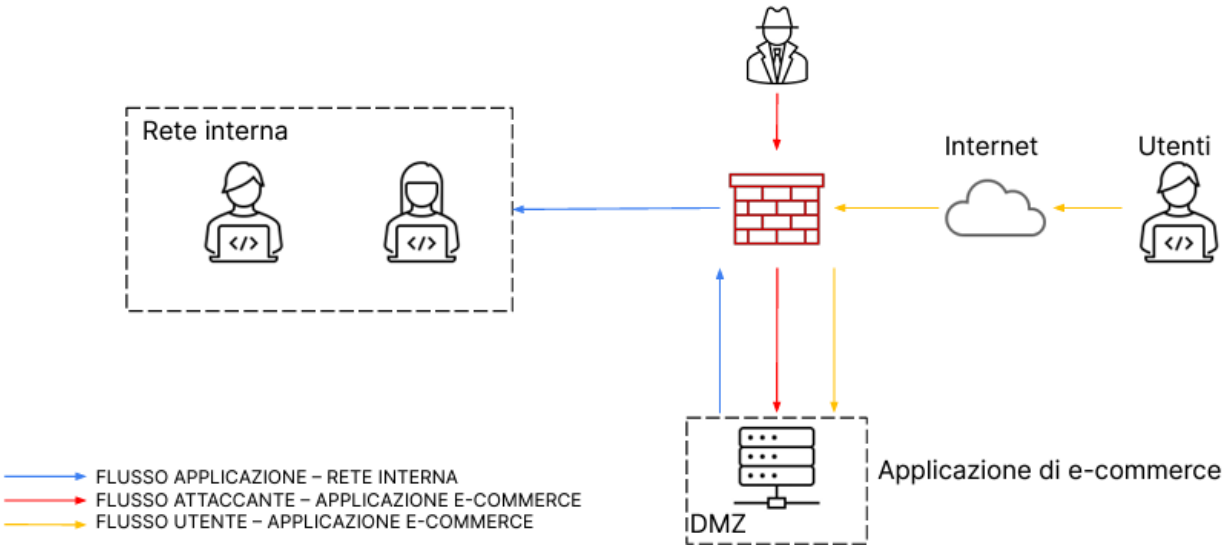
Analisi dei log: caso reale

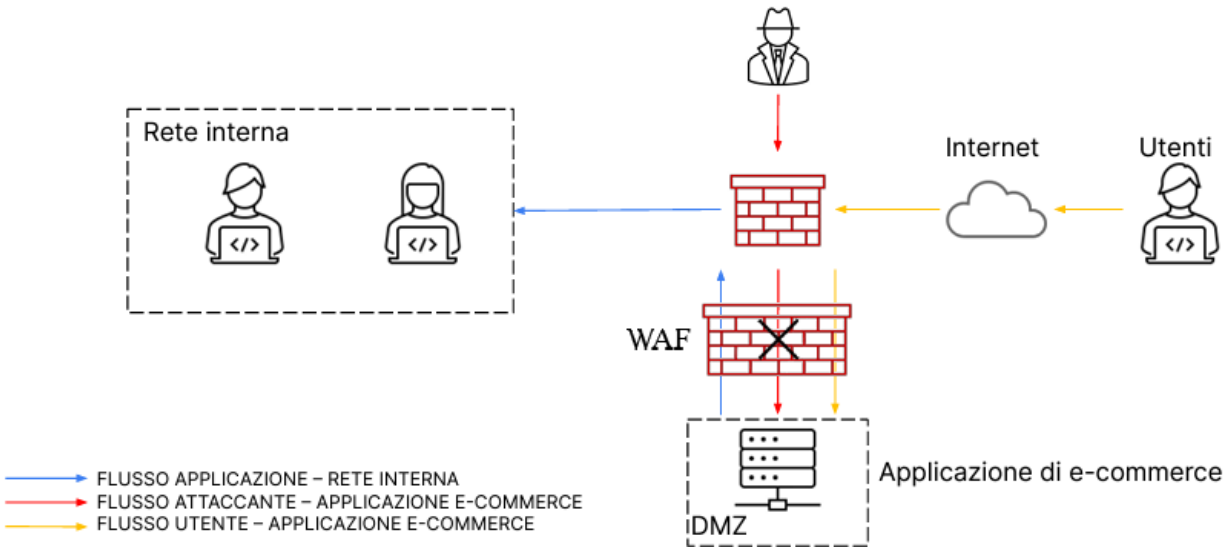
Nell'esercizio di oggi vedremo un esempio di un caso reale, in cui un attaccante ha manomesso il server **DMZ** di un'azienda di e-commerce. In particolare, vedremo come mitigare e prevenire eventuali attacchi **SQLi** e **XSS**, oltre a modificare l'architettura di rete per bloccare l'accesso alla rete interna della compagnia da parte del black hat. Analizzeremo inoltre la perdita monetaria della compagnia in caso di attacco **DDoS**, suggerendo anche in questo caso delle possibili soluzioni per prevenire il problema.

