МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
КІБЕРБЕЗПЕКИ ТА ЗАХИСТУ ІНФОРМАЦІЇ
КАФЕДРА ТЕХНІЧНИХ СИСТЕМ КІБЕРЗАХИСТУ

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

“Захист веб-застосунків за допомогою WEB APPLICATION FIREWALL на базі рішення F5 NETWORKS ”

на здобуття освітнього ступеня магістра
зі спеціальності *125 Кібербезпека та захист інформації*
освітньої програми Технічних систем інформаційного та кібернетичного захисту
(назва програми)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ Микита М'ЯСНИКОВ
(підпис) *Ім’я, ПРІЗВИЩЕ здобувача*

Виконав: здобувач повної вищої освіти групи СЗДМ-61
Микита М'ЯСНИКОВ
Ім’я, ПРІЗВИЩЕ

Керівник: Ігор ІВАНЧЕНКО
к. т. н., доцент Ім’я, ПРІЗВИЩЕ

Рецензент: _____

Київ 2025

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ КІБЕРБЕЗПЕКИ ТА ЗАХИСТУ ІНФОРМАЦІЇ

Кафедра Технічних Систем Кіберзахисту

Ступінь вищої освіти Магістр

Спеціальність 125 Кібербезпека

Освітня програма Технічні системи інформаційного та кібернетичного захисту

ЗАТВЕРДЖУЮ
Завідувач кафедри ТСКБ
_____Олександр Туровський
“___” _____ 2025 року

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

М'ясников Микита Сергійович
(прізвище, ім’я)

1. Тема кваліфікаційної роботи: “ Захист веб-застосунків за допомогою Web Application Firewall на базі рішення F5 Networks”
керівник кваліфікаційної роботи *к. т. н., доц.*, Ігор ІВАНЧЕНКО
(прізвище, ім’я, науковий ступінь, вчене звання)
затверджені наказом Державного університету інформаційно-комунікаційних технологій від «30» жовтня 2025 року № 467
2. Строк подання кваліфікаційної роботи 20.12.2025 р.
3. Вихідні дані до кваліфікаційної роботи: сучасні методи захисту веб-застосунків від кіберзагроз; принципи роботи та архітектура Web Application Firewall на базі платформи F5 BIG-IP; наукова та технічна література, експлуатаційна документація виробника, міжнародні стандарти та рекомендації з кіберзахисту.
4. Перелік питань, які мають бути розроблені
4.1. Аналіз сучасних кіберзагроз і типових атак на веб-застосунки.
4.2. Аналіз існуючих підходів і засобів захисту веб-застосунків, зокрема із застосуванням Web Application Firewall.
4.3. Розроблення та практичне впровадження рекомендацій щодо налаштування та використання Web Application Firewall на базі рішення F5 Networks для підвищення рівня безпеки веб-застосунків.
5. Перелік ілюстративного матеріалу: *презентація PowerPoint*
6. Дата видачі завдання: “15” листопада 2025р.

КАЛЕНДАРНИЙ ПЛАН

№ зп	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Визначення актуальності проблеми.	11.11.2025 р.	
2.	Аналіз наукової та технічної літератури з питань теми кваліфікаційної роботи.	13.11.2025 р.	
3.	Аналіз сучасних кіберзагроз і типових атак на веб-застосунки.	18.11.2025 р.	
4.	Аналіз існуючих підходів і засобів захисту веб-застосунків, зокрема із застосуванням Web Application Firewall.	27.11.2025 р.	
5.	Розроблення та практичне впровадження рекомендацій щодо налаштування та використання Web Application Firewall на базі рішення F5 Networks для підвищення рівня безпеки веб-застосунків.	10.12.2025 р.	
6.	Оформлення результатів дослідження.	20.12.2025 р.	
7.	Підготовка доповіді до захисту.	30.12.2025 р.	
8.	Оформлення презентації.	02.01.2026 р.	
9.	Отримання рецензії на роботу.	05.01.2026 р.	
10.	Захист в ДЕК.	__.01.2026	

Здобувач вищої освіти _____ Микита М'ЯСНИКОВ
(підпис) *(ім'я, прізвище)*

Керівник кваліфікаційної роботи _____ Ігор ІВАНЧЕНКО
(підпис) *(ім'я, прізвище)*

РЕФЕРАТ

Кваліфікаційна робота присвячена аналізу сучасних загроз безпеці веб-застосунків, дослідженню підходів до їх захисту та практичному застосуванню Web Application Firewall на базі рішення F5 Networks. Робота складається зі вступу, трьох розділів, що містять 32 рисунки, висновків і списку використаних джерел із 41 найменувань. Загальний обсяг роботи становить 75 аркуші, з яких 5 аркушів займають перелік умовних скорочень та список використаних джерел.

Метою роботи є аналіз основних кібератак на веб-застосунки, дослідження принципів захисту за допомогою Web Application Firewall та оцінка ефективності застосування рішення F5 Networks для захисту веб-серверів.

Об'єктом дослідження є процеси забезпечення інформаційної безпеки веб-застосунків у мережевих системах.

Предметом дослідження є основні способи кібер-атак на веб-застосунки та засоби їх захисту з використанням Web Application Firewall на базі рішення F5 Networks.

Методи дослідження. У процесі виконання кваліфікаційної роботи застосовано аналіз і узагальнення науково-технічних джерел з питань безпеки веб-застосунків; аналіз сучасних типів атак; практичне налаштування та тестування політик безпеки Web Application Firewall; аналіз логів для виявлення атак; вивчення рекомендацій і стандартів у сфері захисту веб-ресурсів.

У результаті виконання роботи проаналізовано сучасні загрози безпеці веб-застосунків та підходи до їх захисту, досліджено архітектуру та функціональні можливості Web Application Firewall, а також практично реалізовано захист веб-застосунку за допомогою рішення F5 Networks. На основі отриманих результатів оцінено ефективність використання WAF для виявлення та блокування атак, сформульовано рекомендації щодо його впровадження та налаштування в реальних інформаційних системах.

Галузь застосування. Результати та рекомендації, отримані в роботі, можуть бути використані під час проєктування, впровадження та експлуатації

систем захисту веб-застосунків у корпоративних і державних інформаційних системах, а також у діяльності фахівців з кібербезпеки та адміністраторів веб-інфраструктури.

Апробація:

А.М. Котенко, Н.С. Бондаренко, М.С. М'ясников ЗАХИСТ КОНФІДЕНЦІЙНОЇ ІНФОРМАЦІЇ У МЕРЕЖЕВИХ КАНАЛАХ ВІД НЕСАНКЦІОНОВАНОГО ДОСТУПУ ШЛЯХОМ ВИКОРИСТАННЯ ОПЕРАЦІЙНОЇ СИСТЕМИ LINUX. *Всеукраїнської науково- практичної конференція: Актуальні проблеми безпеки інформаційно-телекомунікаційних систем*. м. Київ, 05 листопада 2025 року, Київ: РВЦ ДУІКТ. – 2025. С.52-53. https://duikt.edu.ua/uploads/p_2779_72486260.pdf

Ключові слова: ІНФОРМАЦІЙНА БЕЗПЕКА, ВЕБ-ЗАСТОСУНКИ, БРАНДМАУЕР, WEB APPLICATION FIREWALL, F5 NETWORKS

ABSTRACT

The qualification work is devoted to the analysis of modern threats to the security of web applications, the study of approaches to their protection and the practical application of the Web Application Firewall based on the F5 Networks solution. The work consists of an introduction, three sections containing 32 figures, conclusions and a list of sources used from 41 items. The total volume of the work is 75 sheets, of which 5 sheets are occupied by a list of conventional abbreviations and a list of sources used.

The purpose of the study is to analyze the main cyberattacks on web applications, study the principles of protection using the Web Application Firewall and assess the effectiveness of using the F5 Networks solution to protect web servers.

The object of the study is the processes of ensuring information security of web applications in network systems.

The subject of the study is the main methods of cyberattacks on web applications and means of their protection using the Web Application Firewall based on the F5 Networks solution.

Research methods. In the process of performing the qualification work, the analysis and generalization of scientific and technical sources on web application security issues were applied; analysis of modern types of attacks; practical configuration and testing of Web Application Firewall security policies; analysis of logs to detect attacks; study of recommendations and standards in the field of web resource protection.

As a result of the work, modern web application security threats and approaches to their protection were analyzed, the architecture and functionality of the Web Application Firewall were studied, and web application protection using the F5 Networks solution was practically implemented. Based on the results obtained, the effectiveness of using WAF to detect and block attacks was assessed, recommendations were formulated for its implementation and configuration in real information systems.

Field of application. The results and recommendations obtained in the work can be used during the design, implementation and operation of web application protection systems in corporate and government information systems, as well as in the activities

of cybersecurity specialists and web infrastructure administrators.

Keywords: INFORMATION SECURITY, WEB-APPLICATION, WEB APPLICATION FIREWALL, F5 NETWORKS

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ

WAF	Web Application Firewall
ASM	Application Security Manager
APM	Access Policy Manager
LTM	Local Traffic Manager
OWASP	Open Web Application Security Project
XSS	Server-Side Request Forgery
SSRF	Server-Side Request Forgery
CSRF	Client-Side Request Forgery
SSL	Secure Socket Layer
TLS	Transport Layer Security
API	Application Programming Interface
REST	Representational State Transfer
ЦОД	Центр Обробки Даних
TMOS	Traffic Management Operating System

ВСТУП

Актуальність теми. Актуальність обраної теми зумовлена тим, що на сучасному етапі розвитку інформаційних технологій веб-застосунки є основою більшості інформаційних систем, які використовуються як у комерційному, так і в державному секторах, при цьому кількість атак на них постійно зростає. Аналіз сучасних інцидентів інформаційної безпеки свідчить, що навіть за наявності базових засобів захисту веб-ресурси залишаються вразливими до атак прикладного рівня, зокрема SQL-ін’єкцій, XSS чи автоматизовані спроби підбору облікових даних. Це свідчить про необхідність використання спеціалізованих рішень, здатних аналізувати логіку HTTP-запитів і поведінку користувачів у режимі реального часу. Саме тому дослідження можливостей Web Application Firewall та практичне застосування рішення F5 Networks є актуальним завданням, оскільки воно дозволяє не лише поглибити теоретичні знання з захисту веб-застосунків, а й отримати практичні навички налаштування сучасних засобів кіберзахисту, які реально використовуються в професійному середовищі.

Метою роботи є аналіз основних кібератак на веб-застосунки, дослідження принципів захисту за допомогою Web Application Firewall та оцінка ефективності застосування рішення F5 Networks для захисту веб-серверів.

Об’єктом дослідження є процеси забезпечення інформаційної безпеки веб-застосунків у мережевих системах.

Предметом дослідження є основні способи кібер-атак на веб-застосунки та засоби їх захисту з використанням Web Application Firewall на базі рішення F5 Networks.

Методи дослідження. У процесі виконання кваліфікаційної роботи застосовано аналіз і узагальнення науково-технічних джерел з питань безпеки веб-застосунків; аналіз сучасних типів атак; практичне налаштування та тестування політик безпеки Web Application Firewall; аналіз логів для виявлення атак; вивчення рекомендацій і стандартів у сфері захисту веб-ресурсів.

Для досягнення цієї мети в роботі необхідно виконати наступні **завдання**:

1. Аналіз сучасних кіберзагроз і типових атак на веб-застосунки.

2. Аналіз існуючих підходів і засобів захисту веб-застосунків, зокрема із застосуванням Web Application Firewall.

3. Розроблення та практичне впровадження рекомендацій щодо налаштування та використання Web Application Firewall на базі рішення F5 Networks для підвищення рівня безпеки веб-застосунків.

Практичне значення одержаних результатів. Розроблені в роботі рекомендації щодо налаштування та використання Web Application Firewall на базі рішення F5 Networks можуть бути застосовані під час проєктування, впровадження та експлуатації систем захисту веб-застосунків у корпоративних та публічних інформаційних системах, де критично важливим є забезпечення високого рівня кібербезпеки та стійкості до сучасних веб-атак.

Апробація результатів кваліфікаційної роботи відбулася на III Міжнародній науково-практичній конференції “ Сучасні аспекти діджиталізації та інформатизації в програмній та комп’ютерній інженерії” 5 грудня 2025 року.

Розділ 1 ТЕОРЕТИЧНІ ОСНОВИ ЗАХИСТУ ВЕБ-ЗАСТОСУНКІВ

1.1 Сучасні загрози веб-застосункам

На сучасному етапі розвитку інформаційних технологій спостерігається стрімкий розвиток бізнес-процесів, що супроводжується масовим переходом сервісів у цифровий простір. Веб-застосунки стали невід'ємною частиною критичної інфраструктури, фінансового сектору, державного управління та сфери послуг. Проте паралельно з розширенням функціональних можливостей програмного забезпечення зростає і кількість векторів атак, спрямованих на компрометацію конфіденційної інформації, порушення цілісності даних та обмеження доступності ресурсів. Технологічність веб-серверів зумовлює появу нових архітектурних підходів таких як мікросервісна архітектура, контейнеризація та активне використання API. Це, у свою чергу, ускладнює способи захисту. Традиційні засоби мережевого захисту, зокрема міжмережеві екрани L3/L4 рівнів виявляються недостатньо ефективними проти атак прикладного рівня (L7), оскільки вони не здатні аналізувати логіку HTTPS-трафіку та розпізнавати специфічні маніпуляції всередині сесій користувачів. (рис. 1.1)

Рис.1.1. Приклад клієнт-серверної взаємодії

Аналіз інцидентів кібербезпеки за останні роки свідчить про те, що саме веб-застосунки залишаються найбільш вразливою ланкою в системі захисту підприємства. Мотивація зловмисників варіюється від фінансової вигоди до промислового шпигунства. Враховуючи високу складність сучасного коду та стислі терміни розробки, виникнення помилок у конфігурації або логіці роботи застосунку стає майже неминучим. Саме тому виникає гостра потреба у впровадженні спеціалізованих рішень, таких як Web Application Firewall здатних здійснювати глибокий аналіз запитів у режимі реального часу. [1]

У процесі розробки стратегії захисту веб-застосунків велике значення має аналіз та розуміння можливих векторів атак. Найбільш визнаним та авторитетним джерелом у цьому контексті є Open Web Application Security Project (OWASP) — відкрита некомерційна організація, діяльність якої спрямована на покращення безпеки програмного забезпечення.

Найвідомішим продуктом даної організації є документ OWASP Top 10. Це не просто перелік найбільш розповсюджених помилок у коді, а комплексний аналітичний звіт, що базується на висновках експертів з безпеки з усього світу та аналізі величезних масивів даних про виявлені уразливості. Основна мета проєкту полягає в інформуванні розробників, системних архітекторів та фахівців з кібербезпеки про найбільш критичні ризики, що загрожують сучасним веб-ресурсам. Періодичне оновлення списку OWASP Top 10 відображає еволюцію ландшафту загроз.

Методологія оцінки кожного пункту рейтингу базується на поєднанні кількох факторів:

- Розповсюдженість (Prevalence). Статистичні дані про те, наскільки часто певна категорія уразливостей зустрічається у реальних застосунках під час проведення аудитів та тестувань на проникнення.
- Виявляємість (Detectability). Оцінка того, наскільки легко розробник або автоматизований сканер може знайти дану проблему в системі.

- Складність експлуатації (Exploitability). Визначення технічного рівня підготовки, необхідного зловмиснику для успішного здійснення атаки через конкретну уразливість.
- Технічний вплив (Technical Impact). Потенційні наслідки для конфіденційності, цілісності та доступності даних у разі успішної компрометації.

Для інженерів з кібербезпеки, що працюють із рішеннями класу WAF , OWASP Top 10 слугує базовим фреймворком для налаштування політик безпеки. Важливо розуміти, що сучасні редакції списку змістили акцент із суто технічних векторів атак на більш фундаментальні проблеми, такі як "Insecure Design" (небезпечне проєктування) або "Software and Data Integrity Failures" (порушення цілісності програмного забезпечення та даних).

Класифікація OWASP дозволяє структурувати підхід до захисту наступним чином:

- Пріоритезація ресурсів: Спрямування зусиль на нейтралізацію найбільш критичних та ймовірних ризиків.
- Відповідність стандартам: Багато галузевих стандартів (наприклад, PCI DSS для платіжних систем) вимагають обов’язкового захисту від загроз, перелічених в OWASP Top 10.
- Створення сигнатурних баз: Виробники засобів захисту, включаючи F5 Networks, інтегрують логіку перевірок у свої пристрої саме на основі категорій цього проєкту.

OWASP Top 10 є динамічним стандартом, який не лише констатує наявність технічних дір у програмному забезпеченні, а й спонукає до впровадження комплексного підходу "Security by Design". [2]

Рис. 1.2. Класифікація OWASP Top 10

У чинній редакції рейтингу OWASP категорія **Broken Access Control** посідає перше місце, що свідчить про найвищу частоту її виникнення та значний рівень ризику для прикладних систем. Ця уразливість полягає у неспроможності застосунку належним чином забезпечити дотримання політики обмеження доступу, через що користувачі отримують можливість діяти поза межами своїх цільових повноважень. Основними сценаріями експлуатації даної проблеми є несанкціоноване підвищення привілеїв, маніпуляція параметрами у запитах та обхід механізмів автентифікації. Зловмисники можуть переглядати конфіденційні записи інших клієнтів, модифікувати дані або отримувати доступ до адміністративних панелей керування шляхом прямого звернення до захищених URL-адрес, які не мають належної серверної перевірки прав. Складність протидії цій загрозі полягає в тому, що контроль доступу глибоко інтегрований у бізнес-логіку програмного продукту. Традиційні методи фільтрації трафіку часто виявляються неефективними, оскільки запити зловмисника можуть виглядати легітимними за структурою, але бути незаконними за суттю.

Категорія **Cryptographic Failures** посідає другу позицію в актуальному рейтингу OWASP, що обумовлено критичним значенням захисту конфіденційних даних у сучасних розподілених системах. Основними технічними проявами цієї уразливості є використання застарілих та скомпрометованих алгоритмів шифрування , застосування слабких криптографічних ключів, а також відсутність належного налаштування протоколів TLS/SSL. Особливу небезпеку становить передача конфіденційних параметрів, таких як ідентифікатори сесій або персональні дані, у відкритому вигляді через незахищений HTTP-протокол. Це створює передумови для реалізації атак типу Man-in-the-Middle , що призводять до повної втрати конфіденційності. [3]

Категорія **Injection** традиційно посідає одне з ключових місць у переліку найбільш небезпечних загроз для веб-серверів. Сутність даного типу атак полягає у впровадженні зловмисником шкідливого коду або спеціально сформованих вхідних даних у запит, що згодом передаються інтерпретатору як частина команди або запиту. У результаті виконання таких маніпуляцій інтерпретатор змінює заплановану логіку роботи застосунку, що дозволяє виконувати несанкціоновані дії в системі. До цієї категорії належать різноманітні вектори впливу, зокрема SQL-ін’єкції, впровадження команд операційної системи (OS Command Injection), а також Cross-Site Scripting (XSS). Експлуатація ін’єкцій може призвести до повного розкриття вмісту баз даних, несанкціонованої модифікації чи видалення запитів, а також до виконання довільного коду на стороні сервера. Основним чинником виникнення таких уразливостей є відсутність чіткого розмежування між даними користувача та командами, що виконуються програмним забезпеченням. Оскільки сучасні веб-інтерфейси оперують значними обсягами динамічного контенту, складність аналізу кожного вхідного параметра зростає, що робить ін’єкції стабільно актуальною загрозою для систем з архітектурою будь-якої складності.

