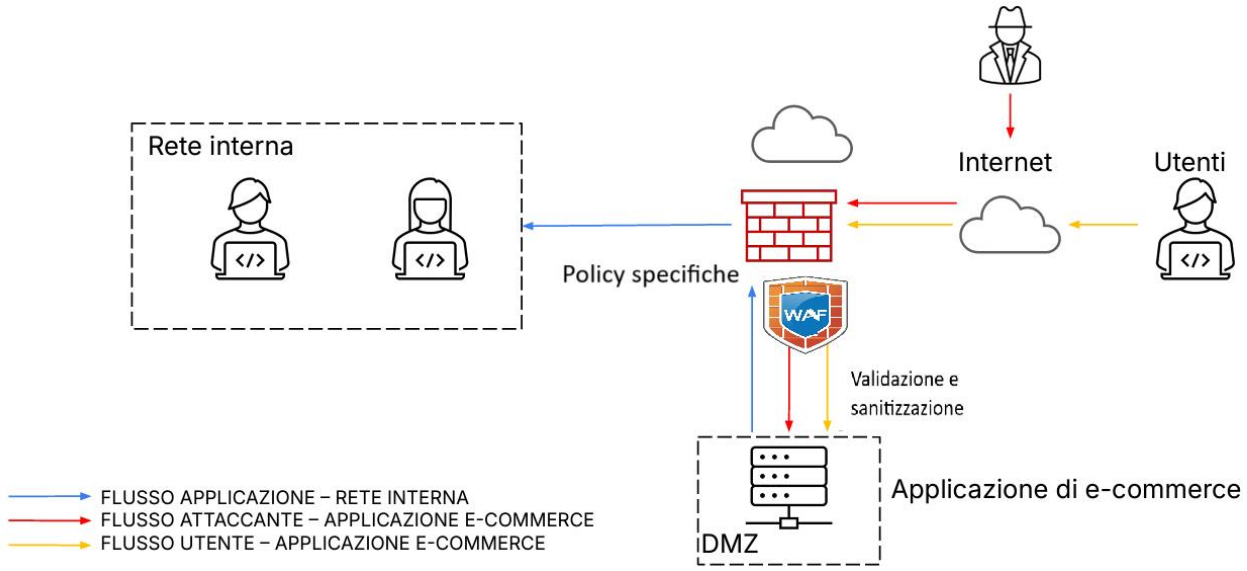
PROGETTO DI FINE MODULO M5

1. Azioni preventive:

Con riferimento alla figura modificata, sono state identificate e implementate una serie di **azioni preventive** per proteggere l'applicazione di e-commerce da attacchi di tipo **SQL injection (SQLi)** e **Cross-site scripting (XSS)**, che sono tra le minacce più comuni per le applicazioni Web esposte a Internet.



Azioni preventive implementate:

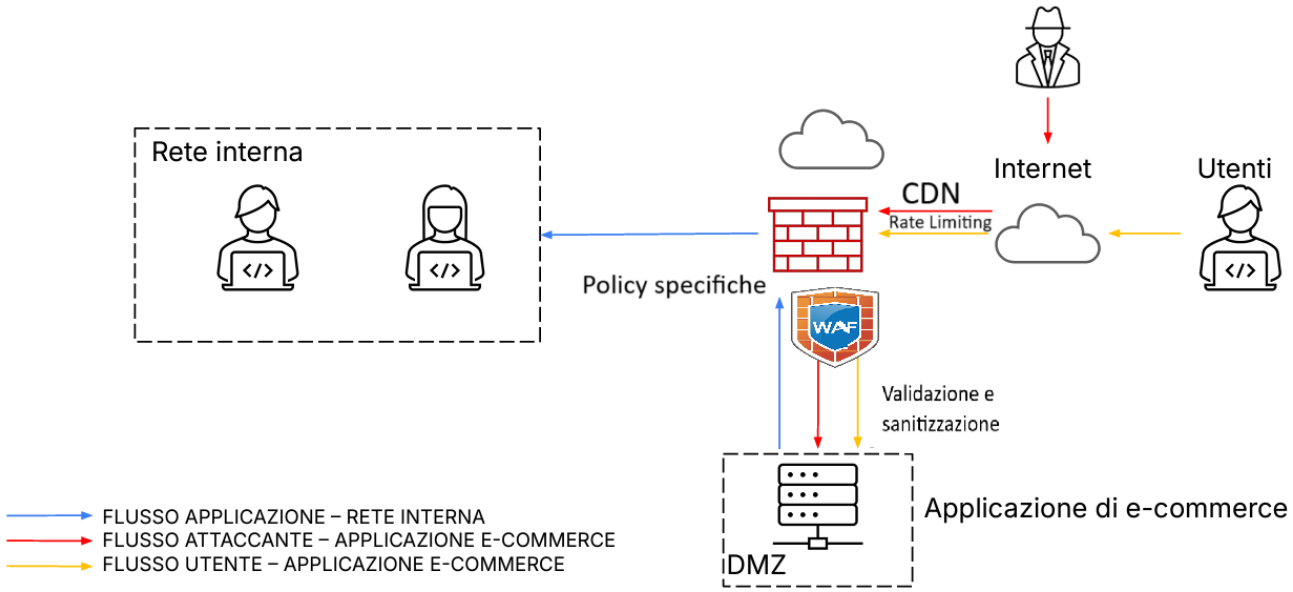
1. **Validazione e sanitizzazione degli input**
   - Tutti i dati in ingresso inviati dagli utenti vengono validati e filtrati prima di essere elaborati dall'applicazione (es rimozione caratteri non validi, escaping di caratteri speciali, utilizzo di whitelist per i valori ammessi).
   - Questa misura è efficace per bloccare comandi malevoli inseriti in input (tipici degli attacchi SQLi e XSS).
2. **Firewall WAF con Regole Specifiche**
   - È stato rafforzato il firewall con regole di controllo sui pacchetti e analisi del traffico per individuare pattern tipici di attacchi SQLi e XSS.
   - Il firewall WAF funge da secondo filtro per bloccare traffico sospetto diretto all'applicazione web.
3. Policy specifiche al primo firewall per impedire l'accesso alla rete interna ad utenti non autorizzati

