced to enforce more accuracy and effective countermeasures across the infrastructure. In this regard, new allocation and optimization algorithms could be taken into account simultaneously, depending on the policy requirements (e.g., greedy, scored, MILP...). Besides, those algorithms should also consider security properties during the process to make the best decision for the sake of security.

Finally, we want to highlight that the results of this PhD thesis, as well as the implementation of the different components have been, and are being exploited and reused in H2020 EU projects such as ANASTACIA and INSPIRE 5G+.

Chapter 4

Publications Composing the PhD Thesis

4.1. Enhancing IoT security through network softwarization and virtual security appliances

Title	Enhancing IoT security through network softwarization and virtual security appliances
Authors	Alejandro Molina Zarca and Jorge Bernal Bernabé and Ivan Farris and Yacine Khettab and Tarik Taleb and Antonio Skarmeta-Gómez
Type	Journal
Journal	International Journal of Network Management
Impact factor (2018)	1.231
Publisher	Wiley
Pages	e2038
Volume	28
Issue	5
Year	2018
Month	July
ISSN	1099-1190
DOI	`https://doi.org/10.1002/nem.2038`
URL	`https://onlinelibrary.wiley.com/doi/abs/10.1002/nem.2038`
State	Published
Author's contribution	The PhD student, Alejandro Molina Zarca, is the main author of the paper

Journal details: International Journal of Network Management
ISSN: 1099-1190
Publisher: Wiley
Impact factor (2018): 1.231
Website: `https://onlinelibrary.wiley.com/journal/10991190`

Authors – Personal details	
Name	**Alejandro Molina Zarca**
Position	PhD student of the Department of Information and Communications Engineering
University	University of Murcia
Name	**Dr. Jorge Bernal Bernabé**
Position	Postdoctoral Researcher of Department of Information and Communications Engineering
University	University of Murcia
Name	**Ivan Farris**
Position	PhD student
University	Aalto University
Name	**Yacine Khettab**
Position	PhD student
University	Aalto University
Name	**Tarik Taleb**
Position	Full Professor
University	Aalto University
Name	**Dr. Antonio F. Skarmeta Gómez**
Position	Professor of the Department of Information and Communications Engineering
University	University of Murcia

Abstract

Billions of Internet of Things (IoT) devices are expected to populate our environments and provide novel pervasive services by interconnecting the physical and digital world. However, the increased connectivity of everyday objects can open manifold security vectors for cybercriminals to perform malicious attacks. These threats are even augmented by the resource constraints and heterogeneity of low-cost IoT devices, which make current host-based and static perimeter-oriented defense mechanisms unsuitable for dynamic IoT environments. Accounting for all these considerations, we reckon that the novel softwarization capabilities of Telco network can fully leverage its privileged position to provide the desired levels of security. To this aim, the emerging software-defined networking (SDN) and network function virtualization (NFV) paradigms can introduce new security enablers able to increase the level of IoT systems protection. In this paper, we design a novel policy-based framework aiming to exploit SDN/NFV-based security features, by efficiently coupling with existing IoT security approaches. A proof of concept test bed has been developed to assess the feasibility of the proposed architecture. The presented performance evaluation illustrates the benefits of adopting SDN security mechanisms in integrated IoT environments and provides interesting insights in the policy enforcement process to drive future research.

4.2. Enabling Virtual AAA Management in SDN-Based IoT Networks

Title	Enabling Virtual AAA Management in SDN-Based IoT Networks
Authors	Alejandro Molina Zarca and Dan Garcia-Carrillo and Jorge Bernal Bernabé and Jordi Ortiz and Rafael Marin-Perez and Antonio Skarmeta
Type	Journal
Journal	Sensors
Impact factor (2018)	3.275
Publisher	MDPI
Pages	295
Volume	19(2)
Year	2019
Month	January
ISSN	1424-8220
DOI	`https://doi.org/10.3390/s19020295`
URL	`https://www.mdpi.com/1424-8220/19/2/295`
State	Published
Author's contribution	The PhD student, Alejandro Molina Zarca, is the main author of the paper

Journal details: Sensors
ISSN: 1424-8220
Publisher: MDPI
Impact factor (2018): 3.031
Website: `https://www.mdpi.com/journal/sensors`

Authors – Personal details	
Name	**Alejandro Molina Zarca**
Position	PhD student of the Department of Information and Communications Engineering
University	University of Murcia
Name	**Dr. Dan Garcia-Carrillo**
Position	Postdoctoral Researcher
Research Centre	Odin Solutions
Name	**Dr. Jorge Bernal Bernabé**
Position	Postdoctoral Researcher of Department of Information and Communications Engineering
University	University of Murcia
Name	**Dr. Jordi Ortiz**
Position	Postdoctoral Researcher of Department of Information and Communications Engineering
University	University of Murcia
Name	**Dr. Rafael Marin-Perez**
Position	Postdoctoral Researcher
Research Centre	Odin Solutions
Name	**Dr. Antonio F. Skarmeta Gómez**
Position	Professor of the Department of Information and Communications Engineering
University	University of Murcia

Abstract

The increase of Software Defined Networks (SDN) and Network Function Virtualization (NFV) technologies is bringing many security management benefits that can be exploited at the edge of Internet of Things (IoT) networks to deal with cyber-threats. In this sense, this paper presents and evaluates a novel policy-based and cyber-situational awareness security framework for continuous and dynamic management of Authentication, Authorization, Accounting (AAA) as well as Channel Protection virtual security functions in IoT networks enabled with SDN/NFV. The virtual AAA, including network authenticators, are deployed as VNF (Virtual Network Function) dynamically at the edge, in order to enable scalable device's bootstrapping and managing the access control of IoT devices to the network. In addition, our solution allows distributing dynamically the necessary crypto-keys for IoT Machine to Machine (M2M) communications and deploy virtual Channel-protection proxys as VNFs, with the aim of establishing secure tunnels among IoT devices and services, according to the contextual decisions inferred by the cognitive framework. The solution has been implemented and evaluated, demonstrating its feasibility to manage dynamically AAA and channel protection in SDN/NFV-enabled IoT scenarios.