Insecure Design є одним із сучасних та водночас найбільш складних для виявлення типів загроз, що входить до переліку OWASP . На відміну від

технічних вразливостей, які виникають через помилки в коді або неправильну конфігурацію, небезпечне проєктування пов’язане з недоліками на етапі архітектурного планування веб-застосунку. Фактично, це ситуація, коли система з самого початку створюється без урахування базових принципів безпеки або з помилковими припущеннями щодо поведінки користувачів і можливостей зловмисників. До проявів Insecure Design можна віднести відсутність чіткої моделі загроз, некоректне розмежування прав доступу, покладання критично важливих рішень на клієнтську частину або логіку бізнес-процесів, яка легко обходиться. Такі недоліки не завжди можна усунути простими виправленнями коду, оскільки вони вимагають перегляду загальної логіки роботи системи. Особливу небезпеку цей тип вразливостей становить у складних веб-застосунках із великою кількістю бізнес-операцій, де помилки в проєктуванні можуть призвести до витоку даних, фінансових втрат або порушення цілісності системи. Тому врахування принципів безпечного проєктування ще на початкових етапах розробки є критично важливим для забезпечення надійного захисту веб-застосунків. [4]

Одним із найпоширеніших типів вразливостей, що входить до списку OWASP є **Security Misconfiguration**. Даний вид загроз виникає в наслідок неправильного або неповного налаштування компонентів веб-застосунку, серверного обладнання чи допоміжних сервісів. Часто такі помилки з’являються через використання стандартних конфігурацій, які не були адаптовані до вимог безпеки конкретної системи. Прикладами Security Misconfiguration є увімкнені тестові або адміністративні обліковки, надмірні права доступу, а також розкриття службової інформації через детальні повідомлення про помилки. У багатьох випадках ці вразливості не пов’язані безпосередньо з логікою застосунку, але можуть значно спростити зловмиснику процес атаки та надати додаткову інформацію про внутрішню структуру системи. Особливо актуальною проблема неправильної конфігурації є в умовах використання хмарних платформ, контейнеризації та мікросервісної архітектури, де кількість налаштувань суттєво зростає. Недостатній контроль налаштувань або

відсутність регулярного аудиту безпеки може призвести до серйозних наслідків, включно з компрометацією даних та порушенням доступності сервісів. Саме тому правильна та систематична конфігурація безпеки є важливою складовою захисту сучасних веб-застосунків

Vulnerable and Outdated Components є однією з типових загроз безпеці веб-застосунків. Даний тип вразливостей виникає у випадках, коли в системі використовуються бібліотеки, фреймворки або інші програмні компоненти з відомими уразливостями, які не були своєчасно оновлені. У сучасній веб-розробці майже кожен застосунок залежить від великої кількості сторонніх компонентів, що значно ускладнює контроль їх актуального стану. Проблема посилюється тим, що розробники не завжди відстежують оновлення або вразливості у використаному програмному забезпеченні. У результаті веб-застосунок може містити компоненти з відомими проблемами безпеки, інформація про які є у відкритому доступі. Це значно спрощує роботу зловмисників, оскільки їм не потрібно шукати нові вектори атак — достатньо скористатися вже задокументованими експлойтами.

Identification and Authentication Failures належать до категорії загроз, що виникають у разі неправильної реалізації механізмів перевірки користувачів у веб-застосунках. Даний тип вразливостей пов’язаний із ситуаціями, коли система неналежним чином перевіряє особу користувача або не забезпечує достатній рівень захисту облікових даних. Поширеною проблемою є використання слабких або передбачуваних паролів, а також відсутність обмежень на кількість спроб входу. У таких випадках зловмисники можуть отримати доступ до облікових записів шляхом підбору паролів або використання викрадених облікових даних. Крім того, ризики зростають за відсутності додаткових механізмів захисту, таких як багатофакторна автентифікація. [5]

Тип загроз, що пов’язаний із відсутністю належного контролю за змінами у коді, конфігураціях або даних, які використовуються веб-застосунком називається **Software and Data Integrity Failures**. Дана проблема виникає у випадках, коли система не перевіряє походження, цілісність або достовірність

програмних компонентів і даних, що завантажуються або оновлюються під час роботи. Поширеним прикладом є використання оновлень, бібліотек або сторонніх модулів без перевірки цифрових підписів чи контрольних сум. У такій ситуації існує ризик підміни легітимних компонентів на модифіковані або шкідливі. Аналогічні загрози можуть виникати під час автоматичних процесів розгортання, коли відсутній контроль цілісності на етапах CI/CD. Порушення цілісності також можуть стосуватися даних, які зберігаються або передаються між компонентами системи. За відсутності відповідних механізмів перевірки можливе несанкціоноване внесення змін, що впливає на коректність роботи веб-застосунку. У результаті такі вразливості можуть призвести до виконання шкідливого коду, спотворення даних або повної компрометації системи, що робить контроль цілісності важливим елементом загальної системи безпеки.

Security Logging and Monitoring Failures - це категорія загроз, яка ускладню своєчасне виявлення атак та реагування на інциденти безпеки. Даний тип проблем виникає у випадках, коли події безпеки або зовсім не фіксуються, або зберігаються без належної структури та аналізу. У результаті навіть успішна атака може залишитися непоміченою протягом тривалого часу.Явною ознакою є відсутність журналювання спроб автентифікації, змін прав доступу, помилок у роботі застосунку або підозрілих дій користувачів. Навіть за наявності логів вони часто не аналізуються в реальному часі, що знижує їхню цінність. У таких умовах реагування на інциденти відбувається із запізненням, коли система вже зазнала шкоди. [6]

Вразливість класу **Server-Side Request Forgery (SSRF)** була включена до останньої редакції рейтингу OWASP Top 10 як окрема категорія, що підкреслює її зростаючу актуальність у сучасних хмарних та мікросервісних архітектурах. Суть даної загрози полягає у можливості зловмисника змусити серверний застосунок надіслати сформований запит на довільну, зазвичай внутрішню або непередбачену логікою розробки, адресу. Експлуатація SSRF стає можливою, коли веб-застосунок приймає від користувача URL-адресу або інший параметр для отримання зовнішнього ресурсу (наприклад, імпорту зображень, зчитування

метаданих або взаємодії з API) без належної перевірки та фільтрації пункту призначення. Це дозволяє атакувальнику використовувати вразливий сервер як своєрідний проксі для доступу до ресурсів, що знаходяться за периметром мережевого захисту та закриті для прямого доступу з інтернету. Найбільш критичними наслідками SSRF є сканування внутрішніх портів мережі, отримання доступу до локальних сервісів таких як бази даних. Оскільки запит ініціюється довіреним сервером застосунку, внутрішні системи ідентифікують його як легітимний, що робить цей вектор атаки надзвичайно ефективним для подальшого просування зловмисника всередині інфраструктури організації. [7]

1.2. Підходи до захисту веб-застосунків

Захист веб-застосунків є складним завданням, яке потребує використання різних підходів та методів. Це пов’язано з тим, що жоден окремий механізм безпеки не може повністю захистити застосунок від усіх можливих загроз. Кожен підхід до захисту вирішує лише певні проблеми, тому на практиці найчастіше застосовується поєднання декількох рішень. Саме комплексний підхід дозволяє зменшити імовірність успішної атаки та підвищити загальний рівень безпеки веб-застосунку.

Підходи до захисту веб-застосунків формуються залежно від архітектури системи, типу оброблюваних даних, вимог до доступності та рівня ризику. У деяких випадках основна увага приділяється захисту на етапі розробки, в інших - контролю та фільтрації трафіку під час експлуатації. Важливу роль відіграє також правильна організація процесів адміністрування та підтримки, оскільки навіть добре спроєктований застосунок може стати вразливим через помилки в налаштуваннях або несвоєчасне оновлення компонентів.

Одним із базових підходів до захисту веб-застосунків є впровадження заходів безпеки на рівні програмного коду. Саме на етапі розробки створюється основа майбутньої безпеки системи, тому що більшість вразливостей виникає через помилки в логіці роботи застосунку або некоректну обробку даних. Захист

на цьому рівні спрямований на запобігання типових помилок ще до того, як веб-застосунок буде введено в експлуатацію. [8]

Одним з основних елементів цього підходу є контроль введених користувачем даних. Веб-застосунки зазвичай приймають значну кількість інформації через форми, параметри запитів або файли. Якщо ці дані не перевіряються належним чином, зловмисник може використати їх для виконання атак. Тому важливо здійснювати перевірку типу, формату та допустимих значень усіх вхідних даних. Такий підхід дозволяє зменшити ризик виконання поганих команд або зміни логіки роботи застосунку.

Правильна реалізація механізмів автентифікації та авторизації відіграє важливу роль у захисті на рівні програмного коду. Необхідно чітко розмежовувати доступ користувачів до різних функцій веб-застосунку відповідно до їхніх прав. Помилки в цих функція можуть призвести до ситуацій, коли користувач отримує доступ до ресурсів, на які він не має відповідних прав. Тому при розробці особлива увага приділяється перевірці прав доступу на кожному етапі виконання запитів. Важливим аспектом є правильна обробка помилок і непередбачуваних ситуацій. Повідомлення про помилки не повинні містити детальної технічної інформації про внутрішню структуру застосунку, оскільки це може бути використано для подальших атак. Натомість користувачу має надаватися узагальнене повідомлення, а детальна інформація фіксується у внутрішніх журналах. Такий підхід дозволяє зменшити витік службових даних і водночас зберегти можливість аналізу помилок. Захист на рівні програмного коду також передбачає використання перевірених бібліотек і фреймворків, а також регулярне оновлення програмних компонентів. Застарілі або непідтримувані залежності можуть містити відомі вразливості, які легко експлуатуються. Використання актуальних версій програмного забезпечення знижує ймовірність успішних атак та підвищує загальну стабільність системи.

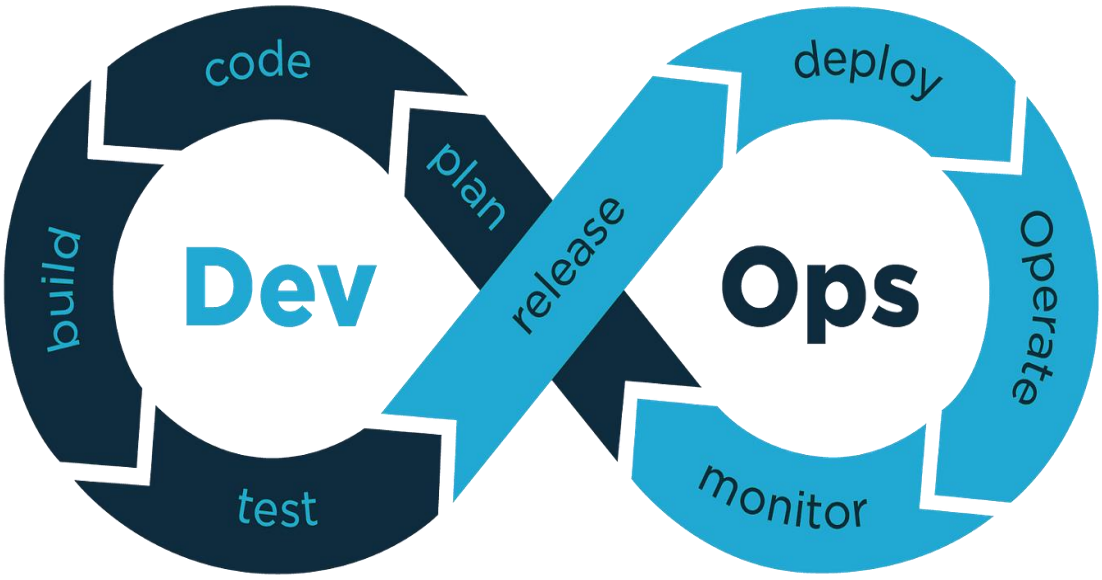


Рис.1.3. Процес безперервної розробки і удосконалення безпеки застосунку

Іншим важливим підходом є захист на рівні інфраструктури та мережі. Він включає використання міжмережевих екранів, сегментацію мережі, обмеження доступу до серверів та застосування захищених протоколів обміну даними. Ці підходи дозволяють контролювати мережевий трафік, який надходить до веб-серверу або виходить з нього. За допомогою таких засобів можна обмежити доступ лише до необхідних портів і сервісів, блокуючи всі інші з’єднання. Це дозволяє зменшити ризик атак, які спрямовані на використання відкритих або неправильно налаштованих мережевих сервісів. Важливу роль у межах цього підходу відіграє сегментація мережі. Вона передбачає розподіл мережевої інфраструктури на окремі зони з різним рівнем доступу. Наприклад, сервери з веб-застосунком можуть бути відокремлені від внутрішніх сервісів або баз даних. Такий підхід дозволяє обмежити поширення атаки у випадку компрометації одного з компонентів системи та зменшує потенційні наслідки інцидентів безпеки. Окрему увагу слід приділити контролю доступу до серверів, на яких розміщено веб-застосунок. Доступ до адміністративних інтерфейсів має бути обмежений лише для уповноважених користувачів і здійснюватися з

використанням захищених методів автентифікації. Використання принципу мінімальних дозволів дозволяє зменшити ризик отримання повного доступу до системи у разі компрометації облікового запису. [9]

Один з інфраструктурних підходів є використання захищених протоколів обміну даними. Передача інформації між клієнтом і сервером повинна здійснюватися з використанням шифрування, що дозволяє запобігти перехопленню або зміні даних під час передавання. Це особливо важливо для веб-застосунків, які працюють з обліковими даними користувачів або іншою чутливою інформацією. Водночас слід зазначити, що захист на рівні інфраструктури та мережі має певні обмеження. Такий підхід ефективно знижує ризики атак на мережевому рівні, однак не завжди дозволяє виявити або заблокувати атаки, які використовують помилки в логіці самого веб-застосунку. Саме тому інфраструктурний захист має розглядатися як один з елементів комплексної системи безпеки, а не як універсальне рішення.

Також важливим підходом до захисту веб-застосунків є постійний моніторинг та аналіз подій безпеки. Даний підхід спрямований не стільки на запобігання атакам, а на їх своєчасне виявлення та зменшення можливих наслідків. Навіть, за наявності інших засобів захисту повністю виключити імовірність інцидентів неможливо, тому контроль подій безпеки відіграє ключову роль у підтриманні стабільної та безпечної роботи веб-застосунку. Моніторинг подій безпеки передбачає збір інформації про роботу веб-застосунку, дії користувачів, запити до сервера та системні події. Для цього використовуються журнали подій, у яких фіксуються спроби входу в систему, помилки автентифікації, звернення до захищених ресурсів та інші важливі дії. Аналіз таких даних дозволяє виявляти підозрілу активність, яка може свідчити про спроби несанкціонованого доступу або проведення атак. Особливу увагу в межах цього підходу приділяють виявленню аномальної поведінки. До неї можуть належати багаторазові невдалі спроби входу, різке збільшення кількості запитів, звернення до не типових функцій або використання незвичних параметрів у запитах. Своєчасне виявлення таких ознак дозволяє швидко

реагувати на можливу загрозу ще до того, як вона призведе до серйозних наслідків для веб-застосунку або даних користувачів. [10]

Важливим елементом моніторингу є коректне налаштування логування. Події безпеки мають фіксувати у достатньому обсязі, щоб забезпечити можливість подальшого аналізу. Журнали повинні зберігатися у захищеному вигляді та бути доступними лише для уповноважених осіб. Це дозволяє уникнути їх підміни або видалення у разі спроб приховати сліди атаки. Моніторинг і аналіз подій безпеки також є важливим інструментом під час розслідування інцидентів. У разі виникнення проблем журнал подій дозволяє відтворити послідовність дій, визначити джерело загрози та оцінити масштаб впливу. Отримана інформація може бути використана для усунення вразливостей і покращення загальної системи захисту веб-застосунку. Ефективність цього підходу значною мірою залежить від регулярності аналізу зібраних даних. За відсутності належного контролю журнали подій перетворюються на формальність і не виконують своєї функції. Саме тому моніторинг має бути безперервним процесом, інтегрованим у загальну систему безпеки.

Окрему увагу серед підходів до захисту веб-застосунків слід приділити контролю та аналізу HTTP-трафіку. Даний підхід орієнтований на перевірку запитів і відповідей, які передаються між клієнтом і сервером, з метою виявлення потенційно небезпечної або шкідливої активності. Оскільки більшість атак на веб-застосунки реалізується саме через HTTP-запити, контроль цього рівня є важливим елементом загальної системи безпеки. Захист на рівні HTTP-трафіку зазвичай реалізується за допомогою спеціалізованих засобів - Web Application Firewall. Такі рішення аналізують вхідні та вихідні запити, порівнюють їх із заданими правилами та шаблонами і, у разі виявлення підозрілих дій, можуть блокувати або обмежувати обробку запиту. Це дозволяє запобігати типовим атакам без необхідності втручання у програмний код веб-застосунку. Перевагою цього підходу є можливість захисту вже розгорнутих і працюючих веб-застосунків, навіть якщо вони містять певні вразливості. Контроль HTTP-трафіку дозволяє зменшити ризики експлуатації таких вразливостей до моменту

їх усунення. Крім того, подібні засоби часто забезпечують ведення журналів подій безпеки, що спрощує аналіз інцидентів і контроль стану системи. [11]

Але використання WAF не є універсальним рішенням і не замінює інші підходи до захисту. Ефективність цього підходу значною мірою залежить від правильного налаштування правил і регулярного їх оновлення. Саме тому контроль і аналіз HTTP-трафіку має застосовуватися як складова комплексної системи захисту веб-застосунків.

Підходи до захисту веб-застосунків охоплюють різні рівні — від програмного коду до мережевої інфраструктури та процесів експлуатації. Кожен із цих підходів має власне призначення та обмеження, тому найбільш ефективним є їх поєднання.

1.3. Принципи роботи Web Application Firewall

Web Application Firewall є одним із ключових засобів захисту веб-застосунків, робота якого базується на детальному аналізі прикладного трафіку, що передається між клієнтом і сервером. Основна задача такого засобу полягає у виявленні та блокуванні запитів, які можуть становити загрозу безпеці веб-застосунку, таких як спроби несанкціонованого доступу, вставляння шкідливого коду або використання вразливостей у логіці застосунку. На відміну від традиційних мережевих пристроїв, які контролюють лише мережеві пакети, WAF зосереджується на аналізі логіки запитів і відповідей, враховує параметри, структуру HTTP-запитів, заголовки та вміст переданих даних. Це дозволяє виявляти атаки, які не помітні при поверхневому аналізі мережевого трафіку. [12]

У типовій архітектурі WAF розміщується між користувачем та веб-застосунком, виконуючи роль проміжної ланки, або зворотнього проксі (рис. 1.4). Початково сесія створюється між клієнтом і WAF, і друга сесія між фаєрволом та сервером. Таким чином усі запити від клієнта спочатку проходять

через WAF, де вони розшифровуються та аналізуються відповідно до визначених правил і політик безпеки, а потім шифруються знову і відправляються на обробку серверу. Ці правила можуть визначати допустимі методи запитів, обмеження на кількість параметрів, довжину рядків, допустимі символи або структуру запиту. Тільки запити, які відповідають політикам безпеки, передаються далі до веб-застосунку. Аналогічно WAF може контролювати відповіді сервера перед тим, як вони надійдуть користувачеві, перевіряючи їх на наявність конфіденційних даних або індикаторів компрометації. Додатково WAF здатний відслідковувати аномалії у поведінці користувачів та трафіку. Наприклад, різке збільшення кількості запитів за короткий проміжок часу, повторювані спроби доступу до захищених ресурсів або нестандартні параметри в запитах можуть бути виявлені як потенційна загроза. Такий аналіз дозволяє реагувати на атаки на ранньому етапі і зменшити імовірність успішного проникнення. Багато сучасних WAF підтримують різні режими роботи, включаючи режим блокування або режим спостереження, коли всі порушення лише фіксуються в журналах. Це дозволяє поступово адаптувати систему до особливостей конкретного веб-застосунку, знижуючи ризик блокування легітимного трафіку та підвищуючи точність політик безпеки. [13]

