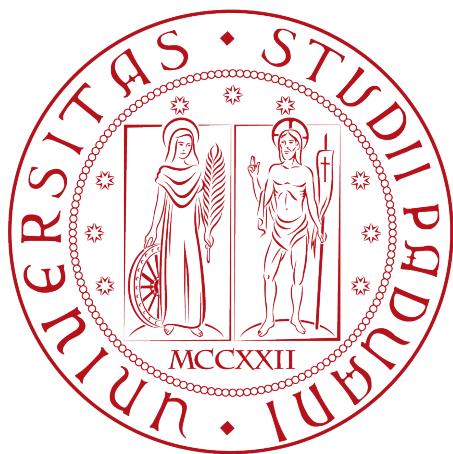


Università degli Studi di Padova

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

Andrea Perozzo: *Implementazione e Ottimizzazione di un Web Application Firewall per la protezione di applicazioni web*, Tesi di laurea, © Settembre 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* 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 realizzare una configurazione di difesa più avanzata.



## 1.1 L'azienda

*Kirey* 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* 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<sup>[g]</sup>*;
- 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.

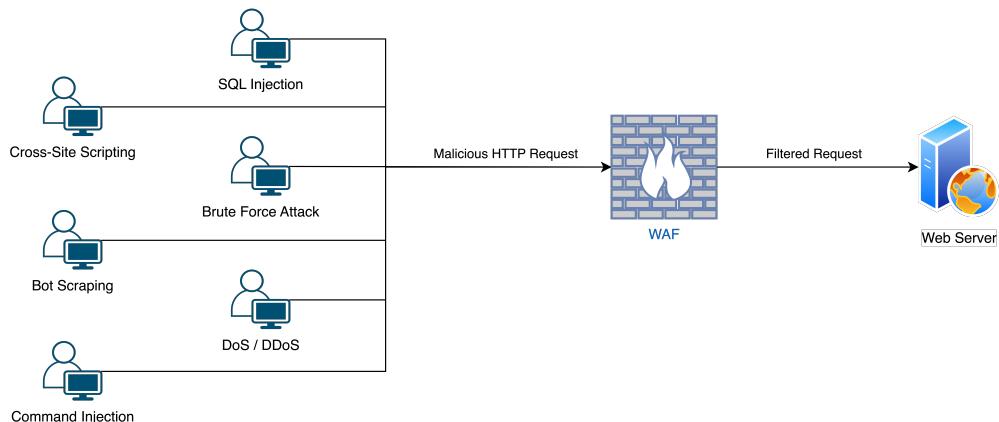


Figura 2.1: Schema WAF

### 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 *WAF*;

- 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* 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 *WAF* 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.

# Capitolo 3

## Analisi dei requisiti

*In questo capitolo vengono analizzati i requisiti del progetto, con l'obiettivo di definire le funzionalità del sistema e le interazioni previste tra l'Amministratore e la piattaforma di gestione del WAF.*

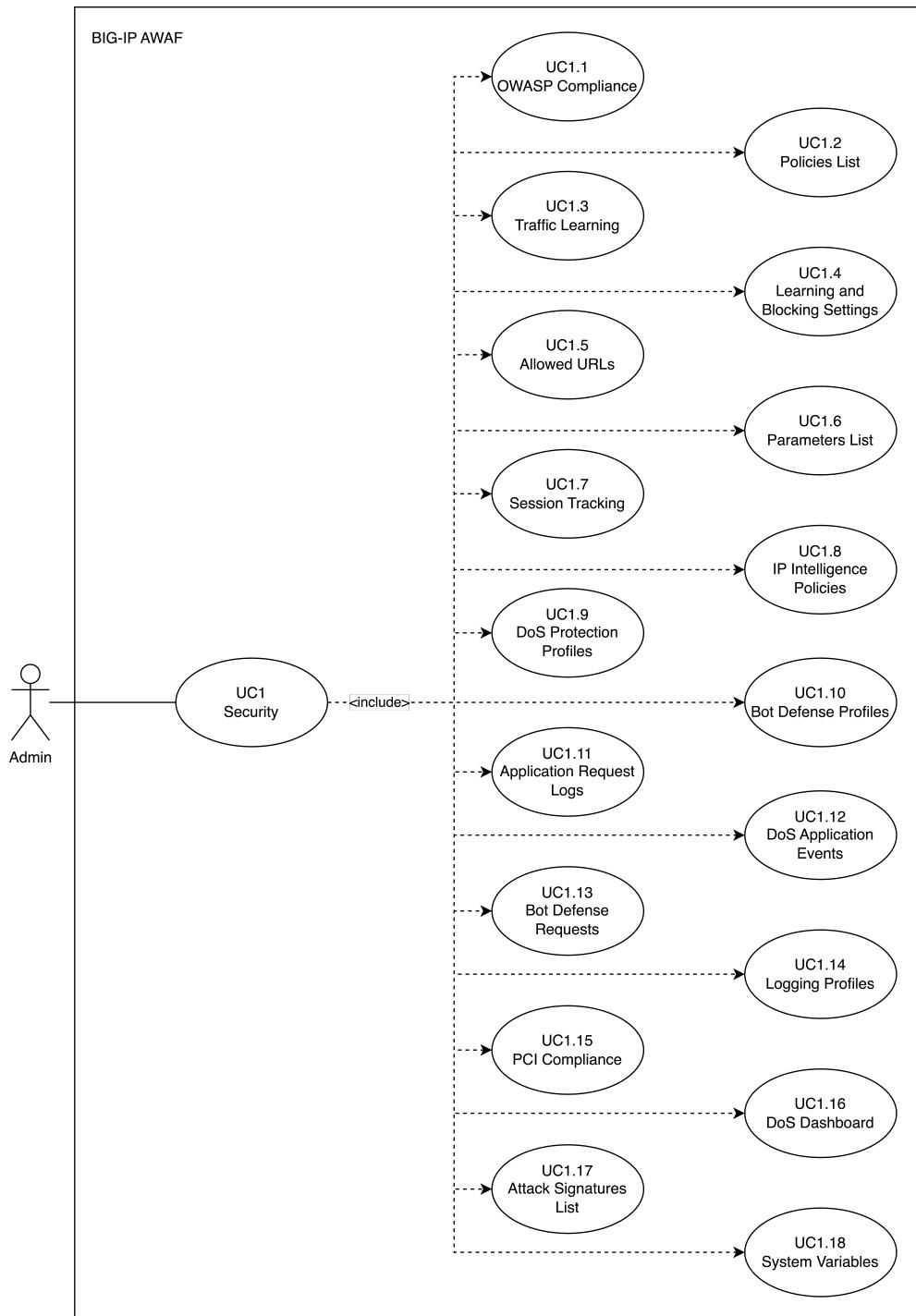
### 3.1 Casi d'uso

I casi d'uso riportati in questo capitolo sono stati definiti per descrivere in maniera formale e coerente le principali funzionalità offerte dal sistema, così come configurabili dall'Amministratore. Ogni caso d'uso rappresenta una singola operazione o gruppo omogeneo di operazioni disponibili attraverso l'interfaccia di amministrazione del [WAF](#).

Ciascun caso è presentato in un formato strutturato e standardizzato per garantire chiarezza e uniformità. In particolare, ogni caso d'uso contiene le seguenti sezioni:

- **Titolo** - descrive in modo sintetico l'obiettivo dell'interazione;
- **Attore Principale** - identifica il soggetto che esegue l'azione;
- **Precondizioni** - indicano lo stato del sistema necessario affinché il caso d'uso sia applicabile;
- **Postcondizioni** - descrivono la situazione attesa al termine dell'interazione;
- **Descrizione** - fornisce un resoconto discorsivo dell'operazione svolta e del suo scopo;

Ove rilevante, i casi d'uso sono accompagnati da diagrammi che ne rappresentano visivamente le interazioni principali.

**Figura 3.1:** Use Case 1

**UC1: Accesso e gestione della sezione *Security***

**Attore Principale:** Amministratore

**Precondizioni:** L'Amministratore deve disporre delle credenziali di accesso al sistema *WAF*.

**Postcondizioni:** Accesso alla sezione *Security* effettuato con successo e funzionalità disponibili per la configurazione.

**Descrizione:** L'Amministratore accede alla sezione *Security* dell'interfaccia di gestione del sistema *WAF*, dove ha la possibilità di configurare, monitorare e aggiornare tutte le funzionalità legate alla sicurezza applicativa e di rete. Questa sezione rappresenta il punto di ingresso principale per tutte le attività di sicurezza nel sistema.

