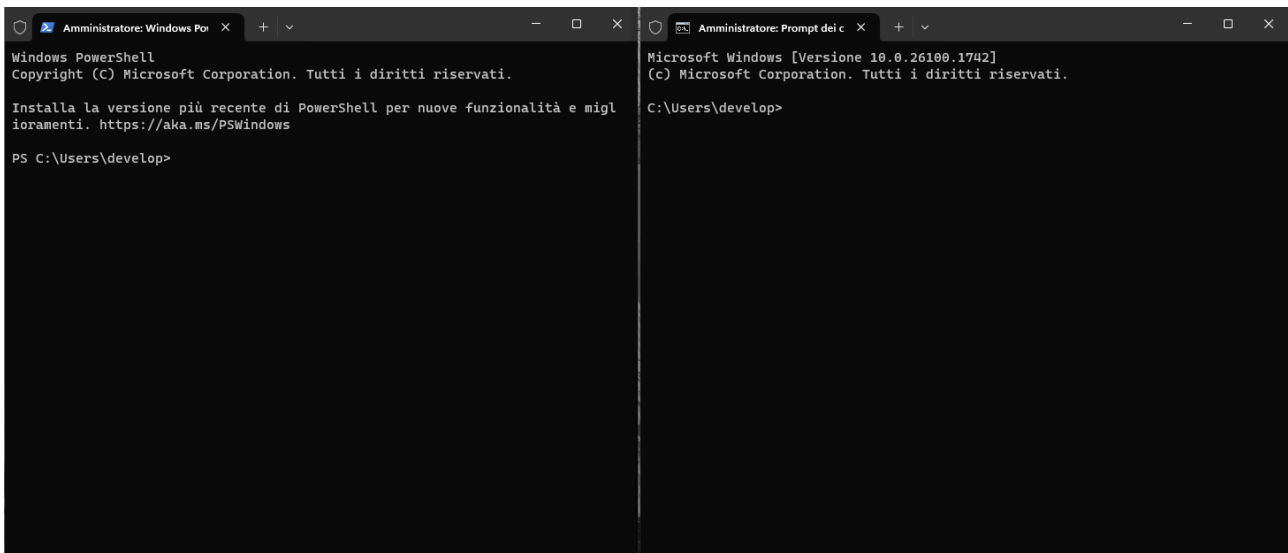


Cyber Security & Ethical Hacking

Report Pratico di Analisi e Automazione

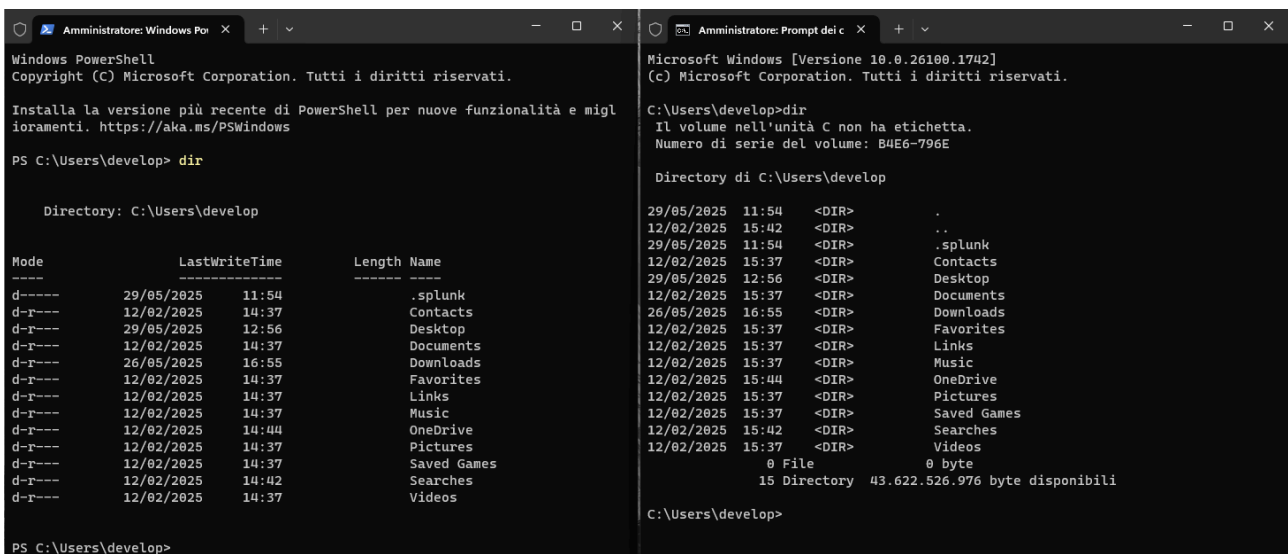
Esercizio 1 – Usare Windows PowerShell

1.1 Accedere alla console PowerShell e al Prompt dei Comandi



2.1 Confrontare l'esecuzione di comandi comuni nei due ambienti

Usiamo il comando **dir** che serve a **visualizzare il contenuto della directory corrente**, ovvero l'elenco di file e cartelle presenti in quella posizione. È uno dei comandi più basilari e diffusi sia nel Prompt dei Comandi (cmd) che in PowerShell.



In PowerShell, l'output del comando dir mostra:

- Mode: indica il tipo di contenuto e i relativi attributi
