Parte 2: Esplorare i comandi del Prompt dei Comandi e di PowerShell.

a. Output del comando dir

Quando inserisci dir al prompt in entrambe le finestre (Prompt dei comandi e PowerShell), gli output saranno simili, ma con alcune differenze di formattazione.

- **Nel Prompt dei comandi:** L'output di dir elenca i file e le sottocartelle della directory corrente. Mostra la data e l'ora dell'ultima modifica, se è un file o una directory (<DIR>), la dimensione del file (se è un file) e il nome del file o della directory. Alla fine, fornisce un riepilogo del numero di file, delle dimensioni totali e del numero di directory e dello spazio libero disponibile.
- **In PowerShell:** L'output di dir (che è un alias per Get-ChildItem) elenca anche i file e le sottocartelle. La formattazione è più tabellare, mostrando colonne come Mode (che indica il tipo di elemento, ad esempio d per directory, - per file), LastWriteTime, Length (dimensione del file) e Name. Tende a essere più dettagliato e orientato agli oggetti, il che è una caratteristica distintiva di PowerShell.

b. Risultati di altri comandi

Comandi come ping, cd, e ipconfig funzionano in modo simile sia nel Prompt dei comandi che in PowerShell.

- **ping**: Questo comando viene utilizzato per testare la raggiungibilità di un host su una rete IP. L'output mostrerà i pacchetti inviati, ricevuti, persi e il tempo medio di andata e ritorno (RTT). I risultati saranno praticamente identici in entrambi gli ambienti.
- **cd (Change Directory)**: Questo comando viene utilizzato per cambiare la directory corrente. L'effetto sarà lo stesso: ti sposterai in una directory diversa all'interno del file system. La sintassi e il comportamento di base sono gli stessi.
- **ipconfig**: Questo comando mostra tutte le configurazioni di rete TCP/IP correnti e aggiorna le impostazioni DHCP e DNS. L'output includerà informazioni come gli indirizzi IP, la subnet mask, il gateway predefinito e i server DNS. Anche in questo caso, i risultati saranno coerenti tra Prompt dei comandi e PowerShell, poiché PowerShell esegue semplicemente il programma ipconfig.exe sottostante.

Parte 4: Esplorare il comando netstat usando PowerShell.

Qual è il gateway IPv4?

Dall'output di netstat -r fornito, nella sezione **IPv4 Route Table**, sotto **Active Routes**, puoi vedere la riga:

0.0.0.0 0.0.0.0 192.168.1.1 192.168.1.5 25

Il **Gateway** per la destinazione di rete 0.0.0.0 (che rappresenta la rotta predefinita per l'accesso a reti esterne) è **192.168.1.1**. Questo è il tuo gateway IPv4.

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

Selezionando un PID (Process ID) come il **756** (nell'esempio fornito) e aprendo la finestra di dialogo **Proprietà** da **Gestione Attività**, puoi ottenere una vasta gamma di informazioni.