4.3. Security Management Architecture for NFV/SDN-aware IoT Systems

Title	Security Management Architecture for NFV/SDN-aware IoT Systems
Authors	Alejandro Molina Zarca and Jorge Bernal Bernabe and Ruben Trapero and Diego Rivera and Jesus Villalobos and Antonio Skarmeta and Stefano Bianchi and Anastasios Zafeiropoulos and Panagiotis Gouvas
Type	Journal
Journal	IEEE Internet of Things Journal
Impact factor (2018)	9.515
Publisher	IEEE
Pages	8005-8020
Volume	6
Year	2019
Month	March
ISSN	2327-4662
DOI	`https://doi.org/10.1109/JIOT.2019.2904123`
URL	`https://ieeexplore.ieee.org/document/8664092`
State	Published
Author's contribution	The PhD student, Alejandro Molina Zarca, is the main author of the paper

Journal details: IEEE Internet of Things Journal
ISSN: 2327-4662
Publisher: IEEE
Impact factor (2018): 9.515
Website: `https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=6488907`

Authors – Personal details	
Name	**Alejandro Molina Zarca**
Position	PhD student of the Department of Information and Communications Engineering
University	University of Murcia
Name	**Dr. Jorge Bernal Bernabé**
Position	Postdoctoral Researcher of Department of Information and Communications Engineering
University	University of Murcia
Name	**Ruben Trapero**
Position	Researcher
Research Centre	ATOS
Name	**Dr. Diego Rivera**
Position	Postdoctoral Researcher
Research Centre	Montimage
Name	**Jesus Villalobos**
Position	Researcher
Research Centre	ATOS
Name	**Dr. Antonio F. Skarmeta Gómez**
Position	Professor of the Department of Information and Communications Engineering
University	University of Murcia
Name	**Stefano Bianchi**
Position	Project Coordinator
Research Centre	Softeco
Name	**Anastasios Zafeiropoulos**
Position	Researcher
Research Centre	Ubitech
Name	**Anastasios Zafeiropoulos**
Position	Researcher
Research Centre	Ubitech

Abstract

The Internet of Things brings a multi-disciplinary revolution in several application areas. However, security and privacy concerns are undermining a reliable and resilient broadscale deployment of IoT-enabled Critical Infrastructures (IoTCIs). To fill this gap, this paper proposes a comprehensive architectural design that captures the main security and privacy challenges related to Cyber-physical Systems and IoT-CIs. The architecture is devised to empower IoT systems and networks to make autonomous security decisions through the usage of novel technologies such as Software Defined Networking (SDN) and Network Function Virtualization (NFV), as well as endowing them with intelligent and dynamic security reaction capabilities by relying on monitoring methodologies and cyber-situational tools. The architecture has been successfully implemented and evaluated in the scope of ANASTACIA H2020 EU research project.

4.4. Virtual IoT HoneyNets to Mitigate Cyberattacks in SDN/NFV-Enabled IoT Networks

Title	Virtual IoT HoneyNets to Mitigate Cyberattacks in SDN/NFV-Enabled IoT Networks
Authors	Alejandro Molina-Zarca and Jorge Bernal Bernabé Antonio Skarmeta and Jose M. Alcaraz Calero
Type	Journal
Journal	IEEE Journal on Selected Areas in Communications
Impact factor (2018)	9.302
Publisher	IEEE
Pages	1262 - 1277
Volume	38
Issue	6
Year	2020
Month	April
ISSN	0733-8716
DOI	`https://doi.org/10.1109/JSAC.2020.2986621`
URL	`https://ieeexplore.ieee.org/document/9060972`
State	Published
Author's contribution	The PhD student, Alejandro Molina Zarca, is the main author of the paper

Journal details: IEEE Journal on Selected Areas in Communications
ISSN: 0733-8716
Publisher: IEEE
Impact factor (2018): 9.302
Website: `https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=49`

Authors – Personal details	
Name	**Alejandro Molina Zarca**
Position	PhD student of the Department of Information and Communications Engineering
University	University of Murcia
Name	**Dr. Jorge Bernal Bernabé**
Position	Postdoctoral Researcher of Department of Information and Communications Engineering
University	University of Murcia
Name	**Dr. Antonio F. Skarmeta Gómez**
Position	Professor of the Department of Information and Communications Engineering
University	University of Murcia
Name	**Dr. Jose M. Alcaraz Calero**
Position	Ful Professor
University	University of the West of Scotland

Abstract

As the IoT adoption is growing in several fields, cybersecurity attacks involving low-cost end-user devices are increasing accordingly, undermining the expected deployment of IoT solutions in a broad range of scenarios. To address this challenge, emerging Network Function Virtualization (NFV) and Software Defined Networking (SDN) technologies can introduce new security enablers, thereby endowing IoT systems and networks with higher degree of scalability and flexibility required to cope with the security of massive IoT deployments. In this sense, honeynets can be enhanced with SDN and NFV support, to be applied into IoT scenarios thereby strengthening the overall security. IoT honeynets are virtualized services simulating real IoT networks deployments, so that attackers can be distracted from the real target. In this paper, we present a novel mechanism leveraging SDN and NFV aimed to autonomously deploy and enforce IoT honeynets. The system follows a security policybased approach that facilitates management, enforcement and orchestration of the honeynets and it has been successfully implemented and tested in the scope of H2020 EU project ANASTACIA, showing its feasibility to mitigate cyber-attacks.

