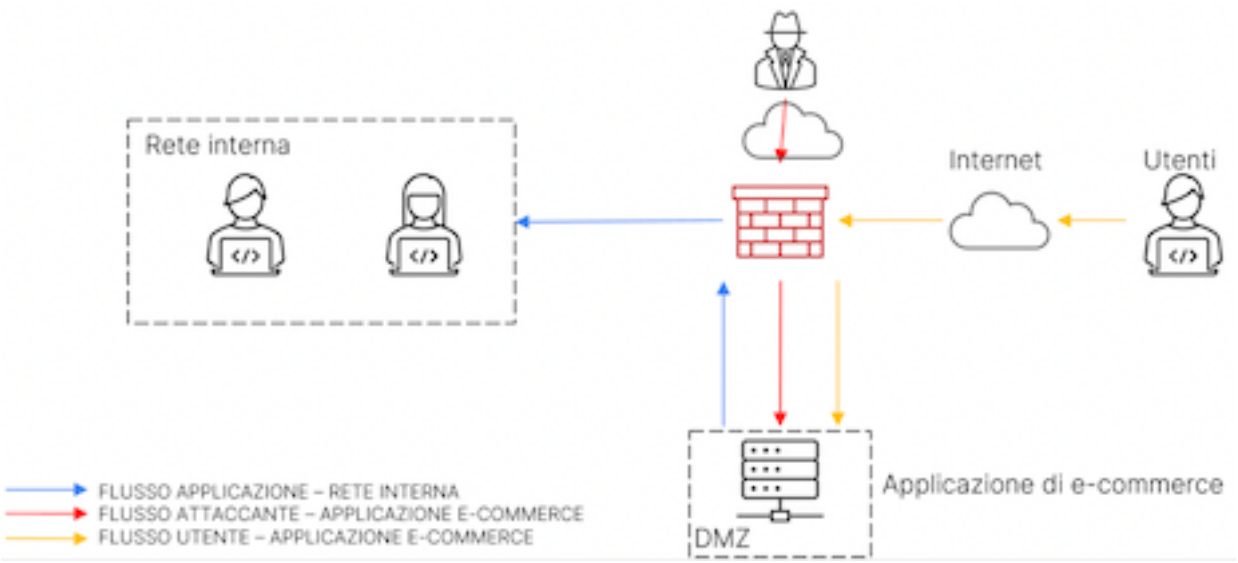
Nell' esercizio di oggi dobbiamo condurre delle azioni preventive in caso di un attacco SQLi (o XSS) sulla nostra applicazione web da parte di un attaccante. Dobbiamo verificare l'impatto sul business in caso di attacco DDoS. Evitare la propagazione del malware che ha infettato l'applicazione Web sulla nostra rete.



L' SQLi è una tecnica hacking che prevede l'inserimento e l'esecuzione di codice SQL non previsto all'interno di applicazioni web sul database.

AZIONI PREVENTIVE

Per prevenire questi attacchi, diversi sono le tecniche da applicare, come per esempio:

1)Web Application Firewall

protegge le applicazioni web da attacchi dannosi tra i quali l' Injection.
Esso protegge le web app monitorando e bloccando qualsiasi traffico HTTP/HTTPS dannoso che viaggia verso la nostra applicazione web e impedisce che i dati non autorizzati escano. Il WAF applica una serie di criteri che possono essere modificati in base alle esigenze rispetto alla nostra web app. Il WAF analizza le comunicazioni prima che queste raggiungono l'app o l'utente.

2) Funzione PHP mysqli_real_escape_string()

consiste in una pre-elaborazione dei dati inviati tramite un modulo web,convertendo i caratteri speciali ostili alla query e rendendo innocui i tentativi di hacking.

