Windows PowerShell

Analisi Operativa e Verifica delle Funzionalità di Sistema

Ardelean Pop Catalin
Master in Cyber Security Specialist – Epicode

20 febbraio 2026

Indice

1 Traccia dell'esercizio **2**

2 Scenario di laboratorio **2**

3 Confronto tra Prompt dei Comandi e PowerShell **2**

3.1 Comando dir 2

3.1.1 Filtraggio e selezione avanzata 3

3.2 Altri comandi testati 4

3.2.1 Esempio pratico di confronto 4

4 Analisi dei Cmdlet **4**

5 Analisi del comando netstat **5**

5.1 Visualizzazione opzioni 5

5.2 Tabella di routing 5

5.3 Connessioni attive e PID 6

6 Analisi PID tramite Task Manager **6**

7 Svuotamento del Cestino **7**

8 Comandi utili per un analista di sicurezza **7**

9 Conclusioni **8**

1 Traccia dell'esercizio

L'esercizio ha l'obiettivo di esplorare le funzionalità di Windows PowerShell confrontandole con il Prompt dei Comandi tradizionale.

I passaggi richiesti sono:

- Accedere alla console PowerShell.
- Confrontare i comandi del Prompt dei Comandi e di PowerShell.
- Analizzare i cmdlet.
- Esplorare il comando netstat.
- Svuotare il Cestino tramite PowerShell.

L'attività è finalizzata alla comprensione dell'automazione e delle capacità di analisi offerte da PowerShell in ambito sicurezza.

2 Scenario di laboratorio

L'ambiente utilizzato è composto da:

- Sistema operativo: Windows 11 Pro
- PowerShell eseguito con privilegi standard e amministrativi
- Connessione di rete locale 192.168.1.0/24

Configurazione di rete rilevata:

- IP locale: 192.168.1.14
- Subnet mask: 255.255.255.0
- Gateway IPv4: 192.168.1.1

3 Confronto tra Prompt dei Comandi e PowerShell

3.1 Comando dir

Esecuzione del comando:

```
dir
```

Nel Prompt dei Comandi il comando mostra l'elenco dei file, le dimensioni e lo spazio disponibile sul disco. In PowerShell, `dir` è un alias di:

```
Get-ChildItem
```

PowerShell restituisce oggetti strutturati contenenti:

- Mode
- LastWriteTime
- Length
- Name

Questa struttura a oggetti consente elaborazioni avanzate tramite pipeline, ad esempio:

```
Get-ChildItem | Where-Object {$_.Length -gt 1MB} | Sort-Object
   Length -Descending
```

Il comando sopra elenca tutti i file con dimensione maggiore di 1 MB e li ordina dal più grande al più piccolo, operazione che nel Prompt dei Comandi richiederebbe script più complessi o l'uso di tool esterni.

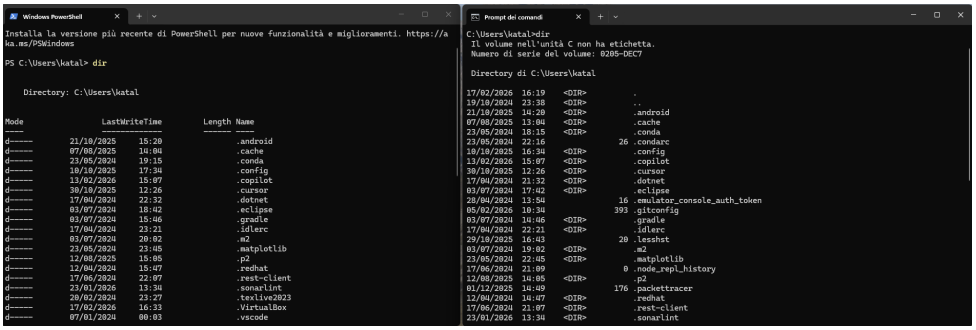

Figura 1: Output del comando dir in PowerShell e Prompt dei Comandi

3.1.1 Filtraggio e selezione avanzata

PowerShell permette di filtrare e selezionare solo determinate proprietà dei file:

```
Get-ChildItem | Select-Object Name, LastWriteTime, Length
```

In questo modo è possibile ottenere report mirati senza dover elaborare manualmente l'output testuale.

3.2 Altri comandi testati

```
ping 8.8.8.8
ipconfig
cd
```

Tutti questi comandi funzionano in PowerShell per compatibilità con il Prompt dei Comandi, ma PowerShell offre cmdlet più avanzati per gestione rete e sistema, come:

- `Test-Connection` — equivalente avanzato di `ping` con supporto per pipeline e report.
- `Get-NetIPAddress` — analizza in dettaglio le configurazioni IP del sistema.
- `Set-Location` — equivalente di `cd`, ma permette anche navigazione tramite provider diversi (registro di sistema, certificati, ecc.).

