

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

КАФЕДРА СИСТЕМ ТА ТЕХНОЛОГІЙ КІБЕРБЕЗПЕКИ

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Технологія захисту веб-ресурсів на базі WAAP»

зі спеціальності 125 Кібербезпека та захист інформації

(код, найменування спеціальності)

освітньо-професійної програми Інформаційна та кібернетична безпека

(назва програми)

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

Владислав САМОЙЛЕНКО

(підпис)

Виконав: здобувач вищої освіти групи БСДМ-62

САМОЙЛЕНКО Владислав

(прізвище, ім'я)

Керівник

д-р техн. наук, професор ГАЙДУР

Галина

(науковий ступінь, вчене звання, прізвище, ім'я)

Рецензент

(науковий ступінь, вчене звання, прізвище, ім'я)

Київ 2025

РЕФЕРАТ

Текстова частина кваліфікаційної роботи: 87 сторінки, 52 рисунки, 3 таблиці, 11 джерел.

Об'єкт дослідження – процес виявлення загроз кібернетичної безпеки в веб додатках за допомогою WAAP.

Предмет дослідження – WAAP(web application and API protection) вендора F5 Networks.

Мета роботи – оглянути рішення та його концепцію, розробити рекомендації щодо забезпечення інформаційної безпеки веб додатку на базі рішення F5xС WAAP.

Методи дослідження – опрацювання літератури за даною темою, аналіз експлуатаційної документації, практичний досвід налаштування та керування системою на реальному досвіді базуючись на best practice вендорів.

У результаті дослідження визначено призначення WAAP, його основні можливості та складові, а також проведено оцінку ефективності застосування цієї технології для захисту веб застосунків і API організацій різного профілю.

Розроблено практичні рекомендації для фахівців з кібербезпеки щодо використання WAAP як базової технології захисту сучасної цифрової інфраструктури, що дозволяє забезпечити стійкість застосунків перед актуальними та майбутніми кіберзагрозами.

Галузь використання – кібербезпека веб додатків

ІНФОРМАЦІЙНІ ВЕБ РЕСУРСИ КОМПАНІЙ, WAAP, ТЕХНОЛОГІЯ ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ ВЕБ ДОДАТКІВ, API, КІБЕРБЕЗПЕКА, КІБЕРЗАГРОЗИ

ABSTRACT

Text of the qualification work: 87 pages, 52 figures, 3 tables, 11 sources.

Object of research – the process of detecting cybersecurity threats in web applications using WAAP.

Subject of research – WAAP (Web Application and API Protection) by the vendor F5 Networks.

Purpose of the work – to review the solution and its concept, and to develop recommendations for ensuring the information security of a web application based on the F5xC WAAP solution.

Research methods – review of literature on the topic, analysis of operational documentation, and practical experience in configuring and managing the system based on real-world use and vendor best practices.

As a result of the research, the purpose of WAAP, its core capabilities, and components were identified, and the effectiveness of this technology for protecting web applications and APIs of organizations with different profiles was evaluated.

Practical recommendations were developed for cybersecurity professionals on using WAAP as a baseline technology for securing modern digital infrastructure, enabling application resilience against current and future cyber threats.

Field of application – web application cybersecurity.

CORPORATE WEB RESOURCES, WAAP, WEB APPLICATION SECURITY TECHNOLOGY, API, CYBERSECURITY, CYBER THREATS.

ЗМІСТ

	Стор.
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	11
ВСТУП	12
1 ДОСЛІДЖЕННЯ ПРОБЛЕМИ ЗАХИСТУ ВЕБ ДОДАТКІВ.....	14
1.1 Актуальність проблеми забезпечення інформаційної безпеки.....	14
1.2 Порівняння класичної архітектури на базі WAF та сучасної на базі WAAP	22
1.3 Аналіз підходів до захисту веб додатків , які вирішує WAAP.....	30
2 АНАЛІЗ МЕТОДІВ ТА ЗАСОБІВ ЗАХИСТУ ВЕБ ДОДАТКІВ НА БАЗІ F5 NETWORKS	33
2.1 Огляд технології Distributed Cloud та рішення F5xC	33
2.2 Ключові компоненти архітектури F5xC	35
3 ТЕХНОЛОГІЯ НАЛАШТУВАННЯ F5xC WAAP ТА РЕКОМЕНДАЦІЙ ЩОДО РОБОТИ ТА КОНФІГУРАЦІЇ	49
3.1 Варіант налаштування консолі та публікація додатку.....	49
3.2 Налаштування технології F5xC WAAP.....	54
3.3 Експериментальні результати дашбордів та статистики F5xC WAAP.....	60
3.4 Рекомендації щодо застосування технології захисту безпеки додатків.....	82
ВИСНОВКИ	85
ПЕРЕЛІК ПОСИЛАНЬ	87
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація)	88

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

WAAP – Web Application and API Protection
WAF – Advanced Web Application Firewall/Web Application Firewall
API - Application Programming Interface
LTM – Local Traffic Manager
AFM – Advanced Firewall Manager
DNS – Domain Name System
APM – Access Policy Manager
ADC – Application Delivery Controller
ПЗ – Програмне забезпечення
ROI – Return of Investments
PII - Personally Identifiable Information
SQL- Structured Query Language
XSS - Cross-Site Scripting,
CSRF - Cross-Site Request Forgery
HTTP - HyperText Transfer Protocol
CVE - Common Vulnerabilities and Exposures
VE – Virtual Edition
HW - Hardware
ML – Machine Learning
SaaS – Software as a Service
ДЦ – Дата Центр
HA/DR – High Availability/Disaster Recovery
ІІІ – Штучний Інтелект
TMOS – Traffic Management Operation System
PoP – Point of Presents
FQDN – Fully Qualified Domain Name
DLP – Data Leakage Prevention

ВСТУП

Актуальність дослідження: Стрімкий розвиток цифрових технологій та зростаюча залежність сучасного суспільства від веб-застосунків і онлайн-сервісів формують нові вимоги до забезпечення інформаційної безпеки. Якщо ще кілька років тому безпека асоціювалася переважно з мережевою інфраструктурою та класичними методами захисту на рівні L3/L4, то сьогодні ключовим об'єктом атак стають саме веб-застосунки та API, які є основою більшості сучасних цифрових сервісів. Це стосується як комерційних платформ, так і державних рішень, що забезпечують доступ до критичних реєстрів та персональних даних громадян.

Актуальність проблематики підтверджується і воєнним досвідом України: кіберфронт став одним із визначальних елементів сучасної гібридної війни. Атаки на державні сервіси, критичну інфраструктуру, логістичні та фінансові системи стали не винятком, а буденністю. Як показують численні випадки, компрометація навіть одного API-ендпоїнта здатна призвести до масштабних витоків персональних даних, порушення доступності сервісів та суттєвих економічних збитків. У таких умовах питання захисту веб-ресурсів перестає бути суто технічним — воно перетворюється на елемент національної безпеки.

Складність архітектур сучасних застосунків зростає: використання мікросервісів, зовнішніх API, хмарних та мультихмарних середовищ створює нові вектори атак, які більше не можуть бути ефективно закриті традиційними WAF-рішеннями. Як зазначено в дипломному проєкті, класичний WAF орієнтований на сигнатурний аналіз, статичні правила та захист від заздалегідь відомих загроз. Проте сучасна загрозова модель включає API-зловживання, бот-атаки, атаки на бізнес-логіку та складні поведінкові сценарії, що вимагають принципово іншого підходу.

Тому логічним етапом еволюції засобів безпеки стає WAAP (Web Application and API Protection) — комплексне рішення, яке поєднує WAF, API Security, Bot Management, DDoS-захист і поведінкову аналітику. Особливу актуальність має

вивчення рішень WAAP у контексті розподілених хмарних платформ, таких як F5 Distributed Cloud (F5xC), які дозволяють забезпечувати безпеку на edge-рівні, у мультихмарних і локальних середовищах одночасно.

У ситуації, коли веб-додатки є основним інтерфейсом взаємодії користувачів із державними та комерційними сервісами, а персональні дані перетворюються на стратегічний ресурс, питання побудови ефективної моделі захисту веб-застосунків та API стає критично важливим. Тому дослідження можливостей та практичних аспектів застосування F5xC WAAP є актуальним і має як наукове, так і прикладне значення для забезпечення кіберстійкості організацій різних типів.

Об'єкт дослідження – процес виявлення загроз кібернетичної безпеки в веб додатках за допомогою WAAP.

Предмет дослідження – WAAP(web application and API protection) від світового вендора F5 Networks це компанія яка надає не тільки послуги по забезпеченню безпеки веб додатків і корпоративних мереж, а ще і відома своїми балансувальними. Загалом цього вендора можна часто зустріти у замовників з Fortune 500, в різних міксах рішень.

Мета роботи – оглянути рішення та його концепцію, розробити рекомендації щодо забезпечення інформаційної безпеки веб додатку на базі рішення F5xC WAAP.

Наукові завдання:

- провести аналіз сучасних проблем кібербезпеки та веб додатків та способів забезпечення безпечного досвіду користування;
- провести аналіз лінійки рішень F5 Networks;
- провести аналіз переліку можливостей AWAFF F5 Networks, порівняти їх з F5xC WAAP та рекомендацій щодо роботи з ним;
- розробити рекомендації для фахівців з кібербезпеки для захисту веб додатків.

Практичне значення одержаних результатів: полягає в аналізі проблем та розробці рекомендацій щодо забезпечення інформаційної безпеки веб додатків при використанні лінійки рішень від вендора F5 Networks.

1. ДОСЛІДЖЕННЯ ПРОБЛЕМИ ЗАХИСТУ ВЕБ ДОДАТКІВ

1.1 Актуальність проблеми забезпечення інформаційної безпеки

Дуже багато змін відбулося в сфері інформаційної безпеки, а громадяни України отримали багато досвіду в протистояннях на всіх можливих фронтах, включно з інформаційним.

Саме такий досвід дає нам повне розуміння важливості цього напрямку, в сучасному світі інженери інформаційної безпеки цінуються більше розробників, бо вже багато програмного забезпечення(далі ПЗ) написано і створено, тепер стоїть виклик в правильній побудові захисту навколо додатків і інфраструктури. В сучасному світі веб додатки це не просто landing page, це повноцінний комплекс надання послуг, починаючи від замовлення таксі, перегляду фільмів, бронювання квитків, надсилання/посилок, онлайн банкінг і ще багато чого іншого. Важливо підкреслити, що складність архітектури збільшується з доданням API(application programming language) в додатки. Навіщо писати свій календар для додатку, якщо є багато відкритих API ? Навіщо нам зберігати інформацію про всі авіа квитки у себе на платформі, якщо за допомогою API call ми можемо отримати актуальну інформацію на момент запиту клієнта з платформи авіа перевізника ? Питання звісно риторичні.

Тому ми робимо заключення що важливо мати комплексний захист не тільки мережевого стеку, а ще і веб додатків в комплексі з API.

Якщо коротко то кібербезпека — це стрімко зростаюча сфера, мета якої — захищати цифровий світ (комп'ютерні системи, мережі, мобільні пристрої) та їхні дані від кібератак. У сучасному світі технології стали невід'ємною частиною нашого життя, а обсяг даних і інформації, що зберігаються та передаються онлайн, зріс у рази. Зростаюча залежність від технологій робить необхідним захист наших систем і даних від кіберзагроз, таких як хакерські атаки, шкідливе ПЗ, фішинг, програми-вимагачі та інші форми кіберзлочинності.

Основна мета кібербезпеки — забезпечити, щоб конфіденційні та критично важливі дані залишалися приватними та захищеними, а додаток працював навіть не знаючи про потенційні загрози. Для цього потрібно впроваджувати як технічні, так і нетехнічні заходи, використовуючи різні практики та технології для гарантування безпеки цифрових активів. Заходи безпеки мають застосовуватися на різних рівнях — мережевому, на рівні кінцевих пристроїв, даних та застосунків(покриваючи весь стек Model OSI).

Нетехнічна сторона кібербезпеки зосереджується на впровадженні кіберосвіти, створенні нових політик безпеки, а також розробці планів реагування на інциденти та планів відновлення після катастроф. Для цього існує багато регламентів та стандартів, які вимагають в кожній компанії наслідування цим правилам.

Ці заходи є важливими для того, щоб кожен був обізнаний про найновіші кіберзагрози та мав необхідні знання й інструменти для правильного реагування на будь-яке порушення безпеки.

Через зростання значення безпеки даних і приватності кібербезпека стала центральним елементом нових регуляцій і законів для організацій та урядів у всьому світі. Управління кіберкомплаєнсом швидко стає пріоритетом для багатьох організацій у всіх галузях, що підкреслює важливість підтримання сильних практик кібербезпеки.

Якщо говорити про статистику, то вона виглядає наступним чином:

- Глобальні збитки від кіберзлочинності, за прогнозами, досягнуть 8 трильйонів доларів у 2023 році. По факту вийшло до 10 трильйонів
- Середня глобальна вартість витоку даних становить 4,35 мільйона доларів.

Figure 1.
Measured in USD millions

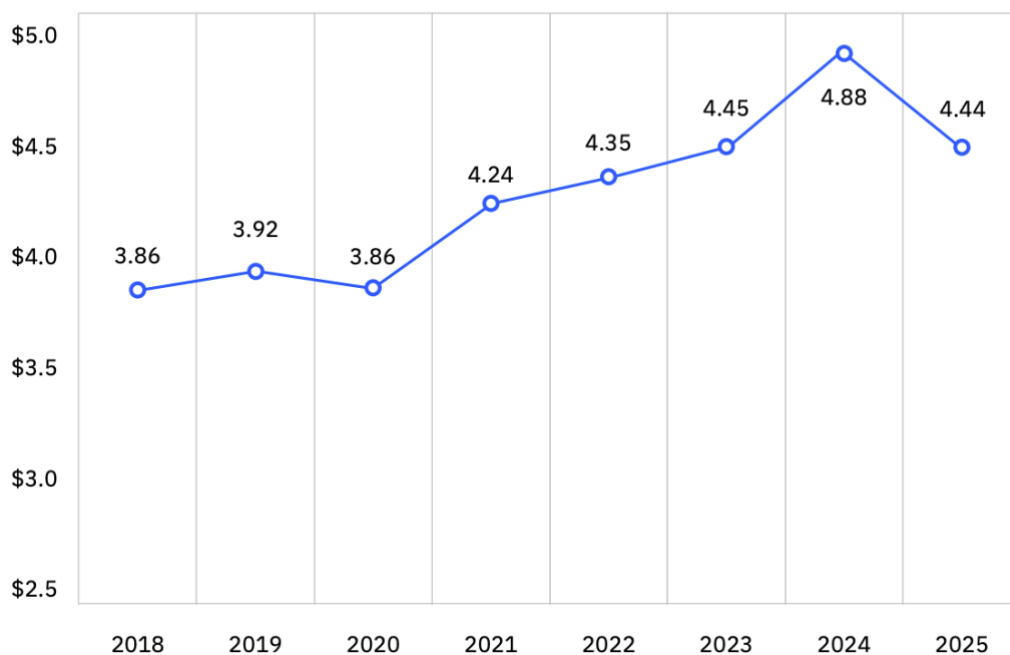
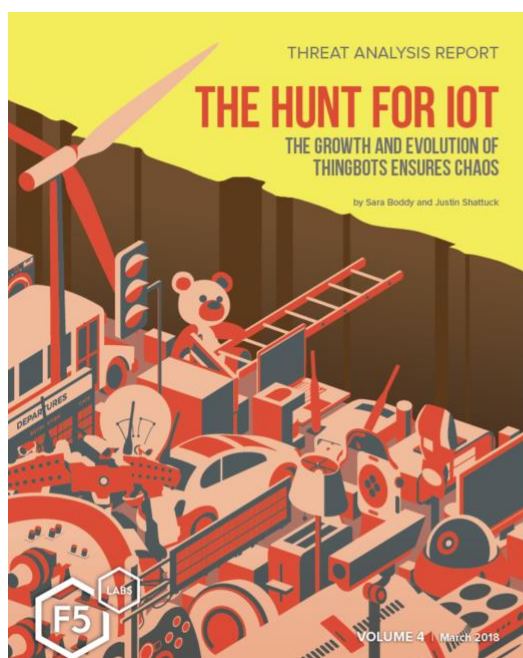


Рис 1.1. Статистика вартості витоку даних по рокам

- За оцінками, у 2023 році було викрадено 33 мільярди записів.
- Трафік згенерований ботами перевищує 80% всього інтернет трафіку(включно з трафіком від добрих ботів, закінчуючи шкідливими ботнетами)



thingbots

offer **300 Gbps** DDoS attacks for **\$20**

have evolved into **multi-purpose** attack bots for hire

can launch up to **12 Tbps** DDoS attacks

61% of top attacking IPs have been the same for **2+ years**

Рис1.2. Статистика по трафіку від ботів описана в звіті про аналіз загроз

Додатків стає більше, відповідно «Будь-яке фундаментальне технічне чи технологічне нововведення, надаючи можливості для вирішення одних соціальних проблем і відкриваючи широкі перспективи для розвитку особистості і суспільства, завжди викликає загострення старих або породжує нові, раніше невідомі проблеми, стає джерелом нових потенційних небезпек.

Без належної уваги до питань забезпечення безпеки наслідки переходу суспільства до нових технологій можуть бути катастрофічними для нього і його громадян. Саме так йде справа в області атомних, хімічних та інших екологічно небезпечних технологій, у сфері транспорту. Аналогічно йде справа і з інформатизацією суспільства.

Важливим є порівняння безпеки і її відсутності (в оригіналі Security vs Insecurity)

Таблиця 1.1

Порівняння мислення з безпекової точки зору та її протилежності

Аспект	Security	Insecurity
Мислення	Запобігання, виявлення, реагування	Латка, коли зламається / сподіваюся, що нічого не станеться
Профіль витрат	Початкові інвестиції + передбачувані періодичні витрати	Непередбачуваний «податок на стихійне лихо» у разі виникнення збою
Операційний вплив	Безперебійна робота, безвідмовна робота, гнучкість	Збої, перебої, ручне пожежогасіння
Ризик	Нижчий залишковий ризик за різними векторами загроз	Високий залишковий ризик; використовуються прогалини

Вигода для бізнесу	Забезпечує зростання, довіру, нові послуги	Підриває репутацію, призводить до відтоку кадрів, штрафів
Модель прибутковості	ROI: уникнення збитків, економія операційних витрат, швидший вихід на ринок	Негативна рентабельність інвестицій у більшості сценаріїв (ви платите більше, пізніше)

Відповідно як результат ми можемо зрозуміти, що безпека це про правильну надійність і проактивність, в свій ще час мислення через протилежний критерій, такий як небезпека це слабкість(в мисленні в тому числі) і реактивність. А як відомо реактивні рішення не мають взагалі нічого спільного з якісними і правильними підходами та рішеннями.

В нашій країні вся держава і основні сервіси доступні в смартфоні, звичайно мова зараз йде саме про додаток ДіЯ. Це чудовий приклад того, чим ми можемо користуватися кожен день навіть не усвідомлюючи, що там взагалі відбувається зі сторони сервісів і серверів. Насправді з цим додатком все набагато простіше, сама по собі аплікація це просто метод відображення для кінцевого користувача великої кількості різних баз даних і реєстрів. Тобто чисто в теорії коли ми розуміємо, що насправді сам по собі додаток не має ніяких баз даних, то читати новини про черговий взлом і витоки з цієї аплікації стає ще веселіше, адже нам вже відомо що в самій аплікації ніякої інформації немає. Так вона є на серверах інших організаційних структур і так до неї все ще можна отримати доступ. Але чи означає це, що такий взлом неможливий ? І що саме взагалі ми можемо вважати взломом в цій ситуації ? Давай розберемося.

ДіЯ спілкується з іншими сервісами за допомогою API, тобто в теорії можна скомпрометувати API call, зробити туди ін'єкцію і отримати доступ до великої кількості інформації, але чи все так просто насправді ?

Відповідь на це запитання, якраз і є однією з основних тем піднятих в цьому дипломному проекті. Для інженерів з інформаційної безпеки важливо розуміти, що

саме вони захищають і як саме працює сервіс, в цьому випадку доволі доречним є Reverse Engineering і звичайно знання з боку Offensive(атакуючого). Коли ти знаєш як працює додаток і до чого він вразливий стає простіше будувати захист.

Важливо підкреслити, що сучасний бізнес, який в тому чи іншому форматі присутній в цифровому середовищі чудово розуміє всі виклики, але не всі готові інвестувати в безпеку. Це помилка молодих або ще не зрілих бізнесів(small/medium business).

Наприклад бізнес надає послуги з перевезення посилок по Україні, їм для надання сервісу потрібно отримати від вас певний набір РП(персональні дані), що може включати в себе, але не обмежуватись – Прізвище, Адреса, Номер Телефону, в деяких випадках навіть платіжна інформація.

Цей бізнес в основній своїй суті є логістичним, а цифровізація це вимушений крок і просто комфорт для юзерів, бізнес своєчасно не звертає увагу на безпеку, тому що для них не очевидно, як це може збільшити прибутки компанії, будь-який бізнес це в основному про прибуток, адже якщо немає прибутку то це вже благодійна організація. Тепер подивимось на все це через призму побудови правильної моделі захисту РП, адже насправді всі ваші дані коштують грошей, якісь більше, якісь менше, але це гроші, які зловмисники готові платити для того, щоб дізнатись ваш номер телефону, який потім використають для підтвердження 2FA і доступу на особисту пошту чи в хмарного сховища. Або для отримання тої самої комбінації логіну і паролю, яка дає зловмисникам великий спектр для потенційних атак.

Уявімо що бізнес не захистив вас і вашу інформацію, а ви потенційно зазнали збитків через це, починаючи від спаму в месенджерах, бо ваш номер потрапив в безкоштовні бази активних юзерів і до втрати доступу до банкінгу або навіть втрати хеш токенів для доступу до криптогаманців.

Для бізнесу це репутаційна втрата, відтоки клієнтів і в деяких випадках навіть судові позови, що для компанії може коштувати значну частину прибутку.

Тут в нас піднімається важливе питання про яке починає думати і вивчати CISO, що таке ROI ?

ROI у кібербезпеці — це показник, який показує, як інвестиції у захист окуповуються зменшенням ризиків, уникненням збитків і підвищенням ефективності роботи. Його важливо розглядати не як «прибуток», а як запобігання втратам.

ROI формується через три основні компоненти.

1. Перший — скорочення фінансових ризиків. Кібератаки сьогодні коштують бізнесу дорожче, ніж будь-коли: середня вартість витоку даних сягає мільйонів доларів, а глобальні збитки від кіберзлочинності зростають щороку. Інвестиція у захист зменшує імовірність інциденту й дає компанії можливість уникнути витрат на відновлення, штрафів, компенсацій та простою сервісів.

2. Другий — підвищення операційної ефективності. Сучасні засоби кібербезпеки автоматизують виявлення загроз, прискорюють реагування та зменшують навантаження на команди. Це вивільняє ресурси, скорочує час простою систем та стабілізує роботу бізнес-процесів, що прямо впливає на фінансовий результат.

3. Третій — підтримка комплаєнсу. Регулятори по всьому світу посилюють вимоги до захисту даних. Інвестиції в безпеку дозволяють уникнути штрафів, довести відповідність нормам і зберегти можливість працювати на регульованих ринках.

Тут ми розуміємо, що сучасна кібербезпека, це не просто багато грошей інвестованих в якісь рішення, а це ще правильне керування бюджетами, розуміння і запобігання ризикам, які можуть виникати через недостатній захист і ще багато чого іншого. Сучасна кібербезпека це доволі складно організований по своєму живий організм, який постійно вимагає уваги для вивчення тенденцій. Отже що дорожче — сам витік чи інструменти, які могли б йому запобігти?

Кібербезпека сьогодні перетворилась на фундаментальний елемент нашого існування — від особистого цифрового простору до інфраструктур цілих держав і міжконтинентальних систем. Вона більше не є вузькотехнічним напрямом: це багаторівнева екосистема, що визначає безпеку суспільства, стійкість економіки та навіть геополітичну рівновагу.

У сучасному світі лінія фронту проходить не лише по окопах, а й через серверні кімнати, дата-центри, хмарні середовища й персональні смартфони. Війни стають гібридними, і їхня логіка зміщується від фізичного протистояння до боротьби за інформацію. Замість того, щоб руйнувати інфраструктуру артилерією, противник може зупинити роботу цілої енергосистеми суто через доступ до мережевого обладнання і пошкодивши ядро мережі. Замість вторгнення танкових колон — викрадення масиву даних, компрометація державних систем або дезінформаційні операції, що руйнують довіру в суспільстві.

Цифровий простір став новим театром бойових дій, де активами є дані, а зброєю — шкідливий код, фішингові кампанії, вірусні ланцюги(kill chain), бот-мережі та атаки на критичну інфраструктуру. Індивід, бізнес, уряд — кожен рівень уразливий, і кожен є частиною загального захисного контуру. Компрометація однієї ланки може спричинити наслідки на рівні організацій, секторів або цілих регіонів.

Саме тому кібербезпека набуває значення стратегічної оборони. Держави інвестують у кіберкомандування, підприємства впроваджують багаторівневі системи захисту й комплаєнсу, а громадяни навчаються цифрової гігієни так само, як колись — правилам безпеки на вулиці. Відмова від цих інвестицій означає не лише ризик фінансових втрат, а й потенційну втрату суверенітету, контролю над інформаційними потоками й здатності захищати власні інтереси.

Таким чином, кібербезпека стала не просто технологічним аспектом, а умовою виживання в епоху, де інформація стала стратегічним ресурсом, а цифрові загрози — повноцінним елементом сучасних конфліктів.

Я дуже радий бути в цей момент в країні, яка має найбільше можливостей отримати правильний досвід і використати його в подальшому для покращення світового рівня безпеки. І більше за все я радію, що маю можливість в такий момент допомагати нашим бізнесам і державному сектору ставати краще і безпечніше для кожного з нас.

1.2 Порівняння класичної архітектури на базі WAF та сучасної на базі WAAP

Можемо вважати класичною або традиційною архітектурою саме на базі WAF.

