Esplorazione di Windows PowerShell

Michele Storelli

13 Giugno 2025

Obiettivi

L'obiettivo di questo laboratorio è esplorare alcune delle funzioni di PowerShell, nello specifico:

- Accedere alla console PowerShell.
- Esplorare i comandi del Prompt dei Comandi e di PowerShell.
- Esplorare i cmdlet.
- Esplorare il comando `netstat` usando PowerShell.
- Svuotare il cestino usando PowerShell.

Contesto / Scenario

PowerShell è un potente strumento di automazione, fungendo sia da console di comando che da linguaggio di scripting. Questo laboratorio si concentra sull'utilizzo della console per eseguire comandi disponibili sia nel prompt dei comandi che in PowerShell, evidenziando le capacità di PowerShell nell'automazione e nella gestione del sistema operativo Windows.

Risorse Richieste

- 1 PC Windows con PowerShell installato e accesso a internet.

Risposte alle Domande

Esplorare i comandi del Prompt dei Comandi e di PowerShell.

a. Inserisci `dir` al prompt in entrambe le finestre. Quali sono gli output del comando `dir`?

Come mostrato nell'immagine 1, l'output del comando `dir` sia in PowerShell (a sinistra) che nel Prompt dei Comandi (a destra) è simile, ma con alcune differenze nel formato. Entrambi i comandi elencano il contenuto della directory corrente (`C:\Users\miche`).

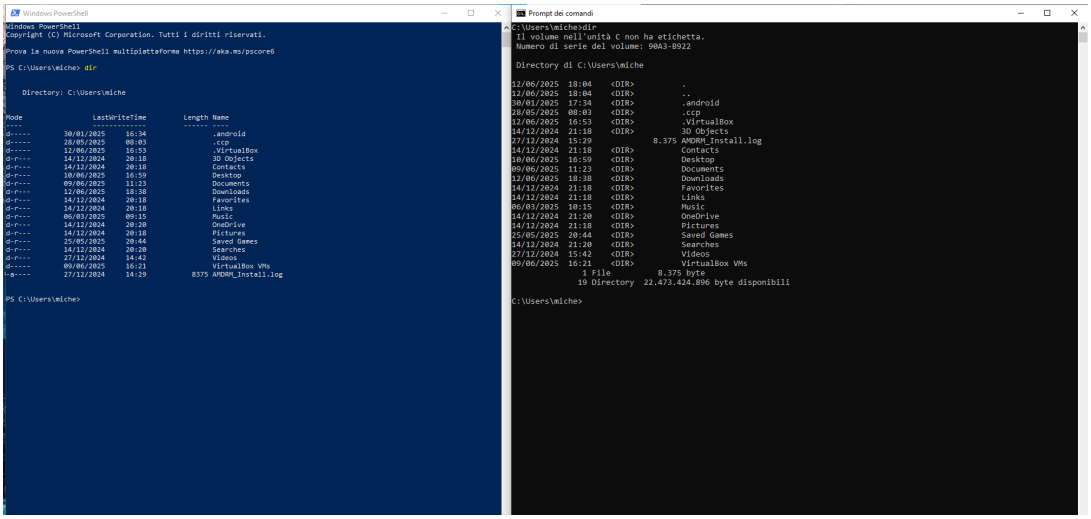

Figure 1: Output del comando `dir` in PowerShell (sinistra) e Prompt dei Comandi (destra).

- **PowerShell (sinistra):** Presenta una colonna "Mode" che indica il tipo di elemento (d per directory, -a- per file), "LastWriteTime" (data e ora dell'ultima modifica), "Length" (dimensione per i file, vuota per le directory) e "Name".
- **Prompt dei Comandi (destra):** Mostra la data e l'ora, il tag <DIR> per le directory, la dimensione per i file, e il nome. Inoltre, fornisce un riepilogo finale del numero di file, directory e spazio disponibile.

Prova un altro comando che hai usato nel prompt dei comandi, come `ping`, `cd` e `ipconfig`. Quali sono i risultati? Qual è il comando PowerShell per `dir`?

Come mostrato nell'immagine 2, il comando `ipconfig` è stato eseguito con successo in PowerShell (sinistra) e nel Prompt dei Comandi (destra). Entrambi gli output sono praticamente identici, mostrando la configurazione IP per le schede di rete.

