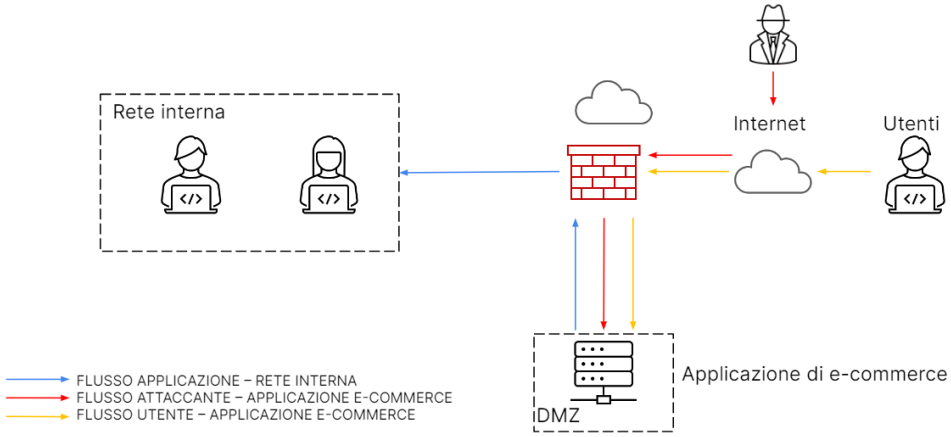
ESERCIZIO M5 D8

PROGETTO – INCIDENT RESPONSE

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.



3

1- AZIONI PREVENTIVE

Prevenzione contro SQL Injection (SQLi):

1. Validazione dei dati di input: Assicurarsi di validare e sanificare correttamente tutti i dati di input forniti dagli utenti; utilizzare metodi come le istruzioni di parametrizzazione o gli statement preparati per evitare l'interpolazione di input non valido nelle query SQL.

2. Utilizzo di meccanismi di accesso ai dati sicuri: sarebbe preferibile l'utilizzo di framework ORM (Object-Relational Mapping) o librerie di accesso ai dati che automatizzano la generazione di query SQL sicure. Questi strumenti spesso gestiscono correttamente l'escaping dei caratteri speciali e prevengono gli attacchi SQLi.

3. Pratiche di autenticazione e autorizzazione solide: implementare un sistema di autenticazione sicuro per verificare le credenziali degli utenti. Usare algoritmi di hashing robusti per memorizzare le password degli utenti e evita l'uso di metodi di autenticazione obsoleti. Inoltre, dovremmo assicurarci che gli utenti abbiano solo i privilegi necessari e applicare rigorose politiche di autorizzazione per limitare l'accesso ai dati sensibili.

4. Limitazione dei privilegi del database: configurare l'utente del database utilizzato dall'applicazione in modo da avere solo i privilegi strettamente necessari per l'esecuzione delle operazioni richieste. Può essere importante ridurre al minimo l'accesso in scrittura al database per l'applicazione stessa.

Prevenzione contro Cross-Site Scripting (XSS):

