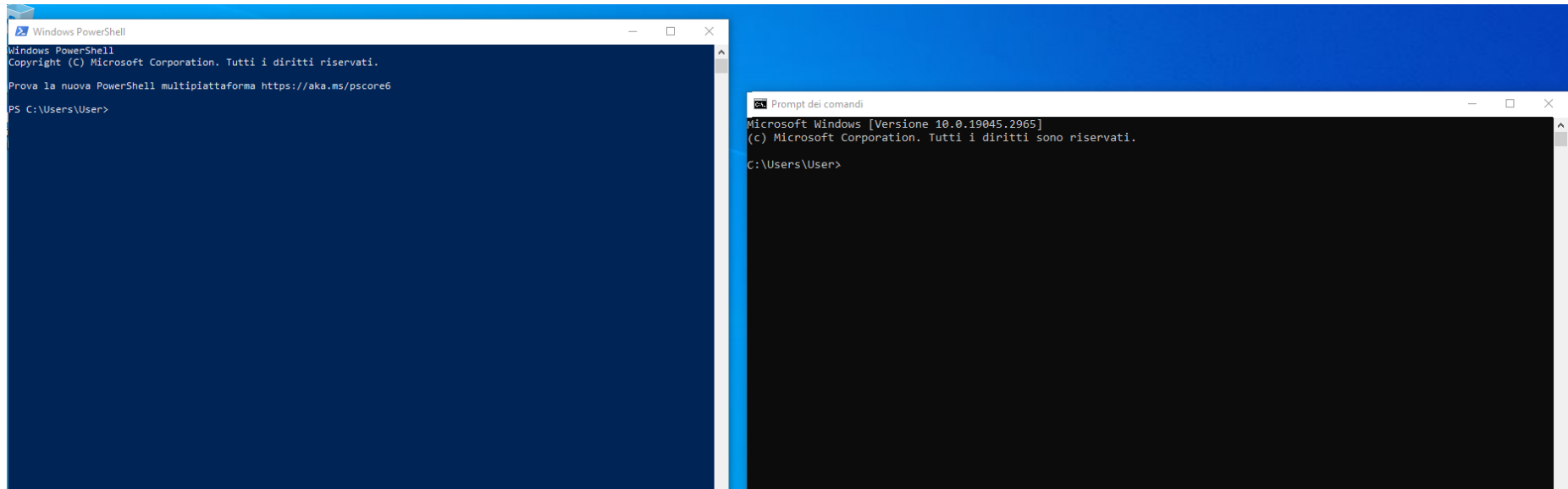


## Consegna S11L5

### Laboratorio 1 – Windows PowerShell

Prima cosa da fare è cliccare su “Start” e cercare “powershell”, una volta trovata avviamo l’applicazione.

Ripetiamo la stessa cosa con il “command prompt”.



Procediamo con l'esplorazione dei comandi, cominciamo con "dir". Quali sono i suoi output?

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

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

PS C:\Users\User> dir

Directory: C:\Users\User


Mode                LastWriteTime         Length Name
----                -
d-r-- 08/09/2024 23:19             30 3D Objects
d-r-- 08/09/2024 23:19             10  Contacts
d-r-- 08/09/2024 23:19             10  Desktop
d-r-- 08/09/2024 23:19             10  Documents
d-r-- 08/09/2024 23:19             10  Downloads
d-r-- 08/09/2024 23:19             10  Favorites
d-r-- 08/09/2024 23:19             10  Links
d-r-- 08/09/2024 23:19             10  Music
d-r-- 22/01/2025 12:04             10  OneDrive
d-r-- 08/09/2024 23:22             10  Pictures
d-r-- 08/09/2024 23:19             10  Saved Games
d-r-- 08/09/2024 23:21             10  Searches
d-r-- 08/09/2024 23:19             10  Videos

PS C:\Users\User>
```

```
cmd Prompt dei comandi
Microsoft Windows [Versione 10.0.19045.2965]
(c) Microsoft Corporation. Tutti i diritti sono riservati.

C:\Users\User>dir
Il volume nell'unità C non ha etichetta.
Numero di serie del volume: 76FF-0D4F

Directory di C:\Users\User

22/01/2025 12:04 <DIR>      .
22/01/2025 12:04 <DIR>      ..
08/09/2024 22:19 <DIR>      3D Objects
08/09/2024 22:19 <DIR>      Contacts
08/09/2024 22:19 <DIR>      Desktop
08/09/2024 22:19 <DIR>      Documents
08/09/2024 22:19 <DIR>      Downloads
08/09/2024 22:19 <DIR>      Favorites
08/09/2024 22:19 <DIR>      Links
08/09/2024 22:19 <DIR>      Music
22/01/2025 12:04 <DIR>      OneDrive
08/09/2024 22:22 <DIR>      Pictures
08/09/2024 22:19 <DIR>      Saved Games
08/09/2024 22:21 <DIR>      Searches
08/09/2024 22:19 <DIR>      Videos
           0 File             0 byte
          15 Directory 57.837.064.192 byte disponibili

C:\Users\User>
```

Entrambe le finestre elencano una lista di cartelle e file, con le informazioni su ogni elemento della lista, come il tipo, la dimensione del file, data e ora dell'ultima modifica.

Sia PowerShell che cmd mostrano risultati simili.

Proviamo altri comandi come ping, cd e ipconfig.

Ping:

```
PS C:\Users\User> ping 192.168.50.10
```

```
Esecuzione di Ping 192.168.50.10 con 32 byte di dati:  
Risposta da 192.168.50.10: byte=32 durata=1ms TTL=64  
Risposta da 192.168.50.10: byte=32 durata=1ms TTL=64  
Risposta da 192.168.50.10: byte=32 durata=1ms TTL=64
```

```
Statistiche Ping per 192.168.50.10:  
Pacchetti: Trasmessi = 3, Ricevuti = 3,  
Persi = 0 (0% persi),  
Tempo approssimativo percorsi andata/ritorno in millisecondi:  
Minimo = 1ms, Massimo = 1ms, Medio = 1ms  
Control-C  
PS C:\Users\User>
```

```
C:\Users\User>ping 192.168.50.10
```

```
Esecuzione di Ping 192.168.50.10 con 32 byte di dati:  
Risposta da 192.168.50.10: byte=32 durata=1ms TTL=64  
Risposta da 192.168.50.10: byte=32 durata<1ms TTL=64  
Risposta da 192.168.50.10: byte=32 durata<1ms TTL=64
```

```
Statistiche Ping per 192.168.50.10:  
Pacchetti: Trasmessi = 3, Ricevuti = 3,  
Persi = 0 (0% persi),  
Tempo approssimativo percorsi andata/ritorno in millisecondi:  
Minimo = 0ms, Massimo = 1ms, Medio = 0ms  
Control-C  
^C  
C:\Users\User>
```

cd:

```
PS C:\Users\User> cd Desktop  
PS C:\Users\User\Desktop>
```

```
C:\Users\User>cd Desktop
```

```
C:\Users\User\Desktop>_
```

Ipconfig:

```
PS C:\Users\User\Desktop> cd ..  
PS C:\Users\User> ipconfig
```

```
Configurazione IP di Windows
```

```
Scheda Ethernet Ethernet:
```

```
Suffisso DNS specifico per connessione:  
Indirizzo IPv4. . . . . : 192.168.50.30  
Subnet mask . . . . . : 255.255.255.0  
Gateway predefinito . . . . . : 192.168.50.1  
PS C:\Users\User>
```

```
Configurazione IP di Windows
```

```
Scheda Ethernet Ethernet:
```

```
Suffisso DNS specifico per connessione:  
Indirizzo IPv4. . . . . : 192.168.50.30  
Subnet mask . . . . . : 255.255.255.0  
Gateway predefinito . . . . . : 192.168.50.1
```

```
C:\Users\User>
```

Come si nota per tutti i comandi entrambe le finestre danno risultati simili.

Esploriamo ora i “cmdlets” in PowerShell

```
PS C:\Users\User> Get-Alias dir
```

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

Il comando “dir” sta a indicare le cartelle, in questo caso “Get-ChildItem”.

I “cmdlets” sono comandi specializzati usati in PowerShell, sono dei framework di automazione sviluppato da Microsoft. Ogni “cmdlet” esegue una singola funzione e può essere combinato con altri “cmdlet” per realizzare operazioni più complessi. Eccone alcuni:

- Get-Process: elenca i processi attualmente in esecuzione
- Set-Item: modifica il valore di un oggetto
- Get-Help: fornisce la guida e la documentazione per i comandi PowerShell

Ora vediamo “netstat” in PowerShell

“netstat -h” mostra le opzioni per ogni input.

```
PS C:\Users\User> netstat -h

Visualizza le statistiche del protocollo e le connessioni di rete TCP/IP correnti.

NETSTAT [-a] [-b] [-e] [-f] [-n] [-o] [-p proto] [-r] [-s] [-t] [-x] [-y] [interval]

-a          Visualizza tutte le connessioni e le porte di ascolto.
-b          Visualizza l'eseguibile coinvolto nella creazione di ogni connessione o
            porta di ascolto. In alcuni casi, host di eseguibili noti
            più componenti indipendenti e in questi casi il
```

“net-stat -r” mostra le routing table cone le route attive.

La voce “IPv4 gateway” indica appunto l’IP designato come gateway, in questo esempio tale indirizzo è “192.168.50.1”

```
PS C:\Users\User> netstat -r

=====
Elenco interfacce
10...08 00 27 96 c2 10 .....Intel(R) PRO/1000 MT Desktop Adapter
1.....Software Loopback Interface 1
=====

IPv4 Tabella route
=====
Route attive:
Indirizzo rete      Mask      Gateway      Interfaccia  Metrica
0.0.0.0             0.0.0.0    192.168.50.1  192.168.50.30 281
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
192.168.50.0        255.255.255.0  On-link      192.168.50.30 281
192.168.50.30       255.255.255.255  On-link      192.168.50.30 281
192.168.50.255     255.255.255.255  On-link      192.168.50.30 281
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      192.168.50.30 281
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      192.168.50.30 281
=====
Route permanenti:
Indirizzo rete      Mask      Indir. gateway  Metrica
0.0.0.0             0.0.0.0    192.168.50.1    Predefinito
=====

IPv6 Tabella route
=====
Route attive:
Interf Metrica Rete Destinazione      Gateway
1      331  ::1/128      On-link
1      331  ff00::/8     On-link
=====
Route permanenti:
Nessuna
```

In PowerShell avviato da amministratore possiamo vedere i processi dei protocolli TCP. Per fare questo utilizziamo il comando “netstat -abno”

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

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

PS C:\Windows\system32> netstat -abno

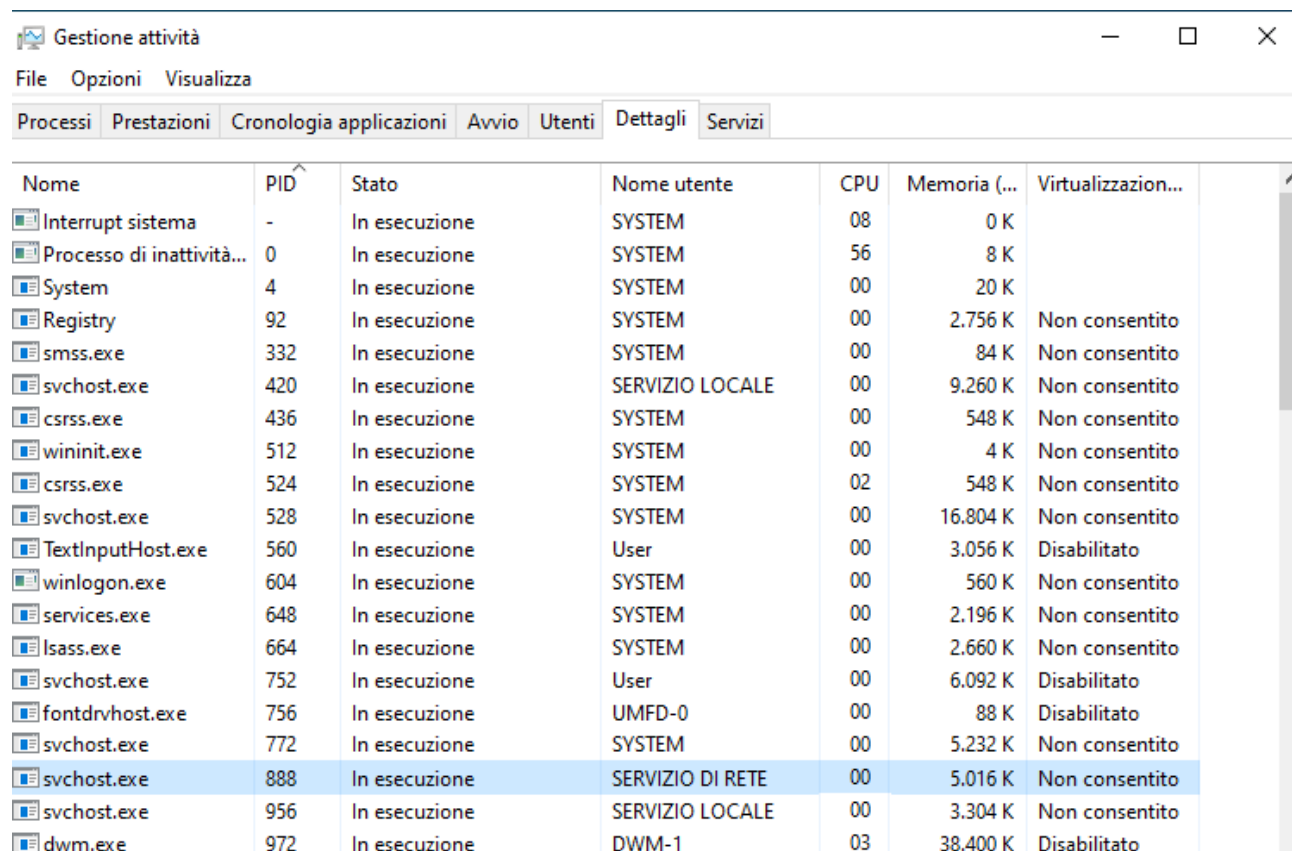
Connessioni attive

Proto Indirizzo locale      Indirizzo esterno      Stato      PID
TCP    0.0.0.0:135              0.0.0.0:0              LISTENING  888
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  1152
CDPSvc
[svchost.exe]
TCP    0.0.0.0:49664            0.0.0.0:0              LISTENING  664
[lsass.exe]
TCP    0.0.0.0:49665            0.0.0.0:0              LISTENING  512
Impossibile ottenere informazioni sulla proprietà
TCP    0.0.0.0:49666            0.0.0.0:0              LISTENING  1084
EventLog
[svchost.exe]
TCP    0.0.0.0:49667            0.0.0.0:0              LISTENING  528
Schedule
[svchost.exe]
TCP    0.0.0.0:49668            0.0.0.0:0              LISTENING  1972
[spoolsv.exe]
TCP    0.0.0.0:49669            0.0.0.0:0              LISTENING  648
Impossibile ottenere informazioni sulla proprietà
TCP    192.168.50.30:139        0.0.0.0:0              LISTENING  4
Impossibile ottenere informazioni sulla proprietà
TCP    [::]:135                 [::]:0                 LISTENING  888
RpcSs
[svchost.exe]
TCP    [::]:445                 [::]:0                 LISTENING  4
Impossibile ottenere informazioni sulla proprietà
TCP    [::]:49664               [::]:0                 LISTENING  664
[lsass.exe]
TCP    [::]:49665               [::]:0                 LISTENING  512
Impossibile ottenere informazioni sulla proprietà
TCP    [::]:49666               [::]:0                 LISTENING  1084
EventLog
[svchost.exe]
TCP    [::]:49667               [::]:0                 LISTENING  528
Schedule
[svchost.exe]
TCP    [::]:49668               [::]:0                 LISTENING  1972
```