- LastWriteTime: data e ora dell'ultima modifica
- Length: la dimensione del file (vuoto per le cartelle)
- Name: il nome dell'elemento (file o cartella)

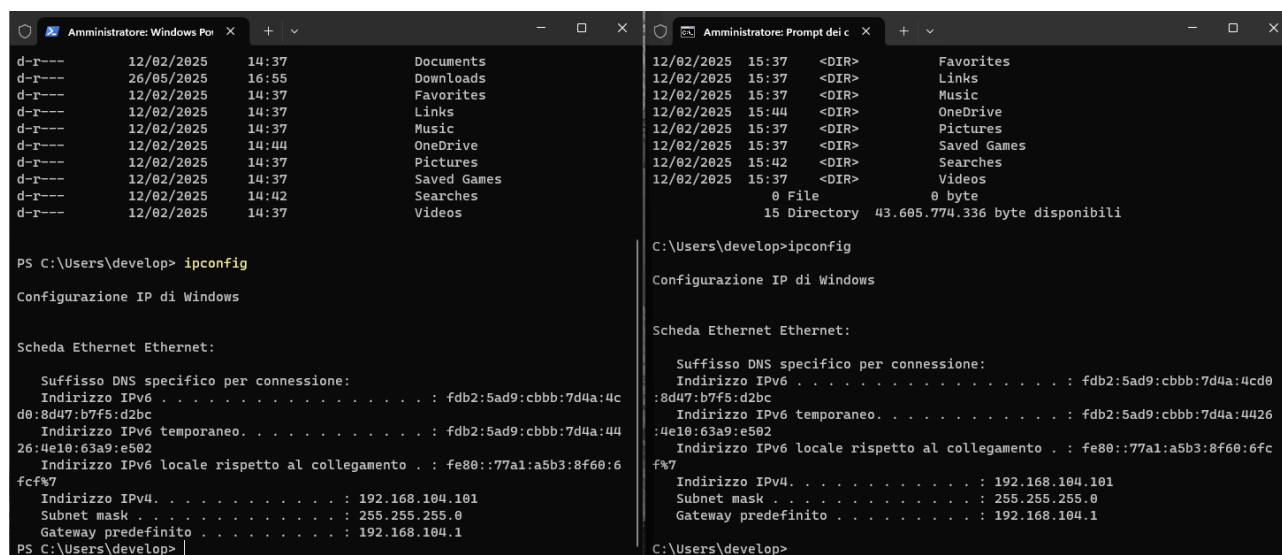
L'elenco è formattato in **colonne ben leggibili**, e ogni riga rappresenta un oggetto interno che può essere utilizzato da altri comandi.

In Prompt dei Comandi, l'output del comando dir mostra:

- Una **riga per ogni elemento**, con:
 - Data e ora ultima modifica
 - Indicazione <DIR> per le cartelle
 - Nome del file o della cartella
- Inoltre mostra:
 - Le directory speciali . (corrente) e .. (superiore)
 - Il numero totale di file e cartelle nella directory
 - Lo spazio libero sul disco

