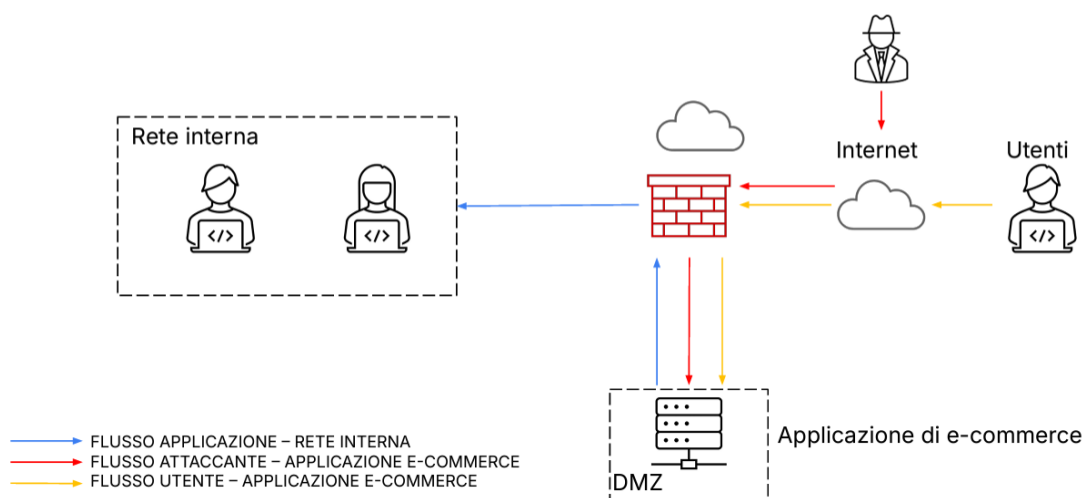


## Miglioramento della sicurezza di un'applicazione di e-commerce: PREVENZIONE, IMPATTO E RISPOSTA AGLI ATTACCHI

In questo esercizio, mi concentrerò sulla protezione e miglioramento dell'architettura di sicurezza di un'applicazione di e-commerce. L'obiettivo è analizzare le potenziali vulnerabilità e implementare misure preventive e reattive per proteggere l'applicazione dagli attacchi comuni, come l'SQL Injection (SQLi), il Cross-Site Scripting (XSS), e i Distributed Denial of Service (DDoS).



Il punto di partenza è questa architettura di rete che rappresenta un'infrastruttura di e-commerce in cui l'applicazione è ospitata nella DMZ, separata dalla rete interna e protetta da un firewall. Il diagramma evidenzia i principali flussi di traffico: il flusso di comunicazione sicura tra l'applicazione e la rete interna, il traffico tra gli utenti 'legittimi' e l'applicazione, e un potenziale flusso di attacco da parte di un attore malevolo. Il firewall gioca un ruolo cruciale nel filtrare il traffico proveniente da Internet, prevenendo accessi non autorizzati e proteggendo sia l'applicazione e-commerce che la rete interna da potenziali minacce.

Partendo da questa configurazione, apporterò delle modifiche per rafforzare la sicurezza del sistema. Mi concentrerò, quindi, su:

- ✓ Azioni preventive contro SQLi e XSS, con l'aggiunta di strumenti e pratiche di protezione specifiche;
- ✓ Calcolo dell'impatto sul business in caso di un attacco DDoS, e misure per mitigare tali attacchi;
- ✓ Strategie di risposta in caso di infezione da malware, per evitare che si propaghi all'interno della rete interna;
- ✓ Soluzione completa, combinando prevenzione e risposta per creare un'architettura di rete più resiliente.

L'esercizio avrà come fine ultimo quello di comprendere meglio come difendere un'applicazione web in uno scenario realistico, mettendo alla prova le competenze su firewall, load balancing, WAF, e mitigazione DDoS. Al termine, l'architettura iniziale risulterà potenziata con miglioramenti specifici che renderanno l'applicazione più sicura e resistente agli attacchi.

### **1. Azioni preventive: protezione contro SQL Injection (SQLi) e Cross-Site Scripting (XSS)**

Il punto 1 mira a rafforzare la sicurezza dell'applicazione web utilizzando misure preventive che impediscono l'inserimento e l'esecuzione di codice malevolo, sia a livello del server che del client.

Queste strategie, combinate, riducono significativamente il rischio di attacchi di SQLi e di XSS, garantendo che l'applicazione possa gestire in modo sicuro gli input ricevuti dagli utenti.

