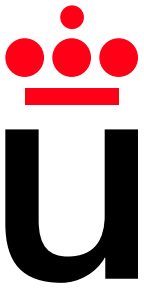


Universidad
Rey Juan Carlos

Escuela Técnica Superior
de Ingeniería Informática

Grado en Ingeniería de la Ciberseguridad

Curso 2021-2022

Trabajo Fin de Grado

SEGURIDAD ADAPTATIVA PARA APLICACIONES EN LA NUBE

Autor: Javier Sánchez García-Ochoa
Tutora: Marta Beltrán Pardo
Co-tutor: Miguel Calvo Matalobos

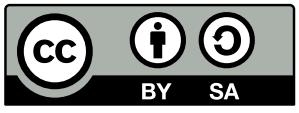
SEGURIDAD ADAPTATIVA PARA APLICACIONES EN LA NUBE

Autor: Javier Sánchez García-Ochoa

Tutora: Marta Beltrán Pardo
Co-tutor: Miguel Calvo Matalobos

Escuela Técnica Superior de Ingeniería Informática
Universidad Rey Juan Carlos

Julio de 2022



Agradecimientos

Gracias papá y mamá por enseñarme tanto, por vuestro esfuerzo y apoyo incondicional a pesar de no despegarme del ordenador y por quererme y confiar en mí en todo momento. A mis hermanos, abuelos y toda mi familia, que siempre está preocupada y orgullosa. A Giovanna, por estar a mi lado y alentarme cuando no me quedaban fuerzas. A Berme, Rubén, Alex y Andrea, por ser los mejores compañeros de aventuras, y a Miki y Lucía, por las jornadas de trabajo y estudio interminables.

Y como no, a Marta y Miguel, por toda la confianza depositada en mí, por tutorizarme con este trabajo y a lo largo de toda la carrera, por descubrirme una salida profesional y por sacar adelante este Grado con tanto esfuerzo y dedicación.

Resumen

La computación en la nube es un paradigma que se está implantando en empresas y organizaciones de todo tipo a un ritmo de adopción vertiginoso. Su flexibilidad y velocidad al aprovisionar recursos de manera casi instantánea, su alta escalabilidad, su fiabilidad o el modelo de pago por uso encamina a muchos clientes a trasladar sus cargas de trabajo a infraestructuras de nube pública. Sin embargo, las características de este modelo también originan amenazas de ciberseguridad, por lo que se deben desarrollar nuevas estrategias de seguridad que las hagan frente.

En esta línea de investigación surgen las medidas de seguridad dinámica o adaptativa, que permiten a los sistemas modificar su arquitectura, comportamiento o configuración en función de la variación del contexto. Una innovadora y reciente propuesta ideada como sistema de seguridad basada en riesgo es RiAS, un modelo genérico capaz de adaptarse a cualquier entorno actual, ya sea *Cloud*, *Internet-of-Things* (IoT), ciudades inteligentes, etc. Para lograr su propósito, obtiene diversa información del contexto de operaciones del sistema para generar un conocimiento, en base al cual se toman decisiones sobre si se debe realizar cualquier adaptación o no, para terminar efectuándola en caso afirmativo. Además, define una estructura de actores que interactúan con el modelo, con unas responsabilidades delimitadas.

En este trabajo se realiza una propuesta de seguridad, utilizando RiAS como modelo, para un entorno de comercio electrónico desplegado en la nube pública de Amazon Web Services (AWS), sobre el cual se tratan de mitigar diversos riesgos de seguridad. En este proceso, se ha procurado aprovechar las ventajas de los entornos en la nube también para el diseño y despliegue de RiAS.

Tras completar el despliegue de la herramienta y comprobar su correcto funcionamiento, se han realizado diferentes experimentos que confirman todas las ventajas de RiAS y los modelos de seguridad adaptativa para entornos *Cloud*, como su escalabilidad o el escaso periodo de tiempo empleado en aplicar las medidas. Además, se ha conseguido implementar la propuesta sin apenas exceder la capa gratuita que AWS pone a disposición de sus clientes y sin sacrificar por ello capacidades, resaltando el bajo coste de la solución.

Palabras clave:

- Seguridad adaptativa
- Seguridad basada en riesgo
- Computación en la nube
- Amazon Web Services

Índice de contenidos

Índice de tablas

Índice de figuras

Índice de códigos

Índice de algoritmos

1 Introducción

1.1. Contexto del Trabajo Fin de Grado

En los últimos años, la migración de cargas de trabajo a infraestructuras de nube pública ha sufrido un aumento considerable, especialmente en grandes corporaciones, aunque también en pequeñas empresas y *startups* [1]. Su escalabilidad, el modelo de pago por uso, la presencia global y el incremento de personal técnico cualificado hace que cada vez sean más las organizaciones que optan por este modelo.

Este cambio de paradigma va ineludiblemente acompañado de nuevas amenazas y ciberincidentes que se deben afrontar. Sin embargo, en muchas ocasiones, se analiza la posición de seguridad de igual manera que en entornos tradicionales, sin evaluar todas aquellas nuevas características de la computación en la nube y los riesgos asociados a las mismas. Para evitarlo, se están realizando trabajos de investigación y desarrollando medidas de seguridad que sean efectivas frente a estos nuevos desafíos.

Este trabajo, en concreto, estudia una novedosa propuesta, *Risk-based Adaptive Security* (RiAS) [2], capaz de reaccionar ante cambios en el contexto de operaciones y modificar la postura de seguridad del sistema que protege de forma autónoma. Asimismo, está diseñada para actuar en cualquier entorno de operación, bien sea computación en la nube, IoT, servidores tradicionales, redes móviles, redes inteligentes, etc. En este trabajo se tratará de explorar una posible implementación de dicha solución en un entorno de comercio electrónico desple-

gado en *Amazon Web Services* (AWS), la nube pública de Amazon, que es el proveedor líder en la industria [3].

1.2. Objetivos

1.2.1. Objetivos generales

El principal objetivo de este trabajo es analizar y definir medidas de seguridad adecuadas para entornos de computación en la nube, mediante el desarrollo y adaptación de RiAS, una propuesta de seguridad adaptativa basada en riesgo, a un caso de uso real de una tienda de comercio electrónico, de forma totalmente integrada haciendo uso de servicios *Cloud* tanto para la plataforma del comercio como la solución de seguridad. Para comprobar su eficacia y validar su efectividad frente a una gran cantidad de amenazas, se simularán escenarios de riesgo y se realizarán distintas mediciones, a fin de poder realizar comparaciones.

1.2.2. Objetivos específicos

En primer lugar, se tratará de adquirir un conocimiento previo sobre las características de los entornos de computación en la nube, su estado de seguridad actual, las necesidades y requisitos demandados para definir medidas de seguridad de calidad, y el valor que aportan los controles más comúnmente utilizados actualmente.

Una vez logradas estas competencias previas, se continuará con la descripción del entorno de pruebas y los escenarios de riesgo que se contemplarán, así como la propia implementación de la propuesta de seguridad adaptativa. Para esto último, se procederá con un estudio que fundamente la elección de los servicios *Cloud* a utilizar, y se tratará de integrar todas las herramientas utilizando modelos de despliegue específicos de la computación en la nube, en caso de haberlos.

1.3. Planificación del trabajo

El trabajo desempeñado puede ser categorizado en tres fases diferentes pero dependientes, aunque no necesariamente debe haberse completado una para poder comenzar con la siguiente. La primera de ellas es el estudio de la arquitectura, incluyendo un análisis del trabajo previo y de la situación actual en relación con la seguridad adaptativa en entornos de computación en la nube. La segunda incluye el diseño de la solución, planteando una tienda de comercio electrónico

como entorno de operaciones y tratando de aplicar el modelo RiAS al mismo. La tercera y última consiste en la implementación en la nube pública AWS de dicha propuesta, incluyendo una parte de medición y validación que aporta información para extraer tanto ventajas como posibles deficiencias de la solución. La Figura 1.1 muestra un detalle de las tareas concretas desempeñadas a lo largo del trabajo.

1.4. Estructura del documento

El resto de este documento está estructurado en diferentes capítulos que reflejan las tres fases de trabajo descritas anteriormente, cubriendo desde el análisis teórico y el estudio del estado del arte, hasta la implementación de la propuesta y su validación, pasando previamente por la fase de diseño de la misma.

El capítulo dos incluye una introducción a la gestión de la seguridad en entornos de computación en la nube, así como un detalle de los motivos que marcan la necesidad de desarrollar e implementar nuevas medidas de seguridad diferentes a las de entornos tradicionales, y los desafíos aún por resolver. Asimismo, recoge también un análisis de las prácticas recomendadas para mitigar riesgos en este tipo de entornos, y una breve introducción a la seguridad dinámica o adaptativa.

En el capítulo tres se explora la arquitectura de un entorno de comercio electrónico, así como las medidas de seguridad aplicables al mismo más comúnmente empleadas. Sobre dicho medio, se realiza una propuesta de seguridad dinámica basada en riesgo, desarrollada a través de dos escenarios de ejemplo, sobre los que realizar las adaptaciones en las medidas de seguridad previamente analizadas e incorporadas al entorno.

El capítulo cuatro comprende el despliegue de la tienda de comercio electrónico en la nube y el diseño, implementación, validación y evaluación de la solución de seguridad adaptativa. La propuesta detalla todos los pasos a seguir para desplegar el modelo escogido, RiAS, sobre el entorno anteriormente definido, así como las decisiones y detalles de la implementación que se han seguido para definir una solución de seguridad completamente funcional e integrada. Tras dicha fase de implementación, se ha realizado una fase de validación y evaluación, que ha arrojado información acerca del desempeño de la propuesta y su posible despliegue en producción.

Por último, en el capítulo cinco se exponen las conclusiones tanto generales como específicas extraídas de este trabajo y del funcionamiento de la propuesta, así como posibles mejoras y líneas de trabajo futuro.

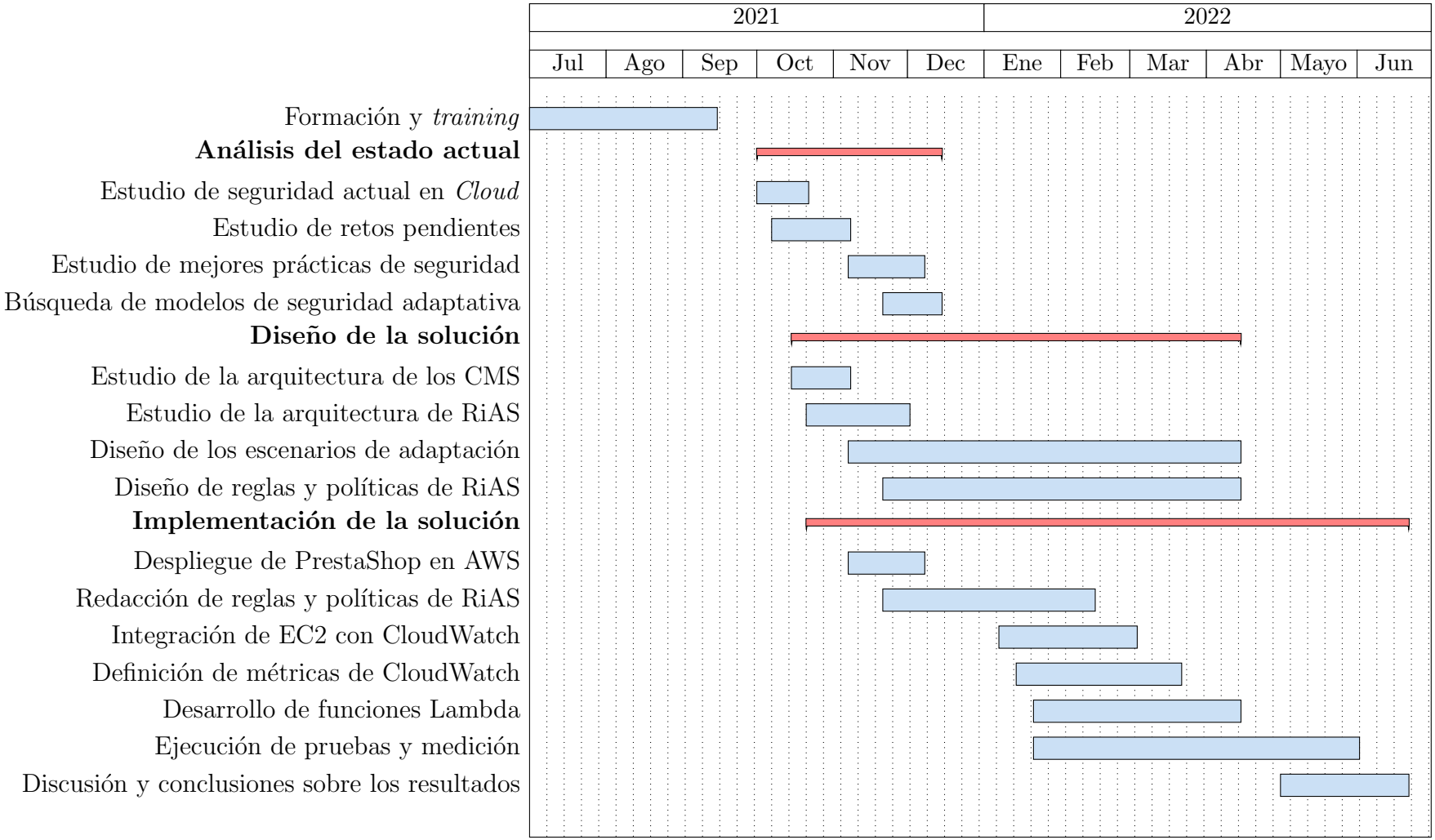


Figura 1.1: Planificación del trabajo

2

Estado del arte

En este capítulo se analizan las características más importantes de los modelos de seguridad en entornos de computación en la nube, así como sus diferencias con la seguridad en entornos tradicionales y los motivos que propician la aparición de nuevas estrategias de ciberseguridad. Para hacer frente a este cambio y a los desafíos que plantea, se mencionan algunas recomendaciones y guías que proporcionan diferentes organizaciones relevantes del sector, y se investigan varias soluciones de seguridad adaptativa, de las cuales una será desarrollada en capítulos posteriores.

2.1. Seguridad en *Cloud Computing*

La computación en la nube se está convirtiendo en un paradigma popular entre empresas y organizaciones de todo tipo gracias al rendimiento mejorado, la calidad de sus servicios de computación y los costes reducidos que ofrecen [4]. Esta implantación y gran acogida va acompañada consecuentemente de medidas de seguridad sobre los nuevos activos. Además, la amplia variedad de clientes y de su actividad y sus modelos de negocio, requiere una enorme cantidad de productos y soluciones diferentes capaces de satisfacer cada necesidad que se plantee.

Todo esto supone un gran reto para los proveedores de servicios *Cloud* (*Cloud Service Providers*, CSPs), que deben ser capaces de cubrir todos los requisitos que necesitan sus clientes y proveerles de productos y modalidades de uso lo más adaptado posible a estos. Pero también es un reto para el personal encargado de

la seguridad de todos los sistemas, tanto del lado del cliente como del proveedor. La amplia variedad en la oferta y sus infinitas configuraciones y combinaciones posibles están causando un aumento de vulnerabilidades en la nube que pueden permitir el acceso de atacantes a las infraestructuras. Ya en 2019, *Symantec* alertaba del peligro de configurar mal los productos, que fue la causa de la fuga de hasta 70 millones de registros de contenedores S3 de *Amazon Web Services* (AWS) en 2018 [5], mientras que *IBM* confirma un aumento de vulnerabilidades de hasta un 150 % en los últimos 5 años, superando las 2000 en 2021 [6]. También es claro el aumento en los ataques dirigidos a las infraestructuras *Cloud*: *AWS*, en el primer trimestre de 2020, detectó un 23 % más de ataques de denegación de servicio (*Denial-of-Service*, DoS) que en el mismo periodo del año anterior, y 0.4 millones más (un 57 %) de amenazas de *malware* que en el último trimestre de 2019 [7].

A disposición de sus clientes, los CSPs ponen una amplia variedad de productos para implantar controles y medidas defensivas que protejan sus activos. Estos deben conocer a la perfección toda la gama de soluciones de seguridad que se les ofrece para poder escoger la óptima en cada caso, así como ser capaces de desplegarla, configurarla para obtener el mayor rendimiento y protección y mantenerla adecuadamente.

2.1.1. Diferencias con los entornos tradicionales

Entre entornos tradicionales y Cloud, aunque hay algunas similitudes, surgen muchas diferencias que hay que tener en cuenta en el momento de plantear la seguridad de los sistemas.

En situaciones en que una organización tiene sus propios servidores en un edificio propio o gestionado por un tercero, esta puede implementar con facilidad sistemas de seguridad físicos sobre los equipos. Sin embargo, se han dado cuenta de que esto es muy costoso y rígido, ya que la gestión (instalación, actualización, retirada de equipos inservibles u obsoletos, etc) supone un gasto que se minimiza al optar por la nube.

En modelos *Cloud* existen múltiples opciones para decidir la localización geográfica de los datos. Acercarlos a los clientes puede ser muy beneficioso para evitar retardos y tiempos de espera en los accesos, mejorando la experiencia y satisfacción de los mismos con la organización. Sin embargo, esto influye en los reglamentos y aplicaciones legales que se deben cumplir, y que en ocasiones pueden ser diferentes a las aplicables en la localización de la organización o diferentes entre las distintas zonas geográficas contratadas, si es el caso.

La infraestructura *Cloud* es mucho más dinámica y flexible, ya que permite aprovisionar servicios bajo demanda en cuestión de minutos y a miles de kilómetros de los centros de datos. Este cambio puede ser muy positivo en cuanto a

costes y disponibilidad de los sistemas, pudiendo contratar únicamente los productos y recursos necesarios en cada situación en función de la demanda. Además, esta ventaja es muy valorada por las empresas y su personal financiero, puesto que únicamente se paga por uso, pasando de un modelo CAPEX (*Capital Expenditure*), con el que se deben realizar inversiones iniciales y pagar gastos fijos por la compra y adquisición de bienes, a OPEX (*Operational Expenditure*), con el que se puede predecir el gasto y pagar sólo por lo necesario en cada momento [8, 9].

2.1.2. Por qué no vale la seguridad tradicional

En los entornos *Cloud* se mantienen cuestiones de seguridad presentes también en entornos tradicionales, relativas por ejemplo a comunicaciones, redes, privacidad, seguridad de las aplicaciones y/o de servicios web, etc. [10]. Sin embargo, hay características de la nube que propician la aparición de nuevos aspectos a tener en cuenta.

Dadas las características básicas sobre la propiedad de los recursos físicos en entornos Cloud, es sencillo comprender que el encargado de la seguridad física en este caso es el proveedor, liberando al cliente de unas medidas que en entornos tradicionales sí que debía aplicar. No obstante, esto puede suponer un problema para el cliente, puesto que una vez salvaguarda los datos en un almacenamiento remoto, pierde el control sobre ellos, que pasan a estar en manos del CSP. Hay usuarios que pueden descuidarse con este traspaso y obviar detalles de las políticas de seguridad del proveedor, ignorando además posibles problemas de seguridad como vulnerabilidades o *malware* que puedan utilizarse para lograr acceso a sus datos a través del CSP, como por ejemplo ataques a nivel de *kernel* [10].

Por otro lado, en las infraestructuras tradicionales es muy sencillo identificar y, por consecuente, proteger el perímetro de los sistemas de la organización. En la nube todo está altamente conectado y la infraestructura es muy dinámica y cambiante [9]. Esto es causado por la virtualización, la capa de software encargada de agrupar y asignar los recursos físicos necesarios, proporcionando a los clientes unos recursos virtuales. La virtualización puede causar que unos mismos recursos físicos estén divididos y se encuentren distintos clientes trabajando sobre ellos simultáneamente, conocido como *multi-tenancy*. Además, cabe destacar que en diversas infraestructuras *Cloud* los recursos físicos asignados a cada cliente varían entre las distintas ejecuciones de su software, modificándose en función de diversos factores determinados por el proveedor. Si el CSP no desarrolla un aislamiento adecuado entre recursos virtuales, la virtualización y el entorno *multi-tenant* pueden causar graves problemas de seguridad al cliente.

Estas diferencias en la infraestructura física con relación al modelo tradicional hace necesario cambiar el enfoque de la seguridad y adoptar nuevas medidas más centradas en los datos en lugar de sobre la infraestructura, ya que ahora la parte

de la infraestructura es dependiente del proveedor, por lo que el usuario no tiene mucho margen de maniobra sobre él [9].

Las citadas distinciones dan lugar a lo que se conoce como responsabilidad compartida. Este término hace referencia a las diferentes obligaciones que tienen tanto el CSP como el cliente, y que es fundamental que ambos tengan en cuenta para conseguir un nivel óptimo de seguridad. El proveedor es responsable de la infraestructura subyacente a los servicios que proporciona y que hace posible dicha oferta (seguridad física, hardware, sistema operativo *host*, virtualización, redes, etc.). A partir de ahí comienza la responsabilidad del cliente, desde la configuración de los productos y sus actualizaciones hasta el firewall que los proteja. Las responsabilidades concretas, no obstante, dependerán de cada producto determinado: no implican las mismas obligaciones una solución de infraestructura como servicio (*Infrastructure-as-a-Service*, IaaS), que una de plataforma como servicio (*Platform-as-a-Service*, PaaS) o una de software como servicio (*Software-as-a-Service*, SaaS), por ejemplo [11].

En este sentido, el equipo de Amazon Web Services propone el modelo de responsabilidad compartida que se muestra en la Figura 2.1, diferenciando entre «seguridad de la nube», que corresponde al CSP, y «seguridad en la nube», que es tarea del cliente.

2.2. Retos y problemas sin resolver

Las amenazas pueden ser de naturaleza interna (dentro de la red corporativa de la organización) o externa (provenientes de la red a través de Internet), y las posibles soluciones a las mismas únicamente pueden diseñarse de manera adecuada si se definen de antemano unos niveles de confianza entre las dos partes involucradas. Estos hacen referencia, por ejemplo, a tratar al CSP como si fuera completamente inseguro o incluso malicioso, a desconfiar del mismo sólo en algunos aspectos, o a confiar plenamente en el mismo.

