Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(повне найменування вищого навчального закладу)
Факультет комп'ютерно-інформаційних систем і програмної інженерії
(назва факультету)
Кафедра кібербезпеки
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр
(освітній рівень)

на тему: *"Аудит безпеки Amazon Selling Partner API"*

Виконав: студент VI курсу, групи СБмз-61

Спеціальності:
125 «Кібербезпека»
(шифр і назва напряму підготовки, спеціальності)

Волошин Р. М.
підпис (прізвище та ініціали)

Керівник *Козак Р. О.*
підпис (прізвище та ініціали)

Нормоконтроль *Лобур Т. Б.*
підпис (прізвище та ініціали)

Завідувач кафедри *Загородна Н.В.*
підпис (прізвище та ініціали)

Рецензент
підпис (прізвище та ініціали)

м. Тернопіль – 2022

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра кібербезпеки
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Загородна Н.В.
(підпис) (прізвище та ініціали)
«__» _____ 2022 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Магістр
(назва освітнього ступеня)

за спеціальністю 125 Кібербезпека
(шифр і назва спеціальності)

Студенту Волошин Роман Миколайович
(прізвище, ім'я, по батькові)

1. Тема роботи Аудит безпеки Amazon Selling Partner API

Керівник роботи Козак Р. О., к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 06 » грудня 2022 року № 4/7-987

2. Термін подання студентом завершеної роботи 14.12.2022

3. Вихідні дані до роботи Наукові публікації про аудит API щодо кібербезпеки

4. Зміст роботи (перелік питань, які потрібно розробити)
Вступ. 1. Application programming interface як складова е-маркетплейсів 1.1 Розвиток електронної комерції (e-commerce). 1.2 Компоненти е-маркетплейсу 1.3 Application Programming Interface (API). 1.3.1 Переваги застосування API для е-маркетплейсів. 1.3.2 Архітектурні стилі API для е-маркетплейсів. 2. Аудит кібербезпеки API. 2.1 RESTful API 2.2 Безпека RESTful API. 2.3 Моделі зрілості API 2.3.1 Модель зрілості API Амундсена. 2.3.2 Модель зрілості API Річардсона. 2.4 Метод виявлення вразливості API. 2.5 Система аудиту безпеки API на основі трафіку. 3. Amazon SELLING PARTNER API аудит. 3.1. Ознайомлення із досліджуваним SAAS-рішенням. 3.2 Архітектура досліджуваного SAAS-рішення. 3.3 Amazon Selling Partner API. 3.4 Event-driven архітектура SP-API. 3.5 Data Protection Policy та її важливість для проходження аудиту. 3.6 Selling Partner API Guard. 3.6 Порівняння результатів. 4. Охорона праці та безпека в надзвичайних ситуаціях. 4.1 Охорона праці. 4.2 Організація оповіщення і зв'язку у надзвичайних ситуаціях техногенного та природного характеру. Висновки. Список літературних джерел. Додатки.

5. Перелік графічного матеріалу. 1. Титулка. 2. Мета та задачі дослідження. 3. Роздрібні онлайн продажі в світі. 4. Розподіл ринку електронної комерції за регіонами. 5. Найкращі онлайн-площадки світу. 6. Графік зростання кількості реєстрацій нових API щорічно. 7. Модель зрілості Річардсона для оцінки відповідності RESTful API. 8. Структура аудиту. безпеки. 9. Двостороння інтеграція "SAAS рішення" із онлайн маркетплейсами та перевізниками 10. Архітектура досліджуваного SAAS рішення. 11. Amazon API Guard. 12. Отримані вразливості (високий пріоритет) щодо застосування SP-API для досліджуваного SAAS рішення. 13. Порівняльний аналіз застосування Amazon API Guard та агрегованого стеку додатків. 14. Висновки.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Осухівська Г.М., к.т.н., доцент		
Безпека в надзвичайних ситуаціях	Клепчик В.М., проректор з адміністративно-господарської роботи та будівництва		

7. Дата видачі завдання 14.11.2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	14.11.2022-15.11.2022	*Виконано*
2.	Підбір наукових джерел про електронну комерцію та аудити API.	16.11.2022-20.11.2022	*Виконано*
3.	Переклад та опрацювання наукових джерел про аудити кібербезпеки існуючих API для маркетплейсів	21.11.2022-23.11.2022	*Виконано*
4.	Виконання аудиту Amazon Selling Partner API щодо кібербезпеки	24.11.2022-27.11.2022	*Виконано*
5.	Оформлення розділу “Application Programming Interface як складова е-маркетплейсів”	28.11.2022-30.11.2022	*Виконано*
6.	Оформлення розділу “Аудит кібербезпеки API”	01.12.2022-04.12.2022	*Виконано*
7.	Оформлення розділу “Amazon Selling Partner Audit”	05.12.2022-07.12.2022	*Виконано*
8.	Виконання завдання до підрозділу «Охорона праці»	08.12.2022-09.12.2022	*Виконано*
9.	Виконання завдання до підрозділу «Безпека в надзвичайних ситуаціях»	10.12.2022-11.12.2022	*Виконано*
10.	Оформлення кваліфікаційної роботи	12.12.2022-13.12.2022	*Виконано*
11.	Нормоконтроль	14.12.2022-15.12.2022	*Виконано*
12.	Перевірка на плагіат	9.12.2022	*Виконано*
13.	Попередній захист кваліфікаційної роботи	16.12.2022	*Виконано*
14.	Захист кваліфікаційної роботи	21.12.2022	

Студент _____ *Волошин Р. М.*
(підпис) (прізвище та ініціали)

Керівник роботи _____ *Козак Р. О.*
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Аудит безпеки Amazon Selling Partner API // Кваліфікаційна робота освітнього рівня «Магістр» // Волошин Роман Миколайович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра кібербезпеки, група СБмз-61 // Тернопіль, 2022 // С. – 73, рис. – 20, табл. – 2 , кресл. – 14, додат. – 1.

Ключові слова: API АУДИТ, AMAZON SELLING PARTNER API, AMAZON API GUARD, RESTful API, SAAS.

В кваліфікаційній роботі вирішується проблема успішного проходження Amazon Selling Partner API аудит, що використовується SAAS рішенням для інтеграції з онлайн-площадкою Amazon. В роботі наведено основні переваги застосування API для маркетплейсів, наведено переваги та недоліки застосування RESTful API та основні його вразливості. Детально розглянуто RESTful API та відповідні моделі зрілості Амундсена та Річардсона. Наведено методи виявлення вразливості API та систему аудиту безпеки на основі трафіку.

Імплементовано аудит існуючого додатку з використанням Amazon API Guard та відповідного стеку додатків в галузі кібербезпеки. Здійснено порівняльний аналіз з використанням критеріїв ціни, тривалості та результативності.

ANNOTATION

Amazon Selling Partner API cybersecurity audit // Qualification work of the educational level “Master” // Voloshyn Roman Mykolayovych // Ternopil Ivan Pulyuy National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Cybersecurity, СБmz-61 group // Ternopil, 2022 // P. – 73, fig. – 20, table – 2, drawing – 14, apendix – 1.

Key words: API AUDIT, AMAZON SELLING PARTNER API, AMAZON API GUARD, RESTful API, SAAS.

The qualification work solves the problem of successfully passing the Amazon Selling Partner API audit, which is used by the SAAS solution for integration with the Amazon online platform. The paper presents the main advantages of using API for marketplaces, the advantages and disadvantages of using RESTful API and its main vulnerabilities. A detailed review of RESTful APIs and the corresponding maturity models of Amundsen and Richardson. API vulnerability detection methods and a traffic-based security audit system are presented.

Implemented an audit of an existing application using Amazon API Guard and the corresponding application stack in the field of cyber security. A comparative analysis was carried out using the criteria of price, duration and effectiveness.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

Amazon MWS	Amazon Marketplace Web Service
API	Application Programming Interface
B2B	Business to Business
CI/CD	Continuous integration / Continuous delivery
CRUD	Create, read, update, delete
HTTPS	Hyper Text Transfer Protocol Secure
IT	Information Technology
OWASP	Open Web Application Security Project
REST	Representational State Transfer
RPC	Remote procedure call
SDK	Software development kit
SQS	Simple Queue Service
SP API	Amazon Selling Partner API
TLS	Transport Layer Security
ОС	Операційна система

ВСТУП

Актуальність теми. Роздрібна торгівля, електронна комерція, онлайн покупки – цим сьогодні вже нікого не здивуєш. Людям значно легше замовити товар в інтернет магазині чи на відомому українському сервісі Rozetka та отримати доставкою від Нової пошти. Проте за цією легкістю купівлі схована складна архітектура, значна кількість сервісів, які інтегруються один з одним завдяки API. Враховуючи тенденції щодо розвитку онлайн маркетплейсів та збільшенню товарообігу через інтернет появляється питання захисту персональної інформації покупців і архітектурних рішень систем, API. Оскільки, значна частина сервісів використовують для обміну інформацією RESTful API, то дослідження теми аудити кібербезпеки для API найбільшого маркетплейсу в світі Amazon Selling Partner API є надзвичайно актуальним.

Мета і задачі дослідження. Метою даної кваліфікаційної роботи освітнього рівня «Магістр» є успішне виконання аудиту Amazon Selling Partner API, для отримання доступу до усіх наявних його методів при реалізації SAAS рішення.

Для досягнення поставленої мети було потрібно виконати наступні завдання:

- Дослідити моделі зрілості API;
- Провести огляд переваг та недоліків RESTful API;
- Проаналізувати методологію аудитів API на основі трафіку;
- Проаналізувати вимоги щодо успішного проходження аудиту на основі політик Amazon;
- Дослідити можливості Amazon API Guard;
- Провести експериментальні дослідження та отримати результати аудиту;
- Здійснити порівняльний аналіз інструментів для аудиту API.

Об’єкт дослідження. Процес аудиту кібербезпеки API.

Предмет дослідження. Amazon Selling Partner API та його вимоги щодо проходження аудиту.

Наукова новизна одержаних результатів кваліфікаційної роботи полягає у тому, що отримано порівняльні результати на основі експериментального виконання аудиту щодо доцільності використання нативних інструментів Amazon чи агрегації існуючих додатків, що дозволяє виявляти вразливості Amazon Selling Partner API з меншими втратами часу та грошей.

Практичне значення одержаних результатів. Надано рекомендації щодо доцільності застосування відповідних інструментів аудиту API відповідно до реалізації архітектури веб додатку чи рішення.

Апробація результатів магістерської роботи. Основні результати проведених досліджень обговорювались на: X науково-технічній конференції «Інформаційні моделі, системи та технології» (м.Тернопіль, 7-8 грудня, 2022).

Публікації. Основні результати кваліфікаційної роботи опубліковано у одній праці конференції (див. Додаток А).

Структура й обсяг кваліфікаційної роботи. Кваліфікаційна робота складається зі вступу, чотирьох розділів, висновків, списку літератури із 23 найменувань та 1 додатка. Загальний обсяг кваліфікаційної роботи складає 77 сторінки, з них 73 сторінок основного тексту, який містить 20 рисунки та 2 таблиці.

1 APPLICATION PROGRAMMING INTERFACE ЯК СКЛАДОВА Е-МАРКЕТПЛЕЙСІВ

1.1 Розвиток електронної комерції (e-commerce)

За останні роки електронна комерція надзвичайно швидко набула популярності завдяки зручності та легкості обміну товарами та послугами в різних регіонах світу. Як і інші сегменти ринку, роздрібний сектор також змінюється, переживаючи своєрідну значну революцію внаслідок цифрової трансформації та появи передових технологій [1].

Електронна комерція стає невід'ємною частиною бізнес-стратегії, де вона робить значний внесок в економічний розвиток країни. Об'єднання концепції інформаційних і комунікаційних технологій у бізнесі призвело до революції у відносинах між окремими особами та організаціями, а також всередині організацій. Удосконалення та впровадження таких технологій, як штучний інтелект і концепції великих даних, у підприємствах забезпечує масове налаштування, збільшення кількості потенційних клієнтів, а також підвищення продуктивності та отримання прибутку для всього бізнесу.

Очікується, що індустрія електронної комерції зростатиме завдяки ряду інших факторів, таких як зростання Інтернет-послуг із високошвидкісною смугою пропускання, безпрецедентне проникнення смартфонів (m-commerce, mobile-commerce), стабільність ринку, персоналізація продуктів і локалізація, а також зниження тарифів на Інтернет. Крім того, очікується, що збільшення безперебійних Інтернет-послуг (4G, технологія StarLink) збільшить доступ споживачів до онлайн-покупок, що сприятливо вплине на загальне зростання ринку електронної комерції.

На більшості ринків споживачі віддають перевагу покупкам на онлайн-платформам, а не звичайним варіантам придбання. Головним чином це можна пояснити тим фактом, що при такому варіанти покупки пропонують різноманітний асортимент продуктів, чудову зручність для клієнтів, безперебійні

послуги клієнтам. Також сучасні онлайн-покупці акцентують увагу на таких факторах, як гарантія та швидка доставка у своїх рішеннях про покупку, а електронна комерція задовольняє ці вимоги клієнтів [2]. Щоб задовольнити мінливий конкурентний сценарій для компаній стратегічно важливо позиціонувати себе на ринку електронної комерції.

Триваюча пандемія COVID-19 експонентно прискорила зростання онлайн-продажів і вплинула на загальну поведінку споживачів у всьому світі. Щоб стабілізувати медичну ситуацію, було посилено кілька обмежень, одним із яких був карантин по всьому світу, через що цифрові канали стали альтернативними рішеннями для покупок. При цьому криза ще більше підвищила загальні продажі, тим самим максимізуючи прибутки та доходи, отримані компаніями електронної комерції в різних регіонах світу.

Також очікується, що зростання кількості малих і середніх підприємств також сприятиме зростанню сегмента електронної комерції. На основі звітної інформації прогнозується, що в 2023 році на такі регіони, як Північна Америка та Азіатсько-Тихоокеанський регіон, припадатиме більша частина загального обсягу роздрібних продажів електронної комерції [3].

У 2021 році споживачі в усьому світі витратили майже 5,21 мільярдів доларів на онлайн-продажі, що свідчить про значне зростання у порівнянні з показником попереднього 2020 року (рис.1.1).

На основі географії ринок сегментований на регіони Північної Америки, Європи, Азіатсько-Тихоокеанського регіону, Латинської Америки, Близького Сходу й Африки. Частка Азіатсько-Тихоокеанського регіону становила майже 40,92% від загального ринку (рис. 1.2). Це в основному пов’язано зі збільшенням переваг серед галузей щодо здійснення своїх операцій, зокрема, через платформу електронної комерції B2B.

У 2019 році на Північну Америку припадало майже 24,99% загальної частки, що свідчить про значне зростання разом із європейськими регіонами. В період пандемії найбільш популярними були товари з сегменту харчової промисловості та напоїв, краси та косметики, моди, засобів особистої гігієни та

фармацевтики. Ринок електронної комерції в Азіатсько-Тихоокеанському регіоні, оцінений у 3,69 трильйона доларів. Це можна пояснити збільшенням кількості користувачів Інтернету та поширенням смартфонів у різних регіонах.

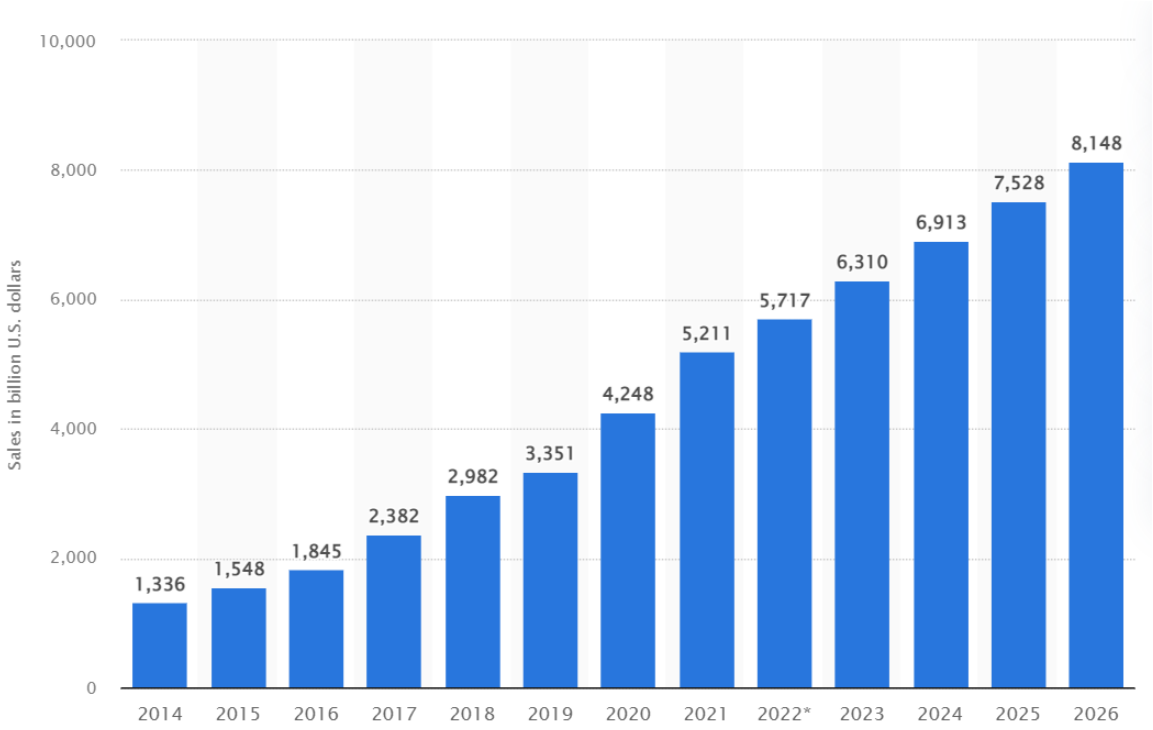

Рисунок 1.1 – Роздрібні онлайн продажі в усьому світі з 2014 по 2026 рік [3]

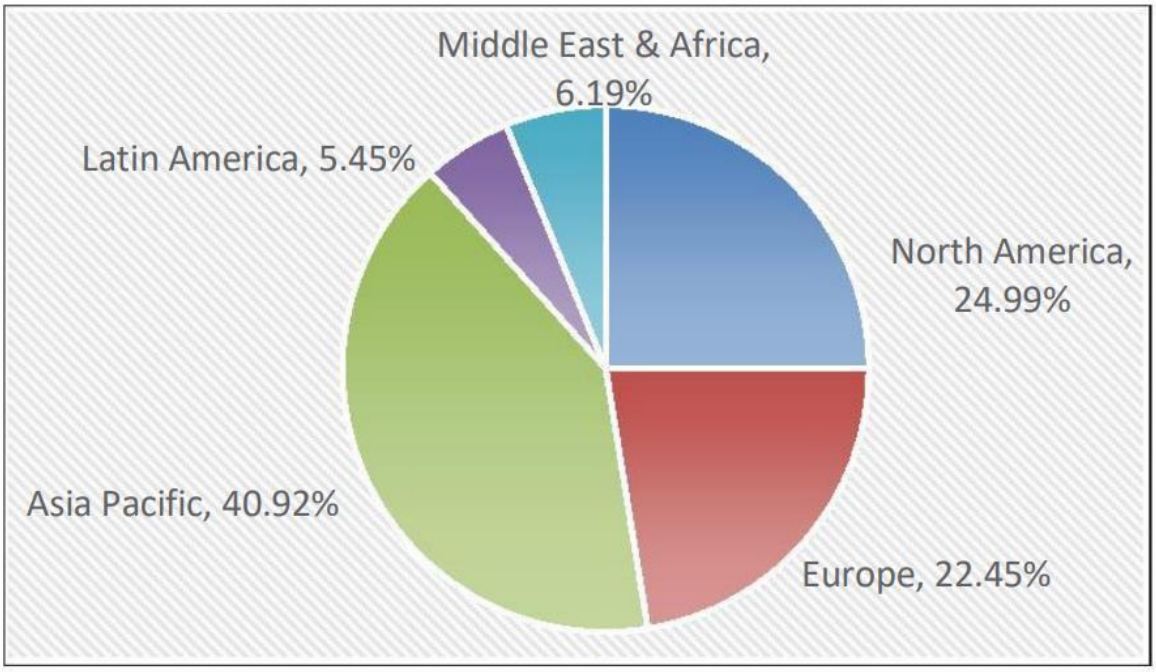

Рисунок 1.2 – Розподіл ринку електронної комерції за регіонами [4]

1.2 Компоненти е-маркетплейсу

Е-маркетплейс (електронний ринок, маркетплейс) – інтернет-ринок, зазвичай B2B (business-to-business), на якому покупці та продавці обмінюються товарами або послугами. Зазвичай існують три типи е-маркетплейсів: приватні, державні та консорціуми.