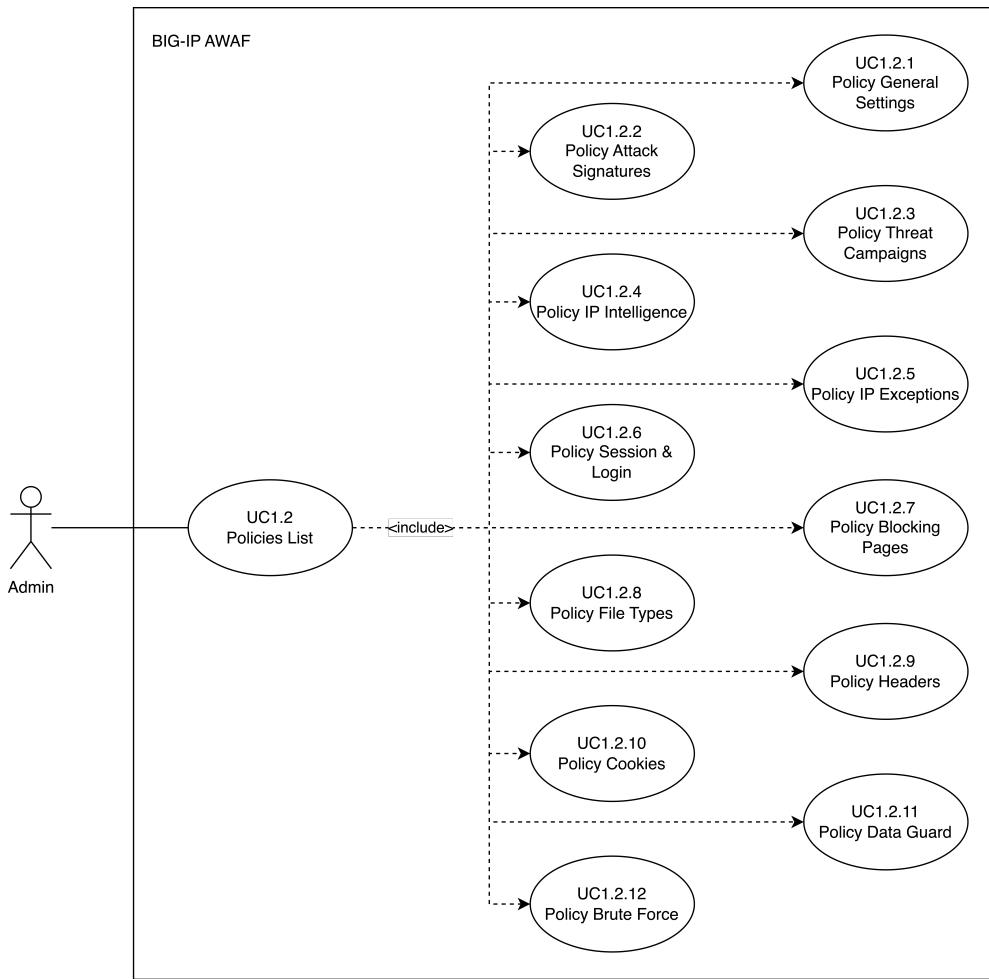
**UC1.1: Visualizzazione dei livelli di compliance *OWASP* delle *policy* esistenti**

**Attore Principale:** Amministratore

**Precondizioni:** Almeno una *policy* deve essere già configurata nel sistema *WAF*

**Postcondizioni:** Report aggiornato con il livello di conformità *Open Worldwide Application Security Project (OWASP)* per ciascuna *policy* esistente.

**Descrizione:** L'Amministratore ha la possibilità di accedere alla sezione di *OWASP Compliance Overview* del sistema *WAF*, dove può visualizzare un *report* riepilogativo che mostra il grado di conformità delle *policy* esistenti rispetto agli standard *OWASP Top 10*. Questa funzionalità permette di monitorare in modo continuo l'efficacia delle configurazioni di sicurezza, identificando eventuali categorie non sufficientemente coperte. Il *report* può inoltre essere utilizzato come riferimento per eventuali attività di *tuning* o ottimizzazione delle *policy*.

**Figura 3.2:** Use Case 1.2**UC1.2: Visualizzazione delle *policy* esistenti****Attore Principale:** Amministratore**Precondizioni:** Almeno una *policy* deve essere già configurata nel sistema *WAF*.**Postcondizioni:** Lista aggiornata contenente tutte le *policy* attualmente configurate.**Descrizione:** L'Amministratore può accedere alla sezione *policy list* del sistema *WAF*, dove viene visualizzato un elenco completo delle *policy* di sicurezza esistenti. Questa funzionalità consente di ottenere rapidamente una panoramica della configurazione corrente del sistema, semplificando le attività di gestione, verifica e manutenzione delle *policy* stesse.**UC1.2.1: Configurazione delle impostazioni generali della *policy*****Attore Principale:** Amministratore**Precondizioni:** Almeno una *policy* deve essere già configurata nel sistema *WAF*.

**Postcondizioni:** Impostazioni generali della *policy* aggiornate.

**Descrizione:** L'Amministratore può accedere alla sezione delle impostazioni generali di una *policy* di sicurezza, dove può definire il nome, la modalità di *enforcement* (*blocking mode* o *transparent mode*), le opzioni di *staging*, e la validità temporale della configurazione. Questa funzionalità consente di controllare il comportamento di base della *policy* e di adattarne l'applicazione alle esigenze operative del sistema.

#### UC1.2.2: Gestione delle firme di attacco all'interno della *policy*

**Attore Principale:** Amministratore

**Precondizioni:** Una *policy* deve essere attiva e modificabile.

**Postcondizioni:** Set di firme di attacco abilitato o aggiornato per la *policy* selezionata.

**Descrizione:** L'Amministratore può selezionare, attivare o disattivare gruppi specifici di firme di attacco all'interno di una *policy*. Questa funzionalità permette di associare al traffico applicativo solo le firme rilevanti per il contesto specifico, riducendo il rischio di falsi positivi e migliorando le prestazioni del sistema.

#### UC1.2.3: Abilitazione della protezione da campagne di attacco note

**Attore Principale:** Amministratore

**Precondizioni:** Il sistema deve essere aggiornato con le firme delle campagne di attacco.

**Postcondizioni:** Protezione contro campagne di attacco abilitata nella *policy*.

**Descrizione:** L'Amministratore può abilitare la protezione da campagne di attacco conosciute, basata su *intelligence* aggiornata fornita dal *vendor* del *WAF*. Questa funzionalità consente di rilevare e bloccare attacchi avanzati e coordinati riconducibili a *threat actors* noti, rafforzando la capacità di difesa proattiva del sistema.

#### UC1.2.4: Configurazione della protezione *IP Intelligence* nella *policy*

**Attore Principale:** Amministratore

**Precondizioni:** Il modulo *IP Intelligence* deve essere attivo nel sistema.

**Postcondizioni:** La *policy* è configurata per applicare azioni specifiche in base alla reputazione degli *Internet Protocol (IP)*.

**Descrizione:** L'Amministratore può abilitare la protezione *IP Intelligence* all'interno della singola *policy*, definendo le azioni da intraprendere in base alla categoria di rischio degli indirizzi *IP* (es. *spam*). Questo permette di applicare filtri proattivi sul traffico in ingresso direttamente a livello di *policy*.

#### UC1.2.5: Gestione delle eccezioni sugli indirizzi *IP*

**Attore Principale:** Amministratore

**Precondizioni:** Una *policy* deve essere attiva e con *IP Intelligence* abilitata.

**Postcondizioni:** Indirizzi *IP* inseriti nell'elenco delle eccezioni.

**Descrizione:** L'Amministratore può definire un elenco di indirizzi *IP* o *subnet* da escludere dalle verifiche di sicurezza previste dalla *policy*, ad esempio per motivi di *testing* o fiducia. Le eccezioni possono includere o escludere specifiche firme, violazioni o comportamenti. Questa funzionalità consente di adattare con precisione il comportamento della *policy* in contesti controllati.

#### UC1.2.6: Configurazione delle impostazioni di sessione e *login*

**Attore Principale:** Amministratore

**Precondizioni:** Deve essere attiva almeno una *policy* configurata.

**Postcondizioni:** Impostazioni di tracciamento sessioni e *login* aggiornate.

**Descrizione:** L'Amministratore può configurare il tracciamento delle sessioni utente all'interno della *policy*, definendo parametri come *session cookie*, durata della sessione, *URL* di *login*, e regole di identificazione per il rilevamento di anomalie. Questo consente di rafforzare il monitoraggio delle attività utente e di rilevare comportamenti sospetti.

#### UC1.2.7: Personalizzazione delle pagine di blocco e risposta

**Attore Principale:** Amministratore

**Precondizioni:** Deve essere attiva almeno una *policy* configurata in *blocking mode*.

**Postcondizioni:** Pagine di blocco personalizzate associate alla *policy*.

**Descrizione:** L'Amministratore può personalizzare i contenuti delle pagine di risposta mostrate agli utenti quando viene rilevata una violazione. È possibile modificare testo, codice *HyperText Markup Language (HTML)*, codici di stato *HyperText Transfer Protocol (HTTP)* e comportamento di reindirizzamento. Questa funzionalità consente di migliorare la *user experience* anche in caso di blocchi, fornendo messaggi coerenti e contestualizzati.

