Universitat Oberta de Catalunya

uoc.edu

Gestión de vulnerabilidades en entornos Cloud

Sergio Fernández Benito
Máster en Seguridad de las Tecnologías de la Información y las Comunicaciones
Seguridad en la internet de las cosas

Manel Jesús Mendoza Flores

02/06/2020

Licencias alternativas (elegir alguna de las siguientes y sustituir la de la página anterior)

A) Creative Commons:

Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada 3.0 España de Creative Commons

Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-CompartirIgual 3.0 España de Creative Commons

Esta obra está sujeta a una licencia de Reconocimiento-NoComercial 3.0 España de Creative Commons

Esta obra está sujeta a una licencia de Reconocimiento-SinObraDerivada 3.0 España de Creative Commons

Esta obra está sujeta a una licencia de Reconocimiento-CompartirIgual 3.0 España de Creative Commons

Esta obra está sujeta a una licencia de Reconocimiento 3.0 España de Creative Commons

B) GNU Free Documentation License (GNU FDL)

Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
A copy of the license is included in the section entitled "GNU Free Documentation License".

C) Copyright

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Gestión de vulnerabilidades en entornos Cloud*
Nombre del autor:	*Sergio Fernández Benito*
Nombre del consultor/a:	*Víctor García Font*
Nombre del PRA:	*Manel Jesús Mendoza Flores*
Fecha de entrega (mm/aaaa):	06/2020
Titulación:	*Máster Universitario en Seguridad de las TIC (MISTIC)*
Área del Trabajo Final:	*Seguridad en la Internet de las Cosas*
Idioma del trabajo:	*Español*
Palabras clave	*Cloud, vulnerabilidades, gobierno*

Resumen del Trabajo (máximo 250 palabras): *Con la finalidad, contexto de aplicación, metodología, resultados y conclusiones del trabajo.*

Muchas organizaciones están llevando su infraestructura, total o parcialmente, a la nube, debido a las ventajas que ofrece.

Sin embargo, no está exenta de problemas de seguridad, al igual que los entornos tradicionales.

En este trabajo se trata la seguridad en la nube desde el punto de vista de configuración de los recursos, un aspecto exclusivo de estos entornos y que, ante el desconocimiento o las prisas por realizar despliegues, puede provocar grandes problemas a la organización.

Se analizará un entorno Cloud con los recursos habituales configurados erróneamente a propósito, se desplegarán herramientas de terceros sobre estos entornos y se analizará la configuración de los recursos, dando como resultado una serie de problemas a resolver con diferentes criticidades.

Estas plataformas de terceros basan su criterio en diferentes estándares de seguridad y legales, utilizándolos como guías a la hora evaluar un entorno desde un punto de vista objetivo y regulatorio.

De estos problemas encontrados se realizará un plan de acción, en el cual se detallará punto por punto qué implicaciones tienen estas vulnerabilidades y su potencial impacto, evidenciando el peligro que entrañan.

Una vez elaborado dicho plan, se llevará a cabo la solución de los problemas encontrados.

Finalmente, se volverá a utilizar las herramientas para escanear de nuevo los entornos utilizados y comparar los resultados obtenidos con el estado inicial de los recursos desplegados, confirmando que los ajustes llevados a cabo han

dado como resultado un número de vulnerabilidades reducido al máximo y, por lo consiguiente, una infraestructura segura a nivel de diseño.

Abstract (in English, 250 words or less):

Cloud Service Providers have become popular the last years, and because of that, it's more and more usual to see organizations migrating their infrastructure to the cloud.

Nevertheless, is not devoid of security flaws, much like their regular counterparts.

The purpose of this work is to secure cloud environments through its resources, configuration and a new and exclusive approach of this solutions. The likes that could cause serious problems given the need of fast deployments or the lack of knowledge of the DevOps team.

A cloud environment with the common resources purposely misconfigured will be analyzed using third-party tools, which will report a series of security problems to be solved with different levels of severity.

These third-party tools are based on different legal regulations, security frameworks and standards, using them as a guide to assess these environments from an unbiased point of view and a proper legal standpoint.

All the reported vulnerabilities will be analyzed and explained, getting as a result, an action plan that will explain each vulnerability that has been found and their risk potential. Showing the unforeseen consequences of not amending them.

Once this problems and vulnerabilities have been amended following the given action plan, the cloud environments will be scanned again and the results will be put against their initial state, and after all the problems have been fixed and being assured that the actions that have been taken have reduced the risks. Therefore, as a result, getting a secure infrastructure by design.

Índice

Lista de figuras

1. Introducción

1.1 Contexto y justificación del Trabajo

En los últimos años se está popularizando en las empresas el disponer de la infraestructura de servidores en un entorno Cloud por las ventajas que supone, principalmente:

- Ahorro de costes por poder dimensionar de manera precisa los recursos
- Mantenimiento incluido
- Alta disponibilidad

Sin embargo, estos entornos no están exentos de problemas de seguridad, tanto por posibles configuraciones inseguras de los activos como por incidentes en los entornos de ejecución (ya sea infecciones con malware, intrusiones en los equipos debido a exploits, etc…).

Existen varias aproximaciones para afrontar los problemas de seguridad en la nube, con las siguientes tecnologías: [1]

- **Cloud Security Posture Management (CSPM)**: Conocido también como gobierno, consiste en protección pasiva basada en la revisión de la configuración de los diferentes elementos existentes en el entorno, con el objetivo de identificar potenciales problemas de seguridad y configuraciones que sobreexpongan activos a la red. Es el enfoque sobre el que se centrará el trabajo al ser un paradigma completamente nuevo y exclusivo de entornos cloud.
- **Cloud Workload Protection (CWP)**: Protección a nivel de las cargas de cómputo (máquinas virtuales, containers y funciones). Esta solución por lo general requiere de la instalación de un agente en los activos a proteger que posteriormente reportan a una consola centralizada. Suelen disponer de funcionalidades muy variadas, como por ejemplo protección antimalware, obtención de CVEs asociados a software y sistema operativo en el activo, vigilancia de la integridad de ficheros o un firewall a nivel de capa 3 o 7 (WAF). Estas soluciones son aplicables generalmente también a un entorno on-premise, por lo tanto, no se profundizará en este aspecto.
- **Cloud Access Security Brokers (CASB)**: Este tipo de solución está enfocada a la prevención de filtración de datos entre aplicaciones cloud (no solo Azure o AWS, suelen incluir aplicaciones SaaS como Office 365, Dropbox, Box, Mailchimp, etc…). Controlan los ficheros que viajan entre herramientas SaaS con el propósito de evitar la filtración de información sensible al exterior. Es un paradigma muy similar a los DLPs (Data Loss Prevention) clásicos. Al no tratarse de vulnerabilidades de seguridad si no de prevención de filtración de datos tampoco se va a abarcar este enfoque en el trabajo.

Estas tecnologías no son del todo excluyentes entre sí. Por ejemplo, en el caso de CSPM y CWPP, es posible controlar las comunicaciones de red; en modo pasivo en el caso de CSPM a través de los Flow logs de los CSPs, y en modo activo, llegando a bloquear comunicaciones en el caso de CWPP:

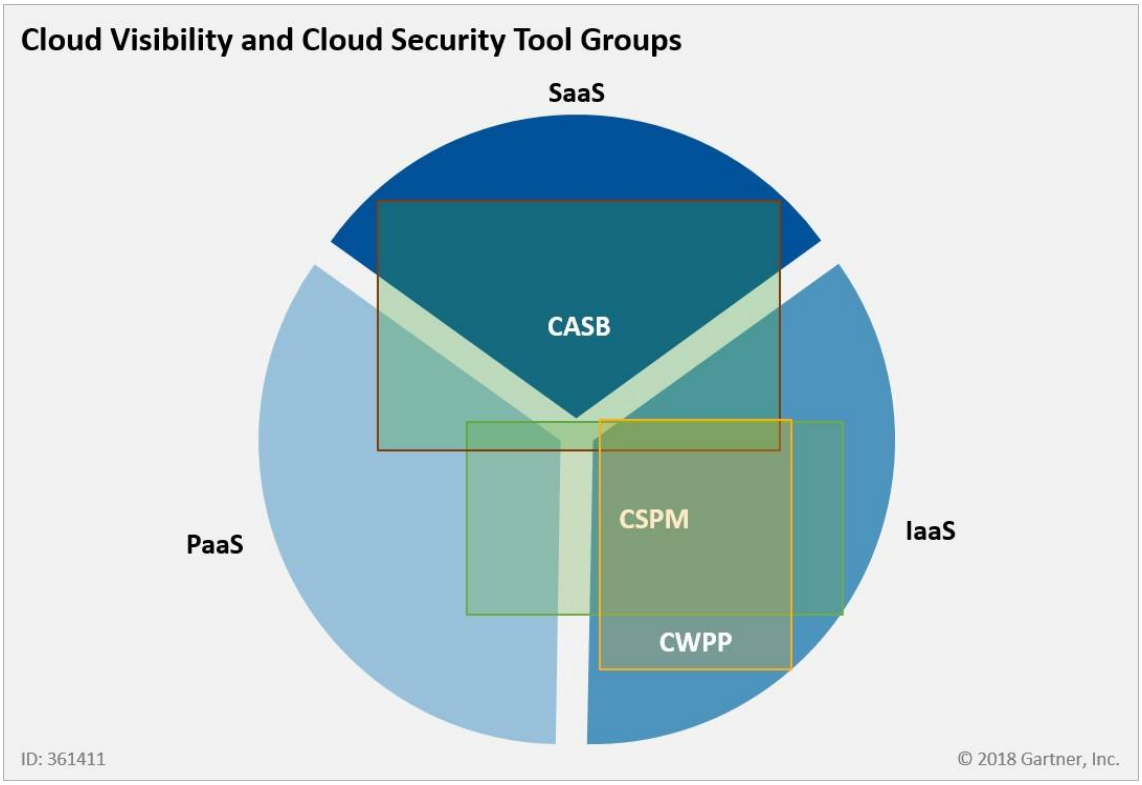


Ilustración 1: Diagrama proporcionado por Gartner donde se muestra que existe superposición entre tecnologías

Como se ha comentado, el trabajo se centrará en las soluciones CSPM, ya que tratan un aspecto exclusivo de Cloud: la configuración de sus recursos.
Ya existen antecedentes de grandes empresas afectadas debido a la configuración errónea o insegura de sus recursos en la nube:

- El banco británico Capital One sufrió en 2019 una filtración de datos que realmente no fue debida a un ataque, sino que se expuso de manera pública y sin autenticación alguna la información contenida en Buckets S3 de AWS, con información sensible de clientes. [2]
- En abril de 2019, Facebook sufrió una filtración de datos que incluía registros de más de 540 millones de usuarios, incluyendo datos personales, fotos e incluso contraseñas. Esta filtración se debió a que un grupo de desarrolladores de la compañía dejó expuesta a todo internet una base de datos en AWS con toda la información sensible, de tal modo que cualquier persona podía acceder a los datos libremente. [3]

Pese a que en ambos ejemplos el CSP involucrado ha sido AWS, Amazon no es responsable de estas filtraciones, ya que se deben a una mala gestión de las herramientas proporcionadas por AWS.
Según el modelo de responsabilidad compartida, Amazon tan sólo provee seguridad de sus propias herramientas (seguridad **de** la nube frente a seguridad **en** la nube), esto es, tanto del hardware sobre el que se despliegan los recursos como el software utilizado para gestionarlos:

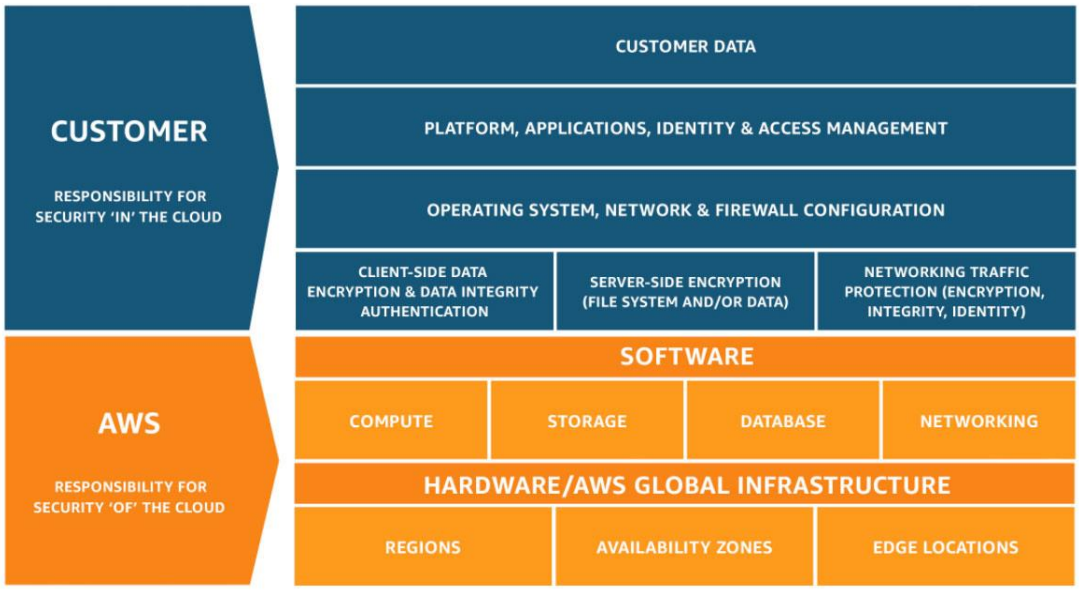


Ilustración 2: Modelo de responsabilidad compartida según Amazon [4]

Ilustración 3: Modelo de responsabilidad compartida según Microsoft [5]

Estos antecedentes demuestran que nos encontramos ante un problema real y que puede afectar en muchos niveles no sólo a las empresas, sino también a los usuarios finales de los servicios de estas, ya que pueden ver como sus datos personales son accesibles para todo el mundo y, en última instancia, podrían filtrarse sus datos bancarios, permitiendo a cualquier usuario malintencionado a obtener rédito económico de la situación.

En un análisis de GlobalDots [6] se detalla que, según Gartner, en el año 2020 el 95% de los problemas de seguridad en Cloud serán resultado de malas configuraciones de los entornos de la nube.

Es por este motivo por el cual las herramientas CSPM cobran un especial atractivo, ya que, como se ha comentado, nos permiten monitorizar la configuración de los recursos de manera centralizada, facilitando la detección de problemas de seguridad tales como [7]:

- Bases de datos o almacenamiento sin cifrar
- Tráfico sin cifrar que puede contener información sensible
- Permisos o roles excesivos
- Ausencia de autenticación en múltiples factores en cuentas críticas
- Configuración de redes pobres o que sobreexponen activos
- Almacenamiento expuesto a internet sin restricciones
- Ausencia de logado de accesos o actividades críticas, tales como creación y borrado de recursos

Además, junto con la visibilidad de todos los recursos, las herramientas CSPM ofrecen funcionalidades adicionales derivadas de la visibilidad del entorno, como, por ejemplo:

- Monitorización de creación y borrado de recursos
- Vigilancia continua de los recursos desplegados (continuous assessment)
- Revisión de la infraestructura con el objetivo de cumplir frente a determinados estándares o reglamentos
- Automatización de las remediaciones a tomar para solucionar los problemas detectados
- Análisis de las plantillas utilizadas por los desarrolladores y modificación de las mismas para que los recursos desplegados sean generados de manera segura por defecto
- Protección de identidad

1.2 Objetivos del Trabajo

Un aspecto donde más se puede profundizar en cuanto a seguridad pasiva y que más nos puede ayudar a evitar incidentes de seguridad es la segmentación de los activos, controlando los Security Groups (elementos de configuración que permiten indicar de manera explícita las comunicaciones permitidas y denegadas) de los siguientes elementos:

- Redes virtuales
- Subredes
- Interfaces de red de máquinas virtuales
- Containers y Kubernetes/OpenShift

De este modo podemos evitar movimientos laterales derivados de intrusiones en equipos de la red, con lo cual se considera como uno de los primeros objetivos para tener en cuenta para poder proteger los activos en una infraestructura Cloud de las vulnerabilidades que pudieran presentar.

Con esto se pretende realizar el hardening (refuerzo de la seguridad) de todos los recursos disponibles en nube, obteniendo un plan de acción por cada entorno y herramienta.

Adicionalmente se trabajará en contrastar las vulnerabilidades encontradas por cada herramienta, así como su plan de acción para detectar posibles discrepancias y mejoras, y permitiéndonos enfocarnos en aquellos tipos de activos que más desprotegidos quedan tras el uso de estas herramientas.

Una vez que se hayan tomado las medidas determinadas por el plan de acción, se volverá a escanear los entornos con las mismas herramientas para verificar que se han solucionado todos los problemas detectados y que se han minimizado los riesgos de seguridad.

1.3 Enfoque y método seguido

El enfoque seguido consiste en el despliegue y uso de herramientas, tanto open source como comerciales, para localizar las vulnerabilidades generadas por una incorrecta configuración de los recursos, así como aplicar un plan de acción para reducir, en la medida de lo posible, estas vulnerabilidades.

Con el fin de mostrar casos de uso lo más reales posible, se desplegará un entorno en dos CSPs (AWS y Azure) con los elementos más habituales que se pueden encontrar en un entorno habitual, el cual se detalla más adelante.

Gran parte de los recursos que se utilizarán serán desplegados con configuraciones incorrectas intencionadamente, de este modo se podrá reflejar problemas de gravedad real que pueden desembocar en diferentes escenarios:

- Exfiltración de datos
- Intrusión en la infraestructura (acceso no autorizado)
- Diferentes ataques web (SQLi, XSS, shellshock, etc…)

Sobre estos entornos se desplegarán los diferentes aplicativos a probar, que harán uso de las APIs propias de cada CSP.

El trabajo se centra en la seguridad desde el gobierno, utilizando herramientas que revisaran la configuración de los diferentes elementos de la nube desde la API del proveedor. Con esta información se elaboraría un plan de acción o procedimiento para asegurar una correcta configuración desde la seguridad pasiva.

Para ello, se ha elaborado el siguiente listado de tareas:
Despliegue de un entorno de pruebas en Azure con los servicios típicos:

- Base de datos
- Servidor web
- Storage Account
- Servidor de backend
- Balanceadores
- Clúster de Kubernetes

Despliegue de un entorno de pruebas en AWS con los servicios típicos:

- Base de datos
- Servidor web
- Bucket S3
- Servidor de backend
- Balanceadores
- Clúster de Kubernetes

Despliegue de la herramienta Cloudsploit de Aqua en estos entornos:

- Obtención de los datos de toda la infraestructura
- Escaneo de los datos obtenidos para buscar vulnerabilidades en la configuración
- Búsqueda de elementos que incumplan estándares comunes
- Definición de plan de acción para solventar los problemas

Despliegue de la herramienta Prisma de Palo Alto en estos entornos:

- Conexión con la plataforma Cloud y aplicación de las reglas asociadas a los estándares comunes
- Recolección de activos que violan políticas y definición de plan de acción para solventar los problemas

Despliegue de la herramienta Dome9 de Checkpoint en estos entornos:

- Conexión con la plataforma Cloud y aplicación de las reglas asociadas a los estándares comunes
- Recolección de activos que violan políticas y definición de un plan de acción para solventar los problemas encontrados

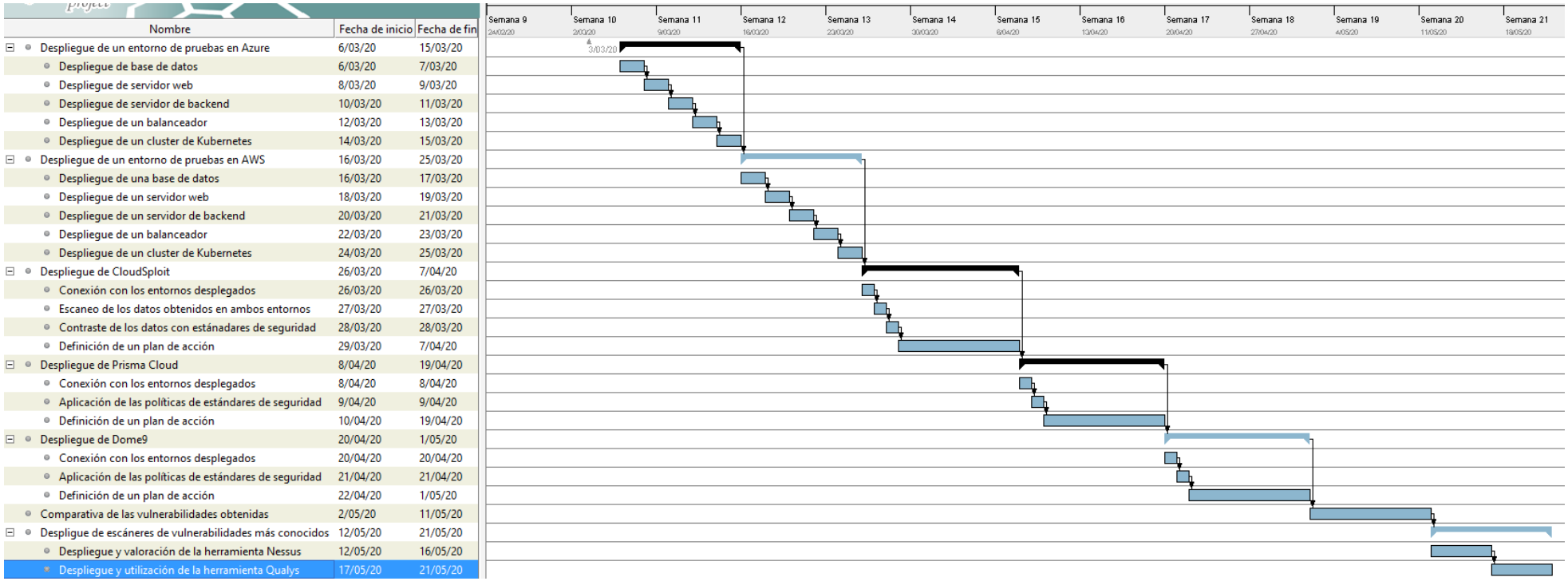
Se contrastará la configuración de los recursos con las directrices proporcionadas por diferentes estándares y leyes, con el fin de utilizar referencias ampliamente aceptadas y asegurar el cumplimiento legislativo en la medida de lo posible:

- **CIS**: Siglas del Center for Internet Security, es una organización sin ánimo de lucro que agrupa el conocimiento de una comunidad global de profesionales de IT que proporcionan guías y controles a seguir para mejorar la ciberseguridad de empresas y servicios. [8]
- **PCI**: Payment Card Industry Data Security Standard, conjunto de medidas para la seguridad de datos de medios de pago definidas por un comité formado por los principales proveedores de tarjetas, American Express, MasterCard y Visa, entre otros. [9]
- **NIST-800**: National Institute of Standards and Technology, organismo estadounidense formado en 1901 y parte del departamento de comercio de Estados Unidos, es un organismo dedicado al desarrollo de guías y estándares en el ámbito tecnológico. Los estándares NIST-800-53 y NIST-800-171 son los más extendidos. [10]
- **RGPD**: Reglamento General de Protección de Datos europeo, de obligado cumplimiento para todos los proveedores de servicios en internet. Las medidas implementadas por los fabricantes se centran en los artículos 25 (Protección de datos desde el diseño y por defecto), 30 (Registro de las actividades de tratamiento), 32 (Seguridad del tratamiento) y 46 (Transferencias mediante garantías adecuadas. [11]

Una vez que se haya obtenido los resultados de las plataformas utilizadas basándonos en estas guías y políticas, se pondrán en común para realizar una comparativa, identificando qué vulnerabilidades no han sido detectadas en todas las plataformas.

Por último, definido el plan de acción se llevarán a cabo las medidas indicadas en el mismo, comprobando en las diferentes herramientas que se han solucionado los problemas detectados y obteniendo, por tanto, la verificación de que el entorno ha sido securizado.

1.4 Planificación del Trabajo



Nombre	Fecha de inicio	Fecha de fin
Despliegue de un entorno de pruebas en Azure	6/03/20	15/03/20
Despliegue de base de datos	6/03/20	7/03/20
Despliegue de servidor web	8/03/20	9/03/20
Despliegue de servidor de backend	10/03/20	11/03/20
Despliegue de un balanceador	12/03/20	13/03/20
Despliegue de un cluster de Kubernetes	14/03/20	15/03/20
Despliegue de un entorno de pruebas en AWS	16/03/20	25/03/20
Despliegue de una base de datos	16/03/20	17/03/20
Despliegue de un servidor web	18/03/20	19/03/20
Despliegue de un servidor de backend	20/03/20	21/03/20
Despliegue de un balanceador	22/03/20	23/03/20
Despliegue de un cluster de Kubernetes	24/03/20	25/03/20
Despliegue de CloudSploit	26/03/20	7/04/20
Conexión con los entornos desplegados	26/03/20	26/03/20
Escaneo de los datos obtenidos en ambos entornos	27/03/20	27/03/20
Contraste de los datos con estánadares de seguridad	28/03/20	28/03/20
Definición de un plan de acción	29/03/20	7/04/20
Despliegue de Prisma Cloud	8/04/20	19/04/20
Conexión con los entornos desplegados	8/04/20	8/04/20
Aplicación de las políticas de estándares de seguridad	9/04/20	9/04/20
Definición de un plan de acción	10/04/20	19/04/20
Despliegue de Dome9	20/04/20	1/05/20
Conexión con los entornos desplegados	20/04/20	20/04/20
Aplicación de las políticas de estándares de seguridad	21/04/20	21/04/20
Definición de un plan de acción	22/04/20	1/05/20
Comparativa de las vulnerabilidades obtenidas	2/05/20	11/05/20
Despligue de escáneres de vulnerabilidades más conocidos	12/05/20	21/05/20
Despliegue y valoración de la herramienta Nessus	12/05/20	16/05/20
Despliegue y utilización de la herramienta Qualys	17/05/20	21/05/20

Ilustración 4: Planificación inicial del trabajo

1.5 Breve sumario de productos obtenidos

Tras realizar la conexión entre los entornos y las herramientas, y estas escanear la infraestructura, se ha obtenido un listado de vulnerabilidades detectadas en el entorno desplegado, tanto en AWS como en Azure.

Estas vulnerabilidades están, asimismo, asociadas a reglamentos comunes y estandarizados, tales como el Reglamento General de Protección de datos, de tal modo que nos permiten conocer el nivel de cumplimiento requerido por cada uno de estos estándares, ayudando a encarar una posible auditoría.

En base a estas vulnerabilidades detectadas se ha llevado un análisis de cada una de ellas, destacando su motivación y los riesgos que implican su no resolución, descubriendo que es extremadamente sencillo descuidar el entorno y dejar expuestos elementos sensibles de la infraestructura como pueden ser las bases de datos, por ejemplo.

Una vez que se ha llevado a cabo este análisis, se ha procedido a resolver los problemas reportados, para ver si las herramientas eran capaces de reflejar estos cambios y demostrar, efectivamente, que nuestros entornos habían mejorado el nivel de cumplimiento por cada uno de los estándares seleccionados.

1.6 Breve descripción de los otros capítulos de la memoria

A continuación, se introduce brevemente el contenido de los capítulos que forman el trabajo:

- **Infraestructura analizada**: En este capítulo se describe detalladamente la infraestructura desplegada, tanto en AWS como en Azure, utilizada para buscar las vulnerabilidades sobre las que trata el trabajo. Se detalla, además, las peculiaridades y particularidades de cada una de las nubes.
- **Plataformas utilizadas**: En este apartado se pasa a valorar las opciones disponibles, tanto comerciales como Open Source, obteniendo una conclusión de cuáles se utilizarán durante el trabajo.
  Además, se entra en detalle sobre cada una de ellas, poniendo en contexto su existencia. Por último, se detalla el funcionamiento de estas, indicando como se obtienen información de cada uno de los proveedores de cloud pública.
- **Resultados Prisma Cloud**: En este apartado se pasa a detallar, en AWS y en Azure, todas las vulnerabilidades obtenidas con Prisma Cloud y un plan de acción para solventarlas.
- **Resultados Dome9**: Este capítulo proporciona la misma información que el anterior, pero, en este caso, con la herramienta Dome9 de Checkpoint.
- **Resultados CloudSploit**: Al igual que los apartados anteriores, se detalla las vulnerabilidades obtenidas con la herramienta, en este caso CloudSploit de Aqua, y un plan de acción para solventarlas.
- **Comparativa de resultados**: Finalmente, una vez que se ha analizado la información proporcionada por cada herramienta y seguido el plan de acción, se pasa a contrastar los resultados obtenidos y destacar las peculiaridades de cada una de las herramientas.

2. Infraestructura analizada

La infraestructura analizada consta de los elementos habituales de una infraestructura tradicional on-premise (máquinas virtuales y balanceadores) junto con servicios PaaS (Platform as a Service) típicos de las nubes, así como plataformas autoescalables (Kubernetes).

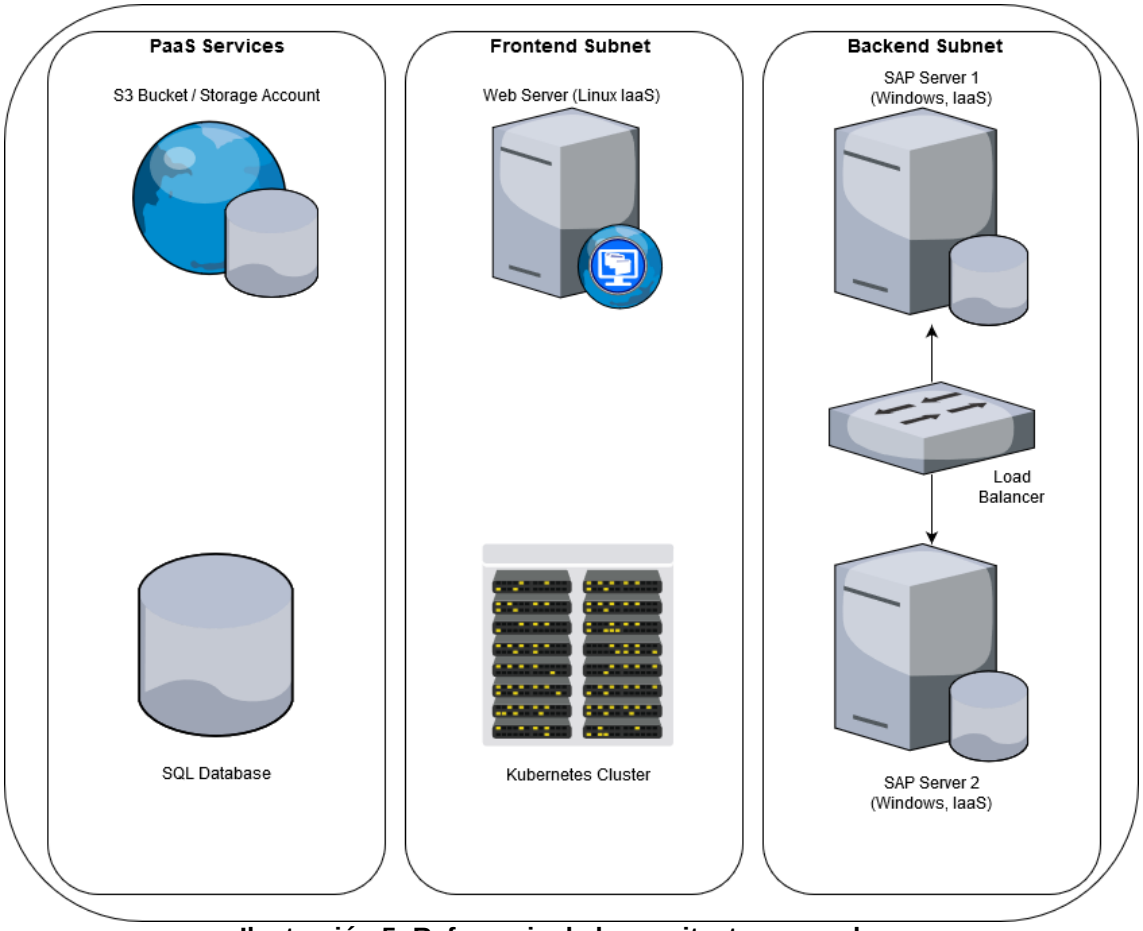


Ilustración 5: Referencia de la arquitectura a evaluar

A continuación, se realiza una equivalencia entre los términos utilizados en AWS y Azure con los conceptos tradicionales:

Concepto	AWS	Azure
Máquina virtual	EC2/Instancia	Máquina virtual
Almacenamiento arbitrario (ficheros)	S3 Bucket	Storage Account (STA)
Clúster de Kubernetes	Elastic Kubernetes Service (EKS)	Azure Kubernetes Service (AKS)
Base de datos	Relational Database Service (RDS)	Base de datos
Balanceador de carga	Elastic Load Balancer (ELB)	Balanceador de carga
Unidad de disco	Elastic Block Store (EBS)	Unidad de disco
Interfaz de red	Elastic Network Interface (ENI)	NIC (Network Interface Controller)

Ilustración 6: Equivalencia de conceptos entre proveedores Cloud

En la infraestructura de ejemplo se contemplan los siguientes elementos:

- **Servidores SAP**. Estas máquinas virtuales basadas en Windows son el resultado de un lift and shift (subir a la infraestructura cloud los servidores virtualizados tal cual existían en el entorno on-premise) y contienen un aplicativo interno habitual, como es SAP, encargado de procesar datos y remitirlos a los clientes (por ejemplo, el departamento contable). Estas máquinas conforman el backend de la empresa y no están expuestas a internet.
- **Balanceador de carga**. Aunque existe la posibilidad de utilizar un dispositivo basado en tecnología de terceros (F5, A10, etc…), tanto Azure como AWS ofrecen elementos de infraestructura gestionados por el proveedor, simplificando la tarea de despliegue y mantenimiento. En este caso, se opta por el balanceador de carga ofrecido por cada uno de los proveedores de nubes públicas.
- **Servidor Web**. Una máquina virtual basada en Linux con un servicio (software) de servidor web corriendo. Esta máquina podría contener la web de la empresa y presumiblemente un aplicativo de e-commerce. Utiliza los servicios de bases de datos PaaS aprovechando las ventajas del pago por uso de los proveedores de nubes públicas. Al ofrecer un servicio web, esta máquina está expuesta al público.
- **Clúster de Kubernetes**. Aunque no es un servicio exclusivo de los entornos cloud, se compenetra muy bien con estos gracias a su paradigma de servicios autoescalables y efímeros, disponiendo únicamente de la cantidad de recursos justa para satisfacer las necesidades en todo momento, sin exceder los recursos reservados y abaratando costes. Este clúster contendría una aplicación más específica que un simple servicio web, como, por ejemplo, un asistente virtual que ayudara a los clientes con la toma de decisiones en un proceso de compra. Podría disponer de contenedores con imágenes enfocadas a machine learning y otros procesamientos de datos.
- **S3 Bucket / Storage Account**. Este servicio exclusive de los entornos cloud permite almacenar datos desestructurados (es decir, ficheros arbitrarios sin relación con las bases de datos) sin necesidad de utilizar tiempo de procesamiento (CPU) para gestionarlos y facilitar su acceso. Se utilizaría para almacenar recursos utilizados por el servidor web (imágenes de artículos, imágenes de perfil, scripts y otros recursos web, etc…) de tal modo que estos no se almacenaran en el disco duro de la máquina virtual, evitando el riesgo de pérdida de datos.
- **Base de datos SQL (PaaS)**. Las bases de datos, al igual que los balanceadores, pueden ser desplegadas con un servidor tradicional o, en cambio, pueden utilizar los servicios del proveedor de la nube, no sólo facilitando el ahorro de costes, si no también replicación, alta disponibilidad o una garantía superior ante pérdidas de datos. Esta base de datos contendría los usuarios del aplicativo web, así como mensajes, artículos, etc… típicos de una plataforma de e-commerce.

El inventario completo de recursos desplegados queda del siguiente modo:

Recurso	AWS	Azure
SAP Server 1	WinServer1	WinServer-2019-1
SAP Server 2	WinServer2	WinServer-2019-2
SAP Server 1 HDD	vol-0a06bfea3112154f9	2019_1_OsDisk_1_ 52b3f67c18f8409a ae85c6ef9d5ae9e9
SAP Server 2 HDD	vol-0ef52c835b2aa9f6d	2019_2_OsDisk_1_ ae91bec98de18e2 2cde749a2d09b20b
SAP Server 1 NSG	launch-wizard-2	WinServer-2019-1-nsg
SAP Server 2 NSG	launch-wizard-3	WinServer-2019-2-nsg
Web Server	WebServer	UbuntuServer1804
Web Server HDD	vol-0a3b0ba64213cee92	UbuntuServer1804_ OsDisk_1_6b95ff86 28894fc9a47eeff668423fab
Web Server NSG	launch-wizard-1	UbuntuServer1804-nsg
Load Balancer	BackendLB	BackendLB
Frontend Subnet	DMZ	DMZ
Backend Subnet	BackendSubnet	BackendSubnet
Kubernetes Cluster	EKS-Cluster	AKS-Cluster
Kubernetes NSG	eks-remoteAccess	aks-agentpool-98637346-nsg
SQL Database	database-users-instance-1	tfmsql/users_database
Resource Group	-	TFM-RG
S3/Storage Account	s3tfmseryoferben	tfmsta

Ilustración 7: Inventario de recursos desplegados

Existe, sin embargo, algún recurso adicional creado automáticamente derivado del uso de diferentes herramientas (almacenamiento del cloudshell, almacenamiento de plantillas de CloudSploit, etc…).

