Implementación de un *framework* de remediación de vulnerabilidades automatizado en servicios de computación en AWS haciendo uso de infraestructura como código.

Implementation of an automated vulnerability remediation framework in computing services on AWS using infrastructure as code.

Amanda Patricia Urdaneta Ruíz
a.urdaneta@uniandes.edu.co

Jhon Fredy Triana Marin
jf.triana@uniandes.edu.co

Ronald Ferney Paéz
r.paezm@uniandes.edu.co

Maestría en Seguridad de la Información
Departamento de Ingeniería de Sistemas y Computación
Universidad de Los Andes.
Bogotá, Colombia abril 2021

1. Contexto

Para lograr la transformación digital las empresas adoptan nuevas e innovadoras maneras de hacer negocios con base en los avances tecnológicos, donde la adopción de la computación en la nube ha sido uno de los principales recursos al que acuden para apalancar estos objetivos estratégicos que se han propuesto.

La computación en la nube se está convirtiendo rápidamente en la norma y algunas industrias están más adelante que otras. Las industrias que se espera inviertan más en la nube son: Fabricación ($ 19.7 mil millones), Servicios profesionales ($ 18.1 mil millones) y Banca ($ 16.7 mil millones). (Sysgroup, 2018)

Brindar acceso a los datos desde cualquier lugar es la principal razón para la adopción de la nube. El mercado híbrido en la nube se valoró en USD 52.16 mil millones en 2020 y se espera que llegue a USD 145 mil millones en 2026. Además, la nube híbrida conduce a la eficiencia de costos, la seguridad, la agilidad y la escalabilidad. Uno de los beneficios más significativos de una nube híbrida (entre otras cosas) es su apoyo a una fuerza laboral remota. Proporciona a las empresas la flexibilidad de administrar y acomodar fácilmente a sus equipos remotos con acceso a datos descentralizados. (dzone, 2021)

Sin embargo, como todo cambio, esta adopción y migración hacia nuevas tecnologías trae consigo nuevos riesgos como:

- Las configuraciones incorrectas de las características de seguridad en la nube.
- Despliegue de infraestructura en la nube que se crea fuera de la visibilidad de la oficina de TI.
- Poco control sobre quién puede acceder a datos sensibles.
- Robo de datos alojados en la infraestructura de la nube por parte de un actor malintencionado.
- Falta de controles de seguridad consistentes en entornos locales y de múltiples nubes.
- Amenazas y ataques avanzados.
- Propagación lateral de un ataque.

Conociendo esto, en este artículo, encontrará los pasos para implementar un framework de remediación de vulnerabilidades automatizado desplegado en AWS haciendo uso de infraestructura como código.

Palabras Clave: Framework, remediación, vulnerabilidades, servicios de computación, nube, alertamiento, infraestructura como código, IAC, AWS.

2. Planteamiento del problema

Sabemos que, la transformación digital es un término del que se ha estado hablando en diferentes foros durante los últimos 5 años como una estrategia que todo negocio debía asumir para apalancar sus objetivos y metas, buscando mantenerse al día y en competencia con sus aliados comerciales.

Las transformaciones digitales suceden cuando las empresas adoptan nuevas e innovadoras maneras de hacer negocios con base en los avances tecnológicos. La adopción de la computación en la nube ha sido uno de los principales recursos al que acuden para apalancar estos objetivos estratégicos que se han propuesto.

Para entender mejor las tendencias de la nube, en 2018 Refinitiv realizó un estudio entre 250 líderes senior de grandes instituciones financieras y los resultados revelaron que:

- Los beneficios más significativos de la migración a la nube pública son la reducción de costos (para 66% de los participantes), una mayor integridad en los datos (66%), la escalabilidad en la infraestructura (65%) y la facilidad en el procesamiento y análisis de los datos (64%). (Refinitiv, 2018, pág. 4)
- “En los próximos 2 - 3 años, aproximadamente 67% de las empresas financieras utilizarán la nube pública para suplir sus necesidades de datos de mercado”. (Refinitiv, 2018, pág. 4)

El uso de la nube no parece disminuir en el corto o mediano plazo, sobre todo con la coyuntura a la que se ve enfrentado el mundo como lo es una pandemia, que obligó a las empresas a reinventar sus procesos y casos de negocio migrando su infraestructura y servicios *core*.

Sin embargo, como todo cambio, esta adopción y migración hacia nuevas tecnologías trae consigo nuevos riesgos como:

- Las configuraciones incorrectas de las características de seguridad en la nube.
- Despliegue de infraestructura en la nube que se crea fuera de la visibilidad de TI
- Control incompleto sobre quién puede acceder a datos sensibles
- Robo de datos alojados en la infraestructura de la nube por parte de un actor malintencionado
- Falta de controles de seguridad consistentes en entornos locales y de múltiples nubes
- Amenazas y ataques avanzados
- Incapacidad para monitorear sistemas y aplicaciones en busca de vulnerabilidades.
- Propagación lateral de un ataque de una carga de trabajo en la nube a otra

Según estudio de *DyvyCloud*, “las empresas luchan por implementar las configuraciones y controles de seguridad adecuados en la nube, lo que resultó en más de 33 mil millones de registros expuestos solo en 2018 y 2019.” (DivvyCloud, 2018, pág. 6).