2.1.1 Esecuzione di un altro comando

Comando utilizzato: ipconfig



```
PS C:\Users\develop> dir
d-r---      12/02/2025      14:37      Documents
d-r---      26/05/2025      16:55      Downloads
d-r---      12/02/2025      14:37      Favorites
d-r---      12/02/2025      14:37      Links
d-r---      12/02/2025      14:37      Music
d-r---      12/02/2025      14:44      OneDrive
d-r---      12/02/2025      14:37      Pictures
d-r---      12/02/2025      14:37      Saved Games
d-r---      12/02/2025      14:42      Searches
d-r---      12/02/2025      14:37      Videos

PS C:\Users\develop> ipconfig

Configurazione IP di Windows

Scheda Ethernet Ethernet:

    Suffisso DNS specifico per connessione:
    Indirizzo IPv6 . . . . . : fdb2:5ad9:cbbb:7d4a:4c
d0:8d47:b7f5:d2bc
    Indirizzo IPv6 temporaneo. . . . . : fdb2:5ad9:cbbb:7d4a:44
26:4e10:63a9:e502
    Indirizzo IPv6 locale rispetto al collegamento . : fe80::77a1:a5b3:8f60:6
fcf%7
    Indirizzo IPv4. . . . . : 192.168.104.101
    Subnet mask . . . . . : 255.255.255.0
    Gateway predefinito . . . . . : 192.168.104.1

PS C:\Users\develop>

C:\Users\develop> ipconfig

Configurazione IP di Windows

Scheda Ethernet Ethernet:

    Suffisso DNS specifico per connessione:
    Indirizzo IPv6 . . . . . : fdb2:5ad9:cbbb:7d4a:4cd0
:8d47:b7f5:d2bc
    Indirizzo IPv6 temporaneo. . . . . : fdb2:5ad9:cbbb:7d4a:4426
:4e10:63a9:e502
    Indirizzo IPv6 locale rispetto al collegamento . : fe80::77a1:a5b3:8f60:6fc
f%7
    Indirizzo IPv4. . . . . : 192.168.104.101
    Subnet mask . . . . . : 255.255.255.0
    Gateway predefinito . . . . . : 192.168.104.1

C:\Users\develop>
```

Risultato ottenuto in entrambi gli ambienti:

Il comando ipconfig ha restituito la configurazione della rete attiva, mostrando le informazioni della scheda Ethernet

Osservazioni:

- L'output è identico sia in **PowerShell** che in **Prompt dei Comandi**, poiché ipconfig è un comando di rete di basso livello integrato nel sistema operativo e non gestito da PowerShell.
- È utile per diagnosticare problemi di rete e identificare la connessione attiva, specialmente in ambienti dove il gateway, la subnet o gli indirizzi IP sono fondamentali per l'analisi delle comunicazioni o degli attacchi di rete.

3.1 Comprendere e utilizzare i cmdlet PowerShell

In PowerShell, i comandi sono strutturati come cmdlet, abbreviazione di *command-lets*, con una sintassi standard basata su coppie verbo-sostantivo. Questa struttura consente maggiore chiarezza, automazione e integrazione con l'ambiente operativo.

Il vero comando PowerShell per dir è **Get-ChildItem**.

Questo cmdlet fa parte di un ecosistema molto più avanzato rispetto ai comandi testuali tradizionali, permettendo automazioni e filtri complessi su file, cartelle, servizi, processi, registry e altro ancora.

In PowerShell, il comando dir è semplicemente un **alias**, cioè una scorciatoia, del cmdlet completo: **Get-ChildItem**

Quindi, digitare dir, ls o gci in PowerShell esegue in realtà lo stesso comando: **Get-ChildItem**

Che cos'è Get-ChildItem?

Get-ChildItem è un cmdlet che:

- Elenca file e cartelle in una directory (come fa dir)
- Ma in modo **più potente**, perché ogni voce restituita è un **oggetto .NET** con proprietà accessibili (dimensione, tipo, data, attributi, ecc.)
- Può essere combinato con filtri, ordinamenti, e output personalizzati

Approfondimento sui Cmdlet

Per ottenere informazioni più dettagliate su tutti i cmdlet disponibili in PowerShell, si può eseguire localmente: **Get-Command**

4.1 Analizzare le connessioni e i processi con netstat

Il comando `netstat` (*network statistics*) consente di analizzare lo stato delle connessioni di rete del sistema operativo, mostrando informazioni su porte aperte, indirizzi remoti, protocolli attivi e processi associati.