2. Impatti sul Business:

La perdita stimata è di **15.000 €**, senza considerare ulteriori danni indiretti come:

- Calo della fiducia degli utenti
- Reclami e assistenza clienti
- Costi tecnici per mitigazione e recupero del servizio

Per proteggere l'applicazione da questo tipo di attacchi, è possibile **estendere la prima immagine** con le seguenti modifiche:



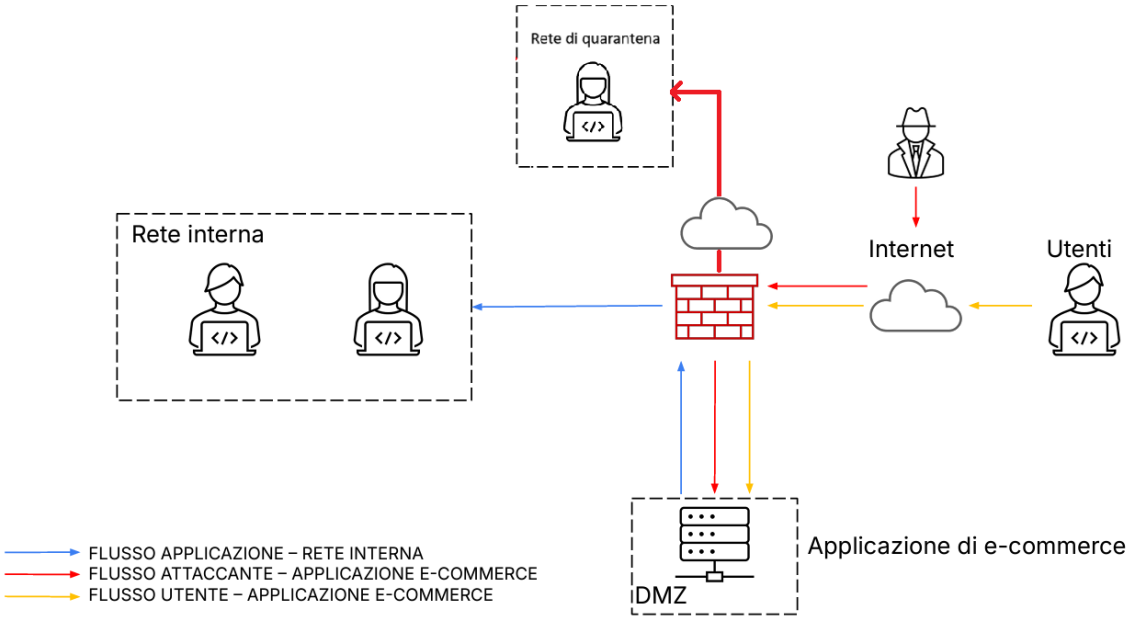
Per proteggere l'applicazione da questo tipo di attacchi, è possibile **estendere la prima immagine** con le seguenti modifiche:

1. **CDN con protezione DDoS**
   - Inserita tra Internet e firewall.
   - Filtra e distribuisce il traffico su più nodi, mitigando l'effetto degli attacchi volumetrici.
   - Es.: **Cloudflare**, **Akamai**, **AWS Shield**.
2. **WAF (Web Application Firewall)**
   - Già presente nella prima immagine, va potenziato per bloccare anche traffico sospetto su larga scala.
   - Rileva pattern noti di DDoS a livello applicativo (HTTP flood, slow POST, ecc.).
3. **Rate Limiting**

- Limita il numero di richieste per utente/IP, evitando il sovraccarico del sistema.

