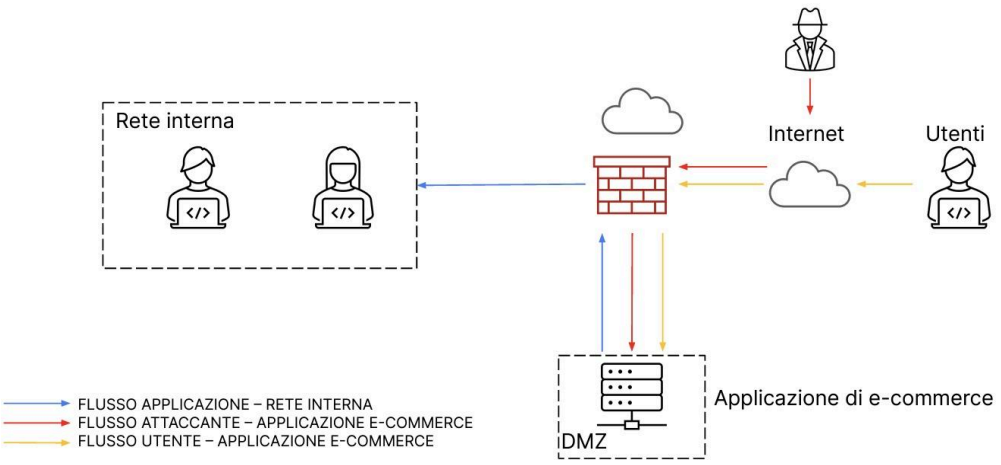
Traccia:
Con riferimento alla figura in slide 2, rispondere ai seguenti quesiti.
1. Azioni preventive: quali azioni preventive si potrebbero implementare per difendere l'applicazione Web da attacchi di tipo SQLi oppure XSS da parte di un utente malintenzionato? Modificate la figura in modo da evidenziare 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.



Azioni preventive 1

XSS: per prevenire questa tipologia di attacco è bene sanitizzare la web app tramite **escaping** e **encoding** degli input trasformando i simboli del codice javascript (quali </>' maggiormente utilizzati per introdurre degli script) in altri caratteri (quali ad esempio $ and commerciale) inserendo ex ante nel codice html/javascript la funzione php **htmlspecialchars**.

Questo perchè se la web app non fosse sanitizzata si potrebbero ottenere cookie, token, fingerprinting, redirecting (posso rimandare l'utente su un'altra pagina) e logging ecc a cui la vittima accede dando i propri dati.

SQLi: per prevenire questa tipologia di attacco, in Javascript si potrebbe utilizzare il prepared statement il quale andrà a cercare tutto quello inserito.

Il classico esempio di sql injection potrebbe essere il seguente:

SELECT * FROM users WHERE id = 19' OR 1=1 #

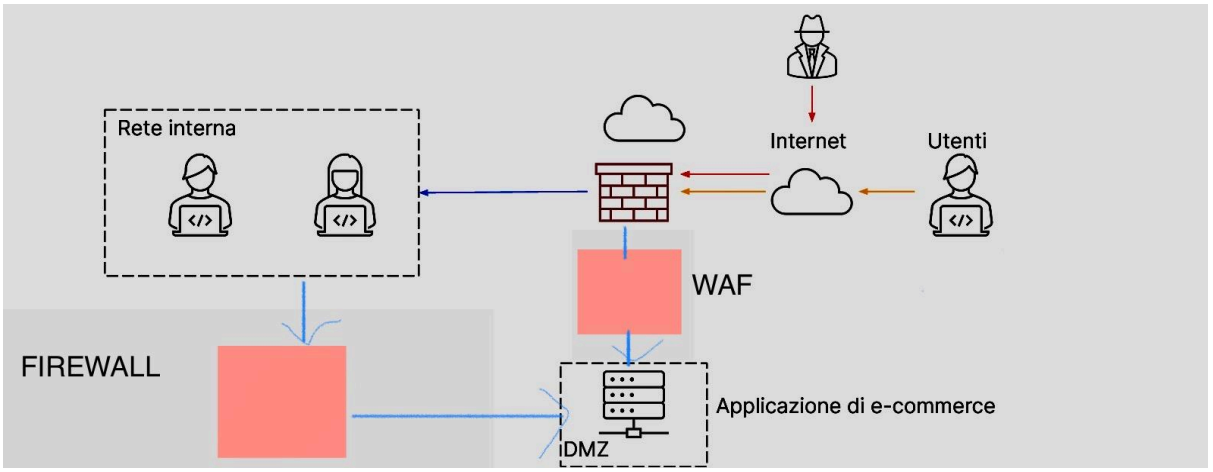
Si potrebbe inserire questo in Java:

SELECT * FROM users WHERE id = ?; del sistema operativo.