Visualizzare le opzioni disponibili: **netstat -h**

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

Mostra le statistiche del protocollo e le connessioni di rete TCP/IP correnti.
NETSTAT [-a] [-b] [-e] [-f] [-i] [-n] [-o] [-p proto] [-r] [-s] [-t] [-x] [-y] [interval]

-a      Mostra tutte le connessioni e le porte di ascolto.
-b      Mostra l'eseguibile coinvolto nella creazione di ogni connessione o
        porta di ascolto. In alcuni casi, eseguibili noti ospitano
        più componenti indipendenti e in questi casi la
        sequenza dei componenti coinvolti nella creazione della connessione
        o della porta di ascolto viene visualizzata. In questo caso, il nome dell'eseguibile
        è in [] in basso, in alto si trova il componente chiamato,
        e così via fino al raggiungimento di TCP/IP. Tenere presente che questa opzione
        può essere dispendiosa in termini di tempo e non andrà a buon fine a meno che non si disponga delle
        autorizzazioni sufficienti.
-c      Visualizza un elenco di processi ordinati in base al numero di TCP o UDP
        porte attualmente utilizzate.
-d      Mostra il valore DSCP associato a ogni connessione.
-e      Mostra le statistiche Ethernet. Potrebbe essere in combinazione con l'opzione
        -s.
-f      Mostra Fully Qualified Domain Names (FQDN) per gli indirizzi
        stranieri.
-i      Mostra il tempo in cui una connessione TCP si trova nel suo stato corrente.
-n      Mostra i numeri di indirizzi e porte in formato numerico.
-o      Mostra l'ID processo di proprietà associato a ogni connessione.
-p proto Mostra le connessioni per il protocollo specificato dal protocollo; il protocollo
        può essere: TCP, UDP, TCPv6 o UDPv6. Se usato con l'opzione -s
```

Visualizzare la tabella di routing: **netstat -r**

```
PS C:\Users\develop> netstat -r

=====
Elenco interfacce
 7...42 73 64 8f c5 19 .....Red Hat VirtIO Ethernet 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    192.168.104.1 192.168.104.101 271
  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
  192.168.104.0       255.255.255.0 On-link      192.168.104.101 271
  192.168.104.101     255.255.255.255 On-link      192.168.104.101 271
  192.168.104.255     255.255.255.255 On-link      192.168.104.101 271
  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      192.168.104.101 271
  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      192.168.104.101 271
=====
Route permanenti:
  Indirizzo rete      Mask      Indir. gateway Metrica
  0.0.0.0             0.0.0.0    192.168.104.1 Predefinito
=====

IPv6 Tabella route
=====
Route attive:
  Interf Metrica Rete Destinazione Gateway
  1 331 ::1/128 On-link
  7 271 fdb2:5ad9:cbbb:7d4a::/64 On-link
  7 271 fdb2:5ad9:cbbb:7d4a::4426:4e10:63a9:e502/128 On-link
  7 271 fdb2:5ad9:cbbb:7d4a::4cd0:8d47:b7f5:d2bc/128 On-link
  7 271 fe80::/64 On-link
  7 271 fe80::77a1:a5b3:8f60:6fcf/128 On-link
  1 331 ff00::/8 On-link
  7 271 ff00::/8 On-link
=====
```

Visualizzare tutte le connessioni attive con i relativi processi: **netstat -abno**
Risultato: elenca tutte le connessioni TCP/UDP aperte

```
Amministratore: Windows Po...
1 331 ff00::8 On-link
7 271 ff00::8 On-link

Route permanenti:
Nessuna
PS C:\Users\develop> netstat -abno

Connessioni attive

Proto Indirizzo locale Indirizzo esterno Stato PID
TCP 0.0.0.0:135 0.0.0.0:0 LISTENING 1048
RpcSs
[svchost.exe]
TCP 0.0.0.0:445 0.0.0.0:0 LISTENING 4
Impossibile ottenere informazioni sulla proprietà
TCP 0.0.0.0:5040 0.0.0.0:0 LISTENING 5600
CDPSvc
[svchost.exe]
TCP 0.0.0.0:8000 0.0.0.0:0 LISTENING 4000
[splunkd.exe]
TCP 0.0.0.0:8089 0.0.0.0:0 LISTENING 4000
[splunkd.exe]
TCP 0.0.0.0:8191 0.0.0.0:0 LISTENING 616
[mongod.exe]
TCP 0.0.0.0:49664 0.0.0.0:0 LISTENING 996
Impossibile ottenere informazioni sulla proprietà
TCP 0.0.0.0:49665 0.0.0.0:0 LISTENING 816
Impossibile ottenere informazioni sulla proprietà
TCP 0.0.0.0:49666 0.0.0.0:0 LISTENING 1940
EventLog
[svchost.exe]
TCP 0.0.0.0:49667 0.0.0.0:0 LISTENING 672
Schedule
[svchost.exe]
TCP 0.0.0.0:49668 0.0.0.0:0 LISTENING 3388
[spoolsv.exe]
TCP 0.0.0.0:49671 0.0.0.0:0 LISTENING 960
Impossibile ottenere informazioni sulla proprietà
TCP 127.0.0.1:8065 0.0.0.0:0 LISTENING 8212
[Python3.9.exe]
TCP 127.0.0.1:8089 127.0.0.1:52196 ESTABLISHED 4000
[splunkd.exe]
TCP 127.0.0.1:8191 127.0.0.1:49796 ESTABLISHED 616
[mongod.exe]
TCP 127.0.0.1:8191 127.0.0.1:49860 ESTABLISHED 616
```

Analisi con il Task Manager

1 Aprire **Task Manager** - **2** Scheda **Dettagli** - **3** Ordinare per **PID** - **4** Cercare il **PID** indicato da **netstat** (es. **1048**) - **5** Clic destro > **Proprietà**