Основна задача і причина виникнення - WAF виник як окремий захисний шар(L7 OSI Model), що стоїть перед веб-додатком і фільтрує HTTP-трафік. Основне завдання якого — знайти і заблокувати класичні web-атаки, які описані у OWASP Top 10 (SQL-ін'єкції, XSS, CSRF, командні ін'єкції тощо) і ще багато чого іншого, сучасний WAF, це комплексний підхід, який зазвичай використовує архітектуру Reverse Proxy(але зустрічаються і випадки Full Proxy(Reverse + Forward)).

Основні цілі WAF – це три головні пункти:

1. Захистити веб-додатки від відомих векторів атак.
2. Зменшення ризику компрометації даних, особливо у додатках, які працюють з клієнтською інформацією.
3. Закриття вразливості, які бізнес не встигає виправити у коді.

Сюди можна ще багато чого додати, бо це доволі базові вимоги, які в сучасному світі може покрити будь який “green button/one button WAF”.(*Green-button WAF — це традиційний сигнатурний WAF, який працює за принципом «увімкни й забудь», забезпечуючи мінімальний рівень безпеки без глибокої поведінкової аналітики, API-контексту та сучасних засобів виявлення загроз.*)

Основні підходи які WAF використовує:

1. Сигнатурний підхід. Порівняння трафіку зі списком відомих патернів атак описаних за допомогою CVE.
2. Правила та політики. Набори статичних правил, які адміністратор налаштовує відповідно до додатку.
3. Позитивна модель безпеки. Дозволяється лише те, що точно відповідає очікуваній моделі трафіку.
4. Негативна модель безпеки. Дозволяється все, а забороняється тільки явно шкідливий трафік.

Типові особливості WAF:

- Працює на рівні HTTP(L7 OSI Model).

- Захищає переважно монолітні веб-додатки.
- Зазвичай розгортається локально, на периметрі(може бути як віртуальному виконанні так і у залізному)
- Реагує на вже відомі загрози. Або в рішеннях рівня enterprise може мати ML для вивчення реального трафіку користувачів для побудови політики безпеки.
- Потребує ручного налаштування, регулярного тюнінгу. Або в рішеннях рівня enterprise це можна делегувати на ML для автоматичної побудови і тюнінгу політики, але звісно з правильним контролем адміністратора і стейджингом сигнатур. (*Signature staging — це процес тестування нових або оновлених сигнатур WAF у "неблокуючому" режимі, щоб зібрати статистику, оцінити їхню точність і адаптувати політики перед тим, як увімкнути їх у режим блокування.*)

Проблематика традиційного WAF

- Великий обсяг false positives.
- Через статичні правила і сигнатури WAF часто блокує легітимний трафік.
- Обмежена видимість.
- WAF бачить лише те, що заходить на периметр. Якщо трафік уже розподілений у хмарі або мікросервісах — він неефективний.
- Складність підтримки.
- Кожна зміна додатку вимагає оновлення правил. Що ускладнює процес розгортання нових версій і вимагає постійної взаємодії між відділами розробки і безпеки, що як ми знаємо в сучасному світі не завжди закінчується добре.
- Необхідність експертного тюнінгу.
- Якщо WAF погано налаштований — він або «німо стоїть» пропускаючи все, це ще називається неправильний policy tightening, або «стріляє у все підряд», що може бути неправильно вибраною моделлю безпеки. (*Policy tightening — це поетапне звуження, уточнення і посилення політик безпеки на основі аналізу реального трафіку, з метою переведення WAF/WAAP із «м'якого» захисту в*

«повний блок» без ризику для легітимних користувачів. Як правило це наступний крок після Signature Staging.)

- Нездатність захищати API. WAF проєктувався у час, коли API не були ключовою частиною логіки. Сьогодні ж більшість атак спрямовані саме на API, як і логіка побудови сучасних аплікацій. Сучасні проблеми вимагають сучасних інструментів для вирішення проблем.

З іншого боку у нас нова модель: WAAP — Web Application and API Protection, що є логічним розвитком класичного WAF, який прибирає основні недоліки застарілої моделі.

Якщо коротко то WAAP — це еволюція WAF, яка враховує реалії сучасних хмарних систем, мікросервісів, API-економіки та динаміки DevOps-процесів.

WAAP не замінює WAF — він включає в себе WAF, але розширює функціонал на інші площини, чим відповідно і стає більш широким і зручним продуктом для захисту в сучасних архітектурах.

Основні цілі WAAP:

- Захищати веб-додатки й API у будь-якому середовищі (он-прем, хмара, мульти-хмара, edge).
- Блокувати не лише класичні веб-атаки, а й API-зловживання, логічні атаки, бот-трафік.
- Надавати автоматизовану аналітику з мінімальним тюнінгом.
- Працювати у DevOps-циклі і підтримувати швидкість релізів.
- Забезпечувати масштабування і видимість у розподілених системах.

WAAP працює не лише на рівні HTTP-правил, хоча все ще основним рівнем роботи залишається L7. Він використовує багатосарові підходи:

1. Машинне навчання і поведінкова аналітика

Система вивчає нормальний трафік і виявляє відхилення у реальному часі що дозволяє:

- ловити 0-day атаки;
- зменшити залежність від сигнатур;

- автоматизувати виявлення ботів;
- адаптувати захист до змін у додатку.

2. Вбудований API Security

WAAP аналізує запити на рівні:

- методів,
- схем,
- параметрів,
- токенів,
- аномалій використання,
- API-зловживань (BOLA, excessive data exposure, mass assignment).

Рішення enterprise рівня спроможні навіть вивчати API endpoints, взаємодію і методи, для формування правильної політики безпеки.

3. Захист від ботів

WAAP включає рішення проти бот-мереж:

- JavaScript-челенджі,
- ML-класифікацію поведінки,
- захист від credential stuffing,
- захист від скрейпінгу.

4. DDoS-захист у хмарі

Включає автоматичне масштабування та глобальне фільтрування. Більша частина сучасних рішень є саме SaaS, що надає нам можливість використовувати ДЦ вендорів для захисту від різних типів volumetric DDoS, SlowLoris та інше.

Основна проблематика, яку вирішує WAAP

- API стають головною ціллю атак, тому що це 70% архітектури сучасного додатку.
- Класичний WAF не розуміє логіки API або вимагає розширення і доповнення іншими модулями
- Мікросервіси розносять трафік між різними зонами і сегментами інфраструктури.

- Периметру більше немає, як однієї точки входу— і WAF на периметрі не бачить внутрішні взаємодії.
 - Хмарні і мульти-хмарні архітектури збільшують масштаб і складність роботи додатку, сучасні тренди на HA/DR(high availability/ disaster recovery) вимагають саме таких архітектур для гарантування SLA 99.9%.
 - Потрібне рішення, яке легко тиражується, а не вимагає окремої “коробки” на кожен сегмент.
 - Додатки оновлюються щодня. Класичні правила безпеки, які створені ручному режимі просто не встигають.
 - Нові вектори атак — бізнес-логіка, обходи аутентифікації, API-misuse.
- Це те, що WAF не бачить, і це те для чого потрібен WAAP.

Реалізація WAAP

1. Хмарна архітектура (SaaS або distributed cloud)

WAAP працює у глобальній хмарі з реплікацією і автоматичним оновленням.

Приклади: F5 Distributed Cloud WAAP, Cloudflare WAAP, Akamai App & API Protector.

2. Повна видимість через один портал

Адміністратор бачить:

- веб-трафік,
- API-виклики,
- стан ботів,
- DDoS-події,
- поведінкові аномалії.

3. Zero Trust підхід, який вимагає щоб усі запити проходили перевірку:

- автентифікація,
- авторизація,
- контекст запиту,
- поведінкові фактори.

4. Політики на рівні мікросервісів і API-ендпоінтів

Захист стає більш гранульованим — для кожного endpoint свій профіль.

5. Автоматичне оновлення та ML-оновлення

- Немає потреби вручну тюнити 1000 правил.
- WAAP вчиться на реальному трафіку.

Чому від WAF переходять до WAAP (як основна тенденція ринку)

1. Периметр зник. В сучасному світі дуже складно знайти одну точку входу, сучасні архітектури це хмари, VPN-less доступ, edge-локації, SaaS.

2. API займають 80–90% трафіку сучасних додатків. WAF створений для HTML-сторінок, а не для JSON-обміну(в основному).

3. Кількість атак на API зростає у 5–7 разів швидше, ніж на класичні web-додатки.

4. Бізнес хоче швидкості релізів. WAF сповільнює DevOps, WAAP інтегрується з DevOps.

5. Немає сенсу купувати коробки у світі хмари. WAAP працює у SaaS, автоматизується і не потребує підтримки заліза.

Отже підсумовуючи класичний WAF — це продукт епохи монолітних веб-сайтів і статичного периметру. Сучасна реальність — це API-економіка, мікросервіси, хмара, edge і гібридні атаки.

WAAP — це логічна еволюція, яка збирає в одному рішенні:

- WAF,
- API Security,
- Bot Management,
- DDoS-захист,
- ML-аналітику,
- інтеграцію з DevOps.

Це не просто «кращий WAF». Це нова модель безпеки, орієнтована на динамічні середовища, де дані рухаються між сервісами, регіонами і хмарами, а атаки стають розумнішими і складнішими.

Якщо взяти до уваги все сказане вище, то WAAP має багато переваг, але все рівно не є конкурентом для WAF, це рішення яке виросло з WAF. Основні з переваг WAAP підсумовані нижче:

1. WAAP забезпечує уніфіковане рішення для безпеки, захищаючи від різноманітних загроз, таких як SQL-ін'єкції, XSS, DDoS-атаки та специфічні вразливості API. Такий цілісний підхід гарантує, що вебзастосунки та API залишаються захищеними, надійними й продуктивними.

2. У відповідь на еволюцію кіберзагроз WAAP використовує машинне навчання та штучний інтелект, щоб виявляти та реагувати на нові атаки в реальному часі, захищаючи від складних та цілеспрямованих загроз. API є критичними і вразливими. WAAP забезпечує їх безпеку, контролюючи та блокуючи шкідливі запити, що покращує продуктивність і надійність API.

3. Хмарні механізми безпеки(в тому числі і за рахунок контейнеризації) WAAP масштабуються разом із вашими застосунками, забезпечуючи стабільний рівень захисту в локальних, хмарних і гібридних середовищах.

4. WAAP інтегрує CDN WAF та edge WAF для зменшення затримок і підвищення продуктивності, забезпечуючи безпеку без шкоди для користувацького досвіду. WAAP допомагає відповідати регуляторним і сучасним вимогам (GDPR, HIPAA, PCI DSS), надаючи деталізоване логування, моніторинг та звітність. Також в скоро часі ми очікуємо випуск нового регламенту EU AI Act для регулювання моделей ШІ, і відповідно сучасне рішення так само повинно буде готове до нових викликів.

5. Зручні інтерфейси та автоматизація в рішеннях WAAP спрощують управління засобами безпеки, зменшуючи потребу у значних ресурсах.

Порівняння можливостей WAF та WAAP

Можливості	WAAP	AWAF
Основна мета	Уніфікований захист для веб-додатків та API, включаючи виявлення, захист стану та виконання;	Захист веб-додатків від HTTP-експлойтів та відомих атак OWASP-top
Охоплення	Веб-, мобільні та машинні API; багатомарні та периферійні сліди;	Переважає трафік браузер-додаток на певних сайтах
Робота з API	Перевірка з урахуванням схеми, розбір JSON/XML, перевірка токенів, політики швидкості, виявлення шаблонів зловживань;	Обмежена обізнаність щодо API; розглядає виклики API як загальний HTTP, якщо не додано власні правила
Робота з даними	Перевіряє структуру корисного навантаження, типи та поля відповідно до специфікацій API (OpenAPI);	Правила, орієнтовані на регулярні вирази та підписи; перевірка схеми є рідкістю
Запобігання ботам та зловживанням	Вбудована оцінка ботів, поведінкові сигнали, елементи керування заповненням та вилученням облікових даних. Часто додатковий модуль;	Базові сигнатури ботів та репутація IP-адрес
L7 DDoS & Rate Limiting	Квоти на кінцеву точку, обмеження, що враховують орендарів, адаптивне дроселювання на периферії та в джерелі;	Грубі обмеження IP-адрес або шляхів; менше контексту щодо ідентифікацій.

Discovery & inventory	Автоматичне виявлення тіньових та зомбі-API, генерування інвентаризацій та ранжування ризику	Зазвичай не входить до базового комплекту, вимагає додаткових ліцензій
Identity & access context	Інтегрується з IdP, JWT, mTLS; забезпечує вбудований контекст автентифікації	Обмежений контекст ідентифікації без налаштування інтеграції
Автоматизації та машинне навчання	Виявлення аномалій, сповіщення про дрейф зі специфікацій API, автоматичні пропозиції політик	Ручне налаштування правил; вищий ризик хибних спрацьовувань
Розгортання	Хмарно-орієнтований, edge WAF з інтеграцією CDN, sidecar/ingress для Kubernetes, multicloud	Залізний інстанс або віртуальна машина
Телеметрія	Метрики методу для кожного API, маяки SDK клієнта, QoS та бізнес-KPI	Події безпеки та журнали запитів; менше метрик, специфічних для API
Інтеграція з API gateway	Розроблено для роботи разом з шлюзами; шаблони «WAF шлюзу API» підтримуються спільними політиками та телеметрією.	Може розташовуватися до або після шлюзу; залежить від ручної координації.

1.3 Основні виклики, які вирішує WAAP

Раніше ми говорили про прогалини в захисті а тепер розберемо, як WAAP усуває ці всі недоліки, забезпечуючи кращий захист і стабільність для компанії та її адміністраторів.

1. Складні та загрози, що постійно еволюціонують

Виклик: Загрози постійно змінюються, і кіберзлочинці розробляють нові тактики та техніки для експлуатації вразливостей у вебзастосунках і API. Традиційні засоби безпеки часто не встигають адаптуватися до таких швидких змін.

Вирішення за допомогою WAAP: Сервіси WAAP використовують машинне навчання та штучний інтелект для виявлення і миттєвого реагування на нові загрози. Такий проактивний підхід дає змогу визначати й блокувати навіть найскладніші атаки до того, як вони завдадуть шкоди.

2. Захист API від зловмисних атак

Виклик: API все частіше стають ціллю атак, адже саме вони часто відкривають доступ до критичних даних і функціональності. Забезпечити їхню безпеку складніше, ніж у традиційних вебзастосунків, через специфіку обміну даними.

Вирішення за допомогою WAAP: WAAP забезпечує спеціалізований захист API, контролюючи та фільтруючи трафік. Система визначає та блокує шкідливі запити, запобігає витокам даних і гарантує безпечну роботу API. Це значно покращує продуктивність і стабільність API.

3. Керування розподіленими та динамічними середовищами

Виклик: Сучасні вебзастосунки розгортаються у локальних, хмарних та гібридних середовищах. Це ускладнює керування всіма компонентами та їхній захист.

Вирішення за допомогою WAAP: WAAP використовує хмарні засоби безпеки, пропонуючи єдину платформу для захисту вебзастосунків і API у будь-якому середовищі. Централізований підхід спрощує керування та гарантує єдину політику безпеки незалежно від розташування застосунку.

4. Робота з високими обсягами трафіку та захист від DDoS-атак

Виклик: Високі обсяги трафіку можуть перевантажити вебзастосунки та API, спричиняючи проблеми з продуктивністю або простої. DDoS-атаки також можуть повністю паралізувати роботу серверів.

Вирішення за допомогою WAAP: Використання CDN WAF та edge WAF допомагає WAAP оптимізувати роботу та стійкість застосунків. Трафік розподіляється ефективніше, а шкідливі запити відсіюються ще до потрапляння на сервер. Завдяки цьому вебзастосунки та API залишаються доступними навіть під час пікових навантажень чи DDoS-атак.

5. Дотримання вимог до безпеки та конфіденційності

Виклик: Організації повинні відповідати вимогам GDPR, HIPAA, PCI DSS та інших стандартів. Недотримання може привести до штрафів і втрати довіри клієнтів.

Вирішення за допомогою WAAP: WAAP пропонує потужні засоби комплаєнсу — деталізоване логування, моніторинг і звітність. Це дозволяє забезпечити відповідність регуляторним вимогам та прозорість усіх процесів.

6. Зменшення затримок і покращення користувацького досвіду

Виклик: Засоби безпеки можуть додавати затримку у відповіді системи, що негативно впливає на користувацький досвід.

Вирішення за допомогою WAAP: Сучасні WAAP-рішення мінімізують затримки завдяки розміщенню механізмів безпеки на мережевому периметрі, ближче до користувача. Це покращує швидкість відповіді та загальну взаємодію, не знижуючи рівень безпеки.

7. Спрощення керування безпекою

Виклик: Захист вебзастосунків і API може бути складним та вимагати багато ресурсів. Неправильні налаштування часто створюють додаткові ризики.

Вирішення за допомогою WAAP: WAAP пропонує просте керування через зручні інтерфейси та автоматизацію. Централізовані панелі керування дають змогу легко контролювати стан безпеки й суттєво зменшують навантаження на команди.

2. АНАЛІЗ МЕТОДІВ ТА ЗАСОБІВ ЗАХИСТУ ВЕБ ДОДАТКІВ НА БАЗІ F5 NETWORKS

2.1 Огляд технології Distributed Cloud та рішення F5xC

F5 Networks — американська компанія, заснована у 1996 році, яка починала з балансування трафіку та розробила BIG-IP, яка працює на TMOS і з часом стала одним із глобальних лідерів у сфері безпеки застосунків та керування мережевими сервісами. Компанія зосереджена на тому, щоб робити роботу додатків стабільною, швидкою та захищеною у будь-якому середовищі — у дата-центрі, у хмарі та на edge.

Сильна сторона F5 — глибоке розуміння та детальна робота з L7-трафіком. Їхні рішення вміють читати структуру запиту, аналізувати поведінку користувача та будувати точний профіль загроз. Саме тому WAF, WAAP і API-захист F5 вважають одними з найефективніших у світі. BIG-IP багато років був і залишається стандартом enterprise-класу для фінансових установ, телекомів і великих корпорацій, а його гнучкість і можливість повного кастомізованого контролю L7-трафіку не має аналогів. Саме тому лідери Fortune 500 використовують рішення цього вендора.

Сучасний F5 трансформувався у платформу Distributed Cloud, яка дає глобальний edge-захист, multi-cloud networking, Zero Trust і автоматизовану безпеку API. Це дозволяє компаніям отримувати не просто WAF, а єдиний контур безпеки, доставки та видимості додатків по всьому світу. Завдяки поєднанню багаторічного досвіду, інновацій і придбаних технологій (NGINX, Shape Security, Volterra) F5 пропонує одну з найсильніших екосистем безпеки для сучасних, розподілених і високонавантажених застосунків.

Отже F5xC також відомий як F5 Distributed Cloud є рішенням типу SaaS.

Важливо також розібратися з тим, що таке Distributed cloud історично у нас в світі всі додатки були монолітними, але як ми вже проговорили сучасний світ дещо

відрізняється і потребує адаптації. Спочатку під монолітний додаток у нас був один ДЦ, пізніше ми почали будувати декілька ДЦ, щоб мати відмовостійкість.

Після цього нам стала доступна хмара, неважливо яка публічна чи приватна. Ми почали використовувати її так само додаючи як альтернативу до звичайних ДЦ. Такий тип архітектури вже називали гібридним.

Після цього, коли оренда хмарних систем виявилась зручнішою в керуванні і в деяких випадках навіть вигіднішою для масштабування ми почали все більше використовувати хмарних провайдерів і менше віддавати переваг наземним ДЦ(виключаємо звісно приклади з державними структурами і регуляціями, які під час повномасштабного вторгнення доречі трохи оптимізувались). Тут у нас почала ставати актуальною інша проблематика, в різних публічних хмарах різні системи білінгу і зручніше розбивати аплікацію на різні компоненти і тримати все в різних хмарах. Умовно основна БД у нас в Azure, фронт і бек аплікації на GCP. Цю архітектуру прийнято називати – мультиклаудна(multicloud) архітектура.

Найсучаснішою є distributed cloud архітектура це модель побудови хмарних платформ, у якій обчислення, безпека, мережа та сервіси доставки розподілені між багатьма географічними локаціями, а не зосереджені в одному центральному дата-центрі чи одному регіоні хмари.

У такій архітектурі є єдиний централізований control plane, але data plane рознесений по сотнях або тисячах вузлів: хмарні регіони, edge-локації, локальні дата-центри, мікро-ЦОДи, IoT-сайти, телеком-майданчики. Це дозволяє обробляти дані й забезпечувати безпеку там, де виникає трафік, з мінімальною затримкою та максимальною стійкістю.

По факту це логічна еволюція класичної хмари.

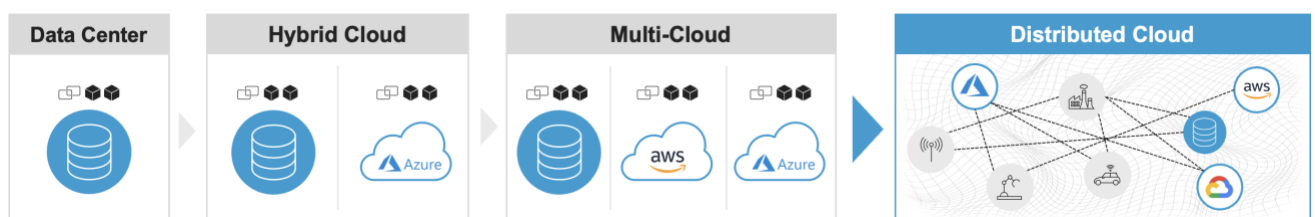


Рис 2.1. Схематичне зображення еволюції архітектури для додатків

F5 Distributed Cloud або F5xC — це глобальна розподілена платформа безпеки, мережі та delivery, побудована на edge-інфраструктурі F5 та здатна працювати у будь-якому середовищі: хмара, мульти-хмара, on-prem, IoT-локації, дата-центри, операторські майданчики за допомогою CE ноди(про це детальніше далі).

Вона створена як єдина control plane, що керує великою кількістю сервісів: WAAP, DDoS, API Security, Bot Defense, Global Load Balancing, Multi-Cloud Networking, Zero Trust, Service Mesh, Endpoint-захистом додатків, data-path схеми та інше.

1. F5xC — це не просто WAAP. Це універсальна платформа керування додатками, безпекою та мережею, яка працює як глобальний розподілений edge-cloud. Її архітектура базується на трьох ключових принципах:

1.1. Control-plane централізований, data-plane розподілений. Управління — у SaaS-хмарі F5XC. Трафік — на edge-вузлах ближче до користувача.

1.2. Уніфікація безпеки й мережі. WAAP, API Security, Bot Defence, DDoS, Mesh, LB, Zero Trust — все у єдиній політиці.

1.3. Будь-яке середовище має один контур керування. AWS, Azure, GCP, bare metal, VMware, Kubernetes, IoT — усе під'єднується як окремі ноди.

2. Архітектурні рівні F5XC. Архітектура платформи складається з трьох шарів:

2.1. Global SaaS Control Plane. Це керуючий рівень F5XC і завжди це SaaS і не знаходиться на інфраструктурі клієнта. Він містить:

- API-орієнтоване управління (REST / GraphQL).
- Central Policy Engine для WAF, API Security, Zero Trust, routing.
- Global Identity & Authentication (OIDC, SAML, mTLS, Cert manager).
- Configuration as Code (GitOps / Terraform).
- Central Observability — log-pipeline, metrics, traces, L7-аналітика.
- ML/AI Engines:
 - поведінкове виявлення атак,

- класифікація ботів,
- anomaly detection у API,
- auto-tuning політик.

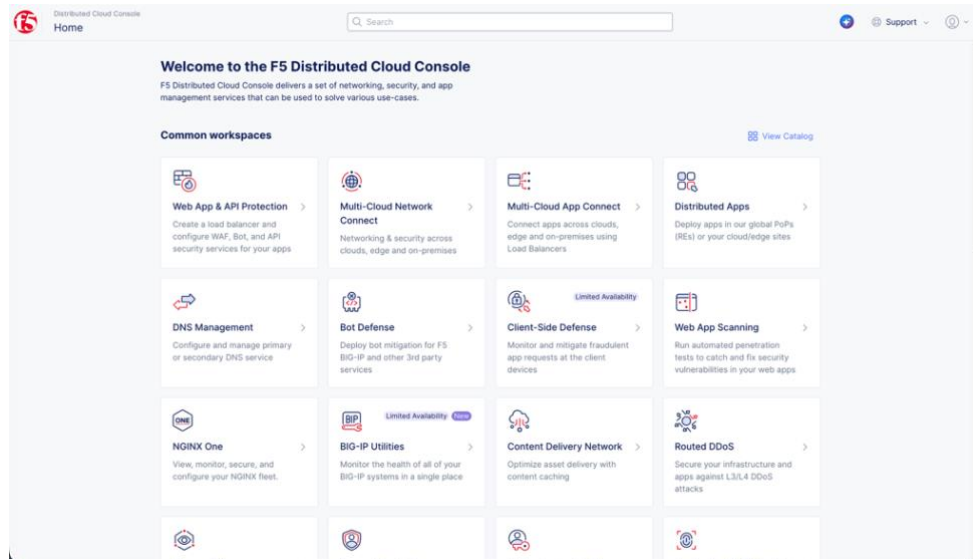


Рис 2.2. Центральна консоль F5xC

2.2. Distributed Data Plane (Global Network & Regional PoPs) що є глобальною edge-мережею F5xC:

- 35+ PoP (Points of Presence) по всьому світу.
- Ultra-low-latency backbones.
- Прямі інтерконекти з Azure / AWS / GCP.
- Повноцінний Distributed Cloud CDN.
- L3/L4/L7 фільтрація трафіку.
- Глобальний Anycast.

