REPORT TECNICO

PowerShell, IoC, Nmap e Analisi SQL Injection

Redatto da: *Nicolò Calì Cybersecurity Student*

Data: *20/02/2026*

Oggetto: Amministrazione di sistema, network scanning e analisi di vulnerabilità web.

1. Introduzione

Il presente report documenta l'attività pratica focalizzata su molteplici aspetti della sicurezza informatica.

L'esercitazione è suddivisa in fasi principali:

- L'esplorazione delle funzionalità di automazione e gestione tramite **Windows PowerShell**.
- L'analisi di minacce basate su **IoC**.
- L'utilizzo dell'utility **Nmap** per la ricognizione di rete.
- l'analisi di un attacco di **SQL Injection** tramite l'ispezione di traffico di rete con **Wireshark**.

L'ambiente di test comprende un sistema operativo Windows e la macchina virtuale Linux CyberOps Workstation.

2. Utilizzo di Windows PowerShell

2.1 Esplorazione dei comandi di base

La prima fase dell'attività si è concentrata sul confronto tra l'interprete classico **Prompt dei Comandi** e l'ambiente avanzato **Windows PowerShell**. Eseguendo comandi di uso comune come "`dir`", "`ping`", "`cd`" e "`ipconfig`" in entrambi gli applicativi, si è potuto verificare il comportamento e la risposta del sistema operativo.

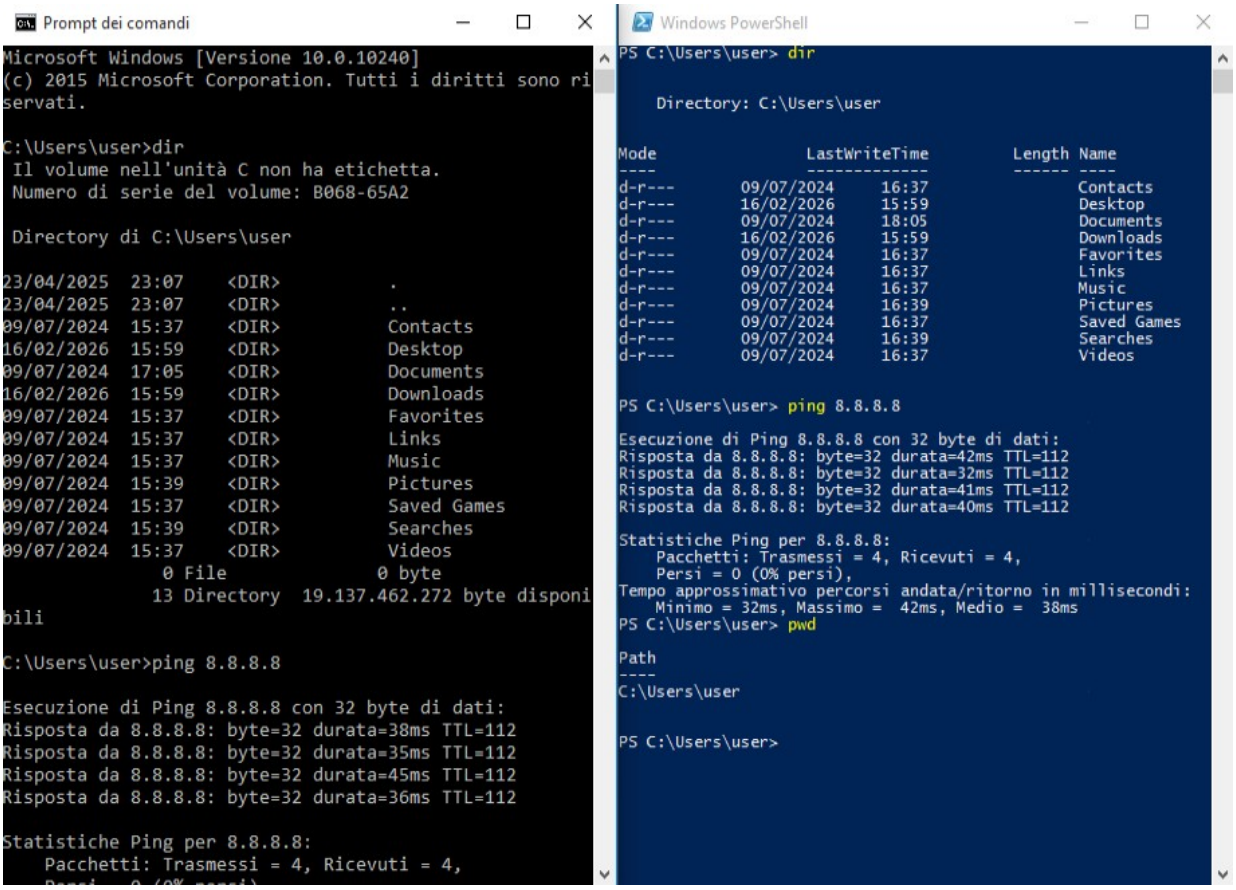

Fig. 1 - Confronto dell'output del comando dir tra Prompt dei Comandi e PowerShell.

D: Quali sono gli output del comando dir? Quali sono i risultati (degli altri comandi)?

R: L'output del comando `dir` in PowerShell **differisce** da quello del Prompt dei Comandi in quanto restituisce dettagli aggiuntivi come gli attributi e i permessi (Mode), oltre alla data di ultima modifica, la lunghezza e il nome.

Gli altri comandi come `ping` restituiscono output visivamente simili.

2.2 Alias e Cmdlet

PowerShell utilizza comandi nativi chiamati "**cmdlet**", strutturati secondo la sintassi **Verbo-Nome**. Per mantenere la retrocompatibilità e facilitare l'uso, utilizza degli "Alias" che mappano i vecchi comandi DOS ai nuovi cmdlet.

Tramite un'apposita verifica, si è identificato il comando sottostante a "dir".