```
Amministratore: Windows Po...
1 331 ff00::8 On-link
7 271 ff00::8 On-link

Route permanenti:
Nessuna
PS C:\Users\develop> netstat -abno

Connessioni attive

Proto Indirizzo locale Indirizzo esterno Stato PID
TCP 0.0.0.0:135 0.0.0.0:0 LISTENING 1048
RpcSs
[svchost.exe]
TCP 0.0.0.0:445 0.0.0.0:0 LISTENING 4
Impossibile ottenere informazioni sulla proprietà
TCP 0.0.0.0:5040 0.0.0.0:0 LISTENING 5600
CDPSvc
[svchost.exe]
TCP 0.0.0.0:8000 0.0.0.0:0 LISTENING 4000
[splunkd.exe]
TCP 0.0.0.0:8089 0.0.0.0:0 LISTENING 4000
[splunkd.exe]
TCP 0.0.0.0:8191 0.0.0.0:0 LISTENING 616
[mongod.exe]
TCP 0.0.0.0:49664 0.0.0.0:0 LISTENING 996
Impossibile ottenere informazioni sulla proprietà
TCP 0.0.0.0:49665 0.0.0.0:0 LISTENING 816
Impossibile ottenere informazioni sulla proprietà
TCP 0.0.0.0:49666 0.0.0.0:0 LISTENING 1940
EventLog
[svchost.exe]
TCP 0.0.0.0:49667 0.0.0.0:0 LISTENING 672
Schedule
[svchost.exe]
TCP 0.0.0.0:49668 0.0.0.0:0 LISTENING 3388
[spoolsv.exe]
TCP 0.0.0.0:49671 0.0.0.0:0 LISTENING 960
Impossibile ottenere informazioni sulla proprietà
TCP 127.0.0.1:8065 0.0.0.0:0 LISTENING 8212
[Python3.9.exe]
TCP 127.0.0.1:8089 127.0.0.1:52196 ESTABLISHED 4000
[splunkd.exe]
TCP 127.0.0.1:8191 127.0.0.1:49796 ESTABLISHED 616
[mongod.exe]
TCP 127.0.0.1:8191 127.0.0.1:49860 ESTABLISHED 616
```

Gestione attività

Dettagli

| Nome | PID | Stato | Nome utente | CPU | Memoria | Architettura | Descrizione |
|---------------------------|------|---------------|---------------|-----|-----------|--------------|---------------------------|
| Interrupt sistema | - | In esecuzione | SYSTEM | 00 | 0 K | | Chiamate di procedura... |
| Processo di inattività... | 0 | In esecuzione | SYSTEM | 95 | 8 K | | Percentuale di tempo ... |
| System | 4 | In esecuzione | SYSTEM | 00 | 12 K | | NT Kernel & System |
| Registry | 196 | In esecuzione | SYSTEM | 00 | 2.520 K | | NT Kernel & System |
| smss.exe | 576 | In esecuzione | SYSTEM | 00 | 168 K | | Gestione sessioni di W... |
| mongod.exe | 616 | In esecuzione | SYSTEM | 00 | 114.076 K | x64 | MongoDB Database S... |
| svchost.exe | 672 | In esecuzione | SYSTEM | 00 | 3.956 K | Arm64 | Processo host per serv... |
| csrss.exe | 744 | In esecuzione | SYSTEM | 00 | 792 K | | Processo runtime clie... |
| wininit.exe | 816 | In esecuzione | SYSTEM | 00 | 216 K | | Applicazione di avvio ... |
| csrss.exe | 824 | In esecuzione | SYSTEM | 00 | 816 K | | Processo runtime clie... |
| winlogon.exe | 888 | In esecuzione | SYSTEM | 00 | 844 K | Arm64 | Applicazione Accesso ... |
| svchost.exe | 956 | In esecuzione | SYSTEM | 00 | 1.200 K | Arm64 | Processo host per serv... |
| services.exe | 960 | In esecuzione | SYSTEM | 00 | 2.832 K | | App Servizi e Controller |
| lsass.exe | 996 | In esecuzione | SYSTEM | 00 | 5.908 K | | Local Security Authori... |
| svchost.exe | 1016 | In esecuzione | SYSTEM | 00 | 6.516 K | Arm64 | Processo host per serv... |
| svchost.exe | 1032 | In esecuzione | SYSTEM | 00 | 448 K | Arm64 | Processo host per serv... |
| svchost.exe | 1036 | In esecuzione | SYSTEM | 00 | 596 K | Arm64 | Processo host per serv... |
| svchost.exe | 1048 | In esecuzione | SERVIZIO ... | 00 | 5.404 K | Arm64 | Processo host per serv... |