Ora apriamo “Gestione Attività”, ci spostiamo in “Dettagli” e ordiniamo i “PID” in ordine crescente.

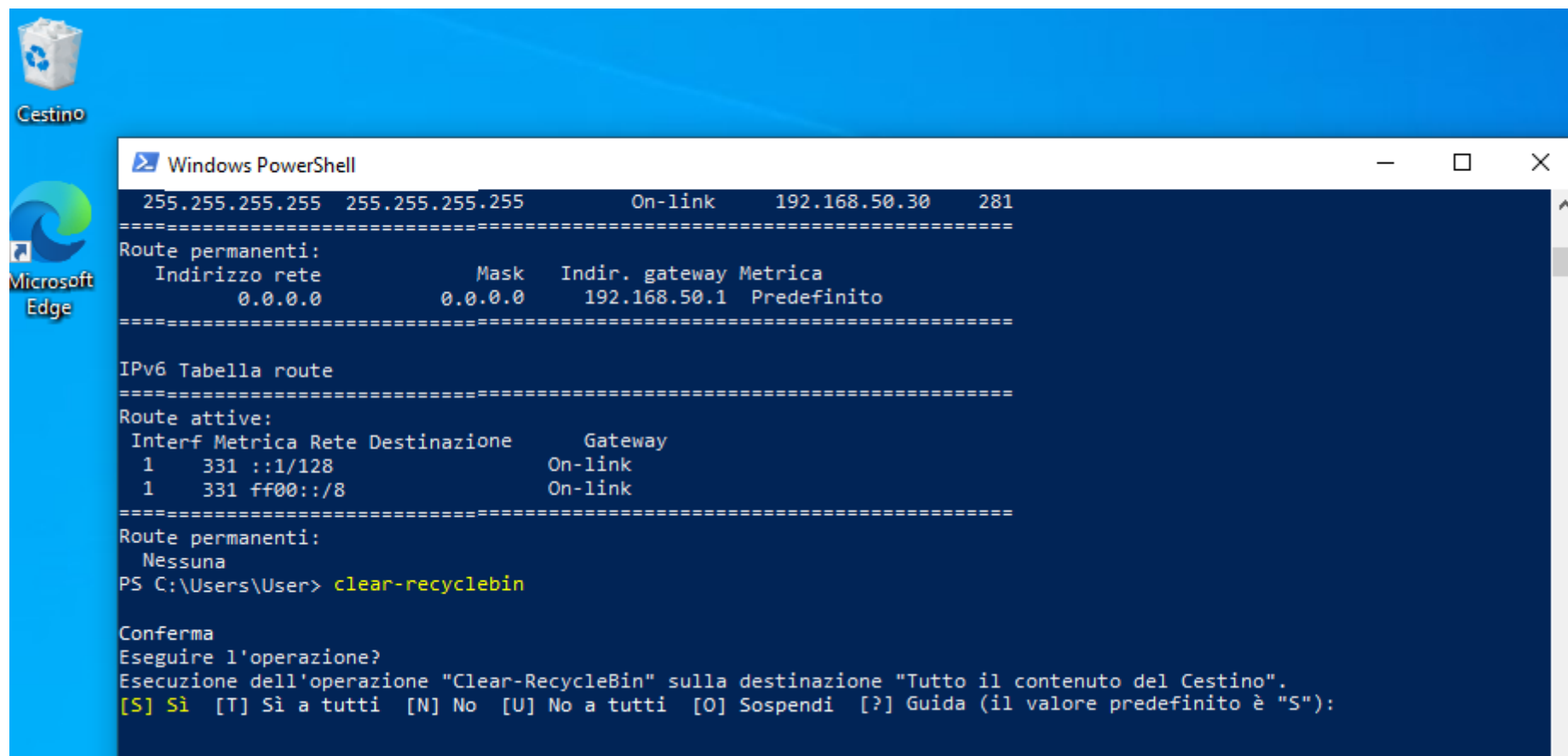
Identifichiamo il “PID” risultato dal comando “netstat -abno”, se vogliamo più dettagli clicchiamo il tasto destro per vedere le proprietà.

Le informazioni ottenute dal PID selezionato sono: PID 888 associato col processo “svchost.exe”. L'utilizzatore del processo è il Servizio di rete e sta utilizzando 5016K di memoria.