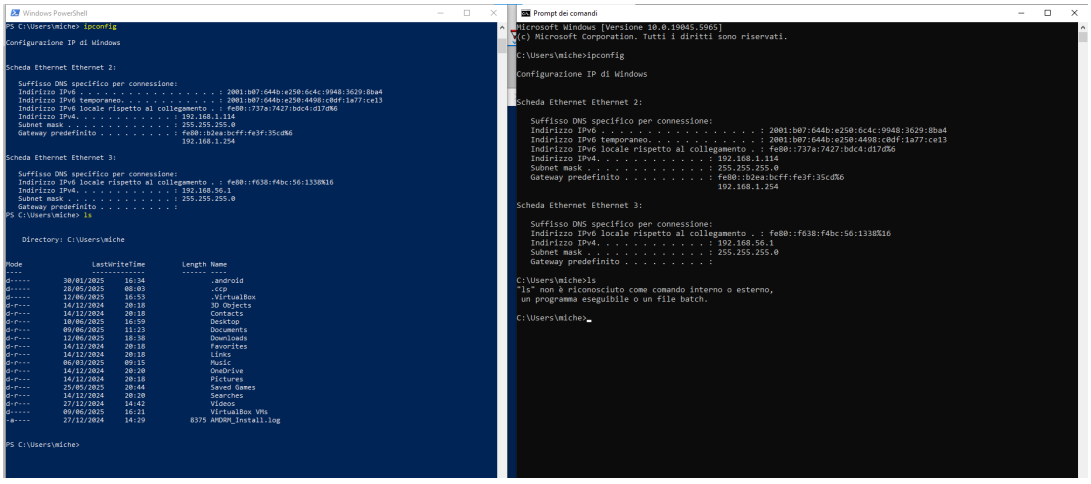

Figure 2: Output del comando `ipconfig` in PowerShell (sinistra) e Prompt dei Comandi (destra).

Il comando `cd` (Change Directory) funziona allo stesso modo in entrambi gli ambienti per navigare tra le directory. Il comando `ping` (non mostrato direttamente in queste immagini, ma funzionerebbe in modo simile) viene anche riconosciuto ed eseguito in entrambi.

Il comando PowerShell equivalente per `dir` è `Get-ChildItem`. Come evidenziato nell'immagine 3, PowerShell utilizza gli alias per i comandi più comuni del Prompt dei Comandi per facilitare la transizione degli utenti. L'output di `Get-ChildItem` è identico a quello di `dir` in PowerShell.

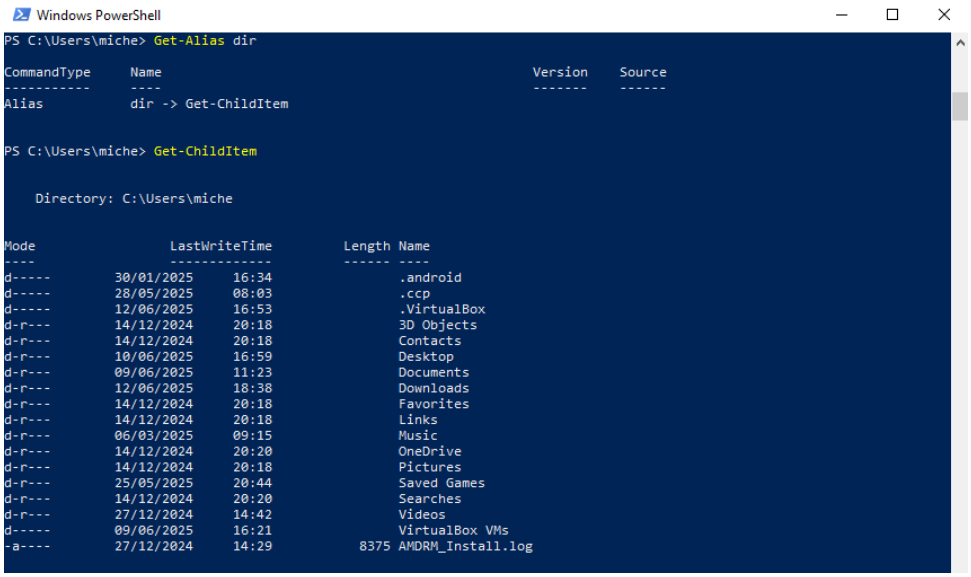

Figure 3: Alias per `dir` e output di `Get-ChildItem` in PowerShell.

Per visualizzare la tabella di routing con le rotte attive, inserisci `netstat -r` al prompt. Qual è il gateway IPv4?