3. Response:

L'applicazione Web è stata infettata da un **malware**. L'obiettivo **non** è bloccare l'attaccante, ma impedire che il malware si **propaghi sulla rete interna** o ad altri sistemi.



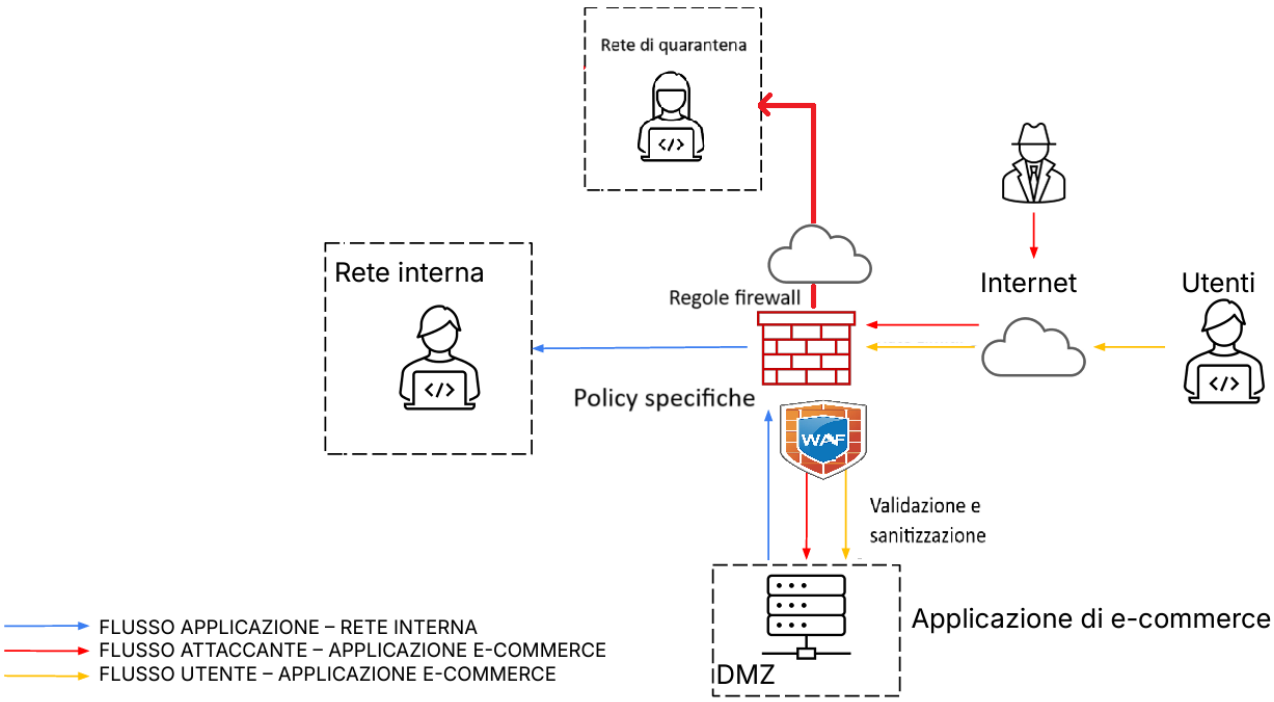
Per impedire la propagazione del malware, si interviene immediatamente con **misure di contenimento**. Nella figura modificata vengono rappresentati i seguenti elementi:

Isolamento di rete (Network Segmentation & Quarantine):

- La macchina compromessa viene **isolata** da tutte le altre componenti della rete interna tramite regole sul firewall.
- Viene spostata in una zona chiamata **quarantine network** (rete di quarantena), per limitare la riproduzione del malware e l'accesso al resto della rete da parte dell'attaccante.
- Solo l'attaccante continua ad avere accesso alla macchina infetta, come richiesto.

4. Soluzione completa

In questa soluzione vengono mostrate insieme l'azione preventiva e la response in un unico diagramma.


Rete di quarantena
Rete interna
Regole firewall
Internet
Utenti
Policy specifiche
WAF
Validazione e
sanitizzazione
DMZ
Applicazione di e-commerce
FLUSSO APPLICAZIONE – RETE INTERNA
FLUSSO ATTACCANTE – APPLICAZIONE E-COMMERCE
FLUSSO UTENTE – APPLICAZIONE E-COMMERCE