Е-маркетплейси відіграють центральну роль в економіці, сприяючи обміну інформацією, товарами, послугами та грошовими транзакціями. У процесі своєї роботи та використання вони створюють економічну цінність для покупців, продавців, ринкових посередників і для суспільства в цілому.

З розвитком електронної комерції з'являються нові ринки. Найбільші гравці пропонують практично всі види товарів, які можна придбати у звичайних магазинах. Їхні платформи часто пропонують товари, доступні лише на відповідній електронній площадці. Наприклад, маркетплейс Etsy спеціалізується на виробах ручної роботи. З метою не втрачати таку цільову аудиторію Amazon запустив окремий каталог – Amazon Handmade. Однак, як і існують спеціалізовані інтернет-магазини, маркетплейси також стають сегментованими. Таким чином, були запущені платформи, зосереджені на конкретних продуктах і бізнес-сферах: мода (LaModa, Kasta), електроніка (Rozetka, Alibaba, Allegro), предмети для дому чи ручної роботи.

Е-маркетплейси виконують три основні функції:

- співставлення (зустріч) покупців і продавців;
- сприяння обміну інформацією, товарами, послугами та платежами, пов'язаними з ринковими операціями;
- забезпечення інституційної інфраструктури (захисту прав споживачів та їх персональних даних), такої як законодавча та нормативна база, яка забезпечує ефективне функціонування маркетплейсу як ринку.

Основним місце здійснення електронних платежів є маркетплейс, саме тому під час пандемії COVID-19 спостерігалося різке зростання застосування IT та

відповідно оновлених систем захисту кожної онлайн покупки та транзакції. Така тенденція призвела до:

- більшої інформаційної насиченості транзакційного та реляційного середовища;
- зниження витрат покупців на пошук інформації;
- зменшення інформаційної асиметрії між продавцями та покупцями;
- більшого часового розриву між часом покупки та часом володіння фізичними продуктами, придбаними на електронному ринку;
- меншої тривалості між часом покупки та часом володіння цифровими продуктами, придбаними на електронному ринку;
- можливості покупців і продавців перебувати в різних місцях.

Основними компонентами маркетплейсів вважають (рис. 1.3):

- клієнти: 1,6 мільярда людей у всьому світі, що користуються інтернетом, є потенційними покупцями товарів і послуг. Ці споживачі шукають вигідні пропозиції, індивідуальні речі, предмети колекціонування, розваги, соціалізацію тощо.
- продавці: існують мільйони вітрин магазинів, які рекламують і пропонують величезну різноманітність товарів. Ці магазини належать компаніям, державним установам або приватним особам. Продавці можуть продавати безпосередньо зі своїх веб-сайтів або з електронних ринків.
- продукти та послуги. Однією з головних відмінностей між ринком і електронним ринковим простором є оцифрування продуктів і послуг. Хоча обидва типи ринків можуть продавати фізичні продукти, ринковий простір також може продавати цифрові продукти, тобто товари, які можна трансформувати в цифровий формат і миттєво доставляти через інтернет.
- інфраструктура включає електронні мережі, апаратне забезпечення (сервери), програмне забезпечення.
- інтерфейс (front-end), за допомогою якого клієнти взаємодіють із маркетплейсом. Компоненти інтерфейсу можуть включати портал продавця,

електронні каталоги, кошик для покупок, пошукову систему, систему аукціону та платіжний шлюз.

- ядро (back-end): виконуються усі дії, пов’язані з агрегацією та виконанням замовлень, управлінням запасами, закупівлями у постачальників, бухгалтерським обліком і фінансами, страхуванням, обробкою платежів, пакуванням і доставкою [5].
- посередники: третя сторона, яка діє між продавцями та покупцями. Роль електронних посередників часто відрізняється від ролі звичайних посередників (таких як оптовики). Наприклад, онлайн-посередники допомагають зіставляти покупців і продавців, надають деякі інфраструктурні послуги та допомагають клієнтам і/або продавцям ініціювати та завершувати транзакції.
- ділові партнери: перевізники (Нова пошта, FedEx, USPS), провайдери банківських послуг (МоноБанк, ПриватБанк, Chase).
- служби підтримки: надають послуги сертифікації та депонування (для забезпечення безпеки) до постачальників контенту.

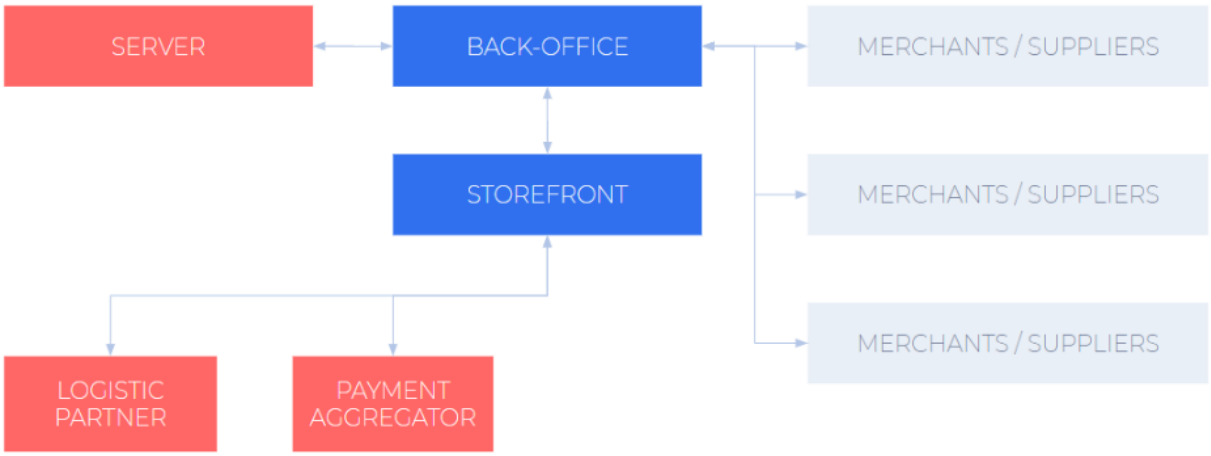

Рисунок 1.3 – Класична архітектура е-маркетплейсу [6]

Сполучені Штати є найбільшою економікою світу та другими після Китаю ринком електронної комерції з річним обсягом продажів у 792 мільярди доларів. Онлайн-роздрібна торгівля становить 14% від загального обсягу роздрібних

продажів. Багато світових компаній електронної комерції, таких як Amazon і eBay, виникли в США і відповідно ця країна є лідером світових технологій [7].

Електронна комерція в США характеризується низкою проблем. Перш за все – низька щільність населення та великі відстані між великими містами призводять до того, що доставка є дорожчою та тривалішою. Податки з продажів та інші закони значно відрізняються між штатами, створюючи перешкоди для торгівлі. Ці фактори можуть є причиною того, що онлайн-роздрібна торгівля як частка загальної роздрібної торгівлі набагато вища в Китаї (25%) і Великобританії (23%), ніж у США.

Перші п’ять основних гравців на ринку електронної комерції в США представлені на рис. 1.4.

#	TYPE	NAME	REGION/COUNTRY	PRODUCT CATEGORY	US VISITS/MONTH
1		Amazon	Global	General	2.0B
2		eBay	Global	General	688.9M
3		Walmart.com	USA	General	388.8M
4		Etsy	Global	Arts, Crafts & Gifts	238.4M
5		Target.com	USA	General	175.7M

Рисунок 1.4 – Найбільші онлайн-маркетплейси в США [8]

Безсумнівно, Amazon домінує на ринку електронної комерції США, маючи понад 2 мільярди відвідувачів на місяць. eBay є на другому місці з приблизно 689 мільйонами відвідувачів на місяць. Amazon є лідером, навіть враховуючи, що eBay є чистим ринком і на Amazon припадає 55% від загального обсягу продажів.

Наступним у списку є роздрібний торговець Walmart, а Target — на двох місцях внизу, але ці два маркетплейси мають дуже різні стратегії електронної

комерції. Електронна комерція Walmart була відроджена після придбання Jet.com, а після багатьох років занедбаності ринок Walmart зараз стрімко зростає.

Дев’ять найбільших основних гравців на ринку електронної комерції в світі представлені на рис. 1.5.

#	Type	Name	Region/Country	Product Category	Visits/month
1		Amazon	Global	General	5.69B
2		eBay	Global	General	2.98B
3		Shopee	Southeast Asia	General	631.19M
4		Rakuten	Global	General	590.84M
5		AliExpress	Global	General	526.4M
6		Walmart	North America	General	514.03M
7		Mercado Libre	Latin America	General	446.97M
8		Etsy	Global	Arts, Crafts & Gifts	397.5M
9		Taobao	China	General	333.14M

Рисунок 1.5 – Найкращі онлайн-ринки світу [9]

В кінці 2022 року багато нових маркетплейсів закриваються, деякі функції існуючих будуть замінені. Після 23 років роботи китайський веб-сайт електронної комерції eachnet.com офіційно закривається. Менш ніж через шість місяців після свого дебюту в Індії мережа соціальної комерції Shopee покидає країну через “невизначеність світового ринку”. Shopify переходить на іншу програму дропшиппінгу, створену DSers. Згідно з повідомленням Facebook, його функція живих покупок буде припинена 1 жовтня на користь Reels. Внаслідок цього багато продавців переходять на багатоканальну стратегію продаж (multi-channel selling).

Ще одна нова тенденція, заснована на присутності продавця на різних маркетплейсах. Проте продавець стикається з проблемою синхронізацію кількості товару та ціни на кожній з площадок електронної комерції. Саме тому дуже важливо е-маркетплейсу наявність АР інтерфейсу для автоматичного та періодичного оновлення інформації щодо товарів.

1.3 Application Programming Interface (API)

Інтерфейс прикладного програмування (API) – це набір команд, функцій, протоколів і об’єктів. API взаємодіє із зовнішніми системами, виконуючи загальні операції. API є гнучкий, простий у використанні та ефективний інтерфейс. По суті є мостом між модулями, програмним забезпеченням і розробниками, API є невід’ємною частиною сучасних мобільних додатків, SAAS (Software as a service) і веб-додатків. Він широко використовується в орієнтованих на клієнта, партнерів і внутрішніх додатках [10], таких як банківська справа, роздрібна торгівля, безпілотні автомобілі та розумні будинки. З безперервним поглибленням мережевих процесів різних галузей модель обслуговування API стає все більш популярною.

В даний час за допомогою інструментів API можна швидко створювати програми, а це означає, що навіть недосвідчені розробники можуть розгортати або інтегрувати програми. Такий гнучкий розвиток, як правило, не має надійного дизайну безпеки чи вказівок з інтеграції додатків, і вони навіть не повністю враховують вплив на безпеку та можуть виявити вразливість логіки додатка. Наприклад, під час розробки та впровадження розширень неправильні обмеження ресурсів або дозволів можуть призвести до атак на відмову в обслуговуванні. Таким чином, широке використання API збільшило ризик безпеки користувачів і витоку конфіденційності, і API поступово стали ціллю для кібер-зловмисників [11].

Незважаючи на те, що програма може реалізувати надійну перевірку введення та контроль доступу у своєму коді, ці дані зазвичай не копіюються в

тому самому класі, коли надсилаються на сервер через API. Тому зловмисник може обійти контроль клієнта. Атаки через API здебільшого здійснюють вхід із юридичною особою, а потім імітують звичайні операції, такі як багато джерельні та низькочастотні запити. Таким чином, механізми безпеки, надані традиційними шлюзами безпеки API, такі як автентифікація особи, керування повноваженнями, обмеження швидкості та перевірка вмісту запиту, не можуть відповідати вимогам безпеки. Безпека API є фундаментальною частиною безпеки мережі. Без безпечного API неможливо досягти швидких інновацій.

1.3.1 Переваги застосування API для е-маркетплейсів

Приблизно половину всього веб-трафіку генерують боти. Боти набувають багатьох форм, іноді таких простих, як сценарій Bash, що складається з команд curl, іноді у вигляді сценарію безголовного браузера, такого як PhantomJS або іноді навіть у вигляді широкомасштабного поширеного веб-сканера, що працює на базі, як-от Apache Nutch.

API маркетплейсу – це готовий інтерфейс для ефективного з’єднання виробників API та споживачів API, і, як і будь-який інший електронний ринок, він надає інструменти та засоби для цього. Хоча сьогодні існують деякі загальнодоступні торгові майданчики API, які надають каталог існуючих загальнодоступних API. При цьому інтерфейс може бути доступним для всіх користувачів, закритим чи надаватись у користування лише певній когорті користувачів, що пройшли відповідну перевірку.

На основі аналізу кількості впровадження API, наданого ProgrammableWeb.com основним ресурсом спільноти для аматорів і професіоналів у галузі програмування веб-додатків, робимо висновок щодо надзвичайної популярності застосування цього інтерфейсу. Цей ресурс збирає загальнодоступні кінцеві точки API у повному каталозі з інформацією, яку повідомляють розробники. На рис.1.6 показано кількість записів веб-API, зареєстрованих з 2005 року і як показує тенденція кількість нових зареєстрованих

інтерфейсів буд продовжувати зростати. Станом на кінець 2022 року застосування API в галузі електронної комерції займає 7 місце.

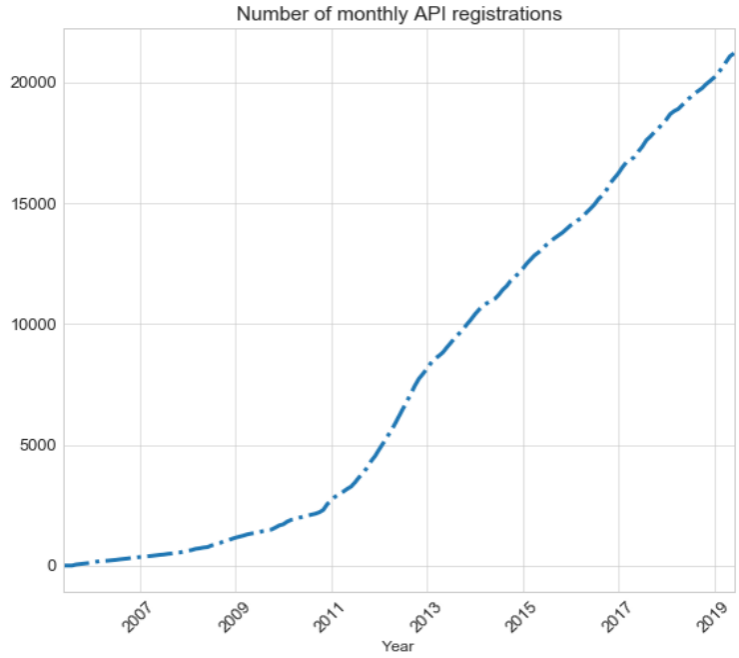

Рисунок 1.6 – Графік зростання кількості реєстрацій нових API щорічно [12]

Зростання реєстрації API як складової онлайн маркетплейсу спричинено наступними причинами:

- API є невід'ємною частиною ведення онлайн-бізнесу, що керується даними. Інтерфейс дозволяє бізнес-користувачам використовувати програмне забезпечення та додатки для підвищення продуктивності та підвищення прибутку. Від інструментів соціальної співпраці до більш інноваційних підходів до охоплення клієнтів, використання API приносить користь підприємству.
- забезпечує покращення швидкості, гнучкості, узгодженості та точності, тобто більш плавну інтеграцію бізнес-процесів у додатках у поєднанні з іншими типами технологій B2B.
- швидкий обмін документами як частина динамічних бізнес-транзакцій, наприклад, замовлення на купівлю (Purchase Order, PO), вказує на еволюцію електронної комерції B2B.

1.3.2 Архітектурні стилі API для е-маркетплейсів

API можна розділити на такі основні типи: RPC API та API, які дотримуються архітектурного стилю REST, або RESTful API.

REST API характеризується набором процедур або методів, які клієнтська програма може викликати і які виконуються сервером для виконання завдання, наприклад, обмін даними або виклик служби перевірки даних (Amazon, eBay, Etsy, Walmart).

RPC API по суті працюють шляхом заміни обміну повідомленнями об’єктів у пам’яті міжмережевими об’єктними повідомленнями в об’єктно-орієнтованих програмах [13]. У двох словах, це можна проілюструвати, розглядаючи використання бібліотеки коду не в локальному середовищі, а через мережу, таким чином надсилаючи/отримуючи повідомлення до/з бібліотеки коду через мережу, а не через локальну пам’ять (Walmart Canada).

На рис. 1.7 представлено розподіл застосування архітектурних стилів API.