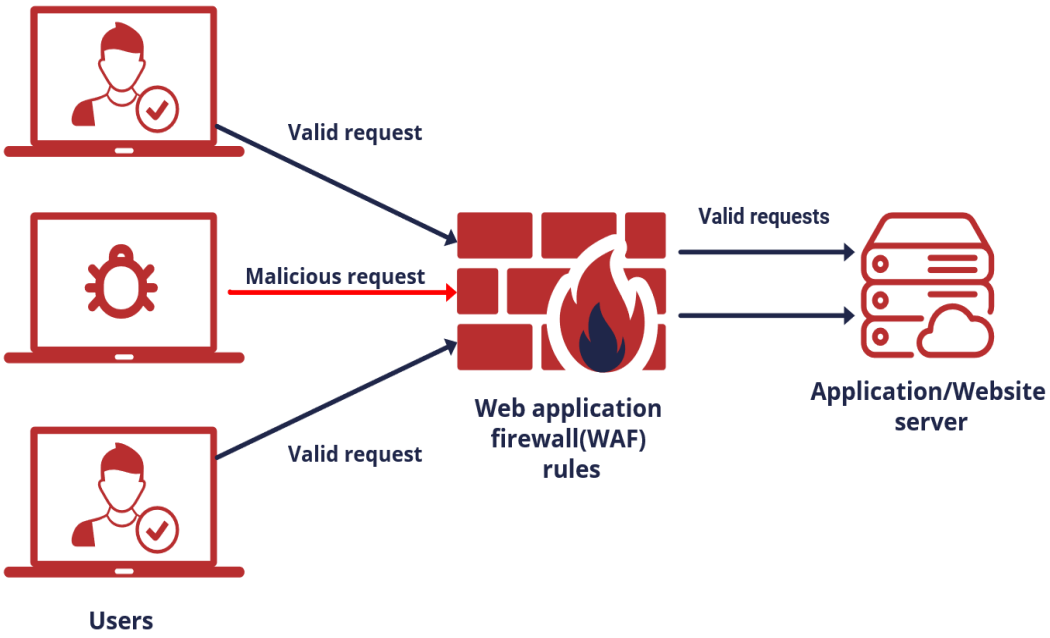


Рис. 1.4. Приклад мережевої архітектури з використанням Web Application Firewall

Одним із підходів, що використовується в роботі Web Application Firewall, є позитивна модель безпеки. Сутність цього підходу полягає в тому, що за замовчуванням дозволяються лише ті запити та дії, які вважаються коректними та очікуваними для конкретного веб-застосунку. Усі інші запити, які не відповідають заданим правилам, розглядаються як потенційно небезпечні та можуть бути заблоковані. Позитивна модель безпеки базується на чіткому описі допустимої поведінки веб-застосунку. Для цього визначаються правила, які описують структуру HTTP-запитів, допустимі параметри, їх формат, довжину та можливі значення. Наприклад, для певного поля може бути дозволено лише числове значення або рядок обмеженої довжини. Якщо запит не відповідає таким вимогам, він вважається некоректним незалежно від того, чи містить він явні ознаки атаки. Використання позитивної моделі дозволяє зменшити ризик успішної експлуатації вразливостей, пов’язаних з некоректною обробкою вхідних даних. Оскільки дозволяється лише заздалегідь визначений набір дій, можливості для виконання небажаних операцій значно обмежуються. Це особливо актуально для веб-застосунків зі стабільною структурою та чітко визначеною логікою роботи. Для ефективної роботи цього способу захисту необхідно коректно описати всі допустимі сценарії взаємодії користувача із застосунком. Якщо певна легітимна дія не врахована у правилах, вона може бути помилково заблокована. Тому впровадження позитивної моделі зазвичай потребує детального аналізу роботи веб-застосунку та поетапного налаштування політик безпеки. Позитивна модель безпеки часто використовується у поєднанні з режимом спостереження (Learning mode), коли WAF не блокує запити, а лише фіксує відхилення від очікуваної поведінки. Це дозволяє зібрати статистику реального трафіку, скоригувати правила та зменшити кількість помилкових спрацювань перед переходом до активного режиму блокування. Такий підхід спрощує адаптацію системи захисту до особливостей конкретного веб-застосунку. [14]

Наступним підходом, який часто використовується при впровадженні WAF-у - це негативна модель безпеки. Вона базується на ідентифікації та

блокуванні відомих небезпечних запитів або дій. На відміну від позитивної моделі, де дозволяються лише очікувані дії, негативна модель передбачає фільтрацію трафіку на основі заборонених патернів, відомих вразливостей та розповсюджених атак. Всі запити, які не відповідають критеріям загрози, проходять без обмежень, тоді як запити, які підпадають під шаблони атаки, блокуються або обробляються згідно з політикою безпеки. Основою роботи негативної моделі є використання сигнатурного аналізу. WAF порівнює вхідні HTTP-запити з базою відомих шаблонів атак, таких як SQL-ін’єкції, міжсайтовий скриптинг (XSS), спроби обходу авторизації чи інші поширені загрози. Якщо запит відповідає хоча б одній сигнатурі, система сприймає його як потенційно шкідливий і реагує відповідно до налаштувань. Це дозволяє захищати веб-застосунок від повторюваних атак, які вже були описані фахівцям з безпеки. Негативна модель також включає перевірку на поведінкові патерни атак. Такий аналіз дозволяє виявляти підозрілі дії, навіть якщо вони не збігаються з конкретними сигнатурами. Наприклад, багаторазові невдалі спроби входу, різке збільшення кількості запитів до одного ресурсу або нетипова структура параметрів можуть вказувати на атаку на веб-сервер. WAF у цьому випадку може швидко відреагувати і заблокувати підозрілу активність. Негативна модель безпеки добре підходить для веб-застосунків із великою кількістю різноманітних запитів та динамічною логікою роботи. Вона дозволяє швидко реагувати на загрози без необхідності детально описувати всі очікувані дії користувачів. Наявність широкого набору сигнатур і правил дозволяє системі автоматично блокувати численні атаки без втручання розробників.

Проте ефективність негативної моделі значною мірою залежить від актуальності бази сигнатур і правил. Якщо база застаріла або не охоплює нові типи атак, деякі загрози можуть залишатися непоміченими. Тому адміністрування WAF із негативною моделлю передбачає регулярне оновлення правил і постійний моніторинг роботи системи. Це дозволяє підтримувати високий рівень безпеки та швидко реагувати на нові методи атак. [15]

Ще одним аспектом є баланс між безпекою та доступністю веб-застосунку. Надмірно жорсткі правила негативної моделі можуть призвести до блокування легітимних запитів, що порушує роботу застосунку для користувачів. Для уникнення таких ситуацій використовується тестування та налаштування правил, а також ведення журналів подій, які допомагають відстежувати і аналізувати блокування запитів.

Основною перевагою Web Application Firewall є гнучке налаштування політик безпеки, що дозволяє адаптувати правила до особливостей конкретного веб-застосунку. Такий підхід дозволяє уникнути помилкових блокувань легітимних запитів і забезпечує стабільність роботи сервісу для кінцевих користувачів. Гнучкість налаштування реалізується через можливість створення окремих політик для різних частин веб-застосунку або для різних типів трафіку. Наприклад, сторінки з формами для введення даних можуть мати більш суворі правила перевірки параметрів, тоді як статичні ресурси, такі як зображення або стилі, можуть оброблятися з мінімальною перевіркою. Така сегментація дозволяє ефективно контролювати критичні точки входу, зберігаючи швидкість і доступність інших частин системи.

Особливу роль у гнучкому налаштуванні відіграє поєднання позитивної і негативної моделей безпеки. Позитивна модель визначає очікувану поведінку застосунку та дозволяє виконувати лише ті дії, які відповідають встановленим правилам. Вона ефективно запобігає виконанню небажаних операцій і обмежує можливості для експлуатації вразливостей. Негативна модель, у свою чергу, фокусується на виявленні відомих загроз і шаблонів атак, блокуючи лише ті запити, які співпадають із відомими сигнатурами. Використання обох моделей одночасно дозволяє гнучко налаштовувати політики під конкретні сценарії використання веб-застосунку. Наприклад, для частин системи з високим ризиком компрометації можна застосовувати переважно позитивну модель, щоб дозволяти лише чітко визначені дії, а для менш критичних ресурсів використовувати негативну модель для блокування відомих атак без обмеження

легітимного трафіку. Такий підхід зменшує ймовірність помилкових спрацювань і підвищує ефективність захисту без втрати доступності. [16]

Гнучке налаштування політик також передбачає можливість режимного застосування WAF. У режимі спостереження всі запити аналізуються та журналюються, але не блокуються, що дозволяє оцінити ефективність правил і скоригувати політики перед включенням активного захисту. У режимі блокування система відхиляє небезпечні запити відповідно до встановлених правил, забезпечуючи реальний захист веб-застосунку.

Журналювання та моніторинг подій безпеки є важливим елементом роботи Web Application Firewall і забезпечує можливість аналізу поведінки веб-застосунку та виявлення потенційних загроз. Всі запити, які блокуються або визнаються підозрілими, фіксуються у спеціальних журналах подій (рис. 1.5). Таке ведення обліку дозволяє відстежувати дії користувачів, реагувати на інциденти та проводити подальший аналіз ефективності налаштованих правил безпеки. Журнали містять інформацію про тип атаки, час звернення, IP-адресу джерела, запитувані ресурси та параметри запиту, що дозволяє відтворити хід подій у разі інциденту. Безперервний моніторинг подій у WAF дозволяє виявляти аномальну поведінку користувачів та системні відхилення, які можуть свідчити про спроби несанкціонованого доступу, сканування вразливостей або реалізацію атак типу SQL-ін’єкції або інших. Регулярний перегляд журналів сприяє швидкому реагуванню на інциденти та зменшує час, протягом якого зловмисник може впливати на систему.

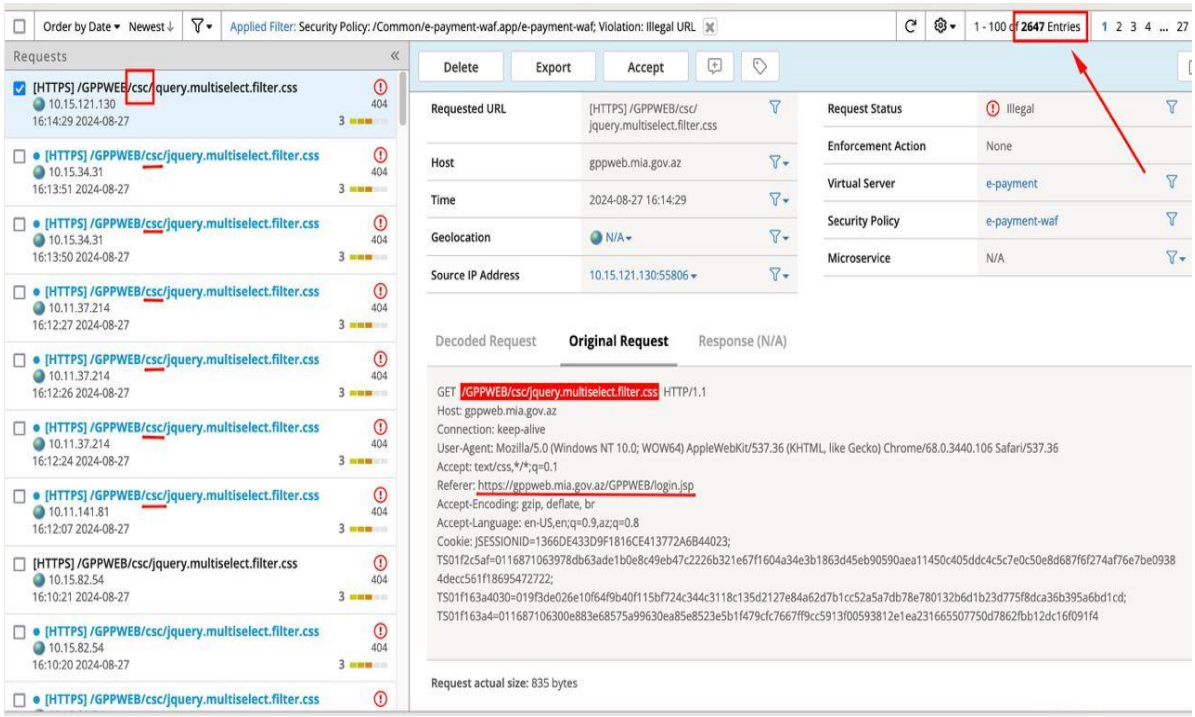


Рис. 1.5. Приклад журналювання і аналізу атак з веб-консолі WAF-у

Важливою складовою сучасного моніторингу є централізація даних про події безпеки. WAF може інтегруватися з SIEM системами, що дозволяє збирати журнали з різних джерел у єдиному середовищі. Така інтеграція забезпечує глибокий аналіз подій у контексті загальної інфраструктури безпеки організації, виявлення складних атак, які поодинці можуть залишатися непоміченими . Через SIEM можливе корелювання даних з інших джерел, наприклад мережевих пристроїв, серверів або антивірусних систем, що підвищує точність виявлення загроз і ефективність заходів реагування. Журнали подій, передані до SIEM, можуть використовуватися для аналізу тенденцій атак, виявлення повторюваних спроб компрометації, оцінки навантаження на систему та перевірки ефективності налаштованих правил WAF. Також це дозволяє проводити аудит безпеки та звітність перед керівництвом чи зовнішніми контролюючими органами. Регулярний аналіз журналів допомагає вдосконалювати політики безпеки, уточнювати правила блокування та виявляти нові типи загроз, які раніше не були враховані. [17]

Принципи роботи Web Application Firewall базуються на постійному аналізі прикладного трафіку, застосуванні правил безпеки та контролі поведінки запитів. Використання WAF дозволяє забезпечити додатковий рівень захисту веб-застосунків і зменшити ризики реалізації атак.

Висновки до розділу 1

Було досліджено, що розвиток веб-технологій підвищує зручність і доступність сервісів, водночас збільшує ризики компрометації даних та порушення цілісності систем. Розглянуто сучасні загрози веб-застосункам, серед яких визначено найбільш поширені види атак та вразливостей, що можуть бути використані зловмисниками.

Досліджено основні підходи до захисту веб-застосунків. Оглянуто як на етапі розробки можна суттєво підвищити якість додатку та значно підвищити рівень безпеки. Було визначено, що хоч способи мережевого засобу є досить ефективними в протидії атакам мережевого рівня, наприклад DDoS, але повністю не ефективними при атаках прикладного рівня. Встановлено, що для захисту веб-серверів від атак прикладного рівня найкращим способом є використання Web Application Firewall та використання методів журналювання та моніторингу подій безпеки для швидкого і ефективного реагування.

Детально розглянуто принципи роботи Web Application Firewall, який виступає ключовим інструментом захисту веб-застосунків. Встановлено, що WAF здійснює аналіз вхідних та вихідних запитів, застосовує правила фільтрації, виявляє аномалії та підозрілі дії, а також може інтегруватися з системами SIEM для централізованого збору та аналізу подій безпеки. Така організація роботи дозволяє забезпечити контроль над HTTP-трафіком, виявляти відомі та нові загрози.

Розділ 2 АНАЛІЗ ТА АРХІТЕКТУРА РІШЕННЯ F5 NETWORKS ДЛЯ ЗАХИСТУ ВЕБ-ЗАСТОСУНКІВ

2.1 Огляд компанії F5 Networks та платформи BIG-IP

У сучасних умовах активного розвитку веб-технологій та зростання кількості кіберзагроз особливого значення набувають комплексні рішення для захисту веб-застосунків і мережевої інфраструктури. Однією з компаній, що спеціалізується на розробці таких рішень, є F5 Networks. Продукти цієї компанії широко використовуються в корпоративних мережах, ЦОДах та хмарних середовищах, зокрема для забезпечення безпеки, доступності та продуктивності веб-ресурсів. Компанія F5 Networks була заснована у 1996 році в США та з початку своєї діяльності орієнтувалася на розробку рішень для оптимізації мережевого трафіку та балансування навантаження. З часом фокус компанії розширився, і сьогодні F5 є одним із провідних постачальників рішень у сфері захисту застосунків. Продукти F5 використовуються як великими міжнародними корпораціями, так і державними установами, фінансовими організаціями та постачальниками хмарних сервісів. Основною ідеєю, закладеною в продуктах F5, є забезпечення стабільної та безпечної роботи застосунків незалежно від середовища їх розгортання. У зв’язку з цим компанія розробляє рішення, які поєднують у собі функції балансування навантаження, керування трафіком, шифрування, автентифікації та захисту від різноманітних мережевих і прикладних атак. Центральне місце серед таких рішень займає платформа BIG-IP. [18]

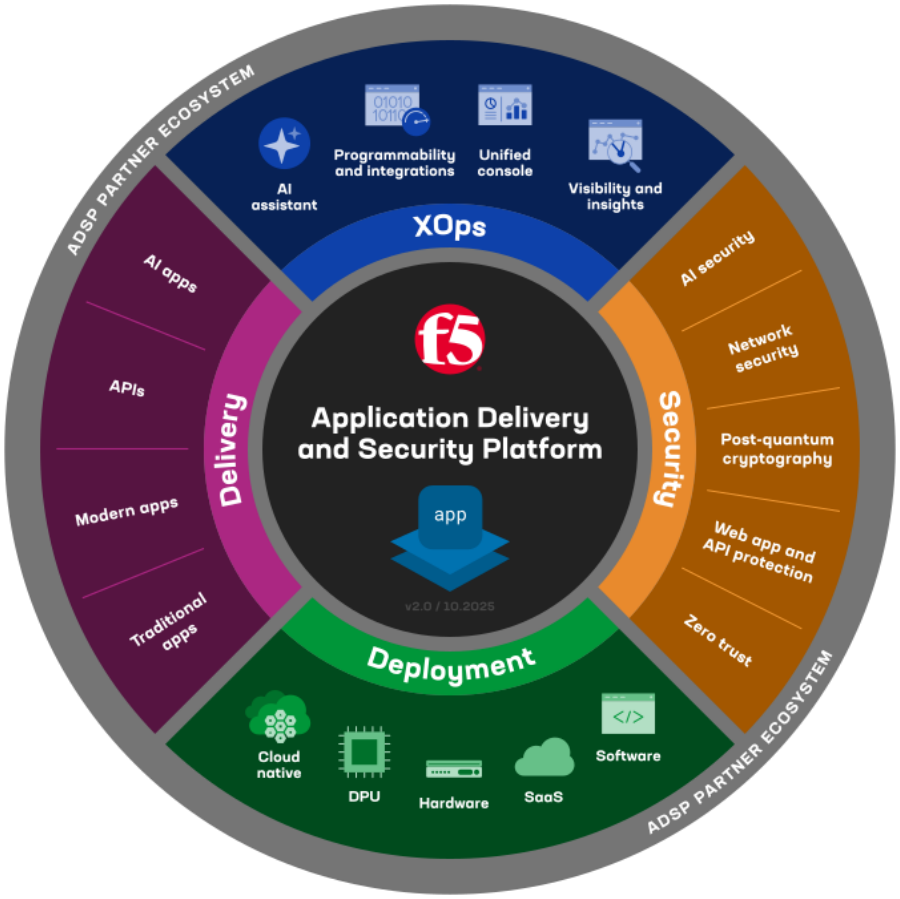


Рис. 2.1. Можливості комплексного рішення від компанії F5 Networks

Платформа BIG-IP є флагманським продуктом компанії F5 Networks і представляє собою багатофункціональне програмне або апаратне рішення для управління трафіком і безпекою застосунків. BIG-IP реалізує концепцію Application Delivery Controller, який працює на всіх рівнях моделі OSI. Це дозволяє здійснювати глибокий аналіз трафіку та приймати рішення не лише на основі IP-адрес або портів, а й з урахуванням параметрів HTTP/HTTPS, сесій користувачів і логіки роботи веб-застосунків. [19]

Однією з ключових особливостей платформи BIG-IP є її модульна архітектура, яка суттєво впливає на підхід до побудови системи захисту та управління веб-застосунками. Архітектура BIG-IP побудована за принципом єдиного програмного ядра, до якого можуть підключатися спеціалізовані функціональні модулі. Кожен з таких модулів відповідає за окремий напрям

обробки трафіку або забезпечення безпеки, що дозволяє формувати конфігурацію платформи відповідно до конкретних вимог організації. У результаті рішення не є статичним, а може змінюватися та розширюватися разом із розвитком інформаційної інфраструктури. Застосування модульного підходу дає змогу уникнути використання надлишкового функціоналу та зосередитися лише на тих можливостях, які є актуальними для конкретного середовища. Наприклад, у випадку, коли основною задачею є забезпечення високої доступності веб-сервісів, платформа може бути доповнена лише модулем балансування навантаження. У більш складних сценаріях, де важливу роль відіграють контроль доступу та захист від прикладних атак, можливе підключення додаткових модулів без необхідності повної заміни рішення. Така гнучкість є особливо актуальною в умовах постійної зміни ландшафту кіберзагроз.

Одним із базових та найбільш поширених модулів платформи BIG-IP є Local Traffic Manager (LTM). Даний модуль відповідає за керування локальним трафіком і реалізує функції балансування навантаження між серверами застосунків. LTM аналізує вхідні запити клієнтів та розподіляє їх між доступними серверними вузлами відповідно до заданих алгоритмів, таких як round-robin, least connections або з урахуванням стану сервера. Це дозволяє підвищити доступність веб-застосунків, забезпечити рівномірне використання ресурсів та мінімізувати ризики відмов у разі перевантаження окремих компонентів інфраструктури.

Модуль Access Policy Manager (APM) орієнтований на реалізацію контролю доступу до веб-ресурсів та захищених сервісів. Його основним завданням є перевірка автентичності користувачів і застосування політик авторизації перед наданням доступу до застосунків. APM підтримує різні методи автентифікації, зокрема використання логіна і пароля, сертифікатів, одноразових паролів, а також інтеграцію з зовнішніми системами ідентифікації, такими як LDAP, Active Directory або RADIUS. Це дозволяє використовувати єдиний підхід до управління доступом у межах організації.

