

# **UTILIZZO DI WINDOWS POWER SHELL**



Nichol Galessiere



13/12/2024

# Indice

- Obiettivo
- Introduzione
- Parte 1: accedere alla console di PowerShell
- Parte 2: Esplora i comandi del prompt dei comandi e di PowerShell
- Parte 3: Esplora i cmdlet
- Parte 4: Esplora il comando netstat utilizzando PowerShell
- Parte 5: Svuotare il cestino tramite PowerShell
- Considerazioni



# OBIETTIVO

L'obiettivo del laboratorio è esplorare alcune delle funzioni di PowerShell.

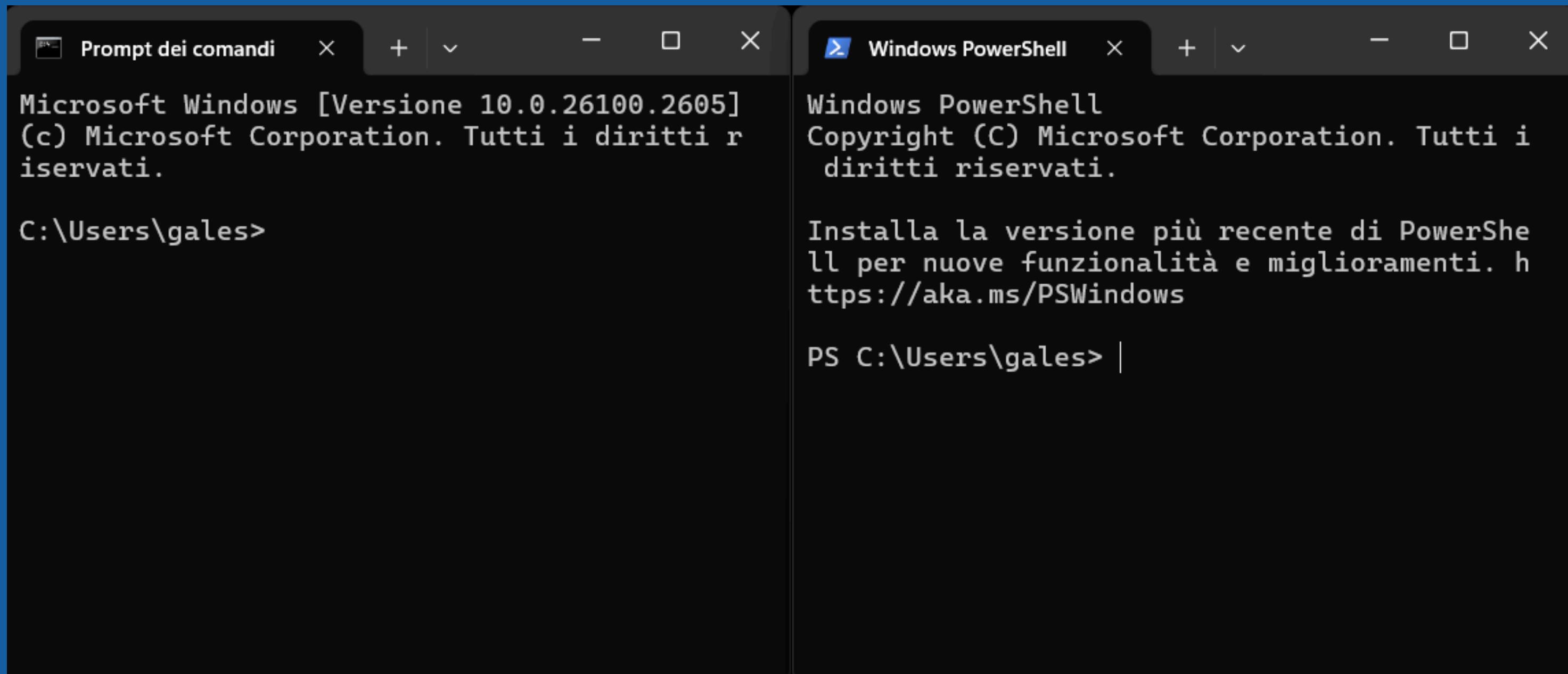
- **Parte 1: accedere alla console di PowerShell.**
- **Parte 2: Esplora i comandi del prompt dei comandi e di PowerShell.**
- **Parte 3: Esplora i cmdlet.**
- **Parte 4: Esplora il comando netstat utilizzando PowerShell.**
- **Parte 5: Svuotare il cestino tramite PowerShell.**

# Introduzione

Windows PowerShell è una shell di comandi e linguaggio di scripting avanzato sviluppata da Microsoft, utilizzata principalmente per l'automazione delle attività di amministrazione di sistemi e per l'interazione con il sistema operativo Windows in modo più potente e flessibile rispetto al tradizionale Prompt dei comandi (CMD). PowerShell è stato progettato per semplificare la gestione di computer locali e remoti, facilitando operazioni complesse su file, processi, configurazioni di sistema, e altro ancora.