#### UC1.2.8: Gestione dei tipi di *file*

**Attore Principale:** Amministratore

**Precondizioni:** La *policy* deve essere attiva e configurata per ispezionare le richieste *HTTP*.

**Postcondizioni:** Elenco dei tipi di *file* controllati aggiornato.

**Descrizione:** L'Amministratore può definire un insieme di estensioni di *file* da monitorare o proteggere all'interno del traffico *HTTP*. Per ciascun tipo di *file* è possibile configurare azioni specifiche, come il blocco della richiesta, l'applicazione di firme di attacco o l'inserimento in modalità *staging*. Questa funzionalità consente di rafforzare la sicurezza delle applicazioni limitando o controllando l'accesso a determinati tipi di contenuti statici o dinamici.

#### UC1.2.9: Gestione degli *Header*

**Attore Principale:** Amministratore

**Precondizioni:** La *policy* deve essere attiva e configurata per analizzare i messaggi *HTTP*.

**Postcondizioni:** Configurazione degli *header* aggiornata.

**Descrizione:** L'Amministratore può configurare il comportamento della *policy* in relazione agli *header* presenti nelle richieste. È possibile specificare *header* da bloccare, monitorare o marcire come sensibili, oltre a definire regole per l'*enforcement* o lo *staging*. Questa funzionalità è utile per identificare richieste sospette che manipolano *header* noti o inseriscono *header* anomali, migliorando la protezione contro attacchi avanzati come *bypass* di autenticazione.

#### UC1.2.10: Gestione dei *Cookie*

**Attore Principale:** Amministratore

**Precondizioni:** La *policy* deve essere attiva e configurata per ispezionare i *cookie* delle richieste.

**Postcondizioni:** Regole di gestione e protezione dei *cookie* aggiornate.

**Descrizione:** L'Amministratore può configurare il comportamento della *policy* relativamente ai *cookie* trasmessi dalle applicazioni. È possibile applicare firme di attacco, mascherare contenuti sensibili, monitorare modifiche, e abilitare funzionalità di *enforcement* o *staging* su ciascun *cookie*. Questa funzionalità è fondamentale per prevenire tecniche di manipolazione della sessione e altre forme di compromissione dell'integrità dei dati di sessione lato *client*.

#### UC1.2.11: Configurazione della protezione dei dati sensibili

**Attore Principale:** Amministratore

**Precondizioni:** La *policy* deve essere attiva e associata a un'applicazione che gestisce dati potenzialmente sensibili.

**Postcondizioni:** Protezione dei dati sensibili attivata e configurata.

**Descrizione:** L'Amministratore può attivare la funzionalità *Data Guard* all'interno della *policy*, abilitando la mascheratura automatica di informazioni sensibili contenute nelle risposte dell'applicazione, come numeri di carte di credito, codici fiscali, dati personali o identificativi univoci. È possibile definire *pattern* di riconoscimento, configurare il livello di esposizione consentita (es. ultime quattro cifre visibili) e specificare quali risposte devono essere monitorate. Questa funzionalità consente di prevenire il rischio di *data leakage* in scenari di errore applicativo o esposizione involontaria.

#### UC1.2.12: Configurazione della protezione contro attacchi di *Brute Force*

**Attore Principale:** Amministratore

**Precondizioni:** La *policy* deve essere attiva e deve essere stato definito almeno un *URL* di *login*.

**Postcondizioni:** Meccanismo di protezione contro tentativi di *Brute Force* configurato.

**Descrizione:** L'Amministratore può configurare le impostazioni di protezione contro

gli attacchi *Brute Force*, definendo soglie di accesso, frequenza massima dei tentativi e azioni di mitigazione automatica, come *CAPTCHA*, blocco *IP* temporaneo o *logging* delle violazioni. È inoltre possibile specificare le credenziali di riferimento (*username*, *password*) e i parametri implicati nel processo di autenticazione, per rendere il sistema capace di riconoscere tentativi automatizzati o distribuiti. Questa funzionalità è fondamentale per proteggere gli *endpoint* di *login* da attività malevole finalizzate al furto di credenziali.

### UC1.3: Visualizzazione dell'apprendimento del traffico

**Attore Principale:** Amministratore

**Precondizioni:** Almeno una *policy* deve essere già configurata nel sistema *WAF*.

**Postcondizioni:** *Report* aggiornato con le informazioni apprese sul traffico.

**Descrizione:** L'Amministratore può accedere alla sezione *Traffic Learning* del sistema *WAF*, dove viene presentato un *report* dettagliato contenente tutte le informazioni apprese sul traffico applicativo. Questa funzionalità consente di monitorare l'efficacia della configurazione della *policy* e di individuare eventuali anomalie, comportamenti sospetti o *pattern* ricorrenti nel traffico. Il *Traffic Learning* rappresenta inoltre uno strumento fondamentale per il processo di *tuning* continuo delle *policy* di sicurezza, poiché fornisce suggerimenti che l'Amministratore può accettare o rifiutare per affinare progressivamente la protezione.

### UC1.4: Configurazione delle impostazioni di apprendimento e blocco del traffico

**Attore Principale:** Amministratore

**Precondizioni:** Almeno una *policy* deve essere già configurata nel sistema *WAF*.

**Postcondizioni:** Impostazioni di apprendimento e di blocco del traffico aggiornate.

**Descrizione:** L'Amministratore può accedere alla sezione *Learning and Blocking Settings* del sistema *WAF*, dove ha la possibilità di configurare le modalità di apprendimento automatico e i criteri di blocco applicati al traffico. Questa funzionalità consente di affinare progressivamente le *policy* di sicurezza, bilanciando la protezione dalle minacce con l'esigenza di ridurre i falsi positivi. In questo modo è possibile adattare dinamicamente la configurazione del *WAF* all'evoluzione del traffico e ai *pattern* di comportamento delle applicazioni protette.

### UC1.5: Configurazione degli *URL* consentiti

**Attore Principale:** Amministratore

**Precondizioni:** Almeno una *policy* deve essere già configurata nel sistema *WAF*.

**Postcondizioni:** Definiti gli *URLs* consentiti.

**Descrizione:** L'Amministratore può accedere alla sezione *Allowed URLs* del sistema *WAF*, dove ha la possibilità di configurare gli *URL* consentiti per il traffico applicativo. Questa funzionalità consente di definire in modo preciso quali risorse possono essere raggiunte dalle richieste, contribuendo a migliorare la sicurezza complessiva del sistema. Gli *URL* consentiti possono essere specificati in base a criteri come il dominio, il percorso e i parametri della *query*.

### UC1.6: Visualizzazione della lista dei parametri

**Attore Principale:** Amministratore

**Precondizioni:** Almeno una *policy* deve essere già configurata nel sistema *WAF* e deve essere stato appreso o configurato almeno un parametro.

**Postcondizioni:** Lista dei parametri visualizzata e consultata.

**Descrizione:** L'Amministratore può accedere alla lista dei parametri appresi o configurati all'interno del sistema *WAF*. Questa funzionalità consente di visualizzare in modo dettagliato le informazioni relative a ciascun parametro, inclusi il tipo di valore accettato, i metacaratteri consentiti, il comportamento in fase di *enforcement* o di *staging*, e l'applicazione delle firme di attacco. La consultazione della lista dei parametri permette di verificare la corretta configurazione della *policy* e di identificare eventuali necessità di *tuning*.

### UC1.7: Configurazione del tracciamento delle sessioni

**Attore Principale:** Amministratore

**Precondizioni:** Almeno una *policy* deve essere già configurata nel sistema *WAF*.

**Postcondizioni:** Il tracciamento delle sessioni risulta attivato e configurato secondo le impostazioni definite.

**Descrizione:** L'Amministratore può configurare il tracciamento delle sessioni utente nel sistema *WAF*. Questa funzionalità consente di monitorare in maniera approfondita l'attività delle sessioni che generano violazioni, abilitando il *logging* completo delle richieste provenienti da sessioni, indirizzi *IP* o dispositivi identificati come sospetti. Attraverso il tracciamento delle sessioni è possibile ottenere visibilità sui comportamenti anomali successivi a un attacco, migliorando le capacità di rilevamento e risposta agli incidenti di sicurezza.

