W20D4 - Progetto

Security Operation

Implementazione misure di prevenzione e response sull’infrastruttura di una app e-commerce

1. Azioni preventive SQLi & XSS - Metodologie

Attacchi SQLi

Una delle tecniche più efficaci per prevenire attacchi di tipo SQLi è l'utilizzo di **query parametrizzate** (o istruzioni preparate). Queste query consentono di separare il codice SQL dai dati, mitigando così il rischio di inserimenti malevoli. Ad esempio, la query potrebbe contenere un punto interrogativo (?) come segnaposto per i parametri, i quali vengono sostituiti con valori sicuri solo al momento dell'esecuzione della query mediante il comando execute. Questa pratica garantisce che qualsiasi input dell'utente venga trattato come dato e non come codice eseguibile.

In aggiunta, è fondamentale **sanificare l'input** degli utenti, prestando particolare attenzione a caratteri problematici come virgolette e parentesi, che potrebbero essere sfruttati per compromettere la sicurezza dell'applicazione. Strumenti di validazione e sanitizzazione, come regex o librerie dedicate, possono essere impiegati per garantire che i dati in ingresso rispettino i formati attesi.

Attacchi XSS

Per mitigare attacchi Cross-Site Scripting (XSS), è consigliata l'integrazione di un Content Security Policy (CSP). Questa politica permette di specificare le fonti attendibili da cui il browser può caricare contenuti e script. Sebbene la CSP non elimini completamente le vulnerabilità da un'applicazione web, contribuisce a rendere più difficoltoso per un attaccante sfruttarle.

1. Azioni preventive SQLi & XSS - WAF

Implementazione di un Web Application Firewall (WAF)

Integrare un **Web Application Firewall (WAF)** è una strategia cruciale per proteggere le applicazioni web da SQLi e altri tipi di attacco. Operando a livello 7 (Applicazione), il WAF funge da proxy inverso, filtrando e monitorando il traffico HTTP tra l'applicazione di e-commerce e internet. Il WAF monitora pattern per exploit potenziali di vulnerabilità SQL injection, XSS, CSRF, e attacchi DDoS ispezionando il livello applicativo delle richieste e risposte per intercettare e neutralizzare le minacce prima che possano raggiungere le applicazioni web.

Come misura preventiva per l'applicazione di e-commerce, si può implementare l'utilizzo di un **WAF con mitigazione anti-DDoS** e un **modello di sicurezza ibrido**, ossia operante sia su modello blacklist, che protegge contro attacchi noti, che modello allow list, che ammetterebbe solo traffico precedentemente approvato.

Data la loss expectancy in caso di attacco DDoS sull'app, si presume che l'e-commerce disponga di un budget piuttosto consistente. Inoltre, avendo l'esigenza di proteggere dati sensibili degli utenti, sarebbe opportuna l'implementazione di varie metodologie e tecnologie di sicurezza per proteggere la sicurezza dei dati e la reputazione della compagnia in caso di attacchi andati a buon fine.

1. Azioni preventive SQLi & XSS - WAF

A seconda delle esigenze di budget, si potrebbe optare per una WAF basata su **cloud** o **firewall-as-a-service (FWaaS)**, facile da installare, economicamente accessibile e che solitamente viene aggiornata costantemente dal vendor per proteggere dalle minacce più recenti. Inoltre, i FWaaS non richiedono spazio fisico e manutenzione delle apparecchiature, a differenza dei WAF basati su rete che sono invece implementati mediante hardware. I WAF possono anche bloccare traffico malevolo e prevenire infezioni da malware o attacchi da bad bot. Molti WAF recenti basati su cloud utilizzano tecnologie avanzate di machine learning per adattare le regole dei WAF ai metodi di attacco più recenti.

I **WAF basati su rete** sono implementati mediante hardware e generalmente più costosi, poiché richiedono manutenzione e molto spazio fisico. Hanno tuttavia il vantaggio di ridurre la latenza, essendo installati localmente.

Un **WAF basato su host** può essere invece integrato nel software di un'applicazione, offre maggiore personalizzazione ed è una soluzione meno costosa di un WAF basato su rete. Questa soluzione però presenta alcuni svantaggi, come il consumo di risorse del server locale, la complessità di implementazione e onerosi costi di manutenzione, poiché queste componenti richiedono solitamente l'impiego di risorse ingegneristiche.