| fontdrvhost.exe | 1060 | In esecuzione | UMFD-0 | 00 | 264 K | Arm64 | Usermode Font Driver ... |
| fontdrvhost.exe | 1064 | In esecuzione | UMFD-1 | 00 | 724 K | Arm64 | Usermode Font Driver ... |
| ApplicationFrameHo... | 1084 | In esecuzione | develop | 00 | 4.636 K | Arm64 | Application Frame Host |
| svchost.exe | 1172 | In esecuzione | SERVIZIO ... | 00 | 2.352 K | Arm64 | Processo host per serv... |
| svchost.exe | 1216 | In esecuzione | SYSTEM | 00 | 1.364 K | Arm64 | Processo host per serv... |
| svchost.exe | 1304 | In esecuzione | SERVIZIO L... | 00 | 1.092 K | Arm64 | Processo host per serv... |

Proprietà - svchost

svchost

Tipo di file: Applicazione (.exe)

Descrizione: Processo host per servizi di Windows

Percorso: C:\Windows\System32

Dimensioni: 64,2 KB (65.752 byte)

Dimensioni su disco: 36,0 KB (36.864 byte)

Data creazione: lunedì 1 aprile 2024, 10:43:00

Ultima modifica: lunedì 1 aprile 2024, 10:43:00

Ultimo accesso: Oggi 13 giugno 2025, 20 minuti fa

Attributi: ☐ Solo lettura ☐ Nascosto

Dopo aver individuato un **PID** (Process ID) tramite il comando netstat -abno, è possibile ottenere informazioni dettagliate sul processo corrispondente utilizzando il **Task Manager** e la **finestra Proprietà** del processo.

Dalla scheda "Dettagli" del Task Manager:

- **Nome del processo** (es. chrome.exe, svchost.exe, ecc.)
- **PID** (per confermare che corrisponde a quello individuato con netstat)
- **Utilizzo della CPU e memoria** (utile per identificare processi sospetti o pesanti)
- **Stato del processo** (in esecuzione, sospeso...)
- **Utente che ha avviato il processo** (es. SYSTEM, Administrator, nome utente)

Queste informazioni aiutano a determinare:

- Se il processo è legittimo o sospetto
- Se è avviato da un utente normale o da un servizio di sistema
- Se consuma risorse in modo anomalo

Dalla finestra "Proprietà" del processo:

- **Percorso del file eseguibile** (es. C:\Program Files\Google\Chrome\Application\chrome.exe)
- **Descrizione del file** (es. "Google Chrome", "Windows Host Process", ecc.)
- **Versione del file e copyright**
- **Data di creazione/modifica del file**
- **Firma digitale** (in alcuni casi)

Queste informazioni permettono di:

- Verificare se il file si trova nella **posizione corretta** (es. svchost.exe deve essere in C:\Windows\System32\, non altrove)
- Capire se si tratta di un **programma ufficiale o potenzialmente malevolo**
- Valutare l'affidabilità del processo attraverso la **firma digitale**

Conclusione:

L'analisi incrociata tra netstat, **Task Manager** e **Proprietà** permette di:

- Collegare le **connessioni di rete attive** ai relativi **processi in esecuzione**
- Identificare comportamenti **anomali o sospetti**
- Raccogliere informazioni utili per un'eventuale indagine forense o per applicare contromisure

5.1 Svuotare il Cestino usando PowerShell

In questa fase dell'esercizio è stato sperimentato il comando PowerShell per **svuotare completamente il Cestino**, cioè eliminare definitivamente i file al suo interno senza dover utilizzare l'interfaccia grafica.

Comando eseguito: **Clear-RecycleBin**

