

Obligatorio

Seguridad en Redes y Datos

Sexto Semestre 2024

Alexis D'Andrea N. ° 230556

Nicolas Martins N. ° 292534

1. Índice

1. Índice	1
2. Introducción.....	3
3. Arquitectura	4
3.1 Componentes	5
3.1.1 Jump-SRV	5
3.1.2 WAF	5
3.1.3 WEB.....	5
3.1.4 Amazon RDS.....	6
3.1.5 SIEM.....	6
3.1.6 ALB (Application Load Balancer).....	6
3.1.7 Auto Scaling Group	6
3.2 Resumen del Flujo de Tráfico	7
4. Acceso Administrativo	8
4.1 Google Authenticator	8
4.1.1 Habilitamos Google Authenticator en SELINUX	9
5. WAF.....	10
5.1 Reglas configuradas.....	11
5.1.1 Rate limiting – Fuerza bruta	11
5.1.2 Otras reglas aplicadas:.....	13
5.1.2.1 Bloqueo de Países Extranjeros.....	14
5.1.2.2 Protección contra XSS (Cross-Site Scripting)	14
5.1.2.3 Bloqueo de Acceso a Archivos Ocultos	14
5.1.2.4 Challenge a Solicitudes Sospechosas.....	14
5.1.2.5 Bloqueo de Rutas de Administración.....	14
5.1.2.6 Redirección tráfico HTTP a HTTPS	15
5.1.2.7 Protección contra bots.....	15
5.1.2.7.1 Bot Fight Mode:	15
5.1.2.7.2 Block AI Bots:.....	15
6. Hardening	16
6.1 Algunas de las medidas de hardening que aplicamos fueron:.....	17
6.1.1 Configuración de permisos y acceso:	17
6.1.2 Mejora en la Seguridad del Servicio SSH:	17
6.1.3 Auditoría del sistema y logging:.....	17

6.1.4	Desactivación de servicios innecesarios:.....	17
6.1.5	Aplicación de políticas de contraseña:	17
6.1.6	Ajuste de Permisos en Archivos de Cron:.....	17
6.1.7	Deshabilitación del Almacenamiento USB:	17
6.1.8	Instalación y Configuración de AIDE para Integridad del Sistema:.....	17
6.1.9	Antes del hardening:.....	18
6.1.10	Después del hardening:.....	19
6.1.11	Score obtenido luego del hardening	24
7.	SIEM.....	25
7.1.1	Visualización de alertas de seguridad con graficas.....	25
7.1.2	Bloqueo de ataques de fuerza bruta	25
7.1.2.1	Visualización de la alerta:	26
7.1.3	Bloqueo por cambio de usuario.....	26
7.1.3.1	Visualización de las alertas:	27
7.1.4	Reiniciar Wazuh Agent frente a cambios.....	27
7.1.4.1	Visualización de la alerta:	28
7.1.5	Detección malware usando Yara	28
7.1.5.1	Configuraciones en el servidor de Wazuh.....	31
7.1.5.2	Emulación de ataque:	32
7.1.5.3	Visualización de las alertas:	34
8.	Autenticación Federada	35
9.	Anexo.....	40
9.1	Instalación Wordpress:.....	40
9.2	Terraform:	43
9.3	Bibliografía:.....	54
9.4	Declaración de autoría	55

2. Introducción

En este proyecto, diseñamos e implementamos una solución integral para garantizar la seguridad, escalabilidad y disponibilidad de una infraestructura en la nube de AWS. Nuestra propuesta aborda los desafíos actuales en seguridad en redes y datos, enfocándonos en prevenir amenazas comunes, gestionar eficientemente el acceso a recursos críticos y asegurar la protección de datos sensibles.

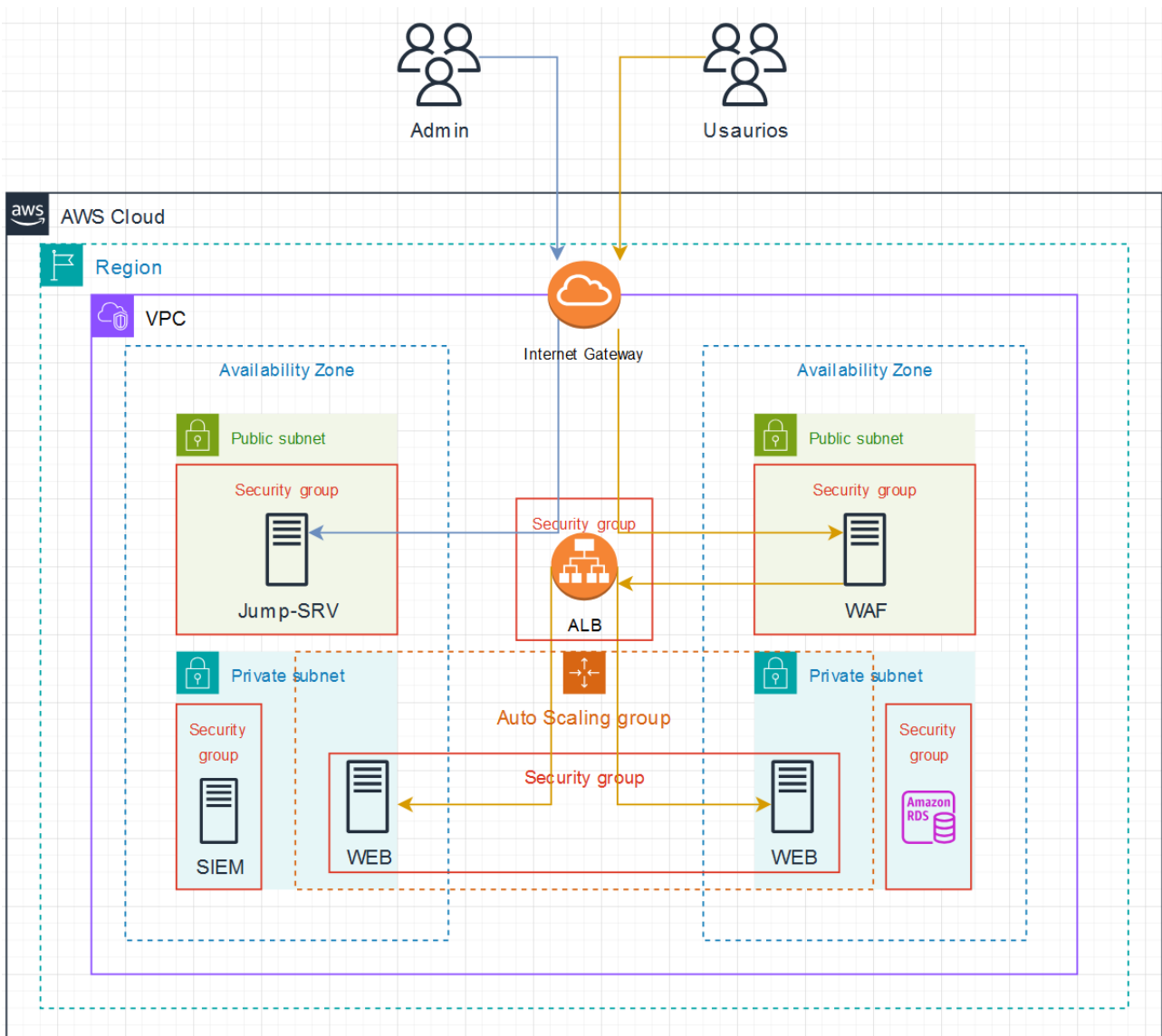
La arquitectura presentada combina automatización y buenas prácticas de seguridad, aprovechando herramientas como Terraform para el despliegue y la configuración de infraestructura, Cloudflare como WAF y Wazuh como solución SIEM. Esto nos permitió optimizar el despliegue, monitorización y protección de los sistemas involucrados, manteniendo un enfoque centralizado en la protección ante ciberataques y el cumplimiento de estándares de seguridad.

No solo se desplegaron soluciones de seguridad, sino que también se tuvo en cuenta la ubicación de cada equipo, teniendo en cuenta su criticidad y exposición, de tal manera que fuera lo más seguro posible para cada componente de la red. Además de aplicarle hardening a cada equipo y en todos los casos admitir el mínimo acceso/privilegio.

A través de esta solución, buscamos no solo mitigar riesgos, sino también establecer una base sólida que facilite futuras expansiones y adaptaciones a nuevas necesidades tecnológicas y del negocio.

3. Arquitectura

Creamos con Terraform la siguiente arquitectura de infraestructura en AWS, diseñada para nuestra aplicación web, teniendo en cuenta la seguridad, escalabilidad y disponibilidad que consideramos necesaria para la misma.



Esto nos permite automatizar las tareas y realizar cambios rápidamente, además de conocer cada componente de la red de forma rápida, ya que todos estarían representados gráficamente y en nuestro código. De la misma forma que las configuraciones.

Beneficios que también contribuyen con la seguridad de la infraestructura, al tener todo inventariado y conocer con facilidad las configuraciones al tenerlas de manera centralizada en el código, además de facilitar el despliegue y cambios que sean necesarios.

3.1 Componentes

A continuación, se describen los distintos componentes utilizados en nuestra solución.

3.1.1 Jump-SRV

Este será nuestro servidor de salto, se encuentra en una subred pública, para que los administradores puedan acceder a esta máquina y luego conectarse para administrar todos los equipos de la red, de manera segura, sin exponer directamente esos recursos a internet.

Esto se logra gracias a que sus security groups que permiten el acceso a la máquina de salto solo desde IPs específicas (las de los administradores) para asegurar que el acceso a cada equipo de la red esté restringido.

3.1.2 WAF

El Web Application Firewall se ubica aquí para filtrar el tráfico entrante y proteger contra ataques de aplicaciones web como SQL Injection, Cross-Site Scripting (XSS), etc, siendo una de las principales barreras frente ataques a nuestra web. Luego el tráfico podrá pasar hacia el Application Load Balancer, permitiendo solo tráfico HTTP/HTTPS desde el WAF.

Nota: En este caso se muestra un WAF interno en la infraestructura, pero en realidad se utilizó Cloudflare, debido a las limitaciones de nuestras cuentas de AWS.

3.1.3 WEB

Los servidores de la aplicación web se encuentran en esta subred privada y están configurados en un Auto Scaling Group para ajustar automáticamente la capacidad en función de la demanda. Estos servidores solo podrán recibir peticiones HTTP/HTTPS desde el load balancer que distribuirá el tráfico entre los dos.



3.1.4 Amazon RDS

La base de datos relacional de Amazon (RDS) está en esta subred privada para almacenar datos de la aplicación de forma segura. Al estar en una subred privada, solo es accesible por los servidores de la aplicación y por el puerto específico de MySQL.

3.1.5 SIEM

El sistema de información y gestión de eventos de seguridad (SIEM) recopilará y analizará eventos de seguridad de todos los equipos en la red, para detectar amenazas y administrarlas.

3.1.6 ALB (Application Load Balancer)

El balanceador de carga de aplicaciones distribuirá el tráfico entre múltiples instancias de servidores web, que se ubican en distintas AZ para asegurar una alta disponibilidad y escalabilidad de la aplicación. Este componente está configurado para recibir tráfico filtrado por el WAF y dirigirlo a los servidores web en la subred privada.

3.1.7 Auto Scaling Group

El Auto Scaling Group administrará el escalado horizontal de los servidores web, permitiendo agregar o eliminar instancias automáticamente en función de la demanda de la aplicación. Esto permitirá que la infraestructura sea flexible y eficiente en costos.

3.2 Resumen del Flujo de Tráfico

Para que los usuarios puedan acceder a la aplicación web, entran por el Internet Gateway, luego pasan al WAF para su inspección. El WAF filtra el tráfico, permitiendo solo solicitudes legítimas hacia el ALB.

El ALB distribuye las solicitudes entre los servidores web en el Auto Scaling Group dentro de las subredes privadas y los servidores web acceden a la base de datos RDS para obtener o almacenar datos.

Mientras esto el SIEM, registra y actúa en base a los eventos y alertas de seguridad generados en la infraestructura.

Los equipos de los administradores son los únicos que pueden acceder de forma segura a través de SSH al Jump-SRV en la subred pública, desde donde pueden realizar tareas de administración en todos los recursos internos.

4. Acceso Administrativo

Para el acceso administrativo configuramos un Jump Server, en el cual solo se podrán acceder desde las IPs de los equipos de los administradores y por SSH, mediante un puerto que no será el tradicional (22) y también se le agregará un banner en el cual se le dará aviso al usuario que acceda al mismo, advirtiéndole que somos los propietarios de este.

El acceso desde las IPs de los equipos de los administradores se logró mediante filtrado de tráfico por parte del security group del servidor, además de filtrado de tráfico, a través del firewall local del servidor.

```
# sudo systemctl start firewalld
# sudo systemctl enable firewalld
# sudo firewall-cmd --permanent --add-rich-rule="rule family='ipv4' source address='<IP y
mascara del equipo administrador>' port port=2222 protocol=tcp accept"
# sudo firewall-cmd --permanent --remove-service=ssh
# sudo firewall-cmd --reload
```

Para el banner, creamos un archivo con su contenido y luego en el archivo de configuración de SSH agregamos el path del mismo. En esta misma configuración, cambiamos el puerto de SSH por el 2222.

```
# sudo echo "Advertencia: Acceso no autorizado. Este sistema es propiedad del obligatorio" >
/home/ec2-user/banner.txt
# sudo sed -i 's|#Banner none|Banner /home/ec2-user/banner.txt|' /etc/ssh/sshd_config
# sudo sed -i 's|#Port 22/Port 2222/' /etc/ssh/sshd_config
# sudo systemctl restart sshd
```

El parámetro “PasswordAuthentication no”, que inhabilita a las conexiones a través de autenticación mediante contraseña, ya viene por defecto seteado en no, por lo cual no fue necesario configurarlo. Y para la generación de las claves SSH se optó por utilizar “vockey” que son las claves que vienen cargadas por defecto en AWS, pero en caso de ponerlo en producción, se optaría por ponerle diferentes claves SSH en los equipos.

