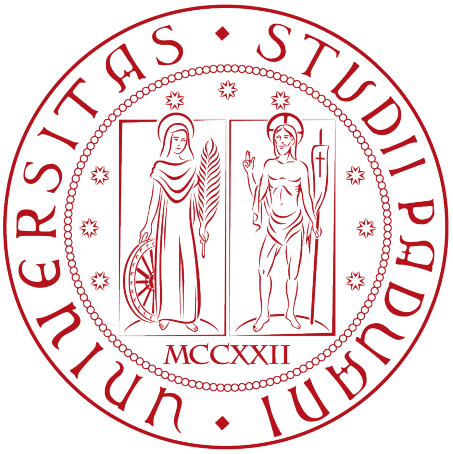
Università degli Studi di Padova

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

Sommario

Il presente documento descrive il lavoro svolto durante il periodo di stage, della durata di circa trecento ore, dal laureando Pinco Pallino presso l'azienda Azienda S.p.A. Gli obbiettivi da raggiungere erano molteplici.
In primo luogo era richiesto lo sviluppo di ... In secondo luogo era richiesta l'implementazione di un ... Tale framework permette di registrare gli eventi di un controllore programmabile, quali segnali applicati Terzo ed ultimo obbiettivo era l'integrazione ...

Indice

Elenco delle figure

Elenco delle tabelle

Capitolo 1

Introduzione

Le applicazioni *web* costituiscono spesso l'anello più esposto all'esterno, e quindi il principale vettore di attacco per attori malevoli intenzionati a sottrarre dati sensibili o a compromettere i sistemi aziendali. In questo contesto, i Web Application Firewall (WAF) rappresentano una soluzione per rafforzare la sicurezza a livello applicativo, proteggendo da attacchi noti come SQL injection (SQLi), Cross-Site Scripting (XSS), Denial of Service (DoS) e molti altri.

Lo *stage* svolto presso *Kirey Group*, della durata di due mesi, ha avuto come obiettivo l'implementazione e l'ottimizzazione di un WAF a protezione di una *web* application server. Durante questo periodo, ho potuto approfondire in maniera pratica vari aspetti della sicurezza applicativa, configurare policy e sperimentare tecniche di *test* e ottimizzazione delle regole di sicurezza.

Ho scelto questo progetto di *stage* perché nutro interesse nella *cybersecurity* e aver avuto l'opportunità di lavorare su un prodotto di sicurezza professionale come il WAF di F5 è stata particolarmente stimolante.

Durante la prima fase dello *stage*, ho seguito un percorso di formazione strutturato attraverso una serie di laboratori pratici (20 in totale), svolti su una Virtual Machine (VM) di Ubuntu server con ambiente di test rappresentato inizialmente da *Juice Shop*, una nota applicazione vulnerabile usata per il *training* in *cybersecurity*. Tramite questi laboratori ho acquisito competenze pratiche nell'uso di strumenti quali F5, Burp Suite, tecniche di analisi dei log e gestione delle policy.

A partire dalla terza settimana, ho iniziato la fase di sperimentazione autonoma, scegliendo la web app *NodeGoat* come ambiente *target* da proteggere. In questa fase ho proceduto a configurare il WAF per difendere l'applicazione da traffico malevolo reale e a validare l'efficacia delle regole implementate.

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.

Nel contesto del mio *stage*, ho avuto l'opportunità di formarmi sotto la guida del *tutor* aziendale Stefano Marchetti, focalizzandomi sul tema della protezione delle *web application* mediante WAF.

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.

La scelta di affrontare questo tema nasce soprattutto dall'interesse personale verso la sicurezza applicativa e dalla volontà di acquisire competenze in crescente richiesta.

1.3 Organizzazione del testo

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

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

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

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

Nel settimo 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

Processi e metodologie

Brevissima 1introduzione al capitolo

2.1 Processo sviluppo prodotto

Capitolo 3

Descrizione dello stage

Breve 1introduzione al capitolo

3.1 Introduzione al progetto

3.2 Analisi preventiva dei rischi

Durante la fase di analisi iniziale sono stati individuati alcuni possibili rischi a cui si potrà andare incontro. Si è quindi proceduto a elaborare delle possibili soluzioni per far fronte a tali rischi.

1. Performance del simulatore hardware
Descrizione: le performance del simulatore hardware e la comunicazione con questo potrebbero risultare lenti o non abbastanza buoni da causare il fallimento dei test.
Soluzione: coinvolgimento del responsabile a capo del progetto relativo il simulatore hardware.

3.3 Requisiti e obiettivi

3.4 Pianificazione

Capitolo 4

Analisi dei requisiti

Breve 1introduzione al capitolo

4.1 Casi d'uso

Per lo studio dei casi di utilizzo del prodotto sono stati creati dei diagrammi. I diagrammi dei casi d'uso (in inglese *Use Case Diagram*) sono diagrammi di tipo Unified Modeling Language (UML) dedicati alla descrizione delle funzioni o servizi offerti da un sistema, così come sono percepiti e utilizzati dagli attori che interagiscono col sistema stesso. Essendo il progetto finalizzato alla creazione di un tool per l'automazione di un processo, le interazioni da parte dell'utilizzatore devono essere ovviamente ridotte allo stretto necessario. Per questo motivo i diagrammi d'uso risultano semplici e in numero ridotto.

