Progetto Security Operation

W20 D4 | 31.01.2025

Panoramica

L'applicazione di e-commerce è progettata per essere accessibile agli utenti tramite Internet, ma, al tempo stesso, deve garantire che eventuali attacchi non compromettano né l'applicazione stessa né la rete interna aziendale. Per questo motivo, il sistema è stato riorganizzato e protetto seguendo alcune best practice fondamentali per la sicurezza delle reti.

Richieste

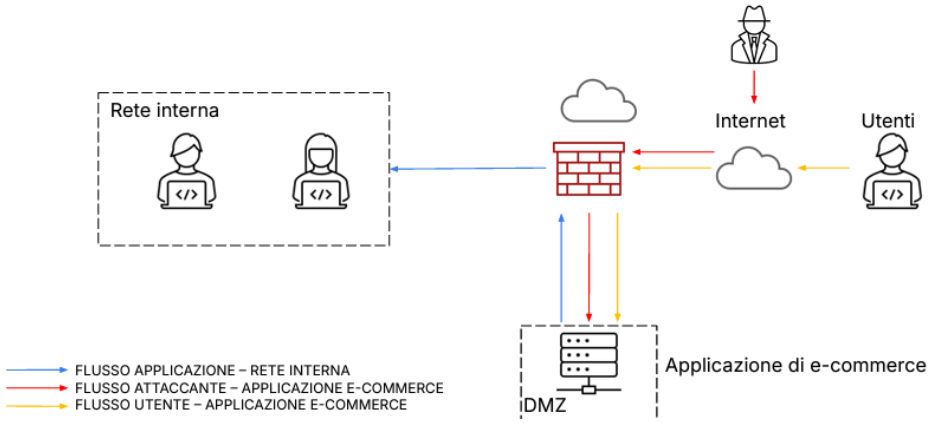
1. **Azioni preventive:** quali azioni preventive si potrebbero implementare per difendere l'applicazione Web da attacchi di tipo SQLi oppure XSS da parte di un utente malintenzionato? Modificate la figura in modo da evidenziare le implementazioni
2. **Impatti sul business:** l'applicazione Web subisce 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.

Indice

Azioni preventive

La protezione di un'applicazione web contro attacchi di tipo **SQL Injection (SQLi)** e **Cross-Site Scripting (XSS)** è fondamentale per garantire la sicurezza del sistema e dei dati degli utenti. Entrambi questi tipi di attacchi possono compromettere seriamente la sicurezza e l'integrità delle applicazioni web.

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.



SQL Injection (SQLi)

SQL Injection è un tipo di vulnerabilità che si verifica quando un utente malintenzionato riesce a iniettare codice SQL dannoso in una query, con l'obiettivo di eseguire comandi non autorizzati sul database dell'applicazione. Per prevenire questo tipo di attacco, è possibile implementare diverse misure preventive:

1. **Uso di query preparate e dichiarative**: Le query preparate (o "prepared statements") separano i comandi SQL dai dati, riducendo drasticamente il rischio di SQL Injection. In pratica, si definisce la struttura della query e i parametri vengono aggiunti successivamente, impedendo che l'input dell'utente possa alterare la logica della query.
2. **Sanificazione e validazione dei dati in ingresso**: È essenziale convalidare e sanificare tutti i dati provenienti dall'utente prima che vengano utilizzati nelle query SQL. Un'accurata validazione evita l'inserimento di valori pericolosi o non attesi, mentre la sanificazione rimuove o codifica eventuali caratteri speciali che potrebbero essere interpretati come comandi SQL.
3. **Limitare i privilegi dell'utente**: È importante che gli utenti e le applicazioni che interagiscono con il database abbiano solo i privilegi necessari per l'esecuzione delle operazioni richieste. Ad esempio, un'applicazione che visualizza solo i dati non dovrebbe avere privilegi di scrittura sul database, riducendo il rischio che un attacco di SQL Injection possa compromettere la base di dati.
4. **Uso di ORM (Object-Relational Mapping)**: L'adozione di un framework ORM automatizza molte delle best practices di sicurezza, proteggendo automaticamente dalle SQL Injection. Gli ORM gestiscono in modo sicuro l'interazione tra l'applicazione e il database, evitando errori legati alla scrittura manuale delle query.
5. **Uso di WAF (Web Application Firewall):** Introdurre un WAF per filtrare e monitorare il traffico HTTP verso l'applicazione Web. Il WAF può bloccare attacchi comuni come SQLi e XSS prima che raggiungano l'applicazione.