- **Scheda Dettagli:** Questa scheda fornisce informazioni dettagliate sul processo in esecuzione. Per il PID 756, che nell'esempio di netstat -abno potrebbe essere associato a un servizio o a un'applicazione, la scheda Dettagli mostrerebbe:
    - **Nome Immagine:** Il nome del file eseguibile del processo (ad esempio, svchost.exe, chrome.exe, ecc.).
    - **PID:** L'identificativo unico del processo.
    - **Stato:** Lo stato attuale del processo (ad esempio, In esecuzione, Sospeso).
    - **Nome utente:** L'account utente sotto cui il processo è in esecuzione.
    - **CPU:** La percentuale di utilizzo della CPU.
    - **Memoria (Set di lavoro privato):** La quantità di memoria RAM utilizzata dal processo che non può essere condivisa con altri processi.
    - **Memoria (Set di lavoro):** La quantità totale di memoria fisica attualmente utilizzata dal processo.
    - **Descrizione:** Una breve descrizione dell'eseguibile.
    - **Nome della compagnia:** Il nome dell'azienda che ha prodotto il software.
    - **Versione del prodotto:** La versione del software.
    - **Percorso completo dell'immagine:** Il percorso completo del file eseguibile sul disco.
    - **Linea di comando:** La riga di comando completa utilizzata per avviare il processo, inclusi eventuali argomenti.
- **Finestra di dialogo Proprietà (dopo aver cliccato con il tasto destro e selezionato "Proprietà"):** Questa finestra di dialogo fornisce ancora più dettagli sull'eseguibile associato al PID:
    - **Scheda Generale:** Tipo di file, percorso, dimensione, data di creazione, data di modifica, data di ultimo accesso.
    - **Scheda Sicurezza:** Permessi NTFS per il file, mostrando quali utenti o gruppi hanno accesso e quali tipi di accesso sono consentiti (lettura, scrittura, esecuzione, ecc.).

- **Scheda Dettagli:** Informazioni sulla versione del file (come versione del prodotto, versione del file, data di creazione, lingua, copyright). Queste sono spesso le stesse informazioni di base che si trovano nella scheda Dettagli in Gestione Attività, ma organizzate in modo più specifico per le proprietà del file.
- **Scheda Compatibilità:** Impostazioni per l'esecuzione del programma in modalità di compatibilità per versioni precedenti di Windows.
- **Scheda Versioni precedenti:** Consente di ripristinare versioni precedenti del file se la protezione del sistema è abilitata.

In sintesi, da queste informazioni, puoi identificare l'applicazione o il servizio associato a un determinato PID, il percorso del suo eseguibile, l'utente che lo ha avviato, il suo consumo di risorse e persino i dettagli sulla sua versione e sicurezza. Questo è fondamentale per l'analisi dei processi e la risoluzione dei problemi.

Parte 5: Svuotare il cestino usando PowerShell.

Cosa è successo ai file nel Cestino?

Dopo aver eseguito il comando clear-recyclebin e aver confermato con Y (Yes), i file presenti nel Cestino vengono **eliminati permanentemente** dal PC. Non saranno più recuperabili tramite il Cestino stesso. Questo comando esegue l'equivalente dell'operazione "Svuota Cestino" che faresti graficamente.

Domanda di Riflessione: PowerShell per l'analista di sicurezza

PowerShell è uno strumento incredibilmente potente per gli analisti di sicurezza, offrendo capacità di automazione e gestione che vanno ben oltre i comandi di base. Ecco alcuni comandi e scenari che un analista di sicurezza potrebbe trovare utili:

Monitoraggio e Forensics:

- **Get-WinEvent**: Questo cmdlet consente di recuperare eventi dai registri eventi di Windows (ad esempio, Sicurezza, Sistema, Applicazioni). È fondamentale per l'analisi dei log, la rilevazione di attività sospette (tentativi di accesso falliti, modifiche alle impostazioni di sicurezza, ecc.) e le indagini forensi.
    - *Esempio:* Get-WinEvent -LogName Security | Where-Object {$_.Id -eq 4625} (per trovare tentativi di accesso falliti).
- **Get-NetTCPConnection**: Visualizza le connessioni TCP attive, molto simile a netstat -an, ma con un output più strutturato e facile da filtrare in PowerShell. Utile per identificare connessioni non autorizzate o attività di malware.
    - *Esempio:* Get-NetTCPConnection -State Established | Select-Object LocalAddress, RemoteAddress, RemotePort (per visualizzare connessioni attive).