Innanzitutto, esaminiamo i metodi di prevenzione per gli attacchi SQLi e XSS. La seguente immagine mostra la situazione di partenza della rete presa in esame.



Prevenzione SQLi e XSS.

Entrambe queste modalità di attacco si basano sulla presenza di **vulnerabilità** a livello dell'**input utente** richiesto da un web server, permettendo ad un attaccante di eseguire del codice malevolo. A livello di architettura di rete, possiamo pensare di inserire un **WAF** prima della DMZ per identificare e bloccare tentativi di SQL injection e XSS, come mostrato nell'immagine seguente.



Oltre alla presenza del WAF, si dovrebbero considerare ulteriori soluzioni per mitigare ulteriormente la vulnerabilità in questione, implementando ad esempio:

- Sistemi di **sanificazione** dell'**input utente**, i quali andrebbero a "regolare" le stringhe immesse rimuovendo istruzioni e codici malevoli;
- **Librerie di sicurezza**, implementate a livello del codice sorgente ed in grado di mitigare entrambi i rischi;
- **Limitare l'accesso a directories contenenti dati sensibili**, implementando un sistema di credenziali per gli utenti di root o con accesso a tali dati.
- Introduzione di ulteriori sistemi di **monitoraggio del traffico**, come **IDS** e **IPS**, i quali potrebbero essere impostati per segnalare, ed eventualmente bloccare, pacchetti contenenti codice malevolo dove non dovrebbe essere presente alcun tipo di codice;
- **Aggiornamenti** continui dei sistemi, in grado di risolvere eventuali vulnerabilità presenti e, laddove una di queste dovesse essere già stata sfruttata, di sovrascrivere il codice malevolo presente sul server, come nel caso di **XSS stored**.

In ogni caso, è consigliato effettuare regolarmente procedure di **Pen-Testing** per scovare e risolvere eventuali punti di vulnerabilità non precedentemente conosciuti, andando a risolverli prima che questi diventino un problema.

DDoS e Impatto sul Business

Supponiamo che la nostra compagnia abbia subito un attacco di tipo **Distributed Denial of Service**, avente come effetto l'irraggiungibilità del web server per **10 minuti**. Considerando che l'e-commerce incassa **1500 euro al minuto**, la perdita di guadagno complessiva sarebbe di **15000 euro**.

Inoltre, non dobbiamo dimenticare il danno "intangibile" che la compagnia subirebbe: un attacco andato a buon fine potrebbe minarne l'immagine e la reputazione, andando quindi a compromettere i profitti futuri.

Tra le misure di prevenzione e mitigazione, suggeriamo:

- Configurazione di un **WAF** in grado di **riconoscere** e **mitigare** gli attacchi **DDoS**, oltre all'eventuale installazione e configurazione di **IDP** e **IPS** per aumentare la probabilità di riconoscere tale attacco non appena questo si verifichi mitigandone gli effetti da subito;
- Impostare dei **limiti alla banda** disponibile per ciascun utente, ad esempio bloccando un numero eccessivo di richieste (indicatore di un attacco DDoS) proveniente da uno stesso indirizzo IP, andando poi eventualmente ad escludere tale IP;
- Implementare **CDNs (Content Delivery Networks)** e **multipli Data Center** per distribuire il carico del traffico tra diversi server, diminuendo la possibilità di interruzione del servizio.