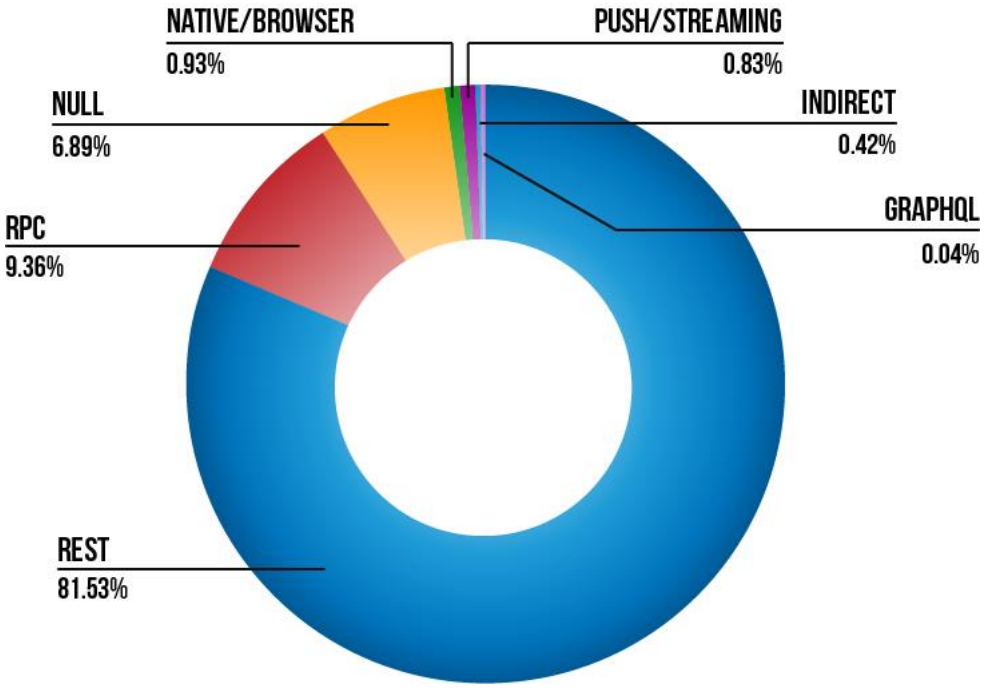

Рисунок 1.7 – Діаграма розподілу застосування архітектурних стилів API [14]

Також виділяють наступні типи API, які застосовуються в електронній комерції:

- браузерний API – це API, які доступні в локальній системі, середовищі виконання або API ОС, які можна інтегрувати в веб-програми та мобільні програми. Наприклад, API JavaScript у веб-браузері або API для конкретного пристрою, як-от для акселерометра у смартфоні.
- Push/Streaming – це для API реального часу, де дані передаються назад до системи виклику за допомогою таких технологій, як Websockets або Webhooks (Shopify, eBay, Amazon).
- GraphQL – це новіша технологія API, яку ми нещодавно додали до нашого списку архітектурних стилів. API GraphQL мають здатність використовувати один запит, щоб отримати необхідні дані з кількох ресурсів, як у гіпермедіа. Але API GraphQL також запозичують концепції, які ми спостерігали як в REST, так і в API у стилі RPC (Shopify).

Очікувано, що REST є найбільш часто використовуваним архітектурним стилем, враховуючи його видатність у дизайні API за останнє десятиліття плюс. Наступним за частотою використання є RPC, завдяки багатьом API у стилі SOAP і XML-RPC, які з’явилися, особливо на початку існування нашого каталогу.

2 АУДИТ КІБЕРБЕЗПЕКИ API

2.1 RESTful API

Моделі зрілості API можна використовувати для оцінки рівня відповідності деяким принципам, визначеним самою моделлю. Це корисно не лише для постачальників (наприклад, аналіз прогалин або планування вдосконалення), але й для користувачів, щоб вибрати, який конкретний API використовувати та, загалом, як і коли його можна використовувати найефективнішим способом. Існують дві різні моделі для оцінки цих аспектів: модель зрілості Амундсена та модель зрілості Річардсона.

REST (Representational State Transfer) – це стиль архітектури API для створення веб-служб і сервісів. Він був представлений у 2000 році в докторській дисертації [16] одним із засновників протоколу HTTP Роєм Філдінгом. REST – це простий інтерфейс для передачі інформації, який не використовує рівні програмного забезпечення сторонніх розробників. Тобто при відправці даних немає етапу конвертації, інформація надходить у вихідному вигляді, що позитивно впливає на навантаження клієнта, але додає навантаження на мережеву частину.

Перевагами REST API є наступні пункти:

- відкритість взаємодії;
- проста реалізація;
- кешування даних на рівні HTTP;
- можливість працювати з кількома форматами представлення даних;
- стабільність за рахунок високого рівня абстракції.

До недоліків REST API належать наступні особливості:

- відсутність єдиної стандартизованої структури;
- високе навантаження на мережу;
- надлишок або недостатність інформації, що може призвести до необхідності надсилання додаткового запиту.

Робота з даними організована у форматах JSON або XML. Веб-сервіс, який відповідає всім вимогам і умовам архітектури REST, називається веб-сервісом RESTful.

Архітектурними вимогами RESTful (Fielding Constraints) API є наступні:

- не повинно міститись інформація щодо стану (без стану): стан для обробки запиту може бути лише в самому запиті, тому сервер не зберігає жодної інформації про сеанс;
- кешування;
- загальний інтерфейс: забезпечує послідовну, незалежну від програми взаємодію з веб-сервером.

Також існує перелік підобмежень: управління ресурсами через їх представлення, ідентифікація ресурсів, “самодостатність” повідомлень, робота з гіпермедіа:

- сконцентрованість на архітектурі клієнт-сервер.
- можливість доставки виконуваного коду на сторону клієнта.
- незалежність від кількості рівнів програми чи додатку.

У REST весь зв’язок базується на використанні методів HTTP: GET, POST, PUT, PATCH і DELETE.

REST найчастіше використовується як API керування для операцій CRUD (Create, Read, Update, Delete) та для реалізації взаємодії з ресурсами в легких масштабованих службах. Ресурс зазвичай є об’єктом моделі даних або таблицею бази даних.

З усіх специфікацій REST реалізує найвищий рівень абстракції та найкраще підходить для розробки простішого CRUD API. Він підтримує баланс надійності та простоти у використанні. Перехід до клієнт-серверної системи, без статистики та схильність до кешування (важливо для переліку досягнень, сортування та фільтрації) також є перевагами для таких служб. Велике, в порівнянні з іншими специфікаціями, навантаження на мережу нівелюється специфікою вирішуваної

задачі. Саме тому REST є основою архітектурного стилю прототипу проекту, що розробляється.

2.2 Безпека RESTful API

Створення безпечних RESTful API також накладає певні стандартні вимоги:

- використання протоколу HTTPS: криптооперація необхідна для забезпечення цілісності переданих даних;
- Rate-limits: Необхідно стежити за навантаженням на API. Скидання запитів при перевантаженні або підключення додаткових систем для балансування;
- автентифікація: ідентифікація користувача / програми / пристрою;
- журнал аудиту: запис дій шляхом створення запису у файлі журналу;
- контроль прав доступу (авторизація) або ж визначення прав доступу для роботи з ресурсами;
- доступ до бізнес-логіки програми.

При роботі з архітектурою REST прийнято виділяти два рівні безпеки, оскільки REST API є інтерфейсом для мережевої взаємодії з веб-додатком:

- перший рівень - отримання доступу до API;
- другий рівень - отримання доступу безпосередньо до програми.

Захист рівня API передбачає правильну організацію аутентифікації, авторизації, реєстрації та розділення прав доступу. Розробник повинен переконатися, що лише авторизовані клієнти можуть використовувати API і мати доступ лише до авторизованих операцій.

Безпека прикладного рівня включає перевірку вразливостей кінцевих точок служби – URL-адрес, відповідальних за взаємодію з інтерфейсом.

Згідно даних спільноти OWASP (Open Web Application Security Project) існує (на різних рівнях) близько десяти вразливостей розробки API [17]. Найбільш небезпечними є:

- недостатнє ведення журналів і моніторинг: відсутність або неправильне бачення файлу журналів і моніторингу системи.
- порушена авторизація на рівні об’єкта: відсутність поділу та примусового контролю доступу до ресурсів.
- порушена автентифікація користувача: уразливості, пов’язані з автентифікацією користувача.
- відсутність ресурсів і обмеження швидкості: невідповідність перевірок і обмежень.
- порушена авторизація функціонального рівня: відсутність контролю доступу.
- масове призначення: уразливості, пов’язані з десеріалізацією отриманих об’єктів.
- неправильна конфігурація безпеки: помилки в роботі з налаштуваннями безпеки програми.

Перш за все, необхідно звернути увагу на проблеми, пов'язані з контролем доступу, аутентифікацією та авторизацією. Згідно з дизайном REST API не має стану, тому доступ контролюється через локальні кінцеві точки.

Існує кілька найпоширеніших методів (схем автентифікації): базова автентифікація, ключ API, веб-токени JSON, OAuth 2.0, автентифікація на основі токенів, автентифікація на основі файлів cookie та OpenID.

Базова автентифікація (Basic Authentication, BA) – це найпростіша схема автентифікації на основі HTTP. Клієнт або додаток формує HTTP-запит, у заголовку якого міститься поле “Авторизація”. Рядок у формі: Basic <ім'я користувача: пароль> (закодований у Base64) передається як значення цього поля. Для забезпечення безпеки та збереження конфіденційності даних у поєднанні з базовою автентифікацією завжди необхідно використовувати захищені протоколи HTTPS/TLS [18], оскільки ідентифікатор користувача та пароль входять до Мережі у вигляді звичайний текст, закодований у Base64 (який просто піддається декодуванню).

OAuth 2.0 – це найбільш складний протокол, який відповідає за авторизацію користувачів. Дозволяє веб-програмі отримувати права на використання ресурсів клієнта в іншій службі. Це дає можливість надати сторонній програмі обмежений доступ до контрольованих ресурсів користувача без безпосередньої передачі логіна та пароля цій програмі. OAuth можна використовувати на будь-якій платформі, оскільки протокол спирається на базовий стек веб-технологій, а саме на HTTP-запити, відповіді, перенаправлення URL-адрес.

Стандарт OAuth визначає різні процеси (потоки) для досягнення делегованої авторизації. Ці потоки істотно відрізняються тим, як клієнтська програма отримує необхідну авторизацію. Надання авторизації – це облікові дані, що представляють дозвіл власника ресурсу на доступ до захищеного ресурсу. OAuth визначає чотири типи дозволу авторизації:

- код авторизації;
- неявні дані;
- облікові дані пароля власника ресурсу;
- облікові дані клієнта.

В залежності від того, який дозвіл авторизації використовується, визначається відповідний потік OAuth і виконано.

Після отримання дозволу на авторизацію клієнт використовує його для запиту маркера доступу для сервера авторизації. Маркер доступу – це облікові дані, які клієнт може використовувати для доступу до ресурсу на сервері ресурсів. Хоча OAuth 2.0 спеціально розроблено для роботи через HTTP, у стандарті не визначено конкретних API. Доступний звіт про оцінку CAMSS для OAuth 2.0 [19], який класифікує OAuth 2.0 як відповідний до Регламенту ЄС щодо європейської стандартизації.

Ключ API – ключ у вигляді рядка символів, який користувач передає разом із запитом на сервер. Сервер підтвердить ідентичність клієнта, якщо його ключ міститься в базі даних програми. Сам ключ видається програмою при реєстрації користувача. Ця схема використовується для захисту від несанкціонованого

доступу та дозволяє накласти обмеження на виклики API. Ключ API можна передати: як параметр запиту, у заголовку запиту, як значення cookie.

Загалом ключі API не вважаються безпечними. Фактично, після надання ключа сервери не мають контролю над тим, наскільки безпечно використовується ключ; наприклад, оскільки ключі API зазвичай доступні для клієнтів, комусь легко вкрасти ключ API та використати його. Провайдер не може перевірити, чи вхідний запит використовує вкрадений ключ.

Аутентифікація на основі файлів cookie – метод, заснований на перевірці вмісту файлів cookie, всередині яких зберігається вся необхідна інформація про сеанс. Користувач ініціює запит на вхід. У разі успішного входу сервер надсилає відповідь, у заголовку якої міститься поле Set-Cookies, що містить назву поля cookie, значення, термін дії cookie тощо. Наступного разу, коли користувачеві знадобиться отримати доступ до API, він передасть значення збереженого поля Cookie JSESSIONID із ключем “Cookie” у заголовку запиту.

JSON Web Tokens (JWT) – це механізм автентифікації, заснований на використанні спеціального типу токена, а саме: токена JWT. Це структура даних JSON. Такий токен містить заголовок із загальною інформацією, тіло з корисним навантаженням (ідентифікатор користувача, група, дані) і криптографічний підпис. Ця схема є одним із найбезпечніших механізмів передачі даних між двома сторонами, тому вона вважається кращим методом контролю доступу до REST API. Користувач для роботи з API, надсилаючи запит, додає до нього персональний JWT, попередньо виданий сервером.

OpenID – це стандартизований метод децентралізованої схеми автентифікації. Відмінною особливістю є можливість створення єдиного облікового запису для аутентифікації відразу на декількох сервісах (створення унікального цифрового ідентифікатора) через послуги сервісу-посередника. За механізмом роботи цей метод схожий на OAuth 2.0, але OpenID призначений виключно для аутентифікації користувача. Нова версія OpenID Connection була трансформована в надбудову для аутентифікації через OAuth 2.0, отримала надійний механізм шифрування та цифрові підписи.

2.3 Моделі зрілості API

Моделі зрілості API можна використовувати для оцінки рівня відповідності деяким принципам, визначеним самою моделлю. Це корисно не лише для постачальників (наприклад, аналіз прогалин або планування вдосконалення), але й для користувачів, щоб вибрати, який конкретний API використовувати та, загалом, як і коли його можна використовувати найефективнішим способом. Існують дві різні моделі для оцінки цих аспектів: модель зрілості Амундсена та модель зрілості Річардсона.

2.3.1 Модель зрілості API Амундсена

Значно відомою моделлю зрілості дизайну є модель Амундсена (Amundsen, 2017). Ця модель визначає рівні відповідності на основі ступеня абстрагування наданого API від базової реалізації. Рівні відповідності моделі Амундсена є наступними:

- рівень 0: орієнтований на дані — внутрішня модель впровадження безпосередньо доступна на рівні API;
- рівень 1: об’єктоцентричний — API безпосередньо не відкриває внутрішню модель, але надає засоби (методи) для маніпулювання об’єктами внутрішньої моделі;
- рівень 2: орієнтований на ресурси — API описується як набір ресурсів, які можуть споживатися клієнтськими програмами; на цьому рівні ресурси незалежні від об'єктів внутрішньої моделі;
- рівень 3: орієнтований на доступ — API описується як набір ресурсів, які використовують гіпермедійні представлення для надання доступних дій (операцій і посилань), які можна виконати на описаному ресурсі.

Рівні 0 і 1 вважаються внутрішніми моделями, оскільки вони розкривають внутрішні структури реалізації на рівні API. Рівні 2 і 3 вважаються зовнішніми моделями, оскільки вони відокремлюють зовнішню модель, яку надає API, від внутрішньої моделі, яка використовується реалізацією.

2.3.2 Модель зрілості API Річардсона

Повністю сумісний RESTful API повинен задовольняти всім обмеженням, визначеним архітектурним стилем REST. Модель зрілості Річардсона [16] є добре відомою моделлю для оцінки відповідності впровадження RESTful API. Модель визначає чотири рівні зрілості впровадження (рис. 2.1).

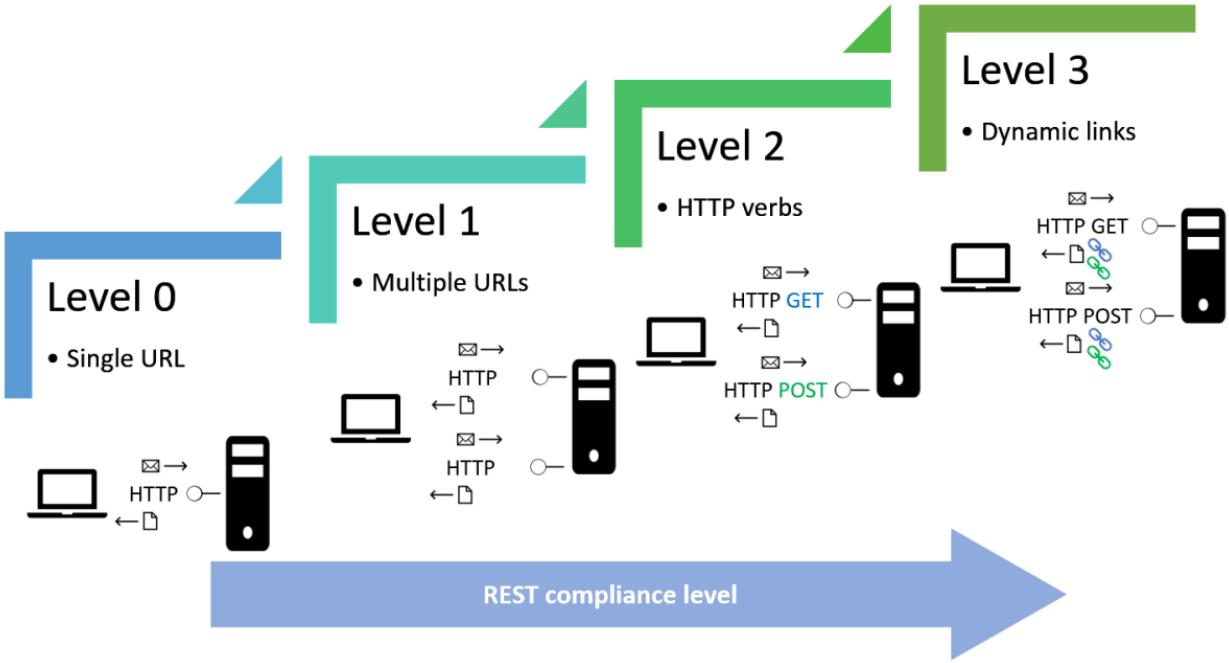

Рисунок 2.1 – Модель зрілості Річардсона [20] для оцінки відповідності RESTful API

Рівні зрілості моделі є наступними:

- рівень 0: API на цьому рівні використовують HTTP як транспортний протокол для віддаленої взаємодії, як правило, з єдиною кінцевою точкою, опублікованою сервером; по суті, це API RPC через топологію мережі, побудовану навколо протоколу HTTP;
- рівень 1: замість використання однієї кінцевої точки на цьому рівні API використовують різні кінцеві точки для різних ресурсів;
- рівень 2: на цьому рівні API використовують дієслова HTTP для характеристики запитуваної операції;

- рівень 3: на цьому рівні API використовують HATEOAS, тобто вони використовують гіпермедіа для керування переходами програми; відповіді від сервера містять посилання для доступних операцій.

2.4 Метод виявлення вразливості API

API було створено для мережевого зв'язку, тому вразливості в API в основному походять від аномалій потоку даних. Поширення конфіденційних даних через Інтернет вимагає суворої перевірки потоку даних у пов'язаних API. Тому виявлення витоків і поглиблене відстеження конфіденційних даних є важливим питанням у дослідженнях безпеки API. У традиційних дослідженнях відстеження потоків даних метод аналізу потоку даних має широке застосування [21]. Динамічне розповсюдження шкідливого коду широко використовується в трьох основних сферах:

- виявлення та захист зловмисного коду;
- аналіз вразливостей програмного забезпечення та видобуток інформації;
- виявлення витоку конфіденційної інформації.

Подібно до традиційної технології аналізу пошкоджень (taint checking), різноманітне сучасне веб-програмне забезпечення з конфіденційними даними, наприклад міні-програми, розподілені програми, також може використовувати аналіз пошкоджень після деякого редизайну та трансформації для завершення аналізу потоку даних.

Веб-технологія орієнтована на взаємодію між кількома комп'ютерами на основі мережі. Навпаки, технологія аналізу пошкоджень більше підходить для тестів, що виконуються на одній обчислювальній системі, оскільки вона залежить від обсягу пам'яті. У цьому протиріччі традиційна ідея міграції технологій полягає в тому, щоб окремо поширювати пошкодження на клієнта та сервер, а потім аналізувати їх окремо. Однак у цьому окремому тесті потік викликів API

між тестованими суб'єктами не відстежується належним чином, і фокус зосереджений на потоці даних клієнта або самого сервера.