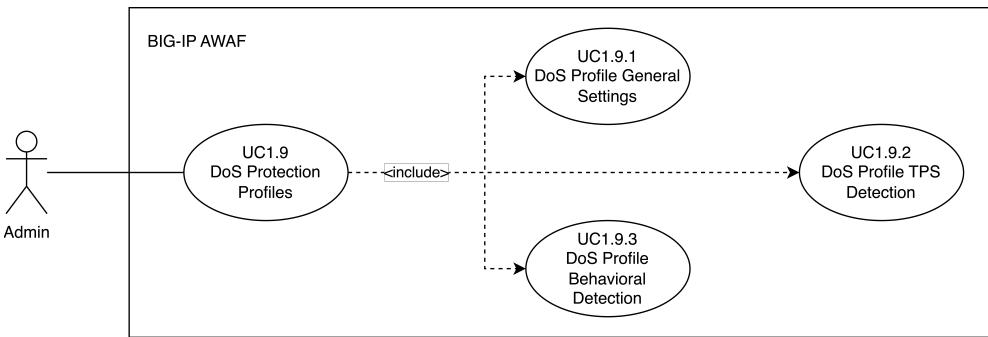
### UC1.8: Gestione delle *policy* di *IP Intelligence*

**Attore Principale:** Amministratore

**Precondizioni:** Il sistema *WAF* deve essere configurato per l'utilizzo della funzionalità di *IP Intelligence*.

**Postcondizioni:** Le *policy* di *IP Intelligence* verranno visualizzate o modificate secondo le configurazioni effettuate.

**Descrizione:** L'Amministratore può accedere alla sezione dedicata alla gestione delle *policy* di *IP Intelligence*, dove ha la possibilità di consultare, creare o modificare le *policy* associate ai diversi profili di rischio degli indirizzi *IP*. Questa funzionalità consente di definire regole di trattamento automatico per il traffico proveniente da sorgenti ritenute pericolose, sulla base di *blacklist* categorizzate (es. *malware*, *phishing*, *spam*, ecc.). Le azioni applicabili comprendono, ad esempio, il blocco immediato, la redirezione o la registrazione di un evento. Il controllo proattivo degli *IP* permette di rafforzare la postura difensiva del sistema.

**Figura 3.3:** Use Case 1.9

### UC1.9: Configurazione dei profili di protezione *DoS*

**Attore Principale:** Amministratore

**Precondizioni:** Il sistema *WAF* deve essere configurato per supportare la protezione dagli attacchi *DoS*.

**Postcondizioni:** Profilo *DoS* configurato e associato al traffico da proteggere.

**Descrizione:** L'Amministratore può configurare uno o più profili di protezione *DoS*, definendo soglie di traffico, criteri di rilevamento e contromisure da applicare in caso di attacco. Questa funzionalità consente di proteggere l'applicazione da minacce di tipo *DoS*, sia volumetriche (basate su numero di richieste al secondo) che comportamentali. Il profilo può includere tecniche di mitigazione come il blocco delle richieste, l'iniezione di *CAPTCHA* o la verifica dell'integrità del *client*. I profili vengono successivamente assegnati ai *Virtual Server* per applicare la protezione in tempo reale.

#### UC1.9.1: Configurazione generale del profilo di protezione *DoS*

**Attore Principale:** Amministratore

**Precondizioni:** Il modulo *DoS Protection* deve essere attivo nel sistema.

**Postcondizioni:** Parametri generali del profilo *DoS* aggiornati.

**Descrizione:** L'Amministratore può accedere alle impostazioni generali di un profilo di protezione *DoS*, dove può definire il nome del profilo, lo stato di attivazione, e i *Virtual Server* o le *policy* a cui associare la protezione. In questa sezione è inoltre possibile configurare la modalità di *logging* e il comportamento di *default* del sistema in presenza di traffico anomalo. Questa configurazione iniziale costituisce la base su cui vengono applicati i meccanismi di rilevamento e mitigazione più avanzati.

#### UC1.9.2: Configurazione del rilevamento basato sul numero di richieste

**Attore Principale:** Amministratore

**Precondizioni:** Il profilo *DoS* deve essere già stato creato e associato a un *Virtual Server*.

**Postcondizioni:** Soglie *Transactions Per Second (TPS)* definite per identificare picchi anomali di traffico.

**Descrizione:** L'Amministratore può configurare soglie di traffico espresse in termini di *TPS* per rilevare e bloccare attacchi *DoS* di tipo volumetrico. È possibile definire livelli di soglia predefiniti, limiti personalizzati per *IP* o *subnet*, e azioni di risposta automatica in caso di superamento. Questa modalità di rilevamento è efficace per identificare rapidamente aumenti improvvisi e anomali del numero di richieste *HTTP* verso l'applicazione.

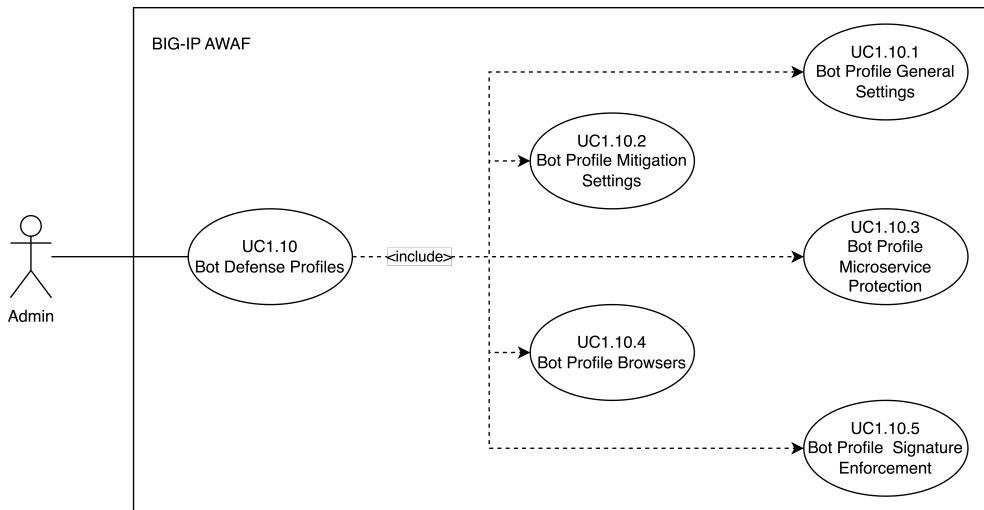
### UC1.9.3: Configurazione del rilevamento comportamentale e da stress

**Attore Principale:** Amministratore

**Precondizioni:** Il profilo *DoS* deve essere attivo e il traffico deve essere monitorato.

**Postcondizioni:** Meccanismi di rilevamento basati sul comportamento abilitati e calibrati.

**Descrizione:** L'Amministratore può attivare e configurare la modalità di rilevamento comportamentale e da stress, che consente al sistema di apprendere il comportamento normale del traffico applicativo e identificare deviazioni sospette nel tempo. Questa funzionalità è utile per intercettare attacchi sofisticati e distribuiti, non sempre rilevabili con soglie fisse. Il rilevamento può includere analisi del carico, durata delle connessioni e altri *pattern* anomali.



**Figura 3.4:** Use Case 1.10

### UC1.10: Configurazione dei profili di *bot defense*

**Attore Principale:** Amministratore

**Precondizioni:** Il sistema *WAF* deve supportare la funzionalità di protezione contro i *bot*.

**Postcondizioni:** Profilo di *bot defense* configurato e applicato alla *policy* desiderata.

**Descrizione:** L'Amministratore può configurare uno o più profili di *bot defense*,

definendo le strategie di rilevamento e mitigazione nei confronti del traffico generato da *bot* automatizzati. Questa funzionalità consente di distinguere tra *bot* benigni e *bot* malevoli (es. *spammer*, attori fraudolenti), e di applicare contromisure adeguate come *CAPTCHA* o blocco diretto. Il profilo può essere personalizzato in base a categorie di *bot*, reputazione *IP* e comportamenti sospetti. Una volta creato, il profilo può essere associato a una *policy* attiva per proteggere specifiche aree dell'applicazione.

#### **UC1.10.1: Configurazione generale del profilo di *bot defense***

**Attore Principale:** Amministratore

**Precondizioni:** Il modulo *bot defense* deve essere attivo nel sistema.

**Postcondizioni:** Impostazioni generali del profilo configurate correttamente.

**Descrizione:** L'Amministratore può accedere alle impostazioni generali di un profilo di *bot defense*, dove può definire il nome del profilo, lo stato di attivazione, il tipo di traffico da analizzare e il comportamento predefinito da adottare nei confronti dei *bot* non classificati. Questa sezione consente inoltre di abilitare il *logging* e configurare opzioni generali di integrazione con le *policy*. La configurazione generale rappresenta la base su cui vengono applicate le altre regole di rilevamento e mitigazione.

#### **UC1.10.2: Configurazione delle strategie di mitigazione *bot***

**Attore Principale:** Amministratore

**Precondizioni:** Il profilo *bot defense* deve essere già stato creato e attivo.

**Postcondizioni:** Azioni di mitigazione aggiornate per ciascuna categoria di *bot*.

**Descrizione:** L'Amministratore può configurare le azioni da intraprendere nei confronti dei *bot*, in base alla loro classificazione (benigni, sospetti, malevoli). È possibile specificare contromisure come blocco, *CAPTCHA*, redirezione, oppure semplici *log*. La sezione consente inoltre di impostare soglie di attivazione e criteri avanzati per la risposta automatizzata. Questa funzionalità è fondamentale per bilanciare sicurezza ed esperienza utente.