**Per prevenire attacchi di SQL Injection (SQLi) e di Cross-Site Scripting (XSS)** sull'applicazione di e-commerce, adotterò diverse misure di sicurezza fondamentali.

In primo luogo, parto con **l'implementare un Web Application Firewall (WAF)**, che rappresenta uno dei metodi più efficaci per proteggere l'applicazione e sicuramente il primo che si prende in considerazione. Il WAF viene posizionato tra il firewall di rete e la DMZ e andrà ad agire da filtro, che analizza e blocca le richieste in base a regole predefinite. Esso è

particolarmente utile contro l'SQLi e il XSS, in quanto riconosce e blocca i pattern di attacco noti prima ancora che raggiungano l'applicazione.

In generale, **Il WAF non solo aggiunge un livello di sicurezza aggiuntivo, ma permette anche di configurare e aggiornare facilmente le sue regole per rispondere alle eventuali minacce future**, rappresentando così la soluzione preventiva più flessibile e allo stesso modo potente.

Un'altra misura cruciale che si potrebbe prendere in considerazione è **la sanitizzazione e validazione dell'input utente**. Questo processo consiste nel filtrare i dati forniti dai vari e numerosi utenti per rimuovere caratteri speciali o elementi pericolosi. "Ripulire" gli input evita che contenuti sospetti, come tag HTML o script JavaScript, possano infiltrarsi nell'applicazione. In questo modo, si previene una delle cause principali degli attacchi di XSS, in quanto questa misura va ad impedire che il codice malevolo sia riconosciuto e interpretato dal browser. Particolare accorgimento è quello di assicurarci **una validazione rigorosa e completa, in primis verificando che ogni input rispetti il formato atteso, non solo sui campi visibili agli utenti ma anche sui parametri URL e sui campi nascosti**.

Un altro metodo complementare e spesso trascurato è **l'uso delle query parametrizzate** o, meglio, delle prepared statements per tutte le interazioni con il database. **Le query parametrizzate impediscono agli attaccanti di inserire un codice SQL potenzialmente malevolo** perché separano i dati dagli input. Ciò rende praticamente impossibile per l'attaccante alterare la logica delle query, proteggendo così il sistema dai tentativi di SQL Injection. Questo metodo di protezione risulta essenziale soprattutto in quelle applicazioni in cui gli utenti inseriscono informazioni in modo frequente, poiché riduce notevolmente il rischio di compromissione.

Per ultimo, un'altra importante strategia è **l'implementazione di una Content Security Policy (CSP)**, che permette di selezionare quali fonti esterne l'applicazione può caricare. La CSP è particolarmente efficace contro il XSS, poiché va a bloccare l'esecuzione di script provenienti da

fonti non autorizzate. **Impostando delle regole che limitano le fonti dei contenuti, si riduce la superficie d'attacco, poiché anche nel caso in cui un attaccante dovesse riuscire a inserire codice malevolo, questo non verrebbe effettivamente eseguito.**

Oltre a queste misure principali, ci sono altri accorgimenti che, pur essendo meno diretti rispetto ai precedenti metodi, possono contribuire alla protezione contro l'SQLi e il XSS. Un primo esempio riguarda **l'uso di codifiche speciali per gli input** (come HTML entities), che riduce il rischio che il browser interpreti elementi pericolosi. Allo stesso modo, **l'abilitazione delle secure cookies e del protocollo HTTPS** protegge i dati trasmessi, prevenendo molteplici attacchi tipo Man-in-the-Middle (MITM) che potrebbero facilitare il XSS.

- ✚ Ognuna di queste misure che ho descritto contribuisce a limitare la possibilità di attacchi malevoli e garantisce un livello di sicurezza elevato. Sicuramente, adottare queste soluzioni in maniera combinata tra loro assicura che l'applicazione possa gestire in modo sicuro ed efficace gli input degli utenti, preservando la sicurezza e l'integrità del sistema.

## **2.Valutazione dell'impatto di un attacco DDoS**

In questo secondo punto, mi concentrerò **sull'impatto economico di un attacco Distributed Denial of Service (DDoS)** sull'applicazione e-commerce e sulle possibili soluzioni per mitigare questo tipo di attacco. Un attacco di tipo DDoS è particolarmente pericoloso perché può rendere un servizio completamente inaccessibile, con gravi conseguenze economiche e reputazionali. In primo luogo, calcolare l'impatto economico risulta essenziale per capire quanto è urgente adottare misure preventive e valutare il ritorno sull'investimento di tali soluzioni.