Dove il '?' Andrà a prendere tutto l'input inserito, nel nostro caso "19' or 1=1#", in questo caso una volta interrogato il database non andrà a restituire tutti i dati richiesti, ma si andrà a cercare letteralmente la cercare "1' or 1=1#" nel database e ovviamente non restituirà alcun risultato.

Altrimenti un altro modo per evitare attacchi di SQLi e XSS sarebbe l'introduzione di una web application firewall!

Questa web application firewall bloccherebbe le richieste provenienti da IP noti ad attacchi. Sempre questa WAF contiene quanto detto pocanzi in merito alla santificazione degli input per evitare XSS e SQLi.



Impatto sul business

La non raggiungibilità del web application all'azienda costerà 15000 euro.

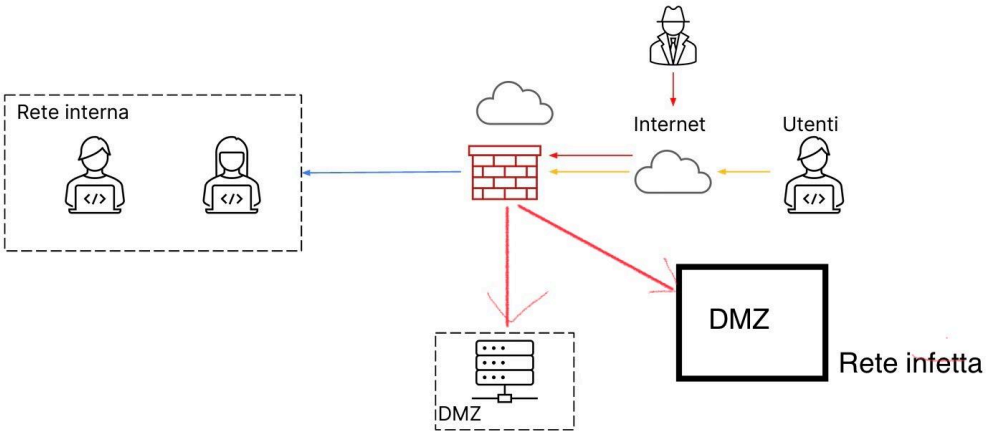
Per prevenire un attacco Ddos è bene implementare un sistema di monitoraggio della disponibilità del servizio per rilevare rapidamente gli attacchi e rispondere prontamente e bloccare il traffico filtrando gli IP che mandano richieste per bloccare il traffico sospetto.

Response

Una volta entrato il malware all'interno della macchina e non si è interessati si potrebbe lasciarlo lì, il tutto con lo scopo di studiare i suoi comportamenti.
mentre nel server di backup si potrebbe reindirizzare tutto il traffico dell'ecommerce.

Architettura di rete:
L'applicazione di e-commerce deve essere disponibile per gli utenti tramite Internet per effettuare acquisti sulla piattaforma.

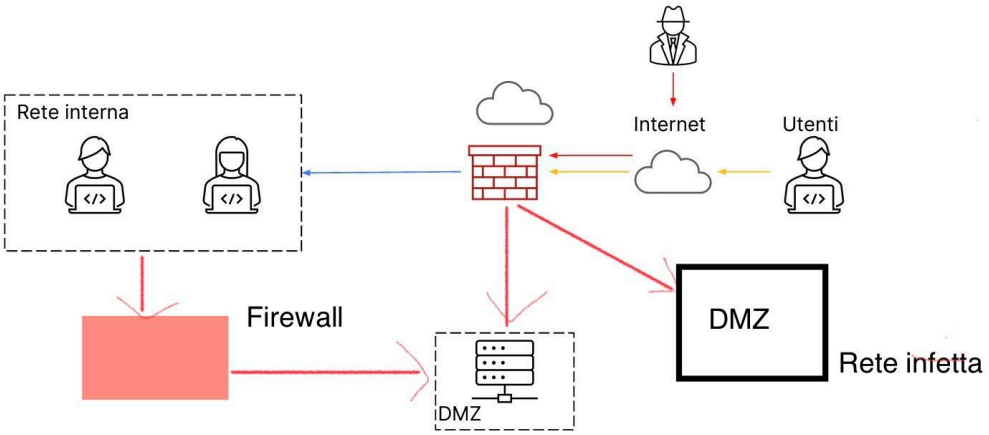
La rete interna è raggiungibile dalla DMZ per via delle policy sul firewall, quindi se il server in DMZ viene compromesso potenzialmente un attaccante potrebbe raggiungere la rete interna.



Soluzione completa

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.



Modifica aggressiva dell'infrastruttura

La modifica più aggressiva sarebbe la formattazione del disco rigido e reinstallazione del sistema operativo oppure l'isolamento totale della rete (il che non avrebbe impatti negativi e non permetterebbe la vendita dell'ecommerce).