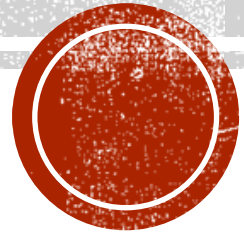
PROGETTO MODULO 5

Mohamed Asry



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**

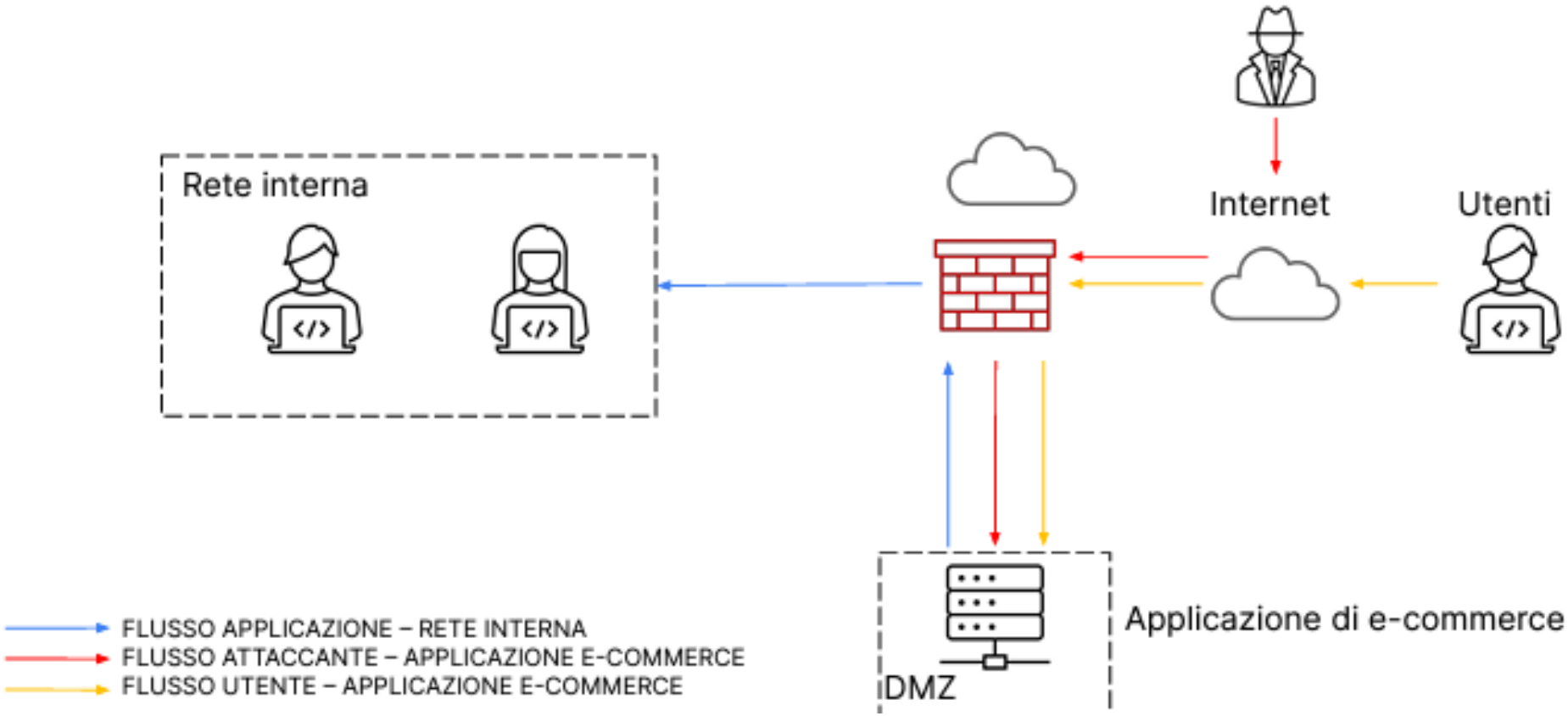
3. **Response**: l'applicazione Web viene infettata da un malware.
   La vostra priorità è che il malware non si propaghi sulla vostre rete, mentre non siete interessati a rimuovere l'accesso da parte dell'attaccante alla macchina infettata.
   Modificate la figura in slide 2 con la soluzione proposta.