Es precisamente este último caso el más habitual en entornos Cloud, donde los clientes asumen que no se pueden comprometer las capas de aplicación, hipervisor y/o sistema operativo que pone a disposición el proveedor y establecen las medidas de seguridad acorde a esta creencia. Además, en cuanto a la red, usualmente se considera segura la red corporativa contenida en la nube y las relaciones entre los servicios del CSP, y solamente se establecen puntos de acción en la periferia de la misma, vigilando exclusivamente el tráfico entre el cliente y los servicios en la nube, conocido generalmente como «tráfico Norte-Sur» (véase Figura 2.2).

Sin embargo, la realidad determina que también pueden materializarse amenazas aprovechando vulnerabilidades en el hipervisor o sistema operativo, por ejemplo, o utilizar como punto de entrada algún activo de la red interna direc-

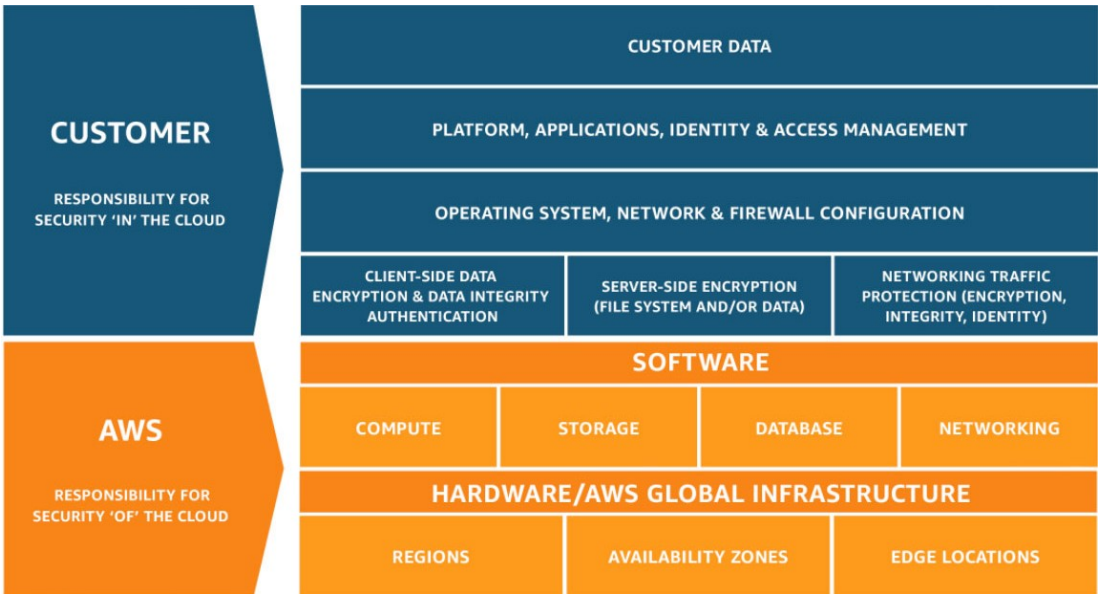


Figura 2.1: Modelo de responsabilidad compartida propuesto por AWS [11]

tamente sin vulnerar el perímetro establecido, como puede ser sirviéndose de un empleado en un ataque de *Phising*. Además, el perímetro se ha fragmentado mucho con la necesidad de incorporar conexiones entre negocios (*Business to Business*, B2B) y la creación de puestos de trabajo remotos y/o móviles. Esto implica que los CSPs deben valorar y desarrollar soluciones que permitan a los clientes operar con seguridad asumiendo que haya vulnerabilidades en el hipervisor, sistema operativo o aplicación que se utilice, según se opere con un modelo de entrega IaaS, PaaS o Saas, respectivamente, a la vez que los clientes no deben confiar y dar por sentado que no puedan acontecer ataques a sus sistemas utilizando estos puntos de entrada.

Asimismo, en cuanto a la red, no se debe descuidar el tráfico lateral dentro del centro de datos y las comunicaciones entre distintos servicios del proveedor, conocido como «tráfico Este-Oeste» (véase Figura 2.2), por el que posibles atacantes podrían pivotar para alcanzar unos activos desde otros a los que hayan logrado ya acceso [12]. Esto también aplica a los CSPs, que deben asegurar que un usuario con fines maliciosos no sea capaz de sobrepasar su perímetro dentro de la infraestructura del proveedor y obtener información de sistemas dedicados a otros clientes.

Aunque los *firewalls* están ideados principalmente para proteger el perímetro, puede ser una buena solución a este problema su uso en combinación con VLANs para defender el tráfico Este-Oeste, situándolos en los puntos de conexión de las distintas redes internas virtuales. No obstante, hay que prestar atención a la arquitectura de red, puesto que pueden producirse cuellos de botella que disminuyan significativamente la calidad de las comunicaciones.

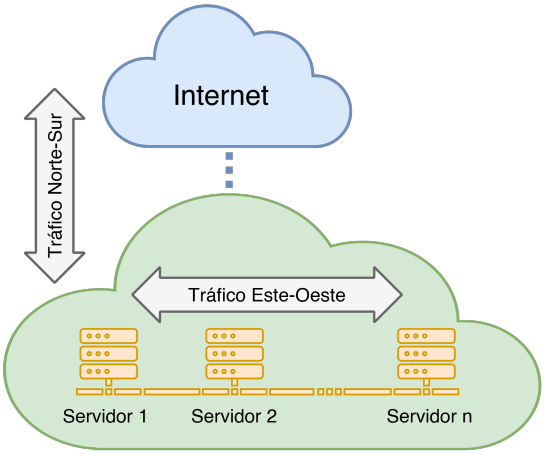


Figura 2.2: Tráfico en infraestructuras *Cloud*

En el Apéndice A. Desafíos de seguridad en el *Cloud* se recogen con detalle nuevos desafíos que surgen en relación con los datos, las comunicaciones y la virtualización característica de la nube según [13] y [14].

2.3. Mejores prácticas para mitigar riesgos

Con el objetivo de reducir al mínimo posible el riesgo de que se materialice una amenaza en la infraestructura Cloud y tratando de reducir los problemas planteados en la sección anterior, es conveniente considerar un compendio de buenas prácticas que implantar y desarrollar, desde el punto de vista del cliente.

Uno de los aspectos más importantes es formar a todos los equipos involucrados. Deben ser conocedores del contexto en el que trabajan y las diferencias con el modelo tradicional, el modelo de seguridad compartida concreto y las responsabilidades y roles que debe desempeñar cada miembro de la organización. Además, el personal encargado del despliegue, configuración y operación deben estar cualificados con la formación técnica necesaria en las tecnologías y productos de seguridad *Cloud* que deban utilizar, para poder desempeñar sus funciones con el mayor grado de seguridad posible. Cabe destacar también que deben tener a su disposición una documentación oficial o fuentes contrastadas en las que poder buscar aquello que desconozcan, de manera que se evite que escojan por error otras que no tengan en cuenta la seguridad y les haga originar problemas de manera involuntaria.

Otra medida que se debe tomar es especificar claramente quién es el responsable de tomar cada tipo de decisión de seguridad sobre los servicios contratados en la nube, evitando que se queden dilemas sin resolver porque nadie quiera asumir la responsabilidad o nadie sepa a quién pedir o dónde buscar una solución. Si

todo el personal conoce quiénes son estos encargados, serán capaces de hacerles llegar cada problemática que se plantee de manera que sea la persona indicada la que señale la solución necesaria.

Una disposición que ayuda mucho a minimizar el riesgo en las organizaciones es su capacidad de estar preparados frente a las posibles amenazas. Esto implica dedicar recursos tanto humanos como materiales a la monitorización y fortificación de los sistemas y a la detección de amenazas. La efectividad de estos procesos se incrementa si se destinan también recursos a *Threat Hunting* para buscar nuevas amenazas y tipologías de ataques de manera proactiva antes de que se lleguen a materializar en la organización [15]. Además, se deben tener redactadas políticas y planes de respuesta ante incidentes que estos equipos y el personal de *Digital Forensics and Incidents Response* (DFIR) puedan seguir cuando ocurran incidentes. Es muy importante que estos documentos estén actualizados y se vayan mejorando de manera constante con la última información disponible, para evitar que queden desactualizados y se gestionen de manera errónea los incidentes.

En cuanto a la autenticación para el uso de las tecnologías, recientemente se está impulsando activamente el uso del multifactor de autenticación, que combina algo que se conoce (típicamente una contraseña) con algo que se posee (como por ejemplo una tarjeta criptográfica o un dispositivo móvil) y/o con algo que se es (biometría). Otra posibilidad que se puede utilizar en este sentido son los mecanismos de autenticación sin contraseña, que utilizan enfoques biométricos y dispositivos como teléfonos móviles sin utilizar en ningún caso una contraseña o cualquier otro elemento que se deba recordar. El hecho de utilizar alguno de estos mecanismos de autenticación, especialmente en sistemas con un impacto crítico en el porvenir de la organización, puede disminuir hasta un 99,9 % la probabilidad de que una cuenta sea comprometida [16].

Para que se puedan implantar todas estas medidas de manera satisfactoria y lograr que todas impulsen a la organización en un mismo sentido, es fundamental establecer una estrategia de seguridad única. De esta manera se logrará que todos los equipos establezcan sus objetivos pensando en unas metas comunes y alineadas con el rumbo de la empresa, evitando que los grupos trabajen de manera aislada, lo que puede causar que alguno de ellos tome acciones que devalúen o socaven involuntariamente los esfuerzos de otros equipos, creando una fricción que ralentice el progreso general [17].

2.3.1. *Cloud Security Alliance* (CSA)

La *Cloud Security Alliance* (CSA) es una organización sin ánimo de lucro que define buenas prácticas y conciencia sobre ellas con el fin de lograr un entorno de computación en la nube más seguro. Además, opera un programa de certificación muy valorado en el sector, el *CSA Security, Trust & Assurance Registry*,

conocido como *STAR*, que evalúa la seguridad de los CSPs [18]. En función del nivel de evaluación, se pueden distinguir tres grados de certificación: nivel 1 (autoevaluación), nivel 2 (evaluación independiente) y nivel 3 (auditoría continua) [19].

Entre la gran cantidad de documentos e información que aporta a la comunidad, se encuentra la *Security Guidance*, que alcanza ya su versión 4.0. En ella se proponen medidas y buenas prácticas para que las organizaciones puedan cumplir sus objetivos y metas a la vez que mitigan los riesgos asociados a la migración de sus servicios a productos de computación en la nube. Para clasificar los ámbitos de estudio y aplicación de la guía, se divide el espectro en catorce dominios, detallados en la Sección B.1. Guía *Security Guidance 4.0* [20].

La CSA también genera de manera trienal un informe recogiendo las amenazas más destacables en entornos *Cloud* para sensibilizar sobre ellas y el riesgo y las vulnerabilidades en la nube. Actualmente, hay cuatro publicaciones disponibles:

- *Top Threats to Cloud Computing: Egregious Eleven* [21], de 2019, contiene 11 amenazas.
- *The Treacherous Twelve: Cloud Computing Top Threats in 2016* [22], destaca 12 amenazas.
- *The Notorious Nine: Cloud Computing Top Threats in 2013* [23], resalta nueve amenazas.
- *Top Threats to Cloud Computing V1.0* [24], primera versión publicada en 2010, que tan solo recogía 7 amenazas.

La Figura 2.3 muestra de forma gráfica las diferentes amenazas recogidas en cada uno de los cuatro informes, ordenadas de manera descendente según su relevancia, y mostrando su evolución a lo largo del tiempo en el caso de aquellas que aparecen en más de un informe.

Estos informes ponen de manifiesto que, al contrario de lo que pueda parecer, las amenazas más comunes no surgen por la ausencia de algunos de los productos específicos de seguridad, sino que están relacionadas con los productos que sustentan los activos del cliente, concretamente con su uso y configuración. Centrando el análisis en el informe de 2019 [21], se incluyen, por ejemplo, la «Mala configuración y control de cambios inadecuado», la «Gestión insuficiente de identidades, credenciales, accesos y claves», la «Amenaza interna» y el «Abuso y uso nefasto de servicios en la nube», todas ellas causadas por problemas ajenos al uso de productos concretos de seguridad.

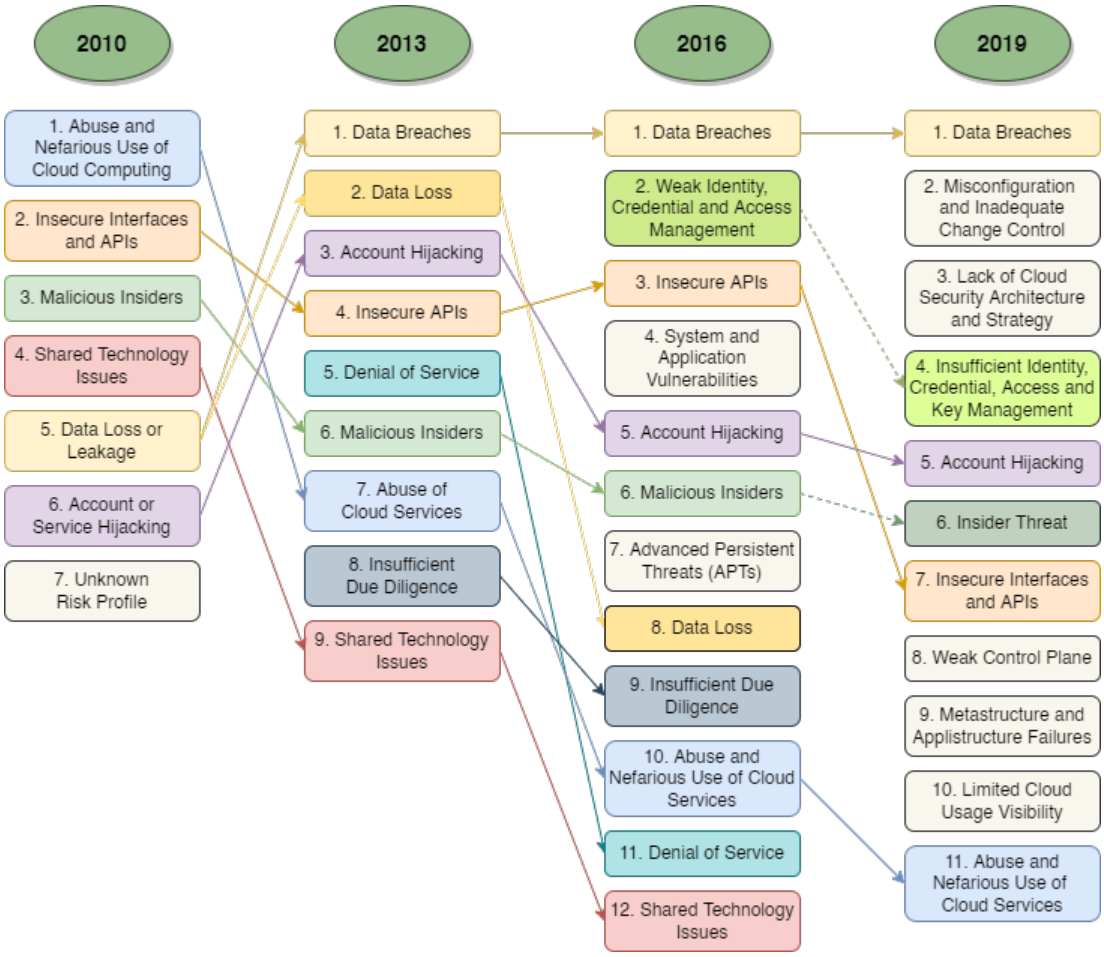


Figura 2.3: Relación de las amenazas incluidas en los informes *CSA Top Threats*

2.3.2. Seguridad en Amazon Web Services

Los CSPs también proponen medidas y soluciones para mejorar la seguridad de sus clientes. Como se ha mencionado en el capítulo anterior (véase Capítulo 1. Introducción), en este caso se ha optado por analizar *Amazon Web Services* (AWS), que pone a disposición de los clientes una gran cantidad de productos con los que aplicar seguridad. Concretamente, categorizados como «*Seguridad, identidad y conformidad de AWS*», Amazon cuenta con 25 productos en su catálogo [25]. Esto permite cubrir gran parte del espectro y defenderse frente a gran cantidad de amenazas. Sin embargo, escoger el servicio óptimo y configurarlo de manera adecuada, a pesar de la documentación disponible, puede ser complicado.

Centrando el análisis en servicios y productos concretos de AWS, se pueden aplicar medidas que mitiguen el riesgo de que una amenaza se materialice. La Sección B.2. Medidas de seguridad en los servicios más relevantes de AWS contiene algunas de las tareas más relevantes de los productos más comunes de AWS,

junto a aspectos a tener en cuenta durante su despliegue y uso [26].

CIS Benchmarks

El *Center for Internet Security* (CIS) es otra organización sin ánimo de lucro impulsada por la comunidad y reconocida globalmente por empresas y organizaciones gracias a sus *CIS Controls* y *CIS Benchmarks*, un compendio de controles y buenas prácticas enfocados a la seguridad de los datos y los sistemas de información. Además, proporcionan servicios para proteger los sistemas frente a amenazas emergentes. Este es el caso de *CIS Hardened Images*, unos entornos de computación en la nube seguros, escalables y disponibles bajo demanda [27].

Entre los diferentes *CIS Benchmarks* publicados por dicha organización, se puede encontrar uno sobre los servicios de AWS, cuya última versión hasta la fecha es la 1.4.0, publicada el 28 de mayo de 2021 [28]. En él, para cada uno de los productos analizados, se proponen diferentes comprobaciones que se han de valorar en el entorno *Cloud* concreto y que ayudan a mitigar posibles amenazas. De las gran cantidad de disposiciones que se incluyen es posible que no todas sean compatibles; simplemente se trata de una evaluación y habrá casos en los que se cumplan algunas de ellas y otros en los que se validen unas diferentes, en función de las características del entorno evaluado.

Los productos de AWS que conforman el alcance del documento son:

1. *Identity and Access Management (IAM)*: gestiona el acceso a los recursos controlando la autenticación y la autorización.

2. *IAM Access Analyzer*: identifica los recursos compartidos con entidades externas y los accesos a los mismos.

3. *Simple Storage Service (S3)*: almacena objetos y datos de muy diversos tipos asegurando escalabilidad, disponibilidad, seguridad y rendimiento óptimo.

4. *Elastic Compute Cloud (EC2)*: ofrece capacidad informática, pudiendo elegir los recursos, capacidades y sistema operativo necesarios.

5. *Relational Database Service (RDS)*: proporciona capacidad escalable y automatiza las tareas administrativas derivadas de desplegar bases de datos relacionales.

6. *CloudTrail*: monitoriza y registra la actividad de las cuentas de usuarios y el uso que hacen de las APIs.

7. *CloudWatch*: monitoriza el rendimiento, el uso de recursos y el estado general de los servicios.

8. *Config*: analiza las configuraciones de los recursos, monitorizando y registrando las mismas para facilitar auditorías.

9. *Simple Notification Service (SNS)*: comunica aplicaciones entre sí (*Application-to-Application*, A2A) y aplicaciones y personas (*Application-to-Person*, A2P) mediante un sistema de mensajería de publicación y suscripción.

10. *Virtual Private Cloud (VPC)*: facilita control absoluto sobre el entorno de redes virtuales entre recursos, ofreciendo opciones en cuanto a ubicación, conectividad y seguridad.

Estos 10 servicios pueden ser agrupados en 5 categorías diferentes en función de sus cometidos. Véanse estas junto a algunas de las comprobaciones que el CIS propone para el caso de AWS y los productos involucrados en la Sección B.3. *CIS Benchmark 1.4.0*.

2.3.3. Seguridad dinámica

Uno de los mayores intereses de los administradores de sistemas es poder evaluar y conocer el riesgo de los activos para así determinar la mejor manera en que pueden defenderlos de entre una lista de posibles contramedidas, además de conocer la relación entre el coste y el beneficio que se va a obtener para poder ajustarse al presupuesto establecido. Sin embargo, el riesgo varía a lo largo del tiempo, sobre todo en entornos *Cloud* dados su dinamismo y flexibilidad, y no hay ningún control capaz de ajustarse a todos los escenarios posibles. Este caso pone de manifiesto que las soluciones hasta ahora sugeridas, tanto los planteamientos genéricos de la CSA (véase Subsección 2.3.1. *Cloud Security Alliance (CSA)*) como específicos para los distintos proveedores, en este caso AWS (véase Subsección 2.3.2. Seguridad en Amazon Web Services), no son suficientes. Siguen siendo controles estáticos y difíciles de modificar y adaptar a los entornos elásticos y cambiantes propios de la computación en la nube, que requieren medidas de seguridad diferentes según se produzcan cambios en el contexto o en función de la infraestructura concreta de cada momento. Este dinamismo y variación en los controles aplicados en cada momento se conoce como seguridad dinámica [29, 30] y permite hacer frente a muchos de los desafíos expuestos en la Sección 2.2, detectando indicios de cuándo se pueden estar materializando las diversas amenazas mencionadas y tomando decisiones en función de cada caso.

Para automatizar dichas decisiones e implantarlas en tiempo real, sin depender de la disponibilidad del personal encargado de ello, surge la seguridad adaptativa: sistemas capaces de modificar su arquitectura, comportamiento o configuración en función de un estado o contexto determinados gracias a una lógica de adaptación. En trabajos previos se distinguen dos enfoques diferentes [2]:

- **Seguridad basada en el contexto**: basada en información del entorno y contexto, como por ejemplo la geolocalización, reputación de una dirección IP, tipo de dispositivo, número de peticiones realizadas, contexto del valor del negocio, momento temporal, etc., tratada de manera agregada, puede determinar medidas más efectivas que las acciones preventivas tradicionales. Suele aplicarse en procedimientos de autenticación, autorización y control de acceso en sistemas distribuidos basándose en el contexto de cada petición o intento de acceso concreto [31, 32, 33].
- **Seguridad incremental o inteligente**: se basa en la recolección y estandarización de datos de la red, aplicaciones, ficheros de log, equipos de la infraestructura, etc. para posteriormente analizarlos con técnicas como *Big Data* y/o herramientas como los sistemas de gestión y correlación de eventos (*Security Information and Event Management*, SIEM) de manera que se detectan anomalías y desviaciones del comportamiento habitual.