File Opzioni Visualizza						
Processi Prestazioni Cronologia applicazioni Avvio Utenti Dettagli Servizi						
Nome	PID	Stato	Nome utente	CPU	Memoria (...)	Virtualizzazione...
Interrupt sistema	-	In esecuzione	SYSTEM	08	0 K	
Processo di inattività...	0	In esecuzione	SYSTEM	56	8 K	
System	4	In esecuzione	SYSTEM	00	20 K	
Registry	92	In esecuzione	SYSTEM	00	2.756 K	Non consentito
smss.exe	332	In esecuzione	SYSTEM	00	84 K	Non consentito
svchost.exe	420	In esecuzione	SERVIZIO LOCALE	00	9.260 K	Non consentito
csrss.exe	436	In esecuzione	SYSTEM	00	548 K	Non consentito
wininit.exe	512	In esecuzione	SYSTEM	00	4 K	Non consentito
csrss.exe	524	In esecuzione	SYSTEM	02	548 K	Non consentito
svchost.exe	528	In esecuzione	SYSTEM	00	16.804 K	Non consentito
TextInputHost.exe	560	In esecuzione	User	00	3.056 K	Disabilitato
winlogon.exe	604	In esecuzione	SYSTEM	00	560 K	Non consentito
services.exe	648	In esecuzione	SYSTEM	00	2.196 K	Non consentito
lsass.exe	664	In esecuzione	SYSTEM	00	2.660 K	Non consentito
svchost.exe	752	In esecuzione	User	00	6.092 K	Disabilitato
fontdrvhost.exe	756	In esecuzione	UMFD-0	00	88 K	Disabilitato
svchost.exe	772	In esecuzione	SYSTEM	00	5.232 K	Non consentito
svchost.exe	888	In esecuzione	SERVIZIO DI RETE	00	5.016 K	Non consentito
svchost.exe	956	In esecuzione	SERVIZIO LOCALE	00	3.304 K	Non consentito
dwm.exe	972	In esecuzione	DWM-1	03	38.400 K	Disabilitato

Da PowerShell si può controllare una vasta rete di computer. Si può aggiungere anche nuove impostazioni di sicurezza e simili, e si può controllare quali servizi stanno lavorando. Si possono inoltre dare comandi per semplificare azioni che richiederebbero più passaggi, come svuotare il cestino.



```
Windows PowerShell
255.255.255.255 255.255.255.255 On-link 192.168.50.30 281
=====
Route permanenti:
  Indirizzo rete      Mask      Indir. gateway  Metrica
      0.0.0.0          0.0.0.0      192.168.50.1  Predefinito
=====

IPv6 Tabella route
=====
Route attive:
  Interf Metrica Rete Destinazione Gateway
    1     331  ::1/128             On-link
    1     331 ff00::/8             On-link
=====
Route permanenti:
  Nessuna
PS C:\Users\User> 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"):
```

Cosa succede se diamo conferma? Il “Cestino” viene svuotato.

In conclusione, PowerShell è stato ideato per automatizzazione e configurazione dei dispositivi. Avvalendosi delle ricerche su internet possiamo trovare tutti i comandi che possono servirci a semplificare il lavoro come security analyst.



## Laboratorio 2

Esaminare il traffico HTTP e HTTPS tramite Wireshark

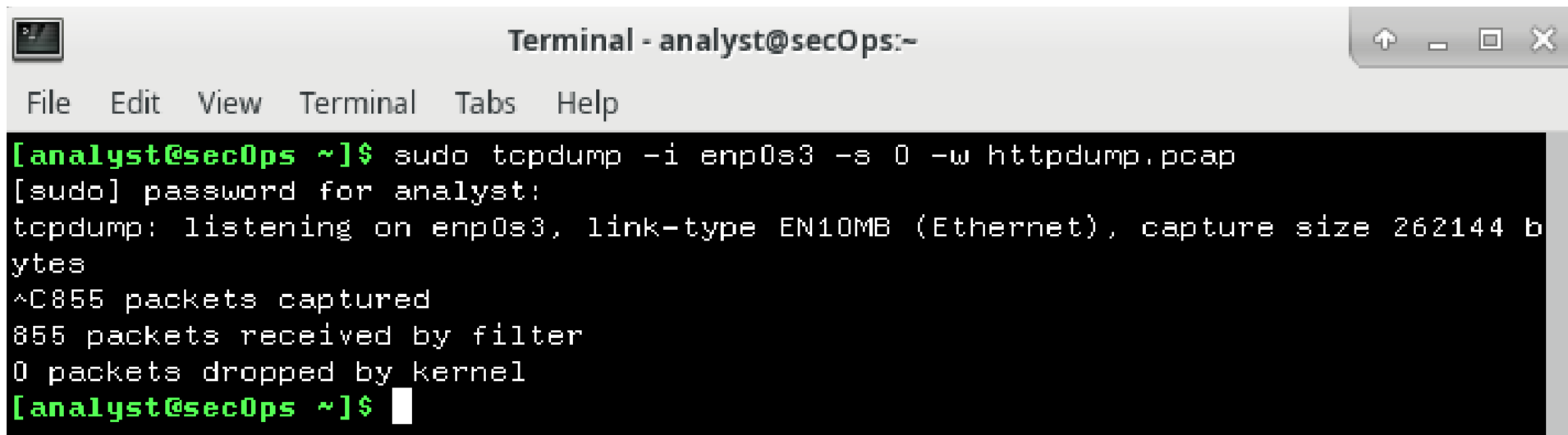
Vediamo ora come utilizzare Wireshark per analizzare traffico HTTP. Per svolgere questa analisi ci spostiamo su MV Linux.

Per prima cosa andiamo ad eseguire la cattura del traffico. Per fare ciò andiamo ad utilizzare il comando “sudo tcpdump -i enp0s3 -s 0 -w httpdump.pcap”.

Questo comando serve per mettersi in ascolto su una determinata porta o, in questo caso, una determinata interfaccia di rete e salverà in un file denominato “httpdump.pcap”.

Procediamo quindi collegandoci in HTTP ad un sito ed effettuando il login.