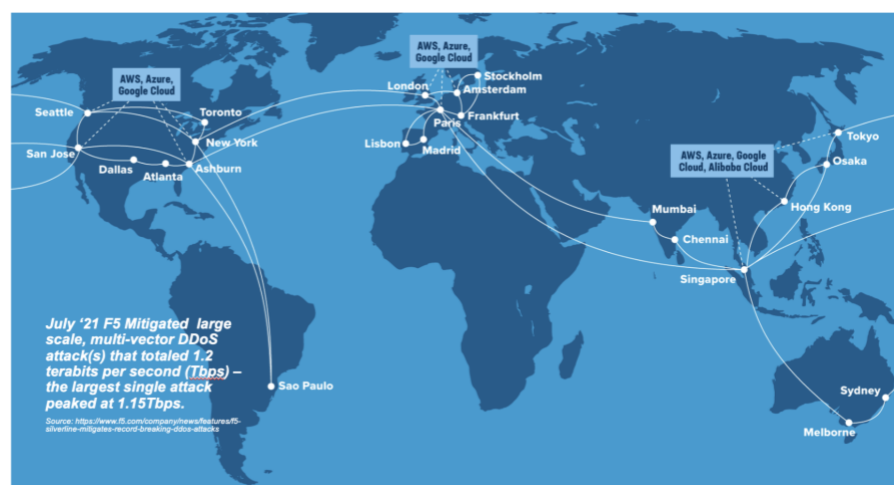


Рис 2.3. Розміщення точок присутності вендора

Точки присутності підтримують та виконують наступне:

1. WAF/WAAP на edge.
2. DDoS L3–L7 очищення.
3. TLS termination / re-encryption.
4. API discovery та захист.
5. Global Load Balancing (GSLB).
6. Bot Defense та rate-limiting.

Це робить F5XC ближчим по логіці до Cloudflare / Akamai, але з глибшою інтеграцією з приватними середовищами замовника.

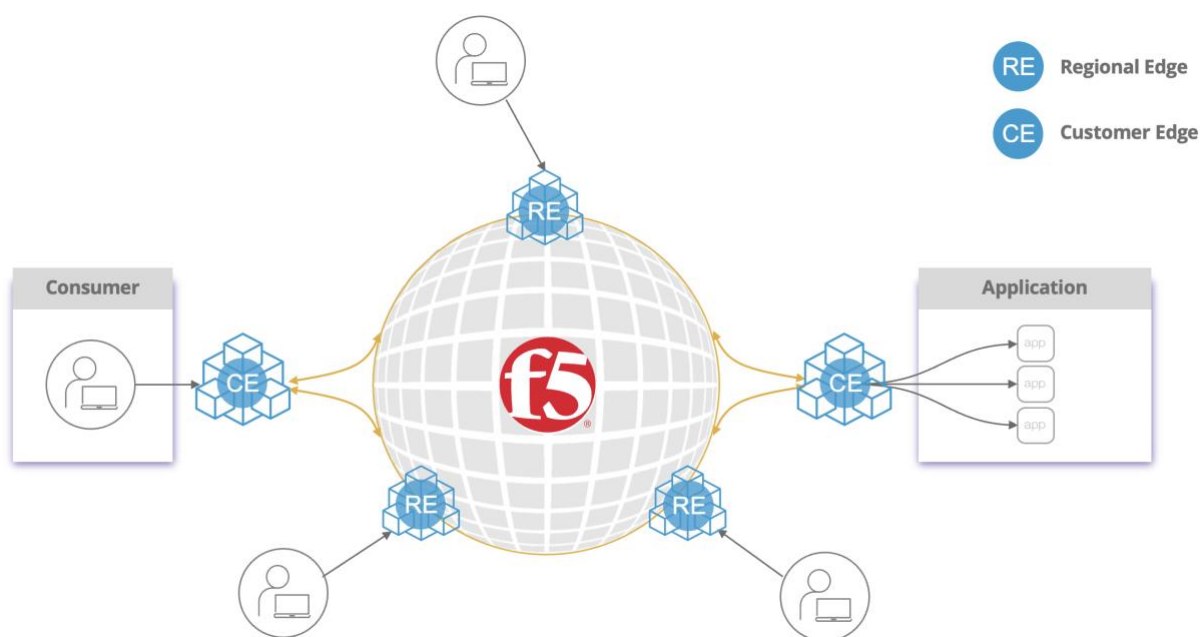


Рис 2.4. Схематичний приклад гібридного використання точок присутності вендора і CE нод розташованих безпосередньо з боку сервісу замовника.

2.3. Customer-Hosted Nodes (Customer Edge - CE). CE-ноди встановлюються у середовищі клієнта:

- Kubernetes (k8s кластер або окремі ноди).
- Bare metal, VMware ESXi.
- AWS/Azure/GCP (віртуальні edge-вузли).
- IoT/remote локації (1–2 вузли).

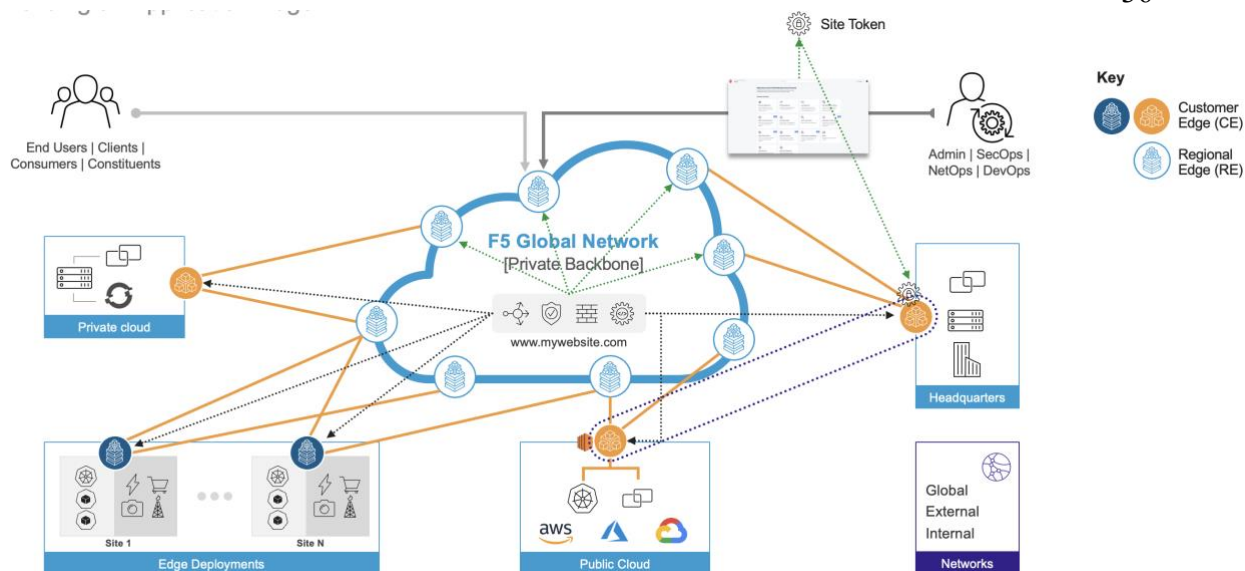


Рис 2.5. Приклад класичної рознесеної архітектури додатку, яка з'єднується і керується через центральну консоль F5xС

СЕ підтримують наступний функціонал:

1. локальний WAAP / WAF;
2. Mesh Gateway;
3. приватний ingress / egress;
4. локальний LB (L4/L7);
5. Zero Trust проксі;
6. мікросервісний routing.

СЕ інтегрується з control-plane через зашифрований тунель, який будується автоматично від час розгортання системи. По факту під капотом звичайний IP-Sec тунель, але який не потрібно ніяк вручну адмініструвати.

Трохи детальніше про технологію СЕ, щоб сформуванати комплексне розуміння, що таке взагалі СЕ.

Customer Edge у F5 Distributed Cloud — це фактично «точка присутності F5 у твоєму середовищі», невеликий, але дуже функціональний вузол, який ти розгортаєш у своїй хмарі, дата-центрі або навіть на віддаленій локації. Він виглядає як проста віртуальна машина, але виконує одну з ключових ролей в архітектурі F5XC: переносить частину data-plane з глобальних PoP безпосередньо ближче до

твоїх сервісів. Тобто він дає можливість отримати вендорський PoP в будь яку з локацій, де він ще не доступний, тим самим отримати повний функціонал з мінімальними затримками для роботи додатку і замовників, які використовують цей сервіс.

Чому для деяких замовників наявність локальної точки присутності це важливо?

По-перше, SE дає змогу працювати з трафіком локально, не ганяючи все в інтернет до найближчого PoP, прибираючи зайві затримки. Це критично для внутрішніх сервісів, приватних API та додатків, які не мають виходити назовні і працюються виключно в інтранеті.

По-друге, SE створює той самий “єдиний контур”, про який F5XC та сам вендор постійно говорить: одна й та сама політика безпеки застосовується як на edge, так і всередині твоєї інфраструктури. Тобто він стає продовженням distributed cloud, тільки вже у твоїй приватній зоні.

Архітектурно SE — це lightweight-вузол, який складається з control-агента, data-plane проксі та сервісів мережевої взаємодії. Він підключається до глобального control plane через захищений тунель і одразу стає частиною mesh-платформи. Коли SE запускається у Kubernetes, він фактично виконує роль ingress/egress gateway з увімкненою L7-логікою, мікросегментацією, автоматичним mTLS і повним обліком API-трафіку. У віртуальних машинах чи bare metal він працює аналогічно, просто не прив'язаний до контейнерного оркестратора.

З точки зору роботи SE досить “розумний” компонент архітектури: він сам реєструється у control plane, тягне необхідні політики, отримує сертифікати, формує overlay-канали з іншими SE або PoP і будує локальну маршрутизацію. Вендором навмисно була вибрана така модель розгортання, для того, щоб понизити поріг входу для нових користувачів. По факту під капотом просто набір автоматизацій і певних скриптів, які починають розгортатися відразу після валідації токенів. Все, що ти робиш — задаєш правила та політики в одному порталі, а SE виконує їх на місці, без ручного тюнінгу. Він може працювати як локальний WAF/WAAP-захист,

як service mesh gateway, як L4/L7 балансувальник або як private link до твоїх хмар та інше.

Головна цінність у тому, що він об'єднує приватні середовища, мульти-хмарні середовища та edge-регіони у один логічний fabric. Без нього F5XC був би просто ще одним edge-хмарним сервісом. А з CE F5XC перетворюється на повноцінну distributed cloud платформу, яку можна використовувати і імплементувати на будь-яку інфраструктуру — навіть ту, що не має доступу в інтернет або працює з чутливими даними. В цьому випадку це є основною перевагою відносно інших вендорів, які надають схожі послуги, це саме той функціонал, який нам демонструє те що вендор чітко розуміє замовника і задачі для яких дане рішення буде утилізовано.

2.2 Ключові компоненти архітектури F5xC

1.F5xC WAAP. Включає повний стек безпеки - WAF

- сигнатурний захист;
- ML-аналітика;
- automatic policy tuning;
- signature staging + policy tightening;
- вбудований positive security model.

API Security

- API discovery;
- автоматичне створення OpenAPI схем;
- anomaly detection;
- виявлення BOLA, mass assignment, injection in JSON, GraphQL misuse.

Bot Defense

- ML-класифікація;
- JS-челенджі;
- device fingerprinting;
- захист від credential stuffing та скрейпінгу.

L7 DDoS

- захист на edge-PoP;
- behavioural detection.

2. F5xC Global Load Balancing (GSLB)

- Anycast-routing.
- Health-check-и L7.
- Cross-cloud балансування.
- Автоматичний failover між регіонами/clouds.
- Multi-cloud service discovery.

3. F5xC Multi-Cloud Networking (MCN). Найсильніший компонент платформи який дозволяє:

- будувати mesh-мережу між AWS, Azure, GCP;
- згорнути overlay VPN автоматично між середовищами;
- створювати tenant-segmentation;
- управляти маршрутизацією на L3/L7;
- організувати Zero Trust East-West рух між сервісами.

Якщо спрощено то MCN це SD-WAN, Service Mesh, і cloud networking в одному.

F5 Distributed Cloud – The Platform

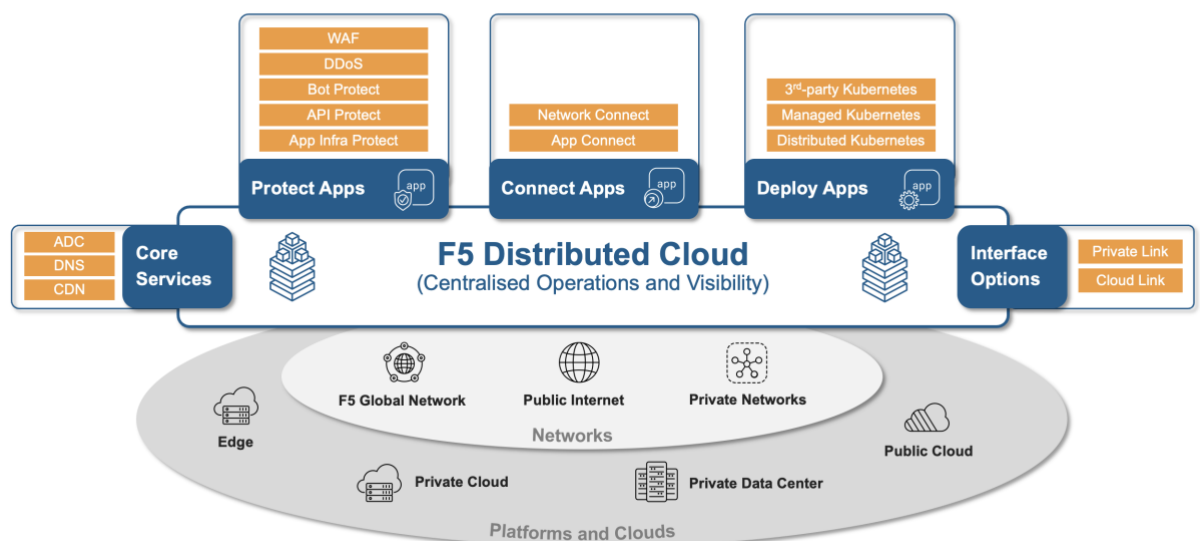


Рис 2.6. Схематичне зображення всіх перелічених функціональних модулів

4. F5xC Mesh (Service Mesh / API Gateway)

- mTLS між сервісами.
- Sidecar-less mode (gateway-based mesh).
- L7 routing.
- Контроль API flows.
- AuthZ/AuthN через OIDC/SAML/JWT.
- Rate limits / QoS для кожного endpoint.

5. Observability Suite. F5xC дає наступний pipeline:

- request-level traces;
- L7 трафік аналіз;
- API usage;
- real-time dashboards;
- logs → Elastic / Datadog / Splunk / S3;
- anomaly detection.

6. Моделі розгортання архітектури

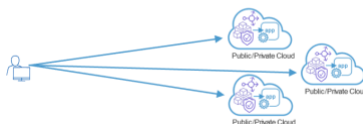
- SaaS Edge
- Захист виконується на PoP-ах F5XC.
- Hybrid Edge
- PoP + Customer Edge nodes.
- Private Edge

Regional Edge (RE)



- Only leverages the F5 global network
- Provides L3/L4 DDoS via network
- Advertises via Anycast
- Origins accessed via Public NAT/FQDN
- Global network can be access controlled (ACL) at origin sites

Customer Edge (CE)



- Does not leverage the F5 global network
- Site includes WAAP security services except L3/L4 DDoS
- Self-managed advertisement based on deployed locations
- SaaS managed Load balancing/WAF endpoints

Hybrid (RE + CE)



- Requires deployment at customer edge
- Provides for private address/FQDN access of origins
- WAF and additional security services at both the customer site and global network
- L3/L4 DDoS is provided via network

Рис 2.7. Варіанти розгортання з детальним описом переваг

7. Особливості платформи (одні з ключових)

- Єдина платформа для безпеки, мережі та delivery.
- Справжня multi-cloud підтримка, а не просто проксі через один cloud.
- Edge-обчислення на глобальних PoP.
- Поєднання WAF + API Security + Mesh + MCN в одному control-plane.
- Реальна Zero Trust модель East-West + North-South.
- Високий рівень автоматизації через ML.
- Повний visibility усіх flows у будь-якому cloud/on-prem.
- Sidecar-less Service Mesh — менше складності у Kubernetes.
- Гнучкі сценарії для enterprise, включно з banking/healthcare/gov.

8. Архітектурний принцип “Identity is the new perimeter”. F5xC реалізує Zero Trust концепцію через:

- identity-based routing;
- мікросегментацію;
- контекстні політики L7;
- continuous authentication;
- взаємний TLS між сервісами.

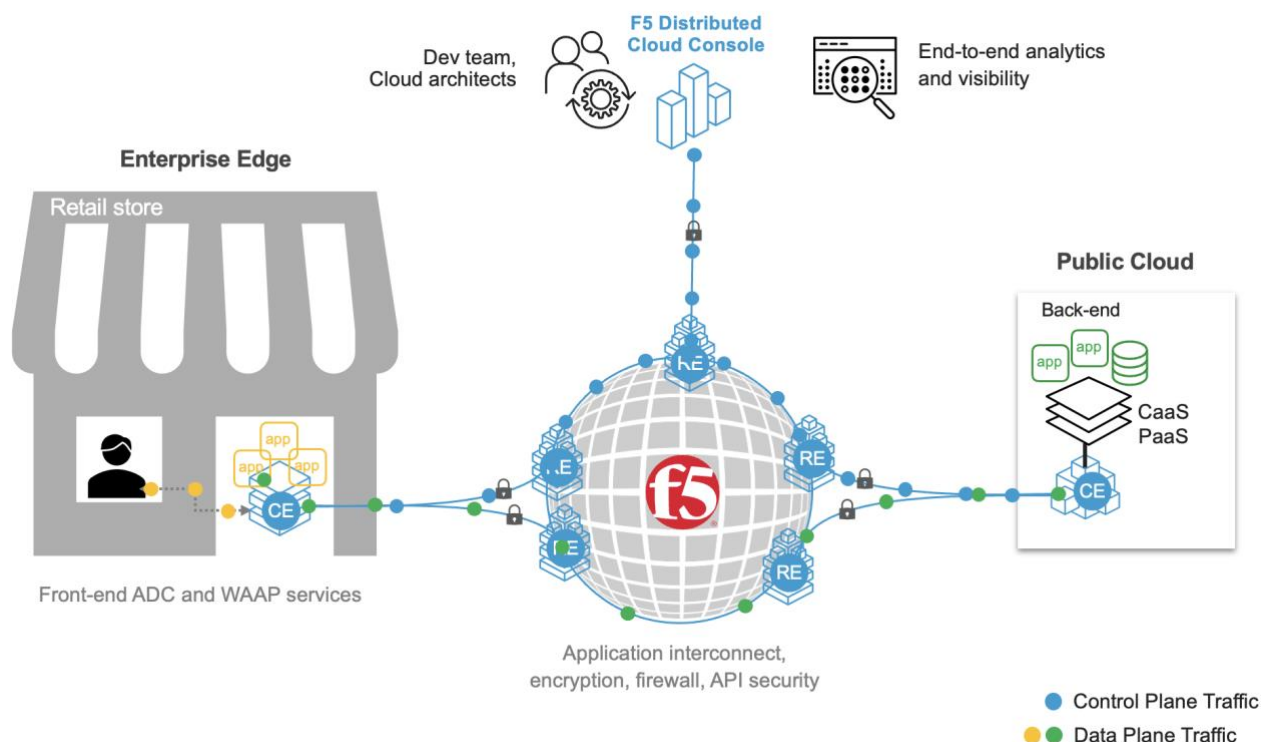
Як результат периметру більше нема. Є глобальний policy fabric, що проходить через edge-PoP і CE-ноди.

9. Як F5xC працює в реальному житті

1. Користувач заходить на додаток → трафік приходить на PoP або використовує CE ноду.

2. PoP/CE робить:
- TLS offload,
 - WAF/WAAP,
 - Bot detection,
 - API protection,
 - L7 routing.

3. Далі PoP/CE передає трафік у хмару клієнта або інший CE вузол.
4. CE робить mesh-різання, локальні політики, microsegmentation.
5. Observability фіксує увесь шлях запиту на дашборді системи.



22 ©2022 F5

Рис 2.8. Комплексна схема поєднання всіх користувача та сервісів з демонстрацією потоків трафіку

Важливо для розуміння розділяти різні рішення і різних вендорів, тому що в світі кібербезпеки не існує очевидно поганих рішень (окрім звичайно opensource для клієнтів рівня enterprise) існує проблема з розумінням і використанням правильного продукту. Відповідно частіше за все це зустрічається через відсутність достатньої компетенції у адміністраторів. Тому наступна матриця може стати корисною для розуміння функціональних відмінностей між рішеннями і вендорами.

Матриця порівняння функціональних модулів у різних вендорів

Платформа	WAAP	API sec	MCN	Mesh	Edge	Коментар
F5xC	наявний	наявний	наявний	наявний	наявний	Єдина платформа, що об'єднує все з описаного
Cloudflare	наявний	частково	відсутній	відсутній	наявний	Сильний edge, слабка multi-cloud мережа
Akamai	наявний	наявний	відсутній	відсутній	наявний	Один з топових CDN провайдерів, але обмежена платформа мережі
ZScaler	відсутній	відсутній	відсутній	відсутній	наявний	SASE підхід: продукт направлений на користувачів, а не на додатки
Palo Alto Networks	частково	наявний	частково	частково	відсутній	Не має edge мережі взагалі. Але сильний лідер в сфері

						мережевих файрволів.
--	--	--	--	--	--	-------------------------

Якщо підсумувати це порівняння стає очевидно, що F5xC це поєднання всіх інших вендорів(включно з класичною платформою F5 BIG-IP) в одному єдиному сервісі, який ще і є лідером в кожній з заявлених ніш і має великий досвід в реалізації доставки і безпеки додатків.

Перед тим як ми перейдемо до огляну самої консолі, функцій та особливостей, пропоную підвести підсумок всього попереднього матеріалу, щоб було більш структуроване явлення про рішення.

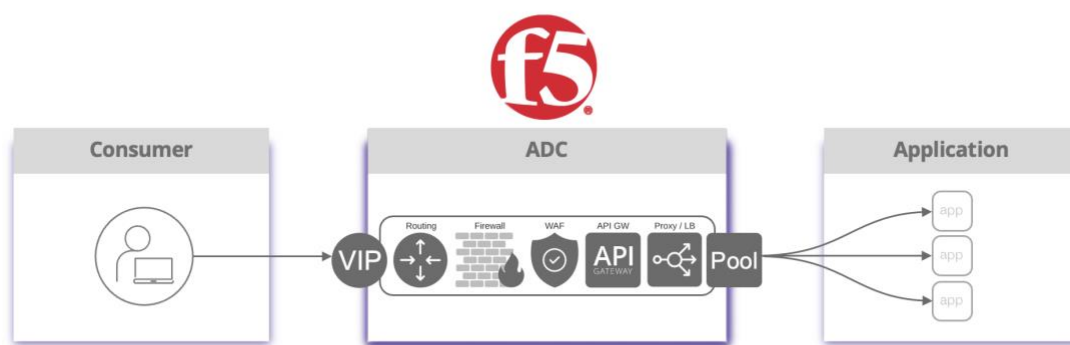


Рис 2.9. Класична архітектура публікації додатку

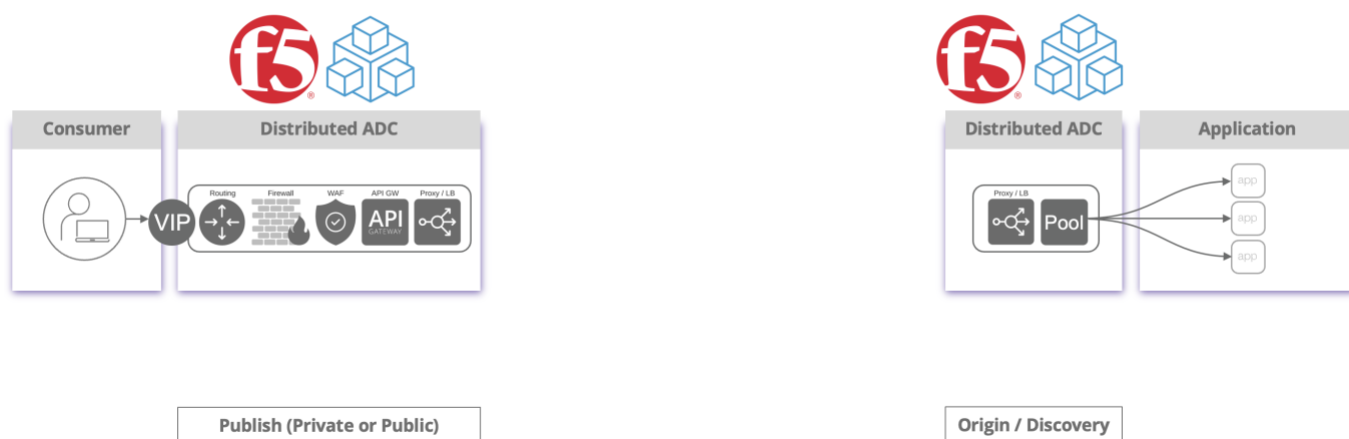


Рис 2.10. Наступний крок в розвитку коли з'являється потреба в рознесенні архітектури, з різних причин, будь то складність додатку, відмовостійкість чи масштабування і тд



Рис 2.11. Для зберігати зручність керування в сучасних мультиклаудних архітектурах потрібне рішення яке може все це центрально обробляти, публікувати, захищати і мати детальну прозорість для забезпечення достатнього рівня observability, саме для цього і імплементується F5xC

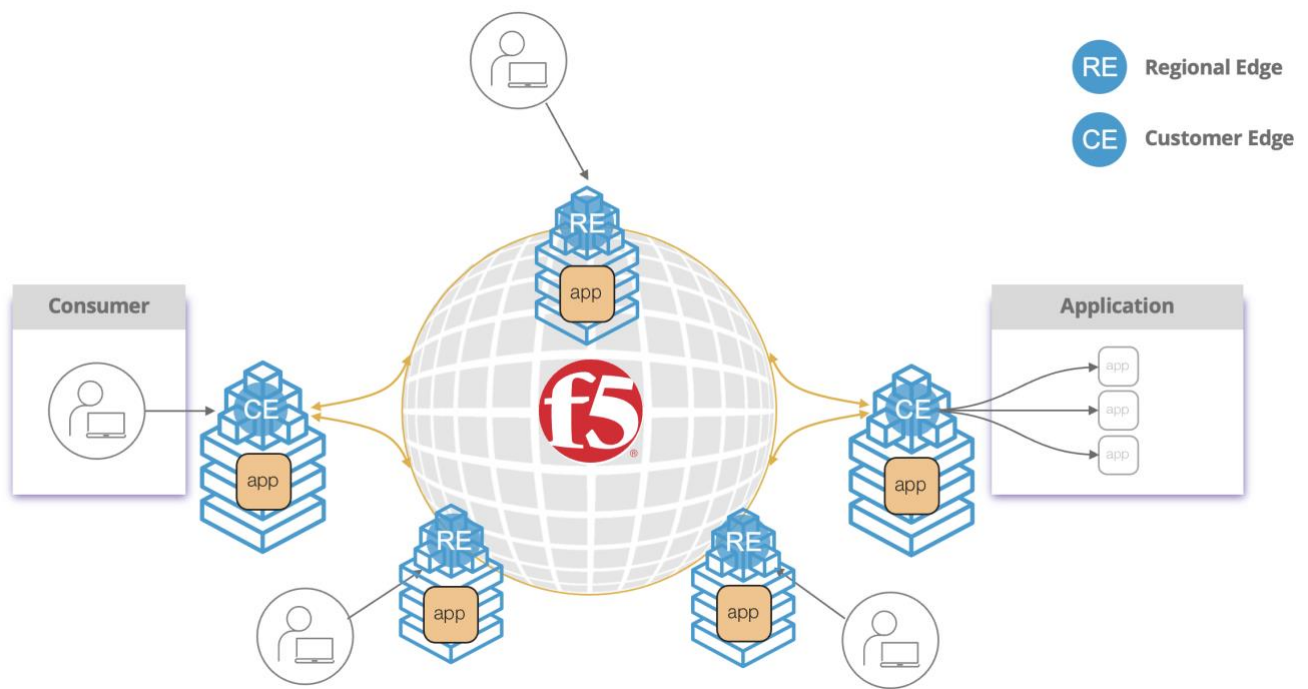


Рис 2.12. F5xС платформа надає нам можливість поєднати все в одному централізованому місці і забезпечити всі відповіді до сучасних викликів.

