Relazione sull'uso di Windows PowerShell

Obiettivi L'obiettivo del laboratorio è esplorare le funzionalità di Windows PowerShell, un potente strumento di automazione che funge sia da console di comando che da linguaggio di scripting.

Attività Svolte

1. **Accesso alla console di PowerShell**
    a. Apertura della console di PowerShell e del Prompt dei comandi.
2. **Esplorazione dei comandi del Prompt dei comandi e di PowerShell**
    a. Esecuzione del comando `dir` in entrambi gli ambienti, ottenendo un elenco di file e sottodirectory.
    b. Test di comandi comuni come `ping`, `cd` e `ipconfig`, con output simili tra Prompt e PowerShell.
3. **Esplorazione dei cmdlet**
    a. Identificazione dei comandi di PowerShell utilizzando `Get-Alias dir`, che mostra che `dir` è un alias di `Get-ChildItem`.
    b. Ricerca online per approfondire i cmdlet di PowerShell.
4. **Utilizzo del comando netstat con PowerShell**
    a. Esecuzione di `netstat -h` per visualizzare le opzioni disponibili.
    b. Uso di `netstat -r` per visualizzare la tabella di routing IPv4 e IPv6.
    c. Apertura di una seconda istanza di PowerShell con privilegi elevati.
    d. Esecuzione di `netstat -abno` per ottenere informazioni sui processi attivi e sulle connessioni di rete.
    e. Verifica dei processi associati tramite il Task Manager utilizzando il PID.
5. **Svuotamento del Cestino tramite PowerShell**
    a. Verifica della presenza di file nel Cestino e loro eliminazione con il comando `Clear-RecycleBin`.
    b. Conferma della cancellazione permanente dei file.

Conclusioni PowerShell è uno strumento fondamentale per la gestione e l'automazione delle attività su Windows, particolarmente utile in ambito cybersecurity. La sua capacità di eseguire comandi avanzati e di gestire processi e connessioni di rete lo rende essenziale per un analista di sicurezza.

Riflessione Finale PowerShell offre molteplici comandi utili per la sicurezza informatica. Alcuni esempi includono:

- `Get-EventLog` per monitorare i log di sistema.
- `Get-Process` per visualizzare i processi attivi.
- `Test-NetConnection` per verificare la connettività di rete.

- `Set-ExecutionPolicy` per gestire le policy di esecuzione degli script.

L'uso di PowerShell consente di semplificare operazioni ripetitive e migliorare l'efficienza nella gestione della sicurezza dei sistemi.