Firewall interno

L’implementazione di un firewall interno tra la DMZ e la rete interna fornisce un extra layer di protezione con policy che limitino e monitorino il più possibile le richieste dalla web app che espone servizi sulla rete alla rete interna. Questo può aiutare a garantire che il traffico sia ulteriormente filtrato e a prevenire che eventuali minacce si propaghino verso la rete interna dalla DMZ.

1. Azioni preventive SQLi & XSS - CSP

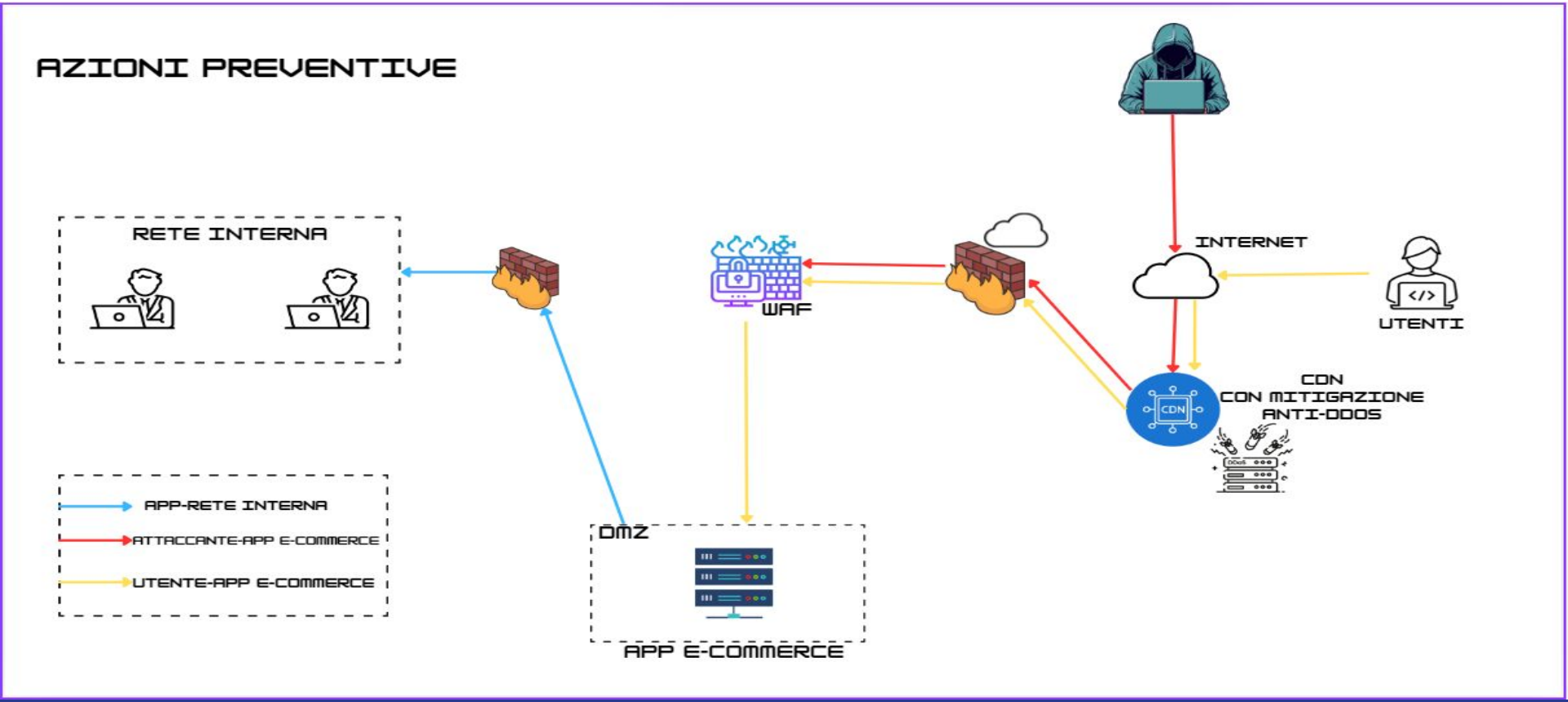
Content Security Policy (CSP)

Oltre a un WAF, si può aggiungere un extra layer di sicurezza con l'implementazione di **Content Security Policy (CSP)** per individuare e mitigare attacchi di tipo **XSS**. Questi attacchi sfruttano il fatto che i browser si fidano della fonte del contenuto, dunque una CSP può aiutare a prevenirli specificando le fonti che i browser dovrebbero considerare come attendibili per eseguire script e caricare contenuti quali JavaScript, CSS e immagini.