4.5. Semantic-Aware Security Orchestration in SDN/NFV-Enabled IoT Systems

Title	Semantic-Aware Security Orchestration in SDN/NFV-Enabled IoT Systems
Authors	Alejandro Molina-Zarca and Miloud Bagaa and Jorge Bernal Bernabé and Tarik Taleb and Antonio Skarmeta
Type	Journal
Journal	Sensors
Impact factor (2018)	3.275
Publisher	MDPI
Pages	3622
Volume	20
Issue	13
Year	2020
Month	Jun
ISSN	1424-8220
DOI	`http://dx.doi.org/10.3390/s20133622`
URL	`https://www.mdpi.com/1424-8220/20/13/3622`
State	Published
Author's contribution	The PhD student, Alejandro Molina Zarca, is the main author of the paper

Journal details: Sensors
ISSN: 1424-8220
Publisher: MDPI
Impact factor (2018): 3.275
Website: `https://www.mdpi.com/1424-8220/20/13/3622`

Authors – Personal details	
Name	**Alejandro Molina Zarca**
Position	PhD student of the Department of Information and Communications Engineering
University	University of Murcia
Name	**Dr. Jorge Bernal Bernabé**
Position	Postdoctoral Researcher of Department of Information and Communications Engineering
University	University of Murcia
Name	**Miloud Bagaa**
Position	Postdoctoral researcher
University	Aalto University
Name	**Tarik Taleb**
Position	Full Professor
University	Aalto University
Name	**Dr. Antonio F. Skarmeta Gómez**
Position	Professor of the Department of Information and Communications Engineering
University	University of Murcia

Abstract

IoT systems can be leveraged by Network Function Virtualization (NFV) and Software-Defined Networking (SDN) technologies, thereby strengthening their overall flexibility, security and resilience. In this sense, adaptive and policy-based security frameworks for SDN/NFV-aware IoT systems can provide a remarkable added value for self-protection and self-healing, by orchestrating and enforcing dynamically security policies and associated Virtual Network Functions (VNF) or Virtual network Security Functions (VSF) according to the actual context. However, this security orchestration is subject to multiple possible inconsistencies between the policies to enforce, the already enforced management policies and the evolving status of the managed IoT system. In this regard, this paper presents a semantic-aware, zero-touch and policy-driven security orchestration framework for autonomic and conflict-less security orchestration in SDN/NFV-aware IoT scenarios while ensuring optimal allocation and Service Function Chaining (SFC) of VSF. The framework relies on Semantic technologies and considers the security policies and the evolving IoT system model to dynamically and formally detect any semantic conflict during the orchestration. In addition, our optimized SFC algorithm maximizes the QoS, security aspects and resources usage during VSF allocation. The orchestration security framework has been implemented and validated showing its feasibility and performance to detect the conflicts and optimally enforce the VSFs.

Chapter 5

Bibliography

5.1. References

[1] “Common Information Model (CIM), DMTF.” http://www.dmtf.org/standards/cim.

[2] Y. Gao, Y. Peng, F. Xie, W. Zhao, D. Wang, X. Han, T. Lu, and Z. Li, “Analysis of security threats and vulnerability for cyber-physical systems,” in *Proceedings of 2013 3rd International Conference on Computer Science and Network Technology*, Oct 2013, pp. 50–55.

[3] C. Basile, A. Lioy, C. Pitscheider, F. Valenza, and M. Vallini, “A novel approach for integrating security policy enforcement with dynamic network virtualization,” in *Network Softwarization (NetSoft), 2015 1st IEEE Conference on.* IEEE, 2015, pp. 1–5.

[4] D. Garcia-Carrillo and R. Marin-Lopez, “Lightweight coap-based bootstrapping service for the internet of things,” *Sensors*, vol. 16, no. 3, 2016.

[5] J. Granjal, E. Monteiro, and J. S. Silva, “Security for the internet of things: a survey of existing protocols and open research issues,” *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, pp. 1294–1312, 2015.

[6] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, “Security, privacy and trust in internet of things: The road ahead,” *Computer Networks*, vol. 76, pp. 146–164, 2015.

[7] S. Lal, T. Taleb, and A. Dutta, “Nfv: Security threats and best practices,” *IEEE Communications Magazine*, vol. PP, no. 99, pp. 2–8, 2017.

[8] Y. Choi, “Implementation of content-oriented networking architecture (cona): a focus on ddos countermeasure,” in *Proceedings of European NetFPGA developers workshop*, 2010.

[9] S. T. Ali, V. Sivaraman, A. Radford, and S. Jha, “A survey of securing networks using software defined networking,” *IEEE transactions on reliability*, vol. 64, no. 3, pp. 1086–1097, 2015.

[10] L. Galluccio, S. Milardo, G. Morabito, and S. Palazzo, “Sdn-wise: Design, prototyping and experimentation of a stateful sdn solution for wireless sensor networks,” in *Computer Communications (INFOCOM), 2015 IEEE Conference on.* IEEE, 2015, pp. 513–521.

[11] T. Luo, H.-P. Tan, and T. Q. Quek, "Sensor openflow: Enabling software-defined wireless sensor networks," *IEEE Communications Letters*, vol. 16, no. 11, pp. 1896–1899, 2012.

[12] S. Shin, P. Porras, V. Yegneswaran, M. Fong, G. Gu, and M. Tyson, "Fresco: Modular composable security services for software-defined networks," in *20th Annual Network & Distributed System Security Symposium.* NDSS, 2013.

[13] B. T. De Oliveira, L. B. Gabriel, and C. B. Margi, "Tinysdn: Enabling multiple controllers for software-defined wireless sensor networks," *IEEE Latin America Transactions*, vol. 13, no. 11, pp. 3690–3696, 2015.

[14] ETSI ISG NFV, "Etsi gs nfv-sec 003 nfv; architectural framework v1.2.1," 2014.

[15] T. Yu, V. Sekar, S. Seshan, Y. Agarwal, and C. Xu, "Handling a trillion (unfixable) flaws on a billion devices: Rethinking network security for the internet-of-things," in *Proceedings of the 14th ACM Workshop on Hot Topics in Networks.* ACM, 2015, p. 5.

