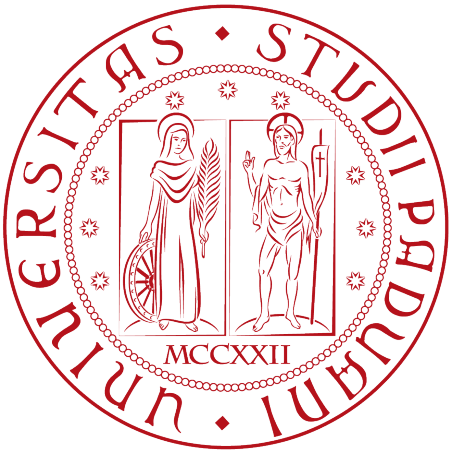
Università degli Studi di Padova

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* 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)*[g]costituiscono una valida soluzione a livello applicativo, offrendo protezione contro minacce diffuse come *SQL injection (SQLi)*[g], *Cross-Site Scripting (XSS)*[g], *Denial of Service (DoS)*[g]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. Questi esercizi 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. Questo contesto 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.

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 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

Descrizione dello stage

Questo capitolo descrive più in dettaglio come si è svolto lo stage, la metodologia di lavoro adottata e il rapporto con l'azienda e con il tutor aziendale. Vengono inoltre analizzati i principali rischi individuati e gli obiettivi definiti in fase di pianificazione.

2.1 Introduzione al progetto

L'obiettivo principale del progetto è stato quello di progettare e configurare un Advanced Web Application Firewall (AWAF) in grado di proteggere una *Web Application* server da attacchi noti e sconosciuti. In particolare, ci si è concentrati sulla protezione da minacce quali SQLi, XSS, Cross-Site Request Forgery (CSRF), attacchi di Brute Force e da bot malevoli.

Il progetto è stato suddiviso in due macro-fasi:

- una prima fase di apprendimento pratico, tramite lo svolgimento di una serie di 20 laboratori guidati, volti ad acquisire competenze sull'utilizzo e configurazione del prodotto AWAF della piattaforma BIG-IP di F5;
- una seconda fase autonoma, dedicata all'implementazione concreta della protezione su una *Web Application* scelta dallo studente. In questa fase è stata selezionata l'applicazione vulnerabile *NodeGoat*, su cui sono state applicate policy avanzate di sicurezza tramite il WAF.

Lo stage ha rappresentato un'opportunità formativa importante per acquisire competenze pratiche nel settore della sicurezza informatica.

2.2 Analisi preventiva dei rischi

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

1. Difficoltà nell'apprendimento e configurazione di un WAF
Descrizione: La configurazione di un AWAF come quello di F5 è complessa e richiede competenze a livello di sicurezza applicativa e a livello di *networking*..

Soluzione: Fase iniziale di formazione tramite laboratori guidati e disponibilità del tutor aziendale per chiarimenti tecnici.

2. Difficoltà nel bilanciare protezione ed esperienza utente
Descrizione: Configurare policy troppo restrittive nel WAF avrebbe potuto causare falsi positivi, compromettendo l'esperienza utente legittima..
Soluzione: Iterativo processo di *tuning* delle policy: inizialmente in modalità di apprendimento (*transparent mode*), successivo affinamento con *blocking mode* dopo verifica dei log.

2.3 Requisiti e obiettivi

Il progetto ha previsto i seguenti requisiti e obiettivi, suddivisi per priorità:

Obiettivi obbligatori

- Analisi e valutazione delle vulnerabilità presenti.
- Configurazione e implementazione del WAF.
- Esecuzione di *test* e simulazioni di attacchi per verificare l'efficacia delle soluzioni adottate.
- Ottimizzazione delle regole di sicurezza per ridurre i falsi positivi.
- Redazione di una documentazione tecnica che descriva il lavoro svolto e le metodologie adottate

Obiettivi desiderabili

- Monitoraggio continuo per valutare i progressi e l'efficacia delle soluzioni implementate

Obiettivi facoltativi

- Configurazione e gestione del WAF su piattaforme cloud per garantire scalabilità e flessibilità.