6. **Security Headers:** Configurare Header di sicurezza HTTP com X-Content,Typer-Options e Content-Security-Policy.
7. **Azioni Preventive con SOC:** Il SOC può monitorare continuamente il traffico di rete e le attività dell'applicazione per rilevare anomalie e potenziali attacchi SQLi e XSS. In Caso di rilevamento di un attacco, il SOC può rispondere rapidamente per mitigare l'impatto e contenere la minaccia.

Cross-Site Scripting (XSS)

Il Cross-Site Scripting è un attacco in cui un utente malintenzionato inietta del codice JavaScript nel contenuto di una pagina web, che viene poi eseguito nel browser degli altri utenti che visitano la pagina vulnerabile. Per prevenire questo tipo di attacco, è possibile adottare le seguenti misure:

1. **Escape dei dati in uscita**: Quando l'applicazione visualizza i dati inseriti dagli utenti (ad esempio, in un form o in un commento), è fondamentale fare l'escape di tutti i caratteri speciali (come <, >, &, etc.), per evitare che vengano interpretati come codice HTML o JavaScript. In questo modo, anche se l'utente inserisce un codice potenzialmente malevolo, esso verrà trattato come un semplice testo.
2. **Implementazione di politiche di Content Security (CSP)**: La Content Security Policy (CSP) è una tecnologia che consente di definire una politica di sicurezza per i contenuti che possono essere caricati ed eseguiti in un'applicazione web. Implementando una CSP, è possibile limitare l'esecuzione di script non autorizzati, riducendo il rischio di XSS. Ad esempio, si possono consentire solo script provenienti da domini sicuri e vietare l'esecuzione di script inline.
3. **Filtraggio degli input**: Una corretta validazione e filtraggio dei dati in ingresso è essenziale per prevenire XSS. Ad esempio, è possibile rimuovere o codificare tutte le sequenze che potrebbero rappresentare codice HTML o JavaScript pericoloso.
4. **Utilizzo di librerie di sicurezza**: È possibile utilizzare librerie di sicurezza preesistenti, come OWASP ESAPI (Enterprise Security API), che forniscono funzioni per gestire

automaticamente la protezione contro XSS e altre vulnerabilità comuni nelle applicazioni web.

L'implementazione di queste misure permette di ridurre significativamente il rischio di attacchi SQLi e XSS, aumentando la sicurezza complessiva dell'applicazione web.

Architettura di Rete con SOC proposta evidenziando le implementazioni

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. Il SOC monitora continuamente il traffico di rete e le attività dell'applicazione per rilevare e rispondere rapidamente a qualsiasi minaccia.