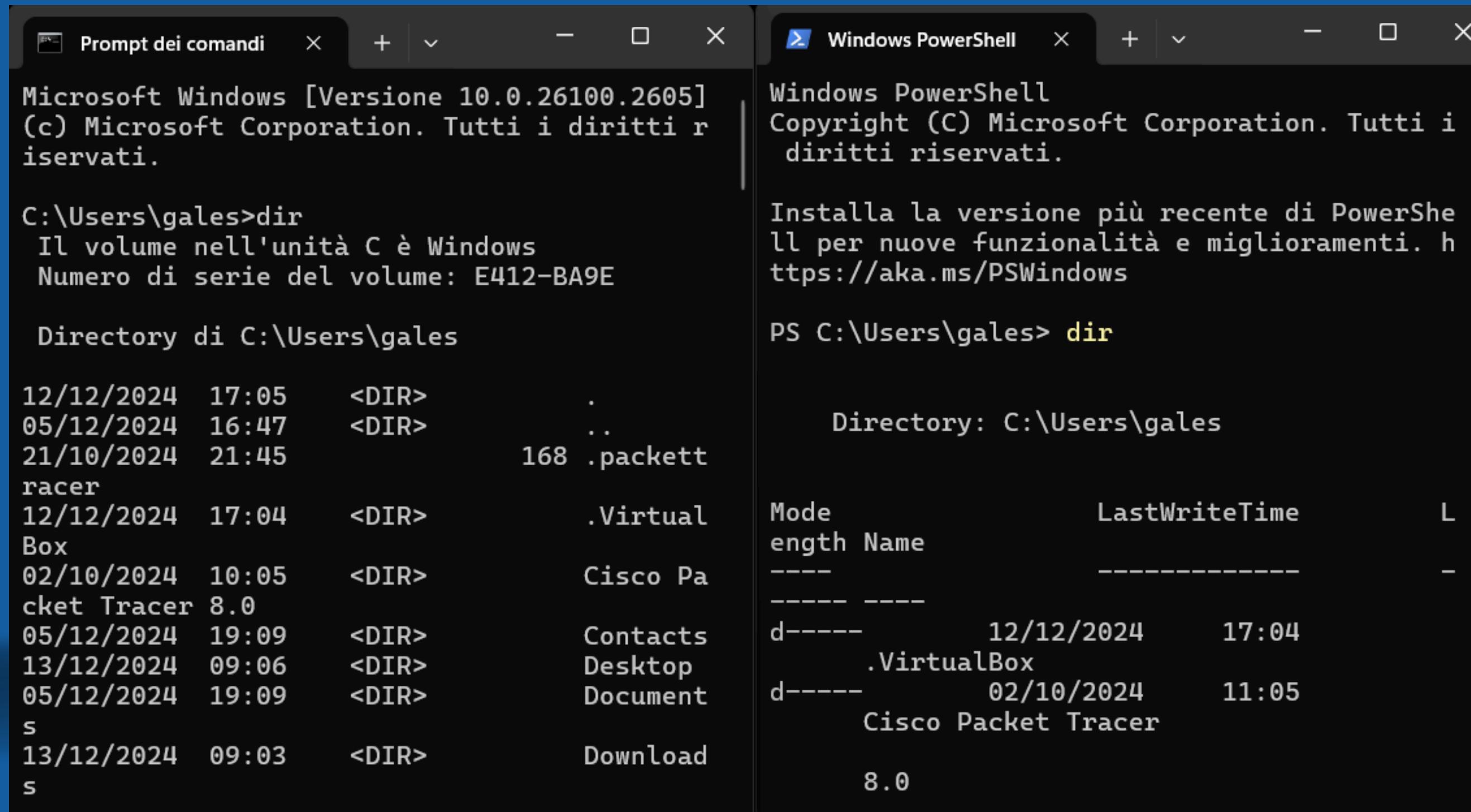


# Parte 1: accedere alla console di PowerShell.



In questo passaggio stiamo eseguendo un confronto tra il Prompt dei comandi e PowerShell. Molti comandi che si usano nel Prompt dei comandi sono compatibili con PowerShell, ma quest'ultimo offre comandi più avanzati. Comprendere le differenze ci aiuterà a capire quando e perché scegliere PowerShell.

# Parte 2: Esplora i comandi del prompt dei comandi e di PowerShell.



The image shows two side-by-side command-line windows on a Windows operating system. The left window is titled "Prompt dei comandi" and the right window is titled "Windows PowerShell". Both windows display the output of a "dir" command.

**Prompt dei comandi Output:**

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

C:\Users\gales>dir
Il volume nell'unità C è Windows
Numero di serie del volume: E412-BA9E

Directory di C:\Users\gales

12/12/2024 17:05    <DIR>          .
05/12/2024 16:47    <DIR>          ..
21/10/2024 21:45            168 .packettracer
12/12/2024 17:04    <DIR>          .VirtualBox
02/10/2024 10:05    <DIR>          Cisco Packet Tracer 8.0
05/12/2024 19:09    <DIR>          Contacts
13/12/2024 09:06    <DIR>          Desktop
05/12/2024 19:09    <DIR>          Document
13/12/2024 09:03    <DIR>          Download
```

**Windows PowerShell Output:**

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

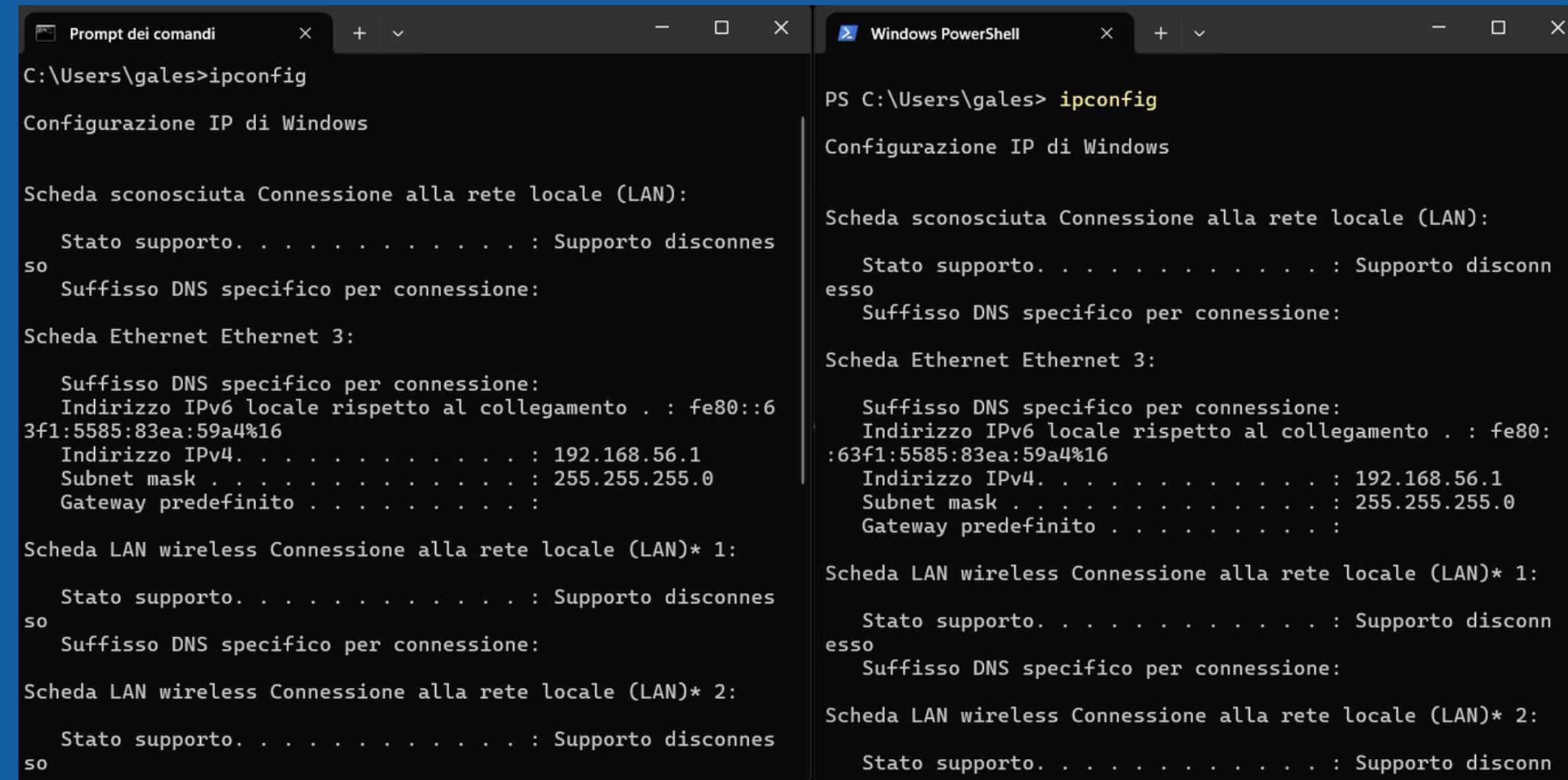
Install la versione più recente di PowerShell per nuove funzionalità e miglioramenti. https://aka.ms/PSWindows

PS C:\Users\gales> dir

Directory: C:\Users\gales

Mode                LastWriteTime     Length Name
----                -----          ----
d-----        12/12/2024      17:04   .VirtualBox
d-----        02/10/2024      11:05   Cisco Packet Tracer 8.0
```