Sebbene una Content Security Policy non elimini una vulnerabilità da una applicazione web, può fare in modo da renderla difficile da sfruttare per un attaccante. Per implementare questa soluzione, il web server può aggiungere un header HTTP header chiamato Content-Security-Policy a ogni risposta. Ad esempio, si possono disabilitare gli script inline e abilitare solo script provenienti da origini sicure (HTTPS).

Inizialmente si potrebbe definire una whitelist per bloccare tutto ed eseguire la CSP in modalità report-only per fare in modo che il browser valuti le regole ma non blocchi ancora il contenuto. Questo permette di rivedere gli errori e verificare quali dovrebbero essere bloccati e quali no. Effettuando questa operazione per alcune settimane si dovrebbe riuscire ad ottenere una panoramica utile per settare delle regole efficaci, disabilitare la modalità report-only e iniziare a bloccare le risorse che non sono in whitelist.

1. Azioni preventive SQLi & XSS - fig. 1



2. Impatti sul Business

Considerando una perdita di 1.500 euro al minuto a causa di un attacco DDoS che impatta l'applicazione per 10 minuti, l'impatto totale sul business (Single Loss Expectancy) sarebbe di:

$$\textbf{SLE} = 1.500\ \text{euro/min} \times 10\ \text{min} = \textbf{15.000 euro}$$

2. Impatti sul Business - Azioni preventive

Azioni Preventive per Mitigare Attacchi DDoS

Implementazione di un WAF

Il WAF non solo protegge da attacchi SQLi e XSS, ma è anche fondamentale per la difesa contro gli attacchi DDoS. I WAF più avanzati utilizzano tecniche di analisi comportamentale per identificare anomalie nel traffico, come attività di botnet o pattern sospetti, attivando notifiche agli analisti di cyber security per affrontare rapidamente le minacce.

2. Impatti sul Business - Azioni preventive

Utilizzo di un Content Delivery Network (CDN)

I CDN possono alleggerire il carico diretto su un singolo server distribuendo il traffico su una rete globale di server e riducendo l'impatto sul business di attacchi DDoS e migliorando al contempo la velocità di risposta dell'applicazione web. I CDN possono anche mitigare l'impatto di attacchi DDoS sul layer 3 (rete).

Un CDN è un gruppo di server proxy distribuito geograficamente in data center nel mondo che memorizza nella propria cache i contenuti vicino agli utenti finali (un CDN è più vicino all'utente finale di quanto non lo sia il server di origine). Inoltre, consente la riduzione della latenza della connessione, ridurre il round-trip time (RTT) addizionale richiesto dalla connessione sicura su SSL/TLS e permette il trasferimento rapido delle risorse statiche necessarie per il caricamento dei contenuti internet tra cui pagine HTML, codice CSS, librerie JavaScript, immagini e video. Se configurato correttamente, un CDN può rendere un server di origine 'invisibile', fungendo da scudo per le richieste in entrata.

Un CDN con diffusione delle reti Anycast può aumentare la superficie di rete dell'organizzazione in modo che possa assorbire più facilmente i picchi di traffico volumetrici e prevenire eventuali interruzioni dovute, tra l'altro, ad attacchi di tipo DDoS, disperdendo il traffico tipicamente su molteplici data center vicini. Quando si ricevono richieste da un singolo indirizzo IP associato alla rete Anycast, queste vengono distribuite sulla base di una metodologia di prioritizzazione che solitamente viene ottimizzata per ridurre la latenza, selezionando il data center con la distanza minore dal client.

Nel caso di un attacco DDoS, dopo che altri tool di mitigazione hanno filtrato parte del traffico proveniente dall'attaccante, la rete Anycast distribuisce ciò che rimane del traffico malevolo tra molteplici data center, facendo in modo che nessuno di essi riceva più richieste di quante possa supportare. Una CDN con anycast aumenta la superficie della rete ricevente in modo che il flusso di rete di attacco non filtrato proveniente, ad esempio, da una botnet, venga assorbito da ognuno dei data center della CDN.