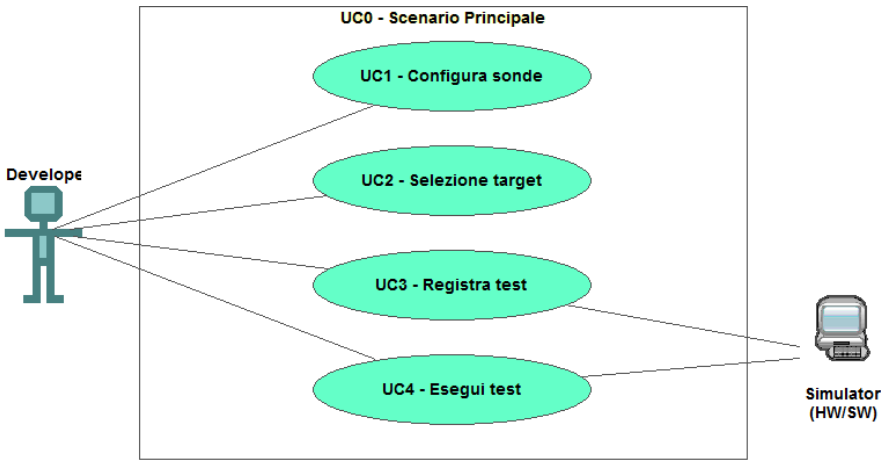


Figura 4.1: Use Case - UC0: Scenario principale

UC0: Scenario principale

Attori Principali: Sviluppatore applicativi.

Precondizioni: Lo sviluppatore è entrato nel plug-in di simulazione all'interno dell'IDE.

Descrizione: La finestra di simulazione mette a disposizione i comandi per configurare, registrare o eseguire un test.

Postcondizioni: Il sistema è pronto per permettere una nuova interazione.

4.2 Tracciamento dei requisiti

Da un'attenta analisi dei requisiti e degli use case effettuata sul progetto è stata stilata la tabella che traccia i requisiti in rapporto agli use case.
Sono stati individuati diversi tipi di requisiti e si è quindi fatto utilizzo di un codice identificativo per distinguerli.
Il codice dei requisiti è così strutturato R(F/Q/V)(N/D/O) dove:

R = requisito

F = funzionale

Q = qualitativo

V = di vincolo

N = obbligatorio (necessario)

D = desiderabile

Z = opzionale

Nelle tabelle 4.1, 4.2 e 4.3 sono riassunti i requisiti e il loro tracciamento con gli use case delineati in fase di analisi.

Tabella 4.1: Tabella del tracciamento dei requisti funzionali

Requisito	Descrizione	Use Case
RFN-1	L'interfaccia permette di configurare il tipo di sonde del test	UC1

Tabella 4.2: Tabella del tracciamento dei requisiti qualitativi

Requisito	Descrizione	Use Case
RQD-1	Le prestazioni del simulatore hardware deve garantire la giusta esecuzione dei test e non la generazione di falsi negativi	-

Tabella 4.3: Tabella del tracciamento dei requisiti di vincolo

Requisito	Descrizione	Use Case
RVO-1	La libreria per l'esecuzione dei test automatici deve essere riutilizzabile	-

Capitolo 5

Progettazione e codifica

Breve 1introduzione al capitolo

5.1 Tecnologie e strumenti

Di seguito viene data una panoramica delle tecnologie e strumenti utilizzati.

Tecnologia 1

Descrizione Tecnologia 1.

Tecnologia 2

Descrizione Tecnologia 2

5.2 Ciclo di vita del software

5.3 Progettazione

Namespace 1

Descrizione namespace 1.

Classe 1: Descrizione classe 1

Classe 2: Descrizione classe 2

5.4 Design Pattern utilizzati

5.5 Codifica

Capitolo 6

Conclusioni

6.1 Consuntivo finale

6.2 Raggiungimento degli obiettivi

6.3 Conoscenze acquisite

6.4 Valutazione personale

Appendice A

Appendice A

Citazione

Autore della citazione

Acronimi e abbreviazioni

AWAF Advanced Web Application Firewall. 12

CSRF Cross-Site Request Forgery. 12, 13

DoS Denial of Service. 1, 12

HTML HyperText Markup Language. 12–14

HTTP HyperText Transfer Protocol. 12–14

HTTPS HyperText Transfer Protocol Secure. 12–14

JS JavaScript. 13, 14

SQL Structured Query Language. 13

SQLi SQL injection. 1, 12, 13

UML Unified Modeling Language. 5, 13

VM Virtual Machine. 1, 13

WAF Web Application Firewall. 1, 2, 12, 13

WWW World Wide Web. 12–14

XSS Cross-Site Scripting. 1, 12–14

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 Cross-Site Request Forgery (CSRF). 11

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 Advanced Web Application Firewall (AWAF). 12

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. 12

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. 1, 2

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. 11

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. 11

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, 2, 12

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. 11

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. 11

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*. 11

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). 11

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

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

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*. 13

SQL *Structured Query Language*, linguaggio *standard* utilizzato per l'interrogazione, la manipolazione e la definizione di dati all'interno di un *database* relazionale. 11

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. 11

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. 1

UML *Unified Modeling Language* è un linguaggio di modellazione e specifica basato sul paradigma *object-oriented*. L' svolge un'importantissima funzione di "lingua franca" nella comunità della progettazione e programmazione a oggetti. Gran parte della letteratura di settore usa tale linguaggio per descrivere soluzioni analitiche e progettuali in modo sintetico e comprensibile a un vasto pubblico. 11

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. 11

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. 11

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. 11

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. 11

Bibliografia

Riferimenti bibliografici

James P. Womack, Daniel T. Jones. *Lean Thinking, Second Editon*. Simon & Schuster, Inc., 2010.

Siti web consultati

Manifesto Agile. URL: http://agilemanifesto.org/iso/it/.