- **Get-Process**: Elenca tutti i processi in esecuzione sul sistema. Combinato con Where-Object e Select-Object, può aiutare a identificare processi sconosciuti o sospetti, o processi che consumano risorse in modo anomalo.
    - *Esempio:* Get-Process | Sort-Object CPU -Descending | Select-Object -First 10 Name, CPU, WorkingSet (per trovare i 10 processi che usano più CPU).
- **Get-AuthenticodeSignature**: Verifica la firma digitale di un file eseguibile o di uno script, un passo importante per determinare se un file è stato manomesso o proviene da una fonte attendibile.
    - *Esempio:* Get-AuthenticodeSignature -FilePath C:\Windows\System32\cmd.exe
- **Get-ItemProperty (per il Registro di Sistema)**: Consente di interrogare e modificare le chiavi e i valori del Registro di Sistema. Essenziale per indagare su persistenza di malware, configurazioni di sicurezza e modifiche non autorizzate al sistema.
    - *Esempio:* Get-ItemProperty -Path "HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Run" (per vedere i programmi che si avviano con Windows).

Gestione della Sicurezza e Risposta agli Incidenti:

- **Set-NetFirewallRule / New-NetFirewallRule**: Cmdlet per gestire il firewall di Windows. Un analista può usarli per bloccare rapidamente il traffico da indirizzi IP malevoli, bloccare porte specifiche o creare regole di firewall per isolare sistemi compromessi.
    - *Esempio:* New-NetFirewallRule -DisplayName "Blocca Indirizzo Malevolo" -Direction Inbound -Action Block -RemoteAddress 192.168.1.100
- **Stop-Process**: Permette di terminare processi in esecuzione, utile per bloccare l'attività di malware o processi non autorizzati.
    - *Esempio:* Stop-Process -Name "malware.exe" -Force
- **Remove-Item**: Elimina file e directory. Può essere usato per rimuovere malware o file sospetti.
    - *Esempio:* Remove-Item -Path "C:\Malware\suspicious.exe"
- **Get-CimInstance / Invoke-CimMethod (per WMI)**: Interagire con WMI (Windows Management Instrumentation) per ottenere informazioni dettagliate sul sistema operativo, hardware e software. Può essere usato per raccogliere dati sull'inventario, vulnerabilità, o per eseguire azioni remote su sistemi.
    - *Esempio:* Get-CimInstance -ClassName Win32_ComputerSystem (per ottenere informazioni sul sistema).

Automazione e Scripting:

- **Script di PowerShell:** La vera forza di PowerShell sta nella sua capacità di combinare questi cmdlet in script complessi per automatizzare compiti ripetitivi. Un analista di sicurezza potrebbe scrivere script per:
    - Raccolta automatica di log da più server.
    - Scansione di file sospetti in specifiche directory.
    - Applicazione di configurazioni di sicurezza standardizzate.
    - Risposta automatizzata a eventi specifici (es. blocco di un IP dopo tentativi di accesso falliti).
- **Moduli di PowerShell:** Esistono numerosi moduli di terze parti e ufficiali (es. PSReadLine, Pester per testing, moduli di sicurezza specifici) che estendono le funzionalità di PowerShell per compiti di sicurezza.

PowerShell, grazie alla sua profonda integrazione con il sistema operativo Windows e la sua natura di linguaggio di scripting orientato agli oggetti, offre un controllo e una flessibilità ineguagliabili per le operazioni di sicurezza, dal monitoraggio alla risposta agli incidenti, fino all'automazione di compiti complessi.

Bonus 1: Esplorazione di Nmap

c. Cos'è Nmap? Per cosa viene usato nmap?

Dal manuale di Nmap (man nmap):

- **Cos'è Nmap?** Nmap, abbreviazione di "Network Mapper", è un **utility open source per l'esplorazione di rete e l'audit di sicurezza**.
- **Per cosa viene usato Nmap?** Nmap viene utilizzato per:
    - **Scoperta di host:** Identificare i computer attivi su una rete.
    - **Identificazione di servizi:** Determinare quali servizi (ad es. server web, SSH, FTP) sono in esecuzione su un host e su quali porte.
    - **Identificazione della versione:** Rilevare la versione del software/servizio in esecuzione.
    - **Rilevamento del sistema operativo (OS detection):** Determinare il sistema operativo dell'host scansionato.
    - **Scansione delle vulnerabilità:** Sebbene Nmap non sia uno scanner di vulnerabilità completo, può essere usato per identificare configurazioni errate o versioni di software note per avere vulnerabilità, anche tramite i suoi script (NSE).

e. Guarda l'Esempio 1. Qual è il comando nmap usato? Cosa fa l'opzione -A? Cosa fa l'opzione -T4?

