M5 - Progetto Finale - Azioni Preventive

Per difendere l'applicazione Web da attacchi di tipo SQL Injection (SQLi) e Cross-Site Scripting (XSS) da parte di un utente malintenzionato, si potrebbero implementare le seguenti azioni preventive:

Web Application Firewall (WAF)

- Implementazione: Un WAF dovrebbe essere posizionato di fronte all'applicazione e-commerce (nella DMZ) per filtrare il traffico HTTP/HTTPS. Agirebbe come uno scudo tra gli utenti esterni (inclusi gli attaccanti) e l'applicazione.
- Funzionalità: Il WAF è in grado di rilevare e bloccare pattern di attacco noti per SQLi e XSS, analizzando le richieste in entrata e le risposte in uscita. Può bloccare richieste malformate o contenenti codice malevolo prima che raggiungano l'applicazione.
- Benefici: Protezione a livello applicativo, senza modifiche al codice dell'applicazione esistente. Può mitigare un'ampia gamma di vulnerabilità. È particolarmente utile quando le modifiche al codice non sono immediatamente possibili.

Input Validation e Output Encoding a livello di applicazione

- Implementazione: Questa è una misura a livello di codice dell'applicazione e-commerce. Ogni input ricevuto dall'utente (es. campi di testo, parametri URL) deve essere rigorosamente validato per assicurarsi che corrisponda al formato atteso e che non contenga caratteri speciali o codice potenzialmente dannoso. Allo stesso modo, tutti i dati provenienti dall'applicazione che vengono visualizzati nel browser dell'utente devono essere correttamente "encoded" per prevenire l'esecuzione di script malevoli (XSS).
- Funzionalità: Per SQLi, la validazione include l'uso di prepared statements/parametrized queries per tutte le interazioni con il database, evitando la concatenazione diretta di stringhe per la costruzione delle query. Per XSS, l'encoding assicura che caratteri come <, >, &, " vengano convertiti nelle loro entità HTML equivalenti.
- Benefici: Elimina la causa radice delle vulnerabilità SQLi e XSS, rendendo l'applicazione intrinsecamente più sicura. Questa è la difesa più robusta e consigliata, poiché agisce direttamente sul punto debole dell'applicazione.

Principio del minimo privilegio per l'account del database

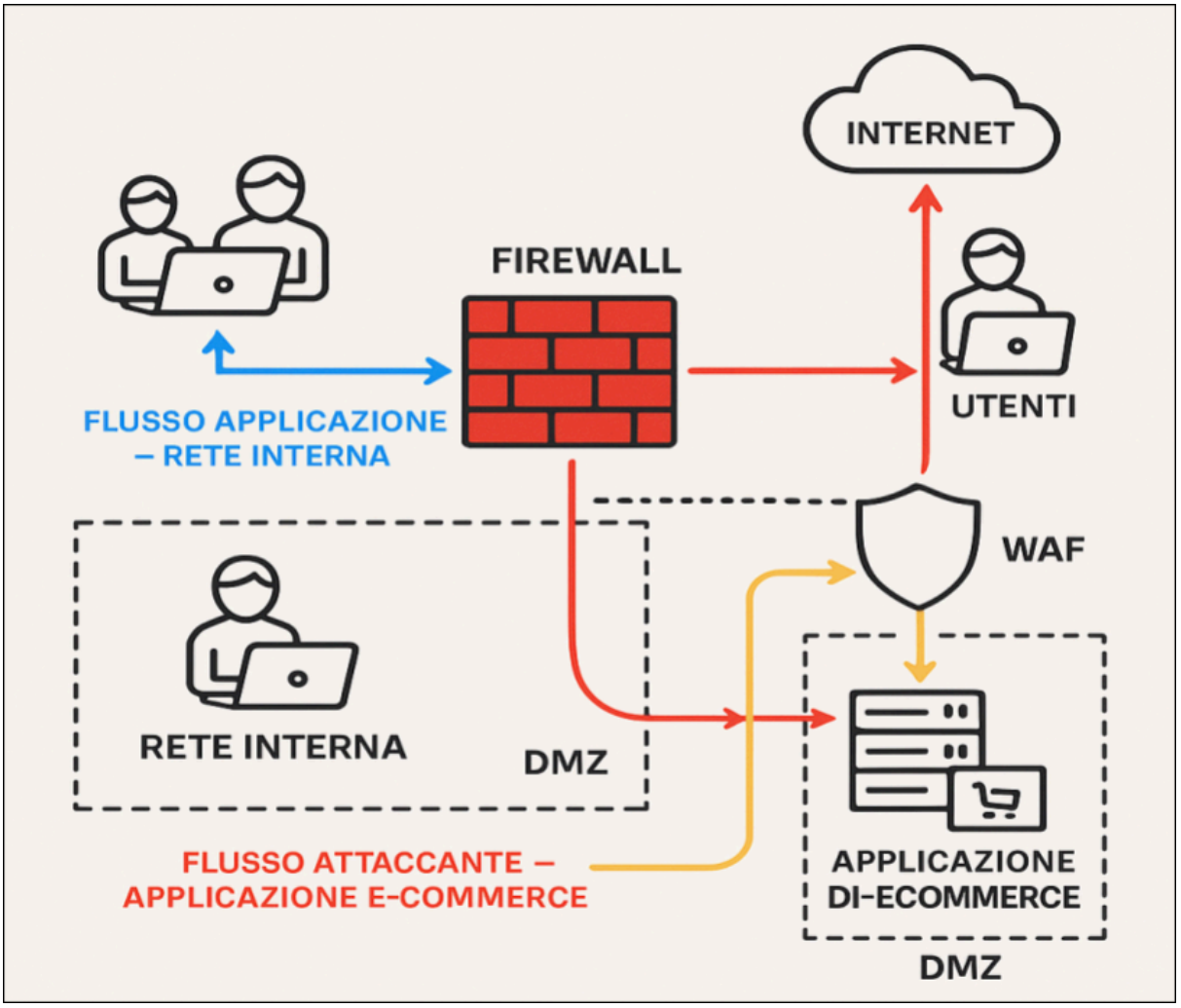
- Implementazione: L'account utente del database utilizzato dall'applicazione e-commerce dovrebbe avere solo i privilegi minimi strettamente necessari per le sue operazioni. Ad esempio, dovrebbe avere permessi di SELECT, INSERT, UPDATE, DELETE solo sulle tabelle pertinenti e non permessi di DROP, ALTER, o GRANT.
- Funzionalità: In caso di un SQLi riuscito, questa misura limita significativamente il danno potenziale, impedendo all'attaccante di eseguire operazioni distruttive o di elevare i privilegi all'interno del database.
- Benefici: Riduce il raggio d'azione di un attacco SQLi in caso di breccia, minimizzando le conseguenze negative.