2.3 Analisi della Rete con Netstat

Successivamente, si è esplorato il comando **netstat** per l'analisi delle connessioni di rete e delle tabelle di routing. Si è eseguito "**`netstat -h`**" per consultare la manualistica delle opzioni, e "**`netstat -r`**" per visualizzare le rotte attive del sistema host.

```
PS C:\Users\user> get-Alias dir

CommandType     Name                                               Version    Source
-----------     ----                                               -------    ------
Alias           dir -> Get-ChildItem


PS C:\Users\user> netstat -r
===========================================================================
Elenco interfacce
  4...08 00 27 64 c3 8d ......Intel(R) PRO/1000 MT Desktop Adapter
  1...........................Software Loopback Interface 1
  6...00 00 00 00 00 00 00 e0 Microsoft ISATAP Adapter
  5...00 00 00 00 00 00 00 e0 Microsoft Teredo Tunneling Adapter
===========================================================================

IPv4 Tabella route
===========================================================================
Route attive:
     Indirizzo rete            Mask          Gateway     Interfaccia Metrica
          0.0.0.0          0.0.0.0      192.168.0.1    192.168.0.177     10
        127.0.0.0        255.0.0.0         On-link         127.0.0.1    306
        127.0.0.1  255.255.255.255         On-link         127.0.0.1    306
  127.255.255.255  255.255.255.255         On-link         127.0.0.1    306
      192.168.0.0    255.255.255.0         On-link     192.168.0.177    266
    192.168.0.177  255.255.255.255         On-link     192.168.0.177    266
    192.168.0.255  255.255.255.255         On-link     192.168.0.177    266
        224.0.0.0        240.0.0.0         On-link         127.0.0.1    306
        224.0.0.0        240.0.0.0         On-link     192.168.0.177    266
  255.255.255.255  255.255.255.255         On-link         127.0.0.1    306
  255.255.255.255  255.255.255.255         On-link     192.168.0.177    266
===========================================================================
Route permanenti:
  Nessuna
```

Fig. 2 - Identificazione del cmdlet per "dir" e visualizzazione della tabella di routing IPv4.

D: Qual è il comando PowerShell per dir?

R: Il comando PowerShell per dir è **Get-ChildItem**

D: Qual è il gateway IPv4?

R: Il gateway IPv4 è **192.168.0.1**

2.4 Correlazione tra connessioni di rete e processi (PID)

Al fine di identificare quali applicazioni stiano generando traffico di rete o siano in ascolto su specifiche porte, si è eseguita un'istanza di PowerShell con privilegi elevati. Utilizzando il comando "**`netstat -ano`**", è stato possibile estrapolare il **PID** (*Process ID*) associato a ciascuna connessione **TCP/IP** attiva.

Successivamente, si è incrociato questo dato con la scheda Dettagli di **Gestione Attività** (*Task Manager*).

Selezionando un PID specifico e ispezionandone le Proprietà, è possibile eseguire un'operazione di base per l'identificazione di potenziali processi malevoli o anomali.

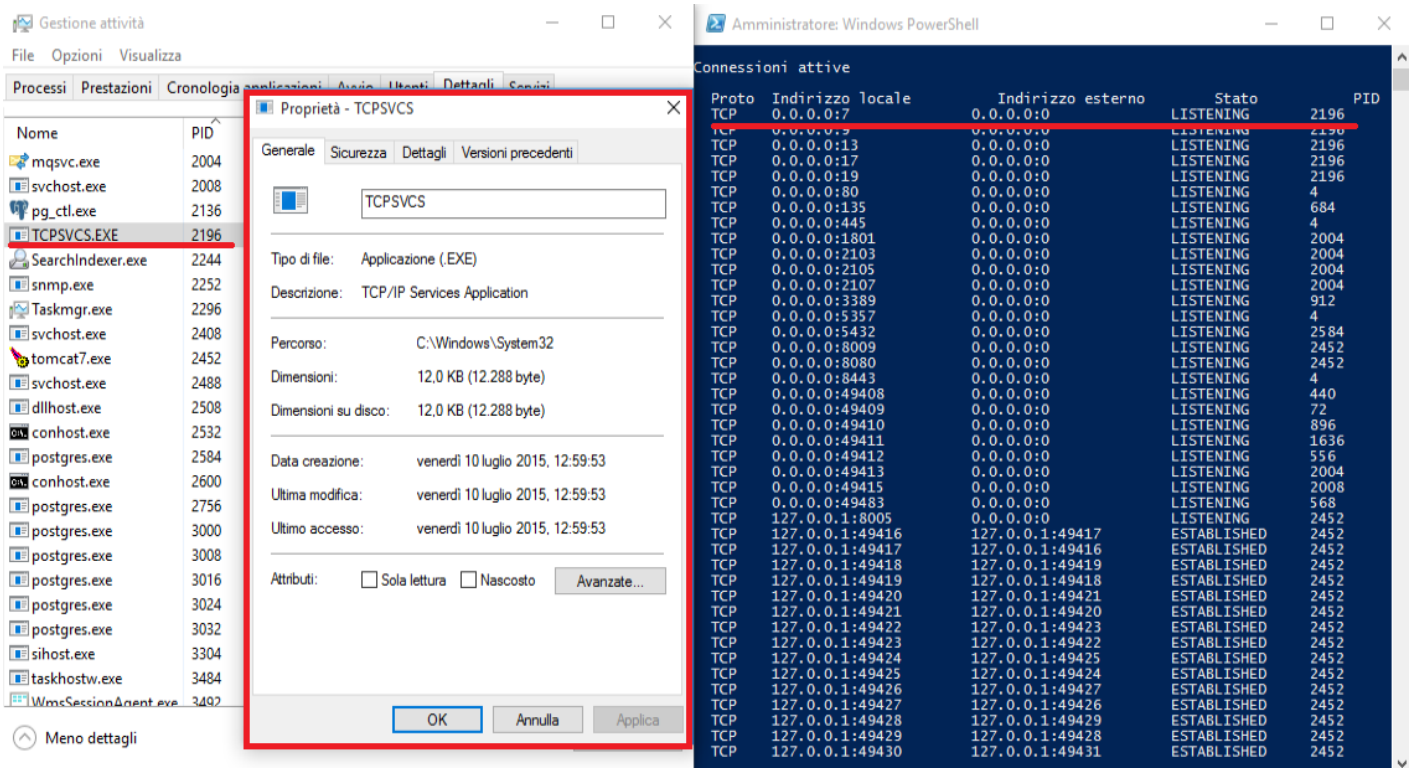

Fig. 3 - Associazione tra connessione di rete e processo applicativo tramite PID in Task Manager.

D: Quali informazioni puoi ottenere dalla scheda Dettagli e dalla finestra di dialogo Proprietà per il PID selezionato?

R:

- Dalla scheda **Dettagli** è possibile identificare il nome dell'eseguibile, l'utente che lo ha lanciato e il consumo di risorse.
- Dalla finestra **Proprietà** si ottengono dati cruciali per l'analisi di sicurezza, come il percorso fisico del file su disco (*Location*), le dimensioni, le date di creazione/modifica e, soprattutto, la presenza di eventuali Firme Digitali per verificarne l'autenticità.

