EPICODE

ESERCIZIO W19D1 EXTRA

MINACCE COMUNI

Mungiovì Fabio

TASK

Per ogni scenario proposto identifica:

- OWASP Top 10 (se presente);
- MITRE ATT&CK Enterprise (tecnica principale);
- Mitigazione suggerita da MITRE ATT&CK.

Scenari:

1. Un'azienda ha ricevuto segnalazioni da utenti che hanno subito attacchi XSS.
   Gli utenti hanno inserito dati in un form online che eseguiva script dannosi nel loro browser.
   Questo ha permesso agli attaccanti di rubare i cookie di sessione e impersonare altri utenti.
2. Un attaccante è riuscito a ottenere accesso non autorizzato ai dati aziendali sfruttando una vulnerabilità SQL Injection nell'interfaccia di login di un'applicazione.
   L'attaccante ha manipolato l'input per eseguire comandi SQL non autorizzati, estraendo dati sensibili dal database.
3. Un attaccante è riuscito a eseguire codice arbitrario sul server sfruttando una vulnerabilità di deserializzazione non sicura del client in una funzione che accetta oggetti serializzati dall'utente.
   Manipolando l'oggetto inviato, l'attaccante ha ottenuto l'esecuzione remota di codice sul server.

ESECUZIONE

Scenario 1: Attacco XSS (Cross-Site Scripting)

Un'azienda ha ricevuto segnalazioni da utenti che hanno subito attacchi XSS.
Gli utenti hanno inserito dati in un form online che eseguiva script dannosi nel loro browser.
Questo ha permesso agli attaccanti di rubare i cookie di sessione e impersonare altri utenti.

- **OWASP Top 10 (2021):**
    - **A03:2021 - Injection**. Sebbene l'XSS non sia una "iniezione" nel database come la SQL Injection, è comunque una forma di iniezione di codice, in questo caso script, all'interno di una pagina web visualizzata dal browser dell'utente.
- **MITRE ATT&CK Enterprise (Tecnica principale**):
    - **T1059.004: Command and Scripting Interpreter: JavaScript.** Questa tecnica descrive l'esecuzione di codice JavaScript nel contesto del browser web di un utente. In questo scenario, gli attaccanti hanno iniettato script JavaScript dannosi che sono stati eseguiti sul browser delle vittime per rubare i cookie di sessione.
- **Mitigazione suggerita da MITRE ATT&CK (per T1059.004):**
    - **Input Validation:** Validare e sanificare tutti gli input forniti dagli utenti per impedire l'inserimento di script dannosi.
    - **Output Encoding:** Codificare i dati prima di visualizzarli nell'HTML.
      Questo assicura che il browser interpreti i dati come testo semplice e non come codice eseguibile.
    - **Content Security Policy (CSP):** Implementare una Content Security Policy per limitare le fonti da cui il browser può caricare script, immagini e altri contenuti, rendendo più difficile l'esecuzione di script non autorizzati.
    - **Web Application Firewall (WAF):** Utilizzare un WAF per rilevare e bloccare tentativi di attacco XSS a livello di rete prima che raggiungano l'applicazione.

Scenario 2: Attacco SQL Injection

Un attaccante è riuscito a ottenere accesso non autorizzato ai dati aziendali sfruttando una vulnerabilità SQL Injection nell'interfaccia di login di un'applicazione.
L'attaccante ha manipolato l'input per eseguire comandi SQL non autorizzati, estraendo dati sensibili dal database.

- **OWASP Top 10 (2021):**
    - **A03:2021 - Injection.** La SQL Injection è l'esempio più classico di vulnerabilità di injection, dove comandi SQL malevoli vengono "iniettati" in un input per manipolare il database.
- **MITRE ATT&CK Enterprise (Tecnica principale):**
    - **T1190: Exploit Public-Facing Application.** Questa tecnica si riferisce all'uso di vulnerabilità in applicazioni accessibili pubblicamente (come un'interfaccia di login web) per ottenere l'accesso o eseguire azioni non autorizzate. La SQL Injection è un metodo comune per sfruttare tali applicazioni.
- **Mitigazione suggerita da MITRE ATT&CK (per T1190):**

- **Input Validation:** È la misura più critica. Tutti gli input utente devono essere validati, filtrati e sanitizzati per prevenire l'iniezione di codice SQL.
- **Prepared Statements con Parametri (o ORM):** Utilizzare istruzioni SQL preparate con parametri o Object Relational Mappers (ORM) anziché concatenare stringhe per costruire query SQL.
  Questo separa il codice SQL dai dati di input, impedendo che l'input venga interpretato come parte del comando SQL.
- **Least Privilege:** Assicurarsi che l'applicazione utilizzi credenziali con il minimo privilegio necessario per accedere al database, limitando i danni in caso di compromissione.
- **Web Application Firewall (WAF):** Un WAF può aiutare a rilevare e bloccare i tentativi di SQL Injection.
- **Vulnerability Scanning:** Eseguire scansioni regolari per identificare e correggere le vulnerabilità prima che possano essere sfruttate.

Scenario 3: Deserializzazione Non Sicura

Un attaccante è riuscito a eseguire codice arbitrario sul server sfruttando una vulnerabilità di deserializzazione non sicura del client in una funzione che accetta oggetti serializzati dall'utente. Manipolando l'oggetto inviato, l'attaccante ha ottenuto l'esecuzione remota di codice sul server.

- **OWASP Top 10 (2021):**
  - **A08:2021 - Insecure Deserialization.** Questa categoria si riferisce specificamente alle vulnerabilità che sorgono quando un'applicazione deserializza dati non affidabili senza un'adeguata validazione o integrità. Se un attaccante può manipolare un oggetto serializzato, può innescare l'esecuzione di codice arbitrario.
- **MITRE ATT&CK Enterprise (Tecnica principale):**
  - **T1210: Exploitation of Remote Services.** Questa tecnica descrive come gli attaccanti possano sfruttare vulnerabilità in servizi di rete o funzioni accessibili da remoto per ottenere l'esecuzione di codice. La deserializzazione non sicura in un'applicazione lato server che accetta input dal client rientra perfettamente in questa categoria, portando all'esecuzione remota di codice (RCE).
- **Mitigazione suggerita da MITRE ATT&CK (per T1210):**
  - **Update Software:** Mantenere aggiornati tutti i software, librerie e framework utilizzati, poiché le vulnerabilità di deserializzazione vengono spesso corrette nelle nuove versioni.
  - **Filter Network Content:** Filtrare il traffico di rete per bloccare tentativi noti di sfruttamento di vulnerabilità di deserializzazione.
  - **Application Isolation and Sandboxing:** Isolare le applicazioni o le funzioni che gestiscono la deserializzazione in ambienti "sandbox" per limitare i danni in caso di compromissione.
  - **Privileged Account Management:** Assicurarsi che il servizio o l'applicazione che esegue la deserializzazione operi con il minimo privilegio necessario, per ridurre l'impatto dell'esecuzione di codice.
  - **Input Validation/Sanitization:** Anche se la deserializzazione è intrinsecamente rischiosa con input non affidabile, validare e rifiutare input non validi o sospetti può aiutare a mitigare alcuni attacchi.