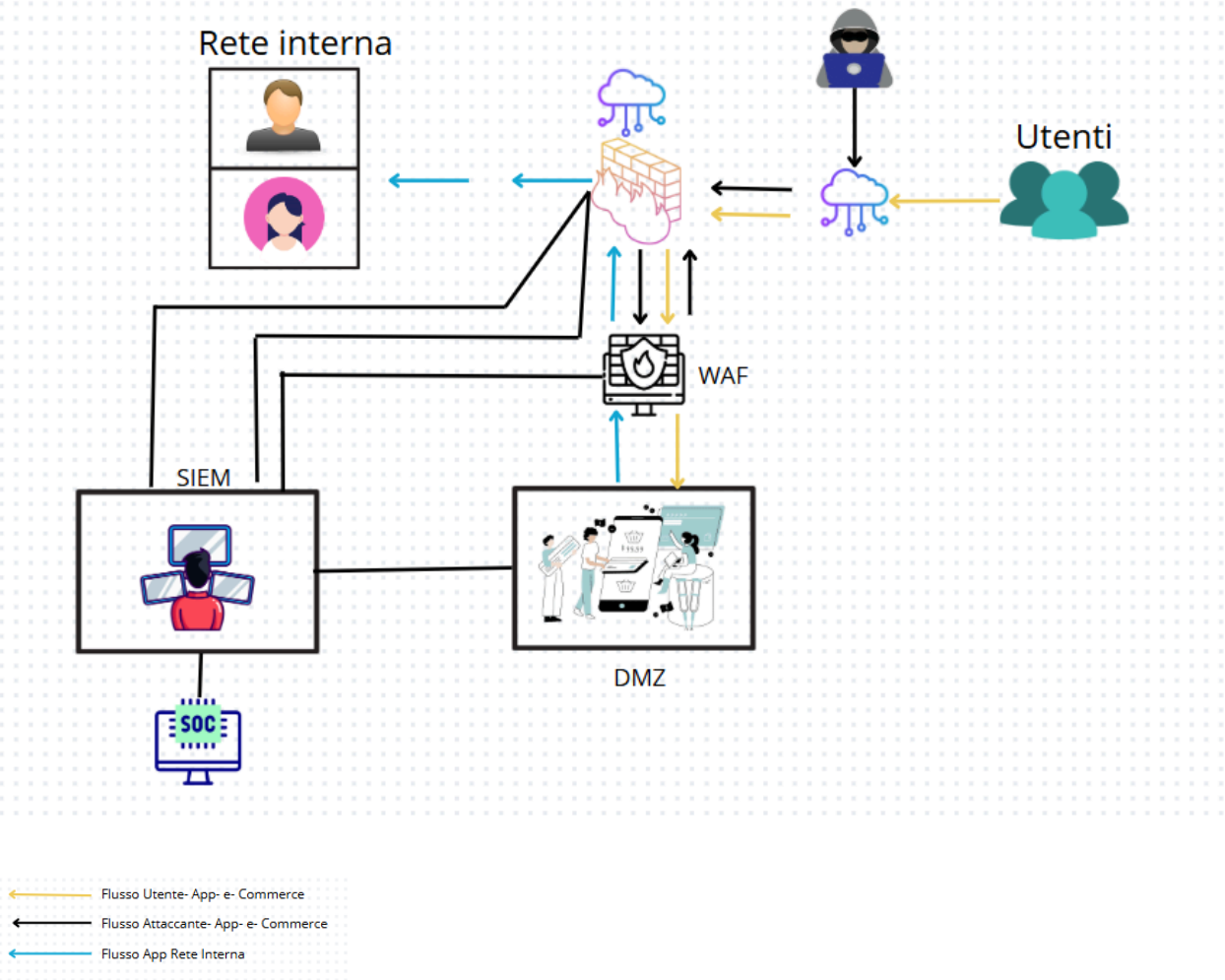
Descrizione testuale della figura aggiornata:

Implementazioni:

- **Web Application Firewall (WAF) tra Firewall e DMZ**
  - Il WAF analizza il traffico HTTP/HTTPS e blocca richieste malevole contenenti attacchi SQLi o XSS.
  - Si posiziona tra il firewall e la DMZ per proteggere i server web da exploit applicativi.
- **SIEM in Cloud per il monitoraggio degli attacchi**
  - Tutti i log del WAF, firewall, e altri dispositivi di sicurezza vengono inviati al SIEM in cloud gestito dal SOC.
  - Il SIEM analizza le richieste sospette e permette una risposta rapida a eventuali tentativi di attacco.
- **Modifica delle Policy del Firewall**
  - Configurare regole per filtrare richieste sospette e ridurre la superficie di attacco.
  - Limitare l'accesso solo ai servizi e alle porte strettamente necessarie.

-

Immagine 1:


Rete interna
Utenti
WAF
SIEM
SOC
DMZ
Flusso Utente- App- e- Commerce
Flusso Attaccante- App- e- Commerce
Flusso App Rete Interna

Impatto sul Business in caso di attacco DDoS

Gli attacchi Distributed Denial of Service (DDoS) sono uno dei tipi di attacchi più comuni e dannosi contro le applicazioni web. Questi attacchi mirano a rendere un'applicazione o un servizio online non disponibile, sovraccaricando i suoi server con una quantità enorme di traffico, bloccando gli utenti legittimi. Se un'applicazione web subisce un attacco DDoS e diventa inaccessibile per un periodo di tempo, ciò può avere gravi conseguenze sul business.

