

Utilizzo di Windows PowerShell

L'obiettivo principale di questo esercizio è acquisire familiarità con alcune delle funzionalità base di PowerShell, esplorando comandi comuni, cmdlet e strumenti di diagnostica di rete.

Le attività si suddividono in:

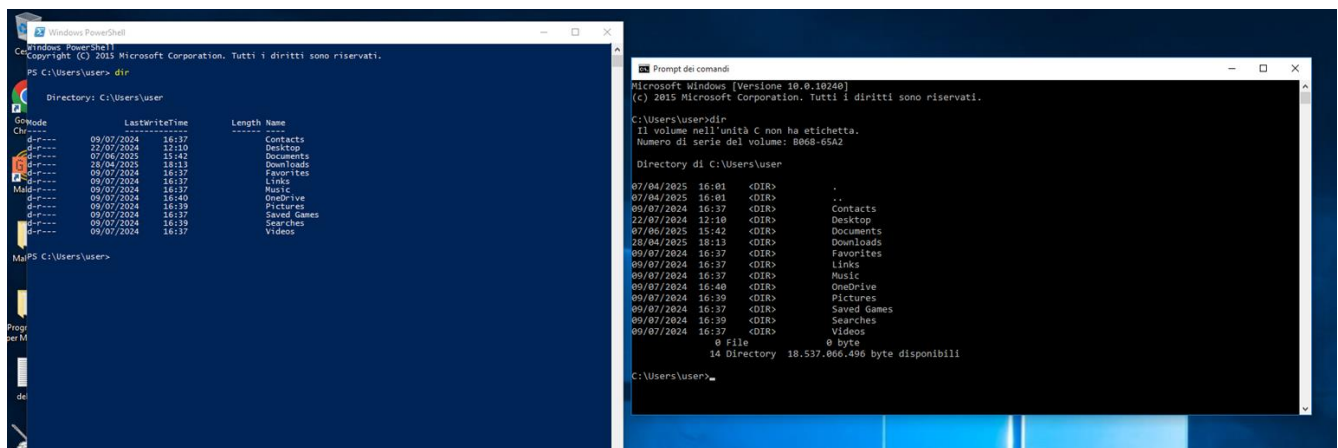
- Accesso alla console PowerShell
- Confronto tra comandi del Prompt dei Comandi e PowerShell
- Analisi dei cmdlet
- Utilizzo del comando netstat per la rete
- Pulizia del Cestino tramite comando PowerShell

Parte 1: Avvio di PowerShell

La prima fase ha previsto l'apertura di PowerShell dal menu Start. In parallelo è stato avviato anche il Prompt dei Comandi, per permettere un confronto diretto tra i due ambienti. Sebbene visivamente simili, PowerShell si distingue per il supporto a comandi più evoluti e una struttura più orientata alla programmazione e automazione.

Parte 2: Confronto tra comandi del Prompt e di PowerShell

Per iniziare, è stato utilizzato il comando dir in entrambi gli ambienti. Il risultato è stato simile: l'elenco dei file e delle cartelle della directory corrente. Tuttavia, PowerShell fornisce un output più chiaro e dettagliato.



Successivamente sono stati testati comandi di rete come cd .. e ipconfig. Entrambi funzionano anche in PowerShell, dimostrando una compatibilità con i comandi classici del Prompt, pur offrendo maggiore versatilità.

```
Windows PowerShell
d-r-- 09/07/2024 16:37 -----
d-r-- 22/07/2024 12:10 Desktop
d-r-- 07/06/2025 15:42 Documents
d-r-- 26/04/2025 18:13 Downloads
d-r-- 09/07/2024 16:37 Favorites
d-r-- 09/07/2024 16:37 Links
d-r-- 09/07/2024 16:37 Music
d-r-- 09/07/2024 16:40 OneDrive
d-r-- 09/07/2024 16:39 Pictures
d-r-- 09/07/2024 16:37 Saved Games
d-r-- 09/07/2024 16:39 Searches
d-r-- 09/07/2024 16:37 Videos

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

Configurazione IP di Windows

Scheda Ethernet Ethernet:

    Suffisso DNS specifico per connessione:
    Indirizzo IPv6 locale rispetto al collegamento . : fe80::102d:836c:a57:ea6c%4
    Indirizzo IPv4. . . . . : 192.168.108.11
    Subnet mask . . . . . : 255.255.255.0
    Gateway predefinito . . . . . : 192.168.108.170

Scheda Tunnel isatap.{92D61F82-1D19-45C9-B7CF-2E5AF2D63627}:

    Stato supporto. . . . . : Supporto disconnesso
    Suffisso DNS specifico per connessione:

Scheda Tunnel Teredo Tunneling Pseudo-Interface:

    Suffisso DNS specifico per connessione:
    Indirizzo IPv6 . . . . . : 2001:0:2001::2001:0:048:68ed:8ee0
    Indirizzo IPv6 locale rispetto al collegamento . : fe80::2091:c048:68ed:8ee0%5
    Gateway predefinito . . . . . : ::

PS C:\Users>
```