# Parte 2: Esplora i comandi del prompt dei comandi e di PowerShell.



The image shows two side-by-side command-line windows. The left window is titled "Prompt dei comandi" and the right window is titled "Windows PowerShell". Both windows are running the "ipconfig" command. The output in both windows is identical, displaying network configuration details for multiple interfaces (Scheda sconosciuta, Scheda Ethernet Ethernet 3, Scheda LAN wireless Connessione alla rete locale (LAN)\* 1, Scheda LAN wireless Connessione alla rete locale (LAN)\* 2). The output includes connection status (Supporto disconnesso), specific DNS suffixes, and IPv4/IPv6 addresses along with their subnet masks and default gateways.

```
C:\Users\gales>ipconfig
Configurazione IP di Windows

Scheda sconosciuta Connessione alla rete locale (LAN):
  Stato supporto. . . . . : Supporto disconnesso
  Suffisso DNS specifico per connessione:

Scheda Ethernet Ethernet 3:
  Suffisso DNS specifico per connessione:
  Indirizzo IPv6 locale rispetto al collegamento . : fe80::6
3f1:5585:83ea:59a4%16
  Indirizzo IPv4. . . . . : 192.168.56.1
  Subnet mask . . . . . : 255.255.255.0
  Gateway predefinito . . . . . :

Scheda LAN wireless Connessione alla rete locale (LAN)* 1:
  Stato supporto. . . . . : Supporto disconnesso
  Suffisso DNS specifico per connessione:

Scheda LAN wireless Connessione alla rete locale (LAN)* 2:
  Stato supporto. . . . . : Supporto disconnesso

PS C:\Users\gales> ipconfig
Configurazione IP di Windows

Scheda sconosciuta Connessione alla rete locale (LAN):
  Stato supporto. . . . . : Supporto disconnesso
  Suffisso DNS specifico per connessione:

Scheda Ethernet Ethernet 3:
  Suffisso DNS specifico per connessione:
  Indirizzo IPv6 locale rispetto al collegamento . : fe80:
:63f1:5585:83ea:59a4%16
  Indirizzo IPv4. . . . . : 192.168.56.1
  Subnet mask . . . . . : 255.255.255.0
  Gateway predefinito . . . . . :

Scheda LAN wireless Connessione alla rete locale (LAN)* 1:
  Stato supporto. . . . . : Supporto disconnesso
  Suffisso DNS specifico per connessione:

Scheda LAN wireless Connessione alla rete locale (LAN)* 2:
  Stato supporto. . . . . : Supporto disconnesso
```

Il comando dir è utilizzato per elencare i contenuti di una directory (file e cartelle). È uno dei comandi di base utilizzati nel Prompt dei comandi di Windows.

In PowerShell, il comando dir è un alias per il cmdlet Get-ChildItem, che restituisce lo stesso risultato, ma in PowerShell si può anche lavorare con gli oggetti risultanti. Quindi, mentre in Prompt dei comandi si ottiene semplicemente un elenco, in PowerShell si può fare molto di più con i risultati.

# Parte 3: Esplora i cmdlet.

```
Windows PowerShell

Gateway predefinito . . . . . : 1
92.168.1.254

Scheda Ethernet Ethernet:

    Stato supporto. . . . . : S
    upporto disconnesso
    Suffisso DNS specifico per connessione:
PS C:\Users\gales> Get-Alias dir

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

PS C:\Users\gales>
```

I cmdlet sono **comandi nativi di PowerShell, non eseguibili autonomi**. I cmdlet vengono raccolti nei moduli di PowerShell che **possono essere caricati su richiesta**. I cmdlet possono essere scritti in qualsiasi linguaggio .NET compilato o nel linguaggio di scripting di PowerShell stesso. 8 nov 2024

PowerShell utilizza alias per semplificare l'uso dei comandi. Gli alias sono abbreviazioni o sinonimi per i cmdlet di PowerShell. Ad esempio, dir è un alias per il cmdlet Get-ChildItem, che serve per ottenere l'elenco dei file e delle cartelle in una directory.

