lunedì 25 novembre 2024 14:05

Traccia:

Per ogni scenario proposto identifica:

● OWASP Top 10 (se presente);

● MITRE ATT&CK Enterprise (tecnica principale);

● Mitigazione suggerita da MITRE ATT&CK.

Scenari:

1. Esercizio Traccia 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.

L'attacco XSS è classificato all'interno del progetto OWASP Top 10, una lista dei principali rischi di sicurezza per le applicazioni web stilata dall'organizzazione internazionale OWASP (Open Web Application Security Project). Secondo OWASP, il rischio XSS rientra nella categoria delle "Injection" (A03:2021), in cui un sistema permette l'inserimento di codice dannoso, e anche nelle "Identification and Authentication Failures" (A07:2021) poiché può portare al furto di sessioni utente.

Tecniche di Attacco secondo MITRE ATT&CK
MITRE ATT&CK, una struttura per la classificazione delle tecniche di attacco, identifica l'XSS sotto la tecnica **T1059.007 - Command and Scripting Interpreter: JavaScript**. Questa tecnica descrive l'uso del linguaggio JavaScript per eseguire codice dannoso nel browser della vittima, sfruttando vulnerabilità di input non sanitizzato.

Quali Rischi comporta un attacco XSS?
Gli attacchi XSS comportano diversi rischi significativi, tra cui:

- **Furto di dati sensibili**: Come accennato, un attaccante può rubare i

cookie di sessione e impersonare l'utente per accedere ai suoi account.

- **Impersonificazione**: L'attaccante può agire come l'utente, con tutte le sue autorizzazioni, ad esempio, inviando messaggi o compiendo azioni fraudolente.
- **Perdita di fiducia nell'applicazione**: Quando un'applicazione viene compromessa da attacchi XSS, gli utenti possono perdere fiducia nella sicurezza dell'applicazione.

Mitigazioni Suggerite: Come Proteggersi dagli Attacchi XSS
Fortunatamente, esistono diverse tecniche per mitigare e prevenire gli attacchi XSS. MITRE ATT&CK fornisce alcune linee guida chiave:

1. **Convalida e sanitizzazione degli input**: Uno dei metodi principali per prevenire l'XSS è assicurarsi che tutti i dati inseriti dagli utenti siano filtrati e sanitizzati. Questo significa che ogni campo di input deve essere controllato per rilevare e rimuovere eventuali script o caratteri dannosi. Tecniche di convalida rigorose impediscono l'inserimento di codice JavaScript o HTML nei form.
2. **Protezione delle sessioni utente**: I cookie di sessione possono essere configurati per essere accessibili solo tramite protocolli sicuri. L'impostazione dei flag HttpOnly e Secure sui cookie di sessione impedisce agli script JavaScript di accedere ai cookie, proteggendoli da eventuali furti.
3. **Utilizzo di Content Security Policy (CSP)**: La Content Security Policy è un meccanismo che permette di specificare quali risorse possono essere caricate all'interno della pagina web. Con una CSP ben configurata, è possibile ridurre l'esecuzione di codice non autorizzato, limitando le origini da cui il browser può caricare gli script.
4. **Aggiornamento e monitoraggio costante**: È importante tenere aggiornati tutti i componenti dell'applicazione e monitorare regolarmente il comportamento dell'applicazione. Se vengono rilevati errori di scripting o comportamenti anomali, è possibile intervenire tempestivamente per mitigare l'attacco

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.

L'attacco SQL Injection è al primo posto nella lista **OWASP Top 10** (A01:2021 - Broken Access Control). È uno degli esempi più comuni di vulnerabilità da injection, in cui l'applicazione non protegge adeguatamente

l'interazione tra il codice e il database.

Tecniche di Attacco secondo MITRE ATT&CK
MITRE ATT&CK classifica la SQL Injection come parte della tecnica **T1505.003 - Server Software Component: SQL Injection**. Questa tecnica descrive il modo in cui un attaccante manipola i comandi SQL per ottenere accesso non autorizzato ai dati o per eseguire operazioni dannose sul database.

Quali rischi comporta una SQL Injection?
Le SQL Injection possono causare danni significativi, tra cui:

- **Furto di dati sensibili**: Gli attaccanti possono accedere a informazioni riservate come credenziali, dettagli finanziari, o informazioni personali.
- **Compromissione dell'integrità dei dati**: I dati possono essere modificati o eliminati, con conseguenti danni per l'azienda.
- **Accesso persistente**: Una volta compromesso il database, l'attaccante può inserire backdoor per accedere nuovamente al sistema in futuro.
- **Danni reputazionali e legali**: La perdita di dati sensibili può danneggiare la reputazione dell'azienda e portare a conseguenze legali.

