

Report Analitico - Esercizio 1: Usare Windows PowerShell

Studente: Nicola Cassandra

Data: 20/02/2026

Obiettivo: Imparare ad utilizzare Windows PowerShell

Parte 1 e 2: Esplorare i comandi del Prompt dei Comandi e di PowerShell

In questa fase iniziale abbiamo confrontato il comportamento dei due ambienti a riga di comando di Windows eseguendo comandi di base.

- **Quali sono gli output del comando `dir`?** Come si evince dall'immagine, l'output è differente nei due terminali. Nel Prompt dei Comandi classico, `dir` restituisce un semplice elenco testuale che mostra data, ora, la dicitura `<DIR>` o la dimensione in byte, e il nome del file. In PowerShell, invece, l'output è strutturato sotto forma di *oggetti* e viene presentato in una tabella con colonne ben definite: **Mode** (i permessi e gli attributi), **LastWriteTime**, **Length** e **Name**.

The image shows two side-by-side terminal windows. The left window is Windows PowerShell, and the right window is the classic Windows Command Prompt. Both windows show the output of the `dir` command in the `C:\Users\User` directory.

Windows PowerShell Output:

```
PS C:\Users\User> dir

Directory: C:\Users\User

Mode                LastWriteTime         Length Name
----                -
d-r--             08/09/2024   23:19             30 Objects
d-r--             08/09/2024   23:19             Contacts
d-r--             08/09/2024   23:19             Desktop
d-r--             08/09/2024   23:19             Documents
d-r--             08/09/2024   23:19             Downloads
d-r--             08/09/2024   23:19             Favorites
d-r--             08/09/2024   23:19             Links
d-r--             08/09/2024   23:19             Music
d-r--             20/02/2026   10:38             OneDrive
d-r--             08/09/2024   23:22             Pictures
d-r--             08/09/2024   23:19             Saved Games
d-r--             08/09/2024   23:21             Searches
d-r--             08/09/2024   23:19             Videos
```

Windows Command Prompt Output:

```
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

20/02/2026  10:38  <DIR>          .
20/02/2026  10:38  <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
20/02/2026  10:38  <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       57.986.088.960 byte disponibili

C:\Users\User>
```

- **Quali sono i risultati di comandi come `ping`, `cd` e `ipconfig`?** Guardando l'immagine che segue questa risposta (che mostra l'esecuzione di `ipconfig`), notiamo che i risultati sono **identici** in entrambe le finestre. PowerShell è infatti progettato per essere retrocompatibile e riconosce gli eseguibili standard di Windows (come `ipconfig.exe` o `ping.exe`), restituendo lo stesso identico output di rete.

```
PS C:\Users\User> ipconfig
Configurazione IP di Windows

Scheda Ethernet Ethernet:
    Suffisso DNS specifico per connessione: homenet.telecomitalia.it
    Indirizzo IPv6 . . . . . : fd17:625c:f037:2:7899:5949:534:2173
    Indirizzo IPv6 temporaneo. . . . . : fd17:625c:f037:2:a1be:fe77:4521:fb9d
    Indirizzo IPv6 locale rispetto al collegamento . : fe80::7de5:ce64:b266:fed3%5
    Subnet mask . . . . . : 255.255.255.0
    Gateway predefinito . . . . . : fe80::2%5
    10.0.2.2

PS C:\Users\User> netstat
Connessioni attive

Proto Indirizzo locale Indirizzo esterno Stato
TCP 10.0.2.15:49759 4.207.247.139:https ESTABLISHED
TCP 10.0.2.15:49805 199.232.214.172:http TIME_WAIT
TCP 10.0.2.15:49806 199.232.214.172:http TIME_WAIT
TCP 10.0.2.15:49809 199.232.214.172:http TIME_WAIT
TCP 10.0.2.15:49811 135.232.92.97:https TIME_WAIT
TCP 10.0.2.15:49813 a2-16-70-4:http ESTABLISHED
TCP 10.0.2.15:49819 a2-20-242-17:https CLOSE_WAIT
TCP 10.0.2.15:49822 4.207.247.139:https ESTABLISHED
TCP 10.0.2.15:49824 172.184.231.71:https TIME_WAIT
TCP 10.0.2.15:49825 a23-55-48-58:http TIME_WAIT
TCP 10.0.2.15:49826 135.232.92.97:https TIME_WAIT
TCP 10.0.2.15:49827 172.184.231.71:https TIME_WAIT
TCP 10.0.2.15:49828 a23-55-48-58:http TIME_WAIT
TCP 10.0.2.15:49829 135.232.92.97:https TIME_WAIT
TCP 10.0.2.15:49830 128.85.113.134:https TIME_WAIT
TCP 10.0.2.15:49831 109.61.38.38:http TIME_WAIT
TCP 10.0.2.15:49832 135.232.92.97:https TIME_WAIT
TCP 10.0.2.15:49834 a2-20-114-43:https ESTABLISHED
TCP 10.0.2.15:49835 a2-20-114-43:https ESTABLISHED
TCP 10.0.2.15:49836 a2-20-114-43:https ESTABLISHED
TCP 10.0.2.15:49839 a2-20-114-43:https ESTABLISHED
TCP 10.0.2.15:49843 a2-20-114-43:https ESTABLISHED
TCP 10.0.2.15:49844 72.146.92.132:http TIME_WAIT
TCP 10.0.2.15:49846 72.146.92.132:http TIME_WAIT
TCP 10.0.2.15:49849 40.126.53.6:https ESTABLISHED

PS C:\Users\User>
```