# Parte 4: Esplora il comando netstat utilizzando PowerShell.

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

Socket Handle Count

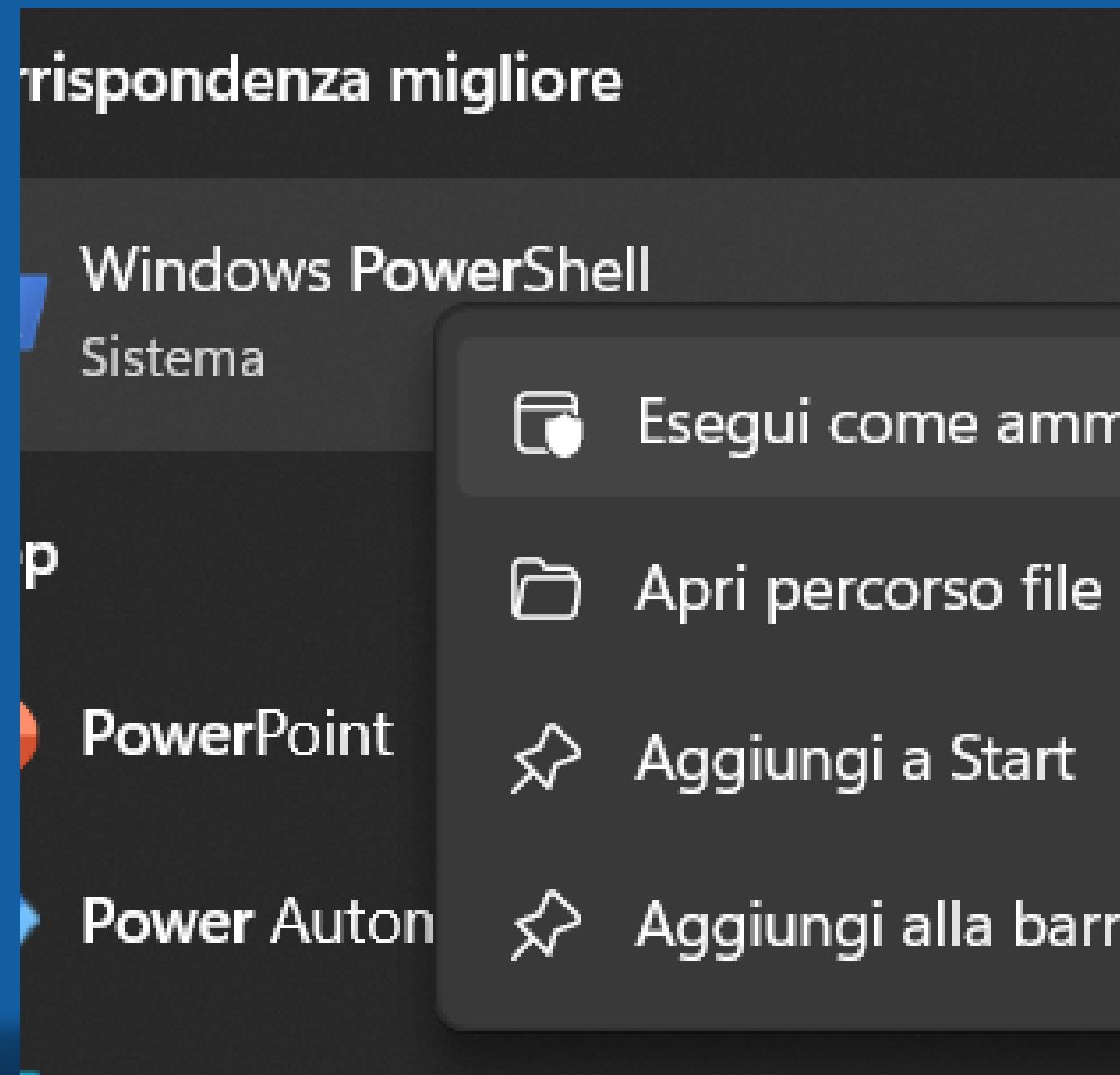
  PID      Count  Closing Count
    6912        1          0
    7684        3          0
    1296        4          0
    3100        4          0
    1840       12          1
    2100        1          0
    19512       1          0
    14396       1          0
    11840       11         0
    1368        4          0
    1884        2          0
    6236        4          0
    4968        4          0
    15216       12         0
    1396        4          0
    11128       10         0
    9084       34          0
    17536       2          0
```

```
PS C:\Users\gales> netstat -r
=====
Elenco interfacce
  4.....ExpressVPN TUN Driver
  16...0a 00 27 00 00 10 .....VirtualBox Host-Only Ethernet Adapter
    7...16 d4 24 55 71 2d .....Microsoft Wi-Fi Direct Virtual Adapter
  14...16 d4 24 55 71 3d .....Microsoft Wi-Fi Direct Virtual Adapter #2
  11...14 d4 24 55 71 2d .....MediaTek MT7921 Wi-Fi 6 802.11ax PCIe Adapter
  20...bc 0f f3 73 e0 65 .....Realtek Gaming GbE Family Controller
    1.....Software Loopback Interface 1
=====