```
PS C:\Users\develop> 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:\Users\develop>
```

Risultato:

- Tutti i file presenti nel Cestino vengono **eliminati definitivamente**.
- Non è possibile recuperarli con un semplice ripristino da interfaccia utente.
- Il comando può essere molto utile in fase di **pulizia automatizzata del sistema**, ad esempio in script di manutenzione o gestione post-attacco.

Conclusione:

Il comando Clear-RecycleBin consente di **automatizzare l'eliminazione dei file nel cestino** direttamente da terminale, risultando particolarmente utile per:

- Script di manutenzione
- Procedure di hardening del sistema
- Scenari in cui si desidera evitare tracce recuperabili

6.1 Domanda di Riflessione Finale

Altri comandi PowerShell utili nel monitoraggio di un sistema.

PowerShell offre una vasta gamma di comandi utili in ambito di **monitoraggio, diagnosi e sicurezza del sistema**. Alcuni dei cmdlet più rilevanti includono:

Get-Process

- Elenca **tutti i processi attivi** sul sistema, con nome, PID, utilizzo di CPU e memoria.
- Può essere combinato con filtri per individuare processi anomali.

Get-Process | Sort-Object CPU -Descending

Get-Service

- Mostra tutti i **servizi attivi e inattivi**, utile per verificare lo stato di antivirus, firewall o servizi sospetti.

Get-Service | Where-Object {\$_.Status -eq 'Running'}

Get-EventLog

- Permette di leggere i **log eventi di sistema**, applicazioni e sicurezza.
- Fondamentale per l'analisi post-attacco o per il monitoraggio di attività insolite.

Get-EventLog -LogName Security -Newest 50

Start-Transcript

- Registra **tutte le operazioni eseguite in PowerShell** in un file di testo.
- Utile in contesti di audit, formazione o forense.

Start-Transcript -Path C:\Logs\session.log

Get-ACL

- Mostra i **permessi (Access Control List)** associati a file e cartelle.
- Utile per verificare chi può accedere a risorse sensibili.

Get-Acl C:\Users\ | Format-List

Conclusione:

PowerShell è uno strumento estremamente potente per la sicurezza informatica. Permette di monitorare, registrare, filtrare e analizzare dati in tempo reale, riducendo la necessità di strumenti esterni.

Con i giusti cmdlet, può diventare un vero e proprio **sistema di sorveglianza e gestione proattiva del sistema operativo**.

Esercizio 2 – Studio IoC

L'obiettivo è analizzare il comportamento di un file eseguibile sospetto all'interno della sandbox interattiva **Any.Run**, con lo scopo di identificare indicatori di compromissione (IoC), comprendere il ciclo di vita dell'attacco e valutare il rischio associato.

Ambiente di analisi

- **Sandbox utilizzata:** Any.Run
- **File analizzato:** Muadnrd.exe, disponibile su GitHub
- **Sistema bersaglio:** Windows 10 x64 (sandbox virtuale gestita)

The screenshot displays the Any.Run web interface for analyzing the file `Muadnrd.exe`. The top section shows the file's metadata, including its name, size (105 KB), and a download link. Below this, a network traffic log is visible, showing HTTP requests and responses. The bottom section displays a list of processes running in the sandbox, including `svchost.exe`, `firefox.exe`, and `Jvczthe.exe`. The interface also includes a sidebar with navigation options like 'New analysis', 'Reports', and 'TI'.

| Process | PID | Process name | Content |
|-------------|------|---|----------------|
| svchost.exe | 2256 | NetworkService-p-s DnsCache | 0 b + binary |
| firefox.exe | 6852 | https://github.com/MELITERRE/frew/blob/main/... | 154 b + binary |
| firefox.exe | 6596 | https://github.com/MELITERRE/frew/blob/main/... | 34k b + binary |
| firefox.exe | 6744 | -contentproc-channel-1824-parentBuildID 2... | 502 b + binary |
| firefox.exe | 6816 | -contentproc-channel-2208-parentBuildID 2... | 316 b + binary |
| firefox.exe | 7048 | -contentproc-channel-3028-childID 1-sFor... | 479 b + binary |