#### **UC1.10.3: Configurazione della protezione dei microservizi**

**Attore Principale:** Amministratore

**Precondizioni:** Il traffico applicativo deve includere microservizi da proteggere.

**Postcondizioni:** Regole di protezione per i microservizi attivate.

**Descrizione:** L'Amministratore può attivare la protezione specifica per microservizi all'interno del profilo di *bot defense*, definendo *URL* o *pattern* relativi a microservizi che necessitano di una protezione distinta. Questa funzionalità consente di personalizzare l'approccio difensivo per *endpoint* più esposti a *Brute Force* o analisi automatizzate. Sono disponibili tecniche mirate per identificare anomalie nei flussi di richieste tra *client* e servizi.

#### **UC1.10.4: Gestione dei *browser***

**Attore Principale:** Amministratore

**Precondizioni:** Il profilo *bot defense* deve essere attivo e in fase di *tuning*.

**Postcondizioni:** Comportamento dei *browser* classificati configurato correttamente.

**Descrizione:** L'Amministratore può definire come il sistema deve gestire le richieste provenienti da *browser* conosciuti. È possibile classificare specifici *browser* come attendibili, sospetti o bloccati, e associare loro comportamenti di risposta mirati. Questa configurazione consente di gestire con maggiore precisione i *client* legittimi rispetto ai *bot* mascherati da *browser* comuni.

#### UC1.10.5: Gestione delle firme di rilevamento *bot*

**Attore Principale:** Amministratore

**Precondizioni:** Il modulo *bot defense* deve essere aggiornato con firme attive.

**Postcondizioni:** Firme di *bot* selezionate, attivate o disattivate secondo *policy*.

**Descrizione:** L'Amministratore può consultare l'elenco delle firme di rilevamento *bot* e decidere quali abilitare nel profilo corrente. Le firme rappresentano *pattern* noti di comportamento o identità di *bot*, riconosciuti tramite *fingerprinting* e altre tecniche. È possibile attivare o disattivare specifiche firme per ridurre i falsi positivi o adattare la protezione al contesto applicativo.

#### UC1.11: Visualizzazione dei *log* delle richieste applicative

**Attore Principale:** Amministratore

**Precondizioni:** Almeno una *policy* deve essere attiva e applicata al traffico.

**Postcondizioni:** Richieste registrate consultate tramite il modulo di *logging* applicativo.

**Descrizione:** L'Amministratore può accedere alla sezione dedicata ai *log* delle richieste applicative, dove può visualizzare le interazioni *HTTP* intercettate dal sistema *WAF*, comprese quelle che hanno generato violazioni, sono state messe in evidenza oppure hanno attivato firme di attacco. Il *log* mostra informazioni dettagliate come indirizzo *IP* sorgente, *URL* richiesto, tipo di violazione e punteggio di gravità. Questa funzionalità è fondamentale per eseguire analisi post-evento, diagnosticare falsi positivi e verificare il comportamento delle *policy* di sicurezza applicate.

#### UC1.12: Visualizzazione degli eventi *DoS* a livello applicativo

**Attore Principale:** Amministratore

**Precondizioni:** Deve essere attivo almeno un profilo di protezione *DoS* applicato a una *policy*.

**Postcondizioni:** Eventi *DoS* rilevati dal sistema consultati tramite il modulo di *logging*.

**Descrizione:** L'Amministratore può visualizzare gli eventi *DoS* registrati a livello applicativo, ovvero tutte le occorrenze in cui il sistema *WAF* ha rilevato comportamenti riconducibili a un attacco di tipo *DoS*. Il *log* mostra informazioni dettagliate sulle soglie superate, sulle contromisure attivate (blocco, *CAPTCHA*, ecc.), sugli indirizzi *IP* coinvolti e sull'impatto dell'evento. Questa funzionalità è utile per analizzare l'efficacia dei profili *DoS* configurati e per identificare eventuali *pattern* ricorrenti di traffico malevolo.

**UC1.13: Visualizzazione delle richieste intercettate dalla *bot defense***

**Attore Principale:** Amministratore

**Precondizioni:** Deve essere attivo almeno un profilo di *bot defense* associato a una *policy*.

**Postcondizioni:** Richieste sospette riconducibili a *bot* visualizzate tramite il modulo di *logging*.

**Descrizione:** L'Amministratore può consultare il registro delle richieste classificate come generate da *bot*, visualizzando le informazioni rilevate dal sistema *WAF* in fase di analisi comportamentale o reputazionale. Il *log* include dettagli su tipo di *bot* rilevato (benigno, sospetto o malevolo), indirizzo *IP* e azione intrapresa (come blocco, *CAPTCHA* o redirezione). Questa funzionalità consente di monitorare l'efficacia delle contromisure di *bot defense* e di individuare *pattern* anomali nel traffico automatizzato.

**UC1.14: Gestione dei profili di *logging***

**Attore Principale:** Amministratore

**Precondizioni:** Devono essere presenti una o più *policy* a cui poter associare un profilo di *logging*.

**Postcondizioni:** Profilo di *logging* configurato o aggiornato secondo le impostazioni desiderate.

**Descrizione:** L'Amministratore può visualizzare e gestire i profili di *logging* del sistema *WAF*, definendo le modalità con cui vengono registrati gli eventi di sicurezza. Ogni profilo consente di specificare il formato di *output* dei *log*, i dati da includere, la destinazione del *logging* (locale o remota) e le condizioni che attivano la registrazione. Questa funzionalità è essenziale per integrare il *WAF* con strumenti esterni di monitoraggio o correlazione degli eventi, e per assicurare una tracciabilità coerente delle violazioni e del traffico rilevante.

**UC1.15: Visualizzazione del *report* di conformità *PCI***

**Attore Principale:** Amministratore

**Precondizioni:** Deve essere attiva almeno una *policy* di sicurezza applicativa.

**Postcondizioni:** *Report Payment Card Industry (PCI)* generato e visualizzato.

**Descrizione:** L'Amministratore può generare e visualizzare il *report* di conformità *PCI*, che riassume il grado di copertura delle *policy* di sicurezza rispetto ai requisiti definiti dallo standard *PCI*. Il *report* include informazioni su firme di attacco abilitate, tecniche di protezione attive e componenti della *policy* che contribuiscono alla protezione dei dati sensibili. Questa funzionalità è utile per verificare la conformità del sistema agli standard di sicurezza richiesti in contesti che gestiscono dati di pagamento.

**UC1.16: Visualizzazione della dashboard *DoS***

**Attore Principale:** Amministratore

**Precondizioni:** Deve essere configurato almeno un profilo di protezione *DoS*.

**Postcondizioni:** *Dashboard* aggiornata con i dati relativi agli eventi *DoS*.

**Descrizione:** L'Amministratore può accedere alla *dashboard* dedicata agli attacchi *DoS*, dove vengono visualizzati in tempo reale i dati relativi alle violazioni rilevate, alle soglie superate, e alle azioni di mitigazione applicate. La *dashboard* mostra grafici, *trend* e metriche aggregate utili per comprendere l'efficacia delle protezioni attive, individuare *pattern* di attacco ricorrenti e prendere decisioni informate sul *tuning* dei profili *DoS*.

### UC1.17: Gestione delle firme di attacco

**Attore Principale:** Amministratore

**Precondizioni:** Il modulo di protezione applicativa deve essere attivo nel sistema *WAF*.

**Postcondizioni:** Firme di attacco visualizzate, aggiornate o personalizzate.

**Descrizione:** L'Amministratore può consultare e gestire l'elenco delle firme di attacco utilizzate dal sistema *WAF* per rilevare tentativi malevoli come *SQLi*, *XSS* e altre tecniche note. Questa funzionalità consente di visualizzare il dettaglio di ciascuna firma, abilitarla o disabilitarla in modo selettivo, e creare firme personalizzate se necessario. La gestione delle firme permette di adattare la protezione alle specificità dell'applicazione protetta, riducendo i falsi positivi senza compromettere la sicurezza.

### UC1.18: Configurazione delle variabili di sistema

**Attore Principale:** Amministratore

**Precondizioni:** Il sistema *WAF* deve essere correttamente configurato e operativo.

**Postcondizioni:** Valori delle variabili di sistema aggiornati secondo le nuove impostazioni.

**Descrizione:** L'Amministratore può accedere alla configurazione delle variabili di sistema del modulo di sicurezza applicativa, modificando impostazioni avanzate che influiscono sul comportamento globale del *WAF*. Tra le variabili configurabili rientrano, ad esempio, limiti dimensionali per parametri e *URLs*, opzioni per la gestione dei caratteri *jolly (wildcard)*, e soglie generali per la rilevazione degli attacchi. Questa funzionalità consente un *tuning* profondo del sistema, utile in scenari avanzati o per ambienti ad alta personalizzazione.