2. Impatti sul Business - Azioni preventive

Soluzioni di Rate Limiting

Per limitare il numero di richieste che un utente può inviare in un determinato periodo di tempo e prevenire attacchi brute force, attacchi DoS e DDoS e webscraping da bot malevoli, che possono sovraccaricare il server con molteplici richieste e impattare la disponibilità dell'applicazione web, si può adottare una soluzione di rate limiting che può prevenire attacchi DDoS sul livello 3 (rete). Questa soluzione viene implementata sull'applicazione stessa anziché sul server. Molti WAF e CDN offrono soluzioni di rate limiting integrate e utilizzano analisi comportamentale basata su metodologie di machine learning per determinare i limiti corretti da applicare.

Il **rate limiting** può ad esempio essere implementato sulla pagina di login con una soluzione ibrida che tenga in conto sia l'indirizzo IP da cui proviene la richiesta che lo username. Una soluzione ibrida sembra più efficace perché:

- se venisse applicata solo sulla base dell'indirizzo IP da cui proviene la richiesta, questo non proteggerebbe da attacchi in cui viene impiegato l'IP spoofing usando botnet.
- se applicata solo sulla base dello username che cerca di effettuare il login, un attaccante potrebbe lanciare attacchi brute force usando liste di username e password comuni e riuscirebbe ad effettuare il login per almeno alcune delle credenziali presenti nel database dell'applicazione web (anche utilizzando lo stesso indirizzo IP).

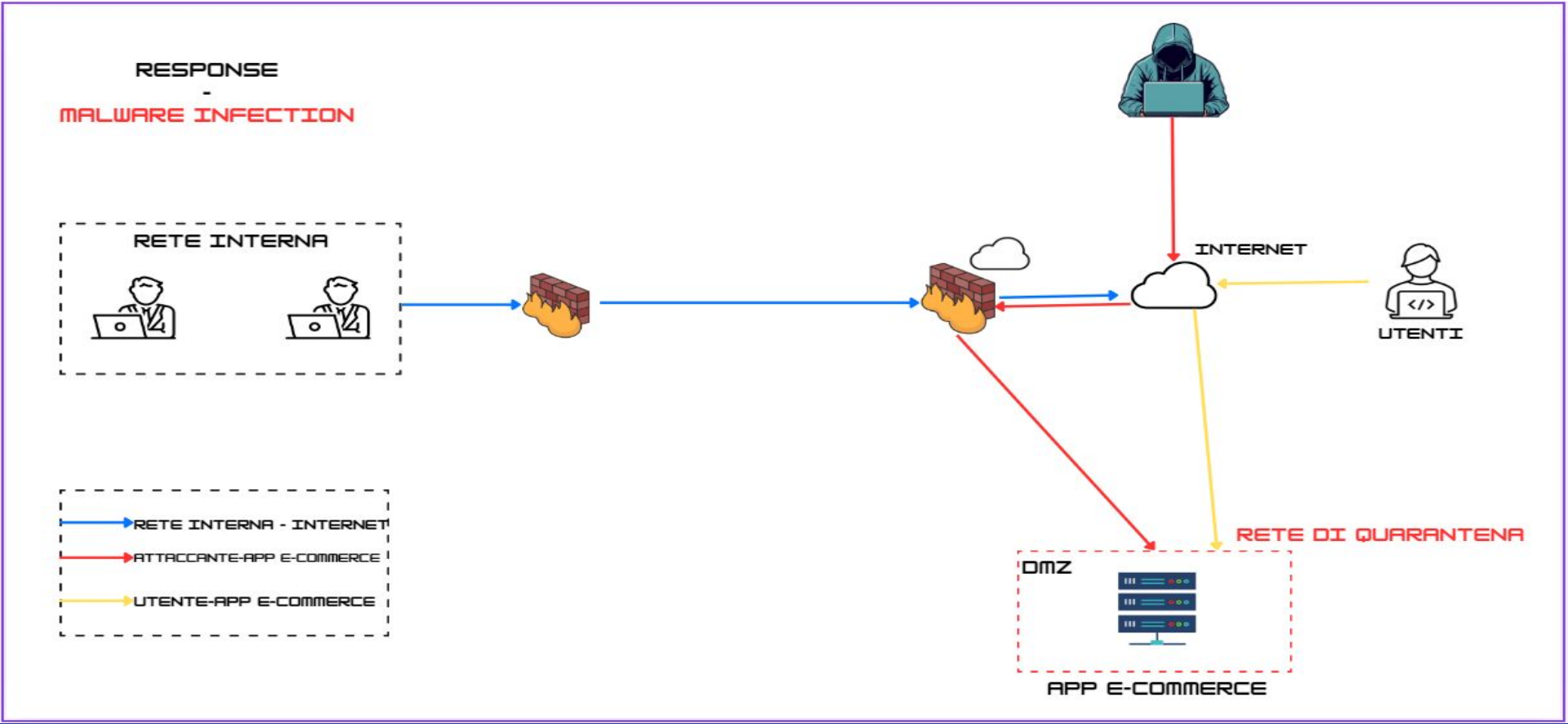
2. Impatti sul Business - Azioni preventive

