ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЗАХИСТУ ІНФОРМАЦІЇ

КАФЕДРА ІНФОРМАЦІЙНОЇ ТА КІБЕРНЕТИЧНОЇ БЕЗПЕКИ

Пояснювальна записка

до магістерської роботи
на тему:

«ТЕХНОЛОГІЯ ЗАХИСТУ ІНТЕРНЕТ-ДОДАТКІВ ЗА ДОПОМОГОЮ AWS WEB APPLICATION FIREWALL»

Виконав студент 6 курсу, групи БСДМ-61
спеціальності 125 Кібербезпека
освітньо-професійної програми «Інформаційна та кібернетична безпека»
(шифр і назва спеціальності)
Кучер В. І.
(прізвище та ініціали)
Керівник Власенко В. О.
(прізвище та ініціали)
Рецензент
(прізвище та ініціали)
Нормоконтролер Чумак Н. С.
(прізвище та ініціали)

КИЇВ – 2021

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ОС – операційна система

ПК – персональний комп’ютер

ACL – Access Control List

API – Application Programming Interface

AWS – Amazon Web Services

JSON – JavaScript Object Notation

REST – Representational State Transfer

OWASP – Open Web Application Security Project

SQL – Structured Query Language

WAF – Web Application Firewall

ВСТУП

Актуальність дослідження. Сьогодні в сучасних організаціях широко застосовуються веб-додатки, які входять до складу їх інформаційних систем. Вразливості веб-додатків експлуатуються зловмисниками для досягнення різних злочинних цілей, що ще більш загострює проблему забезпечення кібербезпеки інформаційних систем організацій.

Сучасні архітектури розгортання веб-додатків досить складні та потребують інтеграції багатьох гетерогенних технологій, створюючи потенціал для численних уразливостей. Прискорені цикли розробки та постійні оновлення веб-додатків ще більше загострюють ситуацію. У таких умовах бізнес-ризики організацій занадто великі, щоб ігнорувати цю проблему, оскільки вразливості у веб-додатках наражають небезпеку на критично важливі бізнес-операції та конфіденційні дані. Значні фінансові втрати можуть виникнути внаслідок непередбачених затримок у бізнес-процесах, крадіжки інтелектуальної власності та втрати довіри клієнтів, а також репутації бренду.

Захист вразливих веб-додатків може здійснюватися або шляхом усунення вразливостей у них або шляхом застосування спеціалізованих засобів захисту веб-додатків – Web Application Firewall (WAF). Застосування технології WAF на сьогоднішній день є найбільш гнучким та точним інструментом для всебічного захисту від веб-загроз.

Amazon Web Services (AWS) є глобальною хмарної інфраструктурою, яка пропонує понад 200 повнофункціональних сервісів в ЦОД по всьому світу, є найбезпечнішою, масштабною та надійною хмарною платформою. За допомогою сервісів AWS можна створити хмарну інфраструктуру та швидко запустити веб-додаток з доступом по всьому світу і легко управляти робочими навантаженнями або розгорнути додаток ближче до кінцевих користувачів, щоб забезпечити затримку не більше декількох мілісекунд.

AWS довіряють мільйони активних клієнтів по всьому світу, в тому числі найдинамічніші стартапи, найбільші підприємства та урядові установи.

Вищесказане визначає актуальність теми даної магістерської роботи, основний зміст якої становлять дослідження щодо технології захисту веб-додатків організації на прикладі AWS Web Application Firewall.

Об'єкт дослідження – процес забезпечення захисту веб-додатків організації.

Предмет дослідження – технологія захисту веб-додатків організації.

Мета роботи – розробити порядок застосування технології захисту веб-додатків організації та рекомендації щодо його реалізації.

Наукові завдання:

дослідити сутність проблеми забезпечення захисту веб-додатків організації;

встановити сутність завдань захисту веб-додатків організації;

проаналізувати існуючі технології захисту веб-додатків організації;

проаналізувати методи та засоби забезпечення захисту веб-додатків організації;

проаналізувати основні функції та принципи реалізації захисту веб-додатків організації.

Методи дослідження – опрацювання літератури за даною темою, аналіз експлуатаційної документації, міжнародних стандартів та їх порівняння.

Практичне значення одержаних результатів: запропоновано порядок застосування технології захисту веб-додатків організації із застосуванням рішення AWS Web Application Firewall, а також розроблено рекомендації фахівцям з кібербезпеки щодо застосування технології захисту веб-додатків організації.

Результати магістерської роботи апробовані на Всеукраїнській науковій конференція «Актуальні проблеми кібербезпеки», яка відбулася 27 жовтня 2021 року в Державному університеті телекомунікацій, м. Київ.

1 АНАЛІЗ ПРОБЛЕМИ ЗАБЕЗПЕЧЕННЯ КІБЕРБЕЗПЕКИ ВЕБ-ДОДАТКІВ ОРГАНІЗАЦІЇ

1.1. Роль та місце веб-додатків в сучасному бізнесі

Сучасний бізнес широко застосовує “діджитал продукти”, наприклад, освітній сайт або інтернет-магазин [4]. Один із варіантів розробки такого продукту – створення веб-додатку. Це повноцінна програма, яку клієнти використовують через браузер.

Веб-додаток – це повноцінна програма, доступ до якої користувач отримує через Інтернет, тобто вона не потребує встановлення пристрою. Веб-додаток інтерактивний і дозволяє користувачам взаємодіяти з різними елементами: наприклад, залишити заявку на покупку товару, оформити покупку авіаквитка або прокоментувати пост друга [4].

Веб-додатки створюються та застосовуються в сучасних організаціях практично в будь-якій сфері. Ось кілька ідей веб-додатків [4]:

соціальні мережі;

ігри;

освітні продукти;

системи бронювання квитків та готелів;

онлайн-магазини;

фінансові рішення;

веб-версії програмного забезпечення тощо.

Розглянемо види веб-додатків. Веб-додатки можна класифікувати по-різному: залежно від їхнього функціоналу та призначення. Давайте докладніше розберемо ці типи додатків, щоб краще розуміти, як вони працюють і які більш підходять для конкретних бізнес-завдань.

Є три основні шаблони побудови сайтів [4]:

MPA (Multi-Page Application): багатосторінковий додаток, який надсилає

запит на сервер і повністю оновлює сторінку, коли з нею відбувається дія;

SPA (single-page application): односторінковий додаток, що містить HTML-сторінку, яка динамічно оновлюється залежно від дій користувача без повного перезавантаження;

PWA (progressive web application): додаток, який користувач встановлює та може використовувати у режимі офлайн.

Інша класифікація полягає в призначенні веб-додатків. Ось найпопулярніші види додатків для бізнесу [4]:

корпоративні портали – завдяки ним можна автоматизувати багато бізнес-процесів за допомогою одного продукту. Корпоративний портал дозволяє працювати з документами, відстежувати роботу співробітників, спілкуватися з контрагентами, проводити PR-заходи, пов'язувати підрозділи компанії тощо;

CRM (Customer Relationship Management) – дозволяє налаштувати вирву продажів, керувати взаєминами з клієнтами, утримувати клієнтську базу та скоротити документообіг;

ERP (Enterprise Resource Planning) – ERP-система забезпечує нові можливості: стандартизувати форми звітності, контролювати процеси, покращити взаємодію відділів та інтегрувати контрагентів у робочий процес;

системи електронної комерції – надають можливість детально розповідати клієнтам про продукти, приймати заявки та, власне, продавати товари чи послуги. При цьому скорочується шлях товару до споживача та знизите витрати на здійснення угоди.

Переваги веб-додатків [4]:

економія – у ході розробки вам не доведеться створювати окремі програми для різних операційних систем – вони працюють однаково в будь-яких браузерах: Internet Explorer, Opera, Safari, Google Chrome тощо;

безпека – веб-система має єдину точку входу, тому можна централізовано налаштувати її захист. Крім того, дані користувачів зберігаються у хмарі, тому при пошкодженні жорсткого диска інформація буде збережена;

доступ із різних пристроїв – користувач може взаємодіяти з веб-додатком

через комп’ютер, смартфон, планшет тощо, за умов доступу до Інтернету;

відсутність клієнтського ПЗ – користувачам не потрібно нічого скачувати і, що важливіше, оновлювати. Ви можете міняти інтерфейс клієнта, а оновлення до останньої версії відбудеться при черговому завантаженні сторінки;

масштабованість – навіть якщо навантаження на систему збільшиться, вам не доведеться збільшувати потужність клієнтських місць. Зазвичай веб-додатки можуть обробляти більше даних лише силами апаратних ресурсів, тому вам доведеться переписувати код і змінювати архітектуру.

Розглянемо порядок розроблення веб-додатку. Для створення веб-додатків потрібні різноманітні інструменти, які допоможуть створити структуру, красиво оформити продукт і зробити його інтерактивним. Ось основні технології розробки веб-додатків: HTML, CSS, JavaScript, PHP, TypeScript, Java, SQL.

Розробка веб-додатків включає кілька етапів і може бути досить довгим і трудомістким процесом. Ось основні етапи веб-розробки [4]:

постановка цілей та завдань програми. Клієнт визначає, навіщо та який продукт йому потрібен. Це включає як основні функції, а й глобальні цілі – продати, навчити тощо;

опрацювання технічного завдання. Команда розробки щільно спілкується із замовником, щоб максимально точно визначити вимоги до фінального продукту;

прототипування. Підрядник створює прототип сайту та показує клієнту приклади майбутніх веб-сторінок. Важливо підібрати хорошого виконавця, який має достатній досвід та розуміється на актуальних технологіях веб-розробки;

створення макету дизайну веб-додатку. Дизайнери працюють над зовнішнім виглядом продукту та узгоджують результат із замовником;

розробка та верстка. Розробник повністю створює веб-сторінки, використовуючи дизайн-макети та технічне завдання. Процес поділяється на дві частини: *backend* та *frontend*. Backend-частина включає внутрішні процеси сайту, такі як синхронізація пристроїв або авторизація користувачів. Frontend-частина – це зовнішній вигляд продукту, тобто те, як кнопки реагують на натискання і як з’являються вікна, що спливають. Після цього етапу програма вже практично

готова до використання;

заповнення контентом. Підрядник робить фінальну роботу над програмою: додає потрібний текст, картинки та відео на свої місця;

тестування. Тестувальники перевіряють, що програма працює коректно і всі об'єкти відображаються належним чином.

1.2. Структура, функції та процеси веб-додатків організації

Розглянемо роботу сучасних веб-додатків. Веб-додатки працюють за принципом “клієнт-сервер” [4]. У цьому випадку клієнт-браузер зв’язується з веб-сервером через мережу. Зміст веб-додатку на пристрої користувача формується, коли він надсилає певний запит.

Залежно від типу веб-додатків принципи їх роботи можуть відрізнятися:

статичні сторінки – користувач робить запит у браузері, а веб-сервер обробляє його та відправляє у відповідь заздалегідь створену веб-сторінку. Це може бути, наприклад, матеріал новин або інші дані, які не залежать від дій користувача;

динамічні сторінки – вони, навпаки, не надсилаються безпосередньо від веб-сервера браузеру. Спочатку вони направляються на сервер додатків, де зчитується код та підбираються дані для формування сторінки. Лише після цього сторінка відправляється на веб-сервер, а потім у браузер.

В основі роботи веб-додатків застосовується багато технологій: від допоміжних бібліотек JavaScript і модулів CSS (Cascading Style Sheets) до веб-серверів і навіть операційних систем. Розуміючи роль цих взаємозв’язків різних технологій та їх загальних реалізацій у стеку додатків, стає набагато простіше швидко ідентифікувати їх і шукати неправильні конфігурації [1]. Часто сучасні веб-додатки складаються з декількох додатків, пов’язаних з REST API. Ці API-інтерфейси не мають стану і існують лише для виконання запитів від одного додатку до іншого. Це означає, що вони фактично не зберігають жодної інформації про запитувача.

Багато сучасних клієнтських (UI) додатків запускаються в браузері способами, більш схожими на традиційні настільні програми. Ці клієнтські додатки керують своїми власними циклами життєвого циклу, запитують власні дані та не вимагають перезавантаження сторінки після завершення початкового завантаження. Це не рідкість, коли автономний додаток, розгорнутий у веб-браузері, взаємодіє з великою кількістю серверів.

Розглянемо додаток для розміщення зображень, який дозволяє користувачеві входити в систему – ймовірно, у нього буде спеціалізований сервер хостингу/розповсюдження, розташований по одній URL-адресі, та окрема URL-адреса для управління базою даних та обліковими записами.

Можна з упевненістю сказати, що сучасні додатки часто насправді є комбінацією множини окремих, але взаємодіючих додатків, що працюють разом. Це можна віднести до більш чітко визначених мережевих протоколів і шаблонів архітектури API.

Основними технологіями сучасного веб-додатку є [1]:

REST API;

JSON або XML;

JavaScript;

фреймворк SPA (React, Vue, EmberJS, AngularJS);

система автентифікації та авторизації;

один або кілька веб-серверів (зазвичай на сервері Linux)

один або кілька пакетів програмного забезпечення веб-сервера (ExpressJS, Apache, NginX);

одна або кілька баз даних (MySQL, MongoDB тощо);

локальне сховище даних на клієнті (файли cookie, веб-сховище, IndexDB).

Деякі з цих технологій існували десять років тому, але було б несправедливо сказати, що вони не змінилися за цей час. Бази даних існують протягом десятиліть, але бази даних NoSQL і бази даних на стороні клієнта, безумовно, є новітніми розробками [1].

Розробка додатків JavaScript з повним стеком також була неможлива, поки

NodeJS і npm не почали швидко застосовуватися. Ландшафт для веб-додатків за останнє десятиліття або близько того змінювався настільки швидко, що багато з цих технологій перетворилися з невідомих на ті, які застосовуються майже всюди.

Зараз з’являється ще більше технологій: наприклад, Cache API для локального зберігання запитів і веб-сокети як альтернативний мережевий протокол для зв’язку клієнт-сервер (або навіть клієнт-клієнт). Згодом браузери мають намір повністю підтримувати варіант коду складання, відомий як веб-збірка, що дозволить використовувати мови, які не є JavaScript, для написання клієнтського коду у браузері [1].

У той же час, кожна з нових і майбутніх технологій несе з собою нові вразливості в безпеці, які можна знайти та використати зі злочинною метою.

REST API. REST означає передачу стану репрезентації, що є чудовим способом визначення API, який має кілька унікальних властивостей: він повинен бути окремим від клієнтського API REST призначені для створення високомасштабованих, але простих веб-програм.

Відокремлення клієнта від API, але дотримання суворої структури API, дозволяє клієнтській програмі легко запитувати ресурси від API, не маючи можливості здійснювати виклики до бази даних або виконувати логіку на стороні сервера. Він повинен бути без стану. За конструкцією API REST приймає лише вхідні дані та надає вихідні дані. API не повинні зберігати будь-який стан з’єднання клієнта.

Однак це не означає, що REST API не може виконувати автентифікацію та авторизацію. Натомість авторизація повинна бути токенізована та надіслана під час кожного запиту. Він повинен легко кешуватися. Щоб належним чином масштабувати веб-додаток, що поставляється через Інтернет, REST API повинен мати можливість легко позначати свої відповіді як такі, що кешуються чи ні. Оскільки REST також містить дуже чіткі визначення того, які дані будуть обслуговуватися з якої кінцевої точки, це насправді дуже легко налаштувати на правильно розробленому REST API.

В ідеалі, кешами слід керувати програмно, щоб випадково не передати

привілейовану інформацію іншому користувачеві. Кожна кінцева точка повинна визначати специфічний об’єкт або метод. Зазвичай вони визначаються ієрархічно; наприклад, */moderators/joe/logs/12_21_2018*. При цьому API REST можуть легко використовувати такі дієслова HTTP, як GET, POST, PUT і DELETE. В результаті одна кінцева точка з кількома дієсловами HTTP стає самодокументованою.

JavaScript Object Notation. REST – це специфікація архітектури, яка визначає, як HTTP-дієслова мають відображатися з ресурсами (кінцевими точками API та функціональними можливостями) на сервері. Більшість API REST сьогодні використовують JSON як формат переданих даних.

Зверніть увагу на це: API-сервер додатку повинен спілкуватися зі своїм клієнтом (зазвичай деякий код у браузері або мобільному додатку). Без взаємозв’язку клієнт/сервер ми не можемо мати збережений стан на різних пристроях і зберігати цей стан між обліковими записами. Весь стан потрібно було б зберігати локально.

Оскільки сучасні веб-додатки вимагають багато зв’язку між клієнтом і сервером (для обміну даними на нижньому рівні та запитів у формі HTTP-дієслів), неможливо надсилати дані в спеціальних форматах. Формат передачі даних має бути стандартизованим. JSON є одним з потенційних рішень цієї проблеми.

JSON – це відкритий стандартний (не пропрієтарний) формат файлу, який відповідає низці переваг:

він дуже легкий (зменшує пропускну здатність мережі);

він вимагає дуже малого аналізу (зменшує навантаження на обладнання сервера/клієнта); він легко читається людиною;

він є ієрархічним (може представляти складні взаємозв’язки між даними);

об’єкти JSON представлені дуже подібно до об’єктів JavaScript, що спрощує використання JSON та створення нових об’єктів JSON у браузері.

Усі основні браузери сьогодні підтримують синтаксичний аналіз JSON з достатньою швидкістю, що, на додаток до попередніх пунктів, робить JSON чудовим форматом для передачі даних між сервером без стану та веб-браузером.

JavaScript. Так, сервер – це комп’ютер (як правило, потужний), який

знаходиться в центрі обробки даних (іноді називається хмарою) і відповідає за обробку запитів до веб-сайту. Іноді ці сервери насправді можуть бути кластером багатьох серверів. В інших випадках це може бути просто один легкий сервер, який використовується для розробки або реєстрації.

З іншого боку, клієнт – це будь-який пристрій, до якого користувач має доступ, яким він керує для використання веб-додатку. Клієнтом може бути мобільний телефон, кіоск торгового центру або сенсорний екран в електромобілі, але для наших цілей це зазвичай буде просто веб-браузер.

Сервери можна налаштувати на запуск практично будь-якого програмного забезпечення, яке тільки можна собі уявити та будь-якою мовою.

Веб-сервери сьогодні працюють на Python, Java, JavaScript, C++ тощо. Клієнти (зокрема, браузер) не мають такого розмаїття. JavaScript – це не тільки мова програмування, але й єдина мова програмування для сценаріїв на стороні клієнта у веб-браузерах. JavaScript – це динамічна мова програмування, яка спочатку була розроблена для використання в інтернет-браузерах. Зараз JavaScript використовується в багатьох додатках, від мобільних пристроїв до Інтернету речей. JavaScript є унікальною мовою, оскільки розробка пов’язана зі зростанням браузера та його партнера, об’єктної моделі документа (DOM).

SPA Frameworks. Старіші веб-сайти, як правило, створювалися на основі комбінації спеціального сценарію для маніпулювання DOM і великої кількості повторно використаного коду шаблону HTML. Це не була масштабована модель, і хоча вона працювала для доставки статичного контексту кінцевому користувачеві, вона не працювала для доставки складних, багатих на логіку програм [1].

Програмне забезпечення для настільних комп’ютерів у той час було надійним у функціональності, дозволяючи користувачам зберігати та підтримувати стан програми. Веб-сайти в старі часи не забезпечували такого типу функціональних можливостей, хоча багато компаній воліли б доставити свої складні програми через Інтернет, оскільки це забезпечувало багато переваг від простоти використання до запобігання піратству [1].

Фреймворки односторінкових додатків (SPA) були розроблені для подолання функціонального розриву між веб-сайтами та настільними додатками. Фреймворки SPA дозволяють розробляти складні програми на основі JavaScript, які зберігають власний внутрішній стан і складаються з компонентів інтерфейсу, які можна повторно використовувати, кожен з яких має власний автономний життєвий цикл, від візуалізації до логічного виконання.

Сьогодні в Інтернеті широко поширені фреймворки SPA, які підтримують найбільші та найскладніші додатки (такі як Facebook, Twitter і YouTube), де функціональність є ключовою, а додатки майже настільні. Найбільш розповсюджені на сьогоднішній день фреймворки SPA з відкритим кодом: ReactJS, EmberJS, VueJS та AngularJS (рис. 1.1). Усі вони побудовані на основі JavaScript і DOM, але приносять додаткову складність як з точки зору безпеки, так і з точки зору функціональності.

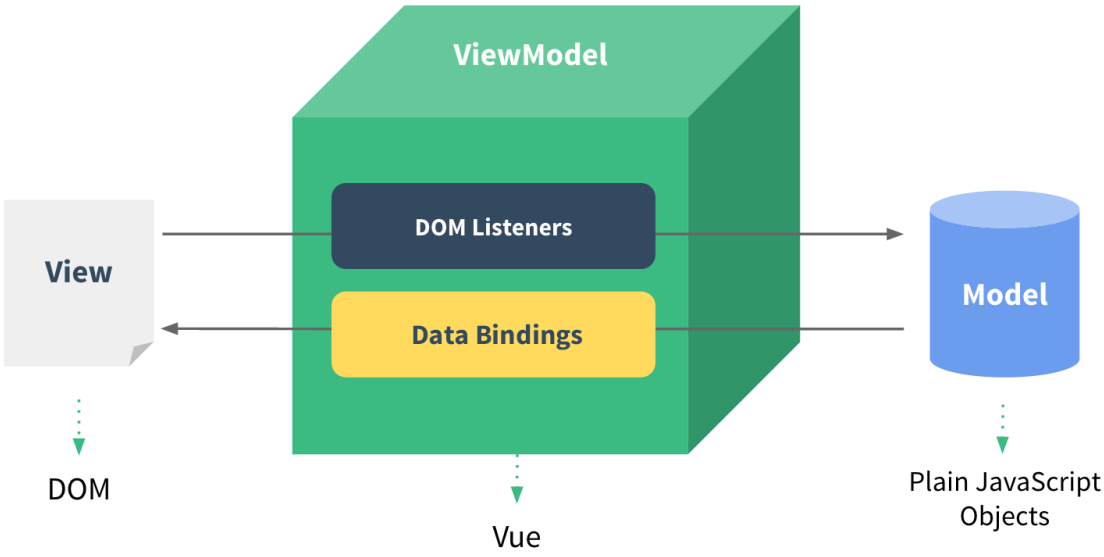

Рис. 1.1. VueJS – фреймворк односторінкового додатка, на який надбудовується веб-компоненти [2]

Як приклад, Vue.js – це прогресивне середовище Javascript з відкритим вихідним кодом для створення інтерфейсів користувача, які прагнуть поступового впровадження. Vue.js в основному використовується для фронт-енд і вимагає проміжного рівня HTML та CSS. Питання, пов’язані з версією Vue.js, мають бути

позначені [vuejs2] або [vuejs3]. Vue.js забезпечує двосторонню прив'язку даних, обчислювані властивості, CSS-прив'язки, HTML-шаблони, часткову візуалізацію і може бути розширена за допомогою компонентів, міксинів та плагінів [2].

Системи автентифікації та авторизації. У світі, де більшість додатків складається з клієнтів (браузерів/смартфонів) і серверів, а сервери зберігають дані, спочатку надіслані від клієнта, системи повинні бути на місці, щоб забезпечити майбутній доступ до даних, що зберігаються, від правильного користувача.

Сьогодні більшість веб-додатків вибирають з набору протоколів автентифікації, залежно від характеру бізнесу. Наприклад, протокол OAuth чудово підходить для веб-сайтів, які хочуть інтегруватися з більшими веб-сайтами. OAuth дозволяє великому веб-сайту (наприклад, Facebook, Google тощо) надавати партнерському веб-сайту маркер, що підтверджує ідентичність користувача.

OAuth може бути корисним для користувача, оскільки дані користувача потрібно оновлювати лише на одному сайті, а не на кількох сайтах. Але OAuth може бути небезпечним, оскільки один скомпрометований веб-сайт може призвести до кількох скомпрометованих профілів. Базова автентифікація HTTP та автентифікація дайджеста все ще широко використовуються сьогодні, причому дайджест є більш популярним, оскільки має більше захисту від перехоплення та атак відтворення.