Un concepto destacable que forma parte de la seguridad dinámica es la seguridad basada en el riesgo. Este modelo monitoriza, cuantifica y evalúa el riesgo de los activos de la organización comparando con los estándares aceptables que se hayan definido para así determinar la hoja de ruta. Estos estándares o riesgo tolerable por la compañía debe ser establecidos previamente y se deben haber desarrollado las medidas que se implantarán en caso de que sea necesario. Aunque la mayoría de aplicaciones se centran también en el control de acceso, también se utilizan en sistemas de control industrial, redes definidas por software (*Software Defined Networks*, SDNs) o procesamiento de datos [34, 35].

El ciclo *MAPE-K* es uno de los modelos más extendidos que se utilizan como base para desarrollar soluciones de seguridad adaptativa, bien sea basándose en el riesgo o no. Está formado por cuatro acciones que se ejecutan en bucle, sustentadas por un quinto componente (véase Figura 2.4) [36]:

- **Monitorizar** el contexto y los recursos gestionados.
- **Analizar** los datos recogidos buscando cambios significativos.
- **Planificar** la adaptación que se debe realizar (o no).
- **Ejecutar** los controles que correspondan.
- Base de **conocimiento** compartido entre todos los elementos anteriores.

Risk-based Adaptive Security

Una novedosa propuesta en el ámbito de la seguridad adaptativa y la seguridad basada en el riesgo es *Risk-based Adaptive Security* (RiAS) [2]. Ha sido diseñada

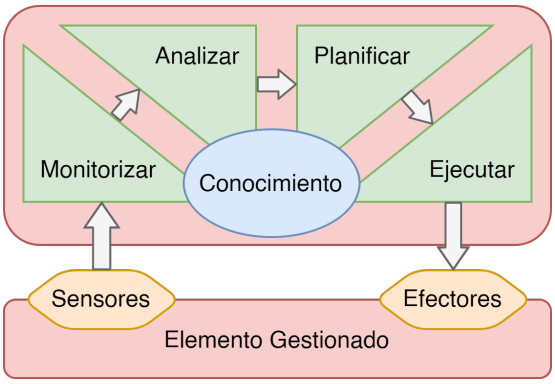


Figura 2.4: Elementos del modelo MAPE-K

y desarrollada para poder utilizarse de forma genérica, abarcando múltiples casos de uso y dominios de aplicación, ya sea *Cloud*, IoT, aplicaciones web, redes SDN, etc. Esto lo convierte en ideal para los entornos cambiantes, distribuidos y heterogéneos característicos de la computación en la nube, eliminando la problemática del uso controles estáticos en este tipo de entornos.

Para lograr la adaptación, RiAS realiza mediciones utilizando métricas, como las siguientes:

- **Indicadores clave de riesgo (*Key Risk Indicators*, KRIs)**: probabilidad de que se materialice una amenaza superando el apetito por el riesgo de la organización, lo que causaría un impacto negativo en la misma. Por ejemplo: porcentaje de incidentes que involucran datos personales de clientes, número de sistemas sin parches actualizados, cantidad de incidentes de seguridad atribuidos a vulnerabilidades en sistemas de terceros, etc. [37].

- **Indicadores de ataques (*Indicators of Attacks*, IoAs)**: comportamientos o eventos dinámicos que justifican una alta probabilidad de que se está sufriendo un ciberataque en el momento o que se va a sufrir en un futuro. Estos pueden corresponder a cualquier fase del ciclo de vida del ataque, conocido como *cyber kill chain*, suelen necesitar un contexto para determinar si realmente se trata de un ataque o no, e impulsan una respuesta proactiva. Son, por ejemplo, escaneos de puertos, accesos no autorizados, procesos anormales, intentos de acceso a directorios sin autorización, actividad fuera de horas de trabajo, etc.

- **Indicadores de compromiso (*Indicators of Compromise*, IoCs)**: evidencia estática y conocida que determina que la seguridad de una red o sistema ha sido vulnerada, para activar una respuesta reactiva. Por ejemplo: detección de firmas de virus, presencia de archivos determinados, consultas DNS de dominios maliciosos, comunicación con direcciones IP incluidas en listas negras, etc.

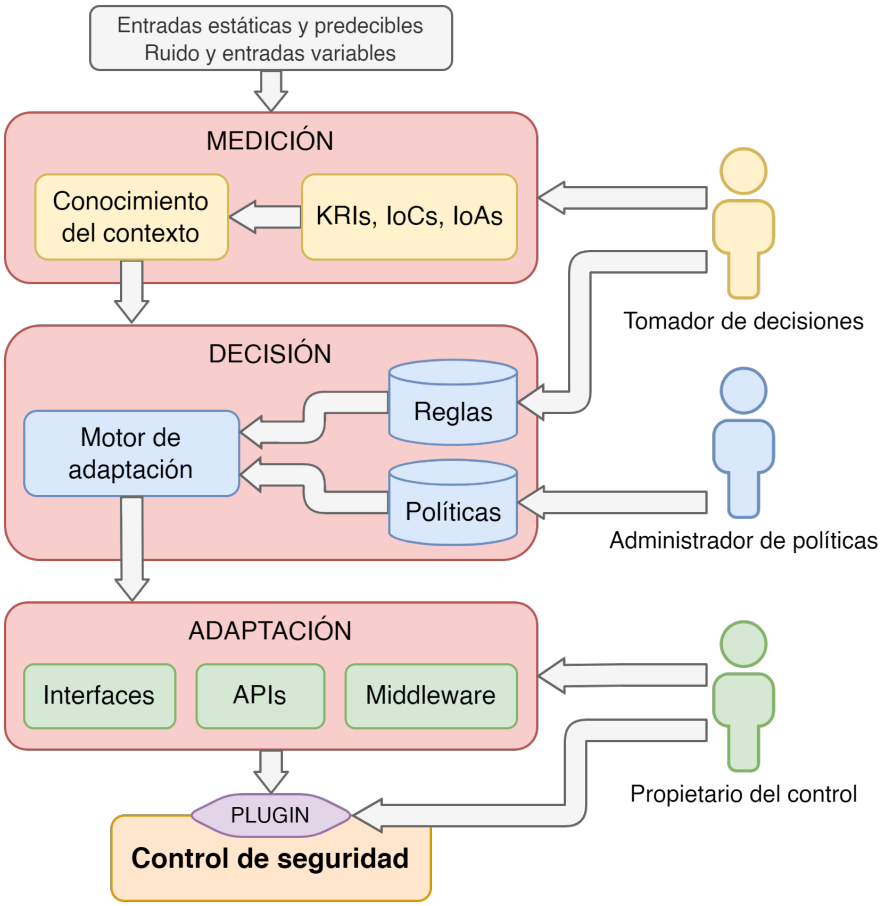


Figura 2.5: Estructura de RiAS (basado en [2])

- **Cuantificadores del riesgo**: evaluación del riesgo de un sistema o equipo en un momento concreto, proporcionado por una metodología de análisis de riesgo, un sistema de medición del riesgo, o cualquier proceso que determine un nivel de riesgo conforme a una escala.

Utilizando estas entradas se pueden generar diversas reglas que determinen cuándo, cómo y dónde se deben realizar adaptaciones, cuyas decisiones impactarán en los controles de seguridad, integrados y definidos a través de APIs y *middleware*. El flujo que sigue RiAS para realizar las adaptaciones puede observarse en la Figura 2.5.

3

Solución propuesta para la seguridad adaptativa

A lo largo de este capítulo se describe el caso de uso planteado para el desarrollo de la prueba de concepto. Además de especificar la arquitectura de los entornos de comercio electrónico y argumentar la necesidad de implantar medidas de seguridad adaptativas en ellos, se detalla la lógica de la prueba de concepto y el marco teórico de trabajo planteado.

Para la implantación de las medidas sobre el entorno de comercio electrónico, se han definido dos escenarios que afectan a dominios diferentes de la arquitectura y se han escogido dos controles de seguridad para los mismos. Además, se han detallado los KRIs que permiten estimar el riesgo en cada escenario y los requisitos que se deben satisfacer para poder medirlos adecuadamente.

Por último, se ha detallado la elección de una solución de seguridad adaptativa basada en riesgo y su relación con el caso de uso planteado, así como los pasos necesarios para su implementación y la relación entre sus elementos y la arquitectura definida.

3.1. Motivación

Como se ha mencionado en el Capítulo 1. Introducción, a lo largo de este trabajo se va a desplegar y adaptar la solución RiAS sobre un entorno de comercio

electrónico. Es muy común que este tipo de negocios sufran fluctuaciones a lo largo del tiempo, dado que la demanda de diversos productos es muy superior en unos periodos que en otros. Por ejemplo, en campañas especiales como la navideña, los periodos de rebajas, días especiales con descuentos como el «Día sin IVA» o el «Black Friday» (viernes negro), etc. la cantidad de compradores aumenta considerablemente, causando una gran demanda de productos, y una agitación y revuelo poco habitual en busca de la mejor oferta. Sin embargo, igual que las compras se intensifican en estas épocas, también disminuyen una vez pasan las fechas señaladas. Esta fluctuación impacta directamente en la infraestructura tecnológica necesaria para hacer frente a la demanda, requiriendo mayor cantidad de recursos cuando aumenta la demanda, de modo que el sistema de información sea capaz de atender las peticiones de los compradores.

El interés de gran parte de la población por las compras y por obtener el mayor descuento posible, es un terreno ideal para ciberdelincuentes y atacantes, que buscan sacar provecho de la situación. Esto conlleva un aumento del riesgo en las operaciones de este tipo de negocios, dado que tanto la probabilidad de que la organización sea víctima de un ciberataque, como el impacto que causaría la interrupción de las operaciones en un momento de gran volumen de ventas, aumentan considerablemente con eventos especiales como los mencionados. Es por ello que, además de necesitarse una mayor cantidad de recursos para soportar la actividad propia del comercio, requieren una serie de medidas de seguridad que en un momento de pocas operaciones podrían no ser tan necesarias, porque la probabilidad será seguramente menor, y además el impacto no será tan catastrófico como en fechas señaladas.

Este factor de riesgo detallado a modo de ejemplo, sumado a otros tanto generales como específicos de cada negocio en particular, implican cambios en el contexto, justificando la importancia de implantar un sistema adaptativo de seguridad que sea capaz de adaptar las medidas y controles de seguridad al riesgo y las características concretos de cada momento y/o situación.

3.2. Arquitectura de la solución

Gran parte de las tiendas en línea están sustentadas por sistemas de gestión de contenidos (*Content Management System*, CMS), herramientas software para la creación y administración de sitios web que facilitan a los administradores y editores la gestión de los elementos publicados, ya sean productos de una tienda, artículos de un blog, noticias de un medio de comunicación, etc., sin apenas conocimiento técnico, ya que evitan tener que editar manualmente el código de cada fichero o página que conforme el sitio web. Algunos ejemplos de CMSs de código abierto son *WordPress* [38], *Joomla* [39] o *PrestaShop* [40], aunque también existen soluciones propietarias como *Kentico* [41] o *Shopify* [42].

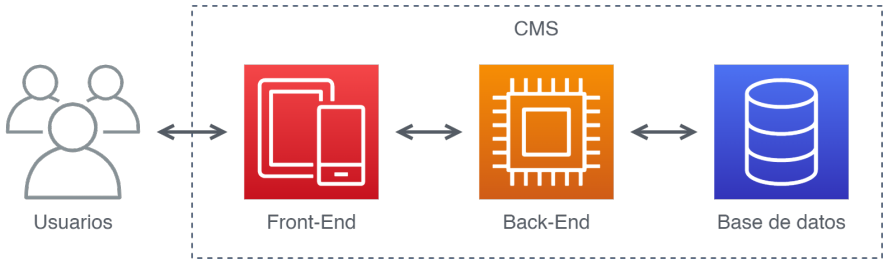


Figura 3.1: Capas de un CMS común

La mayoría de estas herramientas se basan en una arquitectura de tres capas (véase Figura 3.1), capa de presentación, capa lógica y capa de datos, implementadas a través de los siguientes elementos:

1. **Capa de presentación**: capa más cercana al usuario, formada, normalmente, por el *front-end* de la aplicación, implementa la interfaz gráfica con todas las utilidades que permiten tanto a clientes como a proveedores y administradores desarrollar sus operaciones. La interacción con esta capa se realiza por medio de los clientes remotos de los usuarios, generalmente navegadores web o aplicaciones móviles, aunque también pueden ser asistentes de voz, por ejemplo.

2. **Capa lógica**: capa intermedia, controla la lógica de negocio de la tienda, realizando todas las tareas de procesamiento y gestión de los datos de la capa subyacente, y generando la información mostrada en la interfaz de usuario. Suele conocerse como *back-end*.

3. **Capa de datos**: administrada por el sistema gestor de base de datos (SGBD), proporciona el almacenamiento de toda la información que da soporte al CMS, que, para una tienda en línea, se corresponderá con los productos, precios, clientes, pedidos, etc., junto a sus correspondientes relaciones lógicas.

3.2.1. Seguridad tradicional en sistemas de gestión de contenidos

En arquitecturas de tres capas, como la planteada para el CMS, suelen considerarse tres pilares básicos en materia de ciberseguridad que permiten definir capacidades de seguridad que, en la medida de lo posible, reduzcan el riesgo; estos pilares son la confidencialidad, la integridad y la disponibilidad (CIA, por sus siglas en inglés, *Confidentiality*, *Integrity* y *Availabily*). Prestando atención a estos y a los elementos del CMS que conforman las tres capas (véase Figura 3.1), se pueden definir algunas medidas de seguridad típicas, como las siguientes:

- **Confidencialidad**: garantiza que únicamente acceden a la información los usuarios autorizados, generalmente a través del cifrado, tanto en reposo como en tránsito en las comunicaciones entre los elementos del CMS.
- **Integridad**: certifica que los activos y/o información no son modificados por usuarios no autorizados, valiéndose habitualmente de la autenticación y la autorización de usuarios, implementadas generalmente en el núcleo del CMS, en la capa lógica. También se pueden añadir elementos a la arquitectura de referencia, como es un WAF, que ayuda a prevenir diferentes ataques web que pueden afectar a la integridad, tales como inyecciones de código SQL o ataques *cross-site scripting* (XSS), entre otros.
- **Disponibilidad**: permite que un usuario autorizado pueda utilizar un activo o acceder a información siempre que lo desee, usualmente mediante la replicación tanto del servidor de la aplicación, que engloba el *front-end* y el *back-end*, como del servidor de la base de datos, para asegurar la escalabilidad y, por ende, la disponibilidad de los sistemas. Para gestionar esta replicación, se hace uso de balanceadores de carga, que distribuyen el tráfico entre las diferentes instancias disponibles.

El cifrado es la medida de seguridad que podría considerarse más estática, ya que apenas presenta variaciones: o bien se cifra, o bien no. Sin embargo, el resto de controles incluyen una componente variable muy importante. Por ejemplo, los balanceadores de carga tienen una naturaleza muy dinámica, ya que las decisiones que tomen dependen en gran medida de su configuración y del contexto: se pueden configurar pares de sistemas en modo activo-activo, en los que se distribuya de igual manera la carga de trabajo, o en modo activo-pasivo, en los que una instancia asuma toda la carga computacional habitual, mientras que la otra únicamente actúe cuando se consuman todos los recursos de la primera. Igual sucede con el WAF, en el que se pueden configurar diferentes reglas para que actúe de infinitas maneras, filtrando en cada caso un tipo diferente de tráfico, unas peticiones web concretas, etc. Asimismo, la autenticación y autorización también puede variar en función del contexto. Respecto a la primera, se pueden configurar métodos como la autenticación multifactor que únicamente actúe en determinados casos, por ejemplo, cuando se acceda desde ciertos países, en horario nocturno, etc. En función del contexto, también pueden modificarse los permisos asignados a cada usuario en la autorización, por ejemplo, los privilegios de cada rol de usuario en una autorización por roles en periodos de mantenimiento de la tienda.

3.3. Escenarios de adaptación

En la Subsección 3.2.1. Seguridad tradicional en sistemas de gestión de contenidos se han analizado varios controles de seguridad tradicionales que podrían

Tabla 3.1: Escenarios para la adaptación

Riesgo	Indicador	Capa
Suplantación del administrador	Cantidad de peticiones a `/admin`	*Front-end*
Integridad del catálogo	Cantidad de cambios de precios sospechosos	Base de datos

modificarse de manera adaptativa en función del contexto de operación, bien sea en periodos fijos, por ejemplo según horarios definidos, o bien de manera totalmente adaptativa, analizando y evaluando el riesgo en cada momento. Para este segundo caso, se pueden usar diferentes KRIs que soporten la toma de decisiones para realizar o no las adaptaciones. Algunos de los KRIs más evidentes y relevantes pueden ser el volumen de ventas, las consultas al sitio web, las modificaciones en información de los productos como los precios, los cambios de stock, las geolocalizaciones de los usuarios junto a las direcciones de envío, etc. Con estos indicadores y la definición de unos umbrales tolerables, se pueden activar y/o desactivar medidas de seguridad de las analizadas anteriormente, como modificación de reglas de un WAF o modificación de los mecanismos de autenticación, o realizar acciones concretas sobre los elementos de la tienda en línea, como cancelación temporal del stock, bloqueo temporal de pedidos, etc.

De todos los múltiples escenarios de riesgo en los que se podría aprovechar la potencia de un sistema de seguridad adaptativo, se han escogido dos, recogidos en la Tabla 3.1, por su potencial impacto y alta probabilidad de ocurrencia en una tienda de comercio electrónico corriente. En ambos casos, se ejecutarán las adaptaciones cuando se superen los umbrales que se establezcan para cada KRI.

3.3.1. Suplantación del administrador

El primer escenario busca detectar una supuesta suplantación del administrador del *front-end* de la tienda en línea. Para ello, se ha de estudiar el uso normal que realiza el administrador y analizar el número de peticiones a la interfaz de administración, denotada genéricamente como `/admin`, para establecer un umbral aceptable de peticiones. Superado este umbral, se activará un segundo factor de autenticación que garantice que es el propio administrador el que está realizando los cambios en la tienda, en lugar de un usuario malicioso que le está suplantando.

Para poder detectar las peticiones realizadas, se pueden monitorizar diferentes registros (*logs*), como, por ejemplo, los del servidor web o los del CMS, según la información que cada uno de ellos aporte. El único requisito es que proporcionen información sobre los accesos a la ruta de administración.

Tabla 3.2: Lógica de adaptación del escenario 1

Escenario de adaptación: suplantación del administrador
Control adaptado: 2FA
Requisitos de adaptación: 1. Activar 2FA si peticionesAdmin mayorQue X en Y tiempo 2. Desactivar 2FA si peticionesAdmin menorIgualQue Z en T tiempo

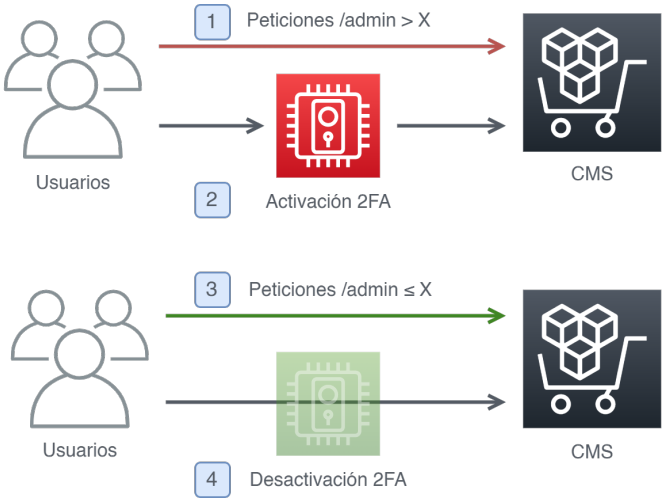


Figura 3.2: Adaptaciones en el escenario 1

Cabe destacar que el hecho de superar el umbral establecido no garantiza que se esté realizando la suplantación, simplemente se trata de un KRI que indica un aumento del riesgo, motivo por el cual se activa un control adicional. Puede ser que se estén realizando labores de administración que requieran de más tareas a las habituales, actividades que no serán interrumpidas al activar el segundo factor de autenticación: únicamente se pedirá la validación adicional y, tras ello, el administrador podrá continuar con sus responsabilidades.

Como muestra la política redactada en la Tabla 3.2, este escenario cuenta con dos condiciones antagónicas, una que activa el segundo factor de autenticación cuando se supera una cantidad determinada de peticiones dentro de una ventana temporal fija, y la otra que lo desactiva en caso contrario. Estos dos casos aparecen también representados de manera gráfica en la Figura 3.2.

El Algoritmo 3.1 muestra en pseudocódigo una propuesta para la implementación de este escenario. Tras filtrar de los *logs* aquellos registros enmarcados en la ventana temporal *win*, se comprueba si las peticiones totales superan el umbral *thres*, y en caso de que el segundo factor de autenticación no estuviera previamente activado, se activará. De la misma manera, si no se supera el umbral y el segundo factor está activado, se desactivará.

Algoritmo 3.1 Flujo del escenario 1

```
1: procedure ESCENARIO1(Log, admin, win, thres)
2:     end ← now
3:     start ← end − win
4:     group ← filter(Log, start, end)
5:     count ← 0

6:     for i ∈ group do
7:         if admin in i.path then
8:             count ← count + 1
9:         end if
10:    end for

11:    if count > thres AND 2FA is not enabled then
12:        ret ← ADAPTION1()
13:    else if count ≤ thres AND 2FA is enabled then
14:        ret ← ADAPTION2()
15:    end if
16:    return ret
17: end procedure

18: procedure ADAPTION1
19:    enable 2FA
20:    return true
21: end procedure

22: procedure ADAPTION2
23:    disable 2FA
24:    return false
25: end procedure
```

3.3.2. Ataque a la integridad del catálogo

El segundo escenario se centra en la integridad del catálogo de la base de datos, que comúnmente almacena, entre otros, información relativa a los productos que comercializa la tienda. Concretamente, este escenario busca detectar descensos en los precios de los productos que superen un porcentaje previamente definido, según las características y políticas de descuentos del comercio.