2.5 Automazione della gestione del sistema

A conclusione dell'esplorazione di PowerShell, si è testata la sua capacità di semplificare azioni che tramite GUI richiederebbero più passaggi.

Si è provveduto a popolare il Cestino di Windows con alcuni file di test, per poi procedere alla sua pulizia forzata tramite riga di comando utilizzando il cmdlet "**`clear-recyclebin`**".

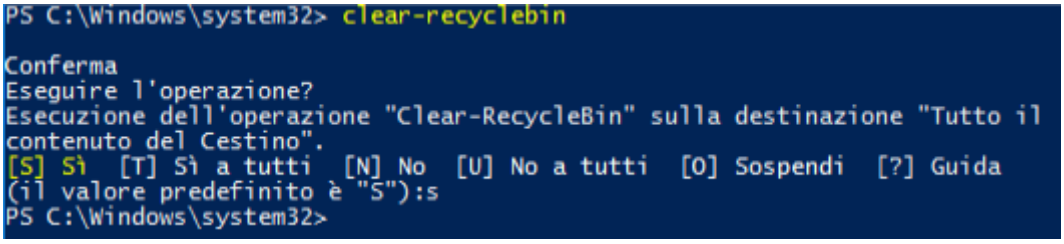

Fig. 4 - Esecuzione del cmdlet clear-recyclebin per l'eliminazione dei file dal cestino.

D: Cosa è successo ai file nel Cestino?

R: Il File è stato eliminato con successo.

D: Usando internet, ricerca comandi PowerShell che potresti usare per semplificare i tuoi compiti come analista di sicurezza. Registra le tue scoperte.

R: Ecco alcuni comandi PowerShell che potrebbero essere utili per un analista:

- **Get-WinEvent**: essenziale per l'estrazione e l'analisi rapida dei log di sicurezza di Windows (*es. tentativi di login anomali*).
- **Get-Process**: utile per fare l'inventario dei processi attivi e individuare eventuali software malevoli in esecuzione.
- **Get-LocalUser**: Mostra la lista degli account locali sul PC. Un analista lo usa per verificare se un attaccante ha creato account "nascosti" o non autorizzati per mantenere l'accesso. (*backdoor access*).

3. Analisi di Indicatori di Compromissione (IoC)

3.1 Vettore di Infezione e Inganno Iniziale

L'analisi dinamica condotta tramite la **sandbox Any.run** mostra come l'attacco abbia origine dal download di file eseguibili "**Jvczfhe.exe**" e "**Muadnrd.exe**" da una repository pubblica su **GitHub**.

Sfruttare un sito con un'alta reputazione come GitHub è una tecnica mirata per aggirare i controlli di sicurezza e i firewall aziendali, che solitamente considerano questo dominio sicuro.

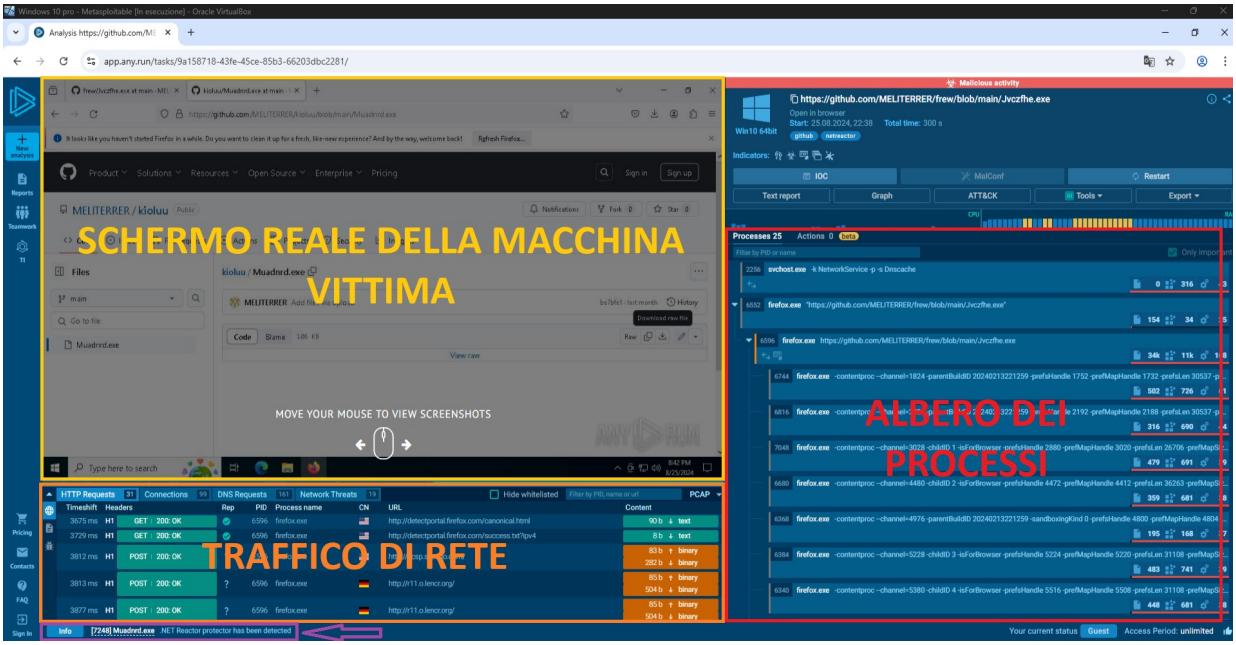

Fig. 5 - Panoramica dell'interfaccia di Any.run.

3.2 Falsi Errori e Albero dei Processi

Una volta eseguito il file, il sistema mostra all'utente un **finto messaggio di errore**, affermando che il file è danneggiato ("*The file is damaged and could not be opened*"). In realtà, questo finto crash è solo un diversivo.

Osservando l'albero dei processi, notiamo che il malware si attiva silenziosamente in background avviando una serie di comandi a cascata, mascherati da processi legittimi di sistema come **Werfault.exe** (*il segnalatore di errori di Windows*).

Fig. 6 - Processi.

3.3 Tecniche di Evasione e Ricognizione del Sistema

Per evitare di essere bloccato dai sistemi di sicurezza automatizzati, il malware utilizza tecniche di evasione specifiche.

