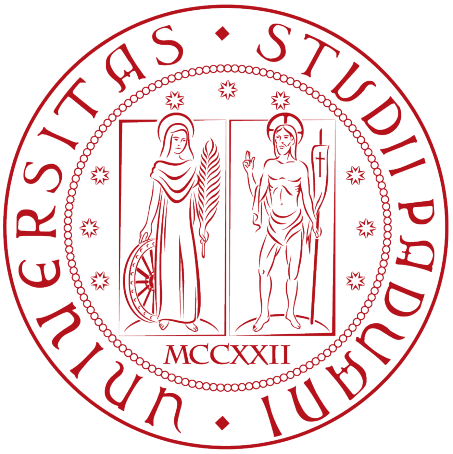
Università degli Studi di Padova

Dipartimento di Matematica "Tullio Levi-Civita"

Corso di Laurea in Informatica



Implementazione e Ottimizzazione di un Web Application Firewall per la protezione di applicazioni web

Tesi di laurea

Relatore
Prof. Davide Bresolin

Laureando
Andrea Perozzo
Matricola 2082849

Anno Accademico 2024-2025

Capitolo 1

Introduzione

Le applicazioni *web* rappresentano spesso l'anello più esposto verso l'esterno e, di conseguenza, il principale punto d'ingresso per attacchi informatici. Per contrastare questo rischio, i *Web Application Firewall (WAF)* costituiscono una valida soluzione a livello applicativo, offrendo protezione contro minacce diffuse come *SQL injection (SQLi)*, *Cross-Site Scripting (XSS)*, *Denial of Service (DoS)* e molte altre vulnerabilità.

Il mio *stage*, svolto presso *Kirey Group* per un periodo di due mesi, ha avuto come obiettivo principale l'implementazione e il perfezionamento di una configurazione di sicurezza capace di proteggere un sistema esposto da traffico malevolo. Questo percorso mi ha permesso di approfondire in modo pratico numerosi aspetti della sicurezza applicativa, dalla definizione delle *policy* alla loro verifica e ottimizzazione tramite appositi strumenti di analisi e *test*.

Ho scelto questo progetto formativo perché da tempo nutro un forte interesse per il mondo della *cybersecurity*, e poter lavorare direttamente su una tecnologia come il *WAF* di F5 si è rivelata un'opportunità stimolante e coerente con i miei obiettivi di crescita.

La prima fase dello *stage* si è incentrata su un'attività di formazione pratica, articolata in una serie di laboratori guidati che mi hanno consentito di acquisire familiarità con le principali funzionalità di un *firewall* applicativo, approfondendo sia le logiche di protezione che la configurazione iniziale delle componenti fondamentali.

Durante i laboratori ho lavorato in un ambiente simulato che riproduceva un'infrastruttura realistica, utilizzando un'applicazione *web* vulnerabile a scopo didattico che mi ha permesso di esercitarmi nell'analisi del traffico, nella definizione delle regole di sicurezza e nella gestione dei relativi *log*, sperimentando al contempo l'effetto delle *policy* applicate.

Questa fase introduttiva ha costituito le basi per affrontare con autonomia la seconda parte del progetto, in cui ho applicato le competenze acquisite per progettare e realizzare una configurazione di difesa più avanzata.

kirey group

1.1 L'azienda

Kirey Group è un *system integrator* e fornitore di soluzioni tecnologiche che opera a livello internazionale. Con sede a Padova (Corso Stati Uniti 14/B) e uffici distribuiti in Italia e all'estero, *Kirey Group* offre consulenza, servizi *IT* e soluzioni personalizzate in ambiti quali *Digital Transformation*, *Cybersecurity*, *Big Data & Analytics*, *Cloud* e *Artificial Intelligence*. Il gruppo collabora con *partner* tecnologici e supporta aziende di diversi settori nell'adozione di tecnologie per migliorare la competitività e la resilienza dei propri sistemi informativi.

1.2 L'idea

Il progetto si propone di implementare e configurare un *WAF* capace di garantire una protezione contro le principali tipologie di attacco, senza introdurre impatti negativi sulle *performance* delle applicazioni.