Parte 3: Scoperta dei cmdlet

PowerShell introduce i **cmdlet**, comandi strutturati nella forma Verbo-Nome (es. Get-Process, Set-Date), pensati per essere autoesplicativi.

Ad esempio, il comando dir in PowerShell è in realtà un alias di Get-ChildItem. Questo approccio favorisce una maggiore coerenza e leggibilità, soprattutto quando si creano script più complessi per l'automazione.

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

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

PS C:\Users>
```

Parte 4: Analisi di rete con netstat

Utilizzando il comando netstat -r, è stato possibile consultare la tabella di routing del sistema. Il dato più rilevante è stato l'individuazione del **gateway IPv4** (192.168.108.170), che rappresenta il punto di accesso verso la rete esterna.

```
Windows PowerShell
PS C:\Users\user> netstat -r

=====
Elenco interfacce
4...08 00 27 c3 09 10 .....Intel(R) PRO/1000 MT Desktop Adapter
1.....Software Loopback Interface 1
6...00 00 00 00 00 00 e0 Microsoft ISATAP Adapter
5...00 00 00 00 00 00 e0 Microsoft Teredo Tunneling Adapter
=====

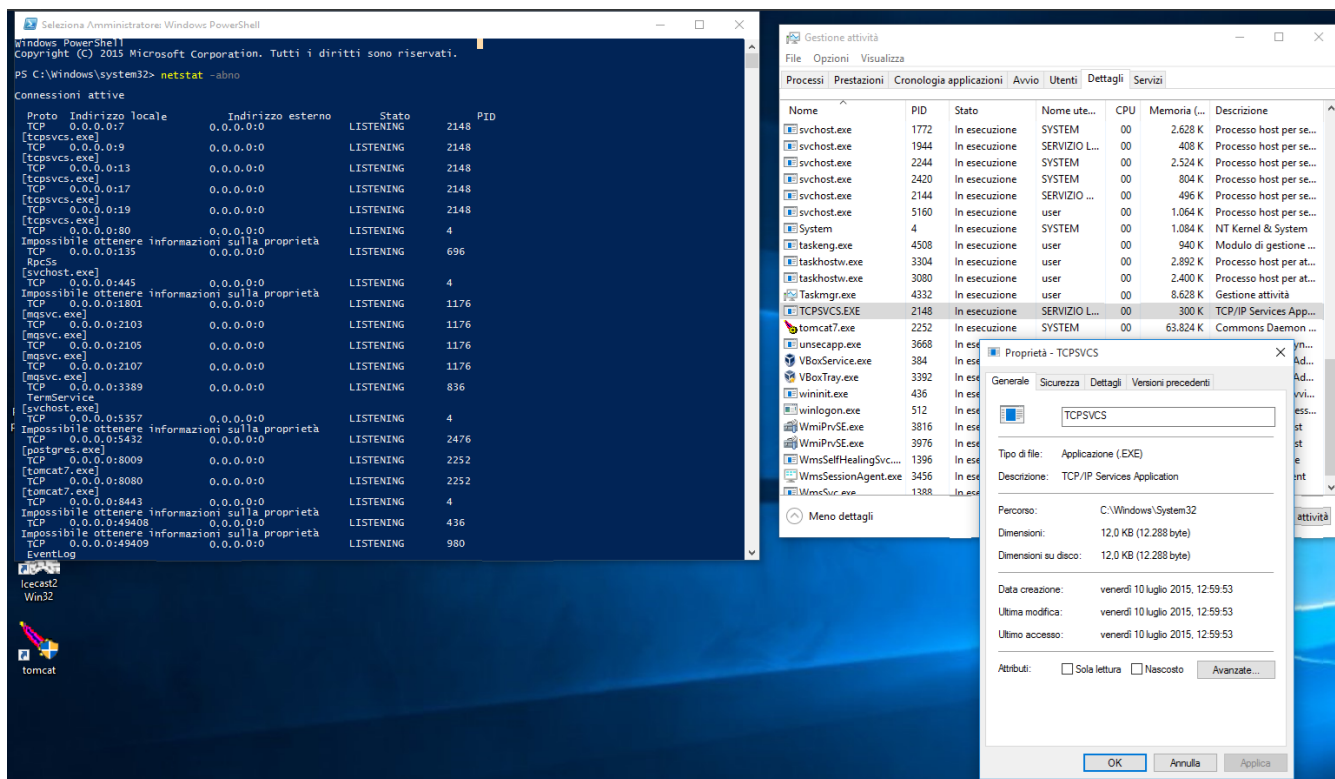
IPv4 Tabella route
Route attive:
=====
Indirizzo rete      Mask      Gateway    Interfaccia  Metrica
-----
0.0.0.0             0.0.0.0   192.168.108.170  192.168.108.11  10
127.0.0.0           255.0.0.0   On-link        127.0.0.1       306
127.0.0.1           255.255.255.255   On-link        127.0.0.1       306
127.255.255.255     255.255.255.255   On-link        127.0.0.1       306
192.168.108.0       255.255.255.0   On-link        192.168.108.11  266
192.168.108.11     255.255.255.255   On-link        192.168.108.11  266
192.168.108.255    255.255.255.255   On-link        192.168.108.11  266
224.0.0.0           240.0.0.0   On-link        127.0.0.1       306
224.0.0.0           240.0.0.0   On-link        192.168.108.11  266
255.255.255.255     255.255.255.255   On-link        127.0.0.1       306
255.255.255.255     255.255.255.255   On-link        192.168.108.11  266
=====