Parte 3: Esplorare i cmdlet

PowerShell utilizza una struttura "Verbo-Nome" per i suoi comandi nativi (i *cmdlet*), ma permette l'uso di comandi legacy tramite un sistema di alias.

- **Qual è il comando PowerShell per *dir*?** Eseguendo il comando *Get-Alias dir*, scopriamo che la riga di comando ci restituisce la mappatura esatta: il vero cmdlet eseguito dietro le quinte da PowerShell quando digitiamo *dir* è *Get-ChildItem*.

```
Windows PowerShell

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

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

Parte 4: Esplorare il comando *netstat* usando PowerShell

Questa sezione del laboratorio si è concentrata sull'analisi delle connessioni di rete e delle tabelle di routing.

- **Qual è il gateway IPv4?** Osservando l'output del comando *netstat -r* (Tabella di routing) nell'immagine al di sotto, possiamo identificare la rotta predefinita (quella con Indirizzo rete *0.0.0.0* e Mask *0.0.0.0*). Il Gateway IPv4 associato a questa rotta

sulla tua macchina virtuale è **10.0.2.2**.

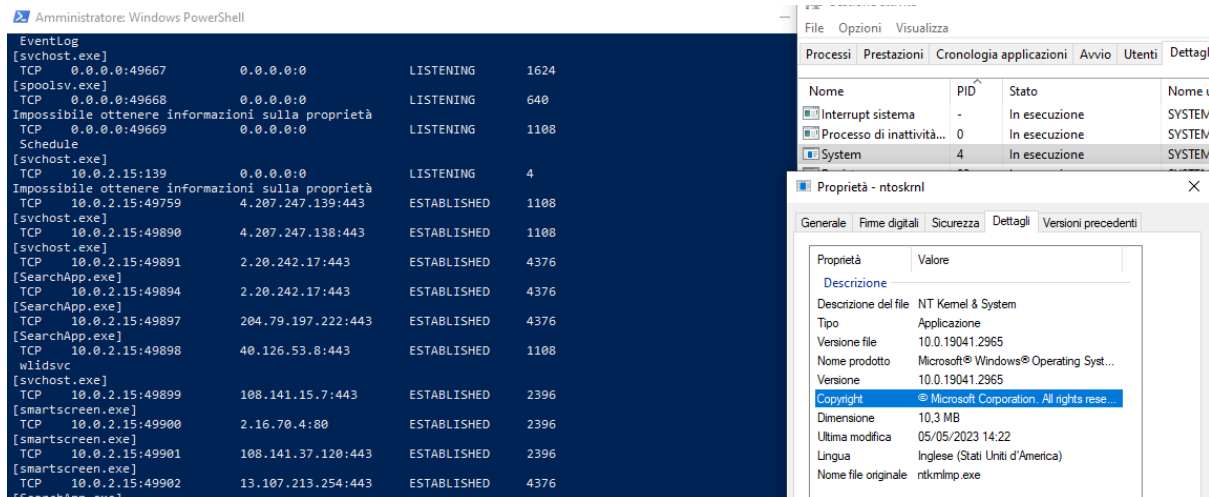
```
PS C:\Users\User> netstat -r
=====
Elenco interfacce
 5...08 00 27 96 c2 10 .....Intel(R) PRO/1000 MT Desktop Adapter
 1.....Software Loopback Interface 1
=====