3. Plataformas utilizadas

Se optará en el desarrollo del trabajo por tres plataformas de gobierno en la nube o CSPM (Cloud Security Posture Management) que nos permiten analizar la infraestructura al completo en busca de configuraciones inseguras que puedan dar como resultado una vulnerabilidad, la cual pueda ser explotada por un atacante.

Existe una gran variedad de herramientas comerciales desarrolladas para este propósito, unos ejemplos son:

- DivvyCloud [12]
- CloudVisory [13]
- Optiv [14]

Debido a la gran cantidad de soluciones encontradas y la ausencia de comparativas entre tantas opciones, se ha optado por acudir a referentes ampliamente conocidos en el terreno de la ciberseguridad, como Palo Alto, Checkpoint o Fortinet.

En el caso de Palo Alto y Checkpoint, compraron soluciones desarrolladas por otras empresas para agregarlas a sus catálogos de productos (Redlock y Twistlock por parte de Palo Alto [15] y Dome9 por parte de Checkpoint [16]), mientras que Fortinet ha desarrollado su propia herramienta para seguridad en Cloud. [17]

Muchos de estos productos no ofrecen una única funcionalidad de las descritas al inicio de este trabajo. Un ejemplo puede ser Prisma, que ofrece bajo una misma solución herramientas de CSPM y CWPP (Cloud Workload Protection Platform) (Redlock y Twistlock, respectivamente), sin embargo, nos centraremos siempre en los aspectos relativos a CSPM.

Aunque la intención inicial era poner en perspectiva las herramientas de los proveedores de seguridad tradicionales en cuanto a firewalls (Fortinet, Palo Alto y Checkpoint), ha sido imposible conseguir una versión de prueba de la herramienta de Fortinet, ya que, al contrario que Prisma y Dome9, no ofrece una manera directa de activar una versión de prueba y es necesario contactar con la empresa.
Se consiguió respuesta de Fortinet para esta demo, indicando que se debía gestionar a través de un mayorista (Arrow, en este caso), pero no se consiguió respuesta de esta empresa pese a que se realizaron numerosos intentos de conseguir una versión trial de la herramienta.

Para complementar las soluciones comerciales, se disponemos también de herramientas Open Source, pero por desgracia, no hay muchas más soluciones que la propuesta por Cloudsploit, perteneciente a la empresa Aqua Security. Actualmente, el resto de las herramientas localizadas no son completas y han sido creadas por desarrolladores que las han adaptado a sus necesidades concretas, de manera ad-hoc. [18]

3.1. Palo Alto Prisma Cloud

Prisma Cloud es la herramienta de seguridad en cloud del conocido proveedor de servicios de seguridad americano Palo Alto.

Esta solución no fue desarrollada por la propia Palo Alto, sino que pertenecía a otra empresa llamada Redlock, la cual daba nombre al producto.

Palo Alto compró la empresa Redlock Inc. en octubre de 2018 [15], incorporando, de esta manera, el producto a su catálogo de soluciones y servicios. Desde entonces, la herramienta ha seguido evolucionando, aumentando su compatibilidad con otros CSP y nuevos recursos de los proveedores cloud ya existentes según evolucionaban.

Prisma se conecta con los entornos cloud de diferentes maneras dependiendo del proveedor para obtener la información de todos los recursos en cada cuenta o suscripción.
Una vez conectado, Prisma recupera la información de los recursos en formato JSON, y sobre estos datos aplica un lenguaje de queries propio (RQL, Redlock Query Languaje) para comprobar los valores de diferentes campos obtenidos en el mencionado JSON.

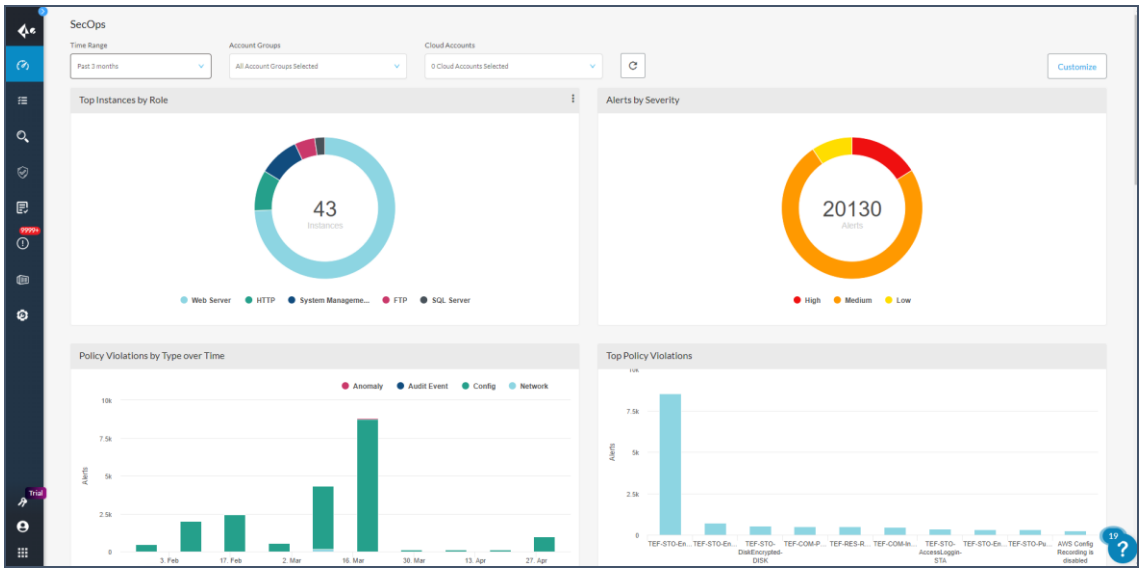


Ilustración 8: Detalle del dashboard principal de Prisma Cloud

Ilustración 9: Detalle de un ejemplo de query RQL

Una vez definidas las políticas y obtenidos la configuración de los recursos, Prisma evalúa los recursos encontrados en busca de aquellos que no cumplan con las mismas y genera una alerta.

3.1.1. Conexión con AWS

Prisma Cloud es una herramienta SaaS desplegada (total o parcialmente, se desconoce) en AWS. Es fácilmente comprobable revisando el rol creado de manera automática siguiendo el proceso de onboarding dado por la herramienta:

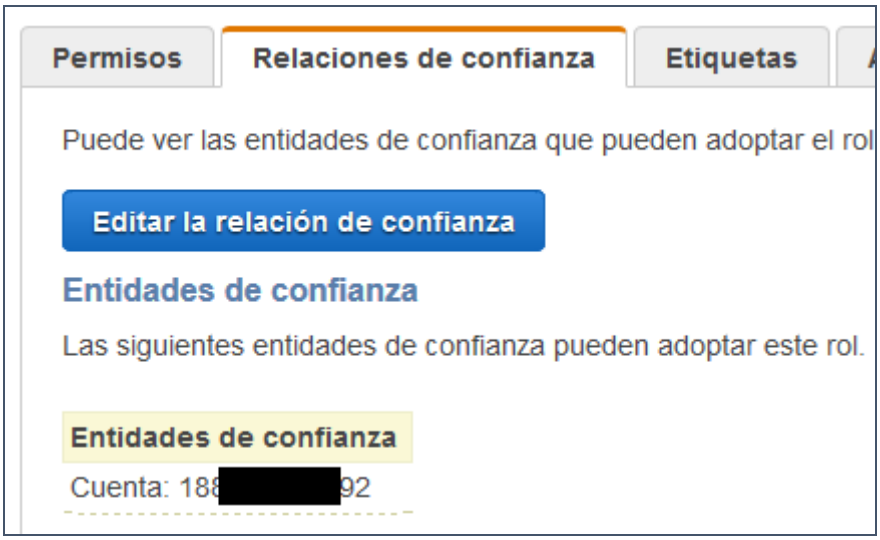


Ilustración 10: Cuenta de AWS que asume el rol generado

Se puede observar que el rol tiene una relación de confianza en la que se especifica qué entidades pueden asumir dicho rol. La entidad que aparece es otra cuenta de AWS, que se corresponde con la cuenta de Amazon desde donde Prisma Cloud se conecta a nuestra cuenta de AWS.

Una vez que Prisma ha asumido el rol creado, puede leer la configuración de todos los elementos indicados en la política:

Ilustración 11: Detalle parcial de los recursos accesibles por la política

En total, observamos que el rol otorga acceso de lectura (completo o limitado) a 226 servicios en total de AWS, permitiendo, de este modo, comprobar la configuración de todos los recursos incluidos en la política del rol y contrastándola con las políticas y controles creados en Prisma.

3.1.2. Conexión con Azure

Como se ha realizado durante la conexión de Prisma Cloud con Azure, la plataforma hace uso de un App Registration (aplicación) generada con Terraform.

Un App registration facilita exponer el API de Azure, utilizando un Application ID exclusivo para este App registration, permitiendo tener diferentes aplicaciones a las cuales se les pueden asignar diferentes roles para diferentes propósitos a través de un Service Principal. [19]

Un Service Principal de Azure es una identidad creada para ser utilizada con aplicaciones y herramientas automatizadas para acceder a los recursos de Azure. Este Service Principal es el elemento al que se le asocian los roles para gestionar el acceso a los diferentes recursos disponibles en el entorno de Azure, y siempre está asignada a los App Registration mencionados en el párrafo anterior. [20]

Según lo que se pretenda conseguir con la herramienta, se le asignará un rol u otro en función de si se quieren activar remediaciones automáticas.

A través de este App registration, Prisma es capaz de listar todos los elementos de la infraestructura, obteniendo su configuración al completo en formato JSON.

Además, es capaz es capaz de leer del activity log de la suscripción dada de alta y recibir los cambios realizados en los recursos al momento, reevaluándolos cuando hayan sido actualizados por los desarrolladores en busca de nuevas vulnerabilidades.

3.2. Checkpoint Dome9 Cloudguard

De manera similar a lo ocurrido con Prisma y Palo Alto, Dome9 era una empresa independiente cuyo nombre coincidía con el del producto.

En octubre de 2018, prácticamente al tiempo que Palo Alto adquiría Prisma Inc., Checkpoint compraba esta empresa para reforzar su catálogo de soluciones de seguridad en la nube. [16]

El planteamiento de la herramienta y funcionamiento es idéntico al de Prisma, aunque incluye más funcionalidades en cuanto a monitorización de redes.

Al igual que en Prisma, se dispone de un lenguaje de query propio, llamado Governance Specification Languaje (GSL). Existe, sin embargo, una diferencia sustancial respecto a Prisma, Dome9 no aplica las queries directamente sobre el JSON obtenido de la API de los CSPs, sino que realiza una primera abstracción de estos datos para normalizarlos y hacer que la tarea de realizar reglas personalizadas sea más sencilla, aunque resta potencial a su motor de consultas.

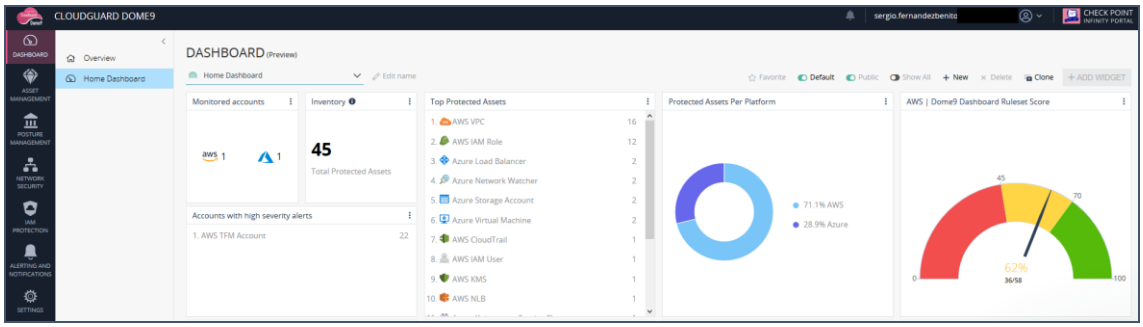


Ilustración 12: Detalle del dashboard principal de Dome9

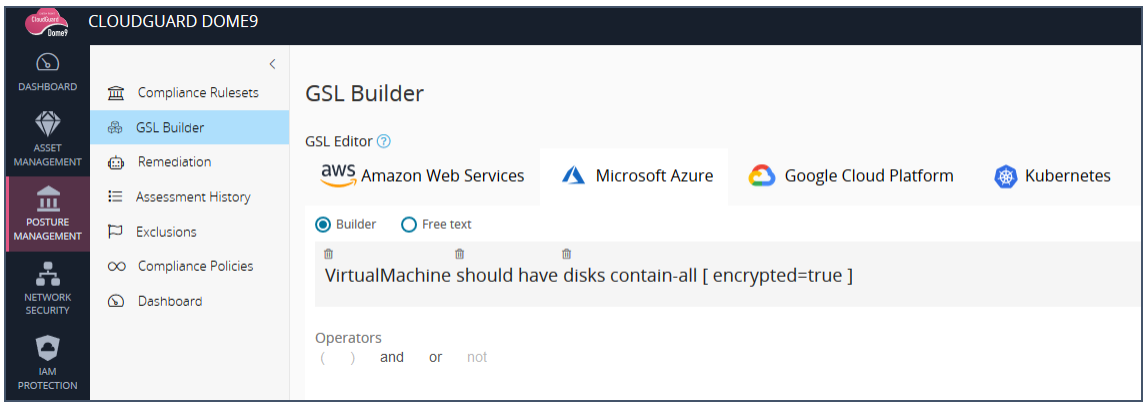


Ilustración 13: Detalle de un ejemplo de query GSL

3.2.1. Conexión con AWS

Nos encontramos en el mismo escenario que con Prisma. Dome9 se ejecuta en AWS (total o parcialmente, esa información no es pública) y utiliza una cuenta de AWS para conectarse vía API a la cuenta objetivo de AWS a través de un rol concreto.

Una vez que se le facilita el rol con los requisitos que solicita, es capaz de asumir dicho rol, al cual habremos dado el permiso de asumir a la cuenta de Dome9, y podrá acceder a nuestros recursos en modo de solo lectura o también con permisos de escritura, según hayamos configurado remediaciones automáticas o no.

3.2.2. Conexión con Azure

El funcionamiento con Azure también es idéntico al visto en Prisma.

Requiere de la creación de un App Service asociado a un Service Principal, al cual se le asocian los roles necesarios (solo lectura o "Contributor" para realizar remediaciones automáticas).

Una vez que se ha creado tanto el Service Principal, App Service y asignación de roles, Dome9 utiliza los siguientes valores para realizar la integración:

- Tenant ID (identificador de Directorio Activo)
- Susbcription ID (identificador de suscripción)
- Client ID (identificador del App Service)
- Client Key (clave del App Service)

Con todos estos parámetros necesarios para autenticarse, Dome9 utiliza la API pública de Azure para obtener toda la información de la suscripción de Azure y monitorizar los recursos que se encuentran en ella.

3.3. Aqua CloudSploit

Cloudsploit apareció en junio de 2015 como un proyecto Open Source enfocado a democratizar la seguridad en cloud.

En noviembre de 2019 fue comprada por la compañía Aqua Security, de cara a poder aumentar su cartera de servicios. [21]

Pese a haber sido adquirido por una empresa, el concepto de Cloudsploit en lo referente a Open Source no ha cambiado. Sigue siendo una herramienta de código libre cuyo código puede localizarse en GitHub, donde se sigue actualizando habitualmente. [22]

Pese a ser de código abierto, el servicio que ofrece Aqua Security con la licencia es su uso como aplicación SaaS, alojado en los propios servidores o nube de Aqua, liberando al usuario final de la necesidad de desplegar la herramienta en un entorno concreto o de mantenerla. [23]

3.3.1. Conexión con AWS

El escenario de integración de Cloudsploit con AWS es el mismo que en ya se ha visto con Prisma y Dome9, no existen diferencias.

La única diferencia se encuentra en el proceso, en el caso de Dome9 toda la creación de roles es manual, mientras que tanto en Prisma como en Cloudsploit, se ofrece un stack que crea todos los elementos necesarios de manera automática a través de plantillas Cloudformation.

3.3.2. Conexión con Azure

De la misma manera que visto en las otras dos herramientas, Cloudsploit utiliza la API pública de Azure consiguiendo credenciales a través de un Service Principal dado de alta como una aplicación.

A este Service Principal se le asocian los roles adecuados y, junto con la creación de una clave secreta, Cloudsploit es capaz de autenticarse contra el CSP y recuperar los datos del entorno.

Al igual que en Dome9, todo el proceso de configuración en el CSP es manual, aunque guiado con instrucciones muy detalladas.

4. Resultados Prisma Cloud

4.1. AWS

4.1.1. Vulnerabilidades detectadas

En este apartado se recogen todas las vulnerabilidades detectadas por Prisma Cloud en nuestro entorno de AWS, descrito en el apartado 2 del trabajo.

En primer lugar, recurrimos al dashboard de Prisma para revisar qué procentaje de incumplimiento detecta en la cuenta en la región utilizada (AWS París) y con los reglamentos definidos anteriormente; CIS, PCI, NIST y RGPD:

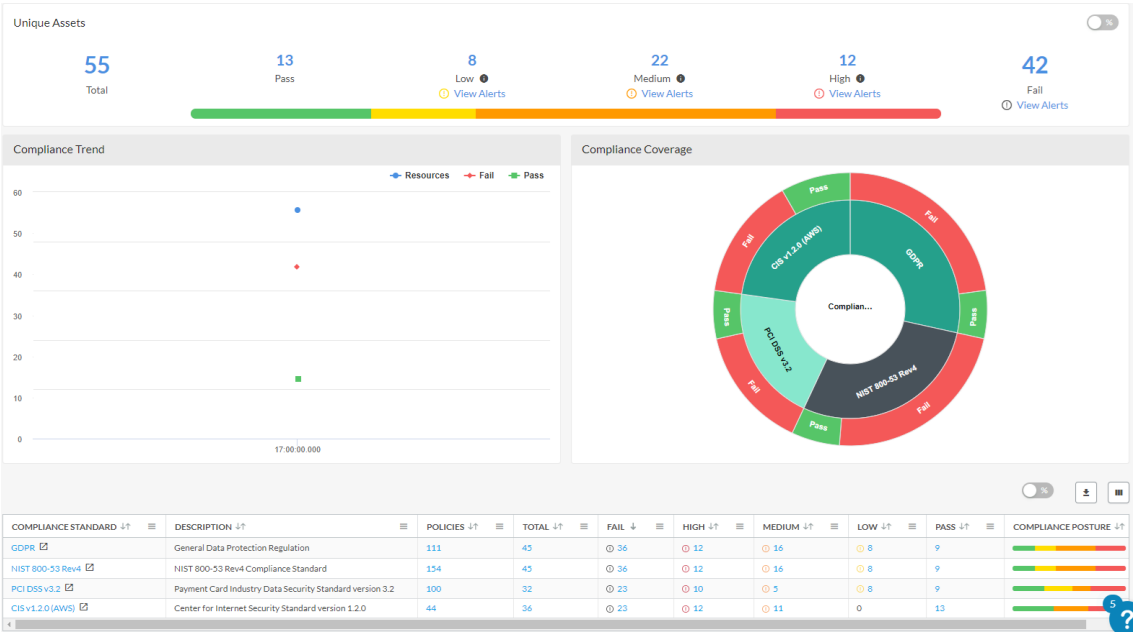


Ilustración 14: Dashboard de Prisma antes de securizar el entorno de AWS

NOTA: El número de fallos que aparece reflejado en Prisma se refiere a los recursos que fallan una política, no al número de política violadas.

Observamos que el nivel de cumplimiento está muy distante de alcanzar siquiera el 50% en cualquiera de los estándares.

Obtenemos un total de 22 vulnerabilidades diferentes, y destaca la aparición de ciertos elementos no contemplados en la infraestructura, como es el caso del bucket S3 “cf-templates-1ewjr29s9uj7t-eu-west-3”, creado automáticamente para las templates de CloudFormation utilizadas para los roles generados por las herramientas, entre otros.

Severidad: Alta **NºVulnerabilidad:** 1 **Reglamento**: CIS, PCI, RGPD **Nombre**: MFA no está activado en la cuenta root **Descripción**: Las cuentas root tienen el máximo nivel de privilegios. Si las credenciales se ven comprometidas, un atacante podría conseguir acceso completo al entorno. **Recursos afectados**: root

Severidad: Alta **NºVulnerabilidad:** 2 **Reglamento**: CIS, RGPD, NIST **Nombre**: Se detectan Security Groups que permiten tráfico de internet al puerto 22 (SSH) **Descripción**: No se debe permitir la entrada desde internet al puerto 22, ya que podría ser objetivo de un ataque de fuerza bruta, facilitando acceso a la red. **Recursos afectados**: launch-wizard-1

Severidad: Alta **NºVulnerabilidad:** 3 **Reglamento**: CIS, RGPD, NIST
Nombre: Se detectan Security Groups que permiten tráfico de internet al puerto 3389 (RDP)
Descripción: No se debe permitir la entrada desde internet al puerto 3389, ya que podría ser objetivo de un ataque de fuerza bruta, facilitando acceso a la red.
Recursos afectados: launch-wizard-3, launch-wizard-2

Severidad: Alta **NºVulnerabilidad:** 4 **Reglamento**: CIS, RGPD, NIST
Nombre: Los Security Groups por defecto de las subnets no restringen todo el tráfico de entrada
Descripción: Cuando se genera una instancia en AWS, si no se le asigna Security Group, hereda el de la subnet.
Recursos afectados: default (Security Group)

Severidad: Alta **NºVulnerabilidad:** 5 **Reglamento**: PCI, RGPD, NIST
Nombre: Security Groups permitiendo tráfico entrante sin filtrar
Descripción: Se detectan Security Groups que permiten tráfico entrante desde internet, sin especificar rango de direcciones IP.
Recursos afectados: launch-wizard-1, launch-wizard-3, launch-wizard-2

Vulnerabilidades de la 6 a la 12, inclusive, relacionadas con las políticas de contraseñas de la cuenta localizadas en el anexo.

Severidad: Media **NºVulnerabilidad:** 13 **Reglamento**: PCI, RGPD, NIST
Nombre: Instancia RDS sin cifrar
Descripción: Se detecta una base de datos RDS sin cifrar. Este cifrado es obligatorio por cumplimiento y razones de seguridad.
Recursos afectados: database-users-instance-1

Severidad: Media **NºVulnerabilidad:** 14 **Reglamento**: PCI, RGPD
Nombre: El logado de accesos a los buckets S3 no está activo
Descripción: El logado de acceso a los buckets S3 permite disponer de todas las trazas de auditoría en elementos sensibles como los S3.
Recursos afectados: cf-templates-1ewjr29s9uj7t-eu-west-3, s3tfmseryoferben

Severidad: Media **NºVulnerabilidad:** 15 **Reglamento**: RGPD, NIST
Nombre: Las subnets de las VPC de AWS no deberían permitir asignación automática de IPs públicas
Descripción: La asignación automática de direcciones IP públicas a las máquinas que se encuentren en determinada subnet podría provocar, de manera inintencionada, la exposición de estas a internet, exponiéndolas a posibles atacantes.
Recursos afectados: DMZ, BackendSubnet

Severidad: Media **NºVulnerabilidad:** 16 **Reglamento**: RGPD, NIST
Nombre: El registro de actividad de AWS Config está desactivado
Descripción: AWS Config permite registrar cambios en la configuración del entorno para monitorizar el comportamiento de los usuarios, obteniendo datos de actividad de cara a una auditoría.
Recursos afectados: AWS París (región)

Severidad: Media **NºVulnerabilidad:** 17 **Reglamento**: RGPD, NIST
Nombre: CloudTrail no integrado con CloudWatch
Descripción: Habilitar CloudWatch permite monitorizar y generar reglas para controlar las acciones reportadas en los logs del CloudTrail.
Recursos afectados: aquacloud-security-scanner2-CloudSploitCloudTrail-10I0HJYCFRP8Z, cloudsploit-security-scanner-CloudSploitCloudTrail-TH0YCB80C5RU

Severidad: Media **NºVulnerabilidad:** 18 **Reglamento**: RGPD, NIST
Nombre: Logs de CloudTrail sin cifrar con claves gestionadas por el cliente (CMK)
Descripción: Cifrar el contenido de los diferentes CloudTrail utilizando claves gestionadas por la organización permite reducir al mínimo la posibilidad de que datos sensibles almacenados en este servicio sean accedidos de manera no autorizada.
Recursos afectados: aquacloud-security-scanner2-CloudSploitCloudTrail-10I0HJYCFRP8Z, cloudsploit-security-scanner-CloudSploitCloudTrail-TH0YCB80C5RU

Severidad: Media **NºVulnerabilidad:** 19 **Reglamento**: CIS
Nombre: Instancias EC2 sin rol IAM
Descripción: Es recomendable asignar un rol IAM a las instancias EC2 que realicen cualquier interacción con recursos de la cuenta, gestionando mejor los permisos y evitando el uso de claves.
Recursos afectados: WebServer, WinServer1, WinServer2

Severidad: Baja **NºVulnerabilidad:** 20 **Reglamento**: CIS, RGPD
Nombre: Políticas de permisos IAM asignadas a usuarios
Descripción: Se han detectado políticas de permisos asignadas directamente a usuarios, cuando deberían aplicarse exclusivamente a grupos y roles para mejorar la gestión y el control de los permisos.
Recursos afectados: 10828-106460816557-list-attached-user-policies

Severidad: Baja **NºVulnerabilidad:** 21 **Reglamento**: PCI, RGPD, NIST
Nombre: Los volúmenes EBS no están cifrados
Descripción: Los discos asociados a las máquinas virtuales deben estar cifrados para prevenir una posible fuga de información. A su vez, tener estos discos cifrados permite compartirlos entre cuentas AWS de manera segura.
Recursos afectados: vol-0a06bfea3112154f9, vol-0ef52c835b2aa9f6d, vol-0a3b0ba64213cee92

Severidad: Baja **NºVulnerabilidad:** 22 **Reglamento**: RGPD, NIST
Nombre: Los buckets S3 no tienen habilitado el cifrado en reposo
Descripción: Es necesario cifrar los buckets S3 que contengan datos sensibles para evitar que ante un incidente de seguridad no se producen accesos indebidos.
Recursos afectados: s3tfmseryoferben

4.1.2. Plan de acción

Empezando por las alertas de criticidad alta, observamos que:

- La cuenta root no tiene MFA (vulnerabilidad número 1)
- El puerto 22 está expuesto a internet (vulnerabilidad número 2)
- El puerto 3389 está expuesto a internet (vulnerabilidad número 3)
- Los SG (Security Group) por defecto de las subnets no restringen el tráfico entrante (vulnerabilidad número 4)
- Hay SG que aceptan tráfico entrante sin filtrar por IP (vulnerabilidad número 5)

Observamos que 4 de las 5 alertas son problemas de segmentación de red. No tener correctamente segmentada la red implica desconocer o no tener inventariadas qué comunicaciones deben transcurrir por cada servicio.

Esto supone un problema grave ya que, en caso de no tenerlo controlado, podríamos permitir comunicaciones no legítimas entre grupos de recursos en un posible caso de infección, permitiendo el movimiento lateral de la infección.

Tras esta breve explicación, los pasos a tomar para solucionar este problema pasarían por, en primer lugar, limitar el acceso a los puertos 22 y 3389 a un rango de Ips concreto (una VPN, por ejemplo), en el caso de no cerrarlo por completo.

En cuanto al SG por defecto de la subnet que no restringe el tráfico de entrada, el problema reside en el caso de generar una instancia en dicha subnet y no asignarle un SG heredaría el de la subnet. Esto podría dejar una máquina expuesta a atacantes sin que el administrador de la nube tuviera constancia de ello, dejando una puerta de entrada a la infraestructura.
Por esto, el paso a seguir es crear una regla en el SG que restrinja todo el tráfico de entrada.

Terminando con la segmentación de red, vemos que hay SG que no filtran el tráfico entrante por IP. Como se ha comentado al comienzo de este apartado, tener inventariadas y controladas las comunicaciones reduce al mínimo la posibilidad de infecciones o accesos no autorizados. Se debe asegurar, por tanto, que las instancias que no tengan que estar expuestas a todo internet (como un servidor web) tengan filtrado por IP en todas las reglas de entrada de su SG.

Finalizando con las vulnerabilidades de severidad alta, se nos alerta que el usuario root (administrador) de la cuenta no tiene activo MFA (factor de autenticación multifactor). En primer lugar, esta cuenta debería utilizarse lo menos posible, ya que se trata del usuario con más privilegios del entorno y la administración debería estar repartida entre diferentes roles.
Teniendo esto en cuenta, si las credenciales de este usuario fueran comprometidas, un atacante podría hacerse con el control absoluto de la infraestructura, secuestrando por completo la cuenta, dando como resultado no sólo la denegación de servicio, sino que además una subida de coste explosiva si el atacante empieza a desplegar recursos indiscriminadamente.
En este caso la solución es muy sencilla, y aumenta enormemente la seguridad del acceso a la cuenta.

Continuando con las alertas de severidad media, vemos varias relacionadas con la política de contraseñas utilizadas por los usuarios (vulnerabilidades de la 6 a la 12).
Las credenciales de acceso de los usuarios son la puerta de entrada al servicio. Obligar a estos a utilizar contraseñas lo más seguras y variadas posibles, evitando su repetición, ayudará a reducir la posibilidad de que se encuentren en diccionarios de contraseñas (por ejemplo, el diccionario Rockyou), reduciendo drásticamente la posibilidad de que un ataque por fuerza bruta o diccionario tenga éxito.
La solución pasa por habilitar todas las restricciones de contraseña indicadas por la herramienta (aumentar la longitud de las mismas, obligar a utilizar símbolos, mayúsculas, minúsculas y números).

A continuación, en la vulnerabilidad número 13, vemos una base de datos RDS sin cifrar. Tanto el marco PCI-DSS como el Reglamento General de Protección de Datos obligan a tener los datos referentes a tarjetas de crédito y datos personales, respectivamente, cifrados en reposo. De este modo se puede evitar un acceso no autorizado a los mismos.

La siguiente alerta, la número 14, nos avisa que los buckets S3 no están registrando la actividad (logado de auditoría). De cara a una posible filtración de datos, el RGPD obliga a determinar el alcance de esta. Sin el registro de accesos es prácticamente imposible determinar el nivel de afectación de los datos. Es, además, obligatorio para verificar el tratamiento de los datos según el RGPD.

Siguiendo con la alerta número 15, referente a la asignación automática de IPs públicas en las subnets, se puede entender fácilmente por qué esta característica supone un problema.
Si una instancia no consta de IP pública no es accesible desde internet. Si se descuida este aspecto se puede dar el caso de una máquina crítica que resulta accesible desde cualquier parte del mundo, dejando una puerta abierta a los atacantes.

Observamos también en la vulnerabilidad número 16 que AWS Config está desactivado. Esta herramienta permite realizar un análisis de la actividad de los usuarios, detectando anomalías. Si existe un usuario en el entorno con intenciones maliciosas (insider) podría ser localizado con esta herramienta antes de que pudiera generar un incidente de seguridad.

Continuando con la vulnerabilidad número 17, se nos nos informa que no se ha habilitado CloudWatch con los logs de CloudTrail. CloudWatch permite monitorizar acciones realizadas dentro del entorno AWS, de manera similar a AWS Config, pero incluyendo las alertas de este servicio. Por lo tanto, es altamente recomendable la activación en conjunto de ambos servicios para generar alertas en caso de detectar cambios no autorizados.

Otro problema reportado por Prisma es no disponer del contenido de los CloudTrail cifrados, concretamente utilizando claves gestionadas por la organización (vulnerabilidad número 18). Los datos de almacenados en ClousTrails contienen de manera habitual información sobre acciones de los usuarios, tales como su nombre de usuario, lugar de conexión, etc… Por ello, es necesario mantenerlos cifrados. Si, además, se realiza con claves gestionadas por la organización, se puede controlar mejor el acceso a estos datos, permitiendo el acceso a las claves de cifrado a aquellas personas que realmente lo vayan a necesitar.

Continuando con la vulnerabilidad número 19, se nos reporta que las instancias EC2 no tienen ningún rol IAM asignado. Esto es un problema siempre y cuando estas instancias interactúen con recursos del entorno, ya que deben almacenar en disco (ya sea en scripts o en ficheros de texto) los datos de la API de AWS. Si se utiliza un rol IAM en su lugar, se evita almacenar claves, ya que la máquina puede asumir dicho rol, quedando a la ver autenticada y autorizada.

Finalizando con las alertas de severidad baja, se nos indica que las políticas de permisos deben asignarse a roles o grupos, nunca a usuarios directamente (vulnerabilidad número 20).
Esto no es una vulnerabilidad por sí misma, pero sí ayuda a mantener un control de los permisos asignados a los usuarios, evitando que un administrador olvide que cierto usuario tiene permisos asignados fuera del rol o grupo que le corresponde, cumpliendo así con el concepto de conceder a los usuarios la menor cantidad de permisos posible.

Finalmente, se nos reporta que ni los volúmenes de disco (EBS) de las máquinas virtuales ni los buckets S3 están cifrados (vulnerabilidades 21 y 22, respectivamente).
Solucionar este problema no es complejo y, como se ha explicado en otras vulnerabilidades, permite reducir enormemente la posibilidad de un acceso no autorizado a los datos, además de ser obligatorio según los reglamentos PCI-DSS y el RGPD.