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.



1. AZIONI PREVENTIVE

Pratiche Generali

- **Aggiornamento e patching**: Assicurarsi che tutti i software, librerie e framework utilizzati siano aggiornati alle ultime versioni per beneficiare delle patch di sicurezza.
- **Audit di sicurezza regolari**: Eseguire audit di sicurezza regolari, inclusi test di penetrazione e code review, per identificare e risolvere vulnerabilità.
- **Formazione del team di sviluppo**: Educare il team di sviluppo sulle migliori pratiche di sicurezza e sugli attacchi comuni, affinché possano scrivere codice più sicuro.

Implementando queste misure preventive, è possibile ridurre significativamente il rischio di attacchi SQLi e XSS e migliorare la sicurezza complessiva della propria applicazione web.

1. AZIONI PREVENTIVE

Prevenzione SQL Injection (SQLi)

- **Utilizzare query parametrizzate (Prepared Statements)**: Le query parametrizzate assicurano che il codice SQL e i dati siano separati, impedendo così l'inserimento di codice maligno. Questo metodo è supportato da molti database e librerie di accesso ai dati.
- **Utilizzare ORM (Object-Relational Mapping)**: Gli ORM, come SQLAlchemy per Python, Eloquent per Laravel (PHP) e Hibernate per Java, astraggono il livello di database e riducono il rischio di SQLi.
- **Validazione e sanificazione degli input**: Validare e sanificare tutti gli input forniti dagli utenti per assicurarsi che contengano solo dati attesi. Ad esempio, per i campi di testo, si può verificare la lunghezza e il tipo di caratteri.
- **Minimizzare i privilegi del database**: Configurare l'utente del database utilizzato dall'applicazione con privilegi minimi necessari. Evitare di utilizzare l'utente root o admin.
- **Utilizzare un Web Application Firewall (WAF)**: Un WAF può rilevare e bloccare attacchi SQLi noti prima che raggiungano l'applicazione.

1. AZIONI PREVENTIVE

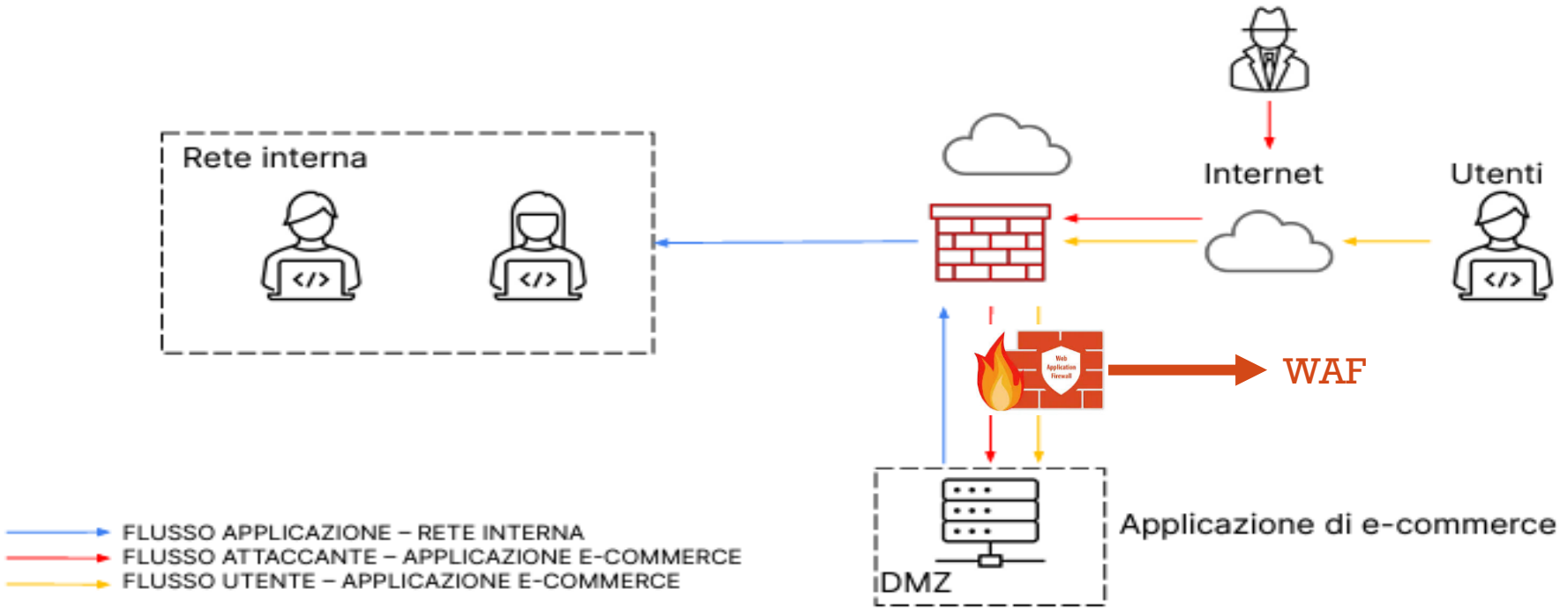
Prevenzione Cross-Site Scripting (XSS)