Misure aggiuntive anti-DDoS

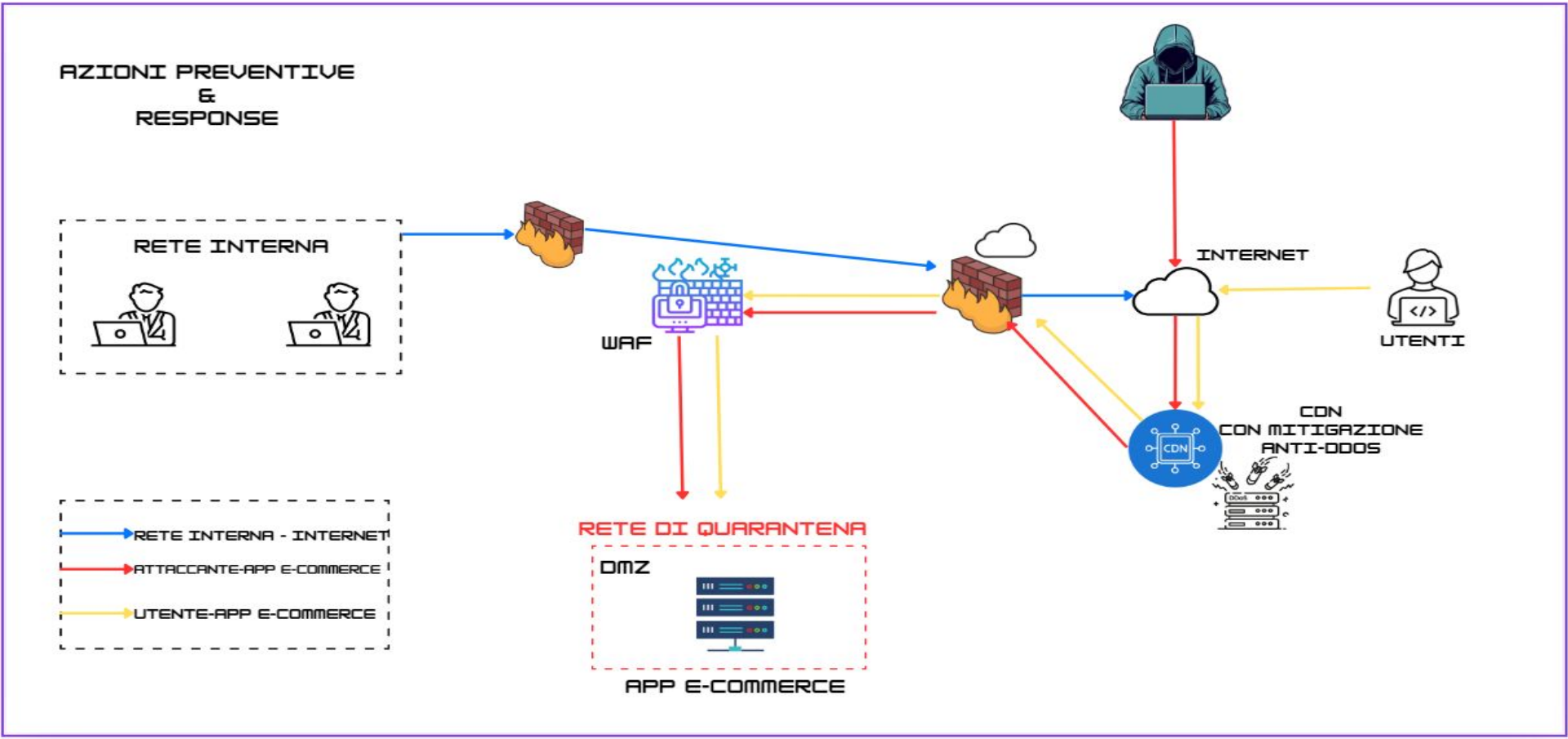
L'utilizzo di **load balancers** può aiutare a prevenire attacchi di tipo DDoS fungendo da scudo per i server e le risorse computazionali, evitando la loro diretta esposizione. Il load balancer è un dispositivo che si trova tra l'utente e il gruppo di server e funge da facilitatore invisibile, garantendo che tutti i server di risorse siano utilizzati allo stesso modo. Gli Application Load Balancer esaminano il contenuto della richiesta, ad esempio le intestazioni HTTP o gli ID di sessione SSL, rilevano eventuali problemi del server e reindirizzano il traffico client ed eventuale traffico malevolo a più server di back-end per ridurre al minimo l'impatto di un attacco quando un server diventa vulnerabile.