Por otro lado, las nubes mal configuradas fueron una de las principales causas de infracciones. Junto con las credenciales robadas o comprometidas, los servidores en la nube mal configurados están vinculados al vector de amenaza inicial más frecuente en las infracciones causadas por ataques maliciosos, siendo estas malas configuraciones referenciadas como el 19% de los casos. (Corporation, July 2020, pág. 34).

El incumplimiento legal debido a configuraciones incorrectas de la nube resultó en un costo promedio por infracción de $4,41 millones USD. (Corporation, July 2020)

También es importante mencionar que las empresas que no habían implementado la automatización de la seguridad registraron un costo total promedio de $6.03 millones USD en violación de datos, más del doble del costo promedio de $2.45 millones para empresas que habían implementado completamente la automatización de la seguridad. (Corporation, July 2020)

Además, hay que mencionar, el modelo de responsabilidad compartida entre el proveedor y el consumidor de la nube. La división de la responsabilidad depende del tipo de la estructura de la nube que se esté usando: IaaS, PaaS o SaaS. Hay una división de la responsabilidad definida por ISO, NIST e incluso la Cloud Security Alliance (CSA) pero, al final, se describe las responsabilidades que tiene el cliente en cuanto al aseguramiento de los datos que ha migrado a la nube.

El cliente es responsable de: La seguridad de los datos a nivel de transporte, almacenamiento y procesamiento, plataforma, aplicaciones, identidad y control de acceso, sistema operativo, configuraciones de Firewall y red. Es decir, es responsable de garantizar la configuración adecuada de los controles de seguridad que apliquen de acuerdo con su negocio y los servicios utilizados.

Por ejemplo, uno de los casos más recientes sobre violación de datos es de una empresa de pruebas de coronavirus en Utah que expuso las identificaciones escaneadas de más de 50,000 pacientes y miles de resultados de pruebas de COVID-19, por mala configuración de los servicios de AWS, dejando público los datos en el *bucket* S3 donde estaban alojados. (comparitech, s.f.), Según el modelo de responsabilidad compartida, el cliente fue 100% responsable por esta violación, dejando exento al proveedor de nube de cargos.

“En el apuro hacia la adopción de IaaS, muchas organizaciones pasan por alto el modelo de responsabilidad compartida para la nube y asumen que la seguridad está completamente a cargo del proveedor de la nube”, dijo Rajiv Gupta, vicepresidente senior de seguridad en la nube de McAfee. (helpnetsecurity, 2019)

En relación con lo mencionado anteriormente, el *Cloud Security Report* del 2020, presentado por Fugue, menciona los principales retos que tienen los ingenieros al manejar manualmente las malas configuraciones en la nube: (fugue, 2020)

- Error humano al faltar configuraciones erróneas críticas (46%)
- Error humano al corregir errores de configuración críticos (45%)
- Dificultades para capacitar a los miembros del equipo sobre la configuración incorrecta (43%)
- Desafíos para contratar suficientes expertos en seguridad en la nube (39%)
- Falsos positivos (31%)

Los problemas mencionados anteriormente podrían llevar al incumplimiento del modelo de responsabilidad compartida como usuarios de la nube, generando brechas de seguridad entre la entidad y sus clientes, que al ser explotadas pueden generar perdidas por costos en cláusulas contractuales, daños en la reputación de las compañías e incluso perdidas de clientes finales.

3. Descripción de la propuesta

3.1. Propuesta de solución

Se requiere automatizar el manejo del ciclo de vida de las vulnerabilidades de seguridad más frecuentes ocasionadas por malas configuraciones dentro del entorno cloud.

En función de lo anterior, se Implementará de un *framework* automatizado para detectar un hallazgo de seguridad, además de realizar la remediación si aplica, enviará la notificación a los encargados de los servicios y llevará seguimiento de lo realizado a través de un gestor de incidentes/casos sobre los servicios de la nube.

Las especificaciones fundamentales a tener en cuenta en la implementación de este *framework son:*

- Conocer el origen y uso de los datos: El desarrollador debe identificar la fuente de los datos y el activo de información a proteger junto con la clasificación del tipo de información que se maneja.
- Definir perfiles de accesos (roles y privilegios sobre la infraestructura): El equipo de seguridad tiene que definir los roles y privilegios que pueden estar asignados a la infraestructura, de modo que un error en una configuración pueda ser remediado por medio del *framework.*
- Conocer las áreas y procesos del negocio que interactúan en el flujo soportado por la infraestructura, para así hacer la parametrización correspondiente.
- Definir los niveles de riesgo a los que se encuentra expuesto el proyecto soportado por la infraestructura y que los clientes están dispuestos a asumir en el desarrollo de su función, y parametrizar los requerimientos específicos que manejará el *framework* de acuerdo con el nivel de riesgo.

A partir de una evaluación de riesgos por el cliente de la nube para determinar que se entienden las consecuencias del uso de cualquier forma de la nube, es importante asegurar que tenga el nivel de protección adecuado, disminuyendo la probabilidad de un mal uso de esta información y que pueda impactar a nivel reputacional, legal y/o financiero.