Para poder detectar las modificaciones, es necesario contar con alguna medida de monitorización de la base de datos, bien sea a través de sus registros (*logs*) o con otros mecanismos que aporten información sobre los cambios de precios en una ventana temporal fija. Además, es imprescindible poder obtener de dichos registros tanto el precio original como el precio tras la modificación, de manera

Tabla 3.3: Lógica de adaptación del escenario 2

Escenario de adaptación: integridad del catálogo
Control adaptado: WAF
Requisitos de adaptación: 1. Restringir WAF si descuentosAbusivos mayorQue X en Y tiempo 2. Suavizar WAF si descuentosAbusivos menorIgualQue Z en T tiempo

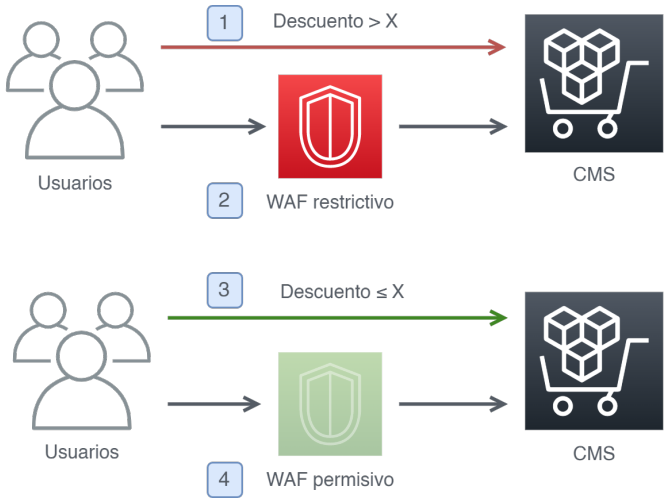


Figura 3.3: Adaptaciones en el escenario 2

que se pueda calcular el descuento y valorar si se supera el umbral establecido.

En este caso, la medida de control consiste en adaptar una protección básica en cualquier servicio web, como es un WAF. Para ello, se pueden definir dos modos, restrictivo y permisivo, con diferentes configuraciones del sistema. El modo permisivo será el predeterminado, aquel con una configuración básica que proporciona los controles de seguridad aceptables para las operaciones habituales del CMS en general, tanto de clientes y proveedores como de administradores. Por otro lado, en el modo restrictivo se configuran las reglas de control del WAF para limitar los orígenes de las comunicaciones que tienen permiso para realizar modificaciones en el catálogo del CMS, de manera que se prevengan cambios no autorizados.

De nuevo, este escenario se implementa mediante dos adaptaciones (véase Tabla 3.3), una que establece el WAF en modo restrictivo, y otra que lo configura de manera más permisiva. La Figura 3.3 muestra de forma gráfica estas dos casuísticas posibles dentro de este escenario.

El pseudocódigo propuesto en el Algoritmo 3.2 para este segundo escenario es muy similar al Algoritmo 3.1 del primer caso. Tras obtener los registros pertenecientes a la ventana temporal definida, se buscan las modificaciones en el

producto objetivo *prod*, y se calcula la diferencia de precio. Si el descuento es mayor al umbral *thres* y el WAF no está ya configurado en modo restrictivo, se ejecutará la primera adaptación, y en caso contrario se lanzará la segunda.

3.4. Elección de solución de seguridad adaptativa

De entre todas las herramientas y soluciones de seguridad adaptativa analizadas en el Capítulo 2. Estado del arte, se ha optado por elegir RiAS. Esta propuesta, como se ha introducido en la Subsección 2.3.3. Seguridad dinámica, define una arquitectura de tres capas (medición, decisión y adaptación) que permiten detectar los dos escenarios enunciados, analizarlos para tomar las acciones necesarias e implantar cambios si se requiere. Además, su enfoque basado en riesgo es óptimo para detectar circunstancias como las escogidas y adoptar medidas preventivas, que no requieren la materialización expresa de un ataque, aunque RiAS también permite actuar de manera reactiva y en tiempo real en estos casos.

Se han descartado otras propuestas por ser poco escalables al centrarse en desarrollar soluciones adaptativas específicas para dominios concretos, como IoT, dispositivos móviles, redes inteligentes, etc. Esto hace que dependan mucho del sistema para el que se implementen, y no permiten generalizar y agrupar diversas adaptaciones bajo un mismo modelo que permita su reutilización y/o su ampliación para su uso en otros dominios o ámbitos de aplicación.

3.4.1. Uso de RiAS

Para poder implantar RiAS de manera correcta para las adaptaciones definidas, es necesario profundizar en su arquitectura, que como se ha introducido en la Subsección 2.3.3. Seguridad dinámica, propone un flujo con tres etapas, sobre las que actúan tres sujetos diferentes (véase Figura 2.5):

- **Medición**: recibe las diferentes entradas del entorno y los indicadores, KRIs, IoCs, IoAs, etc. para generar un conocimiento del contexto. El tomador de decisiones decide cómo se captura el contexto y qué procedimientos o eventos hay que supervisar.
- **Decisión**: el tomador de decisiones genera las reglas que determinan cómo, cuándo y dónde se adaptarán los controles, y el administrador de las políticas define las políticas que gestionan dichas adaptaciones y los motivos por los que se realizan. En función de dichas entradas y del conocimiento recibido de la capa anterior, se decidirá si se debe adaptar, qué se debe adaptar, así como dónde y de qué manera es preciso hacerlo.

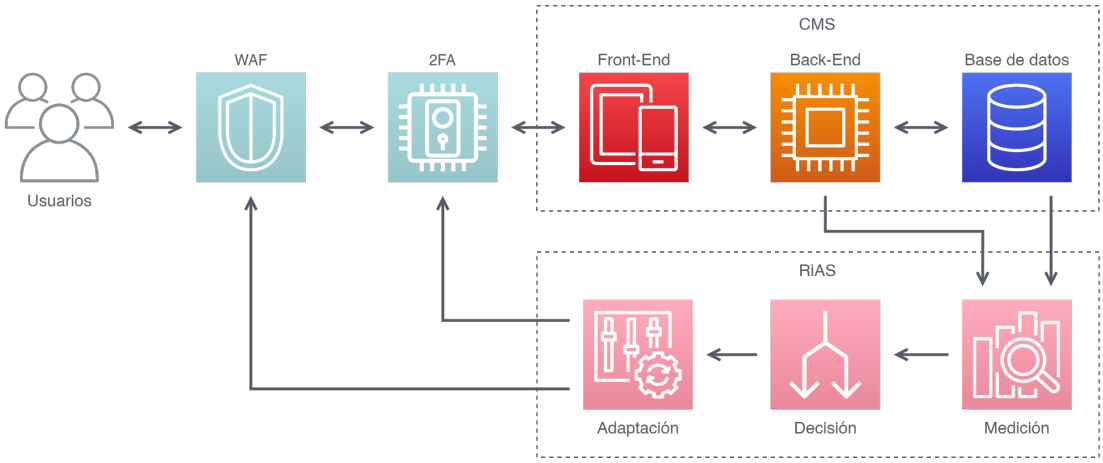


Figura 3.4: Relación de las capas de RiAS con el CMS y los controles añadidos

- **Adaptación**: sobre esta capa actúa el propietario del control, responsable de integrar las capas anteriores con los controles y contramedidas establecidas en las reglas y políticas, proporcionando interfaces, APIs y *middleware*. Esta capa actúa directamente sobre el control de seguridad mediante un *plugin* también gestionado por el propietario del control, que posibilita que se realicen los cambios.

Dados los dos escenarios planteados (véase Sección 3.3. Escenarios de adaptación), la capa de medición recogerá los registros con la información sobre las peticiones web y las alteraciones de la base de datos. Posteriormente, la capa de decisión evaluará los datos y determinará si hay que realizar o no alguna adaptación, qué adaptación en concreto, y en qué lugar. Por último, la capa de adaptación actuará sobre el WAF y el segundo factor de autenticación, si es necesario, modificando su configuración. Añadiendo los dos controles de seguridad a los tres elementos del CMS, y relacionándolos con las tres capas de RiAS, se obtiene la arquitectura de referencia de la prueba de concepto (véase Figura 3.4).

Por otro lado, es necesario realizar un trabajo previo de definición de las reglas, políticas e interfaces que gestionen los procesos de adaptación, basándose en la gobernanza y el contexto de aplicación. Las primeras definen el flujo principal de la adaptación: qué se va a medir, qué y cuándo se va a adaptar, y dónde se va a realizar la adaptación. Las políticas, por su parte, determinan cómo se va a realizar la adaptación, y el motivo y los fines perseguidos con la misma, mientras que la interfaz, ya desde un punto de vista técnico, es la encargada de implementar los cambios en los sistemas. Con estos pasos previos y los agentes involucrados en los mismos, será posible implantar RiAS de manera satisfactoria en 6 fases (véase Figura 3.5) que no necesariamente deben ser consecutivas, consumiéndose en las tres capas del modelo cada uno de los recursos definidos previamente de manera *offline*, sin acciones sobre los sistemas.

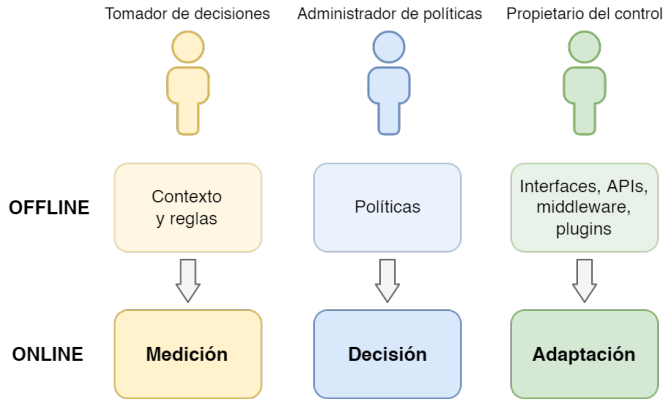


Figura 3.5: Flujo de implantación de RiAS (basado en [2])

Tabla 3.4: Política del escenario 1 a alto nivel

Name: Admin impersonation	*Owner*: Ecommerce website admin
Control: 2FA	*Type*: Reactive
Adaption conditions:	
Predicate 1: *observed*(EVENT-Admin-High when adminRequests greaterThan X for Y time) → *Impersonation high*	
Predicate 2: *observed*(EVENT-Admin-Low when adminRequests lowerEqualThan Z for T time) → *Impersonation low*	

3.4.2. Definición de políticas y reglas

RiAS define los siguientes campos que deben ser declarados en la política: nombre, propietario, control, tipo, y condiciones de adaptación. Los dos primeros son fácilmente interpretables: el nombre es un identificador de la política, mientras que el propietario es el administrador y encargado de definirla, administrarla y actualizarla. El control hace referencia al punto sobre el que se realizarán las adaptaciones, y las condiciones de adaptación son los predicados que definen los parámetros y umbrales que se han de medir, así como las reglas que se dispararán en cada caso. El tipo de la política, por su parte, referencia a cómo se realizará la adaptación: de manera reactiva, al observar cambios producidos en el contexto, o de manera predictiva, al tratar de predecir y anticiparse a dichos cambios.

Utilizando esta información, se puede completar la información definida en la Tabla 3.2, añadiendo al administrador de la plataforma web como propietario, y escogiendo el tipo reactivo, pues las adaptaciones se ejecutarán cuando exista la sospecha de que se está produciendo un ataque de suplantación al administrador (véase Tabla 3.4).

En el caso de las reglas, además del nombre y propietario, se ha de definir la programación temporal, la categoría y los controles que realizar. La programación

Tabla 3.5: Reglas del escenario 1 a alto nivel

Name: Impersonation high	*Owner*: Ecommerce website admin
Timing:	
Event-driven: observed(EVENT-Admin-High)	
Category: Behavioural	
Controls:	
Action: 2FA *enable*()	
Artefact: *use*(2FA configuration file)	
Name: Impersonation low	*Owner*: Ecommerce website admin
Timing:	
Event-driven: observed(EVENT-Admin-Low)	
Category: Behavioural	
Controls:	
Action: 2FA *disable*()	
Artefact: *use*(2FA configuration file)	

puede ser periódica si se realiza en momentos concretos repetitivamente, basada en eventos si se ejecuta al observar los hechos definidos, o bajo demanda si únicamente se efectúa cuando se solicita. Por otra parte, la categoría hace referencia a qué se adapta, y puede ser paramétrica si se realizan cambios en la configuración, arquitectónica si se hacen cambios estructurales, o conductual si se modifica la forma de uso. Por último, los controles son los duplos de acciones realizadas y artefactos utilizados (APIs, interfaces, *middleware*, etc.).

En el primer caso, ambas reglas son basadas en eventos, pues se deben ejecutar cuando se detecten los hechos definidos en la política (cuando la cantidad de peticiones supere o no el umbral definido dentro de una ventana temporal fija), y la categoría será conductual, pues se modificará la forma en que los usuarios interactuarán con el panel de administración. Los controles, que son tanto activar como desactivar el segundo factor de autenticación, comparten el artefacto: el fichero de configuración del servidor web. Estas dos reglas del primer escenario se encuentran definidas en la Tabla 3.5.

Para completar la Tabla 3.3 con la información del segundo escenario, se ha escogido al administrador de seguridad como propietario y se ha vuelto a establecer el tipo reactivo, ya que la adaptación se realizará cuando se sospeche que se están realizando varios cambios de precio abusivos (véase Tabla 3.6).

Las reglas del segundo escenario (véase Tabla 3.7) son muy similares a las del primero. El único parámetro que varía es la categoría, que pasa a ser paramétrica,

Tabla 3.6: Política del escenario 2 a alto nivel

Name: Catalog integrity	*Owner*: Security admin
Control: WAF	*Type*: Reactive
Adaption conditions:	
Predicate 1: *observed*(EVENT-DB-High when abusiveDiscounts greaterThan X for Y time) → *Discount high*	
Predicate 2: *observed*(EVENT-DB-Low when abusiveDiscounts lowerEqualThan Z for T time) → *Discount low*	

Tabla 3.7: Reglas del escenario 2 a alto nivel

Name: Discount high	*Owner*: Security admin
Timing:	
Event-driven: observed(EVENT-DB-High)	
Category: Parametric	
Controls:	
Action: WAF *apply*(restrictive mode)	
Artefact: *use*(WAF configuration API)	
Name: Discount low	*Owner*: Security admin
Timing:	
Event-driven: observed(EVENT-DB-Low)	
Category: Parametric	
Controls:	
Action: WAF *apply*(soften mode)	
Artefact: *use*(WAF configuration API)	

pues en este caso únicamente se realizan cambios en la configuración del WAF, en lugar de en la arquitectura de la aplicación, como sucedía anteriormente.

Algoritmo 3.2 Flujo del escenario 2

```
1:  procedure ESCENARIO2(Log, prod, win, thres)
2:      end ← now
3:      start ← end − win
4:      group ← filter(Log, start, end)
5:      count ← 0

6:      for i ∈ group do
7:          if prod = i.prod then
8:              disc ← 100 ∗ (i.original.price − i.new.price)/i.original.price
9:              if disc > thres then
10:                 count ← count + 1
11:             end if
12:         end if
13:     end for

14:     if count > 0 AND WAF is not restrictive then
15:         ret ← ADAPTION1()
16:     else if count ≤ 0 AND WAF is restrictive then
17:         ret ← ADAPTION2()
18:     end if
19:     return ret
20: end procedure

21: procedure ADAPTION1
22:     restrict WAF
23:     return true
24: end procedure

25: procedure ADAPTION2
26:     soften WAF
27:     return false
28: end procedure
```

4

Implementación y validación

A lo largo de este capítulo se detalla el desarrollo técnico de la prueba de concepto y la implementación de RiAS en un entorno de comercio electrónico desplegado en AWS. Habiendo descrito en el capítulo anterior la arquitectura de la solución y unos escenarios de adaptación, se tratará de implementar su funcionamiento y demostrar el uso de RiAS como solución de seguridad adaptativa para hacer frente a los desafíos de seguridad previamente expuestos.

Asimismo, se proporciona una guía e instrucciones que permitan adaptar los dos escenarios aquí descritos a otros servicios de AWS, otros casos de uso, otros entornos, así como diferentes CSPs. Los servicios utilizados en este caso son aquellos que se han considerado óptimos para los requisitos establecidos, valorando altamente los costes de despliegue y uso, ya que el presupuesto es muy limitado. Para dicha decisión se ha contactado también con arquitectos técnicos de AWS, que han recomendado diferentes alternativas y validado los productos utilizados en la solución final aquí expuesta.

4.1. Implementación de la aplicación de comercio electrónico en AWS

La plataforma escogida para desarrollar la tienda de comercio electrónico es *PrestaShop* [40], un CMS de código abierto que se ha desplegado utilizando el principal servicio de cómputo que proporciona AWS: *Amazon Elastic Compute Cloud* (Amazon EC2). Este permite poner en marcha instancias virtuales con

diferentes sistemas operativos preinstalados y con recursos variables de CPU y memoria según las necesidades. En este caso, se han utilizado dos instancias, una para el servidor de *PrestaShop*, sobre el servidor web Apache, y otra para el motor de la base de datos, MySQL en este caso. Las instancias EC2 hacen uso de otro servicio, *Amazon Elastic Block Store* (Amazon EBS), que provee el almacenamiento persistente de la información contenida en las máquinas virtuales.

Al ser una prueba de concepto, se han desplegado todos los recursos en la misma región geográfica, zona de disponibilidad y red *Virtual Private Cloud* (VPC). Sin embargo, se ha creado un grupo de seguridad diferente para cada instancia virtual, lo que permite limitar la superficie de exposición al realizar una configuración diferente de puertos abiertos al exterior, restringiendo la comunicación entrante a únicamente aquellos necesarios por cada servicio, que serán el 443 para el protocolo web HTTPS en el caso de *PrestaShop*, y 3306 para MySQL, ambos sobre el protocolo TCP. Para labores de mantenimiento de las máquinas, se puede utilizar el protocolo SSH, en cuyo caso se deberá permitir el acceso también al puerto 22 desde la IP de origen desde la que se vayan a realizar las tareas.

La Figura 4.1 muestra esta distribución de los servicios escogidos para realizar el despliegue de *PrestaShop* en AWS, así como la configuración de región, zona de disponibilidad, red VPC y grupos de seguridad.

A la infraestructura mostrada en la Figura 4.1 hay que añadir los controles de seguridad requeridos por los escenarios descritos: el segundo factor de autenticación y el WAF. Como 2FA se ha escogido una aplicación de código libre llamada `apache_2fa` que proporciona esta funcionalidad para Apache utilizando aplicaciones móviles de contraseñas temporales de un solo uso (*Time-based One-Time Password*, TOTP) [43]. En el caso del WAF, se ha optado por utilizar el propio de Amazon, AWS WAF, para integrar el máximo número posible de elementos de su catálogo y así evitar el uso de soluciones instaladas en máquinas virtuales, que se asemejaría en gran medida a un escenario tradicional. Sin embargo, este servicio requiere de al menos uno de los siguientes para funcionar:

- **Amazon CloudFront**: una red de entrega de contenido (*Content Devilery Network*, CDN) que almacena información del servicio web en ubicaciones geográficas próximas a los usuarios, consiguiendo así reducir las latencias.
- **Application Load Balancer**: un balanceador de carga que distribuye el tráfico entrante a diferentes destinos, como instancias EC2, según su estado.
- **Amazon API Gateway**: un servicio administrado de gestión de APIs, incluyendo su publicación, mantenimiento, monitorización y protección a escala, y acepta y procesa las distintas llamadas a la API.
- **AWS AppSync**: un servicio administrado de desarrollo de APIs con GraphQL que conecta de manera segura los diferentes orígenes de datos y escala de manera flexible el motor de ejecución de API GraphQL.

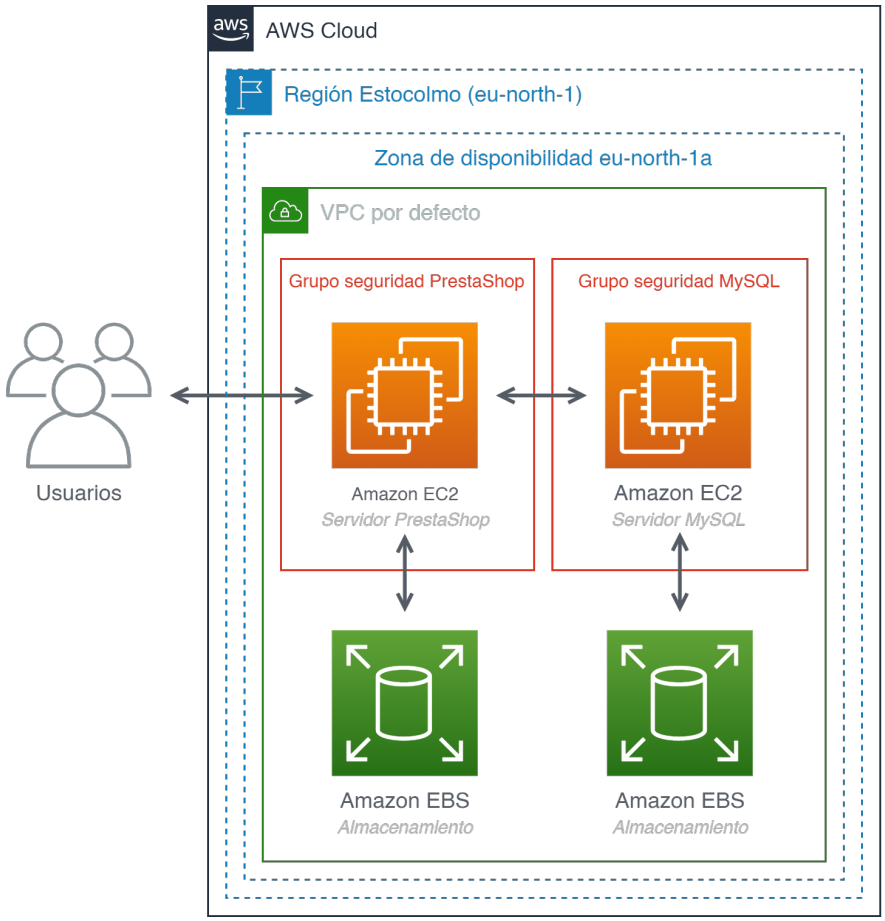


Figura 4.1: Infraestructura básica de la tienda en AWS

De entre los anteriores servicios, se ha escogido el primero, Amazon CloudFront, ya que es óptimo para un servicio web como la tienda electrónica, pudiendo así entregar con baja latencia los diferentes contenidos e información a los clientes. Además, Amazon CloudFront incluye el servicio de seguridad AWS Shield Standard, que proporciona protección frente a ataques DDoS, sin ningún cargo adicional. Asimismo, la infraestructura definida no requiere balanceado de cargas, y tampoco se va a definir o interactuar con ninguna API, por lo que se han descartado los otros tres. Con esto, se obtiene la infraestructura mostrada en la Figura 4.2, que incluye todos los servicios que conforman la tienda de comercio electrónico y los controles de seguridad.