Mitigazioni Suggerite: Come Proteggersi dagli Attacchi SQL Injection
Prevenire una SQL Injection richiede un approccio proattivo e una serie di tecniche di sicurezza. Ecco le principali soluzioni suggerite da **MITRE ATT&CK** e **OWASP**:

1. **Uso di query parametrizzate (Prepared Statements)**
    - Le query parametrizzate garantiscono che i dati dell'utente vengano trattati come semplici valori e non come comandi SQL.
2. **Validazione e sanitizzazione degli input**
    - Tutti gli input degli utenti devono essere rigorosamente controllati per assicurarsi che non contengano caratteri o sequenze dannose, come ' o --.
3. **Restrizioni sui privilegi del database**
    - Configurare gli account del database con i privilegi minimi necessari. Ad esempio, l'account utilizzato dall'applicazione non dovrebbe mai avere permessi di amministrazione sul database.
4. **Implementazione di un Web Application Firewall (WAF)**
    - Un WAF può rilevare e bloccare tentativi di SQL Injection analizzando il traffico in tempo reale e identificando modelli di comportamento sospetti.
5. **Monitoraggio e logging**
    - È importante monitorare costantemente le query SQL e registrare le attività sospette per individuare potenziali attacchi.
6. **Aggiornamenti regolari**

- Gli aggiornamenti di sicurezza dei sistemi, del database e delle librerie utilizzate nell'applicazione sono fondamentali per correggere vulnerabilità note.

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.

La vulnerabilità di deserializzazione non sicura è evidenziata nella lista **OWASP Top 10** come **A08:2021 - Software and Data Integrity Failures**. È classificata come una delle principali cause di esecuzione remota di codice (RCE), con impatti critici sulla sicurezza dei sistemi.

Tecniche di Attacco secondo MITRE ATT&CK
MITRE ATT&CK classifica questa vulnerabilità sotto la tecnica **T1574.002 - Hijack Execution Flow: Unsafe Deserialization**. La tecnica descrive come gli attaccanti sfruttano la deserializzazione di oggetti non sicuri per alterare il flusso di esecuzione e iniettare codice malevolo.

Quali rischi comporta una Deserializzazione Non Sicura?
Le conseguenze di una vulnerabilità di deserializzazione non sicura possono essere devastanti:

- **Esecuzione remota di codice (RCE)**: Gli attaccanti possono ottenere il controllo completo del server, eseguendo qualsiasi comando.
- **Accesso a dati sensibili**: L'attaccante potrebbe accedere o rubare dati riservati memorizzati nel sistema.
- **Installazione di malware**: Il server può essere utilizzato come punto di ingresso per installare malware o ransomware.
- **Interruzione dei servizi**: Azioni malevole possono causare crash del sistema o downtime prolungati.
- **Danni reputazionali e legali**: La compromissione di un server può mettere a rischio i dati degli utenti e portare a conseguenze legali per l'azienda.

Mitigazioni Suggerite: Come Proteggersi dalla Deserializzazione Non Sicura
Prevenire questa vulnerabilità richiede attenzione durante la progettazione e la gestione delle applicazioni. Ecco alcune strategie suggerite da **MITRE ATT&CK** e **OWASP**:

1. **Evitare la deserializzazione non necessaria**

    - Se possibile, evitare del tutto di deserializzare oggetti provenienti da fonti non attendibili. Optare per formati di dati sicuri come JSON o XML, trattati con librerie sicure.
2. **Validazione rigorosa degli input**
    - Controllare che gli oggetti serializzati provengano da fonti fidate e verificare la loro integrità con meccanismi come firme digitali o hash crittografici.
3. **Implementazione di whitelist per classi deserializzabili**
    - Limitare le classi che possono essere deserializzate dal sistema. Ad esempio, configurare la libreria di deserializzazione per accettare solo oggetti predefiniti.
4. **Isolamento del processo di deserializzazione**
    - Eseguire la deserializzazione in un ambiente isolato, come un container o un processo con privilegi limitati, per minimizzare i danni in caso di attacco.
5. **Aggiornamento continuo delle librerie**
    - Le librerie di serializzazione/deserializzazione devono essere sempre aggiornate per correggere eventuali vulnerabilità note.
6. **Logging e monitoraggio**
    - Registrare tutte le operazioni di deserializzazione e monitorare il sistema per rilevare comportamenti anomali