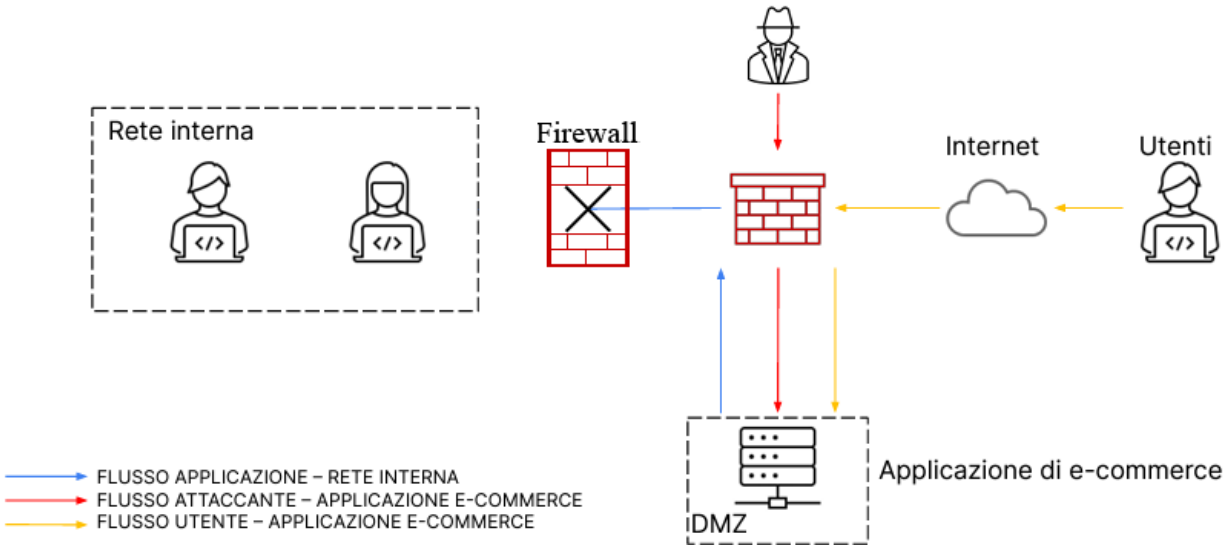
Come per ogni tipo di attacco informatico, è importante avere delle salde **policy** e piani d'azione per dettare e guidare le operazioni in caso di attacco, specificando quali misure adottare per prevenire, rispondere e gestire una eventuale minaccia.

Protezione della Rete Interna della Compagnia

Supponiamo che la web application venga infettata da un **malware**; la nostra priorità è quella di prendere misure per **proteggere la rete interna** della compagnia senza però andare ad interrompere in sevizio agli utenti. La priorità al momento non è quindi quella di escludere l'attaccante dalla DMZ violata ma di proteggere il resto della nostra rete, anche se questo potrebbe andare a mettere gli altri utenti a contatto con il malware: ciò avviene perché, probabilmente, si è valutato che le perdite monetarie e di immagine sarebbero

minori se gli utenti rischiassero di prendere il malware piuttosto che chiudendo completamente il servizio di e-commerce legato alla DMZ.

Per evitare che il malware si propaghi alla rete interna, lasciando però l'accesso alla DMZ, procediamo **separando la nostra rete dalla DMZ**. Per far ciò, possiamo implementare un **firewall tra la DMZ e la rete della compagnia**, impostato in modo tale da **bloccare qualsiasi connessione**, come proposto nell'immagine seguente. In questo modo, né l'attaccante né il malware avranno modo di raggiungere la rete interna; inoltre, una volta risolto il problema ed eliminata la minaccia, potremmo ripristinare la comunicazione semplicemente cambiando le regole del firewall che abbiamo aggiunto senza effettuare ulteriori modifiche alla rete.



Una volta eliminata la minaccia, dovremmo andare ad eseguire una profonda **analisi post-attacco** per capire come questa si sia potuta verificare in primo luogo, andando quindi a sanare le eventuali vulnerabilità trovate. Dovremmo poi ripristinare la DMZ eliminando il malware utilizzando, ad esempio, il backup più recente che non presenti la minaccia.

Infine, per verificare l'effettiva sicurezza del sistema, dovremmo ricorrere ad un approfondito **Pen-Test**, confermando la risoluzione della problematica.

Giuseppe Pariota