L'output del comando `netstat -r` è mostrato nell'immagine 4.

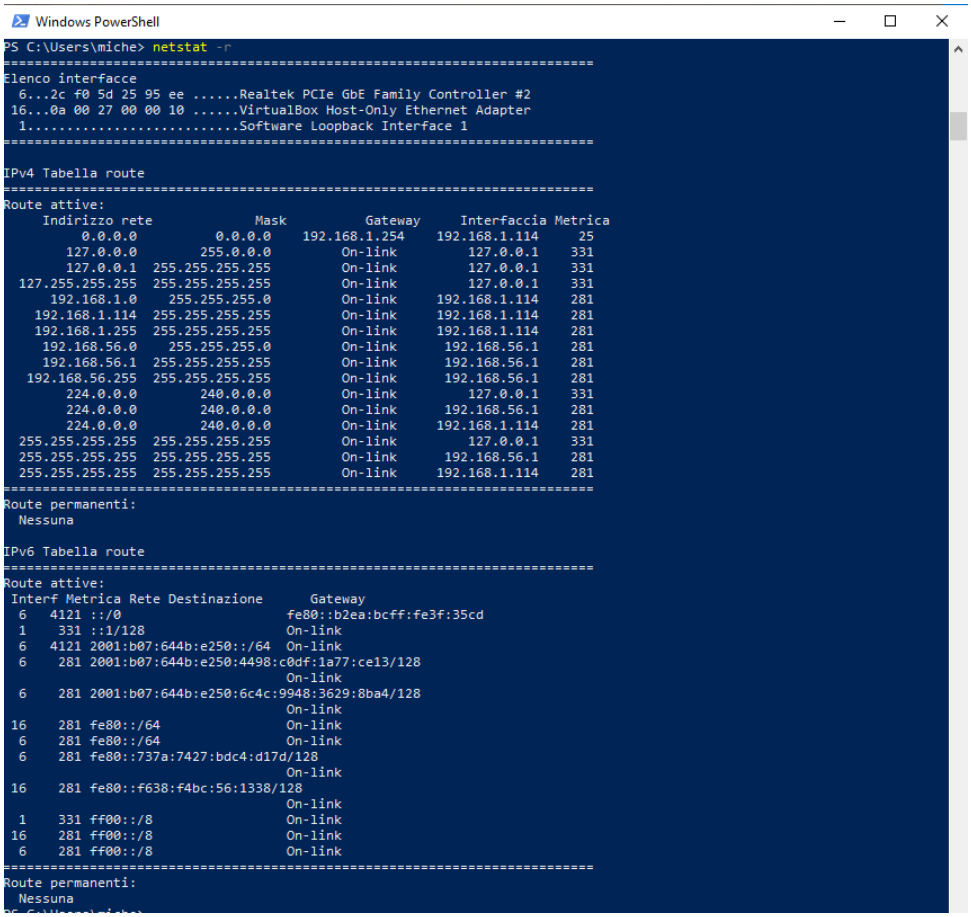

Figure 4: Output del comando `netstat -r` in PowerShell.

Dalla sezione "IPv4 Tabella route" e in particolare dalla riga con "Indirizzo rete" `0.0.0.0` (la rotta

predefinita), possiamo identificare il gateway IPv4.

Il gateway IPv4 è `192.168.1.254`.

Quali informazioni puoi ottenere dalla scheda Dettagli e dalla finestra di dialogo Proprietà per il PID selezionato?

L'immagine 5 mostra la finestra "Gestione attività" e la finestra di dialogo "Proprietà" per un processo selezionato (in questo caso, `svchost.exe`).