IPv4 Tabella route
=====
Route attive:
  Indirizzo rete          Mask        Gateway      Interfaccia Metrica
            0.0.0.0        0.0.0.0    192.168.1.254  192.168.1.54      30
            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
            192.168.1.0      255.255.255.0  On-link        192.168.1.54     286
            192.168.1.54      255.255.255.255  On-link        192.168.1.54     286
            192.168.1.255    255.255.255.255  On-link        192.168.1.54     286
            192.168.56.0      255.255.255.0  On-link        192.168.56.1     281
```

Il comando netstat (network statistics) è uno strumento utile per visualizzare le connessioni di rete attive sul computer, le porte in ascolto, e altre statistiche di rete. L'opzione -h mostra l'aiuto e la sintassi per l'uso del comando. Monitorare le connessioni di rete è importante per rilevare attività sospette o per diagnosticare problemi di connettività.

# Parte 4: Esplora il comando netstat utilizzando PowerShell.

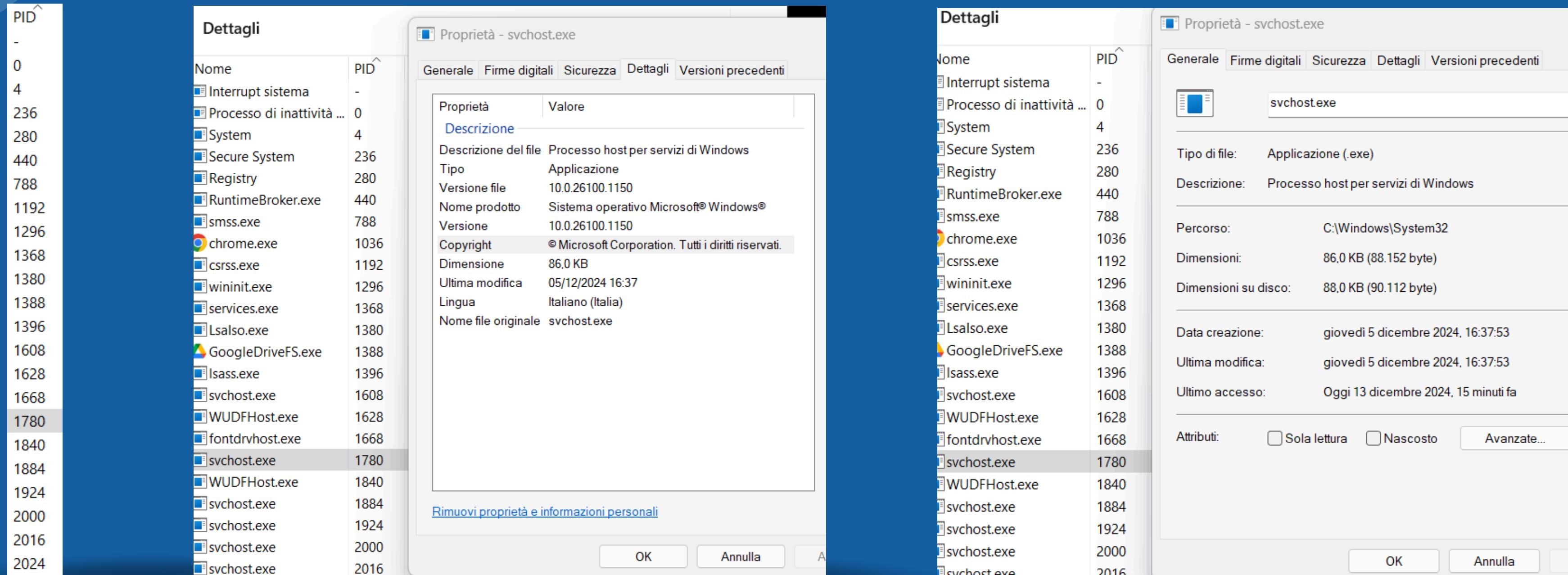