Sicurezza degli header HTTP (Content Security Policy - CSP)

- Implementazione: Implementare una Content Security Policy (CSP) può mitigare gli attacchi XSS. La CSP è un header HTTP che consente agli sviluppatori di specificare quali risorse (script, stili, immagini, ecc.) il browser dovrebbe essere autorizzato a caricare.
- Funzionalità: Se un attaccante riesce a iniettare uno script XSS, la CSP può impedire al browser di eseguirlo se la sua origine non è tra quelle consentite dalla policy.
- Benefici: Aggiunge un ulteriore strato di difesa contro XSS, limitando le capacità degli script iniettati e fornendo una barriera di sicurezza aggiuntiva lato client.

Rappresentazione Architetturale delle Azioni Preventive

Come evidenziato, è stato introdotto un Web Application Firewall (WAF), posizionato strategicamente tra il firewall di bordo rete e l'applicazione di e-commerce (nella DMZ). Questo componente agisce come uno strato di protezione aggiuntivo, ispezionando e filtrando il traffico HTTP/HTTPS in ingresso per prevenire attacchi di tipo SQL Injection e Cross-Site Scripting prima che raggiungano l'applicazione. I flussi di traffico dall'Internet (sia utenti che attaccanti) passano ora attraverso il WAF per un'analisi approfondita, garantendo che solo le richieste legittime e sicure raggiungano il server dell'applicazione. Si noti che altre misure come la validazione dell'input e l'encoding dell'output sono implementazioni a livello applicativo e non sono direttamente rappresentabili in questo schema architetturale.



Conclusioni

La difesa dell'applicazione web da attacchi come SQL Injection e Cross-Site Scripting richiede un approccio a più livelli che combini protezioni perimetrali con best practice di sviluppo sicuro. L'implementazione di un Web Application Firewall (WAF) fornisce una robusta prima linea di difesa esterna, capace di filtrare il traffico malevolo prima che raggiunga l'applicazione. Tuttavia, la sicurezza più efficace si ottiene affrontando le vulnerabilità alla radice: una rigorosa validazione degli input e un corretto encoding degli output a livello di codice sono essenziali per prevenire l'iniezione di codice dannoso. Il principio del minimo privilegio per gli account del database, unitamente all'adozione di header di sicurezza come la Content Security Policy (CSP), completano questa strategia, riducendo significativamente la superficie di attacco e la gravità di eventuali compromissioni. Un approccio combinato e stratificato è fondamentale per garantire la resilienza dell'applicazione di e-commerce contro le minacce web più comuni e pericolose.