## Calcolo dell'impatto economico

L'immaginario è che l'applicazione subisce un attacco DDoS che la rende inaccessibile per 10 minuti. Se il ricavo medio per minuto è di 1.500 €, l'impatto finanziario diretto sarà calcolato moltiplicando il guadagno per minuto per la durata dell'interruzione.

Con questi dati a disposizione, l'impatto economico sull'azienda di e-commerce sarà calcolato in questa maniera:  $1.500 \text{ €} \times 10 \text{ minuti} = 15.000 \text{ €}$

**Questa cifra rappresenta la perdita diretta** derivante dalla mancata possibilità per i clienti di completare transazioni. Tuttavia, è importante considerare **anche una serie di impatti indiretti**: l'inaccessibilità, anche momentanea, può ridurre la fiducia dei clienti, portandoli ad allontanarsi dalla piattaforma e indirizzandoli verso la concorrenza, con potenziali perdite future. In questo scenario, l'impatto economico potrebbe essere molto più elevato di quello immediato che è stato calcolato.

## Misure di mitigazione da adottare

Per evitare che un attacco DDoS causi danni così significativi, è cruciale implementare delle **soluzioni preventive**. La prima azione da intraprendere, che forse risulta anche la più importante, è

**l'implementazione di un servizio di mitigazione DDoS**. Questi servizi, come ad esempio **Cloudflare, AWS Shield o Akamai**, possono rilevare e filtrare il traffico anomalo prima che raggiunga l'infrastruttura dell'applicazione. **I servizi di mitigazione DDoS agiscono, in buona sostanza, come una barriera che protegge il sito dal sovraccarico**, garantendo che il traffico legittimo possa comunque accedere in tranquillità alla piattaforma.


Un'altra opzione è **l'utilizzo di un Load Balancer**. Questo strumento va a distribuire il carico di lavoro tra server diversi, riducendo di molto le probabilità che un singolo server venga sovraccaricato di richieste. Il Load Balancer non solo migliora e aumenta la capacità di gestione del traffico legittimo, ma aiuta anche a proteggere lo stesso server contro i DDoS, poiché diluisce il traffico su più nodi. **L'adozione di un Load Balancer può**

**ridurre significativamente l'impatto di un attacco, mantenendo la disponibilità del servizio anche in presenza di carichi di traffico anomali.**

Un'altra misura preventiva importante è **il rate limiting**, che consente di limitare il numero di richieste che un singolo IP può inviare in un determinato intervallo di tempo (bot). In questo modo, anche se un attaccante cerca di inviare una quantità spropositata di traffico, il sistema limita automaticamente il numero di richieste provenienti da uno stesso IP. **Questa misura risulta particolarmente efficace nel fermare attacchi DDoS di bassa intensità o provenienti da un numero limitato di fonti.**

Soluzioni più avanzate sono data **dall'utilizzo di reti di Content Delivery Network (CDN)**. Una CDN distribuisce il contenuto dell'applicazione su più server sparsi geograficamente. In caso di attacco, la CDN può assorbire il carico distribuendolo sui vari nodi, riducendo così il rischio di sovraccarico. **L'adozione di una CDN potrebbe essere utile non solo per migliorare la sicurezza, ma anche per aumentare le prestazioni dell'applicazione, riducendo i tempi di caricamento e migliorando l'esperienza utente.**

Infine, è possibile adottare una serie di misure proattive come **il monitoraggio continuo del traffico e l'utilizzo di sistemi di allerta automatica**. Questi strumenti consentono di rilevare immediatamente anomalie nel traffico e di reagire rapidamente in caso di attacco. **Il monitoraggio costante permette di rispondere prontamente, riducendo la durata dell'interruzione e, quindi, l'impatto economico complessivo.**

 L'adozione di queste misure preventive offre una protezione efficace e riduce il rischio di interruzioni del servizio, mantenendo al sicuro il flusso di entrate e preservando la reputazione dell'azienda.

### **3. Risposta a un'infezione da malware**

Nel caso in cui l'applicazione web venga infettata da un malware, la priorità principale risulta impedire che quest'ultimo si propaghi nella rete interna. In questa situazione, l'obiettivo principale non sarà rimuovere l'accesso dell'attaccante, ma **contenere il malware** e limitarne gli effetti solo alla macchina compromessa. Le strategie per raggiungere questo scopo sono molteplici, e ciascuna presenta un livello di protezione specifico.

