Epicode – Modulo 5

Alessandro Galli

Prevenzione SQL Injection

•**Utilizzare Prepared Statements (o Query Parametrizzate)**:

Questo metodo permette di separare il codice SQL dai dati forniti dall'utente, prevenendo l'inserimento di codice malevolo.

•**Utilizzare ORM (Object-Relational Mapping)**:

Gli ORM come Hibernate per Java o Entity Framework per .NET aiutano a evitare SQL manuale, rendendo più difficile l'inserimento di codice malevolo.

•**Validazione e Sanitizzazione dei Dati di Input**:

Validare rigorosamente tutti i dati di input per assicurarsi che siano del tipo e nel formato atteso.

Sanitizzare i dati per rimuovere o «escapare» i caratteri pericolosi.

•**Limitare i Permessi del Database**:

Configurare l'applicazione in modo che utilizzi un utente del database con i privilegi minimi necessari. Evitare di usare l'utente root o amministratore del database.

•**Utilizzare Firewalls per le Applicazioni Web (WAF)**:

Implementare un WAF può aiutare a identificare e bloccare i tentativi di attacco SQLi in tempo reale.

Prevenzione XSS (Cross-Site Scripting)

•**Sanitizzazione e Escaping dei Dati di Input e Output**:

Escapare tutti i dati forniti dall'utente prima di renderli nelle pagine web. Questo può essere fatto utilizzando funzioni specifiche per il contesto (HTML, JavaScript, URL, ecc.).

•**Utilizzare Content Security Policy (CSP)**:

Una CSP può ridurre il rischio di XSS specificando quali sorgenti di contenuti sono attendibili.

•**Convalidare e Sanitizzare i Dati di Input**:

Come per SQLi, è essenziale validare tutti i dati di input per assicurarsi che siano conformi alle aspettative.

Sanitizzare i dati rimuovendo eventuali script o markup pericolosi.

•**Utilizzare HttpOnly e Secure Flags per i Cookie**:

Impostare il flag HttpOnly sui cookie per evitare che siano accessibili tramite JavaScript.

Impostare il flag Secure per garantire che i cookie siano trasmessi solo su connessioni HTTPS.

•**Utilizzare Frameworks con Protezioni Integrate**:

Molti framework moderni come Angular, React, e Django offrono protezioni integrate contro XSS e dovrebbero essere utilizzati correttamente.

Analisi impatto sul business – Prevenzione DDoS

Un'attacco DDoS colpisce un'applicazione web rendendo indisponibile il servizio per 10 minuti. Sapendo che la web app appartiene a un e-commerce che incassa 1500 € al minuto, per calcolare l'impatto sul business è sufficiente moltiplicare i 1500 € al minuto per 10, ottenendo così un potenziale impatto economico all' e-commerce di **15000 €.**

Prevenire attacchi DDoS (Distributed Denial of Service) è essenziale per mantenere l'integrità, la disponibilità e la sicurezza dei servizi online. Ecco alcune azioni che possono essere adottate:

- Utilizzo di sistemi di rilevamento delle intrusioni (IDS) e di prevenzione delle intrusioni (IPS): Implementare IDS e IPS per monitorare il traffico di rete e identificare comportamenti anomali che potrebbero indicare un attacco DDoS.
- Soluzioni di mitigazione DDoS basate su cloud: Utilizzare servizi di mitigazione DDoS forniti da terze parti, come Cloudflare, Akamai, AWS Shield, che possono assorbire e filtrare il traffico malevolo.
- Firewall e filtri anti-DDoS: Configurare firewall e router per bloccare il traffico proveniente da indirizzi IP noti per essere utilizzati in attacchi DDoS.
- Limitazione della banda: Impostare limiti di banda per prevenire che un singolo utente possa consumare tutte le risorse di rete.
- Rate Limiting: Implementare tecniche di rate limiting per limitare il numero di richieste che un singolo indirizzo IP può effettuare in un dato periodo di tempo.
- Load Balancing: Distribuire il traffico su più server o data center per evitare che un singolo punto diventi un collo di bottiglia.

Prevenzione DDoS

- Reti di distribuzione dei contenuti (CDN): Utilizzare CDN per distribuire il contenuto su più server geograficamente distanti, riducendo la possibilità che un attacco DDoS possa colpire un singolo punto di accesso.
- Protezione delle applicazioni web: Utilizzare firewall per applicazioni web (WAF) per proteggere le applicazioni web da traffico malevolo e attacchi mirati.
- Aggiornamento regolare delle patch di sicurezza: Assicurarsi che tutti i sistemi e le applicazioni siano aggiornati con le ultime patch di sicurezza per ridurre le vulnerabilità che potrebbero essere sfruttate durante un attacco DDoS.
- Monitoraggio continuo e analisi dei log: Monitorare continuamente i log di rete e analizzare i pattern di traffico per rilevare precocemente attività sospette.
- Segmentazione della rete:Segmentare la rete in diverse zone di sicurezza per limitare la propagazione di un attacco DDoS all'interno dell'organizzazione.
- Formazione del personale:Formare il personale IT e gli utenti sulle pratiche di sicurezza e sulla consapevolezza degli attacchi DDoS.

DRaaS

Il Disaster Recovery as a Service **(DRaaS)** è un servizio che permette di replicare e ospitare server fisici o virtuali da un ambiente primario a uno secondario (di solito basato su cloud) per garantire la continuità operativa in caso di disastro. Questo servizio include il ripristino rapido dei sistemi IT e dei dati aziendali in caso di eventi catastrofici come guasti hardware, attacchi informatici, disastri naturali, o altri incidenti che potrebbero compromettere l'infrastruttura IT.

Componenti Principali

- Replica dei Dati: I dati aziendali vengono continuamente replicati su un'infrastruttura remota (di solito nel cloud). Questo assicura che ci sia sempre una copia aggiornata dei dati prontamente disponibile per il ripristino.Automazione del
- Ripristino: DRaaS automatizza il processo di failover, ovvero il passaggio all'infrastruttura di backup, riducendo il tempo di inattività e garantendo un ripristino rapido delle operazioni.
- Monitoraggio e Gestione: Le soluzioni DRaaS includono strumenti di monitoraggio e gestione che permettono di supervisionare lo stato della replica dei dati e delle risorse di backup.
- Testing e Conformità: È possibile eseguire test periodici del piano di disaster recovery senza interrompere le operazioni aziendali, garantendo che il processo funzioni correttamente quando necessario.

DRaaS

Vantaggi

- Riduzione dei Costi: Rispetto a mantenere una propria infrastruttura di disaster recovery, DRaaS è generalmente più conveniente poiché utilizza il cloud e un modello di pagamento basato sull'uso.

- Scalabilità: Le soluzioni DRaaS sono altamente scalabili, permettendo alle aziende di adattare il servizio in base alle loro necessità, senza dover investire in risorse hardware aggiuntive.

- Ripristino Rapido: Grazie alla replica continua e agli strumenti di automazione, il ripristino dei dati e dei sistemi può avvenire molto rapidamente, riducendo al minimo il downtime.

- Sicurezza: I provider di DRaaS offrono elevate misure di sicurezza per proteggere i dati replicati e garantire la loro integrità durante il trasferimento e l'archiviazione.