3. ТЕХНОЛОГІЯ НАЛАШТУВАННЯ F5xС WAAP ТА РЕКОМЕНДАЦІЙ ЩОДО РОБОТИ ТА КОНФІГУРАЦІЇ.

3.1 Варіант налаштування консолі та публікація додатку

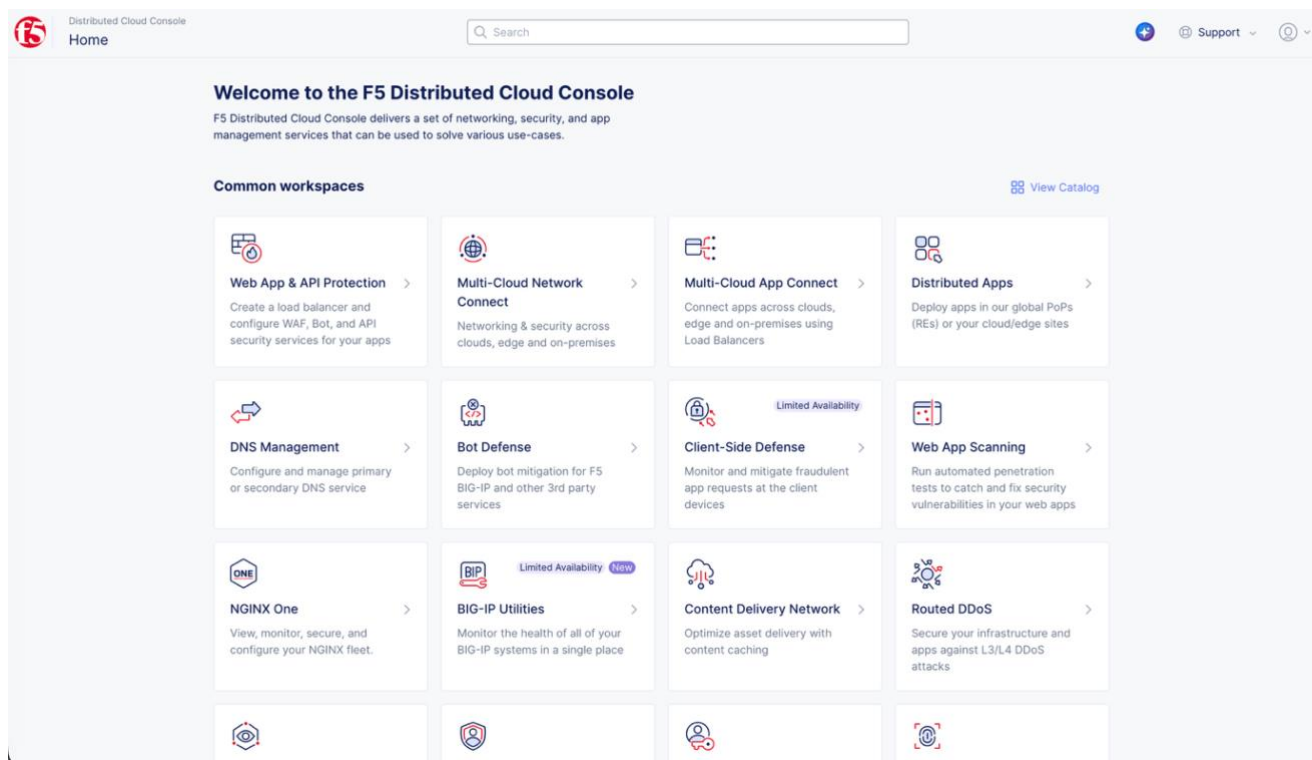


Рис 3.1. Головна сторінка консолі F5xС.

Тут ми бачимо головний інтерфейс консолі F5 Distributed Cloud, своєрідну «точку входу» до всієї екосистеми сервісів F5xС. У верхній частині бачимо привітальне повідомлення та короткий опис платформи, який наголошує, що F5 Distributed Cloud, що це єдина централізована система для керування Web App & API Protection, мережевими сервісами, CDN, Zero Trust та багатьма іншими функціями.

Блоки робочих просторів згруповані за ключовими можливостями:

- Web App & API Protection — основний модуль для роботи з WAAP, захистом вебзастосунків, API та бот-менеджментом.

- Multi-Cloud Network Connect і Multi-Cloud App Connect — інструменти побудови мережі між різними хмарними середовищами.
- Distributed Apps, DNS Management, Client-Side Defense, Bot Defense, DDoS Protection — окремі модулі, які відповідають за розширені політики безпеки та оптимізацію роботи додатків.

Кожен із цих елементів оформлено у вигляді окремої плиточки з коротким описом, що робить консоль інтуїтивно зрозумілою та дозволяє швидко орієнтуватися навіть у великій кількості функцій. Візуально можна стверджувати, що інтерфейс виглядає чисто та сучасно: всі тренди на мінімалістичні іконки збережені, світла кольорова схема (і звичайно ж наявна темна для всіх адміністраторів) та зрозуміла структура підкреслюють орієнтацію платформи на профільне використання та зручність адміністрування.

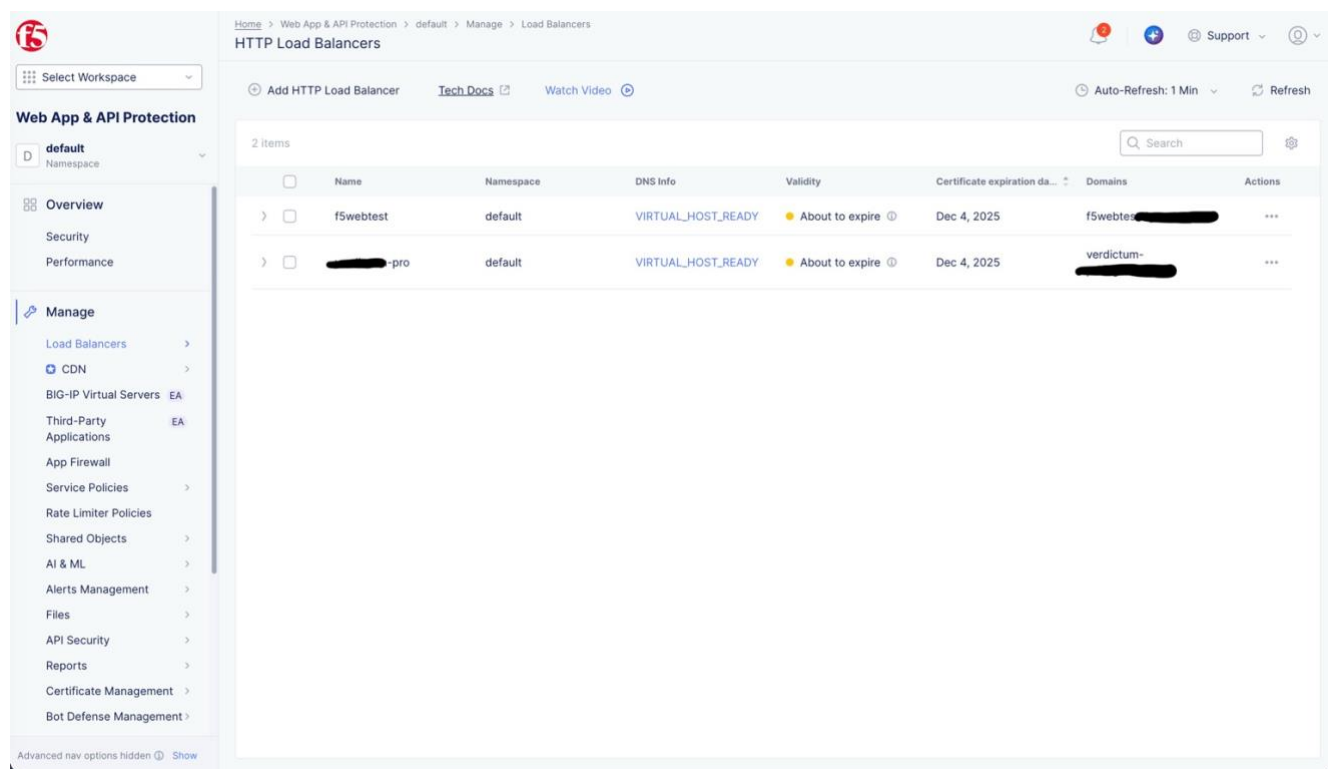


Рис 3.2. Вигляд під-меню WAAP

На цій сторінці адміністратор бачить усі створені балансувальники та їхній поточний стан. Відповідно з цього меню і відбувається публікація і керування

додатком. Тут відображені основні параметри кожного об'єкта: його назва, namespace, статус готовності, дата створення та домен із сертифікатом.

Зліва розташоване меню модуля Web App & API Protection, де зібрані всі інструменти для роботи із захистом вебзастосунків — WAF, політики API, бот-менеджмент, сертифікати та інші налаштування.

The screenshot shows the 'Add HTTP Load Balancer' configuration page in the F5 Web App & API Protection console. The left sidebar contains a menu with options like 'New', 'Domains and LB Type', 'Origins', 'Routes', 'Web Application Firewall', 'Bot Protection', 'API Protection', 'Malware Protection', 'DoS Settings', 'Client-Side Defense', 'Common Security Controls', and 'Other Settings'. The main configuration area is divided into three sections: 'Metadata' with a 'Name' field set to 'test-app', 'Domains and LB Type' with a table containing one domain 'test.app1.com', and 'Load Balancer Type' with a dropdown set to 'HTTPS with Automatic Certificate'. There are also checkboxes for 'HTTP Redirect to HTTPS' and 'Add HSTS Header', and a 'HTTPS Listen Port Choice' dropdown set to 'HTTPS Listen Port'. At the bottom right, there is a blue button labeled 'Add HTTP Load Balancer'.

Рис 3.3. Демонстрація публікації нового додатку.

Інтерфейс для публікації виглядає максимально дружелюбним для нового користувача, все має вигляд простих і очевидних полів де потрібно давати «прості відповіді на питання» для того, щоб рухатись вперед по конфігурації. В прикладі публікується новий додаток, який має FQDN – test.app1.com.

The screenshot shows a selection screen for certificate types. There are three options: 'HTTPS with Automatic Certificate' (marked as 'Default'), 'HTTPS with Custom Certificate', and 'HTTP'. The 'HTTPS with Automatic Certificate' option is selected, and its description is visible: 'HTTPS load balancer with automatic public certificate provisioning. DNS records for the domains will be automatically managed by F5 Distributed Cloud. As a prerequisite, the domain must be delegated to F5 Distributed Cloud (see the DNS Management section) or a DNS CNAME record should be created in your DNS provider's portal (only for Domains not managed by F5 Distributed Cloud).'. At the bottom, there is a dropdown menu with 'HTTPS with Automatic Certificate' selected.

Рис 3.4. Вибір сертифікатів шифрування

The screenshot shows a selection screen for TLS security levels. There are four options: 'High' (marked as 'Default'), 'Medium', 'Low', and 'Custom'. The 'High' option is selected, and its description is visible: 'TLS v1.2+ with PFS ciphers and strong crypto algorithms.'. The other options have their descriptions partially visible: 'Medium' (TLS v1.0+ with PFS ciphers and medium strength crypto algorithms), 'Low' (TLS v1.0+ including non-PFS ciphers and weak crypto algorithms), and 'Custom' (Custom selection of TLS versions and cipher suites). At the bottom, there is a dropdown menu with 'High' selected.

Рис 3.5. Вибір версії TLS

Рис 3.6. Варіації типів Origin Сервера

Рис 3.7. Налаштування Origin Сервера

F5xС дозволяє використовувати публічний IP, приватну адресу, DNS-ім'я або інші типи ресурсів у залежності від архітектури користувача для підключення до Origin сервера. Це робить такий підхід гнучким і зручним для різних сценаріїв — від хмарних сервісів до локальної інфраструктури. Тобто нам при публікації взагалі неважливо де додаток, що не вимагає ускладнень в розширенні доступів.

На зображенні ми можемо побачити детальне заповнення параметрів сервера: IP-адресу, порт, тип підключення, політику health-check та спосіб балансування навантаження.

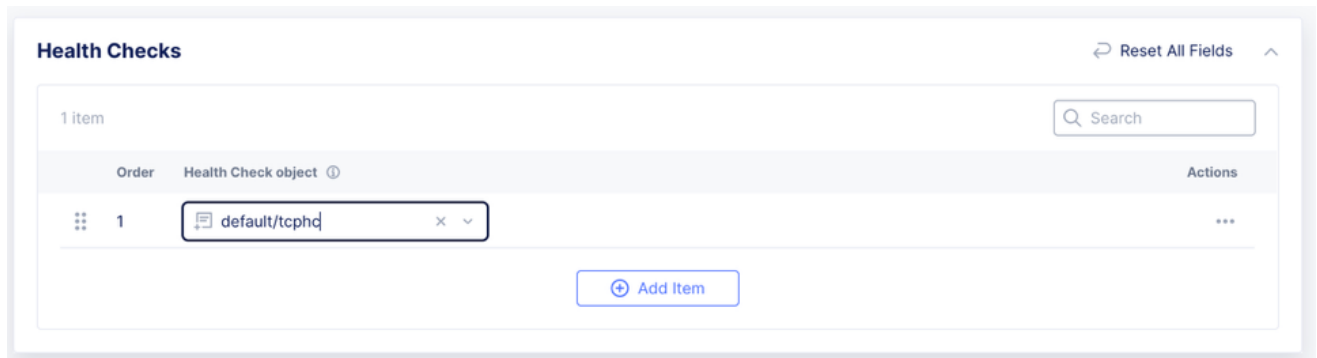


Рис 3.8. Налаштування Health Check на рівні L4

Правильне налаштування health-checkів у балансуванні — це одна з ключових умов стабільної роботи будь-якого вебзастосунку.

Сенс health-checkів дуже простий: балансувальник не може без них зрозуміти, чи живий сервер, чи відповідає він коректно, і чи варто надсилати на нього трафік. Якщо health-check налаштований неправильно або надто поверхнево, балансувальник може продовжувати направляти запити на сервер, який уже недоступний, працює з помилками або віддає некоректні відповіді.

Це призводить до:

- часткових або повних збоїв у роботі застосунку;
- втрати користувацького трафіку;
- затримок, повторних спроб і зрештою погіршення загального UX;

Правильно підібраний health-check гарантує, що балансувальник відправлятиме трафік лише на здорові, стабільні backend-ноди, автоматично виключаючи ті, що не проходять перевірку. Це дозволяє платформі реагувати швидше, ніж це зможе зробити адміністратор, забезпечуючи безперервність сервісу навіть при часткових відмовах.

3.2 Налаштування технології F5xC WAAP

Web Application Firewall

Web Application Firewall (WAF) ⓘ

Enable ⓘ

* Enable ⓘ

Select item ⓘ

WAF Exclusion ⓘ

Inline Rules ⓘ

WAF Exclusion Rules ⓘ

Not configured [Configure](#) >

Data Guard Rules ⓘ

Not configured [Configure](#) >

Cross-Site Request Forgery Protection ⓘ

Not configured [Configure](#) >

GraphQL Inspection ⓘ

Not configured [Configure](#) >

Cookie Protection ⓘ

Not configured [Configure](#) >

Рис 3.9. Огляд можливостей WAF

Metadata ⓘ

* Name ⓘ

testapp

Labels ⓘ

Add Label

Description ⓘ

* Enforcement Mode ⓘ

Monitoring

Security Policy Settings

* Security Policy ⓘ

Default

Advanced configuration

* Allowed Response Status Codes ⓘ

Default

* Mask Sensitive Parameters in Logs ⓘ

Default

* Blocking Response Page ⓘ

Default

Рис 3.10. Створення політики WAF

Тут продемонстровано процес увімкнення та базового налаштування Web Application Firewall (WAF). У верхній частині є перемикач для активації WAF. Після його увімкнення користувач може задати набір правил, обрати політику, налаштувати винятки та визначити рівень контролю. Далі система пропонує конфігурувати важливі підмодулі безпеки:

- Data Guard для захисту чутливих даних(DLP),
- CSRF Protection,
- GraphQL Inspection,

- Cookie Protection та інші механізми, які разом формують багато-шаровий підхід до захисту.

Нижче знаходиться секція з метаданими та режимом Enforcement Mode, де адміністратор може обрати, чи працюватиме політика у режимі моніторингу чи повного блокування. Це дозволяє тестувати правила без ризику зачепити легітимний трафік — критичний етап для будь-якого WAF.

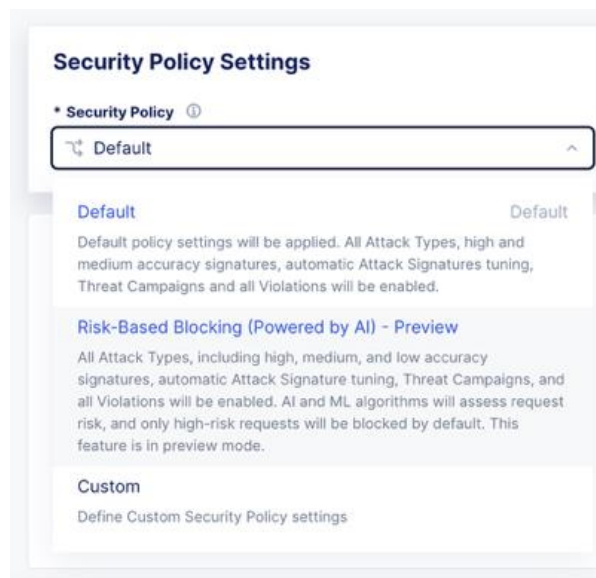


Рис 3.11. Варіації створення політики безпеки

Вибір Security Policy — фактично це центральний елемент роботи WAF. Саме ця політика визначає, як продукт реагуватиме на підозрілий трафік: що він пропустить, що заблокує, а що позначить як ризик.

У випадваючому меню доступно кілька варіантів.

Default — це базовий набір правил, який охоплює основні види атак та сигнатури й підходить для більшості стандартних застосунків.

Risk-Based Blocking - використовує машинне навчання для оцінки ризику кожного запиту. Такий підхід дозволяє більш гнучко відрізнати реальні загрози від шуму, особливо коли трафік складний або нестабільний.

Також є можливість створити Custom політику, коли застосунку потрібні специфічні правила, наприклад під окремі API чи нестандартну логіку. Це корисно

в ситуаціях, коли дефолтних налаштувань мало, а поведінку безпеки потрібно підлаштувати більш детально.

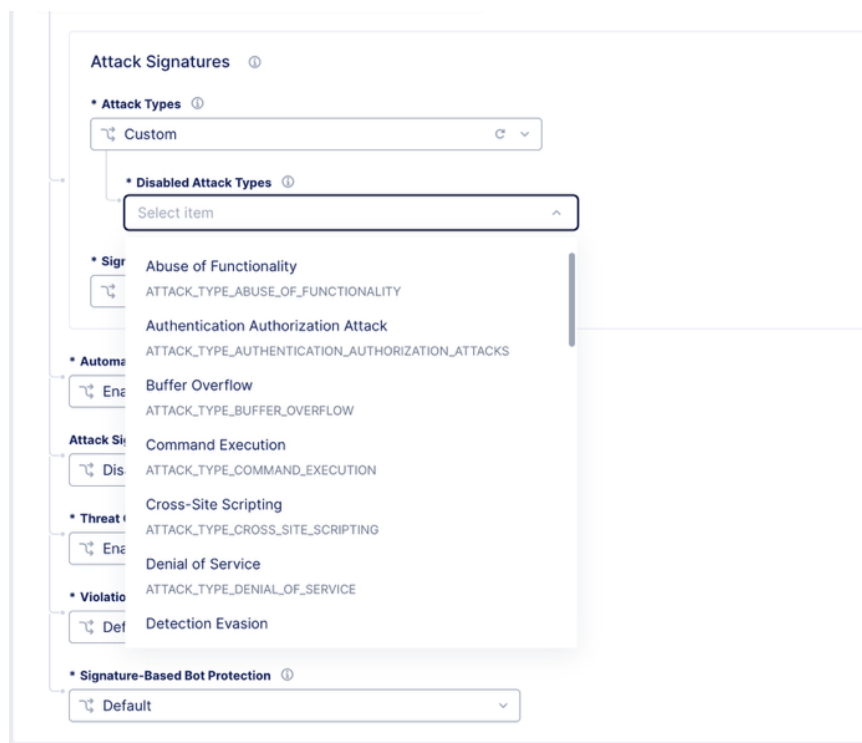


Рис 3.12. Виключення типів атак.

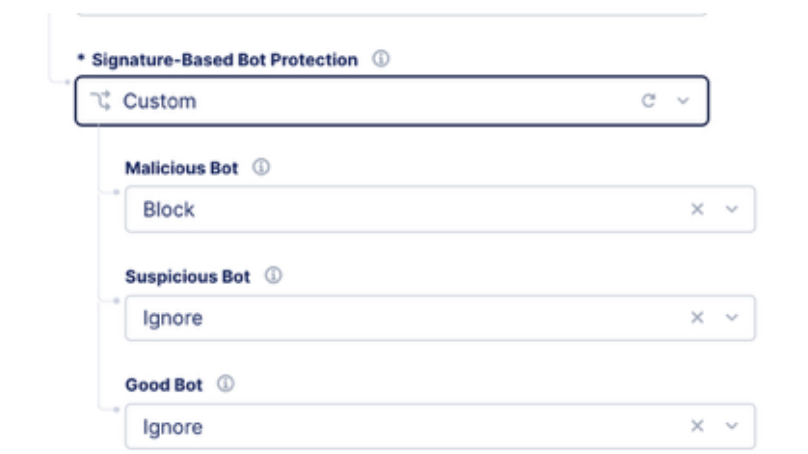


Рис 3.13. Конфігурації реакції політики безпеки на різні типи ботів базуючись на
сигнатури

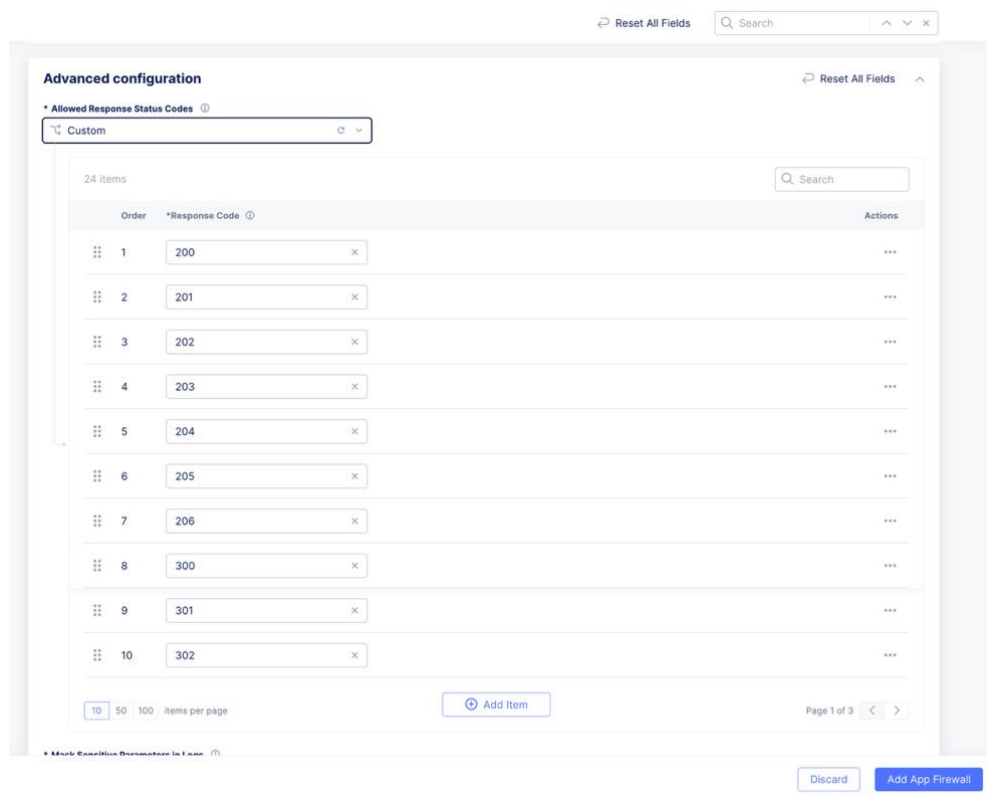


Рис 3.14. Можливість тюнінгу HTTP Status Codes в налаштуваннях політики.