4.1 Google Authenticator

Para seguir reforzando la seguridad de nuestro Jump Server, implementamos doble factor de autenticación para todos los usuarios.

```
# sudo yum install epel-release -y
# sudo dnf install qrencode google-authenticator -y
# sudo dnf install policycoreutils-python-utils -y
# google-authenticator
# sudo vi /etc/pam.d/sshd
    auth required pam_google_authenticator.so
```

```

    auth required pam_permit.so
# sudo vi /etc/ssh/sshd_config
    ChallengeResponseAuthentication yes
    AuthenticationMethods publickey,password publickey,keyboard-interactive
# sudo service sshd restart


```

4.1.1 Habilitamos Google Authenticator en SELINUX

```

# sudo semanage fcontext -a -t ssh_home_t "/root/.google_authenticator"
# sudo restorecon -v /root/.google_authenticator
# sudo ausearch -c 'sshd' --raw | audit2allow -M google_authenticator_policy
# sudo semodule -i google_authenticator_policy.pp

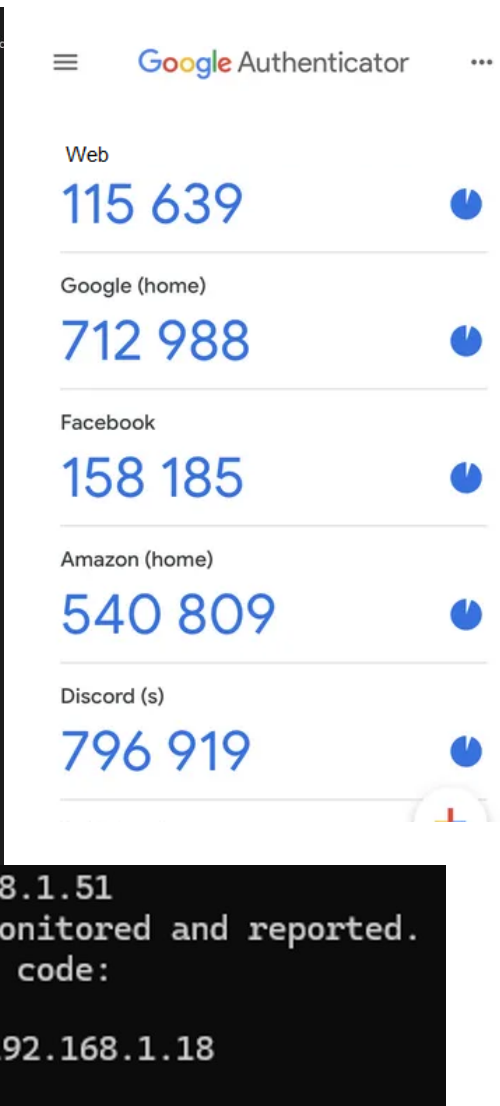
```



```

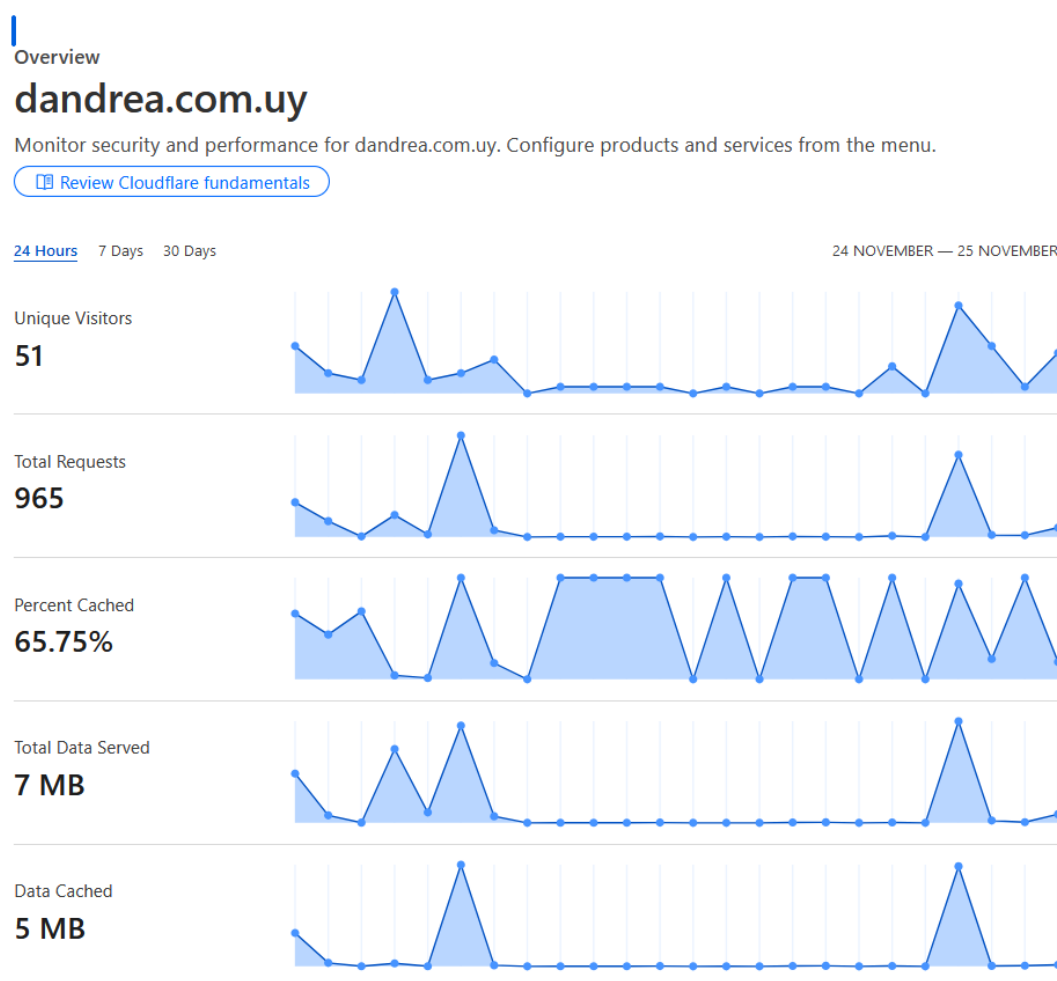
Do you want authentication tokens to be time-based (y/n) y
Warning: pasting the following URL into your browser exposes the OTP secret to Google:
https://www.google.com/chart?chs=200x200&chld=M!0&cht=qr&chl=otpauth://totp/infraestructura@web.obligatorio.srd%3Fsecret=M5MDWMCA4BXZJFZUSCGMY7ZIMY%26issuer%3Dweb.obligatorio.srd
Your new secret key is: M5MDWMCA4BXZJFZUSCGMY7ZIMY
Enter code from app (-1 to skip): 272918
Code incorrect (correct code 410988). Try again.
Enter code from app (-1 to skip): 734633
Code confirmed
Your emergency scratch codes are:
10012824
30323618
54324371
65433467
48392634
PS C:\Users\Ale> ssh infraestructura@192.168.1.51
Authorized uses only. All activity may be monitored and reported.
(infraestructura@192.168.1.51) Verification code:
(infraestructura@192.168.1.51) Password:
Last login: Mon Nov 25 17:14:32 2024 from 192.168.1.18
[infraestructura@web ~]$

```



5. WAF

Siguiendo las implementaciones de seguridad para la aplicación web, decidimos utilizar Cloudflare como nuestro Web Application Firewall (WAF), ya que no pudimos optar por instalar otro WAF, como se muestra en el diagrama de la arquitectura. Esta elección también se basó en las características robustas de Cloudflare, su facilidad de configuración y las ventajas que ofrece al proteger aplicaciones web contra amenazas comunes.



Sus capacidades avanzadas y su infraestructura global, nos permite mitigar ataques sin afectar la disponibilidad del sitio.

Cloudflare nos permite mitigar ataques frecuentes como inyecciones SQL, Cross-Site Scripting (XSS) y ataques de fuerza bruta dirigidos a la página de inicio de sesión de nuestra aplicación. Estas funcionalidades nos resultaron esenciales para garantizar la seguridad de nuestro sitio web.

5.1 Reglas configuradas

5.1.1 Rate limiting – Fuerza bruta

La función de Rate Limiting de Cloudflare nos permitió implementar controles específicos para limitar el número de solicitudes, mitigando así intentos de abuso y ataques automatizados.

Rate limiting rules

Protect your website and API from malicious traffic with rate limiting rules. Configure mitigation criteria and actions for better security.

You have used **1 out of 1** available Rate Limiting Rules.

<div><div>+ Create rule</div><div><input type="text" value="Search..."/> <div>Search</div></div><div><div>Show filters</div></div></div>						
Order	Action	Name	CSR ⓘ	Activity last 24hr	Enabled	
1	Block	Fuerza bruta - Login URI Path	-	<div></div> 58	<input checked="" type="checkbox"/>	

Rule name (required)

Fuerza bruta - Login

Give your rule a descriptive name.

Field	Operator	Value		
URI Path	contains	/login	And	Or

e.g. /content

Expression Preview

[Edit expression](#)

(http.request.uri.path contains "/login")

With the same characteristics...

IP

When rate exceeds...

Requests (required)	Period (required)
10	10 seconds

Then take action...

Choose action

Block

Blocks matching requests and stops evaluating other rules

For duration...

Duration (required)

10 seconds

Configuramos una regla de Rate Limiting con el fin de mitigar ataques de fuerza bruta dirigidos a la página de inicio de sesión.

La regla se activa si una IP realiza más de 10 solicitudes en un período de 10 segundos a cualquier ruta que contenga /login.

Si se excede este límite, la acción configurada es "Block", lo que significa que Cloudflare bloqueará automáticamente la IP que intenta acceder, evitando más solicitudes.

La IP quedará bloqueada durante 10 segundos antes de poder volver a intentar (el tiempo se puede extender si cambiamos nuestro plan de suscripción de Cloudflare).

```
PS C:\WINDOWS\system32> for ($i = 1; $i -le 30; $i++) {  
    Write-Host "Intento $i"  
    try {  
        $response = Invoke-WebRequest -Uri "https://mail.dandrea.com.uy:2096/login" -Method POST -UseBasicParsing  
        Write-Host "Código de estado: $($response.StatusCode)"  
    } catch {  
        Write-Host "Error en la solicitud: $($_.Exception.Message)"  
    }  
}  
Intento 1  
Error en la solicitud: Error en el servidor remoto: (401) No autorizado.  
Intento 2  
Error en la solicitud: Error en el servidor remoto: (401) No autorizado.  
Intento 3  
Error en la solicitud: Error en el servidor remoto: (401) No autorizado.  
Intento 4  
Error en la solicitud: Error en el servidor remoto: (401) No autorizado.  
Intento 5  
Error en la solicitud: Error en el servidor remoto: (401) No autorizado.  
Intento 6  
Error en la solicitud: Error en el servidor remoto: (401) No autorizado.  
Intento 7  
Error en la solicitud: Error en el servidor remoto: (401) No autorizado.  
Intento 8  
Error en la solicitud: Error en el servidor remoto: (401) No autorizado.  
Intento 9  
Error en la solicitud: Error en el servidor remoto: (401) No autorizado.  
Intento 10  
Error en la solicitud: Error en el servidor remoto: (429) Too Many Requests.  
Intento 11  
Error en la solicitud: Error en el servidor remoto: (429) Too Many Requests.  
Intento 12  
Error en la solicitud: Error en el servidor remoto: (429) Too Many Requests.  
Intento 13
```

Para verificar la efectividad de la regla configurada, utilizamos un script en PowerShell que simuló múltiples intentos de acceso a la página de inicio de sesión. El objetivo de esta prueba fue reproducir un ataque de fuerza bruta y confirmar que el WAF estaba aplicando correctamente la regla.

Firewall Events [Create custom rule](#)

[Add filter](#) Previous 30 minutes

Sampled logs [Export](#) [Edit columns](#)

Date	Action taken	Country	IP address	Service
Nov 16, 2024 2:05:30 AM	Block	Uruguay	2800:a4:1ebce800:5d27:768d:f78e:5d2e	Rate limiting rules

Matched service [Export event JSON](#)

Service	Rate limiting rules	Ruleset	default
Action taken	Block	Rule	Fuerza bruta - Login






Request details

Ray ID	8e34f246a8241e8c	HTTP Version	HTTP/1.1
IP address	2800:a4:1ebce800:5d27:768d:f78e:5d2e	Method	POST
ASN	AS6057 Administracion Nacional de Telecomunicaciones	Host	mail.dandrea.com.uy:2096
Country	Uruguay	Path	/login
User agent	Mozilla/5.0 (Windows NT; Windows NT 10.0; es-UY) WindowsPowerShell/5.1.22621.4249	Query string	Empty query string

Date	Action taken	Country	IP address	Service
Nov 16, 2024 2:05:30 AM	Block	Uruguay	2800:a4:1ebce800:5d27:768d:f78e:5d2e	Rate limiting rules
Nov 16, 2024 2:05:30 AM	Block	Uruguay	2800:a4:1ebce800:5d27:768d:f78e:5d2e	Rate limiting rules
Nov 16, 2024 2:05:29 AM	Block	Uruguay	2800:a4:1ebce800:5d27:768d:f78e:5d2e	Rate limiting rules
Nov 16, 2024 2:05:29 AM	Block	Uruguay	2800:a4:1ebce800:5d27:768d:f78e:5d2e	Rate limiting rules

Analizamos los eventos en el panel de Cloudflare y observamos que las IPs que excedían el límite de solicitudes configurado eran bloqueadas. Esto demostró que la regla estaba funcionando según lo esperado y que estaba mitigando eficazmente los intentos de abuso.