Apre il prompt dei comandi (**cmd.exe**) e lancia il comando "**timeout.exe**" per mettere in pausa le proprie attività.

Fig. 7 - Rilevamento delle tecniche evasive.

3.4 Installazione del Payload e Connessioni Anomale

Terminata la fase di evasione, l'attacco entra nel vivo.

Il sistema di analisi rileva un'attività di rete sospetta: il processo cerca di comunicare con l'esterno utilizzando una porta non standard (es. la porta 7702), un comportamento tipico delle **backdoor** per comunicare con il server dell'attaccante.

Fig. 8 - Grafo di esecuzione che evidenzia i processi malevoli identificati come minaccia (Installutil.exe e muadnrd.exe).

Fig. 9 - Rilevamento di traffico di rete anomalo su porte non standard (potenziale backdoor).

L'analisi dinamica rivela l'esecuzione di un malware complesso che sfrutta la reputazione di GitHub per aggirare i filtri di rete.

Il software utilizza tecniche di inganno visivo (*falsi messaggi di errore*) e tecniche di evasione (*timeout.exe*).

L'obiettivo finale, evidenziato dall'uso anomalo di **Installutil.exe** e da connessioni su porte non standard, è l'installazione nascosta di un payload malevolo e la creazione di un canale di comunicazione esterno (**backdoor**).

4. Esplorazione e Ricognizione di Rete (Nmap)

4.1 Analisi della manualistica di Nmap

La prima fase del Bonus ha previsto l'esplorazione del manuale di sistema dell'utility **Nmap** all'interno della **VM CyberOps**. Tramite la funzione di ricerca, si sono analizzate le definizioni e le opzioni di base dello strumento.

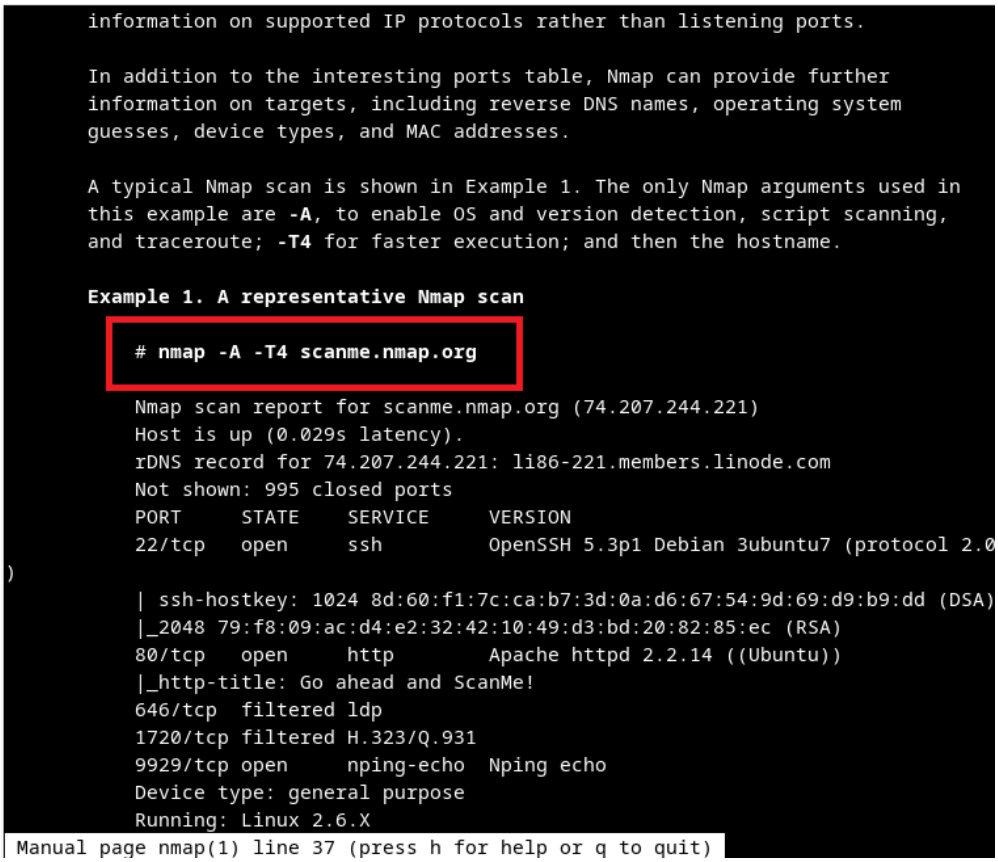

Fig. 10 - Esecuzione manuale Nmap tramite comando man nmap

D: Cos'è **Nmap**? Per cosa viene usato nmap?

R: Nmap è uno strumento **open source** per l'esplorazione della rete e l'auditing di sicurezza. Viene utilizzato per determinare quali host sono attivi, quali servizi offrono, i sistemi operativi in esecuzione e altre caratteristiche di rete.

D: Cosa fa l'opzione **-A**? Cosa fa l'opzione **-T4**?

R: L'opzione -A abilita il rilevamento avanzato, includendo l'identificazione del sistema operativo, la versione dei servizi, lo script scanning e il traceroute. L'opzione -T4 configura un modello di esecuzione più rapido.

D: Qual è il comando nmap usato nell'Esempio 1?

R: Il comando è **`nmap -A -T4 scanme.nmap.org`**

4.2 Scansione del Localhost

Dopo aver compreso la sintassi di base, si è proceduto con una scansione attiva del localhost (**127.0.0.1**) utilizzando il comando "**`nmap -A -T4 localhost`**".

Questa operazione permette di identificare i servizi esposti localmente prima di interagire con la rete esterna.

```
[analyst@secOps ~]$ nmap -A -T4 localhost
Starting Nmap 7.97 ( https://nmap.org ) at 2026-02-20 06:31 -0500
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 tcp ports (conn-refused)
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 4
|      vsFTPd 3.0.5 - secure, fast, stable
|_End of status
22/tcp open  ssh     OpenSSH 10.0 (protocol 2.0)
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 11.49 seconds
[analyst@secOps ~]$
```

Fig. 11 - Scansione delle porte aperte in localhost tramite Nmap.

D: Quali porte e servizi sono aperti?

R: Dalla scansione risultano aperte la porta 21/tcp (**servizio FTP**) e la porta 22/tcp (**servizio SSH**).

D: Per ognuna delle porte aperte, registra il software che fornisce i servizi.

R: Il servizio FTP sulla porta 21 è fornito da "**vsftpd 3.0.5”**. Il servizio SSH sulla porta 22 è fornito da "**OpenSSH 10.0**".