4.2. Implementación de RiAS en AWS

Para la prueba de concepto se ha decidido implementar RiAS siguiendo el modelo de aplicaciones sin servidor, con el cual el CSP se encarga de toda la infraestructura necesaria para ejecutar las cargas de trabajo, liberando al cliente

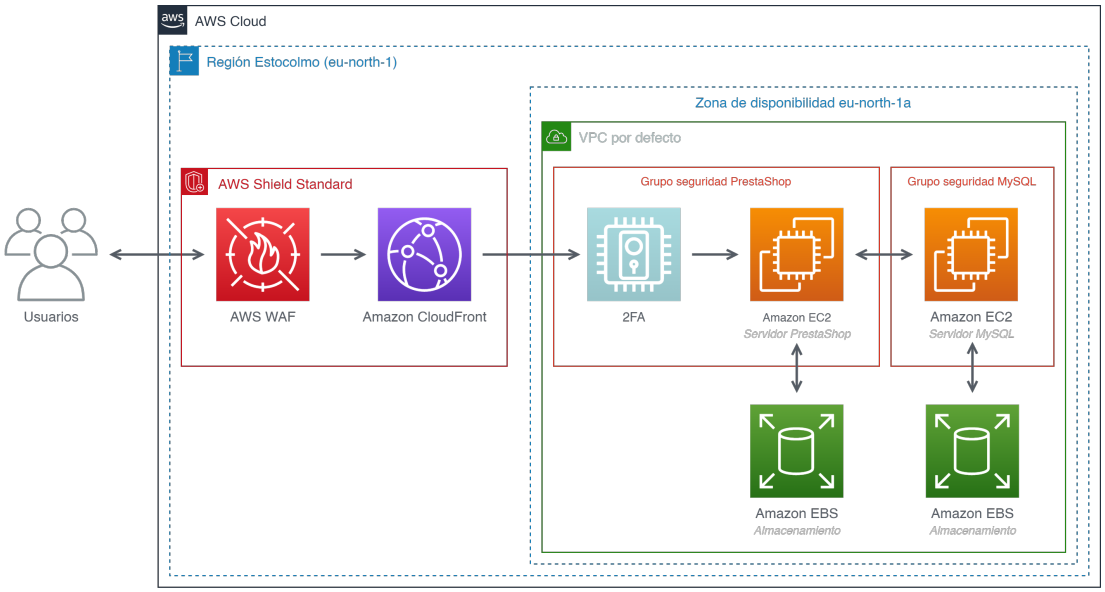


Figura 4.2: Infraestructura completa de la tienda en AWS

de costes y tareas de aprovisionamiento y mantenimiento. Además, como estos proveedores gozan de unos recursos muy superiores a los de cada cliente por separado, pueden asegurar un escalado flexible y alta disponibilidad. Dadas las ventajas que se exponen a continuación, se ha escogido AWS *Serverless Application Model* (AWS SAM), que es la concreción de este paradigma en el entorno de AWS, como modelo para el diseño de la infraestructura de RiAS.

AWS SAM permite a los clientes centrar sus esfuerzos y recursos en su producto o servicio y en la integración de las diferentes capas o servicios de AWS que utilicen. Además, al tener que soportar apenas un mínimo de tareas operativas para poder realizar lanzamientos de su producto o servicio, llegan al mercado de una manera más ágil y veloz que sus competidores. También está presente el pago por uso característico del Cloud, que se ajusta hasta una granularidad de milisegundos y hace que los clientes únicamente paguen por el tiempo de ejecución de código consumido, en lugar de facturar por la capacidad inactiva de los servidores subyacentes (por ejemplo, instancias AWS EC2).

El principal servicio de AWS utilizado para implementar aplicaciones sin servidor es AWS Lambda, una plataforma de computación sin servidor basada en eventos. Esto implica que, en lugar de ejecutar el código de manera continua, únicamente se procesa como respuesta a eventos, tales como llamadas a una interfaz de programación de aplicaciones (*Representational State Transfer Application Programming Interface*, REST API), capturas de datos en sensores del internet de las cosas (*Internet of Things*, IoT), subidas de archivos a servidores web, etc.

Para facilitar la integración y la notificación de los eventos, existen servicios

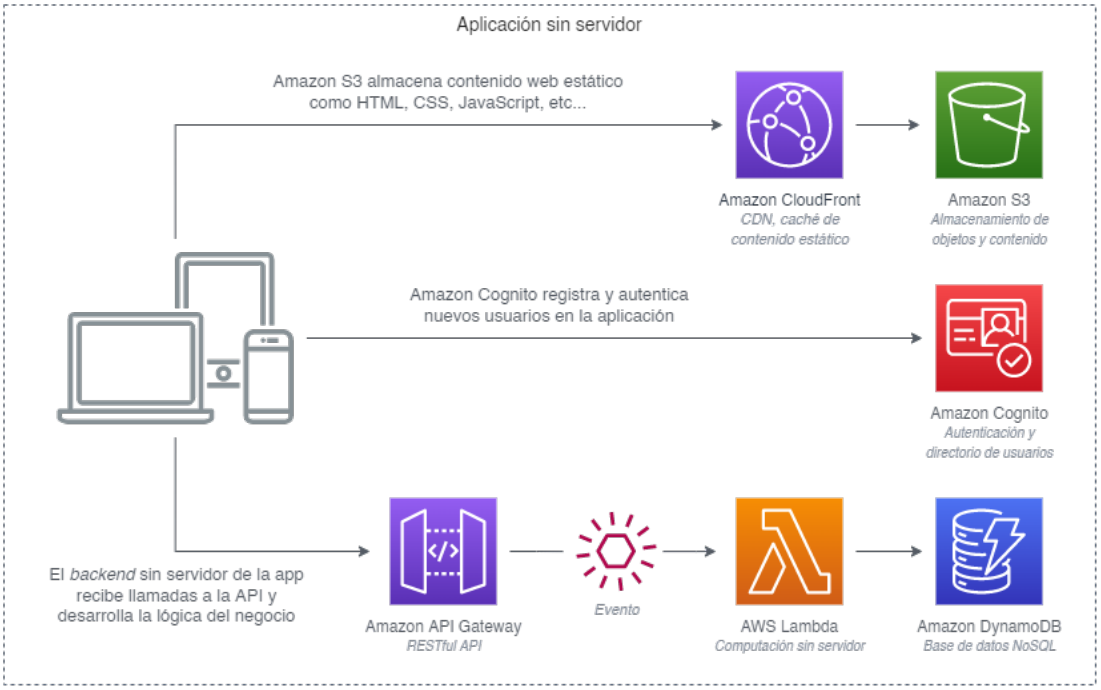


Figura 4.3: Ejemplo de infraestructura de aplicación sin servidor

como Amazon EventBridge o Amazon SNS. El primero es un bus de eventos que envía notificaciones a servicios consumidores de eventos a partir de datos en tiempo real de diversas fuentes. El segundo es el sistema de notificaciones que permite dicha comunicación entre Amazon EventBridge y los consumidores de eventos, como AWS Lambda. Además de la comunicación aplicación a aplicación (A2A), también da soporte a la comunicación aplicación a persona (A2P), todo ello a través de notificaciones *push* móviles, correos electrónicos, SMS, etc.

La Figura 4.3 muestra un ejemplo de aplicación sin servidor en la que todas las tareas de cómputo se realizan utilizando funciones Lambda, que se ejecutan tras las llamadas a la REST API.

Utilizando el catálogo de productos que Amazon propone para desarrollar aplicaciones sin servidor y siguiendo el modelo AWS SAM, se han escogido algunos de los principales para realizar las tres funciones de las capas de RiAS:

- **Medición**: esta labor se desarrolla con Amazon CloudWatch, un servicio de monitorización que permite recopilar y procesar automáticamente *logs* de diferentes servicios. Es necesario instalar en las instancias EC2 un agente proporcionado por AWS que se encarga de leer los registros pertinentes de Apache y MySQL y cargarlos en CloudWatch.
- **Decisión**: de nuevo, se utiliza CloudWatch, ya que también permite establecer acciones automatizadas, conocidas como alarmas, cuando se cumplan

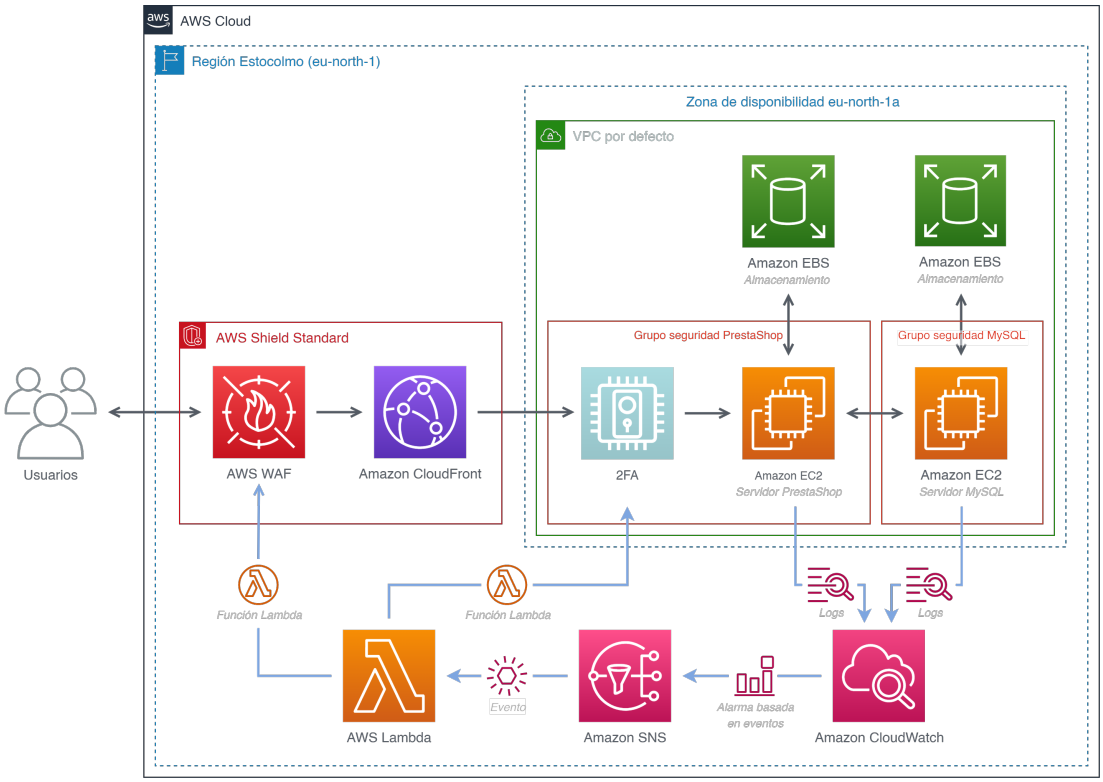


Figura 4.4: Infraestructura completa de la tienda con RiAS en AWS

diversas condiciones en los *logs* que procese. Asimismo, se utiliza el servicio de notificaciones Amazon Simple Notification Service (SNS), donde se publicarán los eventos que se detecten.

- **Adaptación**: se realiza por medio de funciones Lambda, pequeños fragmentos de código Python, en este caso, que se ejecutan en una arquitectura sin servidor con Amazon SNS como desencadenante, es decir, se lanzarán al detectar la publicación de eventos en dicho servicio. Estas funciones serán las encargadas de ejecutar las adaptaciones en el WAF y el 2FA.

Añadiendo estos elementos a la infraestructura existente (véase Figura 4.2), se obtiene la infraestructura de referencia al completo que soporta esta prueba de concepto, recogida en la Figura 4.4, en la cual las flechas de color azul indican el flujo de acciones desempeñadas por RiAS.

4.3. Implementación de los escenarios de adaptación

4.3.1. Pasos *offline*

Una vez definida la implementación de RiAS sobre la arquitectura del sitio de comercio electrónico, es necesario redactar tanto sus reglas como sus políticas con la sintaxis que dicho modelo propone. Las reglas y políticas, que fueron anteriormente descritas en formato de pseudocódigo en la Subsección 3.4.2. Definición de políticas y reglas, deben ser escritas en formato JSON, y definirán el modo de funcionamiento de RiAS y todos los parámetros que influirán en sus acciones y, en definitiva, en su efectividad como solución de seguridad adaptativa. Asimismo, el último paso online recoge el despliegue de los controles y desarrollo de *middleware*, APIs y componentes necesarios para realizar las adaptaciones.

Definición de reglas y políticas del escenario 1

Este escenario pretende mejorar la gestión del riesgo de suplantación del administrador. Para ello, se definen dos eventos disparadores de las adaptaciones de este escenario, `EVENT-AdminRequests-High` y `EVENT-AdminRequests-Low` (véase Código 4.1), que detallan situaciones opuestas. El primero de ellos determina la cantidad de peticiones a la interfaz del administrador que se considera arriesgado superar, y la ventana de tiempo asociada. El segundo, por el contrario, identifica el momento en el que se puede considerar la vuelta a la normalidad, pues la cantidad de peticiones web será inferior o igual a un umbral aceptable en un periodo temporal. Siguiendo la nomenclatura asignada a estas variables en los esquemas de la Tabla 3.4 y la Tabla 3.5, se tienen las siguientes:

- `EVENT-AdminRequests-High`
  - `X`: cantidad de peticiones mínima para considerar un riesgo alto.
  - `Y`: periodo de tiempo en el cual se valora la cantidad `X`.
- `EVENT-AdminRequests-Low`
  - `Z`: cantidad de peticiones límite para considerar un riesgo bajo.
  - `T`: periodo de tiempo en el cual se valora la cantidad `Z`.

Además, hay otros dos eventos que determinan si se activan o no las dos reglas: `EVENT-Enabled2FA` y `EVENT-Disabled2FA`. Como pueden darse repetidamente los dos eventos anteriores, `EVENT-AdminRequests-High` y `EVENT-AdminRequests-Low`,

para evitar intentar la activación (o desactivación) del segundo factor de autenticación cuando ya esté activado (o desactivado), se comprobará previamente si el estado de este control ya es el deseado. De esta forma, con la adición de estos nuevos eventos, se evitarán ejecuciones innecesarias de la adaptación y se eludirán demoras en el reinicio del servidor web, necesario para aplicar el control. Concretamente, únicamente se ejecutará la regla `Impersonation high` cuando se dé el evento `EVENT-AdminRequests-High` y además el 2FA esté desactivado, por el evento `EVENT-Disabled2FA`, y viceversa para el caso contrario.

```
[
    {
        "name": "Admin impersonation",
        "owner": "Ecommerce website admin",
        "type": "reactive",
        "control_id": ["2FA admin"],
        "conditions": [
            {
                "antecedent": [
                    "EVENT-AdminRequests-High",
                    "EVENT-Disabled2FA"
                ],
                "consequent": ["Impersonation high"]
            }, {
                "antecedent": [
                    "EVENT-AdminRequests-Low",
                    "EVENT-Enabled2FA"
                ],
                "consequent": ["Impersonation low"]
            }
        ]
    }
]
```

Código 4.1: Política del escenario 1 en JSON

La definición de las reglas en el formato JSON que propone RiAS es prácticamente una traducción de la especificación a alto nivel de la Tabla 3.5, configurando ambas reglas basadas en eventos, y de categoría conductual, ya que se modifica la manera en la que los usuarios administradores interactúan con el sitio web. Como se detalla más adelante en la Sección 4.3.1. Definición de APIs y *middleware* y la Subsección 4.3.4. Adaptación: AWS Lambda, los artefactos de RiAS son las funciones de AWS Lambda que realizan la adaptación, y sus dependencias.

```
[
    {
        "name": "Impersonation high",
        "owner": "Ecommerce website admin",
        "category": "behavioural",
        "timing": {
            "type": "event-driven",
            "period": null,
            "event-trigger": "EVENT-AdminRequests-High"
        },
        "controls": [
            {
                "control_id": "2FA admin",
                "action": "Enabled",
                "artefact": "AWSLambda-2FA-Enable-Function"
            }
        ]
    }, {
        "name": "Impersonation low",
        "owner": "Ecommerce website admin",
        "category": "behavioural",
        "timing": {
            "type": "event-driven",
            "period": null,
            "event-trigger": "EVENT-AdminRequests-Low"
        },
        "controls": [
            {
                "control_id": "2FA admin",
                "action": "Disable",
                "artefact": "AWSLambda-2FA-Disable-Function"
            }
        ]
    }
]
```

Código 4.2: Reglas del escenario 1 en JSON

Definición de reglas y políticas del escenario 2

Hay que recordar que este escenario tiene que ver con la mejora de la gestión del riesgo de ataque a la integridad del catálogo. Para ello, surge una variable adicional relacionada con los eventos `EVENT-AbusiveDiscount-High` y

`EVENT-AbusiveDiscount-Low`: detallar el concepto de «descuento abusivo». Como se ha mencionado, este hace referencia a un descuento fuera de lo normal que supone un aumento del riesgo sobre la integridad del catálogo de la base de datos. No obstante, es necesario definir el umbral que difiere entre aquellos descuentos que se consideran anormales y aquellos que no suponen un aumento del riesgo. Además, igual que en el escenario previo, se han de definir las cantidades de descuentos y ventanas temporales. Por ello, en este escenario, se deben escoger valores para las siguientes variables:

- General
  - `AbusiveDiscount`: mínimo descuento, en un rango de 0 a 100, considerado excesivo o fuera de lo normal en el contexto de la tienda.
- `EVENT-AbusiveDiscount-High`
  - `X`: cantidad de descuentos mínima para considerar un riesgo alto.
  - `Y`: periodo de tiempo en el cual se valora la cantidad `X`.
- `EVENT-AbusiveDiscount-Low`
  - `Z`: cantidad de descuentos límite para considerar un riesgo bajo.
  - `T`: periodo de tiempo en el cual se valora la cantidad `Z`.

En este escenario, se han vuelto a definir dos eventos que eviten la ejecución consecutiva de una misma regla, para ahorrar tiempo de ejecución y costes en intentar aplicar una configuración que ya se encuentra aplicada. Estos son `EVENT-StrictModeWAF`, que detecta si el WAF ya está activado en modo restrictivo, y `EVENT-NormalModeWAF`, que proporciona el caso contrario, cuando el WAF esté en el modo de funcionamiento normal. Por ello, para activar el modo más restrictivo del WAF, se tiene que dar el evento `EVENT-AbusiveDiscount-High` y además estar el modo normal activado, definido por `EVENT-NormalModeWAF`. Igual sucede para la regla `Discount low`, que únicamente se ejecutará cuando se dé el evento `EVENT-AbusiveDiscount-Low` y el WAF esté ya en modo restrictivo (`EVENT-StrictModeWAF`).

```
[
    {
        "name": "Catalog-integrity",
        "owner": "Security admin",
        "type": "reactive",
        "control_id": ["WAF"],
        "conditions": [
            {
```

```
                "antecedent": [
                    "EVENT-AbusiveDiscount-High",
                    "EVENT-NormalModeWAF"
                ],
                "consequent": ["Discount high"]
            }, {
                "antecedent": [
                    "EVENT-AbusiveDiscount-Low",
                    "EVENT-StrictModeWAF"
                ],
                "consequent": ["Discount low"]
            }
        ]
    }
]
```

Código 4.3: Política del escenario 2 en JSON

De nuevo, la redacción de las reglas para este escenario (véase Código 4.4), son una traducción casi directa de la especificación a alto nivel realizada en la Tabla 3.7. De nuevo, ambas reglas son basadas en eventos, pero en este caso la categoría es paramétrica, pues se modifican únicamente parámetros de configuración del WAF. En la Sección 4.3.1. Definición de APIs y *middleware* y la Subsección 4.3.4. Adaptación: AWS Lambda se encuentra la justificación del hecho de que los artefactos de RiAS sean las funciones de AWS Lambda que realizan la adaptación, así como sus dependencias.

```
[
    {
        "name": "Discount high",
        "owner": "Security admin",
        "category": "parametric",
        "timing": {
            "type": "event-driven",
            "period": null,
            "event-trigger": "EVENT-AbusiveDiscount-High"
        },
        "controls": [
            {
                "control_id": "WAF",
                "action": "RestrictedMode",
                "artefact": "AWSLambda-WAF-Associate-ACL-Function"
            }
        ]
```

```
18      }, {
19          "name": "Discount low",
20          "owner": "Security admin",
21          "category": "parametric",
22          "timing": {
23              "type": "event-driven",
24              "period": null,
25              "event-trigger": "EVENT-AbusiveDiscount-Low"
26          },
27          "controls": [
28              {
29                  "control_id": "WAF",
30                  "action": "NormalMode",
31                  "artefact":
                        "AWSLambda-WAF-Disassociate-ACL-Function"
32              }
33          ]
34      }
35  ]
```

Código 4.4: Reglas del escenario 2 en JSON

Definición de APIs y *middleware*

El último paso *offline* para soportar las tres capas de RiAS es la definición de los *middleware*, APIs y *plugins* necesarios para poder realizar las modificaciones en los controles. Todos estos elementos forman parte de los artefactos de RiAS, que son todos aquellos elementos que relacionan el núcleo de RiAS (las capas de medición y decisión) con los controles donde se producen las adaptaciones.