5.1.2 Otras reglas aplicadas:

Order	Action	Name	CSR ①	Activity last 24hr	Enabled
1	Block	Bloqueo paises extranjeros Country	-	 38	<input checked="" type="checkbox"/>
2	Block	Protección contra XSS URI Query String	-	 27	<input checked="" type="checkbox"/>
3	Block	Bloqueo acceso archivos ocultos URI Path, Threat Score	-	 26	<input checked="" type="checkbox"/>
4	Managed Challenge	Challenge solicitudes sospechosas HTTP Version, User Agent	0%	 0	<input checked="" type="checkbox"/>
5	Block	Bloqueo rutas admin URI Path	-	 27	<input checked="" type="checkbox"/>

5.1.2.1 *Bloqueo de Países Extranjeros*

Esta regla bloquea todo el tráfico que provenga de países distintos a Uruguay. Utilizando la geolocalización IP, se identifica el país de origen de cada solicitud. Al bloquear accesos desde ubicaciones que no forman parte de la audiencia esperada, reducimos significativamente la exposición a ataques automatizados y exploraciones de vulnerabilidades provenientes de fuentes internacionales.

5.1.2.2 *Protección contra XSS (Cross-Site Scripting)*

La regla de protección contra XSS está configurada para bloquear intentos de inyección de código JavaScript malicioso en la URL de la solicitud. Se inspeccionan los parámetros de la consulta (URI, Query y String) en busca de patrones sospechosos como `<script>`, `javascript:`, `onerror=`, y `onload=`. Al bloquear estas solicitudes, prevenimos posibles ataques XSS que podrían comprometer la seguridad de la página web al querer ejecutar scripts maliciosos en el navegador.

5.1.2.3 *Bloqueo de Acceso a Archivos Ocultos*

Esta regla bloquea solicitudes dirigidas a archivos y directorios ocultos, como `.git` y `.env`, que suelen contener información sensible. Además, detecta intentos de recorrido de directorios (`../`) y solicitudes codificadas (`%2e`) para eludir la seguridad. También bloquea solicitudes con un puntaje de amenaza (Threat Score) alto, indicando comportamiento sospechoso. Esta medida protege contra accesos no autorizados a archivos críticos que podrían exponer datos confidenciales.


5.1.2.4 *Challenge a Solicitudes Sospechosas*

Esta regla presenta un desafío gestionado (CAPTCHA) a solicitudes que utilizan la versión antigua de HTTP (HTTP/1.0) o que tienen un User-Agent vacío o desconocido. Estas características son típicas de bots y herramientas automatizadas. Al aplicar un desafío, se valida la legitimidad del cliente antes de permitir el acceso, ayudando a reducir el tráfico malicioso sin bloquear directamente a usuarios legítimos.

5.1.2.5 *Bloqueo de Rutas de Administración*

La regla de bloqueo de rutas de administración a todas las IPs que no sean las de administración, protegiendo secciones críticas del sitio web, como `/admin`, `/wp-admin`, y archivos de configuración como `config.php`. Estas rutas suelen ser objetivos de ataques automatizados que intentan explotar vulnerabilidades o acceder a paneles de control. Al bloquear el acceso a estas rutas, evitamos intentos de fuerza bruta y acceso no autorizado, fortaleciendo la seguridad general de la aplicación.

5.1.2.6 Redirección tráfico HTTP a HTTPS

Order	Name	Enabled
1	Redireccion HTTP a HTTPS URI Full	

Esta regla configurada en Cloudflare establece una redirección automática de todo el tráfico que llegue a través de HTTP hacia HTTPS. Su propósito es asegurar que todas las solicitudes sean servidas de forma segura utilizando el protocolo HTTPS, lo que mejora la seguridad del sitio al cifrar las comunicaciones entre el servidor y los usuarios.

5.1.2.7 Protección contra bots

Security

Bots

Identify and mitigate automated traffic to protect your domain from bad bots.

[Bots documentation](#)

Bot Fight Mode

Challenge requests that match patterns of known bots, before they access your site. This feature includes [JavaScript Detections](#).

Note: Other security products cannot be used to skip Bot Fight Mode. [Learn more](#)



Block AI Bots New

Block bots from scraping your content for AI applications like model training. [Learn more](#)

Note: Blocking AI Bots will also block verified AI bots.



5.1.2.7.1 Bot Fight Mode:

Esta opción activa en Cloudflare, detecta y desafía bots maliciosos utilizando técnicas como detecciones basadas en JavaScript. Presenta challenges (como CAPTCHA) a solicitudes sospechosas antes de que accedan al sitio, bloqueando bots automatizados que intentan hacer scraping, explorar vulnerabilidades o generar tráfico innecesario. Esto ayuda a reducir la carga del servidor y a proteger contra ataques automatizados, sin afectar a los usuarios legítimos.

5.1.2.7.2 Block AI Bots:

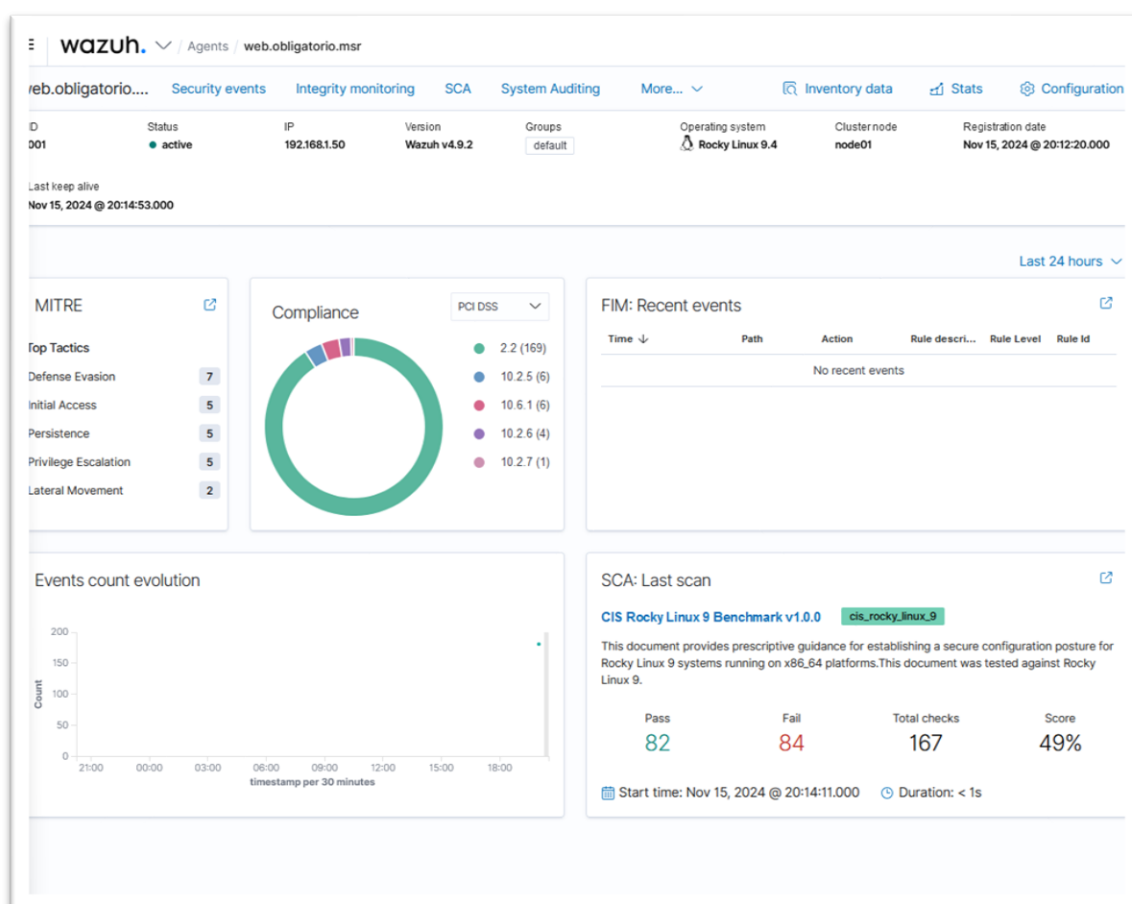
Esta configuración bloquea bots que intentan extraer datos del sitio para aplicaciones de inteligencia artificial, como scraping para entrenamiento de modelos. Al habilitarla, se impide el acceso tanto de bots no verificados como de bots verificados utilizados por empresas de IA, protegiendo el contenido y evitando el uso no autorizado de recursos del sitio para tareas automatizadas de recolección de datos.

6. Hardening

Es el proceso de aplicar configuraciones y medidas de seguridad para reducir vulnerabilidades en un sistema, reforzando su protección contra ataques. Involucra la eliminación de servicios innecesarios, ajuste de permisos, configuración de políticas seguras, y el uso de herramientas de auditoría para asegurar que el sistema cumpla con las mejores prácticas de seguridad.

Las medidas que aplicamos a nuestros servidores para mejorar la seguridad de la infraestructura fueron siguiendo las recomendaciones establecidas en el CIS Rocky Linux 9 Benchmark v1.0.0, utilizando como herramienta de auditoría y monitoreo a Wazuh.

Realizamos un escaneo de nuestros servidores sin aplicar medidas de hardening, lo que resultó en un 49% de cumplimiento de las buenas prácticas de seguridad según el benchmark de CIS. Este resultado inicial reflejaba un estado base con varias configuraciones predeterminadas que podían ser mejoradas para mitigar riesgos y vulnerabilidades.



6.1 Algunas de las medidas de hardening que aplicamos fueron:

6.1.1 Configuración de permisos y acceso:

Ajustamos los permisos de archivos y directorios críticos, como `/etc/passwd`, `/etc/shadow`, y `/etc/hosts`, asegurando que solo usuarios privilegiados puedan modificarlos.

6.1.2 Mejora en la Seguridad del Servicio SSH:

Deshabilitamos el acceso SSH con usuario root y configuramos políticas de autenticación más estrictas, incluyendo el uso obligatorio de claves SSH y la desactivación de métodos de autenticación inseguros.

6.1.3 Auditoría del sistema y logging:

Activamos servicios de auditoría (auditd) para registrar eventos importantes del sistema y configuramos reglas para monitorear cambios en archivos críticos.

6.1.4 Desactivación de servicios innecesarios:

Identificamos y deshabilitamos servicios que no son necesarios para el funcionamiento de nuestros servidores, reduciendo así la superficie de ataque.

6.1.5 Aplicación de políticas de contraseña:

Configuramos políticas de contraseñas más estrictas, requiriendo un mínimo de caracteres, complejidad y expiración regular para mejorar la protección de las cuentas de usuario.

6.1.6 Ajuste de Permisos en Archivos de Cron:

Una de las recomendaciones del CIS Benchmark indicaba que los archivos de cron (`/etc/crontab`, `/etc/cron.hourly/`, `/etc/cron.daily/`, etc.) no contaban con los permisos adecuados, lo que representaba un riesgo de modificación no autorizada. Para mitigar este riesgo, ajustamos los permisos de estos archivos para que solo el usuario root tenga acceso de escritura.

6.1.7 Deshabilitación del Almacenamiento USB:

Para prevenir posibles exfiltraciones de datos y evitar el uso no autorizado de dispositivos de almacenamiento externos, deshabilitamos el soporte para almacenamiento USB.

6.1.8 Instalación y Configuración de AIDE para Integridad del Sistema:

Instalamos AIDE (Advanced Intrusion Detection Environment) para monitorear la integridad del sistema de archivos. Configuramos el servicio para realizar verificaciones periódicas y detectar cambios no autorizados.