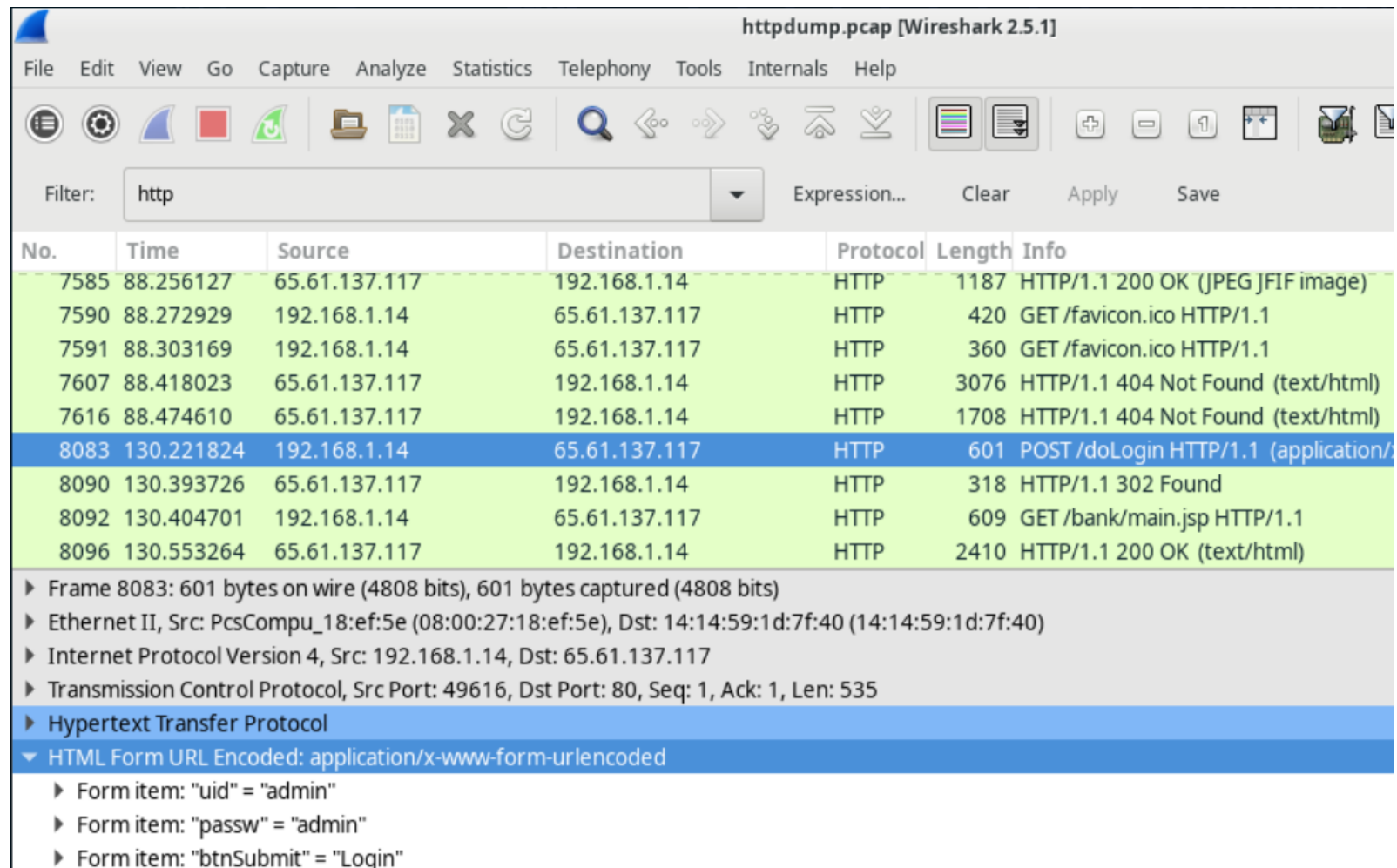
Terminiamo la cattura e avremo il nostro file analizzabile con Wireshark

A terminal window titled "Terminal - analyst@secOps:~" with standard window controls (up arrow, minus, square, X). The menu bar includes "File", "Edit", "View", "Terminal", "Tabs", and "Help". The terminal output shows the command "sudo tcpdump -i enp0s3 -s 0 -w httpdump.pcap" being executed. It prompts for a password, then shows the status: "tcpdump: listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes". It then reports "855 packets captured", "855 packets received by filter", and "0 packets dropped by kernel". The prompt returns to "[analyst@secOps ~]\$".

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

[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
^C855 packets captured
855 packets received by filter
0 packets dropped by kernel
[analyst@secOps ~]$
```

Cattura in HTTP (in chiaro)



httpdump.pcap [Wireshark 2.5.1]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter:  Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
7585	88.256127	65.61.137.117	192.168.1.14	HTTP	1187	HTTP/1.1 200 OK (JPEG JFIF image)
7590	88.272929	192.168.1.14	65.61.137.117	HTTP	420	GET /favicon.ico HTTP/1.1
7591	88.303169	192.168.1.14	65.61.137.117	HTTP	360	GET /favicon.ico HTTP/1.1
7607	88.418023	65.61.137.117	192.168.1.14	HTTP	3076	HTTP/1.1 404 Not Found (text/html)
7616	88.474610	65.61.137.117	192.168.1.14	HTTP	1708	HTTP/1.1 404 Not Found (text/html)
8083	130.221824	192.168.1.14	65.61.137.117	HTTP	601	POST /doLogin HTTP/1.1 (application/...)
8090	130.393726	65.61.137.117	192.168.1.14	HTTP	318	HTTP/1.1 302 Found
8092	130.404701	192.168.1.14	65.61.137.117	HTTP	609	GET /bank/main.jsp HTTP/1.1
8096	130.553264	65.61.137.117	192.168.1.14	HTTP	2410	HTTP/1.1 200 OK (text/html)

- ▶ Frame 8083: 601 bytes on wire (4808 bits), 601 bytes captured (4808 bits)
- ▶ Ethernet II, Src: PcsCompu\_18:ef:5e (08:00:27:18:ef:5e), Dst: 14:14:59:1d:7f:40 (14:14:59:1d:7f:40)
- ▶ Internet Protocol Version 4, Src: 192.168.1.14, Dst: 65.61.137.117
- ▶ Transmission Control Protocol, Src Port: 49616, Dst Port: 80, Seq: 1, Ack: 1, Len: 535
- ▶ Hypertext Transfer Protocol
- ▼ HTML Form URL Encoded: application/x-www-form-urlencoded
  - ▶ Form item: "uid" = "admin"
  - ▶ Form item: "passw" = "admin"
  - ▶ Form item: "btnSubmit" = "Login"

Aggiungendo un filtro per facilitare la ricerca, possiamo andare a vedere il login effettuato.

Nei dettagli del protocollo ci risulteranno in chiaro (essendo HTTP) le credenziali di accesso utilizzate per il login al sito.

Stesso procedimento lo andremo a svolgere per l'analisi del traffico HTTPS, vedendo inoltre le differenze tra traffico criptato o in chiaro.

Andiamo quindi, come prima, ad avviare una cattura con il comando “sudo tcpdump -i enp0s3 -s 0 -w httpsdump.pcap”

Saremo quindi nuovamente in ascolto e ci creerà un file “.pcap” al termine per poterlo analizzare. Procediamo quindi collegandoci in HTTPS ad un sito, effettuiamo il login e terminiamo la cattura.