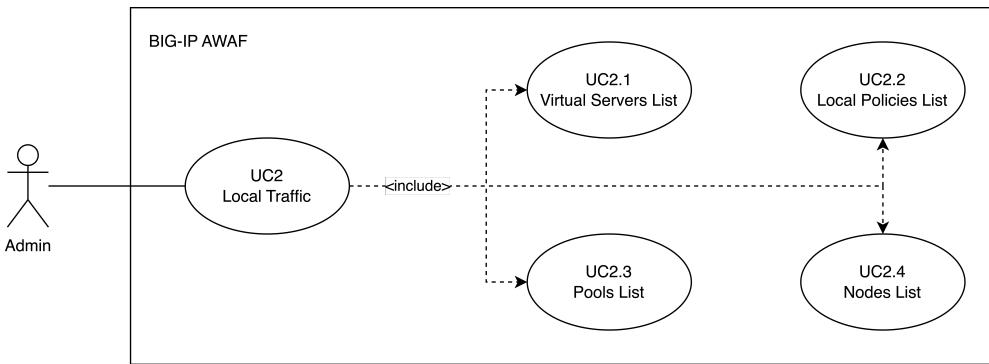
## UC2: Accesso e gestione della sezione *Local Traffic*

**Attore Principale:** Amministratore

**Precondizioni:** L'Amministratore deve essere autenticato nel sistema *WAF*.

**Postcondizioni:** Accesso alla sezione *Local Traffic* effettuato e funzionalità disponibili.

**Descrizione:** L'Amministratore accede alla sezione *Local Traffic*, dove può configurare gli elementi fondamentali per la gestione del bilanciamento del carico e del traffico applicativo. Da qui è possibile visualizzare e modificare *Virtual Server*, *pool*, nodi, e *policy* locali, definendo il comportamento del traffico in ingresso. Questo rappresenta il punto di partenza per la configurazione logica della rete applicativa.

**Figura 3.5:** Use Case 2

### UC2.1: Gestione dei *Virtual Server*

**Attore Principale:** Amministratore

**Precondizioni:** Il sistema deve essere correttamente configurato a livello di rete.

**Postcondizioni:** *Virtual Server* visualizzati e modificati secondo necessità.

**Descrizione:** L'Amministratore può visualizzare la lista dei *Virtual Server* configurati, modificarne i parametri, crearli o rimuoverli. Ogni *Virtual Server* rappresenta un punto di accesso logico per il traffico in ingresso ed è associato a una configurazione che ne determina il comportamento (protocollo, indirizzo *IP*, porta, *policy* applicate, profili di sicurezza). Questa funzionalità consente di definire come il sistema gestisce le connessioni *client*.

### UC2.2: Gestione delle *policy* di traffico locale

**Attore Principale:** Amministratore

**Precondizioni:** Almeno un *Virtual Server* deve essere già configurato.

**Postcondizioni:** Le *policy* locali verranno visualizzate o aggiornate.

**Descrizione:** L'Amministratore può accedere alla lista delle *policy* locali, che definiscono regole dinamiche per la gestione delle richieste in ingresso. Le *policy* possono essere basate su condizioni (es. *header*, indirizzi *IP*, *URLs*) e prevedere azioni come reindirizzamento, riscrittura o inoltro a *pool* diversi. Questa funzionalità consente di gestire comportamenti personalizzati senza modificare il codice applicativo.

### UC2.3: Gestione dei *pool*

**Attore Principale:** Amministratore

**Precondizioni:** Devono essere presenti almeno uno o più nodi configurati.

**Postcondizioni:** Lista dei *pool* visualizzati, creati o modificati.

**Descrizione:** L'Amministratore può visualizzare, creare o modificare i *pool*. Per ciascuno di essi è possibile definire criteri di bilanciamento del carico, *monitor* di disponibilità e metodi di selezione dei nodi. Questa funzionalità è fondamentale per assicurare la scalabilità e la disponibilità dell'applicazione.

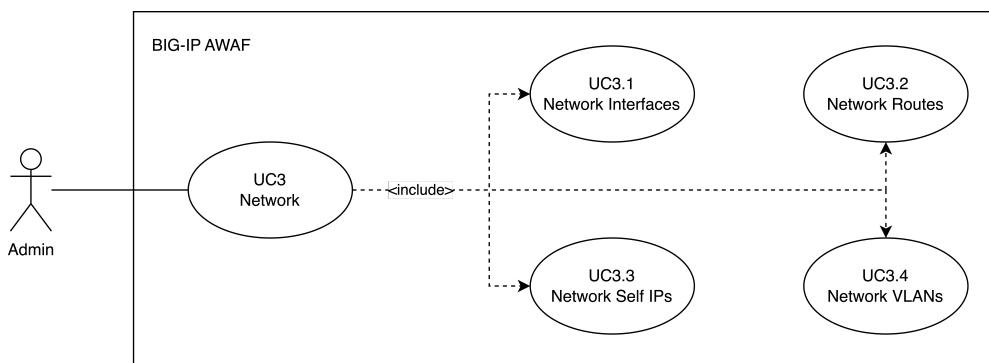
### UC2.4: Gestione dei nodi

**Attore Principale:** Amministratore

**Precondizioni:** La configurazione di rete deve essere correttamente impostata.

**Postcondizioni:** Nodi visualizzati e configurati.

**Descrizione:** L'Amministratore può visualizzare e gestire l'elenco dei nodi, ovvero gli indirizzi *IP* dei *server* fisici o virtuali che fanno parte del *pool*. Ogni nodo può essere monitorato per verificarne lo stato, la disponibilità e la reattività. Questa funzionalità consente di gestire le risorse in modo centralizzato, intervenendo in caso di malfunzionamenti o necessità di manutenzione.



**Figura 3.6:** Use Case 3

### UC3: Accesso e gestione della sezione *Network*

**Attore Principale:** Amministratore

**Precondizioni:** L'Amministratore deve avere accesso ai privilegi di rete.

**Postcondizioni:** Funzionalità di rete accessibili e configurabili.

**Descrizione:** L'Amministratore può accedere alla sezione *Network* del sistema, che consente di visualizzare e modificare la configurazione di basso livello dell'interfaccia di rete, degli indirizzi *IP*, delle *Virtual Local Area Network (VLAN)* e delle *routes*. Questa sezione rappresenta la base infrastrutturale per il funzionamento del sistema e l'instradamento corretto del traffico.

#### UC3.1: Gestione delle interfacce di rete

**Attore Principale:** Amministratore

**Precondizioni:** Accesso alla sezione *Network* abilitato.

**Postcondizioni:** Stato e configurazione delle interfacce aggiornati.

**Descrizione:** L'Amministratore può visualizzare lo stato e i dettagli di ogni interfaccia di rete fisica o virtuale presente nel sistema. È possibile monitorare velocità e configurare parametri avanzati. Questa funzionalità consente di assicurare una connettività stabile tra i componenti della rete.

#### UC3.2: Gestione delle *routes*

**Attore Principale:** Amministratore

**Precondizioni:** Il sistema deve essere connesso a più reti.

**Postcondizioni:** Lista delle *routes* visualizzata e configurata.

**Descrizione:** L'Amministratore può aggiungere, modificare o rimuovere *routes* nella configurazione di rete, definendo il *gateway* per il traffico diretto a reti specifiche. Questa funzionalità è necessaria per garantire il corretto instradamento del traffico tra reti interne ed esterne, soprattutto in ambienti complessi o segmentati.

### UC3.3: Gestione degli indirizzi *IP* del sistema (*Self IPs*)

**Attore Principale:** Amministratore

**Precondizioni:** Le interfacce di rete devono essere già configurate.

**Postcondizioni:** *Self IPs* configurato e associato a una *VLAN*.

**Descrizione:** L'Amministratore può definire uno o più *Self IPs*, ovvero indirizzi *IP* assegnati direttamente al sistema e utilizzati per la comunicazione con gli altri dispositivi nella rete. Ogni *Self IP*s è associato a una *VLAN* e può includere regole locali. Questa configurazione è fondamentale per consentire al sistema di ricevere e inviare traffico nella rete definita.

### UC3.4: Gestione delle *VLAN*

**Attore Principale:** Amministratore

**Precondizioni:** Almeno un'interfaccia di rete deve essere disponibile.

**Postcondizioni:** *VLAN* configurate e assegnate alle interfacce.

**Descrizione:** L'Amministratore può creare e gestire le *VLAN* nel sistema, definendo *ID*, nomi e interfacce associate. La segmentazione del traffico in *VLAN* consente di separare logicamente ambienti di rete diversi e di applicare regole di sicurezza differenziate. Questa funzionalità è fondamentale per la scalabilità e l'isolamento delle architetture.