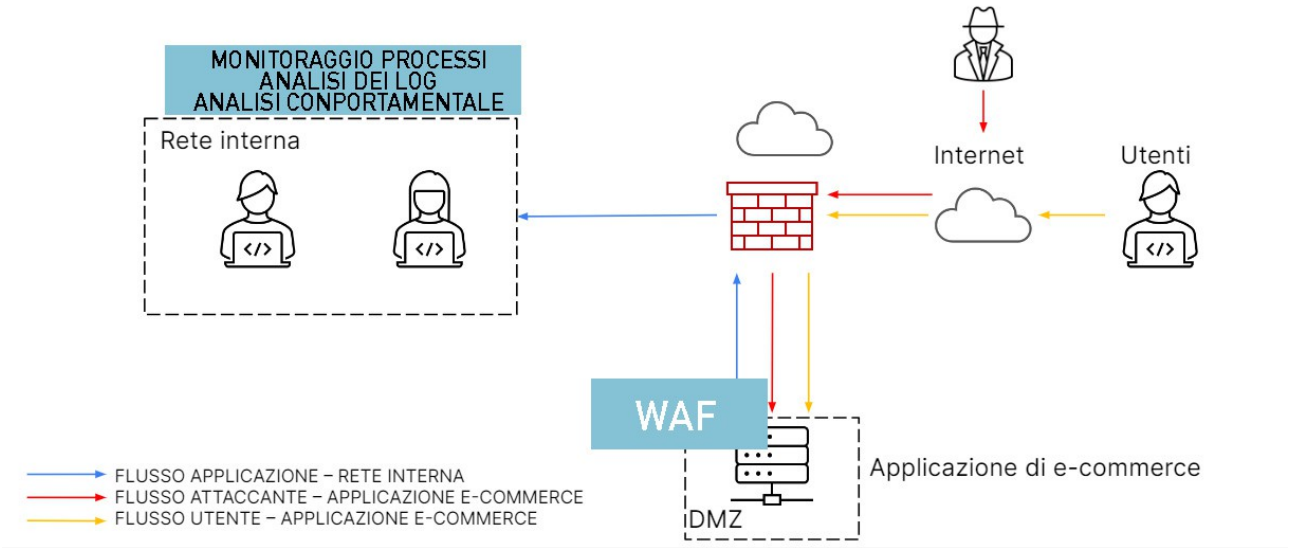
1. Validazione e filtraggio dei dati di input: assicurarsi di validare e filtrare attentamente tutti i dati di input forniti dagli utenti. Rimuovere o disabilitare qualsiasi markup o script potenzialmente pericoloso dai dati di input.
2. Escape corretto dell'output: quando si visualizzano i dati forniti dagli utenti all'interno delle pagine web, assicuriamoci di eseguire un'operazione di "escaping" corretta per evitare l'esecuzione indesiderata di script. Ovviamente dovremmo utilizzare funzioni di escape adeguate in base al contesto di output (ad esempio, HTML escape, JavaScript escape, ecc.).

3. Utilizzo di Content Security Policy (CSP): implementare una CSP adeguata per l' applicazione web. La CSP aiuta a mitigare gli attacchi XSS specificando quali tipi di contenuti sono consentiti nella tua applicazione.

4. Aggiornamento regolare dei framework e delle librerie: assicuriamoci di tenere sempre aggiornati i framework e le librerie utilizzati nella tua applicazione web. Spesso, gli sviluppatori rilasciano patch di sicurezza per risolvere le vulnerabilità note. Mantieni la tua applicazione web aggiornata con le ultime correzioni di sicurezza.

Queste sono solo alcune delle azioni preventive che possiamo implementare per difendere un'applicazione web da attacchi come SQL Injection (SQLi) e Cross-Site Scripting (XSS). Ad ogni modo dovremmo tener presente che la sicurezza delle applicazioni web è un processo continuo e che richiede l'adozione di buone pratiche di sviluppo e test regolari per identificare e mitigare le vulnerabilità.

Da un punto di vista più pratico potremmo prevedere un punto di moniotraggio attivo che possa fare in modo di far analizzare i processi dal team che gestisce la nostra web app creando degli alert e delle regole precise per far in modo che questo tipo di attacchi sia sotto controllo costante