Веб-зв'язок на основі API базується на технології TCP/IP. Дані, запаковані відправником, маршрутизуються та пересилаються через мережу, дані знову зчитуються в пам'ять одержувача. Технологія аналізу пошкоджень покладається на позначку пошкодження на байтовому або бітовому рівні пам'яті, тому пошкодженя частина стане недійсною після передачі через мережу, яку неможливо безпосередньо відстежити. Крім того, базовий стек протоколів мережевої передачі в операційній системі зазвичай доповнюється кодом режиму ядра, а маркування пошкодження не на системному рівні не може позначати змінні стану ядра. Таким чином, платформа для аналізу пошкодження всієї системи стає необхідною умовою для відстеження потоку даних API.

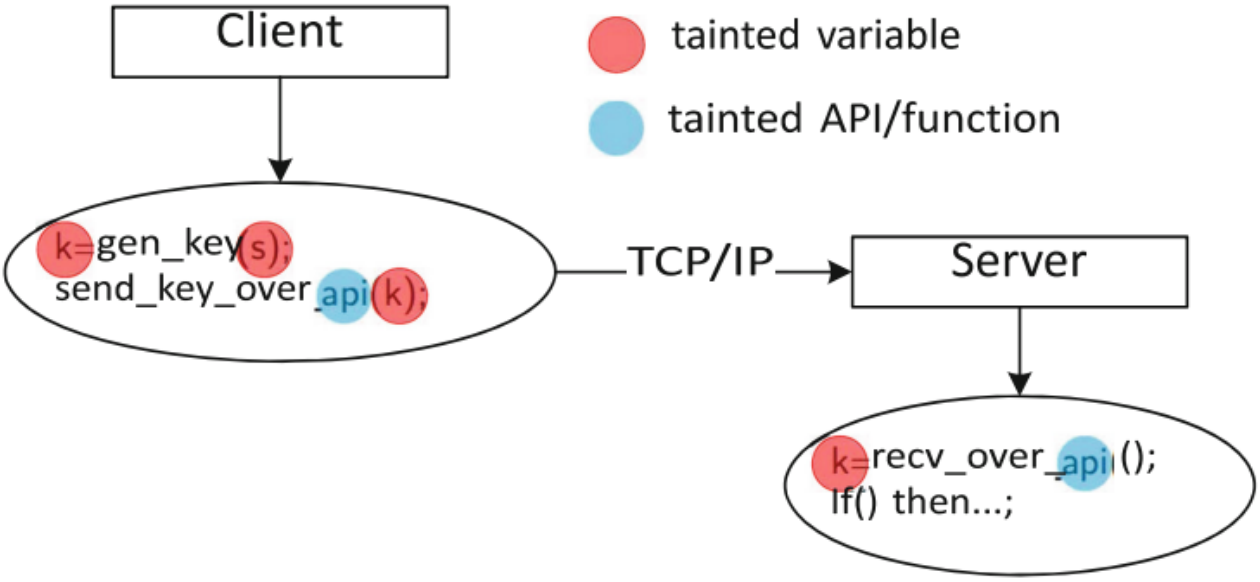

Рисунок 2.2 – Аналіз пошкодження перехресного мережевого зв'язку [22]

Рис 2.2 представляє те, що після того, як клієнтська змінна ключа k надсилається через функцію надсилання ключа через API, вона фактично передається в сокет, а потім передається через мережу TCP/IP. У комунікаційній мережі без RDMA адресу надісланої змінної k одержувачу неможливо сприйняти

клієнтом. Таким чином, окреме розповсюдження пошкодження на стороні клієнта закінчується на цьому, і конфіденційну адресу пам’яті виявити неможливо. У цьому процесі стек викликів функції надсилання ключа через API також є чутливим елементом, оскільки цей процес тісно пов’язаний із певним чутливим API. Щоб вирішити цю проблему, ми пропонуємо метод, який може позначати пошкоджені дані під час міжмережевого зв’язку та водночас відстежувати API та частину, яка ініціювала міжмережевий контакт. За допомогою цього методу клієнт і сервер можуть бути пошкоджені та відстежуватися синхронно в підтримуваній системі.

Основна ідея цього методу базується на системній емуляції клієнта та сервера, а комунікаційний рівень обох є накладним, так що фізичний рівень і рівень зв’язку традиційного мережевого зв’язку реалізуються програмним забезпеченням. Наприклад, коли клієнт надсилає пакет даних на сервер, емулятор системного рівня клієнта відстежуватиме змінні, які складають цей тип пакету даних у просторі пам’яті клієнта; після надсилання клієнтом рівень зв’язку перехоплює дані та безпосередньо копіює їх у буфер, відповідний серверу. При цьому емулятор на сервері повідомляється про те, що йому необхідно заздалегідь відстежити процес читання цього буфера (до завершення повідомлення процес читання не дозволяється виконувати); потім емулятор сервера відстежуватиме процес читання та відповідні змінні та позначатиме ці змінні як зіпсовані дані. У результаті реалізується поширення пошкодження зв’язку в наборі API.

У традиційній формальній перевірці мережевих протоколів дослідження безпеки повністю залежать від вхідних даних, які описують увесь протокол. Отже, коли вхідні дані ненадійні, результат формальної перевірки також непередбачуваний. Принцип є надзвичайно простий. Зрештою, будь-яка перевірка моделі та симуляція системи можуть довести лише незахищеність протоколу, але не безпеку самого протоколу. Те саме міркування цілком застосовне до сфери безпеки API. Стаття [23] пропонує новий вектор атаки: кілька API, інтегрованих в одну програму, спільно використовуватимуть той самий віртуальний адресний простір, тому атаки відбуваються між собою. І атака в основному зосереджена на

крадіжці даних користувача, що, безсумнівно, становить велику загрозу для поточних конфіденційних даних. Однак основним методом цієї роботи є зворотне проектування та статичний аналіз файлу Android APK.

Як показано на рис. 2.3, тестований додаток для Android створено авторами статті [23], щоб перевірити, чи API буде атаковано. Подібно до WeChat, відкритої платформи, яка надає логіни, заявники повинні бути кваліфікованими для надання відкритих послуг. Тому вважається, що програми з умовами входу в WeChat не викрадають дані користувача WeChat.

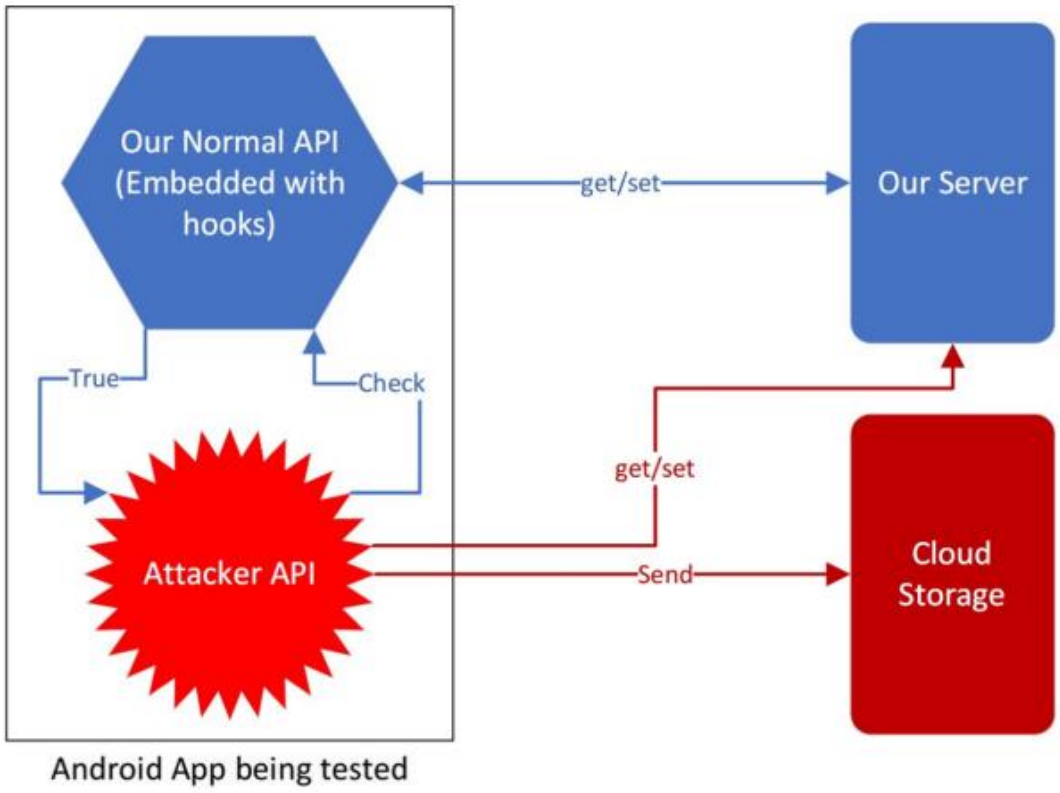

Рисунок 2.3 – Перехресно бібліотечні атаки на API

У міжбібліотечному виклику API абонент і абонент знаходяться в одному просторі пам’яті, тому бібліотека може природно викликати функції в іншій бібліотеці через певні API. У цьому випадку безпека API стає особливо критичною. Шкідлива бібліотека може зателефонувати до досліджуваного API, а потім обґрунтовано отримати з нашого сервера всі види даних за межами безпеки. Авторами було встановлено певний тип маркера як пошкоджені дані. Коли запит

надіслано на стороні додатка, абонент, пов’язаний із маркером, з’явиться в списку відстеження. Створивши розумний список відстеження, було легко знайти незаконні випадки, які з’яилися в списку позначок забруднення. Оскільки досліджувана система використовує рівень мережевої передачі, коли маркер надсилається одержувачу через TCP, він може безпосередньо знайти інформацію про абонента на сервері, таким чином перехоплюючи атаку.

2.5 Система аудиту безпеки API на основі трафіку

На даний момент безпечно та ефективно керувати будь якою API є нелегким процемос. Швидке та просте створення та керування API стало важливим питанням у розробці веб-програм [22]. Традиційні шлюзи безпеки API занадто дорогі для розгортання та обслуговування. Вони не мають ефективних стратегій побудови захисту та не можуть ефективно реагувати на загрози нових автоматизованих інструментів. Розробка методів автоматизованого аудиту API для підвищення безпеки все ще є складною. Враховуючи вище зазначені проблеми в цьому пункті кваліфікаційної роботи пропонується система аудиту безпеки API на основі Інтернет-трафіку, яка розроблена на основі збору розвідувальних даних, щоб надати дослідницькі ідеї щодо захисту безпеки масивних активів API.

API існує у вигляді цифрових даних. Використання API як форми керування цифровими активами може ефективніше покращити безпеку API. З точки зору активів API, аудит безпеки API розділений на три частини, як показано на рис. 2.4.

Модуль виявлення активів API використовує різні методи, такі як аналіз трафіку, дані стикування та імпорт даних для визначення режимів обробки даних API, таких як RESTful і GraphQL. Цей модуль забезпечить повне виявлення активів API, особливо для виявлення невідомих API. Більше того, надається точна ідентифікація, щоб запобігти доступності функцій через незрозумілі API.

Портрет активів API використовує аналіз даних для точного відображення функцій і дозволів API, таких як вхід користувача, реєстрація, запит даних і

дозволи адміністратора. Відповідно до списку профілів API ми можемо швидко перевірити статус кожного API, як-от використання, джерело доступу, виняток.

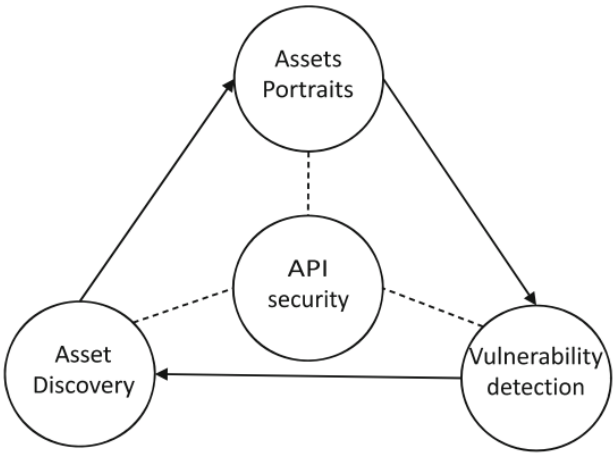

Рисунок 2.4 – Елементи аудиту кібербезпеки

Модуль виявлення безпеки та захисту API використовує активні та пасивні методи для перевірки безпеки API програми та бізнес-вимірів. Під час цієї частини буде виявлено віддалене виконання команд/коду, витік даних, несанкціонований доступ, неавторизований доступ, логічні дефекти тощо. За результатами аудиту необхідно динамічно реагувати та впроваджувати захисні заходи для підвищення складності атак.

При проектуванні системи необхідно сконцентрувати увагу на мережі, даних і бізнес-API. З боку мережі підключення внутрішніх і зовнішніх терміналів додатків через API розширить мережеві атаки, підвищить шляхи передачі ризику та зробить систему більш вразливою для зловмисників. Усе це може призвести до вторгнення на сервер і переривання безперервності роботи. Що стосується даних, якщо відкритий API має недоліки в конструкції або неправильні налаштування дозволів, зловмисники можуть незаконно отримати дані користувача. З точки зору бізнесу, у міру збільшення відкритості API сервісні інтерфейси можуть використовуватися поза рамками, збільшуючи ймовірність подій ризику відповідності бізнесу.

Починаючи з характеристик обробки даних, беручи до уваги ефективне читання та запис, перетворення даних, обробку транзакцій і стратегії кешування, ми можемо розділити систему аудиту на чотири рівні знизу вгору (рис. 2.5): рівень збору, рівень попередньої обробки, рівень виявлення та рівень. рівень інтерфейсів користувача.

Рівень збору в основному використовується для збору даних чотирма способами. Збір трафіку вимагає розгортання вузлів збору та імпорту PCAP, що є основним способом. Імпортування інструментів здійснюється шляхом збору даних безпосередньо в контакті з програмним забезпеченням, таким як інструменти тестування та проксі-інструменти. API також передає інформацію через апаратні пристрої та збирає дані з брандмауерів веб-додатків і шлюзів API за допомогою стикування пристроїв.

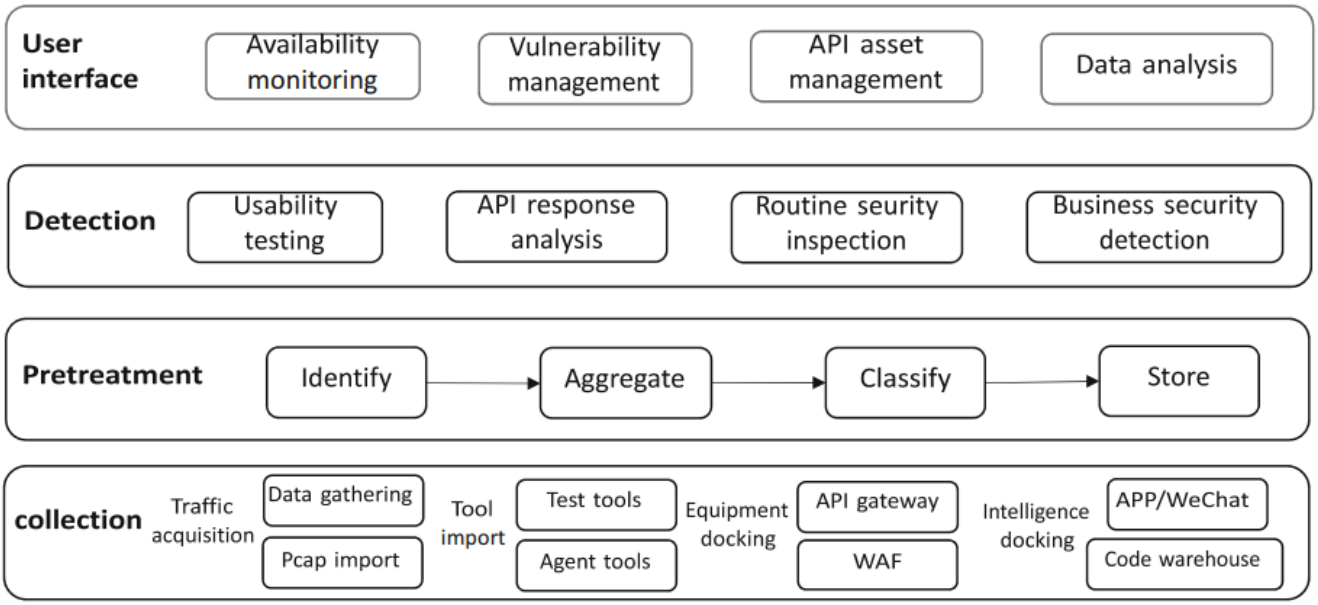

Рисунок 2.5 – Структура аудиту безпеки

Інтелектуальні дані на різних платформах також є важливим компонентом активів API. Збір ресурсів API на основі аналізу в основному здійснюється з таких аспектів: співпраця з постачальниками засобів безпеки APP для отримання інформації про активи API в APP; отримати інформацію про активи API в аплетах; отримати інформацію про активи API, витік у відомі сховища відкритого коду; З

метою отримання інформації про активи API необхідно використовувати інтернет-термінали, браузери, хмарні платформами, Інтернет речей, Інтернетом транспортних засобів та інші компанії. Зазначені вище чотири методи можуть більш повно створити інформаційну систему активів API.

Ядро системи аудиту безпеки API представленої на рис. 2.5 є засноване на потоках. По-перше, необхідно зібрати трафік відповідно до основного драйвера. Потім повторно створити бібліотеку ресурсів API, використовуючи URL-адресу, шаблони запитів тощо; початковий трафік можна безпосередньо сконструювати для формування бібліотеки трафіку, яку можна масово зберігати, швидко шукати та візуалізувати. Нарешті, необхідно виконати попередню обробку даних у бібліотеці активів API та бібліотеці трафіку.

Наступним кроком є отримання активів з бібліотеки ресурсів API для активного виявлення, та аналіз доступності результатів виявлення. Для доступних активів необхідно провести функціональний аналіз і аналіз дозволів, а також створення бібліотеки вразливостей API для знайдених проблем. Після цього необхідно проаналізувати вміст трафіку в базі даних трафіку, вилучити конфіденційну інформацію та записати проблеми в базу даних уразливостей API. За допомогою серії перевірок API навколо трафіку можна швидко й ефективно вирішувати проблеми безпеки API.

Відповідно до топ-10 безпеки API OWASP за 2020 рік [17], можна виявити, що проблеми безпеки API більшою мірою зосереджені в сфері безпеки бізнесу, наприклад, надмірні злочини, витік даних і недійсна автентифікація особи. Безпека звичайних програм в основному залежить від правил виявлення для аналізу запиту та відповіді, а потім оцінки ризику. Однак бізнес-безпека API повинна динамічно порівнювати й аналізувати кілька послідовних даних бізнес-запитів, щоб виявити приховані загрози бізнес-безпеці, такі як неавторизований доступ і збій автентифікації користувача. Щоб підвищити ефективність аудиту трафіку, можна використовувати методи глибокого навчання для постійного виявлення таких проблем.

У випадку достатності вибіркових даних API, ми можемо постійно накопичувати позитивні та негативні висновки під час аудиту безпеки API та постійно навчати та змінювати модель аудиту безпеки за допомогою кластерного аналізу, контрольованого навчання та інших методів. У такому випадку можемо використовувати модель аудиту безпеки, щоб виявити більше невідомих загроз безпеці API.