Para activar y desactivar `apache_2fa` no es necesario definir ningún componente nuevo, tan solo modificar el fichero que define el host virtual de Apache para incluir o no una configuración específica. Dicho fichero, en este caso, es el siguiente:

`/etc/apache2/sites-available/tfgjavshop.conf`

En el caso del WAF, se utilizará la API de AWS para realizar los cambios. En concreto, se harán llamadas a la misma utilizando el *software development kit* que AWS proporciona para Python, denominado `boto3`, que facilita la integración de cualquier código en Python con los servicios de AWS. Se ha escogido este SDK para este lenguaje de programación por los motivos argumentados en la Subsección 4.3.4. Adaptación: AWS Lambda en la cual se detalla también su uso.

Como se deriva de esta explicación y ya se ha introducido de forma breve

previamente, los artefactos en estos dos escenarios son únicamente las funciones Lambda que se describen posteriormente en la Subsección 4.3.4. Adaptación: AWS Lambda, y los elementos que utilizan para realizar sus cometidos, como la conexión SSH, el AWS SDK de Python, etc.

4.3.2. Medición: Amazon CloudWatch

Como se ha mencionado, la función de medición se realizará utilizando Amazon CloudWatch. Para ello, se ha de instalar un agente proporcionado por el equipo de AWS que se encargará de recoger los *logs* que se le indiquen y cargarlos en el servicio de monitorización. La configuración del agente se puede realizar con el script que Amazon proporciona en la siguiente url: `https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/awslogs-agent-setup.py` [44]. En dicho proceso de puesta en marcha, además de indicar los ficheros de registros que se desean cargar y diversas opciones de formato, se deben introducir las claves del usuario de IAM encargado de realizar la transferencia de información hacia CloudWatch. Para ello, es necesario crear un usuario, obtener sus credenciales *AWS Access Key ID* y *AWS Secret Access Key*, y asignarle la política predefinida `CloudWatchAgentServerRole`, que tiene los siguientes permisos, necesarios para el funcionamiento adecuado del agente:

1. `cloudwatch:PutMetricData`
2. `ec2:DescribeVolumes`
3. `ec2:DescribeTags`
4. `logs:PutLogEvents`
5. `logs:DescribeLogStreams`
6. `logs:DescribeLogGroups`
7. `logs:CreateLogStream`
8. `logs:CreateLogGroup`

En las sucesivas Sección 4.3.2. Medición del escenario 1 y Sección 4.3.2. Medición del escenario 2 se disponen los ficheros de registros necesarios para cada escenario, además de prepararlos para poder realizar posteriormente consultas sobre diversos parámetros, que permitan realizar las mediciones y decisiones correspondientes. Una vez preparados los ficheros y configurado el agente, se podrán visualizar todos los registros de los mismos en la interfaz de CloudWatch.

Medición del escenario 1

En el caso del servidor web Apache, por defecto, tan solo se puede visualizar un fichero de registro de errores. Para obtener información sobre todas las peticiones a la ruta del administrador de la tienda, es necesario crear un nuevo *log* personalizado, proceso que únicamente requiere de dos pasos:

1. Definir el formato del *log* y darle un nombre identificativo en el fichero de configuración de Apache (véase Código 4.5), que por defecto se encuentra en la siguiente ruta:

   `/etc/apache2/apache2.conf`

2. Añadir el nombre del *log* personalizado al fichero que define el host virtual (véase Código 4.6), que se puede encontrar en una ruta similar a la siguiente:

   `/etc/apache2/sites-available/prestashop.conf`

```
LogFormat "{ \"time\":\"%{%Y-%m-%d}tT%{%T}t.%{msec_frac}tZ\",
    \"process\":\"%D\", \"filename\":\"%f\", \"remoteIP\":\"%a\",
    \"host\":\"%V\", \"request\":\"%U\", \"query\":\"%q\",
    \"method\":\"%m\", \"status\":\"%>s\",
    \"userAgent\":\"%{User-agent}i\", \"referer\":\"%{Referer}i\" }"
    cloudwatch
```

Código 4.5: Formato del *log* personalizado de Apache

```
<VirtualHost>
    ServerAdmin admin@example.com
    ServerName example.com
    DocumentRoot /var/www/html
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log cloudwatch
</VirtualHost>
```

Código 4.6: Adición del *log* personalizado al sitio web

Medición del escenario 2

Para obtener un registro de los cambios en el catálogo de MySQL que incluya información previa y posterior a la modificación, se puede hacer uso de la utilidad MySQL Enterprise Audit [45], presente únicamente en la versión comercial, aunque puede ser utilizada por un periodo limitado mediante una prueba gratuita ofrecida por Oracle. Concretamente, hay una función de dicha utilidad que permite crear entradas personalizadas en el registro de *log* de auditoría que se defina, `audit_api_message_emit_udf()`, que a su vez forma parte del componente `component_audit_api_message_emit`. La llamada a dicha función se puede realizar utilizando un procedimiento y un disparador definidos en MySQL:

1. Procedimiento que llama a la función `audit_api_message_emit_udf()` con todos los parámetros que se desea que aparezcan en el *log* (véase Código 4.7). Para este escenario, son necesarios los precios anterior y posterior a

la modificación, y también se ha añadido el usuario que realiza los cambios, y el identificador y referencia del producto en cuestión, a fin de facilitar un posterior análisis y seguimiento de los cambios. Además, se ha incluido también el valor del descuento en el propio registro, para evitar tener que realizar su cálculo en pasos posteriores.

2. Un disparador (*trigger*) que se ejecute antes de realizar una modificación en la tabla que contenga la información, que llame al procedimiento previamente definido, habiendo previamente obtenido todos los parámetros deseados, ya que por definición este disparador tiene acceso tanto a la entrada de la tabla previa como a la posterior al cambio (véase Código 4.8).

   La principal tabla que contiene la información sobre los productos y los precios es `prestashop.ps_product`, por lo que ha sido la incluida en la definición del *trigger*. La tabla `prestashop.ps_shop` también recoge los precios de los productos, pero no se ha utilizado por apenas almacenar detalles adicionales de los productos que sean útiles para posteriores investigaciones.

   El disparador debe definirse en la base de datos de la tabla escogida asociada, por lo que se debe declarar en la base de datos `prestashop`.

```
1  DELIMITER $$
2  CREATE PROCEDURE prestashop.audit_api_message_emit_sp(user
       VARCHAR(32), id INT UNSIGNED, ref VARCHAR(64), old_price
       DECIMAL(20, 6), new_price DECIMAL(20, 6), disc INT)
3  BEGIN
4      DECLARE aud_msg VARCHAR(510);
5      select audit_api_message_emit_udf('price_change_trigger',
6              'TRIGGER audit_price_change',
7              'Update product price',
8              'USER', user,
9              'PROD_ID', id,
10             'PROD_REF', ref,
11             'OLD_PRICE', CAST(ROUND(old_price, 6) AS CHAR(10)),
12             'NEW_PRICE', CAST(ROUND(new_price, 6) AS CHAR(10)),
13             'DISCOUNT', disc
14     )
15     into aud_msg;
16 END$$
17 DELIMITER ;
```

Código 4.7: Procedimiento de MySQL que crea la entrada en el log de auditoría

```
DELIMITER $$
CREATE TRIGGER prestashop.audit_update
    BEFORE UPDATE
    ON `prestashop`.`ps_product`
    FOR EACH ROW
BEGIN
    IF OLD.price != new.price THEN
        SET @discount := ROUND(100 * (OLD.price - NEW.price) /
            OLD.price, 0);
        CALL prestashop.audit_api_message_emit_sp(SESSION_USER(),
            NEW.id_product, NEW.reference, OLD.price, NEW.price,
            @discount);
    END IF;
END$$
DELIMITER ;
```

Código 4.8: Disparador de MySQL que llama al procedimiento definido en el Código 4.7

El último paso necesario para obtener las entradas en el registro de auditoría, es establecer los filtros y usuarios que se desean auditar. Para ello, basta con definir un filtro que busque eventos con el valor `message` en el parámetro `class`, y asignárselo al usuario o grupo de usuarios que se deseen auditar, en este caso el usuario `prestashop`, que es el creado con permisos sobre las tablas de la tienda (véase Código 4.9) y el configurado en el panel web de administración. Por ello, será el usuario utilizado al realizar cualquier alteración de la base de datos utilizando la interfaz web de la tienda.

```
SELECT audit_log_filter_set_filter('message_enabled', '{
    "filter": { "class": { "name": "message", "log": true } } }')
    AS 'Result';

SELECT audit_log_filter_set_user('prestashop@%',
    'message_enabled') AS 'Result';
```

Código 4.9: Creación y asignación del filtro de auditoría

4.3.3. Decisión: Amazon CloudWatch

Para el proceso de decisión, se utilizarán dos características de análisis de registros de CloudWatch. En primer lugar, se definirán filtros de métricas, que definen el patrón que se ha de buscar en los *logs* para poder determinar las anomalías. Cabe recordar que en el primer escenario, se debe controlar la cantidad

de peticiones web realizadas a la ruta de administración de la tienda online. Por otro lado, en el segundo, se monitorizarán los descuentos realizados sobre los productos. En ambos casos el filtro se creará en el grupo de registros correspondiente, según se han creado en el paso de medición. Una vez definidos los filtros, se deberán crear alarmas que envíen eventos a Amazon SNS cuando se excedan los umbrales definidos en las reglas de RiAS.

Para la definición de los eventos que evitan la ejecución sucesiva de una misma regla (`EVENT-Enabled2FA` y `EVENT-Disabled2FA` para el primer escenario, y `EVENT-StrictModeWAF` y `EVENT-NormalModeWAF` para el segundo) se activarán y desactivarán las acciones de las diferentes alarmas de CloudWatch. De esta manera, sin tener que eliminar las alarmas, se puede habilitar y deshabilitar su acción, de manera que únicamente se disparen cuando el control esté en el modo contrario al que la alarma pretende configurar. En la implementación inicial, se dejan activadas las alarmas correspondientes con las anomalías y se deshabilitan las acciones de las alarmas que retoman el funcionamiento habitual, ya que al comienzo se estará cumpliendo dicho umbral normal. Y en pasos posteriores, durante la implementación de la adaptación, se activarán y desactivarán mediante llamadas a la API de AWS las reglas contrarias. Así, por ejemplo, se evitará que se dispare la regla que causará la activación del 2FA cuando ya esté previamente activado, y de igual manera ocurrirá con el resto de casos.

Decisión del escenario 1

Para el primer escenario, el filtro tan solo debe seleccionar los registros que contengan la ruta de administración en la petición web. El sitio web que crea PrestaShop es de una sola página (*one-page*), lo que significa que todo el contenido se sirve desde una página, `index.php` en este caso, y la diferente información se muestra modificándola utilizando lenguaje PHP. Por ello, filtrar dicha web es tan simple como realizar la siguiente consulta, utilizando el comodín por si acaso hubiese alguna otra página PHP diferente dentro del subdirectorio del administrador:

```
{ $.request= ‘‘/prestashop_root/admin242r3qua3/*.php‘‘ }
```

Como se desea que la métrica se incremente en una unidad por cada registro seleccionado, el «Valor de métrica» será 1, mientras que el «Valor predeterminado», utilizado cuando no ningún *log* cumpla el filtro, será 0.

Una vez definida la métrica en CloudWatch, es momento de crear las alarmas, que corresponden a las reglas de RiAS. Para ello, tan solo hay que escoger la métrica creada anteriormente y la lógica que hará disparar la alarma (parámetros `X` y `Z` de la regla), y definir la ventana de tiempo apropiada (parámetros `Y` y `T`). CloudWatch ofrece un tipo de límite llamado «Detección de anomalías» para detectar de manera autónoma las desviaciones del comportamiento habitual y

disparar la alerta en dicho momento. Sin embargo, las reglas de RiAS definidas contemplan un valor estático, por lo que se cogerá el tipo de límite del mismo nombre. Para la regla *Impersonation high*, se establecerá que el filtro supere una cantidad de `X` peticiones, y para la regla *Impersonation low*, un valor igual o inferior a dicho umbral (parámetro `Z` = `X`).

Decisión del escenario 2

En el segundo escenario, el filtro debe seleccionar únicamente las entradas del *log* que correspondan a actualizaciones de precios, ya que pueden encontrarse líneas con otro tipo de información de auditoría como puestas en marcha y paradas del motor de base de datos, conexiones, otros disparadores, filtros de otros escenarios de monitorización, etc. Para ello, se puede buscar la cadena de texto dispuesta en el procedimiento definido (véase Código 4.7) que identifica el evento: `‘‘Update product price‘‘`. Además, el filtro ha de seleccionar únicamente aquellas modificaciones en las que el descuento, incluido en el *log*, sea mayor al umbral definido por el parámetro `AbusiveDiscount` de las reglas. El siguiente ejemplo del filtro refleja un valor de 30 para este parámetro:

```
{ $.message_data.message = ‘‘Update product price‘‘ &&
  $.message_data.map.DISCOUNT > 30 }
```

Paso seguido, se han de definir las alarmas de CloudWatch implementando la especificación de las reglas de RiAS, de nuevo utilizando valores estáticos. En este caso, se disparará la regla *Discount high* cuando se realicen más de `X` descuentos abusivos en un periodo inferior a `Y`, y la regla *Discount low* cuando se realicen menos de `Z` descuentos abusivos en un plazo `T`, recomendablemente mayor que el plazo `Y`, para dar tiempo a que se realice la investigación oportuna del incidente.

4.3.4. Adaptación: AWS Lambda

La última fase de RiAS, la adaptación, se realizará utilizando el servicio de computación sin servidor AWS Lambda. Es compatible con hasta 7 lenguajes de programación: Java, Go, PowerShell, Node.js, C#, Ruby y por último Python, que ha sido el escogido en este caso por su simplicidad para realizar las tareas necesarias para esta prueba de concepto.

Para cada escenario se dispondrán dos funciones Lambda diferentes, una para ejecutar cada sentido de la adaptación. Ambas serán disparadas por las notificaciones de los temas de Amazon SNS que las diferentes alarmas de CloudWatch irán emitiendo cuando se den las condiciones. Además, es conveniente recordar que cada función deberá desactivar las acciones de la alarma de CloudWatch que causó su ejecución, y activar las de la alarma opuesta. De esta manera, se asegura

que no se emitirán alarmas iguales de forma sucesiva, lo que causaría varias ejecuciones de funciones Lambda sin efecto real sobre la infraestructura, aumentando los costes de manera innecesaria.

Adaptación del escenario 1

Como ya se adelantó en la Sección 4.3.1. Definición de APIs y *middleware*, para lograr esta adaptación hay que modificar el archivo de configuración del host virtual que da soporte a la tienda, `tfgjavshop.conf`. Para ello, se han creado dos archivos de configuración adicionales a partir del principal, uno que incluye el uso de la utilidad `apache_2fa`, `tfgjavshop.2fa.conf`, y otro que no, `tfgjavshop.no2fa.conf`. Para proceder, tan solo hace falta copiar el contenido de una de estas dos configuraciones adicionales, la que corresponda en cada momento, al fichero utilizado por el sitio web de la tienda. De esta manera, tras reiniciar el servidor web Apache, se conseguirá aplicar la nueva configuración. Por tanto, se operará con los siguientes tres ficheros:

```
/etc/apache2/sites-available/
├── tfgjavshop.conf ..... Configuración utilizada por Apache
├── tfgjavshop.2fa.conf ..... Configuración con apache_2fa activado
└── tfgjavshop.no2fa.conf ..... Configuración con apache_2fa desactivado
```

Para poder realizar estos cambios, se utiliza el protocolo de gestión remota SSH. Para este propósito se ha creado un nuevo usuario en la instancia EC2 del servidor web, a fin de poder asignarle únicamente los permisos necesarios y realizar un control de cambios y auditoría del mismo, por si hubiese cualquier tipo de suplantación o mal uso del mismo. Este usuario se autenticará utilizando un par de claves SSH, pero necesitará también hacer uso de la contraseña para elevar privilegios y poder realizar la copia de los archivos de configuración.

Adaptación del escenario 2

Para este escenario, se ha definido únicamente una lista de control de accesos (*Access Control List*, ACL) web en el *firewall* de AWS. Aunque se podrían haber especificado dos, una más restrictiva y otra menos, no se ha realizado por el alto coste que implica la definición de dichas listas y de las reglas que las integran en AWS WAF. Por ello, en el caso de funcionamiento normal (`NormalMode`), no habrá efecto de ninguna lista del WAF. Por el contrario, en el caso restrictivo (`RestrictedMode`), se aplicará una lista con una única regla que deniegue todas las peticiones web realizadas al CDN provenientes de direcciones IP diferentes a un rango definido. Esta restricción simula ser un bloqueo de todas las peticiones que no se realicen desde las oficinas de la compañía, bien sea de forma presencial o mediante el uso de redes privadas virtuales (*Virtual Private Network*, VPN).

Para poder realizar esta modificación, se ha utilizado el *software development kit* que AWS proporciona para Python, denominado `boto3`, que facilita la integración de cualquier código en dicho lenguaje de programación con los servicios de AWS. La ACL del WAF se asigna al recurso de CloudFront implementado por delante del *front-end* de la tienda. Es por esto que se ha de utilizar el cliente de Amazon CloudFront del paquete `boto3`. Modificar la configuración del recurso de CloudFront se puede realizar con tres simples pasos:

1. Obtener la configuración actual en formato diccionario, con esta función:

   `cloudfront.get_distribution_config()`

2. Modificar el valor de la clave `DistributionConfig.WebACLId` para asignar la ACL correspondiente: el identificador ARN de la ACL para asignarla o una cadena vacía para retirarla.

3. Actualizar la configuración, utilizando la función siguiente:

   `cloudfront.update_distribution()`

4.4. Validación y evaluación

Una vez completado el despliegue de todos los servicios que soportan RiAS y la implementación de sus tres capas, se han simulado ambos escenarios de adaptación para comprobar la integración entre los componentes es la apropiada y la propuesta actúa correctamente de manera integral. No obstante, cada elemento desplegado en AWS, como la carga de registros en CloudWatch, los condiciones disparadoras de sus alarmas, las funciones Lambda, etc., habían sido probados de manera independiente con éxito.

4.4.1. Configuración de los experimentos

Con el fin de facilitar las pruebas y la simulación de los diferentes escenarios de adaptación, se ha escogido un conjunto de parámetros que difieren de los que serían utilizados en un caso real. No obstante, la configuración en entornos de producción es altamente dependiente del contexto de operaciones de los activos en cuestión, por lo que se ha de realizar una fase previa de análisis que permita definir los parámetros más adecuados.

La instancia Amazon EC2 para el servidor de la base de datos ha sido de tipo `t3.micro`, dotada con 2 vCPU y 1 GB de memoria RAM, por encontrarse dentro de los límites de la capa gratuita de AWS y ser suficiente para la carga de trabajo. Sin embargo, a la instancia de PrestaShop se le ha asignado el tipo `t3.small`,

que dispone de 1 GB adicional de memoria RAM, tras detectar que una instancia `t3.micro` no era capaz de atender una cantidad mínima de peticiones web de un único usuario. En ambos casos, el sistema operativo ha sido Ubuntu 20.04.3 LTS. Como la capa gratuita de AWS proporciona hasta 30 GB de almacenamiento sin coste, se ha asignado a cada instancia un tamaño de 10 GB cada uno, a fin de obtener un margen de seguridad para prevenir alcanzar el límite gratuito con facilidad. Cabe destacar también que todos los recursos se han desplegado en la misma región geográfica, eu-north-1, ubicada en Estocolmo, en la zona de disponibilidad A (eu-north-1a).

La parametrización de las reglas y políticas de RiAS es la que sigue:

- Escenario 1
  - `EVENT-AdminRequests-High`
    - `X`: 20 peticiones.
    - `Y`: 2 minutos.
  - `EVENT-AdminRequests-Low`
    - `Z`: 5 peticiones.
    - `T`: 5 minutos.
- Escenario 2
  - `EVENT-AbusiveDiscount-High`
    - `X`: 3 descuentos.
    - `Y`: 2 minutos.
  - `EVENT-AbusiveDiscount-Low`
    - `Z`: 1 descuento
    - `T`: 3 minutos.
  - General
    - `AbusiveDiscount`: 30 %.

Dada esta configuración, se ha confirmado que la propuesta de RiAS implementada es eficaz para ambos escenarios. Por un lado, se ha simulado el comportamiento de un administrador que realiza más de 10 peticiones a la ruta de administración en menos de 2 minutos, y se ha comprobado cómo se activa el segundo factor de autenticación, así como se desactiva en el caso opuesto. Para el segundo escenario, se han realizado los descuentos estando ya autenticado como administrador para facilitar la ejecución. Esta situación de riesgo o ataque podría ser consecuente de otro previo, como un robo de credenciales, la explotación de una vulnerabilidad en el proceso de *login*, una elevación de privilegios de un usuario, etc., que propicie los descuentos desmedidos e injustificados.

4.4.2. Evaluación

Tras comprobar que RiAS funciona exitosamente en el entorno dispuesto, se han analizado diferentes parámetros que pueden proporcionar argumentos justificados a favor o en contra del uso del modelo.