| firefox.exe | 6680 | -contentproc-channel-4480-childID 2-sFor... | 359 b + binary |
| firefox.exe | 6368 | -contentproc-channel-4976-parentBuildID 2... | 195 b + binary |
| firefox.exe | 6384 | -contentproc-channel-5228-childID 3-sFor... | 483 b + binary |
| firefox.exe | 6340 | -contentproc-channel-5380-childID 4-sFor... | 448 b + binary |
| firefox.exe | 6380 | -contentproc-channel-5512-childID 5-sFor... | 448 b + binary |
| firefox.exe | 6456 | -contentproc-channel-5916-childID 6-sFor... | 468 b + binary |
| Jvczthe.exe | 7492 | PE | 1k b + text |
| cmd.exe | 7520 | /c timeout 21 & exit | 8 b + text |

Fasi dell'analisi

2.1 Esecuzione iniziale e processi sospetti

L'utente apre un repository GitHub e scarica il file Muadnrd.exe. Dopo l'esecuzione, viene generato un nuovo processo anomalo:

Processo: Jvczfzhe.exe

PID: 7492

Origine: Firefox.exe

Questo file **non è firmato digitalmente**, ha un nome randomico e viene eseguito senza interazione visibile, suggerendo l'avvio di un **payload malevolo** in background.

2.2 Comunicazioni di rete

L'analisi del traffico evidenzia diverse richieste HTTP e DNS. In particolare:

| Metodo | Dominio | Stato | Note |
|--------|--------------------------|--------|----------------------------------|
| POST | http://l1.lencor.org/ | 200 OK | Dominio sospetto → C2 remoto |
| GET | detectportal.firefox.com | 200 OK | Legittimo (verifica connessione) |
| POST | ocsp.sectigo.com | 200 OK | Legittimo (check certificati) |

Il dominio l1.lencor.org è sospetto: contattato via POST, in modalità cifrata, suggerisce esfiltrazione dati o comunicazione con un server di comando e controllo (C2).

2.3 Tattiche di inganno

A video appare un **falso messaggio di errore**:

"There was an error opening this document. The file is damaged and could not be repaired."

Questa tecnica è comunemente usata per **nascondere l'esecuzione effettiva del malware**, inducendo l'utente a credere che nulla sia accaduto.

2.4 Comportamenti nel sistema

Anche se nello screenshot non si osservano direttamente file creati o modifiche al registro, il tipo di esecuzione e la presenza di un processo anomalo come Jvczfzhe.exe suggeriscono:

- Copia del payload in percorsi nascosti (%AppData%, %Temp%)

- Possibile creazione di chiavi di registro per **persistenza**
- Esecuzione di script invisibili o comandi Powershell

Indicatori di Compromissione (IoC)

| Tipo | Indicatore |
|----------|--------------------------------------|
| File | Muadnrd.exe, Jvczfzhe.exe |
| PID | 7492 |
| Dominio | http://l1.lencor.org/ |
| Processo | firefox.exe lancia payload |
| Tecnica | Falso errore per nascondere attività |

Conclusione:

L'analisi dinamica tramite Any.Run ha evidenziato con chiarezza un comportamento malevolo. Il file eseguibile Muadnrd.exe, scaricato da GitHub, si comporta come un malware:

- Installa e avvia un payload (Jvczfzhe.exe)
- Comunica con un server remoto sospetto (l1.lencor.org)
- Utilizza tecniche di inganno visivo per confondere l'utente
- Probabilmente modifica file e chiavi di registro per mantenersi attivo

In base al comportamento osservato, il campione analizzato è **compatibile con un infostealer o trojan RAT**, ovvero un malware progettato per **rubare informazioni sensibili o controllare il sistema da remoto**.

Questo esercizio ha permesso di mettere in pratica l'analisi dinamica dei malware e di rafforzare la capacità di rilevare indicatori di compromissione reali.

Conclusione Generale:

Il report ha dimostrato come strumenti nativi come PowerShell possano essere utilizzati non solo per l'automazione amministrativa, ma anche per investigazioni forensi e attività di sicurezza. Inoltre, l'analisi dinamica di file sospetti in ambienti isolati come Any.Run ha evidenziato l'importanza della threat intelligence nella prevenzione degli attacchi informatici. Questi esercizi riflettono un approccio operativo concreto alla sicurezza informatica.