3 AMAZON SELLING PARTNER API АУДИТ

3.1 Ознайомлення із досліджуваним SAAS-рішенням

При дослідженні тенденцій електронної комерції було визначено надзвичайну популярність щодо продаж на кількох торгівельних площадках. Тенденція щодо багатоканальності представлення товарів призводить до збільшення операційних витрат на контролювання залишків продуктів та послуг, їх цін та публікації нових товарів на існуючих маркетплейсах.

Для ефективної роботи магазину електронної комерції потрібна велика кількість програм, і відслідковувати всі їхні процеси та деталі є достатньо втомливим. Наприклад, один веб-додаток використовується для керування запасами, інший для керування замовленнями, а ще інший для публікації продуктів. І всі ці програми мають різні інформаційні панелі, які ви повинні розуміти, щоб ефективно ними користуватися.

Це вимагає додаткових витрат і призводить до того, що власники бізнесу втрачають зосередженість на своїй основній бізнес-практиці та цілях. Тож ідеальним вирішенням проблеми є інтеграція окремих програм в одне централізоване середовище та забезпечення їх функціонування з однієї точки. Цей підхід не тільки значно спростить бізнес-практики та процеси, але й підвищить продуктивність.

Саме для вирішення вище представленої моделі використовується веб-додаток, який і буде досліджений в подальшому і для якого буде здійснена інтеграція із Amazon Selling Partner API. Враховуючи NDA не має можливості вказати назву існуючого на ринку додатку, тому в кваліфікаційній роботі використаємо термін “досліджуване SAAS рішення” для позначення програмного продукту.

“Досліджуване SAAS рішення” – це унікальний продукт, який допомагає підприємствам і компаніям ефективно вести свою електронну комерцію та роздрібну торгівлю на різних з однієї інформаційної панелі. Рішення допомогло

тисячам продавців електронної комерції покращити їхні щоденні процеси та інтегруватись з найбільшими маркетплейсами, перевізниками і торговими платформами, такими як Shopify, Etsy, eBay, Walmart і Amazon (рис. 3.1).

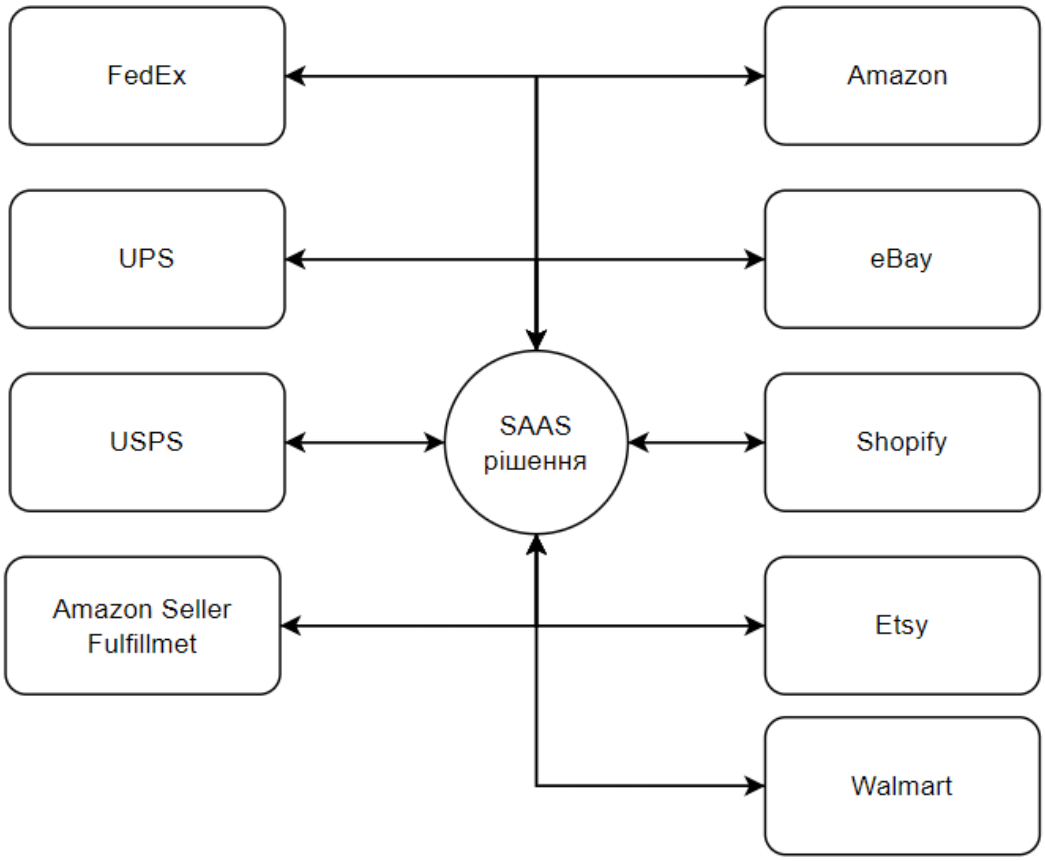

Рисунок 3.1 – Двостороння інтеграція “SAAS рішення” із онлайн маркетплейсами та перевізниками

Для інтеграція, що зображена на рис. 3.1 використовуються наявні API інтерфейси. Однією із проблем стабільності досліджуваного рішення є періодична зміна документації API того чи іншого марктеплейсу, його політик та процедур. Для прикладу досліджуване рішення успішно використовувало Amazon Marketplace Web Service, проте починаючи із четвертого кварталу 2022 року необхідно повністю перейти на нову версію Amazon API. Дане рішення не може бути присутнім на ринку без інтеграції із найбільшим світовим маркетплейсом, відповідно розробникам необхідно виконати міграцію, а експертам з кібербезпеки

пройти аудит щодо інтеграції Amazon Selling Partner API з метою отримання доступу щодо використання всіх наявних методів та даних.

3.2 Архітектура досліджуваного SAAS-рішення

Система “SAAS рішення” складається з наступних компонентів:

- веб-сайт: містить інформацію про проект та блог зі статтями. Побудований на основі Wordpress та MariaDB.
- back-end: містить бізнес-логіку проекту та надає REST API для всіх front-end додатків. Написаний з використанням .NET Core та включає три основні компоненти:
  - Core – виконує запити користувачів, які підходять через REST API;
  - Worker – виконує асинхронні задачі синхронізації даних між досліджуваним рішенняи та сторонніми маркетплейсами (Amazon, eBay, Walmart, Shopify, EasyPost);
  - ViewRenderer – виконує асинхронну підготовку та відправку емейлів користувачам.

Усі компоненти забезпечуються як окремі додатки. Worker та ViewRenderer підтримують завдання від Core через сайт у Redis.

- front-end: Призначені для кінцевого користувача SAAS рішення. Для написання коду використовувався Angular. До даного компоненту входять наступні додатки:
  - Client App – додаток клієнта, призначений для роботи з системою, управління товарами, листингами та ін.
  - Partner App – додаток для партнерської програми, де партнери можуть бачити аналітику за наведеними ними клієнтами
  - Support (Console) App - містить інформацію про клієнтів, їхні запити та підписки.

Схематично це представлено на рис. 3.2.

Для реалізації представленої архітектури використовуємо такі сервіси: PostgreSQL для зберігання даних, для продакшна ми використовуємо AWS RDS. Для управління чергою та кешування використовується Redis.

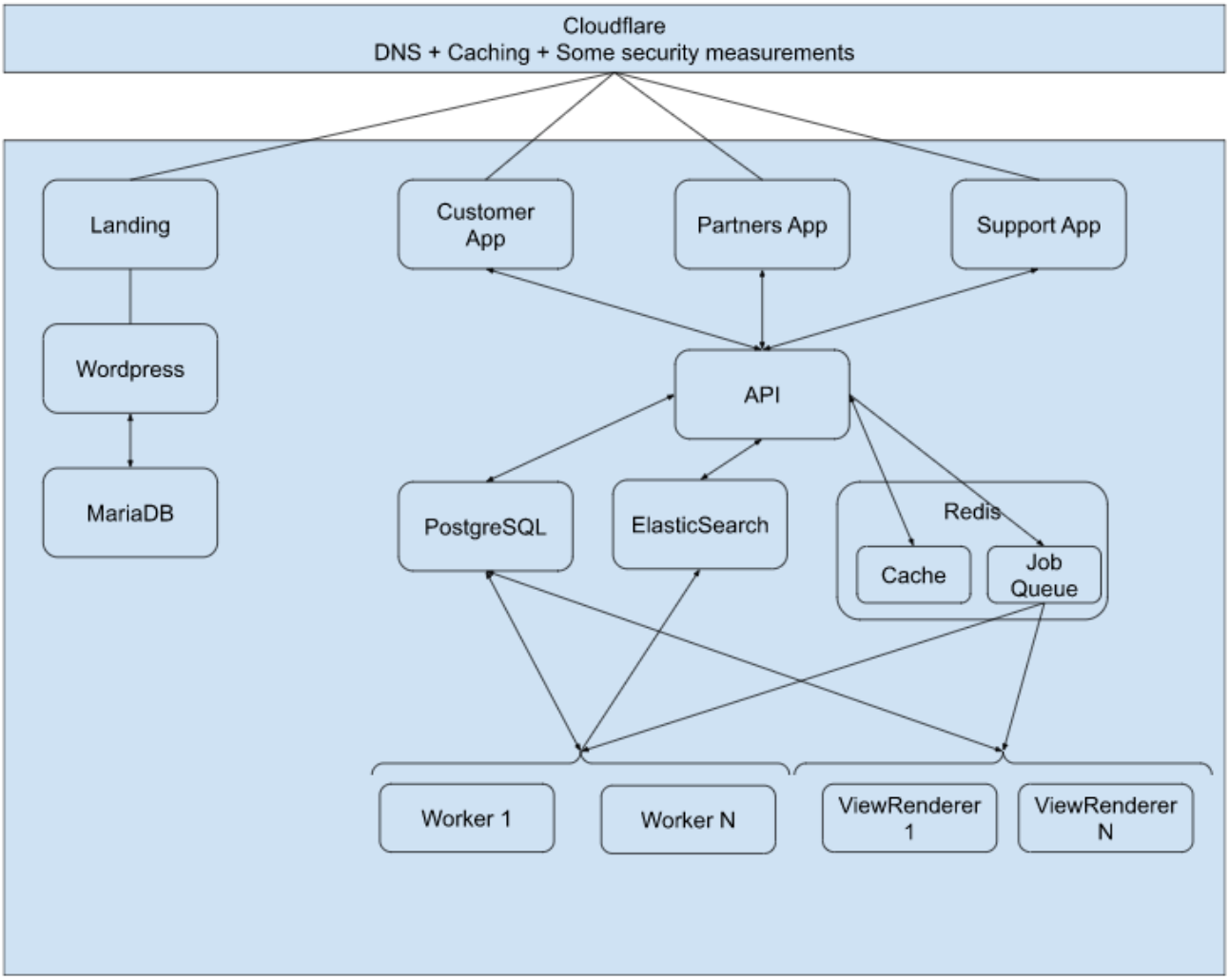

Рисунок 3.2 – Архітектура досліджуваного SAAS рішення

Код проектів зберігається в GitLab і ми використовуємо GitLab CI з власними runner для збірки і деплою проектів.

Кожен проект містить .gitlab-ci.yml в корені проекту, що описує pipeline виконання CI.

Стандартний pipeline виглядає так:

- збірка;

- тестування;
- публікація docker image в Gitlab Containers;
- деплой.

На даний момент для реалізації досліджуваного рішення використовуються 3 сервери:

- EC2 instance / Ubuntu – призначений для продакшна (AWS);
- VPS від Contabo / CentOS - призначений для dev/staging деплою;
- VPS від Contabo / CentOS - містить стек ELK для збору логів.

На усіх серверах встановлено Docker (Swarm mode) для виконання контейнерів та Gitlab Runner для деплою з Gitlab CI.

3.3 Amazon Selling Partner API

Amazon Selling Partner API, API партнера з продажу (SP-API) – це API на основі використання REST технології. Даний інтерфейс допомагає партнерам з продажу Amazon програмно отримувати доступ до своїх даних про замовлення, відправлення, платежі та багато іншого. Програми, що використовують SP-API, можуть підвищити ефективність продажів, зменшити потреби в робочій силі та покращити час відповіді клієнтам, допомагаючи партнерам з продажу розвивати свій бізнес.

Для того щоб отримати доступ до можливості застосування Amazon SP-API потрібно лише один раз зареєструватися як розробник у регіоні та на ринку за вашим вибором, щоб мати змогу створити програму SP-API, яку може авторизувати партнер із продажу з будь-якого регіону чи ринку. Також необхідно лише один набір облікових даних розробника (ваш ідентифікатор ключа доступу до AWS і секретний ключ доступу до AWS), щоб здійснювати дзвінки до будь-якої кінцевої точки SP-API, за умови, що кінцева точка належить до того самого регіону, що й партнер з продажу, який авторизував вашу програму.

Основними перевагами та можливостями щодо використання Amazon SP-API є:

- налаштування робочого процесу авторизації OAuth, який партнери з продажу ініціюють зі сторінки з інформацією про партнера з продажу Appstore або з власного веб-сайту.
- створення SDK, який допоможе з обміном токеном LWA та автентифікацією.
- перевірте веб-додатку з використанням викликів до середовища ізольованого програмного середовища (SandBox).

Попередньою версією Amazon API був Amazon Marketplace Web Service, проте починаючи із третього кварталу 2022 року дана версія є забороненою. Усі додатки та продукти, що використовували зазначену версію повинні перейти на оновлення. Це спричиняє певні ризики для існуючих програмних додатків, адже перехід на нову версію потребує затрат часу та людських ресурсів, при цьому збільшується бюджетні витрати.

Незважаючи на те, що SP-API з продажу та веб-служба Amazon Marketplace (Amazon MWS) є веб-сервісами, які забезпечують програмний доступ до даних клієнтів, існують значні відмінності. Ось деякі ключові відмінності між SP-API та Amazon MWS:

- за допомогою SP-API можна розробляти програми як для продавців, так і для постачальників;
- SP-API обробляє дані як сумісні з REST ресурсами, доступ до яких можна отримати та змінити за допомогою стандартних методів HTTP. Навпаки, Amazon MWS відкриває дані за допомогою операцій, які є специфічними для Amazon MWS;
- авторизація SP-API використовує LWA, реалізацію Amazon OAuth 2.0. Ця модель усуває потребу в ручному обміні маркерами авторизації, як того вимагає Amazon MWS.
- у Amazon MWS партнери з продажу авторизують розробників. За допомогою SP-API партнери з продажу авторизують програми. Використовуючи

SP-API, розробники можуть створювати кілька програм, які потребують різних рівнів доступу до даних партнера з продажу.

- SP-API забезпечує точніший контроль доступу до даних, ніж Amazon MWS. Розробники можуть запитувати доступ лише до тих даних, які їм потрібні, а партнери з продажу можуть надавати дозволи на рівні розділу API, операції або ресурсу даних.
- SP-API дозволяє безпосередньо отримувати власні облікові дані для автентифікації та керувати ними за допомогою AWS Identity and Access Management (IAM). З Amazon MWS розробник отримував облікові дані для автентифікації, надані Amazon за допомогою спеціального процесу реєстрації, і ви отримуєте нові облікові дані, відкривши контакт зі службою підтримки Amazon MWS.
- SP-API використовує AWS Signature Version 4 для автентифікації. Amazon MWS використовує підпис версії 2.

Існує три типи додатків (програм), які можна розробити за допомогою SP-API:

- публічні програми є загальнодоступними в Appstore партнера з продажу та авторизовані продавцем або постачальником;
- додатки приватних продавців доступні лише для вашої організації (або клієнта) і є самоавторизованим;
- програми приватних постачальників доступні лише для вашої організації та авторизовані самостійно.

Кожен тип програми має різні робочі процеси та вимоги.

3.4 Event-driven архітектура SP-API

Архітектурний шаблон, керований подіями (Event-driven) це асинхронна модель проектування архітектури. Ця архітектура використовує події для з’єднання роз’єднаних незалежних компонентів системи. Ці компоненти зазвичай

є мікросервісами, які виконують певні завдання. Замість монолітної програми, що містить всю логіку, архітектура, керована подіями, використовує невеликі компоненти, які генерують події та реагують на них. Ці події представляють зміни стану або інші типи оновлень.

Ключовими елементами керованої подіями архітектури є виробники та споживачі. Зазвичай програма складається з кількох виробників і споживачів. Виробник генерує подію, яка обробляється одним або кількома споживачами для виконання різних завдань. Разом ці завдання виконують бізнес-випадок використання.

Побудова керованої подіями архітектури покращує продуктивність, вартість, надійність, масштабованість і життєвий цикл розробки програми. Приклади цих переваг:

- програми, керовані подіями, працюють краще, реагуючи на події в реальному часі, на відміну від схем, де дані збираються за розкладом, що створює затримки;
- реагування на події зменшує обсяг непотрібної роботи, як-от опитування служби щодо оновлень, що допомагає скоротити витрати за рахунок економії ресурсів і використання;
- шаблон виклику керованої подіями архітектури зменшує навантаження на сторонні служби, усуваючи вузьке місце обмеження швидкості та зменшуючи помилки регулювання;
- відокремлені компоненти можуть масштабуватися та виходити з ладу незалежно, дозволяючи їм адаптуватися до попиту на основі індивідуальних потреб і зменшувати радіус вибуху відмов;
- життєвий цикл розробки скорочується, тому що архітектура простіша і тому її легше адаптувати до нових випадків використання.

SP-API надає Notification API, який дозволяє користувачам створювати керовану подіями архітектуру. За допомогою Notification API є можливість підписатися на різні типи подій і отримувати сповіщення про відповідні зміни у бізнесі Amazon.

SP-API пропонує два робочі процеси для отримання сповіщень. Один із них використовує Amazon Simple Queue Service (Amazon SQS), а інший використовує Amazon EventBridge як маршрутизатори для подій. Залежно від типу сповіщень, на які необхідно підписатися для розробки веб-додатку, необхідно релазувати інтеграцію з одним із цих робочих процесів.

Amazon Simple Queue Service (Amazon SQS) – це повністю керована служба черги повідомлень, яка дозволяє отримувати повідомлення з різних джерел і їх відповідну обробку. Використання Amazon SQS забезпечує масштабоване, високодоступне та безпечне рішення для отримання та обробки подій, які стосуються бізнесу клієнта. Amazon SQS пропонує кілька альтернатив для обробки вхідних повідомлень, наприклад інтеграцію з функціями AWS Lambda або використання API Amazon SQS, щоб забезпечити гнучкість, яку потребує ваша програма (рис. 3.4.).

Типова архітектура робочого процесу сповіщень Amazon SQS у SP-API складається з черги повідомлень і споживача для цих подій. Черга повідомлень розміщується в обліковому записі Amazon Web Services (AWS) розробника та отримує сповіщення про події, на які підписаний партнер з продажу. Обробка повідомлень відбувається асинхронно та базується на бізнес-випадках, які підтримує програма.