```
[analyst@secOps ~]$ sudo tcpdump -i enp0s3 -s 0 -w httpsdump.pcap
[sudo] password for analyst:
tcpdump: listening on enp0s3, link-type EN10MB (Ethernet), capture
ytes
^C1753 packets captured
1753 packets received by filter
0 packets dropped by kernel
[analyst@secOps ~]$
```

## Cattura HTTPS

Analizziamo nuovamente il traffico catturato. Appliciamo un filtro per facilitare la ricerca e vediamo che il traffico è diverso anche nelle info di ogni riga. Selezionando una riga con Application Data possiamo vedere che nel dettaglio del collegamento non sono visibili, come invece succede HTTP, le informazioni di login poiché tutto crittografato con TLSv1.2

httpsdump.pcap [Wireshark 2.5.1]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter:  Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
69	4.017274	192.168.1.14	34.120.5.221	TCP	78	TCP Dup ACK 67#1160062 → 443 [ACK] Seq=296 Ack=
70	4.022999	192.168.1.14	34.120.5.221	TCP	66	60064 → 443 [ACK] Seq=296 Ack=3355 Win=40448 Len=
73	4.029528	34.120.5.221	192.168.1.14	TLSv1.2	141	[TCP Spurious Retransmission], Application Data
74	4.029544	192.168.1.14	34.120.5.221	TCP	78	TCP Dup ACK 70#1160064 → 443 [ACK] Seq=296 Ack=
93	4.094947	192.168.1.14	34.120.5.221	TLSv1.2	243	Application Data
94	4.095014	192.168.1.14	34.120.5.221	TLSv1.2	260	Application Data
95	4.095179	192.168.1.14	34.120.5.221	TLSv1.2	313	Application Data
96	4.095244	192.168.1.14	34.120.5.221	TLSv1.2	104	Application Data
97	4.095305	192.168.1.14	34.120.5.221	TLSv1.2	207	Application Data
98	4.095372	192.168.1.14	34.120.5.221	TLSv1.2	97	Encrypted Alert
99	4.095380	192.168.1.14	34.120.5.221	TCP	66	60062 → 443 [FIN, ACK] Seq=521 Ack=3375 Win=407
101	4.113893	34.120.5.221	192.168.1.14	TLSv1.2	104	Application Data
102	4.113914	192.168.1.14	34.120.5.221	TCP	66	60064 → 443 [ACK] Seq=899 Ack=3393 Win=40448 Len=
103	4.114082	34.120.5.221	192.168.1.14	TLSv1.2	104	Application Data
104	4.114097	192.168.1.14	34.120.5.221	TCP	54	60062 → 443 [RST] Seq=490 Win=0 Len=0
105	4.118215	34.120.5.221	192.168.1.14	TCP	66	443 → 60062 [RST] Seq=3343 Win=0 Len=0

▶ Internet Protocol Version 4, Src: 192.168.1.14, Dst: 34.120.5.221

▶ Transmission Control Protocol, Src Port: 60064, Dst Port: 443, Seq: 296, Ack: 3355, Len: 177

▼ Secure Sockets Layer

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

Content Type: Application Data (23)

Version: TLS 1.2 (0x0303)

Length: 172

Encrypted Application Data: 0000000000000001af621317d39e2a80e2e3637e77ae5d48...

## Bonus “NMAP”

Nmap è uno strumento utile per effettuare scansioni di reti e verifica della sicurezza. È particolarmente utile per scoprire dispositivi su una rete, identificare porte aperte, determinare i servizi in esecuzione e rilevare potenziali vulnerabilità. Andremo ad utilizzarlo in modi diversi per vederne varie funzionalità.

Nel primo caso andremo a scansionare il “localhost” (un indirizzo IP specifico per test: 127.0.0.1).

Useremo il comando “nmap -A -T4 localhost -A” che serve per impostare una scansione avanzata in modo che ci restituisca tutto.

-T4 serve per impostare il livello di velocità della scansione. Possiamo notare dalla scansione le porte aperte (21 e 22) oltre ad altri dettagli.

```
[analyst@secOps ~]$ nmap -A -T4 localhost
Starting Nmap 7.70 ( https://nmap.org ) at 2024-12-13 10:31 EST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000037s 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 12.66 seconds
```

Nel secondo caso si svolgerà una scansione su un'intera rete IP.

Questa procedura sarà utile per individuare e scansionare tutti i dispositivi in quella sottorete comprese porte aperte e servizi.

In questo caso metterò solo lo screen subito successivo al comando perché il risultato del comando è veramente lungo.