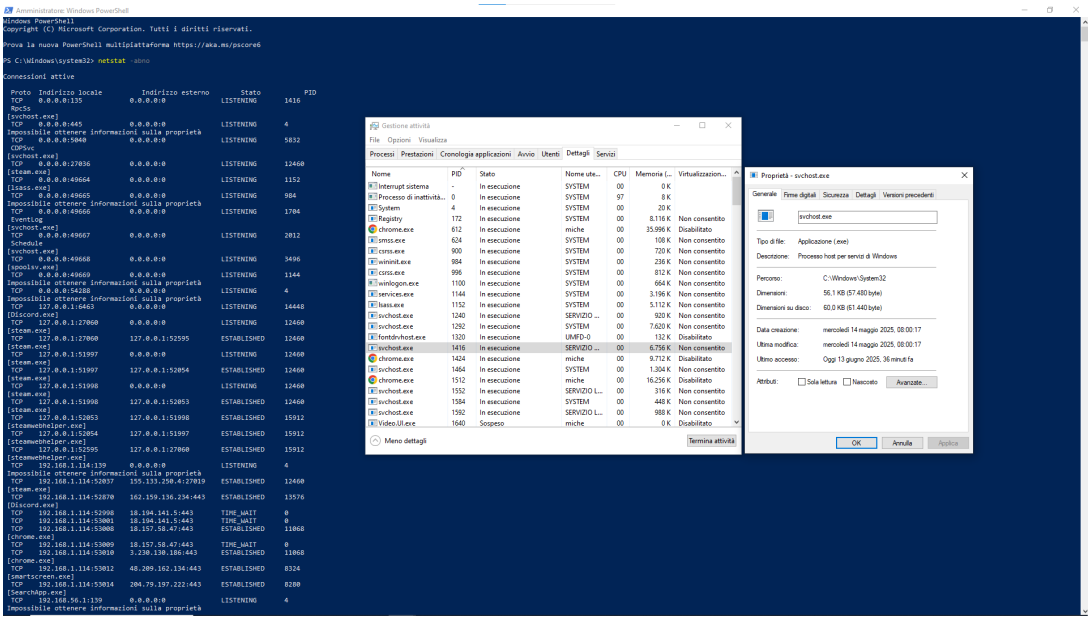

Figure 5: Gestione attività e Proprietà del processo `svchost.exe`.

Dalla scheda "Dettagli" nella finestra "Gestione attività", è possibile ottenere le seguenti informazioni per ogni processo (identificato dal PID - Process ID):

- **Nome:** Nome del file eseguibile (es. `svchost.exe`, `chrome.exe`).
- **PID:** Identificatore univoco del processo.
- **Stato:** Stato corrente del processo (In esecuzione, Sospeso, ecc.).
- **Nome utente:** L'utente o il sistema account con cui il processo è in esecuzione (es. SYSTEM, miche, SERVIZIO LOCALE).
- **CPU:** Percentuale di utilizzo della CPU.
- **Memoria (set di lavoro privato):** Quantità di memoria RAM utilizzata dal processo.
- **Virtualizzazione:** Indica se la virtualizzazione è abilitata per il processo.
- **Descrizione:** Breve descrizione del processo.

Dalla finestra di dialogo "Proprietà" per il processo `svchost.exe` (raggiunta selezionando un processo e cliccando su "Proprietà" o "Apri percorso file" per poi accedere alle proprietà dell'eseguibile), in particolare dalla scheda "Dettagli", possiamo ottenere informazioni più specifiche sull'eseguibile:

- **Nome file:** `svchost.exe`
- **Tipo di file:** Applicazione (`.exe`)
- **Descrizione:** Processo host per servizi di Windows
- **Copyright:** (C) Microsoft Corporation. Tutti i diritti riservati.

- **Versione:** Versione del file (es. 10.0.19041.3636)
- **Dimensioni:** Dimensioni del file in byte (es. 56,1 MB).
- **Data creazione:** Data e ora di creazione del file.
- **Ultima modifica:** Data e ora dell'ultima modifica del file.
- **Ultimo accesso:** Data e ora dell'ultimo accesso al file.

Queste informazioni sono cruciali per l'analisi forense e la sicurezza, permettendo di identificare processi sospetti o versioni obsolete di software.

In una console PowerShell, inserisci `clear-recyclebin` al prompt. Cosa è successo ai file nel Cestino?

Come mostrato nell'immagine 6, dopo aver inserito `clear-recyclebin` al prompt, PowerShell richiede una conferma per l'operazione.

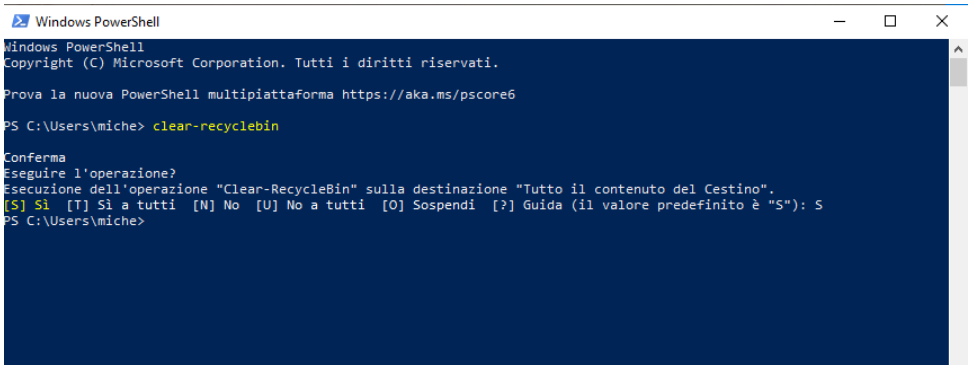

Figure 6: Esecuzione del comando `clear-recyclebin` in PowerShell.