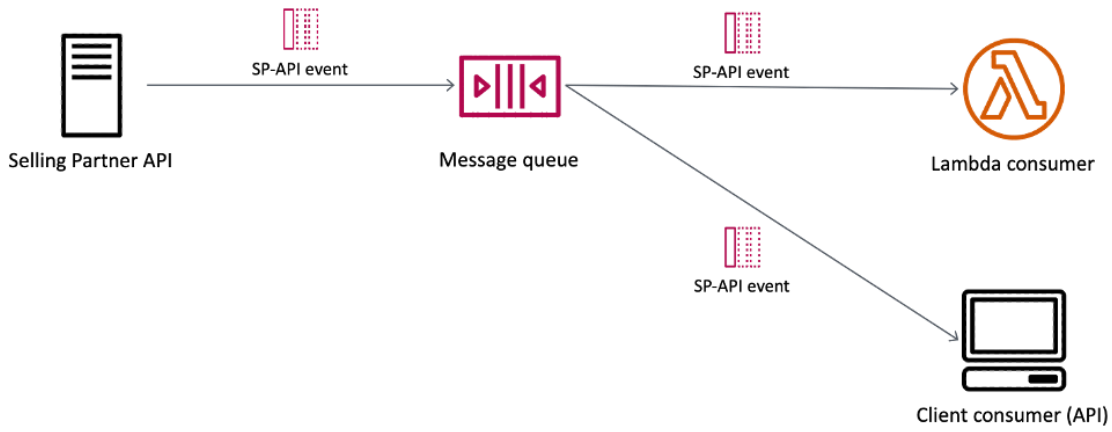

Рисунок 3.3 – Еталонна архітектура робочого процесу Amazon SQS

Amazon EventBridge – це безсерверна шина подій, яка дає змогу отримувати події від різноманітних сервісів і клієнтських програм AWS, а також їх відповідний розподіл до різних цілей для обробки (рис. 3.5). EventBridge – це керована відмовостійка служба, яка масштабується на основі вхідного трафіку. Ви можете використовувати EventBridge для визначення спеціальних правил для фільтрації та трансформації подій перед їх надсиланням до вибраних цілей, що спрощує інтеграцію між програмними компонентами. EventBridge підтримує понад 40 джерел подій типу SAAS для надходження даних і кількох місць призначення, включаючи AWS Lambda, API Gateway і спеціальні кінцеві точки HTTP.

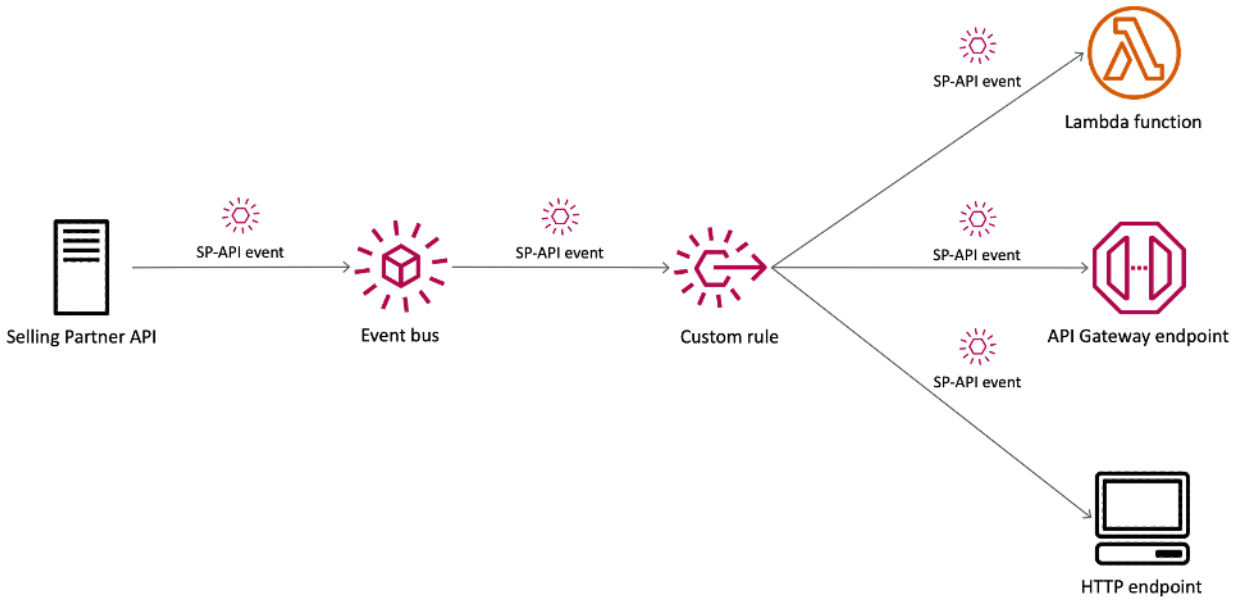

Рисунок 3.4 – Еталонна архітектура робочого процесу EventBridge

Типова архітектура для робочого процесу сповіщень EventBridge у SP-API складається з шини подій, розміщеної в обліковому записі AWS розробника, яка отримує сповіщення про події, на які підписаний партнер з продажу, одне або кілька користувацьких правил і їхні відповідні цілі. Обробка повідомлень відбувається асинхронно та базується на бізнес-випадках, які підтримує програма.

3.5 Data Protection Policy та її важливість для проходження аудиту

Для успішної валідації веб-додатку та опублікування його на торгівельній площадці необхідно пройти аудит щодо використання SP-API. Аудит проходить згідно встановлених правил у Data Protection Policy (DPP). В загальному дана політика включає наступні пункти:

- Захист мережі. Розробники повинні запровадити засоби керування захистом мережі, включаючи мережеві брандмауери та списки контролю доступу до мережі, щоб заборонити доступ до несанкціонованих IP-адрес, запровадити сегментацію мережі, програмне забезпечення для захисту від вірусів і шкідливих програм.
- Керування доступом. Розробники повинні запровадити офіційний процес реєстрації доступу користувачів, щоб призначити права доступу для всіх типів користувачів і послуг, забезпечивши присвоєння унікального ідентифікатора кожній особі, яка має комп'ютерний доступ до інформації, переглядати список людей і служб, які мають доступ до інформації, щонайменше раз на квартал, переконатися, що доступ для звільнених співробітників вимкнено та/або видалено протягом 24 годин.
- Принцип найменших привілеїв. Розробники повинні запровадити детальні механізми контролю доступу, щоб дозволити надавати права будь-якій стороні, яка використовує Програму, і авторизованим операторам Програми відповідно до принципу найменших привілеїв. Доступ до інформації має бути наданий на основі «необхідності знати».
- Керування обліковими даними. Розробники повинні встановити мінімальні вимоги до пароля для персоналу та систем, які мають доступ до інформації, переконатися, що багатофакторна автентифікація (MFA) потрібна для всіх облікових записів користувачів. Розробник повинен переконатися, що ключі API, надані Amazon, зашифровані та доступ до них мають лише необхідні працівники.

- Шифрування під час передавання. Розробники повинні шифрувати всю інформацію, що передається, за допомогою безпечних протоколів, таких як TLS 1.2+, SFTP і SSH-2.
- Управління ризиками та план реагування на інциденти. Розробники повинні мати процес оцінки та управління ризиками, який щорічно переглядається вищим керівництвом.

Необхідно зауважити, що вище представлений список є лише базовим. Для реалізації інтеграції SAAS рішення та отримання персональної індивідуальної інформації потрібно відповідати додатковим вимогам:

- Зберігання даних. Розробники зберігатимуть ідентифікаційну інформацію не довше ніж 30 днів після доставки замовлення та лише з метою та до тих пір, поки це необхідно для виконання замовлень, розрахунку та сплати податків, (iii) складання податкових накладних та інших законних заходів. необхідні документи та відповідати юридичним вимогам, у тому числі податковим або нормативним вимогам.
- Управління даними. Розробники повинні створити, задокументувати та дотримуватися політики конфіденційності та обробки даних і класифікації для своїх програм або послуг, яка регулює належну поведінку та технічний контроль, що застосовуються під час керування та захисту інформаційних активів. Необхідно вести записи про дії з обробки даних, як-от конкретні поля даних, а також те, як вони збираються, обробляються, зберігаються, використовуються, передаються та утилізуються для всіх ідентифікаційних даних, щоб встановити підзвітність і відповідність нормам.
- Управління активами. Розробники повинні підтримувати базову стандартну конфігурацію інформаційної системи та вести інвентаризацію програмного забезпечення та фізичних активів (наприклад, комп’ютерів, мобільних пристроїв) із доступом до ідентифікаційної інформації та оновлювати щокварталу. Розробник повинен запровадити засоби запобігання втраті даних (DLP) для відстеження та виявлення несанкціонованого переміщення даних.

- Шифрування в стані спокою. Розробники повинні шифрувати всю ідентифікаційну інформацію в стані спокою за допомогою принаймні AES-128 або RSA з розміром ключа 2048-біт або вище. Криптографічні матеріали (наприклад, ключі шифрування/дешифрування) і криптографічні можливості (наприклад, демони, що реалізують модулі віртуальної довіреної платформи та надають API шифрування/дешифрування), що використовуються для шифрування ідентифікаційної інформації в стані спокою, мають бути доступні лише для процесів і служб Розробника.
- Практики безпечного кодування. Розробники не повинні жорстко кодувати конфіденційні облікові дані у своєму коді, зокрема ключі шифрування, секретні ключі доступу або паролі. Конфіденційні облікові дані не можна розкривати в публічних сховищах коду. Розробники повинні підтримувати окремі тестові та робочі середовища.
- Журнали та моніторинг. Розробники повинні збирати журнали для виявлення подій, пов’язаних із безпекою, у своїх програмах і системах, включаючи успішне або невдале подію, дату й час, спроби доступу, зміни даних і системні помилки. Розробники повинні запровадити цей механізм реєстрації на всіх каналах (наприклад, API служби, API рівня зберігання, адміністративних панелях), що надають доступ до інформації. Розробники повинні переглядати журнали в реальному часі (наприклад, інструмент SIEM) або кожні два тижні
- Управління вразливістю. Розробники повинні створити та підтримувати план та/або модуль Runbook для виявлення та усунення вразливостей. Розробники повинні захищати фізичне обладнання, що містить ідентифікаційну інформацію, від технічної шкоди.

3.6 Selling Partner API Guard

Selling Partner API Guard – це безсерверна програма, яка сканує облікові записи AWS розробників API Selling Partner, щоб оцінити відповідність їх безпеки політиці конфіденційності Amazon. Для виявлення вразливостей в безпеці та їх

виправлення API Guard сформує звіт протягом 24 години за допомогою спеціальних правил сканування, створених на основі існуючих служб AWS, таких як Security Hub і Macie. Як шаблон CloudFormation, Guard легко підібрати, швидко розгорнути та безпечно використовувати.

Guard автоматизує збір доказів, щоб зменшити ручні зусилля для перевірки. Попередньо створена структура відображає ресурси AWS для контролю вимог політики та надає такі посилання в кінцевому звіті з чіткими рекомендаціями, щоб заощадити час на пошук рішень. При цьому Guard надійно зберігає результати в S3 і надає кінцевим користувачам дозволи лише на читання.

До складу Amazon API Guard входять наступні AWS сервіси:

- GuardDuty відстежує вхідний і вихідний мережевий потік VPC, подій S3, CloudTrail і DNS, а також аналізує ці журнали та перевіряє будь-які аномалії в шаблоні. GuardDuty також перевіряє наявність зловмисного програмного забезпечення в примірниках EC2, ECS і Kubernetes.
- Security Hub перевіряє стандарти безпеки, використовуючи стандартні та користувацькі правила конфігурації. Він також діє як робоче місце для агрегування та відображення результатів усіх служб із стеку Amazon API Guard.
- Inspector відстежує екземпляри EC2 і репозиторії ECR на наявність вразливостей і ненавмисного доступу до мережі на різних портах.
- Macie виявляє особисту інформацію в незашифрованому сховищі RDS, S3 і DynamoDB.
- Access Analyzer оцінює дозволи, які надаються зовнішнім користувачам, щоб переконатися, що засоби керування доступом налаштовано належним чином.
- Config має низку правил, керованих AWS, які виявляють загрози безпеці у конфігурації, наприклад перевіряє, чи є якісь із ваших сегментів S3 публічними. Guard використовує деякі правила Config для перевірки такої конфігурації.

Для інсталяції Amazon API Guard необхідно виконати кілька кроків: інсталяція, включення відповідних сервісів, отримання та опрацювання звіту, деінсталяція стеку. . На рис. 3.5, 3.6, 3.7 та 3.8 зображено відповідні кроки. Необхідно зауважити, що даний сервіс Amazon API Guard може бути використаний лише для тих веб-додатків, які є реалізованими на платформі Amazon AWS. З одного боку це є достатньо зручно, проте якщо ваш додаток реалізований на іншому хмарному середовищі необхідно виконувати ручний аудит або комплекс програм та рішень. Також розробники Amazon SP-API є зацікавленими в тому, щоб розробники чи представники бізнесу переходили на хмарні рішення від Amazon, таким чином концентруючи усі правила та процедури на своїх політиках.

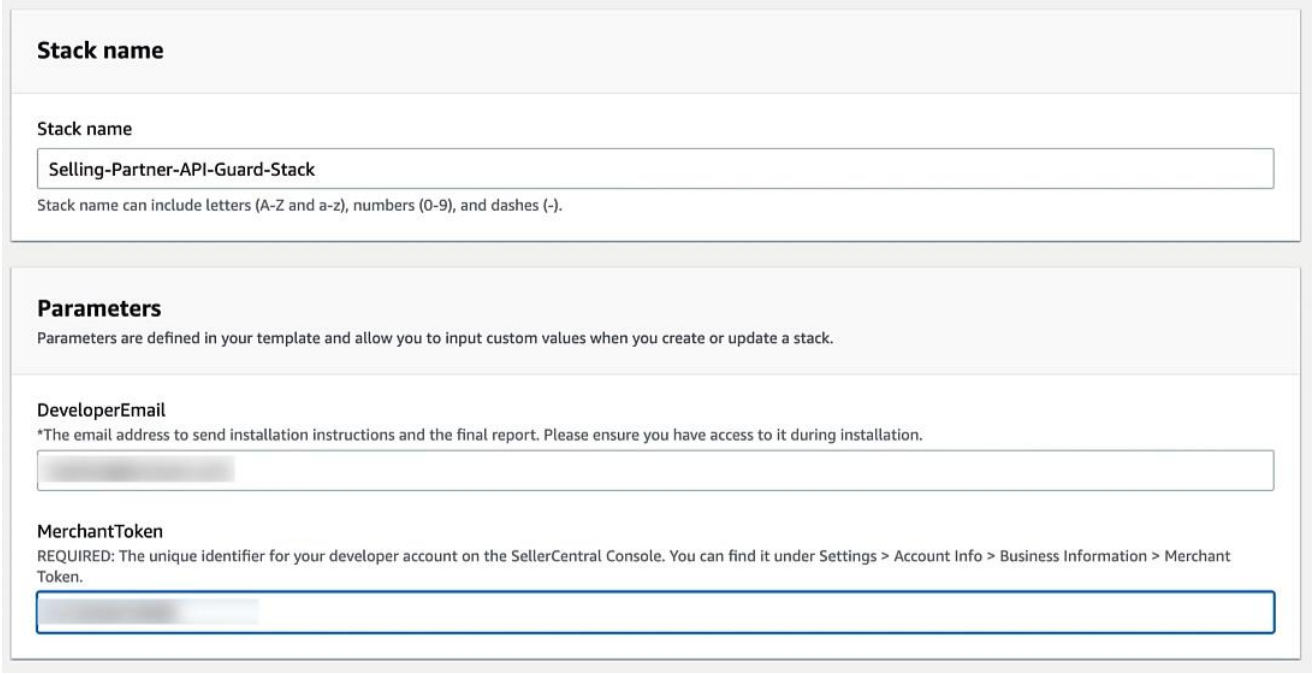

Рисунок 3.5 – Інсталяції стеку на Amazon AWS

```
sh-4.2$ cd GuardCli/
sh-4.2$ sudo ./guardcli enable_services
== guardcli 2022-09-30T19:55:04Z Exporting config ...

Command: enable_services
Usage:
  enable_services -m -g -i
== enable_services 2022-09-30T19:55:04Z Scan Rules enabling BEGIN
AWS version Check...
Seems AWS Version Ok!


AWS Credentials Check...
{
    "UserId": "AROAZPHJB5AQIMFYQCN27:i-051acef6352f267c5",
    "Account": "651176306720",
    "Arn": "arn:aws:sts::651176306720:assumed-role/Selling-Partner-API-Guard-Ec2GuardCliRole0C7F0830-OOHW26OC0AIB/i-051acef6352f267c5"
}
Seems AWS Credentials Ok!


  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   473  100   473    0     0   132k      0 --:--:-- --:--:-- --:--:--  153k


Would you like to scan your S3 buckets for unencrypted PII data (Macie)? (y or n) y
Enabling Macie in Respective AWS Account Configured
Would you like to scan your whole AWS account for Data Protection Policy requirements and possible malicious activity (GuardDuty)? (y or n) y
Enabling GuardDuty in Respective AWS Account Configured
Would you like to scan the identity resources in your organization and accounts that are shared with an external entity (AccessAnalyzer)? (y or n) y
Enabling AccessAnalyzer in Respective Region of AWS Account Configured
Would you like to scan your EC2 instances for unintended network vulnerabilities (Inspector)? (y or n) y
Enabling Inspector...


Enabling Security Hub
Security Hub Not Enabled. Enabling SecurityHub
```