Nel caso specifico, supponiamo che un attacco DDoS renda l'applicazione non raggiungibile per un periodo di **10 minuti**. Se, durante un minuto di operatività, gli utenti spendono in media **1.500 €**, l'impatto economico del periodo di downtime è calcolato come segue:

- **Durata dell'indisponibilità**: 10 minuti
- **Spesa per minuto**: 1.500 €
- **Impatto economico totale**: 10 minuti × 1.500 € = **15.000 €**

Pertanto, l'attacco DDoS porta a una perdita di **15.000 €** di ricavi, considerando solo il periodo di 10 minuti di non disponibilità.

Azioni preventive contro DDoS:

Per mitigare gli effetti di un attacco DDoS, è importante adottare alcune misure preventive:

1. **Uso di Content Delivery Network (CDN)**: I CDN distribuiscono il traffico su più server situati in diverse località geografiche, riducendo la pressione sui server principali e migliorando la capacità di gestione degli attacchi DDoS. Inoltre, i CDN possono filtrare automaticamente il traffico sospetto.
2. **Filtraggio del traffico**: Implementare sistemi di filtraggio per identificare e bloccare il traffico sospetto. Le soluzioni di firewall avanzate e di protezione DDoS, come quelle fornite da Cloudflare o AWS Shield, possono rilevare e fermare gli attacchi in tempo reale.

3. **Rate Limiting**: Implementare il **rate limiting** per limitare il numero di richieste che un singolo indirizzo IP può fare in un determinato periodo di tempo. Questa misura aiuta a prevenire l'overload del sistema in caso di attacchi DDoS di tipo volumetrico.
4. **Scalabilità automatica**: Adottare una strategia di scalabilità automatica per garantire che i server possano gestire un volume di traffico superiore in caso di picchi di richiesta, inclusi gli attacchi DDoS.
5. **Anti-DDoS Cloud tra Internet e Firewall**
    a. Protegge la rete aziendale da attacchi DDoS che potrebbero essere sfruttati come vettore per diffondere malware o generare distrazioni durante un attacco più mirato.

Implementando queste misure, si può ridurre notevolmente il rischio di danni finanziari derivanti da un attacco DDoS, proteggendo la disponibilità dei servizi web.

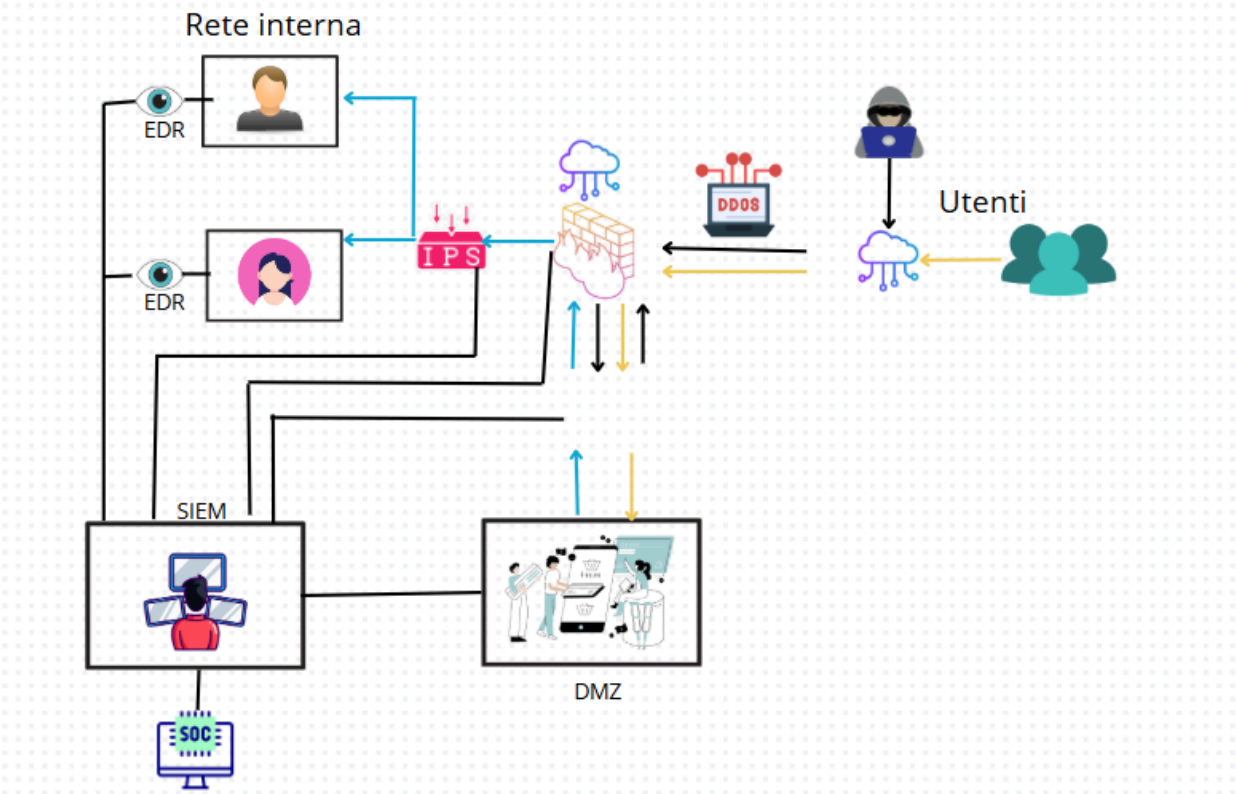
Response: Gestione dell'infezione da malware

