venerdì 22 novembre 2024 18:15

Traccia:

Con riferimento alla figura in slide 2, rispondere ai seguenti quesiti.

1. Azioni preventive: quali azioni preventive si potrebbero implementare per difendere l'applicazione Web da attacchi di tipo SQLi oppure XSS da parte di un utente malintenzionato? Modificate la figura in modo da evidenziare le implementazioni

2. Impatti sul business: l'applicazione Web subisce un attacco di tipo DDoS dall'esterno che rende l'applicazione non raggiungibile per 10 minuti. Calcolare l'impatto sul business dovuto alla non raggiungibilità del servizio, considerando che in media ogni minuto gli utenti spendono 1.500 € sulla piattaforma di e-commerce. Fare eventuali valutazioni di azioni preventive che si possono applicare in questa problematica

3. Response: l'applicazione Web viene infettata da un malware. La vostra priorità è che il malware non si propaghi sulla vostre rete, mentre non siete interessati a rimuovere l'accesso da parte dell'attaccante alla macchina infettata. Modificate la figura in slide 2 con la soluzione proposta.

4. Soluzione completa: unire i disegni dell'azione preventiva e della response (unire soluzione 1 e 3)

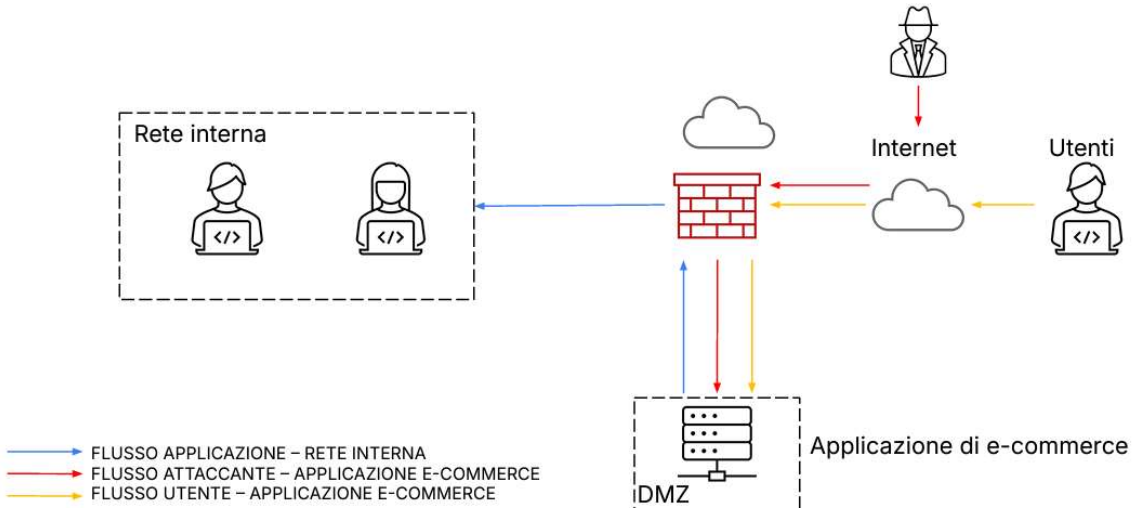
5. Modifica «più aggressiva» dell'infrastruttura (se necessario/facoltativo magari integrando la soluzione al punto 2)

Architettura di rete:
L’applicazione di e-commerce deve essere disponibile per gli utenti tramite Internet per effettuare acquisti sulla piattaforma.

La rete interna è raggiungibile dalla DMZ per via delle policy sul firewall, quindi se il server in DMZ viene compromesso potenzialmente un attaccante potrebbe raggiungere la rete interna.

Architettura di rete:
L’applicazione di e-commerce deve essere disponibile per gli utenti tramite Internet per effettuare acquisti sulla piattaforma.

La rete interna è raggiungibile dalla DMZ per via delle policy sul firewall, quindi se il server in DMZ viene compromesso potenzialmente un attaccante potrebbe raggiungere la rete interna.



1. Azioni preventive: quali azioni preventive si potrebbero implementare per difendere l'applicazione Web da attacchi di tipo SQLi oppure XSS da parte di un utente malintenzionato? Modificate la figura in modo da evidenziare le implementazioni

Introduzione agli attacchi SQLi e XSS

Gli attacchi SQL Injection e Cross-site Scripting sono tra le vulnerabilità più comuni e pericolose che possono colpire un'applicazione Web posizionandosi tra l'altro al terzo posto della classifica OWASP.