Окрему роль у забезпеченні безпеки відіграють модулі AFM (Advanced Firewall Manager) та ASM/Advanced WAF. Модуль AFM виконує функції мережевого міжмережевого екрану та забезпечує захист на рівнях L3–L4 моделі OSI. Він дозволяє фільтрувати трафік за IP-адресами, протоколами та портами, а також протидіяти атакам типу DoS і DDoS. AFM, як правило, використовується для захисту периметра мережі та обмеження несанкціонованого доступу до внутрішніх ресурсів. Модуль ASM призначений для захисту веб-застосунків на прикладному рівні. Його функціонал базується на аналізі HTTP/HTTPS-трафіку та виявленні ознак атак, спрямованих на експлуатацію вразливостей веб-додатків. Advanced WAF дозволяє блокувати такі поширені типи атак, як SQL Injection, Cross-Site Scripting, підміна параметрів запитів та інші загрози, характерні для сучасних веб-систем. Використання даного модуля значно знижує ризик компрометації застосунків та витоку конфіденційної інформації.
Узгоджена робота всіх модулів у межах єдиної платформи BIG-IP забезпечується операційною системою TMOS, що спрощує адміністрування та централізоване управління політиками. Таким чином, модульна архітектура BIG-IP дозволяє реалізувати комплексний підхід до доставки та захисту веб-застосунків, поєднуючи продуктивність, гнучкість і високий рівень безпеки. [20]

Платформа BIG-IP підтримує декілька варіантів розгортання, що дозволяє використовувати її в різних інфраструктурних умовах та адаптувати до конкретних технічних вимог:

- Апаратний пристрій,
- Віртуальна версія,
- Хмарне середовище.

Такий підхід є важливим з огляду на те, що сучасні інформаційні системи рідко обмежуються одним середовищем і часто поєднують класичні дата-центри, віртуалізовані платформи та хмарні сервіси. Універсальність форм-факторів BIG-IP робить можливим її застосування незалежно від масштабу та типу інфраструктури.

Традиційним варіантом впровадження BIG-IP є використання апаратних пристроїв, які постачаються у вигляді спеціалізованих hardware appliance (рис 2.2). Такі рішення являють собою готові до використання мережеві пристрої з попередньо встановленою операційною системою F5 TMOS та оптимізованими апаратними ресурсами. Апаратні платформи розраховані на обробку значних обсягів трафіку та забезпечують високу продуктивність за рахунок використання спеціалізованих процесорів, апаратного прискорення криптографічних операцій і оптимізованих мережевих інтерфейсів. Саме тому hardware-версії BIG-IP зазвичай застосовуються у великих корпоративних мережах, фінансових установах та організаціях з підвищеними вимогами до стабільності й пропускної здатності. Використання апаратних рішень також має переваги з точки зору передбачуваності роботи системи. Оскільки всі компоненти — як програмні, так і апаратні — постачаються одним виробником, спрощується процес підтримки та усунення несправностей. Крім того, такі пристрої зазвичай розміщуються у центрах обробки даних і інтегруються в існуючу мережеву інфраструктуру як окремий логічний елемент, через який проходить увесь вхідний і вихідний трафік веб-застосунків. [21]

Рис 2.2. Апаратна реалізація пристрою F5

Разом із тим, розвиток технологій віртуалізації зумовив появу віртуальних версій BIG-IP, відомих як Virtual Edition. Даний форм-фактор не прив’язаний до конкретного апаратного забезпечення і може бути розгорнутий у середовищах VMware, Hyper-V, KVM або інших платформах віртуалізації. Функціонально віртуальна версія практично не відрізняється від апаратної, оскільки використовує ту саму операційну систему TMOS та підтримує аналогічний набір модулів. Основна відмінність полягає в способі виділення ресурсів, які залежать від можливостей хост-системи. Virtual Edition є доцільним рішенням у випадках, коли необхідна гнучкість та швидке масштабування. Наприклад, у середовищах з динамічним навантаженням можливо оперативно змінювати обсяг доступних ресурсів або створювати додаткові екземпляри BIG-IP без закупівлі нового апаратного обладнання. Це спрощує тестування, впровадження нових сервісів та адаптацію системи безпеки до змін у структурі веб-застосунків.

Окрему категорію становить використання BIG-IP у публічних хмарних середовищах, таких як Amazon Web Services, Microsoft Azure або Google Cloud Platform. У цьому випадку платформа розгортається у вигляді віртуального екземпляра, оптимізованого під конкретного хмарного провайдера. Такий підхід дозволяє інтегрувати механізми доставки та захисту застосунків безпосередньо в хмарну інфраструктуру, не порушуючи її логіки та принципів масштабування. Операційна система TMOS (Traffic Management Operating System), на якій базується BIG-IP, забезпечує узгоджену роботу всіх модулів платформи. TMOS дозволяє централізовано керувати політиками безпеки, правилами маршрутизації та обробки трафіку. [22]

З точки зору інформаційної безпеки платформа BIG-IP відіграє роль проміжного рівня між користувачами та веб-застосунками. Усі запити клієнтів проходять через BIG-IP, що дозволяє виконувати їх аналіз, фільтрацію та перевірку ще до того, як вони потраплять на сервери застосунків. Такий підхід значно зменшує поверхню атаки та дозволяє ефективно протидіяти поширеним загрозам, включаючи атаки типу SQL Injection, Cross-Site Scripting, brute force атаки та спроби експлуатації вразливостей веб-застосунків.

Важливою перевагою рішень F5 є можливість інтеграції з іншими системами безпеки та моніторингу. BIG-IP підтримує взаємодію з SIEM-системами, зовнішніми сервісами аналізу загроз, каталогами користувачів та засобами централізованого управління подіями. Це дозволяє використовувати платформу як частину комплексної системи кіберзахисту організації, а не як ізольоване рішення.

Окремо слід відзначити масштабованість платформи BIG-IP. У разі зростання навантаження або кількості захищених застосунків можливе горизонтальне або вертикальне масштабування рішення. Підтримується кластеризація, що є критично важливим для сервісів з високими вимогами до безперервності роботи. Завдяки цьому BIG-IP може використовуватися як у невеликих організаціях, так і в інфраструктурах з великою кількістю користувачів і високим рівнем трафіку. [23]

З огляду на тенденції розвитку кіберзагроз, компанія F5 Networks постійно оновлює та вдосконалює свої продукти. Значна увага приділяється захисту від автоматизованих атак, бот-трафіку та складних багатовекторних загроз. Платформа BIG-IP еволюціонувала від інструменту балансування навантаження до повноцінного рішення для доставки та захисту застосунків, що відповідає сучасним вимогам кібербезпеки.

2.2 Модуль F5 Advanced WAF: можливості та функціонал

З огляду на зростаючу кількість атак саме на прикладному рівні, використання Web Application Firewall стає обов'язковим елементом захисту сучасних веб-застосунків. У межах платформи BIG-IP дану роль виконує модуль F5 Advanced WAF, який є розвитком класичного модуля ASM та орієнтований на протидію складним, у тому числі автоматизованим і багатовекторним загрозам. Advanced WAF реалізує комплексний підхід до аналізу HTTP/HTTPS-трафіку та дозволяє виявляти атаки ще на етапі їх формування, до безпосередньої взаємодії з веб-застосунком.

Архітектура модуля F5 WAF реалізована як складова платформи BIG-IP та тісно інтегрована з операційною системою TMOS. З точки зору обробки трафіку він працює в контектсі Virtual Server, де WAF-політика прив'язується безпосередньо до об'єкта, що приймає HTTP/HTTPS-запити. Первинну обробку трафіку забезпечує модуль LTM, після чого HTTP-запити передаються в модуль ASM для детального аналізу на прикладному рівні (рис. 2.3). Усередині модуля використовується власний механізм обробки, який розбирає URI, headers, cookies та payload відповідно до HTTP-протоколу. На основі цього аналізу застосовуються політика безпеки, набори сигнатур, аналіз IP-адреси і т д. Рішення про блокування, логування або пропуск запиту приймається ще до передачі трафіку на backend-сервери. Така архітектура дозволяє реалізувати повноцінний inline-захист без впливу на логіку роботи застосунку та з мінімальними затримками. [24]

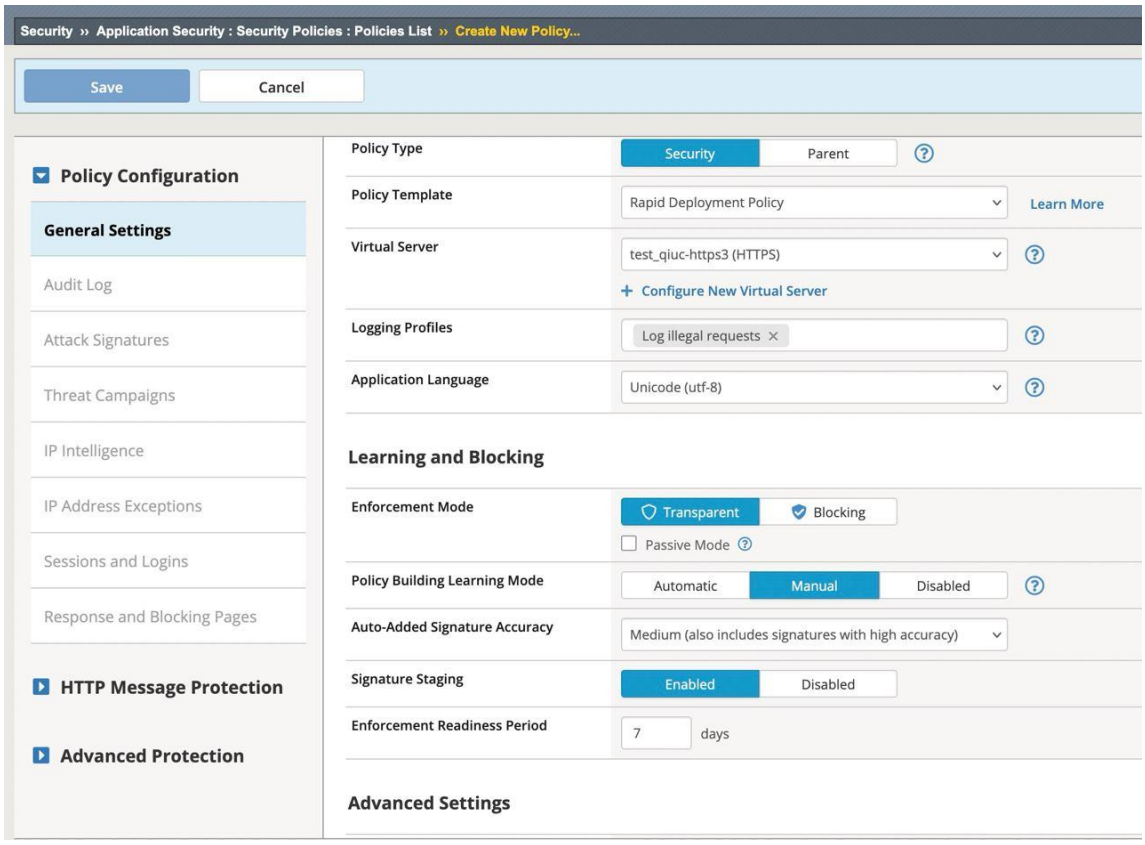


Рис. 2.3. Процес створення базової політики безпеки

В ASM модулі реалізація позитивної моделі безпеки базується на детальному описі легітимної поведінки веб-застосунку у межах Security Policy. Налаштування починається з формування профілю додатку, де визначається його тип, використовувані HTTP-методи, структура URI та очікувані формати даних. У процесі налашутвання WAF-політики формується перелік дозволених URL, параметрів, cookies та headers, які вважаються коректними для конкретного сервісу (рис. 2.4). Для кожного параметра можуть задаватися різні типи даних, допустима довжина, регулярні вирази та обмеження значень. [25]

Позитивна модель, як правило, використовується разом із режимами Transparent та Blocking. На етапі впровадження політика працює в режимі Transparent, де всі порушення лише логуються, що дозволяє скоригувати правила без ризику блокування легітимного трафіку. Після стабілізації політики застосовується Blocking Mode, у якому будь-яке відхилення від створених налаштувань автоматично блокується. Такий підхід забезпечує високий рівень захисту від атак, які складно виявити сигнатурними методами.

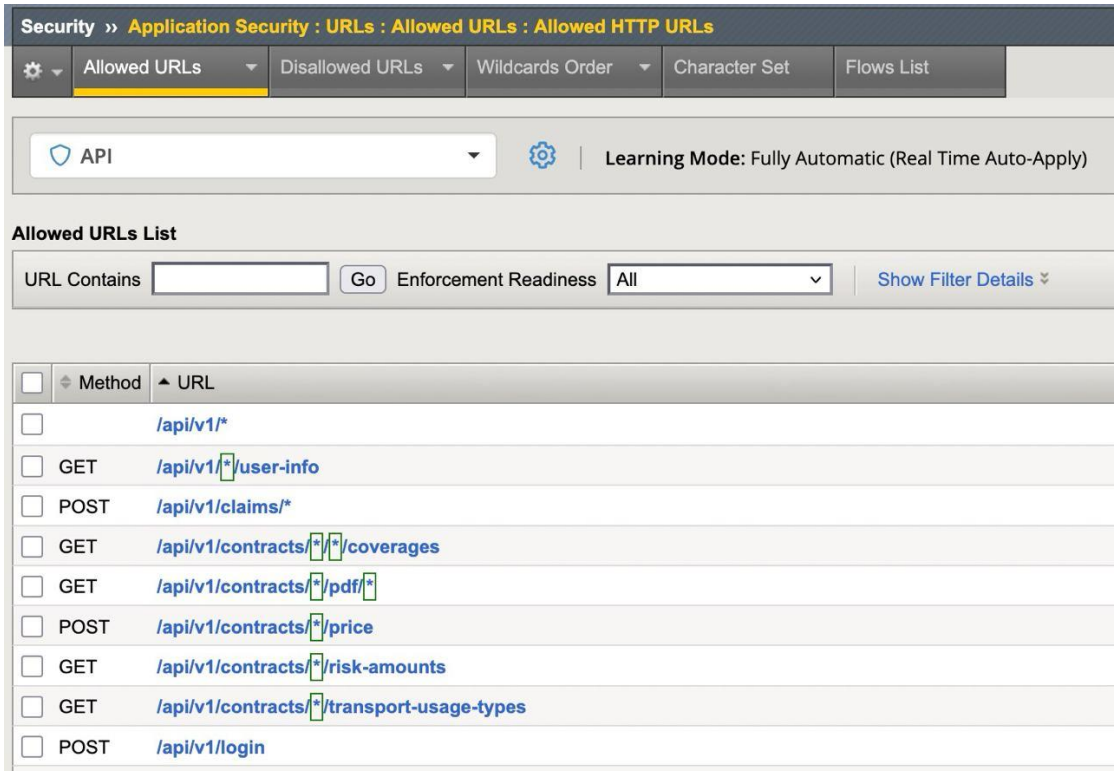


Рис. 2.4. Додавання дозволених URL, як приклад реалізації позитивної моделі безпеки

Негативна модель безпеки створена на основі сигнатурного аналізу. Одним з головних елементів тут є Attack Signatures, які згруповані у Signature Sets і регулярно оновлюються компанією F5. Кожна сигнатура описує характерні ознаки конкретного типу атаки, зокрема структуру вмісту, послідовність символів, аномальні HTTP-заголовки або специфічні параметри запиту.

Під час обробки трафіку WAF застосовує сигнатури до різних частин HTTP-запиту: URI, поле параметрів, вміст пакету або заголовки. Для підвищення точності виявлення використовується контекстний аналіз, тобто сигнатури прив'язуються до конкретних параметрів або типів даних. Наприклад, сигнатури для виявлення SQL ін'єкцій може застосовуватися лише до параметрів, визначених як numeric або database-related, що зменшує кількість хибних спрацювань (рис. 2.5).

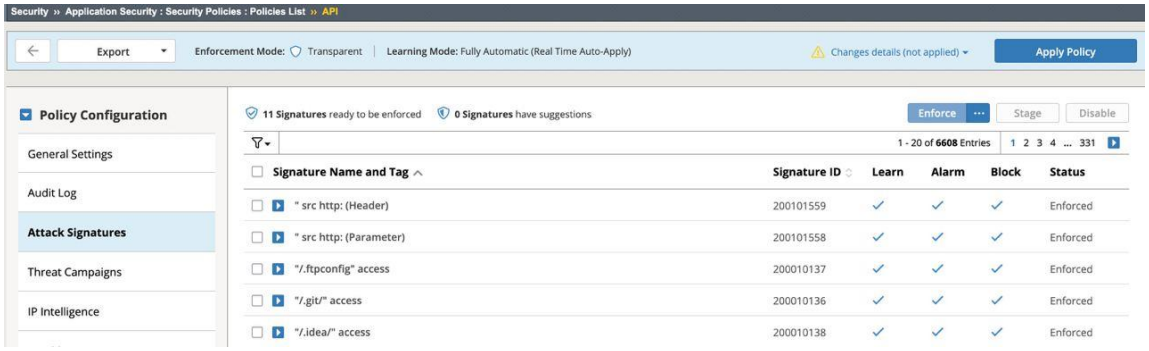


Рис. 2.5. Реалізація негативної моделі безпеки, шляхом додавання списку сигнатур

Налаштування негативної моделі включає вибір рівня суворості (Signature Staging), де нові або потенційно чутливі сигнатури спочатку працюють у режимі Detection Only. Це дозволяє оцінити їх вплив на легітимний трафік перед переведенням у Blocking. Адміністратор також може керувати окремими сигнатурами, вимикати їх або змінювати дію у межах конкретної політики безпеки:

- Learn - в цьому випадку система ніяк не сповіщає адміністратора та не блокує нелегітимні запити. Тільки приймає цю інформацію для покращення політики безпеки.
- Alarm - при ввімкнені цієї опції система записує в журнал сповіщення про нелегітимний запит, але не блокує його і пропускає на сервер.
- Block - система одразу реагує на прояв атаки і блокує запит, не пропускаючи його далі.

Негативна модель особливо ефективна для швидкого захисту від масових та автоматизованих атак, оскільки для цього не потрібно глибокого розуміння логіки застосунку. У поєднанні з постійними оновленнями сигнатур вона забезпечує актуальний рівень захисту від відомих експлойтів і типових векторів атак. [26]

Практична цінність цього пристрою полягає саме у поєднанні обох підходів. Платформа дозволяє використовувати гібридну модель безпеки, де позитивна модель формує базове уявлення про допустиму поведінку застосунку, а негативна - забезпечує оперативне блокування відомих типів атак. Такий підхід значно підвищує рівень захисту та зменшує ймовірність обходу механізмів безпеки.

Процес створення політики безпеки в модулі F5 Advanced WAF є гнучким і може бути реалізований кількома способами залежно від складності веб-застосунку та вимог до рівня безпеки. На практиці виділяють ручний та автоматичний підходи до налаштування політики безпеки.

Ручне налаштування політики застосовується у випадках, коли структура застосунку добре відома або необхідний максимальний контроль над правилами фільтрації. У межах цього підходу адміністратор самостійно створює та описує URL, визначає допустимі HTTP-методи, налаштовує параметри та їх типи даних, а також задає перелік дозволених типів файлів для завантаження. Такий підхід забезпечує високий рівень точності, однак потребує значних зусиль по часу і глибокого розуміння логіки застосунку.

Автоматичний та напівавтоматичний підходи реалізуються за допомогою Learning Mode. У цьому режимі пристрій аналізує реальний трафік, що проходить через віртуальний сервер, і на його основі формує рекомендації щодо створення URL, параметрів та дозволених структур запитів. Залежно від налаштувань, система може або автоматично застосовувати ці зміни, або виносити їх на підтвердження адміністратора.

Окремо слід відзначити можливість створення політики на основі Swagger файлу. У цьому випадку структура API імпортується безпосередньо в WAF, після чого автоматично генеруються URL, методи та параметри відповідно до опису інтерфейсів. Такий підхід особливо ефективний для захисту API, тому що дозволяє швидко сформувати коректну позитивну модель безпеки з мінімальним ризиком помилок. [27]

З точки зору функціональних можливостей, F5 Advanced WAF здатний захищати веб-застосунки від широкого спектра загроз. До них належать класичні атаки з переліку OWASP Top 10, а також ефективно протидіє автоматизованим атакам, що виконуються ботами, зокрема спробам підбору облікових даних, скануванню вразливостей і масовому надсиланню шкідливих запитів.