Un'altra minaccia per la sicurezza delle applicazioni web è rappresentata dagli attacchi che comportano l'infezione di sistemi con malware. In questo scenario, l'obiettivo principale è impedire la propagazione del malware all'interno della rete aziendale, mentre non è immediatamente prioritario rimuovere l'accesso dell'attaccante alla macchina infetta.

Misure per prevenire la propagazione del malware:

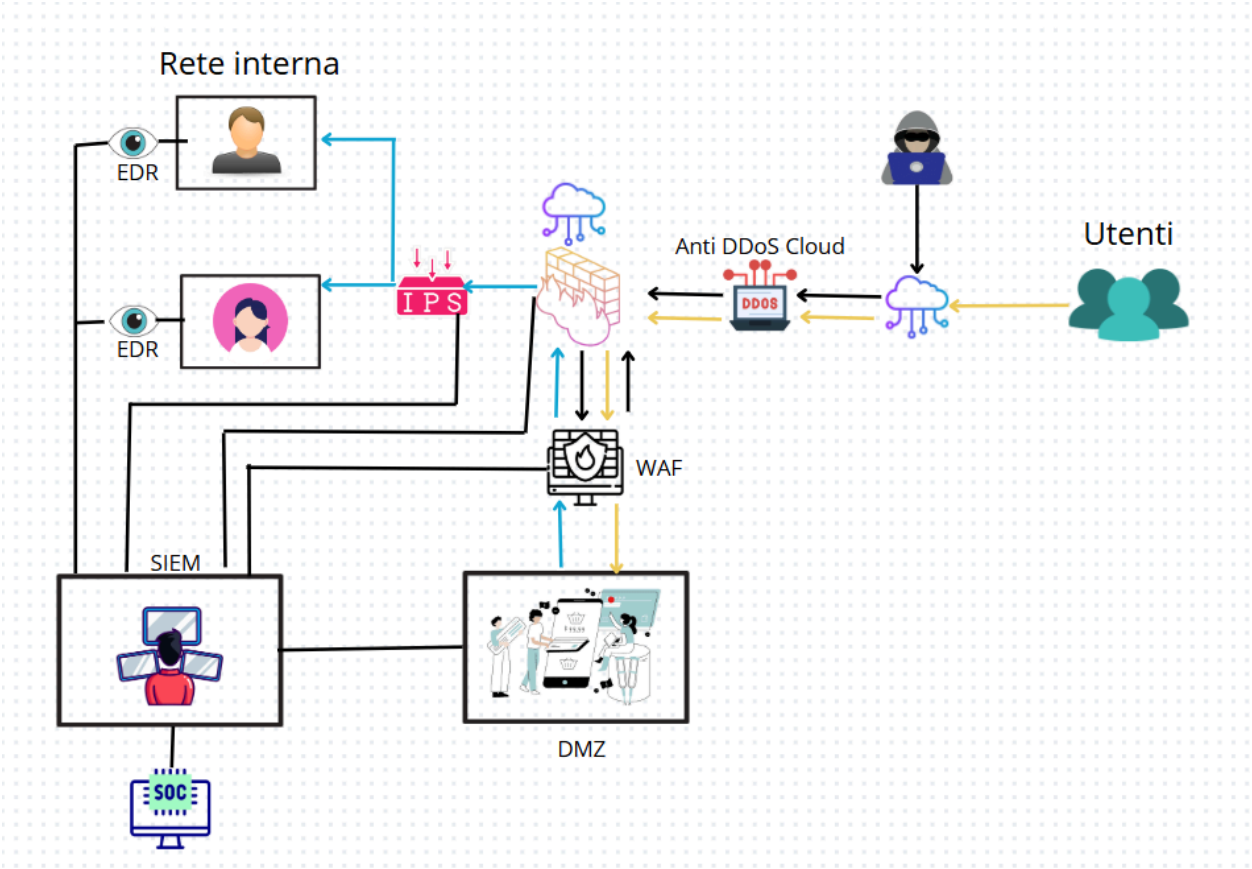
1. **Micro-segmentazione della rete**: La micro-segmentazione della rete riduce la possibilità che una singola macchina compromessa possa avere accesso a sistemi critici. In questo caso, la macchina infetta viene isolata in una sezione separata della rete.
2. **Isolamento della DMZ dalla rete interna**
    a. La DMZ **non deve comunicare con la rete interna**, per evitare che un attacco su un server web compromesso si propaghi all'interno dell'azienda.
    b. Il firewall deve bloccare qualsiasi connessione diretta tra la DMZ e la rete interna.
3. **IPS tra Firewall e Rete Interna**
    a. L'IPS analizza il traffico in ingresso e in uscita per rilevare e bloccare eventuali tentativi di propagazione del malware.
    b. Questo impedisce che il codice malevolo raggiunga dispositivi interni.
4. **EDR su ogni postazione interna**
    a. Ogni dispositivo aziendale deve avere un Endpoint Detection and Response (EDR) per monitorare attività sospette, impedire l'esecuzione di codice malevolo e bloccare tentativi di movimento laterale.
5. **SIEM in Cloud per il monitoraggio della sicurezza**
    a. Tutti i log di EDR, IPS, firewall e altri dispositivi vengono raccolti e analizzati dal SIEM per rilevare attività sospette.
    b. Il SOC può intervenire rapidamente per mitigare eventuali compromissioni.