[16] "SECURity at the network EDge," https://www.secured-fp7.eu/.

[17] J. Bernal Bernabe, J. L. Hernandez Ramos, and A. F. Skarmeta Gomez, "Taciot: multidimensional trust-aware access control system for the internet of things," *Soft Computing*, vol. 20, no. 5, pp. 1763–1779, May 2016. [Online]. Available: https://doi.org/10.1007/s00500-015-1705-6

[18] I. Farris, T. Taleb, Y. Khettab, and J. S. Song, "A survey on emerging sdn and nfv security mechanisms for iot systems," *IEEE Communications Surveys Tutorials*, pp. 1–1, 2018.

[19] P. Salva-Garcia, J. M. Alcaraz-Calero, Q. Wang, J. B. Bernabe, and A. Skarmeta, "5g nb-iot: Efficient network traffic filtering for multitenant iot cellular networks," *Security and Communication Networks*, vol. 2018, no. 9291506, Dec. 2018. [Online]. Available: https://doi.org/10.1155/2018/9291506

[20] L. M. Vaquero, F. Cuadrado, Y. Elkhatib, J. Bernal-Bernabe, S. N. Srirama, and M. F. Zhani, "Research challenges in nextgen service orchestration," *Future Generation Computer Systems*, vol. 90, pp. 20 – 38, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X18303157

[21] J. G. Herrera and J. F. Botero, "Resource allocation in nfv: A comprehensive survey," *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 518–532, Sept 2016.

[22] J. L. Hernandez-Ramos, D. G. Carrillo, R. Marín-López, and A. F. Skarmeta, "Dynamic security credentials pana-based provisioning for iot smart objects," in *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, Dec 2015, pp. 783–788.

[23] J. L. Hernandez-Ramos, A. J. Jara, L. Marin, and A. Skarmeta, "Distributed capability-based access control for the internet of things," vol. 3, pp. 1–16, 01 2013.

[24] J. B. Bernabé, J. M. M. Pérez, J. M. A. Calero, J. D. J. Re, F. J. Clemente, G. M. Pérez, and A. F. Skarmeta, "Security policy specification," in *Network and Traffic Engineering in Emerging Distributed Computing Applications.* IGI Global, 2013, pp. 66–93.

[25] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-level sensor network simulation with cooja," in *Proceedings. 2006 31st IEEE Conference on Local Computer Networks*, Nov 2006, pp. 641–648.

[26] W. Fan, D. Fernández, and V. A. Villagrá, "Technology independent honeynet description language," in *Model-Driven Engineering and Software Development (MODELSWARD), 2015 3rd International Conference on.* IEEE, 2015, pp. 303–311.

[27] S. Scott-Hayward, G. O'Callaghan, and S. Sezer, "Sdn security: A survey," in *2013 IEEE SDN for Future Networks and Services (SDN4FNS)*, Nov 2013, pp. 1–7.

[28] Y. Li and M. Chen, "Software-defined network function virtualization: A survey," *IEEE Access*, vol. 3, pp. 2542–2553, 2015.

[29] I. Farris, J. Bernabe, N. Toumi, D. Garcia-Carrillo, T. Taleb, A. Skarmeta, and B. Sahlin., "Towards Provisioning of SDN/NFV-based Security Enablers for Integrated Protection of IoT Systems," in *IEEE Conference on Standards for Communications and Networking (CSCN-2017)*, 2017.

[30] R. Lopez and G. Lopez-Millan, "Software-Defined Networking (SDN)-based IPsec Flow Protection," Internet Engineering Task Force, Internet-Draft draft-ietf-i2nsf-sdn-ipsec-flow-protection-03, Oct. 2018, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-ietf-i2nsf-sdn-ipsec-flow-protection-03

[31] T. Xu, D. Gao, P. Dong, H. Zhang, C. H. Foh, and H. C. Chao, "Defending against new-flow attack in sdn-based internet of things," *IEEE Access*, vol. 5, pp. 3431–3443, 2017.

[32] S. Ziegler, A. Skarmeta, J. Bernal, E. Kim, and S. Bianchi, "Anastacia: Advanced networked agents for security and trust assessment in cps iot architectures," in *2017 Global Internet of Things Summit (GIoTS)*, June 2017, pp. 1–6.

[33] Q. Yan, F. R. Yu, Q. Gong, and J. Li, "Software-defined networking (sdn) and distributed denial of service (ddos) attacks in cloud computing environments: A survey, some research issues, and challenges," *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 602–622, Firstquarter 2016.

[34] S. Shin, L. Xu, S. Hong, and G. Gu, "Enhancing network security through software defined networking (sdn)," in *2016 25th International Conference on Computer Communication and Networks (ICCCN)*, Aug 2016, pp. 1–9.

[35] G. Selander, J. Mattsson, F. Palombini, and L. Seitz, "Object Security for Constrained RESTful Environments (OSCORE)," Internet Engineering Task Force, Internet-Draft draft-ietf-core-object-security-08, Jan. 2018, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-ietf-core-object-security-08

[36] D. Garcia-Carrillo, R. Marin-Lopez, A. Kandasamy, and A. Pelov, "A coap-based network access authentication service for low-power wide area networks: Lo-coap-eap," *Sensors*, vol. 17, no. 11, 2017. [Online]. Available: http://www.mdpi.com/1424-8220/17/11/2646

[37] "Deserec project: Dependability and security by enhanced reconfigurability," http://www.deserec.eu/.

[38] K. K. Kolluru, C. Paniagua, J. van Deventer, J. Eliasson, J. Delsing, and R. Delong, "An aaa solution for securing industrial iot devices using next generation access control," pp. 737–742, 05 2018.