SQLi: Un attacco di tipo SQL Injection permette di prendere il controllo di un database ottenendo dati sensibili come credenziali, indirizzi, informazioni personali e dettagli delle carte di credito, in alcuni casi, i dati rubati vengono utilizzati per estorcere denaro all'azienda, inoltre un attaccante potrebbe distruggere o alterare informazioni critiche, paralizzando l'operatività aziendale.

Come proteggersi da SQLi:

- **Query parametrizzate**:
  Utilizzare query precompilate, oltre a essere più semplici da scrivere e più facili da comprendere costringono lo sviluppatore a definire prima tutto il codice SQL per poi passare successivamente a ogni parametro della query in modo tale che il database distinguerà sempre il codice dai dati, indipendentemente dall'input che un utente andrà a inserire impedendone così l'injection.

- **Sanitizzazione degli input**:
  E' fondamentarle avere un controllo su cosa inserisce un utente in quanto gli input che non vengono controllati possono essere sfruttati per rubare dati sensibili, eseguire comandi dannosi e prendere il controllo di un server o applicazione. Per far sì che ciò non accada consentire solo caratteri validi come ad esempio, per un campo numerico accettare solo numeri, bloccare caratteri speciali nei dati forniti dagli utenti (es. <, >, ', "), limitare la lunghezza e il tipo di dato.

- **Controllo degli errori**:
  Disattivare sui siti la visibilità delle pagine degli errori. Spesso tali informazioni si

rivelano preziose per l'attaccante, il quale può risalire all'identità e alla struttura del Database.

- **Protezione tramite WAF**:
  Un Web Application Firewall può rilevare e bloccare tentativi di SQLi prima che raggiungano il server.

XSS: Gli attacchi di tipo Cross-Site Scripting permettono a un attaccante di prendere il controllo di una pagina web iniettando del codice malevolo HTML e JavaScript al fine di raccogliere, manipolare e reindirizzare le informazioni riservate degli utenti.

Tramite un XSS si può:

- rubare cookie di sessione per impersonare un utente legittimo.
- Reindirizzati gli utenti a siti di phishing o di malware.
- modificare il contenuto di una pagina web per ingannare gli utenti.
- Eseguire operazioni sulla Web App con i privilegi di un utente amministrativo.

Un XSS può essere di tipo:

Riflesso: Il payload malevolo viene incluso nella richiesta http e riflesso nella risposta del server senza però essere salvato

Persistente: Il payload viene spedito nel sito vulnerabile e salvato nel server dell'applicazione web per poi essere eseguito ogni volta che un utente visita la pagina vulnerabile

DOM (Document Object Model) based: E' un tipo d'attacco dove il payload non interagisce mai con il serve ma sfrutta la vulnerabilità dal lato del client, perciò è molto più difficile da rilevare per i Web Application Firewall (WAF)

Come proteggersi da XSS:

Per prevenire questo tipo di attacchi oltre a una buona configurazione di un Web Application Firewall e alla sanitizzazione e controllo dell'input/output visto precedentemente con l'SQLi possiamo:

- Impostare politiche di sicurezza (**CSP**) che limitano le origini da cui può essere eseguito lo script.
- Utilizzare framework sicuri che incorporano protezioni contro XSS attraverso il binding sicuro dei dati.

- Impostare i cookie come HTTP-only per impedire l'accesso a JavaScript.
- Utilizzare librerie specifiche per la sanificazione degli input come DOMPurify.