В модулів ASM механізми bot protection базуються на аналізі поведінки клієнтів, а не лише на сигнатурному підході. Платформа враховує такі параметри, як частота запитів, послідовність дій, використання HTTP-заголовків та відхилення від типової поведінки браузера. На основі цього трафік класифікується як легітимний, підозрілий або зловмисний.

Для протидії ботам є вбудовані механізми JavaScript Challenge, CAPTCHA та взаємодію з IP Intelligence, що дозволяє виявляти запити з відомих ботнетів.

Захист від атак типу Denial of Service реалізований як окремий набір механізмів, орієнтованих саме на прикладний рівень. На відміну від класичних мережевих DoS-атак, прикладні атаки часто маскуються під легітимний трафік і спрямовані на виснаження ресурсів веб-сервера. Для захисту від таких атак використовується функціонал DoS Protection профілей, які дозволяють контролювати інтенсивність і характер запитів.

Одним із важливих елементів є можливість визначення так званих Heavy URL — ресурсів, обробка яких є найбільш ресурсоємною для серверної частини. Для таких URL можуть встановлюватися окремі порогові значення, що обмежують кількість запитів за певний проміжок часу. Ці порогові значення можуть налаштовуватися вручну або формуватися автоматично шляхом аналізу нормального трафіку. Автоматичний режим дозволяє системі самостійно визначити базовий рівень навантаження та реагувати на аномальні відхилення.

Додатково в даному модулі реалізовано функціонал Behavioral DoS Protection, який базується на поведінковому аналізі клієнтів. Система формує профіль нормальної взаємодії з застосунком і виявляє аномалії, такі як різке зростання частоти запитів або їх нетипові патерни. У разі виявлення атаки можуть застосовуватися різні дії — від логування до тимчасового блокування або застосування challenge-механізмів. Такий підхід дозволяє ефективно протидіяти складним DoS-атакам без впливу на легітимних користувачів.

Додатковим механізмом захисту додатків є функціонал Threat Campaigns. Даний функціонал дозволяє виявляти координовані атаки, які можуть тривати протягом певного часу та змінювати свою поведінку. Threat Campaigns базуються на аналізі глобальної інформації про загрози, що збирається компанією F5 з великої кількості впроваджень по всьому світу. Завдяки цьому платформа може ідентифікувати нові шаблони атак та застосовувати відповідні захисні механізми ще до того, як вони набудуть масового характеру.

Також, ще одним додатковим рівнем захисту є IP Intelligence, який забезпечує репутаційний аналіз запитів. WAF може використовувати інформацію про IP-адреси, пов'язані з ботнетами, проксі, анонімними мережами або відомими зловмисними активностями. На основі цих даних можливо блокувати або обмежувати доступ для підозрілих джерел ще до детального аналізу вмісту запитів. Такий підхід дозволяє зменшити навантаження на систему та підвищити ефективність захисту. [28]

Окремої уваги заслуговує функціонал логування та моніторингу подій безпеки в модулі WAF, який забезпечує повну прозорість дій системи та дозволяє

ефективно аналізувати інциденти. Модуль веде детальні журнали, у яких фіксуються не лише заблоковані атаки, а й підозрілі дії, що відповідають політикам безпеки, але не призвели до негайного блокування. Це дозволяє виявляти потенційні загрози на ранніх етапах і оцінювати поведінку користувачів та клієнтів. Логи включають детальну інформацію про тип атаки, URL, параметри запиту, IP-адресу та застосовані правила політики.

Для полегшення аналізу та кореляції подій підтримується інтеграція з SIEM-системами різних виробників (IMB, Splunk, ArcSight та інші), що дозволяє централізовано збирати, обробляти та аналізувати інформацію про інциденти. Через цю інтеграцію можна автоматизувати процеси реагування на загрози, відстежувати тренди атак та вдосконалювати політики безпеки. Таким чином, модуль стає частиною комплексної системи моніторингу та реагування, забезпечуючи не лише захист, а й постійний контроль за станом безпеки веб-застосунків.

Гарним привілеєм даного модулю є регулярне оновлення сигнатур та механізмів виявлення атак. Компанія F5 постійно відстежує появу нових вразливостей і технік атак у глобальному масштабі, після чого оновлює базу загроз, що використовується модулем. Це дозволяє системі оперативною реагувати на нові типи атак, включаючи модифіковані SQL Injection, XSS, Remote File injection та інші сучасні загрози.

Оновлення сигнатур можуть застосовуватися без зупинки роботи системи, що критично для продуктивних середовищ із високим навантаженням. Платформа підтримує як автоматичне, так і ручне оновлення, що дає адміністратору гнучкість у плануванні змін та контролю за їх впливом на поточний трафік. Крім того, постійне оновлення бази загроз дозволяє підтримувати актуальність негативної моделі безпеки, забезпечуючи ефективне блокування відомих атак і мінімізацію ризику компрометації веб-застосунків. Завдяки цьому Advanced WAF здатний зберігати високий рівень захисту навіть у довгостроковій перспективі, залишаючись надійним компонентом комплексної системи кібербезпеки. [29]

Висновки до розділу 2

Було зроблено огляд компанії F5 Networks та проведено детальний аналіз її платформи BIG-IP як комплексного рішення для захисту веб-застосунків. Було оглянуто історію розвитку компанії, ключові напрями діяльності та особливості модульної архітектури платформи, що забезпечує гнучкість і масштабованість при побудові системи безпеки. Розглянуто різні форм-фактори розгортання BIG-IP, включно з апаратними пристроями, віртуальними та хмарними рішеннями, що дозволяє адаптувати платформу до конкретних інфраструктурних потреб організації.

Окремо було досліджено модуль F5 WAF, його архітектуру та функціональні можливості. Було описано принципи реалізації позитивної та негативної моделей безпеки, механізми захисту від прикладних атак, bot та DoS-атак. Оглянуто можливості логування та моніторингу подій безпеки, включно з інтеграцією з SIEM-системами. Також було відзначено важливість регулярного оновлення сигнатур та механізмів виявлення атак, що забезпечує актуальність захисту та підтримання високого рівня безпеки веб-застосунків у довгостроковій перспективі.

Розділ 3 ПРАКТИЧНЕ ДОСЛІДЖЕННЯ ЗАХИСТУ ВЕБ-ЗАСТОСУНКУ ЗА ДОПОМОГОЮ F5 WAF

3.1 Установлення ключових вимог до практичного дослідження та опис лабораторного середовища

Метою практичного дослідження захисту веб-застосунку з використанням Web Application Firewall на базі рішення F5 BIG-IP є перевірка ефективності застосування WAF для протидії типовим загрозам веб-застосункам, а також набуття практичних навичок налаштування, тестування та аналізу роботи засобів веб-захисту. Основним результатом практичного дослідження має стати підтвердження того, що використання F5 Advanced WAF дозволяє суттєво підвищити рівень безпеки веб-застосунку шляхом виявлення та блокування атак на рівні HTTP/HTTPS, а також забезпечує можливість аналізу інцидентів безпеки.

Основним результатом практичного дослідження має стати підтвердження того, що використання F5 Advanced WAF дозволяє суттєво підвищити рівень захищеності веб-застосунку за рахунок своєчасного виявлення та блокування атак на прикладному рівні. Очікується, що WAF забезпечить ефективний контроль HTTP/HTTPS-трафіку, виявлення аномальної активності та автоматичне реагування на спроби порушення безпеки. [30]

Для досягнення поставленої мети в рамках практичного дослідження визначено такі основні вимоги:

- застосування негативної моделі безпеки, що передбачає виявлення та блокування відомих типів атак на основі сигнатур, правил та поведінкових шаблонів;

- застосування позитивної моделі безпеки, яка базується на формуванні дозволеного профілю легітимної поведінки веб-застосунку та блокуванні відхилень від нього;
- реалізація захисту від атак типу brute force, спрямованих на підбір облікових даних користувачів;
- аналіз спрацювань WAF та розбір інцидентів шляхом перегляду журналів подій, повідомлень про порушення політик безпеки та логів атак.

Практичне дослідження виконується в ізольованому лабораторному середовищі, що моделює реальну інфраструктуру взаємодії клієнта з веб-застосунком через засоби захисту. [31]

Лабораторне середовище складається з таких основних компонентів:

- віртуальна машина на базі операційної системи Linux, яка виконує роль веб-сервера. На даній машині розгорнуто вразливий веб-застосунок у Docker-контейнері, що використовується для імітації типових атак на веб-ресурси;
- F5 BIG-IP VE з увімкненим модулем Advanced Web Application Firewall, який розміщується між клієнтом та веб-сервером і забезпечує фільтрацію, аналіз та блокування шкідливого трафіку;
- віртуальна машина на базі операційної системи Ubuntu, яка використовується в ролі клієнта для доступу до веб-застосунку та виконання тестових атак і перевірок.

Обране лабораторне середовище дозволяє на практиці відтворити типову схему захисту веб-застосунку із застосуванням WAF, оцінити роботу позитивної та негативної моделей безпеки, а також проаналізувати реакцію системи на різні типи атак у контрольованих умовах (рис. 3.1).

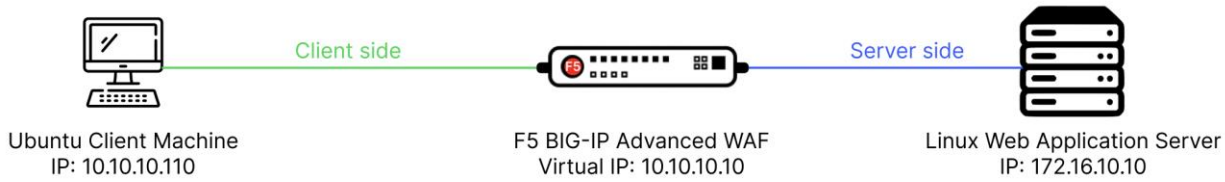


Схема 3.1 Мережева схема лабораторного середовища

3.2 Налаштування F5 WAF для захисту веб-застосунку

На першому етапі виконується створення базової політики безпеки F5 Advanced WAF. Під час ініціалізації політики визначаються основні параметри: назва політики безпеки, Віртуальний сервер до якого застосувати дану політику, режим роботи (Transparent або Blocking), способ навчання політики (Automatic або Manual Learning), а також важливим налаштуванням є вибір способу логування: записування всіх запитів або тільки нелегітимних. Для практичного дослідження було вибрано повне логування, але на практиці в продуктивних впровадженнях зазвичай використовується журналювання тільки підозрілих запитів, для того щоб оптимізувати ресурси пристрою під інші, більш важливі задачі. Базова конфігурація політики забезпечує початковий рівень захисту та слугує основою для подальшого розширення функціональності WAF відповідно до вимог практичного дослідження (рис. 3.1). [32]

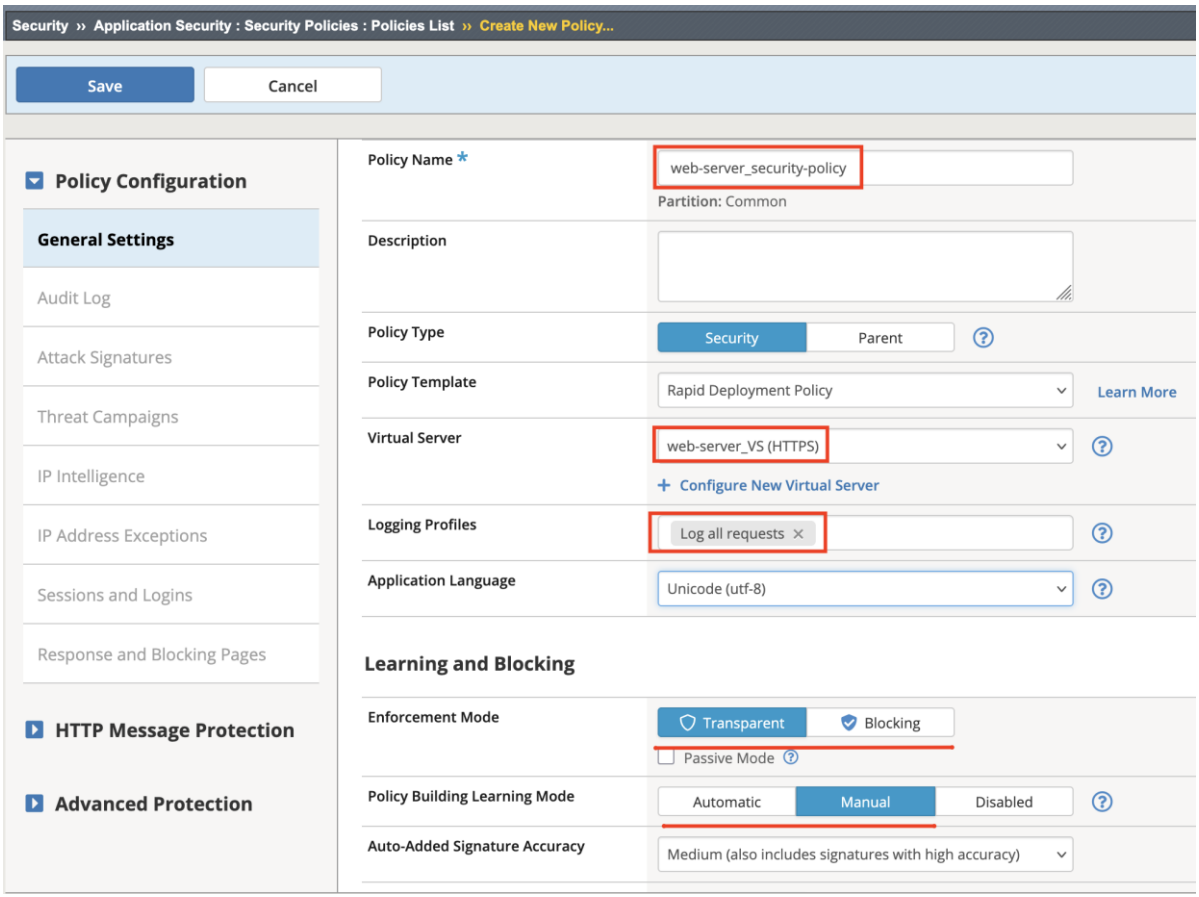


Рис. 3.1. Процес створення базової політики безпеки

Після створення базової політики безпеки наступним етапом буде реалізація позитивної моделі безпеки в межах політики безпеки. Налаштування контролю доступу до ресурсів веб-застосунку шляхом визначення дозволених та заборонених URL (рис. 3.2). Формується список Allowed URLs, до якого вносяться легітимні шляхи доступу до веб-ресурсів, що використовуються в процесі нормальної роботи застосунку (рис. 3.3). А також створюється список Disallowed URLs, який явно визначає список куди “ходити не можна” (рис. 3.4). Даний підхід дозволяє обмежити доступ виключно до відомих і дозволених кінцевих точок, зменшуючи ризик несанкціонованого звернення до прихованих або службових ресурсів.

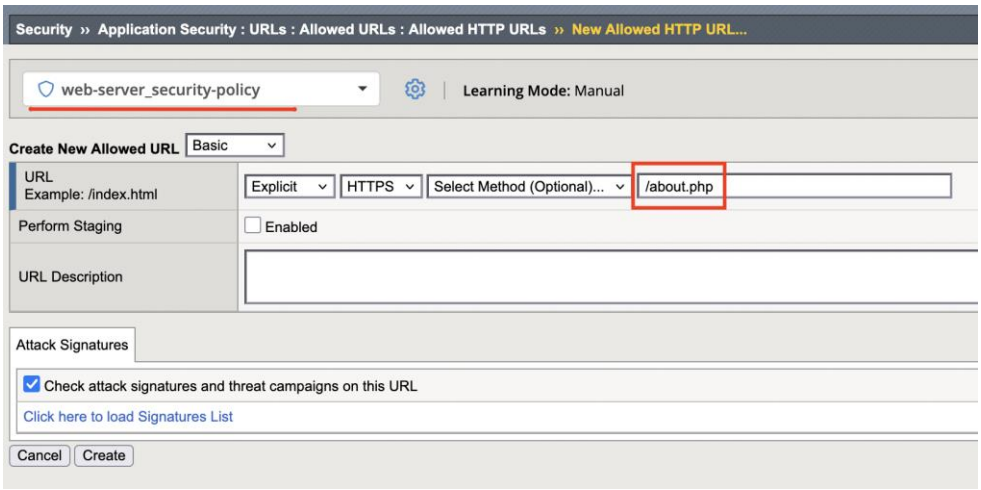


Рис. 3.2 Процес додавання Allowed URL

При додаванні URL-ів обов'язково вказується до якої політики він додається та сам шлях. Він може бути прописаний явно (Explicit) або з використанням декількох допустимих значень (Wildcard). [33]

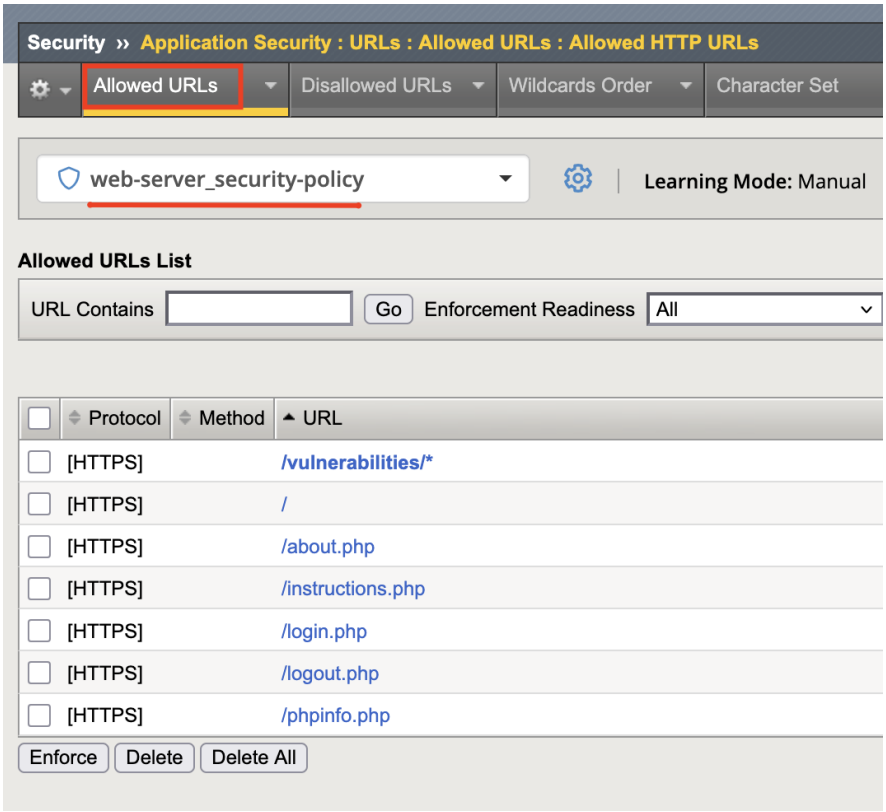


Рис. 3.3. Повністю сформований список дозволених URL-ів

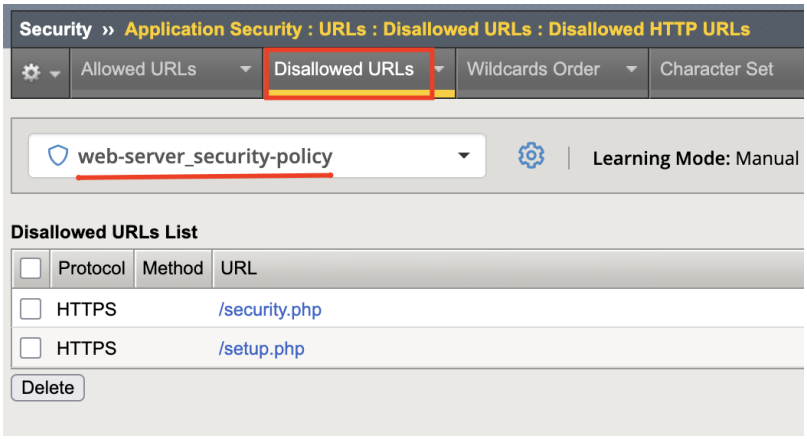


Рис. 3.4. Повністю сформований список заборонених URL-ів

Для зовнішніх користувачів було закрито доступ до “/security.php” та “/setup.php” , оскільки це адміністративні панелі , доступ до яких повинен бути тільки у інженерів відповідальних за застосунок.