Route permanenti:
Nessuna

IPv6 Tabella route
Route attive:
=====
Interf Metrica Rete Destinazione Gateway
-----
5 306 ::/0 On-link
1 306 ::1/128 On-link
5 306 2001::/32 On-link
5 306 2001:0:2851:782c:109f:c001:68ed:8ee0/128 On-link
4 266 fe80::/64 On-link
5 306 fe80::/64 On-link
4 266 fe80::102d:836c:a57:ea6c/128 On-link
5 306 fe80::109f:c001:68ed:8ee0/128 On-link
1 306 ff00::/8 On-link
4 266 ff00::/8 On-link
5 306 ff00::/8 On-link
=====

Route permanenti:
Nessuna
PS C:\Users\user>
```

Con netstat -abno, è stata visualizzata la lista delle connessioni attive, assieme ai processi che le gestiscono. È stato analizzato in particolare il **PID 2148**, associato al processo tcpsvcs.exe. Esplorandolo tramite **Gestione Attività**, la finestra “Proprietà” ha mostrato informazioni chiave come il percorso eseguibile, il produttore Microsoft e la firma digitale. Questi dati sono fondamentali per riconoscere processi legittimi da quelli potenzialmente malevoli.



Parte 5: Svuotare il Cestino con PowerShell

Infine, è stato provato il comando `Clear-RecycleBin` per svuotare il Cestino. Dopo avervi spostato manualmente alcuni file, il comando è stato eseguito, eliminando definitivamente i contenuti senza dover ricorrere all'interfaccia grafica.

```
[svchost.exe]
PS C:\Windows\system32> 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"): s
PS C:\Windows\system32>
```

Riflessione: PowerShell per la Sicurezza Informatica

PowerShell è particolarmente utile nel contesto della cybersecurity grazie alla sua capacità di automatizzare compiti e raccogliere informazioni in modo efficiente. Alcuni comandi rilevanti per la sicurezza includono:

- `Get-EventLog` – per consultare i log di sistema, utile in fase di auditing o incident response.
- `Get-Process` – per monitorare i processi in esecuzione.
- `Get-NetTCPConnection` – per analizzare le connessioni di rete e identificare eventuali attività sospette.

- Get-LocalUser e Get-LocalGroup – per gestire utenti e gruppi locali e controllare accessi non autorizzati.

Conclusione

PowerShell si è dimostrato uno strumento avanzato per la gestione dei sistemi Windows, combinando la potenza di uno scripting engine con la semplicità di una console testuale.

Rispetto al Prompt dei Comandi, offre funzionalità decisamente più evolute, rendendolo indispensabile in contesti professionali dove automazione, controllo e sicurezza sono elementi cruciali.