Рисунок 3.6 – Включення сервісів Amazon API Guard для сканування

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	awsAccou	region	control	controlDe	policyRef	status	serviceNa	severity	nonComp	recommendationText	recommendationUrl				
2	7.01E+10	us-east-1	1.2 Ensure multi-factor authentica	Multi-Fact	NotMappe	NonComp	Security H	MEDIUM	arn:aws:ia	For directions on how to fix this issue, consult the AWS Security Hub CIS doc	https://docs.aws.amazon.com/console/securityhu				
3	7.01E+10	us-east-1	3.3 Ensure a log metric filter and al	Real-time	NotMappe	NonComp	Security H	LOW	AWS::::Ac	For directions on how to fix this issue, consult the AWS Security Hub CIS doc	https://docs.aws.amazon.com/console/securityhu				
4	7.01E+10	us-east-1	Aggregate network exposure: On i	On instan	DPP - Net	NonComp	Inspector	INFORMATIONAL	arn:aws:e	You can edit the Security Group sg-7d22b634 to remove access from the internet					
5	7.01E+10	us-east-1	1.11 Ensure IAM password policy e	IAM passv	NotMappe	NonComp	Security H	LOW	AWS::::Ac	For directions on how to fix this issue, consult the AWS Security Hub CIS doc	https://docs.aws.amazon.com/console/securityhu				
6	7.01E+10	us-east-1	2.4 Ensure CloudTrail trails are inte	AWS Clou	NotMappe	NonComp	Security H	LOW	arn:aws:cl	For directions on how to fix this issue, consult the AWS Security Hub CIS doc	https://docs.aws.amazon.com/console/securityhu				
7	7.01E+10	us-east-1	CloudFormation.1 CloudFormation	This contr	DPP - Logg	NonComp	Security H	LOW	arn:aws:cl	For directions on how to fix this issue, consult the AWS Security Hub Founda	https://docs.aws.amazon.com/console/securityhu				
8	7.01E+10	us-east-1	On instance i-0f87e748ba312266d,	On this in	NotMappe	NonComp	Inspector	HIGH	arn:aws:e	You can edit the Security Group sg-0f93be09acd0ed4d8 to remove access from the internet on port 21					
9	7.01E+10	us-east-1	On instance i-0e54a04245aa3858e,	On this in	NotMappe	NonComp	Inspector	LOW	arn:aws:e	You can edit the Security Group sg-0f93be09acd0ed4d8 to remove access from the internet on port 443					
10	7.01E+10	us-east-1	3.2 Ensure a log metric filter and al	Real-time	NotMappe	NonComp	Security H	LOW	AWS::::Ac	For directions on how to fix this issue, consult the AWS Security Hub CIS doc	https://docs.aws.amazon.com/console/securityhu				
11	7.01E+10	us-east-1	Aggregate network exposure: On i	On instan	DPP - Net	NonComp	Inspector	INFORMATIONAL	arn:aws:e	You can edit the Security Group sg-0f93be09acd0ed4d8 to remove access from the internet					
12	7.01E+10	us-east-1	RDS.10 IAM authentication should	This contr	DPP - Net	NonComp	Security H	MEDIUM	arn:aws:rd	For directions on how to fix this issue, consult the AWS Security Hub Founda	https://docs.aws.amazon.com/console/securityhu				
13	7.01E+10	us-east-1	EC2.3 Attached EBS volumes shoul	This AWS	DPP - Encr	NonComp	Security H	MEDIUM	arn:aws:e	For directions on how to fix this issue, consult the AWS Security Hub Founda	https://docs.aws.amazon.com/console/securityhu				
14	7.01E+10	us-east-1	4.1 Ensure no security groups allow	Security g	NotMappe	NonComp	Security H	HIGH	arn:aws:e	For directions on how to fix this issue, consult the AWS Security Hub CIS doc	https://docs.aws.amazon.com/console/securityhu				
15	7.01E+10	us-east-1	3.11 Ensure a log metric filter and a	Real-time	NotMappe	NonComp	Security H	LOW	AWS::::Ac	For directions on how to fix this issue, consult the AWS Security Hub CIS doc	https://docs.aws.amazon.com/console/securityhu				
16	7.01E+10	us-east-1	3.5 Ensure a log metric filter and al	Real-time	NotMappe	NonComp	Security H	LOW	AWS::::Ac	For directions on how to fix this issue, consult the AWS Security Hub CIS doc	https://docs.aws.amazon.com/console/securityhu				
17	7.01E+10	us-east-1	1.1 Avoid the use of the root user	The root u	NotMappe	NonComp	Security H	LOW	AWS::::Ac	For directions on how to fix this issue, consult the AWS Security Hub CIS doc	https://docs.aws.amazon.com/console/securityhu				
18	7.01E+10	us-east-1	1.13 Ensure MFA is enabled for the	The root u	NotMappe	NonComp	Security H	CRITICAL	AWS::::Ac	For directions on how to fix this issue, consult the AWS Security Hub CIS doc	https://docs.aws.amazon.com/console/securityhu				
19	7.01E+10	us-east-1	1.10 Ensure IAM password policy p	IAM passv	NotMappe	NonComp	Security H	LOW	AWS::::Ac	For directions on how to fix this issue, consult the AWS Security Hub CIS doc	https://docs.aws.amazon.com/console/securityhu				
20	7.01E+10	us-east-1	RDS.11 RDS instances should have	This contr	DPP - Data	NonComp	Security H	MEDIUM	arn:aws:rd	For directions on how to fix this issue, consult the AWS Security Hub Founda	https://docs.aws.amazon.com/console/securityhu				
21	7.01E+10	us-east-1	EC2.18 Security groups should only	This contr	DPP - Net	NonComp	Security H	HIGH	arn:aws:e	For directions on how to fix this issue, consult the AWS Security Hub Founda	https://docs.aws.amazon.com/console/securityhu				
22	7.01E+10	us-east-1	S3.8 S3 Block Public Access setting	This contr	NotMappe	NonComp	Security H	HIGH	arn:aws:s3	For directions on how to fix this issue, consult the AWS Security Hub Founda	https://docs.aws.amazon.com/console/securityhu				
23	7.01E+10	us-east-1	IAM.5 MFA should be enabled for	This AWS	DPP - Net	NonComp	Security H	MEDIUM	arn:aws:ia	For directions on how to fix this issue, consult the AWS Security Hub Founda	https://docs.aws.amazon.com/console/securityhu				
24	7.01E+10	us-east-1	SNS.1 SNS topics should be encryp	This contr	DPP - Encr	NonComp	Security H	MEDIUM	arn:aws:sn	For directions on how to fix this issue, consult the AWS Security Hub Founda	https://docs.aws.amazon.com/console/securityhu				
25	7.01E+10	us-east-1	IAM.6 Hardware MFA should be en	This AWS	DPP - Leas	NonComp	Security H	CRITICAL	AWS::::Ac	For directions on how to fix this issue, consult the AWS Security Hub Founda	https://docs.aws.amazon.com/console/securityhu				
26	7.01E+10	us-east-1	EC2.22 Unused EC2 security groups	This AWS	DPP - Data	NonComp	Security H	MEDIUM	arn:aws:e	For directions on how to fix this issue, consult the AWS Security Hub Founda	https://docs.aws.amazon.com/console/securityhu				
27	7.01E+10	us-east-1	3.9 Ensure a log metric filter and al	Real-time	NotMappe	NonComp	Security H	LOW	AWS::::Ac	For directions on how to fix this issue, consult the AWS Security Hub CIS doc	https://docs.aws.amazon.com/console/securityhu				
28	7.01E+10	us-east-1	On instance i-0e54a04245aa3858e,	On this in	NotMappe	NonComp	Inspector	HIGH	arn:aws:e	You can edit the Security Group sg-0f93be09acd0ed4d8 to remove access from the internet on port 21					
29	7.01E+10	us-east-1	KMS.2 IAM principals should not h	Checks wh	DPP- Cred	NonComp	Security H	MEDIUM	arn:aws:ia	For directions on how to fix this issue, consult the AWS Security Hub Founda	https://docs.aws.amazon.com/console/securityhu				
30	7.01E+10	us-east-1	S3.9 S3 bucket server access loggin	This contr	DPP - Logg	NonComp	Security H	MEDIUM	arn:aws:s3	For directions on how to fix this issue, consult the AWS Security Hub Founda	https://docs.aws.amazon.com/console/securityhu				
31	7.01E+10	us-east-1	1.3 Ensure credentials unused for 9	AWS IAM	NotMappe	NonComp	Security H	MEDIUM	arn:aws:ia	For directions on how to fix this issue, consult the AWS Security Hub CIS doc	https://docs.aws.amazon.com/console/securityhu				
32	7.01E+10	us-east-1	Aggregate network exposure: On i	On instan	DPP - Net	NonComp	Inspector	INFORMATIONAL	arn:aws:e	You can edit the Security Group sg-0f93be09acd0ed4d8 to remove access from the internet					
33	7.01E+10	us-east-1	CloudTrail.5 CloudTrail trails shoul	This AWS	NotMappe	NonComp	Security H	LOW	arn:aws:cl	For directions on how to fix this issue, consult the AWS Security Hub Founda	https://docs.aws.amazon.com/console/securityhu				
34	7.01E+10	us-east-1	EC2.15 EC2 subnets should not aut	This contr	DPP - Net	NonComp	Security H	MEDIUM	arn:aws:e	For directions on how to fix this issue, consult the AWS Security Hub Founda	https://docs.aws.amazon.com/console/securityhu				
35	7.01E+10	us-east-1	RDS.23 RDS instances should not u	This contr	NotMappe	NonComp	Security H	LOW	arn:aws:rd	For directions on how to fix this issue, consult the AWS Security Hub Founda	https://docs.aws.amazon.com/console/securityhu				
36	7.01E+10	us-east-1	S3.13 S3 buckets should have lifec	This contr	DPP - Logg	NonComp	Security H	LOW	arn:aws:s3	For directions on how to fix this issue, consult the AWS Security Hub Founda	https://docs.aws.amazon.com/console/securityhu				
37	7.01E+10	us-east-1	IAM.2 IAM users should not have I	This AWS	NotMappe	NonComp	Security H	LOW	arn:aws:ia	For directions on how to fix this issue, consult the AWS Security Hub Founda	https://docs.aws.amazon.com/console/securityhu				

Рисунок 3.7 – Звіт щодо аудиту Amazon SP-API

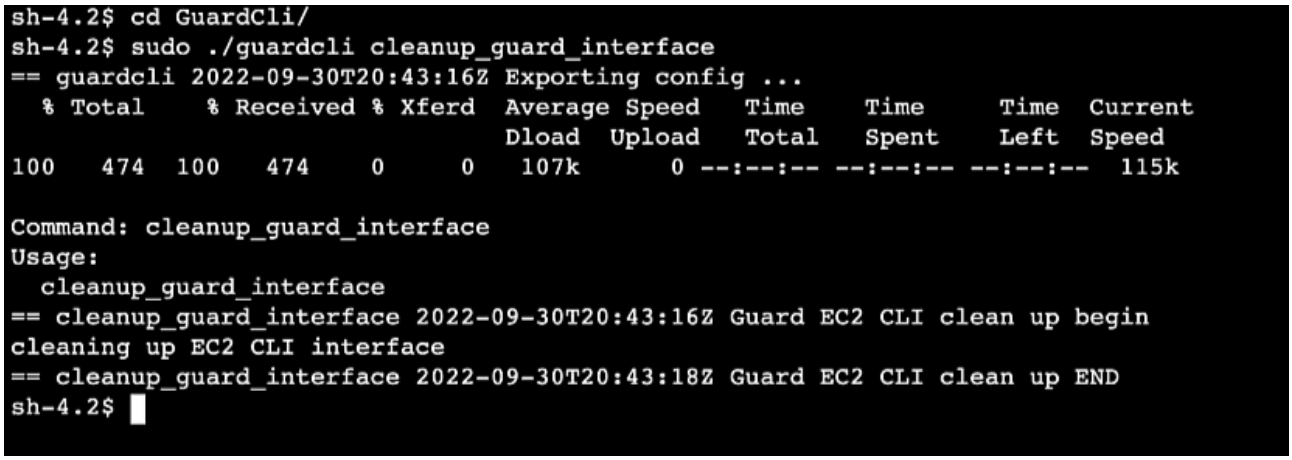

Рисунок 3.8 – Виключення Amazon SP-API Guard

Звіт включає назву пункту з політики безпеки, яку необхідно виконати, пріоритетність, джерело вразливості, формує рекомендацію як виконати правки та вдосконалення з метою проходження наступного аудиту. Використання такого інструменту є надзвичайно ергономічним, проте підходить лише для тих додатків, що розміщенні на сервісах AWS.

Основні вразливості знайдені з використанням Amazon API Guard представлені табл. 3.1.

Таблиця 3.1 Отримані вразливості (високий пріоритет) щодо застосування SP-API для досліджуваного SAAS рішення

Назва сервісу	Вразливість	Пріоритет	Вирішення проблеми
Macie	Об’єкт S3 містить особисту інформацію. (DPP)	Високий	Необхідно шифрувати всю ідентифікаційну інформацію в стані спокою за допомогою AES-128, AES-256 або RSA з розміром ключа 2048 біт.

Продовження таблиці 3. 1

Назва сервісу	Вразливість	Пріоритет	Вирішення проблеми
Security Hub	EC2.18 Групи безпеки повинні дозволяти лише необмежений вхідний трафік для авторизованих портів, DPP	Критична	Необхідно правильно налаштувати групи безпеки.
Security Hub	Hardware MFA має бути ввімкнено для користувача root	Критично	Необхідно увімкнути MFA.
Inspector	Для сегментів S3 має бути ввімкнено шифрування на стороні сервера	Критично	Необхідно шифрувати всю ідентифікаційну інформацію в стані спокою за допомогою AES-128, AES-256 або RSA з розміром ключа 2048 біт.
Inspector	Загальний доступ до мережі: на екземплярі i-0f87e748ba312266d порти доступні з Інтернету через ENI eni-03d0b8b451ce59ae3 і групу безпеки sg-7d22b634. (DPP)	Високий	Необхідно змінити групу безпеки sg-7d22b634, щоб видалити доступ з Інтернету
Inspector	В екземплярі i-0e54a04245aa3858e TCP-порт 21, пов’язаний із «FTP», доступний з Інтернету	Високий	Необхідно змінити групу безпеки sg-0f93be09acd0ed4d8, щоб видалити доступ до Інтернету через порт 21

Продовження таблиці 3. 1

Назва сервісу	Вразливість	Пріоритет	Вирішення проблеми
IAM Access Analyzer	AwsIamRole/arn:aws:iam::070145989624:role/service-role/CWDBSharing-PublicReadOnlyAccess-WYMVQC3C/ дозволяє доступ між обліковими записами, Developer Agreement	Високий	Необхідно змінити або видалити політику, яка надає ненавмисний доступ.

На основі отриманої інформації та рекомендацій було сформовано план дій для подальшого виправлення вразливостей.

3.6 Порівняння результатів

Для порівняння результатів аудиту зазначимо основні метрики: час виконання аудиту, вартість витрачених коштів на аудит, отримані результати.

Оскільки архітектура SAAS рішення є реалізованою на AWS, то першим варіантом для виконання аудиту було використання нативного Amazon API Guard. Необхідно зауважити, що процедура його використання є надзвичайно простою, а звіт повністю адаптованою для того, щоб отримати доступ до усіх методів SP-API. Відповідно розробнику не потрібно вичитувати DPP та перевіряти вказані вимоги.

Для виконання другого варіанту було використано цілий стек програмних рішень як безкоштовних, так і з мінімальною підпискою, та які представлені наступним списком:

- Nmap: безкоштовна утиліта для сканування відкритих портів, виявлення мережі (інвентаризація, оновлення, моніторинг часу роботи хоста) та перевірки її безпеки;

- OpenVAS – це повнофункціональний сканер вразливостей. Його можливості включають не автентифіковані та автентифіковані тести, різноманітні високо та низькорівневі Інтернет-та промислові протоколи, налаштування продуктивності для широкомасштабного сканування та потужні інструменти для реалізації всіх типів тестування вразливостей. Містить внутрішню мову програмування.
- OWASP Zed Attack Proxy безкоштовна альтернатива з відкритим кодом для тестування веб-додатків на можливість проникнення.
- APISec (безкоштовна місячна підписка) комплексне тестування безпеки API для автоматичного створення та написання тестів, адаптованих до вашої архітектури API. При цьому інтегрується із AWS.
- Metasploit (безкоштовна версія) – готовий інструментарій для перевірки вразливості, керування оцінками безпеки та покращення обізнаності про безпеку.

Другий варіант є повністю безкоштовним щодо його застосування. Не беремо до уваги той факт, що стандартний пакет APISec вартує $500 за місяць, що становить таку ж саму вартість як і застосування Amazon API Guard.

Для реалізації першого варіанту було витрачено лише пів години часу для інсталяції даної компоненти, після цього сервіс автоматично без сторонньої допомоги виконував перевірку та тестування.

У другому випадку було витрачено 2 робочих дні для отримання інформації, перевірки даних, налаштувань. При цьому звіти були розподілені відповідно до кожної використаної програми і необхідно було виконувати додаткову агрегацію результатів тестування.

Порівняльний аналіз тестування з використанням двох варіантів наведено у табл. 3.2.

Таблиця 3.2 Порівняльний аналіз застосування Amazon API Guard та агрегованого стеку додатків.

Критерій	Amazon API Guard	Nmap + OpenVAS + OWASP Zed Attack Proxy + APISec + Metasploit
Вартість	$450	безкоштовно
Витрачений час	1 година	17 годин
Результати	Отримано 91 вразливість, виправлення яких забезпечить проходження аудиту	Не знайдено файли на s3 із сенсативною інформацію; політики IAM, керовані клієнтом не повинні виконувати підстановки дій для служб.
Зручність	5	3
Повнота звіту	5	4
Рекомендації	+	-

Результати порівняльного аналізу свідчать про доцільність використання Amazon API Guard у випадку, якщо ваше рішення використовує AWS як середовище розгортання та виконання. Звичайно, якщо ви використовуєте інші хмарні сервіси для реалізації веб-додатків, то необхідно для аудиту API використовувати цілий стек додатків та надсилати деталі звіти для проходження Amazon Selling Partner API Audit.

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Охорона праці

Основними законами, та підзаконними актами, які регулюють правові відносини у сфері охорони праці в умовах виробництва і загалом в суспільстві є:

- Конституція України : Основний Закон держави забезпечує рівність прав чоловіків і жінок, гарантує право на працю, на належні і безпечні умови праці, на заробітну плату, на відпочинок, соціальний захист, на охорону здоров'я.
- Кодекс законів про працю (КЗпП). Регулює правові відносини всіх працівників, підвищенню ефективності суспільного виробництва. У КЗпП зафіксовано питання трудового законодавства, шляхи створення здорових і безпечних умов праці, приділяємо значну увагу охороні праці жінок та молоді, розглянуто питання нагляду та контролю за дотриманням законодавства про працю, відповідальність за порушення законодавства про працю та інші.
- Закони : "Про охорону праці", "Про охорону здоров'я", "Про пожежну безпеку", "Про забезпечення санітарного та епідеміологічного благополуччя населення".

При безпечних умовах праці має бути виключено вплив на працюючих небезпечних і шкідливих виробничих факторів. Але не завжди в умовах реального виробництва це досягається: абсолютна безпека або технічно недосяжна, або економічно недоцільна. Тому при розробці сучасного обладнання прагнуть створити максимально безпечні машини, обладнання, установки і прилади, щоб звести ризик при роботі з ними до мінімуму.

Для реалізації принципів державної політики в галузі охорони праці при Кабінеті Міністрів створено Національну раду з питань безпечної життєдіяльності, Фонд соціального страхування від нещасних випадків та інші структури.

Впровадження досягнень НТП вносить суттєві зміни у сферу матеріального виробництва, впливає на безпеку праці. Застосування нових технологій та

матеріалів, автоматизованого обладнання значно зменшує фізичне навантаження на працюючого, підвищує продуктивність праці та значно поліпшує якість продукції. Застосування досягнень електроніки та кібернетики дозволяє вивести із шкідливих та небезпечних зон працівника та суттєва зменшити ризик виникнення виробничого травматизму та професійних захворювань.

Разом з тим, у зв'язку з ускладненням нових технологій, обладнання виникають нові потенційно небезпечні фактори: іонізуючого випромінювання, хімічно небезпечні речовини, додаткові шумові, вібраційні навантаження, електромагнітні поля тощо. Суттєво зростає роль працівника, поза як помилки його в роботі значно ускладнює наслідки від аварій і призводить до більш значної кількості потерпілих та більш негативних екологічних наслідків.

Впровадження досягнень НТП вимагає більш суворого дотримання правил техніки безпеки та охорони праці заради збереження здоров'я та життя працюючих.

Згідно з чинним законодавством держава бере на себе зобов'язання щодо управління охороною праці, дбає про забезпечення здорових та безпечних умов праці. Державне управління охороною праці здійснює Кабінет Міністрів України через спеціально уповноважений нейтральний орган виконавчої влади з нагляду за охороною праці (Держнаглядохоронпраці), міністерства та інші центральні органи виконавчої влади :

- Держнаглядохоронпраці контролює дотримання правил техніки безпеки та охорони праці, безпечної експлуатації всіх видів обладнання, безпечного ведення технологічних процесів тощо;
- Держсаннагляд контролює дотримання правил виробничої гігієни та санітарії на виробництві та створення здорових умов праці;
- Держпожежнагляд контролює дотримання правил пожежної безпеки та виконання заходів, спрямованих на попередження пожеж;
- Державний інспектор має право безперешкодного входу на територію підприємства (організації) за яким він закріплений, може зупинити роботу підприємства, цеху чи дільниці до усунення недоліків, що ці недоліки загрожують

здоров'ю чи життю працюючих, може видавати пропис на усунення недоліків, може накладати адміністративні стягнення.