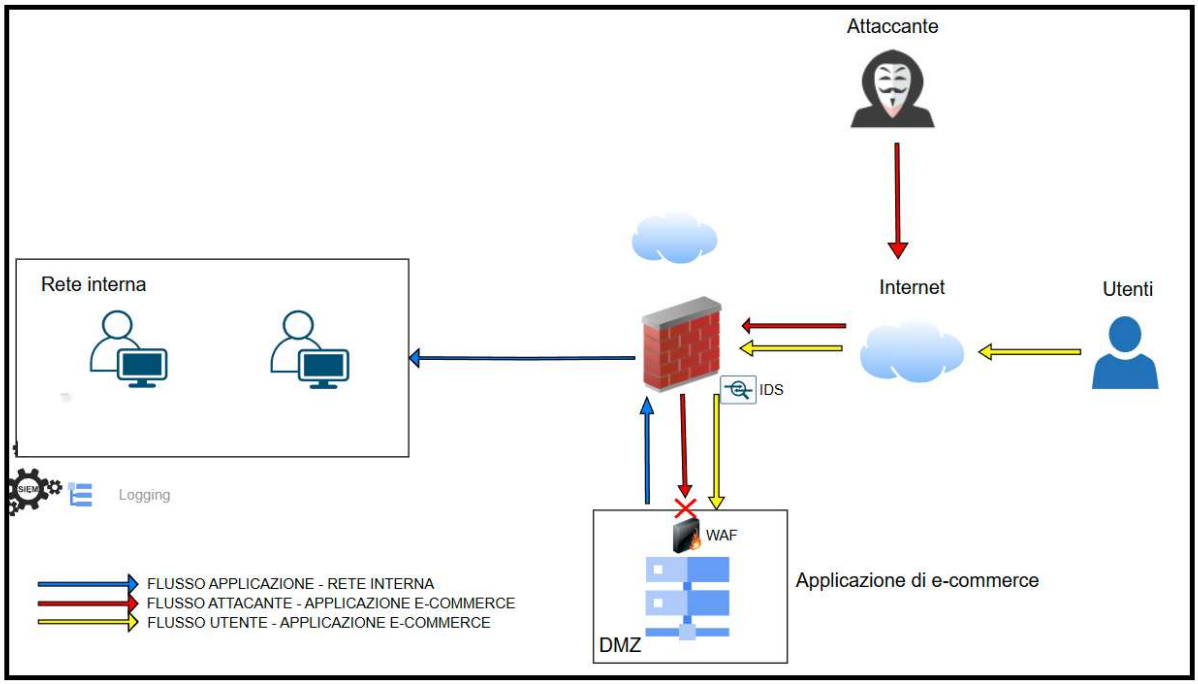
In definitiva per prevenire attacchi di tipo SQL Injection (SQLi) e Cross-Site Scripting (XSS) è fondamentale adottare una serie di misure tecniche e organizzative che lavorino sinergicamente per proteggere l'applicazione Web.

Dall'immagine sottostate si può vedere come implementare un **WAF** direttamente all'interno della DMZ sia cruciale in quanto può filtrare e monitorare il traffico in entrata e in uscita, bloccando eventuali attacchi dall'esterno.

E' importante anche integrare un sistema di prevenzione e rilevamento delle intrusioni (**IDS**) che agisce in modo complementare al WAF registrando e analizzando il traffico per identificare anomalie che potrebbero non essere rilevate immediatamente. Questo è particolarmente utile per tracciare attacchi più sofisticati che potrebbero bypassare le misure di protezione front-end.

Infine, è essenziale l'integrazione di sistemi di **SIEM** per un monitoraggio costante e una gestione centralizzata dei **log**. Il SIEM consente di identificare e rispondere rapidamente ai tentativi di attacco.

Queste implementazioni, insieme a una formazione continua del Team di sviluppo, scansioni regolari dei sistemi e eventuali patch e aggiornamenti in caso di nuove vulnerabilità, garantiscono una protezione completa della Web application.



2. Impatti sul business: l'applicazione Web subisce un attacco di tipo DDoS dall'esterno che rende l'applicazione non raggiungibile per 10 minuti. Calcolare l'impatto sul business dovuto alla non raggiungibilità del servizio, considerando che in media ogni minuto gli utenti spendono 1.500 € sulla piattaforma di e-commerce. Fare eventuali valutazioni di azioni preventive che si possono applicare in questa problematica

Introduzione agli attacchi DDOS

Un attacco DDoS (Distributed Denial of Service) prende di mira siti Web e server interrompendo i servizi di rete nel tentativo di esaurire le risorse di un'applicazione. Durante questo attacco, una serie di bot, o botnet, invia una quantità elevata di richieste e traffico HTTP a un sito Web o a un servizio. In sostanza, più computer prendono d'assalto un computer durante un attacco, estromettendo gli utenti legittimi. Di conseguenza, il servizio può subire rallentamenti o comunque interruzioni per un certo periodo di tempo.