4.3 Scansione della Rete Locale (LAN)

Per mappare gli host attivi all'interno della rete locale, si è prima determinato l'indirizzo IP della macchina in uso e la relativa subnet mask per mezzo del comando "**`ip a`**". Successivamente, si è lanciata una scansione sull'intero range di indirizzi per individuare altri dispositivi e i servizi da essi esposti tramite il comando "**`nmap -A -T4 10.0.2.0/24`**".

Fig. 12 - scansione dell'intera sottorete.

D: A quale rete appartiene la tua VM?

R: L'indirizzo IP della VM è **10.0.2.15** con subnet mask **255.255.255.0** (*cidr /24*). Pertanto, la VM appartiene alla rete 10.0.2.0/24.

D: Quanti host sono attivi? Dai risultati di Nmap, elenca gli indirizzi IP e alcuni dei servizi.

R: Dalla scansione dell'intera subnet è emerso un solo host attivo (**1 host up**), corrispondente all'indirizzo IP della macchina stessa (10.0.2.15). I servizi rilevati su questo host sono **FTP** e **SSH**.

4.4 Scansione di un Server Remoto (scanme.nmap.org)

L'ultimo passaggio dell'esplorazione ha riguardato la scansione di un bersaglio remoto autorizzato. Navigando all'indirizzo **scanme.nmap.org**, si è appreso che il sito è stato configurato appositamente per permettere agli utenti di testare **Nmap** in un ambiente sicuro e legale.

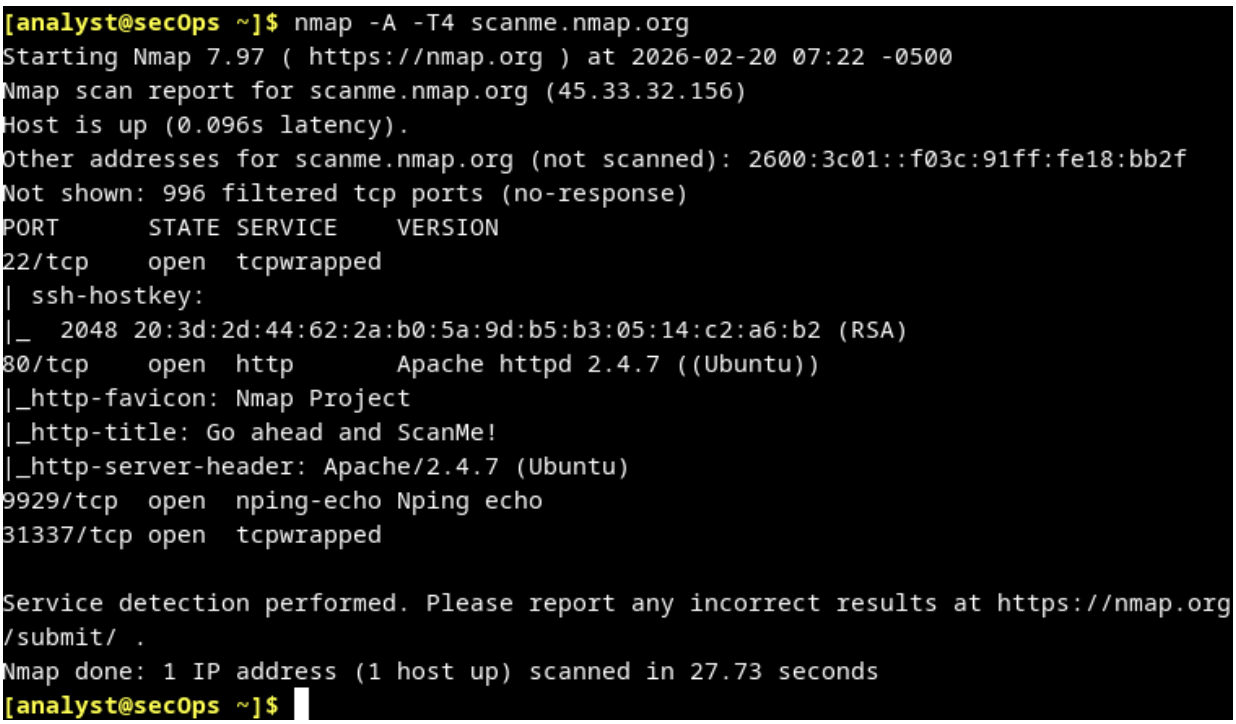

Fig. 13 - Scansione Nmap sul dominio remoto scanme.nmap.org.

D: Quali porte e servizi sono aperti?

R: Risultano aperte le porte **22** (tcpwrapped/SSH), **80** (http, Apache 2.4.7), **9929** (nping-echo) e **31337** (tcpwrapped).

D: Quali porte e servizi sono filtrati?

R: Nmap segnala che **996 porte** TCP risultano filtrate, indicando la probabile presenza di un firewall a protezione del server.

D: Qual è l'indirizzo IP del server?

R: L'indirizzo IP risolto per scanme.nmap.org è **45.33.32.156**.

D: Qual è il sistema operativo?

R: L'enumerazione dei servizi suggerisce che il server utilizzi una distribuzione **Linux Ubuntu**.

D: Come può Nmap aiutare con la sicurezza della rete? Come può Nmap essere usato da un attore malevolo come strumento nefasto?

R: Nmap è fondamentale per la sicurezza difensiva (**Blue Team**) in quanto permette agli amministratori di mappare la propria rete, visualizzare i dispositivi connessi,

verificare la conformità dei firewall e individuare porte e servizi inavvertitamente lasciate aperte.

Di contro, un attore malevolo utilizza le medesime funzioni nella fase di "**Ricognizione**" per scoprire l'architettura della rete bersaglio e identificare servizi vulnerabili o non aggiornati da sfruttare per un attacco.

5. Analisi di un Attacco SQL Injection

5.1 Ispezione del traffico di rete con Wireshark

L'ultimo esercizio ha riguardato l'analisi forense di un attacco a un database MySQL tramite l'ispezione di un file di cattura pacchetti (*.pcap*) utilizzando **Wireshark**.