## 3.2 Tracciamento dei requisiti

L'analisi dei requisiti rappresenta una fase fondamentale nello sviluppo e nella valutazione di un progetto, in quanto consente di individuare, classificare e formalizzare le funzionalità richieste, le caratteristiche desiderate e i vincoli tecnologici che guidano la realizzazione del sistema. In questo progetto, l'obiettivo principale è stato quello di configurare e verificare il funzionamento del modulo *Application Security Manager (ASM)* del sistema *BIG-IP* di *F5*, esplorando in modo sistematico le sue funzionalità attraverso casi d'uso concreti.

Per una visione strutturata e completa, i requisiti sono stati tracciati in corrispondenza diretta con i casi d'uso presentati nel capitolo precedente. Tale corrispondenza permette di validare la copertura funzionale delle operazioni implementate e analizzate, e di facilitare future attività di verifica, manutenzione o estensione del sistema.

I requisiti sono stati classificati secondo la seguente codifica alfanumerica:

- **R** = Requisito
- **F** = Funzionale
- **Q** = Qualità
- **V** = Di vincolo
- **N** = Necessario (Obbligatorio)
- **D** = Desiderabile
- **Z** = Facoltativo

Il codice risultante assume quindi la forma **R(F/Q/V)(N/D/Z)**.

Nelle tabelle 3.1, 3.2 e 3.3 riportate nelle pagine seguenti, vengono elencati i requisiti individuati, ciascuno associato ai casi d'uso che ne dimostrano l'effettiva realizzazione o osservazione durante l'attività di analisi.

**Tabella 3.1:** Tabella del tracciamento dei requisiti funzionali

Requisito	Descrizione	Use Case
RFN-1	Configurare impostazioni generali di una <i>policy</i> .	UC1.2.1
RFN-2	Gestire firme di attacco di una <i>policy</i> .	UC1.2.2
RFN-3	Abilitare protezione contro campagne di attacco note.	UC1.2.3
RFN-4	Applicare protezione <i>IP Intelligence</i> di una <i>policy</i> .	UC1.2.4
RFN-5	Definire eccezioni sugli indirizzi <i>IP</i> .	UC1.2.5
RFN-6	Configurare tracciamento delle sessioni e dei <i>login</i> .	UC1.2.6
RFN-7	Personalizzare pagine di blocco e risposta.	UC1.2.7
RFN-8	Gestire tipi di <i>file</i> da monitorare o bloccare.	UC1.2.8
RFN-9	Configurare analisi degli <i>header</i> nella <i>policy</i> .	UC1.2.9
RFN-10	Gestire e proteggere <i>cookie</i> .	UC1.2.10
RFN-11	Attivare protezione dei dati sensibili.	UC1.2.11
RFN-12	Configurare protezione contro attacchi <i>Brute Force</i> .	UC1.2.12
RFN-13	Analizzare traffico appreso.	UC1.3
RFN-14	Configurare impostazioni di apprendimento e blocco.	UC1.4
RFN-15	Gestire <i>URL</i> consentiti.	UC1.5
RFN-16	Consultare e modificare parametri <i>HTTP</i> .	UC1.6
RFN-17	Configurare tracciamento delle sessioni.	UC1.7
RFN-18	Visualizzare lista <i>policy</i> di <i>IP Intelligence</i> .	UC1.8
RFN-19	Configurare profilo di protezione <i>DoS</i> .	UC1.9.1
RFN-20	Definire soglie <i>TPS</i> nei profili <i>DoS</i> .	UC1.9.2
RFN-21	Abilitare rilevamento comportamentale nei profili <i>DoS</i> .	UC1.9.3
RFN-22	Definire impostazioni generali per profili di <i>bot defense</i> .	UC1.10.1
RFN-23	Configurare mitigazioni per le diverse categorie di <i>bot</i> .	UC1.10.2
RFN-24	Attivare protezione avanzata per <i>endpoint</i> .	UC1.10.3
RFN-25	Gestire <i>browser</i> per rilevamento di anomalie.	UC1.10.4
RFN-26	Selezionare e aggiornare firme di rilevamento <i>bot</i> .	UC1.10.5
RFN-27	Consultare <i>log</i> relativi a richieste applicative.	UC1.11
RFN-28	Visualizzare eventi <i>DoS</i> a livello applicativo.	UC1.12
RFN-29	Analizzare richieste classificate come <i>bot</i> tramite <i>log</i> .	UC1.13
RFN-30	Configurare profili di <i>logging</i> .	UC1.14
RFN-31	Consultare firme di attacco e gestirne attivazione.	UC1.17
RFN-32	Gestire variabili di sistema per configurazione avanzata.	UC1.18
RFN-33	Visualizzare e gestire lista <i>Virtual Server</i> .	UC2.1
RFN-34	Consultare e gestire <i>policy</i> di traffico locale.	UC2.2
RFN-35	Gestire <i>pool</i> per bilanciamento traffico.	UC2.3
RFN-36	Gestire nodi <i>backend</i> .	UC2.4
RFN-37	Configurare interfacce di rete.	UC3.1
RFN-38	Gestire <i>routes</i> statiche.	UC3.2
RFN-39	Assegnare e configurare indirizzi <i>IP</i> del sistema.	UC3.3
RFN-40	Gestire <i>VLAN</i> per segmentazione rete.	UC3.4
RFD-1	Esportare o importare configurazioni <i>policy</i> tramite <i>file</i> .	-
RFZ-1	Inviare automaticamente <i>log</i> a sistema <i>SIEM</i> esterno.	-
RFZ-2	Applicare filtri geografici al traffico.	-
RFZ-3	Integrare <i>WAF</i> con <i>cloud</i> per gestione o <i>backup</i> .	-

**Tabella 3.2:** Tabella del tracciamento dei requisiti qualitativi

Requisito	Descrizione	Use Case
RQN-1	Visualizzare conformità <i>OWASP</i> delle <i>policy</i> .	UC1.1
RQD-1	Accedere a <i>report</i> di conformità <i>PCI</i> .	UC1.15
RQD-2	Visualizzare <i>dashboard</i> <i>DoS</i> per monitoraggio.	UC1.16

**Tabella 3.3:** Tabella del tracciamento dei requisiti di vincolo

Requisito	Descrizione	Use Case
RVN-1	Impiegare piattaforma <i>BIG-IP</i> di <i>F5</i> .	-
RVN-2	Garantire conformità agli standard <i>OWASP</i> e <i>PCI</i> .	UC1.1
RVZ-1	Abilitare supporto opzionale via <i>API REST</i> o <i>CLI</i> .	-

# Acronimi e abbreviazioni

- API REST** Representational State Transfer Application Programming Interface. 30
- ASM** Application Security Manager. 23, 27
- CLI** Command Line Interface. 25, 28
- DoS** Denial of Service. 1, 14, 15, 17–19, 24, 25, 28
- HTML** HyperText Markup Language. 10, 29, 32
- HTTP** HyperText Transfer Protocol. 10, 11, 15, 17, 24, 27–30, 32
- HTTPS** HyperText Transfer Protocol Secure. 27, 29, 32
- IP** Internet Protocol. 9, 10, 12, 13, 15–18, 20–22, 24, 28, 29, 31
- JS** JavaScript. 29, 32
- OWASP** Open Worldwide Application Security Project. 7, 25, 29
- PCI** Payment Card Industry. 18, 25, 30
- SIEM** Security Information and Event Management. 24, 30
- SQL** Structured Query Language. 30
- SQLi** SQL injection. 1, 4, 19, 27, 30, 32
- TPS** Transactions Per Second. 14, 15, 24, 31
- VLAN** Virtual Local Area Network. 21, 22, 24, 31
- VM** Virtual Machine. 3, 4, 28, 31
- WAF** Web Application Firewall. 1–5, 7–9, 12–15, 17–19, 24, 27–32
- WWW** World Wide Web. 29, 32
- XSS** Cross-Site Scripting. 1, 4, 19, 27, 32

# Glossario

**ASM** modulo della piattaforma *BIG-IP* di *F5* progettato per offrire protezione a livello applicativo. *ASM* funge da *WAF*, fornendo funzionalità avanzate come firme di attacco, *IP Intelligence*, protezione da *XSS*, *SQLi* e gestione delle *policy* per difendere le *web application* da minacce note e sconosciute.. [26](#)

**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*.. [3](#), [23](#), [25](#), [27](#), [28](#)

**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](#), [9](#), [10](#)

**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](#), [15–18](#), [24](#), [27](#)

**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](#), [11](#), [12](#), [16](#), [24](#)

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

**Cache** meccanismo di memorizzazione temporanea di dati, utilizzato per ridurre i tempi di accesso e il carico sulle risorse di rete o di calcolo. Nel contesto delle applicazioni *web*, la *cache* consente di evitare richieste ripetitive verso il *server* memorizzando le risposte *HTTP* localmente sul *client* o su nodi intermedi. [28](#)