Assumendo che l'Esempio 1 (senza vederlo direttamente dal man page, ma basandosi sull'uso comune e sul contesto del laboratorio) sia nmap -A -T4 <target>:

- **Comando Nmap usato:** nmap -A -T4 [target] (o un comando simile come nmap -v -A [target])
- **Cosa fa l'opzione -A?** L'opzione -A abilita il **rilevamento del sistema operativo (OS detection)**, il **rilevamento della versione (version detection)**, la **scansione di script predefiniti (default NSE scripts)** e la **tracert** (traceroute). È una scansione aggressiva e completa che fornisce molte informazioni sull'host.
- **Cosa fa l'opzione -T4?** L'opzione -T4 imposta il **timing template (o "aggressiveness")** a "Aggressive". Questo significa che Nmap userà un approccio più rapido per la scansione, aumentando la probabilità di rilevamento ma anche il rischio di sovraccaricare reti o sistemi più lenti. I timing template vanno da T0 (Paranoid) a T5 (Insane), dove T4 è un buon compromesso tra velocità e affidabilità.

b. Quali porte e servizi sono aperti? Per ognuna delle porte aperte, registra il software che fornisce i servizi.

Dall'output di nmap -A -T4 localhost:

- **Porta 21/tcp:**
  - **Stato:** open
  - **Servizio:** ftp
  - **Versione:** vsftpd 2.0.8 or later
  - **Dettagli aggiuntivi:** Permesso l'accesso FTP anonimo (Anonymous FTP login allowed).
- **Altre porte:** L'output mostra "Not shown: 996 closed ports", indicando che, oltre alla porta 21, non ci sono altre porte aperte visibili nell'output fornito.

a. Registra l'indirizzo IP e la subnet mask per la tua VM. A quale rete appartiene la tua VM?

Dall'output di ip address:

- **Indirizzo IP della VM:** 10.0.2.15
- **Subnet Mask (derivata dal prefisso /24):** 255.255.255.0
- **A quale rete appartiene la tua VM?** La VM appartiene alla rete 10.0.2.0/24.

c. Quanti host sono attivi? Dai risultati di Nmap, elenca gli indirizzi IP degli host che si trovano sulla stessa LAN della tua VM. Elenca alcuni dei servizi disponibili sugli host rilevati.

Dall'output di nmap -A -T4 10.0.2.0/24:

- **Quanti host sono attivi?** 4 hosts up (4 host attivi).
- **Indirizzi IP degli host sulla stessa LAN della tua VM:**

    - 10.0.2.15 (la tua VM stessa)
    - 10.0.2.4 (dalla sezione clock-skew)
    - 10.0.2.3 (dalla sezione clock-skew)
    - 10.0.2.2 (dalla sezione clock-skew)
- **Servizi disponibili sugli host rilevati (per 10.0.2.15, che è dettagliato nell'output):**
    - **Porta 21/tcp:** ftp (vsftpd 3.0.3)
        - Accesso FTP anonimo permesso.
        - Dettagli sullo stato del server FTP: "Connected to 10.0.2.15", "Logged in as ftp", "TYPE: ASCII", "vsFTPd 3.0.3 - secure, fast, stable".
    - **Porta 22/tcp:** ssh (OpenSSH 8.2, protocollo 2.0)
    - **Porta 23/tcp:** telnet (Openwall GNU/*/Linux telnetd)
    - **Informazioni di servizio generali:** OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nota: L'output dettagliato degli altri 3 host attivi non è fornito, ma sono elencati sotto la sezione clock-skew, indicando che sono stati rilevati come attivi.

a. Qual è lo scopo di scanme.nmap.org?