Il lavoro si articola in diverse fasi: analisi delle vulnerabilità, configurazione del *WAF* su tecnologia F5, *testing* con strumenti come *Burp Suite*, ottimizzazione delle regole per ridurre i falsi positivi e implementazione di sistemi di monitoraggio in tempo reale.

1.3 Organizzazione del testo

Il secondo capitolo descrive in dettaglio l'organizzazione dello *stage*, il rapporto con l'azienda, la metodologia di lavoro adottata e l'analisi dei rischi.

Il terzo capitolo approfondisce l'analisi dei requisiti definiti per il progetto.

Il quarto capitolo presenta i concetti teorici e gli strumenti tecnologici alla base della soluzione implementata.

Il quinto capitolo descrive il lavoro pratico svolto, le problematiche riscontrate e le soluzioni adottate.

Nel sesto capitolo riporta le considerazioni finali, i risultati raggiunti e possibili margini di miglioramento.

Riguardo la stesura del testo, relativamente al documento sono state adottate le seguenti convenzioni tipografiche:

- gli acronimi, le abbreviazioni e i termini ambigui o di uso non comune menzionati vengono definiti nel glossario, situato alla fine del presente documento;
- per la prima occorrenza dei termini riportati nel glossario viene utilizzata la seguente nomenclatura: *parola*[g];
- i termini in lingua straniera o facenti parti del gergo tecnico sono evidenziati con il carattere *corsivo*.

Capitolo 2

Descrizione dello stage

Questo capitolo descrive in dettaglio l'organizzazione dello stage*, il rapporto instaurato con l'azienda e con il* tutor *aziendale, la metodologia di lavoro adottata e l'analisi preventiva dei rischi.*

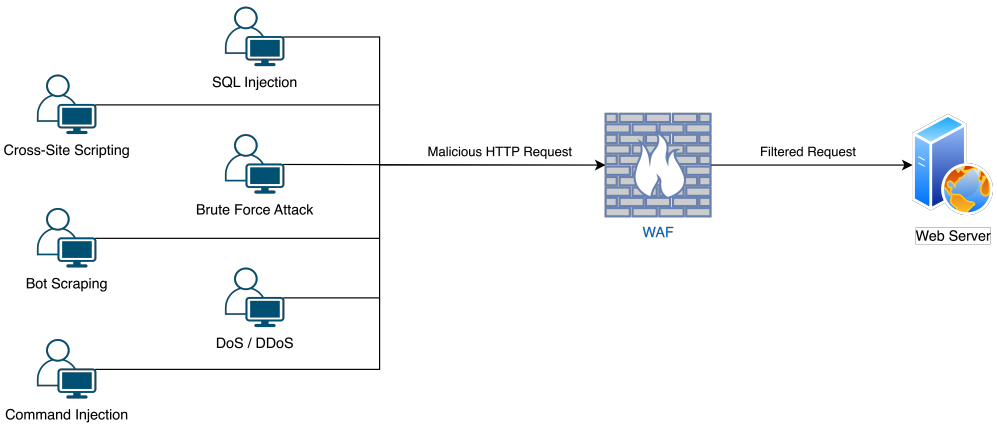


Figura 2.1: Schema WAF

2.1 Organizzazione e metodologia dello stage

Lo *stage* ha avuto una durata complessiva di circa due mesi, corrispondenti a circa 300 ore, articolate in due fasi distinte: una prima fase di formazione guidata e una seconda fase di lavoro autonomo.

Durante le prime due settimane si è svolto un percorso strutturato di apprendimento tramite 20 laboratori pratici, utilizzando un ambiente virtuale realizzato con *VMware Workstation* e composto da tre *Virtual Machine (VM)*:

- una *VM* con *Ubuntu Server*, in cui sono state installate le applicazioni vulnerabili *Juice Shop* (prima settimana) e *NodeGoat* (dalla terza settimana);
- una *VM* con la piattaforma *BIG-IP* di F5, utilizzata per configurare e gestire il *Advanced Web Application Firewall (AWAF)*;

- una *VM* con *Ubuntu Client*, utilizzata per accedere all'interfaccia *web* di gestione tramite *browser* e testare la configurazione.