**CAPTCHA** acronimo di *Completely Automated Public Turing test to tell Computers and Humans Apart*, è un meccanismo di sicurezza utilizzato per distinguere gli utenti umani da programmi automatizzati (*bot*). I *CAPTCHA* vengono spesso impiegati nei moduli *web* per prevenire accessi automatizzati, abusi o *spam*. [12](#), [14](#), [16–18](#)

**CLI** interfaccia testuale che consente all'utente di interagire con un sistema operativo o un'applicazione attraverso comandi digitati da tastiera. La *CLI* è spesso utilizzata per configurazioni avanzate, automazione e operazioni rapide rispetto alle interfacce grafiche. [26](#)

**Cookie** piccolo *file* di testo che un *server* invia al *client* (solitamente un *browser*) per memorizzare informazioni di stato o preferenze dell'utente. I *cookie* sono utilizzati per sessioni di autenticazione, tracciamento delle attività di navigazione e personalizzazione dei contenuti. [11](#), [24](#)

**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. [26](#)

**Endpoint** punto di accesso a una risorsa o a un servizio disponibile su una rete. Nel contesto di questo progetto, rappresenta un *URL* specifico al quale un *client* può inviare richieste per eseguire operazioni su dati o funzionalità esposte da un *server*. [12](#), [16](#), [24](#)

**Enforcement** processo attraverso il quale una regola o una politica viene applicata effettivamente in un sistema. Nel contesto della sicurezza applicativa, l'*enforcement* delle *policy* da parte di un *WAF* implica il blocco o la modifica delle richieste che violano determinate regole. [9](#), [11](#), [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 *WAF*. [1–4](#), [23](#), [25](#), [27](#), [30](#)

**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](#)

**Gateway** dispositivo o nodo di rete che funge da punto di accesso tra reti diverse. Un *gateway* consente la comunicazione tra una rete locale e altre reti, come *Internet*, ed è spesso l'elemento che instrada il traffico verso l'esterno in base alle *routes* configurate. [22](#), [30](#)

**Header** insieme di informazioni aggiuntive incluse in una richiesta o risposta *HTTP*. Gli *header* specificano dettagli come il tipo di contenuto, la lunghezza, le credenziali di autenticazione o il comportamento della *cache*. Sono fondamentali per il funzionamento dei protocolli *web* e la gestione della comunicazione tra *client* e *server*. [11](#), [20](#), [24](#)

**Host** dispositivo o nodo identificabile in rete, dotato di un proprio indirizzo *IP*, che può inviare o ricevere dati. In una rete informatica, un *host* può essere un *computer*, un *server*, una *VM* o qualsiasi altro sistema capace di comunicare tramite protocolli di rete. [30](#)

**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. [26](#)

**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*. [26](#)

**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*. [26](#)

**IP** *Internet Protocol*, protocollo di comunicazione utilizzato per l'inoltro e l'instradamento dei pacchetti dati attraverso le reti informatiche. Ogni dispositivo connesso a una rete basata su *IP* è identificato da un indirizzo univoco denominato indirizzo *IP*. [26](#)

**IP Intelligence** tecnica di analisi che consiste nel raccogliere e utilizzare informazioni contestuali su indirizzi *IP* (es. reputazione, geolocalizzazione, attività sospette) per prendere decisioni di sicurezza. Nei *WAF*, l'*IP Intelligence* consente di bloccare richieste provenienti da *IP* noti per attività malevole o da aree geografiche a rischio. [9, 13, 24, 27](#)

**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 *HTML*. [26](#)

**Juice Shop** *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, 16–18, 24, 29–31](#)

**Logging** processo di registrazione delle attività e degli eventi che si verificano all'interno di un sistema. Il *logging* è fondamentale per l'analisi dei problemi, la sicurezza e il monitoraggio delle applicazioni. Nei *WAF*, i *log* vengono utilizzati per analizzare il traffico e rilevare eventuali attacchi o anomalie. [12–14, 16–18, 24](#)

**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*. [26](#)

**PCI** settore che comprende tutte le organizzazioni coinvolte nell'elaborazione, trasmissione e archiviazione di dati relativi a carte di pagamento. Il termine è spesso associato al *PCI DSS* (*Payment Card Industry Data Security Standard*), uno standard di sicurezza volto a proteggere i dati delle carte di credito. [26](#)

**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](#), [7–20](#), [24](#), [25](#), [27](#), [28](#), [30](#), [31](#)

**Pool** insieme logico di *server* o *host* che collaborano per erogare un servizio condiviso. Nel contesto del *WAF* di [F5](#), una *pool* rappresenta un gruppo di nodi che ricevono richieste dal *Virtual Server* e le gestiscono secondo criteri di bilanciamento del carico. [19–21](#), [24](#)

**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*. [12](#), [30](#)

**API REST** una *Representational State Transfer Application Programming Interface* viene utilizzata per realizzare comunicazioni tra sistemi via *HTTP*, in particolare consente di accedere a risorse e servizi esposti da un'applicazione tramite operazioni *standard* come i principali metodi *HTTP*. [25](#), [26](#)

**Routes** insieme di regole di instradamento che determinano il percorso seguito dai pacchetti all'interno di una rete. Ogni *route* specifica una destinazione, una *subnet*, un *gateway* e, facoltativamente, un'interfaccia di rete. Le *routes* sono essenziali per la comunicazione tra reti diverse. [21](#), [22](#), [24](#), [28](#)

**Session Cookie** *file* temporaneo salvato nel *browser* dell'utente contenente dati relativi alla sessione di navigazione corrente. I *session cookie* vengono utilizzati per mantenere lo stato dell'utente tra richieste successive (es. autenticazione, carrello acquisti) e vengono eliminati automaticamente alla chiusura del *browser*. [10](#)

**SIEM** sistema che centralizza, raccoglie e analizza i dati di *log* e gli eventi di sicurezza generati da dispositivi, applicazioni e infrastrutture *IT*. Un *SIEM* consente il monitoraggio in tempo reale, la correlazione degli eventi e l'identificazione di minacce o anomalie per supportare la risposta agli incidenti e la conformità normativa. [26](#)

**SQL** *Structured Query Language*, linguaggio *standard* utilizzato per l'interrogazione, la manipolazione e la definizione di dati all'interno di un *database* relazionale. [26](#)

**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. [26](#)

**Staging** fase intermedia di test in cui le modifiche o le configurazioni vengono applicate in un ambiente controllato e non ancora attivo in produzione. Nel contesto delle *policy* di sicurezza, lo *staging* consente di valutare l'impatto delle regole senza bloccare il traffico reale. [9–11](#), [13](#)

**Subnet** suddivisione logica di una rete *IP* più ampia, utilizzata per migliorare l'organizzazione, la sicurezza e la gestione del traffico di rete. Ogni *subnet* ha un proprio intervallo di indirizzi *IP* e una *subnet mask* che definisce la dimensione della rete. [10](#), [15](#), [30](#)

**Threat Actors** individui, gruppi o entità che compiono, o hanno la potenzialità di compiere, azioni dannose contro sistemi informatici o reti. I *threat actors* possono includere *hacker*, cybercriminali o *insider* malintenzionati, ciascuno con propri obiettivi e risorse. [9](#)

**TPS** metrica utilizzata per misurare il numero di operazioni o transazioni completate da un sistema in un secondo. Nel contesto dei *WAF* o delle applicazioni *web*, il valore di *TPS* è indicativo del carico di lavoro sostenibile e dell'efficienza del sistema sotto stress. [26](#)

**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](#), [9](#), [27](#)

**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](#), [7](#), [12](#), [13](#), [17](#), [19](#), [31](#)

**Ubuntu Client** installazione *desktop* di Ubuntu, dotata di interfaccia grafica e pensata per l'utilizzo da parte dell'utente finale. Nel contesto del progetto, viene utilizzata come macchina di accesso per interagire con l'interfaccia del *WAF* tramite *browser web*. [4](#)

**Ubuntu Server** versione di Ubuntu ottimizzata per l'utilizzo su *server*, priva di interfaccia grafica e progettata per essere stabile, sicura ed efficiente nelle operazioni di *networking*, virtualizzazione e gestione dei servizi *web*. [3](#)

**Virtual Server** *Virtual Server*, configurazione logica in un *WAF* o in un sistema di bilanciamento del carico che permette di esporre un indirizzo *IP* e una porta a cui indirizzare il traffico in ingresso, inoltrandolo poi verso uno o più *server* reali secondo regole predefinite.. [14](#), [19](#), [20](#), [24](#), [30](#)

**VLAN** tecnica di segmentazione della rete che consente di suddividere una rete fisica in più reti logiche isolate. Le *VLAN* migliorano la sicurezza e la gestione del traffico separando i dispositivi in base alla funzione, alla posizione o ad altri criteri, anche se fisicamente collegati alla stessa infrastruttura. [26](#)

**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. [26](#)

**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* e altri attacchi a livello applicativo. [26](#)

**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*. [26](#)

**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. [26](#)