Dopo aver risposto "S" (Sì) alla richiesta di conferma, **tutti i file presenti nel Cestino sono stati eliminati permanentemente**. L'operazione "Clear-RecycleBin" svuota il Cestino senza possibilità di recupero diretto tramite le funzionalità del sistema operativo.

Domanda di Riflessione

PowerShell è stato sviluppato per l'automazione delle attività e la gestione della configurazione. Usando internet, ricerca comandi che potresti usare per semplificare i tuoi compiti come analista di sicurezza. Registra le tue scoperte.

Come analista di sicurezza, PowerShell offre una vasta gamma di comandi (cmdlet) che possono semplificare notevolmente i compiti di monitoraggio, analisi e risposta. Ecco alcune scoperte e comandi utili:

- `Get-NetTCPConnection`: Questo cmdlet permette di visualizzare le connessioni TCP attive, inclusi gli stati (LISTENING, ESTABLISHED, TIME_WAIT, ecc.), gli indirizzi locali e remoti e le porte. È estremamente utile per identificare connessioni sospette o non autorizzate verso l'esterno o all'interno della rete.
- `Get-WinEvent`: Permette di recuperare eventi dai log di Windows (Sicurezza, Sistema, Applicazione, ecc.). Un analista può filtrare gli eventi per ID, origine, livello e tempo per identificare attività sospette, come tentativi di login falliti, modifiche a politiche di sicurezza, o installazioni di software.
  - **Esempio:** `Get-WinEvent -LogName Security -FilterXPath '*/System/EventID=4625'` per visualizzare i tentativi di login falliti.
- `Get-Process` e `Stop-Process`: `Get-Process` elenca tutti i processi in esecuzione con dettagli come PID, memoria e utilizzo della CPU. `Stop-Process` permette di terminare un processo, utile per bloccare un'applicazione malevola o un processo compromesso.

  - **Esempio:** `Get-Process | Where-Object _.CPU -gt 50` per trovare processi che consumano molta CPU, o `Stop-Process -Name "malware.exe"` per terminare un processo identificato.

- `Get-Service` e `Stop-Service`/`Start-Service`: Questi cmdlet consentono di visualizzare, avviare o arrestare i servizi di Windows. Un analista può utilizzarli per disabilitare servizi non necessari che potrebbero essere sfruttati, o per riavviare servizi compromessi.

- `Get-ItemProperty` e `Set-ItemProperty` (per il Registro di Sistema): PowerShell permette di interagire con il Registro di Sistema. Questo è fondamentale per verificare o modificare chiavi e valori che potrebbero essere stati alterati da malware per la persistenza o per alterare le impostazioni di sicurezza.

- `Get-AuthenticodeSignature`: Questo cmdlet verifica la firma digitale dei file. Può essere usato per determinare se un file è stato manomesso o se proviene da una fonte affidabile, un aspetto cruciale nella prevenzione e nell'analisi di malware.

- `Invoke-WebRequest`: Permette di inviare richieste HTTP/HTTPS. Può essere usato in script per scaricare intelligence sulle minacce, o per testare la connettività a server di comando e controllo (C2) noti in un ambiente controllato.

- `Test-NetConnection`: Offre un modo più robusto per testare la connettività di rete rispetto a `ping`, fornendo dettagli su ping, TCP handshake, route, ecc. Utile per la diagnostica di rete e la verifica delle regole del firewall.

- `Get-Content` e `Set-Content`: Per leggere e scrivere il contenuto dei file. Utile per analizzare log testuali, file di configurazione o per modificare script di sistema.

L'utilizzo di script PowerShell basati su questi cmdlet permette di automatizzare il monitoraggio della sicurezza, la risposta agli incidenti e la raccolta di informazioni forensi, riducendo significativamente il tempo e lo sforzo manuale. La capacità di PowerShell di interagire profondamente con il sistema operativo Windows lo rende uno strumento indispensabile per qualsiasi analista di sicurezza.