Правильне налаштування дозволених HTTP-статус-кодів у політиці WAF це набагато важливіша річ, ніж може здатися на перший погляд. WAF повинен розуміти, яка відповідь сервера вважається нормальною, а яка може свідчити про атаку, помилку застосунку або несанкціоновану поведінку.

Некоректні статус-коди можуть бути ознакою експлойту.

Наприклад, якщо бекенд раптом починає віддавати 500-серію, це може бути ознакою SQL-ін'єкції або інших ін'єкцій, які «ламають» внутрішню логіку.

WAF, знаючи нормальний діапазон відповідей, здатен реагувати швидше і точніше.

Або ж ситуація де не всі статуси мають бути доступні клієнтам.

Деякі внутрішні помилки застосунку не повинні виходити назовні, бо вони можуть «підказати» зловмиснику структуру бекенду або інформацію про модулі, бази даних чи мови програмування.

А найголовніше правильний список допомагає відфільтровувати false positives. Якщо WAF не знає, що для певного API валідною відповіддю є,

наприклад, 206 або 429, він може сприйняти це як аномалію та заблокувати легальний трафік.

*** Blocking Response Page** ⓘ

Custom

Response Code ⓘ

200 OK

Blocking Response Page Body ⓘ

☒ Text ☐ Base64

<html><head><title>Request Rejected</title></head><body>The requested URL was rejected. Please consult with your administrator.

Your support ID is: {{request_id}}

[Go Back]</body>

Рис 3.15. Можливість адаптації сторінки блокування під стандарти компанії.

Web App & API Firewall

HTTP Load Balancer verdictum-pro

Bot Protection

Bot Defense ⓘ

Enable Bot Defense Standard

* Bot Defense Region ⓘ

EU

* Bot Defense Policy ⓘ

Configured Edit Configuration Reset Configuration

API Protection

API Definition ⓘ

Disable

API Protection Rules ⓘ

Not configured Configure

JWT Validation ⓘ

Not configured Configure

API Discovery ⓘ

Enable

Learn From Traffic With Redirect Response ⓘ

Enable Learning From Redirect Traffic

Purge Duration for Inactive Discovered APIs from Traffic ⓘ

7

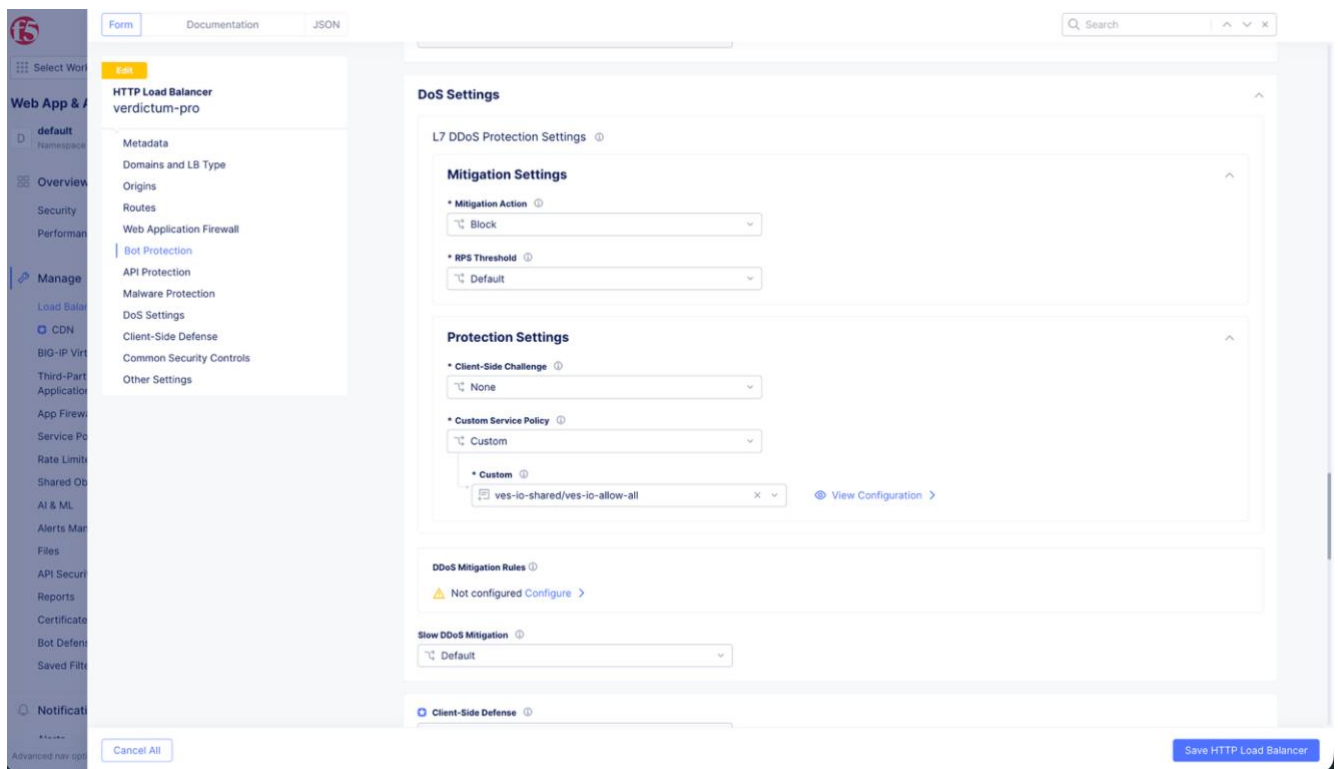
Cancel All Save HTTP Load Balancer

Рис 3.16. Налаштування захисту від ботів та захисту API

Захист від ботів дозволяє відрізнити реальних користувачів від автоматизованих скриптів, які можуть виконувати небажані або небезпечні дії:

перебір облікових даних, скрейпінг контенту, генерацію фейкового трафіку чи підготовку до атаки. У налаштуваннях обирається політика, яка визначає, як система реагуватиме на виявлених ботів — просто фіксуватиме їхню активність або повністю блокуватиме. Доступно два різних режими, базовий сигнатурний та розширений з фінгерпрінтами та додатковими валідаціями.

Блок API Protection саме тут задаються правила та пороги чутливості, за якими WAF аналізує звернення до API. Є можливість налаштовувати рівні захисту, механізми фіксації аномалій та дії при порушеннях дозволяє суттєво зменшити ризик зловживань логікою сервісу.



3.17. Налаштування правил захисту від DDoS

Зрозуміло, що в рамках налаштування WAAP у нас є можливість налаштувати тільки профіль захисту від DDoS на L7, тому що це саме той рівень за який відповідає цей інструмент захисту. Але загалом у нас є можливість налаштувати комплексних захист, який буде включати в себе DDoS захист на L3-4, але це все робиться з іншого сабменю

3.3 Експериментальні результати дашбордів та статистики F5xC WAAP

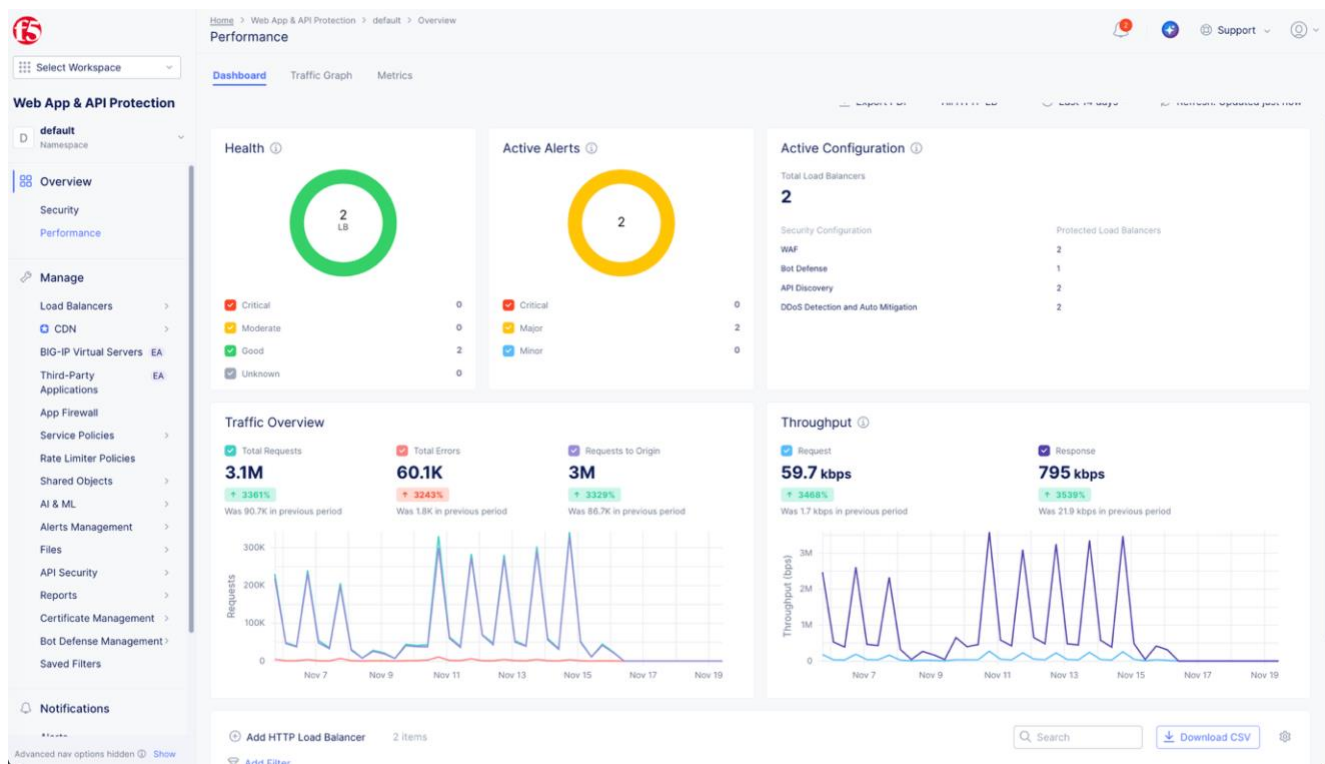


Рис 3.18. Центральний дашборд моніторингу продуктивності та безпеки вебзастосунків у F5 Distributed Cloud WAAP

На зображенні показано сторінку Performance у консолі F5 Distributed Cloud, яка використовується для загального контролю стану вебзастосунків та API. Ця панель дає узагальнену картину того, як працює система захисту і як поводить себе трафік у реальному середовищі.

У верхній частині відображені індикатори здоров'я сервісів і активних сповіщень. Вони дозволяють швидко зрозуміти, чи є критичні проблеми, та оцінити поточний рівень стабільності балансувальників і політик безпеки. Поруч наведено інформацію про активні конфігурації, зокрема кількість балансувальників та увімкнених механізмів захисту, таких як WAF, захист API, бот-менеджмент і DDoS-митигування.

Нижче розміщені графіки огляду трафіку, які показують загальну кількість запитів, кількість помилок і обсяг запитів, що доходять до origin-серверів. Це

дозволяє оцінити не лише навантаження на систему, а й ефективність фільтрації небажаного трафіку. Окремо відображено показники пропускної здатності, які дають уявлення про інтенсивність обміну даними між користувачами та сервісами.

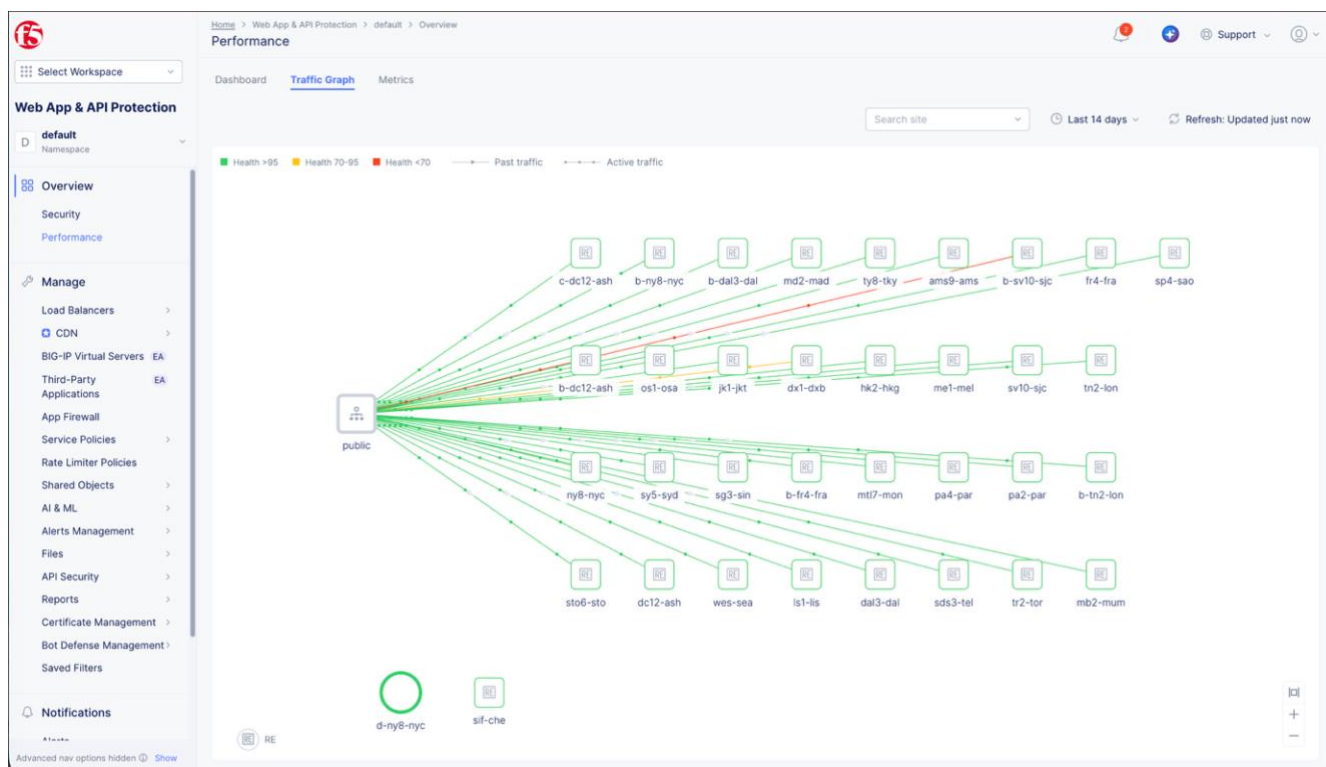


Рис 3.19. Графічне представлення розподілу трафіку та стану регіональних вузлів(PoP) у F5 Distributed Cloud

На зображенні вище продемонстровано сторінку Traffic Graph у консолі F5 Distributed Cloud, яка наочно відображає, як користувацький трафік розподіляється між різними регіональними вузлами платформи. У центрі схеми знаходиться публічна точка входу, від якої трафік спрямовується до численних edge-локацій у різних географічних регіонах.

Кожна точка присутності на схемі відповідає окремій локації або дата-центру, а з'єднувальні лінії показують активні маршрути передачі трафіку. Колір ліній та індикатори стану дозволяють швидко оцінити здоров'я кожного вузла і побачити, де трафік проходить без проблем, а де можуть виникати погіршення або затримки.

Такий граф дає зрозуміти, що трафік не концентрується в одному місці, а рівномірно розподіляється між різними регіонами, що підвищує доступність і

стійкість сервісу. У спрощеному вигляді це наочна демонстрація принципу distributed cloud, коли система автоматично використовує здорові та доступні вузли для обслуговування користувачів. Також адміністратором має можливість детально розуміти, які саме точки присутності вендора задіяні в поточний момент для вирішення запитів користувачів.

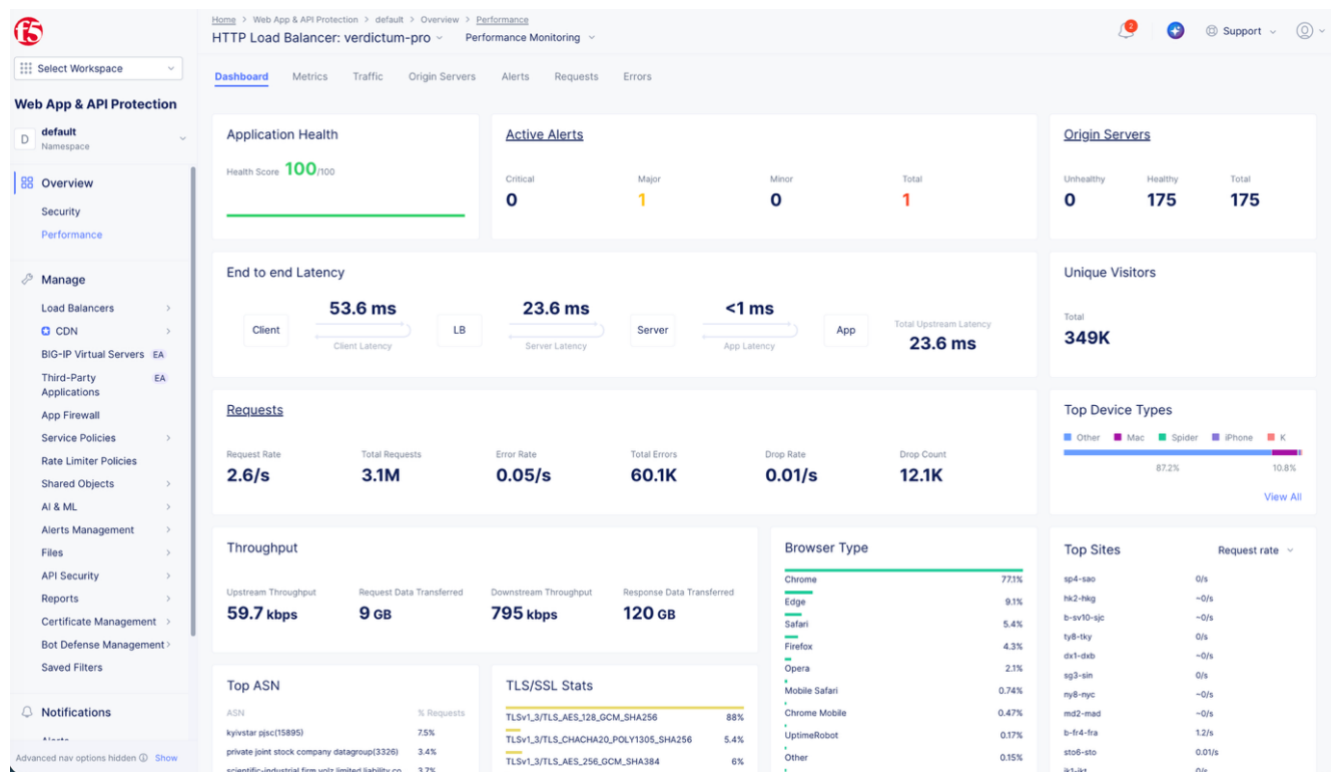


Рис 3.20. Панель моніторингу стану та продуктивності HTTP Load Balancer у F5 Distributed Cloud

Тут представлено сторінку моніторингу конкретного HTTP Load Balancer у консолі F5 Distributed Cloud. Цей екран використовується для оцінки реального стану застосунку та якості обслуговування користувацького трафіку.

У верхній частині показано загальний показник здоров'я застосунку, який у даному випадку свідчить про стабільну роботу сервісу. Поруч відображається інформація про активні сповіщення та стан origin-серверів, що дозволяє швидко зрозуміти, чи всі бекенд-ресурси доступні та працездатні.

Центральна частина екрана поділена і виділена для відображення статистики по основним показникам затримки, де окремо відображається час відповіді з боку

клієнта, балансувальника та серверної частини. Це дає змогу визначити, на якому етапі може виникати затримка та чи впливає безпекова обробка на продуктивність.

Нижче наведено статистику запитів і пропускну здатність, яка показує загальне навантаження на сервіс, кількість оброблених запитів, помилок і переданих даних. Додатково відображається інформація про типи клієнтів, браузерів та джерела трафіку, що допомагає краще зрозуміти характер користувацької активності.

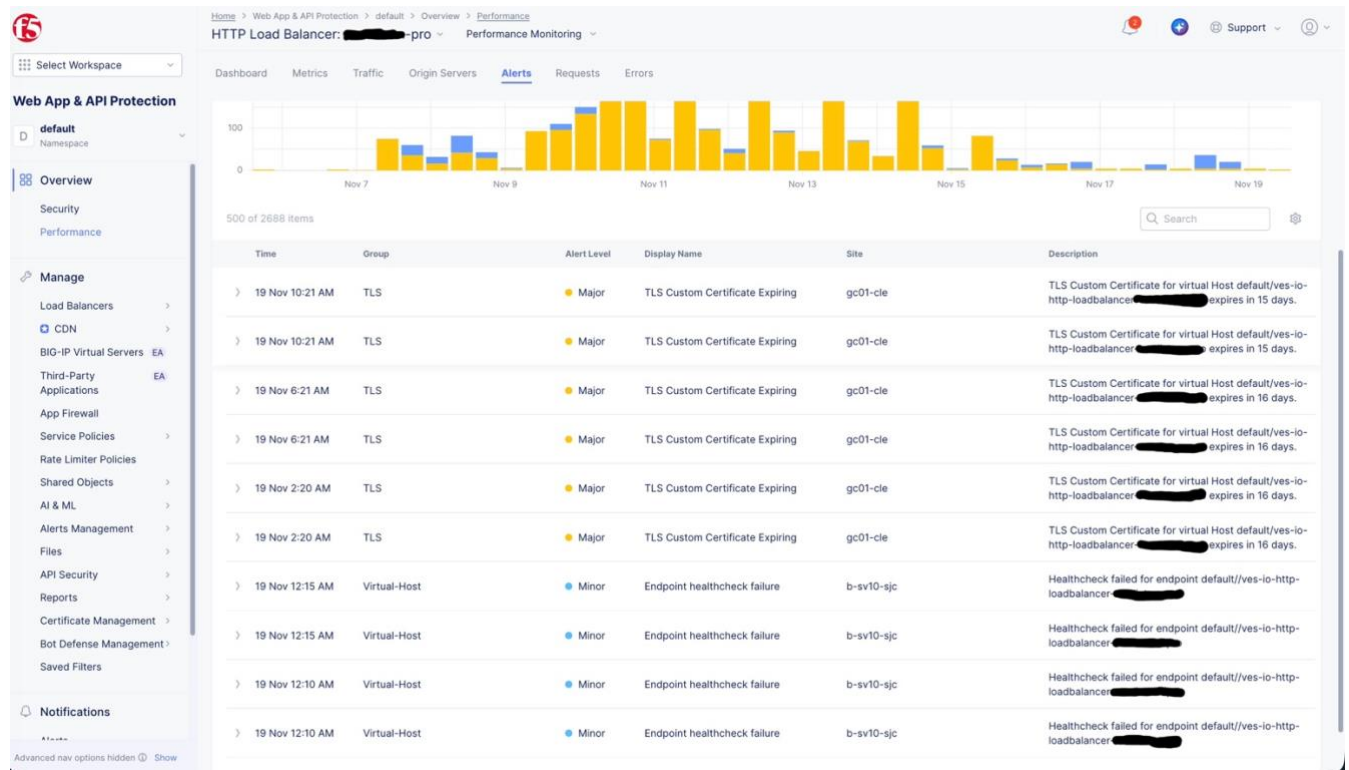


Рис 3.21. Дашборд з журналом сповіщень та подій безпеки HTTP Load Balancer у F5 Distributed Cloud

Далі у нас продемонстровано розділ Alerts для HTTP Load Balancer у консолі F5 Distributed Cloud. Ця сторінка використовується для перегляду подій і сповіщень, які виникають під час роботи сервісу та його захисту.

У верхній частині екрана відображено часову шкалу зі статистикою спрацювань, що дозволяє побачити періоди підвищеної кількості подій і загальну динаміку за вибраний проміжок часу. Нижче наведено таблицю з детальним

переліком сповіщень, де для кожного запису вказано час виникнення, рівень критичності та короткий опис причини.

З наведених даних видно, що система автоматично фіксує як технічні події, пов'язані з TLS-сертифікатами, так і проблеми доступності endpoint-ів, які визначаються за допомогою health-checkів. Це дозволяє своєчасно звернути увагу на потенційні ризики та усунути їх до того, як вони вплинуть на роботу користувачів.