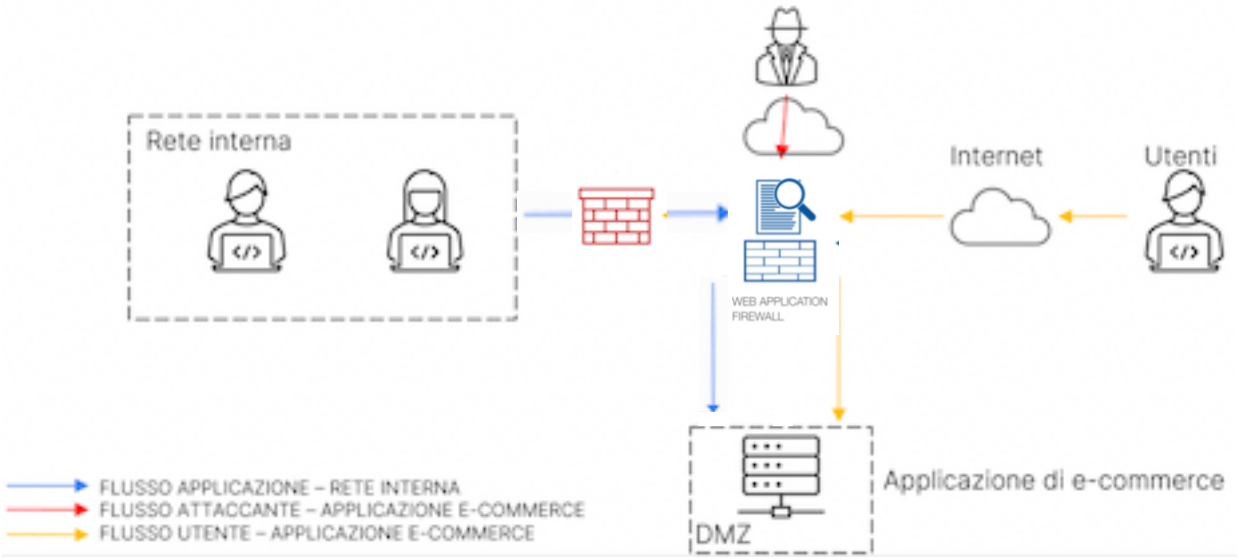
3)Blocco dei caratteri speciali nelle aree di input

Questa tecnica rientra nelle query parametriche con il fine di porre limitazioni al contenuto di qualche campo
sappiamo che ogni campo input di un form presenta:

- Input di testo (come username)
- Password
- Email
- Textarea

Si intuisce facilmente quali valori vanno immessi nei campi, quindi in fase di scrittura codice, si possono bloccare quei caratteri come per esempio gli "apici, simboli speciali o spazi" che vengono utilizzati nell'iniezione del codice malevolo

Prendiamo in esame solamente la prevenzione tramite WAF
Qui di seguito un grafico su come si possa evitare tramite azioni preventive un attacco al nostro e-commerce. Si noti come viene inserito il WAF in modo tale da controllare e analizzare le comunicazioni prima di arrivare al nostro sito. Secondo i criteri preconfigurati, l'attaccante non avrebbe accesso al nostro e-commerce qualora decidesse di usare un attacco SQL.



IMPATTI SUL BUSINESS a seguito attacco DDoS

Ipotizziamo inoltre che la nostra applicazione subisce un attacco di tipo DDos, cioè un attacco che mira a rendere la pagina irraggiungibile ai clienti, con lo scopo di interrompere i servizi di un'azienda utilizzando volumi enormi di traffico e saturandone la banda di comunicazione. Questo tipo di attacco è mirato a mandare in down il sito al fine di procurare una perdita in denaro al target.