Possono aiutare a rimuovere i single points of failure, minimizzare la superficie di attacco e rendere più difficoltoso portare a termine attacchi finalizzati alla saturazione delle risorse. Sono inoltre in grado di effettuare operazioni di monitoraggio del traffico, bloccando eventuali contenuti dannosi e instradare il traffico attraverso un gruppo di firewall di rete per una maggiore sicurezza. Oltre a fornire un extra layer di sicurezza, aiutano a migliorare la disponibilità, scalabilità e le prestazioni di una applicazione web. Sono inoltre generalmente meno onerosi e richiedono meno manutenzione di metodologie di difese hardware.

3. Response - Infezione Malware - Isolamento - fig. 2



4. Soluzione completa - Response & Prevention - fig. 3



5. Modifica “aggressiva” infrastruttura

Intrusion Detection System (IDS)

Un **IDS** può essere implementato tra la DMZ e la rete interna per monitorare il traffico sospetto e bloccare i tentativi di attacco prima che raggiungano la rete interna. Questo dispositivo analizza il traffico in tempo reale e notifica agli amministratori eventuali potenziali minacce.

Potrebbe essere utile posizionare due IDS nella rete: uno tra il WAF e la DMZ per monitorare il traffico in entrata, e l’altro tra il firewall interno e la DMZ per rilevare comportamenti anomali provenienti dai server.

Security Information and Event Management (SIEM)

Implementare una soluzione **SIEM** consente il monitoraggio centralizzato dell'attività di rete, facilitando l'identificazione di anomalie e l'analisi forense. Il SIEM raccoglie e centralizza dati di log provenienti da vari componenti della rete, migliorando la risposta agli incidenti.

Network Access Control (NAC)

L’adozione di un sistema di **NAC** garantisce che solo dispositivi autorizzati possano accedere alla rete interna. Il NAC verifica l'identità degli utenti e la conformità dei dispositivi prima di concedere accesso a risorse critiche.

5. Modifica “aggressiva” infrastruttura

Segmentazione della Rete Interna

La creazione di sottoreti protette da firewall per compartimentare i servizi, come database e applicazioni critiche, offre un ulteriore strato di protezione, riducendo la superficie di attacco.

Firewall interno

Un firewall interno o un NGFW (a seconda del budget) è utile come extra layer di protezione per filtrare il traffico che entra nella rete interna e consentire solo comunicazioni legittime verso porte e protocolli specifici tra i web server nella DMZ e l’application server nella rete interna.

Protezione dei dischi

Per migliorare la resilienza dei sistemi ed eliminare per quanto possibile i SPOF, si può implementare una soluzione di protezione dei dischi tramite l’inserimento di dischi aggiuntivi secondo la configurazione RAID avendo almeno 2 dischi con gli stessi dati (RAID-1) o 3 o più dischi (RAID-5) per poter recuperare i dati sul disco non funzionante in caso di disastro tramite la parità.