Tabla 4.1: Latencias en el escenario 1 de RiAS

		Media	Desv. estándar
Medición	`EVENT-AdminRequests-High`	0,558 s	0,288 s
	`EVENT-AdminRequests-Low`		
Decisión	`EVENT-AdminRequests-High`	67,000 s	25,000 s
	`EVENT-AdminRequests-Low`	138,000 s	101,000 s
Adaptación	`EVENT-AdminRequests-High`	4,000 s	0,098 s
	`EVENT-AdminRequests-Low`	3,900 s	0,141 s

Latencia

El principal interés en relación con soluciones de este tipo es el tiempo consumido en detectar, decidir y realizar la adaptación, ya que la herramienta debe ser capaz de actuar lo más instantáneamente posible para mitigar el riesgo cuando aumenta. Los experimentos realizados se han repetido 10 veces, obteniendo los valores medios y adquiriendo los instantes temporales de cada paso de los registros automáticos que se almacenan de los diferentes servicios de AWS.

La medición recoge el tiempo que transcurre desde que los registros son generados en las instancias EC2 hasta que son cargados a Amazon CloudWatch. Se aporta el mismo valor para las dos adaptaciones de cada escenario, pues la ingesta de *logs* es común a ambas. La latencia en la decisión es el tiempo que transcurre desde que se da la condición de adaptación y se activa la alarma de CloudWatch. En el caso de `EVENT-AdminRequests-Low` y `EVENT-AbusiveDiscount-Low`, en la latencia se ha sustraído el parámetro `T`, la unidad de tiempo que debe transcurrir desde la última petición a la ruta de administración o el último descuento abusivo para dispararse el evento, puesto que este valor no es una demora adicional, sino el tiempo mínimo que ha de transcurrir según indican las reglas de RiAS. Por último, la latencia en la adaptación es el tiempo que tarda en ejecutarse la función Lambda y vuelve a estar activo el servicio.

Como muestra la Tabla 4.1, para el riesgo de suplantación del administrador, las mayores latencias se producen en la capa de decisión, superando el minuto, y con una desviación estándar considerable. La carga de los registros en Amazon CloudWatch, sin embargo, se produce casi en tiempo real, al igual que la adaptación, que apenas toma cuatro segundos en completarse.

En el segundo escenario (véase Tabla 4.2), la latencia en la medición es mayor que en el primer caso. Esto se debe a un pre-procesado que se ha debido

Tabla 4.2: Latencias en el escenario 2 de RiAS

		Media	Desv. estándar
Medición	`EVENT-AbusiveDiscount-High`	7,800 s	0,789 s
	`EVENT-AbusiveDiscount-Low`		
Decisión	`EVENT-AbusiveDiscount-High`	75,300 s	15,727 s
	`EVENT-AbusiveDiscount-Low`	86,900 s	17,381 s
Adaptación	`EVENT-AbusiveDiscount-High`	2,732 s	0,189 s
	`EVENT-AbusiveDiscount-Low`	2,733 s	0,135 s

implementar dentro de la instancia EC2 para adecuar el formato de los registros para su correcta carga en CloudWatch. Asimismo, se ha disminuido ligeramente la latencia en decidir el evento `EVENT-AbusiveDiscount-Low` respecto al evento `EVENT-AdminRequests-Low` del escenario anterior, ya que en este caso se reciben muchos menos registros que analizar, mientras que antes se procesaban muchos *logs* de las peticiones al servidor web. También ha disminuido la latencia en realizar ambas adaptaciones, ya que antes las funciones Lambda debían ejecutar comandos en la instancia EC2 utilizando SSH, mientras que ahora únicamente se realiza una petición a la API de AWS.

Costes

La inversión económica que requiere la propuesta de RiAS desarrollada en AWS puede dividirse en dos categorías: costes fijos y costes variables. Los primeros hacen referencia al coste base que supone tener RiAS desplegado, independientemente de si se realizan o no adaptaciones. Los segundos, por su parte, recogen todos los costes asociados con servicios de AWS que únicamente se emplean cuando se realizan adaptaciones, por lo que su valor mensual dependerá directamente de la cantidad de variaciones realizadas. En los siguientes análisis se detallan también los productos incluidos en el nivel gratuito de AWS, que permite realizar un uso de los mismos sin generar ningún coste en diferentes periodos de tiempo. Todos los recursos gratuitos expuestos a continuación son de carácter mensual e indefinido, por lo que no vencen en ningún momento.

Los gastos fijos, recogidos en la Tabla 4.3, corresponden con el precio de almacenar registros en CloudWatch utilizando su API, y la creación de las alarmas que dispararán las adaptaciones de RiAS. Como los parámetros `Y` y `T` utilizados en las alarmas son múltiplos del minuto, se facturan como alarmas de resolución estándar, que agrupan los datos en periodos de 60 segundos y se incluyen dentro

Tabla 4.3: Gastos fijos mensuales de RiAS en la región eu-north-1 (Estocolmo)

	Capa gratuita	**Capa de pago**
Solicitudes API `PutMetricData`	1 millón de solicitudes	0,00001 USD por solicitud
Registros almacenados	5 GB	0,54 USD por GB
Alarmas de resolución estándar (60 s)	10 métricas	0,10 USD por alarma
Alarmas de alta resolución (10 s)	n/a	0,30 USD por alarma

Tabla 4.4: Gastos variables mensuales de RiAS en la región eu-north-1 (Estocolmo)

		Capa gratuita	**Capa de pago**
Amazon SNS	Notificaciones AWS Lambda	∞	n/a
	Notificaciones email	1.000 de notificaciones	2,00 USD por 100.000 notificaciones
AWS Lambda	Recursos	400.000 GB-segundo	0,0000166667 USD por cada GB-segundo
	Peticiones	1M de peticiones	0,20 USD por 1M de peticiones

de la capa gratuita de AWS. Si se reducen estos parámetros por debajo de dicha cantidad, las alarmas se facturan a una cantidad mayor y sin margen gratuito.

Los gastos variables dependen de los servicios de AWS involucrados en disparar y ejecutar las adaptaciones, que son Amazon Simple Notification Service (SNS) y AWS Lambda, como muestra la Tabla 4.4. El primero de ellos no tiene coste si se utiliza para enviar notificaciones que disparen las funciones Lambda, aunque sí que se aplica un cargo si se activan las notificaciones por email, algo útil para detectar cuándo se efectúan las adaptaciones. Por otro lado, las funciones Lambda de RiAS requieren muy poca capacidad de cómputo, por lo que funcionan correctamente con el mínimo de memoria posible, 128 MB, y no requieren de concurrencia, lo que genera unos costes mínimos si se excede la capa gratuita.

Escalabilidad

Para trasladar esta prueba de concepto a un entorno de una tienda en línea en producción, tan solo se han de recopilar los registros necesarios, crear nuevas métricas y reglas de Amazon CloudWatch y definir los artefactos que se precisen para realizar las adaptaciones. Como la capacidad de cómputo necesaria para RiAS es proporcionada en su totalidad por AWS gracias a haber empleado el modelo AWS SAM, se garantiza un escalado rápido en función de la demanda. No obstante, al cambiar el contexto de operaciones, se han de revisar los parámetros definidos en las reglas y políticas de RiAS y modificarlos si es necesario. También se ha de prestar atención a posibles cuellos de botella que se puedan producir en los controles de seguridad donde actúen los artefactos.

Usabilidad

Adaptar las reglas y políticas de RiAS a otros escenarios de adaptación o casos de uso carece de complejidad, disponiendo de la definición de RiAS (véase [2]) y los ejemplos propuestos en este trabajo. La Apéndice C. Configuración de RiAS muestra los diferentes valores posibles para cada uno de los parámetros con el fin de facilitar la elección. Además, se deben modificar los eventos antecedentes y consecuentes de las políticas, causando una alteración imprescindible también en las reglas.

Flexibilidad

En el proceso de replicar esta prueba de concepto en otro entorno o para otros escenarios, una vez actualizadas las reglas y políticas, pueden surgir varias dificultades. La más habitual sucede al de cargar los registros de datos en Amazon CloudWatch, principalmente por el formato de los mismos, algo que se suele poder resolver cambiando la configuración del sistema de medición, o desarrollando un breve *script* que pre-procese los *logs*, dando lugar a un nuevo fichero de registros que será el empleado para la capa de medición de RiAS. En ocasiones, además, se deberán realizar modificaciones adicionales a los registros para incluir los parámetros que proporcionen la información necesaria en la capa de decisión. También se debe prestar atención a las APIs, interfaces, *middleware*, etc. que permitan efectuar las adaptaciones en los controles de seguridad, y definir el *plugin* correspondiente si no se dispone de ningún elemento de estas características.

5

Conclusiones y trabajos futuros

En este capítulo se detallan las conclusiones derivadas de la realización de este trabajo y propuestas de mejora que podrían ser desarrolladas en el futuro.

5.1. Conclusiones

A lo largo de este trabajo se han analizado los retos de seguridad que surgen en entornos novedosos, como es la computación en la nube, y se ha tratado de plantear una posible solución en el campo de la seguridad adaptativa, haciendo uso de un modelo innovador y reciente para desarrollar una prueba de concepto sobre una arquitectura web de tres capas.

Ha quedado de manifiesto que las medidas de seguridad tradicionales no pueden ser empleadas en este tipo de entornos sin ningún tipo de adaptación, tal cual se aplican en los contextos habituales. Esto es así por el cambio en la propiedad y ubicación física de los sistemas, que da lugar a modelos de responsabilidad compartida, y la elasticidad y estructura cambiante que caracterizan al *Cloud*. La búsqueda de nuevas técnicas con las que proteger los activos lleva a la definición de la seguridad dinámica, explorada en este trabajo, que aporta una numerosa cantidad de ventajas, resultando muy adecuada para el caso que compete.

El diseño, implementación y despliegue de la prueba de concepto ha permitido cumplir notablemente los objetivos, resaltando la facilidad de implementar medidas de seguridad adaptativa en la nube en un periodo de tiempo muy reducido,

y facilitando el posterior mantenimiento y la toma de cada una de las decisiones necesarias para su definición y puesta en funcionamiento. Todo esto, sumado a su bajo coste, proporciona unas características que no hacen sino favorecer y alentar a la puesta en producción de medidas de seguridad de estas características.

Asimismo, ha quedado demostrada también la adecuación de estas medidas para arquitecturas web de tres capas, aportando posibles escenarios de adaptación para cualquiera de dichos niveles. En concreto, se han estudiado los sistemas de gestión de contenidos, caracterizados por un tráfico muy variable según el contexto, como por ejemplo, un alto volumen de compras en el periodo de rebajas de una tienda de comercio electrónico, una gran cantidad de lectores en un blog los días que se publican nuevos contenidos, etc.

5.2. Líneas de trabajo futuro

Debido a la constante actualización y mejora de los servicios de los CSPs, es posible que aparezcan nuevos productos o características que permitan perfeccionar o enriquecer la implementación y despliegue de RiAS propuesta, por lo que el catálogo de servicios de los mismos debería ser revisado antes de su desarrollo.

El mismo análisis realizado con AWS y el diseño de RiAS en relación con los servicios que esta nube proporciona, debería ser realizado para otros proveedores de nube pública, especialmente los más extendidos del mercado, como pueden ser Microsoft Azure o Google Cloud Platform. De esta forma, se completaría la propuesta realizada y se permitiría su migración a estos otros proveedores, muy comunes también hoy en día en entornos de producción. Asimismo, podrían plantearse escenarios de adaptación adicionales, así como estudiar otros CMS y diferentes arquitecturas, con el propósito de extender el análisis realizado y contemplar distintos requisitos que se originen.

Otra posibilidad sería automatizar el despliegue de RiAS en AWS, para incrementar su escalabilidad, lo que permitiría trasladar su uso a entornos con muchos escenarios de adaptación al agilizar su puesta en marcha. Para ello, se puede hacer uso de funciones Lambda, por ejemplo, que reciban como entrada las reglas y políticas definidas en formato JSON, y por medio de llamadas a la API de Amazon creen los recursos necesarios para dar soporte a RiAS.

En pro también de mejorar la escalabilidad del diseño planteado, sería conveniente integrar en el diseño un sistema de gestión de secretos, como AWS Secrets Manager, especialmente a las funciones Lambda que realizan las adaptaciones. De esta manera, se dispondría de un control centralizado de todos los secretos utilizados por dichos fragmentos de código, evitando su definición múltiple como variables de entorno de cada función, y favoreciendo su gestión y eliminación cuando ya no sean necesarios.

Bibliografía

[1] Eurostat, "*Cloud computing services*," Marzo 2022. [En línea]. Disponible en: https://ec.europa.eu/eurostat/databrowser/view/isoc_cicce_use/default/bar?lang=en

[2] M. Calvo y M. Beltrán, "*A Model for Risk-Based Adaptive Security Controls*," *Computers & Security*, vol. 115, p. 102612, Abril 2022. [En línea]. Disponible en: https://www.sciencedirect.com/science/article/pii/S0167404822000116

[3] R. Bala, B. Gill, D. Smith, D. Wright, y K. Ji, "*Magic Quadrant for Cloud Infrastructure and Platform Services*," *Gartner Magic Quadrant*, Julio 2021. [En línea]. Disponible en: https://www.gartner.com/doc/reprints?id=1-271OE4VR&ct=210802

[4] M. A. Khan, "*A survey of security issues for cloud computing*," *Journal of network and computer applications*, vol. 71, pp. 11–29, 2016.

[5] Symantec, "*Internet Security Threat Report*," *Volumen 24*, p. 19, Febrero 2019. [En línea]. Disponible en: https://docs.broadcom.com/doc/internet-security-threat-report-volume-24-en

[6] IBM Security X-Force Threat Intelligence, "*2021 IBM Security X-Force Cloud Threat Landscape Report*," *Special Intelligence Report*, p. 9, 2021. [En línea]. Disponible en: https://www.ibm.com/downloads/cas/WMDZOWK6

[7] AWS Threat Research Team, "*AWS Shield Threat Landscape Report – Q1 2020*," *AWS Security Blog*, 2020. [En línea]. Disponible en: https://aws-shield-tlr.s3.amazonaws.com/2020-Q1_AWS_Shield_TLR.pdf

[8] B. Tokuyoshi. (2017, Octubre) "*Shift Security Spending From Capex to Predictable Opex*." [En línea]. Disponible en: https://www.paloaltonetworks.com/blog/2017/10/shift-security-spending-capex-predictable-opex/ (Último acceso: 2021-10-14).

[9] Red Hat. (2018, Marzo) "*¿Qué es la seguridad de TI: Cloud Security?*" [En línea]. Disponible en: https://www.redhat.com/es/topics/security/cloud-security (Último acceso: 2021-10-14).

[10] A. Singh y K. Chatterjee, "*Cloud security issues and challenges: A survey*," *Journal of network and computer applications*, vol. 79, pp. 88–115, 2017.

[11] M. Taggart, B. Roach, y P. Woods, "*Amazon Web Services: Risk and Compliance*," Marzo 2021. [En línea]. Disponible en: https://docs.aws.amazon.com/whitepapers/latest/aws-risk-and-compliance/aws-risk-and-compliance.pdf

[12] S. Gai, *Building a Future-Proof Cloud Infrastructure: A Unified Architecture for Network, Security, and Storage Services*. Addison-Wesley Professional, 2020.

[13] N. Subramanian y A. Jeyaraj, "*Recent security challenges in cloud computing*," *Computers & Electrical Engineering*, vol. 71, pp. 28–42, 2018. [En línea]. Disponible en: https://www.sciencedirect.com/science/article/pii/S0045790617320724

[14] R. Buyya, S. N. Srirama, G. Casale, R. Calheiros, Y. Simmhan, B. Varghese, E. Gelenbe, B. Javadi, L. M. Vaquero, M. A. S. Netto, A. N. Toosi, M. A. Rodriguez, I. M. Llorente, S. D. C. D. Vimercati, P. Samarati, D. Milojicic, C. Varela, R. Bahsoon, M. D. D. Assuncao, O. Rana, W. Zhou, H. Jin, W. Gentzsch, A. Y. Zomaya, y H. Shen, "*A Manifesto for Future Generation Cloud Computing: Research Directions for the Next Decade*," *ACM Comput. Surv.*, vol. 51, no. 5, Noviembre 2018. [En línea]. Disponible en: https://doi.org/10.1145/3241737

[15] Adaptive Defense, Panda Security, "*¿Qué es Threat Hunting y por qué es necesario?*" Noviembre 2018. [En línea]. Disponible en: https://www.pandasecurity.com/es/mediacenter/adaptive-defense/threat-hunting-por-que-necesario/

[16] A. Weinert, "*Your Pa$$word doesn't matter*," Julio 2019. [En línea]. Disponible en: https://techcommunity.microsoft.com/t5/azure-active-directory-identity/your-pa-word-doesn-t-matter/ba-p/731984

[17] J. Thomas y A. Buck, "*Azure security best practices*," Octubre 2021. [En línea]. Disponible en: https://docs.microsoft.com/en-us/azure/cloud-adoption-framework/secure/security-top-10#11-architecture-establish-a-single-unified-security-strategy

[18] Cloud Security Alliance, "*About CSA*." [En línea]. Disponible en: https://cloudsecurityalliance.org/about

[19] Instituto Nacional de Ciberseguridad, "*CSA Star*." [En línea]. Disponible en: https://www.incibe.es/protege-tu-empresa/sellos-confianza/cloud/csa-star

[20] R. Mogull, J. Arlen, F. Gilbert, A. Lane, D. Mortman, G. Peterson, y M. Rothman, *Security Guidance For Critical Areas of Focus In Cloud Computing v4.0.* Bellingham, WA: Cloud Security Alliance, 2021. [En línea]. Disponible en: https://downloads.cloudsecurityalliance.org/assets/research/security-guidance/security-guidance-v4-FINAL.pdf

[21] J.-M. C. Brook, A. Getsin, G. Jensen, L. Jameson, M. Roza, N. Thethi, A. Kurmi, S. Levy, S. Shamban, V. Hargrave, V. Chin, Z. Lalic, R. Brooks, S. Lumpe, y A. Ulskey, *Top Threats to Cloud Computing: Egregious Eleven.* Bellingham, WA: Cloud Security Alliance, Agosto 2020. [En línea]. Disponible en: https://cloudsecurityalliance.org/artifacts/top-threats-to-cloud-computing-egregious-eleven

[22] J.-M. Brook, S. Field, D. Shackleford, V. Hargrave, L. Jameson, M. Roza, y V. Chin, *The Treacherous Twelve: Cloud Computing Top Threats in 2016.* Bellingham, WA: Cloud Security Alliance, Febrero 2016. [En línea]. Disponible en: https://cloudsecurityalliance.org/artifacts/the-treacherous-twelve-cloud-computing-top-threats-in-2016/

[23] R. Los, D. Shackleford, B. Sullivan, A. Ginsburg, L. JR Santos, E. Scoboria, K. Scoboria, y J. Yeoh, *The Notorious Nine: Cloud Computing Top Threats in 2013.* Bellingham, WA: Cloud Security Alliance, 2013. [En línea]. Disponible en: https://cloudsecurityalliance.org/artifacts/the-notorious-nine-cloud-computing-top-threats-in-2013/

[24] D. Hubbard, M. Sutton, A. Deeba, A. Dancer, B. Shea, C. Balding, D. Hurst, G. Brunette, J. Lee, J. Witty, J. Reavis, J. Howie, J. Zachry, K. Biery, M. Roesler, M. Becker, M. Geide, S. Matsumoto, S. Morrison, W. Thornhill, W. Kandek, A. Reed, D. Cattedu, D. Cullinane, G. Hogben, G. Ollmann, J. Jensen, J. Pennell, N. Puhlmann, y R. Howard, *Top Threats to Cloud Computing V1.0.* Bellingham, WA: Cloud Security Alliance, 2010.

[25] Amazon Web Services (AWS). (2021) "*Cloud Security, Identity and Compliance Products.*" [En línea]. Disponible en: https://aws.amazon.com/products/security/ (Último acceso: 2021-11-02).

[26] H. Kanikathottu, *AWS Security Cookbook: Practical Solutions for Managing Security Policies, Monitoring, Auditing, and Compliance with AWS.* Birmingham: Packt Publishing, Limited, 2020.

[27] Center for Internet Security, "*About us.*" [En línea]. Disponible en: https://www.cisecurity.org/about-us/

[28] C. Spiess, G. Fitzpatrick, A. Pathak, J. Covington, J. Martinez, T. Sandage, M. De Libero, D. Sanoy, G. Frascadore, I. Dragoi, J. Robel, B. Harrison, M. Wicks, A. Sahasrabudhe, J. Phillips, A. Rao, S. Laino, L. Sica, B. Bhat, y N. Gibbon, "*CIS Amazon Web Services Foundations Benchmark*," Center for Internet Security, 31 Tech Valley Drive, East Greenbush, NY 12061, USA, Tech. Rep. v1.4.0, Mayo 2021. [En línea]. Disponible en: https://www.cisecurity.org/benchmark/amazon_web_services/

[29] N. Poolsappasit, R. Dewri, y I. Ray, "*Dynamic Security Risk Management Using Bayesian Attack Graphs*," *IEEE transactions on dependable and secure computing*, vol. 9, no. 1, pp. 61–74, Enero 2012.

[30] A. Kayes, W. Rahayu, T. Dillon, E. Chang, y J. Han, "*Context-aware access control with imprecise context characterization for cloud-based data resources*," vol. 93, pp. 237–255, 2019. [En línea]. Disponible en: https://www.sciencedirect.com/science/article/pii/S0167739X18300785

[31] S. Veloudis, Y. Verginadis, I. Patiniotakis, I. Paraskakis, y G. Mentzas, "*Context-aware Security Models for PaaS-enabled Access Control*," in *Proceedings of the 6th International Conference on Cloud Computing and Services Science - Volume 1 and 2*, ser. CLOSER 2016. Setubal, PRT: SciTePress - Science and Technology Publications, Lda, Enero 2016, pp. 202–212. [En línea]. Disponible en: https://doi.org/10.5220/0005918602020212

[32] A. Primo, V. V. Phoha, R. Kumar, y A. Serwadda, "*Context-Aware Active Authentication Using Smartphone Accelerometer Measurements*," in *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. Los Alamitos, CA, USA: IEEE Computer Society, Junio 2014, pp. 98–105. [En línea]. Disponible en: https://doi.org/10.1109/CVPRW.2014.20

[33] Y. Ashibani, D. Kauling, y Q. H. Mahmoud, "*A context-aware authentication framework for smart homes*," in *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*. Los Alamitos, CA, USA: IEEE Computer Society, Abril 2017, pp. 1–5. [En línea]. Disponible en: https://doi.org/10.1109/CCECE.2017.7946657

[34] B. Benmammar, *Intelligent Network Management and Control: Intelligent Security, Multi-Criteria Optimization, Cloud Computing, Internet of Vehicles, Intelligent Radio*. Newark: Wiley - ISTE, Mayo 2021.