6.1.9 Antes del hardening:

31530	Ensure AIDE is installed.	Command: rpm -q aide	● Failed
31531	Ensure filesystem integrity is regularly checked.	Command: systemctl is-enabled aidecheck.service	● Failed
31532	Ensure cryptographic mechanisms are used to ...	File: /etc/aide/aide.conf	● Failed
31533	Ensure core dump storage is disabled.	File: /etc/systemd/coredump.conf	● Failed
31534	Ensure core dump backtraces are disabled.	File: /etc/systemd/coredump.conf	● Failed
31544	Ensure local login warning banner is configure...	File: /etc/issue	● Failed
31545	Ensure remote login warning banner is configu...	File: /etc/issue.net	● Failed
31555	Ensure chrony is configured.	File: /etc/chrony.conf,/etc/sysconfig/chronyd	● Failed
31614	Ensure permissions on /etc/crontab are configured.	Command: stat -L /etc/crontab	● Failed
31615	Ensure permissions on /etc/cron.hourly are configured.	Command: stat -L /etc/cron.hourly	● Failed
31616	Ensure permissions on /etc/cron.daily are configured.	Command: stat -L /etc/cron.daily	● Failed
31617	Ensure permissions on /etc/cron.weekly are configured.	Command: stat -L /etc/cron.weekly	● Failed
31618	Ensure permissions on /etc/cron.monthly are configured.	Command: stat -L /etc/cron.monthly	● Failed
31619	Ensure permissions on /etc/cron.d are configured.	Command: stat -L /etc/cron.d	● Failed
31624	Ensure SSH root login is disabled.	Command: sshd -T -C user=root	● Failed
31629	Ensure SSH X11 forwarding is disabled.	Command: sshd -T -C user=root	● Failed
31630	Ensure SSH AllowTcpForwarding is disabled.	Command: sshd -T -C user=root	● Failed
31632	Ensure SSH warning banner is configured.	Command: sshd -T -C user=root	● Failed
31633	Ensure SSH MaxAuthTries is set to 4 or less.	Command: sshd -T -C user=root	● Failed
31634	Ensure SSH MaxStartups is configured.	Command: sshd -T -C user=root	● Failed
31636	Ensure SSH LoginGraceTime is set to one minute or le...	Command: sshd -T -C user=root	● Failed

6.1.10 Después del hardening:

31530	Ensure AIDE is installed.	Command: rpm -q aide	● Passed	✓
31531	Ensure filesystem integrity is regularly checked.	Command: systemctl is-enabled aidecheck.service,systemctl is-enabled aidecheck.timer,systemctl status aidecheck.timer	● Passed	✓
31533	Ensure core dump storage is disabled.	File: /etc/systemd/coredump.conf	● Passed	✓
31534	Ensure core dump backtraces are disabled.	File: /etc/systemd/coredump.conf	● Passed	✓
31535	Ensure SELinux is installed.	Command: rpm -q libselinux	● Passed	✓
31536	Ensure SELinux is not disabled in bootloader configur...	File: /boot/grub2/grubenv,/boot/grub2/grub.cfg	● Passed	✓
31537	Ensure SELinux policy is configured.	File: /etc/selinux/config	● Passed	✓
31538	Ensure the SELinux mode is not disabled.	File: /etc/selinux/config	● Passed	✓
31539	Ensure the SELinux mode is enforcing.	File: /etc/selinux/config	● Passed	✓
31541	Ensure SETroubleshoot is not installed.	Command: rpm -qa setroubleshoot	● Passed	✓
31542	Ensure the MCS Translation Service (mcstrans) is not...	Command: rpm -qa mcstrans	● Passed	✓
31543	Ensure message of the day is configured properly.	File: /etc/motd	● Passed	✓
31544	Ensure local login warning banner is configured prope...	File: /etc/issue	● Passed	✓
31545	Ensure remote login warning banner is configured pro...	File: /etc/issue.net	● Passed	✓
31546	Ensure permissions on /etc/motd are configured.	Command: stat /etc/motd	● Passed	✓
31547	Ensure permissions on /etc/issue are configured.	Command: stat /etc/issue	● Passed	✓
31548	Ensure permissions on /etc/issue.net are configured.	Command: stat /etc/issue.net	● Passed	✓
31549	Ensure GNOME Display Manager is removed.	Command: rpm -q gdm	● Passed	✓
31551	Ensure XDCMP is not enabled.	File: /etc/gdm3	● Passed	✓
31553	Ensure system-wide crypto policy is not legacy.	File: /etc/crypto-policies/config	● Passed	✓
31554	Ensure time synchronization is in use.	Command: rpm -q chrony	● Passed	✓
31555	Ensure chrony is configured.	File: /etc/chrony.conf,/etc/sysconfig/chronyd	● Passed	✓
31556	Ensure xorg-x11-server-common is not installed.	Command: rpm -q xorg-x11-server-common	● Passed	✓

31557	Ensure Avahi Server is not installed.	Command: rpm -q avahi-autoipd,rpm -q avahi	● Passed	✓
31558	Ensure CUPS is not installed.	Command: rpm -q cups	● Passed	✓
31559	Ensure DHCP Server is not installed.	Command: rpm -q dhcp-server	● Passed	✓
31560	Ensure DNS Server is not installed.	Command: rpm -q bind	● Passed	✓
31561	Ensure VSFTP Server is not installed.	Command: rpm -q vsftpd	● Passed	✓
31562	Ensure TFTP Server is not installed.	Command: rpm -q tftp-server	● Passed	✓
31563	Ensure a web server is not installed.	Command: rpm -q nginx,rpm -q httpd	● Passed	✓
31564	Ensure IMAP and POP3 server is not installed.	Command: rpm -q dovecot,rpm -q cyrus-imapd	● Passed	✓
31565	Ensure Samba is not installed.	Command: rpm -q samba	● Passed	✓
31566	Ensure HTTP Proxy Server is not installed.	Command: rpm -q squid	● Passed	✓
31567	Ensure net-snmp is not installed.	Command: rpm -q net-snmp	● Passed	✓
31568	Ensure telnet-server is not installed.	Command: rpm -q telnet-server	● Passed	✓
31569	Ensure dnsmasq is not installed.	Command: rpm -q dnsmasq	● Passed	✓
31570	Ensure mail transfer agent is configured for local-only...	Command: ss -lntu	● Passed	✓
31571	Ensure nfs-utils is not installed or the nfs-server servi...	Command: rpm -q nfs-utils	● Passed	✓

31572	Ensure rpcbind is not installed or the rpcbind services...	Command: rpm -q rpcbind	● Passed	▼
31573	Ensure rsync-daemon is not installed or the rsyncd se...	Command: rpm -q rsync	● Passed	▼
31574	Ensure telnet client is not installed.	Command: rpm -q telnet	● Passed	▼
31575	Ensure LDAP client is not installed.	Command: rpm -q openldap-clients	● Passed	▼
31576	Ensure TFTP client is not installed.	Command: rpm -q tftp	● Passed	▼
31577	Ensure FTP client is not installed.	Command: rpm -q ftp	● Passed	▼
31579	Ensure nftables is installed.	Command: rpm -q nftables	● Passed	▼
31581	Ensure at least one nftables table exists.	Command: nft list tables	● Passed	▼
31582	Ensure nftables base chains exist.	Command: rpm -q nftables,nft list ruleset	● Passed	▼
31584	Ensure auditd is installed.	Command: rpm -q audit	● Passed	▼
31585	Ensure auditing for processes that start prior to audit...	Command: grubby --info=ALL	● Passed	▼
31586	Ensure audit_backlog_limit is sufficient.	Command: grubby --info=ALL	● Passed	▼
31587	Ensure auditd service is enabled.	Command: systemctl is-enabled auditd	● Passed	▼
31588	Ensure audit log storage size is configured.	File: /etc/audit/auditd.conf	● Passed	▼
31589	Ensure audit logs are not automatically deleted.	File: /etc/audit/auditd.conf	● Passed	▼
31590	Ensure system is disabled when audit logs are full.	File: /etc/audit/auditd.conf	● Passed	▼
31591	Ensure changes to system administration scope (sud...	Directory: /etc/audit/rules.d/	● Passed	▼
31592	Ensure the running and on disk configuration is the sa...	Command: augenrules --check	● Passed	▼
31593	Ensure only authorized groups are assigned ownershi...	File: /etc/audit/auditd.conf	● Passed	▼
31594	Ensure audit configuration files are 640 or more restri...	Command: find /etc/audit/ -type f \(-name '*.conf' -o -name '*.rules' \) -exec stat -Lc "%n %a" {} +	● Passed	▼
31595	Ensure audit configuration files are owned by root.	Command: find /etc/audit/ -type f \(-name '*.rules' -o -name '*.conf' \) -exec stat -Lc "%U" {} +	● Passed	▼

31596	Ensure audit configuration files belong to group root.	Command: find /etc/audit/ -type f \(-name '*.conf' -o -name '*.rules' \) -exec stat -Lc "%U" {} +	● Passed	✓
31597	Ensure audit tools are 755 or more restrictive.	Command: stat -c "%n %a" /sbin/auditctl /sbin/aureport /sbin/ausearch /sbin/autrace /sbin/auditd /sbin/augeenrules	● Passed	✓
31599	Ensure audit tools belong to group root.	Command: stat -c "%n %a %U %G" /sbin/auditctl /sbin/aureport /sbin/ausearch /sbin/autrace /sbin/auditd /sbin/augeenrules	● Passed	✓
31600	Ensure rsyslog is installed.	Command: rpm -q rsyslog	● Passed	✓
31601	Ensure rsyslog service is enabled.	Command: systemctl is-enabled rsyslog	● Passed	✓
31602	Ensure journald is configured to send logs to rsyslog.	File: /etc/systemd/journald.conf	● Passed	✓
31603	Ensure rsyslog default file permissions are configured.	File: /etc/rsyslog.conf	● Passed	✓
31605	Ensure rsyslog is not configured to receive logs from ...	File: /etc/rsyslog.conf	● Passed	✓
31606	Ensure systemd-journal-remote is installed.	Command: rpm -q systemd-journal-remote	● Passed	✓
31607	Ensure systemd-journal-remote is enabled.	Command: systemctl is-enabled systemd-journal-upload.service	● Passed	✓
31608	Ensure journald is not configured to receive logs from...	Command: systemctl is-enabled systemd-journal-remote.socket	● Passed	✓
31609	Ensure journald service is enabled.	Command: systemctl is-enabled systemd-journald.service	● Passed	✓
31610	Ensure journald is configured to compress large log fil...	File: /etc/systemd/journald.conf	● Passed	✓
31611	Ensure journald is configured to write logfiles to persi...	File: /etc/systemd/journald.conf	● Passed	✓
31613	Ensure cron daemon is enabled.	Command: systemctl is-enabled crond	● Passed	✓
31614	Ensure permissions on /etc/crontab are configured.	Command: stat -L /etc/crontab	● Passed	✓
31615	Ensure permissions on /etc/cron.hourly are configured.	Command: stat -L /etc/cron.hourly	● Passed	✓
31616	Ensure permissions on /etc/cron.daily are configured.	Command: stat -L /etc/cron.daily	● Passed	✓
31617	Ensure permissions on /etc/cron.weekly are configured.	Command: stat -L /etc/cron.weekly	● Passed	✓
31618	Ensure permissions on /etc/cron.monthly are configur...	Command: stat -L /etc/cron.monthly	● Passed	✓
31618	Ensure permissions on /etc/cron.monthly are configur...	Command: stat -L /etc/cron.monthly	● Passed	✓
31619	Ensure permissions on /etc/cron.d are configured.	Command: stat -L /etc/cron.d	● Passed	✓
31620	Ensure permissions on /etc/ssh/ssh_config are confi...	Command: stat -L /etc/ssh/ssh_config	● Passed	✓
31622	Ensure SSH LogLevel is appropriate.	File: /etc/ssh/ssh_config	● Passed	✓
31623	Ensure SSH PAM is enabled.	File: /etc/ssh/ssh_config	● Passed	✓
31624	Ensure SSH root login is disabled.	File: /etc/ssh/ssh_config	● Passed	✓
31625	Ensure SSH HostbasedAuthentication is disabled.	File: /etc/ssh/ssh_config	● Passed	✓
31626	Ensure SSH PermitEmptyPasswords is disabled.	File: /etc/ssh/ssh_config	● Passed	✓
31627	Ensure SSH PermitUserEnvironment is disabled.	File: /etc/ssh/ssh_config	● Passed	✓
31628	Ensure SSH IgnoreRhosts is enabled.	File: /etc/ssh/ssh_config	● Passed	✓

31629	Ensure SSH X11 forwarding is disabled.	File: /etc/ssh/sshd_config	● Passed	✓
31630	Ensure SSH AllowTcpForwarding is disabled.	File: /etc/ssh/sshd_config	● Passed	✓
31631	Ensure system-wide crypto policy is not over-ridden.	File: /etc/sysconfig/ssh	● Passed	✓
31632	Ensure SSH warning banner is configured.	Command: sshd -T -C user=root	● Passed	✓
31633	Ensure SSH MaxAuthTries is set to 4 or less.	File: /etc/ssh/sshd_config	● Passed	✓
31634	Ensure SSH MaxStartups is configured.	File: /etc/ssh/sshd_config	● Passed	✓
31635	Ensure SSH MaxSessions is set to 10 or less.	Command: sshd -T -C user=root	● Passed	✓
31636	Ensure SSH LoginGraceTime is set to one minute or le...	File: /etc/ssh/sshd_config	● Passed	✓

El hardening que aplicamos sigue las recomendaciones del **CIS Benchmark**, que define las mejores prácticas para configurar servidores de forma segura. Para ello nos apoyamos utilizando **OpenSCAP**, una herramienta que analiza el sistema para identificar configuraciones inseguras y aplica cambios necesarios para cumplir con los estándares de seguridad.