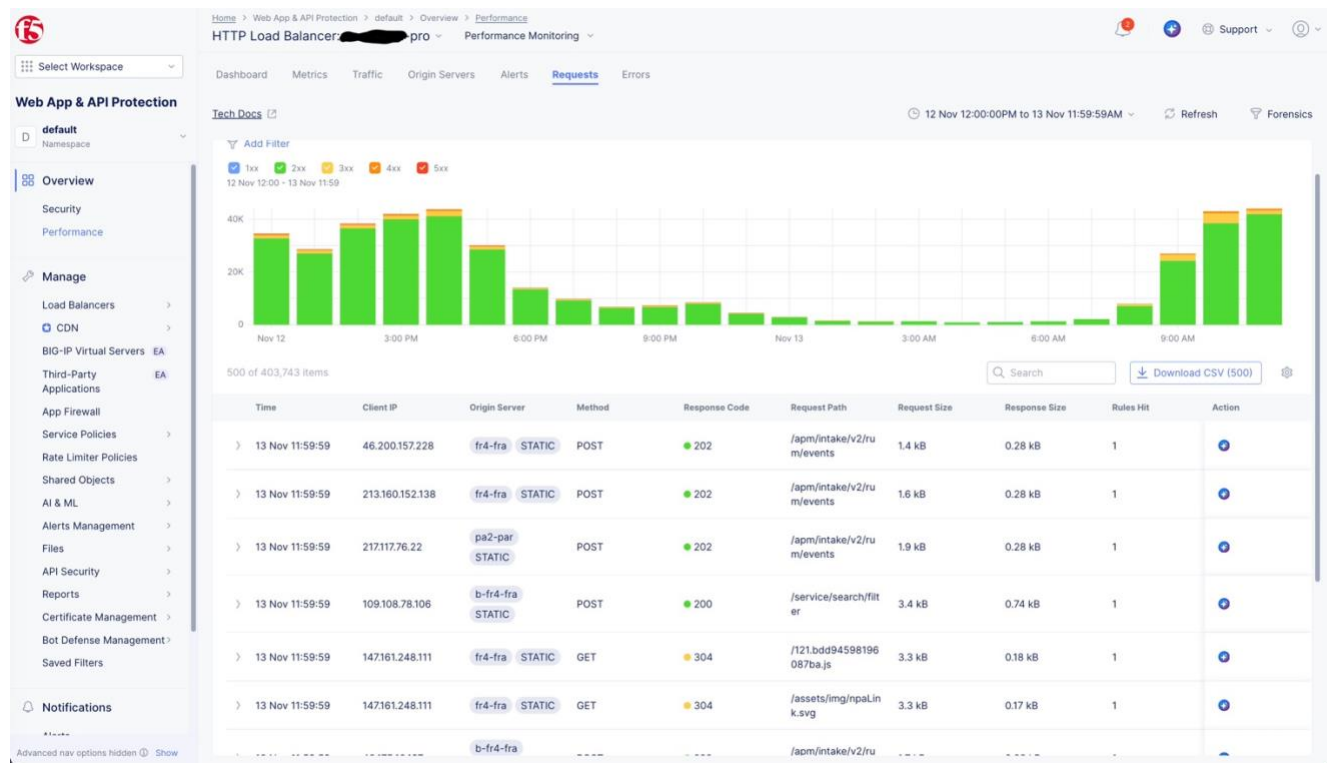


Рис 3.22. Аналіз HTTP-запитів та кодів відповідей у HTTP Load Balancer F5 Distributed Cloud

Розділ Requests для HTTP Load Balancer у консолі F5 Distributed Cloud. Цей екран використовується для детального аналізу користувацьких запитів, які проходять через балансувальник і систему захисту.

У верхній частині відображено графік активності запитів за часовим проміжком, де різними кольорами показано відповіді з різними HTTP-кодами. Це дозволяє швидко оцінити загальну картину трафіку та побачити періоди підвищеного навантаження або зростання кількості помилок.

Нижче наведено таблицю з детальною інформацією про окремі запити. Для кожного запиту показано час, IP-адресу клієнта, сервер-одержувач, HTTP-метод, код відповіді та шлях запиту. Також відображається розмір запиту й відповіді, що допомагає краще зрозуміти характер обміну даними між клієнтом і сервером.

Такий рівень деталізації дозволяє швидко знаходити нетипові або підозрілі запити, аналізувати причини помилок та оцінювати ефективність роботи балансувальника й механізмів WAAP.

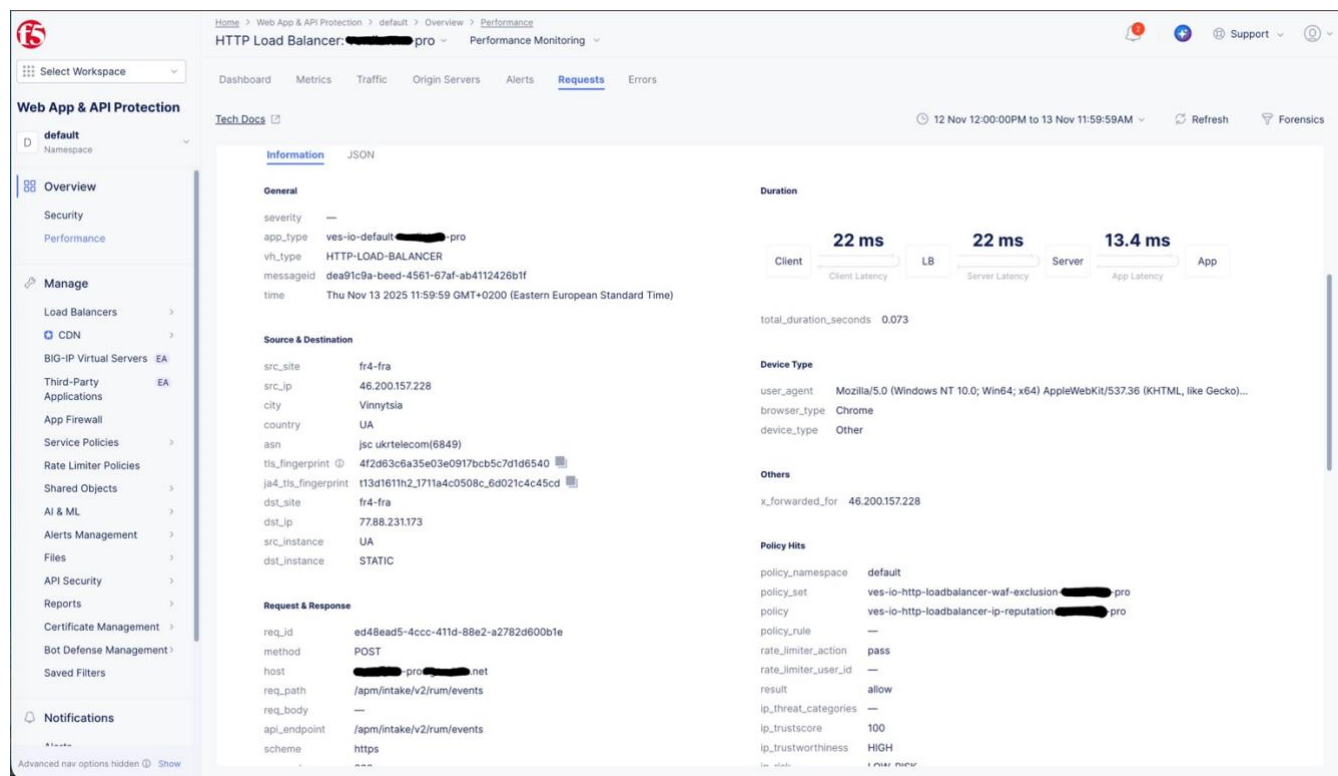


Рис 3.23. Детальна інформація про обробку окремого HTTP-запиту в F5 Distributed Cloud

Цей екран використовується для глибшого аналізу конкретної події, коли потрібно зрозуміти, як саме система опрацювала запит і які рішення були прийняті на різних етапах.

У верхній частині відображено показники затримки, де окремо видно час обробки на стороні клієнта, балансувальника та серверної частини. Це дозволяє швидко оцінити, де саме формується загальна затримка і чи впливає безпекова обробка на швидкодію застосунку.

Нижче наведено інформацію про джерело та призначення запиту, включно з IP-адресою клієнта, географічними даними та типом пристрою і браузера. Також показано параметри самого запиту — HTTP-метод, шлях до ресурсу та протокол. Окремий блок містить інформацію про політики безпеки, які були застосовані під час обробки запиту, та результат їх спрацювання.

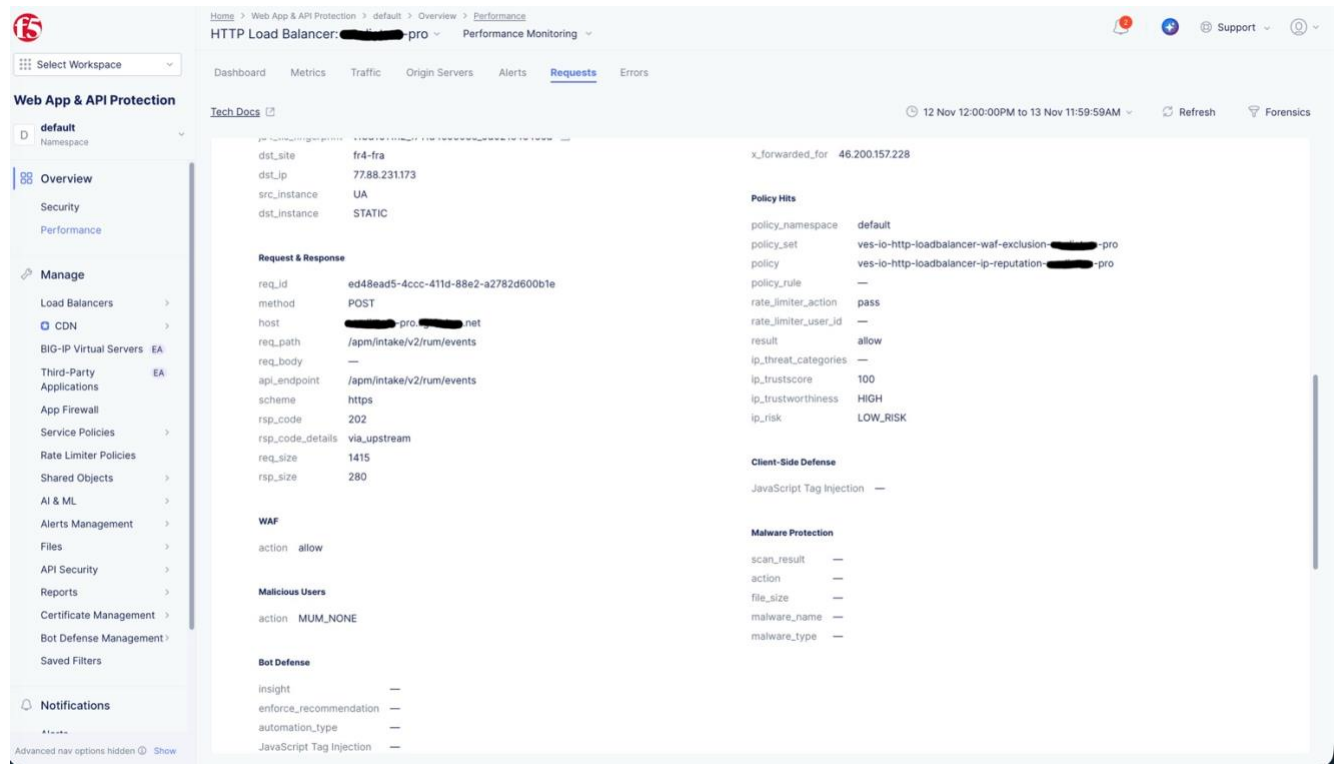


Рис 3.24. Результати застосування політик безпеки до HTTP-запиту в F5 Distributed Cloud WAAP

Тут відображено, які саме механізми безпеки були задіяні під час обробки конкретного запиту та з яким результатом.

У блоці Request & Response наведені базові технічні параметри запиту: HTTP-метод, шлях до ресурсу, код відповіді сервера та розмір переданих даних. Це дозволяє зрозуміти, як завершилася взаємодія між клієнтом і застосунком.

Праворуч показано інформацію про спрацювання політик безпеки. Зокрема видно, що WAF дозволив запит, оцінивши його як низькоризиковий, а також відсутність ознак шкідливої активності з боку користувача, ботів чи клієнтської

сторони. Окремо відображено, що механізми захисту від шкідливого ПЗ та client-side атак не виявили загроз.

У спрощеному вигляді цей екран демонструє, що WAAP не просто пропускає або блокує трафік, а приймає рішення на основі комплексної оцінки — враховуючи поведінку користувача, репутацію джерела та результати перевірок різних модулів безпеки. Це дозволяє забезпечити захист застосунку без зайвого впливу на легітимний трафік.

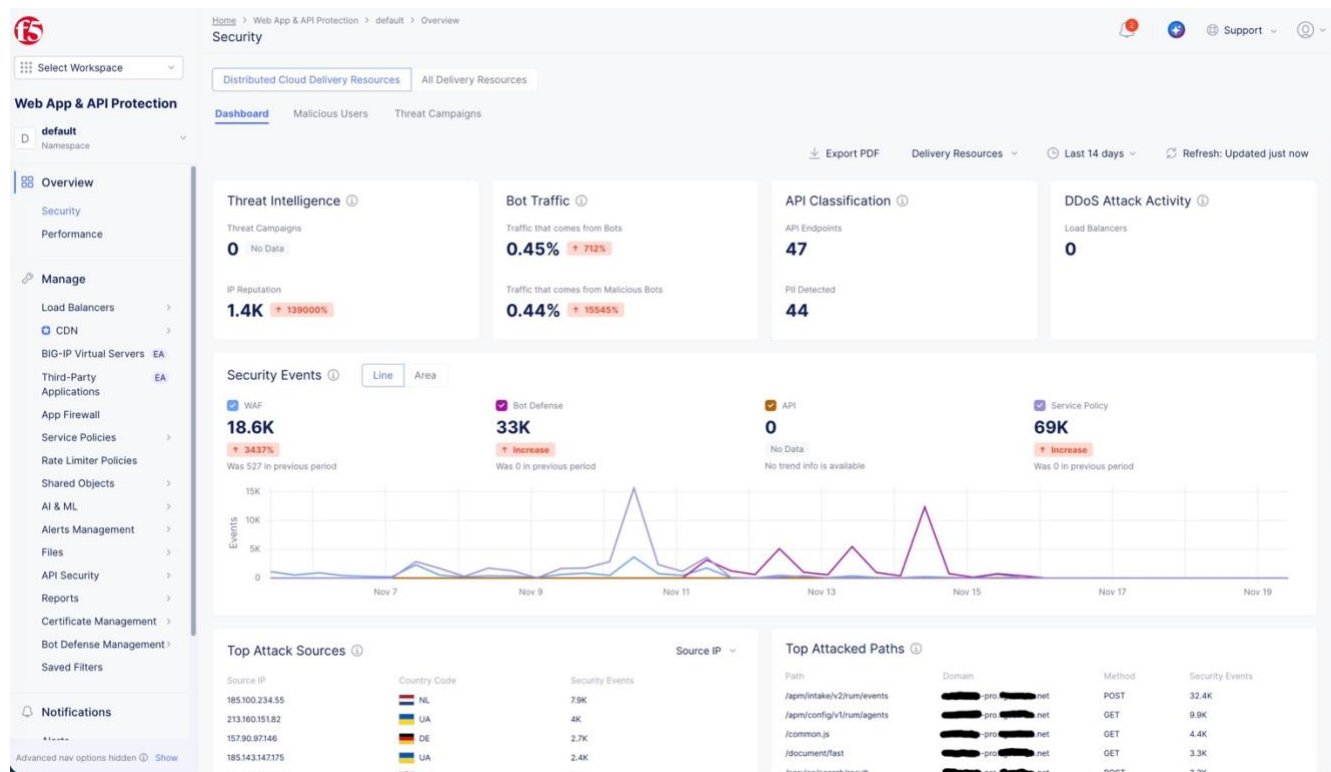


Рис 3.25. Огляд загальних безпекових подій та загроз у F5 Distributed Cloud WAAP

Розділ Security у консолі F5 Distributed Cloud, який надає загальну картину безпекового стану вебзастосунків та API. Цей екран використовується для швидкої оцінки того, з якими загрозами стикається система і які механізми захисту спрацьовують найчастіше.

У верхній частині відображені зведені показники, зокрема інформація про бот-трафік, класифікацію API, виявлення чутливих даних та активність DDoS-атак. Це дозволяє одразу зрозуміти, чи є масові атаки або нетипова поведінка трафіку.

Нижче наведено графік безпекових подій, де окремо показана активність WAF, бот-захисту та сервісних політик. За цими даними можна побачити періоди підвищеної загрозової активності та оцінити динаміку атак у часі.

У нижній частині екрана відображено джерела атак та найбільш атаковані шляхи запитів. Це дає розуміння, з яких регіонів або IP-адрес надходить найбільше підозрілого трафіку і які саме ресурси застосунку є найвразливішими.

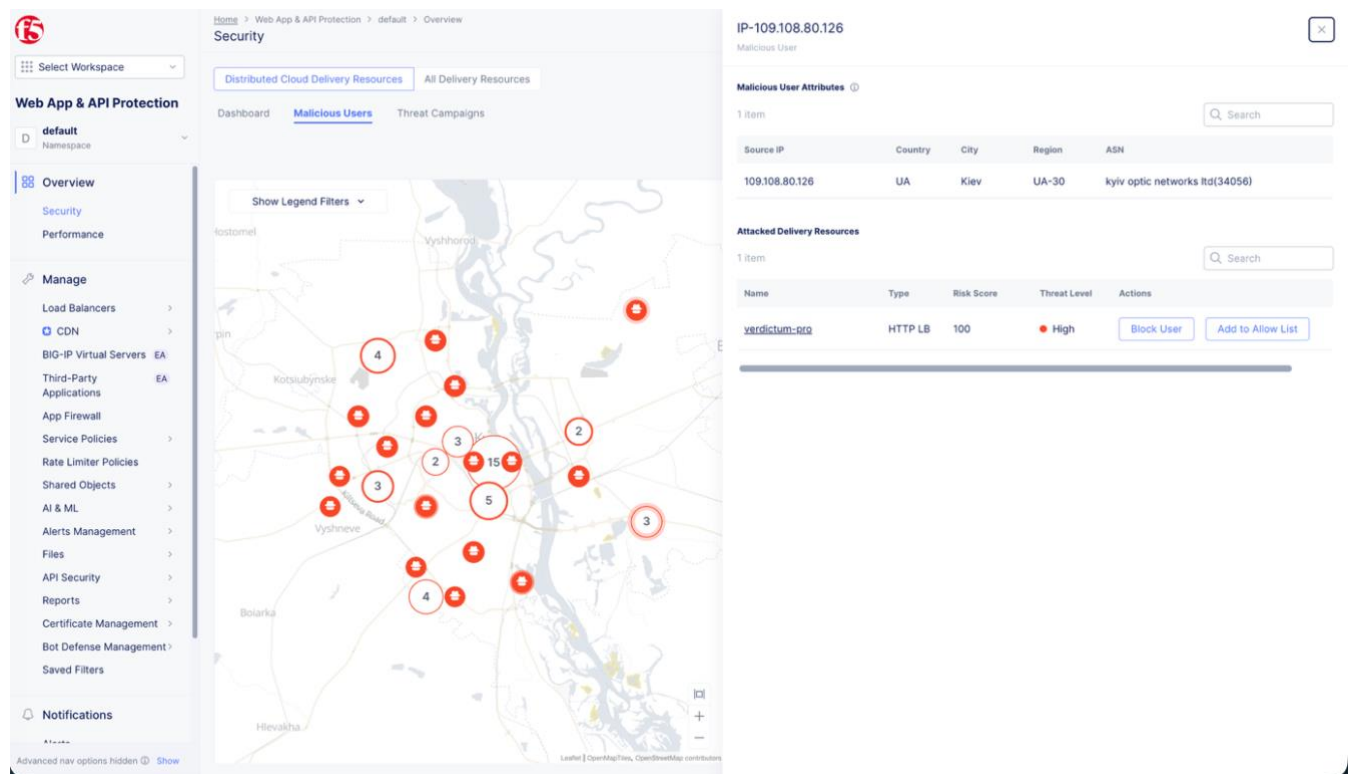


Рис 3.26. Виявлення та аналіз зловмисних користувачів у F5 Distributed Cloud WAAP

Розділ Malicious Users у F5 Distributed Cloud відображає активність користувачів, які були ідентифіковані системою як потенційно небезпечні. Центральна частина екрана містить карту з географічною прив'язкою, де маркерами позначено точки походження підозрілого трафіку. Така візуалізація дозволяє оцінити концентрацію атак у певних регіонах і побачити загальну картину поширення загроз.

У правій частині наведено детальну інформацію про вибране джерело трафіку, включно з IP-адресою, країною, містом, автономною системою та рівнем

ризик. Окремо зазначено, до якого ресурсу або балансувальника здійснювалися запити та яку оцінку загрози присвоїла система. Це дає змогу адміністратору швидко прийняти обґрунтоване рішення щодо подальших дій — обмеження доступу або коригування політик безпеки.

Такий підхід демонструє можливості WAAP щодо централізованого виявлення зловмисної активності, її кореляції з конкретними джерелами та оперативного реагування на загрози в межах єдиної платформи безпеки.

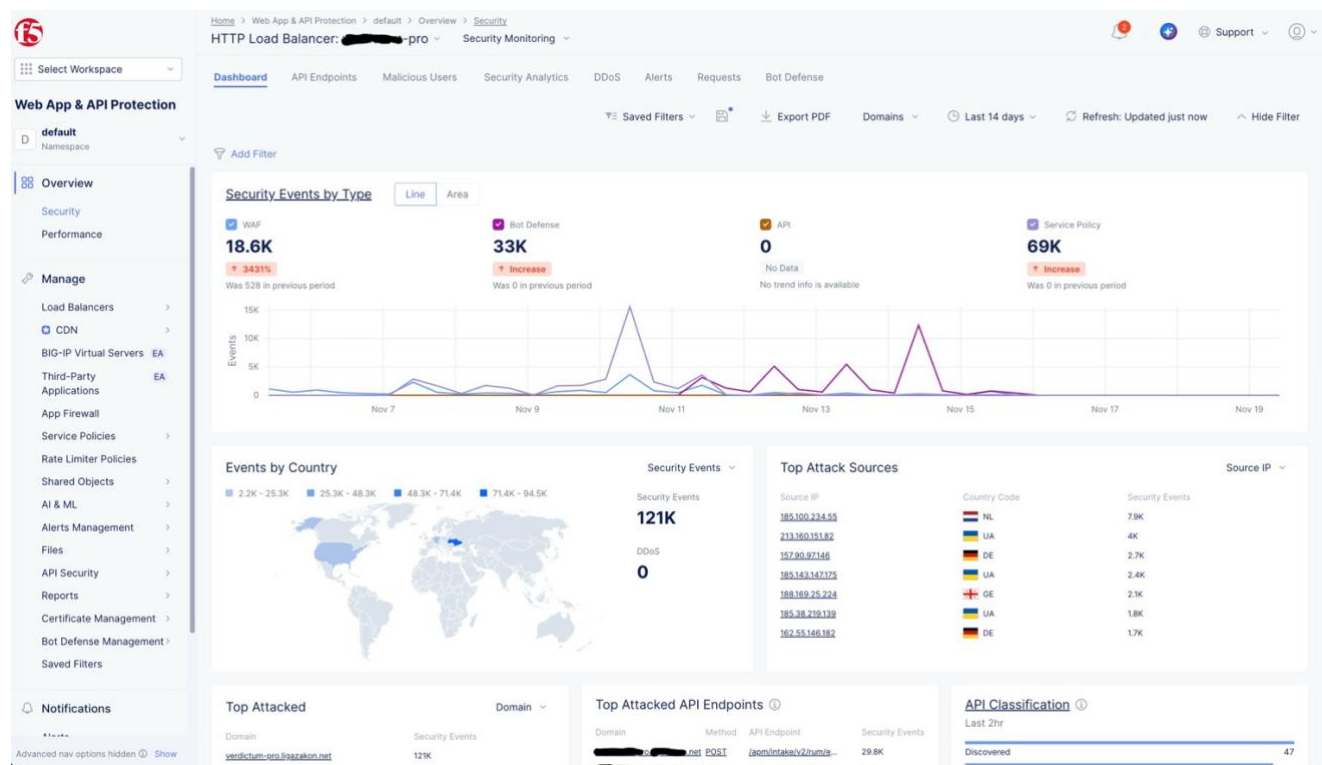


Рис 3.27. Аналітика безпекових подій для HTTP Load Balancer у F5 Distributed Cloud WAAP

Панель Security Monitoring відображає зведену аналітику безпекових подій, що були зафіксовані під час роботи HTTP Load Balancer. У верхній частині представлено динаміку спрацювань різних механізмів захисту, зокрема WAF, bot defense та сервісних політик. Графік дозволяє простежити зміну інтенсивності атак у часі та визначити періоди підвищеної загрозливої активності.

Нижче наведено географічний розподіл подій безпеки, який показує, з яких регіонів надходить найбільша кількість підозрілого трафіку. Поруч розміщено

узагальнену статистику щодо загальної кількості безпекових інцидентів та джерел атак, що дає змогу оцінити масштаб і характер загроз.

Окремі блоки присвячені найбільш атакованим ресурсам і API-ендпоінтам, що дозволяє визначити критичні точки застосунку, які потребують підвищеної уваги з боку служби безпеки. Така аналітика допомагає не лише фіксувати інциденти, а й формувати більш точні політики захисту, спираючись на реальну картину атак і поведінку трафіку.

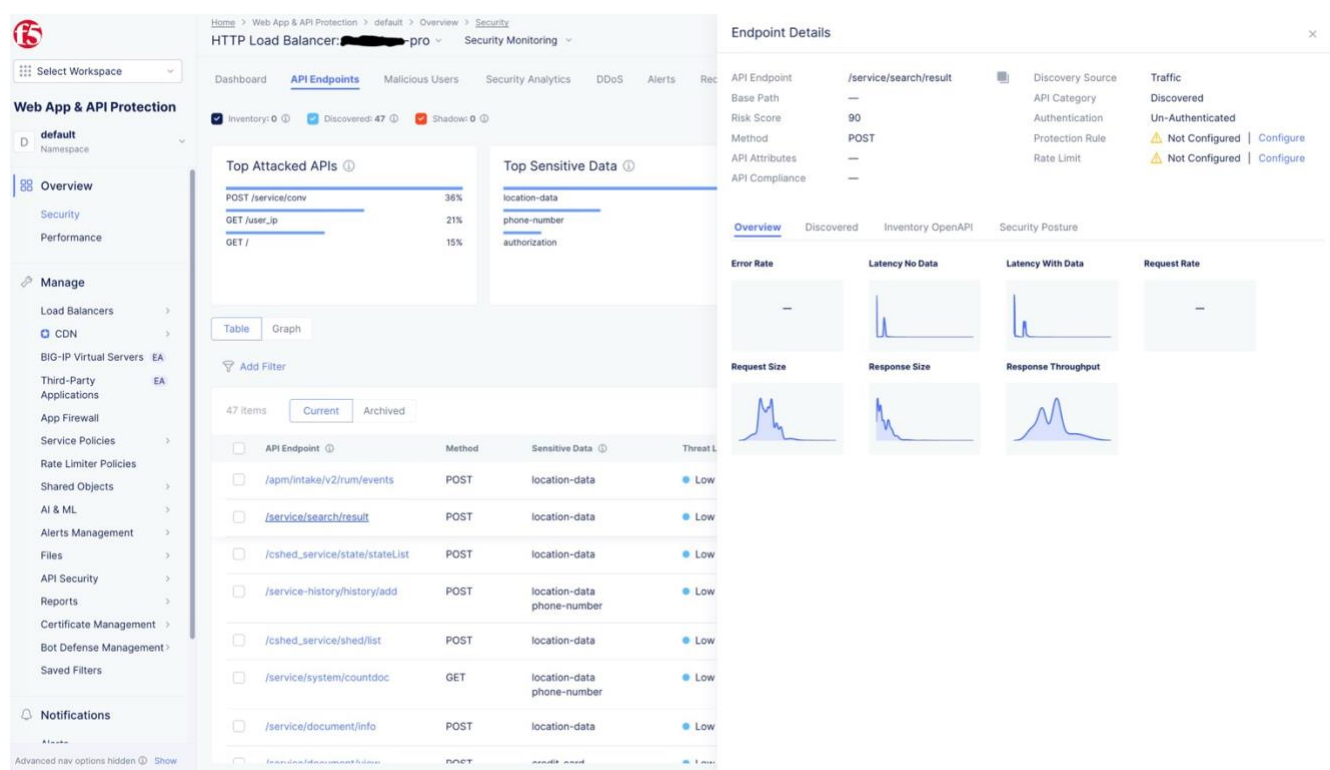


Рис 3.28. Аналіз API-ендпоінтів та чутливих даних у F5 Distributed Cloud WAAP

Розділ API Endpoints у модулі безпеки F5 Distributed Cloud демонструє детальну картину використання API в межах захищеного застосунку. У центральній частині відображено перелік виявлених API-ендпоінтів із зазначенням HTTP-методів, типів чутливих даних та рівня ризику, що дозволяє оцінити, які саме точки доступу є найбільш критичними з точки зору безпеки.