5. Modifica “aggressiva” infrastruttura

Ridondanza server

Per migliorare la tolleranza agli errori dell’infrastruttura, si possono aggiungere elementi ridondanti all’interno dell’architettura come un failover cluster. Se il server attivo dell’applicazione smetterà di funzionare in caso di disastro, l’altro nodo del cluster verrà promosso a nodo attivo tramite un meccanismo automatico di failover. I nodi del cluster saranno connessi tramite una rete dedicata, mentre l’altra rete connette i cluster di failover e i clients.

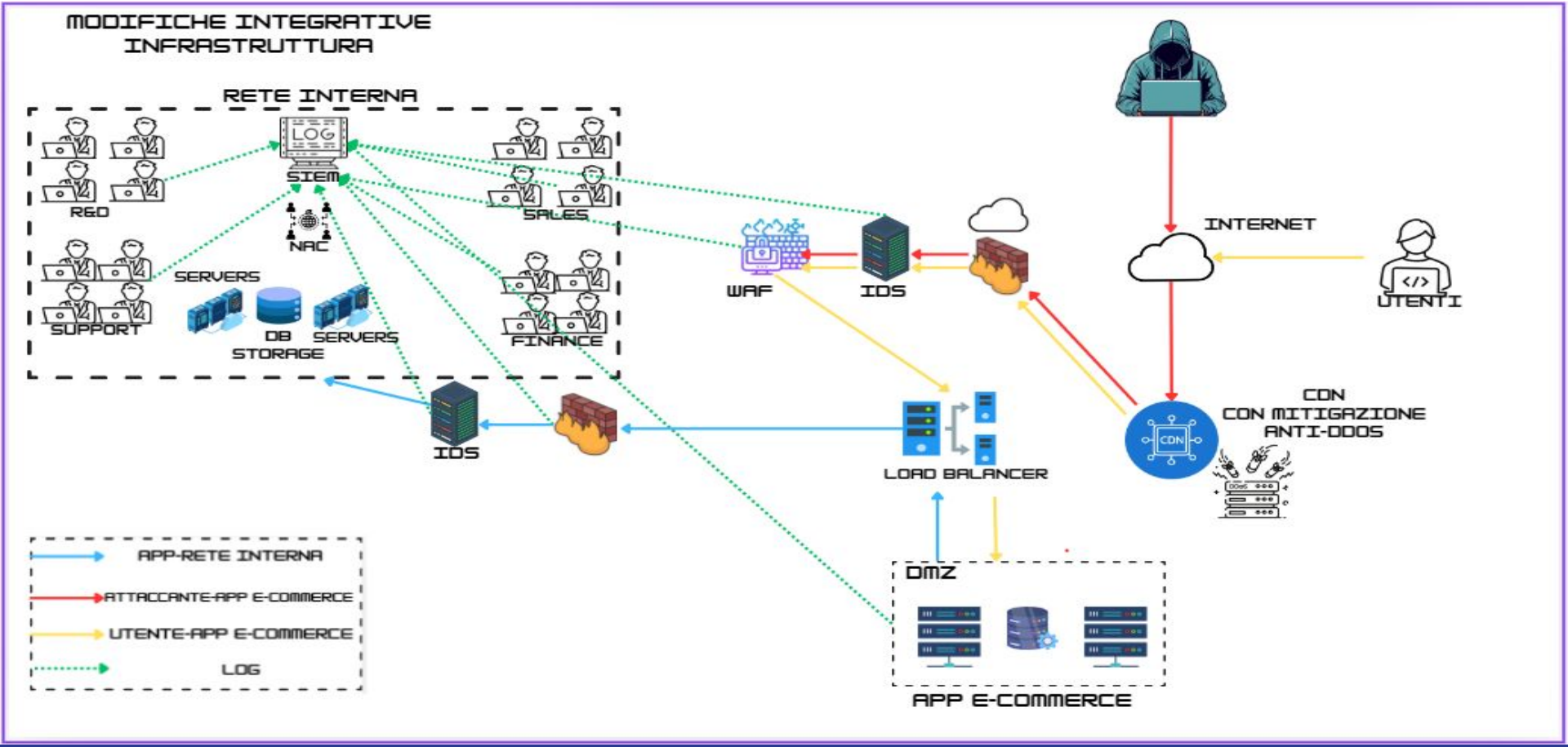
Backup e Ripristino

Disporre di un piano di backup automatizzato per garantire un rapido ripristino dell'infrastruttura in caso di attacco è essenziale per garantire la Business Continuity. Questo dovrebbe essere automatizzato e regolarmente testato, nonché includere strategie di replica dei dati e di failover in caso di attacco o “disastro”. È fondamentale assicurarsi che i backup siano conservati in posizioni sicure e separate, per prevenire che un attacco comprometta sia i dati originali che quelli di backup.