Una vez solucionados los problemas detectados, volvemos a revisar el dashboard de cumplimiento visto anteriormente:

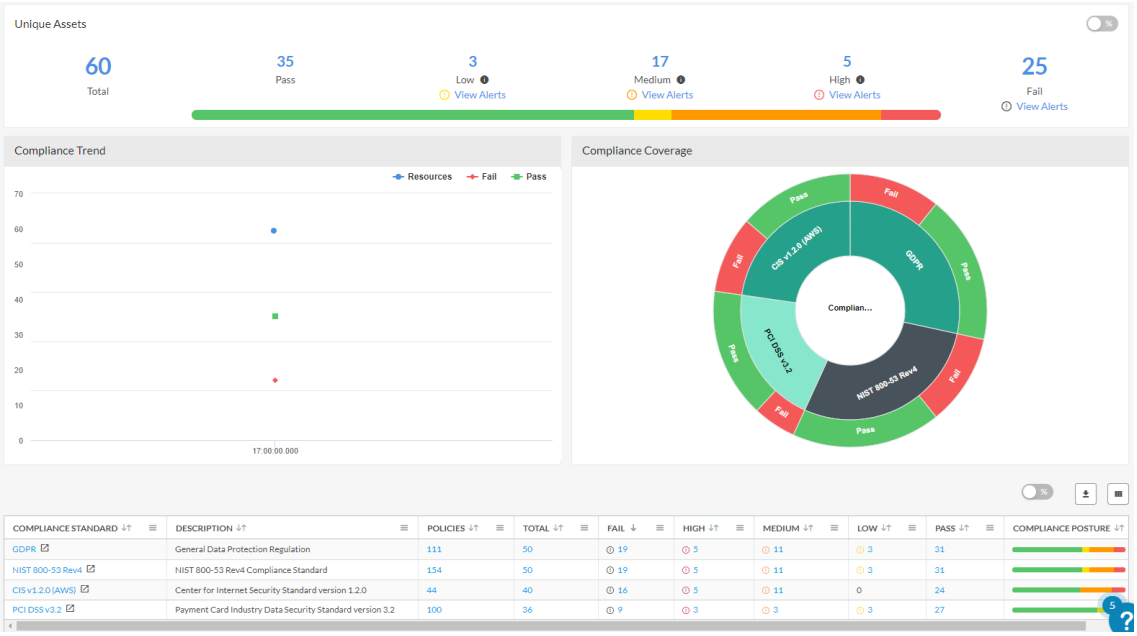


Ilustración 15: Dashboard de Prisma tras securizar el entorno de AWS

Vemos que se ha mejorado en todos los aspectos, pero no se ha mejorado al 100% en ningún caso. Esto es un problema de Prisma, ya que, si buscamos todas las alertas abiertas por estos reglamentos y región, sólo nos aparece una:

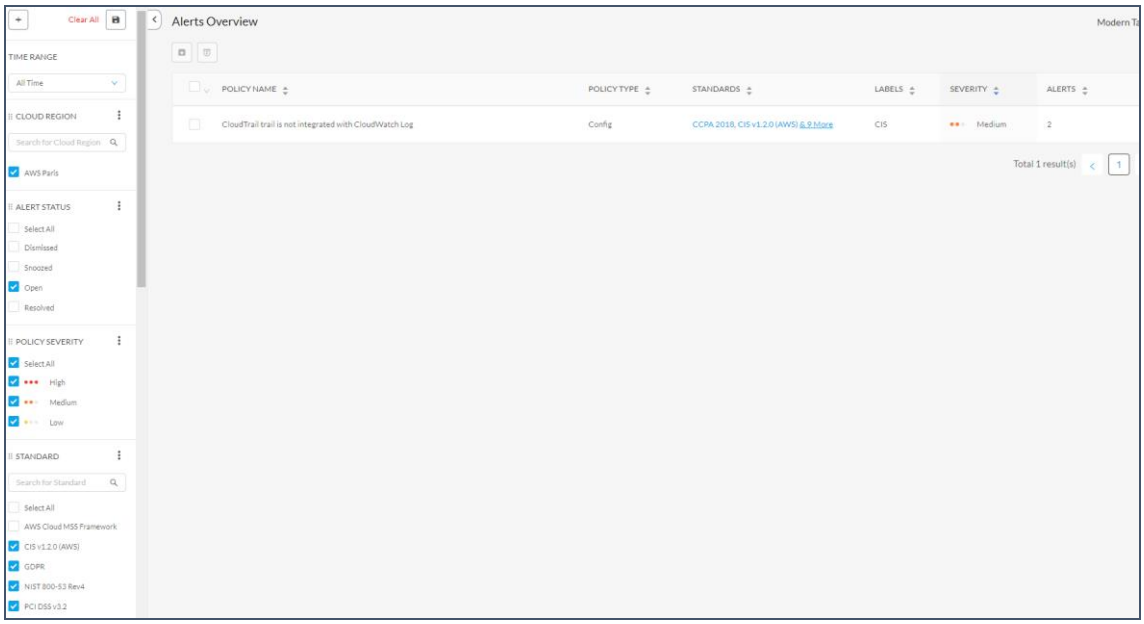


Ilustración 16: Vista de las alertas abiertas en Prisma, evidenciado un único problema pendiente de resolver

Esta alerta no se pudo solucionar por un problema relacionado con AWS Config, que fue imposible resolver, quedando pendiente en todas las herramientas, no solo en Prisma:

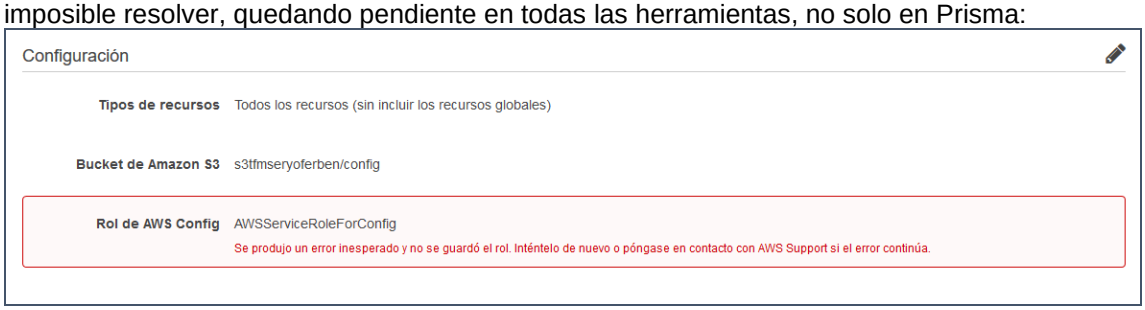


Ilustración 17: Problema de configuración en AWS Config

De no ser por el problema con el dashboard y esta alerta pendiente, el cumplimiento se vería actualizado al 100% en todos los reglamentos vistos.

4.2. Azure

4.2.1. Vulnerabilidades detectadas

En primer lugar, observaremos la información de manera resumida utilizando el dashboard que proporciona Prisma, donde podemos ver los resultados filtrados eligiendo el parámetro que más nos convenga (regiones, tipo de nube, grupos de cuentas, reglamento, etc…).

En el caso que nos atañe, se ha aplicado el filtro por reglamento (CIS, PCI, NIST y RGPD) y por la región donde se ha desplegado toda la infraestructura (West Europe).

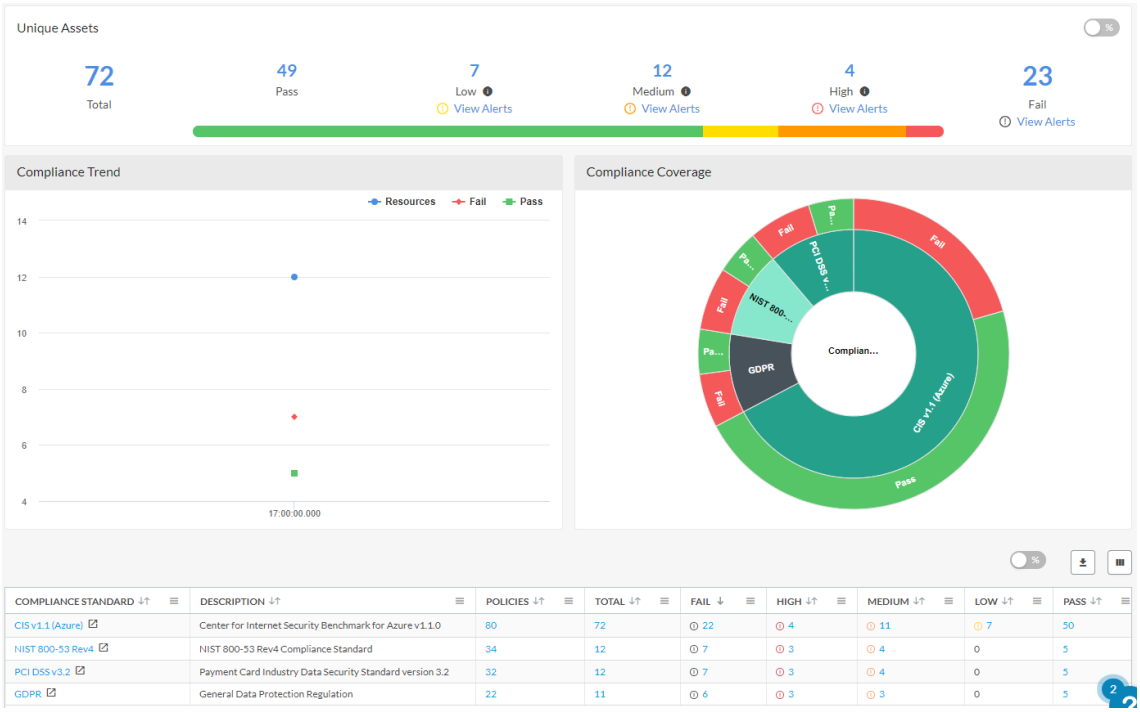


Ilustración 18: Dashboard de Prisma antes de securizar el entorno de Azure

Podemos observar que salvo en el CIS, apenas se llega al 50% de cumplimiento en ninguno de los estándares. Además, genera una alerta por cada recurso que falla un control, teniendo alertas duplicadas, en lugar de agrupadas por recurso.

Entrando en detalle, obtenemos un total de 21 vulnerabilidades, en este caso cobra más protagonismo las reglas asociadas al CIS, en lugar del RGPD como ocurría en AWS.

Sin embargo, vemos también recursos creados automáticamente, como la Storage Account “csb10032000a2811712”, utilizada para el almacenamiento del CloudShell, o varios grupos de recursos, como “MC_TFM2-RG_AKS-Cluster_westeurope”, autogenerado en el despliegue del clúster de Kubernetes.

Por otro lado, destaca la cantidad de recomendaciones obtenidas para el uso de Azure Security Center, no teniendo que ver directamente con la configuración de los recursos desplegados.

Severidad: Alta **NºVulnerabilidad:** 1 **Reglamento**: CIS, PCI, RGPD, NIST **Nombre**: No se debe permitir la entrada desde internet al puerto 3389 en los NSG **Descripción**: No se debe permitir la entrada desde internet al puerto 3389, ya que podría ser objetivo de un ataque de fuerza bruta o un ransomware. **Recursos afectados**: WinServer-2019-1-nsg, WinServer-2019-2-nsg

Severidad: Alta **NºVulnerabilidad:** 2 **Reglamento**: CIS, PCI, RGPD, NIST
Nombre: No se debe permitir la entrada desde internet al puerto 22 en los NSG
Descripción: No se debe permitir la entrada desde internet al puerto 22, ya que podría ser objetivo de un ataque de fuerza bruta
Recursos afectados: UbuntuServer1804-nsg

Severidad: Alta **NºVulnerabilidad:** 3 **Reglamento**: CIS
Nombre: AKS (Azure Kubernetes Service) no utiliza gestión de permisos RBAC
Descripción: Gestionar los permisos a través de RBAC (Role Based Access Control) permite, de manera granular, asignar los permisos de creación y modificación de recursos de una manera mucho más precisa en un AKS.
Recursos afectados: AKS-Cluster

Vulnerabilidades de la 4 a la 8, relacionadas con Azure Security Center, localizadas en el anexo.

Severidad: Media **NºVulnerabilidad:** 9 **Reglamento**: CIS, PCI, RGPD, NIST
Nombre: Storage Accounts sin cifrado en tránsito habilitado
Descripción: Es necesario limitar el tráfico de las STAs a HTTPS utilizando SSL, evitando que los datos puedan ser leídos por un tercero no autorizado.
Recursos afectados: tfmsta

Severidad: Media **NºVulnerabilidad:** 10 **Reglamento**: CIS
Nombre: El logado de acceso de las STAs está desactivado
Descripción: Es necesario logar los accesos a las STAs para, en caso de incidente, poder disponer de datos a nivel de auditoría y poder localizar exfiltraciones de datos.
Recursos afectados: csb10032000a2811712, tfmsta

Severidad: Media **NºVulnerabilidad:** 11 **Reglamento**: CIS
Nombre: La acción por defecto de las STAs es “allow” para todas las redes
Descripción: Restringir el tráfico de entrada a las STAs permite añadir una capa adicional de seguridad y tener los accesos a los datos controlados e inventariados.
Recursos afectados: csb10032000a2811712, tfmsta

Severidad: Media **NºVulnerabilidad:** 12 **Reglamento**: CIS
Nombre: Flow logs con una retención inferior a 90 días
Descripción: La retención de los Flow logs debe ser como mínimo 90 días para poder disponer de información ante un incidente de seguridad.
Recursos afectados: aks-agentpool-98637346-nsg, WinServer-2019-1-nsg, WinServer-2019-2-nsg, UbuntuServer1804-nsg

Severidad: Media **NºVulnerabilidad:** 13 **Reglamento**: CIS
Nombre: Los discos duros de máquinas virtuales no están cifrados
Descripción: Como buena práctica y medida preventiva de fugas de información, los discos duros de las máquinas virtuales deben estar cifrados.
Recursos afectados: WinServer-2019_OsDisk_1_52b3f67c18f8409aae85c6ef9d5ae9e9, 2019_2_OsDisk_1_ae91bec98de18e22cde749a2d09b20b , UbuntuServer1804_OsDisk_1_6b95ff8628894fc9a47eeff668423fab

Severidad: Media **NºVulnerabilidad:** 14 **Reglamento**: CIS
Nombre: Máquinas virtuales sin agente de protección de endpoints instalado
Descripción: Instalar agentes en las máquinas virtuales con el fin de proteger el entorno contra diferentes tipos de malware.
Recursos afectados: WinServer-2019-1, WinServer-2019-2, UbuntuServer1804

Severidad: Media **NºVulnerabilidad:** 15 **Reglamento**: CIS, PCI, NIST
Nombre: Los servidores SQL no tienen un administrador configurado a través de Active Directory
Descripción: Gestionando la identidad de los administradores a través de Active Directory se consigue un control mucho más exhaustivo que con un usuario de base de datos.
Recursos afectados: tfmsql

Severidad: Media **NºVulnerabilidad:** 16 **Reglamento**: PCI, NIST
Nombre: Las reglas de firewall del servidor SQL permiten el acceso sin filtrar.
Descripción: Las reglas del firewall del servidor SQL permiten el acceso desde 0.0.0.0/0, dejando las bases de datos expuestas a internet.
Recursos afectados: tfmsql

Severidad: Media **NºVulnerabilidad:** 17 **Reglamento**: CIS, PCI, RGPD, NIST
Nombre: El logado de las bases de datos SQL debe estar habilitado.
Descripción: El logado de las bases de datos SQL permite recabar información ante incidentes de seguridad y es obligatorio por normativa.
Recursos afectados: tfmsql

Severidad: Media **NºVulnerabilidad:** 18 **Reglamento**: CIS
Nombre: La seguridad avanzada de datos de SQL Server no tiene configurada un destinatario de las alertas.
Descripción: Advanced Data Security (ADS) proporciona un conjunto de capacidades de seguridad en SQL, tales como detección de vulnerabilidades, detección de amenazas, etc...
Recursos afectados: tfmsql

Severidad: Media **NºVulnerabilidad:** 19 **Reglamento**: CIS
Nombre: Las bases de datos SQL no están cifradas con claves gestionadas por el cliente.
Descripción: Gestionar de manera manual las claves de cifrado a través del servicio KeyVault permite mejorar el control de acceso a los datos almacenados.
Recursos afectados: tfmsql

Severidad: Baja **NºVulnerabilidad:** 20 **Reglamento**: CIS, PCI, RGPD, NIST
Nombre: Números de teléfono de contacto no incluidos en ASC
Descripción: Asegurar que los teléfonos de contacto del equipo de seguridad figuran en la configuración de Azure Security Center
Recursos afectados: default (ASC)

Severidad: Baja **NºVulnerabilidad:** 21 **Reglamento**: CIS
Nombre: Resource Groups sin resource locks
Descripción: Los resource locks son candados que permiten evitar cambios en los grupos de recursos. Es necesario imponer un candado en aquellos recursos o grupos de recursos importantes para evitar modificaciones accidentales o no autorizadas.
Recursos afectados: TFM-RG, NetworkWatcherRG, MC_TFM2-RG_AKS-Cluster_westeurope, cloud-shell-storage-westeurope

4.2.2. Plan de acción

En el caso de Azure, vemos que sólo se nos reportan dos vulnerabilidades de severidad alta:

- Puerto 3389 (RDP) abierto a internet (vulnerabilidad número 1)
- Puerto 22 (SSH) abierto a internet (vulnerabilidad número 2)

La remediación vista en AWS también aplica a este caso, filtrando por IP el acceso a estos puertos o, mejor aún, cerrando su acceso por completo.
Adicionalmente, Azure proporciona una herramienta para abrir los puertos exclusivamente cuando se van a utilizar (JIT, Just In Time), permitiendo de este modo la gestión remota de las máquinas virtuales comprometiendo al mínimo la seguridad del entorno.

En la siguiente vulnerabilidad de severidad alta, la número 3, vemos que en el clúster AKS no se ha habilitado el uso de roles RBAC (roles de Azure) para gestionarlos. Utilizar asignación de roles para los usuarios administradores del clúster permite controlar mucho más eficientemente qué personas pueden realizar cambios sobre el mismo, impidiendo cambios no autorizados que pudieran implicar afectación a un servicio.

A continuación, observando las vulnerabilidades de severidad media, 5 de ellas (de la 4 a la 8, inclusive) están relacionadas con el Azure Security Center (ASC). ASC es una herramienta nativa de Azure que permite reforzar la seguridad en el entorno. Activarlo y seguir las recomendaciones facilitadas por Prisma añade una capa de seguridad adicional muy importante, ya que ASC tiene una visibilidad de la infraestructura mucho más profunda que Prisma.

Se puede ver también en la vulnerabilidad número 9 que hay una STA (Storage Account) que no obliga a utilizar conexiones cifradas para el intercambio de datos. Utilizar cifrado en tránsito evita que la información se transmita en claro a través de la red, impidiendo que un usuario cualquiera pueda escuchar la comunicación y tenga acceso a datos sensibles.
Además, el registro de accesos de esta misma STA (junto con otra creada automáticamente para el almacenamiento del Cloudshell) está desactivado (vulnerabilidad número 10).
Como se ha visto en el caso de los buckets S3 de AWS, el no disponer de los datos de acceso implica que, ante una filtración de datos, no se pueda conocer el alcance de la misma y pueda repercutir en multas por mala praxis.

Continuando con las STAs, en la vulnerabilidad número 11 se nos alerta que su acción por defecto a la hora de comprobar accesos es “Allow” (permitido). Esto es equivalente a no tener restricción alguna en las reglas de entrada de un firewall. Si el contenido almacenado incluye datos personales o sensibles una persona que no está destinada a tener acceso a ellos podría no sólo leerlos, podría además publicarlos en páginas web de terceros o chantajear a la empresa.

La siguiente vulnerabilidad, la número 12, nos indica que la retención de los Flow Logs es inferior a 90 días. Los Flow Logs contienen la información de las transferencias de red realizadas en el entorno cloud, con información como Ips de origen y destino y puertos origen y destino.
Además de poder utilizar esta información para monitorizar los accesos a recursos sensibles, dada una infección por malware latente, que realiza comunicaciones periódicas a lo largo del tiempo, podría pasar desapercibida hasta que entrara en funcionamiento. Tener pruebas a nivel forense del origen del comportamiento ayudaría a localizar el vector de entrada de este malware y evitarlo en el futuro.

Al igual que en el entorno AWS, Prisma nos avisa que los discos duros asociados a las máquinas virtuales no están cifrados (vulnerabilidad número 13). Ya se han mencionado las consecuencias de dejar los discos duros desprotegidos contra lecturas no autorizadas, y su solución tiene un impacto mínimo en el rendimiento de las máquinas virtuales.

Observando la vulnerabilidad número 14, se nos alerta que las máquinas virtuales no tienen un agente antimalware instalado. Contar con una herramienta que activamente sea capaz y bloquear comportamientos maliciosos en las máquinas virtuales añade una capa de seguridad que ayuda a proteger la infraestructura.

A continuación, observamos un bloque de problemas encontrados en las bases de datos SQL (vulnerabilidades de la 15 a la 19, inclusive).
En la primera de ellas se reporta que el administrador no está gestionado a través de Azure Active Directory. Aprovechar las ventajas que aporta disponer de Azure Active Directory para asignar el o los administradores de las bases de datos a través de roles o grupos proporciona grandes ventajas a utilizar un nombre de usuario y una contraseña, como, por ejemplo:

- Posibilidad de utilizar doble factor de autenticación
- No existe la posibilidad de perder u olvidar la clave de acceso de administración
- No se almacenan indebidamente las claves de acceso

Todas estas ventajas permiten mejorar notablemente la seguridad de acceso a las bases de datos.
La vulnerabilidad 16 nos muestra que las reglas del firewall del servidor SQL permiten el tráfico desde todo internet (0.0.0.0/0). El primer paso para proteger adecuadamente una base de datos y evitar filtraciones de datos pasa por identificar qué redes deben acceder a la misma, y nunca tener la base de datos expuesta al público.
La vulnerabilidad 17 alerta sobre la ausencia de logado de la base de datos. No sólo es un requisito legal (evidenciar el tratamiento de la información de cara al RGPD), sino que, además, podremos conocer el alcance y motivo de una filtración de datos si esta llega a producirse.
La vulnerabilidad 18 nos informa que la herramienta Advanced Data Security (que monitoriza la seguridad de las bases de datos para detectar problemas) no tiene un destinatario a quien enviar las alertas detectadas. Esta herramienta proporciona información esencial para complementar la seguridad de la base de datos junto con lo que se reporta a través de CSPM, por lo tanto, es altamente recomendable configurarla correctamente.
Cerrando este bloque, la vulnerabilidad 19 nos indica que no se está aplicando el cifrado de los datos con claves gestionadas por el cliente (la organización). El uso de estas claves, aunque implique un proceso manual, es recomendable aplicarlo, puesto que mejora el control de acceso a las bases de datos.

Finalmente, en las vulnerabilidades de severidad baja vemos que se nos recomienda incluir un número de teléfono en las alertas de ASC (vulnerabilidad número 20), para en caso de disponer de un equipo 24x7 pueda hacerse cargo de ellas.
La otra vulnerabilidad de severidad baja nos indica que no hay locks (candados) en los Resource Group (vulnerabilidad número 21). Los candados tienen como utilidad principal la de evitar cambios no autorizados. Un usuario descuidado podría modificar recursos críticos, afectando al servicio.

Tras solucionar los problemas reportados, revisamos el nivel de cumplimiento a través del dashboard mostrado anteriormente:

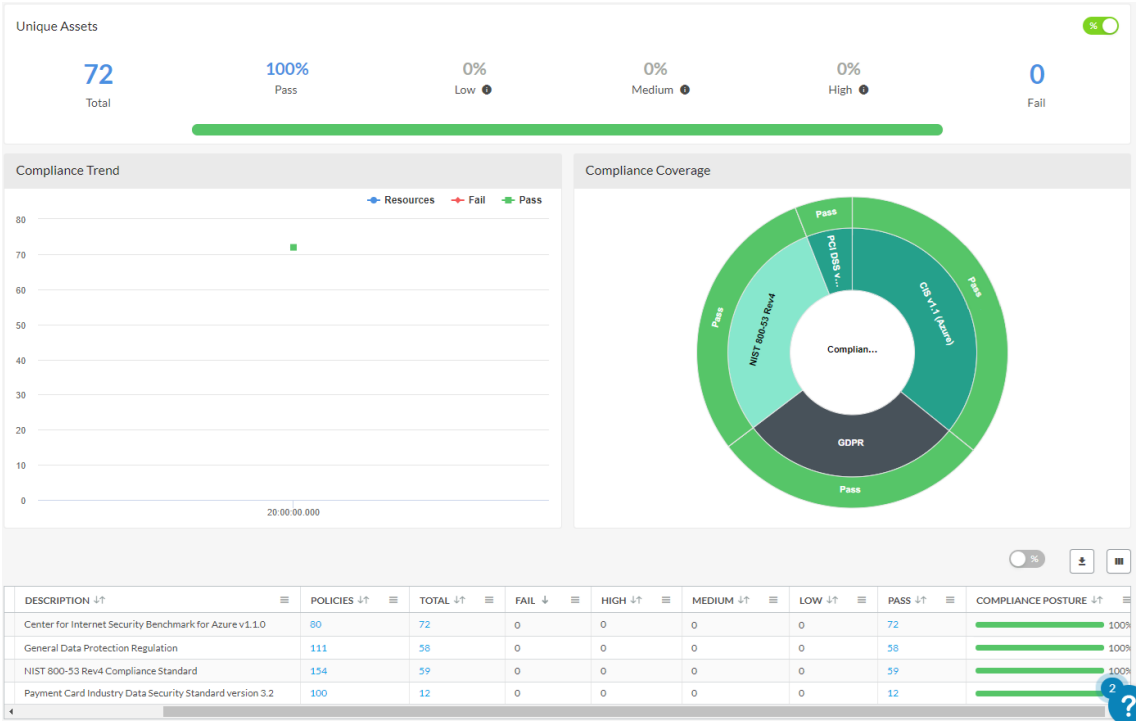


Ilustración 19: Dashboard de Prisma con el cumplimiento de los reglamentos utilizados al inicio

Vemos que se ha conseguido un 100% de cumplimiento en todos los reglamentos, CIS, PCI, NIST y RGPD.

5. Resultados Dome9

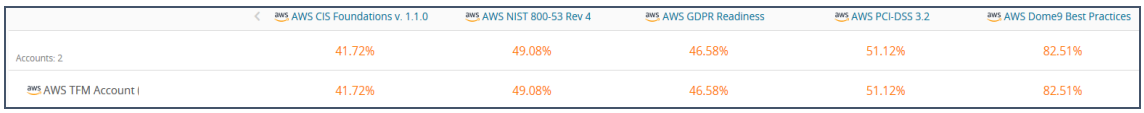
5.1. AWS

5.1.1. Vulnerabilidades detectadas

En el caso de Dome9 sobre AWS, obtenemos un total de 42 vulnerabilidades. El número de vulnerabilidades reportadas en este caso es superior a las vistas con Prisma, pese a ser el mismo entorno.

Aunque Dome9 proporciona un dashboard como se ha visto al inicio, este no muestra un cumplimiento por reglamento o región, como sí hacía Prisma.

En su lugar, ofrece cumplimiento por cada ruleset (o reglamento), indicado con un porcentaje. Sin embargo, no permite filtrar por región, haciendo que los recursos generados por defecto en AWS se muestren en dicho porcentaje, enmascarando el cumplimiento real dentro de la región donde se ha desplegado la infraestructura (París).



	AWS CIS Foundations v. 1.1.0	AWS NIST 800-53 Rev 4	AWS GDPR Readiness	AWS PCI-DSS 3.2	AWS Dome9 Best Practices
Accounts: 2	41.72%	49.08%	46.58%	51.12%	82.51%
AWS TFM Account I	41.72%	49.08%	46.58%	51.12%	82.51%

Ilustración 20: Nivel de cumplimiento por reglamento en el entorno de AWS antes de corregir las vulnerabilidades

Llama la atención que muchas de las recomendaciones consisten en generar alertas de métricas en CloudWatch, de manera similar a lo visto con Prisma en Azure y el Security Center.

Destaca de estas vulnerabilidades que se reporta la primera en lo que se refiere al balanceador de carga, habiendo pasado por alto en Prisma.

Severidad: Alta **NºVulnerabilidad:** 1 **Reglamento**: CIS, NIST
Nombre: Asegurar que el Security Group por defecto de la VPC restringe todo el tráfico.
Descripción: Las VPC se generan con un SG por defecto. Hay que asegurar que dicho SG sea lo suficientemente restrictivo para que, si se da el caso, se genera una instancia sin SG herede el de la VPC siendo este lo suficientemente seguro para no dejar la máquina expuesta.
Recursos afectados: default (SG por defecto de la VPC)

Severidad: Alta **NºVulnerabilidad:** 2 **Reglamento**: CIS, PCI, RGPD, NIST
Nombre: Asegurar que ningún SG permita el tráfico entrante a RDP (puerto 3389/TCP) desde 0.0.0.0/0
Descripción: El servicio RDP de Windows no puede estar expuesto a todo internet, ya que dada una vulnerabilidad un atacante podría hacerse con el control de la máquina.
Recursos afectados: launch-wizard-3, launch-wizard-2

Severidad: Alta **NºVulnerabilidad: 3** **Reglamento**: CIS, PCI, RGPD, NIST
Nombre: Asegurar que ningún SG permita el tráfico entrante a SSH (puerto 22/TCP) desde 0.0.0.0/0
Descripción: El servicio SSH de Linux no puede estar expuesto a todo internet, ya que dada una vulnerabilidad un atacante podría hacerse con el control de la máquina.
Recursos afectados: launch-wizard-1, eks-remoteAccess

Severidad: Alta **NºVulnerabilidad:** 4 **Reglamento**: CIS, PCI, NIST
Nombre: Reforzar la política de contraseñas
Descripción: Es necesario aplicar una política de contraseñas que fuerce el rotado de las mismas cada cierto tiempo, evite la repetición de las mismas y obligue a los usuarios a utilizar caracteres variados (mayúsculas, minúsculas, números y caracteres especiales)
Recursos afectados: Políticas de contraseñas

Severidad: Alta **NºVulnerabilidad:** 5 **Reglamento**: CIS, RGPD, PCI
Nombre: Asegurar que los logs de CloudTrail están cifrados utilizando CMKs
Descripción: AWS ofrece servicios de gestión de claves de cifrado, sin embargo, por defecto se utilizan claves conocidas por el CSP. Utilizar CMKs añade un nivel de seguridad al ser las claves totalmente confidenciales y ser el CSP agnóstico de estas.
Recursos afectados: cloudsploit-security-scanner-CloudSploitCloudTrail-TH0YCB80C5RU

Severidad: Alta **NºVulnerabilidad:** 6 **Reglamento**: CIS, RGPD, NIST
Nombre: Evitar el uso de la cuenta root
Descripción: Es altamente recomendable no utilizar la cuenta root porque es la que más privilegios tiene en todo el entorno AWS.
Recursos afectados: root

Severidad: Alta **NºVulnerabilidad:** 7 **Reglamento**: CIS, PCI, RGPD, NIST
Nombre: Asegurar que se está utilizando un dispositivo hardware para MFA para la cuenta root
Descripción: Utilizar MFA de tipo hardware para autenticar a la cuenta root aparte de su nombre de usuario y contraseña es necesario, puesto que se trata de la cuenta con más privilegios
Recursos afectados: root

Severidad: Alta **NºVulnerabilidad:** 8 **Reglamento**: CIS, PCI, RGPD, NIST
Nombre: Asegurar que se está utilizando un dispositivo virtual para MFA para la cuenta root
Descripción: Utilizar MFA de tipo virtual para autenticar a la cuenta root aparte de su nombre de usuario y contraseña es necesario, puesto que se trata de la cuenta con más privilegios
Recursos afectados: root

Severidad: Alta **NºVulnerabilidad:** 9 **Reglamento**: RGPD, NIST
Nombre: Asegurar que AWS Config está habilitado en todas las regiones
Descripción: Con AWS Config se consigue registrar los cambios realizados a los diferentes recursos del entorno, necesario para verificar el tratamiento de la información.
Recursos afectados: eu-west-3, eu-west-2, eu-west-1

Severidad: Alta **NºVulnerabilidad:** 10 **Reglamento**: RGPD, NIST
Nombre: Asegurar que los Flow Logs de las VPC están habilitados en todas las regiones aplicables
Descripción: Es recomendable activar los Flow Logs en la VPC, de este modo se proporciona visibilidad de las comunicaciones que transcurren en la VPC.
Recursos afectados: eu-west-3, eu-west-2, eu-west-1

Severidad: Alta **NºVulnerabilidad:** 11 **Reglamento**: RGPD, NIST
Nombre: Instancias detectadas sin escaneos de AWS Inspector en los últimos 30 días
Descripción: Se recomienda que AWS Inspector escanee todas las instancias al menos una vez a la semana.
Recursos afectados: WinServer_2019_1, WinServer_2019_2, WebServer_Linux, Kubernetes-Nodo1, Kubernetes-Nodo2

Severidad: Alta **NºVulnerabilidad:** 12 **Reglamento**: PCI, RGPD, NIST
Nombre: Usar cifrado para operaciones de escritura en buckets S3
Descripción: Asegurar que solo se permiten transferencias de datos utilizando SSL.
Recursos afectados: s3tfmseryoferben, cf-templates-1ewjr29s9uj7t-eu-west-3

Severidad: Alta **NºVulnerabilidad:** 13 **Reglamento**: PCI, RGPD, NIST
Nombre: Usar cifrado SSL en los buckets S3
Descripción: Asegurar que solo se permiten transferencias de datos utilizando SSL.
Recursos afectados: s3tfmseryoferben, cf-templates-1ewjr29s9uj7t-eu-west-3

Severidad: Alta **NºVulnerabilidad:** 14 **Reglamento**: RGPD, NIST
Nombre: Asegurar que se debe utilizar MFA para eliminar buckets S3
Descripción: Asegurar que solo se permiten transferencias de datos utilizando SSL.
Recursos afectados: s3tfmseryoferben, cf-templates-1ewjr29s9uj7t-eu-west-3

Severidad: Alta **NºVulnerabilidad:** 15 **Reglamento**: RGPD
Nombre: Asegurar que los buckets S3 están cifrados en reposo
Descripción: Los buckets S3 deben tener el cifrado en reposo habilitado para asegurar que los datos sensibles están protegidos.
Recursos afectados: s3tfmseryoferben

Severidad: Alta **NºVulnerabilidad:** 16 **Reglamento**: PCI, RGPD, NIST
Nombre: Cifrar las bases de datos RDS
Descripción: Utilizar el cifrado en reposo para las bases de datos RDS y para sus snapshots.
Recursos afectados: database-users-instance-1

Severidad: Alta **NºVulnerabilidad:** 17 **Reglamento**: CIS, NIST
Nombre: Asegurar que la política de seguridad del balanceador está actualizada
Descripción: Los balanceadores realizan una negociación TLS entre el cliente y el propio balanceador. Es necesario actualizar la versión de TLS.
Recursos afectados: BackendLB

Severidad: Alta **NºVulnerabilidad:** 18 **Reglamento**: PCI, NIST
Nombre: Security Groups con puertos de administración sobreexpuestos a internet
Descripción:Los Security Groups son la primera capa de seguridad de red, por tanto, no deben permitir tráfico entrante a puertos administrativos.
Recursos afectados: launch-wizard-1, launch-wizard-2, launch-wizard-3, eks-remoteAccess

Severidad: Alta **NºVulnerabilidad:** 19 **Reglamento**: CIS, NIST
Nombre: Deshabilitar la asignación de IP pública automática en las subnets
Descripción: Esta funcionalidad permite asignar de manera automática una dirección IP pública a las instancias que se generen en la subnet, exponiéndolas de manera inmediata a internet y pudiendo ser objetivo de ataques.
Recursos afectados: BackendSubnet, DMZ

Severidad: Media **NºVulnerabilidad:** 20 **Reglamento**: CIS, RGPD, NIST
Nombre: Asegurar que las políticas IAM sólo se utilizan en grupos o roles.
Descripción: Es recomendable que las políticas IAM se asocien directamente a grupos y roles en lugar de usuarios, de este modo se consigue un mayor control de la gestión de accesos.
Recursos afectados: testCLI

Vulnerabilidades de la 21 a la 39 inclusive, referentes a alertas de CloudWatch y políticas de contraseñas, localizadas en el anexo.

Severidad: Media **NºVulnerabilidad:** 40 **Reglamento**: PCI, RGPD, NIST
Nombre: Los buckets S3 deben logar los accesos
Descripción: Es recomendable activar el logado de accesos en todos los buckets para asegurar tener trazabilidad de todos los accesos y cambios.
Recursos afectados: s3tfmseryoferben, cf-templates-1ewjr29s9uj7t-eu-west-3

Severidad: Media **NºVulnerabilidad:** 41 **Reglamento**: RGPD
Nombre: Asegurar que la política de retención de las bases de datos RDS es de al menos 7 días
Descripción: Se recomienda que las bases de datos RDS estén configuradas para mantener al menos 7 días de backups.
Recursos afectados: database-users-instance-1

Severidad: Media **NºVulnerabilidad:** 42 **Reglamento**: NIST
Nombre: Asegurar que los SG restringen el tráfico de salida
Descripción: Se recomienda que ningún SG permita el tráfico de salida de manera indiscriminada.
Recursos afectados: launch-wizard-1, launch-wizard-2, launch-wizard-3, default, eks-remoteAccess

5.1.2. Plan de acción

Dome9 nos reporta ni más ni menos que 19 vulnerabilidades de severidad alta en nuestro entorno de pruebas de AWS.

En primer lugar, se nos advierte de que los Security Group por defecto de la VPC son demasiado permisivos. Como medida base al crear una nueva VPC, es indispensable asignarle un SG absolutamente restrictivo. De este modo, las instancias EC2 y otros recursos que se desplieguen en dicha VPC no quedarán expuestos a todas las comunicaciones entrantes.

Las vulnerabilidades 2 y 3 nos reportan que hay SGs sobreexponiendo los puertos 22 y 3389, respectivamente. Estos puertos son utilizados para la gestión de las máquinas y son, por lo tanto, los más atacados para intentar hacerse con el control de las máquinas (y en el caso de RDP, existen numerosas vulnerabilidades asociadas al protocolo).

A continuación, en la vulnerabilidad número 4, se nos insta a reforzar la política de contraseñas. Las contraseñas de los usuarios son la primera barrera contra intrusos. Reforzarlas obligando a los usuarios a utilizar contraseñas complejas evita que usen contraseñas demasiado simples que puedan encontrarse en diccionarios públicos.

Se reporta en la vulnerabilidad número 5 que no se están utilizando claves gestionadas por el cliente (la organización) para cifrar los datos de CloudTrail. Como se ha visto anteriormente, utilizar este tipo de claves en lugar de las gestionadas por el proveedor de servicios Cloud ayuda a controlar mejor el acceso a los datos cifrados, ya que se limitan los usuarios que tienen acceso a dichas claves.

Agrupando por las que afectan al usuario root de la cuenta observamos:
- Se ha utilizado recientemente (vulnerabilidad número 6)
- No tiene MFA virtual (vulnerabilidad número 7)
- No tiene MFA físico (vulnerabilidad número 8)

Como se ha comentado en el plan de acción de Prisma con AWS, se desaconseja el uso de la cuenta root por su exceso de permisos.
En cuanto al aspecto del MFA, se podría agrupar en una única alerta que nos indicara que se activara el uso de MFA, ya que el ser virtual o físico no afecta al cumplimiento de esta norma.