In questa fase sono stati approfonditi i principali aspetti della configurazione e gestione del *WAF*, in particolare: prevenzione di attacchi quali *Brute Force*, *SQLi*, *XSS*, *Cross-Site Request Forgery (CSRF)* e mitigazione di traffico *bot*. L'apprendimento è avvenuto attraverso un approccio pratico e iterativo, che prevedeva la configurazione iniziale di una *policy* seguita da simulazioni di attacco tramite lo strumento *Burp Suite*, la verifica dei risultati nei *log* e la successiva ottimizzazione delle regole stesse (*tuning*).

Dalla terza settimana fino al termine dello *stage*, l'attività si è svolta in autonomia, configurando un ambiente simile a quello iniziale, ma sostituendo la *web application Juice Shop* con *NodeGoat*. La metodologia iterativa è rimasta invariata, applicando quanto appreso in precedenza per definire e migliorare progressivamente le *policy* di sicurezza in modo autonomo.

2.2 Rapporto con l'azienda e con il tutor aziendale

Lo *stage* è stato svolto all'interno di un ambiente aziendale strutturato e stimolante. Il *tutor* aziendale ha svolto un ruolo fondamentale nell'orientare le prime fasi dello *stage*, fornendo supporto operativo e metodologico durante i laboratori iniziali e assicurando incontri periodici per monitorare l'avanzamento del progetto, discutere eventuali problematiche e validare le soluzioni implementate.

Questo rapporto costante e costruttivo con il *tutor* ha favorito un apprendimento efficace e una crescita autonoma, garantendo al contempo il necessario supporto tecnico e metodologico durante tutto il periodo di *stage*.

2.3 Analisi preventiva dei rischi

Durante la fase iniziale dello *stage* sono stati identificati due principali rischi potenziali, ciascuno associato a una strategia preventiva:

1. Difficoltà nell'apprendimento iniziale della piattaforma
Descrizione: La configurazione del *AWAF* di F5 presenta una complessità intrinseca che avrebbe potuto rallentare la fase iniziale dello *stage*.
Soluzione: Sono stati pianificati laboratori guidati con il supporto del *tutor* così da permettere un apprendimento graduale, accompagnato da chiarimenti settimanali per affrontare eventuali dubbi tecnici.

2. Bilanciamento tra sicurezza ed esperienza utente
Descrizione: Una configurazione troppo restrittiva avrebbe potuto generare un elevato numero di falsi positivi, compromettendo l'usabilità dell'applicazione protetta.
Soluzione: È stato adottato un approccio iterativo: una prima fase in *transparent mode*, con successivo affinamento graduale delle regole fino al passaggio definitivo alla *blocking mode*, dopo accurate verifiche sui *log* generati.

Acronimi e abbreviazioni

AWAF Advanced Web Application Firewall. 3, 4, 6, 7

CSRF Cross-Site Request Forgery. 4, 6, 8

DoS Denial of Service. 1, 6

HTML HyperText Markup Language. 7, 8

HTTP HyperText Transfer Protocol. 6–8

HTTPS HyperText Transfer Protocol Secure. 6–8

JS JavaScript. 7, 8

OWASP Open Worldwide Application Security Project. 7

SQL Structured Query Language. 8

SQLi SQL injection. 1, 4, 6, 8

VM Virtual Machine. 3, 4, 8

WAF Web Application Firewall. 1, 2, 4, 6–8

WWW World Wide Web. 7, 8

XSS Cross-Site Scripting. 1, 4, 6, 8

Glossario

AWAF *Advanced Web Application Firewall*, soluzione di sicurezza avanzata offerta da F5 per proteggere applicazioni *web* da un'ampia gamma di minacce a livello applicativo, comprese vulnerabilità note e attacchi sofisticati come *bot*, *XSS*, *SQLi* e *CSRF*. 5