Dalla descrizione (o navigando al sito), lo scopo di scanme.nmap.org è un **server di test legittimo e autorizzato fornito dagli sviluppatori di Nmap**. Permette agli utenti di Nmap di praticare le scansioni e testare il software senza infrangere alcuna legge o norma etica, poiché è esplicitamente permesso scansionarlo.

Dall'output di nmap -A -T4 scanme.nmap.org:

- **Quali porte e servizi sono aperti?**
    - **Porta 22/tcp:** ssh (OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.8; protocollo 2.0)
    - **Porta 80/tcp:** http (Apache httpd 2.4.7 ((Ubuntu)))
    - **Porta 9929/tcp:** nping-echo (Nping echo)
    - **Porta 31337/tcp:** tcpwrapped (spesso indica un servizio non standard o un servizio che non risponde come previsto da Nmap, ma la porta è aperta)
- **Quali porte e servizi sono filtrati?**
    - **Porta 25/tcp:** smtp
    - **Porta 135/tcp:** msrpc
    - **Porta 139/tcp:** netbios-ssn
    - **Porta 445/tcp:** microsoft-ds
    - **Porta 593/tcp:** http-rpc-epmap
    - **Porta 4444/tcp:** krb524
    - *(Nota: "filtered" significa che la porta è probabilmente protetta da un firewall, e Nmap non riesce a determinare se è aperta o chiusa.)*
- **Qual è l'indirizzo IP del server?**
    - IPv4: 45.33.32.156
    - IPv6: 2600:3c01::f03c:91ff:fe18:bb2f (non scansionato in questo output)
- **Qual è il sistema operativo?**
    - Linux (specificamente OpenSSH 6.6.1p1 Ubuntu e Apache httpd 2.4.7 ((Ubuntu)) suggeriscono una distribuzione Ubuntu Linux).
    - CPE: cpe:/o:linux:linux_kernel

Domanda di Riflessione

Come può Nmap aiutare con la sicurezza della rete?

Nmap è uno strumento indispensabile per la sicurezza della rete in diversi modi:

1. **Network Discovery (Rilevamento della Rete):** Aiuta gli amministratori a mappare la propria rete, identificando tutti i dispositivi connessi (anche quelli non autorizzati o "fantasma").
2. **Audit di Sicurezza:** Permette di identificare quali porte sono aperte su un host, quali servizi sono in esecuzione e le loro versioni. Questo è cruciale per individuare servizi non necessari (che dovrebbero essere chiusi), versioni obsolete di software (che potrebbero avere vulnerabilità note) o configurazioni errate.
3. **Gestione delle Patch:** Nmap può aiutare a verificare se le patch di sicurezza sono state applicate correttamente, controllando le versioni dei servizi.
4. **Verifica delle Regole Firewall:** Consente di testare le regole del firewall, assicurandosi che solo le porte e i servizi desiderati siano accessibili. Le porte "filtrate" indicano che un firewall sta bloccando l'accesso.
5. **Vulnerability Scanning (con NSE):** Gli script del Nmap Scripting Engine (NSE) possono essere utilizzati per rilevare vulnerabilità specifiche, configurazioni errate, o addirittura per sfruttare alcune debolezze.
6. **Valutazione della Postura di Sicurezza:** Fornisce una "fotografia" dello stato di esposizione di un host o di una rete, aiutando le organizzazioni a comprendere i potenziali punti deboli che potrebbero essere sfruttati.

Come può Nmap essere usato da un attore malevolo come strumento nefasto?