Buscando herramientas relacionadas a la problemática planteada nos encontramos con un desarrollo de Flatiron Health que fue presentado en el DefCon del 2020 (Flatiron Health, s.f.; Flatiron Health, s.f.) a la cual sería necesario hacer ajustes para cumplir con los requerimientos básicos esperados dentro de cualquier entidad.

Figura 1

Comparativo de herramientas de remediación de vulnerabilidades en la nube

	AWS FRAMEWORK REMEDIATION	***AWS CONFIG***	***PRISMA***
Plataformas Soportadas	SaaS	SaaS	Multicloud
Soporte	N/A	Depende del contrato establecido	Depende del contrato establecido
Licenciamiento	Open Source	Servicio propio de AWS	Licenciamiento con Palo Alto Networks
Precio	Tiene un costo asociado a la infraestructura que se despliega	$0.001 por regla por region	No hay informacion disponible
Capacitación	Documentación	Documentación Webinars Live Online	Documentación

Nota: Se realiza esta tabla comparativa sobre las diferentes soluciones que pueden llegar a cumplir con la misma funcionalidad que el *framework* presentado en este Proyecto, se comparó el tipo de solución, si tiene soporte, si es licenciada, el precio y la capacitación de la misma.

Conocemos las configuraciones inseguras como “una configuración dentro de un programa de computadora que viola una política de configuración o causa un comportamiento no intencional que afecta la postura de seguridad de un sistema.” (NIST, s.f.)

Dentro de los riesgos de aprovisionar recursos tenemos:

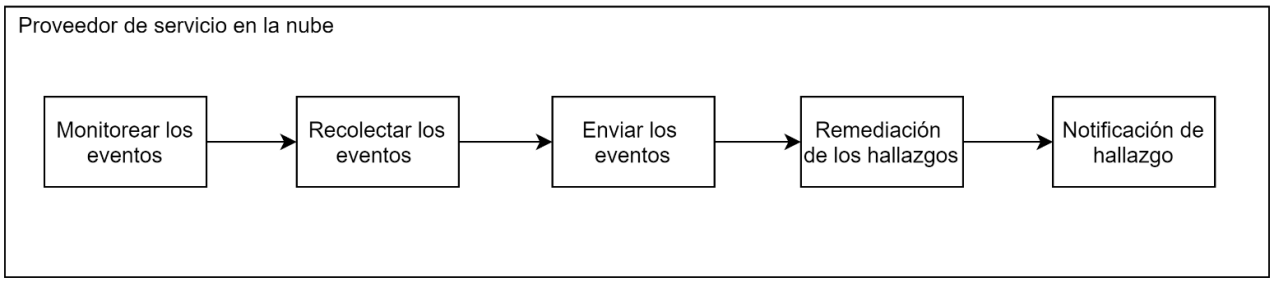
- Puertos de administración abiertos a internet que son vulnerables a ataques de terceros
- Acceso no autorizado a las aplicaciones que pueden convertirse en conexiones maliciosas
- Archivos en la nube con acceso público que pueden ser utilizados por piratas informáticos o ladrones de identidad.
- Ausencia de cifrado de datos en servicios de almacenamiento y bases de datos
- Aprovisionamiento manual de recursos sin visibilidad de creación
- Falta de monitoreo en el entorno *cloud* sobre los comportamientos anómalos y malas configuraciones de los servicios.
- Incremento en los costos de la facturación de uso de la nube.
- Uso indebido de los recursos para ejecutar acciones que no están dentro de las actividades de ADL. (ej: minería de datos)

3.1.1. Diseño de la solución

El desarrollo para la construcción del *framework* según la necesidad comprende los siguientes requerimientos:

Figura 2

Diagrama de flujo



Requerimientos Funcionales:

- El sistema abrirá un tiquet en la herramienta de gestión de casos, para aquellos eventos que sean considerados como posibles vulnerabilidades de seguridad y que no puedan ser remediados automáticamente y necesiten ser validados por un miembro del equipo de seguridad.
- La solución debe contar con archivos de configuración para los servicios definidos donde se podrá identificar cuáles son las excepciones que no se deben tener en cuenta en el *framework*.
- La solución debe remediar automáticamente las vulnerabilidades definidas anteriormente excluyendo las excepciones y que puedan ser corregidas sin intervención humana.
- La solución debe notificar los hallazgos de seguridad que se encuentren a través del *framework* en la herramienta de mensajería de la entidad.
- La solución debe permitir la creación de casos en la herramienta de gestión de incidentes para la gestión y trazabilidad de los incidentes por parte del equipo de seguridad.
- La herramienta estará en la capacidad de detectar eventos relacionados a los servicios de infraestructura desplegados luego de su puesta en marcha en cada una de las cuentas donde se implemente el *framework* sin tener un límite máximo de cuentas miembro.

Requerimientos NO funcionales:

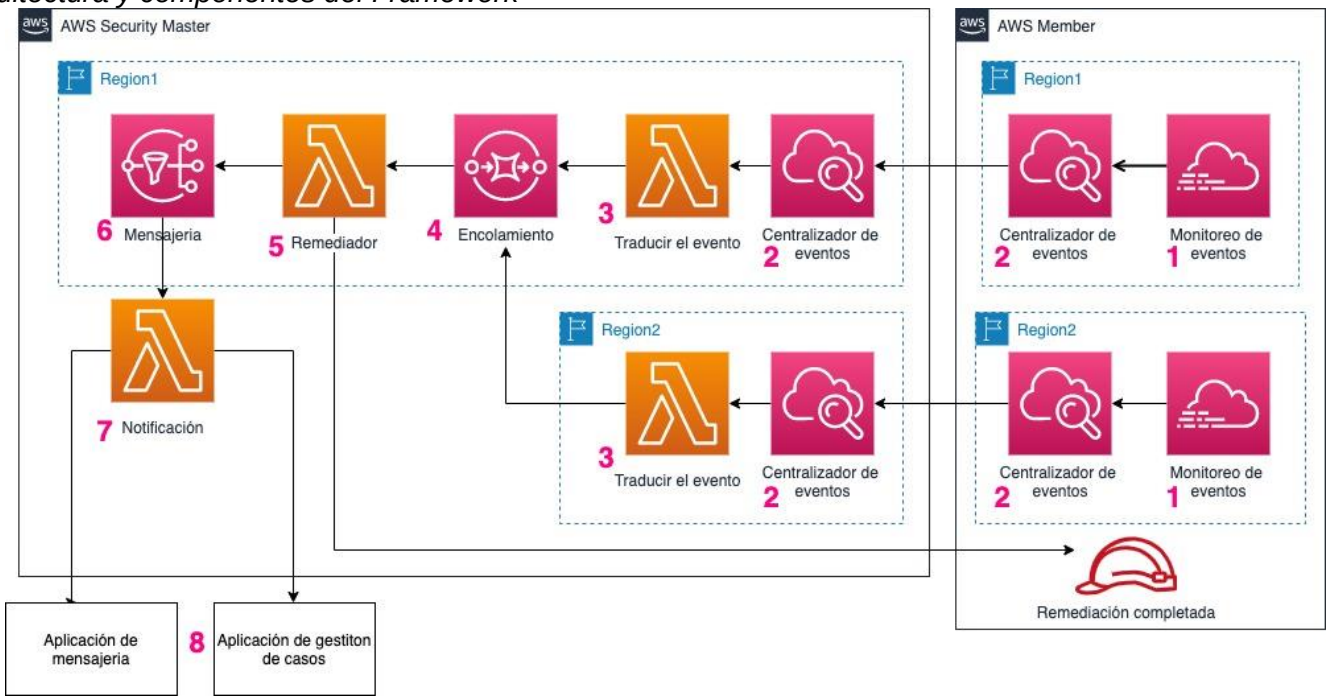
- Usabilidad: Las remediaciones de los hallazgos de seguridad, se ejecutarán de acuerdo a una cola/fila para luego ser ejecutados en orden de aparición, así se evita la presencia de errores por procesar hallazgos al mismo tiempo, Las remediaciones que presenten fallas deberán ser reportadas en un caso abierto en la herramienta de gestión para su posterior seguimiento.
- Concurrencia: La cuenta maestra debe estar en la capacidad de recibir todos los eventos reportados por todas las cuentas miembro.
- Disponibilidad: La disponibilidad de la herramienta deberá ser del 99.04% al año, teniendo como máximo 1 hora de indisponibilidad diaria.
- Seguridad: La comunicación entra la cuenta maestra y las cuentas miembros debe ser por tráfico privado y no se debe exponer información sensible en las herramientas de notificación o de gestión de casos, toda la información debe ser solo informativa.

3.2. Arquitectura y componentes del sistema

La arquitectura y sus componentes constituyen la forma en que se integran las distintas tecnologías para crear el entorno que permita conectar todos los elementos tecnológicos y sus funciones.

Figura 3

Arquitectura y componentes del Framework



1. Servicio para el monitoreo de eventos, el cual identificará los eventos relacionados a hallazgos de seguridad y que posteriormente enviará al servicio centralizador de eventos.
2. Servicio centralizador de eventos, donde se recolectarán todos los eventos de seguridad de los servicios y aplicaciones desplegados sobre cada una de las cuentas de la nube.
3. Función *serverless* desplegado en AWS que se configurará para traducir los eventos recolectados por el monitoreo y centralizador de eventos al formato necesario para la cola.
4. Servicio de mensajería en cola desplegado en AWS que se configurará para mantener en cola los hallazgos encontrados con el fin de darle atención en el orden de aparición a cada uno de ellos.

5. Función *serverless* que se configurará para asumir un rol en la cuenta de destino, auditará y potencialmente remediará los recursos con vulnerabilidades.
6. Función *serverless* que permite enviar la notificación a la herramienta de mensajería de la entidad.
7. Herramienta corporativa de comunicación interna que se integrará con el *framework* para las notificaciones de los hallazgos de seguridad.
8. Herramienta corporativa que se integrará con el *framework* para la creación de casos relacionados a los hallazgos de seguridad.

Componentes necesarios para el funcionamiento de la arquitectura descrita:

- **Software de código abierto:** Necesario para el despliegue de infraestructura como código.
- **Proveedor de servicio de nube:** Aquí se desplegarán los servicios necesarios para este *framework.*
- **Repositorio de código:** Aquí se almacenará el código necesario para el despliegue del *framework*.

4. Implementación

La implementación del *framework* requirió de los siguientes componentes:

- **SQS**: servicio de mensajería en cola desplegado en AWS que se configurará para mantener en cola los hallazgos encontrados con el fin de darle atención en el orden de aparición a cada uno de ellos.
- **Lambda de remediación**: servicio *serverless* desplegado en AWS que se configurará para asumir un rol en la cuenta de destino y auditará, y potencialmente remediará los recursos.
- **Lambda traductora de evento:** servicio *serverless* desplegado en AWS que se configurará para traducir los eventos recolectados por cloud trail y cloudwatch al formato necesario para el SQS.
- **Cloud Trail**: servicio monitoreo de eventos de AWS, el cual identificará los eventos relacionados a hallazgos de seguridad y que posteriormente enviará a cloudwatch.
- **Cloudwatch**: servicio centralizador de eventos de AWS, donde se recolectarán todos los eventos de seguridad de los servicios y aplicaciones desplegados sobre cada una de las cuentas de AWS.
- **Teraform**: software abierto para el despliegue de infraestructura como código.
- **Slack**: herramienta corporativa de comunicación interna que se integrará con el *framework* para las notificaciones de los hallazgos de seguridad.
- **Jira**: herramienta corporativa que se integrará con el *framework* para la creación de casos relacionados a los hallazgos de seguridad.
- **AWS:** Proveedor de servicio de nube donde se desplegarán los servicios necesarios para este *framework.*
- **Github:** repositorio de código.
- **Jenkins:** software abierto para automatización de trabajos en entornos CI/CD.

Para que el *framework* sea más flexible, de acuerdo con las necesidades de cada organización, se diseñó para que su comportamiento pueda ser parametrizado por medio de reglas.

A continuación, se describe de forma general la estructura de estas reglas utilizada para los diferentes servicios de AWS:

- Versión del documento YML.
- Servicio ofrecido por la nube que es objeto de la validación.
- Como hijo del servicio se especifican el o los *tags* que identifican propiedades dentro del servicio y que permiten determinar si se debe o no realizar una validación (True o False), o condiciones que determinan una remediación (puertos específicos para permitir accesos).

Figura 4

Estructura yml con los condicionales de los servicios:

Nota: ejemplo del archivo YML para el servicio denominado *security groups*

Figura 5

Tag dentro de un *security group* que indica los puertos permitidos para acceso desde internet

Requerimientos Técnicos:

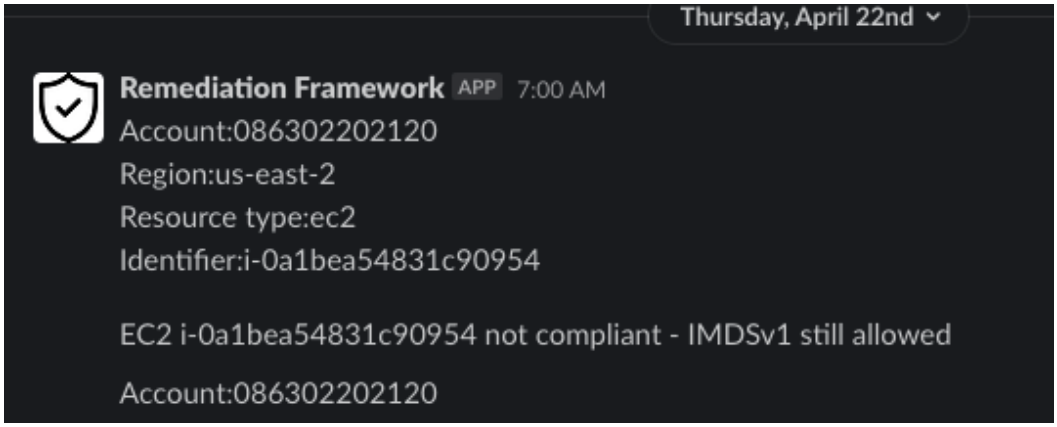
- Diseñar la arquitectura de la infraestructura necesaria para soportar el uso del *framework*:
  Crear un diagrama de arquitectura donde se defina la infraestructura base que se necesita para el funcionamiento del *framework* e integraciones con los componentes adicionales de mensajería y gestión de casos
- Creación de dos módulos de terraform:
  Creación de dos módulos haciendo uso de infraestructura como código para el despliegue necesario de la infraestructura utilizada dentro del *framework*, el cual estaría conformado por:
  - Módulo relacionado a la cuenta principal donde serán desplegadas las funciones de remediación y recibir los eventos relacionados a los hallazgos de seguridad.
  - Módulo para las cuentas miembro donde se consultarán los eventos asociados a los hallazgos de seguridad definidos.
- Desarrollo de remediaciones que no están soportados por el *framework*:
  - Cierre de reglas para los grupos de seguridad que contengan como origen de entrada 0.0.0.0/0 y que el puerto sea diferente de 80 o 443.
  - Cifrado de bases de datos basados en un *tag* específico que determinara si se debe o no cifrar la base de datos.
  - Cifrado de objetos basados en un *tag* específico que determinara si se debe o no cifrar el servicio de almacenamiento S3 de AWS.
- Integración con la aplicación de mensajería empresarial (slack):
  - se debe realizar la integración con el sistema de mensajería empresarial configurando las variables necesarias para la herramienta Slack.
- Integración con la herramienta de gestión de casos Jira:
  - Se debe realizar la integración con el sistema de gestión de casos empresarial, donde se configure una serie de variables (llaves de acceso, tablero, servidor, etc) para consumir el API de Jira.
- Realizar despliegue a través de un entorno CI/CD:
  - Realizar el despliegue de la infraestructura haciendo uso del entorno CI/CD de la entidad, haciendo uso de los módulos creados sobre una cuenta principal y sobre todas las demás cuentas a las que se requiera aplicar el *framework* y que componen la organización de la nube en entidad empezando por las cuentas asociadas a ambientes bajos.

- Realizar un conjunto organizado de pruebas en los ambientes bajos para garantizar que el *framework* está funcionando de acuerdo con lo esperado en cada caso de uso.

Validaciones Funcionales:

Figura 6

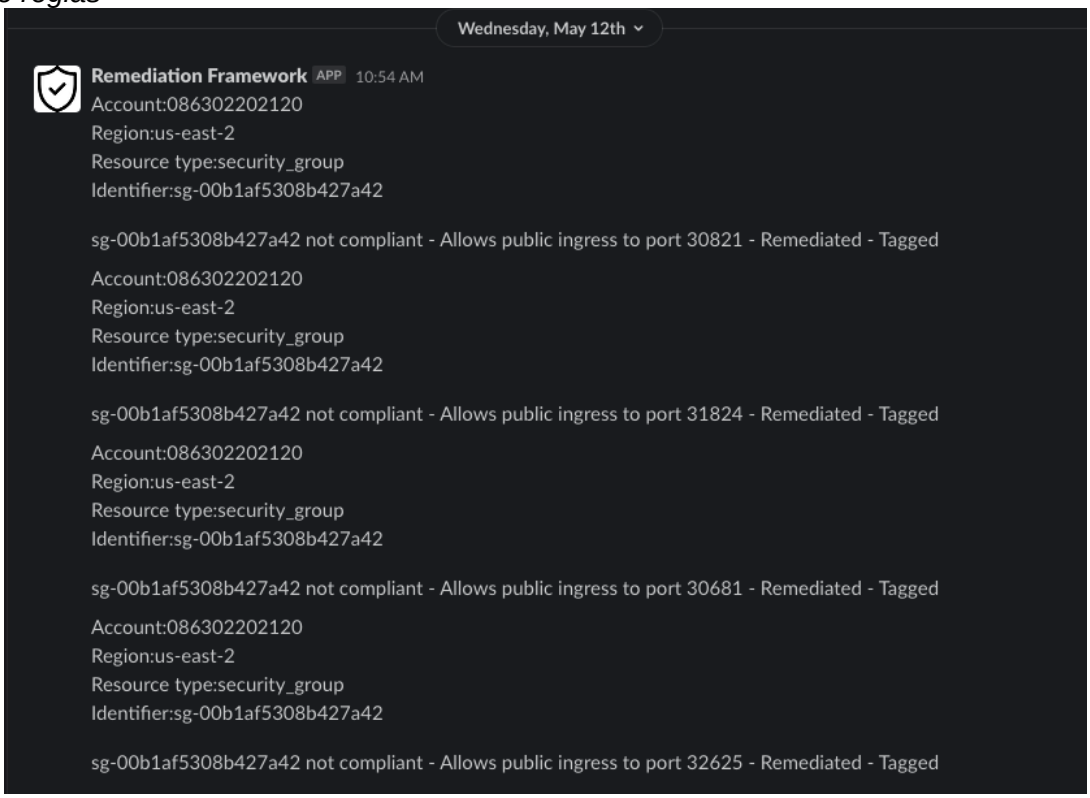
Creación de la instancia



Nota: Instancia creada permitiendo IMDSv1 (Instance Metadata Service Version 1)

Figura 7

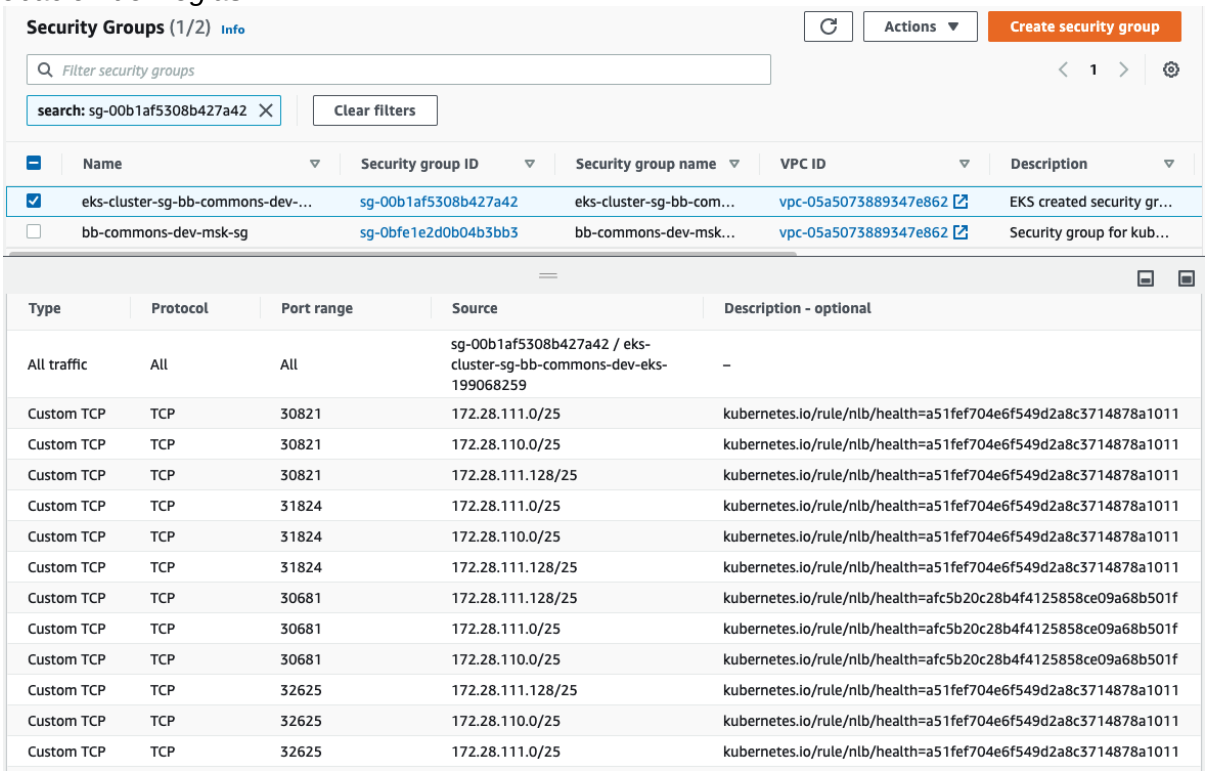
Creación de reglas



Nota: Creación de reglas en los grupos de seguridad con puertos no permitidos.

Figura 8

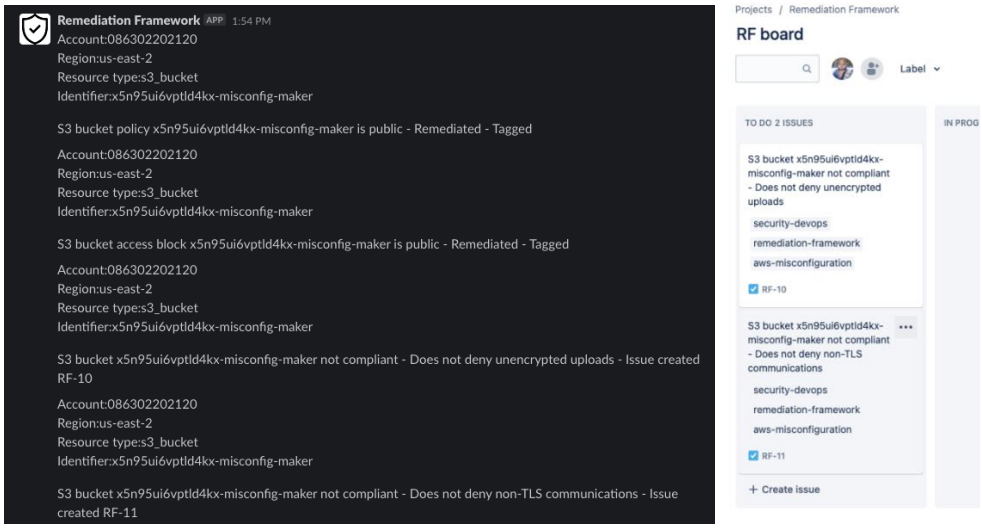
Comprobación de Reglas



Nota: Creación de reglas que eliminan IP's con origen 0.0.0.0/0 y que no son parte de las exclusiones.

Figura 9

Prueba: creación bucket S3 publico



Nota: El *framework* envía las notificaciones de los puntos que se remediaron automáticamente y crea casos en la herramienta de gestión de casos sobre las vulnerabilidades no remediadas automáticamente y necesiten ser validados por un miembro del equipo de seguridad.

5. Conclusiones y recomendaciones

- En los análisis de riesgo realizados en la organización, resalta el riesgo de pérdida de información debido a configuraciones incorrectas, donde si no se remedia o monitorea de manera adecuada se puede materializar el riesgo.
- Se recomienda implementar herramientas que permitan realizar una remediación pertinente sobre las posibles vulnerabilidades que existan en los diferentes ambientes donde esté desplegada la infraestructura y servicios.
- Basado en las pruebas realizadas en ambientes bajos donde se implementó el *framework* de remediación automatizado, éste efectivamente permite tener control y gobierno sobre la infraestructura que se despliega en la nube, garantizando la remediación automática o en su defecto generando las alertas para que el equipo correspondiente pueda realizar las acciones necesarias. Las pruebas del *framework* en ambientes bajos permite identificar si hay fallas e n el código o en la implementación.
- El uso de nuevas tecnologías de desarrollo y despliegue continuo permiten agilizar y dar cumplimiento a las exigencias de los clientes, las áreas de seguridad de las organizaciones deben adaptarse y evolucionar con este tipo de cambios, implementando herramientas que puedan integrarse dentro del ciclo de desarrollo e implementación de la organización, dando agilísimo en el aseguramiento de estos. Solo la automatización de la mayor cantidad de vulnerabilidades que sea posible dentro de la infraestructura en ambientes bajos de la organización le ha permitido al equipo de seguridad bajar sus tiempos de revisión y análisis.
- La integración con la herramienta de comunicación interna de la organización permite estar enterado en tiempo real de las alertas y remediaciones realizadas por el *framework* en una presentación entendible y organizada. Esto permite que la revisión por parte de áreas como auditoría, seguridad de la información o DevSecOps sea más eficiente atacando solo los puntos necesarios.
- Migrar los procesos *core* de la organización a la nube trae muchas ventajas siempre y cuando los componentes que se despliegan en ella estén controlados, monitoreados y con diseños de arquitectura aplicando mejores prácticas. Con este *framework* de remediación, se tendrá una migración controlada, mitigando los riesgos por malas configuraciones que son los más comunes en este tipo de actividades.
- Aunque el proyecto sea implementado por personal de la organización, dentro de los costos del mismo se debe contemplar el valor de los componentes utilizados (Servicios AWS) y las licencias que se lleguen a necesitar en la integración con los componentes oficiales de la compañía.
- Es importante garantizar el correcto funcionamiento del *framework* de acuerdo con lo esperado y al mismo tiempo garantizar que las configuraciones realizadas aplican a la realidad de la empresa, sin causar indisponibilidad en ambientes productivos.

6. Bibliografía

AWS. (s.f.). *AWS*. Obtenido de https://aws.amazon.com/es/lambda/serverless-architectures-learn-more/

Build, M. (s.f.). *Microsoft Build*. Obtenido de Microsoft Build: https://docs.microsoft.com/en-us/azure/devops/learn/what-is-infrastructure-as-code

comparitech. (s.f.). Obtenido de https://www.comparitech.com/blog/information-security/utah-covid-test-center-leak/

Corporation, I. (July 2020). *Ponemon Global Cost of Data Breach Study 2020* . Armonk, NY 10504 .

DivvyCloud. (2018). *Cloud Misconfigurations Report* . NY.

dzone. (2021). Obtenido de https://dzone.com/articles/top-trends-in-cloud-computing-2021

Flatiron Health. (s.f.). *flatironhealth/aws-remediation-framework*. Obtenido de github: https://github.com/flatironhealth/aws-remediation-framework

fugue. (2020). Obtenido de https://www.fugue.co/blog/the-state-of-cloud-security-2020-report-understanding-misconfiguration-risk

helpnetsecurity. (2019). Obtenido de https://www.helpnetsecurity.com/2019/09/25/cloud-misconfiguration-incidents/

ISO27000. (s.f.). *ISO27000*. Obtenido de ISO27000: https://www.iso27000.es/glosario.html

NIST. (s.f.). *NITS*. Obtenido de National Institute of Standard and Technology: https://csrc.nist.gov/glossary/term/misconfiguration

Refinitiv. (2018). *La Nube en America Latina* . New York, NY 10036 : Refinitiv.

SalesForce. (s.f.). *SalesForce*. Obtenido de SalesForce: https://www.salesforce.com/mx/cloud-computing/

Sysgroup. (2018). *sysgroup*. Obtenido de https://www.sysgroup.com/resources/blog/10-cloud-computing-statistics-2018

7. Anexo - Conceptos básicos:

7.1. Vulnerabilidad: Debilidad de un activo o control que puede ser explotada por una o más amenazas. ***(ISO27000, s.f.)***

7.2. Riesgos: El riesgo de seguridad de la información está asociado con el potencial de que las amenazas exploten las vulnerabilidades de un activo de información o grupo de activos de información y, por lo tanto, causen daños a una organización. ***(ISO27000, s.f.)***

7.3. Amenaza: Causa potencial de un incidente no deseado, que puede provocar daños a un sistema o a la organización. ***(ISO27000, s.f.)***

7.4. Computación en la nube: Tecnología que permite acceso remoto a softwares, almacenamiento de archivos y procesamiento de datos por medio de Internet, siendo así, una alternativa a la ejecución en una computadora personal o servidor local. ***(SalesForce, s.f.)***

7.5. Infraestructura como código: Gestión de la infraestructura (redes, máquinas virtuales, balanceadores de carga y topología de conexión) en un modelo descriptivo, utilizando el mismo control de versiones que utiliza el equipo de DevOps para el código fuente. ***(Build, s.f.)***

7.6. Arquitectura/servicio serverless: Una arquitectura sin servidor es una manera de crear y ejecutar aplicaciones y servicios sin tener que administrar infraestructura. Su aplicación continúa ejecutándose en servidores, pero el proveedor de servicio de la nube se encarga de toda la administración de los servidores. ***(AWS, s.f.)***