- **Sanificazione degli output**: Assicurarsi che tutti i dati che vengono renderizzati nelle pagine HTML siano correttamente escapati. Le librerie o i framework moderni spesso hanno funzioni integrate per l'escaping dell'HTML.
- **Content Security Policy (CSP)**: Implementare una CSP per limitare le fonti da cui il browser può caricare risorse come script, fogli di stile e immagini. Questo aiuta a prevenire l'esecuzione di script maligni.
- **Validazione degli input**: Simile alla prevenzione di SQLi, validare e sanificare gli input forniti dagli utenti per assicurarsi che contengano solo dati attesi.
- **HttpOnly e Secure flag sui cookie**: Impostare i flag HttpOnly e Secure sui cookie per proteggerli da accessi JavaScript e per assicurarne la trasmissione solo su connessioni sicure (HTTPS).
- **Sanificazione del markup HTML**: Utilizzare librerie di sanificazione come DOMPurify (per JavaScript) o Bleach (per Python) per pulire i contenuti HTML forniti dagli utenti.

1. AZIONI PREVENTIVE

Per la protezione della Web App da minacce quali XSS e SQLi si può preventivamente adottare una soluzione basata su Web Application Firewall, che a differenza dei firewall standard, sono dedicati per proteggere le Web App da attacchi XSS e SQLi.

Di seguito abbiamo il disegno con il WAF implementato:



2. IMPATTI SUL BUSINESS

L'impatto economico su un'applicazione web colpita da un attacco DDoS che, considerando che in media ogni minuto gli utenti spendono €1.500 sulla piattaforma e-commerce, che rimane non raggiungibile per 10 minuti è pari alla spesa potenziale degli utenti per minuto moltiplicato per i minuti di irragiungibilità dell'applicazione web.

Il mancato guadagno del business di conseguenza è il seguente:

Mancato guadagno/Danno = €1.500 x 10 (minuti) =€15.000 persi

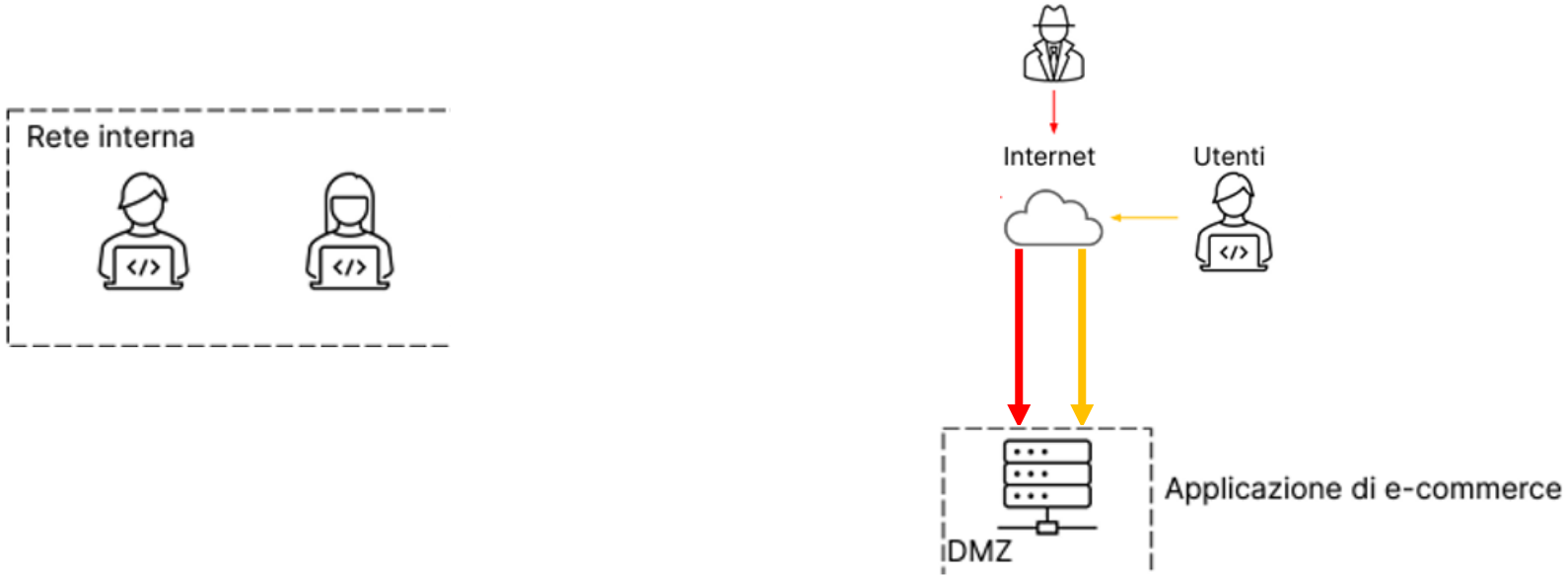
Alcune azioni preventive che si potrebbero prendere in considerazione per evitare incidenti di questo tipo sono le seguenti:

- **Servizi di protezione DDoS**: Servizi specializzati come Cloudflare, Akamai, o AWS Shield possono identificare e mitigare il traffico malevolo prima che raggiunga il server dell'applicazione;
- **Ridondanza e failover**: Configurare infrastrutture di rete ridondanti e meccanismi di failover automatici per garantire che, in caso di attacco a un server, il traffico possa essere reindirizzato a un server di backup senza interruzioni significative;
- **Monitoraggio continuo**: Implementare sistemi di monitoraggio e allerta per rilevare anomalie nel traffico in tempo reale, permettendo una risposta rapida agli attacchi;
- **Piani di risposta agli incidenti**: Avere un piano dettagliato di risposta agli incidenti che includa procedure di comunicazione e mitigazione per ridurre al minimo i tempi di inattività.

3. RESPONSE

Nel caso in cui l'applicazione Web venisse infettata da un malware e la priorità fosse impedire la propagazione del malware sulla rete aziendale, possiamo adottare una strategia basata sull'isolamento della macchina infettata. In questo caso la macchina sarà direttamente collegata ad internet, raggiungibile dall'attaccante ma non più connessa alla rete interna.

Di seguito abbiamo l'architettura di rete dell'applicazione web con la macchina infettata isolata dalla rete interna.



L'applicazione web è isolata dalla rete interna ma raggiungibile da utenti e dall'attaccante tramite internet.

4. SOLUZIONE COMPLETA