BIG-IP piattaforma *hardware* e *software* sviluppata da F5 che offre funzionalità avanzate di bilanciamento del carico (*load balancing*), sicurezza applicativa, gestione del traffico e ottimizzazione delle prestazioni delle applicazioni *web*. Include moduli come *AWAF*. 3, 7

Blocking Mode modalità operativa del *WAF* in cui il traffico riconosciuto come malevolo o non conforme alle *policy* definite viene attivamente bloccato, impedendone il raggiungimento della *web application* protetta. Si contrappone alla *transparent mode*, in cui le richieste non vengono bloccate ma solo monitorate.. 4

Bot programma automatico che effettua operazioni su *Internet*. I *bot* possono essere usati per scopi legittimi (ad esempio motori di ricerca) o malevoli (attacchi automatizzati, *spam*). Un *WAF* spesso implementa meccanismi di difesa contro il traffico generato da *bot* dannosi. 4, 6

Brute Force attacco che tenta di ottenere l'accesso a un sistema o servizio provando sistematicamente tutte le combinazioni possibili di credenziali (*username* e *password*) o chiavi di cifratura, fino a trovare quella corretta. Le moderne difese, come i *WAF*, implementano meccanismi per rilevare e bloccare tali tentativi. 4

Burp Suite suite integrata di strumenti per *test* di sicurezza delle applicazioni *web*. Permette di eseguire analisi del traffico *HyperText Transfer Protocol (HTTP)*/*HTTP Secure (HTTPS)*, attacchi automatizzati, manipolazione di richieste e molto altro. 2, 4

CSRF *Cross-Site Request Forgery* è una vulnerabilità delle applicazioni *web* che consente a un attaccante di indurre un utente autenticato a eseguire, inconsapevolmente, azioni indesiderate su un'applicazione *web* in cui è autenticato, sfruttando la fiducia dell'applicazione nei confronti del *browser* dell'utente. 5

DoS *Denial of Service* è un attacco informatico finalizzato a rendere indisponibile un servizio, una risorsa di rete o un'intera infrastruttura, sovraccaricando i *server* o saturando la banda con richieste malevole o massive. 5

F5 *F5 Networks* è un'azienda statunitense che sviluppa soluzioni *hardware* e *software* per la sicurezza, la disponibilità e l'ottimizzazione delle applicazioni, tra cui i prodotti della famiglia *BIG-IP* e *AWAF*. 1–4, 6

Firewall *Firewall*, sistema *hardware*, *software* o misto, progettato per monitorare e controllare il traffico di rete in entrata e in uscita in base a regole di sicurezza predefinite. Un *firewall* viene utilizzato per proteggere le reti da accessi non autorizzati e da attacchi esterni. I *WAF* rappresentano una tipologia specializzata di *firewall* applicativo.. 1

HTML *HyperText Markup Language*, linguaggio di *markup* utilizzato per strutturare contenuti ipertestuali sul *World Wide Web (WWW)*. Costituisce la base delle pagine *web*, descrivendone la struttura e gli elementi visuali. 5

HTTP *HyperText Transfer Protocol*, protocollo di livello applicativo usato per la trasmissione di documenti ipertestuali (come le pagine *web*) su *Internet*. È il protocollo su cui si basa il *WWW*. 5

HTTPS *HyperText Transfer Protocol Secure*, estensione sicura di *HTTP*. *HTTPS* impiega protocolli di cifratura per garantire la riservatezza e l'integrità dei dati trasmessi tra il *client* e il *server*. 5

JS *JavaScript*, linguaggio di programmazione interpretato, principalmente utilizzato per lo sviluppo di funzionalità dinamiche e interattive nelle pagine *web* lato *client*. È uno dei linguaggi fondamentali del *WWW* insieme a *HyperText Markup Language (HTML)*. 5

Juice Shop *Open Worldwide Application Security Project (OWASP) Juice Shop*, applicazione *web* vulnerabile progettata per scopi di formazione e *test* nel campo della *cybersecurity*. Consente di simulare e analizzare attacchi contro applicazioni *web*, supportando l'apprendimento pratico delle tecniche di protezione.. 3, 4