La stessa potenza che rende Nmap utile per la sicurezza lo rende altrettanto efficace per gli attori malevoli:

1. **Ricognizione (Reconnaissance):** È uno dei primi strumenti che un attaccante userà per "vedere" la rete del bersaglio. Possono identificare host attivi, sistemi operativi, servizi e versioni in esecuzione, il che li aiuta a capire quali vulnerabilità potrebbero esistere.
2. **Identificazione dei Punti di Ingresso:** Scansionando le porte aperte, un attaccante può identificare servizi esposti a internet o alla rete interna, che potrebbero essere punti di ingresso per l'attacco (es. server SSH con credenziali deboli, server web vulnerabili).
3. **Scoperta di Vulnerabilità:** Gli script NSE possono essere usati per scansionare attivamente la rete alla ricerca di vulnerabilità specifiche o di configurazioni errate che possono essere sfruttate.
4. **Bypass di Difese:** Comprendendo le regole di un firewall (identificando le porte "filtrate" vs. "chiuse"), un attaccante può tentare di trovare modi per aggirarle.
5. **Footprinting:** Raccogliendo informazioni sui tipi di sistemi operativi e servizi, un attaccante può personalizzare i suoi attacchi (es. usare exploit specifici per una versione di Apache).
6. **Rilevamento di Backdoor:** Un attaccante che ha già compromesso un sistema potrebbe usare Nmap internamente per scoprire altri host o servizi sulla rete locale, o per verificare se la sua backdoor (magari su una porta insolita) è ancora aperta e funzionante.

Bonus 2 - Attacco a un database MySQL

Quali sono i due indirizzi IP coinvolti in questo attacco di SQL injection in base alle informazioni visualizzate?

Gli indirizzi IP coinvolti in questo attacco di SQL injection sono 10.0.2.15 (l'attaccante) e 10.0.2.4 (il destinatario).

Qual è la versione?

La versione che l'attaccante, utilizzando la quary SQL 1=1, ha trovato è 5.7.12-Oubuntu1.1

Quale utente ha l'hash della password di 8d3533d75ae2c3966d7e0d4fcc69216b?

L'utente che ha come hash della password 8d3533d75ae2c3966d7e0d4fcc69216b è 1337

Qual è la password in chiaro?

La password in chiaro dall'hash 8d3533d75ae2c3966d7e0d4fcc69216b in md5 è charley

Domande di riflessione

Qual è il rischio che le piattaforme utilizzino il linguaggio SQL?

Il rischio principale associato all'utilizzo del linguaggio SQL da parte delle piattaforme (siti web, applicazioni, ecc.) risiede nella potenziale vulnerabilità agli **attacchi di SQL Injection (SQLi)**.

- **Natura del rischio:** SQL è il linguaggio standard per comunicare con la maggior parte dei database relazionali (MySQL, PostgreSQL, SQL Server, Oracle, ecc.). Le applicazioni web e software utilizzano SQL per interrogare, inserire, aggiornare ed eliminare dati dal database. Se un'applicazione non gestisce correttamente l'input dell'utente, un attaccante può iniettare codice SQL malevolo in un campo di input (come un nome utente, password, campo di ricerca). Questo codice iniettato viene poi interpretato dal database come parte della query legittima.
- **Gravità e impatto:** La gravità di un attacco di SQL Injection è estremamente alta e dipende direttamente dalle capacità dell'aggressore e dai privilegi dell'account del database utilizzato dall'applicazione. Un attacco SQLi riuscito può portare a:
    - **Accesso non autorizzato ai dati:** L'aggressore può leggere, modificare o eliminare qualsiasi dato nel database, comprese informazioni sensibili come credenziali degli utenti, dati personali, informazioni finanziarie, ecc. (come abbiamo visto nell'esempio precedente con l'estrazione di username e hash delle password).
    - **Bypass dell'autenticazione:** L'aggressore può accedere all'applicazione come utente privilegiato (es. amministratore) senza conoscere la password.