Часто вони поєднуються з такими інструментами, як 2FA, щоб гарантувати, що маркери автентифікації не скомпрометовані, і що ідентичність користувача, який увійшов в систему, не змінився.

Авторизація є наступним кроком після автентифікації. Системи авторизації складніше класифікувати, оскільки авторизація дуже залежить від бізнес-логіки всередині веб-додатка.

Взагалі кажучи, добре розроблені додатки мають централізований клас авторизації, який відповідає за визначення того, чи має користувач доступ до певних ресурсів або функціональних можливостей. Якщо API написані погано,

вони виконуватимуть перевірки на основі API, що вручну відтворює функціональні можливості авторизації. Часто, якщо ви можете сказати, що програма повторно реалізує перевірки авторизації в кожному API, ця програма, ймовірно, матиме кілька API, де перевірок недостатньо просто через людську помилку [1].

Деякі поширені ресурси, які завжди повинні мати перевірку авторизації, включають налаштування/оновлення профілю, скидання пароля, читання/запис приватних повідомлень, будь-які платні функції та будь-які розширені функції користувача (наприклад, функції модерації) [1].

Веб-сервери. Сучасний веб-додаток «клієнт-сервер» базується на низці технологій, побудованих одна на одній, щоб компонент на стороні сервера та компонент на стороні клієнта функціонували за призначенням.

У випадку з сервером логіка додатку виконується поверх пакета веб-сервера на основі програмного забезпечення, тому розробникам додатків не доводиться турбуватися про обробку запитів та керування процесами.

Програмне забезпечення веб-сервера, звичайно, працює поверх операційної системи (зазвичай це якийсь дистрибутив Linux, як-от Ubuntu, CentOS або RedHat), яка працює поверх фізичного обладнання в центрі обробки даних. Але, що стосується програмного забезпечення для веб-серверів, у сучасному світі веб-додатків є кілька великих гравців.

Так, Apache досі обслуговує майже половину веб-сайтів у світі, тому ми можемо припустити, що Apache також обслуговує більшість веб-програм. Apache є відкритим вихідним кодом, розробляється близько 25 років і працює майже на кожному дистрибутиві Linux, а також на деяких серверах Windows (рис. 1.2).

Найбільшим конкурентом Apache є Nginx (вимовляється «Engine X»). Nginx працює на близько 30% веб-серверів і швидко розвивається. Хоча Nginx можна використовувати безкоштовно, його материнська компанія (зараз F5 Networks) використовує модель, де підтримка та додаткова функціональність платні.

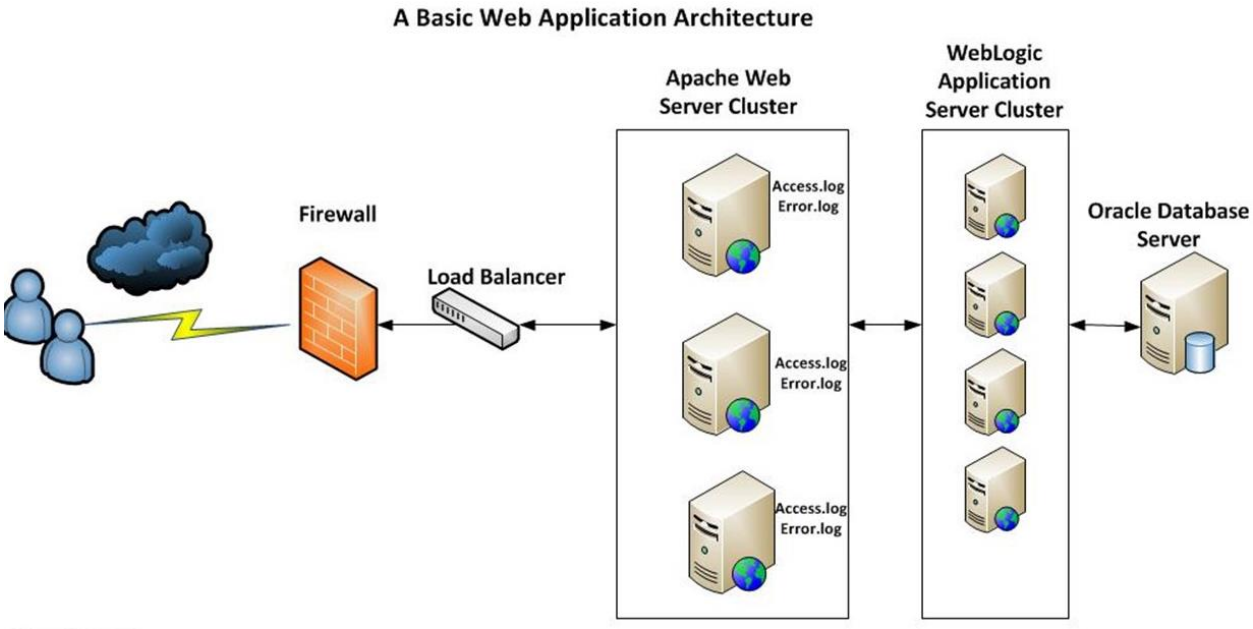

Рис. 1.2. Архітектура основних компонентів веб-додатку

Nginx використовується для додатків великого об’єму з великою кількістю унікальних з'єднань, на відміну від тих з невеликою кількістю з'єднань, які потребують великої кількості даних. Веб-додатки, які одночасно обслуговують багато користувачів, можуть відчути значне покращення продуктивності при переході з Apache на Nginx, оскільки архітектура Nginx має набагато менші витрати на підключення. За Nginx стоїть Microsoft IIS, хоча популярність серверів на базі Windows зменшилася через дорогі ліцензії та відсутність сумісності з пакетами програмного забезпечення з відкритим кодом (OSS) на базі Unix.

IIS є правильним вибором веб-сервера при роботі з багатьма специфічними для Microsoft технологіями, але він може бути тягарем для компаній, які намагаються створити на основі відкритого коду. Існує багато менших веб-серверів, і кожен з них має свої переваги та недоліки безпеки.

Серверні бази даних. Як тільки клієнт надсилає дані для обробки на сервер, сервер часто повинен зберігати ці дані, щоб їх можна було отримати в майбутньому сеансі. Зберігання даних у пам’яті ненадійне в довгостроковій перспективі, оскільки перезапуски та збої можуть призвести до втрати даних. Крім того, пам’ять із довільним доступом є досить дорогою в порівнянні з диском. Зберігаючи дані на диску, необхідно вжити належних запобіжних заходів,

щоб забезпечити надійне та швидке отримання даних, збереження та запитів [1].

Майже всі сучасні веб-додатки зберігають дані, подані користувачами, в базі даних певного типу, часто змінюючи базу даних, що використовується залежно від конкретної бізнес-логіки та варіанту використання. Бази даних SQL як і раніше є найпопулярнішою базою даних загального призначення на ринку. Мова запитів SQL сувора, але надійно швидка та проста для вивчення. SQL можна використовувати для чого завгодно: від зберігання облікових даних користувача до керування об’єктами JSON або невеликими крапками зображень/ Найбільшими з них є PostgreSQL, Microsoft SQL Server, MySQL і SQLite [1].

Коли потрібне більш гнучке сховище, можна використовувати бази даних NoSQL без схем. Бази даних, такі як MongoDB, DocumentDB і CouchDB, зберігають інформацію у вигляді слабо структурованих «документів», які є гнучкими і можуть бути змінені в будь-який час, але не такі прості чи ефективні при запитах або агрегації [1].

Серед сучасних веб-додатків також існують більш досконалі та особливі бази даних. Пошукові системи часто використовують власні вузькоспеціалізовані бази даних, які необхідно регулярно синхронізувати з основною базою даних. Прикладом цього є популярний Elasticsearch [1].

Необхідно зазначити, що кожен тип баз даних несе в собі унікальні проблеми та ризики. SQL-ін’єкція є добре відомим архетипом вразливості, ефективним проти основних баз даних SQL, коли запити не формуються належним чином. Однак атаки в стилі ін’єкції можуть відбутися майже на будь-яку базу даних, якщо хакер бажає вивчити модель запитів бази даних. Також багато сучасних веб-додатків можуть одночасно використовувати кілька баз даних. Додатки з достатньо безпечним генеруванням запитів SQL можуть не мати достатньо безпечних запитів і дозволів MongoDB або Elasticsearch [1].

Сховища даних на стороні клієнта. Традиційно на клієнті зберігається мінімум даних через технічні обмеження та проблеми з міжбраузерною сумісністю. Зараз багато програм зберігають дані на клієнті, часто у вигляді даних конфігурації або великих сценаріїв, які могли б викликати перевантаження

мережі, якщо їх потрібно було завантажувати під час кожного відвідування.

У більшості випадків керований браузером контейнер зберігання, який називається локальним сховищем, використовується для зберігання та доступу до даних ключ/значення від клієнта. Локальне сховище дотримується браузера, який застосовує ту саму політику походження (SOP), яка забороняє іншим доменам (веб-сайтам) отримувати доступ до локально збережених даних один одного. Веб-програми можуть підтримувати стан, навіть коли браузер або вкладка закриті [1].

Підмножина локального сховища, що називається сховищем сеансів, працює ідентично, але зберігає дані лише до закриття вкладки. Цей тип сховища можна використовувати, коли дані важливіші, і його не слід зберігати, якщо інший користувач використовує ту саму машину. У погано спроектованих веб-додатках сховища даних на стороні клієнта також можуть розкривати конфіденційну інформацію, таку як маркери автентифікації або інші секрети.

Нарешті, для більш складних програм підтримка IndexedDB є сьогодні у всіх основних веб-браузерах. IndexedDB – це база даних об’єктно-орієнтованого програмування (ООП) на основі JavaScript, здатна зберігати та виконувати асинхронні запити у фоновому режимі веб-програми [1].

Як висновок, сучасні веб-додатки побудовані на основі низки нових технологій. Завдяки розширеній функціональності веб-додатків, набагато більше різних форм атак можуть бути спрямовані на них.

1.3. Аналіз проблеми забезпечення безпеки веб-додатків

Необхідно підкреслити, що підходи до забезпечення безпеки для веб-додатків докорінно змінилися. Сучасні веб-додатки важче захищати через те, як вони розроблені. Вони набагато більш відкриті та розподілені. Сучасний веб-додаток є сукупністю взаємодій між різними API, службами, кінцевими точками, пристроями тощо. Ці взаємодії охоплюють широкий діапазон кордонів як всередині організації, так і за її межами, і поверхню атаки, що постійно розвивається, важко захистити.

Занадто часто організації намагаються захиститися за допомогою неадекватних інструментів або тих, що не призначені для виконання завдання захисту, або недостатньо швидко адаптуються до нових типів атак. Результатом є занадто багато інструментів, забагато підходів і забагато думок, що призводить до обмеженої ефективності безпеки, впливає на продуктивність додатків і, у гіршому випадку, до атаки (headline-grabbing attack).

Сьогодні атаки веб-додатків є часто є повільними та методологічними, досліджують всю площу великого розподіленого додатка на предмет наявності будь-якої потенційної точки входу. Як тільки ця точка входу знайдена, зловмисник розробляє експлойт спеціально для неї. Такі атаки розумно оптимізовані проти бізнес-моделі додатку, використовуючи автоматизовані інструменти, які постійно досліджують додаток на наявність різних слабких місць [1].

Захисні стратегії стосуються того, як розробляються та створюються сучасні додатки. Гарна безпека починається з хорошого дизайну, який враховує розподілений характер додатку [1].

В [1] підкреслюється, що безпека – це не окремий випадок, а процес, який необхідно включити в весь життєвий цикл розробки програмного забезпечення. Усі сторони, які беруть участь у розробці сучасних додатків, повинні розуміти, як безпека узгоджується з життєвим циклом розробки. Більш чітке узгодження призводить до вибору кращих інструментів і створення найкращих практик, спеціально розроблених для потреб безпеки додатків.

Оскільки одна частина стеку додатків стає все більш безпечною, хакери переходять до націлювання на нові технології. Ці нові технології часто не мають такого ж рівня вбудованого контролю безпеки, і лише шляхом проб і помилок інженери можуть розробити та впровадити належні засоби контролю безпеки [1].

Кожна нова технологія має свою унікальну поверхню атаки та вразливості. Один із способів стати відмінним хакером – це завжди бути в курсі останніх нових технологій, бо вони часто мають вразливості, які ще не опубліковані чи знайдені.

Щоб ефективно використовувати веб-додатки, потрібен широкий спектр навичок. З одного боку, хакерові потрібні знання мережевих протоколів, методів розробки програмного забезпечення та поширених уразливостей, які можна знайти в різних типах додатків. Але з іншого боку, хакер також повинен розуміти додаток, на який він націлений. Хакер повинен розуміти мету додатку з функціональної точки зору; визначити його користувачами; як додаток приносить дохід; чому користувачі вибирають додаток перед конкурентами; хто конкуренти; яка функція міститься в програмі тощо [1].

Зрештою, розвідка веб-додатків полягає в зборі даних і побудові моделі, яка поєднує технічні та функціональні деталі веб-додатка таким чином, що дозволяє повністю зрозуміти призначення та використання веб-додатка. Без того чи іншого хакер не може належним чином націлити свою атаку.

Розвідка веб-додатків відноситься до фази дослідницького збору даних, яка зазвичай відбувається до злому веб-програми. Розвідка веб-додатків зазвичай виконується хакерами, пентестувальниками або мисливцями за помилками, але також може бути ефективним способом для інженерів безпеки знайти слабко захищені механізми у веб-додатку та виправити їх до того, як їх знайде зловмисник.

Навички розвідки самі по собі не мають значної цінності, але стають все більш цінними в поєднанні зі знаннями про напади хакерів та досвідом оборонної техніки безпеки.

Необхідно підкреслити, що більшість методів розвідки слід виконувати лише проти програм, якими ви володієте або маєте письмовий дозвіл на тестування. Розвідку можна здійснити багатьма способами. Іноді просто навігація веб-додатком і запис мережевих запитів – це все, що потрібно, щоб познайомитися з внутрішньою роботою цього додатку [1].

Однак важливо зазначити, що не всі веб-додатки будуть мати користувальницький інтерфейс, який дозволить візуально досліджувати додаток та звернути увагу на її функціональність. Більшість загальнодоступних додатків (часто додатків для бізнес-споживачів, як-от соціальні медіа) матимуть

загальнодоступний користувальницький інтерфейс.

Методи розвідки є цінними для розвитку глибокого розуміння технології та структури веб-додатка та служб, які забезпечують цей веб-додаток.

Розглянемо застосування положень NIST Cybersecurity Framework [7] до веб-безпеки. NIST Cybersecurity Framework складається з трьох частин. Ядро Framework – це набір заходів з кібербезпеки, бажаних результатів і загальних посилань. У ньому представлені стандарти, керівні принципи та методи, що дає змогу повідомити про дії та результати кібербезпеки на всіх рівнях організації. Він містить чотири елементи: функції, категорії, підкатегорії та інформаційні посилання. Рівні впровадження Framework зосереджені на ризиках кібербезпеки та процесах управління цим ризиком. Профіль Framework містить повторювані результати на основі потреб бізнесу, вибраних із категорій і підкатегорій Framework.

Щоб чітко побачити, як NIST Cybersecurity Framework застосовується до веб-безпеки, краще подивитися на структуру ядра Framework. Верхній рівень ядра – це такі функції: ідентифікація, захист, виявлення, відповідь та відновлення. Кожна з функцій містить кілька категорій, наприклад, функція «Ідентифікація» (рис. 1.3) містить такі категорії, як управління активами, бізнес-середовище, управління, оцінка ризиків, стратегія управління ризиками та управління ризиками ланцюга постачання. Кожна з цих категорій має конкретні підкатегорії, які визначають бажані результати.

Рис. 1.3. NIST Cybersecurity Framework [7]

Жоден з елементів CSF NIST не призначений спеціально для Інтернету, але багато з них також стосуються веб-безпеки. Ось кілька яскравих прикладів:

ID.AM-2: Інвентаризація програмних платформ і додатків в організації: цей результат також стосується веб-додатків. Щоб досягти цього результату потрібно знайти спосіб створити перелік усіх веб-додатків організації.

ID.RA-2: розвідку про кіберзагрози отримують з форумів та джерел обміну інформацією: з точки зору Інтернету, це означає, що зовнішні джерела потрібні для того, щоб знати про потенційні події кібербезпеки в Інтернеті, наприклад, нові вектори атак.

PR.DS-5: реалізовано захист від витоку даних: критичні дані часто доступні за допомогою веб-додатків. Щоб уникнути порушення даних і гарантувати безпеку даних, організації повинні впровадити відповідні засоби контролю доступу до Інтернету. Багато нещодавніх витоків були викликані незахищеними документами, доступними через Інтернет.

DE.CM-8: сканування вразливостей: той факт, що сканування вразливостей виділено як окрема категорія, свідчить про його важливість у процесах виявлення відповідно до NIST.

Необхідно зазначити, що ненадійне програмне забезпечення підриває

безпеку критичних інфраструктур, що стосуються, наприклад, охорони здоров'я, оборони, енергетики або фінансів. Програмне забезпечення стає складнішим, пристроїв, підключених до мережі, стає більше, тому важливість забезпечення безпеки програм зростає експоненційно. Швидкий розвиток методів розробки ПЗ призводить до необхідності швидко і безпомилково виявляти, а також усувати загрози, що найчастіше виникають. Необхідно приділяти належну увагу загрозам безпеці (рис. 1.4), які представлено у списку Топ-10 OWASP [3].

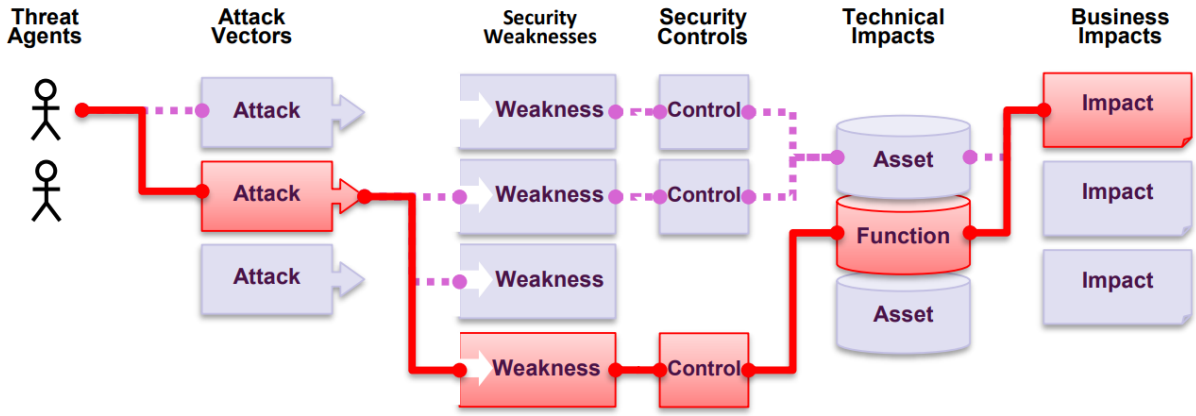

Рис. 1.4. Приклад загрози безпеці вед-додатку ці вед-додатку [3]

Іноді ці способи легко знайти та експлуатувати, іноді дуже складно. Аналогічна ситуація з можливим збитком: його може не бути зовсім, або він може відчутним для бізнесу. Щоб визначити ризики для організації, необхідно оцінювати ймовірності, пов'язані з джерелами загроз, векторами атак і недоліками безпеки, а потім поєднати їх з оцінкою технічної та репутаційної шкоди для конкретної організації. Сума цих чинників визначається сукупний ризик [3].

Десять найбільш поширених вразливостей веб-додатків за [3].

A1:2017 – Впровадження. Вразливості, пов'язані, наприклад, із використанням SQL, NoSQL, OS і LDAP, виникають, коли неперевірені дані відправляються інтерпретатору у складі команди чи запиту. Шкідливі дані можуть змусити інтерпретатор виконати непередбачені команди або звернутися до даних без відповідної авторизації [3].

A2:2017 – Недоліки автентифікації. Функції програм, пов'язані з

автентифікацією та керуванням сесіями, часто некоректно реалізуються, дозволяючи зловмисникам скомпрометувати паролі, ключі або сесійні токени, а також експлуатувати інші помилки реалізації для тимчасового або постійного перехоплення облікових записів користувачів [3].

A3:2017 – Розголошення конфіденційних даних. Багато веб-додатків та API мають поганий захист критичних фінансових, медичних або персональних даних. Зловмисники можуть викрасти або змінити ці дані, а потім здійснити шахрайські дії із кредитними картками або персональними даними. Конфіденційні дані вимагають додаткових заходів захисту, наприклад їх шифрування при зберіганні або передачі, а також спеціальних запобіжних заходів при роботі з браузером [3].

A4:2017 – Зовнішні сутності XML (XXE). Старі або погано налаштовані процесори XML обробляють посилання на зовнішні сутності всередині документів. Ці сутності можуть бути використані для доступу до внутрішніх файлів через обробники URI файлів, спільні папки, сканування портів, віддалене виконання коду та відмова в обслуговуванні [3].

A5:2017 – Недоліки контролю доступу. Дії, які дозволені автентифікованим користувачам, часто некоректно контролюються. Зловмисники можуть скористатися цими недоліками і отримати несанкціонований доступ до облікових записів інших користувачів або конфіденційної інформації, а також змінити дані користувача або права доступу [3].

A6:2017 – Некоректне налаштування параметрів безпеки. Некоректне налаштування безпеки є поширеною помилкою. Це відбувається через використання стандартних параметрів безпеки, неповного або специфічного налаштування, відкритого хмарного зберігання, некоректних HTTP-заголовків та докладних повідомлень про помилки, що містять критичні дані. Всі ОС, фреймворки, бібліотеки та додатки повинні бути не тільки налаштовані належним чином, а й своєчасно коригуватися та оновлюватись [3].

A7:2017 – Міжсайтове виконання сценаріїв (XSS). XSS має місце, коли програма додає неперевірені дані на нову веб-сторінку без їхньої відповідної перевірки чи перетворення, або коли оновлює відкриту сторінку через API

браузера, використовуючи надані користувачем дані, що містять HTML- або JavaScript-код. За допомогою XSS зловмисники можуть виконувати сценарії в браузері жертви, що дозволяють їм перехоплювати сесії, підміняти сторінки сайту або перенаправляти користувачів на шкідливі сайти [3].

A8:2017 – Небезпечна десеріалізація. Небезпечна десеріалізація часто призводить до віддаленого виконання коду. Помилки десеріалізації, що не призводять до віддаленого виконання коду, можуть бути використані для атак із повторним відтворенням, впровадженням та підвищенням привілеїв.

A9:2017 – Використання компонентів з відомими вразливістю [3]. Компоненти, такі як бібліотеки, фреймворки та програмні модулі, запускаються з привілеями програми. Експлуатація вразливого компонента може призвести до втрати даних або перехоплення контролю над сервером. Використання додатків та компонентів API з відомими вразливістю може порушити захист програми та призвести до серйозних наслідків [3].

A10:2017 – Недоліки ведення журналів та моніторингу. Недоліки ведення журналів та моніторингу, а також відсутність або неефективне використання системи реагування на інциденти, дозволяють зловмисникам розвинути атаку, приховати свою присутність та проникнути в інші системи, а також змінити, вилучити чи знищити дані. Проникнення в систему зазвичай виявляють лише через 200 днів і, як правило, сторонні дослідники, а не в рамках внутрішніх перевірок чи моніторингу [3].

Додатки відіграють важливу роль у підтримці ключових бізнес-процесів, тому організації намагаються забезпечити їх безпеку. Згідно з [6] 62% фахівців з кібербезпеки в кращому випадку помірно впевнені в позиції безпеки додатків своєї організації. Не дивно, що приблизно стільки ж вважають свої стратегії безпеки додатків незрілими.

За словами 41% опитаних, веб-додатки, орієнтовані на клієнтів, становлять найбільший ризик для бізнесу (рис. 1.5). Згідно з нещодавньою статтею TechRepublic, майже *кожний веб-додаток сьогодні має принаймні одну вразливість* [6]. Найбільше насторожує керівників безпеки, що час, необхідний

для усунення критичних вразливостей додатків, часто вимірюється місяцями, а не днями чи навіть тижнями.