Окремі блоки зверху показують найбільш атаковані API та типи чутливих даних, які передаються через них, зокрема геолокаційні дані, номери телефонів і

параметри авторизації. Це дає змогу зрозуміти, де зосереджена основна загрозова активність і які ресурси потребують посиленого контролю.

У правій частині екрана відкрито детальну інформацію про конкретний API-ендпоінт. Тут відображено його базовий шлях, HTTP-метод, рівень ризику, статус автентифікації та наявність захисних механізмів, таких як rate limiting або політики захисту. Також наведено графіки затримок, розміру запитів і відповідей, що дозволяє оцінити не лише безпеку, а й поведінку цього API з точки зору продуктивності.

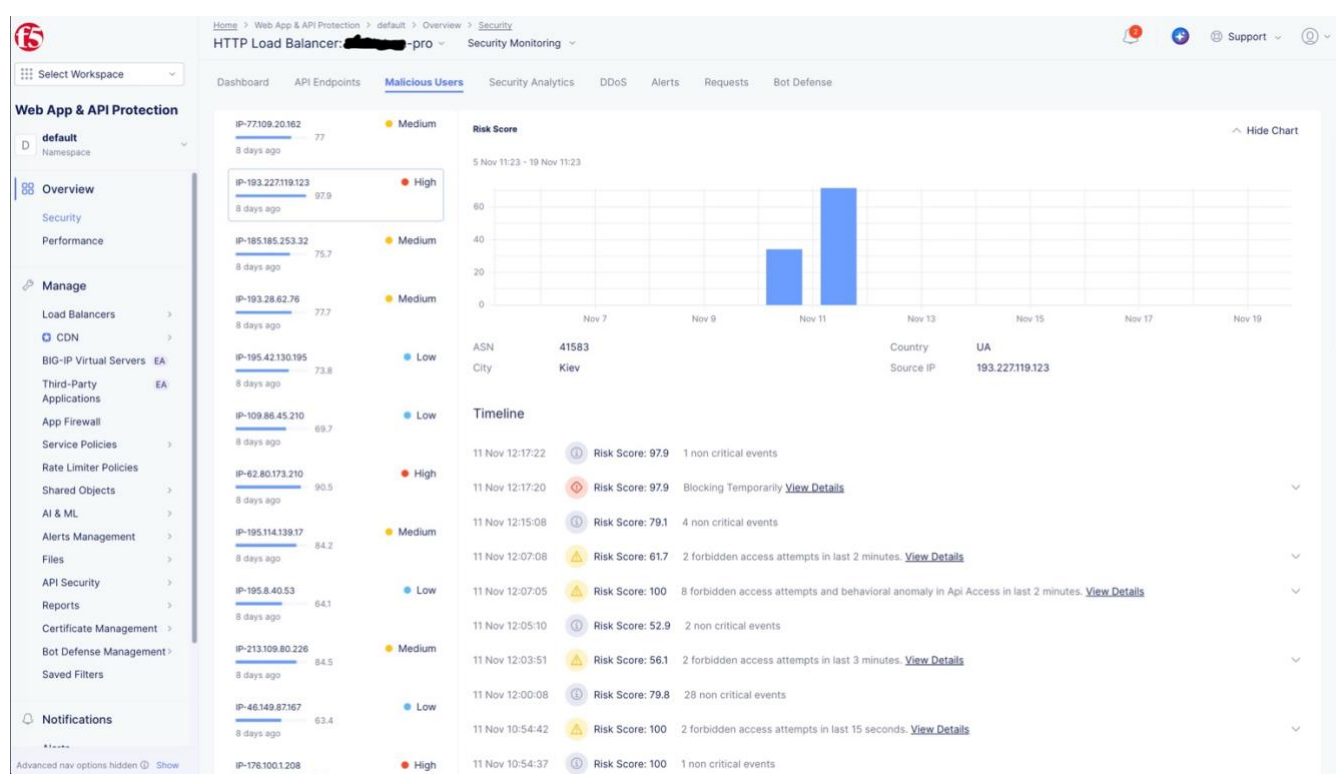


Рис 3.29. Аналіз ризикових користувачів та їхньої активності в F5 Distributed Cloud WAAP

Malicious Users відображає детальну інформацію про користувачів, чия поведінка була оцінена системою як потенційно небезпечна. У лівій частині наведено список IP-адрес із зазначенням рівня ризику та часу останньої активності, що дозволяє швидко визначити найбільш проблемні джерела трафіку.

У центральній частині показано графік зміни risk score у часі для вибраного користувача. Це дає уявлення про динаміку його поведінки та дозволяє побачити моменти, коли активність різко зростала або ставала більш агресивною.

Нижче представлено хронологію подій, де зафіксовано конкретні дії користувача, такі як заборонені спроби доступу, аномальна поведінка при роботі з API або тимчасові блокування. Кожен запис супроводжується оцінкою ризику та коротким описом причини спрацювання захисних механізмів.

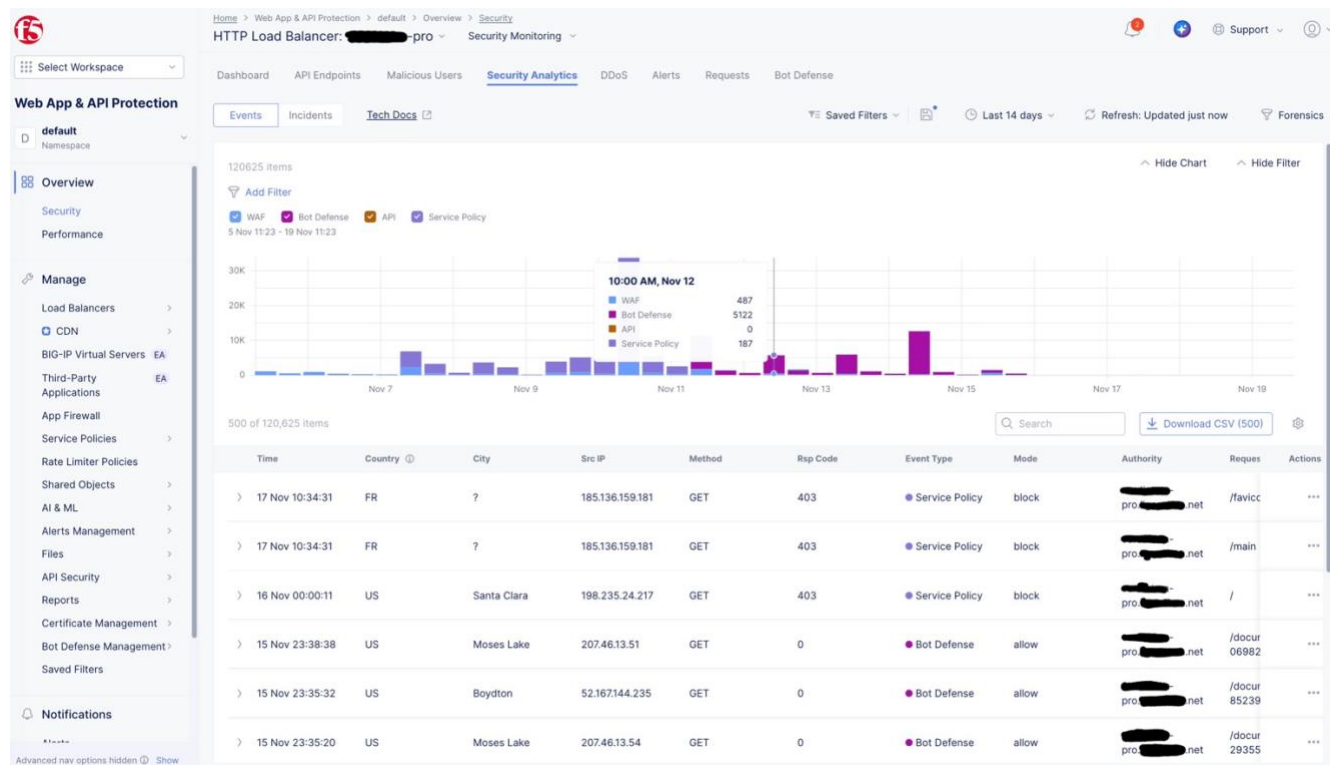


Рис 3.30 - Аналітика безпекових подій та інцидентів у F5 Distributed Cloud WAAP

Security Analytics відображає узагальнену картину безпекових подій, які були зафіксовані під час роботи HTTP Load Balancer. У верхній частині представлено часовий графік, що показує кількість подій різних типів, зокрема спрацювання WAF, бот-захисту, API та сервісних політик. Це дозволяє побачити загальну динаміку загроз і визначити періоди підвищеної активності.

Нижче розміщено таблицю з детальними записами окремих подій. Для кожного інциденту наведено час, країну та місто джерела трафіку, IP-адресу, HTTP-метод, код відповіді та тип спрацювання. Також зазначено режим роботи політики

— блокування або дозволу, що дає змогу зрозуміти, як саме система відреагувала на конкретну активність.

Відповідно ми можемо підраховувати кількість подій, аналізувати їхній контекст, відстежувати поведінку трафіку в часі та приймати рішення щодо подальшого налаштування політик безпеки на основі реальних інцидентів.

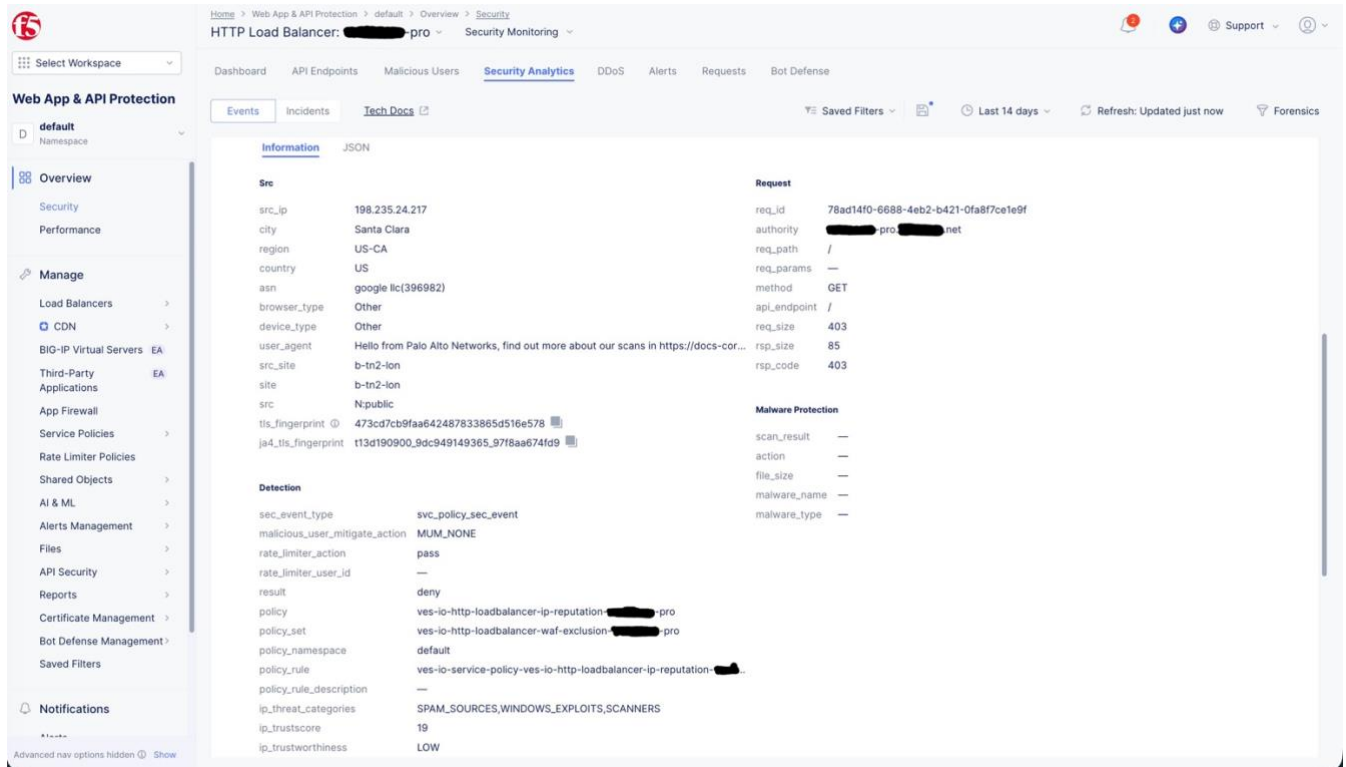


Рис 3.31. Деталізація безпекового інциденту в модулі Security Analytics F5

Distributed Cloud

Додатково у вкладці Security Analytics містить детальну інформацію про окремий безпековий інцидент, зафіксований під час обробки HTTP-запиту. У лівій частині наведені відомості про джерело трафіку, зокрема IP-адресу, географічне розташування, автономну систему та технічні ідентифікатори клієнта. Це дозволяє встановити походження підозрілої активності та оцінити рівень довіри до джерела запиту.

Праворуч відображені параметри самого запиту, включно з HTTP-методом, шляхом до ресурсу, кодом відповіді сервера та розміром переданих даних. Окремо показано результати роботи захисних механізмів, таких як політики сервісного

доступу та репутаційні перевірки, на основі яких було прийнято рішення про блокування запиту.

Нижня частина містить технічні деталі спрацювання політики безпеки, включно з типом події, результатом перевірки та категоріями загроз, до яких було віднесено дану активність.

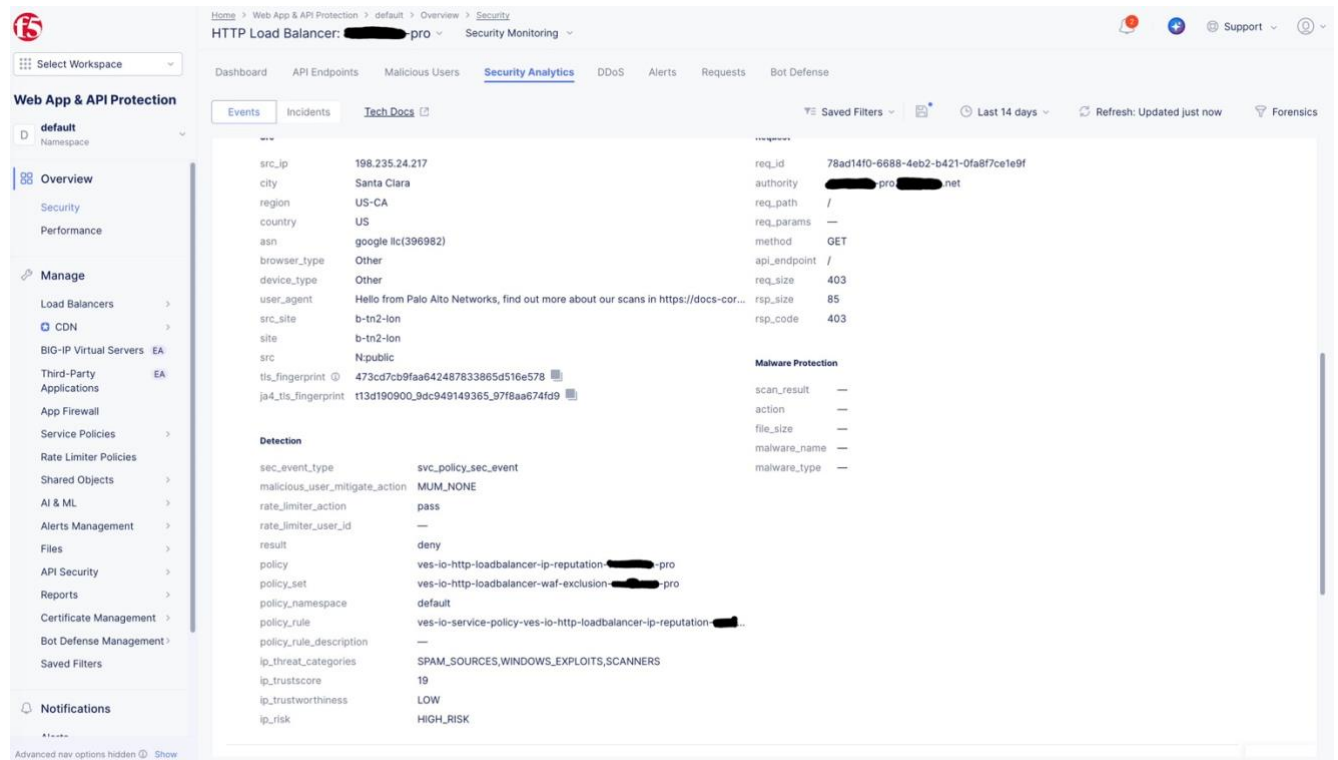


Рис 3.32. Розширена деталізація безпекової події та логіки блокування запиту у F5 Distributed Cloud WAAP

Представлений скріншот відображає поглиблену технічну інформацію про безпекову подію, зафіксовану в модулі Security Analytics під час обробки HTTP-запиту. У лівій частині зосереджені дані про джерело трафіку, включно з IP-адресою, географічним розташуванням, автономною системою та характеристиками клієнта. Це дозволяє визначити походження запиту та оцінити його репутацію з точки зору мережевої безпеки.

Праворуч наведено параметри самого запиту, зокрема HTTP-метод, шлях доступу до ресурсу та код відповіді сервера. Ці дані дають змогу зрозуміти, на якому етапі запит був оброблений і яким було фінальне рішення системи.

Окремий блок присвячений результатам спрацювання політик безпеки. Тут відображено, яка саме політика була застосована, який тип події зафіксовано та до яких категорій загроз було віднесено дану активність. Інформація про рівень ризику та довіри до джерела запиту показує, що рішення про блокування приймалося на основі комплексної оцінки, а не одного окремого правила.

The screenshot shows the F5 Distributed Cloud Security Analytics interface. The main view is titled "Security Analytics" and displays a table of events. The table has columns for "Time", "Location", "IP", "Method", "Risk", "Action", and "Verdict". The first event is highlighted, showing a "HIGH_RISK" event from "Moses Lake" at "207.46.13.51" on "15 Nov 23:38:38". The event details are expanded, showing "Request" information (req_id, authority, req_path, req_params, method, api_endpoint, req_size, rsp_size, rsp_code) and "Detection" information (sec_event_type, insight, enforce_recommendation, automation_type). The detection information indicates a "bot_defense_sec_event" with a "GOODBOT" insight and an "Action_alert" recommendation.

Рис 3.33. Деталізація подій Bot Defense у модулі Security Analytics F5 Distributed Cloud

Цей скріншот відображає аналіз подій, пов'язаних із роботою механізмів захисту від ботів у межах HTTP Load Balancer. Основна увага тут зосереджена на ідентифікації автоматизованого трафіку та прийнятті рішень щодо його обробки. У верхній частині показано перелік подій із зазначенням рівня ризику, типу захисного механізму та дії, яку застосувала система.

Розкритий запис демонструє детальну інформацію про конкретний запит, включно з IP-адресою, географічним розташуванням, типом клієнта та user-agent. Окремо видно, що система класифікувала джерело як автоматизований клієнт, але з ознаками легітимної поведінки, у зв'язку з чим запит був дозволений. Таке

рішення ґрунтується не на простому блокуванні за ознакою «бот», а на аналізі контексту та поведінки.

А нижня частина містить хронологію подібних подій, що дозволяє відстежити повторюваність активності та оцінити її стабільність у часі. Це дає змогу зрозуміти, які автоматизовані запити є допустимими для роботи сервісу, а які можуть становити загрозу.

The screenshot shows the F5 Distributed Cloud Security Analytics interface. The top navigation bar includes links for Home, Web App & API Protection, default, Overview, Security, HTTP Load Balancer, and Security Monitoring. The left sidebar contains a 'Web App & API Protection' section with sub-panels for Overview, Security, Performance, Manage, and Notifications. The main content area is titled 'Security Analytics' and shows a table of events. The selected event is from 15 Nov 23:09:42, originating from UA (Kiev) with IP 134.238.95.182, using a GET method to access a document. The event is classified as 'Bot Defense' with an 'allow' action. Below the table, the 'Information' tab shows details about the request, including the source IP, user agent, and request parameters. The 'Detection' tab shows the event type as 'bot_defense_sec_event' with an insight of 'MALICIOUS' and an enforcement recommendation of 'Action_alert'.

Рис 3.34. Аналіз подій Bot Defense та класифікація автоматизованого трафіку у F5 Distributed Cloud

Далі ми бачимо під меню, яке відображає події, зафіксовані механізмами захисту від ботів у межах модуля Security Analytics. У списку подій наведено запити, які були ідентифіковані як автоматизовані, із зазначенням їх джерела, географічного розташування та застосованої дії з боку системи. Це дозволяє оцінити характер бот-трафіку та його вплив на роботу вебзастосунку.

Розгорнутий запис демонструє детальну інформацію про конкретний запит, включно з IP-адресою, типом клієнта, user-agent та параметрами HTTP-запиту. У блоці детекції видно, що система класифікувала запит як шкідливий, але замість

негайного блокування застосувала м'яку реакцію у вигляді сповіщення. Такий підхід дозволяє зберегти контроль над ситуацією та уникнути помилкових блокувань легітимного трафіку.

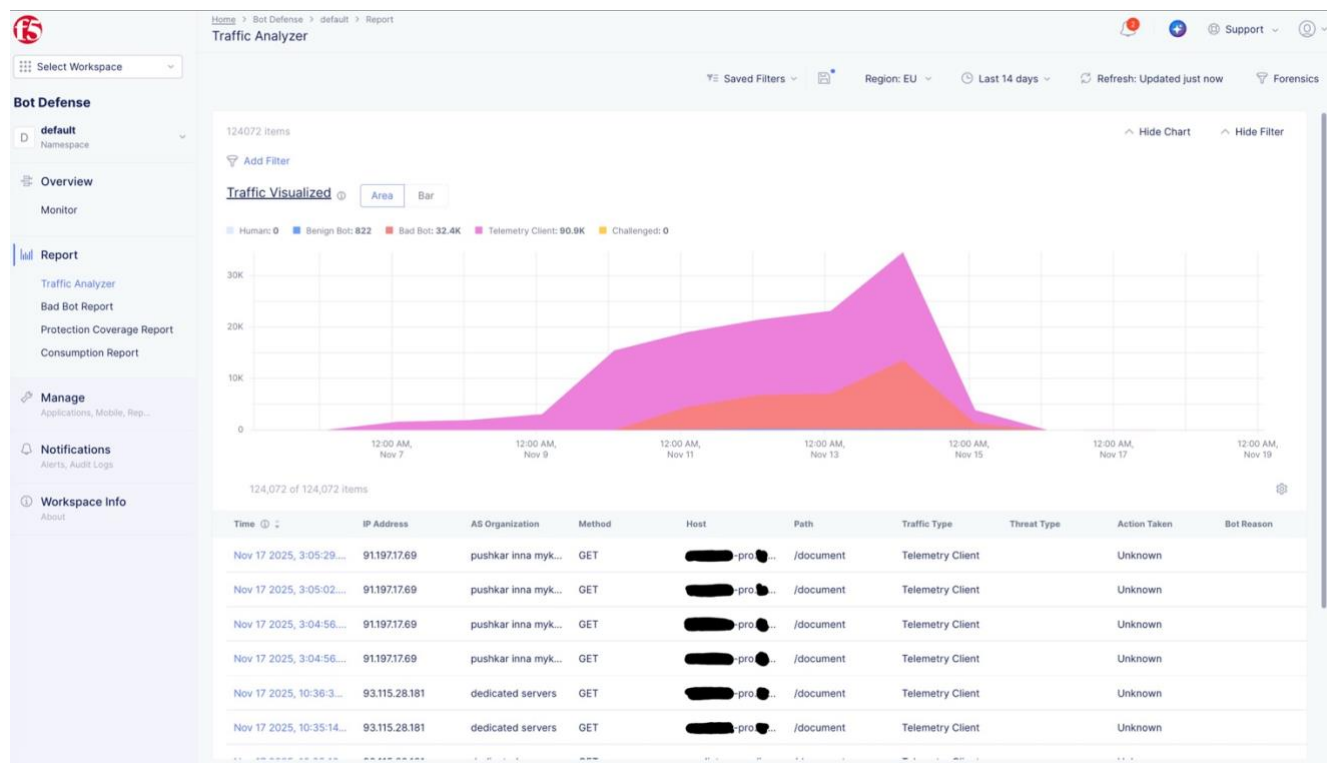


Рис 3.35. Звіт Bot Defense: динаміка та склад автоматизованого трафіку

Зображення дає зрозуміти, яку роль у загальному потоці запитів відіграє автоматизований трафік. Основний акцент тут зроблено не на окремих інцидентах, а на загальній картині: як змінюється інтенсивність ботів у часі та які саме типи автоматизованих клієнтів створюють навантаження на застосунок. Видно, що значну частину складає телеметричний трафік, який формально не є шкідливим, але все одно впливає на ресурси системи. Графік дозволяє швидко помітити пікові періоди активності та співвіднести їх із роботою сервісу або зовнішніми подіями. Це важливо для розуміння, чи є зростання трафіку наслідком реальної користувацької активності, чи результатом роботи автоматизованих клієнтів.