CAPTCHA

L’integrazione di challenges CAPTCHA è una strategia efficace contro gli attacchi DDoS, consentendo di verificare interazioni umane e migliorare la sicurezza dell’applicazione.

5. Modifica “aggressiva” infrastruttura - fig. 4



5. Modifica “aggressiva” infrastruttura - Riepilogo

Il client (utente finale) invia richieste HTTP/HTTPS all’app e-commerce. Il CDN riceve la richiesta, gestisce il traffico riducendo la latenza e protegge da attacchi DDoS. Il traffico passa per il primo IDS, che monitora il traffico per identificare eventuali attività sospette (IoC come scansioni eccessive, traffico in entrata da sorgenti sospette su porte critiche, numero elevato di richieste TCP/UDP su ampi intervalli di porte - sintomo di DDoS, tentativi di SQLi). La richiesta viene inoltrata al WAF. Il WAF filtra il traffico per prevenire attacchi SQLi, XSS e altri attacchi a livello applicativo. In seguito passa per il load balancer, che instrada il traffico ai diversi web server per bilanciare il carico.

Il web server nella DMZ elabora le richieste e genera la risposta per il client o passa i dati all’application server se necessario, per elaborare richieste più complesse. Il firewall interno, tra la DMZ e la rete interna, filtra il traffico verso la rete interna e blocca il traffico malevolo. Il secondo IDS, posto tra il firewall interno e la rete interna, monitora il traffico che passa verso i server interni (db e application server) al fine di identificare attività anomale per aiutare gli analisti a prevenire movimenti laterali in caso di compromissione. L’application server, nella rete interna, riceve richieste dai web server e comunica con il database per accedere ai dati. Una volta completata l’elaborazione della richiesta, l’app server invia la risposta al web server attraverso il flusso protetto, che la trasmette al client finale.

Il NAC controlla e gestisce l’accesso alla rete interna verificando che solo gli utenti autorizzati e autenticati possano connettersi da dispositivi sicuri. Si occupa di verificare l’accesso prima di consentire il traffico verso le risorse della rete.

Tutti gli eventi (accessi, richieste e attività sospette) vengono raccolti da un sistema SIEM per il parsing dei log con successiva normalizzazione e analisi dei dati centralizzata per il rilevamento di eventuali minacce.

Conclusioni

Il caso analizzato sottolinea l'importanza di un approccio strategico, olistico e multilivello alla sicurezza delle applicazioni web.

L’approccio di sicurezza proposto combina diverse tecnologie e best-practices atte a formare una barriera contro eventuali minacce esterne. Queste soluzioni dovrebbero formare parte di un’architettura di sicurezza coesa in cui ogni componente sia volta a prevenire, identificare e reagire in tempo reale agli incidenti di sicurezza. Ogni misura di sicurezza implementata deve essere considerata come parte di un ecosistema più ampio, dove la comunicazione, la collaborazione e la reattività sono fondamentali per garantire la sicurezza dell'applicazione e la protezione dei dati sensibili degli utenti.

Investire risorse per la formazione del personale per creare una cultura di sicurezza all’interno dell’organizzazione è altresì fondamentale, creando consapevolezza riguardo le minacce comuni come attacchi di social engineering o pratiche di sicurezza da seguire per garantire la salvaguardia dei dati sensibili.

Un impegno proattivo e costante nel monitoraggio attivo per rilevare e mitigare attacchi, così come la preparazione di un piano di risposta agli incidenti per garantire la business continuity e la continua valutazione delle vulnerabilità tramite penetration testing, analisi di codice e programmi di bug bounty sono tutte misure che permettono di ridurre al minimo l’incidenza e l’impatto di eventuali attacchi.

Infine, rimanere aggiornati sulle ultime minacce e sulle tecnologie emergenti (IA e machine learning) nel campo della cyber security può fornire un vantaggio strategico e migliorare la capacità di rilevare e rispondere a minacce, considerato che queste sono in continua evoluzione.