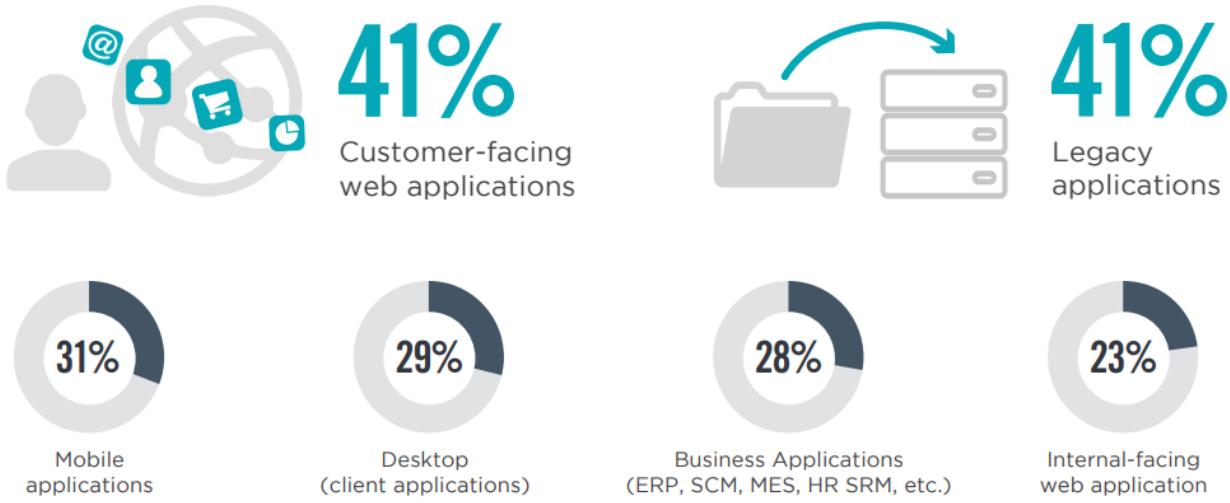

Рис. 1.5. Типи додатків та їх ризик для безпеки [6]

Відповідно до [6] 23% організацій (рис. 1.6) зазнали зламу або компромісу додатків протягом останніх 12 місяців, а 8% зазнали атаки лише протягом попереднього місяця. Більш тривожним є те, що більше третини респондентів навіть не могли повідомити, чи мали вони злами чи компроміс протягом останніх п’яти років.

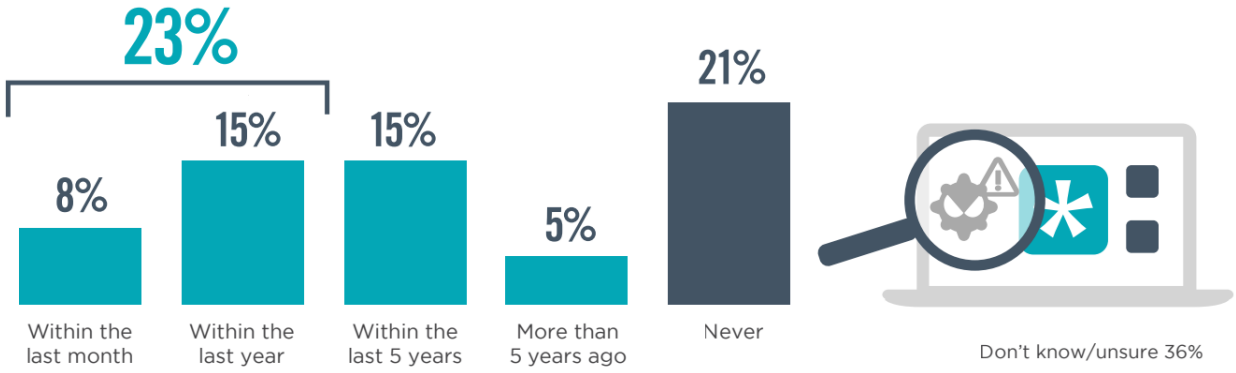

Рис. 1.6. Відповіді на питання коли в останнє було зламано бізнес-додатки [6]

Зрозуміло, що організації повинні покращити спосіб захисту бізнес-додатків, особливо з огляду на те, що все більше додатків розробляється та розміщується в Інтернеті для зовнішнього доступу.

Розглянемо рекомендації компанії Tenable щодо кращого захисту веб-

додатків від останніх загроз кібербезпеці.

Через появу «тіньових IT», у яких відділи, групи та окремі користувачі розгортають бізнес-додатки без повноважень на це або відсутність знань про централізоване управління IT, багато команд із забезпечення безпеки не бачать, які додатки використовуються в організації. Наприклад, 89% респондентів кажуть, що їхні організації не впевнені, що знають усі додатки, які використовуються. Лише 11% кажуть, що впевнені, що знають усі додаткии. Крім того, 17% організацій не можуть навіть надати приблизний діапазон того, скільки додатків вони мають у своєму середовищі [6].

Погана видимість веб-додатків сьогодні є серйозною проблемою для команд безпеки. В одному нещодавньому дослідженні Symantec було визначено, що розрив між уявними та реальними хмарними додатками, що використовуються, становить понад 20 разів [6].

Найкраща практика № 1: автоматичне виявлення веб-додатків. Автоматичне виявлення додатків, розгорнутих в організації, усуває дорогі ручні здогади та небезпечні сліпі зони безпеки. Завдяки новим методам DevOps, веб-додатки створюються та розгортаються все більшою кількістю команд швидше, ніж будь-коли, і дуже важко встигати за ручними списками IP-адрес і доменних імен. Необхідно використовувати переваги рішень, які можуть ідентифікувати веб-додатки за допомогою діапазонів IP-адрес, доменних імен і сканування портів HTTP.

Керівники з кібербезпеки постійно скаржаться на брак ресурсів для захисту від останніх атак. Справді, в [6] підтверджується, що нестача кваліфікованого персоналу (37%) і брак бюджету (35%) є одними з головних перешкод, які гальмують захист від кіберзагроз.

Навички та бюджет також є двома головними проблемами, які гальмують тестування на проникнення додатків в організаціях. Щоб підкреслити дефіцит ресурсів, команди з безпеки додатків часто співвідносяться до розробників як 1:100. Через обмеження навичок, ресурсів і бюджету більшість організацій можуть оцінювати та захищати лише найбільш важливі веб-додатки – менше ніж

10% від загальної кількості веб-додатків – за допомогою ручного тестування на проникнення. Це означає, що інші 90% основних вед-додатків недостатньо захищені та їх необхідно оцінити на предмет ризиків [6].

Найкраща практика №2: автоматизуйте сканування веб-програм. Враховуючи, що попит на навички безпеки продовжує перевищувати пропозицію, автоматизація має вирішальне значення для зниження витрат як на технології, так і на персонал.

Організаціям потрібні рішення для сканування веб-додатків, які можуть швидко, точно й автоматично оцінювати всі веб-додатки. Досліджуйте рішення, які дозволяють вам «налаштувати і забути», плануючи часті та повторювані сканування веб-додатків, щоб захистити корпоративне середовище, що постійно змінюється.

Організації мають справу зі зростаючою складністю свого IT-середовища, включаючи кількість і різноманітність бізнес-додатків, що використовуються. Також існує складність з точки зору інструментів та послуг безпеки. Не дивно, що більше половини (54%) респондентів кажуть, що простота інтеграції є найважливішим критерієм при виборі рішення безпеки програми.

Рішення з роз’єднаними точками, розроблені для конкретних типів активів, створюють відокремлену видимість і надмірні витрати на керування. Тому сканування веб-додатків має бути частиною ширшої інтегрованої платформи Cyber Exposure, щоб допомогти групам безпеки керувати та вимірювати кіберризики на всій поверхні атаки.

Найкраща практика № 3: зробіть безпеку додатків частиною вашої загальної практики Cyber Exposure. Розгорніть прості у використанні та інтуїтивно зрозумілі продукти для сканування веб-додатків, які інтегровані в більш широку платформу Cyber Exposure. Це зменшує потребу в високоспеціалізованому персоналі з безпеки додатків для налаштування, розгортання та керування цими системами.

Крім того, це дає змогу оцінити додатки на предмет кібер-ризиків і визначити пріоритети усунення вразливостей разом з іншими активами на

поверхні атаки.

Отримайте більш широке покриття безпеки, заощаджуючи час і гроші, і перерозподіліть персонал для інших більш цінних заходів. Оскільки з’являється так багато векторів загроз і вразливостей, важко знати, на чому зосередитися. Організаціям в середньому потрібно усувати понад 800 вразливостей на день у майже 1000 активах. Приблизно дві третини цих вразливостей мають оцінку CVSS 7,0 або вище. Це означає, що служби безпеки постійно реагують на невпинний шквал нових кібер-ризиків. Майже половина респондентів опитування (45%) підтверджують, що не відставання від зростаючого числа вразливостей є найбільшою проблемою безпеки додатків [6].

Проблема полягає в тому, що командам з безпеки бракує даних і розуміння, які їм потрібні, щоб визначити пріоритети виправлення, а це означає, що вони не можуть швидко вирішити найважливіші проблеми безпеки. Як наслідок, час, необхідний для усунення лише вразливостей високого та критичного ризику, часто вимірюється місяцями. Це наражає організацію на надмірний і непотрібний кіберризик.

Найкраща практика № 4: визначить пріоритети на основі ризику. Розширене визначення пріоритетів на основі фактичного кібер-ризику, поєднання критичності активів, серйозності вразливості та доступності експлойтів, є важливим для захисту ваших програм.

Відфільтруйте вразливі місця з меншим ризиком, щоб ви могли працювати над усуненням найбільш важливих для бізнесу проблем безпеки, і зосередьтеся на вразливостях, які активно експлуатуються суб’єктами загроз, а не на тих, які могли б бути використані лише теоретично.

Багато організацій просто занадто повільно реагують на інциденти з безпеки або недостатньо активно зупиняють атаки. Майже половина всіх опитаних організацій (48%) сканують програми щоквартально або навіть рідше. Ця нестача швидкості не забезпечить належного захисту додатків у сучасну епоху швидкої розробки програмного забезпечення [6].

Зростання DevOps означає, що випускаються нові веб-додатки, а існуючі

веб-додатки оновлюються набагато швидше, ніж у минулому. Наприклад, зрілі організації DevOps випускають код кілька разів на день, і багато організацій (59%) оновлюють існуючі веб-додатки принаймні раз на місяць. Безпека ще не адаптувалася до нової реальності DevOps [6].

Найкраща практика № 5: інтегруйте безпеку в життєвий цикл розробки програмного забезпечення. Підтримуйте високошвидкісні процеси DevOps, інтегруючи сканування веб-додатків у конвеєр DevOps перед випуском програми.

Необхідно відмітити, що, чим раніше буде усунено дефекти безпеки в життєвому циклі розробки програмного забезпечення, тим менш витратним буде їх усунення. Крім того, необхідно сканувати веб-додатки, які працюють, щонайменше раз на місяць, щоб забезпечити постійну видимість кіберризику [6].

1.4. Аналіз існуючих рішень з кібербезпеки веб-додатків організації

Незважаючи на те, що рішення Web Application Firewall в світі вже встигли набрати достатню популярність, і такі компанії як Amazon надають захищені сервіси на базі продуктів даного класу, на ринку України для багатьох використання WAF ще в новинку. Сильними гравцями на ринку є Fortinet з рішенням FortiWeb і Imperva з SecureSphere Web Firewall Application.

FortiWeb – це WAF, який захищає розміщені веб-додатки від атак, спрямованих на відомі та невідомі експлойти [8].

Використовуючи *багаторівневі та кореляційні методи* виявлення, FortiWeb захищає програми від відомих вразливостей та загроз нульового дня. Служба безпеки веб-додатків від FortiGuard Labs використовує інформацію, засновану на останніх вразливостях додатків, ботах, підозрілих URL-адресах та шаблонах даних, а також спеціалізованих евристичних механізмах виявлення для забезпечення безпеки веб додатків [9].

FortiWeb також пропонує *функцію машинного навчання*, яка дозволяє автоматично виявляти шкідливий веб-трафік. Окрім виявлення відомих атак, ця функція може виявляти потенційні невідомі атаки нульового дня для забезпечення

захисту веб-серверів у режимі реального часу.

FortiWeb дозволяє налаштовувати такі функції [9]:

сканування вразливостей та виправлення;

репутація IP, сигнатури атак веб-застосунків, захист облікових даних, антивірус та FortiSandbox Cloud на базі FortiGuard;

аналіз атак та звітність у режимі реального часу за допомогою розширених інструментів візуальної аналітики;

інтеграція з FortiGate та FortiSandbox для виявлення ATP атак;

виявлення поведінкової атаки;

розширене запобігання хибно позитивним та негативним результатам виявлення.

Платформи обладнання та віртуальних машин FortiWeb доступні для середніх та великих підприємств, а також для постачальників послуг.

Перевагою FortiWeb є те, що дане рішення розроблено спеціально для захисту веб-серверів. Воно забезпечує спеціалізоване виявлення загроз на рівні додатків та захист для служб HTTP та HTTPs, у тому числі: Apache Tomcat; nginx; Microsoft IIS; JBoss; IBM Lotus Domino; Microsoft SharePoint; Microsoft Outlook Web App (OWA); RPC та ActiveSync для Microsoft Exchange Server; Joomla та WordPress

Інтегрований веб-сканер вразливостей FortiWeb значно знижує проблеми, пов’язані із захистом регульованих та конфіденційних даних, за рахунок виявлення схильності корпоративних рішень до найновіших загроз, особливо з OWASP Top 10 [8].

Брандмауер HTTP та захист від атак типу «відмова в обслуговуванні» (DoS) FortiWeb захищають корпоративні веб-додатки від атак. Використовуючи передові методи для забезпечення двоспрямованого захисту від складних загроз, таких як SQL-ін’єкції та атаки міжсайтового скріптингу (XSS), FortiWeb також допомагає захищатися від таких загроз, як крадіжка особистих даних, фінансове шахрайство та корпоративне шпигунство [8].

FortiWeb надає інструменти, необхідні для моніторингу та забезпечення

дотримання державних постанов, передових галузевих практик та внутрішніх політик безпеки, включаючи вимоги до брандмауерів та виправлень із стандарту PCI DSS (рис. 1.7) [8].

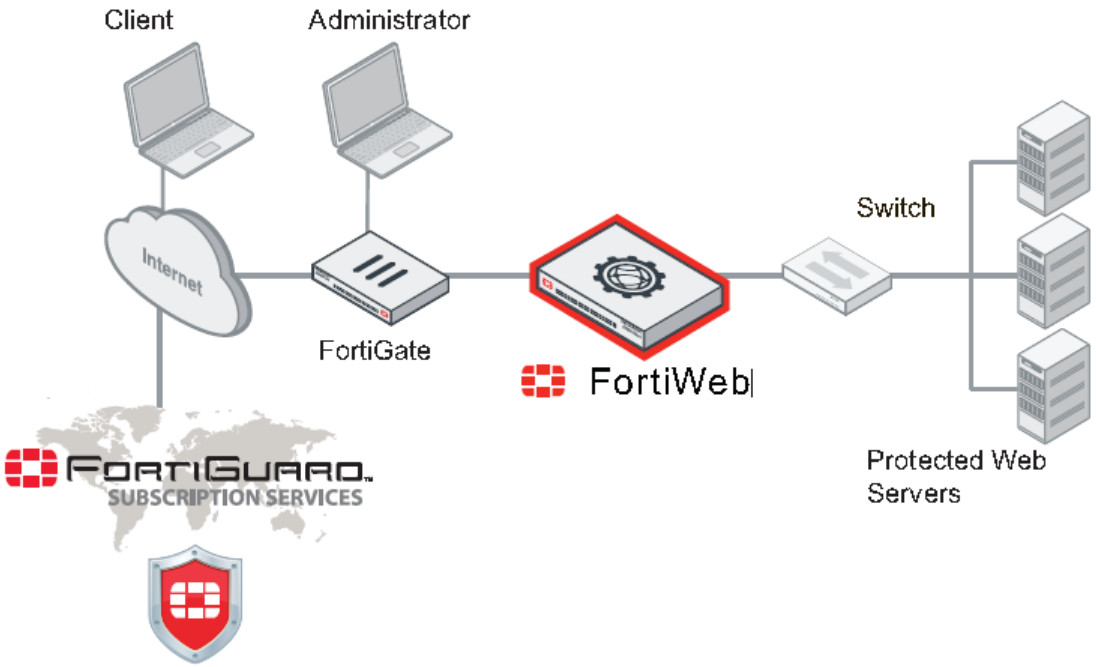

Рис. 1.7. Місце FortiWeb в архітектурі безпеки [8]

FortiWeb можна розгорнути в топології “одна рука”, але частіше він розміщується в оперативному режимі, щоб перехоплювати всі вхідні клієнтські з’єднання та перерозподіляти їх на корпоративні сервери. FortiWeb має спеціальні можливості брандмауера для TCP та HTTP. Оскільки він не призначений для безпеки веб-додатків, відмінних від HTTP/HTTPs, його слід розгортати за брандмауером, таким як FortiGate (рис. 1.7), який орієнтований на безпеку для інших протоколів, включаючи FTP та SSH. Після розгортання FortiWeb його можна налаштувати за допомогою веб-браузера або емулятора терміналу на керуючому комп’ютері [8].

Також серед переваг FortiWeb необхідно відзначити той факт, що в своїх апаратних і програмних рішеннях компанія Fortinet використовує власні технології. Так, для прискорення обробки і шифрування даних застосовуються спеціалізовані процесори FortiASIC. Крім того, розроблена своя операційна система FortiOS, оптимізована під завдання безпеки.

Fortinet якісно супроводжує свою продукцію, постійно виконуючи роботи по виявленню нових загроз у власному центрі FortiGuard аналітичному Центр безпеки. В результаті оновлення сигнатур, чорних списків сайтів і баз репутацій відбувається кілька разів на день.

Важливо також треба відзначити тісну інтеграцію всіх пристроїв Fortinet за рахунок повної сумісності між собою, що дозволяє швидко і просто масштабувати систему. Істотною перевагою систем захисту на базі Fortinet є високий ступінь автоматизації операцій і простота їх супроводу. Це дозволяє скоротити число помилок, викликаних людським фактором і зменшити число співробітників обслуговуючого персоналу.

Imperva SecureSphere Web Firewall Application – один з лідерів в області відображення атак на веб-додатки. Рішення від Imperva забезпечують надійний захист за рахунок одночасного застосування декількох технологій в сфері безпеки: сигнатурний аналіз, перевірка протоколів на аномалії, відстеження сесій, динамічне профілювання (рис. 1.8) [9].

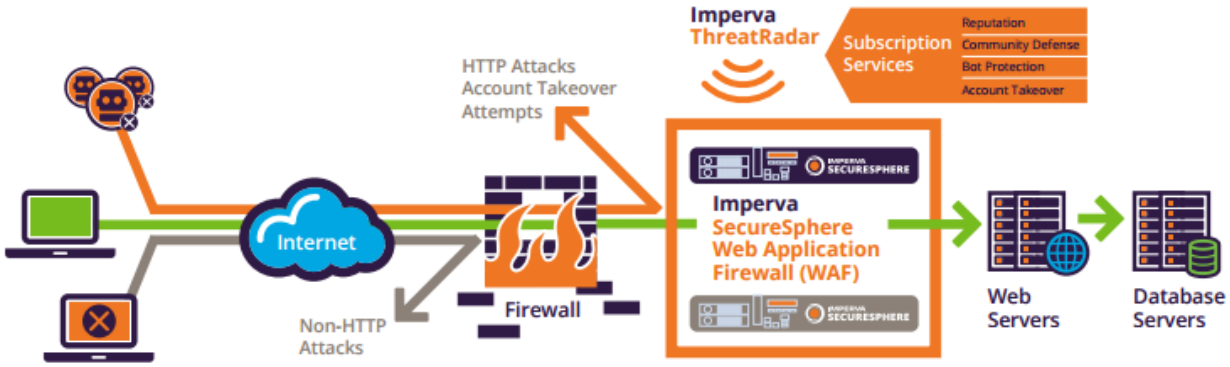

Рис. 1.8. Imperva SecureSphere Web Firewall Application [9]

Сімейство продуктів Secure-Sphere WAF успішно протидіє всім з OWASP TOP 10 атакам, так і іншим менш відомим, але більш витонченим атакам. Пристрої мають можливості по інспектування зашифрованих даних, що пересилаються по протоколу SSL (HTTPS).

Рішення складається з наступних модулів [9]:

SecureSphere Web Firewall Application – захист веб-додатків від кібератак;

ThreatRadar – репутаційна база даних.

Secure-Sphere WAF здатний глибоко аналізувати логіку роботи легального веб-додатки, виконувати інтелектуальне дослідження спроб проникнення і атак; HTTP-протидіяти атакам, включаючи атаки на переповнення буфера, дії шкідливих програм і зловмисників. Рішення має механізм захисту від черв'яків і інших шкідливих атак на веб-сервери і додатки, основу якого складають механізми на основі сигнатур популярної системи Snort і власних SQL-сигнатур, що розробляються дослідним центром ADC (Application Defense Center) компанії Imperva. Вбудований міжмережевий екран здійснює надійний захист від неавторизованих призначених для користувача запитів і атак на мережевому рівні [9].

Попередньо в системі звіти повністю задовольняють вимогам стандартів інформаційної безпеки. Можливе створення призначених для користувача звітів (в тому числі за розкладом) та експорт в різні формати. Додаткові хмарні сервіси дозволяють спростити безпеку і впорається з DDoS-атаками.

Важлива перевага пристроїв SecureSphere WAF: наявність унікального сервісу ThreatRadar, що забезпечує захист від автоматизованих атак. Завдяки швидкому отриманню достовірної інформації про джерела атак, ThreatRadar дозволяє негайно блокувати трафік, що йде від підозрілих джерел, ще до моменту здійснення будь-якого руйнівної дії. Рішення Imperva відрізняються прозорою підтримкою і простим розгортанням [9].

2 АНАЛІЗ МЕТОДІВ ТА ЗАСОБІВ ЗАХИСТУ ВЕБ-ДОДАТКІВ ОРГАНІЗАЦІЇ НА БАЗІ AWS WEB APPLICATION FIREWALL

2.1. Дослідження можливостей використання додатків AWS для бізнеса

Amazon Web Services (AWS) – це найпоширеніша у світі хмарна платформа з найширшими можливостями, що надає понад 200 повнофункціональних сервісів для центрів обробки даних по всій планеті. Мільйони клієнтів, у тому числі стартапи, які стали лідерами за швидкістю зростання, найбільші корпорації та передові урядові установи, використовують AWS для зниження витрат, підвищення гнучкості та прискореного впровадження інновацій [12].

AWS надає незрівнянно більше сервісів та їх функцій, ніж будь-який інший постачальник хмарних послуг: від інфраструктурних технологій, таких як інструменти для обчислення, сховища та бази даних, до інновацій, наприклад машинного навчання та штучного інтелекту, озер даних та аналітики, а також Інтернету речей. З ними клієнт зможе швидше, легше та дешевше перенести поточні додатки в хмару та реалізовувати у ньому будь-які можливі проекти [12].

AWS також надає найширші функціональні можливості для своїх сервісів. Наприклад, AWS пропонує на вибір багато баз даних, спеціально створених для різних типів програм, щоб клієнт міг підібрати правильний інструмент для ефективної та економної роботи [12].

На платформі AWS створено найбільшу та найдинамічнішу спільноту з мільйонами активних клієнтів та десятками тисяч партнерів по всьому світу. Клієнти різного масштабу та практично будь-яких галузей, у тому числі стартапи, великі корпорації та державні організації, використовують AWS для найрізноманітніших завдань. У партнерську мережу AWS (APN) входять тисячі системних інтеграторів, що спеціалізуються на сервісах AWS, та десятки тисяч незалежних постачальників ПЗ (ISV), що адаптують свої технології для роботи на AWS [12].