[39] D. Mehta, A. E.-D. Mady, M. Boubekeur, and D. M. Shila, "Anomaly-based intrusion detection system for embedded devices on internet," in *The Tenth International Conference on Advances in Circuits, Electronics and Micro-electronics*, 2018.

[40] E. Bertino and N. Islam, "Botnets and internet of things security," *Computer*, vol. 50, no. 2, pp. 76–79, Feb. 2017. [Online]. Available: doi.ieeecomputersociety.org/10.1109/MC.2017.62

[41] B. R. Al-Kaseem and H. S. Al-Raweshidyhamed, "Sd-nfv as an energy efficient approach for m2m networks using cloud-based 6lowpan testbed," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1787–1797, Oct 2017.

[42] W. Yang and C. Fung, "A survey on security in network functions virtualization," in *2016 IEEE NetSoft Conference and Workshops (NetSoft)*, June 2016, pp. 15–19.

[43] D. B. Rawat and S. R. Reddy, "Software defined networking architecture, security and energy efficiency: A survey," *IEEE Communications Surveys Tutorials*, vol. 19, no. 1, pp. 325–346, Firstquarter 2017.

[44] F. A. Alaba, M. Othman, I. A. T. Hashem, and F. Alotaibi, "Internet of things security: A survey," *Journal of Network and Computer Applications*, vol. 88, pp. 10 – 28, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804517301455

[45] O. Flauzac, C. González, A. Hachani, and F. Nolot, "Sdn based architecture for iot and improvement of the security," in *2015 IEEE 29th International Conference on Advanced Information Networking and Applications Workshops*, March 2015, pp. 688–693.

[46] J. P. Santos, R. Alheiro, L. Andrade, L. Valdivieso Caraguay, L. I. Barona López, M. A. Sotelo Monge, L. J. Garcia Villalba, W. Jiang, H. Schotten, J. M. Alcaraz-Calero, Q. Wang, and M. J. Barros, "Selfnet framework self-healing capabilities for 5g mobile networks," *Transactions on Emerging Telecommunications Technologies*, vol. 27, no. 9, pp. 1225–1232. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.3049

[47] A. Guerra Manzanares, "Honeyio4: the construction of a virtual, low-interaction iot honeypot," B.S. thesis, Universitat Politècnica de Catalunya, 2017.

[48] P. Bull, R. Austin, E. Popov, M. Sharma, and R. Watson, "Flow based security for iot devices using an sdn gateway," in *2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud)*, Aug 2016, pp. 157–163.

[49] S. Choi and J. Kwak, "Enhanced sdiot security framework models," *International Journal of Distributed Sensor Networks*, vol. 12, no. 5, 2016.

[50] C. Yoon, T. Park, S. Lee, H. Kang, S. Shin, and Z. Zhang, "Enabling security functions with sdn: A feasibility study," *Computer Networks*, vol. 85, pp. 19 – 35, 2015.

[51] C. Shiva Shankar, A. Ranganathan, and R. Campbell, "An eca-p policy-based framework for managing ubiquitous computing environments," 08 2005, pp. 33– 42.

[52] C. Rensing and M. Karsten, "Aaa: a survey and a policy-based architecture and framework," 2002.

[53] A. M. Hadjiantonis, A. Malatras, and G. Pavlou, "A context-aware, policy-based framework for the management of manets," *Seventh IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY'06)*, pp. 10 pp.–34, 2006.

[54] A. L. Shaw, L. Jacquin, A. Lioy, C. Pitscheider, C. Basile, F. Risso, R. Bonafiglia, F. Ciacca, M. Nemirovsky, J. Kuusijärvi, D. Montero, R. Serral-Gracià, M. Yannuzzi, and F. Bosco, "Specification of the secured architecture (alpha version)," Tech. Rep.

[55] ——, "Policy specification)," Tech. Rep.

[56] V. Casola, A. De Benedictis, A. Riccio, D. Rivera, W. Mallouli, and E. M. de Oca, "A security monitoring system for internet of things," *Internet of Things*, vol. 7, p. 100080, 2019.

[57] J. B. Bernabe and A. Skarmeta, "Introducing the Challenges in Cybersecurity and Privacy - The European Research Landscape," in *Challenges in Cybersecurity and Privacy - the European Research Landscape*, ser. RIVER PUBLISHERS SERIES IN SECURITY AND DIGITAL FORENSICS, J. B. Bernabe and A. Skarmeta, Eds. River Publishers, 7 2019, pp. 1–21. [Online]. Available: https://doi.org/10.13052/rp-9788770220873

[58] S. Ziegler, C. Crettaz, E. Kim, A. Skarmeta, J. B. Bernabe, R. Trapero, and S. Bianchi, *Privacy and Security Threats on the Internet of Things.* Cham: Springer International Publishing, 2019, pp. 9–43. [Online]. Available: https://doi.org/10.1007/978-3-030-04984-3_2

[59] M. Ahmad, T. Younis, M. A. Habib, R. Ashraf, and S. H. Ahmed, "A review of current security issues in internet of things," *Recent Trends and Advances in Wireless and IoT-enabled Networks*, p. 11, 2019.

[60] K. Chen, S. Zhang, Z. Li, Y. Zhang, Q. Deng, S. Ray, and Y. Jin, "Internet-of-things security and vulnerabilities: taxonomy, challenges, and practice," *Journal of Hardware and Systems Security*, vol. 2, no. 2, pp. 97–110, 2018.

[61] C. Vorakulpipat, E. Rattanalerdnusorn, P. Thaenkaew, and H. D. Hai, "Recent challenges, trends, and concerns related to iot security: An evolutionary study," in *2018 20th International Conference on Advanced Communication Technology (ICACT).* IEEE, 2018, pp. 405–410.

[62] S. Do, L. V. Le, B. S. P. Lin, and L.-P. Tung, "Sdn/nfv based internet of things for multi-tenant networks," *Transactions on Networks and Communications*, vol. 6, no. 6, pp. 40–40, 2018.