```
PS C:\WINDOWS\system32> netstat -abno
```

Connessioni attive	Proto	Indirizzo locale	Indirizzo esterno	Stato	PID
[svchost.exe]	TCP	0.0.0.0:135	0.0.0.0:0	LISTENING	1780
RpcSs					
[svchost.exe]	TCP	0.0.0.0:445	0.0.0.0:0	LISTENING	4
Impossibile ottenere informazioni sulla proprietà					
[svchost.exe]	TCP	0.0.0.0:5040	0.0.0.0:0	LISTENING	9708
CDPSvc					
[svchost.exe]	TCP	0.0.0.0:7680	0.0.0.0:0	LISTENING	17536
Impossibile ottenere informazioni sulla proprietà					
[svchost.exe]	TCP	0.0.0.0:49664	0.0.0.0:0	LISTENING	1396
Impossibile ottenere informazioni sulla proprietà					
[svchost.exe]	TCP	0.0.0.0:49665	0.0.0.0:0	LISTENING	1296
Impossibile ottenere informazioni sulla proprietà					
[svchost.exe]	TCP	0.0.0.0:49666	0.0.0.0:0	LISTENING	3100
EventLog					
[svchost.exe]	TCP	0.0.0.0:49667	0.0.0.0:0	LISTENING	2556
Schedule					
[svchost.exe]	TCP	0.0.0.0:49668	0.0.0.0:0	LISTENING	4968
[spoolsv.exe]	TCP	0.0.0.0:49674	0.0.0.0:0	LISTENING	1368
Impossibile ottenere informazioni sulla proprietà					
[svchost.exe]	TCP	127.0.0.1:2015	0.0.0.0:0	LISTENING	6236

Netstat -abno: Questo comando fornisce informazioni dettagliate sulle connessioni TCP attive e sui processi che le utilizzano, incluso il nome del programma e il PID (Process ID). È fondamentale in sicurezza per identificare quali processi sono responsabili delle connessioni di rete attive e per rilevare eventuali programmi non autorizzati o malevoli.

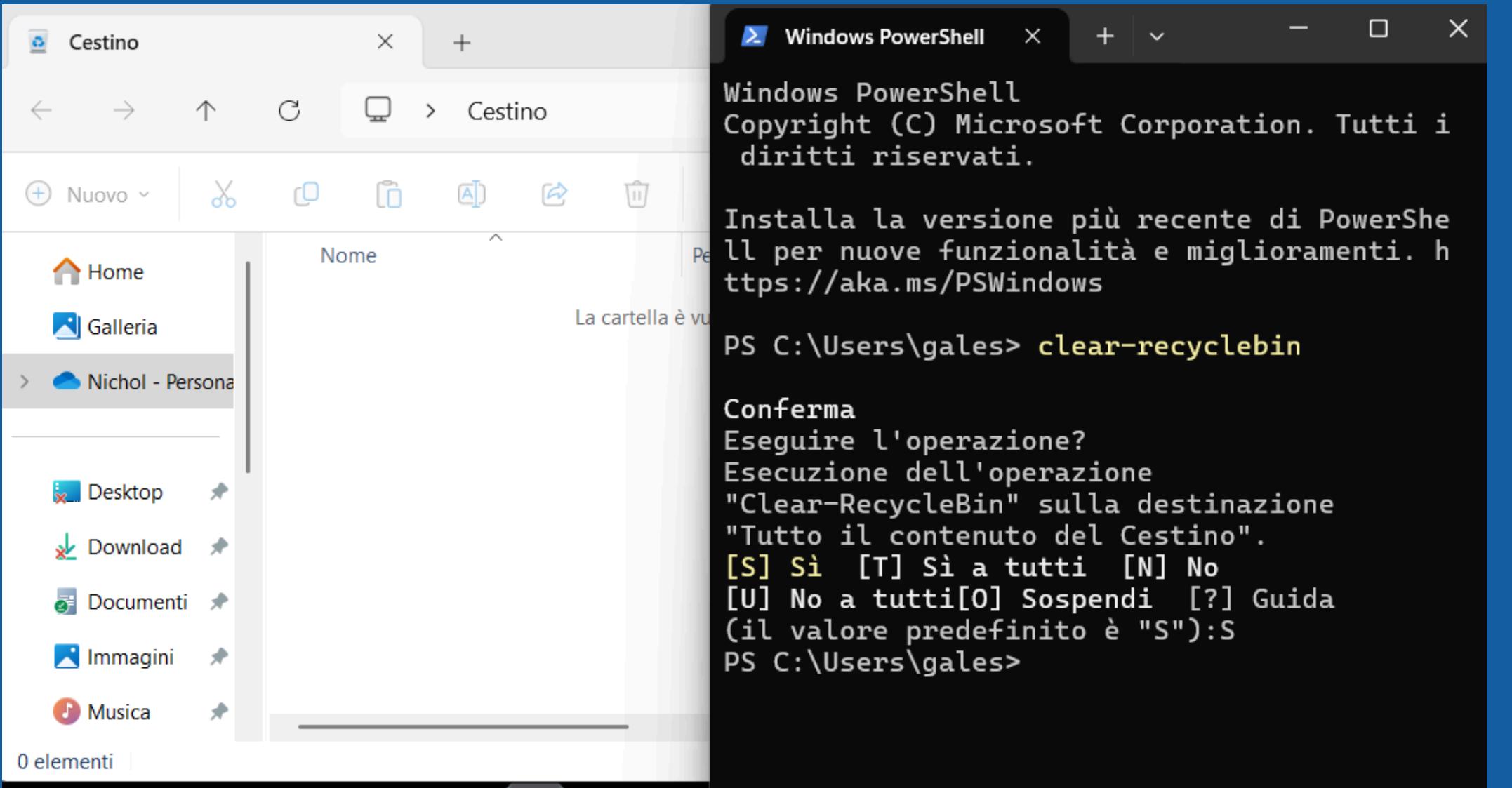
# Parte 4: Esplora il comando netstat utilizzando PowerShell.



Utilizzando il Task Manager, possiamo osservare in dettaglio i processi in esecuzione, ordinati per PID. Se abbiamo un PID dal comando netstat, possiamo cercarlo nel Task Manager per capire quale processo sta utilizzando una connessione di rete.

Questo passaggio aiuta a correlare i dati di rete con i processi attivi nel sistema e a monitorare attività sospette.

# Parte 5: Svuotare il cestino tramite PowerShell.



Il cmdlet Clear-RecycleBin elimina permanentemente tutti i file nel Cestino. È molto utile per automatizzare la pulizia dei file eliminati dal sistema, senza doverlo fare manualmente tramite l'interfaccia grafica.  
Automatizzare attività come svuotare il Cestino aiuta a risparmiare tempo e a mantenere il sistema più ordinato.



# CONSIDERAZIONI

PowerShell è un potente strumento che consente agli amministratori di sistema e agli analisti della sicurezza di gestire e automatizzare operazioni quotidiane. Il laboratorio ha mostrato la versatilità di PowerShell nel gestire operazioni come la gestione dei file, l'analisi delle connessioni di rete e la gestione avanzata dei processi.

Concludendo, l'uso di PowerShell nella sicurezza informatica consente di implementare controlli e risposte automatiche in modo più efficace, riducendo il rischio di errori umani e migliorando l'efficienza operativa. La sua capacità di eseguire script complessi su una rete di computer lo rende indispensabile per la gestione di sistemi su larga scala.

# GRAZIE



Nichol Galessiere

13/12/2024



# **UTILIZZO DI WIRESHARK PER ESAMINARE IL TRAFFICO HTTP E HTTPS**



Nichol Galessiere



13/12/2024

# Indice

- Obiettivo
- Introduzione
- Parte 1: Cattura e visualizza il traffico HTTP
- Parte 2: Cattura e visualizza il traffico HTTPS
- Considerazioni
- Conclusioni

# OBIETTIVO

- **Parte 1: Cattura e visualizza il traffico HTTP**
- **Parte 2: Cattura e visualizza il traffico HTTPS**

# Introduzione

HTTP (Hypertext Transfer Protocol) è un protocollo di comunicazione utilizzato per il trasferimento di dati tra client (browser web) e server web. Il traffico HTTP è non crittografato, quindi i dati scambiati (come password e informazioni sensibili) sono visibili a chiunque possa intercettarli.