AWS – найбільш гнучке та захищене середовище для хмарних обчислень з існуючих. Базова інфраструктура спроектована так, щоб задовольнити вимоги щодо безпеки міжнародних банків, установ у сфері оборони та інших організацій з високими вимогами до захисту даних. Підтримка здійснюється за рахунок різноманітних інструментів для забезпечення безпеки у хмарі, які включають 230 сервісів та можливостей щодо забезпечення безпеки, відповідності вимогам та управлінню. AWS підтримує 90 стандартів безпеки та сертифікацій на відповідність вимогам, і всі 117 сервісів AWS для зберігання клієнтських даних пропонують можливість їх шифрування [12].

Компанія у звіті Magic Quadrant for Cloud Infrastructure & Platform Services (CIPS) 2021 року (рис. 2.1). У контексті «магічного квадранта» сервіси CIPS визначаються як «стандартизовані високо автоматизовані пропозиції, в яких ресурси інфраструктури (наприклад, обчислювальні ресурси, мережа та сховища) доповнюються інтегрованими сервісами платформи» [12].

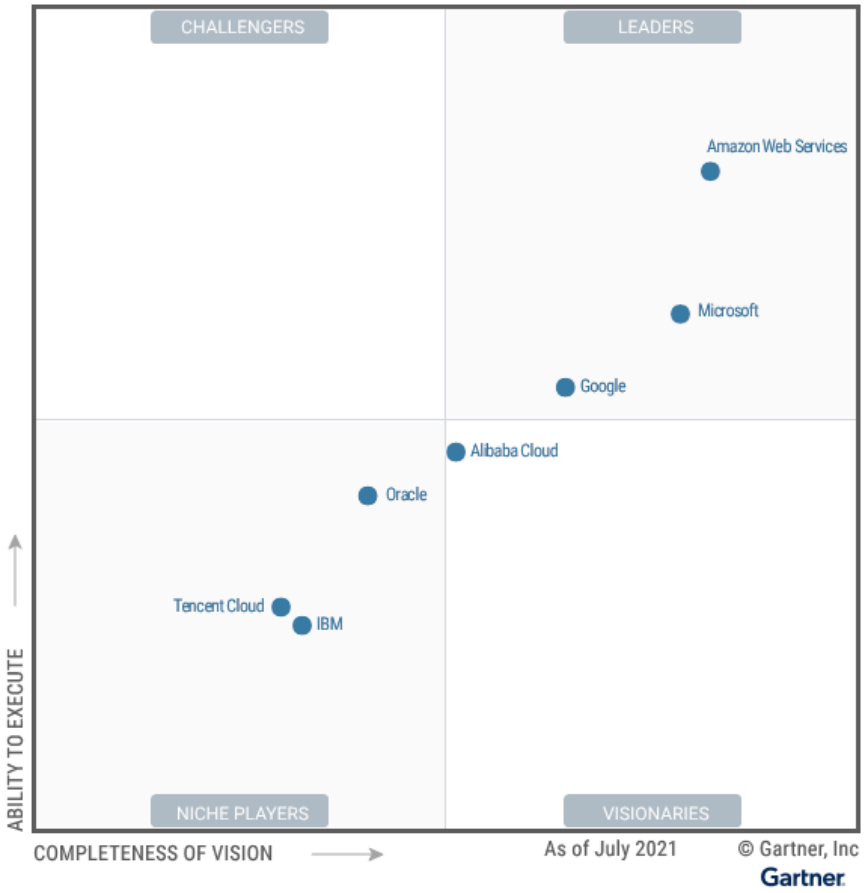

Рис. 2.1. Місце AWS за даними Gartner Research [12]

Додатки AWS для бізнесу – це масштабовані програми на базі AWS з оплатою в міру використання (рис. 2.2) [10].

Category	Service description	AWS service
Line of Business Applications	Easy to use omnichannel cloud contact center	Amazon Connect
	Multichannel marketing communications	Amazon Pinpoint
Productivity Applications	Build apps for managing your team's work with no coding	Amazon Honeycode
	Frustration-free meetings, video calls, and chat	Amazon Chime
	Secure enterprise document storage and sharing	Amazon WorkDocs
	Secure email and calendaring	Amazon WorkMail
	Empower your organization with Alexa	Alexa for Business
Communication Developer Services	Real-time messaging, audio, video, and screen sharing	Amazon Chime SDK
	High-scale inbound and outbound email	Amazon Simple Email Service (SES)
	Flexible mobile SMS and push notifications	Amazon Pinpoint APIs
	Cost-effective SIP trunking and advanced telephony features	Amazon Chime Voice Connector
	Secure file collaboration and management	Amazon WorkDocs SDK

Рис. 2.2. AWS Business Application Services [10]

Розглянемо деякі додатки AWS для бізнесу більш детальніше.

Amazon Connect забезпечує обслуговування клієнтів за менших витрат за допомогою простого використання багатоканального хмарного контактного центру. Користуючись Amazon Connect можна налаштувати контактний центр, який може масштабуватися для підтримки мільйонів клієнтів за лічені хвилини [10].

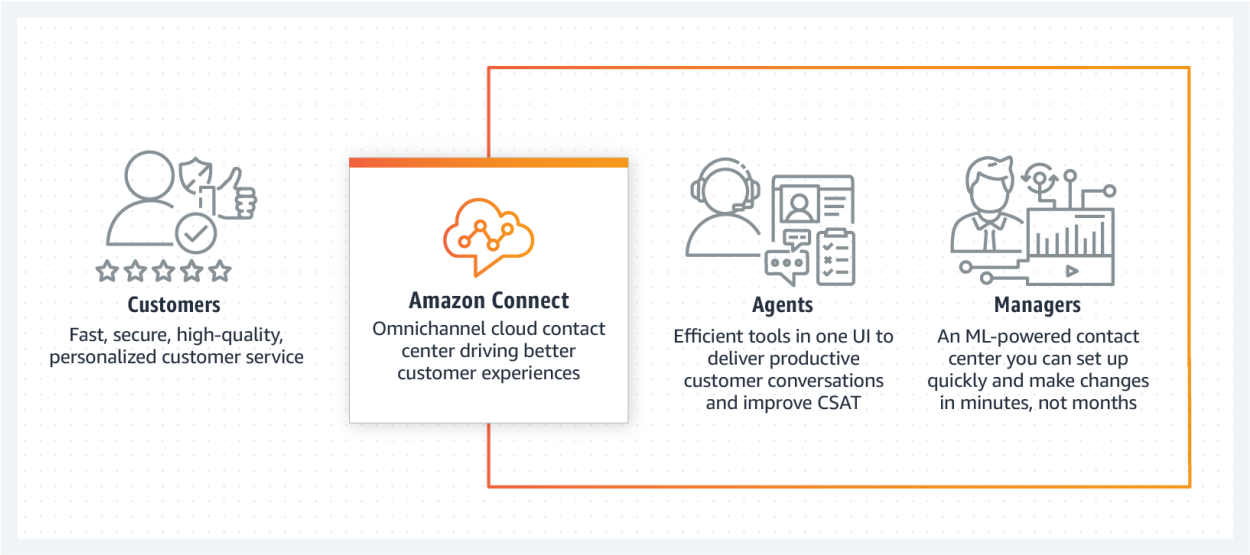

Рис. 2.3. Принцип роботи сервісу Amazon Connect [10]

Amazon Connect забезпечує створення високоякісного багатоканального голосового та інтерактивного чату для підтримки клієнтів у будь-якій локації. Також забезпечується використання інтуїтивного інтерфейсу користувача (UI) для маршрутизації контактів, створення черги та аналітики [10]. Завдяки вбудованому штучному інтелекту (AI) та машинному навчанню (ML) Amazon Connect з легкістю автоматизує взаємодію, розуміє настрій клієнтів, проводить автентифікацію абонентів та включає такі функції, як інтерактивна голосова відповідь (IVR) та чат-боти.

Даний сервіс мотивує агентів працювати більш активно та продуктивно. Є можливість переглядати універсальні профілі клієнтів та рекомендовані варіанти відповідей у режимі реального часу, а також відстежувати наступні завдання для швидкого вирішення клієнтських проблем [10].

Amazon Pinpoint – це гнучкий та масштабований сервіс для вхідних та вихідних маркетингових комунікацій. Він дозволяє взаємодіяти з клієнтами такими каналами, як електронна пошта, SMS, push повідомлення або голосовий зв’язок. Сервіс Amazon Pinpoint простий у налаштуванні та використанні, його можна гнучко підлаштовувати під будь-які сценарії маркетингової взаємодії [10].

Є можливість сегментації аудиторії компанії на кшталт клієнта та індивідуально налаштовувати повідомлення, наповнюючи їх актуальним змістом.

Метрики доставки та метрики компанії в Amazon Pinpoint виміряють дієвість взаємодії. Amazon Pinpoint може зростати разом із бізнесом і масштабуватися до мільярдів повідомлень на день по всіх каналах комунікації [10].

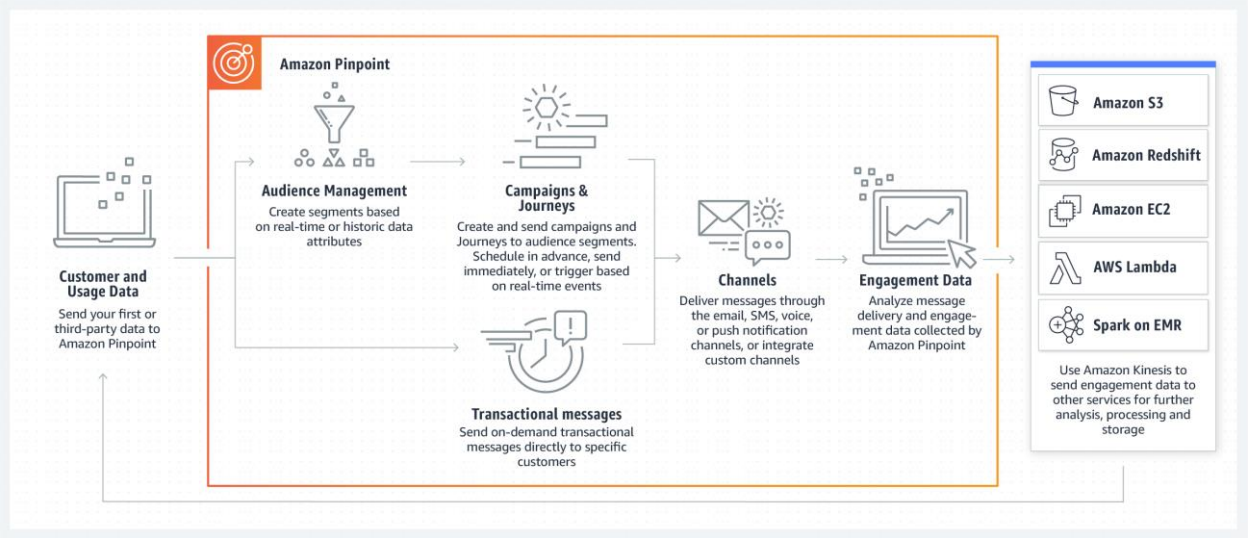

Рис. 2.4. Принцип роботи сервісу Amazon Pinpoint [10]

Amazon Chime – це сервіс зв'язку, який дозволяє організовувати наради, спілкуватися в чаті та здійснювати ділові дзвінки як всередині організації, так і за її межами в тому самому додатку [10].

Розробники можуть використовувати інфраструктуру зв'язку та сервіси, що застосовуються у Amazon Chime. Крім того, за допомогою пакету засобів розробки програмного забезпечення Amazon Chime SDK можна додавати функції демонстрації екрану, аудіо- та відеодзвінків безпосередньо до своїх програм.

Amazon Chime Voice Connector – це сервіс, за допомогою якого підприємства можуть переносити робочі навантаження телефонії в AWS. IT фахівці можуть використовувати сервіс Voice Connector для недорогого SIP транкінгу з локального середовища до хмарних телефонних систем [10].

Сервіс Voice Connector підтримує вхідні дзвінки, вихідні або ті, й інші одночасно. Крім того, розробники можуть використовувати сервіс Voice Connector для дзвінків PSTN (Public Switched Telephone Network) у своїх додатках за допомогою Amazon Chime SDK або для потокового аудіо для аналітики телефонних дзвінків та машинного навчання [10].

Amazon WorkDocs – це повністю керований безпечний сервіс для створення та зберігання контенту та спільної роботи з ним. Завдяки Amazon WorkDocs можна просто створювати та редагувати контент, обмінюватися ним, а також отримувати доступ до нього з будь-якого пристрою в будь-якому місці, оскільки контент централізовано зберігається в AWS [10].

Amazon WorkDocs спрощує спільну роботу з іншими користувачами, а також дозволяє зручно обмінюватися контентом, надавати детальні відгуки та спільно редагувати документи. За допомогою Amazon WorkDocs можна позбутися застарілої інфраструктури загального каталогу, перемістивши файлові ресурси, що спільно використовуються, в хмару. Amazon WorkDocs підтримує інтеграцію з існуючими системами та надає потужний API для розробки власних багатофункціональних програм. Весь контент корпоративних клієнтів є захищеним найбільшою хмарною інфраструктурою у світі, оскільки Amazon WorkDocs створений на основі AWS [10].

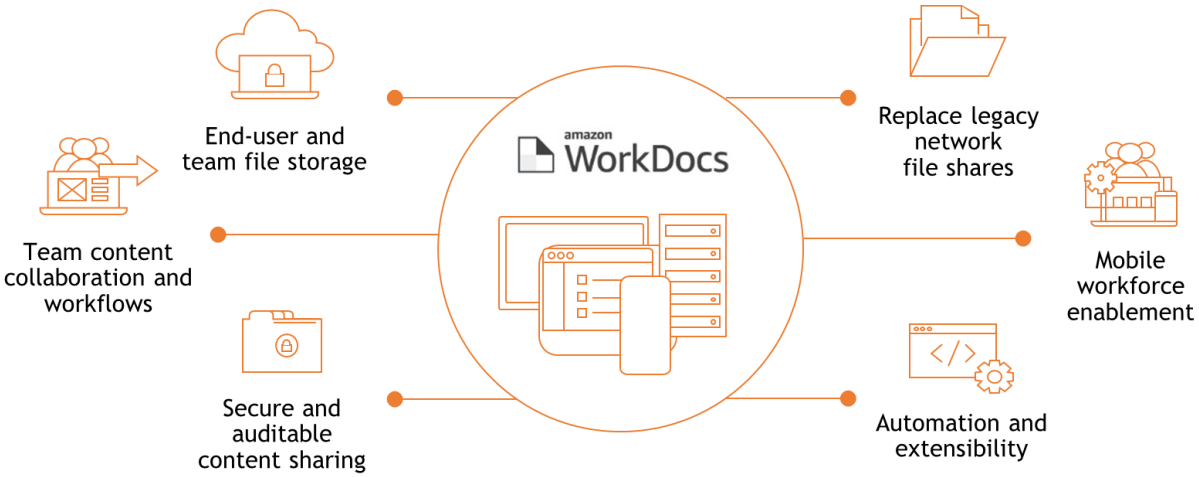

Рис. 2.5. Можливості сервісу Amazon WorkDocs [10]

Amazon WorkMail – це надійний керований сервіс для ділової електронної пошти та календарів з підтримкою існуючих поштових клієнтів для настільних комп’ютерів та мобільних поштових клієнтів [10].

Сервіс Amazon WorkMail дає користувачам можливість безперешкодно отримувати доступ до своєї електронної пошти, контактів та календарів за допомогою зручного поштового клієнта, включаючи Microsoft Outlook, програми

електронної пошти для iOS та Android, а також будь-яких поштових клієнтів, які підтримують протокол IMAP, а також безпосередньо через веб-браузер [10].

Сервіс Amazon WorkMail можна інтегрувати з існуючим корпоративним каталогом, використовувати журнали електронної пошти для забезпечення відповідності вимогам та керувати як ключами, за допомогою яких відбувається шифрування даних, так і місцезнаходженням, де зберігаються ці дані. Крім того, можна налаштувати сумісність із Microsoft Exchange Server та програмно керувати користувачами, групами та ресурсами за допомогою SDK Amazon WorkMail [10].

2.2. Призначення, можливості та функції AWS Web Application Firewall

AWS WAF – це брандмауер для інтернет-додатків, який дозволяє захистити їх (або різні API) від поширених мережевих експлойтів та ботів, здатних вплинути на доступність, створити загрозу безпеці або задіяти надмірну кількість ресурсів API або додатку [11].

Принцип роботи AWS WAF показано на рис. 2.6. AWS WAF надає повний контроль над тим, як саме трафік досягатиме додатків користувача: він зможе створювати правила безпеки, які контролюватимуть трафік від ботів і блокуватимуть такі виконувані за поширеними шаблонами атаки, як SQL-ін’єкції або міжсайтовий скріптинг.

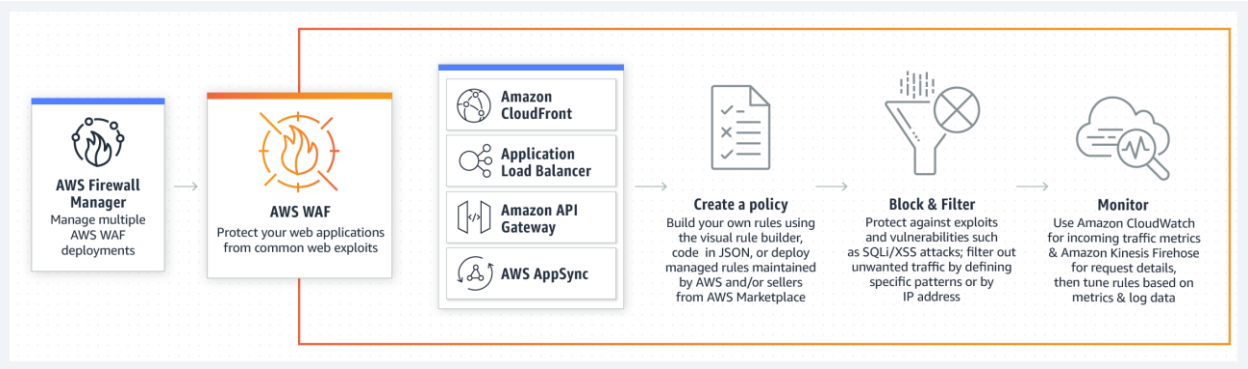

Рис. 2.6. Принцип роботи AWS WAF [11]

Також можна настроювати правила фільтрації для певних шаблонів трафіку.

З даним рішенням можна швидко розпочати роботу, використовуючи керовані правила для AWS WAF, попередньо налаштований набір правил, керованих з боку AWS або продавців AWS Marketplace, для вирішення таких проблем, як 10 основних ризиків безпеки OWASP та автоматизовані боти, які споживають зайві ресурси, відхиляють метрики або призводять до простоїв. Ці правила регулярно оновлюються у міру виникнення нових проблем [11].

AWS WAF пропонує повнофункціональний API, що дозволяє автоматизувати процеси створення, розгортання та обслуговування правил безпеки. Виконати розгортання AWS WAF можна у сервісі Amazon CloudFront у вигляді компонента рішення CDN, на Application Load Balancer, розташованому перед веб-серверами або серверами джерела, що працюють на EC2, на Amazon API Gateway для використовуваних API REST або на AWS AppSync для API GraphQL [11].

Розглянемо можливості AWS WAF.

Адаптивний захист від інтернет атак. Введення в побут і оновлення правил AWS WAF займає менше хвилини, завдяки чому у разі виникнення нової загрози захист завжди залишатиметься на актуальному рівні в рамках усього робочого середовища.

WAF підтримує сотні різних правил, здатних виконати аналіз будь-якої частини веб-запиту з мінімальним впливом на швидкість обробки трафіку. AWS WAF захищає веб-додатки від атак за рахунок фільтрації трафіку на підставі створених правил. Наприклад, можна відфільтрувати будь-яку частину запиту: IP-адресу, заголовки HTTP, тіло повідомлення HTTP або рядки з ідентифікаторами URI. Завдяки цьому стає можливим блокування виконуваних за поширеними шаблонами атак: SQL-ін’єкцій або міжсайтового скриптингу [11].

Керовані правила значно заощаджують час користувача. За допомогою керованих правил для AWS WAF користувач зможе оперативно розпочати роботу та захистити свої інтернет-програми або API від найбільш поширених загроз. На вибір надається безліч типів правил, здатних впоратися з проблемами зі списку десяти головних загроз безпеки за версією Open Web Application Security Project

(OWASP), загрозами, характерними для систем керування контентом (CMS), або зі все більш поширеними загрозами із загального переліку. вразливостей та ризиків (CVE). Керовані правила автоматично оновлюються в міру виникнення нових проблем, так що користувач зможе витратити свій час, що звільнився, на розробку програми, а не на занепокоєння про його захист [11].

Поліпшена можливість відстеження Інтернет-трафіку. AWS WAF дозволяє відстежувати Інтернет трафік у режимі, близькому до реального часу, завдяки чому можна оперативно створювати нові правила або попередження в Amazon CloudWatch. Користувачеві доступні точні засоби контролю над створенням метрик, що дозволяє відстежувати весь трафік, що передається, навіть на рівні правила.

Крім того, система AWS WAF автоматично і докладно заносить у журнал усі події, що відбуваються, протоколюючи дані заголовків кожного дослідженого запиту для використання цих відомостей в автоматизації засобів забезпечення безпеки, проведення аналітики або аудиту [11].

Просте розгортання та обслуговування. Розгортання AWS WAF виконується дуже просто і допомагає забезпечити захист додатків, розгортання яких виконано на Amazon CloudFront у вигляді компонента рішення CDN, на Application Load Balancer, розташованому перед серверами джерела, на Amazon API Gateway для використовуваних API REST або на AWS AppSync для API GraphQL.

Для функціонування системи не треба розгортати додаткове програмне забезпечення, налаштовувати конфігурацію DNS, займатися управлінням сертифікатів SSL/TLS або створювати зворотний проксі-сервер. Завдяки інтеграції з AWS Firewall Manager правила можна задавати та контролювати централізовано, а також використовувати їх повторно у всіх веб-додатках, які потрібно захистити [11].

Простота відстеження, блокування та обмеження трафіку ботів. Завдяки функції AWS WAF для контролю ботів забезпечується видимість і контроль над трафіком роботів, спрямованим до окремих веб-додатків.

За допомогою консолі AWS WAF є можливість відстежувати звичайні боти, такі як монітори стану та пошукові движки, а також отримувати докладну інформацію про категорію, ідентифікацію та інші характеристики трафіку ботів. Також можна блокувати або обмежувати трафік від ботів, таких як скрепери, аналізатори та сканери.

За допомогою AWS Firewall Manager можна розгорнути групу правил під управлінням функції контролю ботів у кількох облікових записах своєї організації AWS Organization [11].

Інтегрований захист під час розробки програми. Кожну з можливостей AWS WAF можна налаштувати за допомогою API сервісу AWS WAF або Консолі керування AWS. Це дозволяє командам DevOps задавати правила, що підвищують рівень безпеки окремих програм вже на стадії розробки. Засоби мережної безпеки можна буде впроваджувати на різних етапах процесу розробки: надати їх розробнику, який спочатку пише код, фахівцю DevOps, що виконує розгортання програми, або професіоналам в галузі безпеки, що задають зведення правил для організації [11].

2.3. Архітектура та порядок функціонування AWS Web Application Firewall

AWS WAF – це брандмауер веб-додатків, який дозволяє відстежувати запити HTTP і HTTPs, які пересилаються в дистрибутив Amazon CloudFront, REST API шлюзу Amazon API, балансувальник навантаження додатків або API AWS AppSync GraphQL. AWS WAF також дозволяє контролювати доступ до вмісту організації. На основі вказаних вами умов, таких як IP-адреси, з яких походять запити, або значення рядків запиту, Amazon CloudFront, Amazon API Gateway, Application Load Balancer або AWS AppSync відповідає на запити або за допомогою запитуваного вмісту, або за допомогою HTTP 403. код статусу (заборонено). Також можна налаштувати CloudFront, щоб повертати користувацьку сторінку помилки, коли запит заблоковано [13].