[63] Á. L. V. Caraguay, P. L. González, R. T. Tandazo, and L. I. B. López, "Sdn/nfv architecture for iot networks." in *WEBIST*, 2018, pp. 425–429.

[64] D. Sinh, L.-V. Le, B.-S. P. Lin, and L.-P. Tung, "Sdn/nfv—a new approach of deploying network infrastructure for iot," in *2018 27th Wireless and Optical Communication Conference (WOCC).* IEEE, 2018, pp. 1–5.

[65] S. Do, L.-V. Le, B.-S. P. Lin, and L.-P. Tung, "Sdn/nfv-based network infrastructure for enhancing iot gateways," in *2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData).* IEEE, 2019, pp. 1135–1142.

[66] H. Lin, "Sdn-based in-network honeypot: Preemptively disrupt and mislead attacks in iot networks," *arXiv preprint arXiv:1905.13254*, 2019.

[67] A. D. Oza, G. N. Kumar, and M. Khorajiya, "Survey of snaring cyber attacks on iot devices with honeypots and honeynets," in *2018 3rd International Conference for Convergence in Technology (I2CT).* IEEE, 2018, pp. 1–6.

[68] M. Banerjee and S. Samantaray, "Network traffic analysis based iot botnet detection using honeynet data applying classification techniques," *International Journal of Computer Science and Information Security (IJCSIS)*, vol. 17, no. 8, 2019.

[69] A. Molina Zarca, J. B. Bernabe, R. Trapero, D. Rivera, J. Villalobos, A. Skarmeta, S. Bianchi, A. Zafeiropoulos, and P. Gouvas, "Security management architecture for nfv/sdn-aware iot systems," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8005–8020, Oct 2019.

[70] W. Fan and D. Fernández, "A novel sdn based stealthy tcp connection handover mechanism for hybrid honeypot systems," in *2017 IEEE Conference on Network Softwarization (NetSoft).* IEEE, 2017, pp. 1–9.

[71] W. Fan, D. Fernández, and Z. Du, "Versatile virtual honeynet management framework," *IET Information Security*, vol. 11, no. 1, pp. 38–45, 2016.

[72] A. Hamza, H. H. Gharakheili, and V. Sivaraman, "Combining mud policies with sdn for iot intrusion detection," in *Proceedings of the 2018 Workshop on IoT Security and Privacy*, 2018, pp. 1–7.

[73] A. Hamza, H. H. Gharakheili, T. A. Benson, and V. Sivaraman, "Detecting volumetric attacks on lot devices via sdn-based monitoring of mud activity," in *Proceedings of the 2019 ACM Symposium on SDN Research*, 2019, pp. 36–48.

[74] M. Ranganathan, D. Montgomery, and O. El Mimouni, "Implementing manufacturer usage descriptions on openflow sdn switches."

[75] D. Comer and A. Rastegatnia, "Osdf: An intent-based software defined network programming framework," in *2018 IEEE 43rd Conference on Local Computer Networks (LCN)*. IEEE, 2018, pp. 527–535.

[76] A. Boudi, I. Farris, M. Bagaa, and T. Taleb, "Assessing lightweight virtualization for security-as-a-service at the network edge," *IEICE Transactions on Communications*, vol. 102, no. 5, pp. 970–977, 2019.

[77] Lear, Droms, and Romascanu, "Manufacturer Usage Description Specification," RFC 8520, 2019. [Online]. Available: https://www.rfc-editor.org/rfc/rfc8520.txt

[78] J. B. Bernabe, J. M. M. Perez, J. M. A. Calero, F. J. G. Clemente, G. M. Perez, and A. F. G. Skarmeta, "Semantic-aware multi-tenancy authorization system for cloud architectures," *Future Generation Computer Systems*, vol. 32, pp. 154–167, 2014.

[79] S. Sicari, A. Rizzardi, D. Miorandi, C. Cappiello, and A. Coen-Porisini, "Security policy enforcement for networked smart objects," *Computer Networks*, vol. 108, pp. 133 – 147, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128616302663

[80] C. Basile, F. Valenza, A. Lioy, D. R. Lopez, and A. Pastor Perales, "Adding support for automatic enforcement of security policies in nfv networks," *IEEE/ACM Transactions on Networking*, vol. 27, no. 2, pp. 707–720, 2019.

[81] F. Valenza, T. Su, S. Spinoso, A. Lioy, R. Sisto, and M. Vallini, "A formal approach for network security policy validation," *JoWUA*, vol. 8, pp. 79–100, 2017.

[82] C. Basile, D. Canavese, A. Lioy, C. Pitscheider, and F. Valenza, "Inter-function anomaly analysis for correct sdn/nfv deployment," *International Journal of Network Management*, vol. 26, no. 1, pp. 25–43, 2016. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/nem.1917

[83] O. Standard, "extensible access control markup language (xacml) version 3.0," 2013.

[84] B. Moore, E. Ellesson, J. Strassner, and A. Westerinen, "Policy core information model–version 1 specification," RFC 3060, February, Tech. Rep., 2001.

[85] D. Barrera, I. Molloy, and H. Huang, "Standardizing iot network security policy enforcement," in *Workshop on Decentralized IoT Security and Standards (DISS)*, vol. 2018, 2018, p. 6.

[86] V. Jethanandani *et al.*, "Yang data model for network access control lists (acls)," RFC 8519, March, Tech. Rep., 2019. [Online]. Available: https://tools.ietf.org/html/rfc8519

[87] A. Uszok, J. M. Bradshaw, M. Johnson, R. Jeffers, A. Tate, J. Dalton, and S. Aitken, "Kaos policy management for semantic web services," *IEEE Intelligent Systems*, vol. 19, no. 4, pp. 32–41, 2004.