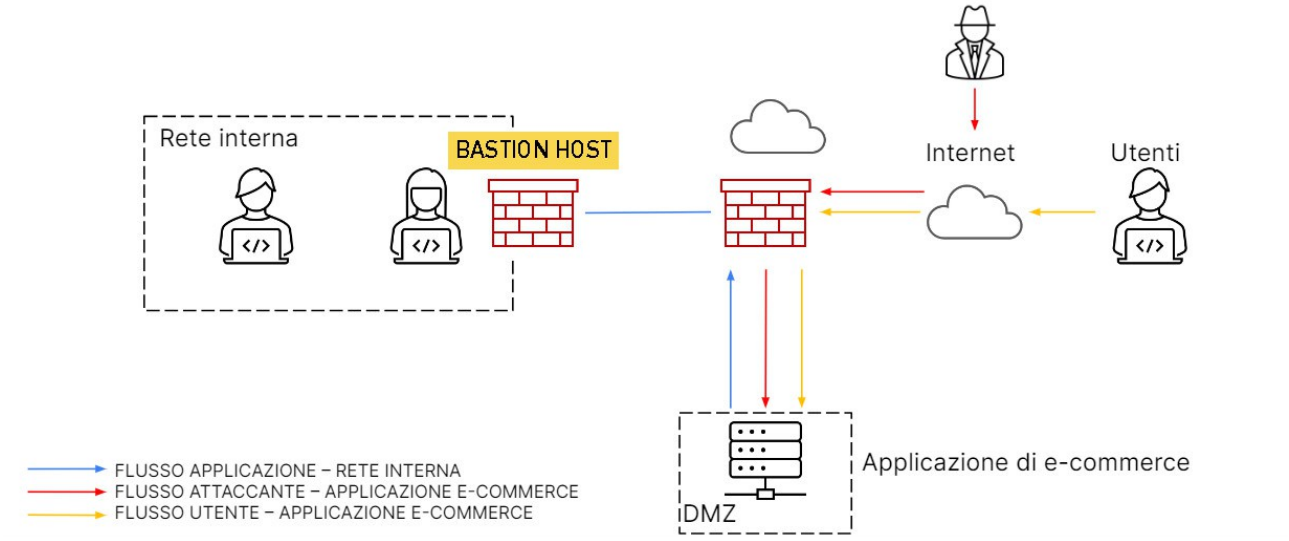


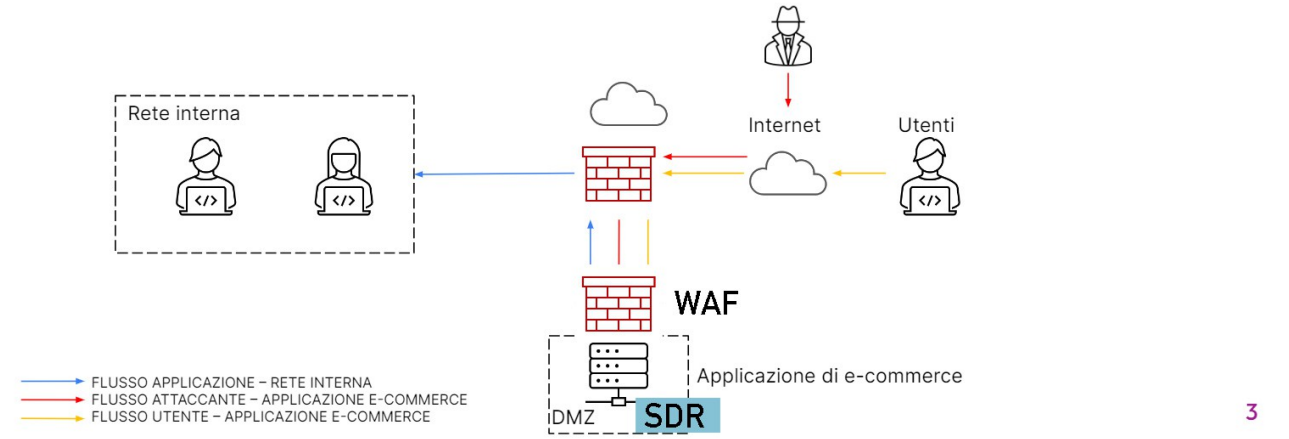
2 – IMPATTI SUL BUSINESS (+ IMPLEMENTAZIONE PUNTO 5)

L'app web sta subendo un attacco DDoS rendendo l'app non raggiungibile per 10 minuti. In media, sulla piattaforma, i clienti spendono 1500€ al minuto. Quindi l'impatto sul business sarà:

1500x10= 15.000€

Quindi a fronte di un danno ingente si potrebbe pensare di proteggere in maniera più precisa la rete interna con un altro firewall (bastion host) o con un WAF (web application firewall) ovvero la configurazione a bastioni dove ogni firewall "protegge la propria rete cercando di arrivare ad una tipologia di approccio "zero trust". Potrebbe anche essere inserito un server di ridondanza (SDR) per aiutare la business continuity.





Per quanto riguarda le azioni preventive che possono essere adottate per affrontare questo problema, ecco alcune valutazioni:

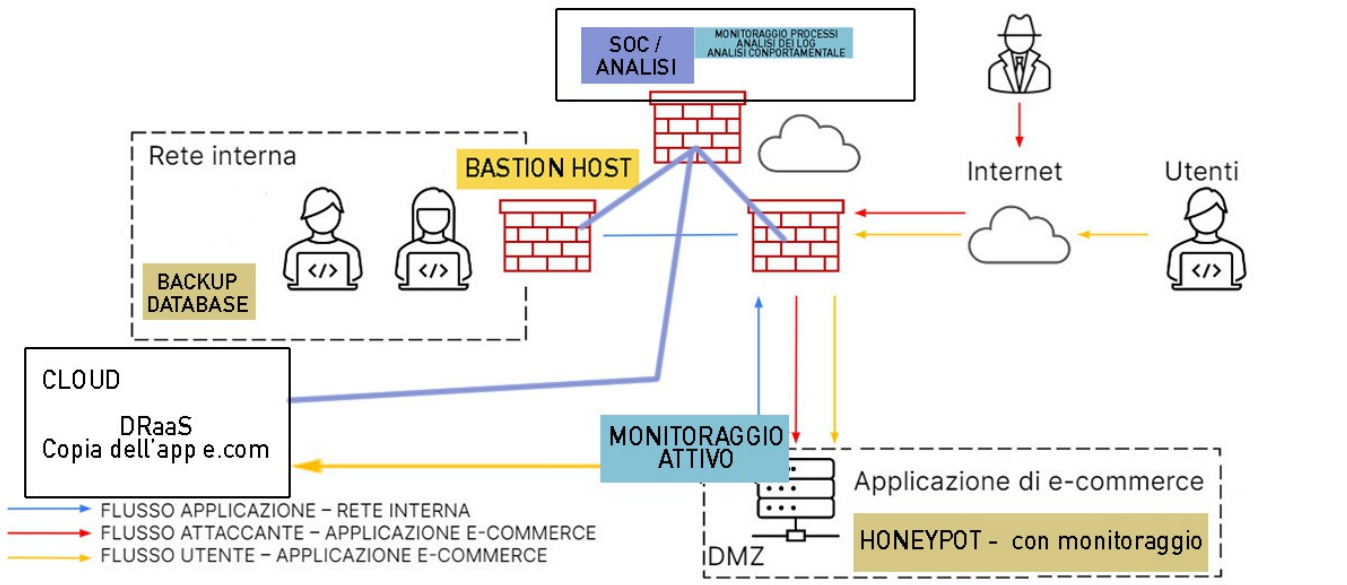
1. Implementazione di un sistema di mitigazione DDoS: si può considerare l'installazione di un sistema di protezione DDoS che rilevi e blocca gli attacchi in tempo reale, riducendo così l'impatto sulle prestazioni del servizio.
2. Scalabilità e ridondanza: assicurarsi che il servizio sia in grado di gestire carichi

di traffico elevati. Si può considerare l'utilizzo di servizi cloud scalabili o la distribuzione di server in diverse regioni geografiche per garantire che il servizio rimanga accessibile anche in caso di attacchi DDoS.

3. Monitoraggio del traffico e rilevamento precoce: come descritto sopra, implementare un sistema di monitoraggio del traffico che consenta di rilevare e rispondere rapidamente agli attacchi DDoS. Si possono utilizzare strumenti di analisi del traffico e impostare avvisi per essere avvisato in caso di attività sospette.
4. Backup e ripristino dei dati: dovremmo assicurarci di avere un sistema di backup regolare dei dati critici e un piano di ripristino in caso di danni o perdita di dati dovuti agli attacchi DDoS.
5. Consapevolezza e formazione del personale: dovremmo fornire al personale informazioni sulla sicurezza informatica, incluso come riconoscere e rispondere agli attacchi DDoS. Un'adeguata formazione può aiutare a mitigare gli effetti negativi degli attacchi e a prendere misure preventive.