DRaaS

Come Funziona

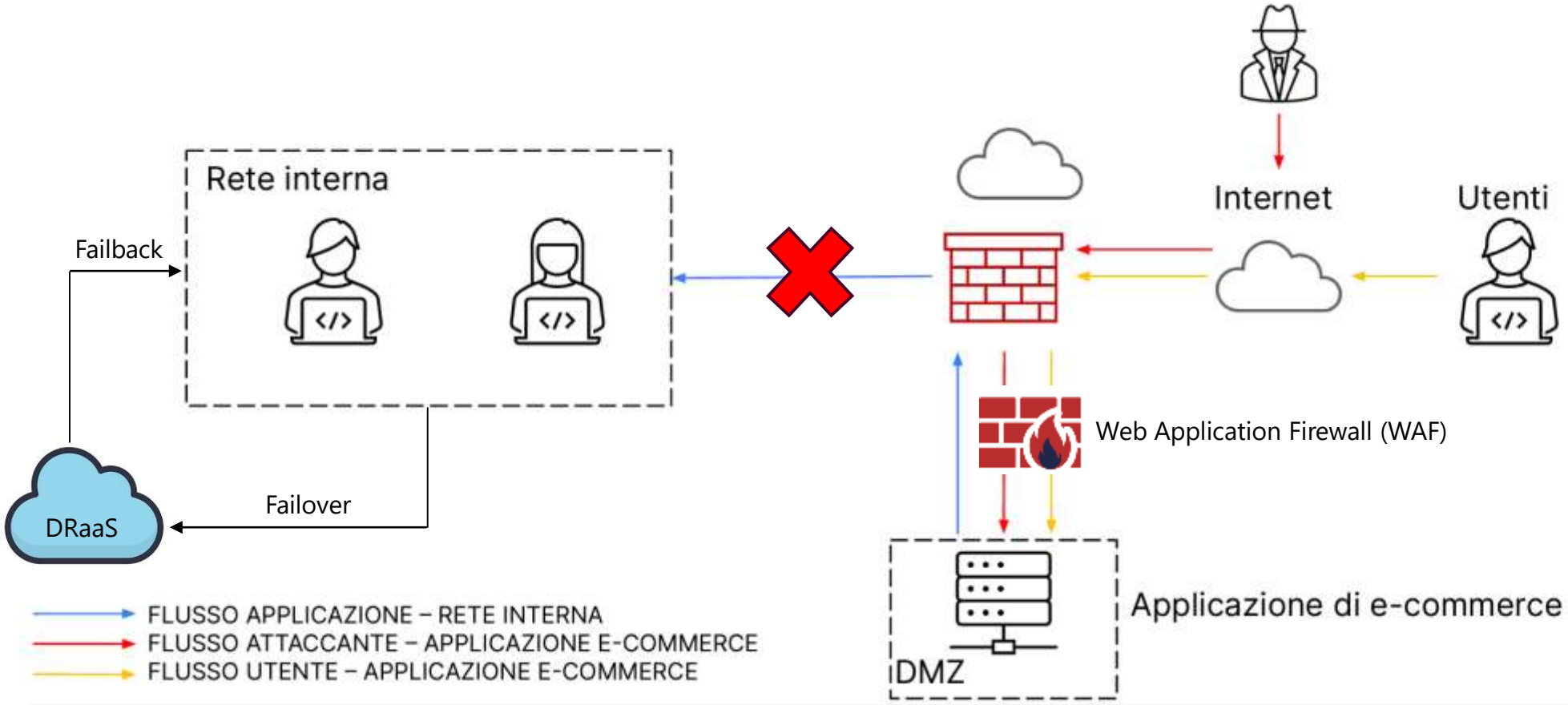
- Setup Iniziale: Si configura il sistema per replicare i dati e le applicazioni su un'infrastruttura remota (cloud).

- Replica Continua: I dati vengono continuamente copiati dal sito primario a quello secondario, assicurando che le copie siano sempre aggiornate.

- Trigger di Failover: In caso di disastro, viene attivato il failover, ovvero il passaggio alle risorse di backup che diventano operative per mantenere la continuità dei servizi.

- Ripristino (Failback): Una volta risolto il problema al sito primario, il processo di failback permette di riportare i dati e le applicazioni dal sito di backup a quello primario, ristabilendo la normale operatività.

Modifiche al sistema



Per prevenire l'accesso alla web app di e-commerce, sarebbe utile settare un **Web Application Firewall** interponendolo fra internet e la DMZ come da figura. Se un attaccante riesce comunque a ottenere l'accesso all'applicazione di e-commerce, potrebbe essere utile togliere la comunicazione con la rete interna per prevenire altri danni e affidarsi al DRaaS descritto in precedenza, effettuando un failover dei dati della rete interna.