[88] F. Valenza and A. Lioy, "User-oriented network security policy specification." *J. Internet Serv. Inf. Secur.*, vol. 8, no. 2, pp. 33–47, 2018.

[89] J. Strassner and S. Schleimer, "Policy framework definition language," *draft-ietf-policy-framework-pfdl-00.txt*, 1998.

[90] E. Bertino, A. Mileo, and A. Provetti, "Pdl with preferences," in *Sixth IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY'05)*. IEEE, 2005, pp. 213–222.

[91] C. DMTF, "Simplified policy language (cim-spl)," *Document Number DSP0231, version*, vol. 1.

[92] P. Ashley, S. Hada, G. Karjoth, C. Powers, and M. Schunter, "Enterprise privacy authorization language (epal)," *IBM Research*, vol. 30, p. 31, 2003.

[93] S. Bleikertz and T. Groß, "A virtualization assurance language for isolation and deployment," in *2011 IEEE International Symposium on Policies for Distributed Systems and Networks*. IEEE, 2011, pp. 33–40.

[94] P. Ashley, S. Hada, G. Karjoth, and M. Schunter, "E-p3p privacy policies and privacy authorization," in *Proceedings of the 2002 ACM workshop on Privacy in the Electronic Society*, 2002, pp. 103–109.

[95] M. Mohsin and M. U. Khan, "Uml-sr: A novel security requirements specification language," in *2019 IEEE 19th International Conference on Software Quality, Reliability and Security (QRS)*. IEEE, 2019, pp. 342–349.

[96] N. Damianou, N. Dulay, E. C. Lupu, and M. Sloman, "Ponder: A language for specifying security and management policies for distributed systems," 2000.

[97] R. Kumar, A. Rensink, and M. Stoelinga, "Locks: a property specification language for security goals," in *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, 2018, pp. 1907–1915.

[98] S.-W. Cheng and D. Garlan, "Stitch: A language for architecture-based self-adaptation," *Journal of Systems and Software*, vol. 85, no. 12, pp. 2860–2875, 2012.

[99] O. W. Group, "Owl web ontology language overview." [Online]. Available: https://www.w3.org/OWL/

[100] A.-R. Sadeghi, C. Wachsmann, and M. Waidner, "Security and privacy challenges in industrial internet of things," in *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*. IEEE, 2015, pp. 1–6.

[101] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "Ddos in the iot: Mirai and other botnets," *Computer*, vol. 50, no. 7, pp. 80–84, 2017.

[102] M. N.-B. A. U. Manufacturer, "Securing small-business and home internet of things (iot) devices," *NIST SPECIAL PUBLICATION*, p. 15A, 1800.

[103] W. Polk, M. Souppaya, and W. Barker, "[project description] mitigating iot-based automated distributed threats (draft)," National Institute of Standards and Technology, Tech. Rep., 2017.

[104] B. Sarikaya, M. Sethi, and D. Garcia-Carillo, "Secure iot bootstrapping: A survey," *Internet Engineering Task Force*, 2018.

[105] M. Kanda, Y. Ohba, S. Das, and S. Chasko, "Pana applicability in constrained environments," in *Smart Object Security Wksp.*, 2012.

[106] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki-a lightweight and flexible operating system for tiny networked sensors," in *29th annual IEEE international conference on local computer networks*. IEEE, 2004, pp. 455–462.

[107] M. Al-Shaboti, I. Welch, A. Chen, and M. A. Mahmood, “Towards secure smart home iot: Manufacturer and user network access control framework,” in *2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA).* IEEE, 2018, pp. 892–899.

[108] S. Pérez, J. L. Hernández-Ramos, S. N. Matheu-García, D. Rotondi, A. F. Skarmeta, L. Straniero, and D. Pedone, “A lightweight and flexible encryption scheme to protect sensitive data in smart building scenarios,” *IEEE Access*, vol. 6, pp. 11 738–11 750, 2018.

[109] O. Garcia-Morchon, S. Kumar, and M. Sethi, “Internet of things security: State of the art and challenges (rfc 8576), 2019,” *Internet Engineering Task Force*, 2019.

[110] J. L. Hernández-Ramos, A. J. Jara, L. Marín, and A. F. Skarmeta Gómez, “Dcapbac: embedding authorization logic into smart things through ecc optimizations,” *International Journal of Computer Mathematics*, vol. 93, no. 2, pp. 345–366, 2016.

[111] J. Galeano-Brajones, J. Carmona-Murillo, J. F. Valenzuela-Valdés, and F. Luna-Valero, “Detection and mitigation of dos and ddos attacks in iot-based stateful sdn: An experimental approach,” *Sensors*, vol. 20, no. 3, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/3/816

[112] M. Bagaa, T. Taleb, J. B. Bernabe, and A. Skarmeta, “A machine learning security framework for iot systems,” *IEEE Access*, pp. 1–1, 2020.

[113] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, “A survey on iot security: Application areas, security threats, and solution architectures,” *IEEE Access*, vol. 7, pp. 82 721–82 743, 2019.

[114] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, “System architecture directions for networked sensors,” *ACM Sigplan notices*, vol. 35, no. 11, pp. 93–104, 2000.