На найпростішому рівні AWS WAF дозволяє вибрати один з таких способів поведінки [13]:

дозволити всі запити, крім тих, які вказано – це корисно, якщо необхідно, щоб Amazon CloudFront, Amazon API Gateway, Application Load Balancer або AWS AppSync обслуговували вміст. для загальнодоступного веб-сайту, але є необхідність заблокувати запити від зловмисників;

блокувати всі запити, окрім тих, які вказано – це корисно, якщо необхідно надавати вміст для обмеженого веб-сайту, користувачів якого легко ідентифікувати за властивостями у веб-запитах, наприклад за IP-адресами, які вони використовують для перегляду веб-сайту;

підраховувати запити, які відповідають вказаним властивостям. Якщо необхідно дозволити або заблокувати запити на основі нових властивостей у веб-запитах, спочатку можна налаштувати AWS WAF на підрахунок запитів, які відповідають цим властивостям, не дозволяючи чи блокуючи ці запити. Це дає змогу підтвердити, що випадково не було налаштувано AWS WAF на блокування всього трафіку на конкретних веб-сайт організації. Якщо є впевненість, що вказано правильні властивості, то можна змінити поведінку, щоб дозволити або заблокувати запити.

Використання AWS WAF має кілька переваг, а саме [13]:

додатковий захист від веб-атак за допомогою умов, які було вказано. Можна визначити умови, використовуючи такі характеристики веб-запитів, як: IP-адреси, з яких походять запити; країна, з якої надходять запити; значення в заголовках запитів; рядки, які з’являються у запитах, або конкретні рядки, або рядки, які відповідають шаблонам регулярних виразів; тривалість запитів; наявність коду SQL, який може бути шкідливим (відомий як ін’єкція SQL); наявність сценарію, який імовірно є шкідливим (відомий як міжсайтовий скриіпт);

правила, які можуть дозволяти, блокувати або підраховувати веб-запити, які відповідають заданим умовам. Крім того, правила можуть блокувати або підраховувати веб-запити, які не тільки відповідають зазначеним умовам, але й перевищують визначену кількість запитів за будь-який 5-хвилинний період;

правила, які можна повторно використовувати для кількох веб-додатків;

керовані групи правил від продавців AWS і AWS Marketplace;

показники в режимі реального часу та вибіркові веб-запити;

автоматизоване адміністрування за допомогою AWS WAF API.

Розглянемо порядок роботи рішення AWS WAF. AWS WAF застосовується для того, щоб керувати тим, як дистрибутив Amazon CloudFront, REST API Amazon API Gateway, балансувальник навантаження додатків або API AWS AppSync GraphQL реагують на веб-запити HTTP(s).

Web ACL – використовується список керування доступом до Інтернету (ACL) для захисту набору ресурсів AWS. Створюється веб-список керування доступом та визначається його стратегія захисту, додаючи правила. Правила визначають критерії перевірки веб-запитів і вказують, як обробляти запити, які відповідають критеріям. Встановлюється дія за замовчуванням для веб-ACL, яка вказує, блокувати чи дозволяти ці запити, які проходять перевірку правил.

Правила – кожне правило містить заяву, яка визначає критерії перевірки та дії, які необхідно виконати, якщо веб-запит відповідає критеріям. Коли веб-запит відповідає критеріям, це збігається. Правила використовуються, щоб блокувати відповідні запити або дозволяти відповідні запити. Можна використовувати правила лише для підрахунку відповідних запитів.

Групи правил – правила можна використовувати окремо або в групах правил для повторного використання. Керовані правила AWS і продавці AWS Marketplace надають групи керованих правил. Також можна визначити власні групи правил.

Після того як буде створено свій веб-список ACL, можна його пов’язати з одним або кількома ресурсами AWS.

Типи ресурсів, які можна захистити за допомогою веб-ACL AWS WAF: дистрибутив Amazon CloudFront, REST API шлюз Amazon API, балансувальник навантаження додатків і API AWS AppSync GraphQL.

AWS WAF доступний у регіонах, перелічених на кінцевих точках служби AWS.

Для REST API шлюзу Amazon API, балансувальника навантаження додатків або API AWS AppSync GraphQL можна використовувати будь-який із регіонів у списку.

Для розповсюдження CloudFront AWS WAF доступний у всьому світі, але для всієї роботи потрібно використовувати регіон Схід США (Северна Вірджинія). Необхідно створити свій веб-ACL, використовуючи регіон США Схід (Северна Вірджинія). Також необхідно використовувати цей регіон для створення будь-яких інших ресурсів, які використовуються у веб-ACL, як-от групи правил, набори IP-адресів і набори шаблонів регулярних виразів.

AWS WAF – це брандмауер веб-додатків, який допомагає захистити веб-додатки від поширених веб-експлойтів, які можуть вплинути на доступність додатків, поставити під загрозу безпеку або споживати надмірні ресурси. AWS WAF дозволяє клієнтам визначати налаштовані правила веб-безпеки, дозволяючи їм контролювати, який трафік дозволяти або блокувати для веб-додатків та API, розгорнутих у Amazon CloudFront, Application Load Balancer або API Gateway.

Налаштування правил WAF може бути складним і обтяжливим для великих і малих організацій, особливо для тих, хто не має виділених груп безпеки. Щоб спростити цей процес, AWS пропонує рішення AWS WAF Security Automations, яке автоматично розгортає єдиний список керування доступом до мережі (ACL) (рис. 2.7) з набором правил AWS WAF, призначених для фільтрації найпоширеніших мережевих атак. Під час початкового налаштування шаблону AWS CloudFormation цього рішення користувачі вказують, які захисні функції слід увімкнути, як показано нижче. Після розгортання цього рішення AWS WAF почне перевіряти веб-запити до своїх існуючих дистрибутивів CloudFront або Application Load Balancer та блокувати їх за потреби.

Необхідно відміти, що веб-застосунки вразливі для безлічі атак. Ці атаки включають спеціально створені запити, призначені для використання вразливості або отримання контролю над сервером, об’ємні атаки, спрямовані на закриття веб-сайту або погані боти та парсери, запрограмовані для очищення та крадіжки веб-контенту [14].

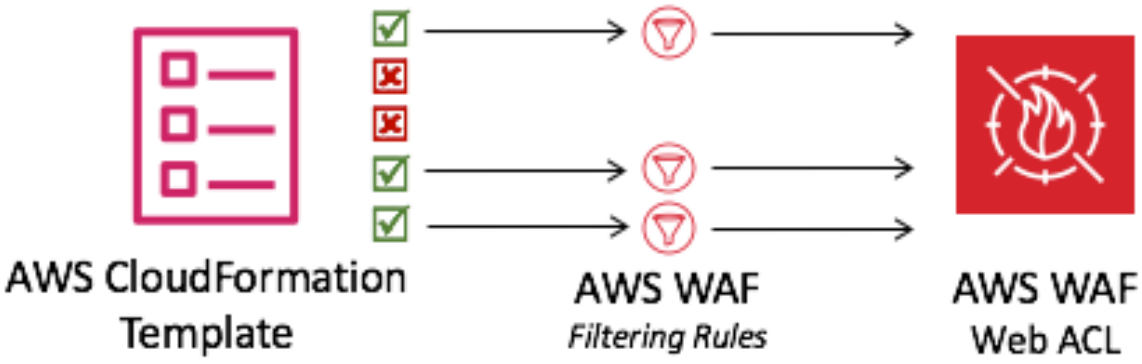

Рис. 2.7. Конфігурація веб-ACL AWS WAF [14]

Це рішення використовує AWS CloudFormation для швидкого та простого настроювання правил AWS WAF, які допомагають блокувати наступні поширені атаки: SQL-ін'єкція: зловмисники вставляють шкідливий код SQL у веб-запити, щоб отримати дані з корпоративної бази даних. Це рішення призначене для блокування веб-запитів, які містять потенційно шкідливий код SQL.

Міжсайтовий скріптинг: також відомий як XSS, зловмисники використовують вразливості на безпечному веб-сайті як засіб для впровадження шкідливих скріптів клієнт-сайт у веб-браузер легітимного користувача. Це рішення призначене для перевірки часто досліджуваних елементів запитів з метою виявлення та блокування XSS-атак.

HTTP-флуди: веб-сервери та інші внутрішні ресурси схильні до ризику розподілених атак типу «відмова в обслуговуванні» (DDoS), таких як HTTP-флуди. Це рішення автоматично запускає правило на основі швидкості, коли веб-запити від клієнта перевищують поріг, що настроюється. Як альтернативу можна посилити цей поріг, обробивши журнали AWS WAF за допомогою функції AWS Lambda або запиту Amazon Athena.

Сканери та зонди: шкідливі джерела сканують та досліджують веб-застосунки, підключені до Інтернету, на предмет наявності вразливостей. Вони надсилають серію запитів, які генерують коди помилок HTTP 4xx, існує можливість використовувати цю історію, щоб допомогти ідентифікувати та заблокувати шкідливі вихідні IP-адреси. Це рішення створює функцію AWS

Lambda або запит Amazon Athena, який автоматично аналізує журнали доступу Amazon CloudFront або Application Load Balancer, підраховує кількість невірних запитів від унікальних вихідних IP-адрес за хвилину.

Відоме походження зловмисників (списки репутації IP-адрес). Ряд організацій підтримують списки репутації IP-адрес, якими користуються відомі зловмисники, такі як спамери, розповсюджувачі шкідливих програм та бот-мережі. Це рішення використовує інформацію з цих списків репутації, щоб допомогти блокувати запити з шкідливих IP-адрес.

Боти та парсери: оператори загальнодоступних веб-додатків повинні бути впевнені, що клієнти, які звертаються до їхнього контенту, ідентифікують себе точно і використовуватимуть служби за призначенням. Однак деякі автоматизовані клієнти, такі як парсери контенту або погані боти, вводять себе в оману, щоб уникнути обмеження. Це рішення допомагає виявляти та блокувати поганих ботів та парсерів.

Розглянемо архітектуру AWS WAF (рис. 2.8). При розгортанні цього рішення з параметрами за промовчанням у хмарі AWS створюється середовище, яке показано на рис. 2.8.

В основі дизайну лежить *веб-список контролю доступу* AWS WAF, який діє як центральна точка перевірки та прийняття рішень для всіх запитів до веб-додатку. Під час початкового налаштування стека AWS CloudFormation користувач визначає, які компоненти захисту слід активувати. Кожен компонент працює незалежно і додає різні правила до веб-ACL [14].

Компоненти цього рішення можна згрупувати за такими напрямами захисту (рис. 2.8) [14]:

ручні списки IP-адрес (A і B): цей компонент створює два конкретні правила AWS WAF, які дозволяють вручну вставляти IP-адреси, які треба дозволити або заборонити.

SQL-ін'єкція (C) та XSS (D): це рішення налаштовує два власних правила AWS WAF, які призначені для захисту від поширених шаблонів SQL-ін'єкцій або міжсайтових сценаріїв (XSS) в URI, рядку запиту або тілі запиту.

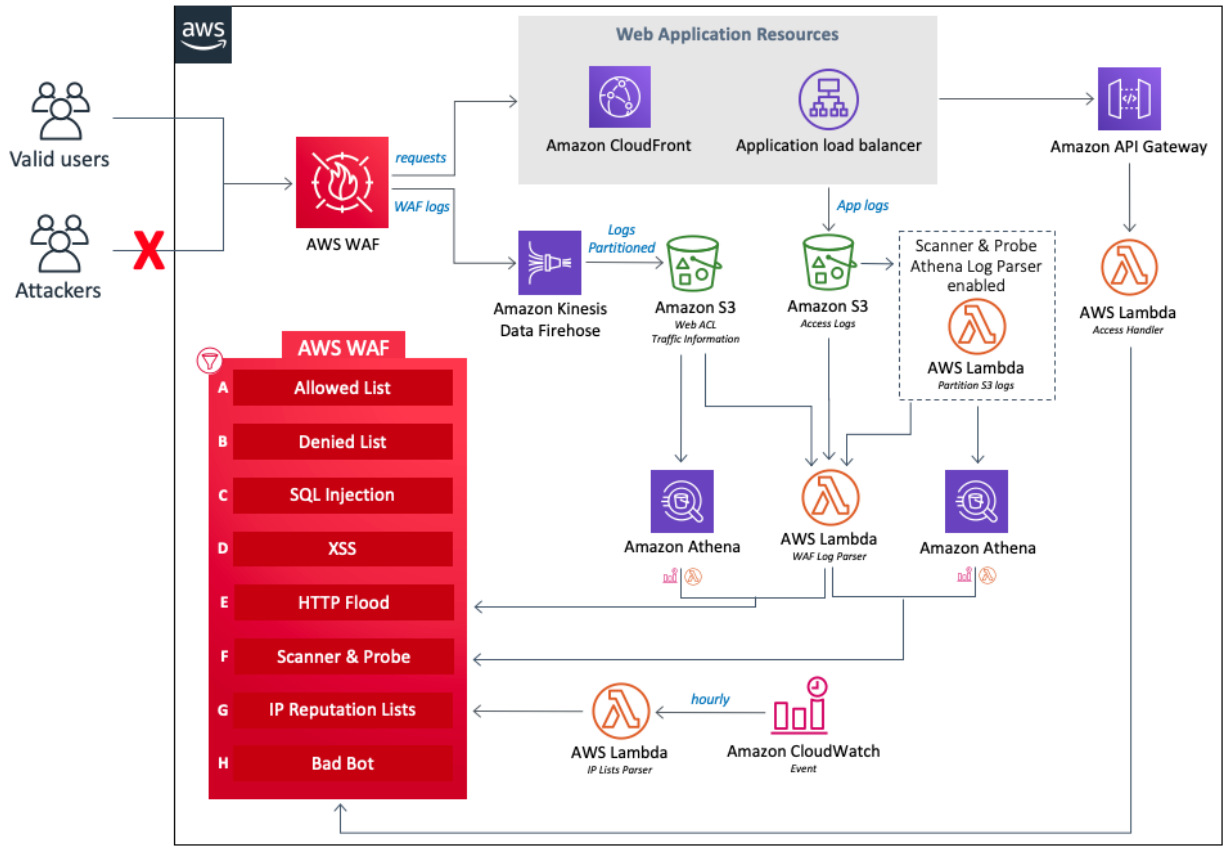

Рис. 2.8. Архітектура AWS WAF Security Automations на AWS [14]

HTTP-флуд (E): цей компонент захищає від атак, що складаються з великої кількості запитів з певної IP-адреси, таких як DDoS-атака на веб-рівні або спроба входу до системи методом грубої сили. За допомогою цього правила ви встановлюєте поріг, який визначає максимальну кількість вхідних запитів, дозволених з однієї IP-адреси протягом п'ятихвилинного періоду. При перевищенні цього порога додаткові запити IP-адреси тимчасово блокуються. Це правило можна реалізувати, використовуючи правило AWS WAF на основі швидкості або обробляючи журнали AWS WAF за допомогою функції AWS Lambda або запиту Amazon Athena.

Сканери та зонди (F): цей компонент аналізує журнали доступу до програм у пошуках підозрілої поведінки, такої як аномальна кількість помилок, згенерованих джерелом. Потім блокує ці підозрілі вихідні IP-адреси на певний користувачем період часу. Це правило можна реалізувати за допомогою AWS Lambda або запиту Amazon Athena.

Списки репутації IP-адрес (G): цей компонент являє собою IP Lists Parser функцію AWS Lambda, яка щогодини перевіряє сторонні списки репутації IP-адрес щодо нових діапазонів для блокування. До цих списків входить Spamhaus Списки Don't Route Or Peer (DROP) та Extended DROP (EDROP), список IP-адрес Proofpoint Emerging Threats, і список вихідних вузлів Tor.

Погані боти (H): цей компонент автоматично встановлює пастку, яка є механізмом безпеки, призначений для заманювання та відображення спроби атаки. Приманка цього рішення – це кінцева точка пастки, яку можна вставити на свій веб-сайт для виявлення запитів, що входять від парсерів контенту і поганих ботів. Якщо джерело звертається до приманки, Access Handler функція AWS Lambda перехопить і перевірить запит, щоб отримати його IP-адресу, а потім додасть його до списку блокування AWS WAF.

Рішення AWS WAF Security Automations призначене для захисту веб-застосунків та API-інтерфейсів, розгорнутих за допомогою Amazon CloudFront, Application Load Balancer або Amazon API Gateway. Розглянемо ці рішення більш детально.

AWS Lambda – це безсерверний, керований подіями обчислювальний сервіс, який дозволяє виконувати код практично для необмеженого типу програми або сервісу без надання серверів та їх обслуговування. Є можливість включення Lambda з понад 200 сервісів і програм, що надаються за моделлю ПЗ як послуга (SaaS). AWS Lambda з іншими сервісами AWS дозволяють створювати веб-додатки, які автоматично масштабуються в потрібному напрямку та працюють у конфігурації з високою доступністю у кількох центрах обробки даних [16].

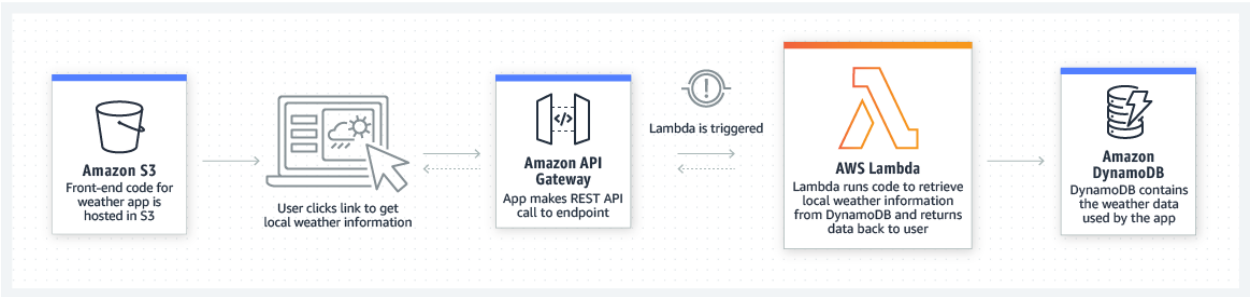

Рис. 2.9. Принцип роботи AWS Lambda [16]

Amazon Simple Storage Service (Amazon S3) – це сервіс зберігання об’єктів, що пропонує найкращі в галузі показники продуктивності, масштабованості, доступності та безпеки даних [18]. Клієнтами можуть бути компанії будь-яких розмірів та з будь-яких сфер діяльності. Вони можуть використовувати даний сервіс для зберігання та захисту будь-яких обсягів даних у різних ситуаціях, наприклад, для забезпечення роботи озер даних, сайтів, мобільних додатків, для резервного копіювання та відновлення, архівації, корпоративних додатків, пристроїв IoT та аналізу великих даних.

Amazon S3 пропонує прості у використанні інструменти адміністрування, які дозволяють організувати дані та точно налаштувати обмеження доступу відповідно до потреб вашого бізнесу або законодавчих вимог. Amazon S3 забезпечує надійність 99,999999999% (тут 11 дев’яток) і зберігає дані мільйонів додатків на користь компаній з усього світу [18].

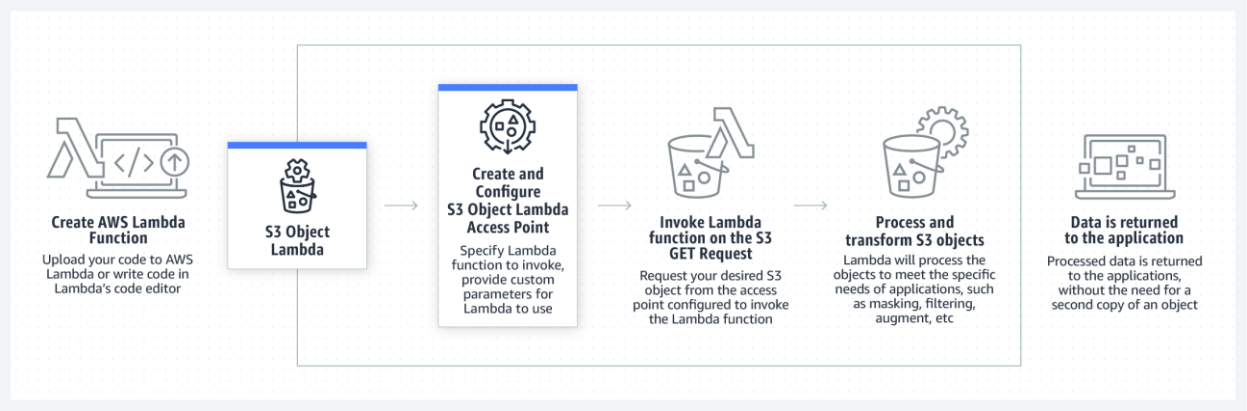

Рис. 2.10. Принцип роботи Amazon S3 [18]

За допомогою функції S3 Object Lambda можна додавати власний код у запити S3 GET, щоб змінювати та обробляти дані, що повертаються до програми. Вперше можна застосувати код користувача для зміни даних, що повертаються стандартними запитами S3 GET, для фільтрування рядків, динамічної зміни розміру зображень, видалення конфіденційних даних та багато іншого. Виконання кодів на базі функцій AWS Lambda здійснюється в інфраструктурі, що повністю керується AWS, що дає можливість виключити необхідність створення та зберігання похідних копій даних або запуску дорогих проксі, при цьому без будь-

яких змін у додатках.

Amazon Athena – це інтерактивний сервіс запитів, що дозволяє просто аналізувати дані у сховищі Amazon S3 за допомогою стандартного SQL. Athena – це безсерверний сервіс, тому не потрібно керувати архітектурою, а платня нараховується лише за виконані запити [19].

Сервіс Athena дуже простий у використанні. Необхідно вказати дані в Amazon S3, задати схему та виконати запити, використовуючи стандартні засоби SQL. Більшість результатів буде одержано протягом секунд. Для підготовки даних до аналізу не потрібно використовувати складні завдання ETL. Таким чином, будь-який спеціаліст із знанням SQL може швидко проаналізувати великий обсяг даних [19].

В Athena передбачена інтеграція з каталогом даних AWS Glue. Це дозволяє створювати єдиний репозиторій метаданих різних сервісів, сканувати джерела даних виявлення схем і наповнювати каталог новими чи зміненими таблицями і визначеннями розділів, і навіть забезпечувати версійність схем.

Amazon CloudWatch – це сервіс моніторингу та спостереження для інженерів DevOps, розробників, інженерів з надійності сайтів та IT-менеджерів. CloudWatch надає дані та дієві аналітичні відомості для моніторингу додатків, реагування на зміни продуктивності в масштабах системи, оптимізації використання ресурсів та отримання єдиного уявлення про працездатність системи [20].

CloudWatch збирає дані моніторингу та операційні дані у вигляді журналів, метрик та подій, допомагаючи отримати єдине представлення додатків, сервісів та ресурсів AWS, що працюють на платформі AWS, а також у локальному середовищі. За допомогою CloudWatch можна виявляти аномальну поведінку у своїх середовищах, налаштовувати попередження, створювати загальні візуальні уявлення журналів та метрик, виконувати автоматизовані дії, усувати неполадки, а також дізнаватися корисні відомості, які допоможуть підтримувати стабільну роботу додатків [20].