3.2.1 Esempio pratico di confronto

```
# Prompt dei Comandi
ping 8.8.8.8

# PowerShell avanzato
Test-Connection 8.8.8.8 -Count 4 | Select-Object Address,
   ResponseTime
```

Il cmdlet `Test-Connection` permette di ottenere direttamente valori strutturati, come indirizzo IP e tempi di risposta, rendendo più semplice l'integrazione in script di monitoraggio o reportistica automatica.

4 Analisi dei Cmdlet

Verifica alias:

```
Get-Alias dir
```

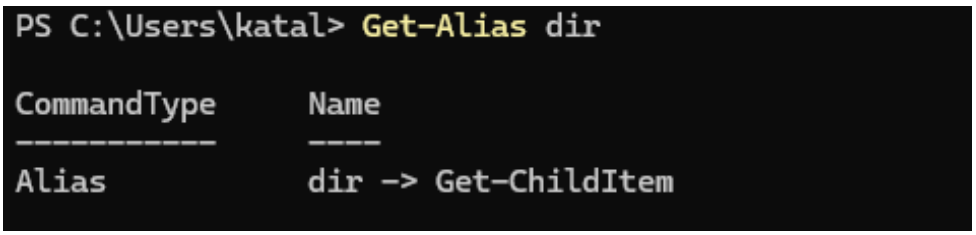

Figura 2: Output del comando Get-Alias dir in PowerShell

Risultato:
dir è alias di Get-ChildItem.
I cmdlet seguono struttura Verbo-Nome:

- Get-Process
- Get-Service
- Get-NetTCPConnection
- Clear-RecycleBin

Questa struttura garantisce coerenza e standardizzazione.

5 Analisi del comando netstat

5.1 Visualizzazione opzioni

```
netstat -h
```

Mostra opzioni come -a, -b, -n, -o, -r.

5.2 Tabella di routing

```
netstat -r
```

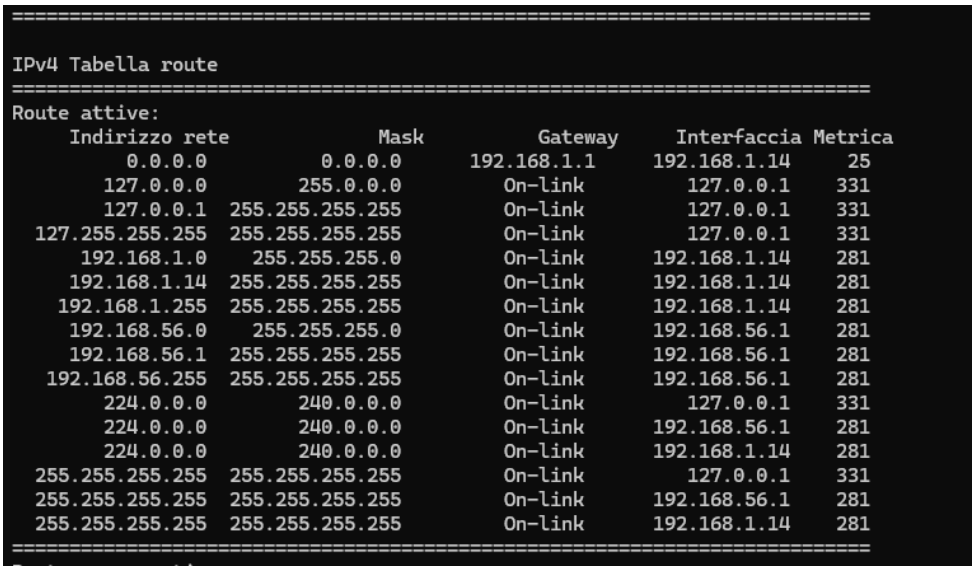

Figura 3: Output del comando netstat -r in PowerShell

Gateway IPv4 rilevato:
192.168.1.1
Rete di appartenenza: 192.168.1.0/24

5.3 Connessioni attive e PID

Esecuzione come amministratore:

```
netstat -abno
```

Analisi rilevata:

- Molte connessioni locali 127.0.0.1 associate a chrome.exe (PID 3400).
- Connessione HTTPS attiva verso 108.138.192.5:443.
- Porta 139 in LISTENING (NetBIOS).