La misura più veloce e immediata, e probabilmente tra le più efficaci, è **l'isolamento del server infetto**. Questa tecnica consiste nel configurare il firewall per bloccare qualsiasi traffico in uscita dal server compromesso verso la rete interna. **In questo modo, il malware rimane confinato nella DMZ e non può raggiungere i sistemi critici della rete interna**, prevenendo la diffusione della minaccia. L'isolamento può essere ottenuto attraverso delle specifiche regole settate nel firewall, che hanno come scopo l'impedimento delle comunicazioni in uscita o la limitazione del traffico solo per specifici indirizzi IP esterni alla rete ma autorizzati.

In parallelo, è utile attuare **un monitoraggio costante del traffico** proveniente dal server infetto. Questo permette di osservare le attività dell'attaccante senza intervenire direttamente e, allo stesso tempo, raccogliere informazioni preziose sul modo di agire del malware.

**Monitorando il traffico in tempo reale, è possibile identificare le modalità di comunicazione del malware e, in un eventuale scenario futuro, anticipare nuovi tentativi di connessione con altri sistemi della rete.** Questo approccio consente di mantenere l'attaccante "sotto osservazione", raccogliendo dati utili per analisi future.

Un'alternativa più elaborata che si può eseguire è **la segmentazione della rete**, dove la rete è suddivisa in molteplici sottoreti isolate tra loro. In questo scenario, il server infetto rimarrebbe all'interno di una determinata sottorete separata da tutte le altre, che comunica con la rete interna solo attraverso riscontri stringenti e monitorati. In buona sostanza, **la segmentazione riduce il rischio che una minaccia si propaghi oltre il**

**segmento in cui è stata rilevata**, aumentando notevolmente la sicurezza complessiva del sistema. Sebbene la segmentazione risulti una misura più complessa da implementare, è particolarmente efficace nelle reti che ospitano servizi critici e ad alto rischio.

Un'altra opzione è **l'utilizzo di honeypot** o di sistemi di trappole per ingannare l'attaccante. Configurando eventualmente un honeypot nella DMZ, l'attaccante può essere indotto a esplorare un sistema falsificato che però non contiene dati reali, ma registra ogni azione che viene compiuta.

**Questa tecnica non solo evita che il malware raggiunga la rete interna, ma consente anche di raccogliere informazioni dettagliate sulle tecniche dell'attaccante**, che possono essere utili sempre per migliorare la sicurezza in futuro.

Per aumentare la protezione, si potrebbe anche abilitare **un sistema di rilevamento delle intrusioni (IDS) e di prevenzione delle intrusioni (IPS)** sul traffico di rete nella DMZ. Un IDS analizza il traffico in cerca di comportamenti anomali o potenzialmente pericolosi e invia un'allerta se rileva attività sospette. Un IPS, invece, blocca automaticamente il traffico sospetto prima che riesca a raggiungere gli altri sistemi. **Questi sistemi, se usati in combinazione, possono rilevare tentativi di propagazione del malware e bloccarli in tempo reale**, proteggendo così la rete interna.

Infine, per avere una protezione più "aggressiva", si potrebbero disattivare temporaneamente tutti i servizi non essenziali sulla macchina infetta, riducendo il numero di vie di accesso per l'attaccante e limitando di conseguenza anche le possibilità di propagazione. **Spegnendo i servizi non necessari, si riduce la superficie di attacco del malware, rendendo più difficile per l'attaccante espandere la propria presenza all'interno del sistema.**

- ✚ Queste misure combinate garantiscono che il malware rimanga confinato, limitando i danni e permettendo di raccogliere informazioni utili senza compromettere la sicurezza della rete interna.



#### 4. Creazione di una soluzione completa

Nel punto 4, l'obiettivo è creare una soluzione completa che vada a combinare le misure preventive precedentemente descritte per proteggere l'applicazione web da attacchi di SQL Injection (SQLi) e di Cross-Site Scripting (XSS), insieme alle strategie di risposta in caso di infezione da malware. L'integrazione di queste due soluzioni permetterà di rafforzare la sicurezza complessiva dell'architettura e garantire che sia pronta non solo a prevenire gli attacchi, ma anche a gestire eventuali compromissioni senza che queste si propaghino nella rete interna.