Для реалізації негативної моделі безпеки в політиці WAF активуються та налаштовуються набори сигнатур атак. В рамках даного дослідження основна увага приділяється захисту від атак типу SQL Injection та Cross-Site Scripting (XSS), які є одними з найбільш поширених і небезпечних загроз для веб-застосунків. Увімкнення відповідних сигнатур дозволяє виявляти шкідливі запити на основі характерних шаблонів і ознак атак, незалежно від конкретної структури веб-ресурсів (рис. 3.5). [34]

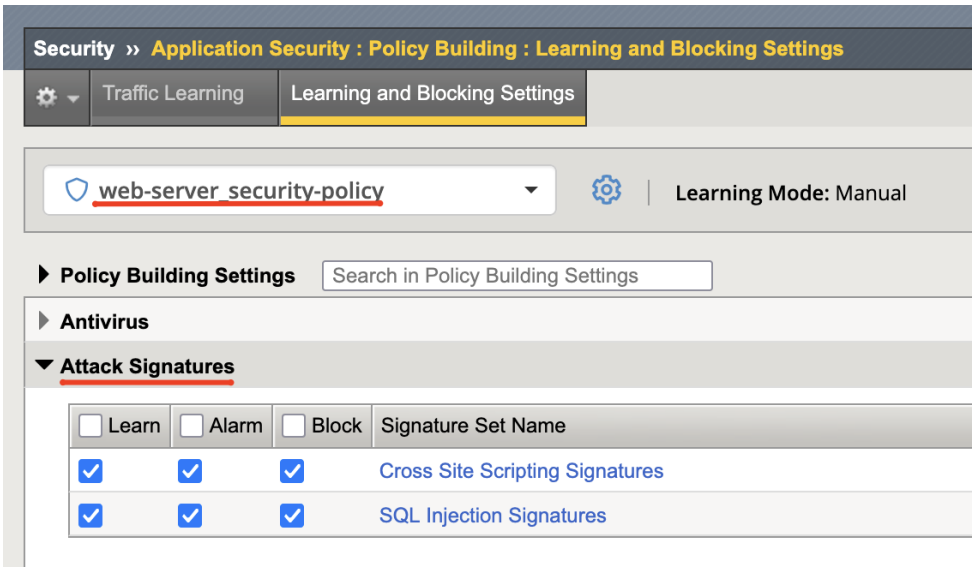


Рис. 3.5. Процес додавання відповідних наборів сигнатур

Окремим етапом налаштування політики безпеки є захист від автоматизованих атак перебору облікових даних користувачів методом грубої сили. Для цього використовується захист від атак типу Brute Force, шляхом визначення точних параметрів сторінки входу, та ідентифікації умов успішних спроб (рис. 3.6). Всі інші спроби вважаються нелегітимними і до них застосовуються правила мітигації згідно налаштувань (рис. 3.7).

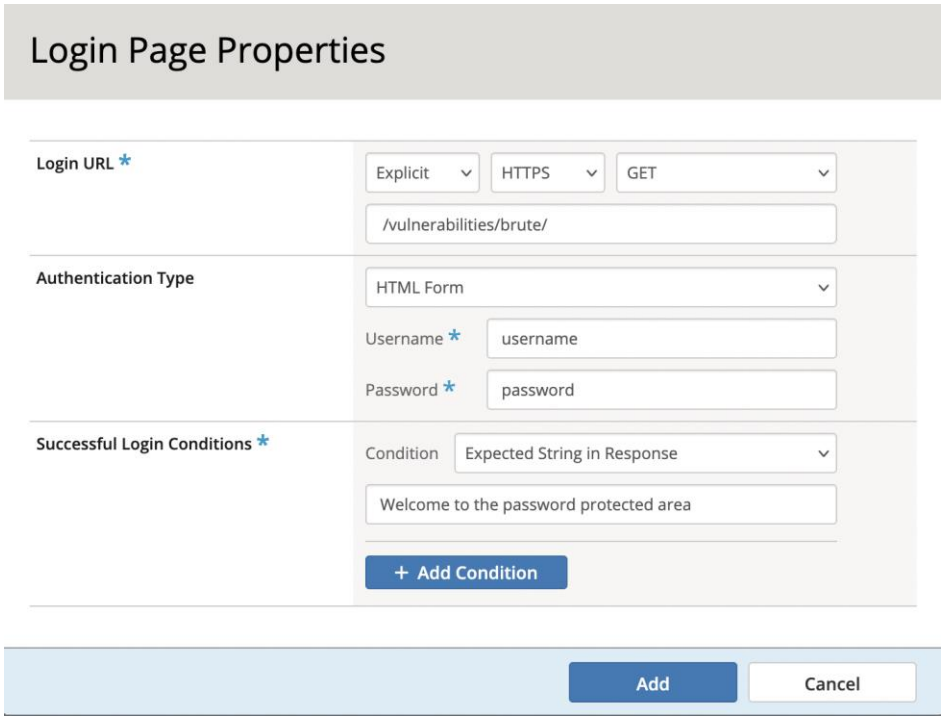


Рис. 3.6. Процес додавання відповідних наборів сигнатур

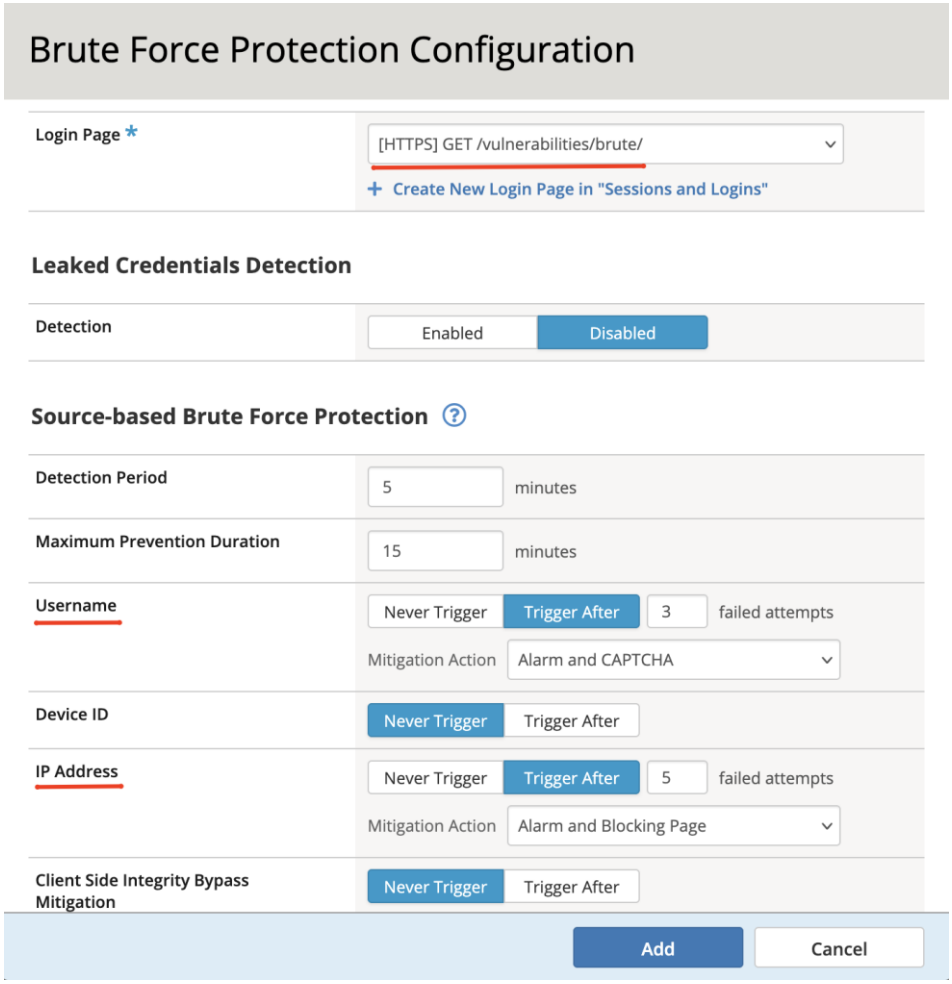


Рис. 3.7. Налаштування правил мітигації для протидії атакам перебору грубої сили

Таким чином була створена комплексна політика безпеки, яка поєднує позитивну та негативну моделі захисту, забезпечує протидію поширеним веб-атакам, та захищає сторінку входу від автоматизованих атак. [35]

3.3 Тестування ефективності захисту

Тестування ефективності захисту веб-застосунку виконується з метою практичної перевірки працездатності налаштованої політики безпеки та оцінки її здатності виявляти й блокувати типові атаки на прикладному рівні. На

початковому етапі тестування веб-застосунок перевіряється в Transparent режимі, це дозволяє зафіксувати базову поведінку застосунку та підтвердити наявність уразливостей, які можуть бути використані зловмисником. У даному режимі тестові запити, що імітують атаки, успішно обробляються веб-застосунком та призводять до очікуваного небезпечного результату, що підтверджує вразливість середовища. [36]

Першочергово було протестовано препозитивну модель безпеки, зокрема контролю доступу до ресурсів за допомогою списків дозволених і заборонених URL. На початковому етапі здійснюється доступ до різних шляхів веб-застосунку без обмежень, включно з нетиповими або небажаними URL (рис. 3.8, 3.9).

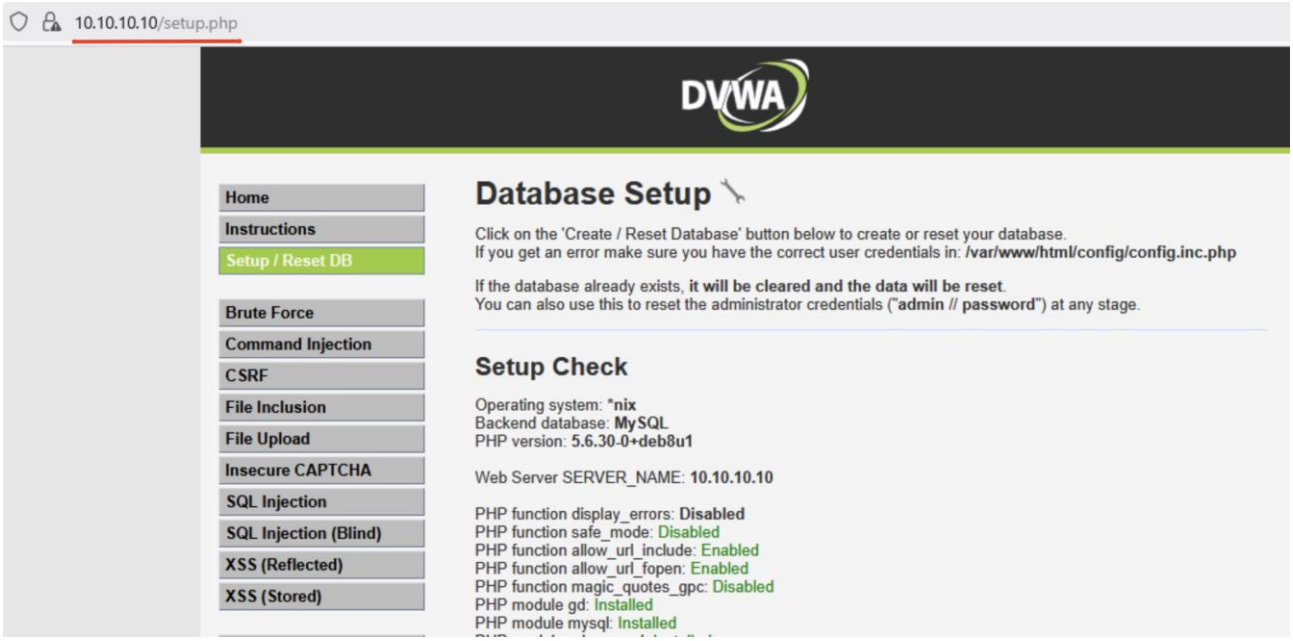


Рис. 3.8. Зовнішній користувач може відкрити URL із забороненого списку

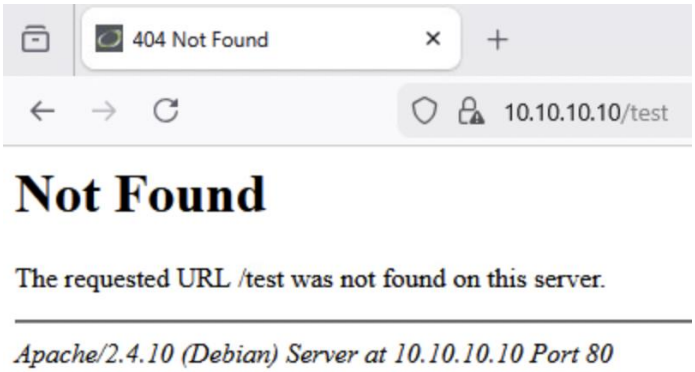


Рис. 3.9. Зовнішній користувач може відкрити URL, якого немає в списку дозволених

Наступним етапом відбувається тестування спроб атакувати веб-застосунок за допомогою SQL ін'єкції та отримати список імен користувачів зареєстрованих в цьому додатку (рис. 3.10). [37]

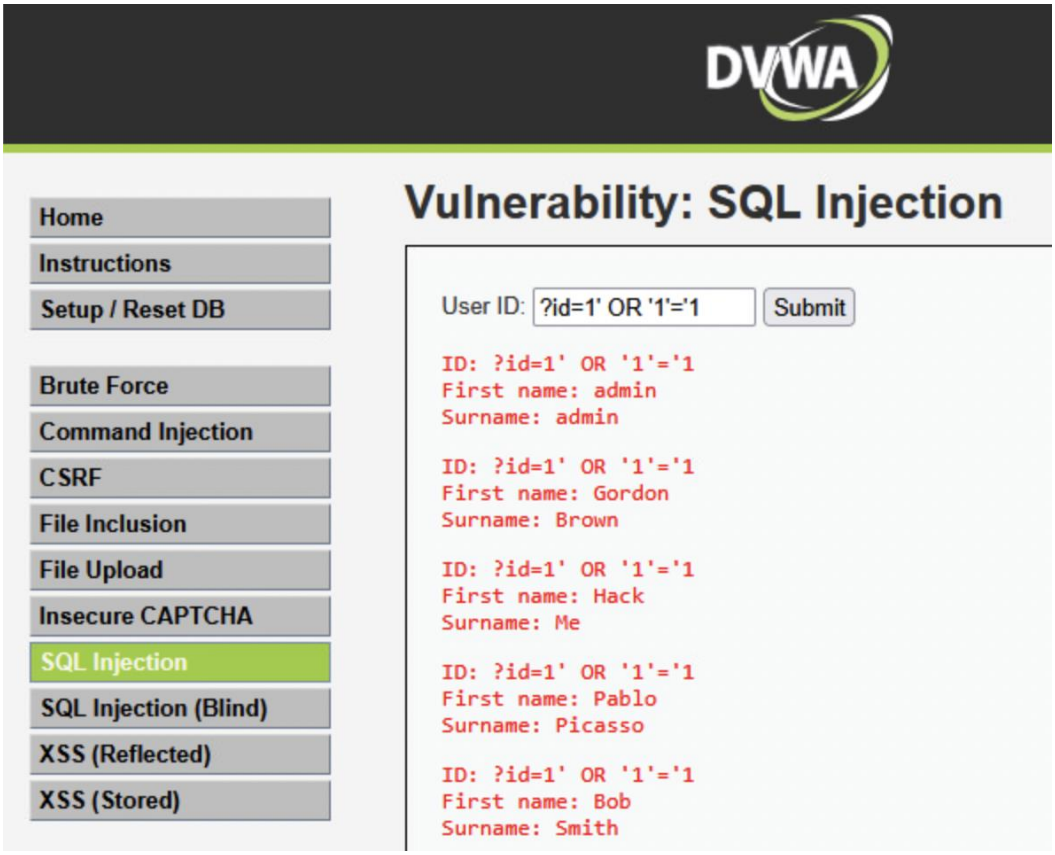


Рис. 3.10. Приклад успішної атаки на веб-сервер, шляхом відправки в форму зловмисного SQL запиту

Аналогічним чином виконується тестування атаки типу Cross-Site Scripting (XSS). Спочатку шкідливий скрипт передається у вхідному параметрі "name". Веб-застосунку без активного захисту обробляє цей скрипт та на стороні клієнта відображає вспливаюче повідомлення про успішну атаку (рис. 3.11).

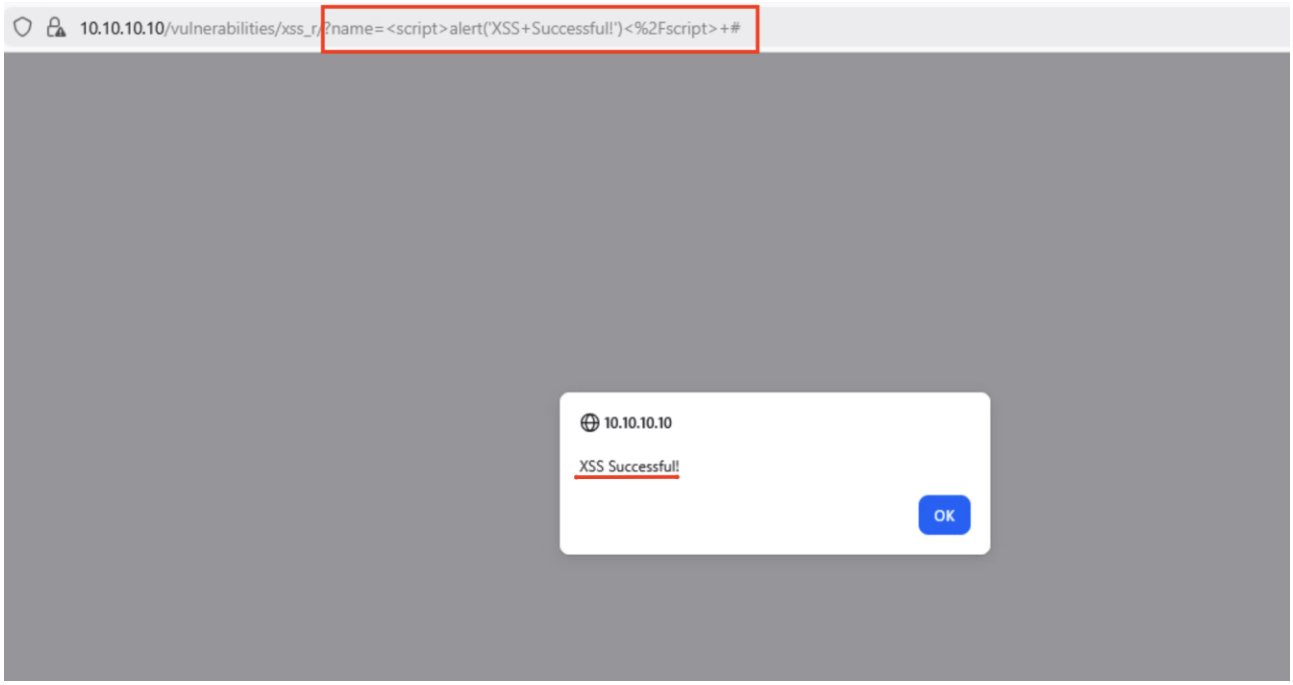


Рис. 3.11. Приклад реалізації успішної атаки типу Cross-Site Scripting

Наступним етапом дослідження було протестовано, що веб-застосунок вразливий до атак методу грубої сили (Brute force). Тестування було здійснено з використанням командного рядка з використанням спеціалізованого програмного забезпечення для проведення тестів на проникнення - утиліти Hydra. Даний інструмент призначений для автоматизованого перебору облікових даних користувачів, що дозволяє перевірити рівень захищеності механізмів автентифікації веб-застосунків. Для реалізації атаки додатково було сформовано словник із можливими варіантами паролів, який використовувався Hydra під час процесу підбору (рис. 3.12).