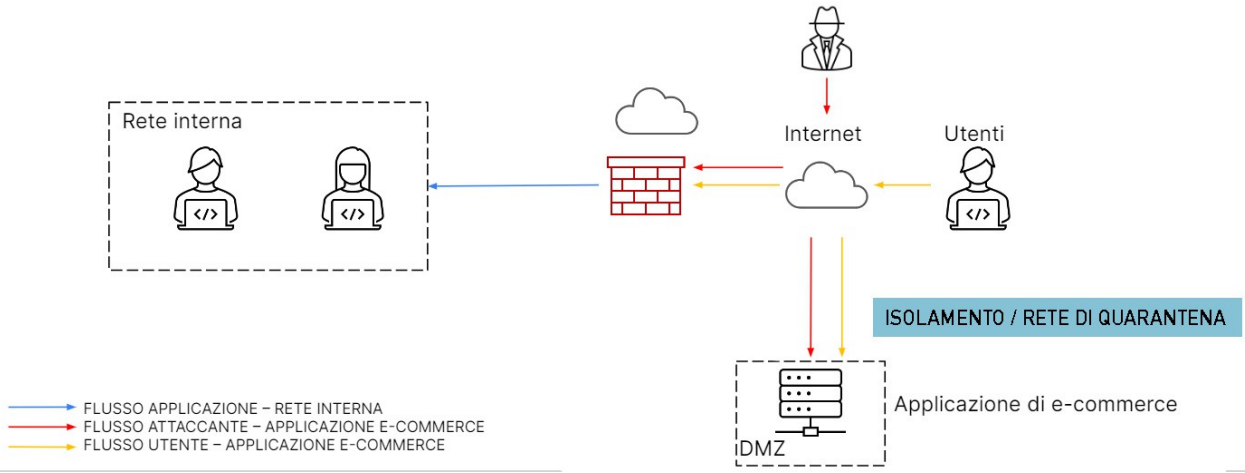
Queste sono solo alcune delle azioni preventive che possiamo considerare per affrontare gli attacchi DDoS e cercare di ridurre l'impatto sul business. La scelta delle misure preventive dipenderà dalle esigenze specifiche e ovviamente dalle risorse disponibili.

Se come proposto nel punto 5 pensassimo ad una modifica pin “aggressiva” della struttura (considerato che una modifica aggressiva aumenterebbe sia le spese di attivazione che di costi generali in modo cosiderevole) potremmo pensare a “blindare” la nostra attività inserendo un centro di monitoraggio stabile esterno dalle reti che possa solo analizzare in entrata quello che succede sulle reti e quindi avere un tempo minimo di risposta alla vulenrabilità ma soprattutto un costante lavoro di intelligence per evitare e prevedere futuri attacchi. Potrebbe essere considarato in un piano di Business continuity e Disater recovery una certa attenzione al backup dei dati e magari un Draas in cluod qualora la nostra struttura on premise andasse down il tutto teso ad eliminare i “single point of failure” SPOF.



3 – RESPONSE

L'app web viene infettata da un malware. Per fare in modo che non si propaghi nella rete potremmo creare una rete di quarantena che isoli il server dove gira la nostra web app. In questo modo l'attaccante è sempre in contatto con la nostra web app ma la web app non ha modo di comunicare con la nostra rete interna.



4 – SOLUZIONE COMPLETA

Cercando di mettere insieme tutti i punti dell'esercizio potremmo pensare che un buon punto di partenza per cercare di alzare il livello della sicurezza del nostro sistema potrebbe essere quello di prevedere un sistema di firewalling a bastioni (WAF) e almeno un punto di monitoraggio attivo (IDS) per fare in modo che la nostra parte del sistema più esposta sia controllata in modo adeeguato e costante quindi qualora ci fossero degli eventi fuori dalle regole da noi impostate il team potrà essere avvertito immediatamente e quindi riuscire a trovare le contromisure necessarie nel minor tempo possible e quindi arginare il danno sia economico che tecnico. In aggiunta un SDR (server di ridondanza).