[115] L. Xia, J. Strassner, C. Basile, and D. Lopez, “Information Model of NSFs Capabilities,” IETF, Internet-Draft draft-ietf-i2nsf-capability-00, 2017, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-ietf-i2nsf-capability-00

[116] ——, “Information Model of NSFs Capabilities,” IETF, Internet-Draft draft-ietf-i2nsf-capability-05, 2019, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-ietf-i2nsf-capability-05

[117] M. Sethi, B. Sarikaya, and D. Garcia-Carrillo, “Secure IoT Bootstrapping: A Survey,” IETF, Internet-Draft draft-sarikaya-t2trg-sbootstrapping-08, 2020, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-sarikaya-t2trg-sbootstrapping-08

[118] A. M. Zarca, J. Bernal, J. Ortiz, and A. Skarmeta, “Policy-based Definition and Policy for Orchestration Initial Report,” University of Murcia, EU Project Deliverable 2.1, 2017. [Online]. Available: http://www.anastacia-h2020.eu/deliverables/ANASTACIA-WP2-T2.1-UMU-D2.1-PolicyBasedDefinitionAndPolicyForOrchestrationInitialReport-v1.0.pdf

[119] ——, “Policy-based Definition and Policy for Orchestration Final Report,” University of Murcia, EU Project Deliverable 2.5, 2018. [Online]. Available: http://www.anastacia-h2020.eu/deliverables/ANASTACIA-WP2-T2.1-UMU-D2.5-PolicyBasedDefinitionAndPolicyForOrchestrationFinalReport-v1.0.pdf

[120] ——, “Final Security Enforcement Manager Report,” University of Murcia, EU Project Deliverable 3.4, 2019. [Online]. Available: http://www.anastacia-h2020.eu/deliverables/ANASTACIA-WP3-UMU-T3.1-D3.4-FinalSecurityEnforcementManagerReport-v1.0.pdf

[121] D. Mehta, A. E.-D. Mady, M. Boubekeur, and D. M. Shila, "Anomaly-based intrusion detection system for embedded devices on internet," in *Proceedings of the Tenth International Conference on Advances in Circuits, Electronics and Micro-electronics, Venice, Italy*, 2018, pp. 16–20.

5.2. Publications

[122] A. Molina Zarca, J. Bernal Bernabe, I. Farris, Y. Khettab, T. Taleb, and A. Skarmeta, "Enhancing iot security through network softwarization and virtual security appliances," *International Journal of Network Management*, vol. 28, no. 5, p. e2038, 2018, e2038 nem.2038. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/nem.2038

[123] A. M. Zarca, D. Garcia-Carrillo, J. B. Bernabe, J. Ortiz, R. Marin-Perez, and A. Skarmeta, "Managing aaa in nfv/sdn-enabled iot scenarios," in *2018 Global Internet of Things Summit (GIoTS)*, 2018, pp. 1–7.

[124] A. Molina Zarca, D. Garcia-Carrillo, J. Bernal Bernabe, J. Ortiz, R. Marin-Perez, and A. Skarmeta, "Enabling virtual aaa management in sdn-based iot networks †," *Sensors*, vol. 19, no. 2, p. 295, Jan 2019. [Online]. Available: http://dx.doi.org/10.3390/s19020295

[125] A. Molina Zarca, J. B. Bernabe, R. Trapero, D. Rivera, J. Villalobos, A. Skarmeta, S. Bianchi, A. Zafeiropoulos, and P. Gouvas, "Security management architecture for nfv/sdn-aware iot systems," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8005–8020, 2019.

[126] J. Bernal, A. Molina, A. Skarmeta, S. Bianchi, E. Cambiaso, I. Vaccari, S. Scaglione, M. Aiello, R. Trapero, M. Bouet, D. Belabed, M. Bagaa, R. Addad, T. Taleb, D. Rivera, A. E.-D. M. andAdrian Quesada Rodriguez, C. Crettaz9, S. Ziegler, E. Kim, M. Filipponi, B. Bajic, D. Garcia-Carrillo, and R. Marin-Perez, "Key innovations in anastacia: Advanced networked agents for security and trust assessment in cps/iot architectures," in *Challenges in Cybersecurity and Privacy: The European Research Landscape*, J. B. Bernabe and A. Skarmeta, Eds. Denmark: River Publishers, 2019, ch. 2, pp. 23–53.

[127] S. N. M. García, A. Molina Zarca, J. L. Hernández-Ramos, J. B. Bernabé, and A. S. Gómez, "Enforcing behavioral profiles through software-defined networks in the industrial internet of things," *Applied Sciences*, vol. 9, no. 21, p. 4576, Oct 2019. [Online]. Available: http://dx.doi.org/10.3390/app9214576

[128] S. N. Matheu, A. Robles Enciso, A. Molina Zarca, D. Garcia-Carrillo, J. L. Hernández-Ramos, J. Bernal Bernabe, and A. F. Skarmeta, "Security architecture for defining and enforcing security profiles in dlt/sdn-based iot systems," *Sensors*, vol. 20, no. 7, p. 1882, Mar 2020. [Online]. Available: http://dx.doi.org/10.3390/s20071882

[129] A. M. Zarca, J. B. Bernabe, A. Skarmeta, and J. M. Alcaraz Calero, "Virtual iot honeynets to mitigate cyberattacks in sdn/nfv-enabled iot networks," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 6, pp. 1262–1277, April 2020.

[130] A. Molina Zarca, M. Bagaa, J. Bernal Bernabe, T. Taleb, and A. F. Skarmeta, "Semantic-aware security orchestration in sdn/nfv-enabled iot systems," *Sensors*, vol. 20, no. 13, p. 3622, Jun 2020. [Online]. Available: http://dx.doi.org/10.3390/s20133622

[131] A. Hermosilla, A. M. Zarca, J. B. Bernabe, J. Ortiz, and A. Skarmeta, "Security orchestration and enforcement in nfv/sdn-aware uav deployments," *IEEE Access*, 2020.

[132] J. Gallego-Madrid, A. Molina-Zarca, R. Sanchez-Iborra, J. Bernal-Bernabe, J. Santa, P. M. Ruiz, and A. F. Skarmeta-Gómez, "Enhancing extensive and remote lora deployments through mec-powered drone gateways," *Sensors 20(15):4109*, 2020.

5.3. Submitted

[133] D. Bringhentia, J. Yusupova, A. M. Zarca, F. Valenzaa, R. Sisto, J. B. Bernabe, and A. Skarmeta, "Autonomic, verifiable and optimized policy-based security enforcement for sdn-aware iot networks," *Elsevier Computer Networks*, in press.