IMPLEMENTAZIONI

1-AZIONI PREVENTIVE

Rate limiting / throttling
Contro brute-force o attacchi automatici.

Autenticazione robusta (MFA)
Non specifica per SQLI/XSS, protegge la superficie d'attacco.

Logging & Alerting
WAF e CSP vanno monitorati.

Aggiornamenti e patch
Molti attacchi sfruttano vulnerabilità già note. Se il framework o DB è obsoleto, la struttura risulta comunque vulnerabile

2-IMPATTI SUL BUSINESS

Monitoring e Alerting in tempo reale

usare sistemi come Prometheus più Grafana, AWS CloudWatch, o servizi SIEM (es. Splunk, ELK stack) per monitoraggio attacchi e performance.

IDS/IPS (Intrusion Detection/Prevention System)

per rilevare intrusioni che bypassano il WAF o sfruttano vulnerabilità non ancora bloccate.

Snort, Suricata, OSSEC.

Security Headers HTTP

prevengono exploit di tipo clickjacking, XSS, e man-in-the-middle

- `Strict-Transport-Security`
- `X-Frame-Options`
- `X-Content-Type-Options`
- `Referrer-Policy`
- `Permissions-Policy`

Aggiornamenti e gestione delle patch

anche con un WAF, se si usa una versione di PHP/Apache/WordPress vulnerabile, si è a rischio.

gestire patch OS e software regolarmente con sistemi come Ansible, WSUS, o aggiornamenti automatici.

Autenticazione forte (MFA + rate limit login)

protezione contro brute-force e account takeover.

autenticazione a due fattori, blocco IP dopo tentativi falliti.

Isolamento dei servizi e microservizi

se un servizio viene compromesso, non deve compromettere tutto il sistema.

separare ambienti (dev/prod), usare container (Docker), segmentare la rete.

Backup regolari e test di ripristino

in caso di attacco riuscito, es. ransomware o defacement, devi poter ripristinare il servizio.

Backup risultano inutili se non vengono testati periodicamente.

3-RESPONSE

Impedire la propagazione verso la rete interna

Configurazione di un **firewall a doppia interfaccia** tra DMZ e rete interna.

Impostare una **regola deny-all** in uscita dalla DMZ verso la LAN (default drop).

Implementare un **firewall segmentato** (es. su pfSense, FortiGate o Cisco ASA) con regole esplicite solo per i servizi che *devono* comunicare.

Bloccare ogni comunicazione in uscita dalla DMZ

Dal firewall, creare una regola per **bloccare tutto il traffico in uscita** dalla DMZ verso Internet.

Se necessario, **usa proxy filtranti** o un **outbound NAT controllato** per limitare le eccezioni (es. aggiornamenti automatici).

Usare anche DNS filtering (es. Pi-hole, Quad9, OpenDNS) per evitare che il malware risolva i domini C&C.

Il traffico in ingresso resta attivo

Blocca immediatamente il traffico in ingresso verso il server compromesso.

Se si ha bisogno di analisi forense, **isolarlo in una VLAN separata** con accesso solo tramite un host di analisi.

Utilizzo un **sistema di contenimento automatico** (es. EDR/EDP + orchestrazione via SOAR o firewall con threat intelligence) per bloccare connessioni sospette.

4- SOLUZIONE COMPLETA

WAF per SQLI, XSS, DDOS

Implementazione pratica:

Scegliere un WAF:

Cloud: Cloudflare, AWS WAF, Azure Front Door

Locale: ModSecurity (con OWASP Core Rule Set), F5 ASM

Configurazione di regole specifiche per:

blocco query SQL sospette

escaping input per XSS

limiti di richiesta per protezione DDOS

Monitora i log WAF per capire se si sta subendo attacchi e da dove.

integrare il WAF con un SIEM per alert automatici.

Sviluppo sicuro dell'applicazione

Cosa implementare:

Dependency scanner: usa strumenti come `OWASP Dependency-Check`, `Snyk`, `npm audit`, `pip-audit`, ecc. per rilevare librerie vulnerabili.

CI/CD con controllo di sicurezza:

Implementa linting + test statici (SonarQube, semgrep)

Blocca il deploy se una vulnerabilità critica è presente

Logging delle azioni sensibili: accessi, modifiche a dati, privilegi elevati.

integra il logging con un sistema centralizzato (ELK stack o Splunk).

Segmentazione rete + DMZ

Creare **VLAN separate** per:

DMZ (es. web server pubblici)

rete interna (DB, servizi interni)

management

Usare **firewall a più zone** (es. pfSense, FortiGate) per filtrare traffico **tra** le VLAN.

Isolare il traffico: DMZ non deve iniziare la comunicazione verso la rete interna.

5- MODIFICA AGRESSIVA NESSUN LIMITE

Identity-Aware Proxy (IAP)

Intercetta l'accesso alle risorse e lo autorizza solo se l'identità utente è verificata (con MFA, gruppi, contesto).

Google IAP, **Azure AD App Proxy**, **Cloudflare Access**

Blocca l'accesso non autenticato a servizi anche se tecnicamente raggiungibili (es. pannelli interni, dashboard, Jenkins, ecc.).

RBAC + ABAC (Controllo degli accessi avanzato)

RBAC (Role-Based Access Control): ogni utente ha solo i permessi strettamente necessari in base al ruolo.

ABAC (Attribute-Based Access Control): aggiungere condizioni come "solo se da IP aziendale", "solo in orario lavorativo", ecc.

Kubernetes

API Gateway

IAM di servizi cloud (AWS, Azure, GCP)

EDR + UEBA

EDR (Endpoint Detection & Response):
Rileva attività sospette su host/server (comandi strani, esfiltrazione, persistence).

UEBA (User and Entity Behavior Analytics):
Analizza i comportamenti per rilevare anomalie (es. login inusuali, escalation silenziose).

CrowdStrike, SentinelOne, Microsoft Defender ATP, Elastic Security.

SIEM + SOAR (monitoraggio + risposta automatica)

SIEM (Security Information and Event Management): raccoglie log da WAF, bastion, firewall, ecc. (es. Splunk, ELK, Wazuh)

SOAR (Security Orchestration, Automation and Response): automatizza reazioni: blocca IP, chiude sessioni sospette, isola macchine (es. Splunk Phantom, Cortex XSOAR)

Test di sicurezza automatizzati

SAST: Static Application Security Testing (es. SonarQube, Semgrep), analizza il codice sorgente per trovare vulnerabilità

DAST: Dynamic Application Security Testing (es. OWASP ZAP, Burp Suite), analizza un'applicazione in esecuzione, simulando il comportamento di una attaccante esterno

IAST: Interattivo, integrato durante i test (es. Contrast Security) unisce i vantaggi di SAST e DAST in modo più preciso

Automatizzali nella pipeline CI/CD → blocca il deploy se qualcosa è critico.

Policy di rotazione segreti

Hashicorp Vault, gestisce in modo sicuro chiavi, token e credenziali

AWS Secrets Manager, serve per archiviare, proteggere, gestire e ruotare automaticamente i segreti

Microsoft Azure Key Vault, gestisce chiavi crittografiche, certificati e password

Doppler, alternativa a quelle elencate sopra gestisce in modo centralizzato e sicuro i segreti

i token/API key/credenziali devono avere durata breve e rotazione automatica.

Controlli lato client (Security hardening)

Impostare **CSP (Content Security Policy)**, `X-Frame-Options`, `X-XSS-Protection`

Usare **Subresource Integrity (SRI)** per proteggere script da CDN

HSTS per forzare HTTPS