[35] M. Qiu, S.-Y. Kung, y K. Gai, "*Intelligent security and optimization in Edge/Fog Computing*," *Future generation computer systems*, vol. 107, pp. 1140–1142, Junio 2020.

[36] P. Arcaini, E. Riccobene, y P. Scandurra, "*Modeling and Analyzing MAPE-K Feedback Loops for Self-Adaptation*," in *Proceedings - 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems 2015 (SEAMS)*. Los Alamitos, CA, USA: IEEE Computer Society, 2015, pp. 13–23. [En línea]. Disponible en: https://doi.org/10.1109/SEAMS.2015.10

[37] R. Lingeswara Satyanarayana Tammineedi, "*Integrating KRIs and KPIs for Effective Technology Risk Management*," *ISACA Journal*, vol. 4, no. 22, pp. 19–23, Julio 2018. [En línea]. Disponible en: https://www.isaca.org/resources/isaca-journal/issues/2018/volume-4

[38] WordPress España. (2022) "*Herramienta de blog, plataforma de publicación y CMS.*" [En línea]. Disponible en: https://es.wordpress.org/ (Último acceso: 2022-06-30).

[39] Joomla! Project. (2022) "*Joomla Content Management System (CMS).*" [En línea]. Disponible en: https://www.joomla.org/ (Último acceso: 2022-06-30).

[40] PrestaShop SA. (2022) "*Crea y desarrolla tu tienda online con PrestaShop.*" [En línea]. Disponible en: https://www.prestashop.com/es (Último acceso: 2022-02-24).

[41] Kentico Software. (2022) "*One vendor. Two products. A headless CMS and DXP.*" [En línea]. Disponible en: https://www.kentico.com/ (Último acceso: 2022-06-30).

[42] Shopify. (2022) "*La mejor plataforma de ecommerce.*" [En línea]. Disponible en: https://www.shopify.es/ (Último acceso: 2022-06-30).

[43] I. Temir. (2017, Septiembre) "*Apache two-factor (2FA) authentication.*" [En línea]. Disponible en: https://github.com/itemir/apache_2fa (Último acceso: 2022-03-12).

[44] Amazon Web Services, "*Configure the older CloudWatch Logs agent on a running EC2 Linux instance*," in *Amazon CloudWatch Logs. User Guide*, Julio 2021, pp. 7–10. [En línea]. Disponible en: https://docs.aws.amazon.com/pdfs/AmazonCloudWatch/latest/logs/cwl-ug.pdf#QuickStartEC2Instance

[45] MySQL, *6.4.5 MySQL Enterprise Audit*, Oracle, Junio 2022, version 8.0. [En línea]. Disponible en: https://dev.mysql.com/doc/refman/8.0/en/audit-log.html

[46] P. Cichonski, T. Millar, T. Grance, y K. Scarfone, "*Computer Security Incident Handling Guide*," *Computer Security*, vol. 800-61, Agosto 2012. [En línea]. Disponible en: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-61r2.pdf

Apéndices

A

Desafíos de seguridad en el *Cloud*

En [13] y [14] se plantean tres categorías importantes en las cuales la computación en la nube, dadas sus características propias, origina nuevos desafíos de seguridad. Es importante valorar estos nuevos escenarios, completando el análisis realizado en Sección 2.2. Retos y problemas sin resolver, para poder tener una visión global e integral de los sistemas y poder definir una estrategia de seguridad apropiada.

A.1. Datos

En relación a la protección de los datos, se pueden distinguir dos escenarios diferentes: datos en tránsito y datos almacenados. El primero de ellos es similar a los modelos tradicionales y, dado que cuentan con más investigación en cuanto medidas de seguridad y protocolos de comunicación adecuados, resultan menos atractivos a los ciberdelincuentes. No obstante, en ambos casos hay que prestar atención a la integridad de los mismos, garantizando que únicamente acceden y/o modifican los datos los usuarios autorizados, y a su procedencia, aspecto clave para las investigaciones post-incidente. La Tabla A.1 recoge los principales aspectos que se deben tener en cuenta sobre la seguridad de los datos según [13].

En [14] se plantean dos escenarios reales, relacionados con los descritos en [13], en los que centrar los trabajos futuros en relación a la protección de los datos:

1. En el primero, más simple, el principal problema es garantizar la protección

Tabla A.1: Desafíos relacionados con los datos [13]

Datos almacenados	**Datos en tránsito**	
Ciclo de vida	Recuperación y eliminación	Segregación
Fuga de datos	Copias de seguridad	Cautividad
	Aislamiento	Ubicación

de los datos en el almacenamiento, permitiendo un acceso y manejo eficientes de los mismos. Manifiesta la importancia de asegurar la escalabilidad y el buen rendimiento sin afectar a la funcionalidad cuando se usan soluciones como el cifrado en el lado del cliente. Para ello, debe garantizarse:

a) La integración sencilla con las tecnologías *Cloud* disponibles.

b) El soporte a una gran cantidad de consultas de diversa índole.

c) Que las soluciones escogidas no originen nuevos problemas que puedan causas fugas de datos.

2. En el segundo, los datos deben ser compartidos y manipulados por diferentes usuarios, en ocasiones incluso distribuidos entre CSPs independientes, que origina retos asociados a la integridad de los datos divididos y la privacidad de las consultas de los mismos. Para garantizar el intercambio selectivo de datos y su integridad, es necesario diseñar soluciones que sean capaces de:

   a) Gestionar permisos de escritura y múltiples editores.

   b) Aplicar eficientemente actualizaciones de políticas si se cuenta con diferentes proveedores independientes.

   c) Intercambiar de manera selectiva los datos entre las partes involucradas en cálculos distribuidos.

   Además, se debe garantizar la privacidad en los accesos y consultas a dicha información, lo que supone un verdadero reto por la complejidad computacional necesaria y la escasa tipología de consultas soportadas. Las soluciones que se planteen deben ser eficientes y escalables, a la vez que permiten el acceso concurrente por diferentes usuarios y garantizan que no hay filtraciones sobre la actividad de cada usuario.

A.2. Comunicaciones

La compartición de recursos e infraestructuras entre diferentes máquinas virtuales hace que las comunicaciones se conviertan en objetivo de muchos ataques,

Tabla A.2: Desafíos relacionados con las comunicaciones [13]

Red	Host	Aplicación
Secuestro de prefijos BGP [a]	Virus, troyanos, gusanos, etc.	Envenenamiento de *cookies*
Ataques a DNS [b]	Denegación de servicio	Denegación de servicio distribuida
Reutilización de direcciones IP [c]	Rotura de contraseñas (*cracking*)	Manipulación de campos ocultos
Ataques de *sniffing*	Perfilado de usuario	Ataques de diccionario
	Footprinting	*Google Hacking*
	Accesos no autorizados	Evasión de CAPTCHA [d]

[a] Protocolo de puerta de enlace de frontera (*Border Gateway Protocol*, BGP)
[b] Sistema de nombres de dominio (*Domain Name System*, DNS)
[c] Protocolo de Internet (*Internet Protocol*, IP)
[d] *Completely Automated Public Turing test to tell Computers and Humans Apart*, CAPTCHA

tanto en la infraestructura virtual dentro del mismo CSP como por el uso de los servicios de diferentes CSPs independientes. La Tabla A.2 muestra algunos de los más relevantes clasificados en tres grupos diferentes: a nivel de red, a nivel de host y a nivel de aplicación [13].

A.3. Cómputo (virtualización)

La implementación del concepto de virtualización, que proporciona la abstracción de los recursos físicos, supone un gran desafío y origina una gran cantidad de retos. Prácticamente todo se puede virtualizar, aunque en entornos *Cloud* suele emplearse con aplicaciones, escritorios, redes y servidores y equipos [13]. En la Tabla A.3 se recogen algunos de los retos más destacables, categorizados en tres niveles diferentes:

- **La capa virtual**: engloba diferentes instancias de las máquinas virtuales ejecutándose sobre la infraestructura virtual del CSP.
- **La capa de virtualización**: contiene el hipervisor o monitor de máquina virtual (*Virtual Machine Monitor, VMM*), que permite el despliegue y

Tabla A.3: Desafíos relacionados con la virtualización [13]

Capa virtual	**Capa de virtualización**	**Capa física**
Clonado de MVs [a]	Amenazas en las redes virtuales	Robos o manipulación del hardware
Aislamiento de MVs [a]	Ataques a la integridad del hipervisor	Denegación de servicio distribuida
Migración de MVs [a]	Ataques de MV a MV [a]	Fallos de hardware
Restauración de MVs [a]	Vulnerabilidades en vTPM [b]	Mal uso de la infraestructura
Aumento descontrolado de MVs (*sprawl*) [a]	Secuestro de hipervisor (*hyperjacking*)	Infraestructura sin mantenimiento
Apropiación de recursos (*poaching*)	Inanición con recursos compartidos	Monitorización de estado del hardware
Saltos entre MVs [a]	Introspección excesiva (VMI) [c]	

[a] Máquina Virtual (MV)
[b] Módulo virtual de plataforma de confianza (*Virtual Trusted Platform Module*, vTPM)
[c] Introspección de máquina virtual (*Virtual Machine Introspection*, VMI)

ejecución de las máquinas virtuales sobre el mismo host gestionando los recursos físicos y manteniendo el aislamiento entre ellas.

- **La capa física**: comprende a los diferentes componentes de hardware como la memoria principal, el almacenamiento, la CPU.

B

Mejores prácticas de seguridad en la nube

A continuación se proponen diversas fuentes de información y guías desarrolladas por organismos de reconocido prestigio y aceptación en la industria acerca de la seguridad en entornos de computación en la nube, centrando el análisis en la nube pública de Amazon, AWS, por ser el entorno utilizado en este trabajo.

B.1. Guía *Security Guidance 4.0*

La guía *Security Guidance 4.0*, publicada por la CSA, propone una serie de medidas enfocadas a ayudar a los consumidores de nube pública a mitigar los riesgos de seguridad asociados a la misma, clasificadas en catorce dominios [20]:

Conceptos y Arquitecturas de la Computación en la Nube: proporciona un marco conceptual, define la computación en la nube y el resto de terminología necesaria y detalla la lógica general.

Gobierno y Gestión del Riesgo Corporativo: plantea herramientas de gobernanza y gestión del riesgo, compara las distintas responsabilidades en función de los modelos de servicio y despliegue y propone recomendaciones para cumplir las distintas normativas.

Cuestiones Legales, Contratos y Descubrimiento Electrónico: resalta los problemas legales que pueden aparecer al migrar la infraestructura a la nube, las implicaciones legales de la computación en nubes gestionadas

por un tercero (ya sea nube pública o privada), y los acuerdos de servicio (*Service-level agreements*, SLAs).

Cumplimiento y Gestión de Auditoría: resalta las implicaciones reguladoras al utilizar un servicio o proveedor de servicios en la nube, la asignación de responsabilidades legales entre CSPs y clientes y la obligación de que el proveedor demuestre dicho cumplimiento legal correctamente.

Gobierno de la Información: asesora acerca de los nuevos enfoques de gobierno que deben acompañar a las nuevas medidas técnicas de protección y determina el impacto que supone la computación en la nube en la privacidad, cumplimiento legal y diseño de políticas corporativas que reflejen la complejidad de tratar con terceras partes.

Plano de Gestión y Continuidad del Negocio: explica la diferencia en cuanto al control de los servicios mediante APIs y consolas web en lugar de servidores y cables como en la infraestructura tradicional, remarca la importancia de aplicar los controles de seguridad necesarios a este plano de gestión y el efecto de la responsabilidad compartida sobre la Continuidad del Negocio y la Recuperación ante Desastres (Business Continuity and Disaster Recovery, BC/DR).

Seguridad de la Infraestructura: trata la seguridad para redes y cargas de trabajo virtuales y las consideraciones para la infraestructura subyacente, complementando los estándares existentes para la seguridad de los centros de procesamiento de datos (CPD) tanto de nubes públicas como privadas.

Virtualización y Contenedores: aborda la agrupación de recursos físicos y las dos nuevas capas que la virtualización agrega a los controles de seguridad: la seguridad de la tecnología de virtualización y los controles sobre los activos virtuales.

Respuesta ante Incidentes: identifica las características específicas de la computación en la nube que deben ser consideradas en la respuesta ante incidentes basándose en la guía del NIST 800-61 revisión 2 [46], detalla el ciclo de vida de dicha respuesta y enumera las consideraciones que debe tener en cuenta el equipo de respuesta cuando trabaja en un entorno Cloud.

Seguridad de Aplicaciones: engloba la seguridad por diseño, el modelado de amenazas, la defensa de aplicaciones en producción y la presión a la que se somete a la seguridad en servicios PaaS e IaaS para poder evolucionar de la manera vertiginosa en que lo hace el desarrollo.

Seguridad y Cifrado de Datos: enfocado a la gobernanza de la información y los datos, proporciona un marco general y una ayuda específica para evaluar la seguridad de los datos basándose en el riesgo para adaptar las políticas de seguridad a las peculiaridades de los entornos Cloud,

como el almacenamiento en entornos de terceros (los proveedores) o en un repositorio de recursos compartidos.

Gestión de Identidades, Derechos y Accesos: trata la relación de confianza y delegación de responsabilidades necesaria entre el CSP y el usuario de los mismos en cuanto a la gestión de claves de acceso.

Seguridad como Servicio: valora algunos de los productos *Cloud* más comunes de SecaaS (*Security-as-a-Service*), proporcionados por proveedores dedicados a dichos servicios o por CSPs genéricos, y la gran variedad de tecnologías posibles que se pueden encontrar en el mercado.

Tecnologías Relacionadas: resalta la importancia de prestar atención también a la seguridad de muchos otros servicios interrelacionados con la nube, tanto aquellas tecnologías que dependen de la misma para operar como aquellas que no, pero que se dan comúnmente en dichos entornos, como por ejemplo las redes definidas por software (*Software Defined Networks, SDN*), *Big Data*, entornos del internet de las cosas (*Internet-of-Things, IoT*), los dispositivos móviles o la computación sin servidor.

B.2. Medidas de seguridad en los servicios más relevantes de AWS

Del amplio catálogo de productos que AWS pone a disposición de sus clientes, la siguiente enumeración recoge algunos de los más relevantes y utilizados, junto a las características de seguridad más importantes que deben valorarse a la hora de hacer uso de los mismos [26]:

- **Gestión de cuentas con *Identity and Access Management (IAM)***: se deben administrar de manera adecuada los usuarios, grupos, roles y permisos, así como las políticas que los determinen.
- **Uso de *Simple Storage Service (S3)***: es conveniente hacer uso de listas de control de acceso (*Access Control Lists*, ACLs), políticas de *bucket*, cifrado, control de versiones y replicación entre regiones.
- **Gestión de grupos de usuarios e identidades con *Cognito***: hay que prestar atención a las definiciones de los grupos, registros de usuarios, flujos de autenticación y autorización e inicios de sesión mediante proveedores de identidades federados.
- **Gestión de claves con *Key Management Service (KMS)* y *CloudHSM***: para la gestión de claves criptográficas, KMS utiliza módulos de seguridad hardware (*Hardware Security Module*, HSM), al igual que CloudHSM, que usa módulos exclusivos para una seguridad mejorada.

- **Seguridad de redes con *Virtual Private Cloud (VPC)***: se han de gestionar las subredes privadas y públicas, configurar tablas de enrutamiento y puertas de enlace y utilizar grupos de seguridad y redes de control de acceso a la red (*Network Access Control Lists*, NACLs) para asegurar el tráfico saliente y entrante.
- **Trabajo con *Elastic Compute Cloud (EC2)***: se pueden ejecutar en VPCs concretas, haciendo uso de grupos de seguridad y utilizando el almacén de parámetros y datos de configuración de *Systems Manager*, además de cifrar la información en los discos de almacenamiento, denominados *Elastic Block Store (EBS)*.
- **Seguridad web al utilizar *Elastic Load Balancing (ELB)*, *CloudFront* y *Web Application Firewall (WAF)***: se debe gestionar correctamente el balanceo de cargas, escogiendo el tipo de balanceador adecuado, determinar de manera precisa y apropiada los puntos de presencia (*Points of Presence*, PoPs) en los que se despliega *CloudFront* y utilizar *Shield Standard* para protegerlo de ataques de denegación de servicio distribuida (*Distributed Denial-of-Service*, DDoS) y configurar correctamente los WAFs para proteger las aplicaciones y APIs frente a exploits y bots.
- **Monitorización con *CloudWatch*, *CloudTrail* y *Config***: herramientas clave para monitorizar los recursos y aplicaciones (*CloudWatch*), la actividad de los usuarios y el uso de las APIs (*CloudTrail*) y evaluar las configuraciones de los recursos para realizar auditorías y controle y satisfacer los requisitos de cumplimiento (*Config*).
- **Tareas de cumplimiento con *GuardDuty*, *Macie* e *Inspector***: con una correcta configuración, ayudan a detectar amenazas y situaciones en las que no se cumplan los requisitos establecidos, descubren datos confidenciales que se deban proteger y realizan evaluaciones de seguridad de manera automática, todo ello mediante algoritmos de aprendizaje automático.

B.3. *CIS Benchmark 1.4.0*

Los CIS Benchmarks son una serie de estándares y recomendaciones, disponibles para tecnologías muy diversas, que buscan promover la implantación de medidas de seguridad en entornos de cualquier índole y proveedor. Se basan en el consenso de expertos y guías de configuración y buenas prácticas ampliamente aceptadas por gobiernos, empresas y organizaciones, así como el entorno académico también. La Tabla B.1 muestra algunas de las comprobaciones más relevantes que el CIS propone para entornos desarrollados en Amazon Web Services y los productos involucrados en las mismas.

Tabla B.1: Algunas de las comprobaciones que propone el *CIS Benchmark 1.4.0* [28]

Categoría	**Producto Amazon Web Services**	**Comprobación *CIS Benchmark 1.4.0* [28]**
Gestión de identidades y accesos	AWS IAM	Garantizar que la autenticación multifactor está activada para la cuenta de administrador
		Garantizar que la política de contraseñas impide su reutilización
		Garantizar que las contraseñas no utilizadas en 45 días o más son deshabilitadas
		Garantizar que se han eliminado todos los certificados SSL/TLS expirados almacenados en AWS IAM
	AWS IAM Access Analyzer	Garantizar que IAM Access Analyzer está habilitado en todas las regiones
Almacenamiento	AWS S3	Garantizar que los contenedores utilizan «cifrado en reposo»
		Garantizar que la política del contenedor está configurada para denegar las peticiones HTTP
	AWS EC2	Garantizar que está activado el cifrado del volumen EBS
	AWS RDS	Garantizar que está activado el cifrado del volumen RDS

Continúa en la siguiente página

Tabla B.1: Algunas de las comprobaciones que propone el *CIS Benchmark 1.4.0* [28] (Continuación)

Categoría	**Producto Amazon Web Services**	**Comprobación *CIS Benchmark 1.4.0* [28]**
Recogida de *logs*	AWS CloudTrail	Garantizar que está activada la validación de ficheros de *log* en *CloudTrail*
	AWS CloudTrail y AWS CloudWatch	Garantizar que los registros de seguimiento de *CloudTrail* están integrados en los *logs* de *CloudWatch*
	AWS Config	Garantizar que *AWS Config* está activado en todas las regiones
Monitorización	AWS CloudTrail, AWS CloudWatch y AWS SNS	Garantizar que existe un filtro de métricas en el log y una alerta para los usos de la cuenta de administrador
		Garantizar que existe un filtro de métricas en el log y una alerta para cambios en las tablas de enrutamiento
Redes	AWS VPC	Garantizar que ningún grupo de seguridad permite el acceso desde 0.0.0.0/0 a los puertos de administración de servidores remotos
		Garantizar que el grupo de seguridad por defecto restringe todo el tráfico
		Garantizar que las tablas de enrutamiento para la interconexión de VPCs son de «mínimo acceso»

C

Configuración de RiAS

Las reglas y políticas de RiAS pueden ser configuradas de diferente manera según se desee que se comporte el modelo de seguridad en relación con la infraestructura. La Tabla C.1 recoge todos los posibles valores de estos parámetros que componen la redacción de dichas reglas y políticas [2].

Tabla C.1: Parámetros de las reglas, políticas y lógica de adaptación de RiAS [2]

Elemento	Cuestión	Opción	Descripción
Reglas	¿Qué se mide?	Directo	Valores instantáneos
		Elaborado	Indicadores calculados
		Monitorizado	Eventos desencadenantes
	¿Qué se adapta?	Paramétrico	Cambios en configuración
		Arquitectónico	Cambios estructurales
		Conductual	Cambios en la forma de uso
	¿Cuándo se adapta?	Periódico	Momento especificado
		Basado en eventos	Cuando sucede un evento
		Bajo demanda	Cuando se solicita
	¿Dónde se decide?	Centralizado	Local
		Distribuido	Remoto
		Híbrido	Mezcla local y remoto
Políticas	¿Cómo se adapta?	Reactivo	Al cambiar el contexto
		Predictivo	Al predecir los cambios
	¿Por qué se adapta?	Contexto	P. ej. nuevas amenazas
		Activos	P. ej. nuevas configuraciones
		Objetivos	P. ej. variaciones del riesgo
Lógica de adaptación	¿Con qué se adapta?	Interfaz	Código y lógica a bajo nivel necesario para aplicar las adaptaciones definidas
		API	
		Middleware	
		Plugin	