- **Scalata dei privilegi:** In alcuni casi, l'attaccante può ottenere un controllo completo del server del database o persino del server sottostante (se il database ha privilegi elevati o se c'è un'ulteriore vulnerabilità).
- **Defacement del sito web:** Modificando i dati nel database che alimentano il sito web, l'aggressore può alterare il contenuto visualizzato.
- **Denial of Service (DoS):** L'attaccante può corrompere o cancellare il database, rendendo l'applicazione inutilizzabile.
- **Esecuzione di comandi di sistema:** Su alcuni sistemi di database e in determinate configurazioni, gli attaccanti possono eseguire comandi direttamente sul sistema operativo sottostante.

In sintesi, il rischio è che una vulnerabilità di SQL Injection possa trasformarsi in un completo controllo sul database dell'applicazione, compromettendo la riservatezza, l'integrità e la disponibilità dei dati.

Metodi per prevenire gli attacchi di SQL Injection

Per prevenire gli attacchi di SQL Injection, è fondamentale adottare pratiche di programmazione sicura e configurazioni infrastrutturali robuste. Ecco due metodi chiave:

1. **Utilizzo di Prepared Statements (Istruzioni Preparate) con parametri:**
   - **Come funziona:** Questo è considerato il metodo più efficace per prevenire la maggior parte degli attacchi SQL Injection. Invece di costruire dinamicamente le query SQL concatenando stringhe (che è ciò che apre la porta all'iniezione), le istruzioni preparate separano il codice SQL dai dati dell'utente. Si definisce prima la struttura della query SQL con dei "segnaposto" (placeholder) per i valori che verranno inseriti, e solo successivamente si forniscono i valori. Il database compila la query *prima* di ricevere l'input dell'utente, trattando l'input come semplice dato, non come codice eseguibile.
   - **Esempio (concettuale):**
     - **Vulnerabile:** query = "SELECT * FROM users WHERE username = '" + user_input + "';"
     - **Sicuro (Prepared Statement):** query = "SELECT * FROM users WHERE username = ?;" poi si associa user_input al ?.
   - **Vantaggi:** Questo approccio garantisce che qualsiasi carattere speciale nell'input dell'utente venga trattato come parte del dato e non come parte della logica SQL, impedendo all'attaccante di manipolare la query.
2. **Validazione e Sanitizzazione dell'Input dell'Utente (Whitelist Validation):**

- **Come funziona:** Prima che qualsiasi input utente venga utilizzato in una query SQL o in qualsiasi altra parte dell'applicazione, dovrebbe essere rigorosamente validato e, se necessario, sanitizzato. La validazione dovrebbe essere basata su una "whitelist" (lista di valori consentiti), piuttosto che su una "blacklist" (lista di valori proibiti).
    - **Validazione:** Verificare che l'input corrisponda esattamente al formato atteso (es. un campo numerico deve contenere solo numeri, un email deve avere un formato di email valido, ecc.).
    - **Sanitizzazione:** Rimuovere o "escape" (neutralizzare) caratteri speciali che potrebbero essere interpretati come codice SQL (es. apici, doppi apici, punti e virgola). Questo è un approccio difensivo e secondario rispetto ai Prepared Statements, ma utile per una difesa in profondità.
- **Vantaggi:** Riduce la superficie di attacco assicurandosi che solo dati "puliti" e attesi raggiungano il database o altre parti dell'applicazione. Anche se un attacco SQLi dovesse in qualche modo superare i Prepared Statements (scenario raro ma possibile con exploit complessi), una rigorosa validazione e sanitizzazione ridurrebbe le possibilità di successo.

Altri metodi importanti, che rientrano nella lista, includono l'implementazione di un **Web Application Firewall (WAF)**, la **disabilitazione di funzionalità non necessarie del database**, il **monitoraggio delle istruzioni SQL** per rilevare anomalie e l'uso di **stored procedure** che incapsulano le query SQL, forzando l'uso di parametri.