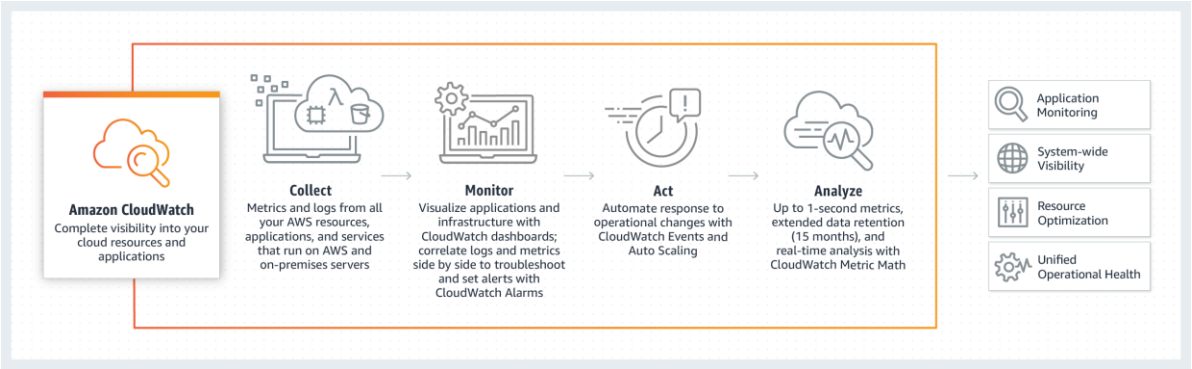

Рис. 2.11. Принцип роботи Amazon CloudWatch [20]

CloudWatch збирає дані моніторингу та операційні дані у вигляді журналів, метрик та подій, а також візуалізує їх за допомогою автоматизованих панелей управління, допомагаючи отримати єдине уявлення про програми, сервіси та ресурси AWS, що працюють на платформі AWS, а також у локальному середовищі. Є можливість порівнювати метрики та журнали, щоб отримати повніше уявлення про працездатність та ефективність використання ресурсів. Є можливість створювати попередження про перевищення зазначених порогових значень метрик або аномальну поведінку в метриках, використовуючи алгоритми машинного навчання. Щоб забезпечити швидке реагування необхідно налаштувати автоматичні дії, які будуть повідомляти вам про попередження, що виникають, і дозволять скоротити середній час усунення проблем шляхом, наприклад, запуску автоматичного масштабування. Є можливість детально аналізувати метрики, журнали та маршрути, щоб краще зрозуміти, як підвищити продуктивність додатків [20].

Amazon CloudFront – це сервіс мережі доставки контенту (CDN), створений для високої продуктивності, безпеки та зручності розробників [21].

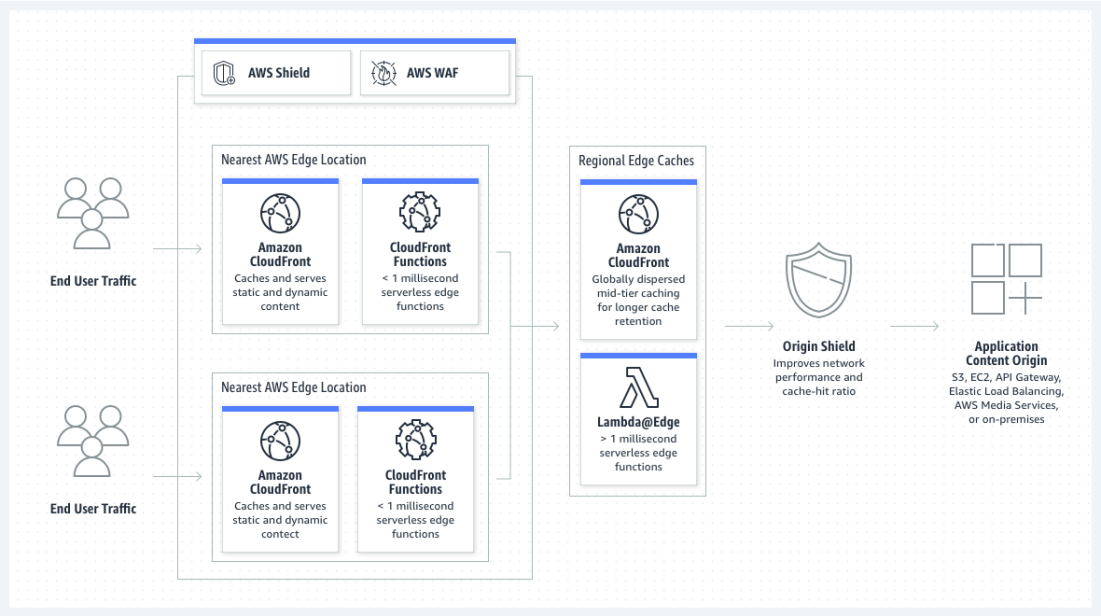

Рис. 2.12. Принцип роботи Amazon CloudFront [21]

AWS CloudFormation дозволяє моделювати та розподіляти ресурси AWS та сторонні ресурси, а також керувати ними завдяки формату «інфраструктура як код» [22].

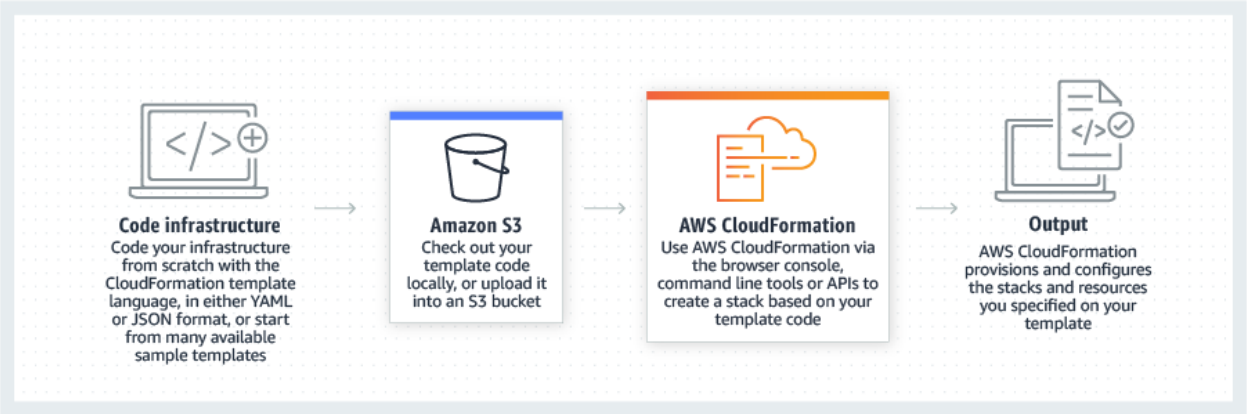

Рис. 2.13. Принцип роботи AWS CloudFormation [22]

Elastic Load Balancing автоматично розподіляє вхідний трафік додатків за кількома цільовими об’єктами та віртуальними пристроями в одній або декількох зонах доступності (AZ) .

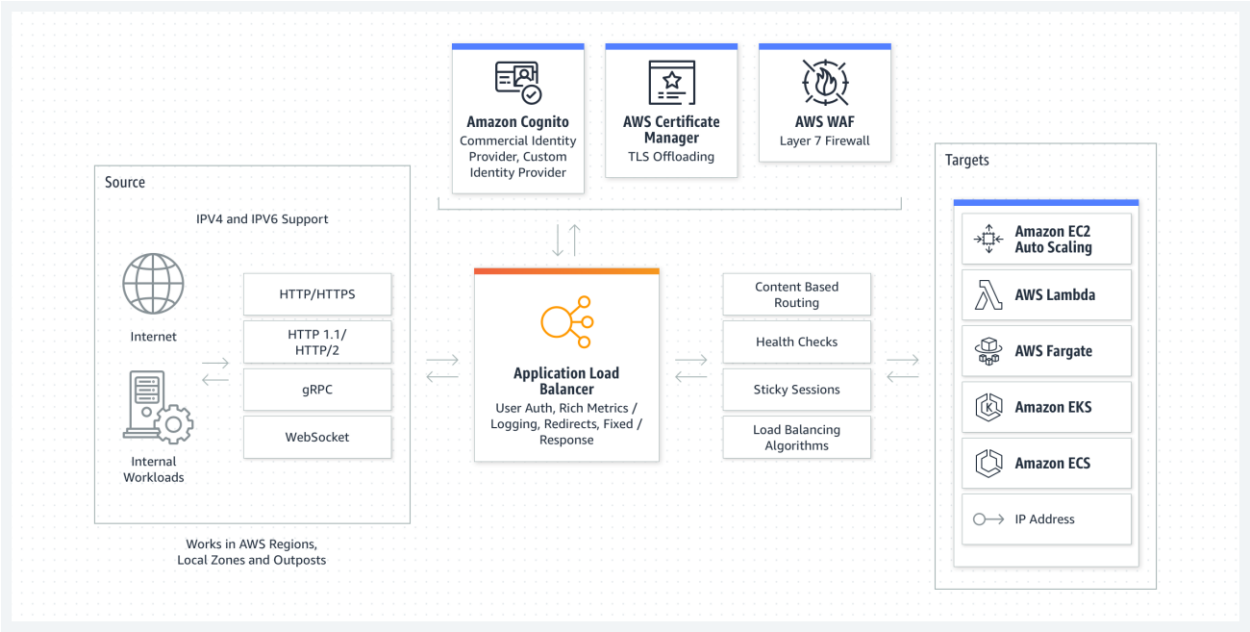

Рис. 2.14. Принцип роботи Elastic Load Balancing [23]

2.4. Призначення, можливості та функції рішення AWS WAF Bot Control

AWS WAF Bot Control забезпечує видимість та контроль над загальним та повсюдним трафіком ботів, який може споживати надлишкові ресурси, спотворювати показники, викликати простої або виконувати інші небажані дії. Є можливість використовувати групу керованих правил управління ботами, щоб заблокувати або обмежити швидкість повсюдних ботів, таких як парсери та сканери, або можна дозволити використання звичайних ботів, таких як монітори стану та пошукові системи. Група керованих правил AWS WAF Bot Control може використовуватися разом з іншими керованими правилами для WAF або з вашими власними правилами WAF для захисту ваших програм.

AWS WAF Bot Control дозволяє відстежувати активність трафіку ботів за допомогою панелей моніторингу, які надають докладну інформацію в режимі реального часу про категорії ботів, ідентифікатори та інші деталі трафіку ботів. Існує можливість використовувати AWS Firewall Manager для розгортання AWS WAF Bot Control для своїх веб-застосунків у кількох облікових записах в організації AWS [15].

Розглянемо переваги застосування AWS WAF Bot Control.

По-перше, надається видимість активності бот-трафіку. Всі клієнти AWS WAF отримують готові інформаційні панелі, які показують, які веб-додатки мають високий рівень активності ботів на основі вибіркових даних. Для клієнтів, які включили керування ботами, надається детальна інформація про дії ботів в режимі реального часу і на рівні запитів.

По-друге, знижуються експлуатаційні витрати та витрати на інфраструктуру. AWS WAF Bot Control допомагає скоротити витрати, пов'язані з веб-трафіком парсерів, сканерів та ботів. AWS WAF Bot Control блокує небажаний трафік ботів на периферії, перш ніж це може збільшити витрати на обробку додатків або вплинути на продуктивність додатку. AWS WAF Bot Control пропонує рівень безкоштовного використання для найпоширеніших випадків використання.

По-третє, AWS WAF Bot Control легко розгортається. Управління ботами включається шляхом додавання групи правил, керованої AWS, до списку управління доступом до Інтернету, що спрощує додавання захисту від ботів для корпоративних веб-додатків, що використовують Amazon CloudFront, Application Load Balancer, Amazon API Gateway або AWS AppSync. Немає потреби в додатковій інфраструктурі, змінах DNS або керуванні сертифікатами TLS [15].

По-четверте, економить час за допомогою керованого захисту від ботів. AWS WAF Bot Control – це керована група правил, яка підтримується та постійно вдосконалюється AWS. AWS WAF Bot Control усуває недиференційовану важку роботу і непотрібну складність моніторингу та захисту веб-додатків від ландшафтів ботів, що постійно змінюються.

По-п'яте, забезпечує гнучкий і настроюваний захист від ботів. AWS WAF Bot Control можна увімкнути без додаткового налаштування для більшості випадків використання, але його також можна легко налаштувати відповідно до конкретних вимог. Є можливість вказати, які запити обробляє AWS WAF Bot Control, різні дії для різних категорій ботів або об'єднати результати AWS WAF Bot Control з правилами WAF, що налаштовуються, щоб дозволити або

заблокувати певних ботів.

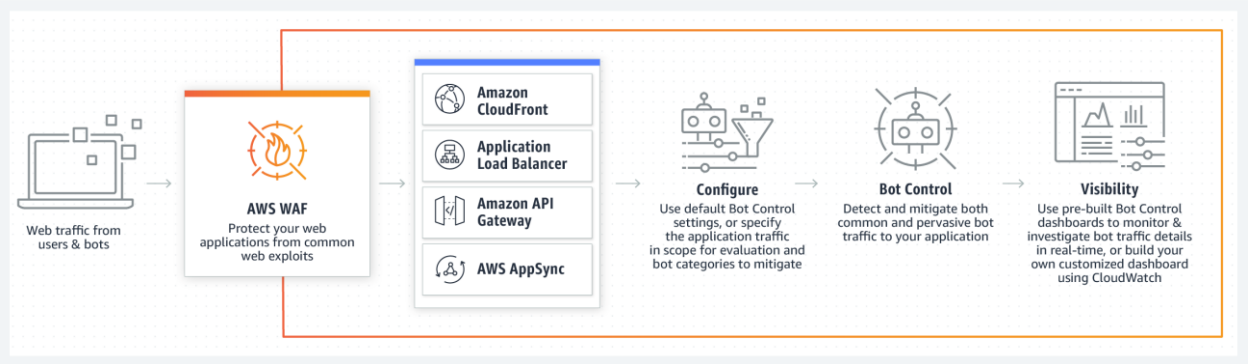

Рис. 2.15. Принцип роботи рішення AWS WAF Bot Control [15]

Випадки застосування AWS WAF Bot Control [15]:

для блокування небажаного трафіку ботів на межі мережі;

для захисту інтелектуальної власності;

для доставки альтернативного контенту у відповідь на трафік ботів.

AWS WAF Bot Control може блокувати небажаний бот-трафік на кордоні мережі під час використання AWS WAF із Amazon CloudFront. AWS WAF Bot Control допомагає мінімізувати вплив ботів на продуктивність корпоративного веб-додатку та може знизити експлуатаційні витрати на інфраструктуру, пов’язані з трафіком від парсерів, сканерів та пошукових ботів. AWS WAF Bot Control також підвищує точність веб-аналітики, видаляючи бот-трафік, який може спотворити веб-сайт та показники конверсії [15].

Деякі боти, такі як парсери та краулери, прочісують веб-сайт, щоб проіндексувати веб-сторінки, завантажити контент або використовувати API небажаними способами для отримання доступу до корпоративних даних. AWS WAF Bot Control класифікує найбільш поширені боти, тому можна блокувати окремих ботів або цілу категорію ботів, наприклад пошукові роботи SEO, парсери або інструменти моніторингу сайту. За промовчанням AWS WAF Bot Control не блокує звичайних ботів, таких як веб-сканери пошукових систем [15].

Використовуючи AWS WAF Bot Control та інші функції WAF, такі як відповіді, що настроюються і впровадження заголовка запиту, можна створювати власні робочі процеси додатків для трафіку ботів. Наприклад, можна дозволити

ботам, які копіюють або очищають дані про ціни, оскільки вони можуть направляти трафік на корпоративний сайт, але є можливість блокувати надмірні запити від ботів, які можуть перевантажити базу даних про ціни в реальному часі. За допомогою AWS WAF можна спрямовувати трафік ботів на альтернативну кінцеву точку, де дані про ціни кешуються і при перенаправленні трафіку користувача на сторінки, які надають дані про ціни в реальному часі [15].

3 ПОРЯДОК ЗАСТОСУВАННЯ ТЕХНОЛОГІЇ ЗАХИСТУ ВЕБ-ДОДАТКІВ ОРГАНІЗАЦІЇ НА БАЗІ AWS WEB APPLICATION FIREWALL

3.1. Порядок розгортання рішення AWS WAF Security Automations

Рішення AWS WAF Security Automations призначене для захисту веб-застосунків та API-інтерфейсів, розгорнутих за допомогою Amazon CloudFront, Application Load Balancer або Amazon API Gateway [14].

Нагадаємо, що AWS CloudFormation дозволяє моделювати та розподіляти ресурси AWS та сторонні ресурси, а також керувати ними завдяки формату «інфраструктура як код». Тому, ввімкнений шаблон AWS CloudFormation слід використовувати як відправну точку для впровадження правил AWS WAF.

Веб-список контролю доступу, який створюється цим рішенням, призначений для комплексного захисту веб-додатків. Стандартна конфігурація додає вісім правил AWS WAF до веб-ACL цього рішення. Також можна вручну змінити веб-ACL, щоб додати додаткові правила, але зверніть увагу, що для окремих веб-ACL існує обмеження до 10 правил (рис. 3.1).

Розглянемо умови відповідності IP. AWS WAF може блокувати до 10 000 діапазонів IP-адрес у нотації безкласової міждоменної маршрутизації (CIDR) для кожної умови відповідності IP. Це обмеження поширюється на кожен список, який створюється цим рішенням. Білий список, чорний список (компонент ручних списків IP-адрес) і чорний список IP-адрес (компонент синтаксичного аналізу списку IP-адрес) – це окремі списки, кожен з обмеженням в 10 000 IP-адрес для отримання додаткової інформації (рис. 3.2).

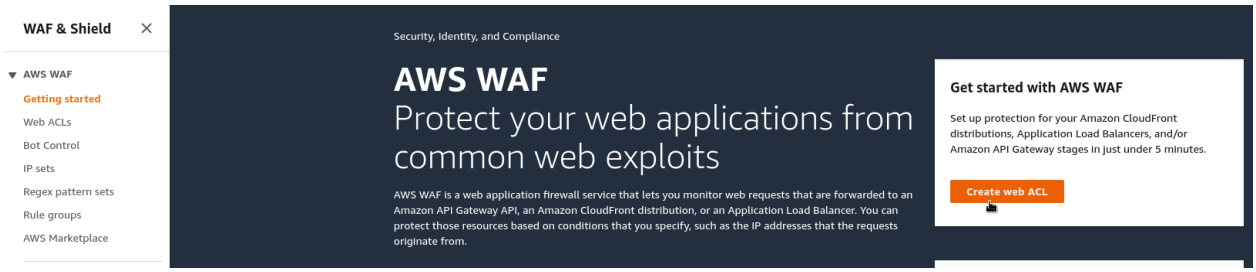

Рис. 3.1. Інтерфейс створення Web ACL

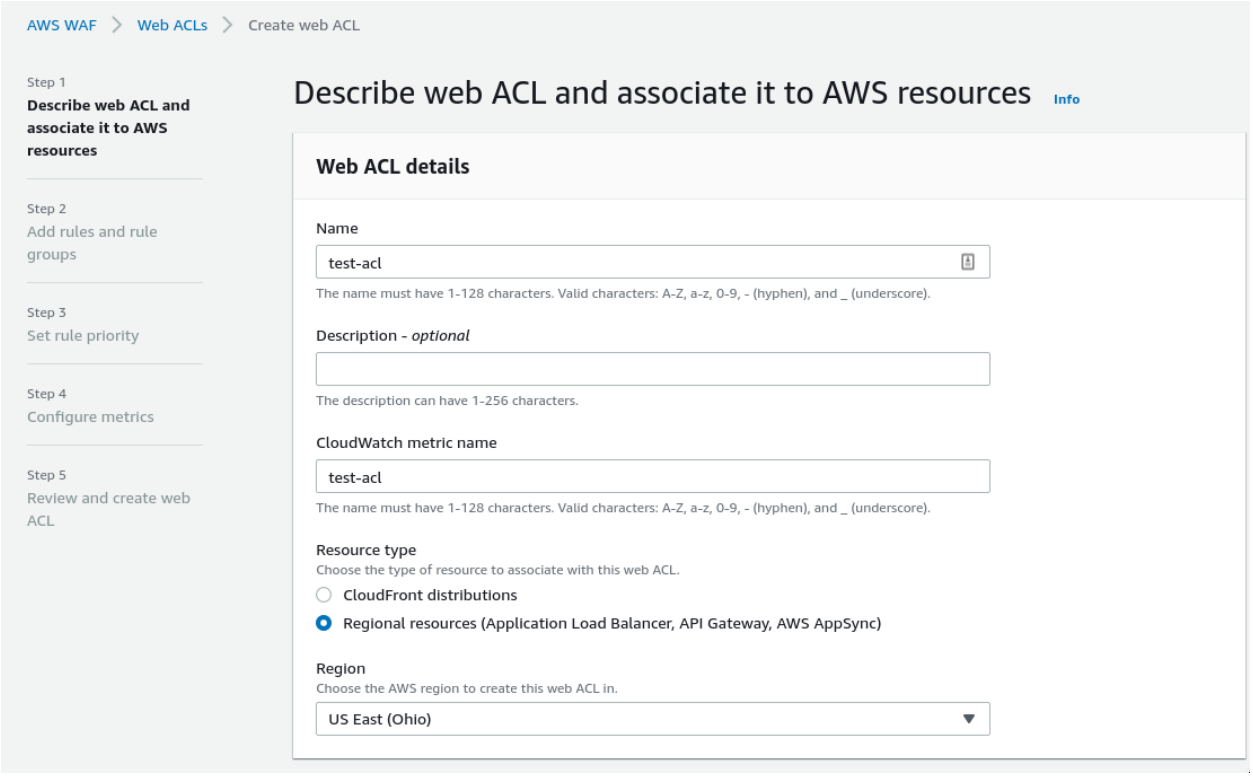

Рис. 3.2. Інтерфейс створення Web ACL

Розглянемо порядок автоматичного розгортання.

Перед запуском шаблону AWS CloudFormation необхідно ознайомтеся з архітектурою та конфігурацією рішення. Необхідно дотримуватися покрокових вказівок, щоб налаштувати та розгорнути рішення AWS WAF Security Automations у своєму обліковому записі. Час розгортання: приблизно 15 хвилин.

Це рішення призначене для роботи з веб-додатками, розгорнутими за допомогою Amazon CloudFront або Application Load Balancer. Якщо ще не налаштовано один із цих ресурсів, необхідно виконати відповідне завдання перед запуском цього рішення.

Налаштування розповсюдження CloudFront: необхідно виконати такі кроки, щоб настроїти CloudFront для розповсюдження як статичного, так і динамічного вмісту вашого веб-додатку:

необхідно створити дистрибутив CloudFront;

необхідно налаштувати статичні та динамічні джерела;

необхідно вказати поведінку дистрибутива.

Необхідно налаштувати балансувальник навантаження додатків. Для цього

необхідно виконати такі кроки, щоб налаштувати балансувальник навантаження програм для розподілу вхідного трафіку у веб-додатку:

необхідно налаштувати балансувальник навантаження та прослуховувач;

необхідно налаштувати параметри безпеки для прослуховувача HTTPs;

необхідно налаштувати групу безпеки;

необхідно налаштувати цільову групу;

необхідно налаштувати цілі для цільової групи;

необхідно створити балансувальник навантаження.

Процедура розгортання архітектури AWS WAF Security Automations складається з наступних кроків.

Крок 1. Запустіть стек.

Запустіть шаблон AWS CloudFormation у обліковому записі AWS.

Цей автоматизований шаблон AWS CloudFormation розгортає рішення AWS WAF Security Automations у хмарі AWS.

Введіть значення для потрібних параметрів: *Stack Name, Application Access Log Bucket Name.*

Перегляньте інші параметри шаблону та за потреби налаштуйте.

Крок 2. Змініть дозволені та заборонені набори (необов’язково).

Вручну додайте відповідні IP-адреси до списку дозволу та заборони AWS WAF.

Після розгортання стека AWS CloudFormation цього рішення можна вручну змінити дозволені та заборонені набори для додавання або видалення IP-адрес при необхідності.

Крок 3. Вставте посилання на приманку Honeypot у свій веб-додаток (необов’язково).

За необхідності додайте кінцеву точку Honeypot веб-додаток.

Очевидно забороніть доступ до кінцевої точки, використовуючи стандарт виключення ботів (тільки CloudFront).

Якщо ви вибрали активацію захисту сканера та датчика на кроці 1, шаблон AWS CloudFormation створює кінцеву точку перехоплення для Honeypot з

низьким рівнем взаємодії, призначену для виявлення та відхилення вхідних запитів від парсерів контенту та поганих ботів. Допустимі користувачі не намагатимуться отримати доступ до цієї кінцевої точки. Однак парсери контенту та боти, такі як шкідливі програми, які сканують вразливості системи безпеки та очищають адреси електронної пошти, можуть спробувати отримати доступ до кінцевої точки Honeypot. У цьому сценарії функція AWS Lambda обробника доступу перевірятиме запит, щоб витягти його джерело,

Використовуйте відповідну процедуру для вбудовування Honeypot для запитів або від роздачі CloudFront, або від Application Load Balancer.