6 Analisi PID tramite Task Manager

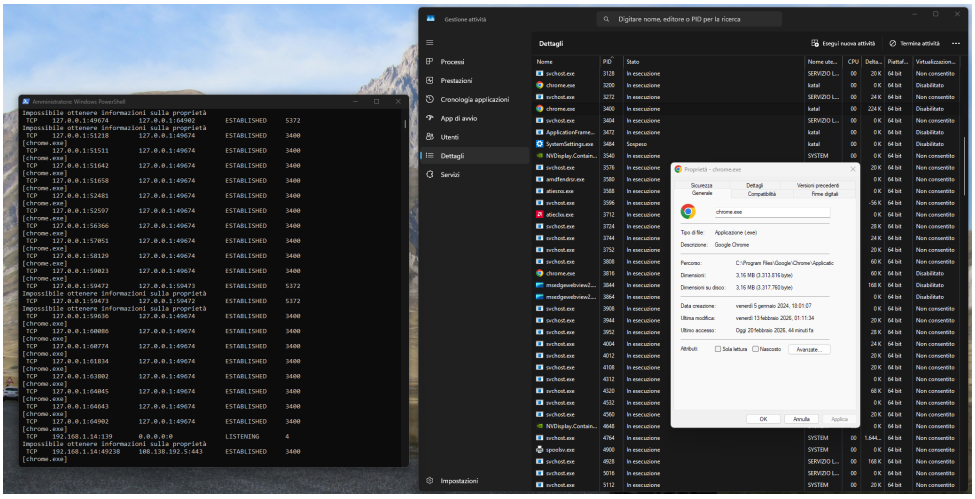

Figura 4: Task Manager con evidenza del processo chrome.exe associato al PID 3400

Il processo analizzato è:

- **PID**: 3400
- **Nome processo**: chrome.exe

Informazioni dettagliate rilevate tramite Task Manager:

- **Percorso**: C:\Program Files\Google\Chrome\Application Il percorso conferma l'eseguibile ufficiale di Google Chrome, utile per distinguere applicazioni legittime da possibili eseguibili malevoli.
- **Tipo**: Applicazione 64 bit Indica che il processo utilizza l'architettura a 64 bit del sistema, consentendo maggiore efficienza nella gestione della memoria e compatibilità con il sistema operativo.

- **Firma digitale valida** La presenza di una firma digitale valida conferma l'integrità del software e la provenienza da un fornitore attendibile.
- **Stato**: In esecuzione Il processo è attivo e impegnato nel fornire funzionalità all'utente, ad esempio la navigazione web.

Queste informazioni sono fondamentali in analisi forense e threat hunting, in quanto consentono di:

- Verificare l'identità del processo e la sua legittimità.
- Correlare attività di rete o consumo di risorse con processi specifici.
- Identificare comportamenti anomali rispetto ai processi attesi.

L'uso di Task Manager in questo contesto fornisce un rapido riscontro visivo e dettagli tecnico-operativo dei processi in esecuzione, indispensabile per analisi immediate o prime fasi di investigazione.

7 Svuotamento del Cestino

Comando eseguito:

```
Clear-RecycleBin
```

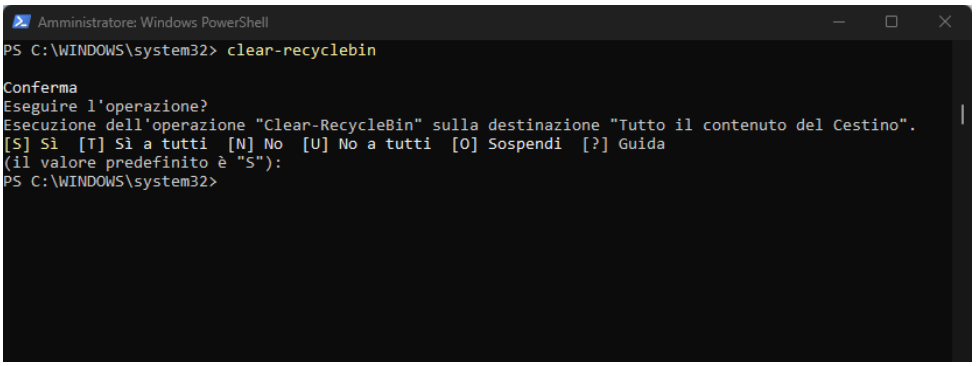

Figura 5: Prompt di conferma per svuotamento del Cestino in PowerShell

Dopo la conferma, i file sono stati eliminati definitivamente.
PowerShell consente quindi automazione di operazioni amministrative critiche.

8 Comandi utili per un analista di sicurezza

```
Get-Process
Get-Service
Get-NetTCPConnection
Get-EventLog -LogName Security
Get-LocalUser
Get-ExecutionPolicy
```

Questi cmdlet permettono:

- Monitoraggio processi
- Analisi servizi
- Verifica connessioni di rete
- Audit eventi di sicurezza
- Controllo utenti locali

9 Conclusioni

L'esercizio ha dimostrato che PowerShell è uno strumento fondamentale per:

- Automazione amministrativa
- Analisi di rete e processi
- Verifica sicurezza sistema
- Incident response di base

La capacità di correlare connessioni di rete ai processi tramite PID e analizzare la tabella di routing rappresenta una competenza essenziale in ambito Cyber Security.

PowerShell si conferma uno strumento chiave per amministratori di sistema e analisti di sicurezza.