Per prima cosa, dobbiamo integrare le azioni preventive menzionate nel **punto 1**. Il **Web Application Firewall (WAF)** deve essere posizionato tra il firewall principale e la DMZ. Il WAF filtrerà tutto il traffico in ingresso, bloccando eventuali attacchi prima che raggiungano l'applicazione. **Il WAF è il primo livello di difesa, che impedisce che input malevoli possano compromettere il sistema.** Oltre al WAF, all'interno dell'applicazione web, **implementeremo meccanismi di sanitizzazione e validazione dell'input**, garantendo che ogni dato fornito dagli utenti sia filtrato e validato per evitare vulnerabilità.

In aggiunta, assicureremo che tutte le query SQL eseguite dall'applicazione siano parametrizzate, separando i dati dalla logica delle query. Questo riduce il rischio di SQL Injection e rende l'applicazione più resistente agli attacchi che mirano a manipolare il database. Inoltre, **configurando una Content Security Policy (CSP)**, limiteremo le fonti da cui l'applicazione può caricare contenuti, prevenendo così l'esecuzione di codice malevolo tramite attacchi XSS.

A questo livello, l'architettura è stata resa sufficientemente robusta contro SQLi e XSS, ma nel caso in cui l'applicazione venga compromessa da un malware, dobbiamo implementare la strategia di risposta descritta nel **punto 3**. Il server infetto nella DMZ sarà isolato utilizzando regole avanzate nel firewall. Queste regole bloccheranno ogni tentativo di comunicazione tra la DMZ e la rete interna, assicurando che il malware non possa propagarsi ulteriormente. **L'isolamento è una delle strategie più**

**importanti per contenere una minaccia**, poiché limita il raggio d'azione del malware e impedisce che possa causare danni alla rete interna.

Allo stesso tempo, manterremo attivo **il monitoraggio del traffico** proveniente dal server compromesso. Questo ci permetterà di osservare le attività dell'attaccante e del malware, raccogliendo dati preziosi che potranno essere utilizzati per l'analisi post-incidente e per migliorare ulteriormente la sicurezza. Il monitoraggio attivo ci aiuterà a individuare eventuali schemi di comportamento sospetto e a reagire tempestivamente nel caso in cui il malware tenti di connettersi a sistemi esterni.

Per rendere la soluzione completa e ancora più efficace, è importante considerare **l'adozione di sistemi di rilevamento e prevenzione delle intrusioni (IDS/IPS)**. Questi strumenti, come già detto in precedenza, analizzano costantemente il traffico e possono rilevare attività anomale o potenzialmente pericolose, bloccando il traffico in tempo reale. **L'IDS/IPS agisce come una difesa reattiva**, intercettando e prevenendo la diffusione di minacce anche nel caso in cui il malware tenti di comunicare con l'esterno o di muoversi lateralmente nella rete.

Infine, la rete deve essere strutturata in modo da ridurre il rischio di compromissione attraverso **la segmentazione della rete**. Creando sottoreti isolate e limitando la comunicazione tra di esse, si riduce la probabilità che un'infezione possa diffondersi al di là del segmento compromesso. Questa segmentazione non solo limita i danni, ma permette anche di isolare meglio le diverse componenti critiche dell'infrastruttura.

## **5. Modifica "PIÙ AGGRESSIVA" dell'infrastruttura**

Nel punto 5, si parla di una possibile **modifica più aggressiva dell'infrastruttura**, che, sebbene sia data come facoltativa, può rivelarsi utile per rafforzare ulteriormente la sicurezza della rete e migliorare la sua capacità di risposta in caso di attacchi di vario genere. Questo tipo di modifica mira a rendere l'infrastruttura non solo più resiliente agli attacchi

di tipo DDoS, ma anche capace di gestire scenari di attacco potenzialmente più complessi, andando a migliorare al contempo l'efficienza complessiva del sistema.

Una delle modifiche più efficaci e aggressive che si possono apportare è una segmentazione **avanzata della rete**. In questa configurazione, la rete interna, la DMZ, e tutti i servizi critici vengono suddivisi in una serie di sottoreti isolate, con un protocollo di controllo molto rigido sulle comunicazioni tra le diverse sezioni. **La segmentazione avanzata riduce notevolmente il rischio che un attacco, una volta infiltrato in una parte del sistema, possa propagarsi ad altre aree sensibili.** L'idea generale è di limitare al massimo i punti di accesso tra i segmenti della rete e applicare regole molto restrittive che impediscano comunicazioni non autorizzate tra le sottoreti.