IPv4 Tabella route
=====
Route attive:
  Indirizzo rete      Mask      Gateway      Interfaccia  Metrica
  0.0.0.0             0.0.0.0    10.0.2.2     10.0.2.15    25
  10.0.2.0            255.255.255.0  On-link      10.0.2.15    281
  10.0.2.15           255.255.255.255  On-link      10.0.2.15    281
  10.0.2.255          255.255.255.255  On-link      10.0.2.15    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
  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      10.0.2.15    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      10.0.2.15    281
=====
Route permanenti:
 Nessuna

IPv6 Tabella route
=====
Route attive:
  Interf Metrica Rete Destinazione      Gateway
  5      281  ::/0             fe80::2
  1      331  ::1/128          On-link
  5      281  fd17:625c:f037:2::/64  On-link
  5      281  fd17:625c:f037:2:7899:5949:534:2173/128  On-link
  5      281  fd17:625c:f037:2:a1be:fe77:4521:fb9d/128  On-link
  5      281  fe80::/64        On-link
  5      281  fe80::7de5:ce64:b266:fed3/128  On-link
  1      331  ff00::/8         On-link
  5      281  ff00::/8         On-link
=====
Route permanenti:
 Nessuna
```

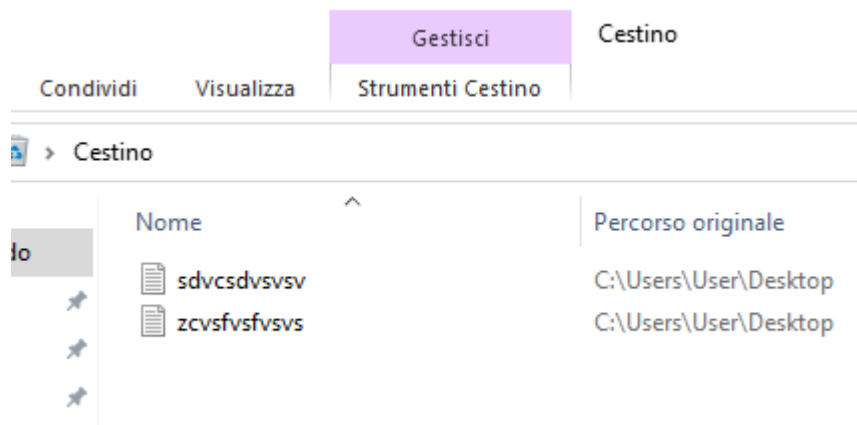
- **Quali informazioni puoi ottenere dalla scheda *Dettagli* e dalla finestra di dialogo *Proprietà* per il PID selezionato?** Nell'immagine che segue al di sotto vediamo l'uso di PowerShell come Amministratore e l'apertura del Task Manager per indagare su un processo specifico. Dalla scheda *Dettagli* otteniamo informazioni sullo stato dell'esecuzione, l'utente che lo ha lanciato (es. SYSTEM), l'utilizzo di CPU e memoria. Aprendo le *Proprietà* (in questo caso per `ntoskrnl.exe`), otteniamo dati cruciali per l'analisi forense e di sicurezza:
 - Descrizione del file (NT Kernel & System)
 - Versione del file e del prodotto
 - Copyright (che aiuta a confermare se il file è legittimo, es. Microsoft Corporation)
 - Dimensione e data di ultima modifica

- Il nome originale del file (**ntkrnlmp.exe**).



Parte 5: Svuotare il cestino usando PowerShell

L'automazione tramite PowerShell permette di eseguire rapidamente task di sistema che richiederebbero più clic nell'interfaccia grafica.



- **Cosa è successo ai file nel Cestino?** Come documentato nelle immagini che seguono la risposta, dopo aver eseguito il comando **clear-recyclebin** e aver confermato premendo **S** (Sì), i file sono stati **eliminati in modo permanente** dal PC

e la cartella del cestino risulta completamente vuota.

