W20D4 – PRATICA

1. Azioni preventive per attacchi SQLi e XSS:

Per difendere l'app di e-commerce da attacchi SQL Injection (SQLi) e Cross-Site Scripting (XSS), è fondamentale implementare diverse misure preventive. Le principali sono:

Sanitizzazione e validazione dell'input: Validare tutti i dati inseriti dagli utenti e assicurarsi che siano del formato previsto prima di utilizzarli nelle query SQL o di visualizzarli nel frontend.*

Prepared Statements: Utilizzare query SQL con parametri "bindati" (ex. .... WHERE id_user = :id_user) per evitare che input "maligni" influenzino la struttura della query SQL.*

Escape Output: Implementare una corretta gestione dell'output, specialmente per il contenuto visualizzato sulle pagine web, per prevenire l'inserimento di script dannosi.

WAF (Web Application Firewall): Utilizzare un firewall per le applicazioni web per rilevare e bloccare i tentativi di attacchi come SQLi e XSS.

Content Security Policy (CSP): Imporre una Content Security Policy "strict" per evitare l'esecuzione di script non autorizzati.

Aggiornamenti e patch: Assicurarsi che sia lato server che lato client siano sempre applicati gli ultimi aggiornamenti con le ultime patch di sicurezza.

2. Impatti sul business con un down di 10 min

Impatto economico = 10 minuti * 1.500 €/minuto = **15.000 €**

Azioni preventive per attacchi DDoS:

CDN con protezione DDoS: Utilizzare una CDN (Content Delivery Network) con capacità di mitigazione degli attacchi DDoS per assorbire il traffico malevolo. (ex. Cloudflare)

Firewall avanzato: Implementare un firewall con funzionalità anti-DDoS che possa rilevare e bloccare il traffico anomalo.

Bilanciamento del carico (Load Balancer): Distribuire il traffico su più server per ridurre l'impatto di un attacco su un singolo server.

* generalmente ci aiuta un query builder oppure un orm nel "lavoro sporco" dietro all'escaping o nel param binding per le query

Limitazione delle richieste: Implementare una rate-limiter per limitare il numero di richieste per IP in un determinato intervallo di tempo.