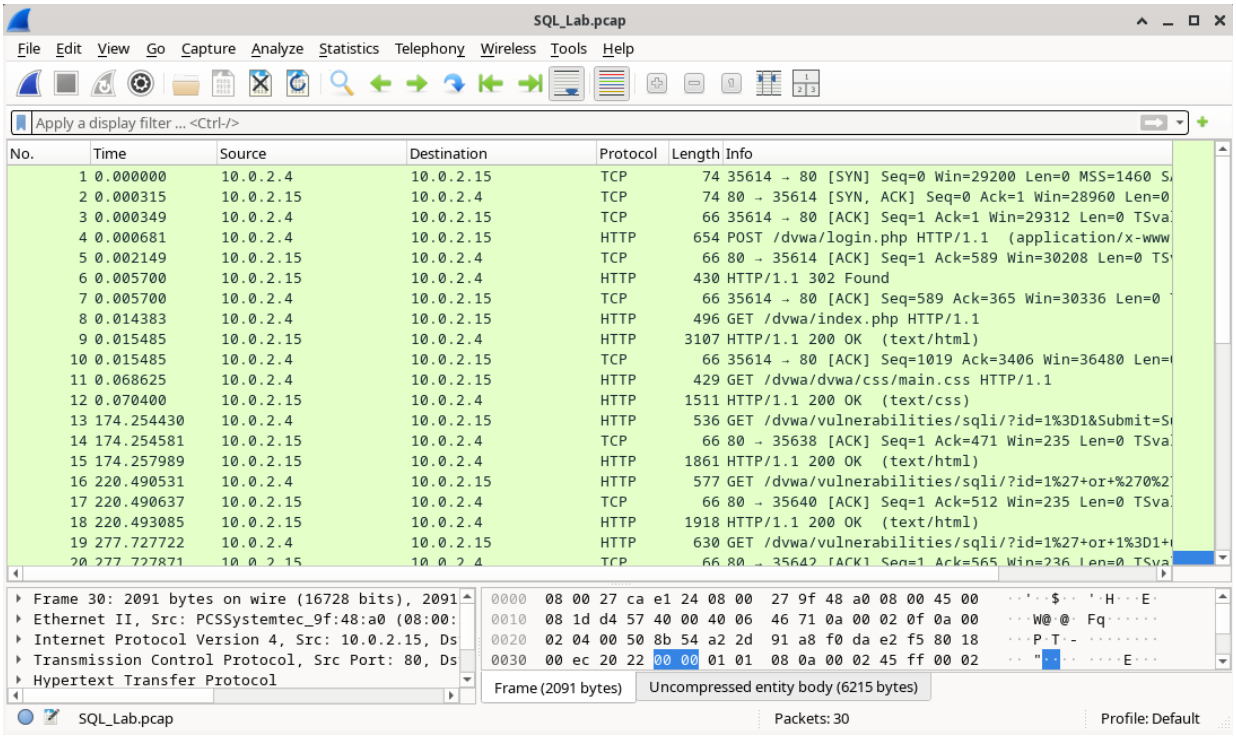

Fig. 14 - Schermata della raccolta dei logs di wireshark.

Aperto il file "SQL_Lab.pcap", si è proceduto all'individuazione degli attori coinvolti analizzando le richieste HTTP.

```
10.0.2.4                    10.0.2.15
```

D: Quali sono i due indirizzi IP coinvolti in questo attacco di SQL injection in base alle informazioni visualizzate?

R: I due indirizzi IP coinvolti sono **10.0.2.4** (Attaccante) e **10.0.2.15** (Vittima).

5.2 Conferma della Vulnerabilità

Per comprendere le dinamiche dell'attacco, si è proceduto alla ricostruzione dei **flussi TCP/HTTP**.

Esaminando la prima richiesta GET sospetta (pacchetto 13) tramite la funzione "**Follow HTTP Stream**", è stato possibile isolare il payload inserito dall'attaccante: la stringa logica "**1=1**".

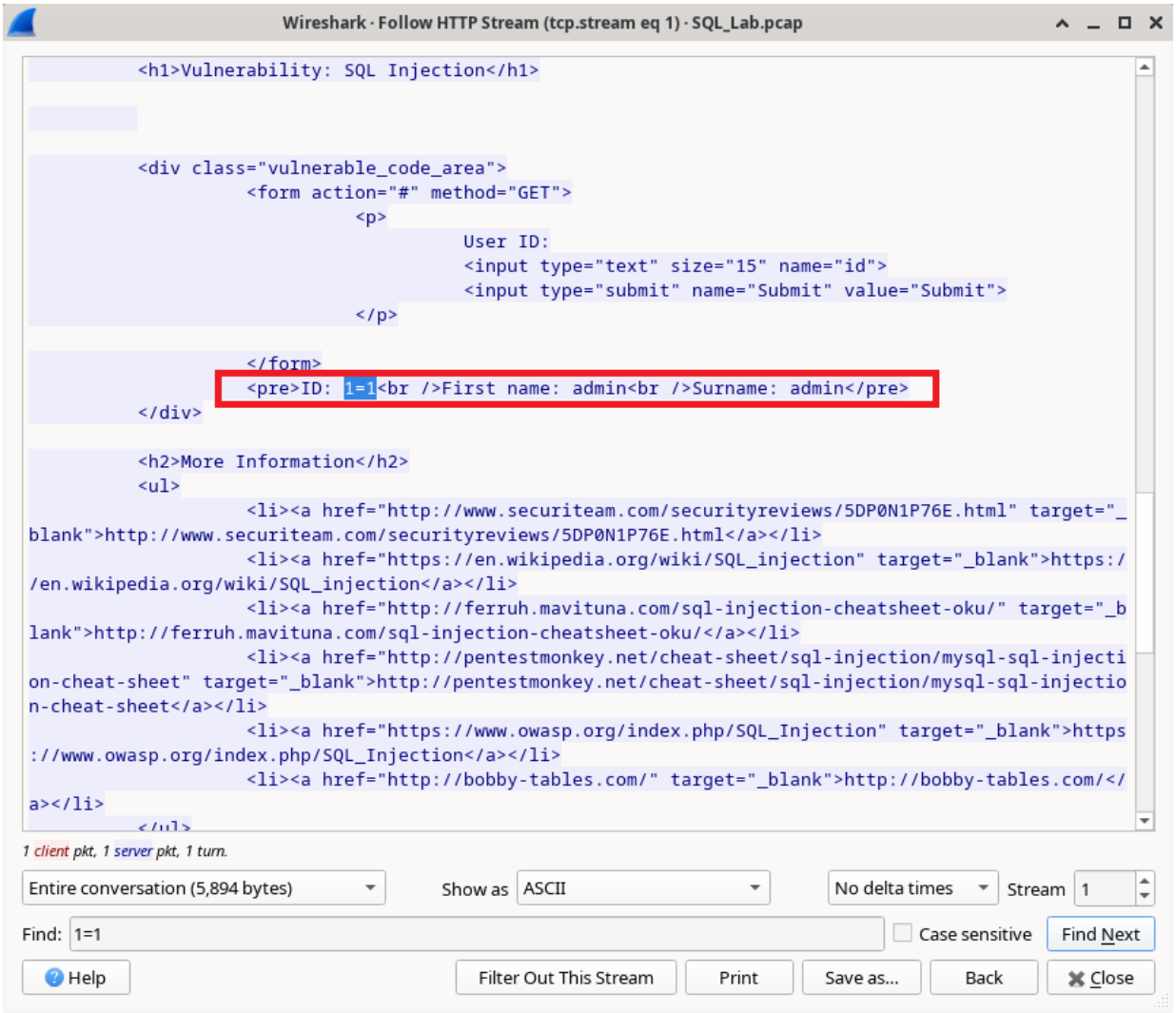

Fig. 15 - Ricostruzione del flusso HTTP e visualizzazione del primo tentativo di SQL Injection (1=1).

L'applicazione, risultata vulnerabile a causa della mancata **sanitizzazione** degli input, ha interpretato la condizione tautologica come vera.

Di conseguenza, invece di restituire un errore di autenticazione, il server ha risposto esponendo i dati del primo record presente nella tabella interrogata (*First name: admin, Surname: admin*).

5.3 Estrazione di Informazioni di Sistema

Una volta confermata la vulnerabilità, l'attaccante ha proceduto con l'enumerazione del database. Analizzando i successivi flussi HTTP (*es. pacchetto 22*), si evince l'uso di comandi SQL avanzati come "**union select null, version()**".

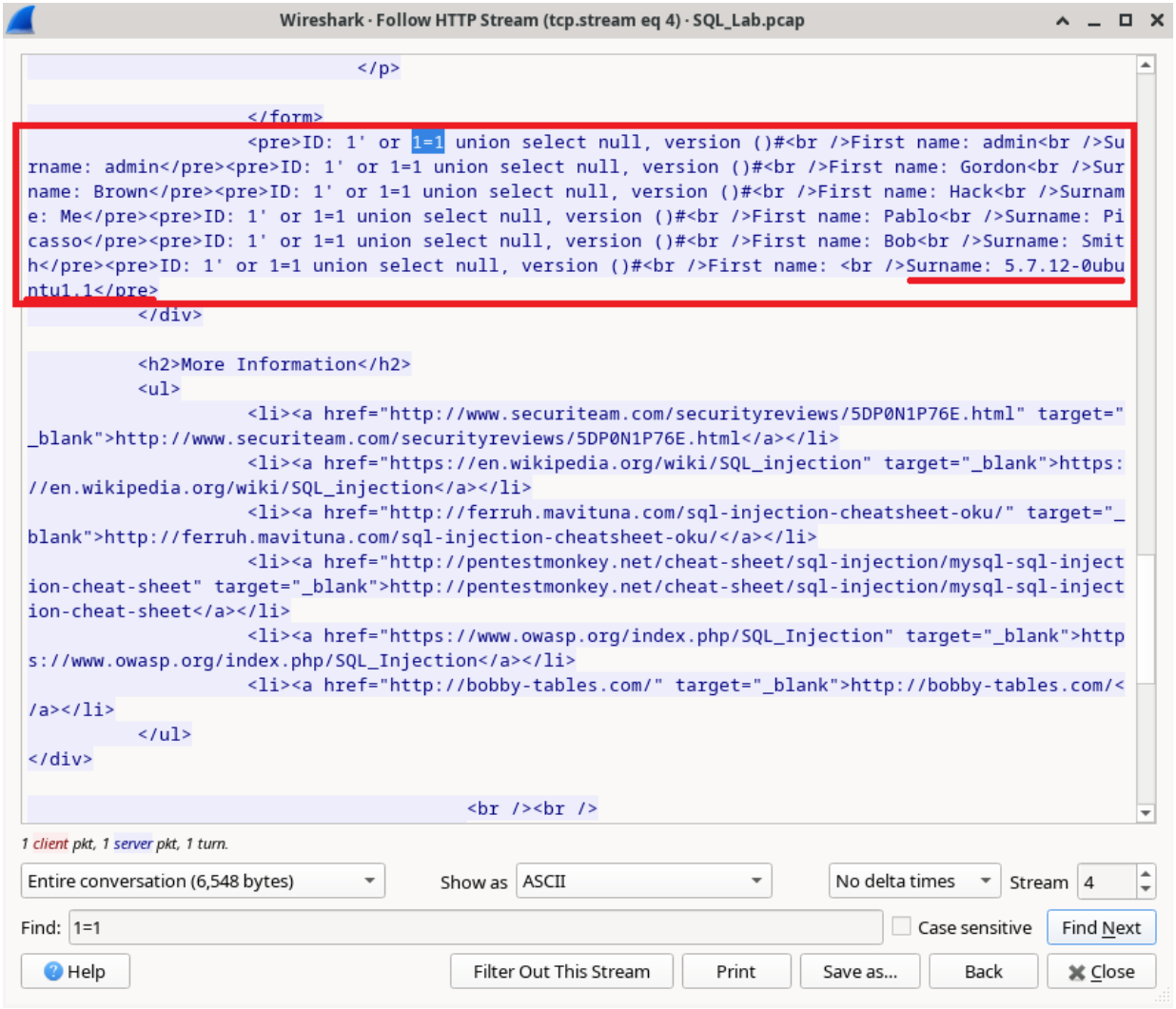

Fig. 16 - Estrazione della versione del database tramite SQL Injection.

Sfruttando questa tecnica, l'attore malevolo è riuscito a far comparire a schermo l'identificatore di versione del sistema di backend, che risulta essere "**5.7.12-0ubuntu1.1**".

Conoscere la versione esatta del software è una fase critica della "**Ricognizione**", poiché permette di individuare specifiche vulnerabilità note (**CVE**) associate a quella macchina.

5.4 Esfiltrazione delle Credenziali

La fase conclusiva dell'attacco ha dimostrato l'impatto critico della vulnerabilità. Inserendo il comando "**1' or 1=1 union select user, password from users#**", l'attaccante ha costretto il database a eseguire il dump dell'intera tabella degli utenti.