Створіть джерело CloudFront для кінцевої точки Honeypot.

Використовуйте цю процедуру для веб-застосунків, розгорнутих за допомогою розповсюдження CloudFront. З CloudFront ви можете увімкнути *robots.txt* файл, який допоможе ідентифікувати парсери контенту та ботів, які ігнорують стандарт виключення роботів. Виконайте такі кроки, щоб вбудувати приховане посилання, а потім явно заборонити його у вашому *robots.txt* файлі.

Крок 4. Поєднайте веб-ACL із веб-додатком.

Необхідно поєднати веб-дистрибутиви Amazon CloudFront або балансувальники навантаження додатків із веб-списком контролю доступу, який створюється цим рішенням. Можна поєднати стільки дистрибутивів або балансувальників навантаження, скільки необхідно.

Використовуйте цю процедуру для веб-додатків, розгорнутих за допомогою Application Load Balancer.

Оновіть свої дистрибутиви Amazon CloudFront або балансувальники навантаження додатків, щоб активувати AWS WAF та ведення журналу за допомогою ресурсів, створених на кроці 1.

Крок 5. Налаштуйте ведення журналу веб-доступу.

Увімкніть ведення журналу веб-доступу для веб-розповсюдження Amazon CloudFront або балансувальника навантаження додатків і надішліть файли журналу до відповідного кошика Amazon S3. Необхідно зберігати журнали в папці *AWSLogs* (префікс журналу *AWSLogs/*).

Необхідно налаштувати Amazon CloudFront або балансувальник навантаження додатків для надсилання журналів веб-доступу до відповідного кошика Amazon S3, щоб ці дані були доступні для функції Log Parser AWS Lambda.

3.2. Технологія застосування рішення AWS Web Application Firewal

AWS WAF допомагає захистити інтерфейсні служби AWS від різних типів атак і відомих вразливостей. У середовищі AWS веб-додаток може працювати на віртуальних машинах EC2 або контейнерах ECS або використовувати безсерверні служби, як-от API Gateway. Це означає, що веб-додатки або веб-API можуть бути доступні через різні кінцеві точки [17].

AWS WAF може блокувати шкідливий трафік і забезпечувати безпеку веб-додатків і API. В цьому допомагають: балансувальник навантаження додатків, дистрибутиви CloudFront, шлюз API та API AppSync GraphQL.

Щоб почати захищати свої ресурси AWS, спочатку потрібно створити список контролю доступу (ACL). Кожен веб-ACL містить набір правил або груп правил. Групи правил можна використовувати повторно, а також можна використовувати керовані групи правил у кількох веб-списках керування доступом [17].

Правило WAF визначає загальні моделі атак, які слід стежити у вхідних запитах, а також дії, які слід виконувати, коли запит відповідає шаблону. Можна створювати власні правила на основі вимог або використовувати існуючі правила, запропоновані AWS або на AWS Marketplace.

Кожне правило визначає, що потрібно перевірити і яку дію виконати, якщо потрібно. Доступні три дії [17]:

дозволити всі вхідні запити, крім тих, які ви спеціально заблокували;

блокування всіх вхідних запитів, крім тих, які ви спеціально дозволили;

враховувати лише запити, які відповідають налаштованим властивостям правила.

Наприклад, у вас може бути дуже просте правило для блокування всіх запитів, що надходять з певної країни або набору IP-адрес.

Можна додати кілька правил до одного веб-списку ACL, поки він не досягне загальної ємності ACL. За замовчуванням кожен веб-список ACL має 1500 одиниць ємності. AWS обчислює ці одиниці по-різному для кожного правила. Наприклад, з точки зору потужності обробки, перевірка запиту зі списком відомих правил IP-адресації коштує дешевше, ніж перевірка на відомі специфічні для Windows експлойти.

Як згадувалося вище, кожне правило WAF може мати дію дозволу, блокування або підрахунку. Дозвіл і блокування – це дії, що припиняють, тому якщо у вашому веб-списку ACL є кілька правил, той, хто першим придумає відповідність, повернеться з налаштованою дією і зупинить оцінку правил, що залишилися. Підрахунок є дією, яка не припиняється, тому якщо ви визначили правило підрахунку і воно відповідає запиту, WAF зареєструє кількість і перейде до оцінки правил, що залишилися в ACL.

Розглянемо порядок створення веб-списку ACL і призначення правила для захисту від відомих загроз.

По-перше, потрібно створити веб-список керування доступом і пов’язати його з наявними ресурсами AWS. У наведеному нижче прикладі ми пов’язуємо веб-ACL з кінцевою точкою Amazon AppSync GraphQL. Додаток розгорнуто та вже працює (рис. 3.3).

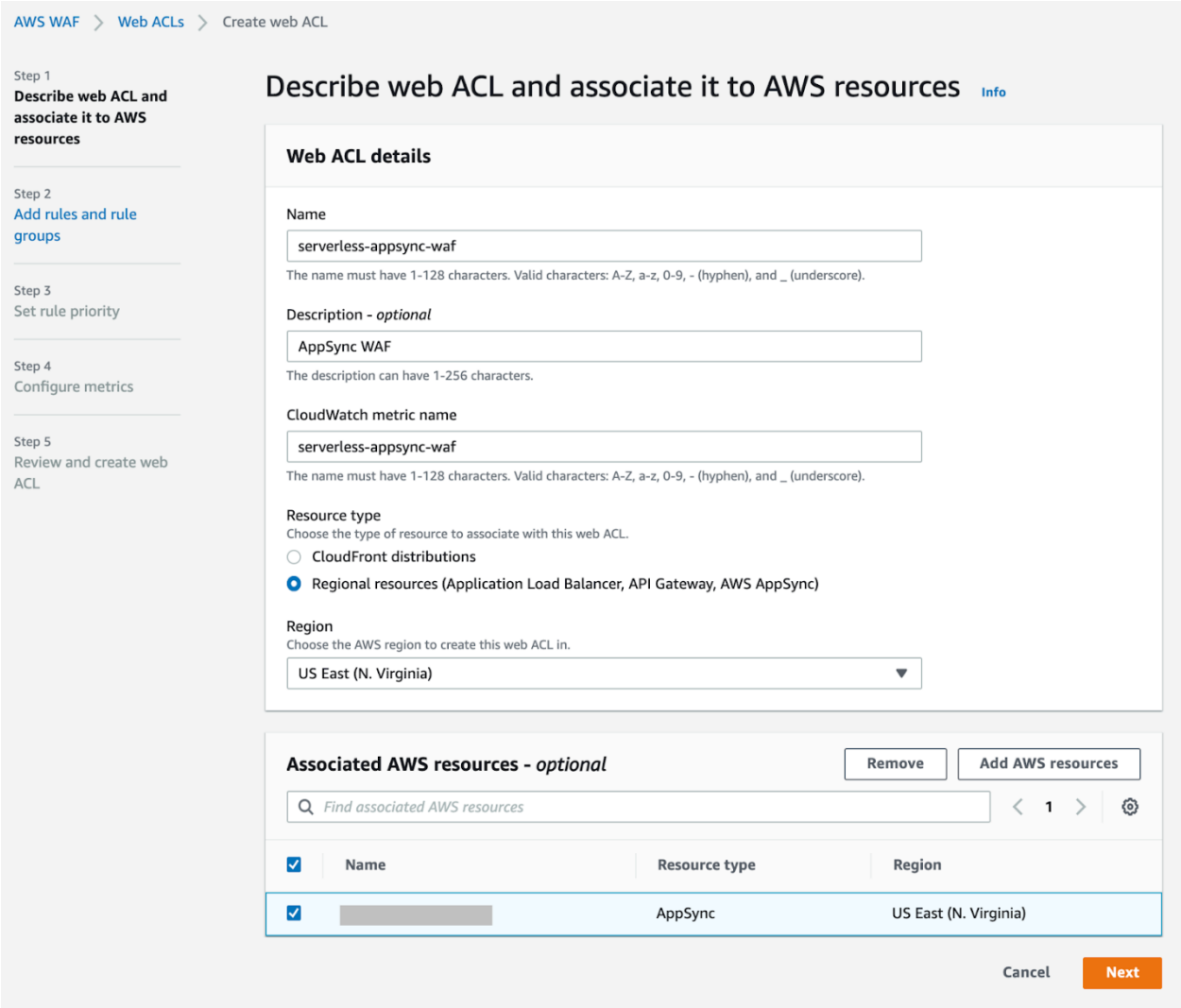

Рис. 3.3. Інтерфейс створення веб ACL

Amazon AppSync – це безсерверний, повністю керований механізм GraphQL для створення великомасштабних додатків GraphQL з такими функціями, як оновлення в реальному часі та автономні можливості.

Далі необхідно призначити правила списку керування доступом. Ви можете вибрати з керованих правил або створити власні.

Призначення правил ACL: для цього прикладу ми будемо використовувати одну з груп керованих правил, список яких ви можете отримати через AWS Marketplace (рис. 3.4).

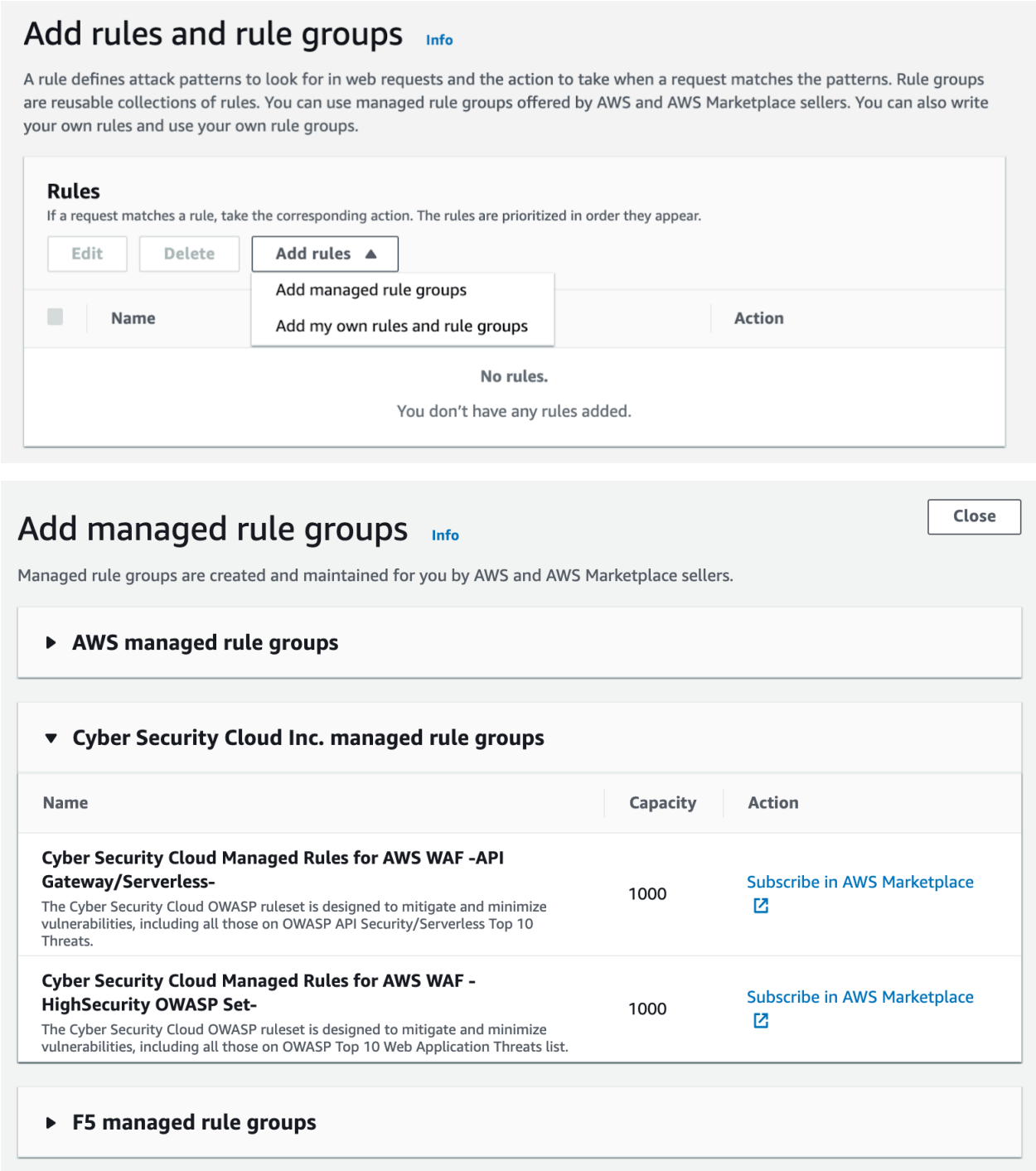

Рис. 3.4. Інтерфейс додавання груп керованих правил через AWS Marketplace

Кожне правило має пов'язану ємність, яка буде зарахована до загальної доступної ємності веб- ACL. Нижче ми вибираємо «Основний набір правил», щоб захистити від широкого спектру поширених уразливостей, включаючи топ-10 OWASP (рис. 3.5)

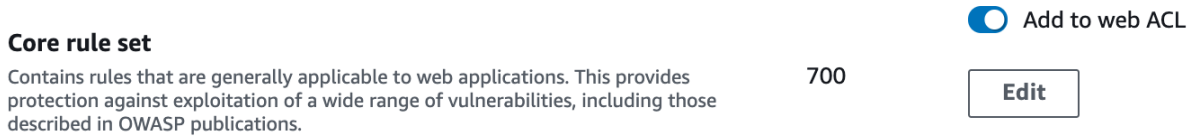

Рис. 3.5. Основний набір правил

Тепер можна побачити спожиту та залишок ємності веб ACL. Також можна налаштувати дію за замовчуванням, яка запускатиметься, якщо жодне з правил, визначених у цьому наборі правил, не відповідає шаблону запиту; тут встановлено значення «Дозволити» (рис. 3.6).

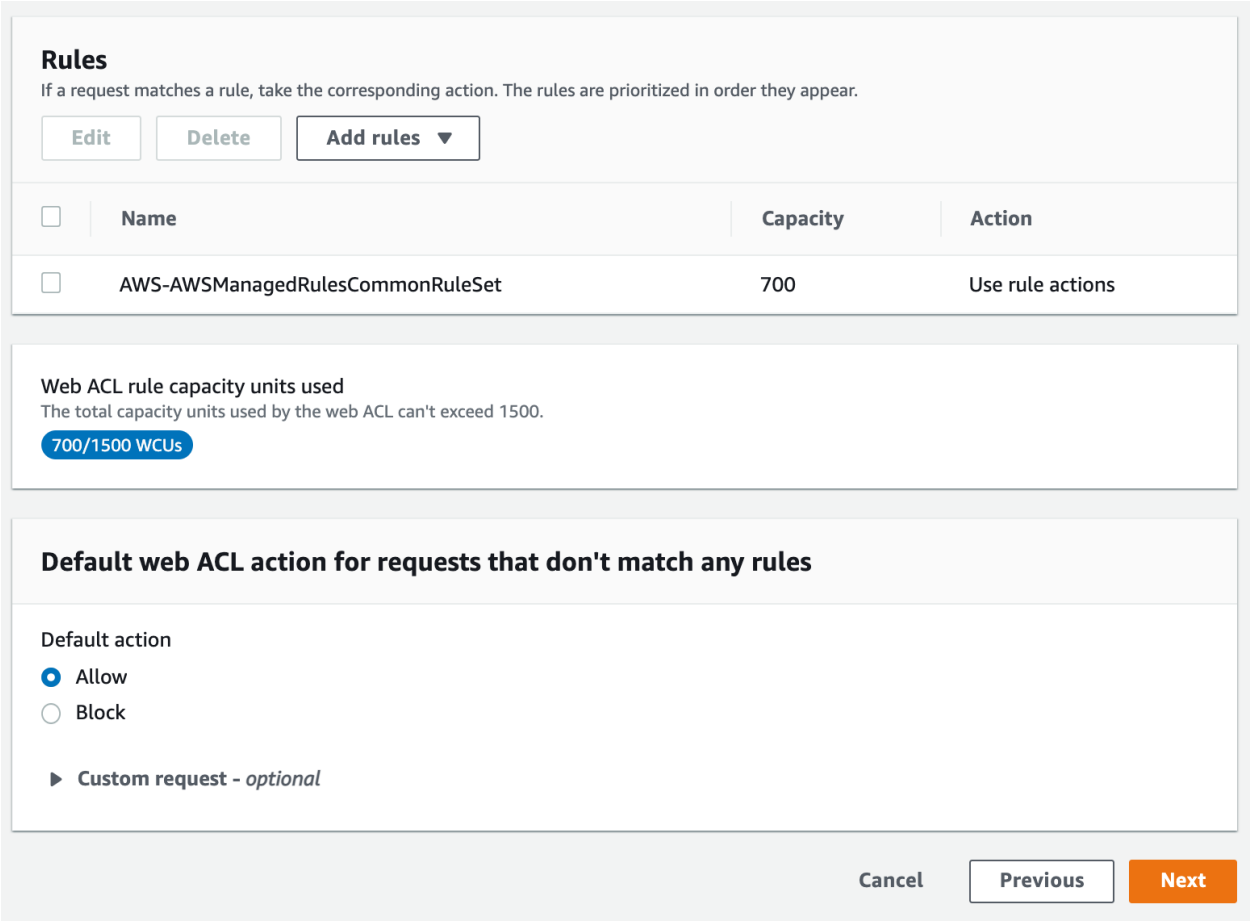

Рис. 3.6. Налаштування веб ACL

Якщо у веб-списку ACL є кілька правил, є можливість визначити пріоритет для їх оцінки в певному порядку. Як згадувалося раніше, дозволи та блокування є завершуючими діями, тому важливо розставити пріоритети правил у правильному порядку (рис. 3.7).

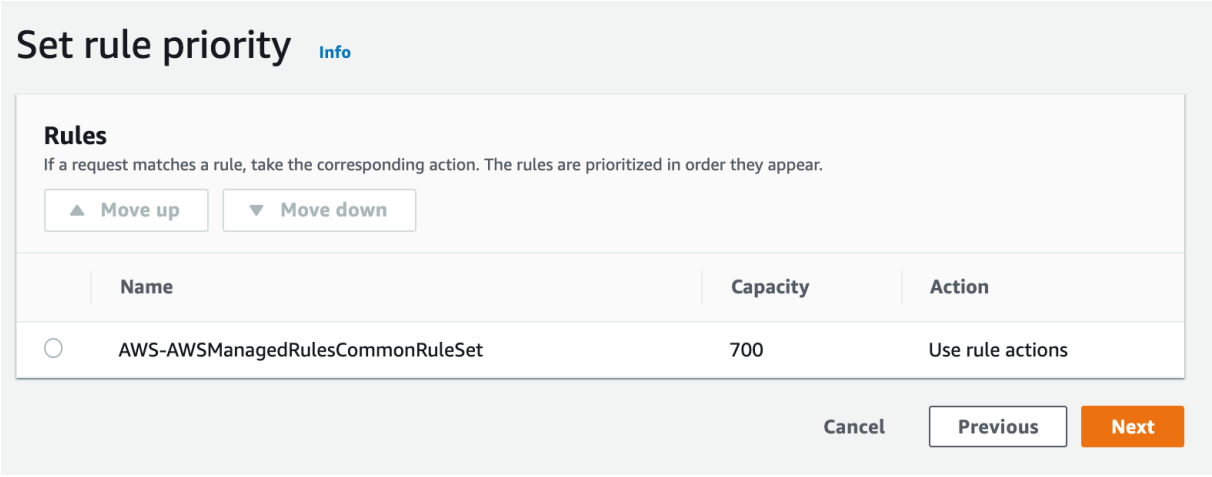

Рис. 3.7. Налаштування пріоритетів правил

Тут у нас є лише один набір правил, тому всі правила всередині цього єдиного правила будуть оцінюватися в тому порядку, в якому вони знаходяться, доки не буде знайдено відповідність.

Налаштуйте журналювання та метрики. Моніторинг і ведення журналів є важливими аспектами безпеки, тому AWS WAF надає можливості ведення журналів і показників, які показують, як обробляються вхідні запити. Журнали та метрики (рис. 3.8) також допомагають гарантувати, що хибнопозитивні тривоги не генеруються (тобто, що налаштовані правила не впливають на дійсні запити).

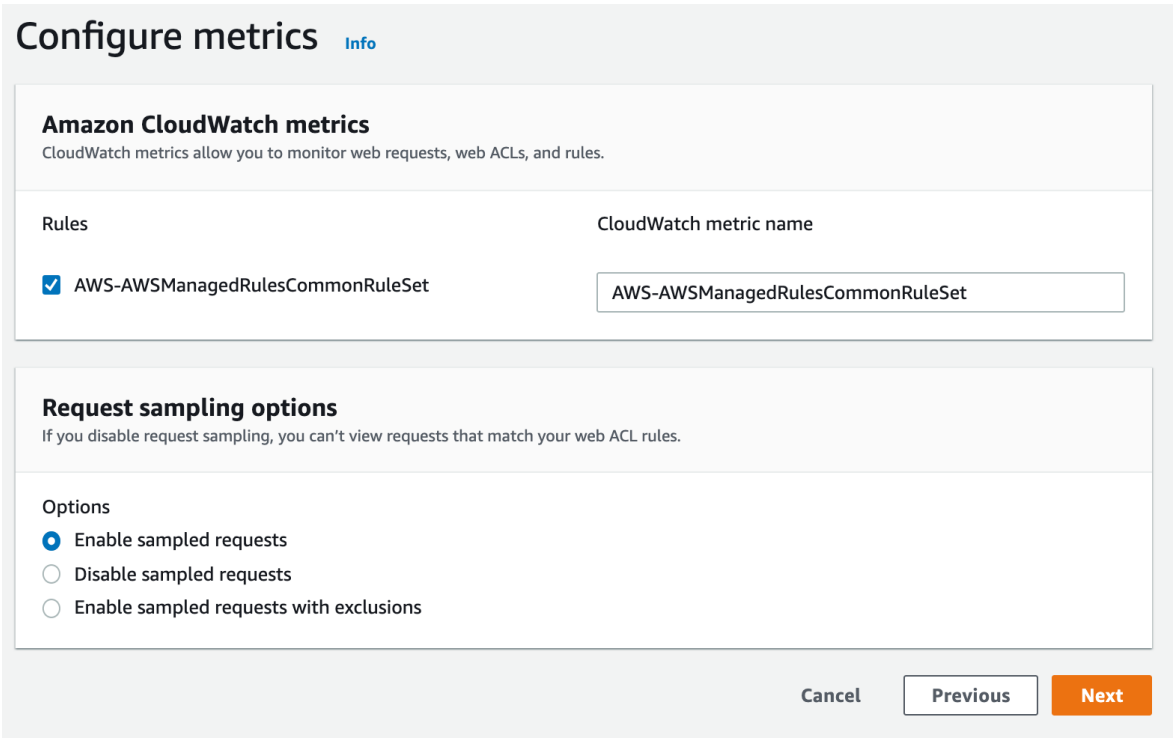

Рис. 3.8. Налаштування метрик

Після створення веб-списку ACL і очікування кількох хвилин, щоб дати йому можливість зібрати журнали запитів, інформаційна панель починає показувати показники та запити вибірки (рис. 3.9).

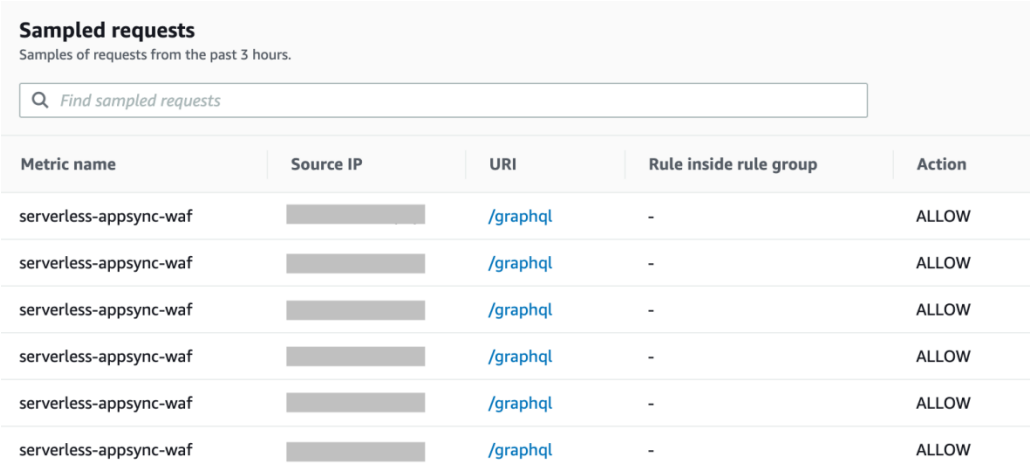

Рис. 3.9. Інтерфейс метрик

Можна розгорнути кожен із зібраних вибіркових запитів, щоб переглянути додаткову інформацію (рис. 3.10).