Quando un'applicazione Web subisce un attacco di tipo DDoS, l'impatto sul business può essere significativo, soprattutto se il servizio diventa irraggiungibile per un periodo prolungato. Immaginiamo che l'attacco renda l'applicazione indisponibile per dieci minuti e che in condizioni normali gli utenti spendano 1.500 euro al minuto, la perdita diretta si calcolerebbe in 15.000 euro. Oltre alla perdita economica immediata, bisogna considerare anche i danni collaterali come il deterioramento dell'esperienza utente, la perdita di fiducia nei confronti della piattaforma e potenziali costi aggiuntivi per il ripristino dei servizi.

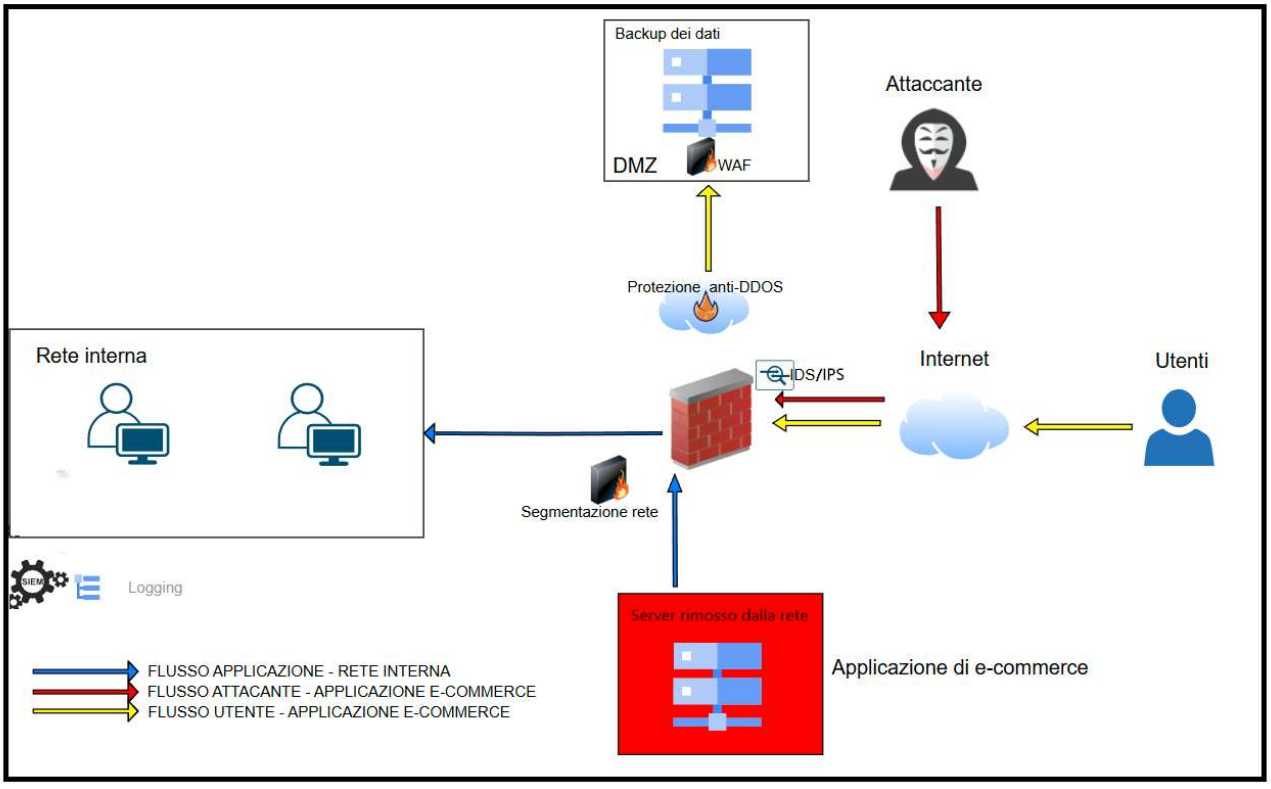
Come proteggersi da DDOS:

Per mitigare il rischio e l'impatto di questi attacchi, bisogna implementare un sistema di protezione contro i DDoS che utilizzi tecniche di filtraggio per distinguere il traffico legittimo da quello malevolo, bloccando le richieste sospette prima che raggiungano i server. Si possono utilizzare strumenti come Cloudflare per gestire grandi volumi di traffico anomalo. In caso di attacco prolungato, si può implementare un piano di disaster recovery, ad esempio prevedendo soluzioni come un failover automatico su un infrastruttura secondaria, portando così il carico sulla piattaforma alternativa e riducendo i tempi di inattività. Può essere una soluzione disporre anche di backup dei sistemi aggiornati, non solo per la base dati, ma anche per l'infrastruttura di rete e le regole di sicurezza come quelle del WAF. Infine, è importante integrare il disaster recovery con un **Business Continuity Plan (BCP)**. Mentre il disaster recovery si concentra sul ripristino

dell'infrastruttura, il BCP copre aspetti come il coordinamento dei team operativi, la comunicazione con i clienti durante l'interruzione del servizio e la gestione delle relazioni con i partner e i fornitori.

5. Modifica «più aggressiva» dell'infrastruttura (se necessario/facoltativo magari integrando la soluzione al punto 2)

In questo scenario, l'obiettivo non è solo prevenire attacchi specifici come SQLi, XSS o malware, ma anche rendere l'intera architettura più robusta contro eventuali interruzioni, propagazioni di minacce e attacchi su vasta scala. In questo caso oltre a una segmentazione della rete, possiamo **rimuovere il sistema infetto** sia dalla rete interna che da internet in modo che l'attaccante non lo riesca più a raggiungere e ripristinare i **dati nel server di backup** della DMZ creato precedentemente con una **protezione** avanzata a livello di rete come **Cloudflare,** in modo da poter mantenere attiva la Web application senza danni al buisness.



3. Response: l’applicazione Web viene infettata da un malware. La vostra priorità è che il malware non si propaghi sulla vostre rete, mentre non siete interessati a rimuovere l’accesso da parte dell’attaccante alla macchina infettata. Modificate la figura in slide 2 con la soluzione proposta.

Quando un'applicazione Web viene infettata da un malware, la priorità assoluta deve essere quella di contenere la minaccia e impedire che questa si propaghi ad altre parti della rete, in particolare alla rete interna. Dato che in questo scenario, non è fondamentale rimuovere immediatamente l’accesso dell’attaccante alla macchina compromessa, la soluzione migliore è quella di **isolare il sistema infetto** per minimizzare i danni e preservare l’integrità dell’infrastruttura.
La prima azione da intraprendere è **segmentare la rete**. La rete interna e la DMZ devono essere separate attraverso un firewall con regole estremamente restrittive. Questo consente di limitare i collegamenti tra le due zone e impedire che eventuali movimenti laterali del malware raggiungano i sistemi critici interni. È fondamentale bloccare ogni traffico non essenziale dalla DMZ verso la rete interna, consentendo solo le comunicazioni strettamente necessarie.
Un altro intervento fondamentale è implementare un sistema di monitoraggio in grado di rilevare attività anomale all’interno della rete. L’uso di un Intrusion Detection System (**IDS**) o, ancora meglio, di un Intrusion Prevention System (**IPS**) consente di identificare rapidamente i comportamenti sospetti del malware, come tentativi di comunicazione con server esterni o di replicazione verso altri nodi della rete. In alcuni casi, l’IPS può anche bloccare automaticamente queste attività.
Durante la fase di isolamento della macchina infetta, è importante raccogliere dati per analizzare il comportamento del malware e identificare le vulnerabilità che sono state sfruttate, così da evitare ulteriori compromissioni.
Infine, occorre implementare una strategia di **logging** centralizzato che raccolga tutte le informazioni sulle attività della macchina compromessa. Questo non solo aiuta nell'analisi post-incidente, ma può anche fornire prove utili in caso di indagini più approfondite.
Queste misure combinate consentono di limitare l’impatto dell’infezione senza la necessità immediata di eliminare l’accesso dell’attaccante, concentrandosi sul contenimento del malware e sulla protezione del resto dell’infrastruttura.

4. Soluzione completa: unire i disegni dell'azione preventiva e della response (unire soluzione 1 e 3)