Log registro strutturato contenente eventi, messaggi o attività registrate da un sistema informatico. I *log* sono fondamentali per il monitoraggio della sicurezza, la diagnosi di problemi e la verifica del comportamento delle applicazioni. 1, 4, 8

NodeGoat *NodeGoat*, applicazione *web* vulnerabile, progettata per scopi didattici e di ricerca nell'ambito della sicurezza applicativa. Viene utilizzata per studiare e testare vulnerabilità comuni e relative contromisure.. 3, 4

OWASP *Open Worldwide Application Security Project*, organizzazione *no-profit* che promuove la sicurezza delle applicazioni *web* attraverso progetti *open-source*, linee guida e *standard* come l'*OWASP Top 10*, che elenca le vulnerabilità più critiche nelle applicazioni *web*. 5

Policy insieme di regole configurate in un sistema (ad esempio un *WAF*) che determinano il comportamento di protezione e le azioni da intraprendere in risposta al traffico applicativo. 1, 4, 6, 8

Query in informatica, una *query* è una richiesta formulata per ottenere informazioni da un sistema di gestione di basi di dati o da un sistema informativo. Nel contesto del *Structured Query Language (SQL)*, una *query* rappresenta un comando per interrogare o manipolare dati contenuti in un *database*. 8

SQL *Structured Query Language*, linguaggio *standard* utilizzato per l'interrogazione, la manipolazione e la definizione di dati all'interno di un *database* relazionale. 5

SQLi *SQL injection* è una tecnica di attacco che consiste nell'inserire comandi *SQL* malevoli in *input* apparentemente innocui dell'applicazione, allo scopo di manipolare le *query* verso il *database* sottostante, accedendo, alterando o eliminando dati sensibili. 5

Transparent Mode modalità operativa del *WAF* in cui il traffico viene solo monitorato e non bloccato. Consente di raccogliere dati sui tentativi di attacco e di validare l'efficacia delle *policy* configurate senza impattare direttamente sull'esperienza utente. Spesso utilizzata durante le fasi di *tuning* iniziale.. 4, 6

Tuning processo iterativo di ottimizzazione delle *policy* di sicurezza, che consiste nell'analizzare i risultati dei *test* e dei *log* per regolare e affinare progressivamente le regole di protezione, riducendo i falsi positivi e migliorando l'efficacia del *WAF*.. 4, 8

Ubuntu distribuzione del sistema operativo *Linux*, molto popolare per la sua semplicità d'uso e ampia comunità. È spesso utilizzata come sistema operativo per *server* e *VM* in ambito di sviluppo e *test*. 3, 4

VM *Virtual Machine*, macchina virtuale: un ambiente *software* che emula un *computer* fisico, consentendo di eseguire sistemi operativi e applicazioni isolati dal sistema *host*. Usata comunemente per *test*, sviluppo e virtualizzazione dei servizi. 5

VMware Workstation *VMware Workstation*, *software* di virtualizzazione che consente di creare e gestire *VM* su un *computer host*. È utilizzato per eseguire più sistemi operativi isolati simultaneamente in un ambiente virtuale.. 3

WAF *Web Application Firewall* è un sistema di protezione che monitora, filtra e analizza il traffico *HTTP*/*HTTPS* verso e da una applicazione *web*, con l'obiettivo di proteggere da attacchi noti e sconosciuti come *SQLi*, *XSS*, *CSRF* e altri attacchi a livello applicativo. 5

WWW *World Wide Web*, sistema di documenti ipertestuali interconnessi accessibili tramite *Internet*. Permette agli utenti di navigare tra pagine *web* tramite *browser* utilizzando protocolli come *HTTP* e *HTTPS*. 5

XSS *Cross-Site Scripting* è una tipologia di vulnerabilità delle applicazioni *web* che consente a un attaccante di iniettare codice *JavaScript (JS)* o *HTML* malevolo nelle pagine visualizzate da altri utenti, con lo scopo di rubare dati sensibili, sessioni utente o manipolare il contenuto della pagina. 5