Se nos indica que activemos el uso de AWS Config en todas las regiones (vulnerabilidad número 9). AWS Config nos permite auditar la actividad de los usuarios y detectar anomalías, generando una alerta de seguridad que nos podría permitir cortar el acceso al usuario si se confirma que está llevando a cabo actividades maliciosas o de dudosa ética.

En la vulnerabilidad número 10 se detecta que no están activados los Flow Logs en las regiones en uso. Estos logs proporcionan información de la actividad ocurrida a nivel de red en el entorno y permite detectar anomalías, además de ser útiles en análisis forense. Adicionalmente, es necesario activarlo para cumplir el RGPD y llevar el control del acceso a los datos que se usan en el servicio.

Se reporta que no se encuentran escaneos realizados con AWS Inspector en las instancias EC2 (vulnerabilidad número 11). AWS Inspector es una herramienta que permite descubrir problemas de configuración internos de la máquina, vulnerabilidades, etc… Por lo tanto, puede proporcionar información muy valiosa de cara a solucionar problemas de seguridad.

A continuación, observamos varias vulnerabilidades relacionadas con el cifrado, tanto en reposo como en tránsito, de los buckets S3 (vulnerabilidades de la 12 a la 15, inclusive). Como se ha visto anteriormente, el cifrado en tránsito es necesario y obligatorio no sólo por cumplimiento, sino para evitar el acceso indebido a los datos mientras viajan por la red.
Para complementar este cifrado, se debe activar también el cifrado en reposo de los datos, para evitar accesos no autorizados a aquellos usuarios que no dispongan de las claves de cifrado.

Como recomendación adicional, se nos sugiere activar MFA para borrado de los buckets. Con esta medida podemos evitar no sólo un borrado accidental, se evitaría además que un intruso borrara intencionadamente el bucket junto con todos sus datos.

Dome9 nos notifica en la vulnerabilidad número 16 de una base de datos RDS con el cifrado en reposo deshabilitado. Por el mismo motivo que los buckets S3 necesitan cifrado en reposo, se recomienda activarlo para las bases de datos, con el agravante de que es conocido que almacenan datos sensibles de usuarios.

Vemos que también se nos recomienda actualizar la versión de cifrado TLS utilizado por el balanceador de carga (vulnerabilidad número 17). Se ha demostrado que las versiones antiguas de TLS (anteriores a 1.2) han sido comprometidas y, por lo tanto, el cifrado en tránsito puede romperse, dejando al descubierto datos confidenciales o altamente sensibles.

Cerrando el bloque de vulnerabilidades de severidad alta nos encontramos con problemas relacionados con la gestión de redes:

- SGs con puertos de administración expuestos a todo internet (vulnerabilidad 18)
- Está habilitado en la subnet la asignación automática de direcciones IP públicas (vulnerabilidad 19)

Estas dos vulnerabilidades en conjunto representan un problema de extraordinaria gravedad, ya que, de manera automática, los recursos que se desplieguen en estas subredes quedan expuestos a internet a través de puertos de administración (22, 3389), dejando una puerta abierta a los atacantes a nuestra infraestructura.
Es importante, por tanto, no sólo segmentar por IP el acceso a puertos administrativos, sino también desactivar la asignación automática de IPs públicas y asignarlas de manera manual sólo a aquellos recursos que realmente lo necesiten.

A continuación, pasamos a las vulnerabilidades de severidad media. La primera de ellas (número 20) nos reporta un usuario (testCLI, creado para utilizar los comandos de AWS en una terminal) que tiene asignadas políticas de permisos directamente, en lugar de tenerlas asignadas a través de la pertenencia a grupos.
Asignar los permisos directamente a los usuarios en lugar de utilizar grupos dificulta la gestión de los mismos, aumentando la probabilidad de asignar permisos excesivos o, en el peor de los casos, olvidar que hay usuarios que pueden realizar determinadas acciones.

Ahora se nos presentan dos grandes grupos de vulnerabilidades:

- Relacionadas con alertas de monitorización de usuarios en Cloudwatch (vulnerabilidades de la 21 a la 34)
- Relacionadas con las políticas de contraseñas (de la 35 a la 39)

El primer grupo hace referencia a las métricas configurables en AWS Config. Como se ha visto, es una herramienta muy potente que permite obtener patrones de comportamiento, y Dome9 sugiere una buena selección de políticas para securizar el entorno.
Además, la primera de ellas nos indica que los logs almacenados en CloudTrail no están integrados con esta herramienta. De esta manera, especifica todos los puntos a remediar para asegurar que la herramienta Cloudwatch funciona como se espera.
En el caso de las contraseñas se repite lo visto en Prisma al analizar la plataforma de AWS (contraseñas que utilicen letras mayúsculas, minúsculas, letras, símbolos, etc…).

Por otro lado, Dome9 ha identificado buckets S3 que no están almacenando la información de acceso a sus datos (vulnerabilidad número 40). Como se ha visto, no solo se incumple el RGPD por no monitorizar el acceso y tratamiento de los datos, sino que implica que, ante un incidente de seguridad y su posterior análisis forense, no podemos acotar el alcance de una filtración de datos o intrusión.

En la vulnerabilidad número 41 Dome9 ha identificado que no se están almacenando backups de la base de datos RDS en al menos 7 días.
La necesidad de copias de seguridad es obvia a estas alturas, y según los datos almacenados (por ejemplo, una base de datos de transacciones) estas copias deben realizarse con bastante frecuencia de cara a minimizar la pérdida de datos en caso de un incidente. Dome9 recomienda que se realicen como mínimo cada 7 días.

Finalmente, en la vulnerabilidad 42 se reporta que no se está controlando el tráfico de salida a través de los Security Groups. En caso de ser conexiones fácilmente identificables no supone un problema limitar el tráfico de salida a los destinos conocidos, pero si se debe monitorizar la salida a internet se debería utilizar una solución dedicada (proxy de navegación) y limitar el tráfico de salida del Security Group para que sólo se pueda acceder al proxy, además de a los recursos internos de la infraestructura que así lo requieran.

Como punto negativo de la herramienta, y como se ha mostrado al inicio de las vulnerabilidades reportadas por esta, el nivel de cumplimiento mostrado apenas se vio afectado tras solucionar los problemas reportados, debido en gran parte a la imposibilidad de filtrar por región y contabilizar Dome9 recursos por defecto que crea AWS en otras regiones.

5.2. Azure

5.2.1. Vulnerabilidades detectadas

En el entorno de Azure, Dome9 nos reporta un total de 27 vulnerabilidades diferentes.

Como se ha visto, la visibilidad del cumplimiento por reglamento no es tan gráfica como en Prisma, limitándonos a ver un porcentaje. Sin embargo, al estar desplegado todo el entorno de Azure en una única región y que, al contrario que AWS, no crea recursos por defecto en regiones que no se utilizan, estos datos son mucho más representativos y realistas que los vistos en AWS.

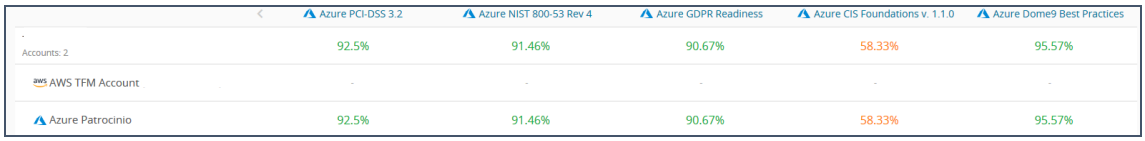


Ilustración 21: Cumplimiento por reglamento según Dome9 en Azure

Observamos que se aproxima más a lo visto en Prisma (en AWS, Dome9 proporcionaba más información que Prisma), ofreciéndonos igualmente numerosas indicaciones para Azure Security Center.

Severidad: Alta **NºVulnerabilidad:** 1 **Reglamento**: CIS, PCI, RGPD, NIST
Nombre: Asegurar que el acceso de entrada de SSH está restringido desde internet
Descripción: Deshabilitar el tráfico SSH de entrada desde internet en los NSG.
Recursos afectados: UbuntuServer1804-nsg

Severidad: Alta **NºVulnerabilidad:** 2 **Reglamento**: CIS, PCI, NIST
Nombre: Asegurar que el acceso de entrada de RDP está restringido desde internet
Descripción: Deshabilitar el tráfico RDP de entrada desde internet en los NSG.
Recursos afectados: WinServer-2019-1-nsg, WinServer-2019-2-nsg

Severidad: Alta **NºVulnerabilidad:** 3 **Reglamento**: CIS
Nombre: Asegurar que la regla de red por defecto de las STA es “Deny”
Descripción: Restringir por defecto todo el tráfico de entrada a las Storage Account ayuda a añadir una capa de seguridad para evitar accesos no autorizados.
Recursos afectados: tfmsta, csb10032000a2811712

Severidad: Alta **NºVulnerabilidad:** 4 **Reglamento**: CIS, PCI, RGPD, NIST
Nombre: Asegurar que el cifrado en tránsito está habilitado para las STAs
Descripción: Se debe restringir el tráfico de la STA a HTTPS exclusivamente para asegurar la protección de los datos en la red.
Recursos afectados: tfmsta

Severidad: Alta **NºVulnerabilidad:** 5 **Reglamento**: CIS
Nombre: Asegurar que existe un perfil de logado de la actividad
Descripción: Habilitar un perfil de logado para exportar los logs de actividad a otra plataforma con más de 90 días de retención, de cara a posibles investigaciones.
Recursos afectados: Suscripción

Severidad: Alta **NºVulnerabilidad:** 6 **Reglamento**: CIS
Nombre: Asegurar que el Network Watcher está activo en la suscripción
Descripción: Es necesario habilitar el Network Watcher en la suscripción para tener constancia de los flujos de red.
Recursos afectados: Suscripción

Severidad: Alta **NºVulnerabilidad:** 7 **Reglamento**: NIST
Nombre: Asegurar que el acceso a la base de datos SQL no está permitido para toda la infraestructura
Descripción: Especificar en las reglas de firewall del servidor SQL las redes que deben tener acceso a este.
Recursos afectados: tfmsql

Severidad: Alta **NºVulnerabilidad:** 8 **Reglamento**: NIST
Nombre: Asegurar que existe una política de red para securizar el tráfico entre pods en el cluster AKS
Descripción: Se debe asegurar la segmentación de microservicios contenidos en el AKS aplicando el criterio de menor privilegio posible.
Recursos afectados: kubernetes-cluster

Severidad: Media **NºVulnerabilidad:** 9 **Reglamento**: CIS
Nombre: Asegurar que los discos duros de las máquinas virtuales están cifrados
Descripción: Asegurando que los discos duros están cifrados, siempre que sea posible, se consigue evitar la lectura no autorizada de los mismos.
Recursos afectados: aks-nodepool1-98637346-vmss_0, aks-nodepool1-98637346-vmss_1, UbuntuServer1804, WinServer-2019-1, WinServer-2019-2

Vulnerabilidades de la 10 a la 23 relacionadas con Azure Security Center agrupadas en el anexo.

Severidad: Media **NºVulnerabilidad:** 24 **Reglamento**: PCI, NIST
Nombre: Asegurar que los datos del servidor SQL se encuentran replicados en sistemas redundantes
Descripción: La replicación de datos de los servidores debe estar habilitada para evitar pérdidas de datos.
Recursos afectados: tfmsql

Severidad: Media **NºVulnerabilidad:** 25 **Reglamento**: CIS, RGPD, PCI, NIST
Nombre: Asegurar que el administrador del servidor se gestiona a través de Active Directory
Descripción: Gestionar la identidad del administrador de las bases de datos a través del Active Directory permite facilitar la gestión de los usuarios privilegiados y mejorar el control de los mismos.
Recursos afectados: tfmsql

Severidad: Media **NºVulnerabilidad:** 26 **Reglamento**: CIS
Nombre: Asegurar que las bases de datos están cifradas utilizando claves gestionadas por el cliente (CMKs)
Descripción: Cifrar las bases de datos utilizando claves gestionadas por la organización en lugar de gestionadas por el CSP mejora el control de accesos y se limita aún más el acceso a la información.
Recursos afectados: tfmsql

Severidad: Media **NºVulnerabilidad:** 27 **Reglamento**: CIS
Nombre: Asegurar que el perfil de logado captura todas las actividades que se realizan en la suscripción
Descripción: Registrar toda la actividad en el plano de gestión del entorno permite localizar anomalías de comportamiento y verificar el alcance de los daños ante un incidente de seguridad.
Recursos afectados: Suscripción

5.2.2. Plan de acción

Al igual que los datos de vulnerabilidades obtenidos en Prisma, Dome9 nos reporta en las vulnerabilidades 1 y 2 que los NSG que están permitiendo el acceso indiscriminado a los puertos 22 y 3390, asociados a SSH y RDP respectivamente.

También nos notifica de los problemas de las STAs ya vistos en Prisma en las vulnerabilidades 4 y 5, avisando que no filtran el acceso a nivel de red ni utilizan el cifrado en tránsito.

Sin embargo, Dome9 ha detectado dos nuevas vulnerabilidades que no habíamos visto en Prisma:

- Ausencia de un perfil de logado (vulnerabilidad 5)
- Activar el Network Watcher (vulnerabilidad 6)

En el caso de la primera, Azure no loga por defecto toda la actividad realizada en la consola, siendo necesario activar un perfil avanzado de logado para asegurar que a nivel forense y de cumplimiento se registra toda la actividad.

En cuanto al Network Watcher, es una herramienta de Azure que recoge la información de los NSG y la convierte en Flow Logs, utilizados para registrar las comunicaciones de red. Ya se ha visto la importancia de estos datos, activar el Network Watcher es una manera de garantizar que estos datos quedan correctamente registrados para su consulta.

A continuación, la vulnerabilidad número 7 nos reporta que no se están controlando las redes que tienen acceso al servidor SQL a través de su firewall. Las bases de datos son de los elementos más sensibles debido a su propósito de almacenar datos confidenciales y personales. Por lo tanto, uno de los pasos para evitar accesos no autorizados pasa por inventariar las redes y dispositivos que deban tener acceso y permitir su acceso en el firewall, denegando el resto del tráfico entrante.

La última de las vulnerabilidades de severidad alta, la número 8, nos insta a crear una política de comunicaciones a nivel del clúster de Kubernetes, del mismo modo que se utilizan los NSG para las máquinas virtuales. En este caso, la segmentación de red se aplica a nivel de pods (contenedores), consiguiendo una segmentación por microservicios.

En cuanto a vulnerabilidades de severidad media, vemos en la vulnerabilidad número 9 que se detectan discos duros sin cifrar, donde se incluyen también los discos duros de los nodos del clúster de Kubernetes.
Como se ha visto anteriormente, en función de los datos almacenados (si son de carácter personal o financieros) es obligatorio cifrarlos. Y aunque no sea obligatorio, siempre es una buena práctica para evitar filtraciones de datos.

A continuación, nos encontramos un bloque de recomendaciones (vulnerabilidades de la 10 a la 23) ofrecidas para Azure Security Center (ASC), muy similares a las vistas anteriormente en Prisma. Estas recomendaciones indican que se deben habilitar todas las políticas integradas que tiene ASC, entre las cuales se incluyen, por ejemplo, monitorizar que las máquinas tengan el sistema operativo actualizado o las vulnerabilidades asociadas al software de la máquina.

Posteriormente se nos presenta un bloque de vulnerabilidades relacionadas con las bases de datos SQL:

- Ausencia de redundancia (vulnerabilidad 24)
- Gestionar el usuario administrador a través de Azure Active Directory (vulnerabilidad 25)
- No se está usando el cifrado a través de claves gestionadas por el cliente o CMK (vulnerabilidad 26)

En la primera de ellas se nos insta a aprovechar las capacidades de un entorno Cloud como Azure para evitar pérdida de servicio si se sufre un incidente con alguna de las bases de datos (ya sea un incidente de seguridad o de otro tipo, como un desastre natural).
Del mismo modo que ha reportado anteriormente Prisma, la recomendación de Dome9 es crear el usuario administrador de las bases de datos a través del Azure Active Directory, lo cual permite aprovechar las herramientas de gestión de usuarios y roles del directorio, facilitando el control de un rol tan sensible.
Cerrando este grupo, vemos que no se han configurado claves de cifrado gestionadas por el cliente (la organización, en este caso). Como se ha descrito en otras vulnerabilidades similares, gestionar de manera manual las claves de cifrado permite restringir el acceso a las mismas, evitando que usuarios no autorizados consigan ver información clasificada como sensible.

Por último, la vulnerabilidad 27 reporta que no se están logando todas las acciones llevadas a cabo por los usuarios en el plano de gestión de la plataforma. Asegurar que se registran todas las actividades es esencial para conseguir un control y monitorización del entorno que de otro modo sería mucho más complicado. Adicionalmente, en caso de sufrir un incidente de seguridad, estos datos pueden proporcionar información muy relevante a averiguar lo ocurrido.

Una vez atajados todos estos problemas, revisamos de nuevo el nivel de cumplimiento por reglamento:

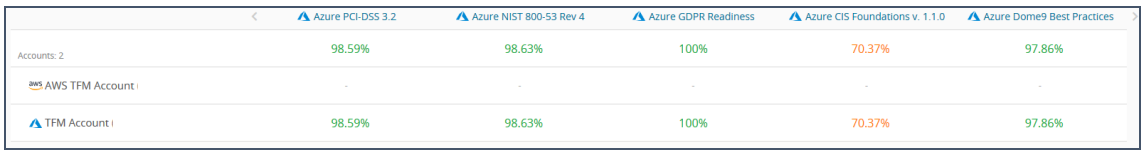


Ilustración 22: Nivel de cumplimiento por reglamento después de securizar el entorno de Azure

Observamos que, efectivamente, el nivel de cumplimiento ha mejorado de manera global, llegando incluso al 100% en el caso del RGPD.

Sin embargo, no es así en el resto de los reglamentos. Profundizando en la herramienta, observamos que existen numerosas vulnerabilidades pendientes:

Ilustración 23: Vista de vulnerabilidades marcadas como pendientes por Dome9 en el entorno de Azure

Vamos a revisar algunas de ellas en detalle:

- Asegurar que el acceso a RDP está restringido desde internet:

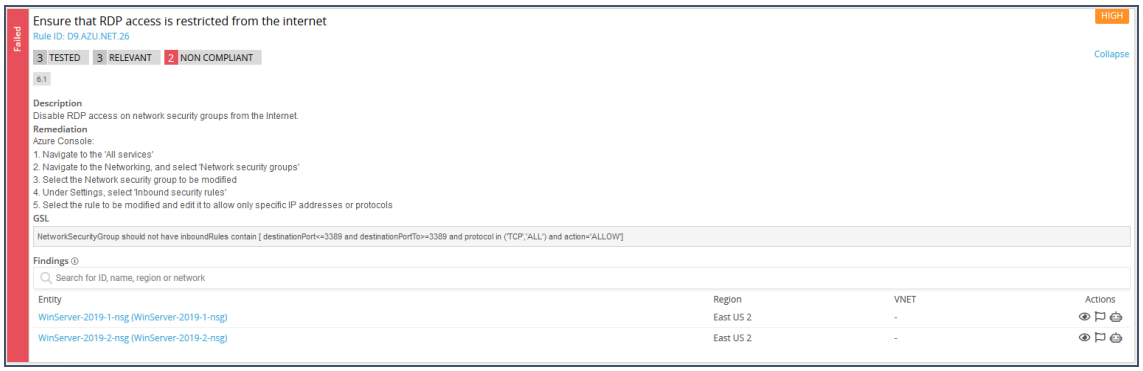


Ilustración 24: Dome9 reportando dos NSG que teóricamente permiten el tráfico entrante a RDP a todo internet

Vemos que Dome9 nos reporta que los NSG asociados a las dos máquinas Windows permiten el tráfico entrante desde todo internet, sin embargo, en Azure, dichos NSG están configurados del siguiente modo, limitando el acceso a RDP a una IP concreta, como se sugiere en la remediación de Dome9:

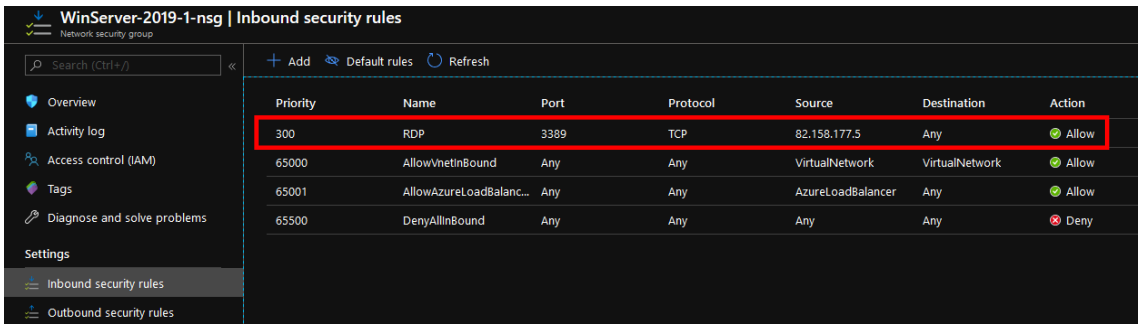


Ilustración 25: Evidencia de que se está restringiendo el tráfico entrante a RDP en uno de los NSG reportados

- Asegurar que la acción por defecto de red de las STA es “deny” y asegurar que los servicios confiables de Microsoft pueden acceder a la Storage Account:

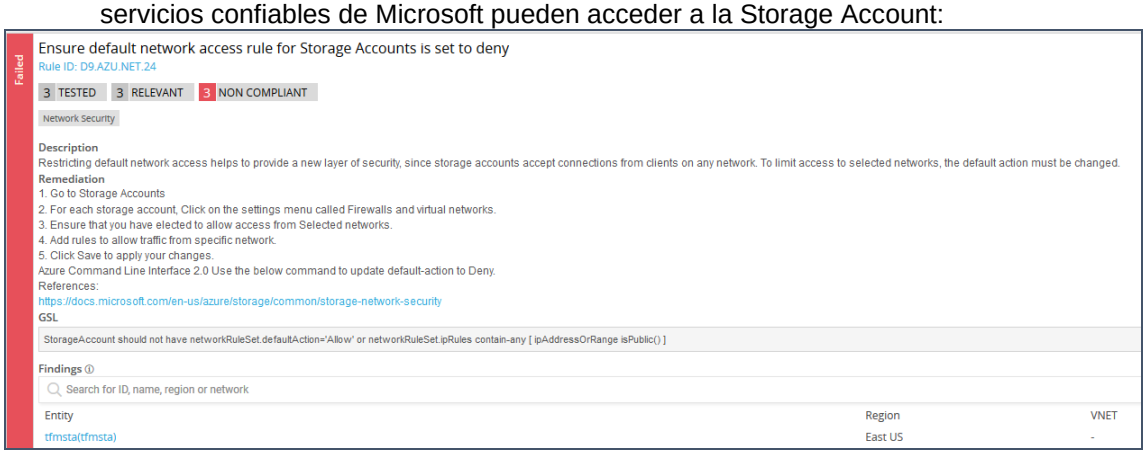


Ilustración 26: Dome9 reportando una STA que supuestamente no filtra el tráfico entrante

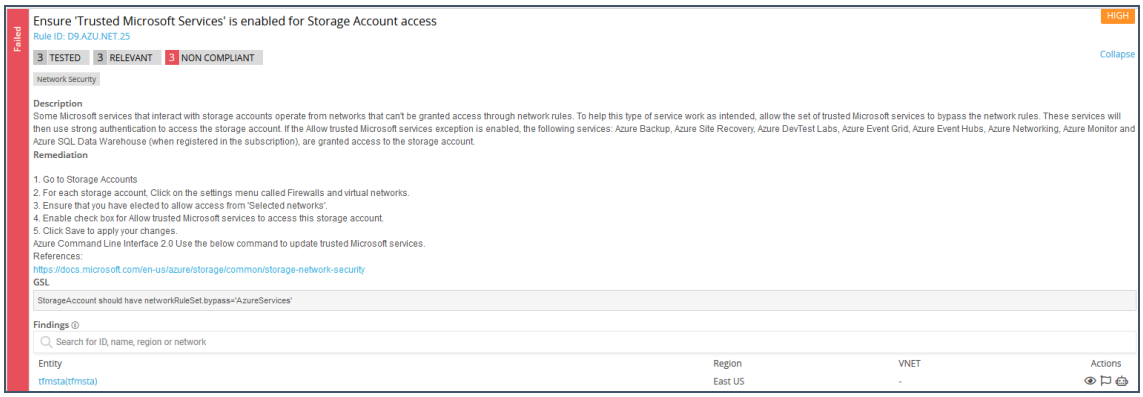


Ilustración 27:Dome9 reportando una STA que supuestamente no permite el acceso a los servicios confiables de Microsoft

Vemos que Dome9 nos está indicando que una Storage Account sigue sin cumplir con lo reportado al inicio en lo referente a comunicaciones y accesos confiables de Microsoft. Sin embargo, cuando accedemos a la configuración del recurso se observa que se ha configurado tal y como se sugiere en la remediación proporcionada:

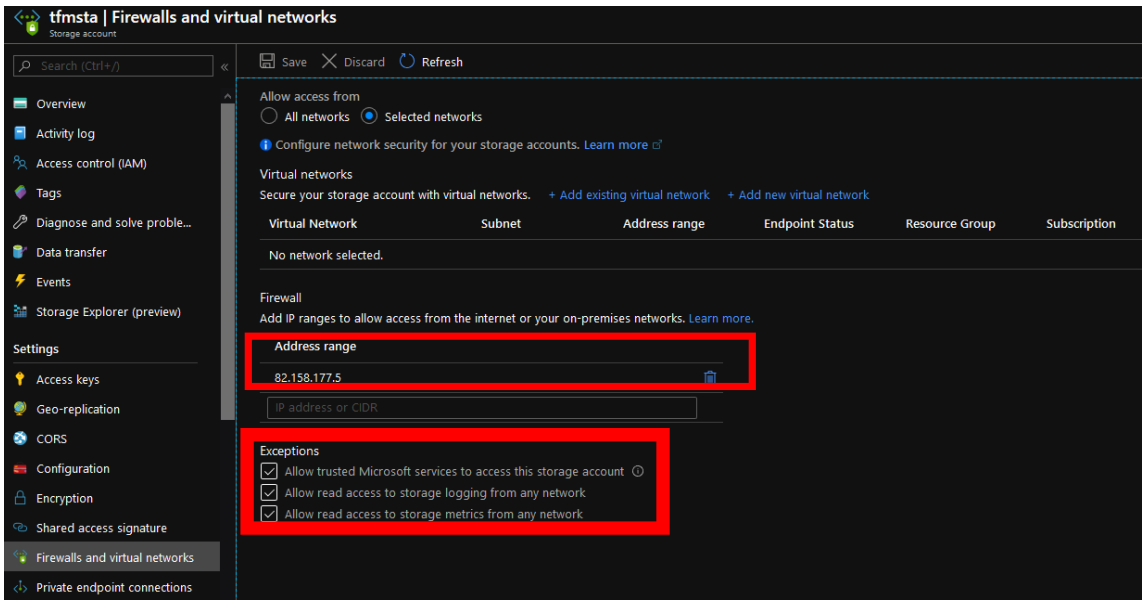


Ilustración 28: Evidencia de que los problemas reportados en la STA están resueltos tal y como indica Dome9

Aunque son solo unos ejemplos, este problema se repetía con todas las vulnerabilidades que la herramienta seguía marcando como no solucionadas (similar al problema de actualización de los dashboards de Prisma, pero en ese caso, sí aparecían como resueltas las vulnerabilidades al buscarlas).

Se realizaron muchas pruebas para solucionar este problema sin éxito, evidenciando que los datos reportados por la herramienta no son confiables al 100%.

6. Resultados CloudSploit

6.1. AWS

6.1.1. Vulnerabilidades detectadas

Finalmente, pasamos a los resultados obtenidos por la herramienta Open Source, CloudSploit.

Pese a ser el mismo entorno que en las anteriores herramientas, observamos únicamente un total de 13 vulnerabilidades, quedando por detrás de las herramientas comerciales en este caso.

La manera que tiene de reportar los resultados a través de sus dashboards hace que sea difícil el seguimiento de los problemas que se detectan en cada cuenta:

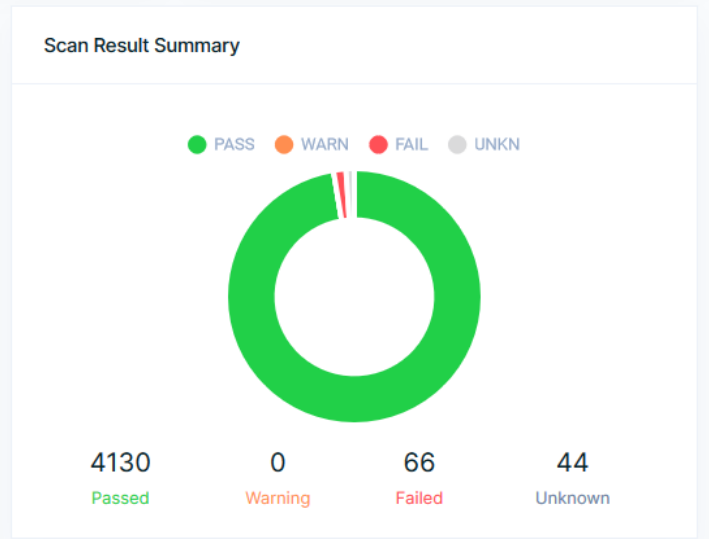


Ilustración 29: Dashboard de cumplimiento de Cloudsploit en el entorno AWS

No existe una manera de filtrar por región (aparecen recursos de todas las regiones generados automáticamente por AWS como VPCs). Adicionalmente, aplica políticas de recursos en esas regiones, aunque no haya ninguno desplegado (por ejemplo, si hay una política que aplica a buckets S3, va a aplicarla en cada región indicando que el recurso sobre el que la ha aplicado es vacío), de ahí que indique que existen más de 4000 recursos que cumplen con los reglamentos.

Por último, no se puede filtrar tampoco por reglamento, de modo que una misma política aparece como múltiples fallos, no porque cada recurso se cuente como un fallo, si no porque están duplicadas las políticas en cada reglamento, haciendo que este cuadro de mando ofrezca nula visibilidad.

Severidad: Alta **NºVulnerabilidad:** 1 **Reglamento**: CIS, RGPD, PCI **Nombre**: Evitar el uso de la cuenta root **Descripción**: La cuenta root tiene acceso ilimitado a todos los recursos de AWS. Es altamente recomendable evitar su uso. Esta vulnerabilidad viola el artículo 25 del RGPD (protección de datos por diseño y por defecto). **Recursos afectados**: root

Severidad: Alta **NºVulnerabilidad:** 2 **Reglamento**: CIS, RGPD, PCI
Nombre: Asegurar el uso de MFA
Descripción: La cuenta root es el usuario con más privilegios en AWS. La autenticación multifactor añade una capa de seguridad adicional respecto al acceso por usuario y contraseña. Esta vulnerabilidad viola el artículo 25 del RGPD (protección de datos por diseño y por defecto).
Recursos afectados: root

Severidad: Alta **NºVulnerabilidad:** 3 **Reglamento**: CIS
Nombre: Asegurar que no hay security groups que permitan el acceso al puerto 22 desde 0.0.0.0/0
Descripción: Se ha detectado un security group permitiendo el tráfico sin filtrar al puerto 22 de entrada, utilizado por el servicio SSH de gestión remota de máquinas Linux. Es recomendable que ningún security group permita la entrada al puerto 22.
Recursos afectados: launch-wizard-1

Severidad: Media **NºVulnerabilidad:** 4 **Reglamento**: RGPD, PCI
Nombre: Asegurar que el cifrado para los datos en reposo está habilitado en las bases de datos RDS
Descripción: Se han de implementar las medidas técnicas necesarias para proteger información personal o bancaria.
Recursos afectados: database-users-instance-1

Severidad: Media **NºVulnerabilidad:** 5 **Reglamento**: CIS, PCI
Nombre: Asegurar que los security group por defecto de las VPC bloquean por defecto todo el tráfico
Descripción: Una VPC tiene asignado por defecto un security group que debería denegar todo el tráfico entrante y permitir todo el tráfico saliente. De este modo se asegura que sólo se exponen los servicios que expesamente deben estar publicados a internet, reduciendo los vectores de entrada de posibles ataques.
Recursos afectados: sg-899385c5, sg-195b2f7a (security groups por defecto de las dos subredes, DMZ y BackendSubnet)

Severidad: Media **NºVulnerabilidad:** 6 **Reglamento**: CIS
Nombre: Asegurar que ningún security group permite el tráfico entrante al puerto 3389 sin filtrar
Descripción: Se detecta un security group que permite el tráfico entrante al puerto 3389, utilizado por el servicio RDP para gestionar máquinas Windows.
Recursos afectados: launch-wizard-2, launch-wizard-3

Severidad: Media **NºVulnerabilidad:** 7 **Reglamento**: CIS, PCI
Nombre: La cuenta no tiene políticas de contraseñas
Descripción: No se han detectado políticas de contraseñas activas que obliguen a los usuarios a aumentar la robustez de sus contraseñas incluyendo números o caracteres especiales.
Recursos afectados: Política de contraseñas

Severidad: Baja **NºVulnerabilidad:** 8 **Reglamento**: CIS, RGPD, PCI
Nombre: Asegurar que la política de contraseñas evita la reutilización de las mismas
Descripción: Para aumentar la seguridad en la infraestructura, es necesario mejorar la seguridad de las contraseñas de acceso de los usuarios.
Recursos afectados: Política de contraseñas

Severidad: Baja **NºVulnerabilidad:** 9 **Reglamento**: CIS, PCI, RGPD
Nombre: Asegurar que las políticas de permisos están asociadas a grupos o roles y no a usuarios
Descripción: Asignar permisos a los usuarios a través de grupos en lugar de hacer la asignación directa permite mejorar la gestión de los permisos y mejorar la seguridad en el entorno.
Recursos afectados: testCLI

Severidad: Baja **NºVulnerabilidad:** 10 **Reglamento**: CIS, RGPD
Nombre: Asegurar que las instancias EC2 que deban hacer cambios en la infraestructura utilizan un rol IAM en lugar de almacenar credenciales de API
Descripción: En los casos en los que existan máquinas virtuales que deban realizar cambios en la infraestructura, es necesario que se realice a través de un rol IAM en lugar de almacenar credenciales de la API de AWS para realizar estas tareas.
Recursos afectados: WinServer1, WinServer2, WebServer

Severidad: Baja **NºVulnerabilidad:** 11 **Reglamento**: CIS, RGPD, PCI
Nombre: Habilitar AWS Config en la región
Descripción: AWS Config permite monitorizar de manera activa los cambios que se realicen en la configuración de los recursos, facilitando la tarea de vigilar las acciones de los usuarios.
Recursos afectados: eu-west-1, eu-west-2, eu-west-3

Severidad: Baja **NºVulnerabilidad:** 12 **Reglamento**: CIS, RGPD, PCI
Nombre: Habilitar los Flow Logs en la región
Descripción: Los Flow Logs generados por los Security Groups nos proporcionan información de las conexiones establecidas a través de cada Security Group, esenciales para identificar conexiones sospechosas ante un incidente de seguridad.
Recursos afectados: eu-west-1, eu-west-2, eu-west-3

Severidad: Baja **NºVulnerabilidad:** 13 **Reglamento**: RGPD
Nombre: Habilitar el cifrado de los buckets S3
Descripción: El cifrado de los objetos contenidos en un bucket S3 debe estar habilitado.
Recursos afectados: s3tfmseryoferben

6.1.2. Plan de acción

Centrándonos primeramente en las vulnerabilidades de severidad más alta, se nos insta a utilizar lo menos posible la cuenta root (vulnerabilidad número 1). Una buena política de administración de un entorno se basa en la separación de roles en el mismo, designando administradores para tareas más concretas (por ejemplo, un administrador de bases de datos, un administrador de servicios IaaS, un administrador de redes), de tal modo que se evita aglutinar todos estos roles en una única persona.
En caso de que un intruso haya conseguido robar las credenciales de uno de estos administradores sólo tendría acceso a una parte de la administración del entorno y no podría hacerse con el control completo del mismo.
Adicionalmente, en la vulnerabilidad número 2, se obliga el uso de la autenticación en varios factores (utilizando un dispositivo físico o virtual, mensaje de texto, llamada, etc…) para que, en caso de que se filtren las credenciales de las cuentas de administración un hipotético intruso no pueda acceder al entorno ni teniendo la contraseña de acceso.

Continuando con los problemas de severidad alta, observamos que hay un security group que permite el acceso al puerto 22 desde todo internet (vulnerabilidad número 3). Para solucionar este problema es necesario revisar qué servicio presta la máquina que tiene asociado el security group y, en caso de ser necesario, filtrar el acceso a este puerto a una serie concreta de direcciones IP o utilizar una VPN para acceder a este servicio.
Es necesario realizar este filtrado debido a que si se detecta una vulnerabilidad explotable podemos dejar expuesto a la red un acceso a nuestra infraestructura, y sin el correcto filtrado en los security groups del resto del entorno, incluso un atacante podría moverse lateralmente por la infraestructura, exfiltrando datos, provocando una denegación de servicio o hacerse con el control de la propia infraestructura.

Reduciendo un escalón de severidad, en la vulnerabilidad número 4 se hace mención a la ausencia de cifrado de los datos almacenados en las bases de datos RDS. No sólo es una buena práctica debido a que la base de datos es ilegible sin la clave de cifrado, sino que, además, según el RGPD es obligatorio cifrar todos los datos de carácter personal (o bancario según el estándar PCI-DSS), incurriendo en una infracción grave con sanciones económicas elevadas.
Por lo tanto, es necesario configurar en las propiedades de las bases de datos el cifrado de las mismas.

Otro problema detectado en la vulnerabilidad número 5 es el uso de security groups no restrictivos en las VPC. El hecho de utilizar un security group sin el debido filtrado provoca que las interfaces de red que se generen en la VPC no estén lo suficientemente protegidas del acceso exterior si el security group asociado a ellas tampoco lo es o si directamente se generan sin un security group.

En la vulnerabilidad número 6 se reporta dos Security Groups que permiten el tráfico entrante al puerto 3389 (RDP). Como se ha visto anteriormente, esto es un problema grave, ya que, al tratarse del puerto utilizado para gestionar máquinas Windows remotamente, muy atacado normalmente y con numerosas vulnerabilidades que permiten el acceso a estas máquinas, abriendo una puerta a la infraestructura de AWS.

Las vulnerabilidades 7 y 8 hacen referencia a políticas aplicables a las contraseñas de los usuarios. El acceso mediante contraseña es la primera barrera que tiene un atacante a la hora de acceder a la infraestructura. Obligar a los usuarios a establecer contraseñas complejas y evitar su reutilización ayuda a reforzar esta primera capa de seguridad.

La vulnerabilidad número 9 nos identifica un usuario (utilizado para usarlo con la consola de comandos de AWS) que tiene políticas de permisos directamente asignadas, en lugar de gestionar los permisos a través de un grupo. Asignar las políticas directamente a los usuarios dificulta el seguimiento de los permisos de cada uno de ellos, empeorando considerablemente la gestión de identidad en nuestro entorno, debilitando la seguridad del mismo al descuidar usuarios con demasiados permisos.

En la vulnerabilidad número 10 se nos aconseja utilizar un rol IAM para las instancias EC2 que necesiten realizar cambios de manera programática en la infraestructura. De este modo, la instancia puede identificarse con ese rol, evitando almacenar claves de API, ya sea en el código o en un fichero de texto en la unidad de disco.

CloudSploit reporta, en la vulnerabilidad número 11, la falta de configuración de la herramienta AWS Config. Esta herramienta permite realizar una monitorización avanzada de las actividades en el plano de gestión de la infraestructura. De este modo, se pueden monitorizar ciertas actividades que, en caso de haber una intrusión, podrían dar lugar a una posible denegación de servicio o filtración de información.

La vulnerabilidad número 12 nos indica que no tenemos habilitados los Flow Logs en la región. Los Flow Logs proporcionan información esencial de cara a monitorizar las comunicaciones y analizar el tráfico en caso de haber una infección o intrusión.

Por último, en la vulnerabilidad número 13, se nos reporta que el cifrado en reposo de los buckets S3 no está habilitado. Reincidiendo en lo descrito anteriormente, en caso de contener información relacionada con datos personales o bancarios, es obligatorio habilitar el cifrado, dada la sensibilidad de estos.

6.2. Azure

6.2.1. Vulnerabilidades detectadas

De manera similar a lo que ocurría en AWS, el dashboard de Cloudsploit no nos permite disponer de una visión clara o útil respecto al cumplimiento, debido a la ausencia de filtrado y la particular manera que tiene de contabilizar el número de recursos:

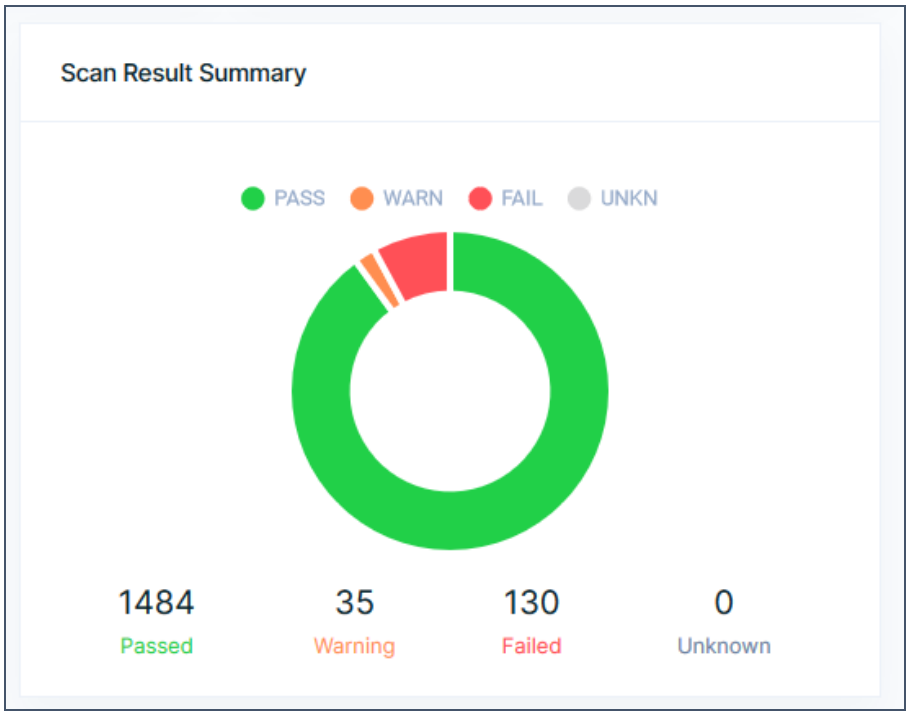


Ilustración 30: Dashboard de Cloudsploit con los datos de Azure

Al contrario que lo visto en AWS, CloudSploit nos proporciona información similar a las herramientas comerciales, reportando un total de 35 vulnerabilidades.

Muchas de estas vulnerabilidades están relacionadas con Azure Security Center, haciendo un total de 16. Estas vulnerabilidades no están directamente relacionadas con el entorno desplegado y no se profundizará en ellas.

Severidad: Alta **NºVulnerabilidad:** 1 **Reglamento**: CIS, RGPD, PCI **Nombre**: Asegurar que el tráfico de las storage account está cifrado en tránsito **Descripción**: Es necesario que el tráfico de datos provenientes de las storage accounts está cifrado en tránsito utilizando HTTPS, de tal manera que la información no circule en claro por la red. **Recursos afectados**: tfmsta

Severidad: Alta **NºVulnerabilidad:** 2 **Reglamento**: CIS **Nombre**: Asegurar que no se permite el tráfico entrante al puerto 3389 **Descripción**: Se detecta un NSG que permite el tráfico entrante al puerto 3389, utilizado por RDP, sin filtrar, es decir, expuesto y accesible a todo el mundo. **Recursos afectados**: WinServer-2019-1-nsg, WinServer-2019-2-nsg

Severidad: Alta **NºVulnerabilidad:** 3 **Reglamento**: CIS **Nombre**: Asegurar que no se permite el tráfico entrante al puerto 22 **Descripción**: Se detecta un NSG que permite el tráfico entrante al puerto 22, utilizado por SSH, sin filtrar, es decir, expuesto y accesible a todo el mundo. **Recursos afectados**: UbuntuServer1804-nsg

Severidad: Alta **NºVulnerabilidad:** 4 **Reglamento**: RGPD
Nombre: No se están almacenando los activity logs de la suscripción
Descripción: No se encuentra una storage account en la que se almacenen los activity logs (eventos de Azure). Es un requisito indispensable del RGPD tener un registro que confirme que los datos son utilizados exclusivamente por las personas autorizadas para tal fin.
Recursos afectados: Suscripción

Severidad: Media **NºVulnerabilidad:** 5 **Reglamento**: CIS, RGPD, PCI
Nombre: Existen discos duros sin cifrar
Descripción: Existen discos duros de máquinas virtuales cuyos datos no están cifrados. Es un requisito indispensable cifrar estos discos duros cuando contienen datos de carácter personal.
Recursos afectados: WinServer-2019_OsDisk_1_52b3f67c18f8409aae85c6ef9d5ae9e9, 2019_2_OsDisk_1_ae91bec98de18e22cde749a2d09b20b, UbuntuServer1804_OsDisk_1_6b95ff8628894fc9a47eeff668423fab

Severidad: Media **NºVulnerabilidad:** 6 **Reglamento**: CIS, PCI
Nombre: Asegurar que las storage accounts sólo permiten el tráfico desde redes confiables
Descripción: Restringir el acceso de red a las storage accounts ayuda a añadir una capa extra de seguridad a la hora de acceder a los datos almacenados. Limitar las redes que pueden acceder a estas previene de posibles fugas de información.
Recursos afectados: csb10032000a2811712, tfmsta

Severidad: Media **NºVulnerabilidad:** 7 **Reglamento**: PCI
Nombre: Asegurar que las storage accounts están cifradas utiliazando claves gestionadas por el cliente
Descripción: Utilizar claves de cifrado gestionadas por la organización en lugar del proveedor de cloud añade un nivel de seguridad que evita que las claves sean públicas ante una filtración de información del proveedor.
Recursos afectados: csb10032000a2811712, tfmsta

Severidad: Media **NºVulnerabilidad:** 8 **Reglamento**: CIS
Nombre: Asegurar que la acción por defecto en las reglas de firewall de las Storage Account es “deny”
Descripción: Es recomendable filtrar las direcciones IP que deban acceder a una STA y denegar el resto del tráfico, consiguiendo evitar accesos no autorizados.
Recursos afectados: tfmsta, csb10032000a2811712

Severidad: Media **NºVulnerabilidad:** 9 **Reglamento**: CIS, PCI
Nombre: Añadir una alerta de actividad cuando se creen, editen o eliminen reglas de firewall en los servidores SQL
Descripción: Realizar modificaciones en el firewall de los servidores SQL puede provocar pérdidas de servicio, y, por lo tanto, es necesario monitorizar cambios en estos recursos para identificar cambios no autorizados o accidentales.
Recursos afectados: -

Severidad: Media **NºVulnerabilidad:** 10 **Reglamento**: PCI
Nombre: Añadir una alerta de actividad cuando se creen, editen o eliminen redes virtuales (VNETs)
Descripción: Es recomendable disponer de una alerta de creación o modificación de redes virtuales en la infraestructura ya que puede afectar a los servicios desplegados en producción.
Recursos afectados: -

Vulnerabilidades de la 11 a la 27 relacionadas con Azure Security Center localizadas en el anexo.

Severidad: Baja **NºVulnerabilidad:** 28 **Reglamento**: CIS, PCI
Nombre: Asegurar que las máquinas virtuales tienen un agente de antivirus instalado
Descripción: Disponer de un agente anti-malware instalado en las máquinas virtuales previene potenciales infecciones en los equipos.
Recursos afectados: WebServer

Severidad: Baja **NºVulnerabilidad:** 29 **Reglamento**: CIS, RGPD
Nombre: Habilitar el logado de auditoría en los servidores SQL
Descripción: Activar el logado de auditoría en las bases de datos permite disponer de información de accesos y modificación de datos, esencial ante un incidente de seguridad.
Recursos afectados: tfmsql

Severidad: Baja **NºVulnerabilidad:** 30 **Reglamento**: CIS
Nombre: Aumentar la retención de los logs de auditoría a un mínimo de 90 días
Descripción: Disponer de una retención mínima de 90 días en los logs de auditoría permite disponer de información de incidentes pasados que pudieran haber pasado desapercibidos.
Recursos afectados: tfmsql

Severidad: Baja **NºVulnerabilidad:** 31 **Reglamento**: CIS
Nombre: Asegurar que el cifrado está habilitado en bases de datos SQL
Descripción: Tener los datos de las bases de datos cifrados en reposo es necesario para poder proporcionar una capa de seguridad básica sobre los datos almacenados.
Recursos afectados: tfmsql

Severidad: Baja **NºVulnerabilidad:** 32 **Reglamento**: CIS
Nombre: Asegurar que existe se registran todas las actividades de gestión en los activity logs
Descripción: Disponer de todas las actividades que se realicen en la suscripción permite identificar el origen de acciones maliciosas.
Recursos afectados: Suscripción

Severidad: Baja **NºVulnerabilidad:** 33 **Reglamento**: CIS, RGPD
Nombre: Crear una alerta cuando se detecte creaciones o modificaciones de NSGs en los activity logs
Descripción: La modificación o creación de Network Security Groups en la suscripción puede afectar a la disponibilidad de los servicios, siendo necesaria la monitorización de su gestión.
Recursos afectados: Suscripción

Severidad: Baja **NºVulnerabilidad:** 34 **Reglamento**: CIS
Nombre: Asegurar que las bases de datos SQL están cifradas con claves gestionadas por el cliente (CMKs)
Descripción: El uso de claves de cifrado propias en lugar de utilizar las del proveedor de cloud mejora el control de las mismas, añadiendo una capa adicional de seguridad.
Recursos afectados: tfmsql

Severidad: Baja **NºVulnerabilidad:** 35 **Reglamento**: RGPD
Nombre: Asegurar que las Storage Account tienen habilitado el cifrado en reposo
Descripción: Es necesario habilitar el cifrado en reposo de las STAs, ya que pueden contener información sensible como nombres de usuario.
Recursos afectados: tfmsta

6.2.2. Plan de acción

Al igual que en las otras herramientas (Prisma y Dome9), se nos informa de que:

- Las STAs no obligan a utilizar cifrado en tránsito (vulnerabilidad número 1)
- Tenemos NSGs permitiendo el tráfico al puerto 22 y 3389 sin filtrar (vulnerabilidades 2 y 3)
- No se están registrando los Activity Logs de la suscripción (vulnerabilidad número 4)

En la vulnerabilidad número 5 se nos informa también de la existencia de discos duros sin cifrar. Es de obligado cumplimiento que todos los datos almacenados se encuentren cifrados en reposo, además de aumentar la dificultad de acceso a los mismos en caso de intrusiones o accesos no autorizados. Además, en caso de que estos contengan datos personales, el Reglamento General de Protección de datos obliga a que se encuentren cifrados.

Las vulnerabilidades 6 y 7 sobre Storage Accounts nos informan de que no se está segmentando por IP ni se están utilizando claves gestionadas por la organización, respectivamente. En la primera de ellas se considera un riesgo puesto que, en caso de tener datos sensibles (snapshots de discos duros, backups de bases de datos, etc…) podría suponer un problema grave en caso de auditoría o filtración de datos.
En cuanto a la gestión de claves, como se ha comentado anteriormente, gestionar las claves de cifrado en la organización ayuda a controlar los accesos a los datos, ya que se puede segregar a los usuarios que deban tener acceso a dichas claves.

Relacionada con la 6, la vulnerabilidad número 8 reporta que existen STAs que se debe establecer la acción de denegar el tráfico entrante a todo el resto de direcciones IP no permitidas explícitamente, dificultando un acceso no autorizado a los datos.

Las vulnerabilidades 9 y 10 hacen referencia a crear alertas de auditoría en base a creación y modificación de recursos. En este caso, tratan sobre creación o modificación de reglas de firewall en los servidores SQL y creación o modificación de redes virtuales. Son dos buenos ejemplos de recursos que, si se configuran incorrectamente, pueden generar una caída del servicio y, por lo tanto, monitorizar este tipo de cambios es una buena práctica para supervisar que estas acciones no provocan efectos adversos.

A continuación, se detalla un bloque de problemas relacionados con Azure Security Center (vulnerabilidades de la 11 a la 27). Estas vulnerabilidades vienen dadas en base a reglas por defecto de este producto, y CloudSploit reporta un problema por cada regla no activa.
Estas reglas son extremadamente útiles porque permiten, entre otros, monitorizar que los discos duros de las máquinas estén cifrados o habilitar de manera automática un antivirus en las máquinas virtuales.
Activar Azure Security Center junto con todo su catálogo de reglas permite, de manera activa, securizar mucho más el entorno de Azure.

En la vulnerabilidad número 28 se reporta una máquina virtual sin un agente anti-malware instalado. Como su propio nombre indica, disponer de software de este tipo permite parar infecciones y archivos maliciosos en la máquina.

Las vulnerabilidades 29 y 30 están relacionadas con el logado de diferente información. En la primera de ellas, dicha información es la auditoría de los servidores SQL. No sólo es obligatorio por el RGPD, si no que, además, permitirá determinar el alcance de una posible filtración.
Por otro lado, se reporta la poca retención de los Flow Logs. Los Flow Logs permiten ayudar en una investigación forense, y, por lo tanto, se recomienda almacenarlos el máximo tiempo posible.

A continuación, la vulnerabilidad número 31 nos reporta la ausencia de cifrado de las bases de datos SQL. Aunque sólo se haya reportado por el CIS, al ser una base de datos de usuarios y contener información personal, el RGPD obliga a cifrar los datos en reposo.

La vulnerabilidad número 32 nos indica que no se están logando todas las actividades en el panel de gestión web de Azure. Como se ha indicado anteriormente en el trabajo, la necesidad de tener un registro de toda la actividad desarrollada en el entorno para facilitar la investigación en el caso de que se de un incidente de seguridad.

La vulnerabilidad número 33 está directamente relacionada con las 29 y 30, en este caso, sobre los NSG, y el impacto ante un fallo de configuración es el mismo; la posible pérdida de servicio.

A continuación, la vulnerabilidad número 34 nos reporta que no se están gestionando las claves de cifrado de las bases de datos, si no que se delega en la gestión propia de Azure. Como se ha indicado otras veces, gestionar de manera manual las claves de cifrado aumenta la seguridad de los datos, evitando que aquellas personas que no deban tener acceso a las claves tampoco puedan acceder a las bases de datos.

Por último, la vulnerabilidad 35 expone el problema de no tener activado el cifrado en reposo de los datos de las Storage Account. En caso de almacenar datos de carácter personal o de tarjetas de crédito es estrictamente necesario activar esta opción, lo cual evitará, no sólo posibles fugas de información, sino además multas por incumplimiento de los reglamentos PCI o RGPD.

7. Comparativa de resultados

A modo de resumen, vemos que la distribución de vulnerabilidades queda del siguiente modo:

	Prisma		Dome9		Cloudsploit	
	AWS	Azure	AWS	Azure	AWS	Azure
Altas	5	3	19	8	3	4
Medias	15	16	23	19	4	6
Bajas	3	2	0	0	6	8
Total	23	21	42	27	13	18

Ilustración 31: Estadística de vulnerabilidades por criticidad y herramienta

En cuanto a número de vulnerabilidades, ha sido Dome9 la herramienta que más problemas ha detectado, con un total de 42 y 27, en AWS y Azure respectivamente, y CloudSploit, la que menos.

Sin embargo, la criticidad, aunque se puede medir bajo un punto de vista objetivo, iba cambiando entre herramientas para la misma vulnerabilidad. Incluso cambiaba entre reglamentos en la misma plataforma.
Por esto, la severidad debe ser meramente informativa, pero no debe considerarse como un parámetro definitivo, ya que, en ocasiones, incluso depende de la necesidad de la organización cumplir con unos requerimientos u otros.
Asimismo, en el caso de cara al Reglamento General de Protección de datos podría ser más grave no tener las bases de datos cifradas en reposo que, por ejemplo, tener una instancia con el puerto 22 expuesto a internet, aunque a nivel de seguridad suponga un riesgo mayor. Este tipo de discrepancias también evidencia que algunos reglamentos dan más prioridad a ciertos aspectos que en otro caso podrían ser considerados más importantes, debido al origen de cada uno de estos estándares.

	AWS				Azure		
	Prisma	Dome9	CS		Prisma	Dome9	CS
EC2	1	1	1	VM	2	0	2
S3	2	5	1	STA	3	2	5
RDS	1	2	1	SQL	5	4	5
SG	5	5	4	NSG	2	3	2
Trail	4	17	1	Logs	1	3	3
Subnet	1	1	0	Subnet	0	0	1
IAM	9	10	5	ASC	6	14	0
EKS	0	0	0	AKS	1	1	0
ELB	0	1	0	Locks	1	0	0

Ilustración 32: Estadística de vulnerabilidades detectadas por recurso y plataforma

Revisando las estadísticas por tipo de recurso en cada caso, podemos observar que muchas de las vulnerabilidades reportadas por Dome9 se refieren a la configuración de CloudWatch, en lugar de indicarlo una única vez como hace Prisma. De esta manera, parece que la información reportada por Dome9 aporta menos valor y requiere cierto filtrado.
En la misma casuística nos encontramos cuando nos informa acerca de políticas de Azure Security Center, reportando 14 vulnerabilidades relacionadas con este recurso, mientras Prisma se queda en 6.

Aunque en ambos entornos, las vulnerabilidades reportadas han sido similares (principalmente en cuanto a bases de datos y segmentación de Security Groups), hay una diferencia destacable, y es que en Azure no se reportan problemas en cuanto a las cuentas de usuario.
Esto se debe a que Azure gestiona las identidades a través de Azure Active Directory, que es la versión en la nube del conocido directorio activo utilizado a lo largo de los años. Debido a esto, la organización de las suscripciones también es diferente. Mientras que en AWS las cuentas son independientes entre sí (aunque se pueden asociar a la misma organización), en Azure existe un Azure Active Directory por organización (tenant), en el cual pueden localizarse numerosas suscripciones, compartiendo identidades entre ellas.

Azure Active Directory ofrece, además, gestión de permisos a través de roles (Global Admin, Application Developer, etc…), facilitando la asignación de permisos a usuarios. Sin embargo, no se comparten estos permisos entre suscripciones. Existen los denominados roles RBAC de Azure, que son exclusivos de Azure y no tienen relación con los roles de Azure Active Directory. Entre otros, podemos destacar el rol "Owner", que sería equivalente a un administrador de la suscripción.
De este modo, se consigue compartir identidades entre todas las suscripciones, pero se consigue también asignar los permisos a los usuarios de manera individual en cada una de ellas.

Debido a esta propiedad de Azure, también es normal ver menos vulnerabilidades en ese entorno (por ejemplo, los problemas reportados respecto a las políticas de contraseñas en AWS no tienen lugar en Azure ya que se administran a nivel de Azure Active Directory).

Otro aspecto llamativo es la baja (o nula en el caso del EKS de AWS) cantidad de problemas reportados en los clústers de Kubernetes. No han aparecido muchas vulnerabilidades específicas debido a que son máquinas virtuales (nodos) y que también disponen de Security Groups (como se ha visto en alguna vulnerabilidad, aparecía el NSG de estos clústers).

Finalmente, observamos las estadísticas por reglamento.

	AWS			Azure		
	Prisma	Dome9	CS	Prisma	Dome9	CS
CIS	13	30	11	20	24	14
PCI	5	11	9	10	5	7
NIST	12	39	0	10	7	0
RGPD	22	35	9	8	3	6

Ilustración 33: Vulnerabilidades reportadas por reglamento y herramienta
NOTA: CloudSploit no disponía de una implementación del reglamento NIST.

Se puede ver claramente que, en AWS, RGPD es el reglamento que más problemas ha sido capaz de detectar, aunque se ve superado por el NIST en Dome9.
Por otro lado, en AWS es el CIS el que más vulnerabilidades obtiene, mientras que RGPD queda muy por detrás.

No hay una motivación aparente de por qué se han repartido de este modo las vulnerabilidades. Sin embargo, es una buena noticia que en AWS sea un reglamento europeo el que detecta más vulnerabilidades, puesto que se trata de un marco legislativo a nivel europeo, en lugar de un conjunto de buenas prácticas o recomendaciones de otros organismos (en el caso del CIS o NIST son de origen estadounidense, ni siquiera contemplan las regulaciones europeas).

Vemos que CloudSploit no integra ninguna implementación de las reglas definidas por el NIST, quedando atrás en ese aspecto respecto a las herramientas comerciales.

8. Conclusiones

Aunque el despliegue del entorno ha sido realizado intencionadamente con configuraciones peligrosas, en entornos reales existe la posibilidad de que estas configuraciones se den debido a diferentes motivos:

- Plantillas desactualizadas
- Desconocimiento por parte de los administradores de la nube
- Despliegues realizados con prisa
- Recursos olvidados

Las herramientas utilizadas nos han proporcionado una visibilidad de todos los recursos del entorno, habilitando de manera efectiva a un profesional de seguridad para reducir los riesgos de la organización y protegiendo la infraestructura contra incidentes provocados por descuidos como los vistos en la introducción de este trabajo.

De este modo, hemos podido observar que muchas de las vulnerabilidades reportadas por las herramientas resultan un peligro real, como la sobreexposición de determinados recursos tales como bases de datos o máquinas virtuales.

Las 3 herramientas, Prisma, Dome9 y CloudSploit, han reportado vulnerabilidades prácticamente idénticas entre ellas (salvando las distancias), donde podemos ver 5 principales focos de atención:

- Segmentación a nivel de red, tanto de máquinas virtuales como de servicios PaaS (Storage Accounts, buckets S3, bases de datos, etc…). Se ha visto que la configuración de red es uno de los pilares de la seguridad en la nube, del mismo modo que lo ha sido tradicionalmente en los entornos on-premise. En este caso existen ciertas peculiaridades, como la existencia de los Security Groups (similares en concepto a los firewalls propios de las máquinas), o la restricción de tráfico a través de la configuración de los recursos, en lugar de utilizar un software específico para ello.
- Cifrado de los datos. Al hablar de los datos almacenados vamos más allá de las bases de datos, existen servicios PaaS, como las Storage Account o los buckets S3, que pueden almacenar información sensible, como podría ser backups de bases de datos o los eventos (logs) de lo que ocurre dentro del entorno, además de poder monitorizar el cifrado de los discos duros de las máquinas, lo cual supone una ventaja importante sobre los entornos on-premise clásicos. En este apartado podemos incluir, además, el cifrado de las comunicaciones, como se ha visto en las Storage Account y buckets S3, esencial para evitar que la información sea transmitida en claro a través de la red.
- Logado de acciones de los usuarios. Esta funcionalidad es una gran ventaja sobre entornos on-premise, donde no es fácil registrar la información sobre quién modifica recursos de la infraestructura. Al ser una funcionalidad integrada en los CSP, es recomendable aprovecharla al máximo para tener un buen detalle de auditoría, sobre todo si no hace falta utilizar herramientas de terceros para este fin.
- Gestión de identidad. Aunque en el caso de Azure existen ciertas peculiaridades, las cuales se explican más adelante, hemos visto que en el entorno de AWS se pueden monitorizar roles sobrepermisivos, mala gestión de permisos e incluso, el nivel de seguridad de las contraseñas. La identidad es un componente altamente importante en la nube, mucho más que en entornos on-premise, debido a que absolutamente todo el entorno está supeditado a esta. De este modo, si la gestión de la identidad no se realiza correctamente, estaremos poniendo en peligro la infraestructura al completo.
- Herramientas de seguridad propias de cada CSP. Como se ha visto, las herramientas de seguridad que ofrecen de manera nativa es un buen complemento a las soluciones CSPM, ya que pueden generar alertas en base al comportamiento de los usuarios o métricas, entre otros. Además, en el caso del Security Center de Azure, es incluso posible evitar el despliegue o la aplicación de configuraciones que impliquen exponer elementos del entorno, como, por ejemplo, evitar abrir puertos de gestión a todo internet.

Entrando más en detalle en la parte de identidad, hemos podido observar que, en AWS, tenemos información sobre usuarios, roles, etc…, mientras que en Azure no se ha visto nada relacionado. Esto se debe a que la identidad en Azure se gestiona (y se securiza) a través del Azure Active Directory, con herramientas de la suite Microsoft 365, al contrario que en AWS, que las identidades sólo existen en la cuenta, no fuera de ellas.
Sí que existen, sin embargo, roles dentro de las suscripciones de Azure, y son visibles desde las herramientas, aunque las asignaciones de estos a usuarios no se pueden listar y, como consecuencia, no hay políticas dedicadas a monitorizar asignaciones de roles en Azure.

Realizando el hardening indicado por las diferentes soluciones no sólo hemos conseguido securizar la infraestructura, sino que, además, se ha mejorado el nivel de cumplimiento en aspectos tan importantes como el Reglamento General de Protección de Datos, de cumplimiento obligado en las empresas europeas y cuyo incumplimiento podría acarrear en cuantiosas sanciones económicas.

Tras este proceso, hemos podido lograr la identificación y solución de vulnerabilidades en los entornos cloud, objetivo principal del trabajo.

En cuanto a la planificación seguida, ha sido realista. Sin embargo, no ha estado exenta de problemas:

- A mitad del despliegue de la infraestructura en la cuenta de AWS se detectaron limitaciones en la modalidad para estudiantes de AWS no resolubles (pese a utilizar el soporte de Amazon) a la hora de desplegar ciertos recursos o utilizar ciertas herramientas. Una de las limitaciones más graves fue el no poder utilizar ni la consola de AWS ni la ejecución de stacks, teniendo que recurrir a una cuenta personal con un coste asociado pagado por el estudiante.
- Según se realizaba el despliegue de la infraestructura en Azure empezó la pandemia del Covid-19 y el consiguiente confinamiento y aumento del teletrabajo. Esto provocó un aumento de la demanda de los servicios de Azure demasiado elevado, a lo cual, Microsoft empezó a aplicar restricciones al despliegue de recursos, quedándonos a la mitad del proceso y retrasando la evaluación del entorno dos semanas. Debido a este problema hubo que prescindir de las herramientas Nessus y Qualys, incluidas en la parte más final de la planificación del trabajo.

En cuanto a la metodología, sólo ha sido necesario introducir cambios leves, como la valoración del estándar de seguridad NIST, que no se tuvo en cuenta en un principio, con la finalidad de complementar lo obtenido en una primera aproximación de las vulnerabilidades encontradas.

En cuanto a líneas futuras para continuar el trabajo, existen varias posibilidades:

- Aprovechar las herramientas que permiten la creación de reglas personalizadas. El trabajo se ha basado en el uso de reglas y políticas ya creadas, pero se nos da la posibilidad de crear nuestros propios controles, que pueden ser más profundos que los iniciales de la plataforma, o más personalizados, como realizar la monitorización del etiquetado de todos los recursos o vigilar que el tráfico saliente estuviera dirigido a través de un firewall.
- Documentar y detallar los pasos seguidos para solucionar cada vulnerabilidad. Esta información es interesante, ya que no existe una única manera de solucionar muchos de los problemas vistos y también valorar el impacto, si existe, de cada remediación. Sin embargo, no se consideró en este trabajo debido a la extensión del mismo.
- Profundizar en las posibilidades de CWPP o CASB. Como se comentó al inicio del trabajo, existen varios tipos de soluciones en lo referente a la seguridad en entornos Cloud, siendo CSPM el elegido para este trabajo. No por ello son menos interesantes las opciones que plantean los CASB (control de la información que circula entre aplicaciones) o CWPP (protección de las cargas a más bajo nivel, monitorizando procesos, redes y ficheros entre otros).

9. Glosario

- **CSPM**: Cloud Security Posture Management. Controles de gobierno a cumplir por los elementos ubicados en la infraestructura en la nube.
- **RGPD**: Reglamento General de Protección de Datos.
- **IaaS**: Infraestructure as a Service.
- **PaaS**: Platform as a Service.
- **SaaS**: Software as a Service.
- **MFA**: Multi-Factor Authentication, autenticación en dos o más factores.
- **[Network] Security Group (NSG)**: Elementos asociados a las interfaces de red o subnets, tanto en AWS como en Azure, que están formados por una serie de reglas que indican los puertos a nivel de red de entrada, salida y las direcciones IP de origen o destino. En Azure se denominan Network Security Group, mientras que en AWS simplemente son denominados Security Group.
- **SSH**: Secure SHell, protocolo de comunicación para la gestión remota de sistemas Linux.
- **RDP**: Remote Desktop Protocol, Escritorio Remoto de Windows, utilizado para el uso remoto de sistemas Windows.
- **RDS**: Relational Database Service, servicio de bases de datos relacionales de AWS.
- **Onboarding**: Proceso de comunicar una cuenta o suscripción de nubes públicas con herramientas de terceros, utilizando APIs públicas u otros métodos.
- **CSP**: Cloud Service Provider, proveedor de nube pública.
- **ARN**: Amazon Resource Name, identificador único de recursos en AWS.
- **Terraform**: Herramienta externa a los CSP creada por HashiCorp, altamente extendida y compatible con AWS y Azure, entre otros, que permite el despliegue automatizado de recursos en base a plantillas.
- **Stack**: Conjunto de tareas en AWS relacionadas con la creación de recursos en base a plantillas CloudFormation.
- **CloudFormation**: Servicio de AWS para el despliegue de recursos en base a plantillas.
- **VPC**: Virtual Private Cloud, red privada virtual donde se añaden subredes en AWS. El concepto es análogo a la VNET de Azure
- **VNET**: Virtual Network, red virtual de Azure donde se despliegan subredes e interfaces de red.
- **CloudTrail**: CloudTrail es un servicio de AWS que recoge los logs de actividad y auditoría de una cuenta, siempre relacionados con la creación y eliminación de recursos, entre otros.
- **CMK**: Customer Managed Keys. Claves gestionadas por el cliente en lugar de las ofrecidas por el CSP.
- **CloudWatch**: Servicio de AWS que utiliza los logs de CloudTrail para monitorizar acciones realizadas y utilización de recursos.
- **IAM**: Identity and Access Management, gestión de identidades y acceso de los usuario (ya sea mediante roles, grupos, etc…).
- **API**: Application Programming Interface, interfaz de datos para comunicación entre dos procesos.
- **ACL**: Access Control List, lista de control de acceso con datos relativos a los permisos de acceso a un recurso concreto.
- **NACL**: Network Access Control List, ACL específica para datos de red, usada en diferentes recursos en AWS.
- **AWS Config**: Servicio de AWS que permite registrar la configuración y sus cambios a modo de auditoría.
- **Flow log**: Información de flujos de comunicaciones entrantes y salientes de los NSG, indicando direcciones IP de origen y destino, puertos de origen y destino y cantidad de bytes transmitidos. Son comunes en todos los CSPs.
- **Cifrado en reposo**: Cifrado de los datos almacenados, ya sea en base de datos o en otros sistemas de almacenamiento.

- **Cifrado en tránsito**: Cifrado de todos los datos que se envían o reciben a través de la red.
- **CWP**: Cloud Workload Protection, protección en tiempo de ejecución de las cargas existentes en la nube.
- **CASB**: Cloud Access Security Broker, herramienta utilizada en entornos cloud para monitorizar la entrada y salida de datos a otras aplicaciones SaaS.
- **Snapshots**: Instantáneas de máquinas virtuales o bases de datos.
- **ASC**: Azure Security Center, herramienta de Microsoft con diferentes funcionalidades de seguridad sobre Azure.
- **JIT**: Just In Time. Aplicable a diferentes ámbitos, consiste en activar o utilizar cierta característica exclusivamente cuando se vaya a utilizar.
- **Hardening**: Técnica de reforzar la seguridad de una máquina, ya sea con medios pasivos (gestión de puertos de entrada, por ejemplo) o activos (control de procesos ejecutables).

10. Bibliografía

[1] Securing your cloud with CSPM (10/04/2020)
https://medium.com/@davidmoremad/securing-your-cloud-with-cspm-303d6c6b6a78

[2] Capital One breach exposes not just data, but dangers of cloud misconfigurations (06/04/2020)
https://www.scmagazine.com/home/security-news/capital-one-breach-exposes-not-just-data-but-dangers-of-cloud-misconfigurations/

[3] Hundreds of millions of Facebook user records were exposed on Amazon Cloud Server (06/04/2020)
https://www.cbsnews.com/news/millions-facebook-user-records-exposed-amazon-cloud-server/

[4] Modelo de responsabilidad compartida (AWS) (12/04/2020)
https://aws.amazon.com/es/compliance/shared-responsibility-model/

[5] Shared responsibility in the cloud (12/04/2020)
https://docs.microsoft.com/en-us/azure/security/fundamentals/shared-responsibility

[6] Cloud Security Posture Management Best Practices (09/05/2020)
https://www.globaldots.com/blog/cloud-security-posture-management-best-practices

[7] Cloud Security Posture Management (Fugue) (05/05/2020)
https://www.fugue.co/cloud-security-posture-management#cspmuses

[8] Center for Internet Security – About us (10/03/2020)
https://www.cisecurity.org/about-us/

[9] PCI Hispano - ¿Qué es PCI DSS? (10/03/2020)
https://www.pcihispano.com/que-es-pci-dss

[10] NIST – About us (10/03/2020)
https://www.nist.gov/about-nist

[11] REGLAMENTO (UE) 2016/679 DEL PARLAMENTO EUROPEO Y DEL CONSEJO de 27 de abril de 2016 (10/03/2020)
https://www.boe.es/doue/2016/119/L00001-00088.pdf

[12] DivvyCloud Protects cloud and container environments (15/03/2020)
https://divvycloud.com/

[13] Cloudisory Cloud Security Posture Management and Workload Protection (15/03/2020)
https://www.cloudvisory.com/cloudvisory-cspm-cwpp-solution.html

[14] Optiv. Cloud Security Posture Management (15/03/2020)
https://www.optiv.com/cybersecurity-dictionary/cloud-security-posture-management

[15] Palo Alto Networks Completes Acquisition of RedLock (18/03/2020)
https://www.paloaltonetworks.com/company/press/2018/palo-alto-networks-completes-acquisition-of-redlock

[16] Check Point beats analysts, buys cloud security company (18/03/2020)
https://en.globes.co.il/en/article-check-point-beats-analysts-buys-cloud-security-co-1001257786

[17] Introducing FortiCWP for Comprehensive Cloud Workload Protection (25/03/2020)
https://www.fortinet.com/blog/business-and-technology/forticwp-cloud-workload-protection.html

[18] List of open source tools for AWS security (10/03/2020)
https://github.com/toniblyx/my-arsenal-of-aws-security-tools

[19] Configure a client application to access web APIs (11/03/2020)
https://docs.microsoft.com/en-us/azure/active-directory/develop/quickstart-configure-app-access-web-apis

[20] Application and service principal objects in Azure Active Directory (12/03/2020)
https://docs.microsoft.com/en-us/azure/active-directory/develop/app-objects-and-service-principals

[21] CloudSploit Joins the Aqua Family (11/03/2020)
https://blog.cloudsploit.com/cloudsploit-joins-the-aqua-family-1825d48a8b72

[22] GitHub: CloudSploit: Cloud security configuration monitoring (20/04/2020)
https://github.com/cloudsploit

[23] About CloudSploit (20/04/2020)
https://cloudsploit.com/about

11. Anexos

Onboarding de cuentas de AWS en Prisma Cloud

Para dar de alta una nueva cuenta de AWS en Prisma la propia herramienta ofrece un asistente que va guiando el proceso.

En primer lugar, nos consulta si se desea que sea Prisma quien, de manera automatizada, resuelva vulnerabilidades detectadas en la configuración de los recursos afectados:

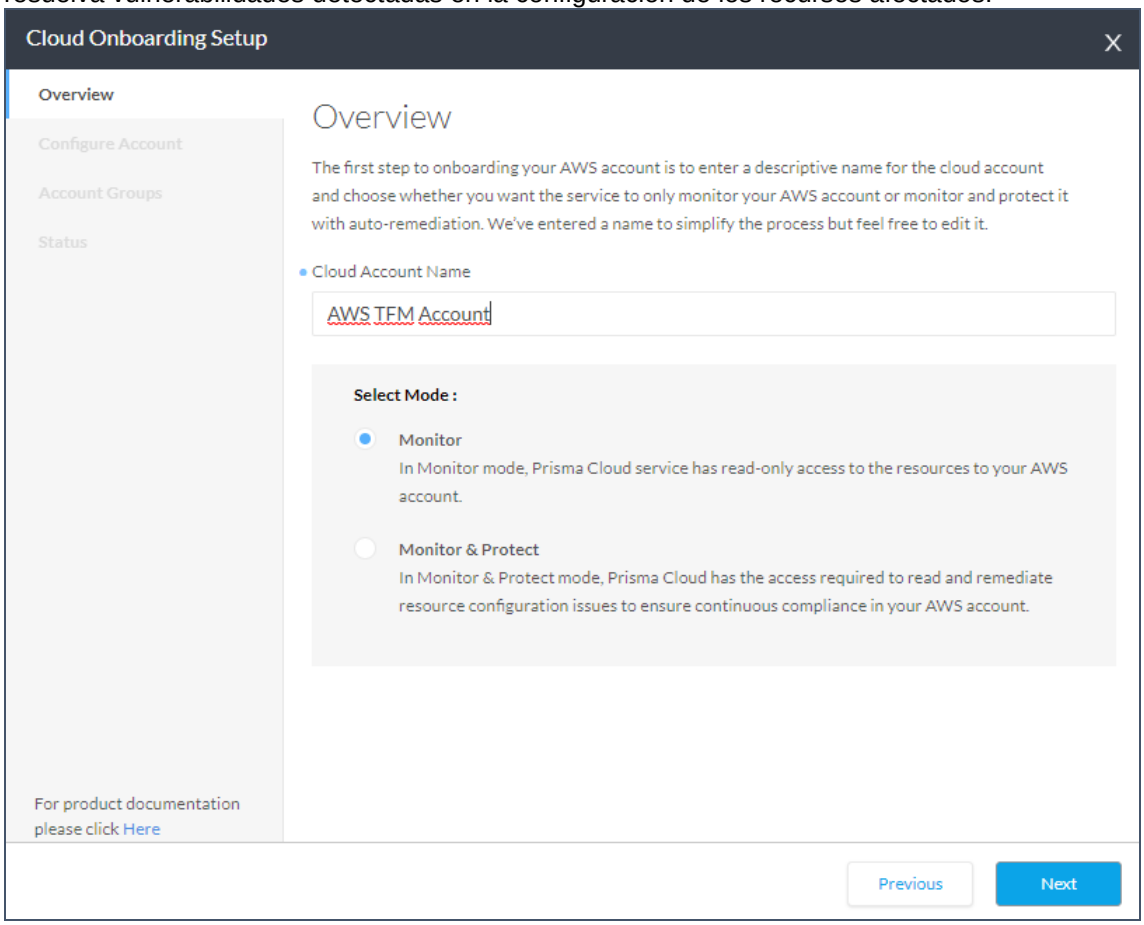


Ilustración 34: Selección de modo de funcionamiento de Prisma con AWS

Para el objetivo del trabajo seleccionaremos el modo monitor, que sólo lee la configuración de los recursos de la cuenta.

A continuación, nos solicita crear un Stack que creará los roles necesarios para el funcionamiento de la herramienta.

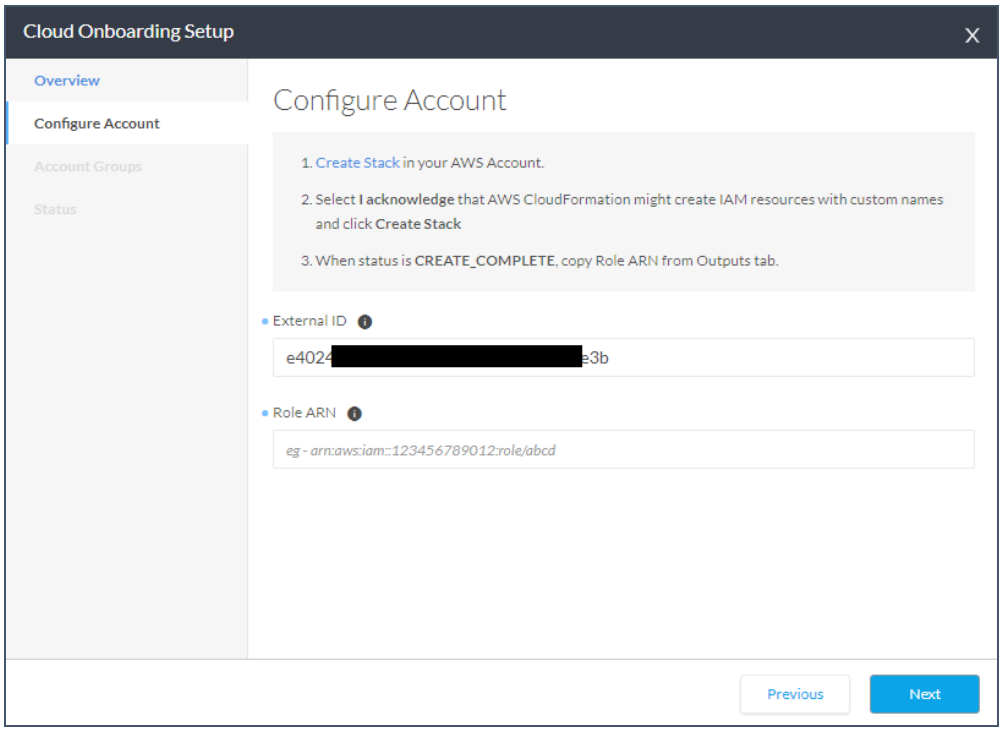


Ilustración 35: Enlace al stack desde Prisma y formulario para incluir ARN del rol generado

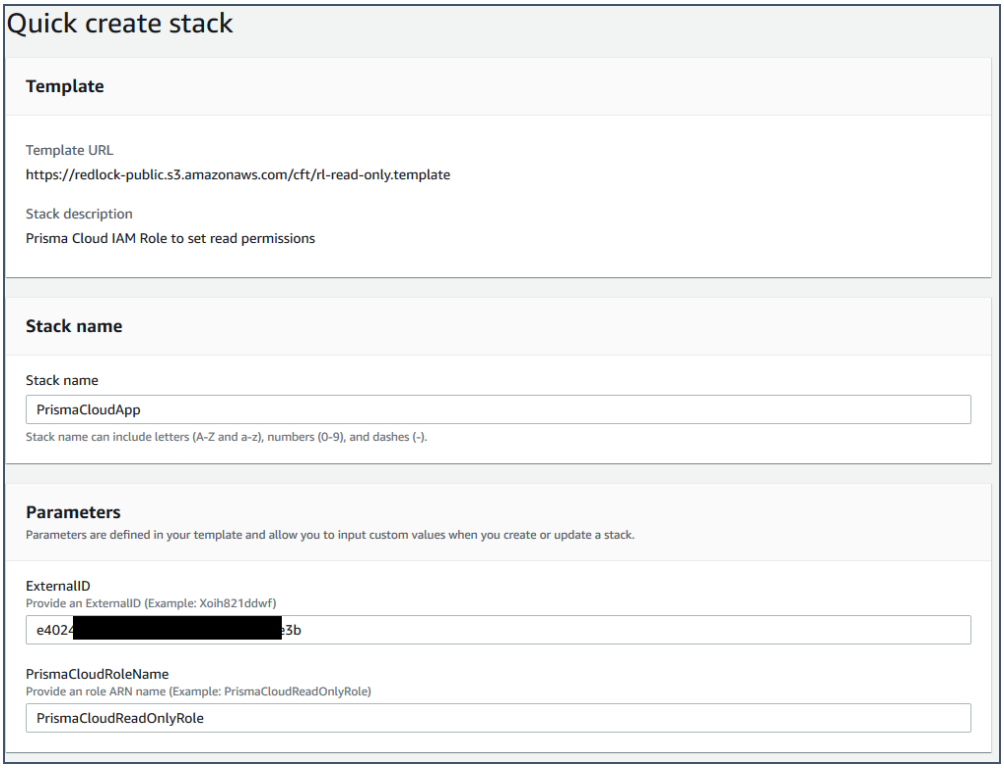


Ilustración 36: Pantalla de creación del stack de Prisma en AWS

Logical ID	Status	Status reason
PrismaCloudApp	CREATE_COMPLETE	-
PrismaCloudRole	CREATE_COMPLETE	-
PrismaCloudRole	CREATE_IN_PROGRESS	Resource creation Initiated
PrismaCloudRole	CREATE_IN_PROGRESS	-
PrismaCloudApp	CREATE_IN_PROGRESS	User Initiated

Ilustración 37: Resultado de la creación del stack en AWS

Una vez que se ha creado el rol con el stack, podemos obtener su identificador ARN de la pestaña Outputs:

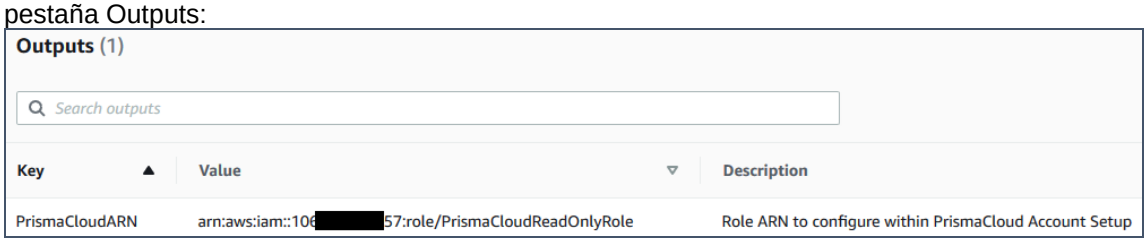


Ilustración 38: ARN obtenido de la creación del rol

Examinando la información del rol, podemos ver que puede ser asumido por una entidad de confianza, que no es más que el número de cuenta de AWS desde donde Prisma se conecta a nuestra cuenta de AWS:

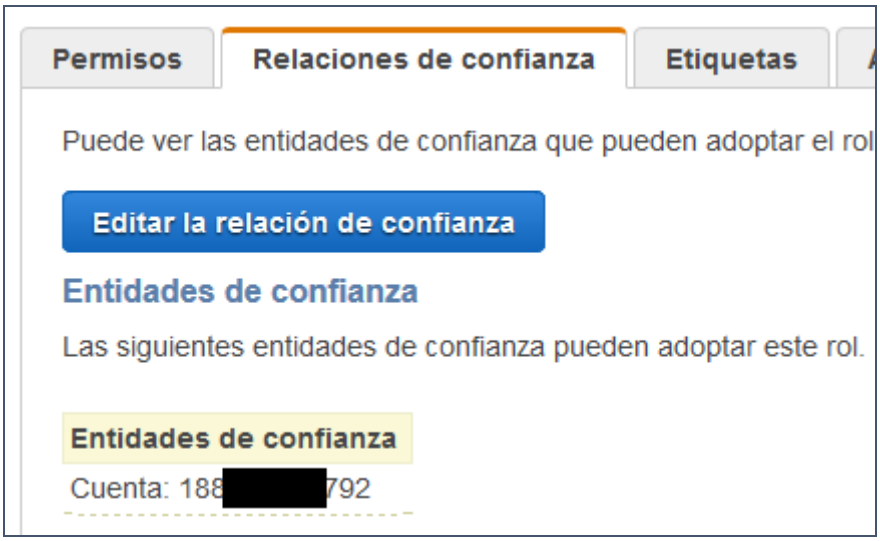


Ilustración 39: Entidad de confianza asociada al rol (cuenta de AWS de Prisma)

Podemos ver, a su vez, los recursos a los que este rol da acceso a la cuenta de Prisma:

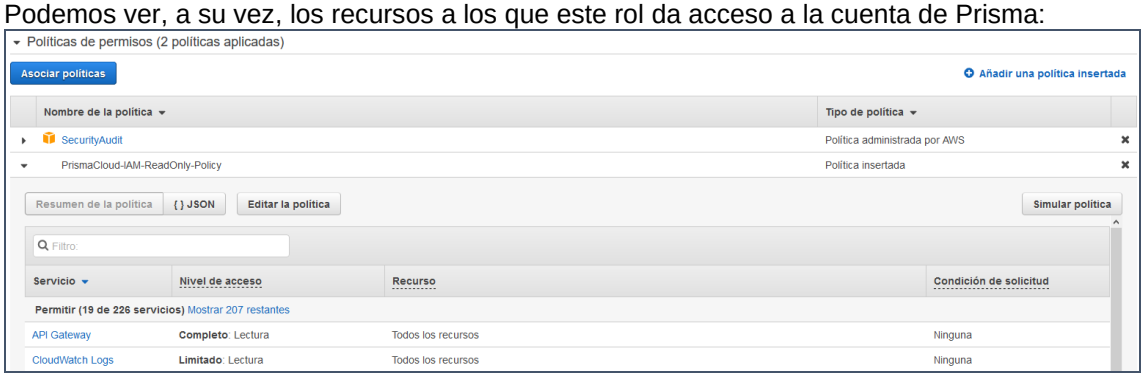


Ilustración 40: Visión superficial de los permisos del rol

Podemos observar que da permisos de lectura (completo o limitado) a un total de 226 servicios diferentes.

Finalmente, copiamos el ARN del rol generado y lo pegamos en el formulario de Prisma donde nos pedía dicho identificador y, tras una comprobación, nos dirá que la cuenta ha sido conectada con éxito:

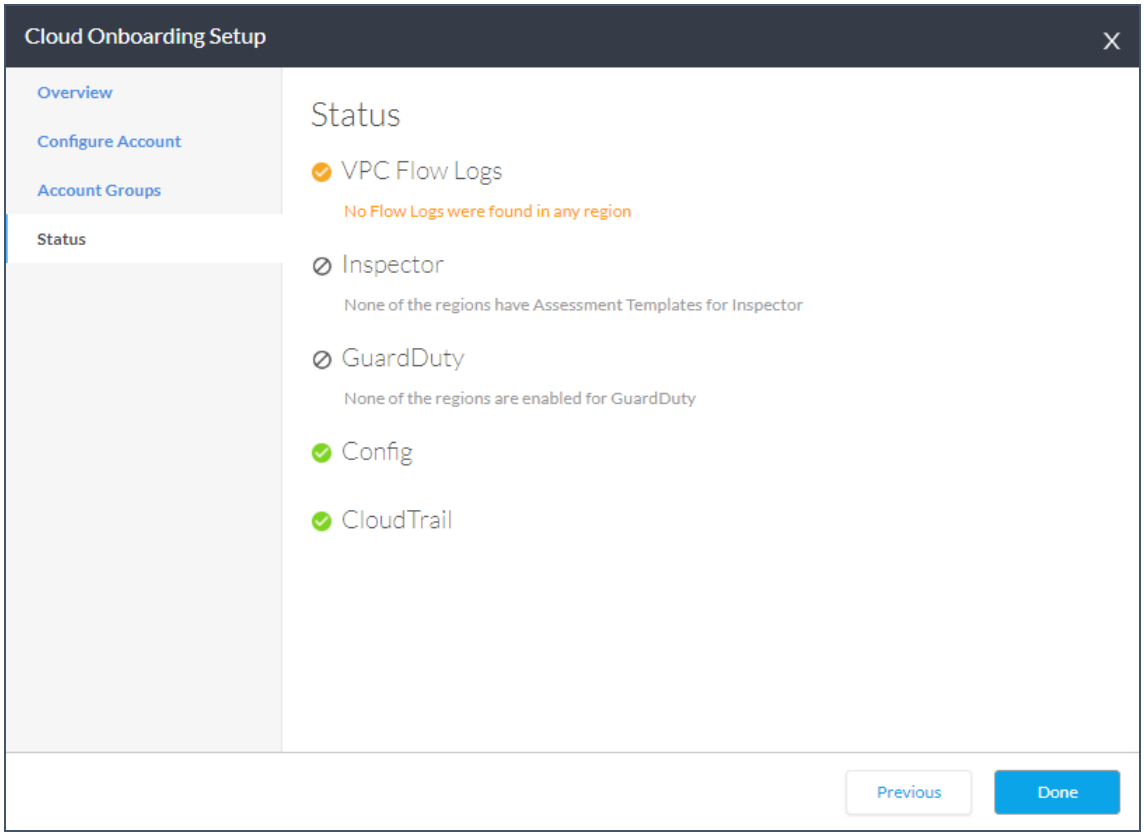


Ilustración 41: Configuración exitosa de la cuenta de AWS en Prisma

Onboarding de suscripciones de Azure en Prisma Cloud

En el caso de Azure, la integración se realiza a través de un App Service asociado a un Service Principal.

Al igual que en el caso de AWS, Prisma nos solicita el modo en el que funcionará la herramienta:

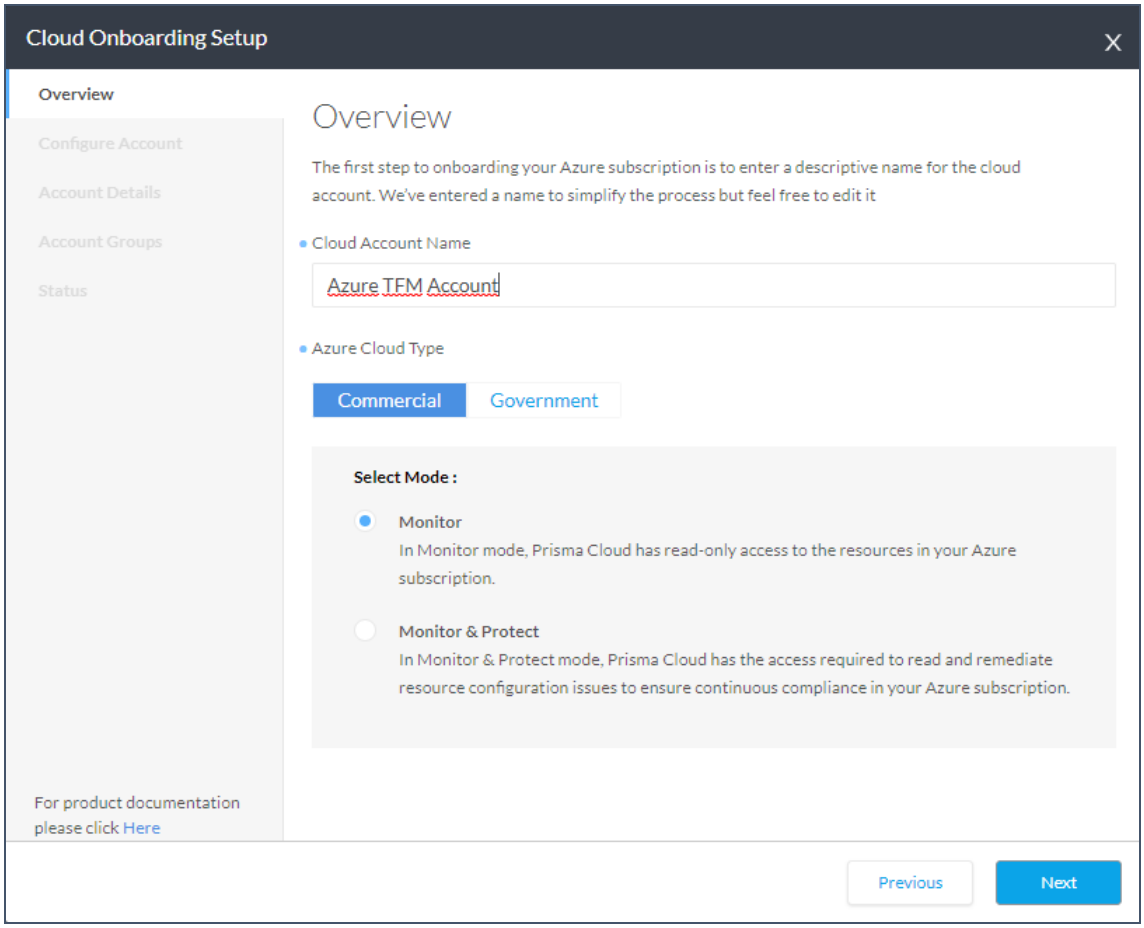


Ilustración 42: Selección de modo de funcionamiento de Prisma con Azure

En este caso, para realizar la creación automatizada de tanto el App Service como de la asignación de roles correspondientes, Prisma nos proporciona una plantilla de Terraform e instrucciones para ejecutarla:

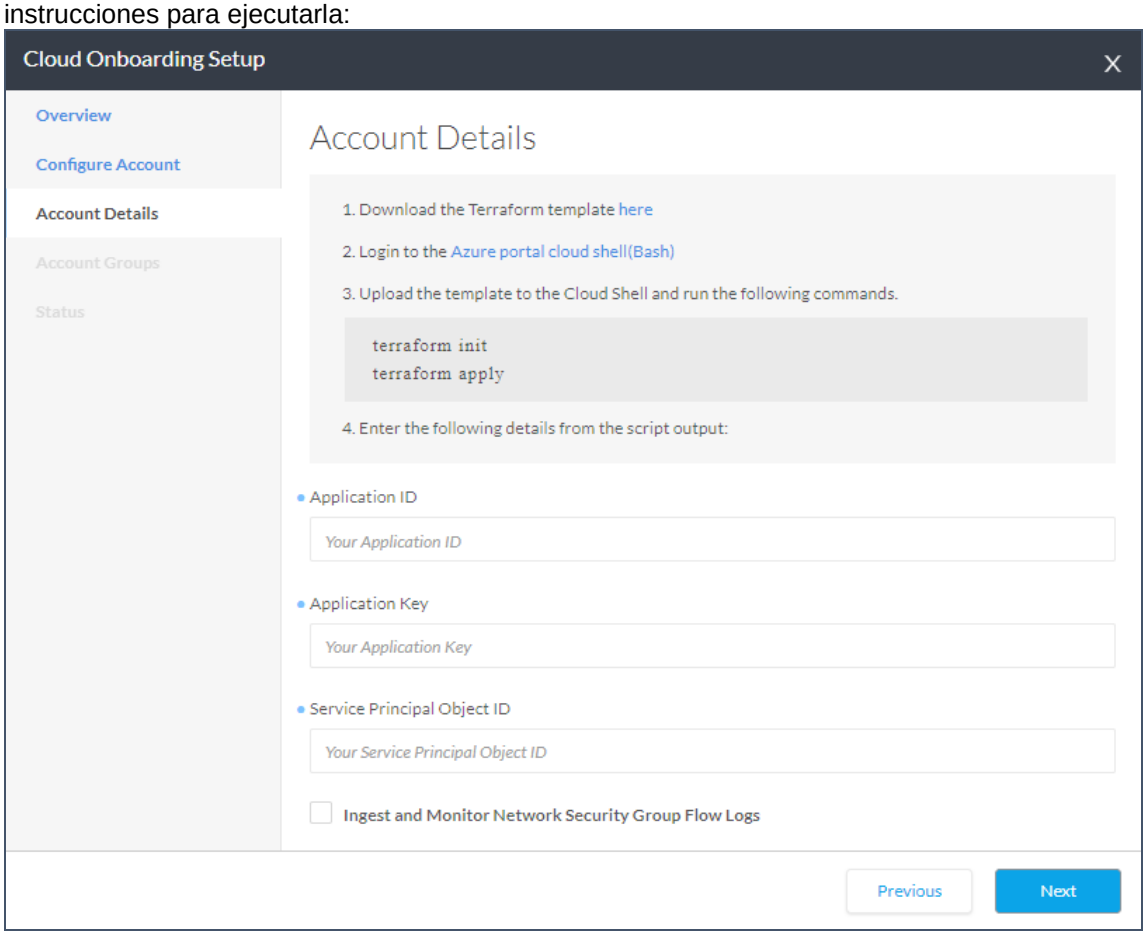


Ilustración 43: Instrucciones para la ejecución de la plantilla de Terraform y formulario solicitando datos

Revisando aspectos de esta plantilla para observar qué despliega exactamente podemos observar que crea el App Service, al que asocia un Service Principal:

```
resource “azuread_application” “prisma_cloud_app” {
  name                    = “Prisma Cloud Onboarding ${random_string.unique_id.result}”
  homepage                = “https://prismacloud.io/”
  available_to_other_tenants = true
}

resource “azuread_service_principal” “prisma_cloud_sp” {
  application_id = azuread_application.prisma_cloud_app.application_id
}
```

Asimismo, podemos ver exactamente qué permisos otorga al rol:

```
variable “custom_role_permissions” {
  type = “list”
  default = [
    “Microsoft.Network/networkInterfaces/effectiveNetworkSecurityGroups/action”,
    “Microsoft.Network/networkWatchers/securityGroupView/action”,
    “Microsoft.Network/networkWatchers/queryFlowLogStatus/action”,
    “Microsoft.Network/virtualwans/vpnconfiguration/action”,
    “Microsoft.ContainerRegistry/registries/webhooks/getCallbackConfig/action”,
    “Microsoft.Web/sites/config/list/action”,
    # “Microsoft.Storage/storageAccounts/*”, # enable remediation for storage account
  ]
}
```

Y posteriormente le otorga los roles predeterminados de Azure “Reader”, “Reader and Data Access” y “Network Contributor”, este último para obtener los flow logs de los NSG:

```
resource “azurerm_role_definition” “custom_prisma_role” {
 name       = “prisma-cloud-policy-${random_string.unique_id.result}”
 role_definition_id = random_uuid.custom_role_uuid.result
 scope      = “/subscriptions/${var.subscription_id}”
 description = “Prisma Cloud custom role created via Terraform”
 assignable_scopes = [“/subscriptions/${var.subscription_id}”]
 permissions {
  actions     = var.custom_role_permissions
  not_actions = []
 }
}

resource “azurerm_role_assignment” “assign_custom_prisma_role” {
 scope      = “/subscriptions/${var.subscription_id}”
 principal_id = azuread_service_principal.prisma_cloud_sp.id
 role_definition_id = “${azurerm_role_definition.custom_prisma_role.id}”
}

resource “azurerm_role_assignment” “assign_reader” {
 scope      = “/subscriptions/${var.subscription_id}”
 principal_id = azuread_service_principal.prisma_cloud_sp.id
 role_definition_name = “Reader”
}

resource “azurerm_role_assignment” “assign_reader_data_access” {
 scope      = “/subscriptions/${var.subscription_id}”
 principal_id = azuread_service_principal.prisma_cloud_sp.id
 role_definition_name = “Reader and Data Access”
}

resource “azurerm_role_assignment” “assign_network_contrib” {
 scope      = “/subscriptions/${var.subscription_id}”
 principal_id = azuread_service_principal.prisma_cloud_sp.id
 role_definition_name = “Network Contributor”
}
```

Siguiendo las instrucciones de Prisma, se ejecuta la plantilla en el Cloudshell de la suscripción de pruebas:

```
sergio@Azure:~$ ls
clouddrive  terraform.tf
sergio@Azure:~$ terraform init

Initializing the backend...

Initializing provider plugins...
- Checking for available provider plugins...
- Downloading plugin for provider "azuread" (hashicorp/azuread) 0.8.0...
- Downloading plugin for provider "azurerm" (hashicorp/azurerm) 2.0.0...
- Downloading plugin for provider "random" (hashicorp/random) 2.2.1...

The following providers do not have any version constraints in configuration,
so the latest version was installed.

To prevent automatic upgrades to new major versions that may contain breaking
changes, it is recommended to add version = "..." constraints to the
corresponding provider blocks in configuration, with the constraint strings
suggested below.

* provider.azuread: version = "~> 0.8"
* provider.random: version = "~> 2.2"

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

Ilustración 44: Ejecución de la plantilla de Terraform desde la Cloudshell de Azure

El resultado de la ejecución es toda la información relacionada con el App Service y el Service Principal necesaria para introducirlo en Prisma y que pueda conectarse a la API de Azure:

```
Apply complete! Resources: 10 added, 0 changed, 0 destroyed.

Outputs:

a__subscription_ids = 34d96e6a-                    4cdd567516ae
b__active_directory_ids = 778ab544-                    8d457fe086fa
c__application_id = ca809fe6-                          f478a
d__application_key = test
e__service_principal_id = b1fd8319-                   -e2f7369598a6
sergio@Azure:~$
```

Ilustración 45: Ejecución finalizada y datos obtenidos

Tras introducir los datos y comprobarlos, Prisma nos indica que la configuración se ha realizado con éxito:

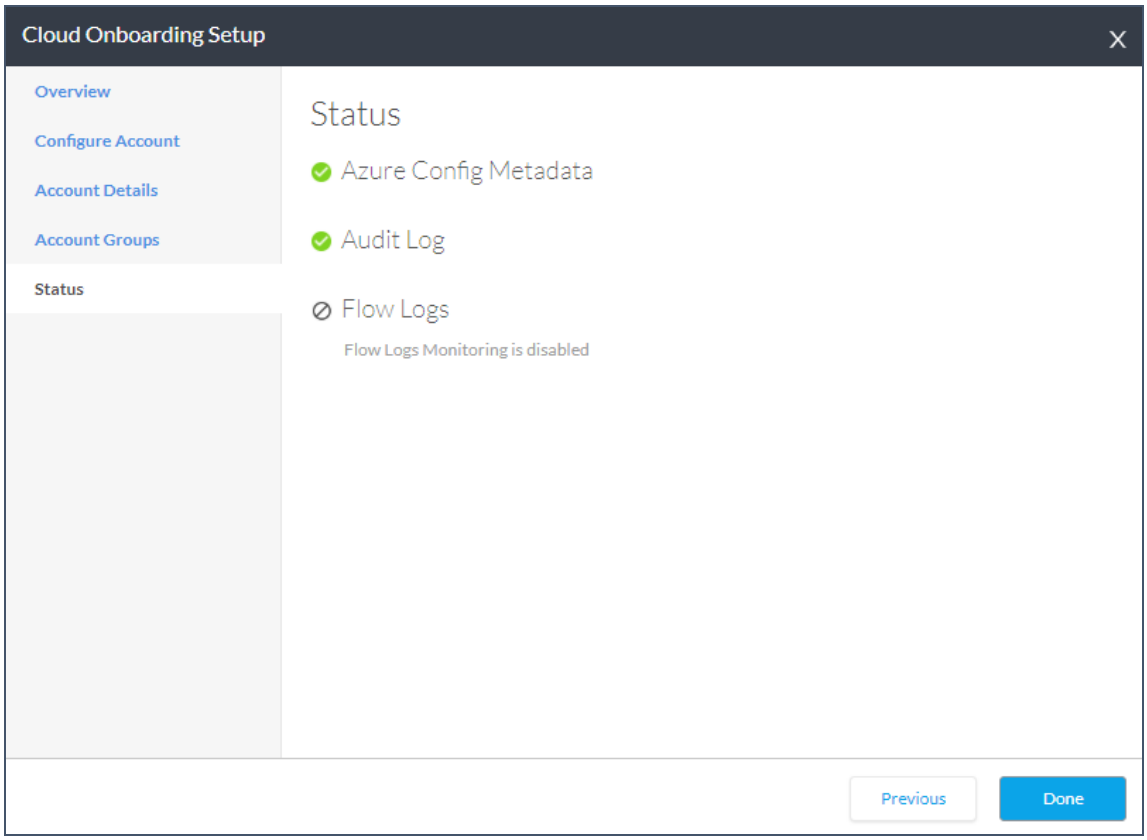


Ilustración 46: Configuración exitosa de la suscripción de Azure en Prisma

Onboarding de cuentas de AWS en Dome9

Al igual que en Prisma, Dome9 ofrece dos modos de funcionamiento; solo lectura y remediaciones automáticas.

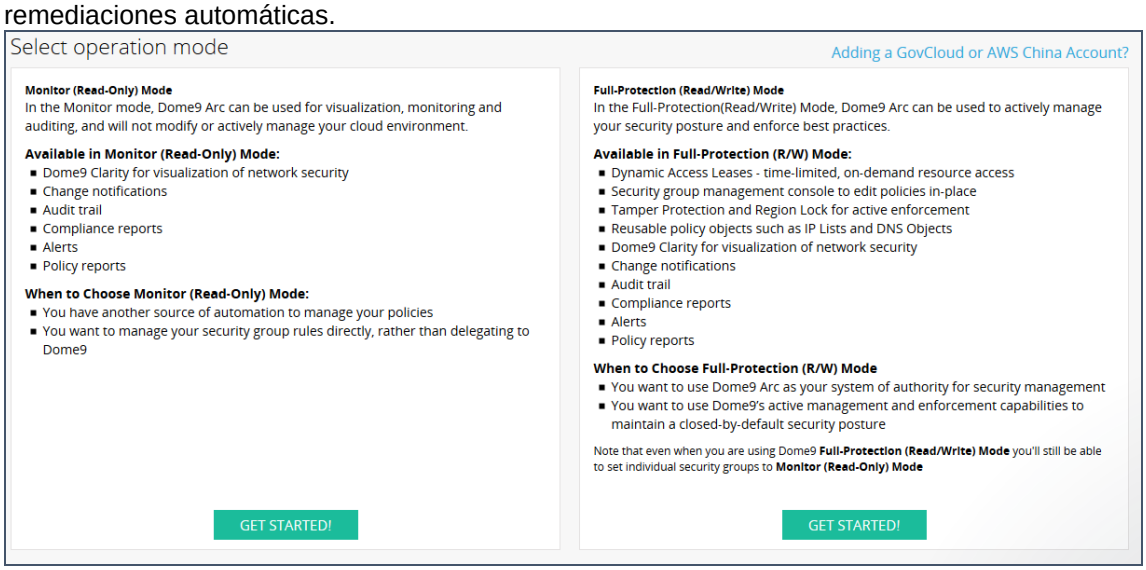


Ilustración 47: Selección de modo de funcionamiento de Dome9 con AWS

En este caso, Dome9 proporciona instrucciones detalladas para realizar la creación de los roles de manera manual, incluso ofreciendo la política de acceso a asociar al rol en un JSON aparte.

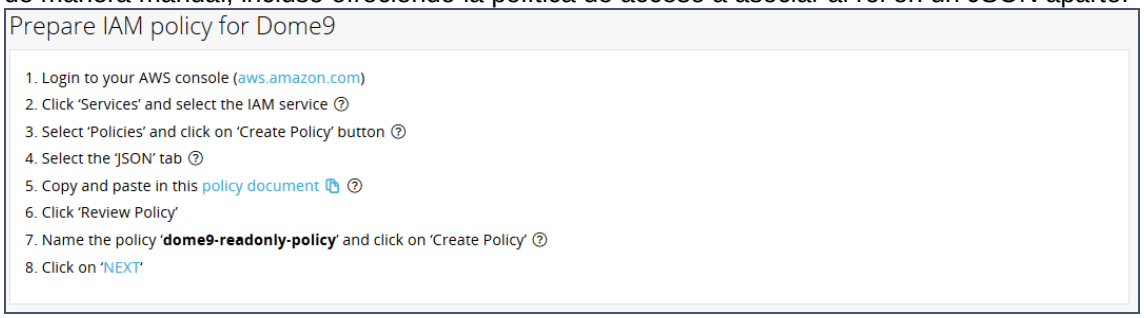


Ilustración 48: Instrucciones paso a paso para preparar la cuenta de AWS

Ilustración 49: Detalle de la política a asociar al rol de Dome9

En este caso, observamos que detalla de manera explícita el acceso a listar y obtener una serie de recursos para poder trabajar sobre la configuración de estos, incluyendo, entre otros, los Bucket S3 o visibilidad del CloudTrail.

Al revisar la política creada en la interfaz de AWS se puede ver en detalle que ofrece acceso de lectura y enumeración a un total de 224 servicios:

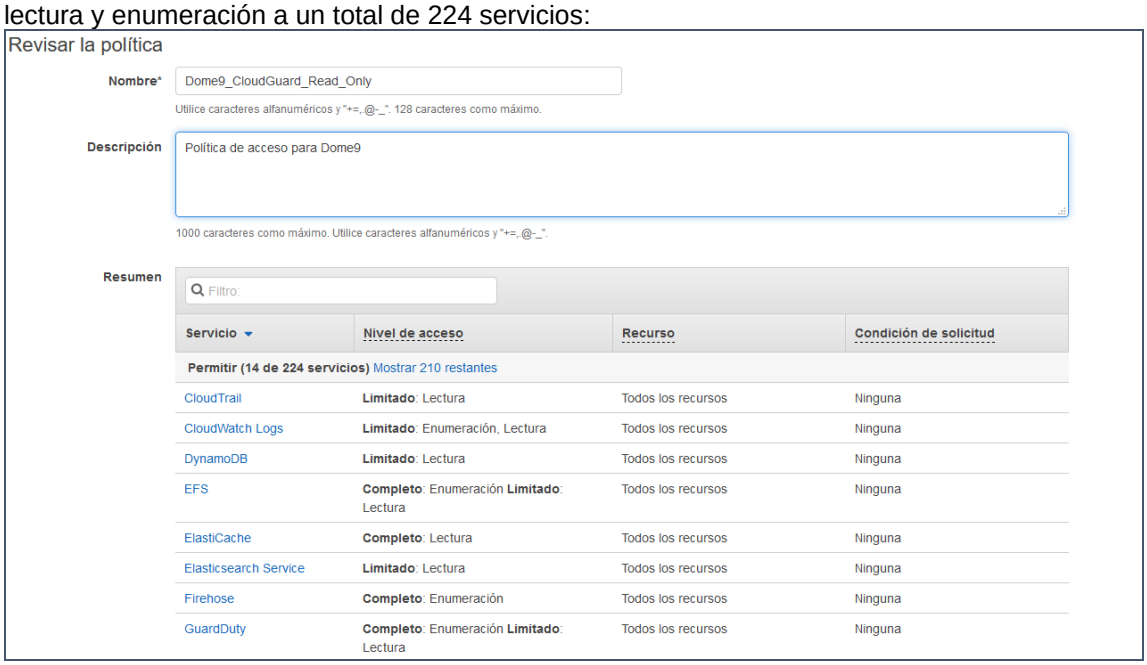


Ilustración 50: Detalle superficial de los recursos disponibles con el rol

Y al igual que en el caso de Prisma, Dome9 se conecta desde una cuenta de AWS para recuperar información de nuestra cuenta. En este caso, tenemos que indicar manualmente la cuenta desde donde van a realizar las conexiones:

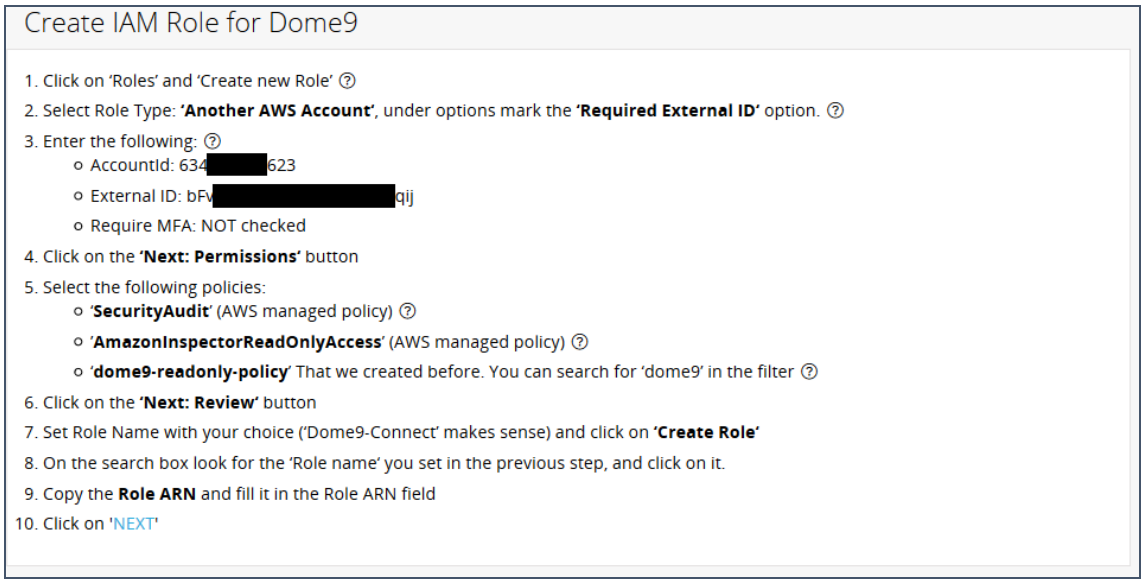


Ilustración 51: Detalle de la cuenta de AWS utilizada por Dome9 e instrucciones adicionales

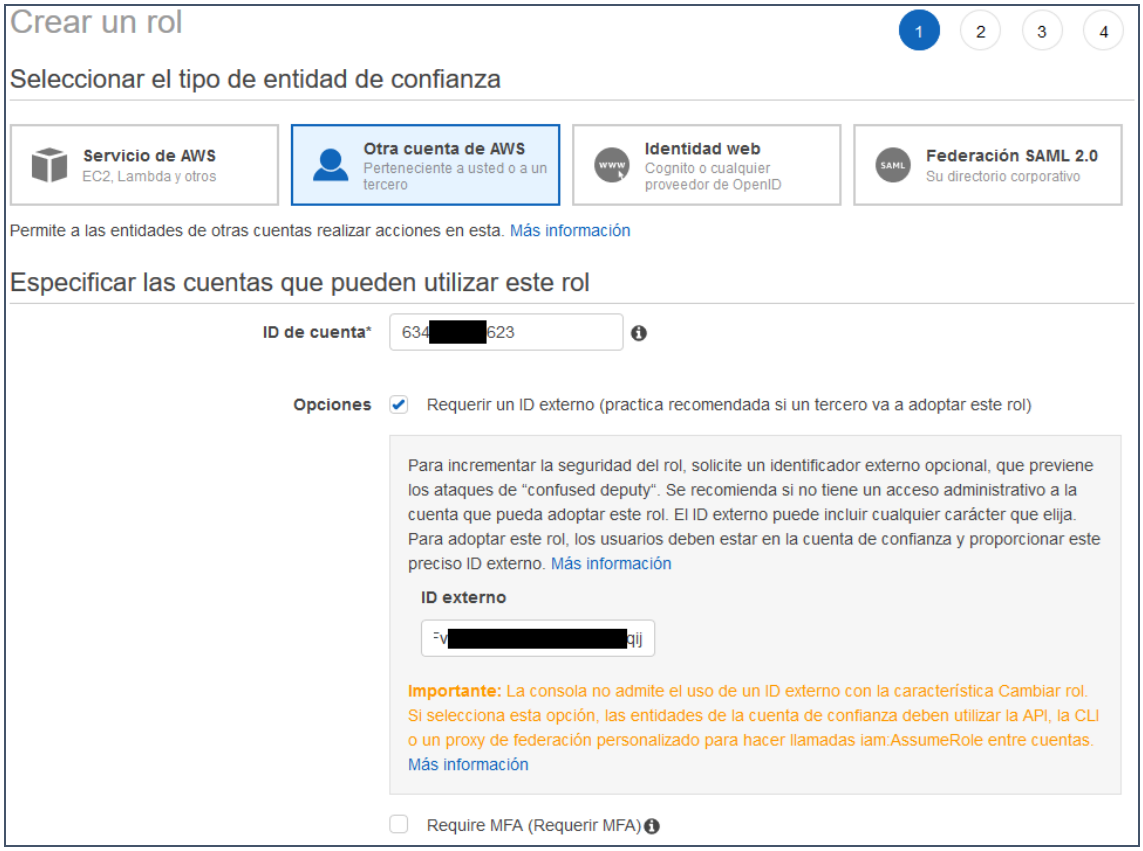


Ilustración 52: Inclusión de la cuenta utilizada por Dome9 e inclusión del external ID

Una vez que se ha finalizado con la creación del rol, utilizaremos su ARN para darlo de alta en Dome9:

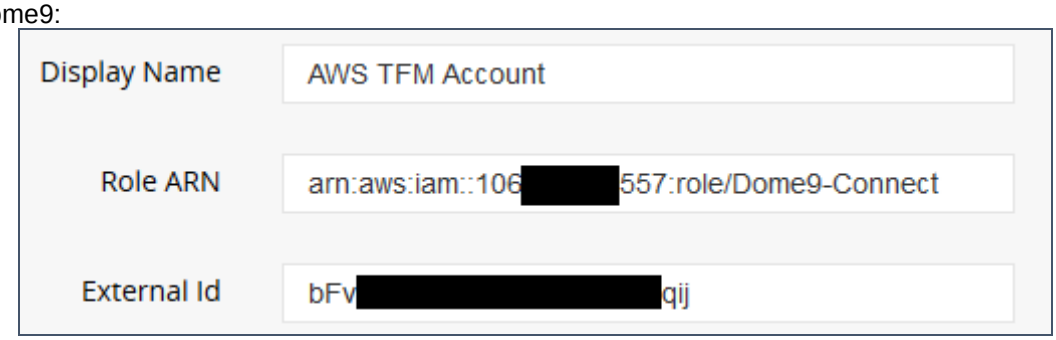


Ilustración 53: Inclusión del ARN del rol generado en Dome9

Una vez que se especifica el ARN del rol, Dome9 verifica que tiene permisos para asumirlo y nos indicará que ha habido éxito con la integración:

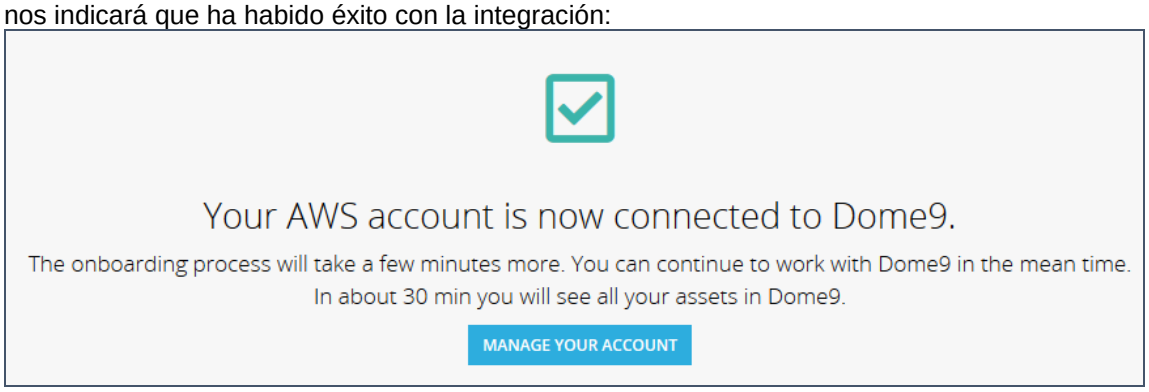


Ilustración 54: Configuración exitosa de la cuenta de AWS en Dome9

Onboarding de suscripciones de Azure en Dome9

Al igual que en AWS y en Prisma, contamos con dos formas de trabajo, solo lectura o remediaciones automáticas:

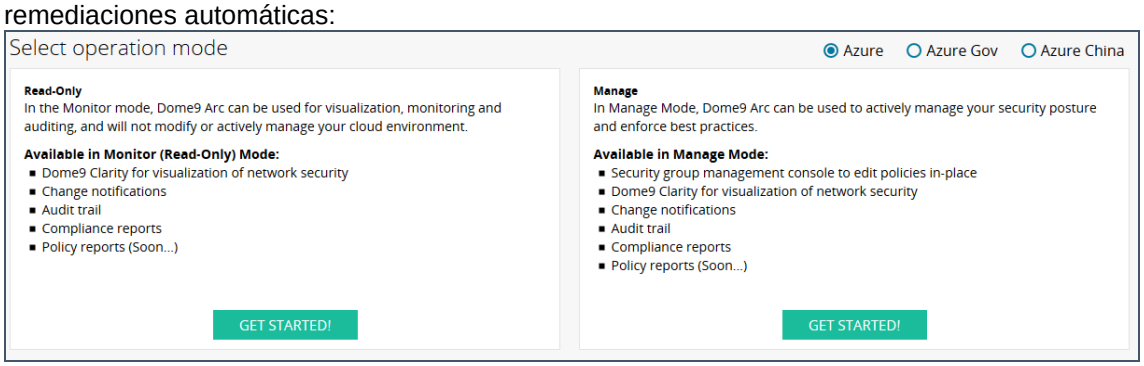


Ilustración 55: Selección de modo de funcionamiento de Dome9 con Azure

El proceso de preparar la conexión con Azure también es manual. Sin embargo, es muy guiado. La configuración es idéntica a la de Prisma con Azure, pero a través de la consola web:

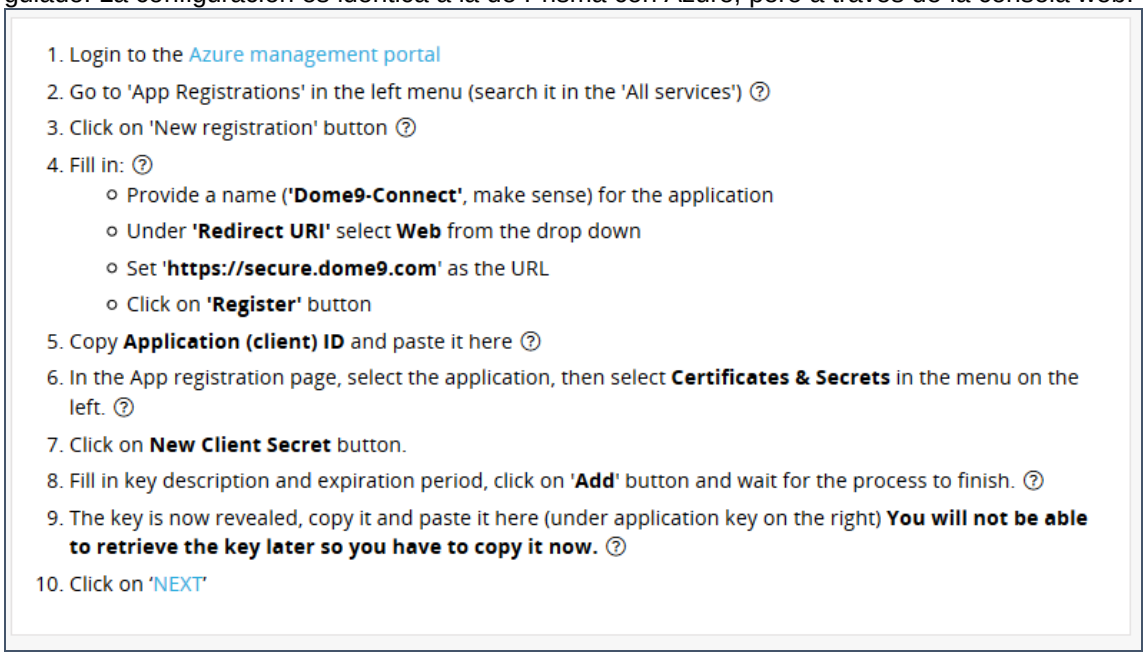


Ilustración 56: Instrucciones paso a paso para la configuración de la suscripción de Azure

Tal y como indican las instrucciones, se crea el App Service indicando la URL de redirección (URL contra la cual va a contrastar que se ha realizado una petición de acceso):

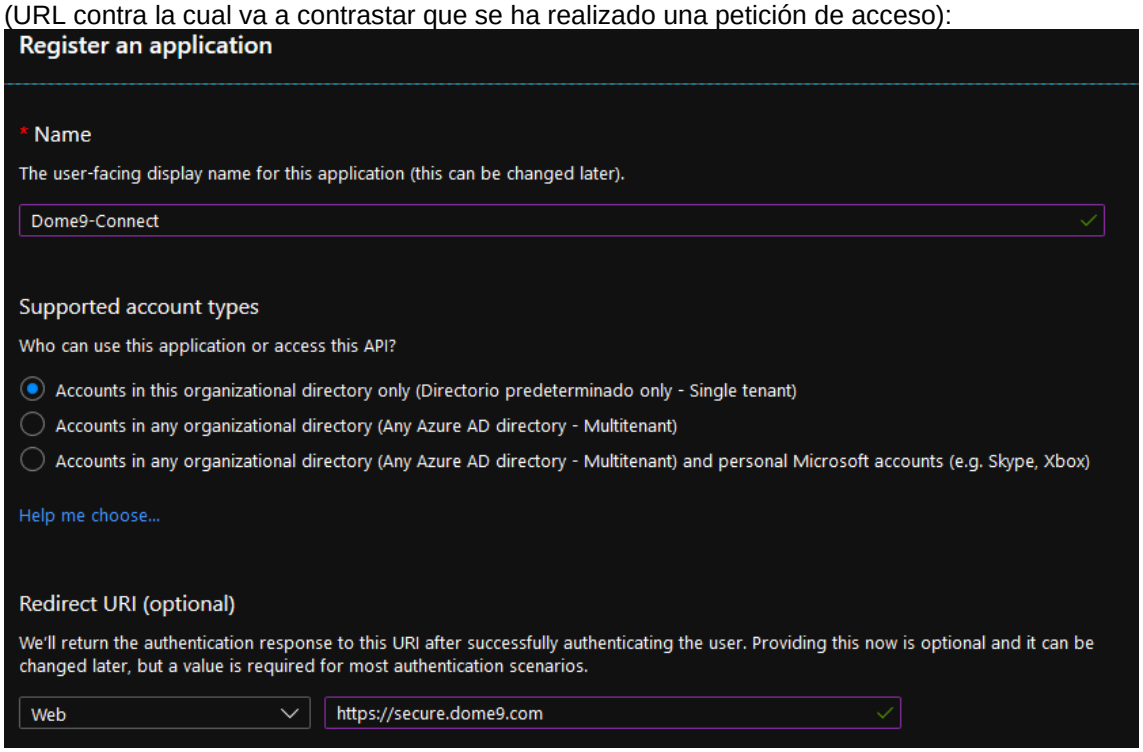


Ilustración 57: Generación de la aplicación en Azure

Tras generar la aplicación, hay que crear una clave (secret) para autenticar a Dome9 y que se le permita el acceso a Azure:

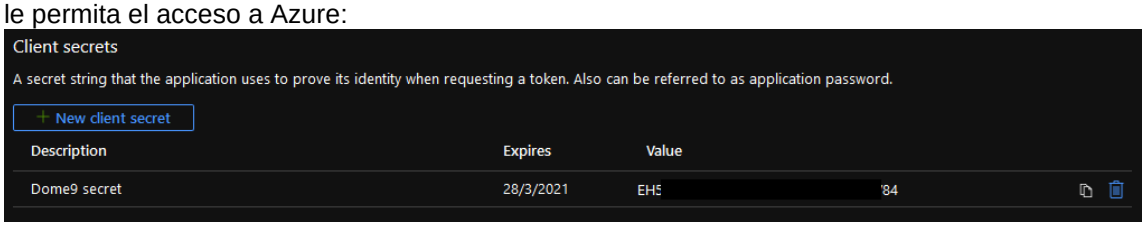


Ilustración 58: Generación y obtención del secreto de la aplicación

Una vez que tenemos el identificador de la aplicación y la key podemos continuar con el proceso:

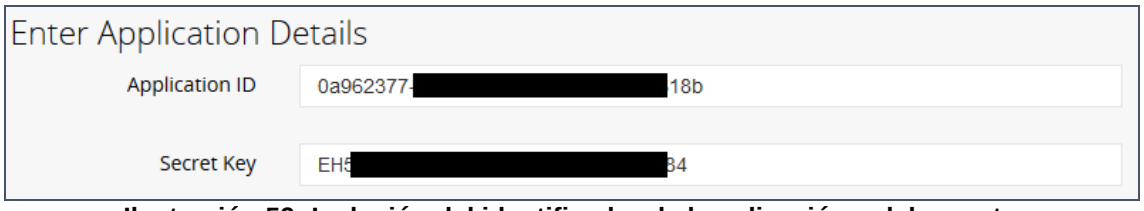


Ilustración 59: Inclusión del identificador de la aplicación y del secreto

Sin embargo, todavía falta asociar los roles adecuados a la aplicación, proceso a través del cual nos guía Dome9:

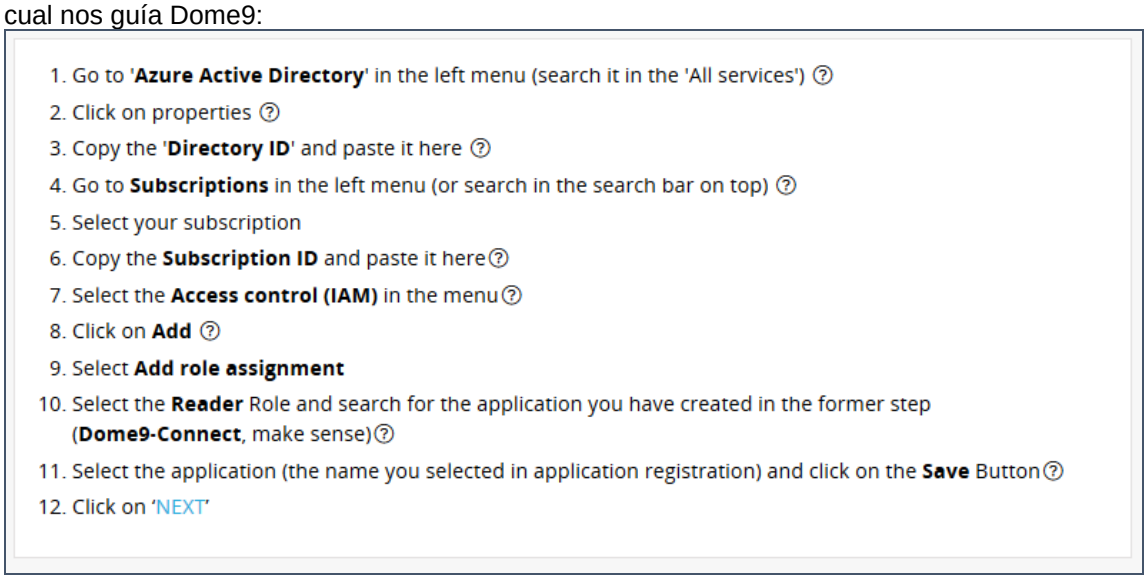


Ilustración 60: Instrucciones para la asignación del rol en Azure

En este caso nos es suficiente con asignarle el rol predeterminado "Reader":

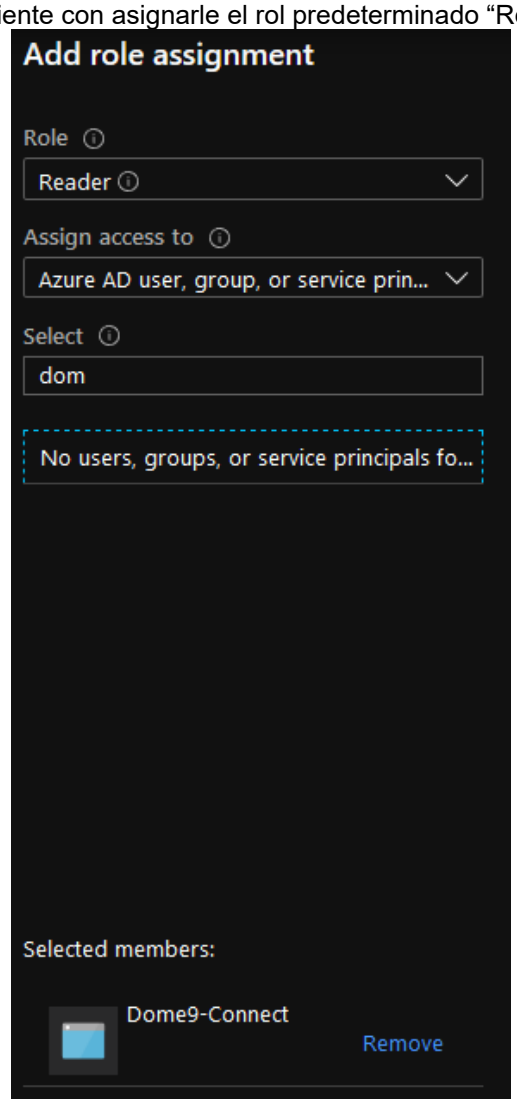


Ilustración 61: Asignación del rol en Azure

Finalmente, terminamos de introducir los datos que nos solicita la herramienta, como el identificador de Active Directory (tenant ID) y el identificador de la suscripción:

Ilustración 62: Inclusión del ID de directorio activo y de la suscripción

Tras realizar la prueba, Dome9 nos indicará que la suscripción de Azure se ha añadido correctamente a la plataforma:

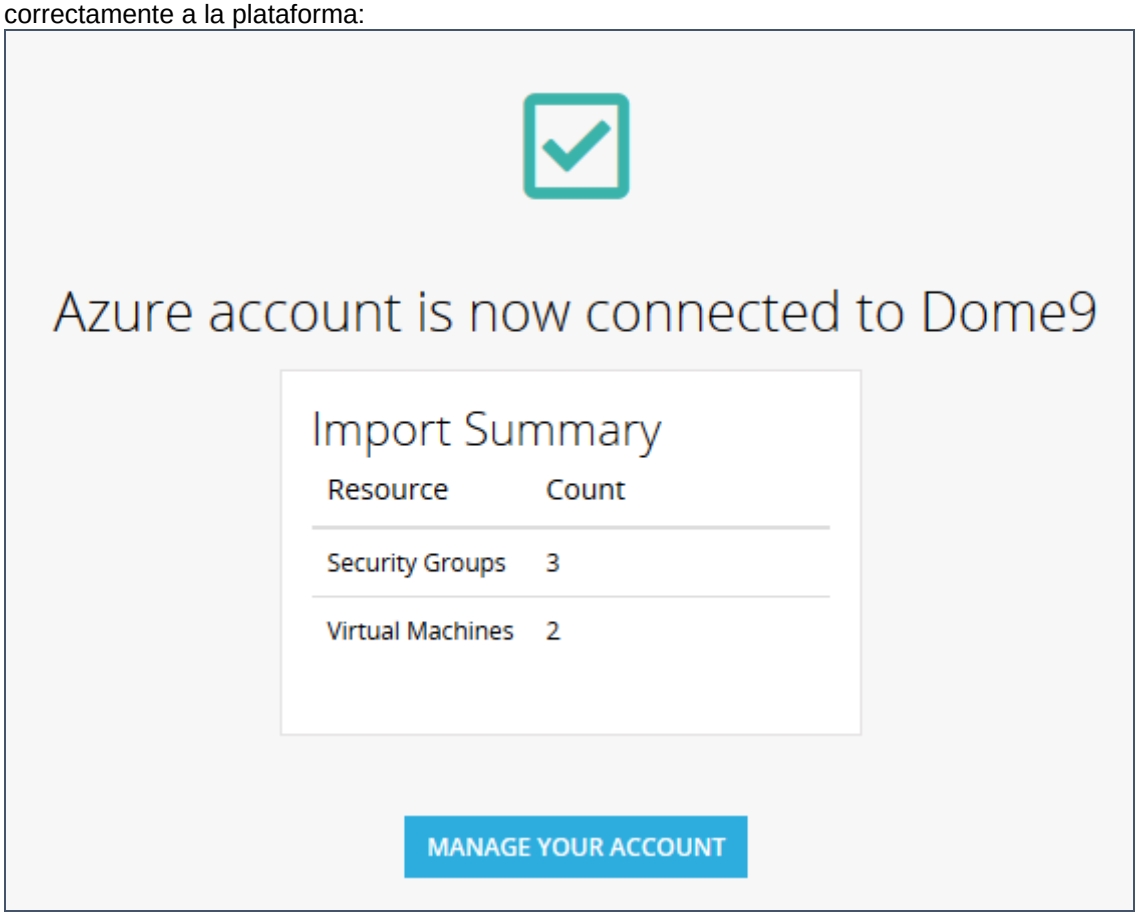


Ilustración 63: Configuración exitosa de la suscripción en Dome9

Onboarding de cuentas de AWS en Aqua Cloudsploit

Para conectar cuentas de AWS en CloudSploit, el procedimiento es prácticamente idéntico al visto en Prisma, aunque hay más métodos disponibles.

La propia herramienta proporciona un Stack para crear todos los recursos necesarios:

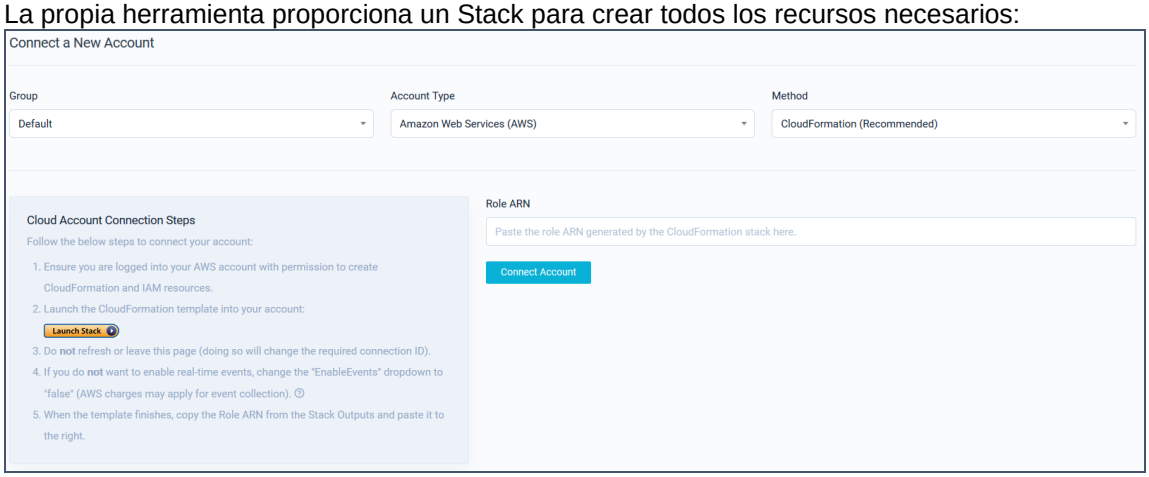


Ilustración 64: Selección del modo de conexión con AWS y acceso al Stack

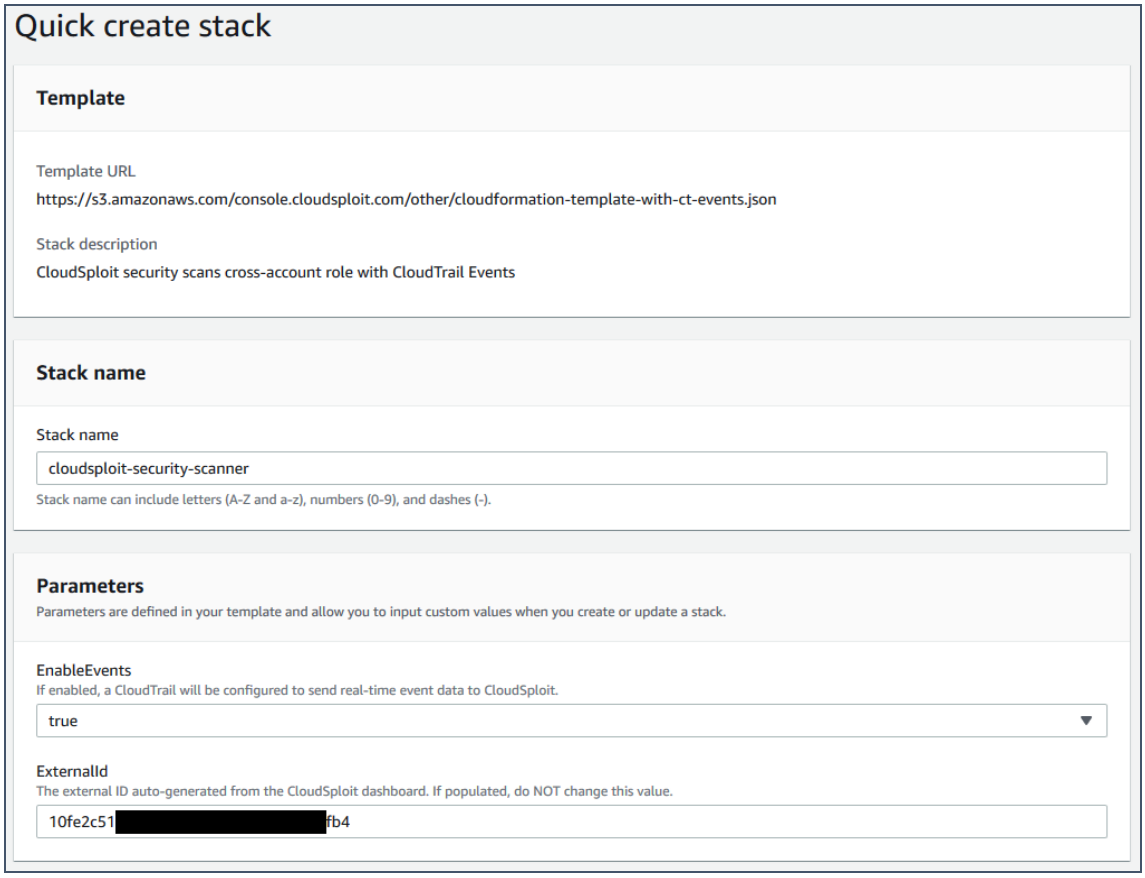


Ilustración 65: Resumen y detalles del Stack

Al igual que con Prisma, una vez que el Stack haya finalizado, copiaremos el ARN del rol creado en el formulario de Cloudsploit para realizar la prueba de conexión:

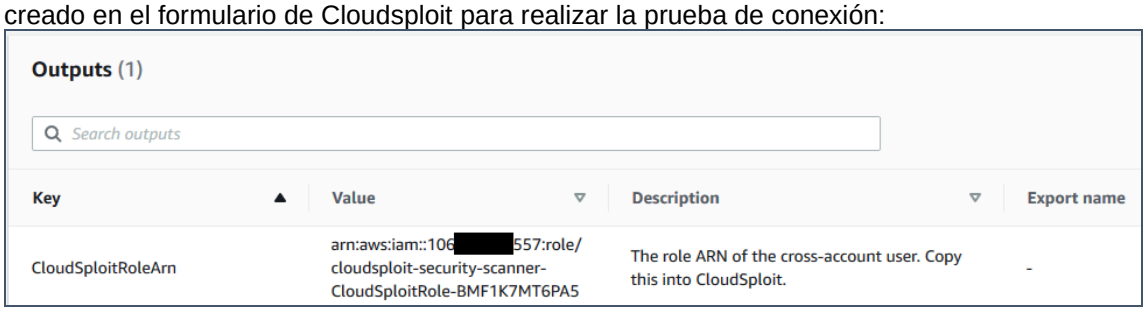


Ilustración 66: ARN resultado de la generación del rol

Este rol incluye la política integrada “SecurityAudit” de AWS sobre un total de 228 servicios:

Ilustración 67: Detalle superficial de los recursos disponibles al rol

Además incluye una política personalizada para complementar el acceso a recursos adicionales:

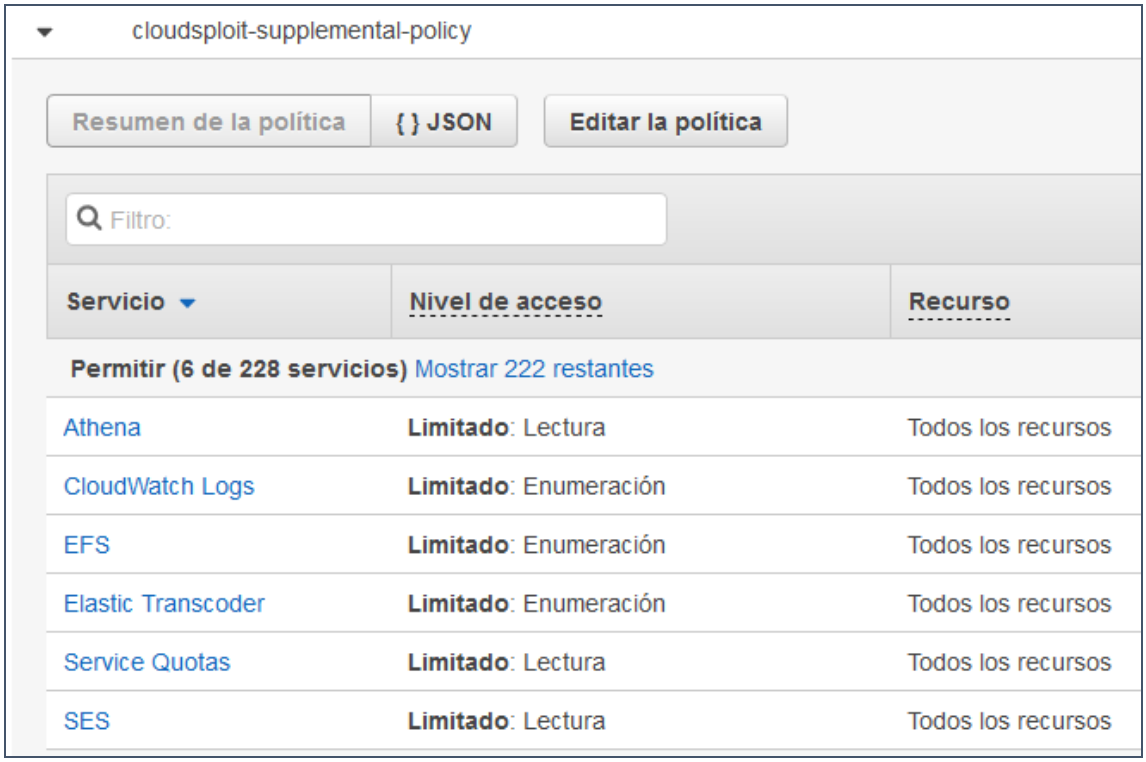


Ilustración 68: Detalle superficial de los recursos adicionales disponibles al rol

Por último, vemos que este rol es asumible por una entidad de confianza, que corresponde con la cuenta de AWS desde donde se ejecuta Cloudsploit:

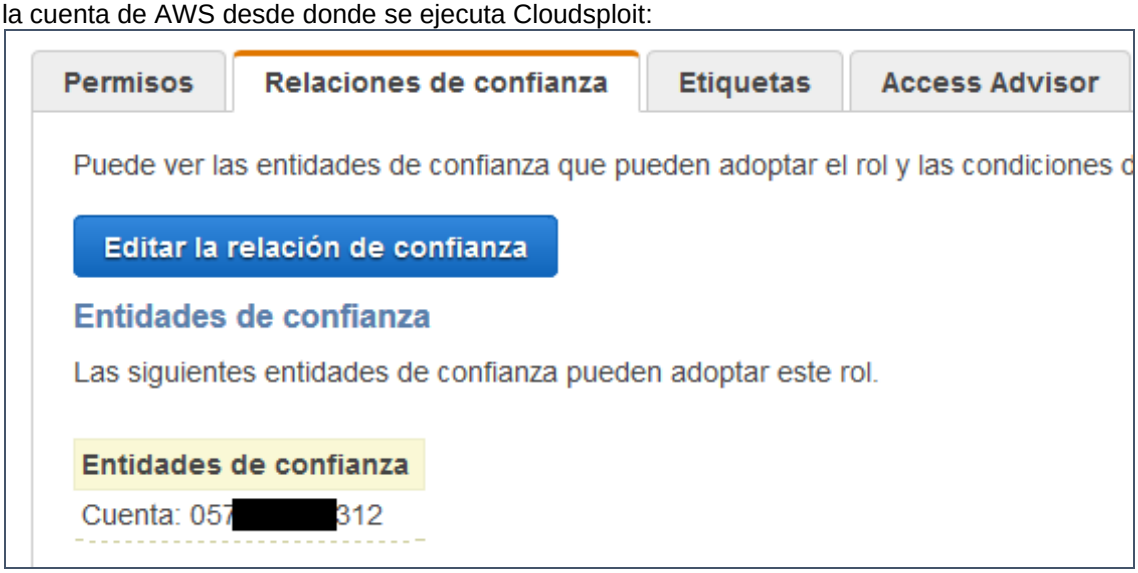


Ilustración 69: Detalle de la cuenta de AWS de Cloudsploit que puede asumir el rol

Tras un test, la herramienta nos notificará que la conexión con el entorno de AWS se ha podido realizar correctamente.

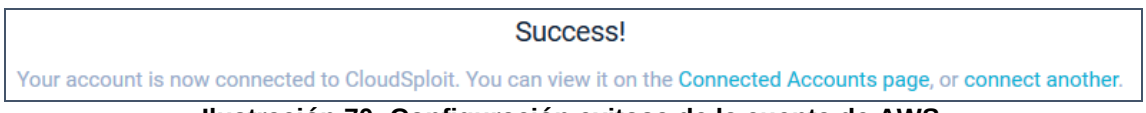


Ilustración 70: Configuración exitosa de la cuenta de AWS

Onboarding de suscripciones de Azure en Aqua Cloudsploit

Al contrario que con AWS, en Azure Cloudsploit ofrece el mismo procedimiento que Dome9 con Azure, el proceso manual de creación del App Service y la asignación del rol.

Se nos solicita el identificador de aplicación (visto en Dome9), la key de la aplicación (visto en Dome9), el identificador de Active Directory (tenant ID) y el identificador de la suscripción.

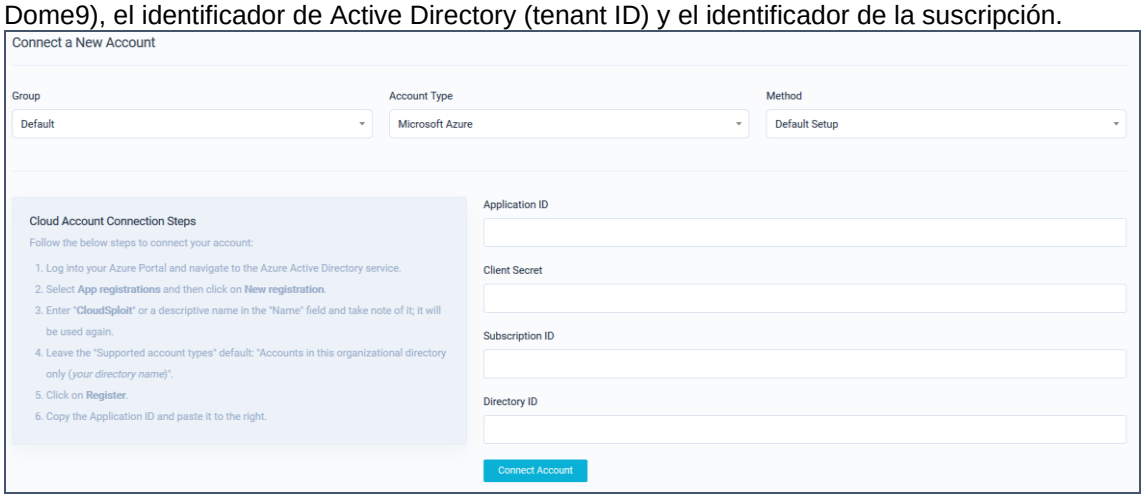


Ilustración 71: Selección del modo de integración con Azure e instrucciones paso a paso

Añadiremos también la asignación del rol de "Reader" a la aplicación que se ha creado para Cloudsploit:

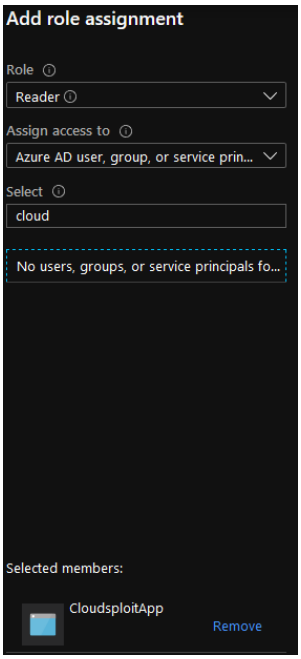


Ilustración 72: Asignación del rol a la aplicación generada

Tras añadir la asignación del rol de "Reader" a la aplicación realizaremos la prueba de conexión. En caso de que tenga éxito observaremos el siguiente mensaje:

Ilustración 73: Configuración exitosa de la suscripción de Azure

Resto de vulnerabilidades de Prisma en AWS

En este apartado se recogen las infracciones relacionadas con las políticas de contraseñas en AWS.

Severidad: Media **NºVulnerabilidad:** 6 **Reglamento**: CIS, RGPD
Nombre: Asegurar que la política de contraseñas obliga a incluir letras mayúsculas
Descripción: Las políticas de contraseñas ayudan a securizar el acceso a la consola de AWS, por lo tanto, es recomendable aumentar la complejidad de las mismas.
Recursos afectados: Política de contraseñas

Severidad: Media **NºVulnerabilidad:** 7 **Reglamento**: CIS, RGPD
Nombre: Asegurar que la política de contraseñas obliga a incluir letras minúsculas
Descripción: Las políticas de contraseñas ayudan a securizar el acceso a la consola de AWS, por lo tanto, es recomendable aumentar la complejidad de las mismas.
Recursos afectados: Política de contraseñas

Severidad: Media **NºVulnerabilidad:** 8 **Reglamento**: CIS, RGPD
Nombre: Asegurar que la política de contraseñas obliga a incluir símbolos
Descripción: Las políticas de contraseñas ayudan a securizar el acceso a la consola de AWS, por lo tanto, es recomendable aumentar la complejidad de las mismas.
Recursos afectados: Política de contraseñas

Severidad: Media **NºVulnerabilidad:** 9 **Reglamento**: CIS, RGPD
Nombre: Asegurar que la política de contraseñas obliga a incluir números
Descripción: Las políticas de contraseñas ayudan a securizar el acceso a la consola de AWS, por lo tanto, es recomendable aumentar la complejidad de las mismas.
Recursos afectados: Política de contraseñas

Severidad: Media **NºVulnerabilidad:** 10 **Reglamento**: CIS, RGPD
Nombre: Asegurar que la política de contraseñas obliga a establecer contraseñas de una longitud mínima de 14 caracteres
Descripción: Las políticas de contraseñas ayudan a securizar el acceso a la consola de AWS, por lo tanto, es recomendable aumentar la complejidad de las mismas.
Recursos afectados: Política de contraseñas

Severidad: Media **NºVulnerabilidad:** 11 **Reglamento**: CIS, RGPD
Nombre: Asegurar que la política de contraseñas impide reutilizar contraseñas
Descripción: Las políticas de contraseñas ayudan a securizar el acceso a la consola de AWS, por lo tanto, es recomendable aumentar la complejidad de las mismas.
Recursos afectados: Política de contraseñas

Severidad: Media **NºVulnerabilidad:** 12 **Reglamento**: CIS, RGPD
Nombre: Asegurar que las contraseñas expiran en 90 días o menos
Descripción: Las políticas de contraseñas ayudan a securizar el acceso a la consola de AWS, por lo tanto, es recomendable aumentar la complejidad de las mismas.
Recursos afectados: Política de contraseñas

Resto de vulnerabilidades de Prisma en Azure

Aquí se recogen vulnerabilidades reportadas por Prisma en Azure relacionadas con Azure Security Center.

Severidad: Media **NºVulnerabilidad:** 4 **Reglamento**: CIS
Nombre: ASC no está activado
Descripción: Asegurar que el tier seleccionado para ASC es “standard pricing” y que monitorice el entorno.
Recursos afectados: default (ASC)

Severidad: Media **NºVulnerabilidad:** 5 **Reglamento**: CIS
Nombre: El despliegue automático del agente de ASC no está habilitado
Descripción: Habilitar el despliegue automático del agente de ASC permite proteger las cargas frente a posibles vulnerabilidades e infecciones, además de proporcionar un sistema de alertas.
Recursos afectados: default (ASC)

Severidad: Media **NºVulnerabilidad:** 6 **Reglamento**: CIS, PCI, RGPD, NIST
Nombre: No existe email de contacto configurado en ASC
Descripción: Asegurar que existe un email configurado en ASC con el fin de que las alertas de seguridad sean recibidas por un responsable.
Recursos afectados: default (ASC)

Severidad: Media **NºVulnerabilidad:** 7 **Reglamento**: CIS, PCI, RGPD, NIST
Nombre: Asegurar que las notificaciones de alertas por email están habilitadas en ASC
Descripción: Con el fin de que las amenazas y vulnerabilidades detectadas sean tratadas, es necesario activar el envío de alertas en ASC
Recursos afectados: default (ASC)

Severidad: Media **NºVulnerabilidad:** 8 **Reglamento**: CIS, PCI, RGPD, NIST
Nombre: Habilitar la opción de enviar alertas al “owner” de la suscripción
Descripción: Con el fin de que todos los responsables del entorno se den por enterados de los problemas de seguridad, es necesario activar el envío de alertas a los owners de la suscripción.
Recursos afectados: default (ASC)

Resto de vulnerabilidades de Dome9 en AWS

Aquí se recogen vulnerabilidades reportadas por Dome9 en AWS muy similares entre sí, relacionadas con alertas de CloudWatch y las políticas de contraseñas de la cuenta.

Severidad: Media **NºVulnerabilidad:** 21 **Reglamento**: CIS, RGPD, NIST
Nombre: Asegurar que todos los logs de CloudTrail están integrados con CloudWatch.
Descripción: Es recomendable monitorizar en tiempo real las llamadas API y cambios detectados en el entorno utilizando los logs de CloudTrail con CloudWatch.
Recursos afectados: cloudsploit-security-scanner-CloudSploitCloudTrail-TH0YCB80C5RU

Severidad: Media **NºVulnerabilidad:** 22 **Reglamento**: CIS, RGPD, NIST
Nombre: Asegurar que existe un filtro de métricas y alarmas para cambios de configuración en AWS
Descripción: Es recomendable monitorizar en tiempo real las llamadas API y cambios detectados en el entorno utilizando los logs de CloudTrail con CloudWatch.
Recursos afectados: -

Severidad: Media **NºVulnerabilidad:** 23 **Reglamento**: CIS, RGPD, NIST
Nombre: Crear una alarma de métricas cuando se detecten cambios en los dispositivos de red
Descripción: Es recomendable monitorizar en tiempo real las llamadas API y cambios detectados en el entorno utilizando los logs de CloudTrail con CloudWatch.
Recursos afectados: -

Severidad: Media **NºVulnerabilidad:** 24 **Reglamento**: CIS, RGPD, NIST
Nombre: Crear una alarma de métricas cuando se detecten fallos de autenticación en la consola de AWS.
Descripción: Es recomendable monitorizar en tiempo real las llamadas API y cambios detectados en el entorno utilizando los logs de CloudTrail con CloudWatch.
Recursos afectados: -

Severidad: Media **NºVulnerabilidad:** 25 **Reglamento**: CIS, RGPD, NIST
Nombre: Crear una alarma de métricas cuando se detecten cambios en las políticas IAM.
Descripción: Es recomendable monitorizar en tiempo real las llamadas API y cambios detectados en el entorno utilizando los logs de CloudTrail con CloudWatch.
Recursos afectados: -

Severidad: Media **NºVulnerabilidad:** 26 **Reglamento**: CIS, RGPD, NIST
Nombre: Crear una alarma de métricas cuando se detecte el borrado, programado o no, de CMKs.
Descripción: Es recomendable monitorizar en tiempo real las llamadas API y cambios detectados en el entorno utilizando los logs de CloudTrail con CloudWatch.
Recursos afectados: -

Severidad: Media **NºVulnerabilidad:** 27 **Reglamento**: CIS, RGPD, NIST
Nombre: Crear una alarma de métricas cuando se detecten accesos a la consola de AWS sin MFA.
Descripción: Es recomendable monitorizar en tiempo real las llamadas API y cambios detectados en el entorno utilizando los logs de CloudTrail con CloudWatch.
Recursos afectados: -

Severidad: Media **NºVulnerabilidad:** 28 **Reglamento**: CIS, RGPD, NIST
Nombre: Crear una alarma de métricas cuando se detecten cambios en tablas de rutas
Descripción: Es recomendable monitorizar en tiempo real las llamadas API y cambios detectados en el entorno utilizando los logs de CloudTrail con CloudWatch.
Recursos afectados: -

Severidad: Media **NºVulnerabilidad:** 29 **Reglamento**: CIS, RGPD, NIST
Nombre: Crear una alarma de métricas cuando se detecten llamadas a las APIs no autorizadas
Descripción: Es recomendable monitorizar en tiempo real las llamadas API y cambios detectados en el entorno utilizando los logs de CloudTrail con CloudWatch.
Recursos afectados: -

Severidad: Media **NºVulnerabilidad:** 30 **Reglamento**: CIS, RGPD, NIST
Nombre: Crear una alarma de métricas cuando se detecten cambios en VPCs
Descripción: Es recomendable monitorizar en tiempo real las llamadas API y cambios detectados en el entorno utilizando los logs de CloudTrail con CloudWatch.
Recursos afectados: -

Severidad: Media **NºVulnerabilidad:** 31 **Reglamento**: CIS, RGPD, NIST
Nombre: Crear una alarma de métricas cuando se detecten cambios en SGs
Descripción: Es recomendable monitorizar en tiempo real las llamadas API y cambios detectados en el entorno utilizando los logs de CloudTrail con CloudWatch.
Recursos afectados: -

Severidad: Media **NºVulnerabilidad:** 32 **Reglamento**: CIS, RGPD, NIST
Nombre: Crear una alarma de métricas cuando se detecte el uso de la cuenta root
Descripción: Es recomendable monitorizar en tiempo real las llamadas API y cambios detectados en el entorno utilizando los logs de CloudTrail con CloudWatch.
Recursos afectados: -

Severidad: Media **NºVulnerabilidad:** 33 **Reglamento**: CIS, RGPD, NIST
Nombre: Crear una alarma de métricas cuando se detecten cambios en las políticas de Buckets S3
Descripción: Es recomendable monitorizar en tiempo real las llamadas API y cambios detectados en el entorno utilizando los logs de CloudTrail con CloudWatch.
Recursos afectados: -

Severidad: Media **NºVulnerabilidad:** 34 **Reglamento**: CIS, RGPD, NIST
Nombre: Crear una alarma de métricas cuando se detecten cambios en las NACL
Descripción: Es recomendable monitorizar en tiempo real las llamadas API y cambios detectados en el entorno utilizando los logs de CloudTrail con CloudWatch.
Recursos afectados: -

Severidad: Media **NºVulnerabilidad:** 35 **Reglamento**: CIS, RGPD, NIST
Nombre: Asegurar que la política de contraseñas requiere como mínimo una letra minúscula
Descripción: Se recomienda incluir al menos una letra minúscula en las contraseñas para aumentar la robustez de las mismas.
Recursos afectados: Políticas de contraseñas

Severidad: Media **NºVulnerabilidad:** 36 **Reglamento**: CIS, RGPD, NIST
Nombre: Asegurar que la política de contraseñas solicita al menos una letra mayúscula
Descripción: Se recomienda incluir al menos una letra mayúscula en las contraseñas para aumentar la robustez de las mismas.
Recursos afectados: Políticas de contraseñas

Severidad: Media **NºVulnerabilidad:** 37 **Reglamento**: CIS, RGPD, NIST
Nombre: Asegurar que la política de contraseñas solicita al menos un símbolo.
Descripción: Se recomienda incluir caracteres especiales en las contraseñas para aumentar la robustez de las mismas.
Recursos afectados: Políticas de contraseñas

Severidad: Media **NºVulnerabilidad:** 38 **Reglamento**: CIS, RGPD, NIST
Nombre: Asegurar que la política de contraseñas solicita al menos un número
Descripción: Se recomienda incluir al menos una letra mayúscula en las contraseñas para aumentar la robustez de las mismas.
Recursos afectados: Políticas de contraseñas

Severidad: Media **NºVulnerabilidad:** 39 **Reglamento**: CIS, RGPD, NIST
Nombre: Asegurar que la política de contraseñas requiere una longitud igual o superior a 14 caracteres
Descripción: Se recomienda que las contraseñas tengan una longitud mínima de 14 caracteres para aumentar la robustez de las mismas.
Recursos afectados: Políticas de contraseñas

Resto de vulnerabilidades de Dome9 en Azure

En este apartado se recogen vulnerabilidades similares entre sí reportadas por Dome9 en el entorno de Azure.

Severidad: Media **NºVulnerabilidad:** 10 **Reglamento**: CIS
Nombre: Asegurar que la política “Monitor SQL Auditing” de ASC está activa
Descripción: Esta política de Azure Security Center permite auditar el acceso a las bases de datos y detectar posibles amenazas.
Recursos afectados: ASC Default

Severidad: Media **NºVulnerabilidad:** 11 **Reglamento**: CIS
Nombre: Asegurar que la política “Monitor OS Vulnerabilities” de ASC está activa
Descripción: Esta política de ASC analiza el sistema operativo de los hosts con el objetivo de encontrar problemas que podrían dejar a una máquina virtual vulnerable ante un ataque.
Recursos afectados: ASC Default

Severidad: Media **NºVulnerabilidad:** 12 **Reglamento**: CIS
Nombre: Asegurar que la política “Monitor JIT Network Access” de ASC está activa
Descripción: Habilitar acceso JIT para las máquinas virtuales y bloquear el tráfico de entrada cuando no es necesario.
Recursos afectados: ASC Default

Severidad: Media **NºVulnerabilidad:** 13 **Reglamento**: CIS
Nombre: Asegurar que la política “Monitor System Updates” de ASC está activa
Descripción: Habilitar la monitorización de actualizaciones de las máquinas virtuales para asegurar que el software se mantiene actualizado con los parches de seguridad correspondientes.
Recursos afectados: ASC Default

Severidad: Media **NºVulnerabilidad:** 14 **Reglamento**: CIS
Nombre: Asegurar que la política “Monitor Adaptive Application Whitelisting” de ASC está activa
Descripción: Habilitar el control adaptativo de aplicaciones, controlando qué aplicativos pueden ejecutarse en las máquinas virtuales, ayudando al hardening de las mismas.
Recursos afectados: ASC Default

Severidad: Media **NºVulnerabilidad:** 15 **Reglamento**: CIS
Nombre: Asegurar que la política “Monitor Storage Blob Encryption” de ASC está activa
Descripción: Habilitar el cifrado automático de los elementos almacenados en las Storage Accounts.
Recursos afectados: ASC Default

Severidad: Media **NºVulnerabilidad:** 16 **Reglamento**: CIS
Nombre: Asegurar que el perfil de auditoría registra todas las acciones
Descripción: El perfil de logado de la suscripción debería estar configurado para exportar cualquier tipo de actividad relacionada con el plano de control o de gestión.
Recursos afectados: ASC Default

Severidad: Media **NºVulnerabilidad:** 17 **Reglamento**: CIS
Nombre: Asegurar que la política “Monitor Vulnerability Assessment” de ASC está activa
Descripción: Habilitar la política relacionada con el escaneo de vulnerabilidades de las máquinas virtuales.
Recursos afectados: ASC Default

Severidad: Media **NºVulnerabilidad:** 18 **Reglamento**: CIS
Nombre: Asegurar que la política “Monitor Endpoint Protection” de ASC está activa
Descripción: Habilitar la política que controla que todos los equipos Windows tengan instalada una solución antimalware para evitar infecciones.
Recursos afectados: ASC Default

Severidad: Media **NºVulnerabilidad:** 19 **Reglamento**: CIS
Nombre: Asegurar que la política “Monitor Disk Encryption” de ASC está activa
Descripción: Habilitar esta política que monitoriza que todos los discos duros estén cifrados.
Recursos afectados: ASC Default

Severidad: Media **NºVulnerabilidad:** 20 **Reglamento**: CIS
Nombre: Asegurar que la política “Enable Next Generation Firewall (NGFW) Monitoring” de ASC está activa
Descripción: Habilitar las recomendaciones de filtrado en la capa 7 de red, ya que este tipo de tráfico no se puede controlar con NSGs..
Recursos afectados: ASC Default

Severidad: Media **NºVulnerabilidad:** 21 **Reglamento**: CIS
Nombre: Asegurar que la política “Monitor SQL Encryption” de ASC está activa
Descripción: Habilitar la política que monitoriza que todas las bases de datos utilicen cifrado en reposo.
Recursos afectados: ASC Default

Severidad: Media **NºVulnerabilidad:** 22 **Reglamento**: CIS
Nombre: Asegurar que la política “Monitor Web Application Firewall” de ASC está activa
Descripción: Habilitar la política que comprueba qué dispositivos deberían disponer de protección contra ataques web.
Recursos afectados: ASC Default

Severidad: Media **NºVulnerabilidad:** 23 **Reglamento**: CIS
Nombre: Asegurar que la política “Monitor Network Security Groups” de ASC está activa
Descripción: Habilitar la política de ASC que monitoriza y recomienda configuración de los NSGs utilizados en las máquinas virtuales.
Recursos afectados: ASC Default

Resto de vulnerabilidades de Cloudsploit en Azure

En este apartado se recogen vulnerabilidades adicionales reportadas en Cloudsploit relacionadas con el Azure Security Center.

Severidad: Media **NºVulnerabilidad:** 11 **Reglamento**: CIS
Nombre: Asegurar que los emails de contacto están especificados en Azure Security Center
Descripción: Asegurar que se ha indicado el email del responsable de seguridad que debe recibir las notificaciones de Azure Security Center.
Recursos afectados: default (ASC)

Severidad: Media **NºVulnerabilidad:** 12 **Reglamento**: CIS
Nombre: Asegurar que los números de teléfono de contacto están especificados en Azure Security Center
Descripción: Asegurar que se ha indicado el número de teléfono del responsable de seguridad que debe recibir las notificaciones de Azure Security Center.
Recursos afectados: default (ASC)

Severidad: Media **NºVulnerabilidad:** 13 **Reglamento**: CIS
Nombre: Asegurar que los números de teléfono de contacto están especificados en Azure Security Center
Descripción: Asegurar que se ha indicado el número de teléfono del responsable de seguridad que debe recibir las notificaciones de Azure Security Center.
Recursos afectados: default (ASC)

Severidad: Media **NºVulnerabilidad:** 14 **Reglamento**: CIS
Nombre: Asegurar que está habilitada la opción de enviar notificaciones a los owners de la suscripción
Descripción: Asegurar que se ha activado el envío de las alertas a los owners (dueños) de la suscripción, con el fin de que estén actualizados de los problemas de seguridad de la suscripción.
Recursos afectados: default (ASC)

Severidad: Media **NºVulnerabilidad:** 15 **Reglamento**: CIS, RGPD
Nombre: Asegurar que la política “Monitor Disk Encryption” está habilitada
Descripción: Asegurar que las políticas de Azure Security Center que monitorizan los recursos están habilitadas.
Recursos afectados: default (ASC)

Severidad: Media **NºVulnerabilidad:** 16 **Reglamento**: CIS
Nombre: Asegurar que la retención de los activity logs es como mínimo de 365 días
Descripción: Los activity logs de la suscripción contienen toda la información de las acciones llevadas en la misma, permitiendo localizar acciones ilegítimas llevadas a cabo por los usuarios.
Recursos afectados: default (ASC)

Severidad: Media **NºVulnerabilidad:** 17 **Reglamento**: CIS, RGPD
Nombre: Asegurar que no hay bases de datos SQL que permiten el tráfico entrante desde 0.0.0.0/0
Descripción: Es necesario restringir el tráfico entrante en las bases de datos para evitar posibles fugas de información y exploits.
Recursos afectados: default (ASC)

Severidad: Baja **NºVulnerabilidad:** 18 **Reglamento**: CIS
Nombre: Asegurar que la política “Monitor System Updates” está habilitada
Descripción: Asegurar que las políticas de Azure Security Center que monitorizan los recursos están habilitadas.
Recursos afectados: default (ASC)

Severidad: Baja **NºVulnerabilidad:** 19 **Reglamento**: CIS
Nombre: Asegurar que la política “Monitor OS Vulnerabilities” está habilitada
Descripción: Asegurar que las políticas de Azure Security Center que monitorizan los recursos están habilitadas.
Recursos afectados: default (ASC)

Severidad: Baja **NºVulnerabilidad:** 20 **Reglamento**: CIS
Nombre: Asegurar que la política “Monitor Endpoint Protection” está habilitada
Descripción: Asegurar que las políticas de Azure Security Center que monitorizan los recursos están habilitadas.
Recursos afectados: default (ASC)

Severidad: Baja **NºVulnerabilidad:** 21 **Reglamento**: CIS
Nombre: Asegurar que la política “Monitor Network Security Groups” está habilitada
Descripción: Asegurar que las políticas de Azure Security Center que monitorizan los recursos están habilitadas.
Recursos afectados: default (ASC)

Severidad: Baja **NºVulnerabilidad:** 22 **Reglamento**: CIS
Nombre: Asegurar que la política “Monitor Vulnerability Assessment” está habilitada
Descripción: Asegurar que las políticas de Azure Security Center que monitorizan los recursos están habilitadas.
Recursos afectados: default (ASC)

Severidad: Baja **NºVulnerabilidad:** 23 **Reglamento**: CIS, RGPD
Nombre: Asegurar que la política “Monitor Storage Blob Encryption” está habilitada
Descripción: Asegurar que las políticas de Azure Security Center que monitorizan los recursos están habilitadas.
Recursos afectados: default (ASC)

Severidad: Baja **NºVulnerabilidad:** 24 **Reglamento**: CIS
Nombre: Asegurar que la política “Monitor Adaptive Application Whitelisting” está habilitada
Descripción: Asegurar que las políticas de Azure Security Center que monitorizan los recursos están habilitadas.
Recursos afectados: default (ASC)

Severidad: Baja **NºVulnerabilidad:** 25 **Reglamento**: CIS, RGPD
Nombre: Asegurar que la política “Monitor SQL Auditing” está habilitada
Descripción: Asegurar que las políticas de Azure Security Center que monitorizan los recursos están habilitadas.
Recursos afectados: default (ASC)

Severidad: Baja **NºVulnerabilidad:** 26 **Reglamento**: CIS, RGPD
Nombre: Asegurar que la política “Monitor SQL Encryption” está habilitada
Descripción: Asegurar que las políticas de Azure Security Center que monitorizan los recursos están habilitadas.
Recursos afectados: default (ASC)

Severidad: Baja **NºVulnerabilidad:** 27 **Reglamento**: CIS
Nombre: Asegurar que la política “Monitor JIT Network Access” está habilitada
Descripción: Asegurar que las políticas de Azure Security Center que monitorizan los recursos están habilitadas.
Recursos afectados: default (ASC)

Plantilla Terraform para el onboarding de Azure en Prisma

A continuación, se muestra la plantilla íntegra, anonimizada, utilizada para dar de alta la suscripción de Azure en Prisma.

Observamos que una vez ha definido varios parámetros en variables, tales como el identificador de suscripción o duración de la contraseña del Service Principal, genera un rol customizado, basado en permisos de Microsoft.

El resto de pasos son más sencillos y se encuentran documentados en la propia plantilla, en color azul.

```
###################################
# EDIT THE FOLLOWING PARAMETERS
#
# tenant_id :               Active directory's ID
#                       (Portal) Azure AD -> Properties -> Directory ID
#
# subscription_id:          Subscription ID that you want to onboard
#                       Custom role are going to be created from this subscription
#                       Please use a permanent subscription
#
# service_principal_password:   Specify the service principal password.
#                       Use a strong password. Preferrably 30+ character long
#                       IF YOU change the password, you need to update each onboarded subscription on Prisma Cloud
#
# Definición en variables del identificador de directorio, suscripción, duración del secret, etc...
variable "tenant_id" {
  type = string
  default = "778aXXXX-XXXX-XXXX-XXXX-XXXXXXXXX6fa"
}
variable "subscription_id" {
  type = string
  default = "34XXXXXX-XXXX-XXXX-XXXX-XXXXXXX16ae"
}
variable "application_password" {
  type = string
  default = "test"
}

# By default setting the password to last for a year
variable "application_password_expiration" {
  type = string
  default = "8760h"
}

# Generación del rol personalizado
# The list of permissions added to the custom role
variable "custom_role_permissions" {
   type = "list"
   default = [
     "Microsoft.Network/networkInterfaces/effectiveNetworkSecurityGroups/action",
     "Microsoft.Network/networkWatchers/securityGroupView/action",
     "Microsoft.Network/networkWatchers/queryFlowLogStatus/action",
     "Microsoft.Network/virtualwans/vpnconfiguration/action",
     "Microsoft.ContainerRegistry/registries/webhooks/getCallbackConfig/action",
     "Microsoft.Web/sites/config/list/action",
     # "Microsoft.Storage/storageAccounts/*", # enable remediation for storage account
   ]
}

###############################
# Initializing the provider
###############################

provider "azuread" {
  subscription_id = var.subscription_id
  tenant_id = var.tenant_id
}
```

```
provider "azurerm" {
  version = "=2.0.0"
  subscription_id = var.subscription_id
  tenant_id = var.tenant_id
  features {}
}

provider "random" {}

############################################################
# Setting up an Application & Service Principal
# Will be shared by all of the onboarded subscriptions
############################################################
resource "random_string" "unique_id" {
  length = 5
  min_lower = 5
  special = false
}

# Creación de la aplicación como recurso de tipo azuread_application, llamado “prisma_cloud_app”
resource "azuread_application" "prisma_cloud_app" {
  name                    = "Prisma Cloud Onboarding ${random_string.unique_id.result}"
  homepage                = "https://prismacloud.io/"
  available_to_other_tenants = true
}

# Creación del Service Principal asociado a la aplicación, llamado “prisma_cloud_sp”
resource "azuread_service_principal" "prisma_cloud_sp" {
  application_id = azuread_application.prisma_cloud_app.application_id
}

# Creación del secreto (api secret) o contaseña de la aplicación
resource "azuread_application_password" "password" {
  application_id      = azuread_application.prisma_cloud_app.id
  value               = var.application_password
  end_date            = timeadd(timestamp(),var.application_password_expiration)
}

############################################################
# Setting up custom roles
# Asignación de los roles (“Reader”, “Reader and Data Access” y “Network Contributor”, así como el rol personalizado de Prisma
############################################################
resource "random_uuid" "custom_role_uuid" {}

resource "azurerm_role_definition" "custom_prisma_role" {
  name        = "prisma-cloud-policy-${random_string.unique_id.result}"
  role_definition_id = random_uuid.custom_role_uuid.result
  scope       = "/subscriptions/${var.subscription_id}"
  description = "Prisma Cloud custom role created via Terraform"
  assignable_scopes = ["/subscriptions/${var.subscription_id}"]
  permissions {
    actions     = var.custom_role_permissions
    not_actions = []
  }
}

resource "azurerm_role_assignment" "assign_custom_prisma_role" {
  scope       = "/subscriptions/${var.subscription_id}"
  principal_id = azuread_service_principal.prisma_cloud_sp.id
  role_definition_id = "${azurerm_role_definition.custom_prisma_role.id}"
}

resource "azurerm_role_assignment" "assign_reader" {
  scope       = "/subscriptions/${var.subscription_id}"
  principal_id = azuread_service_principal.prisma_cloud_sp.id
  role_definition_name = "Reader"
}

resource "azurerm_role_assignment" "assign_reader_data_access" {
  scope       = "/subscriptions/${var.subscription_id}"
```

```
  principal_id = azuread_service_principal.prisma_cloud_sp.id
  role_definition_name = "Reader and Data Access"
}

resource "azurerm_role_assignment" "assign_network_contrib" {
  scope      = "/subscriptions/${var.subscription_id}"
  principal_id = azuread_service_principal.prisma_cloud_sp.id
  role_definition_name = "Network Contributor"
}

# Salida por pantalla de todos los identificadores creados durante el proceso, necesarios para dar de alta la suscripción en Prisma
output "a__subscription_ids" { value = var.subscription_id }
output "b__active_directory_ids" { value = var.tenant_id}
output "c__application_id" { value = azuread_application.prisma_cloud_app.application_id}
output "d__application_key" { value = var.application_password}
output "e__service_principal_id" { value = azuread_service_principal.prisma_cloud_sp.id}
```