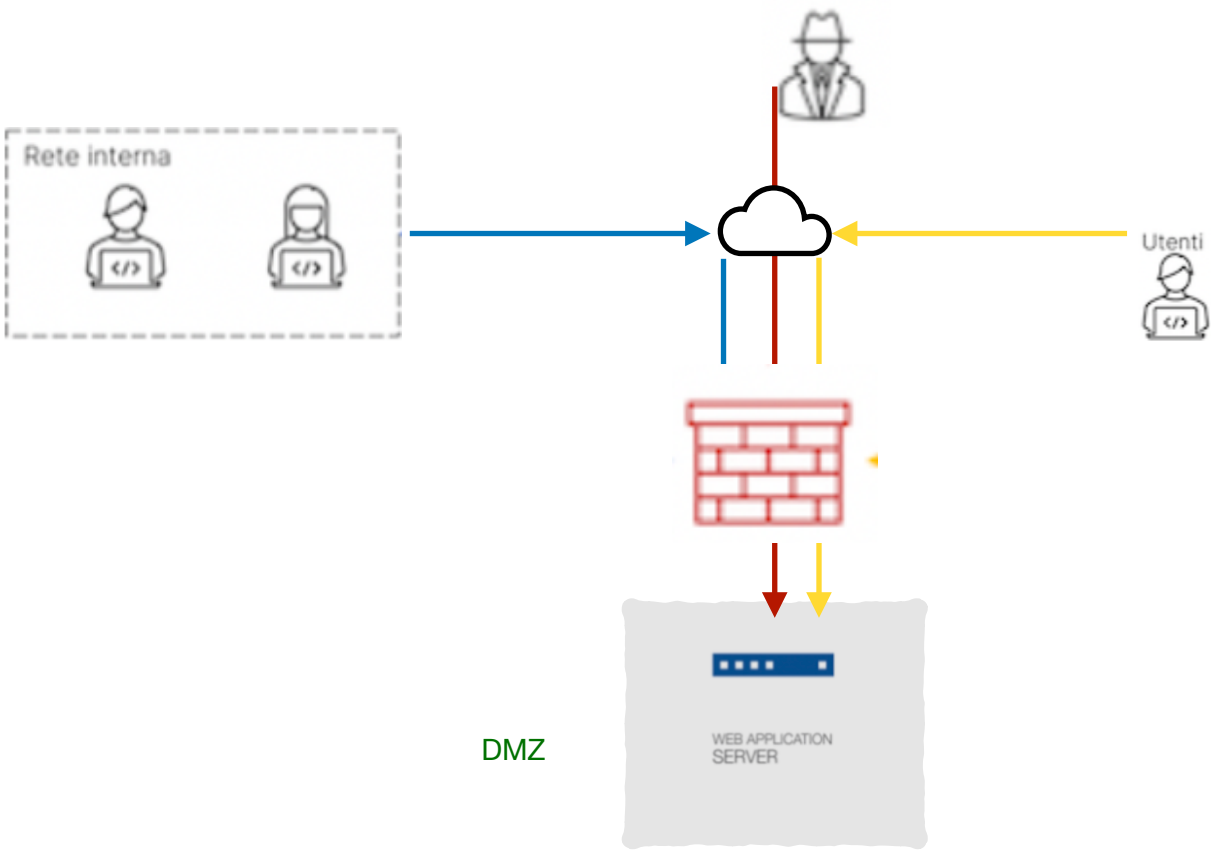
La nostra web app ha appena subito un attacco di tipo DDoS che rende l'applicazione non raggiungibile per 10 minuti. Considerando che in media ogni minuto gli utenti spendono 1500€ sulla nostra piattaforma possiamo calcolarne l'impatto sul business cosi come segue:

Spesa media utenti per minuto = 1500€
Tempo di non raggiungibilità della web app = 10 minuti

1500 x 10 = 15000€

RESPONSE a seguito attacco Malware

Abbiamo subito un attacco Malware e vogliamo far si che quest'ultimo non si propaghi sulla nostra rete.
Il primo passo da fare è quello di isolare la web application dalla Lan rimuovendo la regola di routing che mette in comunicazione la nostra Lan alla DMZ
Lasciamo inoltre aperto in ingresso le connessioni degli utenti-clienti in modo tale che l'e-commerce possa continuare la sua attività.
Sarebbe inutile vietarne l'accesso ai clienti in quanto una volta dentro, l'attaccante avrebbe già il controllo su tutto il database.
Quest'azione sarebbe efficace solamente se il malware è stato appena rilevato e quindi non ha ancora compromesso il servizio. In caso contrario dobbiamo isolare del tutto la web app anche dalla rete internet per poter risolvere l'accatto.



SOLUZIONE completa.

Qui di seguito viene proposta una soluzione definitiva unendo la difesa da attacco SQLi o XXS alla difesa dalla propagazione del Malware.
Come vediamo, l'attaccante si ritrova bloccato dal WAF che gli nega l'accesso alla web app.
Noi, come rete interna, possiamo accedere direttamente al sito non passando tramite la rete internet, mentre l'utente può accedere tranquillamente al sito poichè il suo traffico non è riconosciuto dal WAF come dannoso.