Державний контроль можуть здійснювати органи прокуратури, місцеві державні адміністрації, органи місцевого самоврядування.

Громадський контроль, як правило, у межах своєї компетентності здійснюють професійні спілки через технічну та правову інспекції, комісією з охорони праці та громадських уповноважених з охорони праці.

4.2 Організація оповіщення і зв’язку у надзвичайних ситуаціях техногенного та природного характеру

Одним із головних заходів захисту населення від надзвичайних ситуацій (НС) є його своєчасне оповіщення про небезпеку, обстановку, яка склалася внаслідок її реалізації, а також інформування про порядок і правила поведінки в умовах НС. Під час організації оповіщення і доведення інформації до населення України необхідно керуватися вимогами Положення про організацію оповіщення і зв'язку у надзвичайних ситуаціях, затвердженого постановою Кабінету Міністрів України від 15 лютого 1999 року № 192. Кожний громадянин України повинен знати порядок подавання сигналу “Увага всім!”, діяти за ним та іншими сигналами цивільного захисту (ЦЗ) в умовах НС та особливого періоду.

Встановлено, що система оповіщення та інформування у сфері ЦЗ України включає:

- оперативне доведення до відома населення інформації про виникнення або можливу загрозу виникнення НС, у тому числі через загальнодержавну, територіальні і локальні автоматизовані системи централізованого оповіщення;
- завчасне створення та організаційно-технічне поєднання постійно діючих локальних систем оповіщення та інформування населення із спеціальними системами спостереження і контролю (включаючи державну мережу спостереження і лабораторного контролю) в зонах можливого ураження;

- централізоване використання мереж зв'язку, радіомовлення, телебачення та інших технічних засобів передачі інформації незалежно від форми власності та підпорядкування в разі виникнення НС.

Системи оповіщення населення України мають державний, регіональний, місцевий і об'єктовий рівні. Управління системою оповіщення кожного рівня організовується безпосередньо відповідними органами повсякденного управління системи ЦЗ. Рішення на застосування системи оповіщення приймає відповідний голова державної адміністрації (начальник територіальної підсистеми Єдиної системи цивільного захисту). Відповідальність за організацію і практичне здійснення оповіщення несуть керівники органів виконавчої влади, місцевого самоврядування, підприємств, установ і організацій. Тому керівник об'єкта господарської діяльності і кожний громадянин повинні знати сигнали ЦЗ і уміти правильно за ними діяти.

В результаті наукової розвідки встановлено, що в Єдиній системі ЦЗ України оповіщення населення передбачає спочатку, за будь-якого характеру небезпеки, включення електричних сирен, переривчастий звук яких означає єдиний сигнал небезпеки "Увага всім!". Для вирішення завдань оповіщення на всіх рівнях Єдиної системи ЦЗ створюються спеціальні системи централізованого оповіщення (СЦО). Системою оповіщення будь-якого рівня є організаційно-технічне об'єднання оперативно чергових служб органів управління ЦЗ, спеціальної апаратури управління і засобів оповіщення, а також каналів (ліній зв'язку), які забезпечують передачу команд управління і мовної інформації у НС. СЦО регіонального рівня є основною ланкою системи оповіщення в цілому. Саме з цього рівня планується організація централізованого оповіщення. Завданням СЦО регіонального рівня є оповіщення посадових осіб і сил даного рівня, органів управління, сил місцевого і об'єктового рівнів та їх посадових осіб, а також населення, яке проживає на території, на яку поширюється дія СЦО цього рівня. Інформація, яка доводиться до органів управління і посадових осіб, має оперативний характер, а до населення доводиться інформація про характер і масштаби загрози та про дії в умовах НС, які склалися.

Дослідженням встановлено, що основним способом оповіщення населення про НС в умовах мирного та воєнного часу є передача інформації з використанням державних мереж проводового, радіо і телевізійного мовлення. Для зосередження уваги населення перед передачею інформації вмикаються сирени, виробничі гудки та інші сигнальні засоби, що буде означати подання попереджувального сигналу "Увага всім!", після якого негайно приводяться в готовність радіотрансляційні вузли, радіомовні і телевізійні станції, вмикаються мережі зовнішньої звукофікації. За сигналом населення зобов'язане увімкнути радіотрансляційні та телевізійні приймачі для прослуховування нагального повідомлення. У всіх випадках використання систем оповіщення, з увімкненням сирен, негайно доводиться до населення відповідне повідомлення засобами проводового, радіо та телевізійного мовлення. Тексти повідомлень передаються протягом 5 хвилин державною мовою і мовою, якою користується більшість населення в регіоні з припиненням іншої передачі. Тексти звернень записуються на магнітних стрічках на весь обсяг касети з обох сторін. Фонограми і друковані тексти звернень зберігаються в запечатаних конвертах в оперативних чергових з питань НС, які в необхідних випадках доводяться до населення. Дублікати фонограм і друкованих текстів звернень зберігаються в запечатаних конвертах на радіотрансляційних вузлах, в апаратних радіомовлення, студіях телебачення і використовуються в разі виходу з ладу апаратури оповіщення або аварії на з'єднувальній лінії зв’язку.

У разі повітряної тривоги: “Увага! Говорить Головне управління (управління, відділ) з питань НС облдержадміністрації (міськвиконкому, райдержадміністрації). Громадяни! Повітряна тривога! Відключіть світло, газ, погасіть вогонь у печах. Візьміть засоби індивідуального захисту, документи, запас харчів та води. Попередьте сусідів і допоможіть хворим та людям похилого віку вийти на вулицю. Якнайшвидше дістаньтеся захисної споруди або заховайтеся на місцевості. Дотримуйтеся спокою та порядку. Уважно слухайте повідомлення Головного управління (управління, відділу) з питань НС облдержадміністрації (міськвиконкому, райдержадміністрації)”.

Після повітряної тривоги: “Увага! Говорить Головне управління (управління, відділ) з питань НС облдержадміністрації (міськвиконкому, райдержадміністрації). Відбій повітряної тривоги! Усім повернутися до місць роботи або проживання. Допоможіть у цьому хворим та людям похилого віку. Будьте готові до можливого повторного нападу противника. Завжди майте з собою засоби індивідуального захисту. Уважно слухайте повідомлення Головного управління (управління, відділу) з питань НС облдержадміністрації (міськвиконкому, райдержадміністрації)”.

У разі загрози хімічного зараження: "Увага! Говорить Головне управління (управління, відділ) з питань НС облдержадміністрації (міськвиконкому, райдержадміністрації). Громадяни! Виникла безпосередня загроза хімічного зараження. Одягніть протигази, сховайте дітей у дитячих захисних камерах. Для захисту поверхні тіла використовуйте захисний одяг, комбінезони та чоботи. При собі майте плівкові (полімерні) накидки, куртки або плащі. Перевірте герметизацію житлових приміщень, стан вікон та дверей. Загерметизуйте продукти харчування і запасіться водою. Укрийте сільськогосподарських тварин і корми. Допоможіть хворим та людям похилого віку. Сповістіть сусідів про одержану інформацію. Відключіть електронагрівальні прилади. Надалі дійте відповідно до вказівок Головного управління (управління, відділу) з питань НС облдержадміністрації (міськвиконкому, райдержадміністрації)”.

У разі загрози радіоактивного зараження: “Увага! Говорить Головне управління (управління, відділ) з питань НС облдержадміністрації (міськ-виконкому, райдержадміністрації). Громадяни! Виникла безпосередня загроза радіоактивного зараження. Приведіть у готовність засоби індивідуального захисту та постійно майте їх із собою. Після команди управління (відділу) з питань НС та у справах захисту населення від наслідків Чорнобильської катастрофи надягніть їх. Для захисту поверхні тіла від забруднення радіоактивними речовинами використовуйте захисний одяг, комбінезони та чоботи. При собі майте плівкові (полімерні) накидки, куртки або плащі. Перевірте герметизацію житлових приміщень, стан вікон та дверей. Загерметизуйте продукти харчування і

запасіться водою. Укрийте сільськогосподарських тварин і корми. Сповістіть сусідів про одержану інформацію. Допоможіть хворим та людям похилого віку. Надалі дійте відповідно до вказівок Головного управління (управління, відділу) з питань НС облдержадміністрації (міськвиконкому, райдержадміністрації)".

Уразі загрози біологічного зараження: "Увага! Говорить Головне управління (управління, відділ) з питань НС облдержадміністрації (міськ-виконкому, райдержадміністрації). Громадяни! Виникла безпосередня загроза біологічного зараження. Для захисту поверхні тіла використовуйте захисний одяг, комбінезони та чоботи. Із собою майте плівкові (полімерні) накидки, куртки або плащі. Перевірте герметизацію житлових приміщень, стан вікон та дверей. Загерметизуйте продукти харчування і запасіться водою. Укрийте сільськогосподарських тварин і корми. Допоможіть хворим та людям похилого віку. Сповістіть сусідів про одержану інформацію. Відключіть електронагрівальні прилади. Надалі дійте відповідно до вказівок Головного управління (управління, відділу) з питань НС облдержадміністрації (міськвиконкому, райдержадміністрації)".

ВИСНОВКИ

В першому розділі здійснено аналітичних огляд тенденцій в галузі електронної комерції та онлайн-продаж в цілому. Споживачі за 2021 рік витратили близько 5 мільярдів доларів, купуючи товари через мережу інтернет. Це свідчить про колосальне зростання у порівнянні із 2020 роком. Лідером онлайн-продаж є Азіатсько-Тихоокеанський регіон, частка якого становить 40,92% від загального ринку. Розглянуто основні компоненти е-марткеплейсів. Значну увагу приділено API онлайн-площадки для продаж, що забезпечує покращення швидкості, узгодженості та гнучкості; дозволяє швидкий обмін документами; дозволяє бізнес-користувачам створювати програмне забезпечення для власних потреб. Перелічені фактори сприяють зростанню кількості е-маркетплейсів із відкритими методами API.

Для здійснення аудиту будь якої API необхідно розуміти предметну область та основні методології. В другому розділі висвітлено основні переваги REST API: відкритість взаємодії; проста реалізація; кешування даних на рівні HTTP; стабільність; можливість опрацьовувати різноманітні формати даних. Також представлено основні джерела виникнення вразливостей у REST API: контроль над доступом, аутентифікацією та авторизацією. Розглянуто модель Річардсона, оскільки вона є добре відомою моделлю для оцінки відповідності впровадження RESTful API. Представлено систему аудиту безпеки API на основі трафіку та наведено детальну структуру аудиту безпеки.

В практичній частині представлено детальний огляд готового SAAS рішення для багатоканальних продаж та розглянуто архітектуру, що є реалізованою на AWS. Проаналізовано вимоги щодо відповідності політиці захисту даних. Досліджено функціональні можливості Amazon Selling Partner API. Проведено аудит з використанням нативного інструменту від Amazon (опція 1) та збірки додатків (опція 2): Nmap, OpenVAS, OWASP Zed Attack Proxy, APISec, Metasploit. Результати порівняльного аналізу свідчать про доцільність та оперативність застосування Amazon API Guard лише у випадку, якщо рішення, що

використовує Selling Partner API є розгорнутим на AWS. Вартість одноразового застосування становить $450 у порівнянні із безкоштовним пакетом другої опції, проте витрачений час спеціаліста з кібербезпеки є рівнозначним до витрат за перший сервіс. Результати звіту з використанням першої опції є більш детальним, оскільки повністю відповідають політикам Amazon, результати з використанням другої опції є загальними, інформацію потрібно збирати, аналізувати та узагальнювати для подальшого застосування. У випадку, якщо додаток є розгорнутим на іншій інфраструктурі, то доцільності в застосуванні API Guard не має. Даний сервіс є спроектованим для використання з метою полегшення проходження аудиту та своєрідно підштовхує до переходу на AWS.

В подальших дослідженнях доцільно виконати проходження аудиту застосування Amazon Selling Partner API на готовому рішенні, що є розгорнутим на іншому сервісі, відмінному від AWS.

СПИСОК ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. ACI. (2022). Global eCommerce Retail Sales Up 209 Percent in April, ACI Worldwide Research Reveals. Retrived from https://investor.aciworldwide.com/news-releases/news-release-details/global-ecommerceretail-sales-209-percent-april-aci-worldwide
2. Briedis, H., Kronschnabl, A., Rodriguez, A., & Ungerman, K. (2022). Adapting to the next normal in retail: The customer experience imperative. McKinsey & Company. Retrieved from http://dln.jaipuria.ac.in:8080/jspui/bitstream/123456789/1510/1/Adapting-to-the-next-normal-in-retail-the-customer-experienceimperative.pdf
3. Coppola, D. (2021). E-commerce worldwide: Statistics & facts. Statista.
4. Thakur A., Trends and analysis of e-commerce market: a global perspective. International Journal of Applied Marketing and Management, 2021, vol 6, p. 11-22.
5. E., Turban, D., King, J. Lee. Electronic Commerce 2008: A Managerial Perspective, Prince Hall, p. 42-56.
6. Columbus L. (2018) Predicting The Future Of Digital Marketplaces, Retrived from https://www.forbes.com/sites/louiscolumbu s/2018/10/21/predicting-the-future-ofdigital-marketplaces/#68dd908d1d0e
7. Webretailer (2022). Online marketplaces in the usa: amazon is not the only show in town. Retrieved from https://www.webretailer.com/marketplaces-worldwide/online-marketplaces-usa/
8. Webretailer (2022). The world’s top online marketplaces 2022. RETRIEVED FROM https://www.webretailer.com/marketplaces-worldwide/online-marketplaces/#h-what-has-changed-in-2022
9. Webretailer (2022). Online marketplaces in the uk: amazon and ebay dominate. Retrieved from https://www.webretailer.com/marketplaces-worldwide/online-marketplaces-uk/

10. Adams, R. Jr. (2018). Overcoming disintermediation: A call for librarians to learn to use web service APIs. Library Hi Tech, 36(1), 180–190.
11. Scheller, T., & Kühn, E. (2015). Automated measurement of API usability: The API Concepts Framework. Information and Software Technology, 61, 145–162.
12. Sandos W. (2022) Spotting API Security Trends in ProgrammableWeb's API Directory. Retrieved from https://www.programmableweb.com/news/spotting-api-security-trends-programmablewebs-api-directory/research/2018/01/02
13. Shnier, M., Dictionary of PC Hardware and Data Communications Terms, O'Reilly Media, Inc., Sebastopol, CA, 1996.
14. Sandos W. (2022) Most popular API's. Retrieved from https://www.programmableweb.com/news/spotting-api-security-trends-programmablewebs-api-directory/research/2018/01/02
15. Laptev, N., & Amizadeh, S.. A labeled anomaly detection dataset S5 Yahoo Research, v1. https://webscope.sandbox.yahoo.com/catalog.php?datatype=s&did=70.
16. Jacobson, D., Brail, G. and Woods, D., APIs: A Strategy Guide, O'Reilly Media, Inc., Sebastopol, CA, 2011.
17. OWASP (2022). Owasp Top 10 Security Risk and Vulnerabilities. Retrived from https://sucuri.net/guides/owasp-top-10-security-vulnerabilities-2020/
18. Zachariadis, M., & Ozcan, P. The API Economy and Digital Transformation in Financial Services: The case of Open Banking. Swift Institute.
19. Diprose, J., MacDonald, B., Hosking, J., & Plimmer, B. (2017). Designing an API at an appropriate abstraction level for programming. Journal of Visual Languages and Computing, 39, 22–40
20 Vaccari, L. and Santoro, M., API standards and technical specifications - APIs4DGov, European Commission, Joint Research Centre (JRC), 2019.
21 Hussain, F., Li, W., Noye, B., Sharieh, S., Ferworn, A.: Intelligent service mesh framework for API security and management. In: 2019 IEEE 10th Annual

Information Technology, Electronics and Mobile Communication Conference (IEMCON). pp. 0735–0742. IEEE (2019)

22 Song, Y.: Resarch and implementation of monitoring of monitoring oriented open API service. Ph.D. thesis, Beijing University of Posts and Telecommunications.

23 Ramesh, G., Menen, A.: Automated dynamic approach for detecting ransomware using finitestate machine. Decis. Supp. Syst. 138, 113400 (2020)

ДОДАТКИ

Додаток А

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ІВАНА ПУЛЮЯ

М А Т Е Р І А Л И

X НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

«ІНФОРМАЦІЙНІ МОДЕЛІ, СИСТЕМИ ТА ТЕХНОЛОГІЇ»

7–8 грудня 2022 року

ТЕРНОПІЛЬ
2022

УДК 004
Р. Волошин
(Тернопільський національний технічний університет імені Івана Пулюя)

АУДИТ БЕЗПЕКИ AMAZON SELLING PATRNER API

UDC 004
R. Voloshyn

AMAZON SELLING PATRNER API CYBERSECURITY AUDIT

Selling Partner API (SP-API) – це API на основі технології REST, яка допомагає партнерам з продажу Amazon програмно отримувати доступ до своїх даних про замовлення, відправлення, платежі та багато іншого. Програми, що використовують SP-API, можуть підвищити ефективність продажів, зменшити потреби в робочій силі та покращити час відповіді клієнтам, допомагаючи партнерам з продажу розвивати свій бізнес. Щоб мати змогу створити програму з використанням SP-API розробники повинні пройти реєстрацію з вказанням усіх деталей щодо бізнесу та рівня компанії, на яку вони працюють та аудит безпеки.

Аудит безпеки для Amazon Selling Partner API включає аудит щодо:

- захисту мережі (наявність засобів керування захистом мережі, включаючи мережеві брандмауери та списки контролю доступу до мережі, щоб заборонити доступ до несанкціонованих IP-адрес; сегментація мережі, програмне забезпечення для захисту від вірусів і шкідливих програм на пристроях кінцевих користувачів; публічний доступ лише для схвалених користувачів),
- керування доступом до даних та ресурсів (офіційний процес реєстрації доступу користувачів, щоб призначити права доступу для всіх типів користувачів і послуг, забезпечивши присвоєння унікального ідентифікатора кожній особі, алгоритм "блокування" користувача за спричинені несанкціоновані дії),
- шифрування під час передавання інформації (необхідно шифрувати всю інформацію, що передається, за допомогою безпечних протоколів, таких як TLS 1.2+, SFTP і SSH-2),
- шифрування REST запитів (за допомогою принаймні AES-128 або RSA з розміром ключа 2048-біт або вище),
- зберігання інформації щодо покупців на серверах без шифрування цієї інформації,
- наявність плану реагування на ризики та менеджмент ризиків, систем логування доступів та змін.

Даний аудит містить не лише ці пункти, але і значно більший перелік. У випадку, якщо веб-додаток розмішується на серверах Amazon, то для автоматизації використовують Amazon SP-API Guard. У випадку розміщення програми на інших серверах необхідно використовувати цілий стек програм для аналізу вимог та їх забезпечення, що в кінцевому випадку свідчить про значні витрати.

Література

1. Data Protection Policy. URL: https://sellercentral.amazon.com/mws/static/policy (2022).
2. Acceptable Use Policy. URL: https://sellercentral.amazon.com/mws/static/policy?documentType=AUP (2022).