HTTPS (HTTP Secure) è una versione sicura di HTTP che utilizza il protocollo SSL/TLS per crittografare i dati scambiati tra il browser e il server. Questo garantisce che le informazioni siano protette e non possano essere facilmente intercattate.

Wireshark è un'applicazione di analisi di rete che permette di "catturare" pacchetti di dati in transito su una rete, visualizzandone il contenuto. Utilizzeremo tcpdump per catturare il traffico e successivamente lo analizzeremo con Wireshark.

# Parte 1: Cattura e visualizza il traffico HTTP

Abbiamo utilizzato il comando ip address per visualizzare le interfacce di rete disponibili nella macchina virtuale, insieme agli indirizzi IP associati a ciascuna interfaccia.

Questo passaggio è fondamentale perché permette di sapere quale interfaccia di rete (come enp0s3) utilizzare per catturare il traffico di rete. L'interfaccia di rete è il canale attraverso cui il computer comunica con altri dispositivi sulla rete.

In seguito abbiamo avviato tcpdump.

Abbiamo aperto un browser web sulla macchina virtuale, abbiamo utilizzato un sito che utilizza HTTP (non sicuro).

Poiché il sito è in HTTP, tutto il traffico scambiato tra il browser e il sito è in chiaro, ovvero senza crittografia. Questo permette di osservare i dati come nome utente e password, cosa che non sarebbe possibile se il sito fosse in HTTPS (come nel caso di traffico criptato).

Abbiamo interagito con il sito web, inserendo il nome utente e la password (entrambi come Admin), e quindi abbiamo cliccato su "Accedi".

Perché è importante: Durante questa interazione, il browser ha inviato una richiesta POST al server con i dati del modulo (username e password). Questo tipo di richiesta è ciò che ti permette di fare login nel sito.

# Parte 1: Cattura e visualizza il traffico HTTP

Dopo aver effettuato il login, abbiamo chiuso il browser.

Il traffico generato durante l'invio dei dati del modulo HTTP è stato catturato da tcpdump. Ora possiamo fermare la cattura dei pacchetti.

Siamo tornati alla finestra del terminale dove tcpdump stava catturando i pacchetti e abbiamo premuto CTRL+C per fermare la cattura.

Dopo aver interrotto tcpdump, il file di cattura httpdump.pcap è stato salvato e contiene tutti i pacchetti registrati fino a quel momento. Ora possiamo analizzare questo file per osservare il traffico HTTP.

Abbiamo aperto il file httpdump.pcap con Wireshark, uno strumento grafico per analizzare i pacchetti di rete.

In Wireshark, abbiamo applicato un filtro per mostrare solo i pacchetti HTTP utilizzando il filtro http.

Questo filtro ci permette di concentrarci esclusivamente sui pacchetti che riguardano il traffico HTTP, ignorando altri tipi di traffico (come il traffico DNS, ARP, ecc.).

Abbiamo esplorato i pacchetti HTTP e abbiamo selezionato uno di tipo POST, che è il tipo di richiesta utilizzato per inviare i dati del modulo (username e password).

I pacchetti POST contengono i dati sensibili (nome utente e password), e osservando questi pacchetti possiamo vedere come questi vengano inviati in chiaro attraverso la rete.

Successivamente abbiamo espanso la sezione HTML Form URL Encoded: application/x-www-form-urlencoded per visualizzare i dati inviati nel modulo (nome utente e password).

In questo caso, i dati del login (nome utente e password) sono inviati come una stringa codificata URL in chiaro. Ciò significa che chiunque possa intercettare questo traffico potrebbe leggere facilmente le credenziali.

# Parte 1: Cattura e visualizza il traffico HTTP

