

1. Utilizzo di Windows PowerShell

PowerShell è un ambiente di scripting avanzato per Windows che consente di automatizzare attività amministrative. Durante questo laboratorio, sono stati esplorati i seguenti punti:

1.1 Accesso alla console PowerShell

PowerShell è stato aperto tramite la ricerca di Windows. La versione di PowerShell è stata controllata con il comando ``$PSVersionTable``. Il comando ``dir`` è stato utilizzato per visualizzare file e cartelle. Sono stati testati i comandi ``ping``, ``ipconfig``, ``cd`` e ``netstat``.

1.3 Esplorazione dei Cmdlets

Il comando ``Get-Command`` è stato eseguito per elencare tutti i cmdlet disponibili. Il comando ``Get-Help Get-Process`` è stato utilizzato per ottenere dettagli su un comando. Un processo è stato avviato con il comando ``Start-Process notepad.exe`` e terminato con il comando ``Stop-Process -Name notepad``.

1.4 Netstat in PowerShell

Il comando ``netstat -ano`` è stato utilizzato per visualizzare le connessioni di rete.

1.5 Svuotare il Cestino con PowerShell

Il comando ``Clear-RecycleBin -Confirm:$false`` è stato utilizzato per svuotare il Cestino.

 **Risultati:** PowerShell si è rivelato uno strumento potente per amministrare Windows in modo automatizzato.

2. Analisi del traffico HTTP e HTTPS con Wireshark

Wireshark è un potente strumento per catturare e analizzare il traffico di rete.

2.1 Acquisizione del traffico HTTP


Wireshark è stato avviato e l'interfaccia di rete è stata selezionata. Il traffico è stato filtrato con il filtro ``http``. Un sito web HTTP è stato navigato per osservare le richieste GET e POST.

2.2 Acquisizione del traffico HTTPS

Il test è stato ripetuto accedendo a un sito HTTPS. È stato notato che i pacchetti risultano criptati. Il TLS Handshake è stato identificato.

2.3 Considerazioni di Sicurezza

HTTP non è sicuro e i dati possono essere intercettati. HTTPS, invece, utilizza la crittografia per proteggere le informazioni.

 **Risultati:** Wireshark ha permesso di identificare le differenze tra traffico HTTP e HTTPS, evidenziando l'importanza della crittografia.

3. Esplorazione di Nmap

Nmap è uno strumento essenziale per la scansione delle reti e la sicurezza informatica.

3.1 Installazione e Verifica

Eseguendo il comando “nmap –version”, ho confermato la corretta installazione. Successivamente, ho scansionato una macchina con il comando “nmap -sV -p 80,443 <IPTARGET>”, *identificando il server web e il certificato TLS. Infine, ho utilizzato il comando “nmap -A -T4 <IPTARGET>” per rilevare il sistema operativo, i servizi attivi e possibili vulnerabilità.*

✓ Risultati: Nmap ha fornito dettagli utili per un’analisi di sicurezza di una rete target.



4. Attacco a un database MySQL (Analisi PCAP)

4.1 Apertura del file PCAP

Ho caricato il file SQL_Lab.pcap in Wireshark e ho applicato il filtro mysql per isolare il traffico SQL.

4.2 Identificazione di un attacco SQL Injection

Ho identificato pacchetti con query sospette, come “SELECT * FROM users WHERE username=‘admin’ – ‘;”, e ho osservato il furto di credenziali dal database.

4.3 Analisi dei Dati Estratti

L’attacco ha rivelato tabelle come “users” e “payments”, dimostrando l’importanza di proteggere i database con prepared statements e firewall applicativi.

✓ Risultati: L’analisi ha evidenziato i rischi di un database non protetto e l’importanza di prevenire SQL Injection.

Questo progetto ha permesso di esplorare strumenti essenziali per la sicurezza informatica:

- PowerShell per la gestione avanzata dei sistemi Windows.
- Wireshark per il monitoraggio e l’analisi del traffico di rete.
- Nmap per la scansione delle porte e dei servizi di rete.
- Analisi dei database SQL per comprendere gli attacchi e le contromisure.