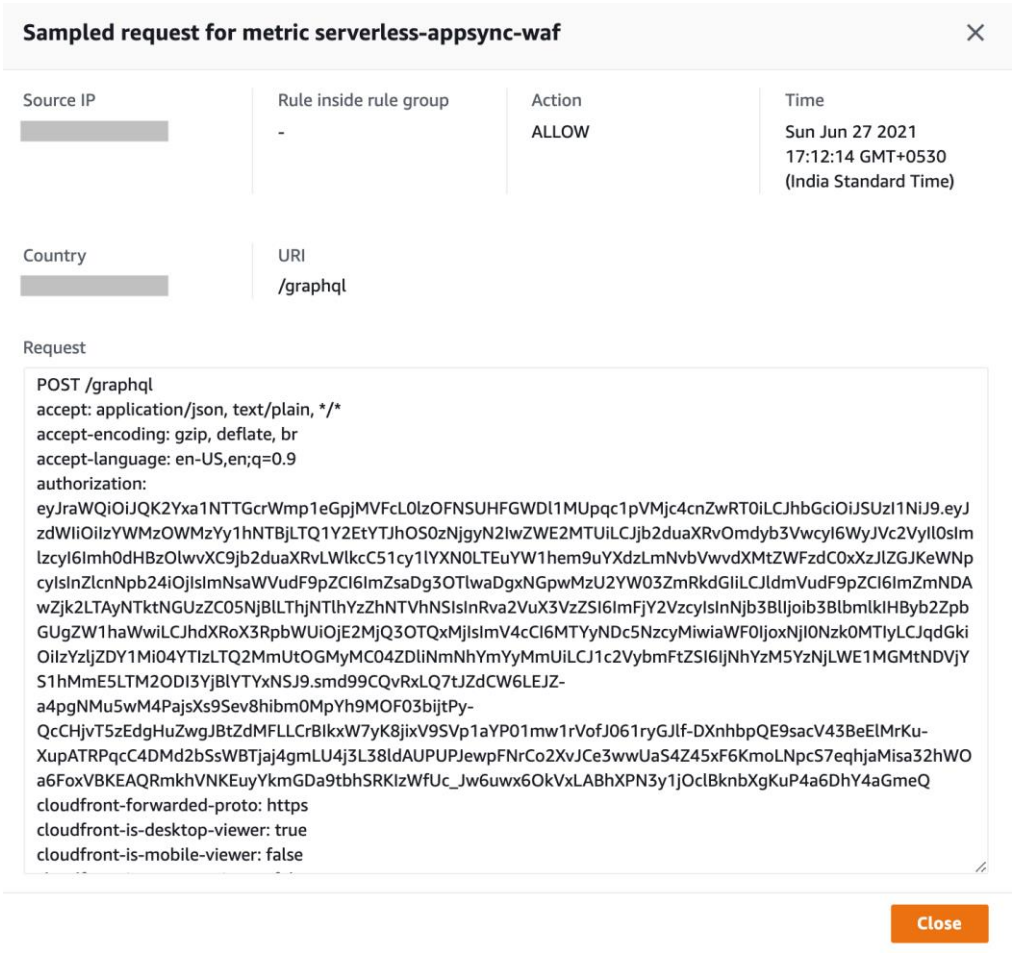

Рис. 3.10. Інтерфейс вибіркових запитів

Розглянемо практичні випадки використання рішення AWS WAF.

Окрім обробки вхідних запитів та прийняття рішення про дозвіл, блокування чи підрахунок, AWS WAF має інші корисні функції, які підходять для сучасних випадків використання.

AWS WAF може здійснювати контроль ботів. Деякі боти (наприклад, сканери пошукових систем) можуть бути корисними, але багато з них є шкідливими або з інших причин небажані. Тому керування ботами є важливим аспектом веб-безпеки. AWS WAF надає групи керованих правил керування ботом, які можна використовувати з іншими правилами для захисту корпоративних веб-додатків.

AWS WAF може здійснювати обмеження швидкості. Звичайні правила WAF призначені для дозволу, блокування або підрахунку кількості запитів, що надходять, але також можна налаштувати правила, щоб обмежити доступ. Правила на основі тарифів такі ж, як і звичайні, але замість того, щоб дозволяти чи блокувати запити, вони відстежують, скільки запитів надходить з окремих IP-адрес за п’ятихвилинний період. Це допомагає налаштувати конкретну кількість запитів, які кожна IP-адреса може генерувати за певний період часу.

В AWS WAF можна налаштовувати відповіді на запит. За замовчуванням, коли запит заблоковано, AWS WAF надсилає клієнту відповідь HTTP 403. Якщо необхідно змінити це, можна налаштувати відповідь за замовчуванням. У деяких випадках також можна налаштувати запит або перенаправити його на основі заголовків HTTP. Наприклад, запити, що надходять від ботів, можуть бути перенаправлені до кешованого джерела даних замість завантаження даних з бази даних.

API AWS WAF дозволяють програмно налаштовувати правила та керувати політиками безпеки в рамках CI/CD або інших автоматизованих процесів. Якщо здійснюється керування кількома обліковими записами AWS, це може допомогти підтримувати узгодженість у різних середовищах. AWS пропонує іншу службу, Менеджер брандмауера, для централізованого керування всіма правилами WAF, але вона може застосовувати правила лише для різних облікових записів AWS в

організації AWS.

Необхідно відмітити, що сучасний ландшафт загроз різноманітний і постійно розширюється. AWS WAF є корисним сервісом у багатьох відношеннях, але для більшості організацій його буде недостатньо для повного захисту хмарних робочих навантажень.

Щоб правильно використовувати AWS WAF, адміністратори повинні розуміти всі потенційні загрози та налаштувати правила ACL для захисту від них. Щоб зробити це повністю та правильно, потрібен рівень знань у сфері безпеки, якого немає у багатьох адміністраторів.

Іншим істотним обмеженням є те, що правила ACL, після визначення, не змінюються та не адаптуються самі по собі. Адміністратори повинні стежити за середовищем загроз (яке постійно розвивається) і налаштовувати параметри AWS WAF для вирішення нових загроз відповідно до поточних вимог безпеки. Це може бути складним і тривалим.

AWS WAF дозволяє адміністраторам оновлювати правила в режимі реального часу, хоча може знадобитися короткий час (від кількох секунд до кількох хвилин), перш ніж нове правило буде застосовано до всіх необхідних кінцевих точок.

Під час застосування правил пов’язані ресурси AWS можуть отримати відкладену відповідь про дозвіл або блокування вхідних запитів через внутрішні помилки. Якщо це станеться, і пов’язаний ресурс є дистрибутивом CloudFront, то зазвичай це дозволить запиту пройти. Для всіх інших регіональних служб, таких як Application Load Balancers або AppSync GraphQL, запит буде відхилено AWS WAF.

Існують архітектурні обмеження. AWS WAF можна налаштувати на служби AWS лише в одному обліковому записі AWS. Однак більшість підприємств використовують стратегію з кількома обліковими записами або кількома хмарами, де весь їхній додаток використовується кількома географічними місцями. Підтримка послідовної базової конфігурації безпеки в таких середовищах є складною, трудомісткою і схильною до помилок. AWS надає додаткові послуги,

як-от Менеджер брандмауера, щоб налаштувати правила WAF для різних облікових записів AWS або AWS Shield Advanced, щоб посилити безпеку та запобігти DDoS-атакам, але вони корисні чи застосовні не в усіх ситуаціях.

Необхідно зауважити, що AWS WAF не є повноцінною платформою безпеки і вона не призначена для цього. Існує багато загроз, від яких AWS WAF не призначено для захисту. Навіть серед загроз, які він покриває, у багатьох випадках його захист є досить елементарним.

Таким чином, організації, які використовують хмару, потребують захисту для своїх робочих навантажень, і AWS WAF є популярним вибором. Його простота налаштування, інтеграція з іншими сервісами AWS та цінова модель є дуже привабливими.

Однак для більшості організацій базового рівня безпеки, який забезпечує AWS WAF, буде недостатньо. Додаткові послуги (такі як AWS Firewall Manager і AWS Shield Advanced) і платні доповнення від AWS Marketplace заповнять деякі з цих пробілів, але не всі, та не вирішують проблем, пов’язані з тим, що рішення, яке самостійно керує, схильне до помилок. і важко підтримувати постійно.

У цих обставинах Reblaze може допомогти. Reblaze – це комплексна платформа безпеки, яка працює на AWS. Вона забезпечує надійний захист, якого не мають вбудовані служби (AWS WAF, AWS Shield тощо). Поряд із WAF наступного покоління, який є більш гнучким і потужним, ніж AWS, він також включає захист від DDoS, розширене керування ботами, контроль потоку сеансів, запобігання захоплення облікового запису, безпеку API, мобільний SDK тощо.

Крім того, Reblaze не обмежується середовищами AWS. Він також може захистити робочі навантаження в Microsoft Azure, Google Cloud Platform, Digital Ocean, гібридних архітектурах, контейнерах та сервісних мережах тощо. А оскільки це повністю керована служба, то веб-безпека підтримується та оновлюється для командою експертів із безпеки, що є більш надійним рішенням.

3.3. Рекомендації щодо застосування технології захисту веб-додатків організації

Як рекомендують в OWASP [3] необхідно розробити та втілити програму забезпечення безпеки додатків.

Безпека додатків є необхідною умовою бізнесу. Під тиском регуляторів та зростаючої кількості атак організації повинні розробляти ефективні процеси та засоби забезпечення безпеки своїх додатків та API. Багато організацій намагаються впоратися з величезною кількістю вразливостей у неймовірному обсязі коду вже випущених програм та API.

OWASP [3] рекомендує розробити програму забезпечення безпеки, щоб проаналізувати та покращити безпеку додатків та API. Забезпечення безпеки потребує ефективної взаємодії різних підрозділів організації, включаючи аудиторів, розробників, керівників та адміністраторів. Безпека повинна бути наочною та вимірюваною, щоб можна було побачити та зрозуміти стан безпеки додатків. Зробіть акцент на роботах та результатах, які реально покращать безпеку та усунуть чи зменшать ризики.

На початку роботи необхідно:

задокументувати всі веб-додатки та пов’язані з ними дані. Великим організаціям рекомендується використовувати з цією метою базу даних управління конфігурацією;

розробити програму забезпечення безпеки додатків та почати її реалізацію;

проведіть аналіз можливостей, порівнявши свою організацію з іншими компаніями, щоб визначити ключові області поліпшення та план дій;

отримати схвалення посібника з цих питань та розробити план підвищення поінформованості про безпеку додатків для всієї організації.

Щоб реалізувати загальний підхід на основі ризиків необхідно:

визначити необхідний рівень захисту веб-додатків з погляду бізнесу. При цьому необхідно керуватися законами про конфіденційність та іншими нормативними документами, що стосуються захищених даних;

розробити модель оцінки найбільш поширених загроз із зазначенням факторів ймовірності та ризику, що відображають стійкість вашої організації до атак;

оцінити та пріоритезувати веб-додатки та API. Занести результати до БД керування конфігурацією.

розробити посібник із коректного визначення необхідного рівня покриття та точності.

Під час підготовки надійної бази необхідно:

розробити спеціальні політики та стандарти, які будуть використовуватися всіма розробниками як основи безпеки додатків;

визначити набір стандартних засобів безпеки, які доповнюватимуть ці політики та стандарти, а також які міститиме посібник з їх використання при проектуванні та розробці;

розробити курси з безпеки додатків, присвячені різним темам і цілям розробки.

Необхідно інтегрувати заходи та засоби безпеки в існуючі процеси. Для цього необхідно:

визначити та впровадити в існуючі процеси розробки та експлуатації заходи щодо безпечної реалізації та контролю. Це включає такі заходи: моделювання загроз, безпечне проектування та аналіз проектів, написання безпечного коду та його аналіз, проведення пентесту та усунення недоліків за результатами;

залучати експертів у предметній галузі та служб підтримки для розробників та проектної команди.

Необхідно забезпечити візуальний контроль процесів. Для цього необхідно:

використовувати різні метрики. Приймати рішення про поліпшення та фінансування необхідно на основі метрик та даних аналітики. Метрики повинні відображати засоби та методи забезпечення безпеки, виявлені та усунені вразливості, покриття додатків, опис помилок за типом та кількістю тощо;

проводити аналіз даних щодо реалізації та контролю для пошуку основних причин та шаблонів вразливостей при проведенні стратегічних та системних

покращень у компанії. Враховуйте помилки та пропонуйте заохочення для просування покращень.

Для правильного управління життєвим циклом веб-додатків пропонуються такі заходи відповідно до [3].

Необхідно зазначити, що веб-додатки відносяться до найбільш складних систем, які люди постійно створюють та обслуговують. Адміністрування додатків необхідно доручати IT-фахівцям, які відповідатимуть за весь їхній життєвий цикл. Необхідно призначати адміністраторів додатків, які відповідатимуть за технічні аспекти додатків протягом їх життєвого циклу, починаючи зі збору вимог і закінчуючи виведенням систем з експлуатації, про що часто забувають.

Для організації управління ресурсами необхідно:

зібрати та обговорити із замовником бізнес-вимоги до веб-додатків, включаючи забезпечення конфіденційності, справжності, цілісності та доступності всіх інформаційних активів, а також очікувану бізнес-логіку;

скласти перелік технічних вимог, включаючи функціональні та нефункціональні вимоги щодо безпеки;

спланувати та обговорити бюджет, що охоплює всі аспекти проектування, створення, тестування та експлуатації, а також роботи із забезпечення безпеки.

На етапі запиту пропозицій та укладання контракту необхідно:

обговорити вимоги з внутрішніми та зовнішніми розробниками, включаючи нормативи та вимоги програми безпеки, наприклад, рекомендації щодо життєвого циклу розробки програмного забезпечення;

оцінити виконання всіх технічних вимог, включаючи етап планування та проектування;

обговорити всі технічні вимоги, включаючи проектування, безпеку та гарантійні зобов’язання;

використовувати шаблони та контрольні списки, а також проконсультуватися з юристом перед його використанням.

На етапі планування та проектування веб-додатків необхідно:

обговорити плани та проекти з розробниками та внутрішніми партнерами,

наприклад, фахівцями з безпеки;

визначити архітектуру та засоби керування безпекою, а також контрзаходи, які відповідають вимогам захисту та очікуваним рівням небезпеки. Все це має забезпечуватись фахівцями з безпеки;

переконатися, що власник додатку приймає інші ризики або надає додаткові ресурси.

забезпечити створення записів з безпеки із зазначенням обмежень, доданих для нефункціональних вимог.

На етапі розгортання, тестування та впровадження веб-додатків необхідно:

автоматизувати безпечне розгортання додатку, інтерфейсів та компонентів, а також отримання необхідних дозволів;

протестувати технічні можливості та інтеграцію з IT-архітектурою, а також організувати бізнес-тестування;

протестуйте штатне та нештатне використання технічних та продуктивних можливостей;

організувати тестування безпеки відповідно до внутрішніх процесів, вимог захисту та передбачуваного рівня небезпеки для кожного веб-додатку;

ввести веб-додаток в експлуатацію та заборонити використовувати старі програми без потреби;

погодити всю документацію, базу даних контролю змін та архітектуру безпеки.

На етапі проведення роботи та контролю змін необхідно:

роботи повинні включати управління безпекою веб-додатку (наприклад, управління оновленнями);

постійно звертати увагу користувачів на безпеку та знайти компроміс між практичністю та безпекою;

спланувати та проводити модифікації, наприклад, перехід на нову версію програми або використання інших компонентів (ОС, програмне забезпечення або бібліотек);

оновлювати документацію, включаючи документацію з контролю змін,

архітектуру безпеки, елементи керування та заходи протидії, а також документацію з поточних завдань або проектів.

На етапі виведення з експлуатації додатку необхідно:

усі важливі дані необхідно заархівувати, а решту безпечно видалити;

здійснити безпечне виведення додатку з експлуатації, включаючи видалення облікових записів, що не використовуються, а також ролей і дозволів;

встановити статус додатку “виведено з експлуатації” у БД контролю змін.

Необхідно підкреслити важливість такого організаційного моменту при побудові системи захисту веб-додатків як тест на проникнення. Саме він стане оптимальним способом перевірки захищеності інформаційної системи за допомогою імітації спрямованих атак. Тест на проникнення дає можливість оцінити захищеність інформаційної системи від несанкціонованого впливу, використовуючи різні моделі вторгнень.

Тест на проникнення для веб-додатків фокусує свою увагу виключно на оцінці рівня захисту веб-додатків. Процес складається з активного аналізу додатків і пошуку в них вразливостей, технічних помилок або інших проблем. Інформацію про всі слабкі місця відображається в підсумковому звіті.

ВИСНОВКИ

В роботі проведено дослідження та аналіз проблеми забезпечення захисту веб-додатків як складової частини інформаційної системи організації, встановлена сутність завдань їх захисту.

Amazon Web Services (AWS) – це найпоширеніша у світі хмарна платформа з найширшими можливостями, що надає понад 200 повнофункціональних сервісів для центрів обробки даних. Тому, AWS широко застосовується у сучасних організаціях.

Проаналізовано існуючі технології захисту веб-додатків організації. Досліджена технологія захисту веб-додатків організації на прикладі рішення AWS Web Application Firewall.

Визначено методи та засоби захисту веб-додатків організації, які реалізовані в AWS Web Application Firewall.

Встановлено основні функції та принципи роботи рішення AWS Web Application Firewall. AWS Web Application Firewall – це програмне забезпечення: міжмережевий екран для інтернет-додатків, який дозволяє захистити їх (або різні API) від поширених мережевих експлойтів та ботів, здатних вплинути на доступність, створити загрозу безпеці або задіяти надмірну кількість ресурсів API або додатку.

AWS Web Application Firewall надає повний контроль над тим, як саме трафік досягатиме додатків користувача за допомогою створюваних правил безпеки, які контролюватимуть трафік від ботів і блокуватимуть такі виконувані за поширеними шаблонами атаки, як SQL-ін’єкції або міжсайтовий скріптинг.

AWS Web Application Firewall допомагає захистити інтерфейсні служби AWS від різних типів атак і відомих вразливостей. У середовищі AWS веб-додаток може працювати на віртуальних машинах EC2 або контейнерах ECS або використовувати безсерверні служби, як-от API Gateway. Це означає, що веб-додатки або веб-API можуть бути доступні через різні кінцеві точки.

AWS Web Application Firewall може блокувати шкідливий трафік та забезпечувати безпеку веб-додатків і API. В цьому допомагають: балансувальник навантаження додатків, дистрибутиви CloudFront, шлюз API та API AppSync GraphQL.

Щоб почати захищати свої ресурси AWS, спочатку потрібно створити список контролю доступу (ACL). Кожен веб-ACL містить набір правил або груп правил. Групи правил можна використовувати повторно, а також можна використовувати керовані групи правил у кількох веб-списках керування доступом. Правило WAF визначає загальні моделі атак, які слід стежити у вхідних запитах, а також дії, які слід виконувати, коли запит відповідає шаблону. Можна створювати власні правила на основі вимог або використовувати існуючі правила, запропоновані AWS або на AWS Marketplace.

У роботі запропоновано порядок розгортання та застосування технології захисту веб-додатків організації із застосуванням рішення AWS Web Application Firewall. Розроблено рекомендації фахівцям з кібербезпеки щодо застосування технології захисту веб-додатків організації.

Таким чином, правильна реалізація технології захисту веб-додатків організації має забезпечити ефективний захист корпоративних даних та кібербезпеку інформаційної системи організації.

ПЕРЕЛІК ПОСИЛАНЬ

1. Andrew Hoffman. Web Application Security Exploitation and Countermeasures for Modern Web Applications. O'Reilly Media. 2020. 331 p. https://www.nginx.com/resources/library/web-application-security.

2. vue.js https://question-it.com/tags/vue.js/info.

3. OWASP Top 10 – 2017. The Ten Most Critical Web Application Security Risks. https://owasp.org/www-pdf-archive/OWASP_Top_10-2017_%28en%29.pdf.pdf.

4. Александр Макаров. Что такое веб-приложения, виды и их преимущества. https://www.azoft.ru/blog/web-apps/.

5. Василенко І. В. Універсальній метод захисту веб-додатків / І. В. Василенко // Системи обробки інформації. – 2016. – Вип. 1. – С. 122-124. – Режим доступу: http://nbuv.gov.ua/UJRN/soi_2016_1_27.

6. 5 Best Practices for Application Security. A How-To Guide. Tenable. https://static.tenable.com/marketing/whitepapers/Whitepaper-5_Best_Practices_for_Application_Security.pdf.

7. Framework for Improving Critical Infrastructure Cybersecurity. Version 1.1. National Institute of Standards and Technology. April 16, 2018. https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.04162018.pdf.

8. FortiWeb 6.4.1. Administration Guide. Fortinet Document Library. https://docs.fortinet.com/document/fortiweb/6.4.1/administration-guide/60895/introduction.

9. Захист веб-додатків: чому це важливо? ITBIZ. https://itbiz.ua/statti-ta-obzori/zaxist-veb-dodatkiv-chomu-ce-vazhlivo/#.

10. AWS Business Applications. Scalable, pay-as-you-go business applications built on AWS. https://aws.amazon.com/business-applications/?nc1=h_ls.

11. AWS WAF – Web Application Firewall. Protect your web applications from common web exploits. https://aws.amazon.com/waf/?nc1=h_ls.

12. Cloud computing with AWS. https://aws.amazon.com/what-is-aws/?nc1=h_ls.

13. AWS WAF, AWS Firewall Manager, and AWS Shield Advanced. Developer Guide. 415 p. https://docs.aws.amazon.com/waf/latest/developerguide/waf-dg.pdf.

14. AWS WAF Security Automations. https://docs.aws.amazon.com/solutions/latest/aws-waf-security-automations/overview.html.

15. AWS WAF Bot Control. https://aws.amazon.com/ru/waf/features/bot-control/?did=ft_card&trk=ft_cardWAF.

16. AWS Lambda. https://aws.amazon.com/ru/lambda/.

17. Spiros Psarris. How to Configure and Use AWS WAF. July 22, 2021. https://www.reblaze.com/blog/how-to-configure-and-use-aws-waf/.

18. Amazon S3. Object storage built to retrieve any amount of data from anywhere. https://aws.amazon.com/s3/?nc1=h_ls.

19. Amazon Athena. https://aws.amazon.com/ru/athena/?whats-new-cards.sort-by=item.additionalFields.postDateTime&whats-new-cards.sort-order=desc.

20. Amazon CloudWatch. https://aws.amazon.com/ru/cloudwatch/.

21. Amazon CloudFront. https://aws.amazon.com/ru/cloudfront/.

22. AWS CloudFormation. https://aws.amazon.com/ru/cloudformation/.

23. Elastic Load Balancing. https://aws.amazon.com/ru/elasticloadbalancing/.

24. Кучер Владислав Ігорович. Технологія захисту інтернет-додатків за допомогою AWS WEB Application Firewall. ВСЕУКРАЇНСЬКА НАУКОВА КОНФЕРЕНЦІЯ «АКТУАЛЬНІ ПРОБЛЕМИ КІБЕРБЕЗПЕКИ». Державний Університет Телекомунікацій. 27 жовтня 2021. Тези доповідей. С. 31 – 34. http://www.dut.edu.ua/uploads/p_2099_79407917.pdf.

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

(Презентація)