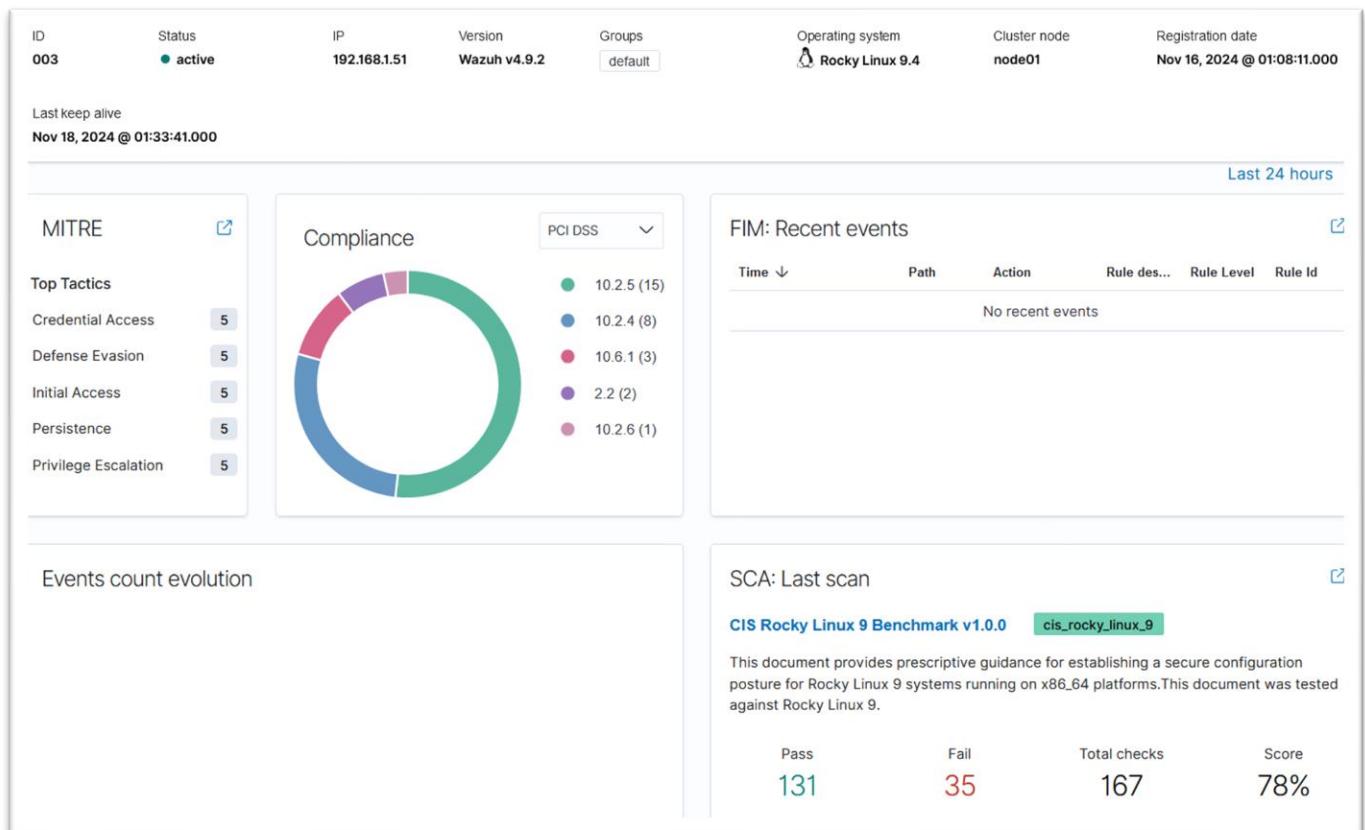
Se instalaron los paquetes necesarios para realizar el análisis y la remediación automatizada del sistema. Verificamos los perfiles, elegimos el de **CIS**, que establece las configuraciones recomendadas.

Antes de remediar realizamos un análisis del servidor en base al perfil seleccionado y genera un informe detallado en formato HTML con los resultados del cumplimiento.

Finalmente, se aplican los cambios necesarios para corregir las configuraciones inseguras y asegurar el servidor según las recomendaciones.

```
# sudo dnf install -y openscap openscap-scanner scap-security-guide
# sudo oscap info /usr/share/xml/scap/ssg/content/ssg-rhel9-ds.xml
# sudo oscap xccdf eval --profile xccdf_org.ssgproject.content_profile_cis --results
/tmp/oscapp-results.xml --report /tmp/oscapp-report.html
/usr/share/xml/scap/ssg/content/ssg-rhel9-ds.xml
# sudo oscap xccdf eval --profile xccdf_org.ssgproject.content_profile_cis --remediate
/usr/share/xml/scap/ssg/content/ssg-rhel9-ds.xml
```

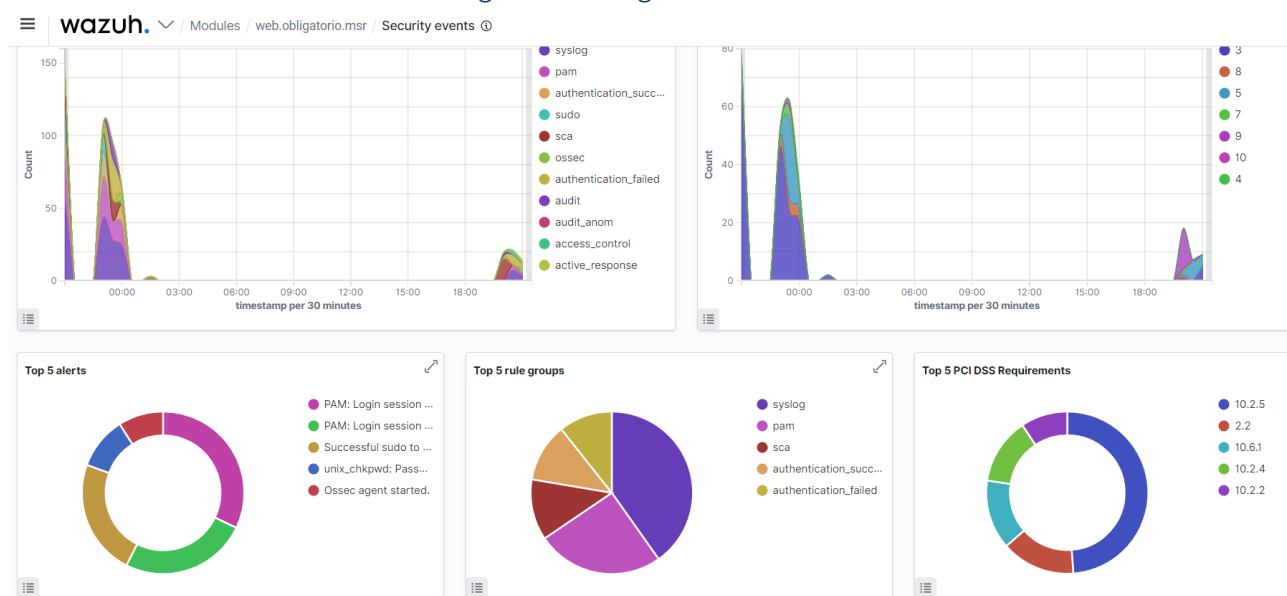

6.1.11 Score obtenido luego del hardening



7. SIEM

Para la solución del SIEM, optamos por Wazuh, en el cual, además de monitorear los eventos en los equipos, configuraremos respuestas activas para que actúen automáticamente en caso de detectar comportamientos que se puedan considerar una amenaza.

7.1.1 Visualización de alertas de seguridad con graficas



7.1.2 Bloqueo de ataques de fuerza bruta

Activamos un bloqueo de IP a los equipos que estén realizando un ataque de fuerza bruta a alguno de los equipos de nuestra red, configurando una regla que bloquea las IPs por 180 minutos, cuando se presenten problemas con autenticación de usuario repetidas veces.

Para ello lo primero que hicimos, fue revisar en los eventos de seguridad, cual es el ID del evento de falla en el logueo.

wazuh. / Modules / web.obligatorio.msr / Security events					
Time ↓	Technique(s)	Tactic(s)	Description	Level	Rule ID
> Nov 16, 2024 @ 20:38:48.783	T1110.001	Credential Access	PAM: User login failed.	5	5503

Abrimos el archivo de configuración en el servidor de Wazuh `/var/ossec/etc/ossec.conf` y verificamos que el bloque `<command>` con el nombre `firewall-drop` contenga la siguiente configuración en un bloque `<ossec_config>`:

```
<command>
  <name>firewall-drop</name>
  <executable>firewall-drop</executable>
  <timeout_allowed>yes</timeout_allowed>
</command>
```

Luego agregamos un bloque de respuesta activa (<active-response>) en este mismo archivo de configuración, utilizando el rule ID:

```
<ossec_config>
  <active-response>
    <command>firewall-drop</command>
    <location>local</location>
    <rules_id>5503</rules_id>
    <timeout>180</timeout>
  </active-response>
</ossec_config>
```

Para finalizar, reiniciamos para que tome los cambios de configuración y simulamos un ataque de fuerza bruta, para validar la configuración realizada.

```
# sudo systemctl restart wazuh-manager
```

7.1.2.1 Visualización de la alerta:

≡ wazuh. ▾ / Modules / web.obligatorio.msr / Security events ⓘ

Security Alerts					
Time ↓	Technique(s)	Tactic(s)	Description	Level	Rule ID
> Nov 16, 2024 @ 21:09:31.659			Host Unblocked by firewall-drop Active Response	3	652
> Nov 16, 2024 @ 21:06:31.381	T1110.001 T1021.004	Credential Access, Lateral Movement	sshd: authentication failed.	5	5760

7.1.3 Bloqueo por cambio de usuario

Este control nos permite detectar ataques de fuerza bruta contra otras cuentas de usuario en el mismo servidor ya que al intentar cambiar de usuario y equivocarse 3 veces consecutivas este control responde automáticamente deshabilitando temporalmente la cuenta del usuario.

Para lograr esto configuramos una regla en “/var/ossec/etc/rules/local_rules.xml” para detectar 3 intentos de autenticación fallidos en 2 minutos lo que genera una alerta de nivel 10 con el ID “120100”.

```
<group name="pam,syslog,">
  <rule id="120100" level="10" frequency="3" timeframe="120">
    <if_matched_sid>5503</if_matched_sid>
    <description>Possible password guess on $(dstuser): 3 failed logins in a short period of time</description>
    <mitre>
      <id>T1110</id>
    </mitre>
  </rule>
</group>
```

Luego, en el archivo de configuración `/var/ossec/etc/ossec.conf` validamos que el comando “disable-account” se encuentre con estas configuraciones:

```
<command>
  <name>disable-account</name>
  <executable>disable-account</executable>
  <timeout_allowed>yes</timeout_allowed>
</command>
```

Agregamos en el mismo archivo de configuración un active-response que ejecuta el comando “disable-account” por 5 minutos.

```
<ossec_config>
  <active-response>
    <command>disable-account</command>
    <location>local</location>
    <rules_id>120100</rules_id>
    <timeout>300</timeout>
  </active-response>
</ossec_config>
```

Para finalizar, reiniciamos para que tome los cambios de configuración y simulamos un ataque de fuerza bruta, para validar la configuración realizada.

```
# sudo systemctl restart wazuh-manager
```

7.1.3.1 Visualización de las alertas:

>	Nov 18, 2024 @ 22:06:20.640			Audit: Passwd was used to lock an account.
>	Nov 18, 2024 @ 22:06:20.637	T1110	Credential Access	Auditd: Limit of failed login attempts reached.
>	Nov 18, 2024 @ 22:06:20.628	T1110.001	Credential Access	unix_chkpwd: Password check failed.
>	Nov 18, 2024 @ 22:06:18.626	T1110.001	Credential Access	unix_chkpwd: Password check failed.
>	Nov 18, 2024 @ 22:06:12.622	T1110.001	Credential Access	unix_chkpwd: Password check failed.

7.1.4 Reiniciar Wazuh Agent frente a cambios

Este control configura Wazuh para detectar y responder automáticamente frente a cambios realizados en la configuración del agente, notificando el cambio en el dashboard y reiniciando el servicio.

En el archivo de configuración `/var/ossec/etc/ossec.conf` validamos que el comando “restart-wazuh” se encuentre con estas configuraciones:

```
<command>
  <name>restart-wazuh</name>
```

```
<executable>restart-wazuh</executable>
</command>
```

Agregamos en el mismo archivo de configuración un “active-response” que vincula el comando “restart-wazuh” .

```
<ossec_config>
  <active-response>
    <command>restart-wazuh</command>
    <location>local</location>
    <rules_id>100009</rules_id>
  </active-response>
</ossec_config>
```

Añadimos la regla en “/var/ossec/etc/rules/local_rules.xml” que indica el evento de cambio en el archivo “ossec.conf” con el ID “100009” notificando en el dashboard con el mensaje seteado en el campo “description”.

```
<group name="restart,">
  <rule id="100009" level="5">
    <if_sid>550</if_sid>
    <match>ossec.conf</match>
    <description>Changes made to the agent configuration file - $(file)</description>
  </rule>
</group>
```

7.1.4.1 Visualización de la alerta:

>	Nov 18, 2024 @ 22:31:30.350		Ossec agent started.	3	503
>	Nov 18, 2024 @ 22:31:28.022	T1562.001	Defense Evasion	3	506
>	Nov 18, 2024 @ 22:31:26.817		Changes made to the agent configuration file - /var/ossec/etc/ossec.conf	5	100009

7.1.5 Detección malware usando Yara

Usamos la integración de YARA con Wazuh para analizar archivos añadidos o modificados en los servidores en busca de malware. YARA es una herramienta diseñada para detectar y clasificar artefactos de malware.

Para ello, lo primero que hicimos fueron las siguientes instalaciones y extracciones en los servidores a ser monitoreados:

```
sudo yum makecache
sudo yum install epel-release
sudo yum update
sudo yum install -y make automake gcc autoconf libtool openssl-devel pkg-config jq
sudo curl -LO https://github.com/VirusTotal/yara/archive/v4.2.3.tar.gz
sudo tar -xvzf v4.2.3.tar.gz -C /usr/local/bin/ && rm -f v4.2.3.tar.gz
cd /usr/local/bin/yara-4.2.3/
```

```
sudo ./bootstrap.sh && sudo ./configure && sudo make && sudo make install && sudo make check
```

Descargamos las reglas de detección de YARA:

[illegible]

Creamos un script llamado `yara.sh` en el directorio `/var/ossec/active-response/bin/`. Esto es necesario para los análisis de respuesta activa de Wazuh-YARA:

```
#!/bin/bash
# Wazuh - Yara active response
# Copyright (C) 2015-2022, Wazuh Inc.
#
# This program is free software; you can redistribute it
# and/or modify it under the terms of the GNU General Public
# License (version 2) as published by the FSF - Free Software
# Foundation.

#----- Gather parameters -----#

# Extra arguments
read INPUT_JSON
YARA_PATH=$(echo $INPUT_JSON | jq -r .parameters.extra_args[1])
YARA_RULES=$(echo $INPUT_JSON | jq -r .parameters.extra_args[3])
FILENAME=$(echo $INPUT_JSON | jq -r .parameters.alert.syscheck.path)

# Set LOG_FILE path
LOG_FILE="logs/active-responses.log"

size=0
actual_size=$(stat -c %s ${FILENAME})
while [ ${size} -ne ${actual_size} ]; do
    sleep 1
```

```
size=${actual_size}
actual_size=$(stat -c %s ${FILENAME})
done

#----- Analyze parameters -----#

if [[ ! $YARA_PATH ]] || [[ ! $YARA_RULES ]]
then
    echo "wazuh-yara: ERROR - Yara active response error. Yara path and rules parameters are mandatory." >> ${LOG_FILE}
    exit 1
fi

#----- Main workflow -----#

# Execute Yara scan on the specified filename
yara_output="$(("${YARA_PATH}"/yara -w -r "${YARA_RULES}" "${FILENAME}")"

if [[ $yara_output != "" ]]
then
    # Iterate every detected rule and append it to the LOG_FILE
    while read -r line; do
        echo "wazuh-yara: INFO - Scan result: $line" >> ${LOG_FILE}
    done <<< "$yara_output"
fi

exit 0;
```

Cambiamos el usuario y grupo owner del archivo yara.sh a root y wazuh y los permisos a 0750:

```
sudo chown root:wazuh /var/ossec/active-response/bin/yara.sh
sudo chmod 750 /var/ossec/active-response/bin/yara.sh
```

Agregamos lo siguiente dentro del bloque <syscheck> en el archivo de configuración /var/ossec/etc/ossec.conf del agente Wazuh para monitorear el directorio /tmp/yara/malware:

```
<directories realtime="yes">/tmp/yara/malware</directories>
```

Reiniciamos el agente de Wazuh para que tome los cambios en la configuración:

```
$ sudo systemctl restart wazuh-agent
```

7.1.5.1 Configuraciones en el servidor de Wazuh

Realizamos los siguientes pasos para configurar Wazuh y alertar sobre cambios en archivos en el directorio de los servidores que serán monitoreados. Los pasos también configuran un script de respuesta activa que se activa cada vez que se detecta un archivo sospechoso.

Agregamos las siguientes reglas al archivo `/var/ossec/etc/rules/local_rules.xml`. Las reglas detectan eventos de FIM en el directorio monitoreado y generan alertas cuando la integración con YARA encuentra malware. En producción, cambiaríamos las reglas para detectar eventos en otros directorios.

```
<group name="syscheck,">
  <rule id="100300" level="7">
    <if_sid>550</if_sid>
    <field name="file">/tmp/yara/malware/</field>
    <description>File modified in /tmp/yara/malware/ directory.</description>
  </rule>
  <rule id="100301" level="7">
    <if_sid>554</if_sid>
    <field name="file">/tmp/yara/malware/</field>
    <description>File added to /tmp/yara/malware/ directory.</description>
  </rule>
</group>

<group name="yara,">
  <rule id="108000" level="0">
    <decoded_as>yara_decoder</decoded_as>
    <description>Yara grouping rule</description>
  </rule>
  <rule id="108001" level="12">
    <if_sid>108000</if_sid>
    <match>wazuh-yara: INFO - Scan result: </match>
    <description>File "$(yara_scanned_file)" is a positive match. Yara rule:
$(yara_rule)</description>
  </rule>
</group>
```

Agregamos los siguientes decodificadores al archivo `/var/ossec/etc/decoders/local_decoder.xml` del servidor Wazuh. Esto permite extraer la información de los resultados del escaneo de YARA:

```
<decoder name="yara_decoder">
  <prematch>wazuh-yara:</prematch>
</decoder>

<decoder name="yara_decoder1">
  <parent>yara_decoder</parent>
  <regex>wazuh-yara: (\S+) - Scan result: (\S+) (\S+)</regex>
```



```
<order>log_type, yara_rule, yara_scanned_file</order>
</decoder>
```

Agregamos la siguiente configuración al archivo de configuración `/var/ossec/etc/ossec.conf` del servidor Wazuh. Esto configura el módulo de respuesta activa para que se active después de que se ejecuten las reglas 100300 y 100301:

```
<ossec_config>
  <command>
    <name>yara_linux</name>
    <executable>yara.sh</executable>
    <extra_args>-yara_path /usr/local/bin -yara_rules
/tmp/yara/rules/yara_rules.yar</extra_args>
    <timeout_allowed>no</timeout_allowed>
  </command>

  <active-response>
    <command>yara_linux</command>
    <location>local</location>
    <rules_id>100300,100301</rules_id>
  </active-response>
</ossec_config>
```

Reiniciamos el servicio de Wazuh para que tome los cambios:

```
$ sudo systemctl restart wazuh-manager
```

7.1.5.2 Emulación de ataque:

Creamos el script `/tmp/yara/malware/malware_downloader.sh` en el servidor que se estará monitoreado para descargar muestras de malware:

```
#!/bin/bash
# Wazuh - Malware Downloader for test purposes
# Copyright (C) 2015-2022, Wazuh Inc.
#
# This program is free software; you can redistribute it
# and/or modify it under the terms of the GNU General Public
# License (version 2) as published by the FSF - Free Software
# Foundation.

function fetch_sample(){

  curl -s -XGET "$1" -o "$2"

}
```

```
echo "WARNING: Downloading Malware samples, please use this script with  caution."
read -p "  Do you want to continue? (y/n)" -n 1 -r ANSWER
echo

if [[ $ANSWER =~ ^[Yy]$ ]]
then
    echo
    # Mirai
    echo "# Mirai: https://en.wikipedia.org/wiki/Mirai_(malware)"
    echo "Downloading malware sample..."
    fetch_sample "https://wazuh-demo.s3-us-west-1.amazonaws.com/mirai"
"/tmp/yara/malware/mirai" && echo "Done!" || echo "Error while downloading."
    echo

    # Xbash
    echo "# Xbash: https://unit42.paloaltonetworks.com/unit42-xbash-combines-botnet-
ransomware-coinmining-worm-targets-linux-windows/"
    echo "Downloading malware sample..."
    fetch_sample "https://wazuh-demo.s3-us-west-1.amazonaws.com/xbash"
"/tmp/yara/malware/xbash" && echo "Done!" || echo "Error while downloading."
    echo

    # VPNFilter
    echo "# VPNFilter: https://news.sophos.com/en-us/2018/05/24/vpnfilter-botnet-a-
sophoslabs-analysis/"
    echo "Downloading malware sample..."
    fetch_sample "https://wazuh-demo.s3-us-west-1.amazonaws.com/vpn_filter"
"/tmp/yara/malware/vpn_filter" && echo "Done!" || echo "Error while downloading."
    echo

    # Webshell
    echo "# WebShell: https://github.com/SecWiki/WebShell-
2/blob/master/Php/Worse%20Linux%20Shell.php"
    echo "Downloading malware sample..."
    fetch_sample "https://wazuh-demo.s3-us-west-1.amazonaws.com/webshell"
"/tmp/yara/malware/webshell" && echo "Done!" || echo "Error while downloading."
    echo
fi
```

Ejecutamos el script `malware_downloader.sh` para descargar muestras de malware en el directorio `/tmp/yara/malware`:

```
$ sudo bash /tmp/yara/malware/malware_downloader.sh
```

7.1.5.3 Visualización de las alertas:

Nov 19, 2024 > @ 00:04:13.871	File "/tmp/yara/malware/webshell" is a positive match. Yara rule: Webshell_Worse_Linux_Shell_1_RID320C	12	108001
Nov 19, 2024 > @ 00:04:13.870	File "/tmp/yara/malware/webshell" is a positive match. Yara rule: Webshell_Worse_Linux_Shell_php_RID3323	12	108001
Nov 19, 2024 > @ 00:04:13.870	File "/tmp/yara/malware/vpn_filter" is a positive match. Yara rule: MAL_ELF_VPNFilter_3_RID2D6C	12	108001
Nov 19, 2024 > @ 00:04:11.869	File "/tmp/yara/malware/vpn_filter" is a positive match. Yara rule: MAL_ELF_VPNFilter_3_RID2D6C	12	108001
Nov 19, 2024 > @ 00:04:11.869	File "/tmp/yara/malware/vpn_filter" is a positive match. Yara rule: MAL_ELF_VPNFilter_3_RID2D6C	12	108001

8. Autenticación Federada

Utilizamos la Autenticación Federada como solución que permite a los usuarios acceder a nuestro sitio web utilizando credenciales de un proveedor externo, como Google, Microsoft o Facebook. Este método mejora la seguridad al reducir la necesidad de gestionar múltiples contraseñas y facilita la experiencia del usuario al unificar el acceso. En nuestro caso, integramos autenticación federada en WordPress utilizando el plugin “Nextend Social Login” con Google como proveedor.

A continuación, describimos los pasos realizados para implementar esta solución en nuestro sitio WordPress.

Luego de [instalar Wordpress](#), dentro de /wp-admin vamos a plugins y buscamos “Nextend Social Login and Register”, lo instalamos y activamos.



Luego seguimos las instrucciones que nos indica el plugin.

Accedimos a la [consola de desarrolladores de Google](#) y creamos nuestro proyecto:



En el menú “Pantalla de consentimiento” seleccionamos el tipo de usuario como externo.

Pantalla de consentimiento de OAuth



La administración de la pantalla de consentimiento de OAuth está cambiando. Esta página se reemplazó por una experiencia nueva y más fácil de usar. Las páginas actuales solo estarán disponibles durante unos días más.

[IR A CONOCER LA NUEVA EXPERIENCIA](#)

Elige cómo deseas configurar y registrar tu app, incluidos los usuarios objetivo. Puedes asociar una sola app con tu proyecto.

User Type

☐ Interno ?

Solo está disponible para los usuarios de tu organización. No necesitarás enviar tu app para verificarla. [Obtén más información sobre el tipo de usuario](#)

☒ Usuarios externos ?

Disponible para cualquier usuario de prueba con una Cuenta de Google. Tu app se iniciará en modo de prueba y solo estará disponible para los usuarios que agregues a la lista de usuarios de prueba. Una vez que la app esté lista para enviarse a producción, puede que debas verificarla. [Obtén más información sobre el tipo de usuario](#)

CREAR

Creamos la aplicación y completamos con los siguientes datos:

Nombre de la aplicación *

App name

El nombre de la aplicación que solicita el consentimiento

Correo electrónico de asistencia del usuario *

alexisxdandrea@gmail.com

Para que los usuarios se comuniquen contigo si tienen preguntas sobre su consentimiento. [Más información](#)

Logotipo de la app

Este es tu logotipo. Ayuda a que las personas reconozcan tu app y aparece en la pantalla de consentimiento de OAuth.

Después de subir un logotipo, deberás enviar tu app para verificarla, a menos que esté configurada solo para uso interno o tenga el estado de publicación "Prueba". [Más información](#)

Archivo de logotipo que debe subirse

[EXPLORAR](#)

Sube una imagen con un tamaño máximo de 1 MB en la pantalla de consentimiento que ayudará a los usuarios a reconocer tu app. Los formatos de imagen permitidos son JPG, PNG y BMP. Para obtener los mejores resultados, los logotipos deben ser cuadrados y de 120 píxeles x 120 píxeles.

Dominio de la app

Para protegerlos a ti y a tus usuarios, Google solo permite que las apps que usan OAuth puedan emplear los dominios autorizados. Se mostrará la siguiente información a los usuarios en la pantalla de consentimiento.

Página principal de la aplicación

<https://dandrea.com.uy>

Proporciona a los usuarios un vínculo a tu página principal

Vínculo a la Política de Privacidad de la aplicación

Proporciona a los usuarios un vínculo a tu página pública de Política de Privacidad

Vínculo a las Condiciones del Servicio de la aplicación

Proporciona a los usuarios un vínculo a tu página pública de Condiciones del Servicio

Dominios autorizados ?

Cuando un dominio se usa en la pantalla de consentimiento o en la configuración del cliente de OAuth, debe contar con un registro previo aquí. Si debes verificar la app, ve [Google Search Console](#) para comprobar si tus dominios están autorizados. [Más información](#)

Dominio autorizado 1 *

dandrea.com.uy

Luego seleccionamos el menú "Credenciales" y creamos las credenciales con el tipo "OAuth client ID" y el tipo de aplicación como "Web Application":

[←](#) ID de cliente para Aplicación web [BORRAR](#)

i La administración de la pantalla de consentimiento de OAuth está cambiando. Esta página se reemplazó por una experiencia nueva y más fácil de usar. Las páginas actuales solo estarán disponibles durante unos días más.

[IR A CONOCER LA NUEVA EXPERIENCIA](#)

Nombre *

Auth0 App

El nombre de tu cliente de OAuth 2.0. Este nombre solo se usa para identificar al cliente en la consola y no se mostrará a los usuarios finales.

i Los dominios de los URI que agregues a continuación se incorporarán automáticamente a tu [pantalla de consentimiento de OAuth](#) como [dominios autorizados](#).

Orígenes autorizados de JavaScript

Para usar con solicitudes de un navegador

[+ AGREGAR URI](#)

URI de redireccionamiento autorizados

Para usar con solicitudes de un servidor web

URI 1 *

<https://dandrea.com.uy/wp/wp-login.php?loginSocial=google>




[+ AGREGAR URI](#)

Additional information

ID de cliente	1079276498568 hm6p05e2oqfpe008g00ves hs040bot45.apps.google ocontent.com
Fecha de creación	21 de noviembre de 2024, 22:50:37 GMT-3

Secretos del cliente

Si estás en proceso de cambiar los secretos del cliente, puedes rotarlos de forma manual sin tiempo de inactividad. [Más información](#)

Secreto del cliente	0000PX hWQzfsVxMv0Hv9jz_RQe Hoe  
Fecha de creación	21 de noviembre de 2024, 22:50:36 GMT-3
Estado	 Habilitado

[+ ADD SECRET](#)

Ponemos nuestra URL de redireccionamiento autorizado y le damos a “Crear”.

Una vez creado, volvemos a la pantalla de consentimiento y publicamos la app:

Estado de publicación

Prueba

[PUBLICAR LA APLICACIÓN](#)

Para finalizar vamos a la configuración del plugin y ponemos el ID de cliente y el secret proporcionado por Google.

Client ID - (Required)

~~1079276498568~~
~~hm6p05e2oqfpe008g00ves~~
~~hs040bot45.apps.google~~
~~ocontent.com~~

If you are not sure what is your Client ID, please head over to [Getting Started](#)

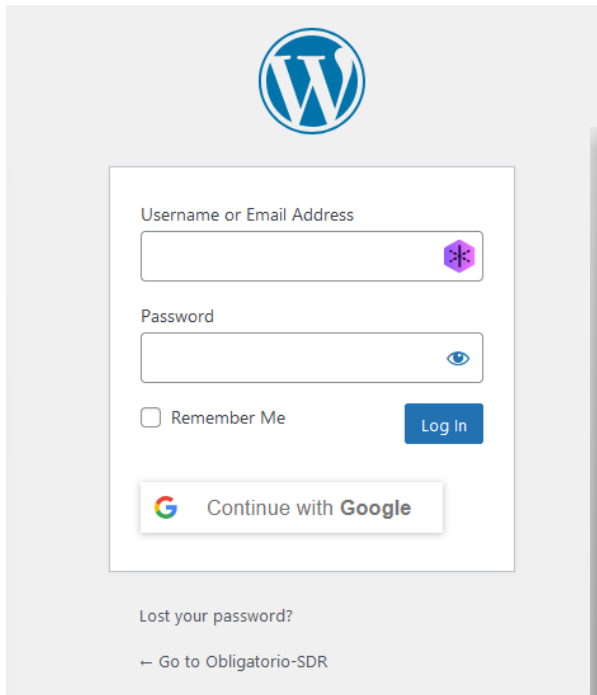
Client Secret - (Required)

~~0000PX~~
~~hWQzfsVxMv0Hv9jz_RQe~~
~~Hoe~~

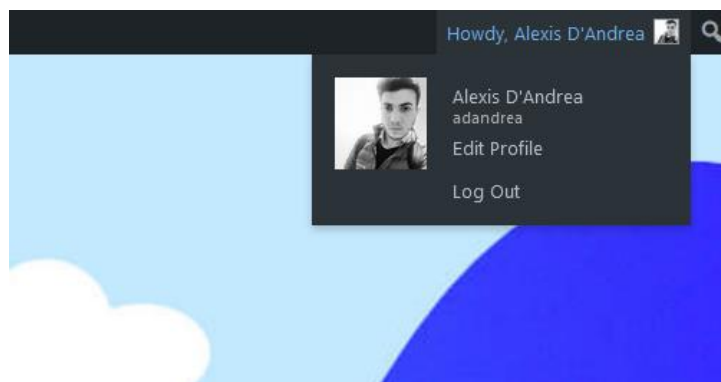
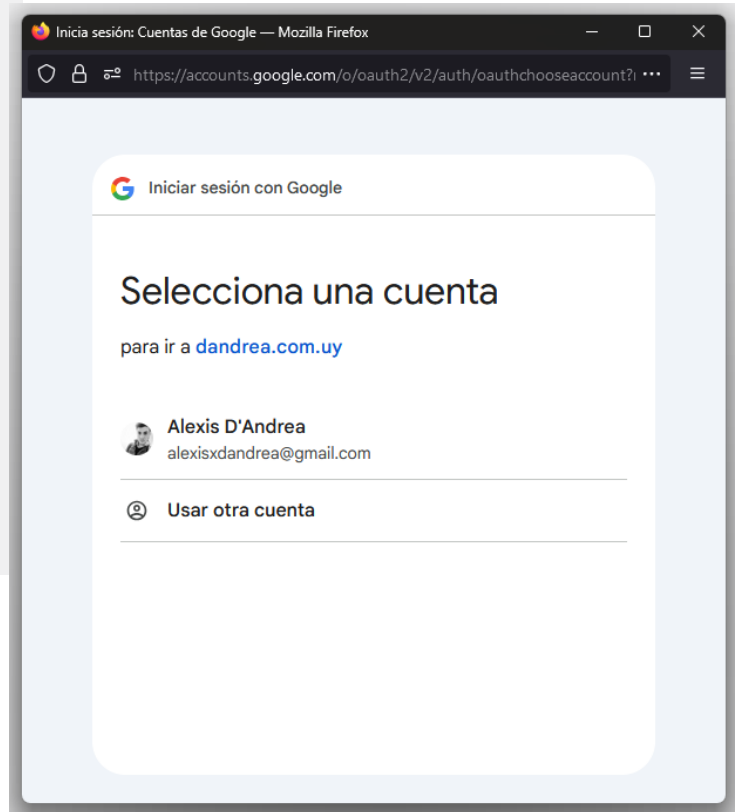
Select account on each login ☒ Enabled
Disable, when you don't want to see the account select prompt on each login.

[Save Changes](#)

Guardamos los cambios y validamos:



The image shows the WordPress login interface. At the top is the WordPress logo. Below it is a login form with two input fields: 'Username or Email Address' and 'Password'. The password field has an eye icon to toggle visibility. Below the password field is a checkbox labeled 'Remember Me' and a blue 'Log In' button. At the bottom of the form is a 'Continue with Google' button. Below the form, there is a link for 'Lost your password?' and a link to 'Go to Obligatorio-SDR'.



9. Anexo

9.1 Instalación Wordpress:

```
sudo yum update -y
sudo yum install mariadb105-server -y
sudo yum install httpd
sudo yum install php
wget http://wordpress.org/latest.tar.gz
tar -xzf latest.tar.gz
cd /var/www/html/wordpress
mkdir /var/www/html/wordpress/wp-content/uploads
sudo chown -R apache:apache /var/www/html/*
sudo chmod -R 777 /var/www/html/wordpress

sudo systemctl start mariadb
sudo systemctl enable mariadb
sudo mysql_secure_installation
mysql -u usuario -p
create database wordpress;
CREATE USER 'user'@'localhost' IDENTIFIED BY 'passuser';
GRANT ALL PRIVILEGES ON wordpress.* TO 'user'@'localhost';
FLUSH PRIVILEGES;
Exit

cd /var/www/html/wordpress
cp wp-config-sample.php wp-config.php
sudo nano wp-config.php
```

```
// ** Database settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress' );

/** Database username */
define( 'DB_USER', 'user' );

/** Database password */
define( 'DB_PASSWORD', 'passuser' );

/** Database hostname */
define( 'DB_HOST', 'localhost' );

/** Database charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );

/** The database collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );
```

En el mismo archivo modificamos las llaves de autenticación desde:

<https://api.wordpress.org/secret-key/1.1/salt/>

copiamos los datos y lo cambiamos por los del archivo:

```
* Authentication unique keys and salts.
*
* Change these to different unique phrases! You can generate these using
* the {@link https://api.wordpress.org/secret-key/1.1/salt/ WordPress.org secr$
*
* You can change these at any point in time to invalidate all existing cookies.
* This will force all users to have to log in again.
*
* @since 2.6.0
*/

define('AUTH KEY',          '7At6WVE2Y2sq5kTNO/!`wI0AkXpq{v! aSeCm0+w $312Jr6eR6$
define('SECURE AUTH KEY',  '<M/1V6hYy%@}Vo|8C7(+DEh0fnq}@[sDO2(!G>I;bWGm+Nc+ mR$
define('LOGGED IN KEY',    'by}sRoqa26*+GcfzKQ7< 3$U4dLZ}HMehP5.W;E;8zKtfO~]OdO$
define('NONCE KEY',        'REI6uBj2U+YLc% Dq93k>dX*-R!7|.dryL|4JwfPDe0p<e>b9m&$
define('AUTH SALT',        '@h/s6I|Y.4)V/=Z$(-T??bQtJ$V$3v] %Qo*Z&`oK,r1|t|Y`wE1$
define('SECURE AUTH SALT', 'WSn*;nljKq6t-%M[]dOk3|!Z 2 >mNZ0SUx|-v.4NIeiiKJ<2kl$
define('LOGGED IN SALT',   'M[o8zVr*]yAP*,J!13K&Lm-9fh $*A`>%TTFDc,wUJW:1IAo8P7$
define('NONCE SALT',       '|l$-J 0Kpwti%oX)0b[|B88tZM&k!?k<BMv#[[]0TjUsCCn ]Y(B$
```

cd /etc/httpd/conf.d

sudo nano wordpress.conf

<VirtualHost *:80>

ServerAdmin root@localhost

DocumentRoot /var/www/html/wordpress

<Directory #/var/www/html/wordpress/#>

AllowOverride All

Require all granted

</Directory>

</VirtualHost>

sudo systemctl restart httpd

Hola

¡Bienvenido al famoso proceso de instalación de WordPress en cinco minutos! Simplemente completa la información siguiente y estarás a punto de usar la más enriquecedora y potente plataforma de publicación personal del mundo.

Información necesaria

Por favor, debes facilitarnos los siguientes datos. No te preocupes, siempre podrás cambiar estos ajustes más tarde.

Título del sitio

Nombre de usuario

Los nombres de usuario pueden tener únicamente caracteres alfanuméricos, espacios, guiones bajos, guiones medios, puntos y el símbolo @.

Contraseña

Strong

[Show](#)

Importante: Necesitas esta contraseña para acceder. Por favor, guárdala en un lugar seguro.

Tu correo electrónico

Comprueba bien tu dirección de correo electrónico antes de continuar.

Visibilidad en los motores de búsqueda

☐ Disuadir a los motores de búsqueda de indexar este sitio

Depende de los motores de búsqueda atender esta petición o no.

[Instalar WordPress](#)

9.2 Terraform:

```
provider "aws" {
  region = "us-east-1"
}

resource "aws_vpc" "vpc_obligatorio" {
  cidr_block      = "10.0.0.0/16"
  enable_dns_support = true
  enable_dns_hostnames = true

  tags = {
    Name = "vpc_obligatorio"
  }
}

resource "aws_security_group" "sg_siem" {
  vpc_id = aws_vpc.vpc_obligatorio.id
  name    = "sg_siem"

  ingress {
    from_port = 22
    to_port   = 22
    protocol  = "tcp"
    security_groups = [aws_security_group.sg_jump_server.id]
  }

  egress {
    from_port = 0
    to_port   = 0
    protocol  = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }

  tags = {
    Name = "sg_siem"
  }
}

resource "aws_instance" "siem" {
  ami          = "ami-063d43db0594b521b"
  instance_type = "t2.micro"
  subnet_id    = aws_subnet.subnet_private_a.id
  security_groups = [aws_security_group.sg_siem.id]

  key_name = "vockey"
```

```
tags = {
  Name = "siem"
}

resource "aws_instance" "waf" {
  ami            = "ami-063d43db0594b521b"
  instance_type  = "t2.micro"
  subnet_id      = aws_subnet.subnet_public_b.id
  security_groups = [aws_security_group.sg_waf.id]

  key_name = "vockey"

  tags = {
    Name = "waf"
  }
}

resource "aws_security_group" "sg_jump_server" {
  vpc_id = aws_vpc.vpc_obligatorio.id
  name    = "sg_jump_server"

  ingress {
    from_port = 22
    to_port   = 2222
    protocol  = "tcp"
    cidr_blocks = ["0.0.0.0/0"] # En este campo se pondria nuestra IP
  }

  egress {
    from_port = 0
    to_port   = 0
    protocol  = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }

  tags = {
    Name = "sg_jump_server"
  }
}

resource "aws_instance" "jump_server" {
  ami            = "ami-063d43db0594b521b"
  instance_type  = "t2.micro"
  subnet_id      = aws_subnet.subnet_public_a.id
```

```
security_groups = [aws_security_group.sg_jump_server.id]

key_name = "vockey"

user_data = base64encode(local.jump_server_user_data)

tags = {
    Name = "jump_server"
}

resource "aws_security_group" "sg_load_balancer" {
    name = "sg_load_balancer"
    vpc_id = aws_vpc.vpc_obligatorio.id
    ingress {
        from_port = 80
        to_port = 80
        protocol = "tcp"
        cidr_blocks = ["0.0.0.0/0"]
# security_groups = [aws_security_group.sg_waf.id]
    }

    egress {
        from_port = 0
        to_port = 0
        protocol = "-1"
        cidr_blocks = ["0.0.0.0/0"]
    }
    tags = {
        Name = "sg_load_balancer"
    }
}

resource "aws_security_group" "sg_waf" {
    name = "sg_waf"
    vpc_id = aws_vpc.vpc_obligatorio.id
    ingress {
        from_port = 22
        to_port = 22
        protocol = "tcp"
        security_groups = [aws_security_group.sg_jump_server.id]
    }
    ingress {
        from_port = 80
        to_port = 80
        protocol = "tcp"
        cidr_blocks = ["0.0.0.0/0"]
    }
}
```

```
}
egress {
    from_port    = 0
    to_port      = 0
    protocol     = "-1"
    cidr_blocks  = ["0.0.0.0/0"]
}
tags = {
    Name = "sg_waf"
}
}

resource "aws_security_group" "sg_appweb" {
    name = "sg_appweb"
    vpc_id = aws_vpc.vpc_obligatorio.id
    ingress {
        from_port    = 22
        to_port      = 22
        protocol     = "tcp"
        security_groups = [aws_security_group.sg_jump_server.id]
    }
    ingress {
        from_port    = 80
        to_port      = 80
        protocol     = "tcp"
        security_groups = [aws_security_group.sg_load_balancer.id]
    }
    egress {
        from_port    = 0
        to_port      = 0
        protocol     = "-1"
        cidr_blocks  = ["0.0.0.0/0"]
    }
    tags = {
        Name = "sg_appweb"
    }
}

resource "aws_security_group" "sg_mysql" {
    name      = "sg_mysql"
    vpc_id    = aws_vpc.vpc_obligatorio.id

    ingress {
        from_port    = 3306
        to_port      = 3306
        protocol     = "tcp"
        security_groups = [aws_security_group.sg_appweb.id]
    }
}
```

```
}  
}  
  
# Crear un Internet Gateway  
resource "aws_internet_gateway" "internet_gateway" {  
  vpc_id = aws_vpc.vpc_obligatorio.id  
  tags = {  
    Name = "internet_gateway"  
  }  
}  
  
# Crear subredes privadas y publicas en las zonas de disponibilidad A y B dentro de  
# la nueva VPC  
resource "aws_subnet" "subnet_private_a" {  
  vpc_id          = aws_vpc.vpc_obligatorio.id  
  cidr_block      = "10.0.1.0/24"  
  availability_zone = "us-east-1a"  
  map_public_ip_on_launch = "true"  
  tags = {  
    Name = "subnet_private_a"  
  }  
}  
  
resource "aws_subnet" "subnet_private_b" {  
  vpc_id          = aws_vpc.vpc_obligatorio.id  
  cidr_block      = "10.0.2.0/24"  
  availability_zone = "us-east-1b"  
  map_public_ip_on_launch = "true"  
  tags = {  
    Name = "subnet_private_b"  
  }  
}  
  
resource "aws_subnet" "subnet_public_a" {  
  vpc_id          = aws_vpc.vpc_obligatorio.id  
  cidr_block      = "10.0.3.0/24"  
  availability_zone = "us-east-1a"  
  map_public_ip_on_launch = "true"  
  tags = {  
    Name = "subnet_public_a"  
  }  
}  
  
resource "aws_subnet" "subnet_public_b" {  
  vpc_id          = aws_vpc.vpc_obligatorio.id  
  cidr_block      = "10.0.4.0/24"
```



```
availability_zone = "us-east-1b"
map_public_ip_on_launch = "true"
tags = {
    Name = "subnet_public_b"
}
}

# Crear grupo de subredes para la base de datos dentro de la nueva VPC
resource "aws_db_subnet_group" "db_subnet_group" {
    name          = "mysql_db-subnet-group"
    subnet_ids    = [aws_subnet.subnet_private_a.id, aws_subnet.subnet_private_b.id]
}

resource "aws_route_table" "route_table" {
    vpc_id = aws_vpc.vpc_obligatorio.id

    route {
        cidr_block = "0.0.0.0/0"
        gateway_id = aws_internet_gateway.internet_gateway.id
    }
    tags = {
        Name = "route_table"
    }
}

resource "aws_route_table_association" "subnet_public_a_asso" {
    subnet_id      = aws_subnet.subnet_public_a.id
    route_table_id = aws_route_table.route_table.id
}

resource "aws_route_table_association" "subnet_public_b_asso" {
    subnet_id      = aws_subnet.subnet_public_b.id
    route_table_id = aws_route_table.route_table.id
}

# Crear una Elastic IP para el NAT Gateway de la AV a
resource "aws_eip" "nat_eip_a" {
    domain = "vpc"
}

# Crear el NAT Gateway en la subred pública de la AV a
resource "aws_nat_gateway" "nat_gateway_a" {
    allocation_id = aws_eip.nat_eip_a.id
    subnet_id     = aws_subnet.subnet_public_a.id
}
```

```
}

# Crear una tabla de rutas para la subred privada a
resource "aws_route_table" "private_route_table_a" {
  vpc_id = aws_vpc.vpc_obligatorio.id

  route {
    cidr_block      = "0.0.0.0/0"
    nat_gateway_id = aws_nat_gateway.nat_gateway_a.id
  }
}

# Asociar la tabla de rutas privada a la subred privada a
resource "aws_route_table_association" "subnet_private_a_asso" {
  subnet_id      = aws_subnet.subnet_private_a.id
  route_table_id = aws_route_table.private_route_table_a.id
}

# Crear una Elastic IP para el NAT Gateway de la AV b
resource "aws_eip" "nat_eip_b" {
  domain = "vpc"
}

# Crear el NAT Gateway en la subred pública de la AV b
resource "aws_nat_gateway" "nat_gateway_b" {
  allocation_id = aws_eip.nat_eip_b.id
  subnet_id     = aws_subnet.subnet_public_b.id
}

# Crear una tabla de rutas para la subred privada b
resource "aws_route_table" "private_route_table_b" {
  vpc_id = aws_vpc.vpc_obligatorio.id

  route {
    cidr_block      = "0.0.0.0/0"
    nat_gateway_id = aws_nat_gateway.nat_gateway_b.id
  }
}

# Asociar la tabla de rutas privada a la subred privada b
resource "aws_route_table_association" "subnet_private_b_asso" {
  subnet_id      = aws_subnet.subnet_private_b.id
```

```
route_table_id = aws_route_table.private_route_table_b.id
}

# Crear un balanceador de carga de aplicación (ALB)
resource "aws_lb" "aplitation_load_balancer" {
  name                = "aplitation-load-balancer"
  internal            = false
  load_balancer_type = "application"
  security_groups     = [aws_security_group.sg_load_balancer.id]
  subnets            = [aws_subnet.subnet_public_a.id,
aws_subnet.subnet_public_b.id]

}

# Definir reglas de escucha y destino para el ALB
resource "aws_lb_listener" "listener" {
  load_balancer_arn = aws_lb.aplitation_load_balancer.arn
  port              = "80"
  protocol          = "HTTP"

  default_action {
    type                = "forward"
    target_group_arn = aws_lb_target_group.obligatorio_target_group.arn
  }
}

# Crear grupos de destino para las instancias de aplicación en las zonas de
disponibilidad A y B
resource "aws_lb_target_group" "obligatorio_target_group" {
  name        = "obligatorio-target-group"
  port        = 80
  protocol    = "HTTP"
  vpc_id      = aws_vpc.vpc_obligatorio.id
  target_type = "instance"

  health_check {
    path          = "/"
    port          = "80"
    protocol      = "HTTP"
    healthy_threshold = 3
    unhealthy_threshold = 3
    timeout        = 5
    interval       = 60
  }
}
```

```
}

resource "aws_lb_listener_rule" "listener_rule" {
  listener_arn = aws_lb_listener.listener.arn
  priority     = 100

  action {
    type             = "forward"
    target_group_arn = aws_lb_target_group.obligatorio_target_group.arn
  }

  condition {
    path_pattern {
      values = ["/var/www/html/index.html"]
    }
  }
}

resource "aws_db_instance" "mysql_db" {
  allocated_storage     = 20
  storage_type          = "gp2"
  engine                = "mysql"
  engine_version        = "5.7.44"
  instance_class        = "db.t3.micro"
  username              = "admin"
  password              = "password"
  skip_final_snapshot   = true
  vpc_security_group_ids = [aws_security_group.sg_mysql.id]
  db_subnet_group_name  = aws_db_subnet_group.db_subnet_group.name
  db_name               = "FondoBlanco"
  tags = {
    Name = "mysql_db"
  }
}

locals {
  webapp_user_data = <<-EOF
  #!/bin/bash
  sudo yum -y install httpd
  sudo systemctl enable httpd
  sudo systemctl start httpd

  EOF
  jump_server_user_data = <<-EOF
  #!/bin/bash
  sudo dnf -y install firewalld
```

```
    sudo echo "Advertencia: Acceso no autorizado. Este sistema es propiedad del
obligatorio" > /home/ec2-user/banner.txt
    sudo sed -i 's|#Banner none|Banner /home/ec2-user/banner.txt|'
/etc/ssh/sshd_config

    sudo sed -i 's|#Port 22/Port 2222/' /etc/ssh/sshd_config
    sudo systemctl start firewalld
    sudo systemctl enable firewalld
    sudo firewall-cmd --permanent --add-rich-rule="rule family='ipv4' source
address='0.0.0.0/0' port port=2222 protocol=tcp accept"
    sudo firewall-cmd --permanent --remove-service=ssh
    sudo firewall-cmd --reload
    sudo systemctl restart sshd

EOF
}

resource "aws_launch_template" "webapp_launch_template" {
  name_prefix    = "webapp_launch_template"
  image_id       = "ami-063d43db0594b521b"
  instance_type  = "t2.micro"
  key_name       = "vockey"

  depends_on = [aws_db_instance.mysql_db]

  network_interfaces {
    associate_public_ip_address = true
    subnet_id                   = aws_subnet.subnet_private_a.id
    security_groups              = [aws_security_group.sg_appweb.id]
  }

  tag_specifications {
    resource_type = "instance"
    tags = {
      Name      = "WebApp"
      terraform = "True"
    }
  }

  user_data = base64encode(local.webapp_user_data)
}

resource "aws_autoscaling_group" "webapp_autoscaling_group" {
  name            = "webapp-autoscaling-group"
  launch_template {
    id = aws_launch_template.webapp_launch_template.id
  }
}
```

```
    version = "$Latest"
  }

  min_size           = 1
  max_size           = 3
  desired_capacity   = 2

  vpc_zone_identifier = [aws_subnet.subnet_private_a.id,
aws_subnet.subnet_private_b.id]

  target_group_arns = [aws_lb_target_group.obligatorio_target_group.arn]

  health_check_type      = "EC2"
  health_check_grace_period = 300

  lifecycle {
    create_before_destroy = true
  }

  depends_on = [aws_launch_template.webapp_launch_template]
}

resource "null_resource" "copy_vockey_to_jump" {
  connection {
    type      = "ssh"
    user      = "ec2-user" # Usuario del Jump Server
    private_key = file("C:/clave.pem") # Clave para acceder al Jump Server
    host       = aws_instance.jump_server.public_ip
    port      = 2222 # Cambia si tu puerto SSH es diferente
  }

  provisioner "file" {
    source      = "C:/clave.pem" # Ruta local del archivo voykey.pem
    destination = "/home/ec2-user/.ssh/clave.pem" # Destino en el Jump Server
  }

  provisioner "remote-exec" {
    inline = [
      "chmod 600 /home/ec2-user/.ssh/clave.pem"
    ]

    ## Desde el jump-server se ingresa a los demas servidores para su gestion:
    ## "ssh -i /home/ec2-user/.ssh/clave.pem ec2-user@IP PRIVADA"
  }
}
```

9.3 Bibliografía:

1. Sitios Internet:

- <https://developers.cloudflare.com/waf/custom-rules/use-cases/>
- [Wazuh documentation](#)
- [Como instalar Wordpress desde la línea de comandos de Linux - Guías y Tutoriales - Hostinglabs](#)
- [Enhancing SSH Security with Two-Factor Authentication \(2FA\) via PAM and Google Authenticator | by Prateek Malhotra | Medium](#)
- <https://registry.terraform.io/providers/hashicorp/aws/latest/docs>
- <https://github.com/ComplianceAsCode/content/blob/master/README.md>
- <https://github.com/adandrea8/Obligatorio-SRD>

2. Plataforma ORT Aulas / Materiales del curso
3. Consultas con el docente Mauricio Campiglia
4. Clases grabadas en Plataforma ORT / Aulas

9.4 Declaración de autoría

Este anexo trata sobre cómo redactar la declaración de autoría. Debe constar del siguiente texto:

Nosotros, Alexis D'Andrea y Nicolas Martins, declaramos que el trabajo que se presenta en esa obra es de nuestra propia mano. Podemos asegurar que:

- La obra fue producida en su totalidad mientras realizábamos Obligatorio del Sexto Semestre de Seguridad en Redes y Datos.
- Cuando hemos consultado el trabajo publicado por otros, lo hemos atribuido con claridad;
- Cuando hemos citado obras de otros, hemos indicado las fuentes. Con excepción de estas citas, la obra es enteramente nuestra;
- En la obra, hemos acusado recibo de las ayudas recibidas;
- Cuando la obra se basa en trabajo realizado conjuntamente con otros, hemos explicado claramente qué fue contribuido por otros, y qué fue contribuido por nosotros;
- Ninguna parte de este trabajo ha sido publicada previamente a su entrega, excepto donde se han realizado las aclaraciones correspondientes.

Firma:

Firma:



Alexis D'Andrea



Nicolas Martins