```
[analyst@sec0ps ~]$ nmap -A -T4 192.168.1.0/24
Starting Nmap 7.70 ( https://nmap.org ) at 2024-12-13 10:34 EST
Nmap scan report for www.adsl.vf (192.168.1.1)
Host is up (0.013s latency).
Not shown: 992 closed ports
PORT      STATE SERVICE      VERSION
22/tcp    filtered ssh
23/tcp    filtered telnet
53/tcp    open  domain       dnsmasq 2.84
| dns-nsid:
|_ bind.version: dnsmasq-2.84
80/tcp    open  http?
| fingerprint-strings:
|_  GetRequest, HTTPOptions:
|_    UNKNOWN 400 Bad Request
|_    Server:
|_      Date: Fri, 13 Dec 2024 09:34:45 GMT
|_      Cache-Control: no-cache,no-store,max-age=0
|_      Pragma: no-cache
|_      X-Frame-Options: DENY
|_      Expires: 0
|_      X-Content-Type-Options: nosniff
|_      X-XSS-Protection: 0; mode=block
|_      Content-Security-Policy: default-src 'self' 'unsafe-inline' 'unsafe-eval'; img-src 'self' data:
|_      Content-Language: en
|_      Content-Type: text/html
|_      Connection: close
|_      <HTML>
|_      <HEAD><TITLE>400 Bad Request</TITLE></HEAD>
|_      <BODY BGCOLOR="#cc9999" TEXT="#000000" LINK="#2020ff" VLINK="#4040cc">
|_      <H4>400 Bad Request</H4>
|_      Invalid Request
|_    NULL:
|_      UNKNOWN 408 Request Timeout
|_      Server:
|_        Date: Fri, 13 Dec 2024 09:34:45 GMT
|_        Cache-Control: no-cache,no-store,max-age=0
|_        Pragma: no-cache
|_        X-Frame-Options: DENY
|_        Expires: 0
|_        X-Content-Type-Options: nosniff
|_        X-XSS-Protection: 0; mode=block
|_        Content-Security-Policy: default-src 'self' 'unsafe-inline' 'unsafe-eval'; img-src 'self' data:
|_        Content-Language: en
|_        Content-Type: text/html
|_        Connection: close
|_        <HTML>
|_        <HEAD><TITLE>408 Request Timeout</TITLE></HEAD>
|_        <BODY BGCOLOR="#cc9999" TEXT="#000000" LINK="#2020ff" VLINK="#4040cc">
|_        <H4>408 Request Timeout</H4>
|_        request appeared within a reasonable time period.
```

Nel terzo caso andremo a scansionare un server web remoto utilizzando come target "scanme.nmap.org"

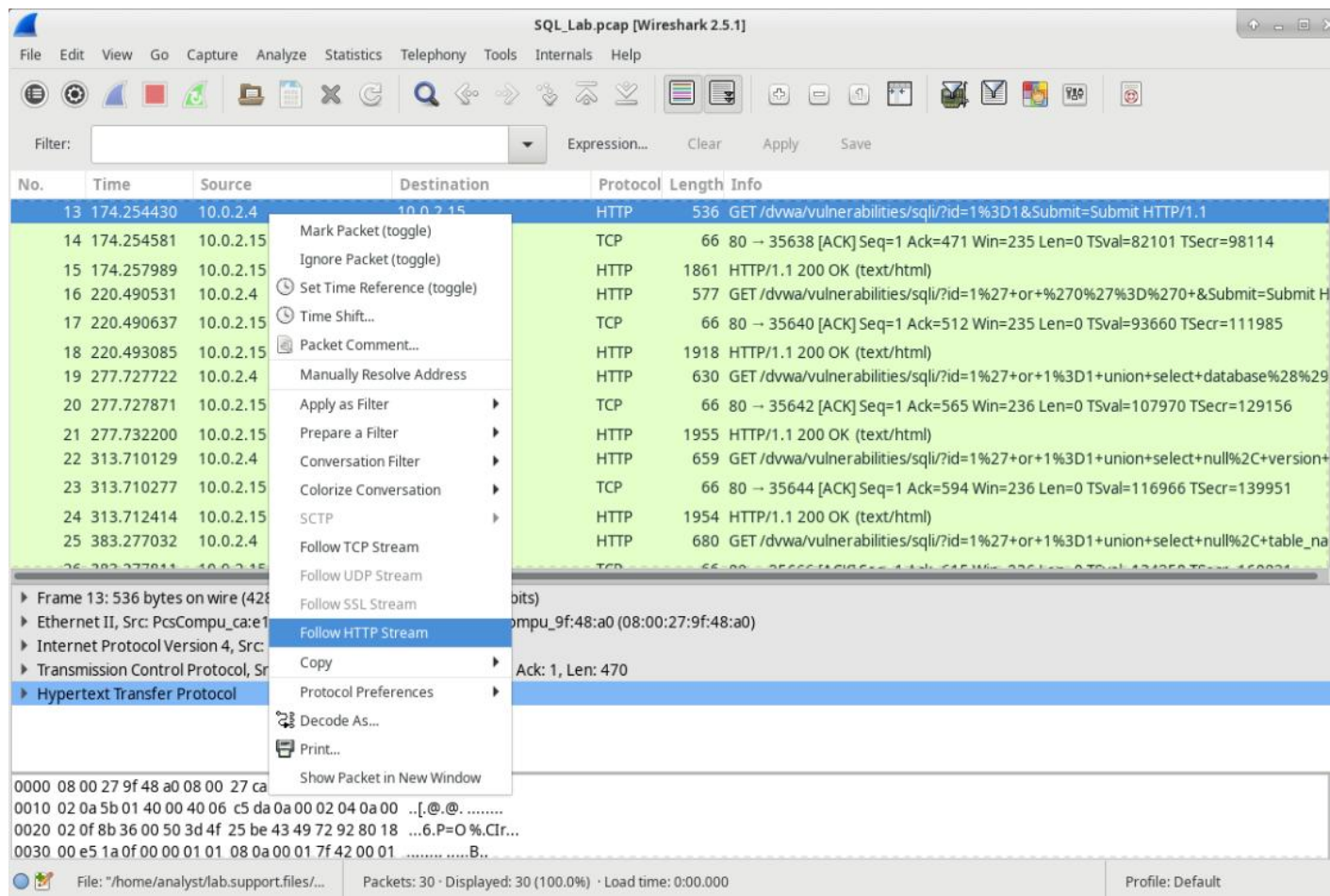
```
[analyst@secOps ~]$ nmap -A -T4 scanme.nmap.org
Starting Nmap 7.70 ( https://nmap.org ) at 2024-12-13 10:42 EST
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.18s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:9
Not shown: 995 closed ports
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu)
|_ 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)