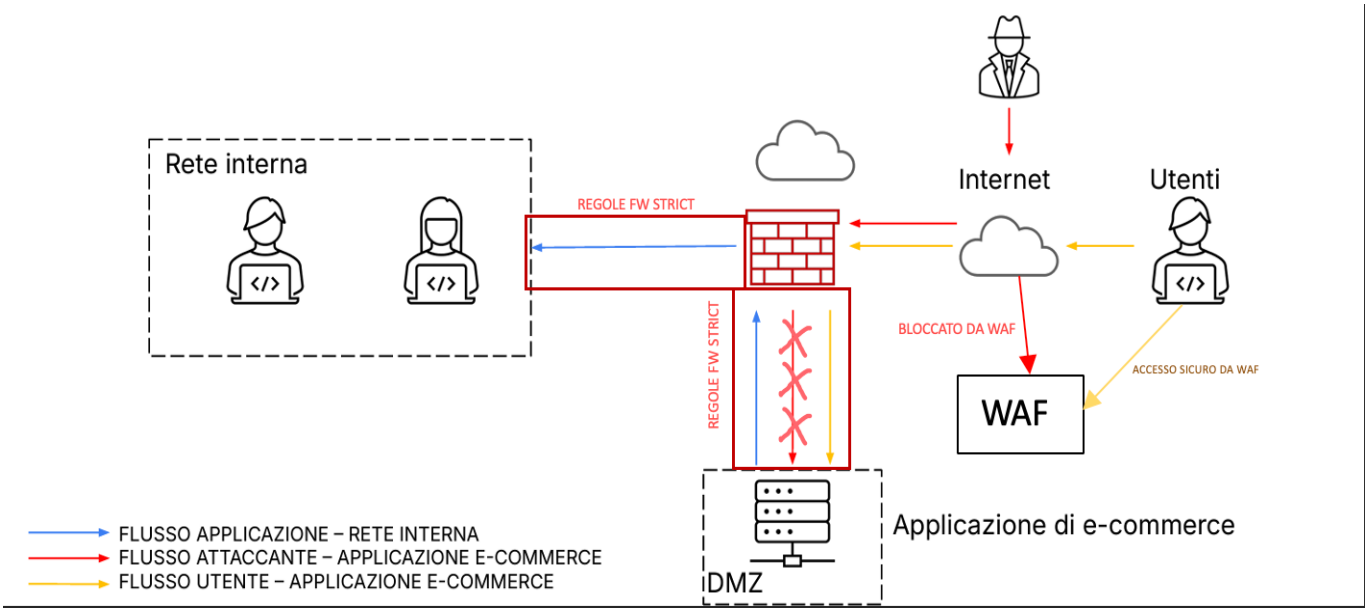
3. Response: infezione da malware

Se l'app venisse attaccata ed infettata si potrebbe implementare:

Segmentazione della rete: Segmentare la rete in sezioni più piccole, con la DMZ isolata dalla rete interna tramite regole di firewall più strict.

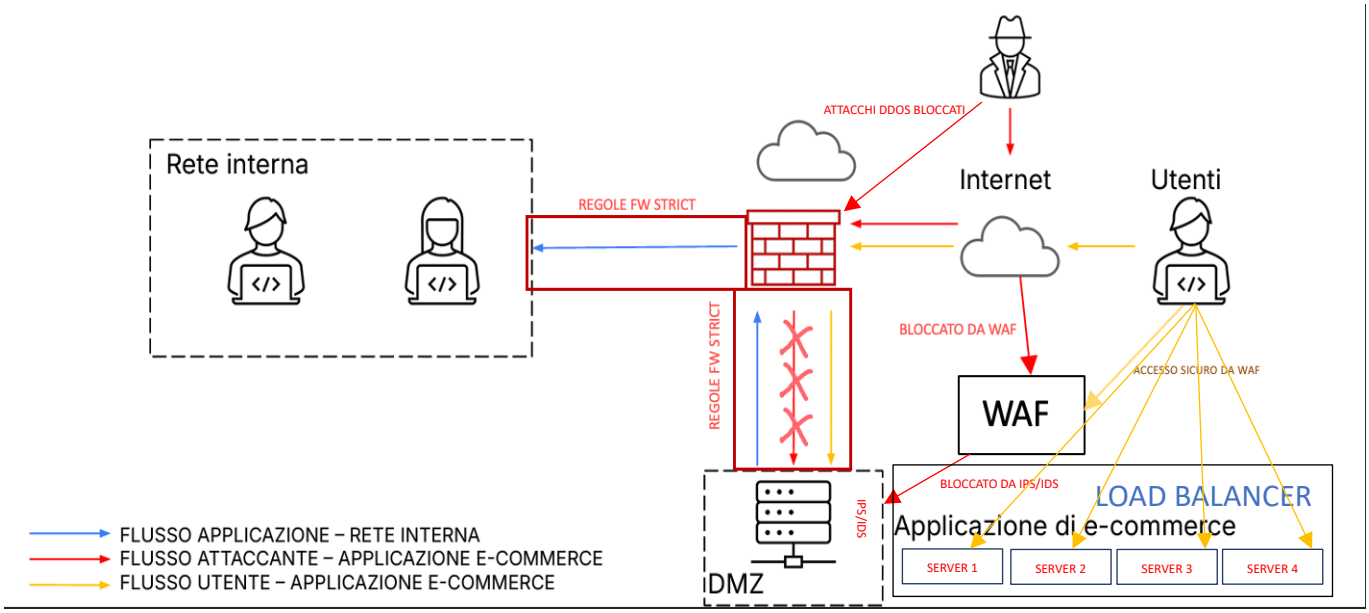
Isolamento della macchina infetta: Disconnettere la macchina infetta dalla rete interna mentre si permette all'attaccante di continuare ad accedere all'area compromessa.

4. Soluzione completa (unire soluzione 1 e 3)



* generalmente ci aiuta un query builder oppure un orm nel "lavoro sporco" dietro all'escaping o nel param binding per le query

5. Modifica “più aggressiva” dell’infrastruttura



* generalmente ci aiuta un query builder oppure un orm nel “lavoro sporco” dietro all’escaping o nel param binding per le query