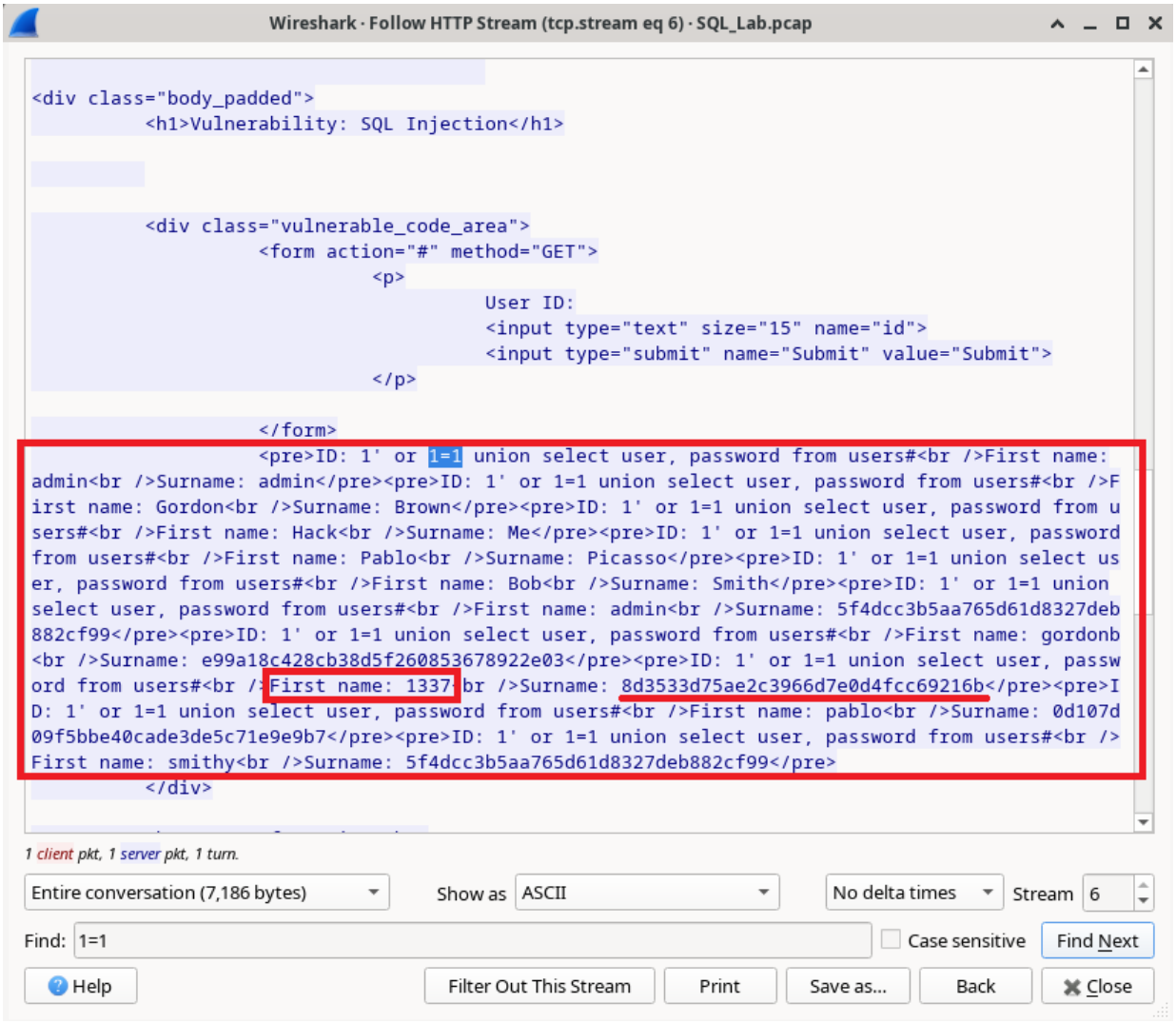

Fig. 17 - Dump delle credenziali (username e password hash) degli utenti del database.

D: Quale utente ha l'hash della password di **8d3533d75ae2c3966d7e0d4fcc69216b**?

R: Analizzando l'output dell'iniezione SQL, si evince che l'hash specificato appartiene all'utente "**1337**".

D: Qual è la password in chiaro?

R: Utilizzando il tool CrackStation da browser ho scoperto che la password in chiaro è **charley**.

Fig.18 - Output di CrackStation.

5.5 Riflessioni Finali e Contromisure

A valle dell'analisi dell'attacco, è fondamentale definire i rischi e le strategie di mitigazione associati all'uso di database SQL.

D: Qual è il rischio che le piattaforme utilizzino il linguaggio SQL?

R: Il rischio primario si concretizza quando **l'applicazione non effettua una corretta validazione e sanitizzazione dell'input utente.**

Questo permette agli attaccanti di iniettare istruzioni SQL arbitrarie, ricevendo risposte non autorizzate dal database. Ciò può portare all'**esfiltrazione** di dati sensibili, alla **manomissione** dei record, alla **falsificazione** di identità o ad altre azioni malevole, la cui gravità finale dipende dagli obiettivi dell'aggressore.

D: Quali sono 2 metodi o passaggi che possono essere adottati per prevenire gli attacchi di SQL injection?

R: Per mitigare efficacemente il rischio di SQL Injection e proteggere l'infrastruttura, è raccomandato adottare le seguenti contromisure:

- Utilizzo di **Query Parametrizzate**: separano nettamente il codice SQL dai dati forniti dall'utente, impedendo che l'input venga interpretato come comando eseguibile.
- Implementazione di **filtri di input** e **WAF**: filtrare rigorosamente l'input dell'utente e posizionare un Web Application Firewall per intercettare pattern di attacco noti, unito alla disabilitazione di funzionalità e capacità non necessarie del database.
-

6. Conclusioni

Il presente laboratorio ha permesso di consolidare competenze operative fondamentali nell'ambito della Cybersecurity.

Attraverso l'uso pratico di strumenti come **PowerShell**, **Any.run**, **Nmap** e **Wireshark**, è stato possibile affrontare scenari reali che spaziano dall'amministrazione e automazione di sistema alla ricognizione di rete, fino all'analisi forense di malware e vulnerabilità web (**SQL Injection**).

Questa esperienza ha dimostrato concretamente come la comprensione profonda delle metodologie offensive (*Red Team*) sia un requisito indispensabile per poter implementare, analizzare e gestire difese efficaci (*Blue Team*).