Immagine 2


Rete interna
EDR
EDR
IPS
DDOS
Utenti
SIEM
SOC
DMZ
Flusso Utente- App- e- Commerce
Flusso Attaccante- App- e- Commerce
Flusso App Rete Interna

Unendo le due soluzioni:

Immagine 3



Conclusione

Le implementazioni proposte rafforzano significativamente la sicurezza dell'infrastruttura IT, garantendo sia la protezione preventiva delle applicazioni web che il contenimento di eventuali minacce informatiche.

L'adozione di un **WAF tra il firewall e la DMZ** mitiga il rischio di attacchi SQLi e XSS, mentre la segmentazione della rete con **IPS, firewall e policy di isolamento della DMZ** impedisce la propagazione di malware in caso di compromissione. Inoltre, l'integrazione di **soluzioni Anti-DDoS in cloud, EDR sugli endpoint e un SIEM centralizzato gestito dal SOC** consente un monitoraggio avanzato delle minacce e una risposta tempestiva agli incidenti di sicurezza.

Un attacco DDoS, se non adeguatamente mitigato, potrebbe avere un impatto significativo sul business, causando interruzioni dei servizi online, danni alla reputazione dell'azienda e perdite economiche dovute a tempi di inattività. La protezione tramite un **Anti-DDoS in cloud** posizionato tra Internet e il firewall riduce drasticamente il rischio di saturazione delle risorse aziendali, permettendo di mantenere la continuità operativa e garantire l'affidabilità dei servizi, anche in presenza di attacchi volumetrici.

Queste misure, unite a una gestione proattiva delle policy di sicurezza, migliorano la resilienza dell'infrastruttura, riducendo i rischi e garantendo una protezione più efficace contro attacchi informatici avanzati e possibili interruzioni operative dovute a eventi DDoS.