Bash команда, яка використовувалась під час тестування:

```
# hydra -V -l admin -P passwords.txt 10.10.10.10 https-get-form
"/vulnerabilities/brute/:username=^USER^&password=^PASS^&Login=Log
in:password incorrect.:H=Cookie: security=low;
PHPSESSID=eb2757f0c328bdc50c1fb6eb6f52c382"
```

(1)

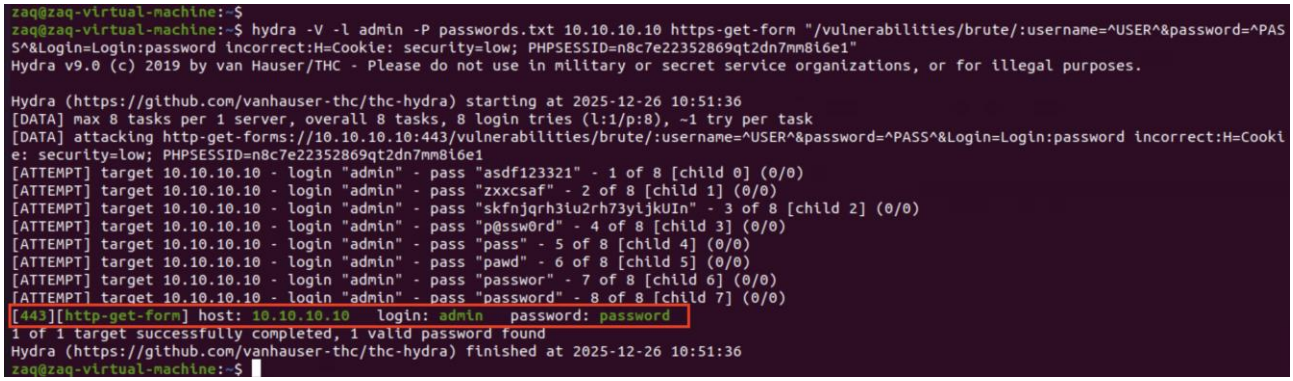


Рис. 3.12. Успішний підбор паролю

Зі скріншоту видно, що атака пройшла успішно і автоматизована програма змогла підібрати пароль до сторінки входу тестового веб-застосунки, при вимкнутій системі блокування.

Про успішну реалізацію атак свідчать записи, зафіксовані в журналі подій. Аналіз журналів показує, що система безпеки коректно ідентифікує потенційно небезпечні запити, однак у режимі прозорої роботи не застосовує блокувальні дії, а лише фіксує відповідні події (рис. 3.13-15).

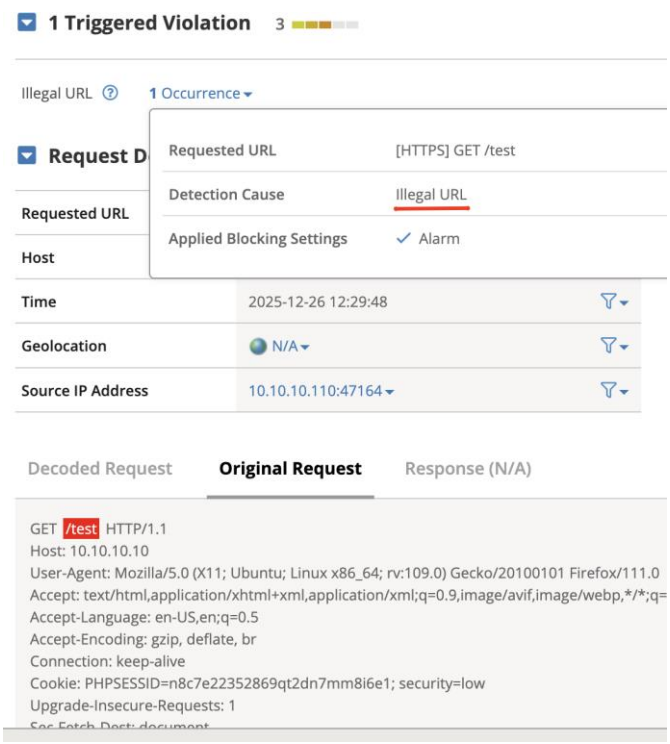


Рис. 3.13. Запис в журналі про доступ до забороненого URL

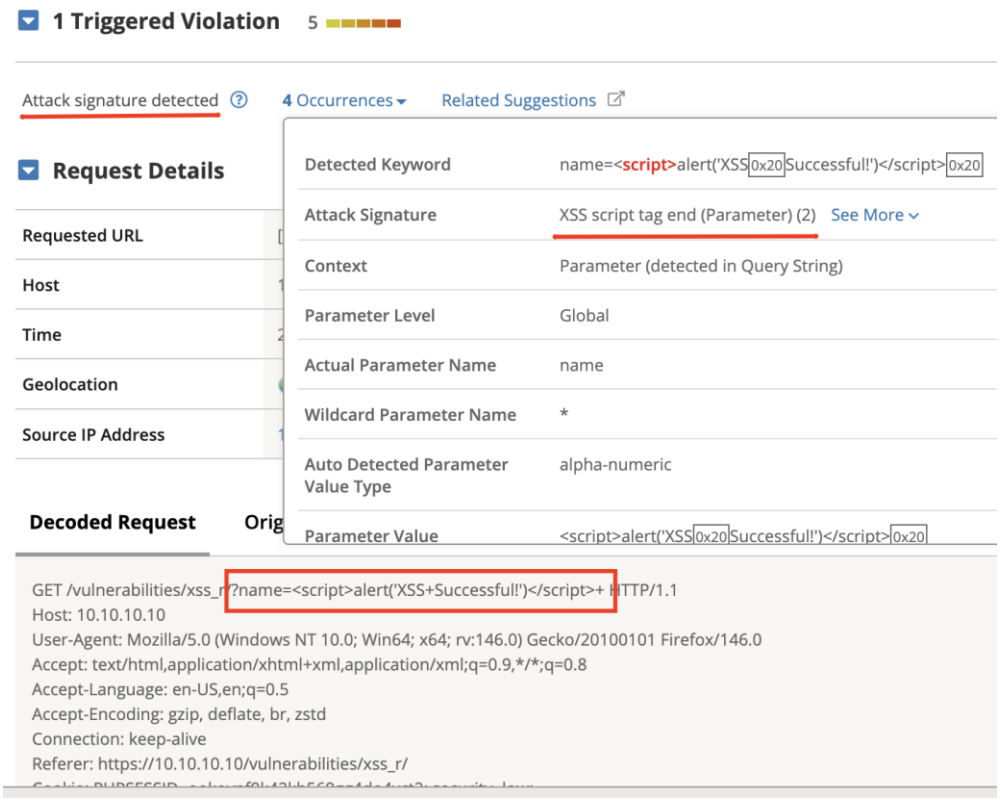


Рис. 3.14. Запис в журналі про вдалу спробу SQL-ін’єкції

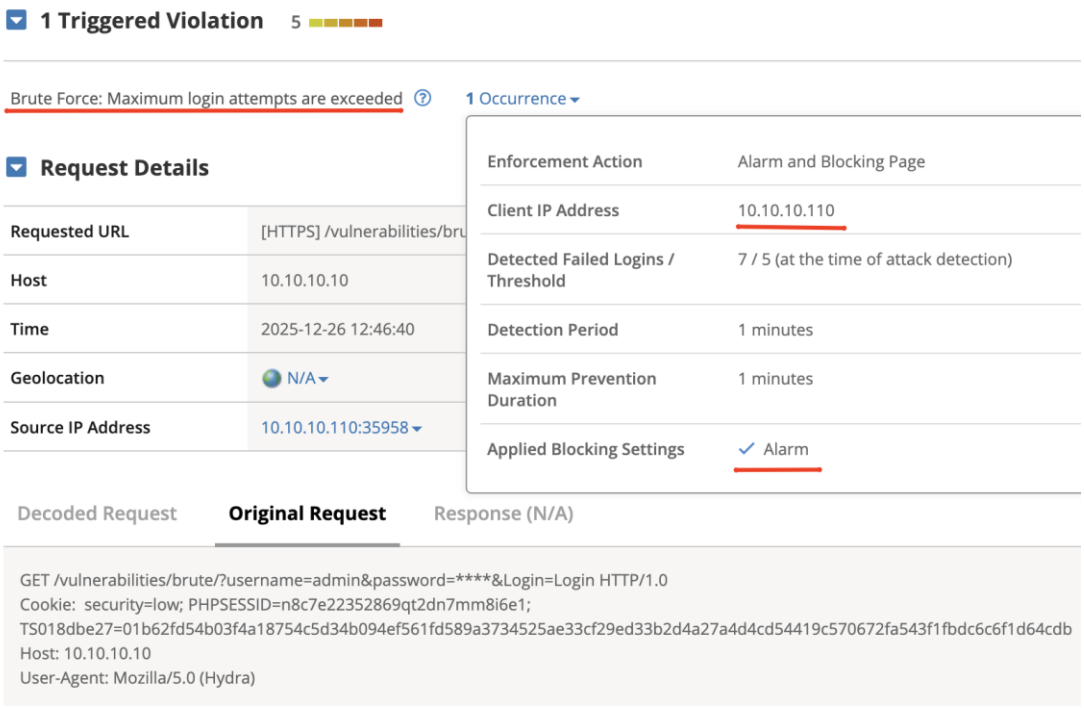


Рис. 3.15. Запис в журналі про вдалу спробу підбору облікових даних користувачів

На скріншотах, отриманих під час тестування, наочно продемонстровано, що система F5 Advanced WAF у режимі Transparent виконує функцію моніторингу та сповіщення про виявлені загрози, однак не блокує шкідливі запити. Усі потенційно небезпечні запити передаються далі до веб-сервера, що дозволяє використовувати даний режим для початкового аналізу трафіку, виявлення вразливостей та коректного налаштування політик безпеки без впливу на доступність веб-застосунку. [38]

Наступним етапом практичного дослідження є переведення політики безпеки у режим Blocking. Активація даного режиму означає перехід від пасивного моніторингу до активного застосування захисних механізмів, за яких система не лише виявляє потенційні загрози, але й автоматично блокує нелегітимні запити, не допускаючи їх передачі до веб-сервера.

Після ввімкнення режиму блокування повторно виконуються ті самі тестові сценарії, які раніше використовувалися на етапі Transparent. Зокрема, здійснюється спроба доступу до неіснуючого або забороненого URL. На відміну від попереднього етапу, така спроба завершується невдало - запит блокується на рівні WAF, а веб-сервер не отримує відповідного звернення (рис. 3.16). Факт блокування підтверджується відповідним записом у журналі подій системи безпеки (рис. 3.17).

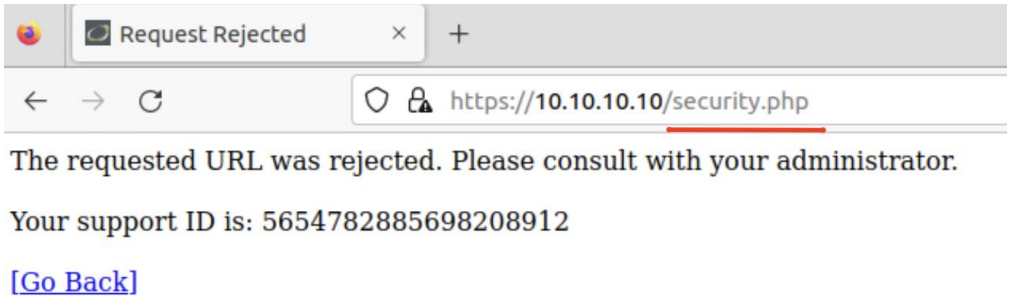


Рис. 3.16. Заблокований запит на заборонений URL

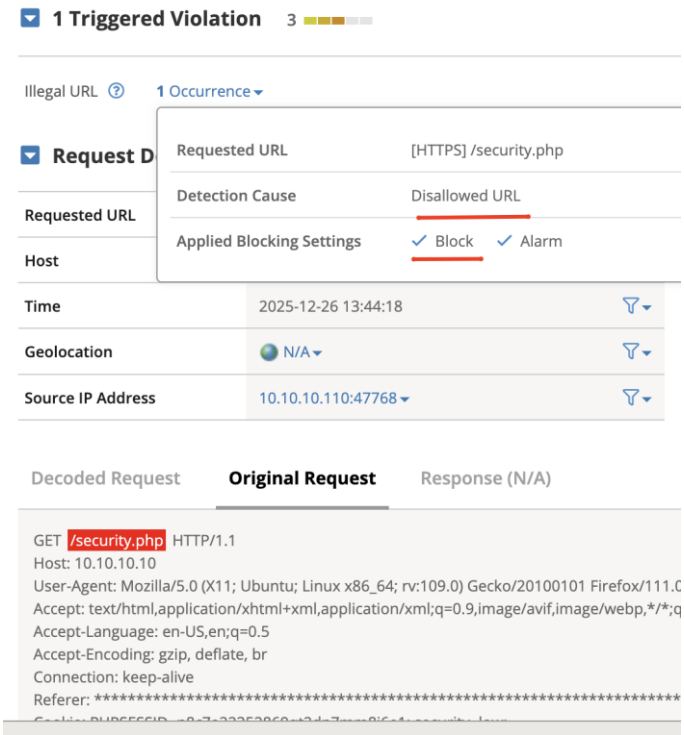


Рис. 3.17. Запис в журналі подій про заблокований запит на заборонений URL

Аналогічним чином перевіряється ефективність захисту від атак типу SQL Injection (рис. 3.18). Після активації режиму Blocking спроба передавання шкідливих SQL-конструкції у параметрах HTTP-запиту призводить до негайного блокування запиту WAF. У результаті виконання атаки стає неможливим, і зловмисник не отримує доступу до внутрішніх даних веб-застосунку, зокрема до списку користувачів. Інформація про виявлення та блокування SQL-ін'єкції фіксується в журналі подій, що дозволяє однозначно підтвердити коректну роботу негативної моделі безпеки (рис. 3.19). [39]

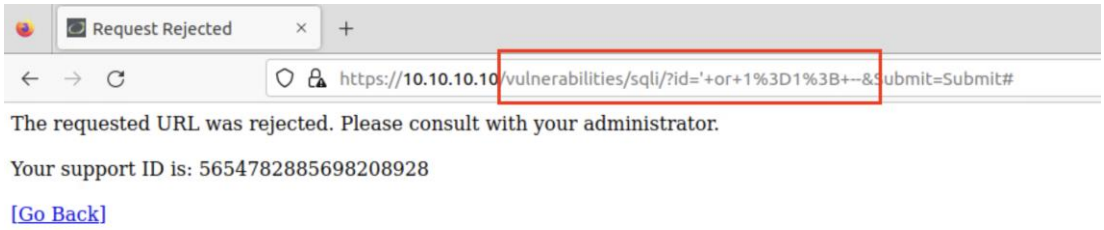


Рис. 3.18. Заблокована спроба SQL-ін'єкції

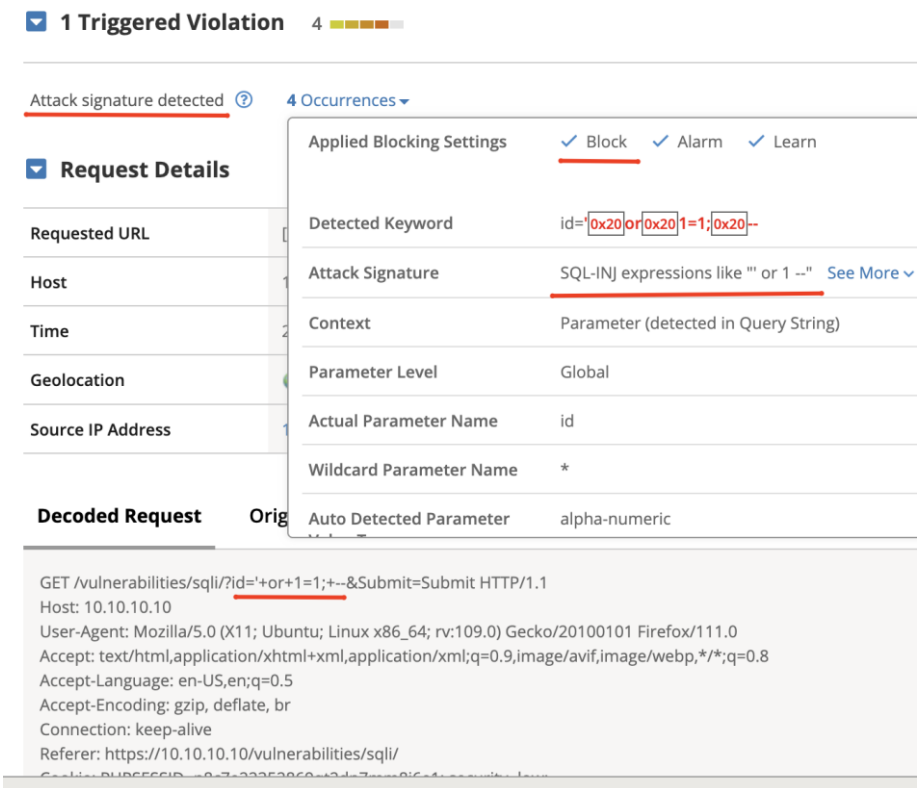


Рис. 3.19. Запис в журналі подій про заблоковану SQL атаку

Окремо перевіряється ефективність захисту від автоматизованого підбору облікових даних (рис. 3.20). Під час повторної спроби виконання атаки типу brute force з використанням спеціалізованого програмного забезпечення встановлено, що атака є невдалою. Унаслідок спрацювання механізмів блокування WAF шкідливе програмне забезпечення не отримує відповіді від реального веб-сервера, що унеможливлює подальший підбір паролів. Відповідна подія також відображається в журналі безпеки, де зафіксовано застосування правил мітигації та блокування джерела атаки (рис. 3.21). [40]

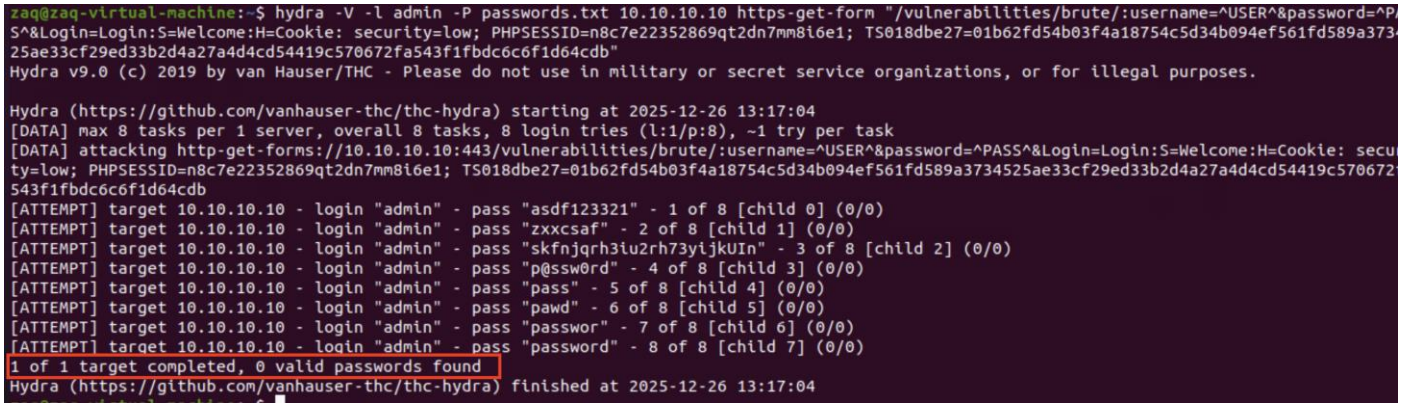


Рис. 3.20. Невдала спроба підбору паролю автоматизованими засобами

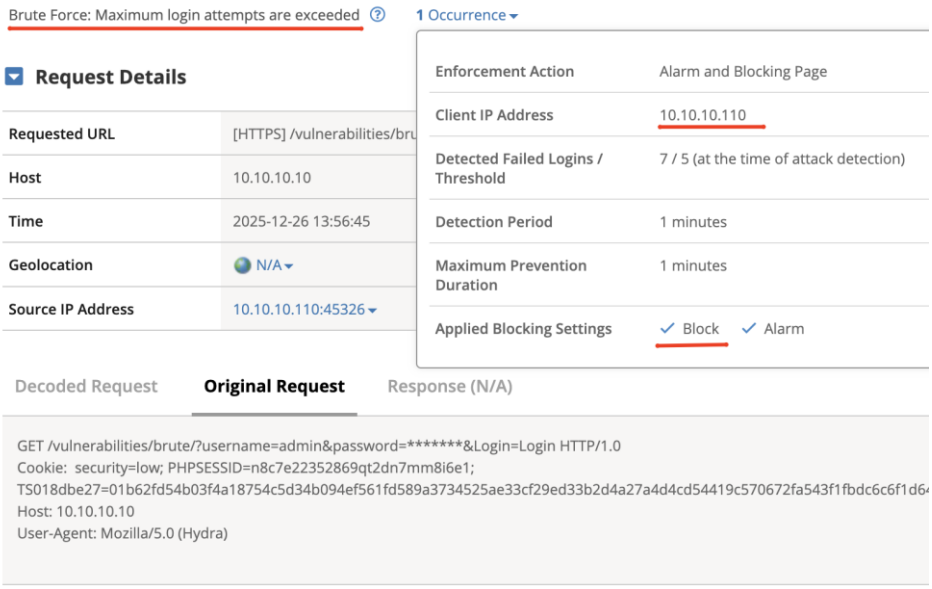


Рис. 3.21. Запис в журналі про заблоковану спробу автоматизованого підбору паролю

На скріншотах, отриманих під час тестування, наочно продемонстровано, що система F5 Advanced WAF у режимі Blocking забезпечує активний захист веб-застосунку шляхом блокування нелегітимних запитів і недопущення їх передачі до веб-сервера. На відміну від режиму Transparent, даний режим дозволяє ефективно протидіяти спробам доступу до заборонених ресурсів, виконанню SQL- та XSS-ін’єкцій, а також атакам типу brute force, що підтверджує доцільність використання Blocking режиму в продуктивному середовищі. [41]

Висновки до розділу 3

Було розглянуто практичні аспекти захисту веб-застосунку з використанням Web Application Firewall на базі рішення F5 BIG-IP з модулем Advanced WAF. Проаналізовано вимоги до практичного дослідження, сформовано лабораторне середовище та визначено основні підходи до побудови захисту веб-ресурсів на прикладному рівні.