In questo contesto, ogni segmento di rete può aderire a specifici livelli di autorizzazione e policy di sicurezza più stringenti. Per esempio, la rete interna che ospita dati sensibili come quelli dei clienti o dei dipendenti può essere isolata in modo che solo determinati dispositivi e servizi possano interagirvi, allo stesso tempo la DMZ continuerà a gestire le interazioni con il traffico esterno. Questo riduce di molto il rischio che un attacco DDoS o una compromissione di un server posto nella DMZ si diffonda all'interno della rete critica.

Un'altra opzione da prendere in considerazione è l'implementazione di **regole firewall avanzate**, da impostare in combinazione con la segmentazione della rete. In questo caso, il firewall adotta procedure specifiche non solo per bloccare il traffico sospetto, ma anche per analizzarlo e filtrarlo in modo intelligente in base a criteri come l'origine, la destinazione, il contenuto e il comportamento. **L'uso di regole firewall "flessibili e intelligenti" permette al sistema di rispondere in tempo reale a comportamenti che possono essere anomali**, bloccando automaticamente il traffico che potrebbe rappresentare una minaccia. Tutto questo si va ad aggiungere e integrare alle tecniche di mitigazione

DDoS descritte nel punto 2, ma rende il firewall generalmente molto più attivo nella gestione delle minacce.

Un'altra modifica aggressiva potrebbe includere **l'implementazione di un sistema di autenticazione più robusto** per l'accesso alla rete e alle risorse critiche. L'adozione di **una autenticazione multifattoriale (MFA)** su ogni livello dell'infrastruttura considerato critico può migliorare notevolmente la sicurezza, richiedendo agli utenti di fornire una combinazione di credenziali per accedere ai sistemi. **L'MFA riduce il rischio di compromissione delle credenziali e garantisce che, anche se un attaccante riesce a ottenere una password, non sarà comunque in grado di accedere al sistema senza passare ulteriori livelli di verifica.** Questa soluzione potrebbe essere estesa anche per limitare l'accesso ai servizi all'interno della rete, applicando un controllo rigoroso sia sugli utenti interni sia su quelli esterni.

Per una protezione più aggressiva, potrebbe rivelarsi essenziale **dotare l'infrastruttura di sistemi di log e monitoraggio avanzato** come i SIEM (Security Information and Event Management). Questo tipo di sistemi raccolgono e analizzano in tempo reale tutti i dati provenienti da tutte le componenti della rete, dai firewall ai server, agli endpoint. **Attraverso l'uso di SIEM, è possibile rilevare in tempo reale comportamenti sospetti, anomalie nel traffico e tentativi di accesso non autorizzati,** permettendo una risposta immediata agli incidenti. I SIEM possono anche essere impostati per inviare alert automatici agli amministratori di sistema in caso di minacce, riducendo di molto il tempo di risposta e contenendo potenziali danni.

Un'altra modifica “aggressiva” potrebbe includere **l'adozione di un'infrastruttura cloud o ibrida**, dove alcune parti dell'infrastruttura sono ospitate su server cloud, riducendo così il carico sui server locali. Le architetture cloud offrono la possibilità di scalare facilmente in caso di attacco DDoS, distribuendo il traffico su una rete globale di server. **L'adozione di un'infrastruttura ibrida offre flessibilità, con la possibilità di beneficiare della scalabilità del cloud e della sicurezza dei server**

**locali**, assicurando che l'applicazione possa resistere anche a picchi di traffico anomali.

Infine, un'altra strategia potrebbe essere **un uso di honeypot più complessi**. Come già detto, gli honeypot sono sistemi falsi progettati per attirare gli attaccanti e monitorare i loro movimenti senza esporre risorse critiche. **L'implementazione di honeypots può deviare gli attaccanti dai veri server, raccogliendo al contempo informazioni preziose sui loro metodi e strumenti**. In questo modo, si aumenta la resilienza del sistema e si migliorano le capacità di rilevamento e risposta.

- ✚ Implementare queste misure rende l'infrastruttura molto più robusta e in grado di gestire attacchi sofisticati o persistenti come i DDoS o le infezioni da malware, assicurando che la rete sia sempre pronta a difendersi da minacce future.

## Diagramma finale con migliorie apportate.