2.4 Pianificazione

Lo *stage* ha avuto una durata di 2 mesi, per un totale di circa 300 ore, articolato come segue:

- **Settimane 1–2**: formazione guidata tramite laboratori pratici con *juice-shop* come *Web Application* di test. In questa fase sono stati esplorati temi quali Brute Force attack prevention, bot mitigation, SQLi protection, XSS protection, CSRF prevention, tuning della policy e logging avanzato.
- **Settimana 3 in poi**: configurazione autonoma di un WAF su **NodeGoat**, con analisi dei flussi applicativi e applicazione di una protezione personalizzata tramite AWAF. Sono state definite e testate policy di sicurezza specifiche per

la protezione di API REST e per la protezione delle componenti più vulnerabili dell'applicazione.

Durante tutto il progetto sono stati svolti incontri di allineamento con il tutor aziendale per la discussione di eventuali criticità e il monitoraggio dell'avanzamento del lavoro.

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.

3. 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.

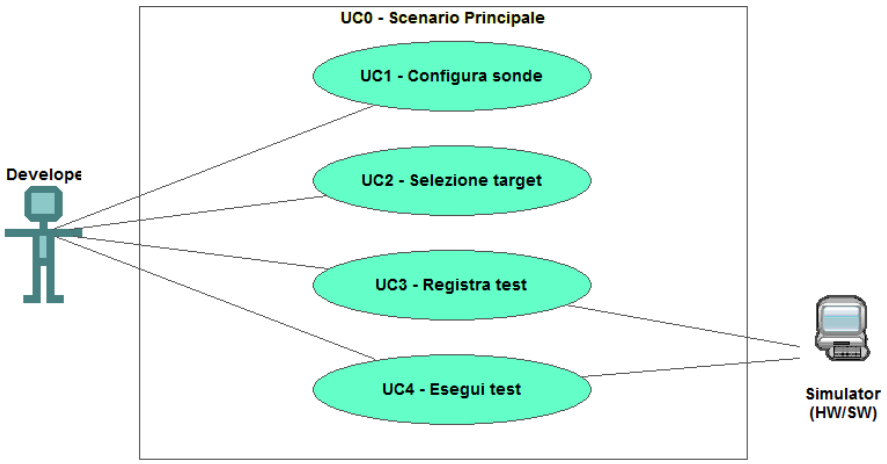


Figura 4.1: Use Case - UC0: Scenario principale

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. 3, 4, 14

CSRF Cross-Site Request Forgery. 3, 4, 14, 15

DoS Denial of Service. 1, 14

HTML HyperText Markup Language. 14–16

HTTP HyperText Transfer Protocol. 14–16

HTTPS HyperText Transfer Protocol Secure. 14–16

JS JavaScript. 15, 16

SQL Structured Query Language. 15

SQLi SQL injection. 1, 3, 4, 14, 15

UML Unified Modeling Language. 7, 15

VM Virtual Machine. 15

WAF Web Application Firewall. 1–4, 14, 15

WWW World Wide Web. 14–16

XSS Cross-Site Scripting. 1, 3, 4, 14–16

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. 13

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, 14

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. 3, 14

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. 3, 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

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. 13

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. 13

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. 1–3, 14

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. 13

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. 13

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*. 13

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). 13

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, 4

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*. 15

SQL *Structured Query Language*, linguaggio *standard* utilizzato per l'interrogazione, la manipolazione e la definizione di dati all'interno di un *database* relazionale. 13

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. 13

UML *Unified Modeling Language* è un linguaggio di modellazione e specifica basato sul paradigma *object-oriented*. L'UML 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. 13

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. 13

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. 13

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. 13

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. 13

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/.