Під час налаштування системи безпеки було встановлено, що ефективний захист веб-застосунку має базуватися на поєднанні позитивної та негативної моделей безпеки. Реалізовано політику WAF, яка забезпечує контроль доступу до ресурсів, виявлення та блокування атак типу SQL Injection, Cross-Site Scripting, а також протидію автоматизованому підбору облікових даних.

У результаті проведеного тестування підтверджено, що використання F5 Advanced WAF дозволяє ефективно виявляти та блокувати нелегітимні запити, запобігати успішній експлуатації вразливостей веб-застосунку та забезпечувати моніторинг інцидентів безпеки.

ВИСНОВКИ

Дослідження було зосереджено на питаннях забезпечення безпеки веб-застосунків шляхом використання Web Application Firewall на базі рішення F5 Networks. У роботі розглянуто актуальні проблеми захисту веб-ресурсів в умовах зростання кількості та складності атак, а також проаналізовано роль WAF як одного з ключових елементів сучасної системи інформаційної безпеки.

Оглянуто основні типи загроз для веб-застосунків, зокрема атаки з переліку OWASP Top 10, які на сьогодні залишаються найбільш поширеними та небезпечними. До них віднесено SQL-ін’єкції, міжсайтовий скриптинг (XSS), атаки типу Brute Force, збої автентифікації та авторизації, а також помилки конфігурації систем безпеки. Проаналізовано характерні особливості кожного типу загроз та їхній вплив на конфіденційність, цілісність і доступність інформації. Встановлено, що значна частина вразливостей виникає внаслідок недоліків на етапах проєктування, розробки або експлуатації веб-застосунків, що обумовлює необхідність застосування комплексного підходу до їх захисту.

Проаналізовано архітектуру та функціональні можливості платформи F5 BIG-IP, а також спеціалізованого модуля F5 Advanced WAF. Детально вивчено принципи роботи WAF, зокрема використання сигнатурного аналізу, механізмів поведінкового виявлення атак, захисту від автоматизованих ботів та централізованого логування подій безпеки. Встановлено, що інтеграція даного рішення дозволяє ефективно протидіяти сучасним веб-загрозам, підвищити рівень захищеності інформаційних ресурсів та забезпечити додатковий рівень безпеки без необхідності внесення змін у програмний код веб-застосунків.

У практичній частині роботи було проведено поглиблене дослідження функціонування веб-застосунку в умовах реалізації різних типів атак з метою оцінки ефективності застосування Web Application Firewall. Практичне тестування включало моделювання атак типу несанкціонованого доступу до заборонених або неіснуючих ресурсів, двох типів ін'єкцій спрямованих на заволодіння даними користувачів, а також Brute Force, спрямованих на

механізми автентифікації. Під час експериментів аналізувалася реакція WAF на підозрілий трафік, коректність спрацювання правил безпеки та процес блокування шкідливих запитів. Отримані результати підтвердили, що правильно налаштований Web Application Firewall забезпечує своєчасне виявлення атак, знижує ризики несанкціонованого доступу та сприяє підвищенню загального рівня захищеності веб-застосунку.

У підсумку можна зробити висновок, що використання Web Application Firewall на базі рішення F5 Networks є ефективним та доцільним підходом до захисту веб-застосунків від сучасних кіберзагроз. Запропоновані підходи та результати дослідження можуть бути використані при проєктуванні та впровадженні систем захисту веб-ресурсів з метою підвищення загального рівня інформаційної безпеки.

ПЕРЕЛІК ПОСИЛАНЬ

1. Rosero G. E. C., Guevara-Vega C. P., Landeta-López P. Website Protection: An Evaluation of the Web Application Firewall, Data & Metadata, Vol. 4, 2025, pp. 190, [Електронний ресурс]. Режим доступу: https://dm.ageditor.ar/index.php/dm/article/view/190

2. Humaira H. N., Hadiana A. I., Ashaury H. Analisis Ketahanan Web Application Firewall Terhadap Serangan SQL Injection, Jurnal Ilmiah Wahana Pendidikan, Vol. 10, No. 5, 2024, pp. 403-412, [Електронний ресурс]. Режим доступу: https://jurnal.peneliti.net/index.php/JIWP/article/view/6341

3. Mulyo Efendi. Analisa WAF (Web Application Firewall) Menggunakan NGINX Terhadap serangan SQL Injection, B. Sc. Thesis, Universitas Mercu Buana, Jakarta, 2021, [Електронний ресурс]. Режим доступу: http://repository.mercubuana.ac.id/72387/1/41516320022%20-%20MULYO%20EFENDI%20-%2002%20Cover.pdf

4. Wu C., Chen J., Zhu S., Feng W., Du R., Xiang Y. WAFBOOSTER: Automatic Boosting of WAF Security Against Mutated Malicious Payloads, arXiv, 2025, [Електронний ресурс]. Режим доступу: https://arxiv.org/abs/2501.14008

5. Montaruli B., Floris G., Scano C., et al. ModSec-AdvLearn: Countering Adversarial SQL Injections with Robust Machine Learning, arXiv, 2023, [Електронний ресурс]. Режим доступу: https://arxiv.org/abs/2308.04964

6. Li Jinfeng, Li Haorong. Evolution of Application Security based on OWASP Top 10 and CWE/SANS Top 25 with Predictions for the 2025 OWASP Top 10, Proceedings of the International Conference on Inventive Computation Technologies (ICICT-2025), IEEE Xplore, 2025, pp. 1178-1186 [Електронний ресурс]. Режим доступу до ресурсу: https://www.researchgate.net/publication/392031422_Evolution_of_Application_Security_based_on_OWASP_Top_10_and_CWESANS_Top_25_with_Predictions_for_the_2025_OWASP_Top_10

7. Amouei M., Rezvani M., Fateh M. RAT: Reinforcement-Learning-Driven and

Adaptive Testing for Vulnerability Discovery in Web Application Firewalls, arXiv, 2023, [Електронний ресурс]. Режим доступу: https://arxiv.org/abs/2312.07885

8. SQL Injection Attack Vulnerabilities of Web Application and Detection, International Journal of Computer Applications, 2023, [Електронний ресурс]. Режим доступу: https://ijcaonline.org/archives/volume185/number38/32942-2023923192/

9. Web application firewall based on machine learning models, PMC – NCBI, 2025, [Електронний ресурс]. Режим доступу: https://pmc.ncbi.nlm.nih.gov/articles/PMC12453791/

10. OWASP Top Ten Web Application Security Risks, Open Web Application Security Project (OWASP), 2025 [Електронний ресурс]. Режим доступу: https://owasp.org/www-project-top-ten/

11. Behl A., Behl K. Cybersecurity and Cyberwar: What Everyone Needs to Know, Oxford University Press, New York, 2017, 312 p.

12. Stuttard D., Pinto M. The Web Application Hacker’s Handbook: Finding and Exploiting Security Flaws, 2nd Edition, Wiley Publishing, 2011, 912 p.

13. Scarfone K., Mell P. Guide to Intrusion Detection and Prevention Systems (IDPS), NIST Special Publication 800-94, National Institute of Standards and Technology, USA, 2012, [Електронний ресурс]. Режим доступу до ресурсу: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-94.pdf

14. Poncelet M., Me L., Debar H. A Survey of Web Application Security, IEEE Security & Privacy Magazine, Vol. 20, No. 3, 2022, pp. 28–37.

15. Web application firewall based on machine learning models, PMC – National Center for Biotechnology Information (NCBI), 2025 [Електронний ресурс]. Режим доступу до ресурсу: https://pmc.ncbi.nlm.nih.gov/articles/PMC12453791/

16. Annas M., Tjut Adek R., Afrillia Y. Web Application Firewall (WAF) Design to Detect and Anticipate Hacking in Web-Based Applications, Journal of Advanced Computer Knowledge and Algorithms, Vol. 1, No. 3, 2024, pp. 52-63, [Електронний ресурс]. Режим доступу до ресурсу: https://ojs.unimal.ac.id/jacka/article/view/16315

17. Aref S., Bassam K. M. H. D. Web Application Firewall Using Machine

Learning and Feature Engineering, Security and Communication Networks, 2022, Article ID 5280158, pp. 1-14, [Електронний ресурс]. Режим доступу до ресурсу: https://www.hindawi.com/journals/scn/2022/5280158/

18. Harini K. Enhancing Web Application Protection with ModSecurity and Reverse Proxy, International Journal for Research in Applied Science and Engineering Technology, Vol. 13, No. 3, 2025, pp. 1476-1480, [Електронний ресурс]. Режим доступу до ресурсу: https://doi.org/10.22214/ijraset.2025.67528

19. Babaey, Vahid, Arun Ravindran. GenXSS: an AI-Driven Framework for Automated Detection of XSS Attacks in WAFs. arXiv, 2025 [Електронний ресурс]. Режим доступу: https://arxiv.org/abs/2504.08176

20. Li, Ke, Heng Yang, and Willem Visser. Evolutionary Multi-Task Injection Testing on Web Application Firewalls. arXiv, 2022 [Електронний ресурс]. Режим доступу: https://arxiv.org/abs/2206.05743

21. Amouei, Mohammadhossein, Mohsen Rezvani, and Mansoor Fateh. RAT: Reinforcement-Learning-Driven and Adaptive Testing for Vulnerability Discovery in Web Application Firewalls. arXiv, 2023 [Електронний ресурс]. Режим доступу: https://arxiv.org/abs/2312.07885

22. Exclusive-Networks. Advanced Application Threats Require an Advanced WAF, white paper, 2021 [Електронний ресурс]. Режим доступу до ресурсу: https://www.exclusive-networks.com/tr/wp-content/uploads/sites/32/2021/07/what-makes-a-waf-advanced.pdf

23. Top Web Application Security, Safe.Security Research Report, Safe.Security, 2025 [Електронний ресурс]. Режим доступу до ресурсу: https://safe.security/wp-content/uploads/web-application-security-threats.pdf

24. F5, Inc. Cloud Application Delivery and Security Products and Services [Електронний ресурс]. Режим доступу: https://www.f5.com/products#f5-platform

25. F5, Inc. BIG-IP Modules datasheet, [Електронний ресурс]. Режим доступу до ресурсу: https://cdn.studio.f5.com/files/k6fem79d/production/23d910005f6c3955333b0a8de2079e879e37475e.pdf

26. F5, Inc. Web Application Firewalls (WAFs) for Dummies. F5 Networks, 2025, [Електронний ресурс]. Режим доступу до ресурсу: https://www.f5.com/go/ebook/web-application-firewalls-for-dummies

27. SecureIQLab, LLC. 2022 Cloud Web Application Firewall CyberRisk Validation Report – F5 BIG-IP Virtual Edition. SecureIQLab, 2022, [Електронний ресурс]. Режим доступу до ресурсу: https://cdn.studio.f5.com/files/k6fem79d/production/de4a100072455bcc7579bf80aecfb13b03936edc.pdf

28. F5, Inc. Deployment and response methodologies | BIG-IP TMOS operations guide, [Електронний ресурс]. Режим доступу: https://my.f5.com/manage/s/article/K10635270

29. F5, Inc. What is a Web Application Firewall (WAF)? F5 Glossary, 2025, [Електронний ресурс]. Режим доступу до ресурсу: https://www.f5.com/glossary/web-application-firewall-waf

30. F5, Inc. Good WAF Security, Getting started with ASM, [Електронний ресурс]. Режим доступу до ресурсу: https://f5-agility-labs-waf.readthedocs.io/en/latest/class3/class3.html

31. F5, Inc. BIG-IP Advanced Web Application Firewall Datasheet. F5 Networks, 2025. [Електронний ресурс]. Режим доступу: https://www.f5.com/pdf/data-sheet/big-ip-advanced-waf-datasheet.pdf

32. F5, Inc. “Create Web Application Firewall”, F5 Distributed Cloud Technical Knowledge, 2025 [Електронний ресурс]. Режим доступу: https://docs.cloud.f5.com/docs-v2/web-app-and-api-protection/how-to/app-security/application-firewall

33. F5, Inc. F5 Agility Labs – Web Application Firewall Solutions, [Електронний ресурс]. Режим доступу: https://clouddocs.f5.com/training/community/waf/html/

34. F5, Inc. Succeeding with application security, [Електронний ресурс]. Режим доступу: https://my.f5.com/manage/s/article/K07359270

35. F5, Inc. Adding URLs to a Security Policy, [Електронний ресурс]. Режим

доступу: https://techdocs.f5.com/kb/en-us/products/big-ip_asm/manuals/product/asm-implementations-11-5-0/31.html

36. F5, Inc. Assigning Attack Signatures to Security Policies, [Електронний ресурс]. Режим доступу: https://techdocs.f5.com/kb/en-us/products/big-ip_asm/manuals/product/asm-implementations-11-5-0/31.html

37. Hydra Tool Documentation, [Електронний ресурс]. Режим доступу: https://www.kali.org/tools/hydra/

38. F5, Inc. Security Deployment Best Practices, [Електронний ресурс]. Режим доступу: https://techdocs.f5.com/en-us/bigiq-7-1-0/big-iq-security/security-deployment-best-practices.html

39. F5, Inc. Creating a Simple Security Policy, [Електронний ресурс]. Режим доступу: https://techdocs.f5.com/kb/en-us/products/big-ip_asm/manuals/product/asm-getting-started-13-0-0/2.html

40. F5, Inc. Configuring brute force attack protection, [Електронний ресурс]. Режим доступу: https://my.f5.com/manage/s/article/K18650749

41. F5, Inc. Deployment and response methodologies | BIG-IP TMOS operations guide, [Електронний ресурс]. Режим доступу: https://my.f5.com/manage/s/article/K10635270

ДОДАТКИ

ДОДАТКИ

Додаток А

Презентаційні матеріали до магістерської роботи

Микита М'ЯСНИКОВ

Захист веб-застосунків за допомогою WEB APPLICATION FIREWALL на базі рішення F5 NETWORKS

- Магістерська робота: Микита М'ЯСНИКОВ
- Спеціальність 125 «Кібербезпека та захист інформації»
- Фокус: веб-застосунки, WAF, F5 Networks, протидія атакам прикладного рівня

Актуальність дослідження

- Веб-застосунки є основою інформаційних систем комерційного і державного секторів.
- Кількість атак на прикладному рівні зростає, а традиційні L3/L4-засоби не аналізують логіку HTTP/HTTPS-запитів.
- Web Application Firewall забезпечує фільтрацію, контроль і блокування небезпечних запитів у режимі реального часу.

Мета, об’єкт і предмет роботи

- Мета: аналіз кібератак на веб-застосунки, дослідження принципів WAF та оцінка ефективності F5 Networks.
- Об’єкт: процеси забезпечення інформаційної безпеки веб-застосунків у мережевих системах.
- Предмет: способи кібератак і засоби захисту веб-застосунків із використанням Web Application Firewall на базі F5 Networks.

Основні завдання дослідження

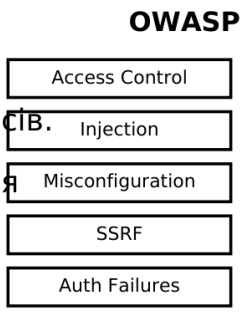
- Проаналізувати сучасні кіберзагрози та типові атаки на веб-застосунки.
- Дослідити підходи і засоби захисту веб-застосунків, зокрема Web Application Firewall.
- Розробити рекомендації щодо налаштування і застосування F5 Networks для підвищення безпеки веб-застосунків.

Структура магістерської роботи

- Розділ 1 - теоретичні основи захисту веб-застосунків і аналіз OWASP-загроз.
- Розділ 2 - архітектура F5 Networks, BIG-IP та можливості F5 Advanced WAF.
- Розділ 3 - практичне дослідження, налаштування WAF, тестування ефективності та оцінка результатів.

Сучасні загрози веб-застосункам

- Основні загрози: Broken Access Control, Cryptographic Failures, Injection, Security Misconfiguration, SSRF.
- Критичними наслідками є витік конфіденційних даних, модифікація інформації та порушення доступності сервісів.
- OWASP Top 10 використовується як базовий орієнтир для пріоритезації політик захисту.



Підходи до захисту веб-застосунків

- Захист на рівні коду: валідація вхідних даних, коректна автентифікація, авторизація і безпечна обробка помилок.
- Інфраструктурний захист: сегментація мережі, обмеження доступу, захищені протоколи обміну даними.
- Контроль HTTP-трафіку: WAF аналізує запити, блокує небезпечні шаблони і формує журнали подій.

Принципи роботи Web Application Firewall

- WAF розміщується між клієнтом і веб-сервером та аналізує прикладний трафік рівня L7.
- Перевіряються URI, параметри запитів, заголовки, cookies, сесії та вміст HTTP-повідомлень.
- Захист реалізується через сигнатури, позитивну модель безпеки, поведінковий аналіз і блокування аномалій.

F5 Networks і платформа BIG-IP

- F5 BIG-IP поєднує балансування навантаження, керування трафіком і захист прикладного рівня.
- Платформа працює як контрольна точка між користувачами, веб-застосунками і серверною інфраструктурою.
- Рішення доцільне для корпоративних і державних інформаційних систем із підвищеними вимогами до безпеки.

Можливості F5 Advanced WAF

- Захист від OWASP Top 10, SQL Injection, XSS, SSRF, автоматизованих атак і спроб підбору облікових даних.
- Підтримка політик безпеки, логування подій, навчання профілю застосунку та аналізу аномалій.
- Можливість поєднання WAF із балансуванням трафіку та SSL/TLS-термінацією.

Лабораторне середовище дослідження

- Практична частина передбачає моделювання захищуваного веб-застосунку та розміщення F5 WAF перед веб-сервером.
- Перевіряються сценарії проходження легітимного трафіку і блокування шкідливих запитів.
- Події безпеки аналізуються за журналами WAF для підтвердження факту виявлення атак.

Налаштування F5 WAF

- Створення віртуального сервера, пулу backend-серверів і профілю захисту веб-застосунку.
- Формування політики безпеки: URL, параметри, методи HTTP, сигнатури атак, режими блокування.
- Перехід від режиму навчання до режиму примусового застосування політик після валідації правил.

Тестування ефективності захисту

- Перевіряються типові атаки: SQL Injection, Cross-Site Scripting, підозрілі параметри, некоректні HTTP-запити.
- Оцінюється здатність WAF відрізняти легітимну активність від атак без порушення роботи застосунку.
- Результати тестів підтверджуються записами журналів подій і статусами блокування.

Оцінка результатів

- Використання F5 WAF підвищує рівень захисту веб-застосунку від атак прикладного рівня.
- Налаштовані політики дозволяють зменшити ризик експлуатації відомих уразливостей до їх усунення в коді.
- Ефективність рішення залежить від коректності політик, актуальності сигнатур і постійного моніторингу.

Практичні рекомендації

- Поєднувати WAF із захищеною розробкою, сегментацією мережі, оновленням компонентів і моніторингом подій.
- Регулярно переглядати політики, журнали блокування, профілі застосунків і винятки безпеки.
- Використовувати F5 WAF як елемент комплексної системи захисту, а не як єдиний засіб кібербезпеки.

Висновки

- Проаналізовано сучасні загрози веб-застосункам і обґрунтовано необхідність захисту рівня L7.
- Досліджено архітектуру та функціональні можливості Web Application Firewall на базі F5 Networks.
- Практичне застосування WAF дозволяє виявляти і блокувати типові атаки, підвищуючи стійкість веб-серверів.