53/tcp    open  domain       dnsmasq 2.84
|_ dns-nsid:
|_ bind.version: dnsmasq-2.84
80/tcp    open  http         Apache httpd 2.4.7 ((Ubuntu))
|_ http-server-header: Apache/2.4.7 (Ubuntu)
|_ http-title: Go ahead and ScanMe!
9929/tcp  open  nping-echo   Nping echo
31337/tcp open  tcpwrapped
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at
Nmap done: 1 IP address (1 host up) scanned in 44.58 seconds
```



## Bonus “MYSQL”

Utilizzando Wireshark andiamo ad analizzare una cattura di traffico di rete relativo ad un attacco SQL injection contro un database SQL. La SQL injection è una vulnerabilità di sicurezza che consente a un attaccante di inserire o manipolare query SQL all'interno di un'applicazione web. Questo avviene quando gli input non sono correttamente sanitizzati. Non essendo filtrati correttamente, l'attaccante può eseguire comandi SQL dannosi, che permettono di accedere, manipolare o distruggere dati nel database.

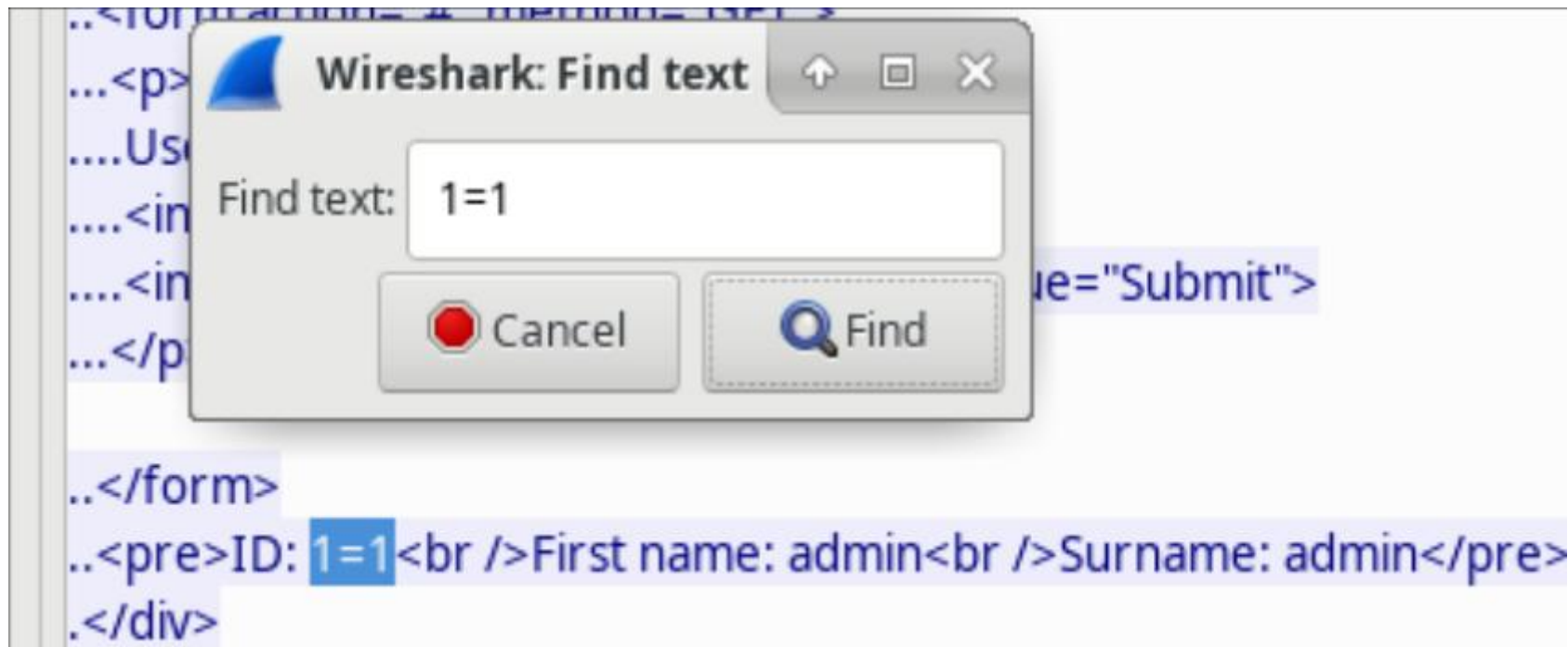




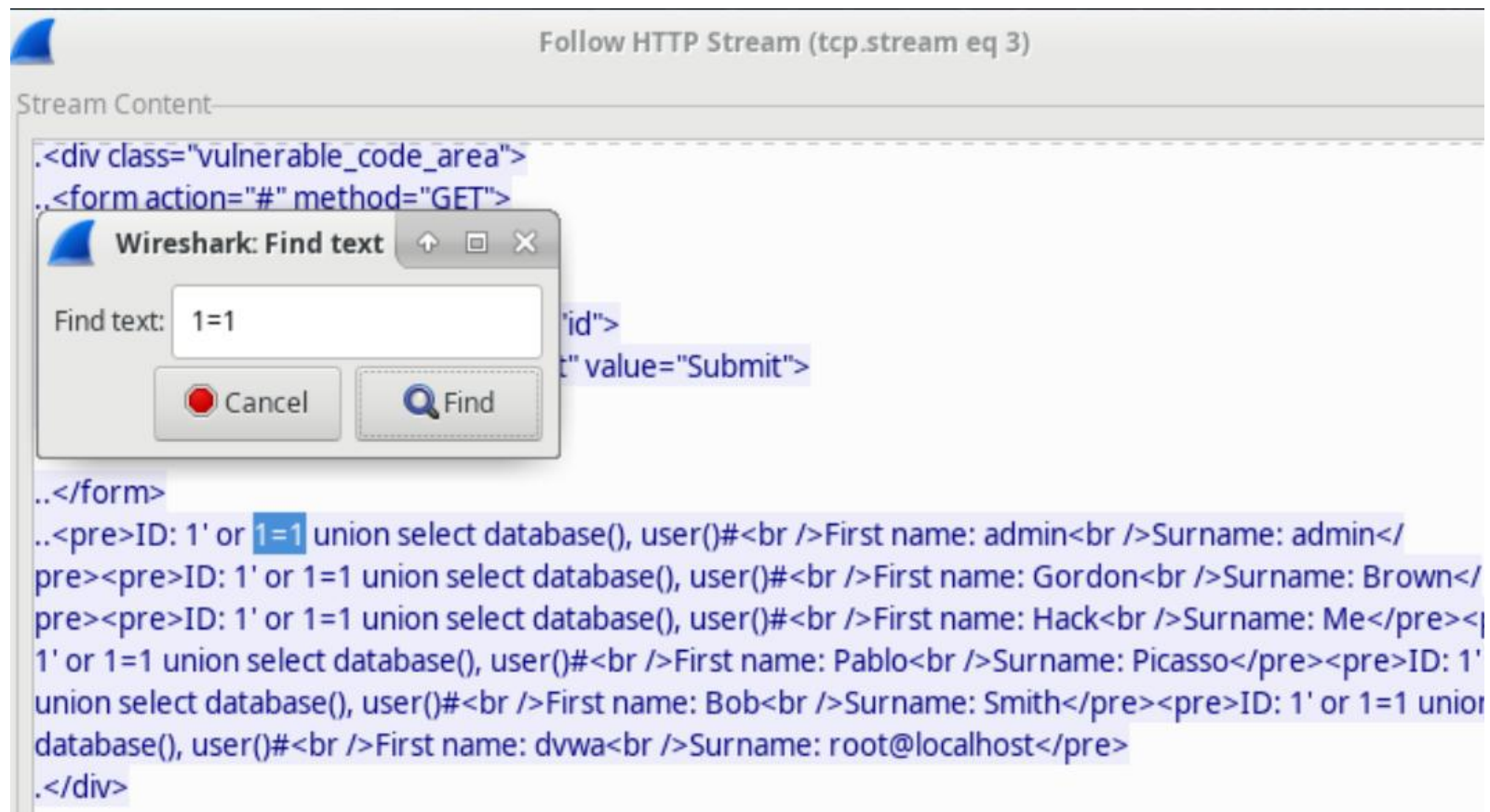
Nello screenshot precedente possiamo vedere i due indirizzi IP coinvolti: 10.0.2.4 (attaccante) e 10.0.2.15 (server vittima).

Andiamo ad aprire la sezione "HTTP Stream" che ci fornirà nel dettaglio la conversazione http tra il client e il server includendo sia la richiesta inviata dal client (ad esempio un browser) sia la risposta del server.

Inizialmente l'attaccante ha inviato una richiesta al server per testare se l'applicazione fosse vulnerabile a una SQL injection. Il comando `1=1` inserito nel campo "UserID" ha restituito un risultato positivo, mostrando che l'applicazione era vulnerabile, poiché il comando `1=1` è sempre vero e il server ha risposto con un record dal database.

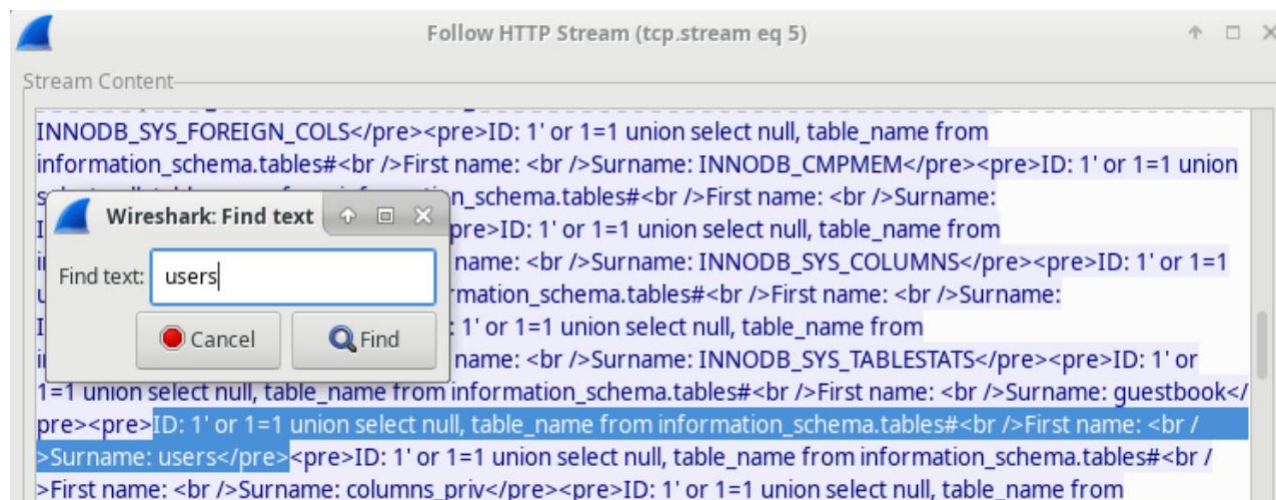


Nel proseguire l'attaccante ha utilizzato una query per ottenere informazioni sul database. Il server ha risposto con il nome del database (DVWA) e l'utente (root@localhost).



L'attaccante ha continuato con una nuova query che ha permesso di ottenere la versione di MySQL in uso

L'attaccante ha quindi cercato di ottenere una lista delle tabelle del database.



Infine, l'attaccante ha utilizzato una query per recuperare gli username e gli hash delle password dalla tabella "users".  
Avendo trovato l'hash ci sono vari tool o siti per trovare una corrispondenza e risalire quindi alla password in chiaro da utilizzare per il furto dell'account.