```
Terminal - analyst@secOps:~  
File Edit View Terminal Tabs Help  
[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: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000  
    link/ether 08:00:27:aa:64:c3 brd ff:ff:ff:ff:ff:ff  
    inet 192.168.1.65/24 brd 192.168.1.255 scope global dynamic enp0s3  
        valid_lft 85162sec preferred_lft 85162sec  
    inet6 fe80::a00:27ff:feaa:64c3/64 scope link  
        valid_lft forever preferred_lft forever  
[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 262  
144 bytes  
[analyst@secOps ~]$
```

## Online Banking Login

Username:

Password:

This connection is not secure.

```
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 262  
bytes  
man  
^C2028 packets captured  
2028 packets received by filter  
0 packets dropped by kernel  
[analyst@secOps ~]$
```

# Parte 1: Cattura e visualizza il traffico HTTP

analyst - File Manager

View Go Help

/home/analyst/

System	Desktop	Downloads	lab.support.files
capture	capture.pcap	httpdump.pcap	

Filter: http

No.	Time	Source	Destination	Protocol	Length	Info
1200	368.164656	192.168.1.65	192.229.221.95	OCSP	497	Request
1201	368.167333	192.168.1.65	192.229.221.95	OCSP	497	Request
1208	368.188053	192.168.1.65	192.229.221.95	OCSP	497	Request
1210	368.193874	192.229.221.95	192.168.1.65	OCSP	801	Response
1213	368.199171	192.229.221.95	192.168.1.65	OCSP	803	Response
1220	368.214366	192.168.1.65	192.229.221.95	OCSP	497	Request
1222	368.220155	192.168.1.65	192.229.221.95	OCSP	497	Request
1223	368.229959	192.229.221.95	192.168.1.65	OCSP	803	Response
1230	368.249794	192.229.221.95	192.168.1.65	OCSP	801	Response
1232	368.253699	192.229.221.95	192.168.1.65	OCSP	803	Response
1610	438.092073	192.168.1.65	65.61.137.117	HTTP	601	POST /doLogin HTTP/1.1 (application/x-www-form-urlencoded)
1613	438.247989	65.61.137.117	192.168.1.65	HTTP	339	HTTP/1.1 302 Found
1615	438.255122	192.168.1.65	65.61.137.117	HTTP	619	GET /bank/main.jsp HTTP/1.1
1616	438.413002	65.61.137.117	192.168.1.65	HTTP	6514	HTTP/1.1 200 OK (text/html)
1716	465.384605	192.168.1.65	172.64.149.23	OCSP	496	Request
1719	465.423936	172.64.149.23	192.168.1.65	OCSP	894	Response

▶ Transmission Control Protocol, Src Port: 57216, 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"

# Parte 2: Cattura e visualizza il traffico HTTPS

In questa parte, abbiamo catturato il traffico HTTPS, che è crittografato, quindi non sarà visibile come il traffico HTTP. Avviamo tcpdump per catturare il traffico HTTPS

Apriamo un browser web e andiamo su [www.netacad.com](http://www.netacad.com). Questo sito utilizza HTTPS, quindi il traffico sarà crittografato.

Facciamo clic su Accedi e inseriamo il nome utente e password NetAcad, poi clic su Avanti.

Chiudiamo il browser e torniamo al terminale. Premiamo CTRL+C per fermare la cattura.

Dopodichè andremo a visualizzare l'acquisizione HTTPS.

Apriamo il file httpsdump.pcap con Wireshark, come fatto in precedenza.

In Wireshark, filtriamo per traffico HTTPS utilizzando il filtro `tcp.port==443` (la porta 443 è utilizzata per HTTPS).

Clicchiamo su Applica.

Ora, sfogliamo i pacchetti e selezioniamo un pacchetto che contenga Dati applicazione. Tuttavia, a differenza di HTTP, i dati di HTTPS sono crittografati e non saranno visibili direttamente.

Espandiamo la sezione Secure Sockets Layer (SSL) o Transport Layer Security (TLS). Qui vedremo che i dati sono crittografati.

Selezioniamo la sezione Dati applicazione crittografati per osservare come i dati siano protetti.

In seguito chiudiamo tutte le finestre di Wireshark e fermiamo la macchina virtuale.

# Parte 2: Cattura e visualizza il traffico HTTPS

In questo passaggio, tcpdump ha catturato tutto il traffico di rete sulla nostra interfaccia di rete e salverà le informazioni in un file pcap. Questo file conterrà i pacchetti di rete relativi a tutto il traffico HTTPS, ma poiché HTTPS è crittografato, non potremo vedere facilmente il contenuto dei pacchetti senza decifrarli.

Quando visitiamo il sito, il browser e il server devono stabilire una connessione sicura. Questo avviene tramite un processo di "handshake" SSL/TLS, che è il primo passo per stabilire la connessione sicura. Il traffico scambiato durante questo processo è criptato, quindi anche se tcpdump cattura i pacchetti, non saremo in grado di vedere il contenuto dei dati.

Durante questo processo di handshake SSL/TLS:

Il server invia il proprio certificato SSL.

Viene effettuata la verifica della validità del certificato.

Viene negoziata una chiave segreta condivisa (session key) per criptare i dati.

Una volta che l'handshake è completato, il traffico tra il client e il server sarà criptato.

Una volta che abbiamo completato l'accesso e il traffico HTTPS è stato catturato, torniamo al terminale dove tcpdump ha registrato il traffico e premiamo CTRL+C per interrompere la cattura.

Con CTRL+C, abbiamo interrotto la registrazione dei pacchetti. A questo punto, il file httpsdump.pcap conterrà tutti i pacchetti che sono stati scambiati tra il browser e il server durante la navigazione del sito HTTPS.

Successivamente abbiamo aperto il file con Wireshark che ci ha mostrato il traffico catturato dopo aver applicato il filtro "tcp.port==443", e potremo vedere un sacco di pacchetti SSL/TLS. Tuttavia, come accennato in precedenza, non vedremo il contenuto del traffico. Wireshark ci mostrerà che i pacchetti sono criptati e potrebbe etichettarli come "Encrypted Application Data" (Dati applicativi criptati).

Anche se i dati dell'applicazione sono criptati, Wireshark ci mostrerà la fase di handshake SSL/TLS, che è visibile come pacchetti separati.

Durante l'handshake, potremo vedere informazioni come:

Il certificato del server e la negoziazione della chiave segreta condivisa.

Dopo l'handshake, il traffico sarà cifrato e visibile solo come "dati applicazione crittografati".

# CONSIDERAZIONI

La principale differenza tra la cattura del traffico in HTTP e quello in HTTPS riguarda il livello di protezione e visibilità dei dati trasmessi. Ecco alcune considerazioni chiave sulle due situazioni:

## Cattura del traffico HTTP

- Non Crittografato: Quando il traffico è in HTTP, i dati che vengono trasmessi tra il client (ad esempio, il browser) e il server non sono crittografati. Ciò significa che chiunque stia monitorando il traffico di rete può facilmente vedere:
  - URL richieste: i siti web visitati.
  - Dati inviati nelle richieste POST o GET: come i parametri nei moduli (ad esempio, nome utente e password inviati in un modulo di login).
  - Contenuti delle risposte: il contenuto delle pagine web, come il testo, le immagini e altri dati.
- In un ambiente HTTP, un attaccante che intercetta il traffico, ad esempio, tramite un attacco di tipo Man-in-the-Middle (MITM), può facilmente leggere, manipolare e alterare i dati scambiati. Ad esempio, se un utente invia una password in un modulo di login, un malintenzionato che intercetta il traffico può leggere quella password in chiaro.
- Facilità di Monitoraggio: Wireshark o tcpdump, quando utilizzati per catturare il traffico HTTP, possono facilmente decodificare e visualizzare i contenuti, perché non c'è alcuna protezione contro l'intercettazione e la decodifica dei pacchetti.

# CONSIDERAZIONI

## Cattura del traffico HTTPS

- Crittografato: Con HTTPS, i dati sono crittografati utilizzando il protocollo SSL/TLS. Questo significa che anche se un attaccante intercetta il traffico, non sarà in grado di leggere o modificare i dati senza la chiave di decrittazione.
  - Handshake SSL/TLS: Quando una connessione HTTPS viene stabilita, avviene un "handshake" tra il client e il server, in cui viene negoziata una chiave segreta condivisa. I dati inviati tra il client e il server sono protetti da questa chiave, e l'algoritmo di crittografia impedisce a chi intercetta il traffico di accedere ai dati sensibili.
  - Sicurezza delle credenziali: Ad esempio, se un utente inserisce una password su un sito HTTPS, la password verrà cifrata prima di essere inviata al server, impedendo che venga letta da chi intercetta il traffico.
- Impossibilità di leggere il traffico: Quando si cattura il traffico HTTPS con strumenti come Wireshark o tcpdump, non è possibile vedere il contenuto dei pacchetti senza la chiave privata del server (o senza il processo di decrittazione, che richiede l'accesso ai certificati e alle chiavi). Anche se catturiamo i pacchetti HTTPS, quello che vediamo sono pacchetti cifrati o messaggi di handshake, ma non il contenuto effettivo (come i dati della sessione HTTP, come i parametri dei moduli di login).
- Protezione contro l'intercettazione: Con HTTPS, anche se un attaccante riesce a catturare il traffico, i dati criptati sono inutilizzabili senza la chiave corretta. L'intercettazione del traffico HTTPS non permette di leggere o manipolare i dati senza rompere la crittografia, il che è estremamente difficile da fare se l'implementazione di SSL/TLS è adeguata.

# CONCLUSIONI

In conclusione, la cattura del traffico HTTPS è molto più sicura rispetto al traffico HTTP. Tuttavia, Wireshark o tcpdump possono ancora essere utilizzati per analizzare il traffico HTTPS, ma i dati rimarranno criptati e non leggibili senza l'accesso alle chiavi di decodifica. D'altra parte, l'analisi del traffico HTTP è più facile, ma meno sicura, poiché i dati sono visibili e vulnerabili a intercettazioni e attacchi.

# GRAZIE



Nichol Galessiere

13/12/2024