Нижче наведено перелік конкретних запитів, де вже видно технічні деталі: звідки надходять звернення, які ресурси вони запитують і як система їх класифікує. У сукупності така подача інформації дозволяє дивитися на бот-захист не як на

механізм блокувати або дозволяти, а як на інструмент аналізу, що допомагає зрозуміти реальну поведінку трафіку та приймати зважені рішення щодо його контролю.

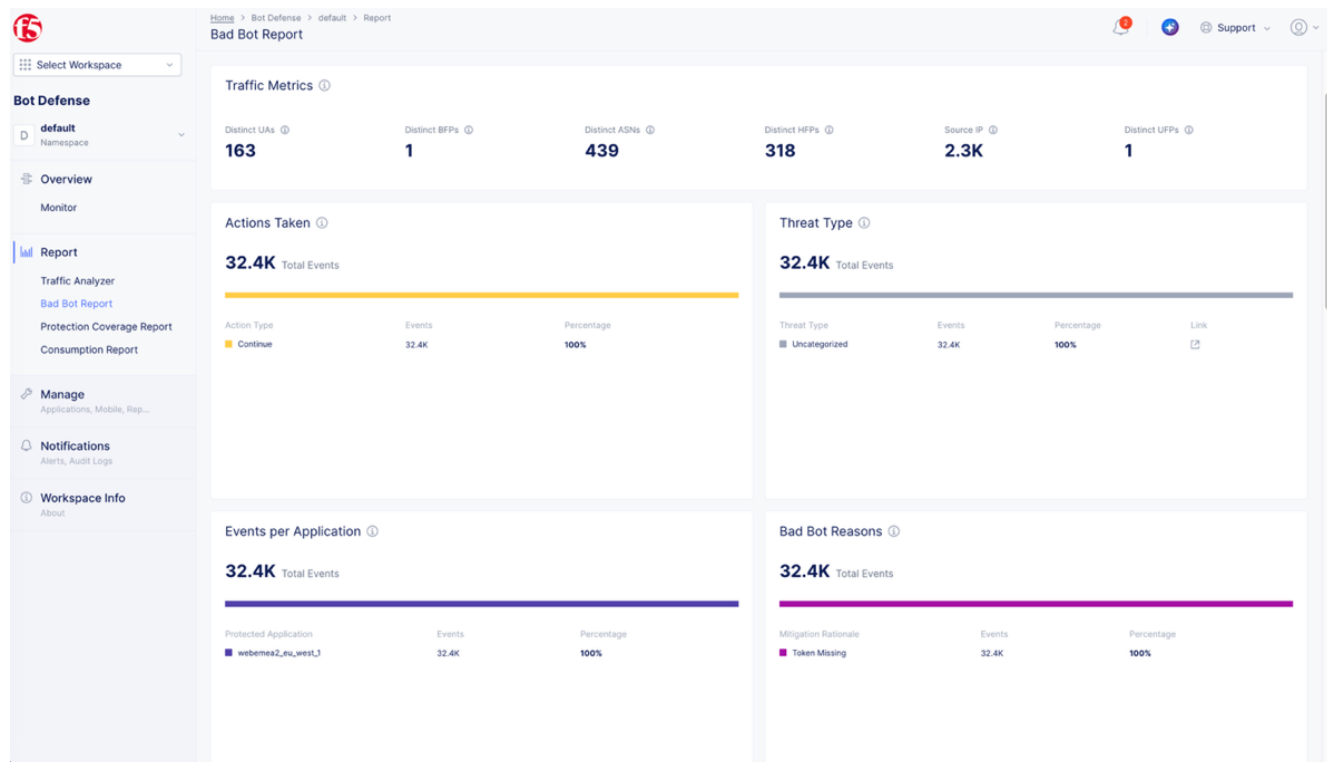


Рис 3.36. Зведений звіт про активність шкідливих ботів у модулі Bot Defense F5 Distributed Cloud

Цей розділ використовується для підбиття підсумків роботи механізмів Bot Defense за вибраний проміжок часу. Він не показує окремі запити чи інциденти, а дає загальне уявлення про те, з якою кількістю автоматизованого трафіку зіткнулася система і як вона його класифікувала.

У верхній частині зведено основні статистичні показники: кількість унікальних джерел, user-agentів та автономних систем, що дозволяє оцінити масштаб і розподіленість бот-активності. Далі наведено інформацію про дії, які застосовувалися до такого трафіку, та типи загроз, до яких він був віднесений. Це допомагає зрозуміти, чи потребує ситуація більш жорстких політик захисту або достатньо спостереження.

Окремий блок присвячений причинам, через які запити були класифіковані як небажані. У цьому випадку основним фактором стала відсутність токена, що є

типовою ознакою автоматизованих клієнтів. Також показано, до якого застосунку була спрямована вся зафіксована активність, що дозволяє швидко локалізувати зону ризику.

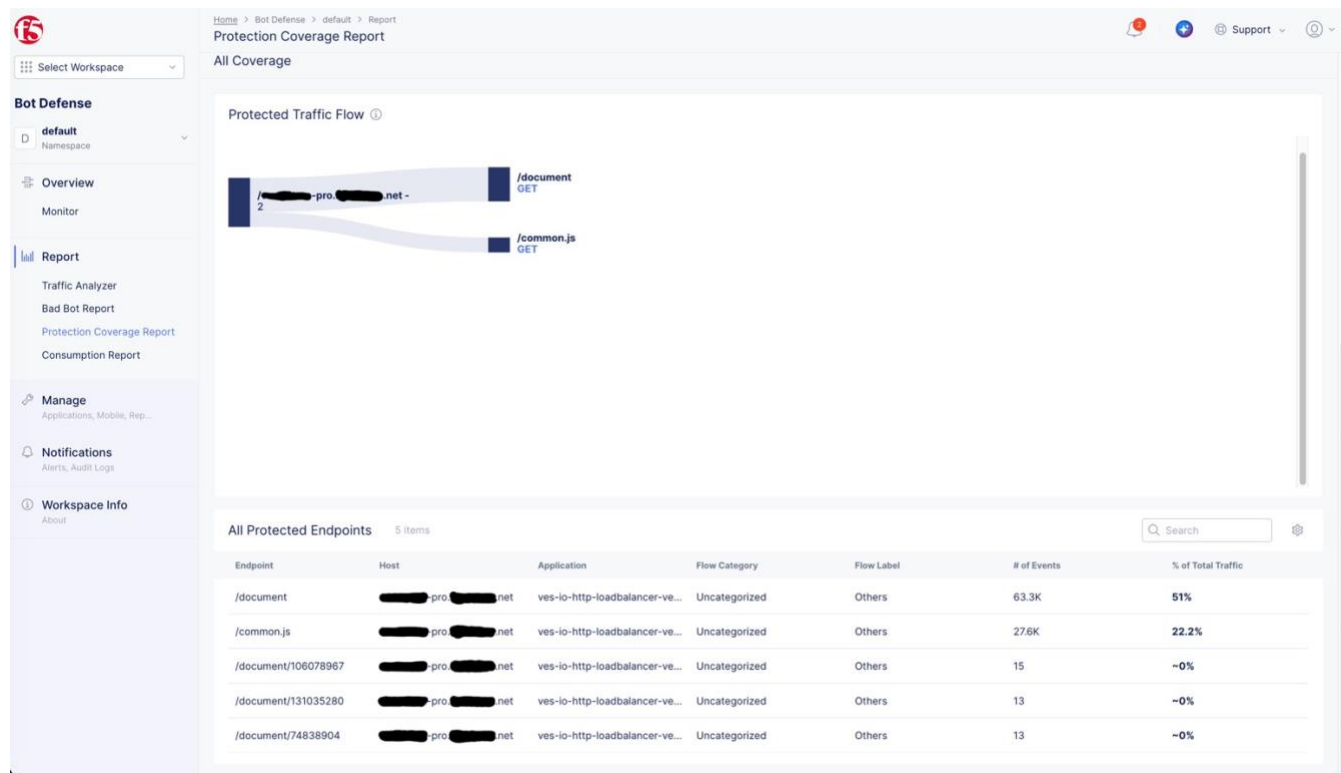


Рис 3.37. Звіт покриття захистом трафіку в модулі Bot Defense F5 Distributed Cloud

Цей розділ призначений для оцінки того, який саме трафік фактично проходить через механізми захисту від ботів і на які ресурси він спрямований. У верхній частині показано схему потоків трафіку, яка наочно демонструє, до яких endpoint-ів звертаються клієнти та які HTTP-запити обробляються системою захисту. Такий вигляд дозволяє швидко зрозуміти, які ресурси застосунку є найбільш активними з точки зору автоматизованих звернень.

Нижче наведено таблицю з переліком захищених endpoint-ів, де для кожного з них вказано домен, застосунок, кількість подій та частку від загального трафіку. Видно, що основне навантаження зосереджене на обмеженій кількості ресурсів, тоді як інші endpoint-и використовуються значно рідше. Це дає змогу визначити, на яких шляхах доцільно посилювати політики Bot Defense, а де достатньо базового контролю. Це допомагає переконатися, що критичні точки доступу не залишаються

поза увагою системи безпеки, а налаштування Bot Defense відповідають реальній картині використання застосунку.

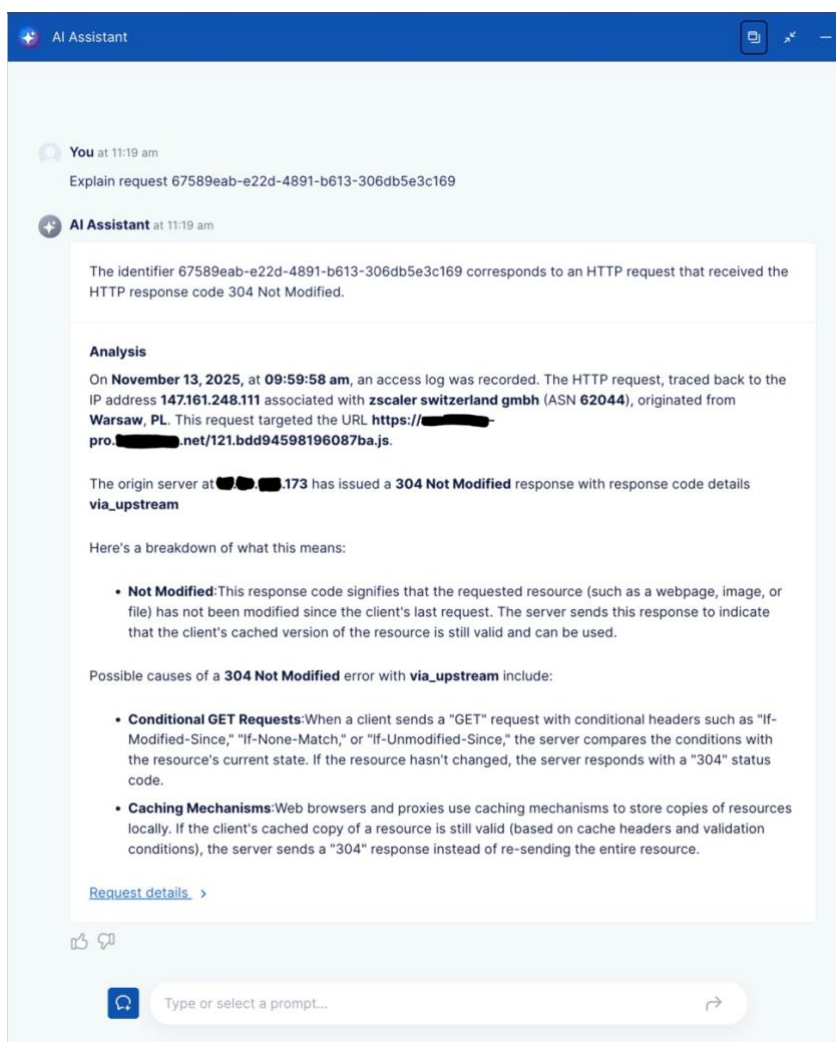


Рис 3.38. AI-асистент для автоматичного аналізу та пояснення HTTP-запитів у F5 Distributed Cloud

Важливо ще підкреслити наявність вбудованого штучного інтелекту для аналізу HTTP-запитів і подій безпеки, який дозволяє автоматично пояснювати поведінку трафіку без ручного розбору логів. На наведеному прикладі показано роботу AI-асистента, який на основі конкретного ідентифікатора запиту формує зрозуміле пояснення того, що відбулося під час обробки HTTP-звернення. Система самостійно визначає час події, джерело трафіку, IP-адресу, автономну систему, географічне походження та цільовий ресурс, після чого інтерпретує отриманий HTTP-статус код. У цьому випадку штучний інтелект пояснює відповідь

304 Not Modified, пов'язуючи її з механізмами кешування та умовними HTTP-запитами, що є типовою поведінкою для статичних ресурсів.

Такий підхід суттєво спрощує роботу спеціалістів з кібербезпеки, оскільки замість аналізу сирих технічних даних вони одразу отримують контекст і причину події у зрозумілій формі. Це знижує час реагування, мінімізує людський фактор і дозволяє швидше відрізнити нормальну роботу застосунку від потенційних інцидентів безпеки.

Усі наведені зображення разом формують цілісну картину того, як на практиці виглядає робота з платформою WAAP на базі F5 Distributed Cloud у реальному виробничому середовищі. Вони не показують абстрактну теорію безпеки, а демонструють живу систему з трафіком, подіями, інцидентами, бот-активністю, API-ендпоїнтами та реальними рішеннями, які приймаються системою щосекунди. Саме в цьому й полягає цінність наведеного матеріалу — він ілюструє не концепцію, а щоденну роботу платформи захисту.

Загальний наратив починається з базових речей — моніторингу стану застосунку, продуктивності та здоров'я балансувальників. Далі поступово відкривається глибина аналізу: від агрегованих графіків трафіку до розбору кожного окремого HTTP-запиту, його маршруту, затримок, політик безпеки та фінального рішення системи. Окремі скріншоти показують, як WAAP працює з API — автоматично виявляє ендпоїнти, визначає чутливі дані та оцінює ризики, не вимагаючи повної ручної інвентаризації.

Значна частина матеріалів присвячена роботі з бот-трафіком. Видно, що платформа не використовує примітивний підхід бот = блокувати, а намагається зрозуміти природу автоматизованої активності: де це легітимна телеметрія, де пошукові боти, а де спроби зловживання. Звіти Bot Defense і аналітика ризикових користувачів показують, як система накопичує контекст, оцінює поведінку в часі та дозволяє ухвалювати зважені рішення без шкоди для нормальної роботи сервісу.

Окрему увагу важливо приділити блоку Security Analytics, де всі події зводяться в єдину часову картину. Саме тут видно, як різні механізми — WAF, Bot Defense, API-захист, сервісні політики — працюють разом, а не ізольовано.

Деталізація окремих інцидентів демонструє прозору логіку прийняття рішень: від ідентифікації джерела до фінального дозволу або блокування.

Частина представлених зображень навмисно замальована або приховує окремі елементи інтерфейсу. Це пов'язано з тим, що матеріал базується на реальному прикладі системи замовника, використаному в навчальних і дослідницьких цілях. З етичних та юридичних причин конкретні домени, IP-адреси та ідентифікатори не можуть бути розголошені, однак це жодним чином не знижує технічну цінність прикладу.

3.4 Рекомендації щодо застосування технології захисту безпеки додатків WAAP

Використання технології WAAP для захисту веб-застосунків і API варто розглядати не як черговий інженерний інструмент, а як комплексний підхід до побудови стійкості цифрової інфраструктури. Сучасні додатки працюють у динамічному середовищі, де поєднуються хмари, локальні середовища, мікросервіси та зовнішні інтеграції. І що важливіше, що сучасні додатки розвиваються дуже швидко. У таких умовах політика безпеки перестає бути набором правил і перетворюється на постійний процес, що розвивається разом із самим застосунком. Саме з цієї причини впровадження WAAP має починатися з усвідомлення контексту: які саме сервіси існують, як вони взаємодіють, які дані обробляють і які ризики супроводжують кожну точку входу та кожен API-метод.

Практика показує, що найбільшу цінність WAAP приносить тоді, коли дозволяє отримати повну видимість того, що раніше залишалось сірою зоною — приховані ендпоінти, неконтрольовані API, неочевидні взаємозв'язки всередині системи. Це перший крок, без якого неможливо побудувати ефективний захист. Після отримання такої прозорості організації мають змогу змінити сам підхід до безпеки: не реагувати на інциденти, а запобігати їм, покладаючись на поведінкову аналітику та автоматичне виявлення аномалій. WAAP добре працює там, де класичні інструменти безсилі, а це у складних взаємодіях даних, у логіці між

сервісами, у тих місцях, де атака може не мати сигнатури, але змінює характер трафіку чи поведінку користувачів.

Окрема увага повинна приділятися API. У більшості сучасних архітектур API містять найбільш критичну бізнес-логіку та відкривають доступ до найважливіших даних. Тому організації, які впроваджують WAAP, мають не просто включити захист, а зробити його частиною життєвого циклу розробки. Оптимальний підхід — коли WAAP відстежує зміни у схемах, попереджає про дрейф API та допомагає автоматично створювати політики, що відповідають реальній поведінці застосунку. Це знижує ризики помилок, які з'являються під час швидких релізів, і дозволяє команді розробки працювати без постійного страху «зламати щось у продакшені».

Ще один важливий аспект — інтеграція WAAP у розподілене середовище. Сучасні застосунки працюють у хмарах, між різними дата-центрами, іноді в країнах із різними регуляторними вимогами. WAAP допомагає вирівняти політику безпеки, зробити її єдиною незалежно від того, де саме розміщений конкретний сервіс. Це не лише впорядковує архітектуру, а й помітно зменшує кількість точок відмови, адже той самий рівень захисту застосовується на edge-локаціях, у приватних середовищах і всередині Kubernetes-кластерів. Поступово платформа стає не додатковим компонентом, а частиною самої інфраструктури, що автоматично підлаштовується під її ритм.

Перевага WAAP стає очевидною також у роботі з бот-трафіком. У багатьох компаній ця проблема довго залишається непомітною, поки кількість ботів не починає впливати на продуктивність або спотворювати бізнес-метрики. WAAP, використовуючи поведінкові ознаки, може відрізнити реального користувача від автоматизованого скрипта ще до того, як бот встигає створити навантаження чи зібрати конфіденційні дані. Це особливо важливо для застосунків, які працюють із автентифікацією, наприклад банківські сервіси, мобільні додатки чи портали з погодженими транзакціями.

Важливо розуміти: ефективність WAAP не залежить лише від інструменту. Вона залежить від того, наскільки організація готова змінити підхід до безпеки. Успішні впровадження показують, що WAAP найбільш корисний там, де бізнес

готовий інтегрувати безпеку в сам продукт, а не розглядати її як зовнішній прошарок. Це означає регулярний перегляд політик, роботу з аналітикою, адаптацію правил під поведінку користувачів та активний діалог між командами розробки, експлуатації та безпеки. Коли всі ці елементи працюють разом, WAAP стає не просто інструментом, а системою, яка підсилює стійкість та розвиток застосунків.

Якщо підсумувати, рекомендації для компаній, що запроваджують WAAP, можна звести до простого принципу: будувати безпеку як частину архітектури, а не як компенсацію її слабких місць. Це означає бачити додаток цілісно — від edge-трафіку до внутрішніх API, від поведінки користувачів до взаємодії мікросервісів. WAAP дозволяє це зробити, але повний ефект з'являється лише тоді, коли технологію застосовують не формально, а як інструмент розвитку й захисту бізнесу. У такому підході безпека перестає бути бар'єром і стає тим, що підтримує зростання, стабільність і довіру користувачів.

ВИСНОВКИ

Під час виконання даної кваліфікаційної роботи було підтверджено, що сфера кібербезпеки перебуває в стані постійної та стрімкої трансформації, а вартість помилки зростає непропорційно швидше, ніж складність самих інформаційних систем. Сучасні кіберзагрози перестали бути абстрактним поняттям або теоретичним ризиком — сьогодні вони є повсякденною реальністю для організацій будь-якого масштабу, від малого бізнесу до об'єктів критичної інфраструктури держави. Особливо гостро це усвідомлюється в умовах, у яких перебуває Україна, де кіберпростір став повноцінним фронтом протистояння поряд із фізичним.

У межах роботи було показано, що класичні підходи до захисту вебдодатків, зокрема традиційні Web Application Firewall, не відповідають сучасним вимогам розвитку цифрових систем. Такі рішення створювалися для монолітних вебресурсів і статичних архітектур, тоді як сучасні застосунки базуються на мікросервісах, розподілених хмарних середовищах та API, які фактично є носіями бізнес-логіки і даних. У цих умовах просте фільтрування HTTP-запитів є недостатнім, а ефективний захист потребує поведінкової аналітики, контекстного аналізу доступу, розуміння логіки взаємодії сервісів і повної прозорості внутрішніх потоків трафіку.

Технологія WAAP, а зокрема її реалізація у вигляді платформи F5 Distributed Cloud, комплексно відповідає на ці виклики. Там, де класичний WAF не має повної видимості API, WAAP оперує схемами, методами, аномаліями та структурою запитів. У середовищі, де традиційний периметр безпеки фактично зникає, distributed cloud-архітектура формує нову модель захисту, яка не прив'язана до фізичного розміщення ресурсів. Водночас edge-архітектура дозволяє не лише не погіршувати продуктивність застосунків, а й у багатьох випадках покращувати її. Зі зростанням складності атак машинне навчання та автоматизовані механізми реагування забезпечують швидкість і точність, недосяжні для ручного аналізу.

Окремо слід відзначити концепцію Customer Edge (CE), яка дозволяє інтегрувати елементи хмарної платформи безпосередньо в інфраструктуру замовника. Такий підхід фактично стирає межі між публічною хмарою, приватним дата-центром і локальними середовищами, забезпечуючи єдину політику безпеки незалежно від місця розміщення сервісів. Це створює уніфікований простір управління ризиками, де правила залишаються однаковими, а контроль — централізованим.

Основний висновок, отриманий у ході дослідження, полягає в тому, що сучасна кібербезпека більше не є статичним набором інструментів або одноразовим впровадженням захисного рішення. Вона повинна розглядатися як жива, адаптивна система, що розвивається разом із застосунком і бізнесом. WAAP зміщує акцент із простого блокування запитів на глибоке розуміння трафіку, поведінки користувачів і потенційних точок виникнення ризиків. Такий підхід органічно інтегрується в сучасні DevOps-процеси й підтримує швидкий розвиток цифрових продуктів, а не протидіє йому.

У ширшому сенсі виконаний проєкт дозволяє розглядати кібербезпеку не лише як технічну дисципліну, а як філософію проактивності, відповідальності та усвідомленого управління ризиками. В умовах, коли інформація стала одним із ключових ресурсів, а атаки можуть бути непомітними, але руйнівними за наслідками, безпека перестає бути витратою — вона стає інвестицією у стабільність і майбутнє.

Традиційне запитання про вартість ефективної кібербезпеки в цьому контексті поступається іншому, значно важливішому: якою є ціна її відсутності та чи готові організації сплачувати саме її.

ПЕРЕЛІК ПОСИЛАНЬ

1. A Complete Guide to Cybersecurity. UpGuard. URL: <https://content.upguard.com/hubfs/resources/eBook%20%20A%20Complete%20Guide%20to%20Cybersecurity.pdf> (дата звернення: 22.11.2025).
2. F5 Networks — огляд вендора та його рішень. Techsvit. URL: <https://techsvit.ua/ua/vendors/f5/> (дата звернення: 24.11.2025).
3. What Is PII? A Complete Guide to Personally Identifiable Information. CoreWin. URL: <https://corewin.ua/blog/personally-identifiable-information-guide/> (дата звернення: 26.11.2025).
4. What Is WAAP? Web Application & API Protection Explained. IORiver. URL: <https://www.ioriver.io/blog/waap> (дата звернення: 26.11.2025).
5. What Is WAAP? F5 Community. URL: <https://community.f5.com/kb/technicalarticles/what-is-waap/311972> (дата звернення: 29.11.2025).
6. What Is Web Application and API Protection (WAAP)? Barracuda. URL: [https://www.barracuda.com/support/glossary/waap#:~:text=Barracuda%20can%20help-,What%20is%20web%20application%20and%20API%20Protection%20\(WAAP\)?,to%20web%20and%20API%20applications](https://www.barracuda.com/support/glossary/waap#:~:text=Barracuda%20can%20help-,What%20is%20web%20application%20and%20API%20Protection%20(WAAP)?,to%20web%20and%20API%20applications) (дата звернення: 29.11.2025).
7. Web App & API Protection Documentation. F5 Distributed Cloud Docs. URL: <https://docs.cloud.f5.com/docs-v2/web-app-and-api-protection> (дата звернення: 02.12.2025).
8. Protect an Application Exposed on Internet with F5 XC WAAP. F5 Community. URL: <https://community.f5.com/kb/technicalarticles/protect-an-application-exposed-on-internet-with-f5-xc-waap/300884> (дата звернення: 02.12.2025).
9. F5 Distributed Cloud WAAP Overview. F5. URL: <https://cdn.studio.f5.com/files/k6fem79d/production/707d70ce4b6c3a3b577cb3d0d04bbf5899389158.pdf> (дата звернення: 04.12.2025).
10. Cybersecurity Update 11. Nomios. URL: <https://www.nomios.co.uk/news-blog/cybersecurity-update-11/> (дата звернення: 04.12.2025).
11. Web Application Security Guide. NordLayer. URL: <https://nordlayer.com/blog/web-application-security/> (дата звернення: 06.12.2025).

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація)