Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ДОПУСТИТИ ДО ЗАХИСТУ:
В.о. завідувача кафедри
кібербезпеки
та захисту інформації
_____ Іван ПАРХОМЕНКО
«17» травня 2024 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи

галузь знань *12 Інформаційні технології*
(шифр і назва галузі знань)

спеціальність *125 Кібербезпека*
(код і назва спеціальності)

освітній ступень *магістр*

освітньо-наукова програма *Кібербезпека*
(назва освітньої програми)

на тему: «Удосконалення захищеної інфраструктури вебдодатку за допомогою клаудпровайдера AWS»

Виконавець: студент ІІ курсу, групи КБм-22
_____ **Микола ГРАСС**
(підпис) (Ім'я, ПРІЗВИЩЕ)

	Ім'я, ПРІЗВИЩЕ	**Підпис**
Науковий керівник	Микола БРАІЛОВСЬКИЙ	
Нормоконтроль	Юрій БАБЕНКО	

Київ 2024

Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ЗАТВЕРДЖЕНО:
В.о. завідувача кафедри
кібербезпеки
та захисту інформації
_____ Іван ПАРХОМЕНКО
«17» листопада 2023 р.

ЗАВДАННЯ
на виконання кваліфікаційної роботи

спеціальності *125 Кібербезпека*
(код і назва спеціальності)

освітній ступень *магістр*

Здобувача КБм-22 (група) Грасса Миколи Вікторовича (прізвище ім'я по-батькові)

Тема кваліфікаційної роботи Удосконалення захищеної інфраструктури вебдодатку за допомогою клаудпровайдера AWS

1. ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Рішення засідання кафедри кібербезпеки та захисту інформації факультету інформаційних технологій протокол № 5 від 15.11.2023 р.

2. МЕТА ТА ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Об'єкт досліджень Процес удосконалення захищенної інфраструктури вебдодатку.

Предмет досліджень Засоби та методи захисту інфраструктури вебдодатків.

Мета Розробка удосконалення захисту інфраструктури вебдодатку за допомогою клаудпровайдера AWS.

Вихідні дані для проведення роботи Методи реагування на інциденти доступу в хмарному серидовищі.

ОЧІКУВАНІ НАУКОВІ РЕЗУЛЬТАТИ

Наукова новизна	покращення існуючих підходів до реагування на інциденти пов’язані з доступом в хмарному серидовищі
Практична цінність	Автоматиція реагування на інциденти інциденти пов’язані з доступом в хмарному серидовищі

4. ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Робота виконана у повному обсязі відповідно до теми.

5. ЕТАПИ ВИКОНАННЯ РОБОТИ

Найменування етапів робіт	Строки виконання робіт (початок-кінець)
Уточнення постановки задачі	17.11.2023 – 29.01.2024
Аналіз літературних джерел	30.01.2024 – 12.02.2024
Аналіз загроз для інфраструктури веб додатків	13.02.2024 – 21.02.2024
Розгляд існуючих методів захисту інфраструкутри веб додатків	22.02.2024 – 26.02.2024
Порівняння засобів захисту різних клаудпровайдерів	27.02.2024 – 04.03.2024
Розробка удосконаленого методу реагування на інциденти пов’язанів з доступом в хмарному серидовищі	05.03.2024 – 10.03.2024
Створення тестової інфраструктури	11.03.2024 – 17.03.2024
Розробка функції бекапу та відновлення політик доступу в AWS	18.03.2024 – 19.03.2024
Інтеграція фунцій до інфраструктури для автоматизованого реагування на інциденти	20.03.2024 – 17.04.2024
Тестування автоматизованого реагування на інцидети доступу в хмарній інфраструктурі	18.04.2024 – 25.04.2024
Оформлення пояснювальної записки згідно методичних рекомендацій	26.04.2024 – 12.05.2024
Подача пакету документів на розгляд ЕК	13.05.2024 – 17.05.2024

6. РЕАЛІЗАЦІЯ РЕЗУЛЬТАТІВ ТА ЕФЕКТИВНІСТЬ

Економічний ефект	Зниження витрат на реагування на інциденти безпеки в хмарному серидовищі
Соціальний ефект	Покращення забезпечення захисту інформаційних систем на підприємствах та інших організаціях.

7. ДОДАТКОВІ ВИМОГИ

Завдання видав _____ Микола БРАІЛОВСЬКИЙ
(підпис) (Ім'я, ПРІЗВИЩЕ)

Завдання прийняв
до виконання _____ Микола ГРАСС
(підпис) (Ім'я, ПРІЗВИЩЕ)

Дата видачі завдання: 17.11.2023 р.
Термін подання кваліфікаційної роботи до ЕК 17.05.2024 р.

РЕФЕРАТ

Пояснювальна записка дипломної роботи складається зі вступу, трьох розділів, загальних висновків, списку використаних джерел та додатків. Основний текст займає 105 сторінок, включає в себе зміст, вступ, три розділи дипломної роботи, висновки та список джерел. Крім того, робота містить 2 додатки із загальною кількістю сторінок 3. У пояснювальній записці дипломної роботи міститься 11 рисунків та 50 літературних джерел.

Об'єктом дослідження є процес удосконалення захищенної інфраструктури веб додатку

Метою роботи є удосконалення інфраструктури вебдодатку за допомогою клауд провайдера AWS

Предметом дослідження є засоби та механізми захисту інфраструктури вебдодатку

Методи дослідження дипломної роботи:

- аналіз літератури;
- аналіз документів;
- системний підхід;
- методи порівняння;
- структурний аналіз

Для досягнення зазначеної мети поставлено наступні завдання:

- дослідити загрози для інфраструктури вебдодатків ;
- проаналізувати засоби та механізми захисту інфраструктури;
- розробити удосконалений метод захисту інфраструктури за допомогою клауд провайдера AWS.

Практичною цінністю отриманих результатів є програмна реалізація покращення захищенної інфраструктури за допомогою клаудпровайдера AWS

Результати здійснених у дипломній роботі досліджень можуть бути використані у проведенні компаніями для покращення безпеки власної інфраструктури.

У подальшому можна вдосконалити програмну реалізацію додавши додаткові модулі та можливості.

Ключові слова: хмарні сервіси, вразливості, порти, мережеві протоколи.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

HTTP	–	Hypertext Transfer Protocol
RCE	–	Remote Code Execution
XSS	–	Cross Site Scripting
AWS	–	Amazon Web Services
UDP	–	User Datagram Protocol
DNS	–	Domain Name System
IP	–	Internet Protocol
GCP	–	Google Cloud Platform
RDP	–	Remote Desktop Protocol
SSH	–	Secure Shell
ACL	–	Access control list
RBAC	–	Role-Based Access Control
ABAC	–	Attribute-based access control
SQL	–	Structured query language
PE	–	Privilege Escalation
MS AD	–	Microsoft Active Directory
CVE	–	Common Vulnerabilities and Exposures
vNIC	–	Virtual Network Interface Cards
MDM	–	Mobile device management
EDR	–	Endpoint Detection and Responce
XDR	–	Extended detection and response
VPC	–	Virtual Private Cloud
SNS	–	Simple Notification Service

ЗМІСТ

ВСТУП

У сучасному світі, де технології розвиваються з неймовірною швидкістю, особливо актуальним стає питання кібербезпеки. Зі зростанням популярності хмарних обчислень, корпорації та індивідуальні користувачі все частіше стикаються з потребою захисту своїх даних у хмарі. Використання хмарних платформ, як Amazon Web Services (AWS), вимагає не тільки розуміння переваг таких технологій, але й усвідомлення потенційних загроз і ризиків [1].

Кібератаки стають все більш винахідливими та руйнівними, здатними порушити не тільки приватність інформації, але й привести до значних фінансових втрат та пошкодження репутації. Однією з ключових загроз хмарній інфраструктурі є несанкціонований доступ до ресурсів, що може статися через недоліки у політиках доступу або через зловмисні дії. Такі інциденти підкреслюють важливість комплексних систем безпеки, які можуть оперативно реагувати на потенційні загрози та ефективно управляти рівнем доступу до критичних ресурсів.

З огляду на це, розробка та впровадження вдосконалених методів захисту інфраструктури у хмарних середовищах стає однією з найважливіших справ

РОЗДІЛ 1
АНАЛІЗ ЗАГРОЗ ІНФРАСТРУКТУРИ ВЕБДОДАТКІВ

1.1 Загрози інфраструктури вебдодатків на мережевому рівні

Мережа, на якій базується інфраструктура вебдодатків, є першою лінією оборони проти кібер-загроз. Зловмисники можуть використовувати різні методи для атак на мережевому рівні, щоб зламати вебдодатки, викрасти дані або порушити роботу онлайн-сервісів.

Забезпечення безпеки мережі є надзвичайно важливим завданням для будь-якої компанії. Вразливості на мережевому рівні можуть призвести до серйозних наслідків, включаючи втрату конфіденційної інформації, пошкодження репутації компанії та фінансових втрат.

Один із ключових аспектів захисту мережі - це регулярне оновлення та патчінг програмного забезпечення та мережевих пристроїв. Вразливості у програмному забезпеченні можуть бути використані зловмисниками для втручання в мережу та злому систем.

Крім того, важливо мати високі стандарти безпеки мережі, включаючи встановлення сильних паролів, використання шифрування трафіку, налаштування брандмауерів та інші заходи. Регулярні аудити безпеки та моніторинг мережі також допомагають вчасно виявляти та виправляти потенційні загрози.

Загалом, захист мережі є критично важливим аспектом безпеки інфраструктури вебдодатків. Посилення заходів безпеки на мережевому рівні допомагає запобігти кібер-атакам, зберегти конфіденційність даних та забезпечити надійну роботу онлайн-сервісів.

Найпоширеніші типи атак на мережевому рівні

Розподілені атаки відмови в обслуговуванні (DDoS) – це небезпечний тип кібер-атак, який може завдати значної шкоди інфраструктурі вебдодатків та мати руйнівні наслідки для онлайн-бізнесу. Ці атаки спрямовані на перевантаження веб-сайту або

сервера величезною кількістю трафіку, роблячи його недоступним для легітимних користувачів [1-3].

Методи DDoS-атак:

Зловмисники використовують різні методи для здійснення DDoS-атак, включаючи:

- Бот-мережі: Зловмисники можуть використовувати мережі заражених комп'ютерів, відомі як бот-мережі, для надсилання одночасних запитів на веб-сайт або сервер. Ці боти можуть генерувати величезну кількість трафіку, який перевантажує цільову систему.
- Атаки типу "відмова в обслуговуванні" (DoS): Ці атаки ґрунтуються на надсиланні великої кількості запитів на веб-сайт або сервер з одного або кількох джерел. Ці запити можуть бути HTTP-запитами, запитами DNS або іншими типами трафіку, які виводять з ладу цільову систему.
- Атаки на протоколи: Зловмисники можуть експлуатувати вразливості в мережевих протоколах, таких як TCP або UDP, для надсилання шкідливого трафіку на веб-сайт або сервер. Ці атаки можуть призвести до переповнення буферів або інших збоїв, які роблять систему недоступною.

DDoS-атаки можуть мати руйнівні наслідки для онлайн-бізнесу, включаючи:

- Втрата доходу: Якщо веб-сайт недоступний протягом тривалого часу, це може призвести до значних фінансових втрат.
- Пошкодження репутації: DDoS-атаки можуть пошкодити репутації компанії та призвести до втрати довіри клієнтів.
- Зниження продуктивності: DDoS-атаки можуть вплинути на продуктивність інших систем та сервісів у мережі.
- Витрати на пом'якшення атаки: Боротьба з DDoS-атакою може бути дорогою, оскільки вона потребує використання спеціалізованих служб та програмного забезпечення.

Атака типу "людина посередині" (MitM) - це серйозний тип кібератаки, яка може завдати значної шкоди як приватним користувачам, так і організаціям. У цій

атаці зловмисник вклинюється між користувачем і веб-сайтом або сервером, перехоплюючи та змінюючи дані, що передаються між ними [3].

Як відбувається атака MitM:

1. Користувач намагається підключитися до веб-сайту або сервера.
2. Зловмисник перехоплює цей запит і надсилає користувачеві фейковий веб-сайт або сервер, який виглядає ідентично легітимному.
3. Користувач вводить свої дані (пароль, номер кредитної картки, особисту інформацію) на фейковому веб-сайті.
4. Зловмисник перехоплює ці дані та використовує їх у своїх цілях.

Наслідки атаки MitM:

- Крадіжка особистої інформації: Зловмисники можуть використовувати вкрадену особисту інформацію для крадіжки особистості, шахрайства або інших злочинних цілей.
- Фінансові втрати: Зловмисники можуть використовувати вкрадені номери кредитних карток або банківські реквізити для здійснення шахрайських транзакцій.
- Втрата конфіденційної інформації: Зловмисники можуть отримати доступ до конфіденційних даних, таких як комерційна таємниця або державна інформація.

Ознаки атаки MitM:

- Незвичні повідомлення про помилки або попередження: Якщо ви бачите незвичні повідомлення про помилки або попередження під час підключення до веб-сайту, це може бути ознакою того, що ви стали жертвою атаки MitM.
- Незвичні сертифікати безпеки: Перевірте сертифікат безпеки веб-сайту. Якщо сертифікат недійсний або виданий не тим, ким має бути, це може бути ознакою атаки MitM.

DNS Spoofing (DNS Poisoning) - DNS-отруєння, також відоме як підміна DNS або отруєння кешу DNS, є хакерською атакою, яка порушує нормальну роботу системи доменних імен (DNS). Під час атаки DNS-отруєння зловмисник обманює

DNS-сервер, змушуючи його надавати неправдиву інформацію для законного доменного імені [4]. Це може мати кілька наслідків:

Перенаправлення трафіку: Користувачі, які намагаються отримати доступ до законного веб-сайту (наприклад, вашого банку), мимоволі потрапляють на шахрайський веб-сайт, який виглядає майже ідентично. Це поширена тактика, яку використовують для крадіжки логінів, фінансової інформації або інших конфіденційних даних.

Відмова в доступі: Отруєний DNS-сервер може не відповісти на запит для певного домену, фактично роблячи веб-сайт недоступним для користувачів. Це можна використовувати для порушення роботи онлайн-сервісів або блокування доступу до критичної інформації.

Перехоплення зв'язку: У деяких випадках зловмисники можуть перехоплювати зв'язок між користувачем і фальшивим веб-сайтом, дозволяючи їм красти дані, що передаються туди й назад.

Як працює DNS-отруєння:

1 Зловмисник знаходить вразливий DNS-сервер. Це може бути сервер вашого інтернет-провайдера, сервер компанії, якою ви користуєтеся, або навіть публічний DNS-сервер.

2 Зловмисник отруює кеш DNS-сервера. Він робить це, надсилаючи серверу фальшиві записи DNS, які пов'язують легітимні веб-сайти з неправильними IP-адресами.

3 Коли ви намагаєтеся відвідати легітимний веб-сайт, ваш комп'ютер запитує IP-адресу у DNS-сервера.

4 Отруєний DNS-сервер надає вашому комп'ютеру фальшиву IP-адресу, що веде вас на фейковий веб-сайт, створений зловмисником.

На фейковому веб-сайті ви можете ввести свою особисту інформацію, думаючи, що ви на легітимному сайті. Зловмисник може використовувати цю інформацію для крадіжки ваших конфіденційних данних

RTP(Routing Table Poisoning) - це процес, який допомагає запобігти петлям маршрутизації шляхом позначення маршруту як недоступного або недійсного в

таблиці маршрутизації маршрутизатора. Після того, як маршрутизатор дізнається, що маршрут отруєно, він більше не використовуватиме його для передачі пакетів або оголошення іншим маршрутизаторам. Отруєння маршруту може відбуватися з різних причин, таких як обриви зв'язку, розділення мережі або будь-які зміни конфігурації [5].

Отруєння маршруту працює шляхом призначення нескінченної метрики або вартості маршруту, який вважається недосяжним або недійсним. Метрика — це числове значення, яке представляє бажаність маршруту, наприклад кількість переходів, пропускна здатність, затримка або надійність. Чим нижча метрика, тим кращий маршрут. Нескінченна метрика означає, що маршрут настільки небажаний, що його ніколи не можна використовувати чи рекламувати.

Різні протоколи маршрутизації мають різні визначення того, що є нескінченною метрикою. Наприклад, у RIP (Routing Information Protocol) нескінченна метрика становить 16 стрибків, тоді як у EIGRP (Enhanced Interior Gateway Routing Protocol) нескінченна метрика становить 2^32 – 1.

Коли маршрутизатор виявляє, що маршрут більше недійсний, він оновить свою таблицю маршрутизації за допомогою нескінченної метрики та надішле повідомлення про оновлення сусіднім маршрутизаторам. Повідомлення про оновлення міститиме пошкоджений маршрут і його нескінченну метрику. Сусідні маршрутизатори отримають повідомлення про оновлення та також позначать маршрут як пошкоджений у своїх таблицях маршрутизації. Потім вони поширять повідомлення оновлення своїм сусідам і так далі, доки всі маршрутизатори в мережі не дізнаються про шкідливий маршрут.

IP spoofing - це техніка, за допомогою якої зловмисники намагаються приховати свою ідентичність в мережі шляхом зміни або підробки джерела IP-адреси в пакетах даних, які вони відправляють. Зазвичай IP-підробка використовується з метою замаскування справжнього джерела атаки, приховання слідів або введення у заблукання систем оборони.

Підміна IP відбувається на рівні мережевого протоколу, коли зловмисник змінює заголовок пакета даних, вставляючи в нього фальшиву IP-адресу. Заголовок

пакета містить інформацію про джерело та призначення даних, і змінюючи цю інформацію, зловмисник може змінити вигляд та маршрутизацію пакета в мережі.

Щоб зрозуміти, як працює IP-підробка, важливо знати, що IP-адреса - це унікальний ідентифікатор пристрою в мережі Інтернет. Коли пристрій відправляє пакет даних, він включає свою IP-адресу як джерело, щоб отримувач міг відповісти на нього. Однак зловмисники можуть змінити цю IP-адресу на фальшиву, що робить важчим виявлення їхньої справжньої ідентичності [7, 8].

Техніка IP-підробки може бути використана для різних видів атак, включаючи:

DoS або DDoS атаки: Зловмисники можуть надсилати велику кількість запитів на сервери з фальшивими IP-адресами, змушуючи їх витрачати ресурси на обробку цих запитів і відмовляти в обслуговуванні легітимних користувачів.

Злам аутентифікації: Змінюючи IP-адресу, зловмисники можуть спробувати отримати несанкціонований доступ до систем, які обмежують доступ за IP-адресою.

Фішинг-атаки: Шахраї можуть використовувати IP-підробку для надсилання листів електронної пошти, які маскуються під повідомлення від відомих джерел, що збільшує ймовірність того, що отримувачі відкриють шкідливі посилання або відправлять конфіденційну інформацію.

1.2 Загрози інфраструктури вебдодатків на рівні хоста

На сучасних серверах, які забезпечують функціонування вебдодатків та зберігання важливої інформації, важливо зрозуміти, що вони становлять мішень для різноманітних кібератак. Забезпечення безпеки на рівні хоста, тобто на рівні самого сервера, є критично важливим завданням для будь-якої організації чи підприємства. Вразливості сервера можуть призвести до серйозних наслідків.

Вразливості хоста - це слабкі місця в програмному забезпеченні, операційній системі або конфігурації сервера, які можуть бути використані зловмисниками для проникнення в систему, крадіжки даних або виведення сервера з ладу, продовжи про загрози для веб додатків на хості, тільки не повторюйся [9]

Загрози для вебдодатків на рівні хоста

Вразливості хоста можуть мати значний вплив на безпеку вебдодатків, що розміщуються на ньому. Ось деякі з ключових ризиків, які слід враховувати:

1. Втрата даних: Зловмисники, які використовують вразливості хоста, можуть отримати доступ до конфіденційних даних, таких як номери кредитних карток, персональна інформація користувачів або комерційна таємниця. Ці дані можуть бути використані для крадіжки особистості, фінансових шахрайств або шантажу.

2. Несанкціонований доступ: Зловмисники можуть отримати повний контроль над сервером, що може призвести до:

- Видалення або зміни вебдодатків: Зловмисники можуть видалити або змінити код вебдодатків, що може призвести до їх непрацездатності або поширення шкідливого програмного забезпечення.
- Використання ресурсів сервера: Зловмисники можуть використовувати ресурси сервера для своїх цілей, таких як розсилка спаму або майнінг криптовалюти.
- Шантаж: Зловмисники можуть загрожувати оприлюднити або знищити дані, якщо їм не буде виплачено викуп.

Зараження шкідливим програмним забезпеченням: Зловмисники можуть заразити сервер шкідливим програмним забезпеченням, яке може:

- Викрасти дані: Шкідливе програмне забезпечення може збирати конфіденційні дані користувачів.
- Поширюватися на інші системи: Шкідливе програмне забезпечення може поширюватися на інші комп'ютери в мережі.
- Використовувати ресурси сервера: Шкідливе програмне забезпечення може використовувати ресурси сервера для своїх цілей, таких як розсилка спаму або майнінг криптовалюти.

Найпоширенішими загрозами на рівні хоста є:

- Вразливості програмного забезпечення: Зловмисники можуть використовувати відомі вразливості в програмному забезпеченні, яке використовується на сервері, для отримання доступу до системи. Це може включати

веб-сервери, бази даних, системи управління контентом (CMS) та інші програмні компоненти.

- Вразливості операційної системи: Зловмисники можуть використовувати недоліки в операційній системі сервера для отримання доступу до системи. Це може включати помилки в ядрі, вразливості в службах або неправильні налаштування.
- Неналежна конфігурація: Зловмисники можуть скористатися неправильними налаштуваннями сервера, щоб отримати доступу до системи. Це може включати слабкі паролі, відкриті порти, неправильні правила брандмауера або неправильні налаштування доступу до файлів
- Фішинг та соціальна інженерія: Зловмисники можуть використовувати фішингові електронні листи або веб-сайти, щоб обманути користувачів розкрити свої паролі або іншу конфіденційну інформацію. Цю інформацію потім можна використовувати для доступу до сервера [10].
- Атаки з нульовим днем: Зловмисники можуть використовувати невідомі вразливості в програмному забезпеченні або операційній системі для отримання доступу до сервера. Ці атаки є особливо небезпечними, оскільки для них немає виправлень.
- Атаки через веб-інтерфейс: Зловмисники можуть використовувати веб-інтерфейс сервера для запуску шкідливого коду або отримання доступу до системи. Це може включати атаки на веб-сайти, веб-служби або веб-панелі керування.
- Атаки з боку сторонніх постачальників: Зловмисники можуть використовувати вразливості в сторонніх постачальниках послуг, таких як хостинг-провайдери або постачальники хмарних послуг, для отримання доступу до сервера.

Зловмисники можуть використовувати вразливості програмного забезпечення на хості для здійснення різноманітних атак, наприклад

- Remote Code Execution(RCE) є серйозною вразливістю системи безпеки, яка дозволяє зловмиснику запускати довільний код на віддаленому комп'ютері або сервері. Це означає, що зловмисник може виконувати на вразливій системі будь-які дії, які зазвичай може виконувати законний користувач із таким самим рівнем

доступу. RCE може мати руйнівні наслідки, оскільки надає зловмиснику майже повний контроль над зараженою системою [11].

Один з видів атак пов’язаних з RCE – SQL ін’єкція або SQLi

SQLi це тип атаки на вебдодатки, при якій зловмисник використовує вразливість в обробці SQL запитів для внедрення шкідливого SQL коду в запити, які взаємодіють з базою даних. Ця атака може призвести до витоку конфіденційної інформації, втрати даних або навіть взяття під контроль бази даних [12].

В основі SQL ін'єкції лежить недостатня перевірка введених даних перед їх використанням у SQL запитах. Зловмисник може вставити шкідливий SQL код у поля вводу або параметри URL, які потім використовуються в запитах до бази даних без достатньої перевірки. Це дозволяє зловмисникам виконувати різноманітні дії, такі як отримання конфіденційних даних, видалення або модифікація існуючих записів в базі даних, або навіть призводити до повного вилучення бази даних.

Успішна атака SQL-ін’єкції може призвести до неавторизованого доступу до конфіденційних даних, наприклад:

- Паролі.
- Реквізити кредитної картки.
- Особиста інформація користувача.

Атаки SQL-ін’єкції протягом багатьох років використовувалися для багатьох гучних витоків даних. Це завдало шкоди репутації та штрафів. У деяких випадках зловмисник може отримати постійний бекдор до систем організації, що призведе до довгострокового злому, який може залишатися непоміченим протягом тривалого періоду.

Ось основні етапи типової SQL-ін'єкції:

1. Вразливий вхід: Веб-сайт або програма має форму введення даних, яка не перевіряє належним чином введені користувачем дані.

2. Введення шкідливого коду: Зловмисник вводить спеціальний код SQL через форму введення даних. Цей код може бути прихований у звичайному тексті або замаскований іншими символами.

3. Виконання коду SQL: Сервер бази даних не може відрізнити шкідливий код від звичайних даних і виконує його.

4. Наслідки атаки: Залежно від типу введеного коду SQL, зловмисник може отримати несанкціонований доступ до конфіденційних даних, змінити або видалити дані, або навіть повністю взяти під контроль сервер бази даних.

Існує декілька типів SQL-ін'єкцій, кожен з яких має різні цілі:

- In-band SQL injection (Внутрішньосмугова SQL-ін'єкція): Зловмисник отримує результати виконаного коду SQL через той самий канал, яким він його ввів (наприклад, через веб-сторінку).
- Out-of-band SQL injection (Позасмугова SQL-ін'єкція): Зловмисник змушує сервер бази даних виконати дії за межами звичайного каналу зв'язку (наприклад, надіслати електронного листа зловмиснику).
- Union-based SQL injection (SQL-ін'єкція на основі UNION): Зловмисник використовує оператор UNION в запиті SQL для отримання даних з інших таблиць бази даних.
- Blind SQL injection (Сліпа SQL-ін'єкція): Зловмисник покладається на непрямі ознаки успішного виконання коду SQL, оскільки він не може безпосередньо побачити результати [13].

Захист від SQL-ін'єкцій є критично важливим для безпеки веб-сайтів та програм, що використовують бази даних. Ось деякі методи захисту:

Валидація вводу: Перевіряйте всі дані, що вводяться користувачами, щоб переконатися, що вони відповідають очікуваному формату і не містять шкідливого коду.

Параметризовані запити: Використовуйте параметризовані запити, які відділяють дані від самого коду SQL. Це запобігає тому, щоб код користувача виконувався як частина запиту SQL.

Екранування спецсимволів: Екрануйте всі спеціальні символи у вхідних даних, щоб вони не інтерпретувалися як частина коду SQL.

Мінімальні права користувачів: Надавайте користувачам баз даних лише мінімальні права доступу, необхідні для виконання їхніх завдань.

Регулярні оновлення: Регулярно оновлюйте програмне забезпечення веб-сайту та бази даних, щоб усунути відомі вразливості.

Запобігання SQL-ін'єкціям є постійним процесом, оскільки зловмисники постійно розробляють нові методи атак. Важливо усвідомлювати цю загрозу та вживати необхідних заходів для захисту своїх систем.

Ще однією небезпечною атакою RCE – є Cross-Site Scripting (XSS)

Cross-Site Scripting (XSS) - це тип атаки на вебдодатки, при якій зловмисник вбудовує шкідливий скрипт у веб-сторінки, які потім виконуються в браузері користувачів. Ця атака використовується для отримання доступу до конфіденційної інформації, крадіжки сесійних файлів або виконання інших шкідливих дій в контексті веб-сторінки.

У XSS атаки зловмисник вбудовує скрипт у веб-сторінки, які потім відправляються користувачам. Коли користувачі відкривають ці сторінки у своєму браузері, вбудований скрипт виконується в їхньому контексті.

Зловмисник може використовувати XSS, щоб надіслати шкідливий сценарій нічого не підозрюючому користувачеві. Браузер кінцевого користувача не може дізнатися, що сценарію не можна довіряти, і виконає сценарій. Оскільки він вважає, що сценарій надійшов із надійного джерела, зловмисний сценарій може отримати доступ до будь-яких файлів cookie, маркерів сеансу чи іншої конфіденційної інформації, яка зберігається у веб-браузері і використовується на цьому сайті. Ці скрипти можуть навіть переписати вміст сторінки HTML [12]

Типи XSS:

Reflected XSS – це такі атаки, коли ін’єктований сценарій відображається на веб-сервері, наприклад, у повідомленні про помилку, результатах пошуку або будь-якій іншій відповіді, яка включає частину або всі вхідні дані, надіслані на сервер як частину запиту. Відображені атаки доставляються жертвам іншим шляхом, наприклад, у повідомленні електронної пошти або на іншому веб-сайті. Коли користувача обманом змушують натиснути зловмисне посилання, надіслати спеціально створену форму або навіть просто перейти на зловмисний сайт, введений код переходить на вразливий веб-сайт, який відображає атаку назад у браузер

користувача. Потім браузер виконує код, оскільки він надійшов із «надійного» сервера. Відображений XSS також іноді називають непостійним або XSS типу I (атака здійснюється через один цикл запиту/відповіді) [14].

Stored XSS– це атаки, коли ін’єктований сценарій постійно зберігається на цільових серверах, наприклад у базі даних, у форумі повідомлень, у журналі відвідувачів, у полі коментарів тощо. Потім жертва отримує шкідливий сценарій із сервера, коли він запитує збережений інформації. Збережений XSS також іноді називають постійним XSS або XSS типу II.

Сліпий міжсайтовий скриптинг(або Blind XSS)

Blind XSS є формою постійного XSS. Зазвичай це відбувається, коли корисне навантаження зловмисника зберігається на сервері та повертається жертві з серверної програми. Наприклад, у формах зворотного зв’язку зловмисник може надіслати зловмисне корисне навантаження за допомогою форми, і коли серверний користувач/адміністратор програми відкриє надіслану форму зловмисника через серверну програму, корисне навантаження зловмисника буде виконано. Сліпий міжсайтовий сценарій важко підтвердити в реальному світі, але одним із найкращих інструментів для цього є XSS Hunter [15].

На додаток до збережених і відображених XSS, інший тип XSS, XSS на основі DOM, був ідентифікований Амітом Кляйном у 2005 році . OWASP рекомендує категоризацію XSS, як описано в статті OWASP: Типи міжсайтових сценаріїв , яка охоплює всі ці терміни XSS, організовуючи їх у матрицю збережених і відображених XSS і серверних і клієнтських XSS, де XSS на основі DOM є підмножина XSS клієнта [16].

Наслідки атаки XSS однакові, незалежно від того, збережена вона чи відображена (або на основі DOM). Різниця полягає в тому, як корисне навантаження надходить на сервер. Не обманюйте себе, думаючи, що сайт «лише для читання» або «програмне забезпечення для брошур» не вразливий до серйозних відображених атак XSS. XSS може спричинити різноманітні проблеми для кінцевого користувача, серйозність яких варіюється від роздратування до повної компрометації облікового запису. Найсерйозніші XSS-атаки передбачають розкриття файлу cookie сеансу

користувача, що дозволяє зловмиснику захопити сеанс користувача та заволодіти обліковим записом. Інші шкідливі атаки включають розголошення файлів кінцевого користувача, встановлення троянських програм, перенаправлення користувача на іншу сторінку чи сайт або зміну представлення вмісту. Уразливість XSS, що дозволяє зловмиснику змінити прес-реліз або новину, може вплинути на курс акцій компанії або знизити довіру споживачів. Уразливість XSS на фармацевтичному сайті може дозволити зловмиснику змінити інформацію про дозування, що призведе до передозування.

Вади XSS може бути важко виявити та видалити з веб-програми. Найкращий спосіб виявити недоліки — це перевірити код на безпеку та знайти всі місця, де вхідні дані HTTP-запиту могли б потрапити до виводу HTML. Зауважте, що для передачі шкідливого JavaScript можна використовувати різні теги HTML. Nessus, Nikto та деякі інші доступні інструменти можуть допомогти сканувати веб-сайт на наявність цих недоліків, але можуть лише подряпати поверхню. Якщо одна частина веб-сайту є вразливою, існує велика ймовірність того, що є й інші проблеми.

Ще однією з найпоширеніших атак на хост є Підвищення прав (Privilege Escalation, PE) - це тип атаки, при якій зловмисник використовує вразливість системи для отримання доступу до більших прав, ніж йому було надано спочатку. Це може дозволити зловмиснику виконувати дії, які зазвичай заборонені для звичайних користувачів, такі як доступ до конфіденційних даних, зміна системних налаштувань або навіть повне захоплення контролю над системою [17].

Є два типи підвищення привілеїв:

Горизонтальна ескалація привілеїв — зловмисник розширює свої привілеї, захоплюючи інший обліковий запис і зловживаючи законними привілеями, наданими іншому користувачеві. Щоб дізнатися більше про горизонтальне підвищення привілеїв, перегляньте наш посібник із бокового переміщення .

Вертикальна ескалація привілеїв — зловмисник намагається отримати більше дозволів або доступу до існуючого облікового запису, який він зламав. Наприклад, зловмисник заволодіває звичайним обліковим записом користувача в мережі та

намагається отримати адміністративні дозволи або root-доступ. Це потребує більшої витонченості та може прийняти форму вдосконаленої постійної загрози [18].

Ескалація привілеїв дозволяє зловмисникам відкривати нові вектори атак на цільову систему. Наприклад, це може включати:

Отримання доступу до інших підключених систем

Розгортання додаткових шкідливих корисних навантажень у цільовій системі

Налаштування параметрів безпеки або привілеїв

Отримання доступу до програм або даних у системі за межами привілеїв початкового зламаного облікового запису

У крайніх випадках отримання кореневого доступу до цільової системи або всієї мережі.

Якщо служби безпеки підозрюють підвищення привілеїв, важливо провести поглиблене розслідування. Ознаки підвищення привілеїв включають зловмисне програмне забезпечення в конфіденційних системах, підозрілі входи та незвичні мережеві зв’язки.

Будь-який інцидент із підвищенням привілеїв слід розглядати як серйозний інцидент із безпекою, і, залежно від зобов’язань організації щодо відповідності, про нього, можливо, доведеться повідомити владі.

Атаки на підвищення привілеїв зазвичай передбачають використання вразливостей, таких як помилки програмного забезпечення, неправильні конфігурації та неправильні елементи керування доступом.

Кожен обліковий запис, який взаємодіє з системою, має певні привілеї. Стандартні користувачі зазвичай мають обмежений доступ до системних баз даних, конфіденційних файлів або інших ресурсів. У деяких випадках користувачі мають надмірний доступ до конфіденційних ресурсів і можуть навіть не знати про це, оскільки вони не намагаються отримати доступ поза межами своїх прав. В інших випадках зловмисники можуть маніпулювати слабкими сторонами системи, щоб збільшити привілеї.

Захоплюючи обліковий запис користувача низького рівня та або зловживаючи надмірними привілеями, або збільшуючи привілеї, зловмисник отримує точку входу

до конфіденційної системи. Зловмисники можуть деякий час перебувати в системі, проводячи розвідку та чекаючи можливості поглибити свій доступ. Згодом вони знайдуть спосіб підвищити привілеї до вищого рівня, ніж обліковий запис, який спочатку було зламано.

Залежно від своєї мети, зловмисники можуть продовжувати горизонтально, щоб отримати контроль над додатковими системами, або підвищувати привілеї вертикально, щоб отримати адміністратор і root-контроль, поки вони не отримають доступ до всього повного середовища.

1.3 Загрози інфраструктури вебдодатків на рівні гіпервізора

Гіпервізор - це програмне забезпечення, яке віртуалізує апаратні ресурси комп'ютера, дозволяючи запускати декілька операційних систем на одній машині. Це робить гіпервізори надзвичайно потужними, але й робить їх уразливими до атак. Загрози на рівні гіпервізора - це тип кібератаки, яка використовує вразливості в гіпервізорі для отримання несанкціонованого доступу до віртуальних машин, що на ньому працюють, або до самого гіпервізора[19].

Загрози на рівні гіпервізора можуть бути надзвичайно серйозними, оскільки гіпервізори відповідають за керування апаратними ресурсами та віртуальними машинами, які працюють на них.

Вразливості гіпервізора - це слабкі місця або помилки в програмному забезпеченні, яке віртуалізує апаратні ресурси комп'ютера та управляє віртуальними середовищами. Вони можуть бути використані зловмисниками для отримання несанкціонованого доступу до віртуальних машин або гіпервізора. Ось деякі типи вразливостей гіпервізора:

1. Вразливості у коді гіпервізора: Ці вразливості виникають через помилки в програмному коді гіпервізора, які можуть бути експлуатовані для отримання несанкціонованого доступу або виконання зловмисного коду.
2. Недоліки в системі віртуалізації: Деякі вразливості можуть бути пов'язані зі специфічними реалізаціями системи віртуалізації, такими як помилки в алгоритмах

планування ресурсів, контроль доступу до пам'яті або управління віртуальними пристроями.

3. Вразливості віртуальних пристроїв: Інколи вразливості можуть виникати через недоліки у віртуальних пристроях, які гіпервізор використовує для забезпечення доступу до апаратних ресурсів, таких як мережеві адаптери або диски.

4. Недостатній контроль доступу: Недоліки у механізмах контролю доступу до віртуальних ресурсів можуть призвести до ситуацій, коли зловмисники отримують доступ до даних або функціональності, до яких вони не мають прав доступу.

5. Вразливості у конфігурації: Неправильна конфігурація гіпервізора може створити додаткові ризики безпеки, такі як недостатньо обмежені права доступу або відкриті мережеві порти.

6. Збої у плануванні ресурсів: Недоліки у механізмах планування ресурсів можуть призвести до перевищення обсягів ресурсів або відмови у використанні віртуальних машин, що може призвести до відмови у обслуговуванні або зниження продуктивності.

7. Недостатня ізоляція віртуальних машин: Іноді гіпервізор може бути налаштований таким чином, що дозволяє одній віртуальній машині отримувати доступ до даних або ресурсів іншої віртуальної машини, що призводить до порушення принципу безпеки ізоляції між віртуальними машинами.

8. Вразливості інтерфейсів API: Інтерфейси API, які використовуються для керування гіпервізором і віртуальними машинами, можуть мати вразливості, що дозволяють зловмисникам отримувати доступ до системи або змінювати конфігурації.

9. Атаки на гіпервізор з віртуальної машини: У разі компрометації однієї з віртуальних машин зловмисник може використати вразливості в коді гіпервізора для отримання контролю над усіма іншими віртуальними машинами, що працюють на тому ж гіпервізорі.

10. Вразливості гостьової операційної системи: Недоліки у безпеці гостьової операційної системи можуть дозволити зловмисникам перехопити керування віртуальною машиною, отримуючи таким чином непрямий доступ до гіпервізора.

11. Недостатня безпека каналів зв’язку: Гіпервізори використовують різні канали зв’язку для взаємодії з віртуальними машинами та з іншими компонентами системи. Незахищені або недостатньо захищені канали можуть бути використані зловмисниками для перехоплення або модифікації переданих даних.

12. Вразливості систем моніторингу та керування: Системи моніторингу та керування віртуальними машинами, такі як панелі управління або інструменти керування віртуалізацією, також можуть мати вразливості, що дозволяють зловмисникам здійснювати атаки на гіпервізор.

1.3.1 Вразливості у коді гіпервізора

Ці вразливості виникають через помилки або недоліки в програмному коді гіпервізора. Вони можуть бути використані зловмисниками для отримання несанкціонованого доступу або виконання зловмисного коду. Основні види таких вразливостей:

Переповнення буфера (Buffer Overflow): Виникає, коли гіпервізор неправильно обробляє дані, дозволяючи зловмиснику записати дані за межі буфера і змінити пам'ять гіпервізора. Це може призвести до виконання шкідливого коду [20].

Виконання довільного коду (Arbitrary Code Execution): Деякі помилки в коді можуть дозволити зловмисникам виконувати свій власний код в контексті гіпервізора, отримуючи таким чином контроль над системою.

Виклик функцій за межами дозволених (Privileged Function Calls): Недоліки в перевірці прав доступу можуть дозволити зловмисникам викликати привілейовані функції, що зазвичай доступні лише системним адміністраторам[21].

Недостатня перевірка вхідних даних (Input Validation Flaws): Вразливості виникають через недостатню перевірку даних, що вводяться. Це може призвести до введення спеціально створених зловмисниками даних, які можуть експлуатуватися для атаки на систему.

Витік пам'яті (Memory Leak): Деякі вразливості можуть призвести до витоку конфіденційної інформації з пам'яті гіпервізора.

Атаки на канали зв’язку (Communication Channel Attacks): Недоліки у способах передачі даних між гіпервізором та віртуальними машинами можуть дозволити зловмисникам перехоплювати або змінювати ці дані.

Приклади вразливостей у коді гіпервізора

CVE-2018-3646: Вразливість гіпервізора Xen дозволила виконання коду з віртуальної машини, що призводило до доступу до даних інших віртуальних машин[22].

CVE-2019-5736: Вразливість у Docker та runc (інструмент, що використовується для запуску контейнерів у Docker) дозволяла виконання довільного коду з контейнера, що могло призвести до захоплення контролю над усією системою[23].

CVE-2020-4004 та CVE-2020-4005: Вразливості в VMware ESXi, Workstation та Fusion дозволяли підвищення привілеїв з гостьової віртуальної машини.

Недоліки в системі віртуалізації

Вразливості, пов'язані з недоліками в системі віртуалізації, виникають через помилки або обмеження в реалізації самої технології віртуалізації. Вони можуть призводити до витоку даних, порушення цілісності системи або захоплення контролю над іншими віртуальними машинами. Нижче наведені основні приклади та типи цих вразливостей:

Приклади недоліків у системі віртуалізації.

1. Вразливості керування пам'яттю (Memory Management Vulnerabilities)

Вразливості керування пам'яттю виникають через помилки в обробці або розподілі пам'яті гіпервізором чи гостьовими операційними системами. Це може призвести до витоку даних, ескалації привілеїв або виконання довільного коду.

- Переповнення буфера (Buffer Overflow):

Суть: Переповнення буфера виникає, коли дані записуються за межі виділеного буфера.

Ризики: Це може призвести до пошкодження пам'яті та виконання довільного коду зловмисника.

Приклад: CVE-2018-3665 дозволяла отримувати доступ до даних інших віртуальних машин через переповнення буфера у функції обробки плаваючої точки [24].

- Використання після звільнення (Use-After-Free):

Суть: Ця вразливість виникає, коли пам'ять, яка вже була звільнена, знову використовується.

Ризики: Зловмисник може отримати доступ до вже виділеної пам'яті або навіть виконувати свій код.

Приклад: CVE-2019-19582 у QEMU дозволяла зловмиснику виконувати довільний код на хості [25].

- Доступ за межі виділеної пам'яті (Out-of-Bounds Access):

Суть: Виникає при доступі до пам'яті за межами виділеного сегмента.

Ризики: Може призвести до витоку даних або виконання довільного коду.

Приклад: CVE-2017-5715 (Spectre) дозволяла процесам читати пам'ять інших процесів.

- Двійкове звільнення пам'яті (Double-Free):

Суть: Помилка виникає, коли одна і та ж ділянка пам'яті звільняється двічі.

Ризики: Це може призвести до некоректного доступу до пам'яті або виконання довільного коду.

Приклад: CVE-2020-10713 (BootHole) дозволяла зловмисникам завантажувати довільний код через двійкові посилання.

- Порушення цілісності пам'яті (Memory Corruption):

Суть: Некоректне оновлення або зміна пам'яті, що призводить до порушення логіки програми.

Ризики: Це може дозволити зловмисникам виконувати довільний код або порушити роботу системи.

Приклад: Вразливість CVE-2018-3646 дозволяла зловмисникам впливати на кеш-пам'ять процесора[26].

- Цілочисельне переповнення (Integer Overflow):

Суть: Переповнення значення змінної цілого типу.

Ризики: Це може призвести до виділення невідповідної кількості пам'яті та виконання довільного коду.

Приклад: CVE-2017-14340 у Xen дозволяла ескалацію привілеїв через цілочисельне переповнення.

- Незачищена пам'ять (Uninitialized Memory):

Суть: Використання пам'яті без попередньої ініціалізації.

Ризики: Може призвести до витоку даних або виконання довільного коду.

Приклад: CVE-2019-15220 дозволяла зловмисникам використовувати незачищену пам'ять для виконання довільного коду.

2. Вразливості системи планування ресурсів (Resource Scheduling Flaws):

- Захоплення ресурсів (Resource Contention): Одній віртуальній машині дозволяється захопити більше ресурсів, ніж належить, що може призвести до відмови в обслуговуванні для інших машин.
- Зловживання пріоритетами: Неправильне призначення пріоритетів дозволяє зловмисникам використовувати більше процесорного часу або інших ресурсів.

3. Вразливості механізмів ізоляції (Isolation Mechanisms Vulnerabilities):

- Перехресний доступ до пам'яті: Неправильне управління доступом до пам'яті дозволяє одній віртуальній машині отримувати доступ до пам'яті інших віртуальних машин або гіпервізора.
- Недостатня ізоляція між віртуальними машинами: Віртуальні машини можуть взаємодіяти одна з одною, порушуючи принцип ізоляції.

4. Логічні помилки у віртуальних мережах (Virtual Network Logical Flaws):

- Спуфінг (Spoofing): Недоліки у віртуальних мережах можуть дозволити зловмиснику підробити IP-адресу або MAC-адресу.
- Витік мережевих пакетів (Packet Leakage): Зловмисник може отримати доступ до мережевих пакетів, призначених для інших віртуальних машин.

5. Вразливості віртуальних пристроїв (Virtual Device Vulnerabilities):

- Віртуальні диски (Virtual Disks): Помилки у реалізації віртуальних дисків можуть дозволити зловмиснику отримати доступ до даних інших віртуальних машин.

- Віртуальні мережеві адаптери (Virtual NICs): Атаки на віртуальні мережеві адаптери можуть дозволити перехоплення мережевого трафіку.

6. Вразливості інтерфейсів API (API Interfaces Vulnerabilities):

- Недостатня автентифікація: API для керування гіпервізором або віртуальними машинами може бути скомпрометований через неправильну автентифікацію.
- Ескалація привілеїв: Недоліки у політиках доступу до API можуть дозволити зловмиснику підвищити свої привілеї.

7. Недоліки у механізмах міграції віртуальних машин (Migration Mechanisms Flaws):

- Витік даних: Зловмисники можуть перехоплювати або змінювати дані під час міграції.
- Обхід політик безпеки: Помилки в механізмах міграції дозволяють обійти політики безпеки під час переміщення віртуальної машини.

1.3.2 Вразливості віртуальних пристроїв

Віртуальні пристрої (мережеві адаптери, диски, USB-пристрої тощо) забезпечують віртуальним машинам доступ до фізичних ресурсів хоста або інших віртуальних пристроїв. Їх помилкова конфігурація або вразливості в реалізації можуть призвести до витоку даних, порушення ізоляції між віртуальними машинами або навіть до компрометації самого гіпервізора.

Основні типи вразливостей віртуальних пристроїв

- Вразливості віртуальних мережевих адаптерів (vNIC)
- Вразливості віртуальних блочних пристроїв (Virtual Block Devices)
- Вразливості у механізмах спільного використання пристроїв (Device Sharing Mechanisms)
- Вразливості у віртуальних графічних пристроях (Virtual Graphics Devices)
- Вразливості у віртуальних звукових пристроях (Virtual Sound Devices)

Вразливості віртуальних мережевих адаптерів (vNIC):

Віртуальні мережеві адаптери (Virtual Network Interface Card, vNIC) забезпечують зв'язок між віртуальними машинами та фізичними мережами або між самими віртуальними машинами. Вразливості у віртуальних мережевих адаптерах можуть призвести до витоку даних, обходу політик контролю доступу, виконання довільного коду та інших проблем.

1. Неправильна обробка мережевих пакетів:

Віртуальні мережеві адаптери отримують і передають мережеві пакети між віртуальними машинами та фізичною мережею. Неправильна обробка мережевих пакетів може виникати через помилки у реалізації мережевих драйверів або протоколів, що призводить до вразливостей, які дозволяють зловмисникам перехоплювати, змінювати або виконувати довільний код.

Одним із поширених типів вразливостей є переповнення буфера, коли мережевий драйвер неправильно обробляє отримані пакети, дозволяючи зловмиснику записувати дані за межі виділеного буфера. Наприклад, у VMware Workstation вразливість CVE-2019-15098 дозволяла зловмиснику надсилати спеціально сформовані пакети на віртуальний мережевий адаптер, що призводило до виконання довільного коду на хості через переповнення буфера[27].

Ще один приклад неправильної обробки мережевих пакетів — відсутність належної фільтрації трафіку. Деякі віртуальні мережеві адаптери можуть не обмежувати обробку небезпечних пакетів, що дозволяє зловмиснику обходити політики контролю доступу. Наприклад, у VMware vSphere була виявлена вразливість (CVE-2020-4004), яка дозволяла зловмисникам надсилати пакети на віртуальні мережеві адаптери віртуальних машин, обходячи політики мережевої фільтрації [28].

2. Зловживання протоколом ARP (ARP Poisoning): Дозволяє підробляти ARP-запити та відповіді, що дозволяє перехоплювати та змінювати мережевий трафік. У Xen або VMware це дозволяє зловмиснику здійснити атаку "людина посередині", підробляючи ARP-відповіді для перехоплення трафіку між віртуальними машинами.

3. Спуфінг MAC-адрес (MAC Address Spoofing)

Віртуальні мережеві адаптери (vNIC) мають унікальні MAC-адреси, які використовуються для ідентифікації кожної віртуальної машини в мережі. MAC-адреса — це апаратна адреса мережевого адаптера, що зазвичай призначається під час налаштування мережевого інтерфейсу та повинна бути унікальною для кожного адаптера.

Зловмисник може підробити MAC-адресу іншої віртуальної машини або фізичного мережевого пристрою, що дозволяє йому:

- Перехоплювати мережевий трафік: Підробивши MAC-адресу іншої машини, зловмисник може перехопити трафік, призначений для цієї машини.
- Обійти політики контролю доступу: Якщо політики мережевого доступу базуються на MAC-адресах, зловмисник може використовувати підроблену адресу для обходу обмежень доступу.
- Видавати себе за іншу машину: Зловмисник може видавати себе за іншу віртуальну машину або сервер, обманюючи користувачів і служби.

Уразливість CVE-2017-4902 у VMware ESXi дозволяла виконувати MAC-спуфінг для перехоплення мережевого трафіку. Ця вразливість виникала через відсутність належних механізмів ізоляції та перевірки MAC-адрес між віртуальними машинами. Зловмисник, який мав доступ до однієї віртуальної машини, міг підробити MAC-адресу іншої віртуальної машини, щоб отримати доступ до її мережевого трафіку.

Зловмисник запускає команду для зміни своєї MAC-адреси на адресу жертви(рис. 1)

```
ifconfig eth0 hw ether 00:1A:2B:3C:4D:5E
```

Рисунок 1.1 - Запуск команди для зміни MAC адреси

Потім зловмисник починає прослуховувати мережевий трафік за допомогою інструментів типу tcpdump або Wireshark.

Мережевий трафік, призначений для жертви, надходить до мережевого інтерфейсу зловмисника, дозволяючи йому перехоплювати або модифікувати дані.\

1.3.3 Вразливості у конфігурації гіпервізора

Вразливості у конфігурації гіпервізора можуть виникати через неправильні налаштування або недоліки у конфігураційних параметрах, які використовуються для управління та налаштування гіпервізора і віртуальних середовищ, які він обслуговує.

Недостатньо обмежені права доступу гіпервізора

Недостатньо обмежені права доступу гіпервізора означають, що деякі користувачі або процеси мають більше привілеїв або доступу до ресурсів, ніж необхідно для виконання їхніх обов'язків або завдань. Це може створити серйозні ризики безпеки, оскільки зловмисники можуть використовувати ці недоліки, щоб отримати доступ до гіпервізора та віртуальних машин.

Ось деякі причини та наслідки недостатньо обмежених прав доступу гіпервізора:

Недостатньо обмежені права адміністратора: Якщо адміністратор має надмірні привілеї, це може призвести до того, що він може виконувати небезпечні операції, які можуть пошкодити гіпервізор або віртуальні машини. Наприклад, неправильне конфігурування ресурсів або налаштування мережевих параметрів може викликати відмову в обслуговуванні або збій безпеки.

Недостатньо обмежені права віртуальних машин: Якщо віртуальні машини мають надмірні привілеї або доступ до ресурсів, це може призвести до того, що вони можуть отримати доступ до конфіденційної інформації або взяти під контроль інші віртуальні машини на тому ж гіпервізорі.

Використання слабких паролів: Якщо паролі для доступу до гіпервізора або адміністративних інтерфейсів є слабкими або не захищеними, зловмисники можуть легко використовувати їх для отримання несанкціонованого доступу.

Неправильна ідентифікація та аутентифікація: Недоліки у механізмах ідентифікації та аутентифікації можуть призвести до того, що зловмисники можуть

підробити облікові записи або отримати доступ до них шляхом перехоплення облікових даних.

Відсутність аудиту та моніторингу: Якщо не здійснюється аудит та моніторинг дій користувачів або процесів у гіпервізорі, зловмисники можуть використовувати недоліки у правах доступу без виявлення.

Методи захисту від недостатньо обмежених прав доступу:

- Впровадження принципу найменших привілеїв: Надайте користувачам і ВМ лише ті привілеї, які їм дійсно потрібні для виконання їхніх завдань.
- Регулярний перегляд прав доступу: Перевіряйте права доступу користувачів і ВМ регулярно, щоб переконатися, що вони не є занадто широкими.
- Використання інструментів управління доступом на основі ролей (RBAC): RBAC може допомогти автоматизувати процес надання та управління правами доступу.
- Шифрування даних може допомогти захистити їх від несанкціонованого доступу, навіть якщо зловмиснику вдасться отримати доступ до ВМ.
- Оновленнями безпеки: Регулярно оновлюйте програмне забезпечення гіпервізора та ВМ, щоб усунути відомі вразливості.

Відкриті мережеві порти

Проблема відкритих мережевих портів полягає в тому, що вони можуть стати вразливими до атак з боку зловмисників, які можуть використовувати ці порти для незаконного доступу або атак на пристрій або мережу. Ось деякі ризики, пов'язані з відкритими мережевими портами:

Експлуатація вразливостей програмного забезпечення: Якщо програма або служба, яка слухає на відкритому порті, має вразливість, зловмисники можуть експлуатувати цю вразливість для отримання доступу до системи або виконання шкідливого коду.

Атаки з переповненням буфера: Зловмисники можуть надсилати спеціально сформовані запити на відкриті порти, щоб спричинити переповнення буфера в програмах або службах, які їх обробляють. Це може призвести до виконання шкідливого коду або відмови в обслуговуванні.

- Злам паролів: Зловмисники можуть намагатися отримати доступ до системи, надсилаючи запити на відкриті порти з метою зламу облікових даних або перебору паролів.
- Внутрішні атаки: Якщо відкриті порти дозволяють зовнішній доступ до внутрішньої мережі, зловмисники можуть використовувати їх для впровадження атак зсередини мережі.
- Витік конфіденційної інформації: Якщо відкриті порти дозволяють доступ до конфіденційної інформації або баз даних, зловмисники можуть використовувати ці порти для витоку чутливої інформації.

Неправильна конфігурація механізмів резервного копіювання може створити серйозні загрози для безпеки даних і вразливість для втрати цих даних. Ось деякі з проблем, які можуть виникнути через неправильну конфігурацію механізмів резервного копіювання:

- Недостатня частота створення резервних копій: Якщо резервні копії створюються недостатньо часто або не включають усю важливу інформацію, то в разі виникнення проблеми (наприклад, випадкового видалення даних або збою в обладнанні) може бути втрачено значну кількість даних.
- Відсутність тестування відновлення: Якщо не проводиться періодичне тестування процедур відновлення з резервних копій, то може статися так, що у випадку кризової ситуації механізм відновлення буде непрацездатним або даватиме неповні або пошкоджені дані.
- Недостатня безпека копій: Якщо резервні копії зберігаються в непридатних місцях або не зашифровані, то це може призвести до неправомірного доступу до конфіденційної інформації.
- Недостатня реплікація: Якщо дані резервних копій не реплікуються на достатньо віддалені або фізично відокремлені місця, то в разі природної катастрофи або іншої серйозної події, яка може вплинути на всі дані в цьому регіоні, резервні копії також можуть бути втрачені.

Недостатнє шифрування даних

Гіпервізори та віртуальні машини часто передають чутливі дані між собою, а також до зовнішніх систем або сховищ. Однак, якщо такі дані не зашифровані належним чином, зловмисники можуть перехопити їх та використати у власних цілях.

Віртуальні середовища зазвичай містять конфіденційні дані, які можуть передаватися через мережу або зберігатися у віртуальних машинах. Недостатнє шифрування таких даних дозволяє зловмисникам, які можуть перехопити мережевий трафік або отримати доступ до сховищ, прочитати їх і скористатися ними. Це створює критичний ризик для безпеки та конфіденційності інформації.

Перехоплення даних під час Live Migration віртуальних машин:

Live Migration віртуальні машини передають свої дані через мережу до іншого хосту гіпервізора. Якщо дані не зашифровані належним чином, зловмисник може перехопити їх і отримати повний знімок пам'яті віртуальної машини, включаючи дані користувачів і облікові дані.

Уразливість у Xen Hypervisor дозволяла перехоплювати дані під час живої міграції через відсутність шифрування. Відкритий канал передачі міг бути використаний зловмисником для отримання доступу до всієї пам'яті віртуальної машини.

Відсутність шифрування між доменом гіпервізора та віртуальними машинами

Віртуальні машини можуть передавати конфіденційні дані між собою та доменом керування (dom0). Якщо такі дані передаються у відкритому вигляді, їх можна легко перехопити та використати.

У Xen Hypervisor передача даних між доменом керування (dom0) та віртуальними машинами (domU) здійснювалася без шифрування. Зловмисник, який отримав доступ до мережевого трафіку, міг перехоплювати конфіденційні дані.

Зберігання віртуальних дисків без шифрування:

Віртуальні диски містять конфіденційні дані, включаючи облікові записи користувачів, особисту інформацію та бізнес-дані. Якщо віртуальні диски не зашифровані, зловмисники можуть легко отримати доступ до цих даних, заволодівши файлом віртуального диска.

Уразливість у VMware vSphere дозволяла отримати доступ до незашифрованих файлів віртуальних дисків через відсутність належного шифрування. Зловмисник, який отримав доступ до файлів, міг відновити конфіденційні дані.

Недостатнє шифрування даних є однією з головних вразливостей, яка може призвести до витоку критичної інформації віртуальних машин або гіпервізора, і потребує особливої уваги для забезпечення безпеки.

Відсутність сегментації мережі для міграції:

Мережевий трафік міграції не ізольований від іншого мережевого трафіку. Зловмисники можуть прослуховувати мережевий трафік або здійснювати атаки на механізм міграції. Відсутність сегментації мережі для трафіку міграції у VMware vSphere дозволяла зловмисникам прослуховувати мережевий трафік або впроваджувати шкідливий код.

Використання застарілих протоколів передачі:

Механізми міграції використовують застарілі протоколи передачі без належного шифрування. Зловмисники можуть перехопити дані або впровадити шкідливий код через недоліки у протоколах передачі. Механізм міграції KVM використовував застарілий протокол передачі VNC без шифрування, що дозволяло зловмисникам перехоплювати дані віртуальних машин.

1.4 Загрози інфраструктури веб додатків на рівні хмари

Вебдодатки, розгорнуті в хмарній інфраструктурі, забезпечують масштабованість та гнучкість, але також стикаються з новими ризиками та загрозами. Хмарна архітектура змінює спосіб обробки та зберігання даних, що призводить до появи нових вразливостей та загроз.

Розміщення вебдодатків у хмарі відкриває нові можливості для бізнесу завдяки спрощенню інфраструктури та ефективному використанню ресурсів. За допомогою хмарних сервісів компанії можуть швидко розгортати та масштабувати свої додатки, зменшуючи витрати на фізичну інфраструктуру. Мультиорендна архітектура хмари

дозволяє використовувати загальні фізичні ресурси кільком користувачам одночасно, створюючи динамічне середовище для зберігання та обробки даних.

Однак такий рівень гнучкості та масштабованості має свою ціну. Віртуалізація, автоматизація та доступність у будь-який час та з будь-якого місця роблять хмару привабливою ціллю для зловмисників. Відсутність належної ізоляції між клієнтами, неправильні конфігурації хмарних ресурсів, вразливості вебдодатків та соціальна інженерія можуть призвести до серйозних наслідків, включаючи витік конфіденційних даних, зловживання ресурсами та навіть повну компрометацію хмарної інфраструктури.

Управління доступом стає однією з найважливіших задач у хмарному середовищі. Використання ключів API, облікових даних із високими привілеями та недостатньо захищених механізмів аутентифікації створює критичні ризики, оскільки компрометація одного облікового запису може надати зловмисникам доступ до всієї хмарної інфраструктури. Це особливо актуально в контексті механізмів розгортання, таких як інфраструктура як код (Infrastructure as Code, IaC), де неправильно налаштовані шаблони можуть автоматично створювати вразливі середовища [29].

Щоб ефективно захистити інфраструктуру вебдодатків на рівні хмари, організаціям необхідно розробити комплексну стратегію безпеки, яка включає правильну конфігурацію ресурсів, контроль доступу та моніторинг активності. Інструменти безпеки мають бути інтегровані в процеси DevOps, щоб виявляти та усувати вразливості ще на етапі розробки. Лише поєднання технологічних заходів та навчання персоналу дозволить організаціям використовувати переваги хмарних сервісів без шкоди для безпеки даних та інфраструктури.

1.4.1 Неправильна конфігурація хмарних ресурсів:

Ця загроза є найпоширенішою саме через людський фактор, Хмарна інфраструктура надає безліч можливостей для налаштування та контролю доступу, але недбале налаштування або невірне розуміння цих можливостей може призвести до серйозних вразливостей. Банально неправильна конфігурація мережевих політик,

облікових записів або сховищ може призвести до несанкціонованого доступу чи витоку даних. Наприклад, відкриті S3-бакети в AWS часто стають причиною витоку конфіденційної інформації.

Відкриті S3-бакети або інші хмарні сховища:

S3-бакети (об'єктні сховища) та інші хмарні сховища (наприклад, Azure Blob Storage, Google Cloud Storage) є популярними сервісами для зберігання даних у хмарі. Однак неправильна конфігурація доступу до таких сховищ може призвести до витоку конфіденційної інформації та інших критично важливих даних.

S3-бакети та інші сховища можуть мати кілька рівнів доступу, від повністю публічного до приватного. Якщо сховище відкрито для публічного доступу, будь-яка людина з інтернету може отримати доступ до його вмісту, включаючи конфіденційні дані.

Відкриті S3-бакети або інші сховища з публічним доступом дозволяють будь-кому читати або записувати дані.

Неправильні політики доступу:

Політики доступу, налаштовані без обмежень, дозволяють несанкціонованим користувачам переглядати, редагувати або видаляти дані. Accenture (2017): Десятки гігабайт конфіденційних даних компанії були викриті через неправильно налаштовані S3-бакети.

Приклади витоків даних через відкриті S3-бакети та інші сховища

Verizon (2017):

Були викриті конфіденційні дані 14 мільйонів клієнтів через відкритий S3-бакет. [30]

Booz Allen Hamilton (2017):

У відкритому S3-бакеті було знайдено 60 000 файлів з конфіденційними даними уряду США [31].

GoDaddy (2018):

Сотні тисяч документів, включаючи конфіденційні дані, були викриті через відкритий S3-бакет [32].

Відкриті RDP/SSH-порти:

Відкриті порти для віддаленого підключення, такі як RDP (Remote Desktop Protocol) або SSH (Secure Shell), забезпечують доступ до серверів та віртуальних машин для адміністраторів та інженерів. Однак у хмарних інфраструктурах вони часто стають мішенню для зловмисників через неправильну конфігурацію або відсутність належних заходів безпеки, що дозволяє використовувати наступні атаки для отримання доступу до віддалених машин:

Експлуатація вразливостей протоколів RDP та SSH можуть містити вразливості, які дозволяють виконувати довільний код або підвищувати привілеї [33].

Зловмисники використовують відомі вразливості для отримання доступу або виконання коду.

Системи з незастосованими патчами стають вразливими.

BlueKeep (CVE-2019-0708): Вразливість у протоколі RDP дозволяла зловмисникам віддалено виконувати код на незахищених системах Windows через незахищений порт RDP [34].

Відсутність двофакторної автентифікації або механізмів обмеження доступу дозволяє зловмисникам використовувати вразливі облікові записи. Відсутність двофакторної автентифікації спрощує отримання доступу через компрометовані облікові дані. Можливість використання одного облікового запису для доступу до кількох ресурсів.

Відсутність мережевих обмежень доступу до портів дозволяє зловмисникам сканувати мережу та виявляти відкриті порти. Відсутність списків контролю доступу (ACL) або мережевої сегментації та відкриті порти для будь-якого IP-адреса у світі є хорошою здобиччю для зловмисників.

1.4.2 Атаки на механізми аутентифікації та авторизації

Механізми аутентифікації та авторизації є ключовими елементами безпеки у хмарних інфраструктурах, оскільки вони забезпечують перевірку та контроль доступу до ресурсів. Зловмисники часто націлюються на ці механізми, щоб отримати несанкціонований доступ до хмарних сервісів або викрасти конфіденційні дані.

По класиці користувачі використовують слабкі паролі та відсутність 2FA, що є Критичним для будь якої хмарної інфраструктури, крім цього існують інші небезпеки пов’язані з аутенитифікацією та авторизацією:

Зловживання ключами API ключі API можуть бути використані зловмисниками для виконання операцій з високими привілеями у хмарних сервісах. Компрометація ключів API у сервісах, наприклад AWS Lambda дозволяє зловмисникам виконувати довільні функції або отримувати доступ до конфіденційних даних.

Недостатній контроль строку дії токенів або неправильно налаштовані механізми відкликання дозволяють зловмисникам використовувати скомпрометовані токени протягом тривалого часу.

Вразливості SAML:

Security Assertion Markup Language (SAML) — це структура на основі XML, яка відіграє ключову роль у забезпеченні безпечної ідентифікації та керування доступом. Він діє як надійний посередник між різними суб’єктами цифрової екосистеми, такими як постачальники ідентифікаційної інформації, постачальники послуг і користувачі. Основною метою SAML є сприяння єдиному входу (SSO), безперебійному та ефективному процесу автентифікації, у якому користувач може отримати доступ до кількох програм і служб за допомогою єдиного набору облікових даних [35]. Хоча SAML пропонує багато переваг порівняно з традиційною автентифікацією за іменем користувача та паролем, уразливості все одно можуть виникати залежно від бібліотеки SAML, яку використовує розробник та налаштування SAML, які використовує розробник.

Однією з найпоширеніших вразливостей SAML є відсутність перевірки підпису та підписання повідомлень.

Перевірку підпису XML можна використовувати для запобігання атакам підробки XML. Він підтверджує автентичність XML-повідомлень у твердженнях SAML. Обгортка XML-підпису — це вразливість, яка використовує слабкі місця в способі обробки XML-підписів і може призвести до зловмисного маніпулювання даними. Під час автентифікації як звичайного користувача в профілі користувача після автентифікації відображається така інформація.

Рисунок 1.2 - Вразливий додаток SAML

Можна підвищити привілеї цього користувача, змінивши запит SAML і змінивши членство в групі на адміністратора.

```
lVTMQ8wDQYDVQQIDAZLYW5zYXMxEDAOBgNVBAcMB1dpY2hpdGExITAfBgNVBAoMGEludGVybmV0IFdpZGdpdHMgUHR5IEx0ZDAeFw0xODAxMTAxNzAzMThaFw0yODAxM
sexdicrUnRbgszkQfgp5Rx+OUyu6WKWCyl8swrz0CWLI16uA5EdSi5evQkJnzVI/e9uvOUDWc/zwSfiEA1ZnwWtW3tr/F09WTvg/6zInqh2TxIwk3uKxyu7HMFekcu2t
0S/baWs8eulYsl8nDQWx0W96DS0pw/jskDxsyIw4rMpqzYfR1XBI2lcALqIjaucDPAZMI9OufoHCSgmUPD1h4g5oIZn/27SkhWqi/hfnBBTc4otzB0h+9q6FhU=</ds:
aml:Issuer>http://127.0.0.1/simplesamlphp/saml2/idp/metadata.php</saml:Issuer><ds:Signature xmlns:ds="http://www.w3.org/2000/09/
//www.w3.org/2001/10/xml-exc-c14n#"/></ds:Transforms><ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/><ds:Di
EiQqPwKnpeLGgKn7YMOIhKLdAgYJjBIZ1/28GXd06t/2mtEzFoMXPCJ8glJe1eXdShCY45pJJ3g==</ds:SignatureValue><ds:KeyInfo><ds:X509Data><ds:X5
oCggEBAOysjXhxnPnXGheeidZphc8PurUT+ToxJBswCk3/uY3lPeAGLaS2a0XvQw9UUJ7qoZ8BiSlegIVD/940E+5T0EPmBJrmJB3VQCEF/tHdo+WCL0JTg1paJCnkS1
PGUbeclFg+ZvfclSCqVpypzy2nJMWC5S8IyOFjo+BQbbry1bR2sKZ9OwIReFDh0qSOh+dzNSlKvfwj5B/phocmzi9UTAlatex02x/QYO3A0iDDj1Qf3cJQj2QJQ/Lqk+
="http://127.0.0.1:8000/metadata/">_647ce0954da7ca55fff2f1b096ad75d561ae217bb7</saml:NameID><saml:SubjectConfirmation Method="ur
ence>http://127.0.0.1:8000/metadata/</saml:Audience></saml:AudienceRestriction></saml:Conditions><saml:AuthnStatement AuthnInsta
mes:tc:SAML:2.0:attrname-format:uri"><saml:AttributeValue xsi:type="xs:string">administrators</saml:AttributeValue></saml:Attrib
bute Name="username" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"><saml:AttributeValue xsi:type="xs:string">yogi
```


Рисунок 1.3 - Запит SAML

На рисунку 1.4 зображено що в додатку після скомпроментованого запиту SAML права було підвищенно до адміністратора

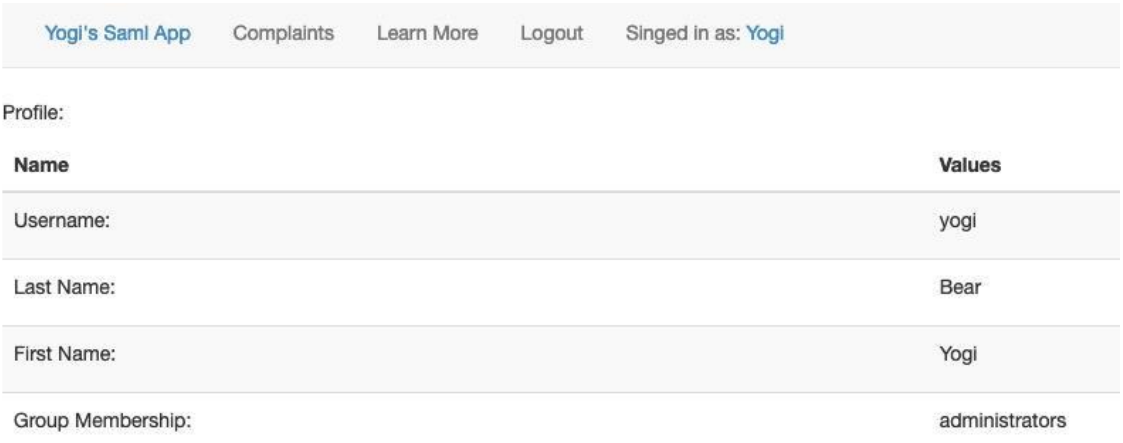

Рисунок 1.4 Підвищення привілеїв

Відкрите перенаправлення

SAML за своєю суттю є асинхронним протоколом. Коли починається процес входу, ініційований сервіс провайдером (SP), він ініціюється створенням запиту автентифікації SAML, який потім направляється до Identity Provider (IdP). Важливо те, що на цьому етапі Постачальник послуг не зберігає жодної інформації щодо запиту. Отже, коли відповідь SAML отримана від IdP, Постачальник послуг не помічає оригінального глибокого посилання, яке викликало запит на автентифікацію. SAML вирішує цю проблему за допомогою включення важливого параметра, відомого як «RelayState».

RelayState служить як параметр HTTP і може бути включений як у запит SAML, так і в відповідь SAML. У сценаріях, коли виконується потік входу, ініційований SP, SP може використовувати параметр RelayState для вбудовування додаткової інформації в запит SAML. Цей механізм гарантує, що важливий контекст і дані не будуть втрачені під час асинхронного процесу [36]. Однак RelayState зловмисник може зловживати цим параметром для проведення відкритих атак перенаправлення

Висновки до розділу 1

У першому розділі було розглянуто загрози вебдодатків на різних рівнях інфраструктури, включаючи мережевий рівень, рівень хоста, гіпервізора та хмари.

Кожен рівень має власні вразливості та ризики, які можуть призвести до серйозних компрометацій та порушенню конфіденційності, цілісності та доступності користувачів вебдодатків. Було розглянуто найпоширеніші рівні на кожному рівні інфраструктури вебдодатків, такі як:

- Атаки типу DDoS
- DNS-спуфінг та атаки на DNS
- IP-спуфінг
- APR poisoning
- Експлуатація вразливостей вебдодатків, такі як RCE, XSS та SQLi
- Атаки пов’язані з неправильною конфігурацією гіпервізора та хмарних сервісів
- Відстуність шифрування

Розглянувши та проаналізувавши зазначені небезпеки було розроблено стратегію побудови захищеної інфраструктури вебдодатку, які будуть описані в наступих розділах

РОЗДІЛ 2
ЗАСОБИ ТА МЕХАНІЗМИ ЗАХИСТУ ІНФРАСТРУКТУРИ

2.1 Загальні методи захисту інфраструктури вебдодатку

Захист інфраструктури вебдодатків є ключовим завданням для будь-якої організації, яка використовує вебдодатки для надання послуг або обробки конфіденційних даних. Оскільки вебдодатки стають все більш критичною частиною бізнес-процесів, кіберзлочинці активно шукають уразливості на різних рівнях архітектури, включаючи мережевий рівень, рівень хоста, гіпервізора та хмарної інфраструктури. Тому комплексний підхід до захисту вебдодатків є необхідним для ефективної протидії сучасним загрозам та забезпечення безпеки даних.

На мережевому рівні вебдодатки стикаються з численними атаками, такими як DDoS-атаки, перехоплення трафіку (атаки типу "людина посередині"), DNS-спуфінг та атаки на протоколи шифрування. DDoS-атаки можуть вивести вебдодатки з ладу, перевантажуючи їх величезною кількістю запитів, тоді як перехоплення трафіку дозволяє зловмисникам отримувати конфіденційну інформацію під час її передачі. Вразливості DNS-системи можуть використовуватися для спрямування користувачів на фальшиві веб-сайти або вебдодатки, а атаки на протоколи шифрування дозволяють обходити механізми захисту та перехоплювати трафік.

На рівні хоста вебдодатки стикаються з такими загрозами, як експлуатація вразливостей операційних систем або програмного забезпечення, атаки на облікові записи користувачів та впровадження шкідливого програмного забезпечення. Відсутність своєчасних оновлень ОС або програмного забезпечення дозволяє зловмисникам експлуатувати відомі вразливості для отримання несанкціонованого доступу або виконання довільного коду. Використання слабких паролів або відсутність багатофакторної автентифікації підвищує ризик компрометації облікових записів користувачів. Злом облікових записів дозволяє зловмисникам отримувати

підвищені привілеї та впроваджувати шкідливе ПЗ, яке може призвести до витоку даних або повної компрометації систем.

Рівень гіпервізора є особливо критичним, оскільки компрометація гіпервізора може призвести до отримання повного контролю над усіма віртуальними машинами. Загрози на цьому рівні включають вразливості у коді гіпервізора, недоліки системи віртуалізації, вразливості віртуальних пристроїв та неправильну конфігурацію гіпервізора. Помилки в програмному коді гіпервізора можуть дозволити зловмисникам виконувати довільний код або отримувати доступ до пам'яті інших віртуальних машин. Недоліки у плануванні ресурсів, контролі доступу до пам'яті або керуванні віртуальними пристроями можуть призвести до витоку даних або втрати контролю над гіпервізором. Неправильна конфігурація гіпервізора, така як відкриті мережеві порти або надмірні привілеї облікових записів адміністраторів, також підвищує ризик компрометації.

Хмарна інфраструктура, яка надає вебдодаткам масштабованість та гнучкість, також стикається з численними загрозами. Основні ризики включають неправильну конфігурацію хмарних ресурсів, зловживання ключами API та обліковими записами з високими привілеями, а також атаки на механізми аутентифікації та авторизації. Неправильна конфігурація хмарних ресурсів, наприклад відкриті S3-бакети або бази даних, може призвести до витоку конфіденційних даних. Компрометація ключів API або облікових записів з високими привілеями дозволяє зловмисникам отримувати доступ до критично важливих даних або керувати хмарною інфраструктурою. Вебдодатки та API, розгорнуті у хмарі, можуть мати вразливості, які зловмисники використовують для компрометації даних або керування інфраструктурою.

Комплексний підхід до захисту інфраструктури вебдодатків включає забезпечення безпеки на всіх рівнях архітектури. Лише таким чином можна ефективно протистояти сучасним загрозам та забезпечити надійний захист конфіденційних даних і цілісність інфраструктури.

Забезпечення захисту вебдодатків потребує впровадження різноманітних методів безпеки на всіх рівнях архітектури. Найпоширенішими практиками захисту інфраструктури вебдодатків є:

2.1.1 Контроль доступу та аутентифікація

Як було описано в попередньому розділі, через недостатність контролю доступу та проблем з аутентифікацією, пов’язана велика кількість загроз.

Контроль доступу та аутентифікація є першою лінією захисту інфраструктури вебдодатків. Їхнє ефективне впровадження допомагає забезпечити лише авторизованим користувачам та службам доступ до критичних ресурсів. Нижче наведено основні методи та практики в цій галузі:

Багатофакторна автентифікація (MFA):

Використання додаткових факторів автентифікації (наприклад, паролі разом із біометричними даними або одноразовими кодами) для підтвердження особи.

Впровадження MFA для всіх облікових записів з високими привілеями (адміністратори, розробники, менеджери) допоможе зберегти критично важливі дані. Використовуйте мобільні додатки або апаратні токени для генерування одноразових кодів.

Основні аспекти принципу найменших привілеїв:

Принцип найменших привілеїв (Least Privilege) передбачає надання користувачам, службам та програмам лише тих прав доступу, які необхідні для виконання їхніх конкретних завдань. Це обмежує потенційний обсяг шкоди у випадку компрометації облікового запису або служби та зменшує ризик несанкціонованого доступу до конфіденційних даних та критичних систем [37].

Обмеження доступу на основі ролей (Role-Based Access Control, RBAC). У разі контролю доступу на основі ролей, права надаються відповідно до ролі користувача або служби. Зазвичай визначаються різні ролі з високими та низькими привілеями, і користувачам призначаються ролі, які відповідають їхнім обов'язкам. Регулярний перегляд ролей та відповідних їм прав доступу дозволяє забезпечити відповідність ролі її обов'язкам та мінімізувати ризик надмірних привілеїв [38].

Обмеження доступу на основі атрибутів (Attribute-Based Access Control, ABAC) . Доступ на основі атрибутів контролюється за допомогою набору атрибутів

користувача або служби, таких як місце розташування, пристрій або відділ. Політики доступу налаштовуються на основі цих атрибутів, що дозволяє надавати більш точні дозволи на доступ. Регулярний перегляд атрибутів забезпечує актуальність контролю доступу [39].

Привілеї для облікових записів користувачів. Користувачам надаються лише ті привілеї, які необхідні для виконання їхніх завдань. Облікові записи поділяються на користувацькі та адміністративні, причому користувачам зазвичай призначаються мінімальні привілеї. Невикористовувані облікові записи деактивуються або видаляються, а тимчасовим обліковим записам надаються обмежені права доступу.

Привілеї для служб та програм. Служби та програми отримують мінімальні привілеї для виконання їхніх функцій. Кожній службі призначається окремий обліковий запис із мінімальними привілеями. Доступ служб до системних файлів та конфіденційних даних забороняється, якщо він не є необхідним. Регулярний перегляд привілеїв служб та програм дозволяє підтримувати їх відповідність завданням.

Обмеження строку дії доступу (Just-in-Time Access). Привілеї надаються лише на обмежений час або для виконання певних завдань. Тимчасові облікові записи або підвищення привілеїв використовуються для конкретних завдань, а доступ до критичних ресурсів надається лише на обмежений період часу. Привілеї видаляються або знижуються одразу після завершення завдання[40].

Аудит привілеїв та прав доступу. Регулярний перегляд та перевірка прав доступу користувачів та служб є ключовим елементом принципу найменших привілеїв. Відповідність прав доступу ролям та завданням користувачів перевіряється шляхом автоматизованих процесів аудиту. Невідповідності або надмірні привілеї виявляються та виправляються.

Обмеження доступу до ключів API та секретів. Ключі API та інші секрети отримують мінімальні привілеї для виконання певних завдань. Політики доступу для кожного ключа API створюються відповідно до принципу найменших привілеїв, причому використовуються різні ключі API для різних служб та програм. Ключі API регулярно обмежуються та оновлюються.

Управління обліковими записами (Identity Management) включає процеси створення, підтримки та контролю доступу облікових записів користувачів та служб. Це забезпечує захист інфраструктури вебдодатків, дозволяючи лише авторизованим користувачам отримувати доступ до потрібних ресурсів. Ефективне управління обліковими записами допомагає мінімізувати ризик компрометації облікових даних та несанкціонованого доступу.

Основні аспекти управління обліковими записами:

Життєвий цикл облікових записів охоплює всі етапи роботи з обліковим записом, від його створення до видалення.

- Створення облікових записів: Нові облікові записи створюються на основі запиту, перевірки та затвердження.
- Підтримка: Забезпечення актуальності та відповідності прав доступу обов'язкам користувача.
- Видалення або деактивація: Видалення облікових записів після завершення співпраці або тривалого невикористання.

Розмежування ролей та привілеїв:

Облікові записи розділяються на користувацькі, адміністративні та службові, кожна з яких має свої привілеї.

- Користувацькі облікові записи: Використовуються кінцевими користувачами з мінімальними привілеями.
- Адміністративні облікові записи: Мають підвищені привілеї для управління системами та ресурсами.
- Службові облікові записи: Призначені для служб та додатків з обмеженими правами доступу.

Автоматизація процесів управління дозволяє покращити ефективність та точність обробки облікових записів.

- Автоматизоване створення та видалення: Нові облікові записи створюються та видаляються автоматично на основі політик безпеки та ролей користувачів.
- Автоматизований моніторинг: Системи автоматично перевіряють відповідність прав доступу користувачів їхнім ролям.

Аудит облікових записів допомагає виявити та виправити надмірні привілеї або невикористовувані облікові записи.

- Періодичні перевірки: Регулярно перевіряються всі облікові записи на відповідність ролям та поточним завданням.
- Аналіз активності: Перевірка журналів активності для виявлення підозрілої або невідповідної активності.

Політики управління обліковими записами визначають правила створення, видалення та підтримки облікових записів.

- Політика створення: Визначає процеси та умови створення нових облікових записів.
- Політика деактивації та видалення: Встановлює строки та правила видалення невикористовуваних облікових записів.
- Політика привілеїв: Визначає правила надання привілеїв користувачам, адміністраторам та службам.

Моніторинг активності облікових записів дозволяє виявляти аномальну поведінку та потенційні спроби зламу.

- Централізований збір логів: Збираються журнали активності облікових записів для подальшого аналізу.
- Системи моніторингу доступу: Використовуються системи для автоматичного виявлення підозрілої поведінки облікових записів.
- Попередження та реагування: Автоматично надсилаються сповіщення про підозрілі дії, такі як повторні невдалі спроби входу.

Управління паролями та секретами забезпечує надійний захист облікових записів від зловмисників.

- Менеджери паролів: Використовуються для генерації та зберігання складних паролів.
- Політика паролів: Визначає мінімальні вимоги до складності паролів та строки їх оновлення.
- Управління секретами: Секрети, такі як ключі API, зберігаються у спеціалізованих системах керування секретами.

Обмеження строку дії доступу (Just-in-Time Access) - це підхід до контролю доступу, при якому користувачам та службам надаються привілеї лише на короткий період часу, необхідний для виконання конкретного завдання. Це допомагає зменшити ризик надмірних привілеїв та знижує ймовірність компрометації системи через несанкціонований доступ.

Основні аспекти обмеження строку дії доступу:

1. Тимчасові облікові записи та ролі:

- Тимчасові облікові записи створюються або тимчасові ролі призначаються для виконання конкретних завдань, після чого вони автоматично деактивуються.
- Тимчасові облікові записи: Створюються лише для виконання певних завдань або проєктів.
- Тимчасові ролі: Привілеї облікових записів підвищуються до певної ролі лише на обмежений період часу.
- Автоматична деактивація: Облікові записи або ролі автоматично деактивуються після завершення періоду дії.

2. Автоматизований процес запиту та надання доступу:

- Процес запиту та надання доступу до критичних систем автоматизується для зручності та швидкості виконання завдань.
- Запит доступу: Користувач надсилає запит на підвищення привілеїв або отримання тимчасової ролі.
- Перевірка та затвердження: Запит перевіряється автоматично або адміністраторами та затверджується для надання доступу.
- Автоматичне надання доступу: Після затвердження привілеї користувача підвищуються автоматично.

3. Обмеження строку дії привілеїв:

- Підвищені привілеї надаються лише на короткий строк, необхідний для виконання конкретного завдання.
- Обмежений строк дії: Привілеї автоматично видаляються після закінчення встановленого часу або завдання.

- Гнучке налаштування: Строк дії привілеїв налаштовується залежно від типу завдання та рівня привілеїв.

4. Моніторинг та аналіз активності доступу:

- Моніторинг активності облікових записів із тимчасовими привілеями допомагає виявляти та реагувати на підозрілі дії.
- Централізований збір логів: Збираються журнали активності тимчасових облікових записів та ролей.
- Аналіз поведінки: Аналізуються журнали на предмет підозрілої активності або невідповідності завданню.
- Попередження та реагування: Автоматичні попередження надсилаються адміністраторам у разі виявлення підозрілої активності.

Політики обмеження строку дії доступу:

5. Чіткі політики обмеження строку дії доступу визначають правила надання та обмеження привілеїв.

- Політика строку дії: Встановлює строки дії тимчасових привілеїв залежно від типу завдання та ролі.
- Політика запиту та затвердження: Визначає процес запиту та затвердження тимчасових привілеїв.
- Політика моніторингу: Визначає процеси моніторингу активності тимчасових облікових записів та ролей.

Застосування обмеження строку дії доступу:

- Адміністративні завдання: Підвищення привілеїв адміністратора для виконання критичних системних завдань.
- Розробка та тестування: Тимчасове підвищення прав доступу розробників та тестувальників до систем.
- Тимчасові проєкти та контракти: Надання доступу зовнішнім консультантам та підрядникам на обмежений строк.
- Резервне адміністрування: Надання резервним адміністраторам тимчасових привілеїв для виконання адміністративних завдань.

2.1.2 Моніторинг та аналіз логів:

Моніторинг та аналіз логів є важливим компонентом захисту інфраструктури вебдодатків, оскільки дозволяє виявляти підозрілу активність, реагувати на інциденти та забезпечувати безпеку даних. Ефективний моніторинг допомагає організаціям швидко ідентифікувати потенційні загрози та знизити ризик компрометації систем.

Основні аспекти моніторингу та аналізу логів:

1. **Централізований збір логів** дозволяє зберігати всі журнали активності систем та додатків в одному місці для подальшого аналізу.

- Системи збору логів: Використовуються системи для централізованого збору логів, такі як Elasticsearch, Logstash, Kibana (ELK Stack) або Splunk.
- Джерела логів: Збираються журнали з операційних систем, веб-серверів, мережевого обладнання, баз даних, додатків та служб.
- Стандартизація логів: Логи форматуються та структуруються у стандартизованому вигляді для зручного аналізу.

2. **Моніторинг системи та додатків** дозволяє відстежувати поведінку та активність компонентів інфраструктури.

- Операційні системи: Відстежується активність облікових записів, системних служб та файлів.
- Веб-сервери: Моніторяться журнали запитів та помилок веб-серверів для виявлення підозрілих запитів.
- Бази даних: Відстежуються журнали доступу до баз даних та запити SQL.
- Мережеве обладнання: Моніторяться журнали маршрутизаторів, брандмауерів, мережевих комутаторів та VPN.

3. **Моніторинг доступу та автентифікації** допомагає виявляти спроби несанкціонованого доступу.

- Автентифікація: Журнали автентифікації облікових записів відстежують успішні та невдалі спроби входу.
- Контроль доступу: Моніторяться журнали доступу до критичних ресурсів, файлів, системних служб та баз даних.

- Аномальна поведінка: Виявляються спроби входу з невідомих місць або повторні невдалі спроби входу.

4. **Моніторинг мережевого трафіку** дозволяє виявляти аномальні з'єднання та спроби атак.

- Аномальна активність: Виявляються підозрілі мережеві з'єднання, сканування портів та надмірний трафік.
- Протоколи та порти: Моніторяться мережеві протоколи та порти для виявлення нетипової поведінки.
- Системи IDS/IPS: Використовуються системи виявлення та запобігання вторгнень для моніторингу мережевого трафіку.

5. **Аналіз логів та кореляція подій** допомагають виявляти потенційні загрози та реагувати на інциденти.

- Системи SIEM: Використовуються системи управління подіями безпеки (Security Information and Event Management, SIEM) для аналізу та кореляції подій.
- Попередження та реагування: Системи SIEM надсилають автоматичні попередження про підозрілу активність адміністраторам.
- Інтеграція з інструментами реагування: Системи SIEM інтегруються з інструментами реагування на інциденти (Incident Response) для швидкого усунення загроз.

6. **Політики моніторингу та аналізу логів** визначають правила збору, зберігання та аналізу журналів.

- Політика збору: Визначає джерела логів та критерії збору журналів активності.
- Політика зберігання: Визначає строки та правила зберігання логів у централізованому сховищі.
- Політика аналізу: Визначає процеси та критерії аналізу логів для виявлення підозрілої активності.

Основні переваги моніторингу та аналізу логів:

- Виявлення загроз у реальному часі: Аналіз логів та кореляція подій дозволяють виявляти потенційні загрози та реагувати на інциденти у реальному часі.

- Покращення контролю доступу: Моніторинг доступу та автентифікації допомагає контролювати спроби доступу до критичних ресурсів.
- Аудит відповідності політикам безпеки: Аналіз логів дозволяє проводити аудит відповідності систем політикам та стандартам безпеки.

Моніторинг та аналіз логів є ключовим компонентом у забезпеченні безпеки інфраструктури вебдодатків, допомагаючи своєчасно виявляти та усувати загрози.

Шифрування та захист даних є невід'ємними компонентами безпеки інфраструктури вебдодатків. Вони допомагають забезпечити конфіденційність, цілісність і доступність даних під час зберігання та передачі, знижуючи ризики несанкціонованого доступу та витоку інформації.

Найкращими практиками захистуінфраструктури вебдодатків є:

1. **Шифрування даних під час зберігання (Encryption at Rest)** забезпечує захист данни у випадку компрометації фізичного або віртуального носія.

- Шифрування файлової системи: Використовується шифрування всієї файлової системи, щоб захистити дані від несанкціонованого доступу.
- Шифрування на рівні дисків: Дані шифруються на рівні фізичних або віртуальних дисків за допомогою BitLocker, LUKS або інших технологій.
- Шифрування на рівні баз даних: Дані шифруються на рівні баз даних для захисту від витоку через компрометацію БД або віртуальної машини.

2. **Шифрування даних під час передачі (Encryption in Transit)** забезпечує захист данних від перехоплення під час передачі через мережі.

- Протоколи SSL/TLS: Використовуються для захисту передачі даних між клієнтом та сервером.
- VPN: Віртуальні приватні мережі (VPN) захищають передачу даних між внутрішніми системами та зовнішніми користувачами.
- Захищений доступ до API: API-запити захищаються за допомогою HTTPS або інших протоколів шифрування.

3. **Управління ключами шифрування (Key Management):**

- Ефективне управління ключами шифрування (KMS) дозволяє забезпечити безпечне створення, зберігання та розподіл ключів.

- Генерація ключів: Ключі генеруються із використанням криптографічно стійких алгоритмів.
- Зберігання ключів: Ключі шифрування зберігаються у захищених сховищах (наприклад, AWS KMS, Azure Key Vault, HashiCorp Vault).
- Ротація та обмеження строку дії: Ключі шифрування регулярно обмежуються та ротуються для забезпечення їхньої актуальності.

2.1.3 Методи захисту мережі та кіцевих точок

Захист мережі та кінцевих точок (ендпоінтів) є критичним аспектом забезпечення безпеки інфраструктури вебдодатків, оскільки він дозволяє ефективно виявляти та протидіяти атакам на різних рівнях архітектури. Нижче наведено основні методи захисту мережі та кінцевих точок.

Захист мережі:

1. Брандмауери (Firewall):
   - Мережеві брандмауери:

Захищають мережеву інфраструктуру від несанкціонованого доступу шляхом фільтрації вхідного та вихідного трафіку.

Забезпечують контроль доступу між сегментами мережі за допомогою списків контролю доступу (ACL).

- Веб-брандмауери (WAF):

Захищають вебдодатки від веб-атак (SQL Injection, XSS) шляхом аналізу HTTP-трафіку. Надають захист від DDoS-атак та інших загроз на рівні вебдодатків.

2. Системи виявлення та запобігання вторгнень (IDS/IPS):

- IDS (Intrusion Detection System):

Виявляють підозрілу активність у мережевому трафіку за допомогою аналізу відомих моделей атак та надсилають попередження адміністраторам у разі виявлення загроз.

- IPS (Intrusion Prevention System):

Виявляють та блокують підозрілий трафік на основі сигнатур та поведінкових моделей. Крім того мають можливість автоматично блокувати або обмежувати трафік під час атак.

3. Віртуальні приватні мережі (VPN):

Захищають передачу даних між внутрішніми системами та зовнішніми користувачами через шифрування та встановлюють захищені з'єднання між віддаленими користувачами та мережевою інфраструктурою організації що забезпечує конфіденційність цілісність та доступність передачі даних.

4. Мережева сегментація:Розділяє мережу на окремі сегменти для ізоляції критичних систем та ресурсів. Зазвичай використовуються віртуальні локальні мережі (VLAN) та списки контролю доступу (ACL) для обмеження доступу між сегментами.

5. Захист DNS:

Фільтрація DNS-запитів: Фільтрує шкідливі або фішингові домени на основі чорних списків. DNSSEC (DNS Security Extensions): Захищає цілісність DNS-запитів через цифрові підписи.

Захист кінцевих точок (ендпоінтів):

1. Антивірусні програми та антивірусні платформи:

Виявляють та видаляють шкідливе ПЗ, таке як віруси, трояни та програми-вимагачі. Також забезпечують регулярне оновлення антивірусних баз для ефективного захисту.

2. Системи виявлення та реагування на кінцевих точках (EDR):

- **EDR(Endpoint Detection and Responce)**

Виявляє та реагує на підозрілу активність на кінцевих точках шляхом аналізу поведінки крім того забезпечує централізоване керування, моніторинг та реагування на інциденти.

- **XDR (Extended Detection and Response):**

Поєднує моніторинг та реагування на інциденти на кінцевих точках, мережі та хмарних сервісах забезпечує розширене виявлення загроз шляхом кореляції подій на різних рівнях.

3. Системи управління кінцевими точками (Endpoint Management):

- MDM (Mobile Device Management): Керує мобільними пристроями для забезпечення їхньої відповідності політикам безпеки.
- UEM (Unified Endpoint Management): Забезпечує централізоване керування комп’ютерами, мобільними пристроями та іншими кінцевими точками.

4. Брандмауери на кінцевих точках:

Контролюють вхідний та вихідний трафік на кожній кінцевій точці.

Можуть обмежувати мережеві з'єднання на основі портів, протоколів та адрес.

2.2 Засоби та методи захисту інфраструктури за допомогою клаудпровайдерів

Клауд-провайдери пропонують широкий спектр сервісів та інструментів для захисту інфраструктури вебдодатків, допомагаючи організаціям інтегрувати ефективні засоби безпеки у свої хмарні архітектури. Відомі провайдери, такі як Amazon Web Services (AWS), Microsoft Azure та Google Cloud Platform (GCP), забезпечують комплексні рішення для контролю доступу, моніторингу, шифрування та захисту мережі. Однак кожен із них має свої унікальні переваги та недоліки, які необхідно враховувати при виборі оптимального хмарного рішення для захисту інфраструктури.

Клауд-провайдери також пропонують інструменти для автоматизації процесів забезпечення безпеки та відповідності стандартам. Це дозволяє організаціям швидко впроваджувати політики безпеки, реагувати на інциденти та автоматизувати процеси моніторингу, що значно полегшує роботу адміністраторів. Крім того, інтеграція сервісів хмарної безпеки з іншими продуктами цих провайдерів (наприклад, корпоративними системами автентифікації або офісними пакетами) допомагає підвищити ефективність та продуктивність роботи.

Вибір відповідного провайдера залежить від конкретних потреб організації, існуючої інфраструктури та вимог до безпеки. Amazon Web Services надає найширший спектр послуг для великих підприємств, Microsoft Azure ідеально

інтегрується з існуючими корпоративними системами Microsoft, а Google Cloud Platform пропонує інноваційні можливості завдяки передовим технологіям аналізу даних та штучного інтелекту. Розуміння унікальних особливостей кожного провайдера дозволяє організаціям підібрати оптимальні інструменти для ефективного захисту своєї інфраструктури.

2.2.1 Amazon Web Services(AWS)

Amazon Web Services (AWS) — провідний клауд-провайдер, який пропонує найбільший асортимент сервісів для забезпечення захисту інфраструктури вебдодатків. Засоби безпеки AWS розроблені для інтеграції у різні архітектури, що дозволяє організаціям ефективно захищати свої дані та інфраструктуру. AWS забезпечує гнучкість, масштабованість та автоматизацію процесів, дозволяючи швидко адаптуватися до мінливих вимог безпеки. Завдяки можливості налаштування індивідуальних політик доступу, моніторингу та реагування на інциденти, AWS підходить як для великих підприємств, так і для середніх компаній, які прагнуть отримати комплексний захист без значних витрат на інфраструктуру.

AWS також пропонує широкий спектр засобів для забезпечення відповідності галузевим стандартам, таких як GDPR, HIPAA та PCI DSS. Вбудовані інструменти моніторингу та аудиту допомагають організаціям забезпечувати відповідність політикам безпеки та отримувати своєчасні попередження про потенційні загрози. Сервіси AWS дозволяють централізовано керувати політиками доступу, ключами шифрування та конфігураціями ресурсів, забезпечуючи комплексний підхід до захисту вебдодатків та інфраструктури.

Незважаючи на складність конфігурації та високу вартість деяких сервісів, AWS залишається провідним вибором для багатьох організацій завдяки своїм можливостям автоматизації, інтеграції та широкому спектру функціональних можливостей. Це робить його оптимальним варіантом для тих, хто шукає гнучке та масштабоване рішення для забезпечення безпеки вебдодатків та даних у хмарі.

Переваги AWS:

1. Широкий спектр послуг

Amazon Web Services (AWS) пропонує потужний і багатогранний набір сервісів для захисту інфраструктури вебдодатків, надаючи гнучкі можливості для контролю доступу, шифрування даних, виявлення загроз та моніторингу активності. Сервіси безпеки AWS інтегровані в усі аспекти хмарної інфраструктури, дозволяючи організаціям детально налаштовувати політики доступу та контролю безпеки.

Однією з ключових особливостей AWS є здатність пропонувати різноманітні засоби контролю доступу та автентифікації. Від індивідуальних облікових записів до інтеграції з корпоративними системами автентифікації, клієнти можуть забезпечувати високий рівень безпеки та застосовувати принцип найменших привілеїв для кожного ресурсу. Також AWS підтримує багатофакторну автентифікацію, єдиний вхід (SSO) та обмеження доступу на основі ролей та атрибутів.

Широкий спектр послуг AWS дозволяє детально контролювати мережеву інфраструктуру та захист вебдодатків. Завдяки поєднанню брандмауерів, захищених віртуальних мереж та захисту від DDoS-атак, організації можуть забезпечувати безпечний доступ до своїх ресурсів із мінімальними затримками. Інтеграція засобів захисту мережі з веб-брандмауером та брандмауером застосунків дозволяє блокувати шкідливі запити на ранніх етапах, мінімізуючи ризик компрометації вебдодатків.

AWS також забезпечує високий рівень захисту даних завдяки можливостям шифрування під час зберігання та передачі. Управління ключами шифрування, підтримка політик шифрування та централізовані сховища секретів дозволяють організаціям контролювати доступ до конфіденційних даних. Вбудовані засоби моніторингу та аудиту забезпечують виявлення загроз у реальному часі та своєчасне реагування на них.

Для централізованого керування безпекою AWS пропонує засоби моніторингу та аналітики, які допомагають організаціям відстежувати та корелювати події безпеки з різних сервісів, надаючи повну видимість активності в хмарній інфраструктурі. Автоматизовані рекомендації та попередження допомагають підтримувати

відповідність стандартам безпеки та забезпечують своєчасне реагування на потенційні загрози.

Крім того, автоматизація та інтеграція засобів безпеки дозволяють організаціям швидко розгортати політики контролю доступу, моніторингу та реагування на інциденти у різних облікових записах та ресурсах AWS. Ці сервіси забезпечують комплексний підхід до захисту інфраструктури вебдодатків, дозволяючи організаціям впроваджувати найкращі практики безпеки та ефективно протистояти сучасним загрозам.

2. Автоматизація безпеки:

Amazon Web Services (AWS) забезпечує потужні можливості автоматизації безпеки, що дозволяють організаціям ефективно контролювати та реагувати на загрози в їхній інфраструктурі вебдодатків. Завдяки використанню безсерверних технологій, автоматизованих сервісів моніторингу та інструментів кореляції подій, AWS допомагає клієнтам впроваджувати автоматизовані процеси захисту.

Ключова перевага автоматизації безпеки в AWS полягає в інтеграції сервісів безпеки з безсерверними обчисленнями через AWS Lambda. Завдяки цьому сервісу організації можуть створювати функції, які автоматично реагують на події безпеки, такі як зміни конфігурацій, виявлення підозрілої активності або порушення політик доступу. Наприклад, можна автоматично відключити обліковий запис користувача, який порушив політику доступу, або повідомити адміністратора про виявлену вразливість у конфігурації мережевого ресурсу.

AWS також пропонує Security Hub, централізовану платформу для моніторингу та кореляції подій безпеки, яка об'єднує дані з різних сервісів, включаючи GuardDuty, Macie та Inspector. Security Hub забезпечує автоматизовані рекомендації та попередження про потенційні загрози, дозволяючи організаціям оперативно реагувати на інциденти. Інтеграція з AWS Lambda дозволяє автоматично запускати функції реагування у разі виявлення підозрілої активності або невідповідності політикам безпеки.

Інструменти моніторингу та аналізу в AWS, такі як CloudTrail та CloudWatch Logs, допомагають автоматизувати збір, аналіз та кореляцію логів активності

облікових записів, ресурсів та мережевого трафіку[41,42]. Ці інструменти забезпечують виявлення аномальної активності, відстеження змін конфігурацій та аналіз відповідності політикам безпеки. CloudWatch Alarms дозволяє автоматично надсилати сповіщення адміністраторам або запускати функції AWS Lambda у разі виявлення підозрілої активності.

Автоматизовані засоби забезпечення відповідності стандартам, такі як AWS Config та Inspector, дозволяють організаціям підтримувати безпеку своїх інфраструктур на високому рівні. AWS Config забезпечує постійний моніторинг конфігурацій ресурсів та автоматично сповіщає про порушення політик безпеки. Inspector проводить автоматизований аналіз вразливостей та відповідності стандартам у хмарних ресурсах, надаючи рекомендації щодо усунення виявлених ризиків [43, 44].

Інструменти автоматизації розгортання, такі як CloudFormation та Terraform, допомагають організаціям швидко впроваджувати стандартизовані конфігурації безпеки у своїй хмарній інфраструктурі. Це забезпечує повторюваність конфігурацій та можливість швидкого відновлення у разі інцидентів.

3. Гнучкість конфігурації:

Amazon Web Services (AWS) надає організаціям надзвичайну гнучкість у налаштуванні політик безпеки, контролю доступу та моніторингу активності, дозволяючи адаптуватися до мінливих вимог і специфіки інфраструктури вебдодатків. Широкий набір сервісів дозволяє детально контролювати привілеї користувачів, управління ключами шифрування, мережеву конфігурацію та захист даних.

Контроль доступу та автентифікація:

Identity and Access Management (IAM): Дозволяє детально налаштовувати політики доступу на основі ролей (RBAC) та атрибутів (ABAC). Гнучкість у визначенні дозволів та політик контролю доступу забезпечує принцип найменших привілеїв для кожного ресурсу, облікового запису та ключа API [45].

AWS Single Sign-On (SSO): Підтримує єдиний вхід (SSO) для спрощення доступу до всіх облікових записів та сервісів AWS [46].

Моніторинг та аналітика:

CloudWatch та CloudTrail: Забезпечують централізований збір та аналіз логів активності облікових записів, ресурсів та мережевого трафіку. Гнучке налаштування метрик та сповіщень дозволяє відстежувати ключові показники безпеки та швидко реагувати на підозрілу активність.

Security Hub: Надає централізовану панель для моніторингу подій безпеки та кореляції даних з різних сервісів, таких як GuardDuty, Macie та Inspector.

Шифрування та захист даних:

Key Management Service (KMS): Забезпечує централізоване управління ключами шифрування для різних хмарних сервісів, включаючи S3, RDS та DynamoDB. Політики доступу та ротації ключів дозволяють гнучко контролювати доступ до конфіденційних даних.

AWS Secrets Manager: Дозволяє централізовано керувати секретами та обмежувати доступ до них за допомогою гнучких політик доступу.

Захист мережі та вебдодатків:

Virtual Private Cloud (VPC): Дозволяє створювати захищені віртуальні мережі та контролювати мережевий доступ до ресурсів AWS. Security Groups та Network ACL забезпечують детальний контроль мережевого трафіку між сегментами мережі та зовнішніми службами.

Web Application Firewall (WAF): Надає гнучкі правила для захисту вебдодатків від веб-атак, таких як SQL Injection та XSS. Інтеграція з CloudFront дозволяє розгорнути WAF на рівні глобальних точок присутності для мінімізації затримок.

Автоматизація безпеки:

AWS Lambda: Інтеграція сервісів безпеки з AWS Lambda дозволяє створювати автоматизовані функції для виявлення та реагування на інциденти безпеки.

CloudFormation: Дозволяє автоматично розгортати стандартизовані конфігурації безпеки у різних облікових записах та ресурсах AWS.

4. Централізоване керування безпекою:

Amazon Web Services (AWS) пропонує централізовані інструменти керування безпекою, які дозволяють організаціям спрощувати процеси моніторингу, реагування

на інциденти та забезпечення відповідності політикам безпеки. Це досягається за допомогою комплексної інтеграції між різними сервісами AWS, які збирають, аналізують та корелюють дані безпеки, надаючи адміністраторам повну видимість активності в інфраструктурі.

AWS Security Hub є централізованою панеллю керування безпекою, яка об'єднує дані з різних сервісів AWS для моніторингу подій безпеки та відповідності стандартам. Основні можливості Security Hub включають:

- Кореляція подій: Інтеграція з GuardDuty, Macie, Inspector та іншими сервісами для централізованого аналізу загроз та подій.
- Оцінка відповідності: Використання вбудованих та кастомних стандартів (CIS AWS Foundations Benchmark, PCI DSS) для автоматичної оцінки конфігурацій ресурсів.

Автоматизовані рекомендації: Надання рекомендацій щодо усунення ризиків та невідповідностей стандартам.

CloudTrail:

- Забезпечує централізований збір та аналіз логів активності облікових записів та ресурсів у хмарній інфраструктурі.
- Відстежує зміни у конфігураціях та політиках безпеки, забезпечуючи повну видимість операцій у хмарі.

CloudWatch

- Дозволяє налаштовувати метрики та сповіщення для ключових показників безпеки.
- Інтеграція з CloudWatch Alarms та AWS Lambda дозволяє автоматизувати сповіщення та реагування на інциденти.

AWS Config забезпечує детальний моніторинг конфігурацій ресурсів у хмарній інфраструктурі та дозволяє автоматизувати процеси забезпечення відповідності політикам безпеки. Основні можливості AWS Config включають:

- Правила конфігурації: Встановлення правил для моніторингу конфігурацій ресурсів відповідно до політик безпеки.

- Звіти відповідності: Автоматична генерація звітів про відповідність конфігурацій ресурсів галузевим стандартам (GDPR, PCI DSS).
- Автоматизовані сповіщення: Надсилання сповіщень адміністраторам у разі виявлення невідповідностей політикам або стандартам.

AWS Organizations

AWS Organizations забезпечує централізоване управління обліковими записами та політиками безпеки у кількох облікових записах AWS. Основні можливості Organizations включають:

- Service Control Policies (SCPs): Дозволяють встановлювати централізовані політики контролю доступу для різних облікових записів.
- Централізоване виставлення рахунків: Спрощує управління витратами на безпеку шляхом об'єднання виставлення рахунків.
- Групування облікових записів: Дозволяє групувати облікові записи для застосування централізованих політик та стандартів.

AWS Inspector:

- Автоматизований аналіз вразливостей та відповідності стандартам у хмарних ресурсах.
- Забезпечує детальні звіти про виявлені ризики та рекомендації щодо їх усунення.

Trusted Advisor:

- Інструмент аналізу найкращих практик безпеки та оптимізації ресурсів у хмарній інфраструктурі.
- Забезпечує рекомендації щодо оптимізації витрат, продуктивності та безпеки.

5. Розширені можливості аналітики

Amazon Web Services (AWS) пропонує потужний набір інструментів для розширеної аналітики, які дозволяють організаціям збирати, аналізувати та корелювати події безпеки, виявляючи загрози на ранніх етапах та забезпечуючи ефективне реагування на інциденти. Використовуючи машинне навчання,

автоматизований аналіз даних та централізовані панелі керування, AWS забезпечує повну видимість активності в інфраструктурі вебдодатків.

AWS використовує машинне навчання для виявлення аномальної активності та потенційних загроз у хмарній інфраструктурі:

GuardDuty:

Використовує машинне навчання та аналіз поведінки для виявлення аномальної активності у мережевому трафіку та логах доступу. Порівнює поточну активність із відомими моделями атак та поведінки зловмисників.

Macie:

Використовує машинне навчання для автоматичного класифікування та захисту конфіденційних даних у сховищах S3. Виявляє аномальні запити та спроби доступу до конфіденційних даних.

Detective:

Аналізує журнали CloudTrail, VPC Flow Logs та GuardDuty для виявлення моделей поведінки зловмисників. Використовує візуалізацію для аналізу інцидентів безпеки та кореляції подій.

Не зважаючи на дуже потужний список сервісів та функціоналу та можливостей точково подувати свою інфраструктуру, AWS також має свої недоліки, а саме:

1. **Складність конфігурації:**

- Велика кількість сервісів: AWS пропонує безліч сервісів для забезпечення безпеки, і хоча це надає гнучкість, одночасно створює складнощі для організацій без досвіду роботи з AWS. Кожен сервіс має свої особливості налаштування, що вимагає глибокого розуміння архітектури.
- Тонкощі налаштування: Детальна конфігурація політик доступу, мережевих правил та моніторингу вимагає значного часу та ресурсів на налаштування. Недостатнє налаштування може призвести до неправильного застосування принципу найменших привілеїв або інших ризиків безпеки.

2. **Цінова політика:**

Хоча AWS використовує політику pay as use, але все одно клаудпровайдер, є достатньо дорогим і просить значну кількість коштів за свої сервіси. Деякі послуги

безпеки AWS, такі як Macie або GuardDuty, можуть бути дорогими при великому обсязі даних або активності. Це особливо стосується організацій, які мають багато даних у S3 або високу мережеву активність. Автоматизовані сервіси безпеки часто генерують додаткові витрати на обчислення та зберігання. Наприклад, зберігання та аналіз логів у CloudWatch може суттєво збільшити рахунки, якщо обсяг даних великий.

3. Обмежена підтримка мультихмарних стратегій:

Зосередженість на екосистемі AWS: Сервіси безпеки AWS здебільшого розроблені для роботи у межах екосистеми AWS, що може бути обмежуючим фактором для організацій, які використовують мультихмарну або гібридну архітектуру. Інтеграція з сервісами безпеки інших клауд-провайдерів або локальних рішень часто потребує додаткових зусиль та налаштування.

4. Складність навчання:

Технічна складність: Інтеграція та конфігурація сервісів безпеки AWS вимагає від спеціалістів високого рівня знань та досвіду роботи з хмарними технологіями.

Специфічні знання: Адміністратори повинні добре розуміти специфічні терміни та концепції AWS для правильного налаштування та використання сервісів.

2.2.2 Microsoft Azure

Microsoft Azure є одним із провідних клауд-провайдерів, який пропонує широкий спектр інструментів та засобів безпеки для захисту інфраструктури вебдодатків. Завдяки інтеграції з Microsoft Active Directory та іншими продуктами Microsoft, Azure забезпечує гнучкий та ефективний захист вебдодатків, даних та мережевої інфраструктури.

Сервіси безпеки Azure тісно інтегровані з іншими продуктами Microsoft, такими як Office 365, Dynamics 365 та Windows Server, забезпечуючи єдину платформу керування доступом і контролю конфігурацій. Azure Active Directory дозволяє централізовано керувати обліковими записами на основі ролей та атрибутів, підтримуючи єдиний вхід (Single Sign-On) та багатофакторну автентифікацію (MFA).

Azure пропонує широкий спектр автоматизованих інструментів для моніторингу та оцінки безпеки, включаючи Azure Sentinel — платформу управління подіями безпеки (SIEM), яка об'єднує дані з різних сервісів для аналізу загроз і кореляції подій. Azure Security Center дозволяє організаціям забезпечувати відповідність політикам безпеки та стандартам, автоматично оцінюючи вразливості та конфігурації ресурсів.

Azure також пропонує потужні інструменти для захисту конфіденційних даних та управління ключами шифрування через Azure Key Vault, який підтримує централізоване керування ключами та секретами з інтеграцією в Azure Active Directory для контролю доступу.

Що стосується захисту вебдодатків, Azure Web Application Firewall (WAF) забезпечує фільтрацію HTTP-трафіку для захисту від веб-атак, таких як SQL Injection та XSS. Інтеграція з Azure Application Gateway дозволяє розгортати WAF на рівні балансування навантаження для ефективного захисту. Крім того, Azure DDoS Protection пропонує засоби захисту від розподілених атак на відмову в обслуговуванні (DDoS), інтегруючись з Azure Virtual Network для ізоляції небажаного трафіку.

Azure Policy та Azure Blueprint забезпечують централізований моніторинг відповідності конфігурацій політикам безпеки та стандартам, а також надають попередньо налаштовані політики для швидкого впровадження найкращих практик безпеки. Ці інструменти допомагають автоматизувати оцінку відповідності стандартам, таким як GDPR, PCI DSS та HIPAA.

Розгорнута мережа технічної підтримки Microsoft забезпечує висококласні консультації та підтримку впровадження та налаштування сервісів безпеки Azure. Вичерпна документація та навчальні ресурси дозволяють адміністраторам швидко опанувати можливості сервісів Azure та інтегрувати їх у свої архітектури безпеки.

Переваги Microsoft Azure:

1. Інтеграція з Microsoft Active Directory

Так як Azure – це Microsoft однією з найбільших перевах Azure від інших клаудпровайдерів – це інтеграція з Microsoft Active Directory(MS AD).

Azure AD дозволяє організаціям централізовано керувати обліковими записами та забезпечувати єдиний вхід (Single Sign-On) до всіх ресурсів Microsoft. Сервіс підтримує багатофакторну автентифікацію (MFA), принцип найменших привілеїв (Least Privilege) та інтеграцію з локальними Active Directory. Крім того він пропонує гнучкі можливості контролю доступу на основі ролей (RBAC), атрибутів (ABAC) та умов та включає автоматизацію процесів керування доступом, таких як реєстрація облікових записів, скидання паролів та управління групами.

Також забезпечує гібридну ідентифікацію, що дозволяє організаціям використовувати існуючі облікові записи та групи в обох середовищах – хмарному та локальному.

Ще однією перевагою є підтримка Office 365. Azure AD спрощує управління доступом до продуктів Office 365 (Outlook, SharePoint, Teams) через єдину платформу автентифікації. Користувачі можуть використовувати одну пару облікових даних для входу до всіх продуктів Microsoft.

2. Інтеграція з продуктами Microsoft

- **Office 365**

Як зазначалось вище Azure має хорошу автоматизацію з Office 365. Крім Azure AD - Azure Security Center дозволяє централізовано контролювати безпеку та управління доступом до Office 365. Додаткові засоби захисту, такі як Azure Information Protection, забезпечують моніторинг облікових записів та захист даних у середовищі Office 365.

- **Dynamics 365**

Інтеграція з Azure AD забезпечує захист конфіденційних даних та управління доступом до бізнес-продуктів Microsoft, таких як Dynamics 365. Гібридна ідентифікація дозволяє використовувати єдиний набір облікових даних для доступу до продуктів Dynamics 365.ч

3. Сервіси автоматизації:

Microsoft не відстає від конкуретів та крім інеграції з AD та офіс надає широкий спектр сервісів, серед них і сервіси автоматизації:

Azure Sentinel - хмарна платформа управління подіями безпеки (SIEM) об'єднує дані з різних сервісів та джерел для централізованого моніторингу та кореляції подій. Платформа легко інтегрується з Azure Security Center та Azure Active Directory для аналізу загроз та використовує машинне навчання для виявлення аномальної активності та потенційних загроз, таких як несанкціонований доступ або витік даних. Крім того - забезпечує автоматизоване реагування на інциденти через Azure Logic Apps та Azure Functions.

Azure Security Center – це комплексний інструмент для моніторингу конфігурацій ресурсів, аналізу вразливостей та забезпечення відповідності стандартам безпеки. Він Автоматично оцінює ресурси на наявність вразливостей та невідповідності стандартам та інтегрується з Azure Policy для автоматизованого виправлення невідповідностей.

4. Широкий спектр засобів безпеки:

Azure пропонує багатий набір засобів для захисту інфраструктури вебдодатків, включаючи Azure Key Vault, Azure Web Application Firewall (WAF) та Azure DDoS Protection.

Azure Key Vault забезпечує централізоване управління ключами шифрування та секретами, інтегруючись з Azure Active Directory для контролю доступу. Сервіс підтримує автоматичну ротацію ключів та доступ до ключів через API.

Azure Web Application Firewall (WAF) забезпечує захист вебдодатків від веб-атак, таких як SQL Injection та XSS. Інтеграція з Azure Application Gateway дозволяє балансувати навантаження, забезпечуючи ефективний захист.

Azure DDoS Protection інтегрований з Azure Virtual Network для ізоляції небажаного трафіку та автоматично виявляє та блокує DDoS-атаки. Професійний план захисту Advanced надає можливість втручання експертів у разі складних атак [48].

5. Автоматизація відповідності стандартам:

Azure Policy забезпечує автоматизований моніторинг та дотримання політик безпеки та стандартів (GDPR, PCI DSS). Сервіс дозволяє налаштовувати правила для

автоматичного виправлення невідповідностей конфігурацій, підвищуючи рівень безпеки.

Azure Blueprint надає набір попередньо налаштованих політик для швидкого впровадження стандартів безпеки. Використовуючи готові шаблони, організації можуть створювати стандартизовані конфігурації та швидко розгортати політики безпеки.

6. Багатоканальна підтримка:

Microsoft пропонує розгорнуту мережу технічної підтримки для консультацій з впровадження та налаштування сервісів безпеки. Доступні різні рівні підтримки (Basic, Developer, Standard, Professional Direct) для різних потреб організацій.

Вичерпна документація, навчальні ресурси, вебінари та курси дозволяють адміністраторам швидко опанувати можливості сервісів Azure та підвищувати свою кваліфікацію.

7. Аналітика та моніторинг:

Log Analytics та Azure Monitor забезпечують централізоване збирання, аналіз та візуалізацію логів та метрик активності. Інтеграція з іншими сервісами дозволяє корелювати події безпеки та виявляти аномальну активність.

Azure Lighthouse забезпечує централізований моніторинг подій безпеки та кореляцію даних з різних сервісів, дозволяючи керувати кількома обліковими записами та ресурсами одночасно.

8. Розширений захист даних та мережі:

Azure Information Protection виявляє, класифікує та захищає конфіденційні дані. Сервіс дозволяє встановлювати політики доступу до конфіденційних документів.

Azure Network Security Groups (NSG) контролюють вхідний та вихідний мережевий трафік між сегментами мережі та хмарними сервісами, забезпечуючи гнучкі правила для обмеження трафіку на основі IP-адрес, портів та протоколів.

Virtual Private Network (VPN) забезпечує захищену передачу даних між локальною мережею та хмарною інфраструктурою через шифрування.

Підтримка різних протоколів шифрування (IKEv2, SSTP, OpenVPN) гарантує безпеку трафіку.

Недоліки Azure:

1. Вимогливість до знань екосистеми Microsoft:

Microsoft Azure тісно інтегрований з екосистемою продуктів Microsoft, таких як Active Directory, Office 365 та Windows Server. Хоча це забезпечує безперебійну роботу для користувачів Microsoft, організації, які не працювали раніше з продуктами Microsoft, можуть відчути труднощі в освоєнні цих інструментів. Інтеграція Azure AD з локальними Active Directory або налаштування керування ролями (RBAC) в Azure може стати складним завданням для адміністраторів без відповідного досвіду.

2. Залежність від регіональних центрів даних:

Microsoft Azure має розгалужену глобальну інфраструктуру, але не всі регіони однаково забезпечені сервісами безпеки. Деякі сервіси доступні лише в певних регіонах, що може обмежувати організації у виборі місця розташування даних та захисту інфраструктури.

Наприклад, Azure Sentinel не підтримується в усіх регіонах, а локальні політики щодо зберігання даних можуть ускладнювати вибір відповідного регіону.

3. **Неоднорідний інтерфейс користувача**:

Панелі управління сервісами Azure Security Center, Sentinel та інших інструментів безпеки мають різні підходи до інтерфейсу користувача та навігації. Це призводить до необхідності навчання адміністраторів роботі з кожним інструментом окремо та може ускладнювати процес моніторингу та налаштування.

4. Обмежена інтеграція з мультихмарними середовищами:

Хоча Azure Sentinel та Azure Security Center надають деяку підтримку мультихмарного моніторингу, вони здебільшого орієнтовані на екосистему Azure та інтеграцію з продуктами Microsoft. Це може ускладнювати організаціям із мультихмарною стратегією отримання єдиного погляду на їхню інфраструктуру.

Azure Security Center підтримує базовий моніторинг ресурсів AWS та GCP, але ця інтеграція може не забезпечувати ті ж можливості аналізу та захисту, що і для ресурсів в Azure.

5. Обмежені можливості налаштування вбудованих засобів безпеки:

Деякі вбудовані засоби безпеки в Azure, такі як Azure Security Center або Azure Firewall, можуть мати обмежені можливості налаштування порівняно зі спеціалізованими інструментами інших провайдерів або незалежних постачальників. Наприклад, Azure Firewall може бути менш гнучким у налаштуванні порівняно з аналогічними продуктами від Palo Alto Networks або Fortinet.

2.2.3 Google Cloud Platform

Google Cloud Platform (GCP) пропонує комплексний набір інструментів та сервісів безпеки для захисту інфраструктури вебдодатків. Завдяки інтеграції з власними технологіями Google, такими як BigQuery та Chronicle, GCP забезпечує потужні можливості аналізу та реагування на загрози. Глибока інтеграція з іншими сервісами Google робить його привабливим вибором для організацій, які вже використовують екосистему Google у своїй діяльності.

Окрім аналітики та машинного навчання, GCP пропонує інструменти безпеки для контролю доступу, моніторингу та аналізу подій, а також управління ключами шифрування. Використання сервісів Identity and Access Management (IAM) та Cloud Identity дозволяє організаціям налаштовувати контроль доступу з урахуванням принципу найменших привілеїв.

GCP також вирізняється потужними можливостями автоматизації та інтеграції. Завдяки таким сервісам, як Cloud Security Command Center, Security Health Analytics та Forseti Security, організації можуть централізовано моніторити події безпеки, аналізувати конфігурації ресурсів та виявляти потенційні вразливості. Інтеграція з системами хмарної інфраструктури Google забезпечує комплексний підхід до захисту даних та інфраструктури вебдодатків.

GCP орієнтований на використання хмарної інфраструктури та сервісів Google для покращення безпеки вебдодатків. Сервіси безпеки тісно інтегровані з іншими продуктами Google, такими як Google Workspace, Kubernetes Engine та Google Cloud Storage, що дозволяє організаціям створювати комплексні архітектури безпеки.

Переваги GCP:

Google Cloud Platform (GCP) пропонує комплексний набір інструментів та сервісів для захисту інфраструктури вебдодатків. Завдяки інтеграції з передовими технологіями Google, такими як BigQuery та Chronicle Security Analytics, GCP забезпечує розширені можливості аналізу загроз та автоматизації безпеки. Глибока інтеграція з іншими продуктами Google робить GCP привабливим вибором для організацій, які вже використовують екосистему Google у своїй діяльності.

1. Розширені можливості аналітики та машинного навчання:

Chronicle Security Analytics — це сервіс, який використовує аналітичні потужності Google для виявлення та реагування на загрози. Chronicle об'єднує журнали безпеки з різних джерел, аналізує їх у реальному часі та застосовує алгоритми машинного навчання для виявлення аномалій. Цей сервіс інтегрується з іншими інструментами безпеки GCP, забезпечуючи централізований підхід до аналізу подій.

BigQuery — це високопродуктивний інструмент для аналізу великих даних, який дозволяє швидко обробляти масиви логів та подій безпеки. Завдяки інтеграції з Cloud Logging та Cloud Monitoring, BigQuery надає можливість корелювати події та виявляти складні загрози.

Security Command Center (SCC) є централізованою платформою керування безпекою, яка інтегрується з сервісами Google Cloud та сторонніми продуктами для аналізу вразливостей, моніторингу конфігурацій та автоматизації відповідності стандартам. SCC дозволяє організаціям отримати єдине уявлення про їхню безпеку та швидко виявляти потенційні ризики [49].

2. Комплексний захист вебдодатків:

GCP пропонує повний спектр засобів для забезпечення безпеки вебдодатків та даних. Identity and Access Management (IAM) забезпечує детальний контроль доступу до ресурсів Google Cloud на основі ролей та атрибутів. Сервіс дозволяє впроваджувати принцип найменших привілеїв для кожного користувача та ресурсу [50].

Cloud Identity пропонує багатофакторну автентифікацію, єдиний вхід (SSO) та інтеграцію з корпоративними системами автентифікації. Це дозволяє організаціям централізовано керувати доступом до ресурсів Google та впроваджувати єдині політики контролю доступу.

Google Cloud Armor надає захист вебдодатків від DDoS-атак та веб-загроз. Використовує технології Google для фільтрації трафіку та забезпечує низьку затримку завдяки глобальній мережі Google. Інтеграція з глобальною мережею дозволяє Cloud Armor забезпечувати захист вебдодатків навіть під час масштабних атак.

Web Security Scanner — це автоматизований інструмент для сканування вебдодатків на наявність поширених вразливостей, таких як XSS або SQL Injection. Інтеграція зі службою Google Kubernetes Engine забезпечує автоматичне сканування контейнерних додатків.

3. Потужна автоматизація та інтеграція:

GCP надає інструменти для автоматизації моніторингу та реагування на інциденти, дозволяючи організаціям централізовано керувати подіями безпеки. Security Command Center (SCC) є централізованою панеллю керування безпекою, яка інтегрується з іншими сервісами Google Cloud для моніторингу та аналізу конфігурацій ресурсів.

Forseti Security — це відкритий інструмент для виявлення вразливостей у конфігураціях хмарних ресурсів, аналізу відповідності політикам та автоматизації перевірки безпеки. Forseti Security дозволяє організаціям швидко виявляти потенційні ризики у своїй інфраструктурі та застосовувати відповідні політики безпеки.

Security Health Analytics автоматизує аналіз конфігурацій ресурсів на наявність вразливостей та невідповідності стандартам. Інтеграція зі Security Command Center дозволяє автоматично виявляти та усувати проблеми безпеки у конфігураціях ресурсів [51].

4. Інтеграція з іншими продуктами Google:

Google Cloud Platform інтегрується з іншими продуктами Google, такими як Google Workspace, Kubernetes Engine та Google Cloud Storage, забезпечуючи комплексний підхід до захисту інфраструктури вебдодатків.

Google Workspace (колишній G Suite) дозволяє централізовано керувати обліковими записами, налаштовувати контроль доступу та захист даних у корпоративних додатках Google. Інтеграція з Cloud Identity забезпечує централізоване керування обліковими записами та єдиний вхід до всіх продуктів Google.

Google Kubernetes Engine (GKE) забезпечує інтеграцію засобів безпеки з контейнерними оркестраторами, такими як Kubernetes, дозволяючи централізовано контролювати доступ та конфігурації кластерів. Використання GKE дозволяє впроваджувати принцип найменших привілеїв для кожного ресурсу та облікового запису [52].

Google Cloud Storage (GCS) пропонує централізоване управління даними та ключами шифрування, дозволяючи впроваджувати найкращі практики зберігання та захисту даних. Інтеграція з Cloud IAM дозволяє централізовано контролювати доступ до кожного об’єкта та встановлювати політики контролю доступу.

Недоліки GCP:

1. Обмежена географічна присутність та покриття регіонів:

Порівняно з іншими великими клауд-провайдерами, такими як AWS або Azure, інфраструктура Google Cloud Platform має менш розгалужену географічну мережу центрів обробки даних. У деяких регіонах відсутні локальні центри обробки даних, що може спричиняти затримки доступу до ресурсів та впливати на відповідність вимогам локального законодавства щодо зберігання даних. Крім того, не всі сервіси GCP доступні у всіх регіонах, що може ускладнювати використання їх можливостей для глобальних організацій.

2. Залежність від внутрішньої екосистеми Google:

GCP орієнтований на інтеграцію з іншими продуктами Google, такими як Google Workspace, Google Analytics та BigQuery. Ця тісна інтеграція може створювати складнощі для організацій, які використовують різні екосистеми або прагнуть мультихмарних рішень. Наприклад, інструменти моніторингу та аналізу в GCP здебільшого оптимізовані для ресурсів Google Cloud, але мають обмежену підтримку інших хмарних провайдерів.

3. Недостатньо розвинені можливості автоматизації відповідності стандартам:

Попри наявність таких інструментів, як Security Command Center та Forseti Security, автоматизація процесів відповідності стандартам у GCP не така розвинена, як в інших провайдерів. Наприклад, Forseti Security вимагає ручної інтеграції та конфігурації для аналізу відповідності політикам безпеки.

4. Обмежена підтримка підприємницьких рішень:

GCP не має такої ж широкої мережі партнерів та підтримки підприємницьких рішень, як AWS або Azure. Це може створювати складнощі для організацій, які потребують спеціалізованих рішень або консультацій щодо впровадження складних архітектур безпеки.

5. Складність конфігурації контролю доступу:

Хоча GCP має потужні засоби контролю доступу, такі як Identity and Access Management (IAM) та Cloud Identity, складність конфігурації політик доступу може призвести до ненавмисних помилок та витоків даних. IAM у GCP має складну систему ролей, що вимагає ретельного налаштування та управління, особливо при застосуванні принципу найменших привілеїв.

6. Незручний інтерфейс користувача:

Консоль Google Cloud Platform має складний інтерфейс користувача з великою кількістю налаштувань, які можуть бути заплутаними навіть для досвідчених адміністраторів. Деякі сервіси, такі як Security Command Center або Cloud IAM, можуть мати неочевидні опції конфігурації та вимагати додаткового часу на налаштування. Крім того, різні сервіси мають інтерфейс, який відрізняється за стилем та принципами роботи, що ускладнює роботу адміністраторам.

2.2.4 Порівняння клаупровайдерів

Amazon Web Services (AWS), Microsoft Azure та Google Cloud Platform (GCP) є трьома найбільшими хмарними провайдерами у світі. Кожен з них пропонує різноманітні інструменти та сервіси безпеки для захисту інфраструктури вебдодатків.

1**. Географічне покриття та регіональна присутність:**

- AWS: Має найширшу географічну присутність з понад 80 зонами доступності в 26 регіонах. Перевага AWS полягає в доступності локальних центрів обробки даних по всьому світу.
- Azure: Присутній у понад 60 регіонах з більш ніж 140 зонами доступності. Забезпечує підтримку великої кількості регіонів, особливо для глобальних організацій.
- GCP: Менш розгалужена мережа центрів обробки даних у порівнянні з AWS та Azure. Має понад 35 зон доступності в 26 регіонах. Недоліком є обмежена доступність деяких сервісів у певних регіонах.

2. **Інтеграція та сумісність:**

- AWS: Пропонує широкий спектр інтеграцій з продуктами третіх сторін через AWS Marketplace, а також підтримує сумісність із системами інших хмарних провайдерів. Засоби моніторингу, як CloudWatch, інтегруються з різними сервісами для централізованого аналізу загроз.
- Azure: Глибоко інтегрований з продуктами Microsoft, такими як Office 365, Dynamics 365 та Windows Server. Azure Security Center забезпечує централізоване керування обліковими записами та політиками доступу. Однак інтеграція з мультихмарними середовищами є обмеженою.
- GCP: Орієнтований на інтеграцію з продуктами Google, такими як Google Workspace та BigQuery. Chronicle Security Analytics та Security Command Center забезпечують централізований аналіз загроз, але підтримка мультихмарних середовищ обмежена.

3. Сервіси безпеки та автоматизація:

AWS:

- Контроль доступу та аутентифікація: Identity and Access Management (IAM) для детального контролю доступу до ресурсів.
- Моніторинг та аналітика: CloudTrail та CloudWatch для централізованого моніторингу подій безпеки.

- Шифрування та захист даних: Key Management Service (KMS) для управління ключами шифрування.
- Захист мережі: Web Application Firewall (WAF) та GuardDuty для моніторингу та захисту вебдодатків.
- Автоматизація відповідності: Security Hub для централізованого аналізу та оцінки відповідності стандартам.

Azure:

- Контроль доступу та аутентифікація: Azure Active Directory (Azure AD) для єдиного входу та багатофакторної автентифікації.
- Моніторинг та аналітика: Azure Sentinel та Security Center для централізованого аналізу загроз та відповідності стандартам.
- Шифрування та захист даних: Azure Key Vault для управління ключами шифрування та секретами.
- Захист мережі: Azure Firewall та Web Application Firewall (WAF) для захисту мережевої інфраструктури.
- Автоматизація відповідності: Azure Policy та Blueprint для автоматизації політик безпеки та відповідності стандартам.
- GCP:
- Контроль доступу та аутентифікація: Identity and Access Management (IAM) та Cloud Identity для контролю доступу.
- Моніторинг та аналітика: Chronicle Security Analytics, Security Command Center (SCC) та Forseti Security для моніторингу конфігурацій та аналізу загроз.
- Шифрування та захист даних: Cloud Key Management Service (Cloud KMS) та Cloud Data Loss Prevention (DLP) для захисту конфіденційних даних.
- Захист мережі: Google Cloud Armor та Web Security Scanner для захисту вебдодатків від DDoS-атак та вразливостей.

Автоматизація відповідності: Security Health Analytics для автоматизації процесів відповідності стандартам.

4. Інтерфейс користувача та зручність використання:

- AWS: Пропонує розвинений інтерфейс користувача з доступом до різних сервісів через єдину консоль. Проте деякі сервіси мають свої панелі керування, що може ускладнювати використання для нових користувачів.
- Azure: Має інтуїтивно зрозумілий інтерфейс користувача з гнучким доступом до сервісів через Azure Portal. Глибока інтеграція з продуктами Microsoft забезпечує знайоме середовище для адміністраторів, які працюють з Windows Server.
- GCP: Консоль Google Cloud Platform має складний інтерфейс користувача з великою кількістю налаштувань, які можуть бути заплутаними навіть для досвідчених адміністраторів. Деякі сервіси, такі як Security Command Center або Cloud IAM, можуть мати неочевидні опції конфігурації.

Загальні висновки:

AWS: Найширший вибір сервісів безпеки, гнучкість у конфігурації та інтеграція з різними інструментами через Marketplace.

Azure: Глибока інтеграція з продуктами Microsoft та потужні засоби автоматизації відповідності стандартам.

GCP: Орієнтований на машинне навчання та аналітику, пропонує розширені можливості автоматизації через Security Command Center та Forseti Security.

Висовки за розділом 2

У розділі 2 розглянуто загальні засоби та механізми захисту інфраструктури вебдодатків, а також ті, які пропонують провідні хмарні провайдери: Amazon Web Services (AWS), Microsoft Azure та Google Cloud Platform (GCP). Аналіз показав, що кожен провайдер пропонує комплексні інструменти безпеки, орієнтовані на контроль доступу, моніторинг, захист мережі та автоматизацію відповідності стандартам. Крім того було проведено порівняння всіх клаудпровайдерів, де було оглянути переваги та кожного. Усі три провайдери пропонують потужні засоби та механізми захисту інфраструктури вебдодатків. Проте, Amazon Web Services (AWS) виділяється найбільшою гнучкістю, масштабованістю та різноманіттям інструментів безпеки. Завдяки своїй широкій географічній присутності, детальному контролю доступу та

потужним інструментам моніторингу та аналізу подій, AWS є оптимальним вибором для більшості організацій. Так AWS не має таку тісну, інтеграцію з сервісами Microsoft та Google, але, якщо компанія не базується на використанні цих сервісів, то AWS є найбільш оптимальним варіантом.

РОЗДІЛ 3
УДОСКОНАЛЕННЯ ІНФРАСТРУКТУРИ ВЕБДОДАТКУ

3.1 Удосконалений метод реагування на інциденти в інфраструктурі

В сучасних хмарних середовищах, таких як AWS, однією з найбільш поширених і значущих загроз є проблеми з доступом до ресурсів. Неправильно налаштовані політики доступу, неконтрольовані привілеї та недостатній моніторинг змін у конфігураціях можуть призвести до витоку даних, несанкціонованого доступу та інших інцидентів безпеки. Більшість загроз у хмарі пов'язані саме з проблемами управління доступом, що підкреслює необхідність удосконалення методів захисту.

Основною причиною цих проблем є складність налаштування політик доступу та управління ними у великих масштабах. Зі збільшенням кількості користувачів і ресурсів, керування правами доступу стає все більш важким завданням. Застарілі або надмірно розширені привілеї користувачів можуть залишатися непоміченими, створюючи потенційні точки входу для зловмисників.

Крім того, традиційні методи моніторингу та реагування на інциденти часто є недостатніми для динамічних і швидко змінюваних хмарних середовищ. Недостатня автоматизація процесів управління доступом і реагування на інциденти збільшує час виявлення та усунення загроз, що може мати серйозні наслідки для безпеки даних.

З огляду на ці виклики, виникає необхідність у розробці вдосконаленого методу реагування на інциденти безпеки, який би враховував специфіку хмарних середовищ і забезпечував швидке та ефективне усунення проблем з доступом. Такий метод повинен включати автоматизовані процеси моніторингу, виявлення та відновлення критичних налаштувань доступу, щоб мінімізувати ризики та забезпечити безпеку хмарної інфраструктури.

Звичайний метод реагування на інциденти в хмарних середовищах включає кілька ключових етапів. Спочатку відбувається виявлення інциденту за допомогою різних інструментів моніторингу, які фіксують підозрілу активність або потенційні

загрози. Після цього проводиться оцінка інциденту, щоб визначити його серйозність і вплив на систему, а також вирішити, які заходи потрібно вжити для усунення проблеми.

Реакція на інцидент зазвичай полягає у виконанні певних дій для зменшення загрози, наприклад, зміни налаштувань доступу або ізоляції уражених ресурсів. Після цього автоматизації переходить до етапу відновлення, де відновлюється нормальна робота шляхом використання резервних копій або повторного налаштування політик доступу. Завершальним етапом є документування інциденту та аналіз причин і наслідків для покращення методів захисту в майбутньому.

Такий метод реагування має свої недоліки, зокрема, довгий час виявлення і усунення загроз, недостатню автоматизацію процесів і високу ймовірність людських помилок.

Удосконалений метод реагування на інциденти враховує сучасні виклики та потреби хмарних середовищ. Він включає автоматизацію процесів моніторингу, виявлення і відновлення, що дозволяє швидше та ефективніше реагувати на загрози.

1. Постійний моніторинг та виявлення: в методі використовуються інструменти для постійного моніторингу змін у налаштуваннях доступу та автоматичного аналізу цих змін. Це дозволяє швидко виявляти несанкціоновані дії та реагувати на них.

2. Автоматичне відновлення критичних налаштувань: У разі виявлення несанкціонованих змін або видалення важливих налаштувань, система автоматично відновлює їх з резервних копій. Це забезпечує безперервність і стабільність роботи системи.

3. Використання тегів для критичних налаштувань: в методі впроваджено систему маркування, яка дозволяє визначати критичні налаштування, що потребують особливої уваги. Це забезпечує їх пріоритетний моніторинг і відновлення.

4. Сповіщення про інциденти: система автоматично надсилає сповіщення про виявлені інциденти та вжиті заходи. Це дозволяє оперативно інформувати відповідальних осіб і забезпечувати контроль за безпекою.

Цей вдосконалений метод дозволяє швидко реагувати на інциденти, мінімізувати ризик втрати даних і підвищити рівень безпеки хмарної інфраструктури, що є важливим для сучасних вебдодатків.

Впровадження цього вдосконаленого методу реагування на інциденти дозволяє досягти кількох важливих результатів:

Швидка реакція на інциденти: Автоматизовані процеси моніторингу та виявлення значно скорочують час від виявлення інциденту до його усунення. Це мінімізує час, протягом якого система залишається вразливою, та зменшує ризик негативних наслідків.

Зменшення ризику втрати даних: Автоматичне відновлення критичних налаштувань з резервних копій забезпечує надійний захист від втрати або компрометації даних. Це підвищує загальний рівень безпеки системи, знижуючи ризики, пов’язані з несанкціонованими змінами.

Підвищення ефективності управління доступом: Використання тегів для маркування критичних налаштувань дозволяє ефективніше керувати правами доступу. Це дозволяє зосередити увагу на найважливіших аспектах безпеки, забезпечуючи належний рівень контролю за доступом до ресурсів.

Оперативне сповіщення про інциденти: Інтеграція з системою сповіщень забезпечує швидке інформування відповідальних осіб про виявлені інциденти та вжиті заходи. Це підвищує прозорість процесу реагування та забезпечує контроль за безпекою на всіх етапах.

Зниження людських помилок: Автоматизація більшості процесів реагування та відновлення знижує ймовірність людських помилок, що можуть виникати під час ручного втручання. Це забезпечує більш надійний і передбачуваний захист системи, підвищуючи її стійкість до загроз.

3.2 Програмна реалізація удосконаленого методу реагування на інциденти

Для програмної реалізації удосконаленого методу необхідно побудувати тестову інфраструктуру, яка дозволить перевірити ефективність запропонованих рішень та забезпечити надійне функціонування системи в реальних умовах. Тестова інфраструктура має імітувати робоче середовище, включаючи основні сервіси та компоненти, що використовуються в хмарних платформах.

Для побудови інфраструктури вебдодатку, було використано інструмент IaC Terraform Використання Terraform для побудови інфраструктури є важливим, оскільки цей інструмент забезпечує автоматизацію та повторюваність процесів, що значно зменшує ризик помилок, пов'язаних з ручним налаштуванням. Terraform дозволяє описувати інфраструктуру як код, що полегшує управління конфігураціями, забезпечує прозорість змін та інтеграцію з системами контролю версій. Це робить інфраструктуру більш масштабованою і портативною, дозволяючи легко розгортати та змінювати ресурси у різних хмарних середовищах. Крім того, Terraform підтримує інтеграцію з іншими інструментами для автоматизації та оркестрації, що дозволяє створювати комплексні та гнучкі рішення для управління інфраструктурою.

Для тестової інфраструктури, було розгорнуто мережу за допомогою AWS VPC, яка складається з:

Мережа: Створено основну мережу з використанням CIDR блоку 10.0.0.0/16, що дозволяє використовувати велику кількість приватних IP-адрес для різних ресурсів

Internet Gateway: Для забезпечення двостороннього зв'язку між ресурсами у VPC та зовнішнім інтернетом було додано Internet Gateway. Це дозволяє ресурсам у публічних підмережах відправляти та отримувати трафік з інтернету.

Підмережі:

Дві публічні підмережі: Розгорнуто публічні підмережі:

- wordpress-public-subnet-1 з використанням CIDR 10.0.1.0/24

- wordpress-public-subnet-2 з використанням CIDR 10.0.2.0/24

для розміщення веб-серверу з вебдодатком та інших сервісів, які вимагають прямого доступу до інтернету. Ці підмережі забезпечені прямим доступом до Internet Gateway.

Дві приватні підмережі:

- wordpress-private-subnet-1 з використанням CIDR 10.0.3.0/24
- wordpress-private-subnet-2 з використанням CIDR 10.0.4.0/24

Використовуються для більш захищених ресурсів, таких як бази даних або внутрішні додаткові сервери, які не вимагають прямого доступу до інтернету. Ресурси в цих підмережах можуть доступатися до інтернету через NAT Gateway або інші засоби, що обмежують зовнішній доступ.

Ця структура VPC забезпечує ефективне розділення та управління мережевими ресурсами, забезпечуючи високий рівень безпеки та гнучкість у управлінні трафіком.

Після цього було створено, тестову конфігурацію доступу, до ресурсів, що включає в собі:

IAM роль: Розгорнуто роль під назвою "AdminWithoutIAM", яка призначена для адміністративного доступу без прямих прав на керування іншими IAM ролями. Для цієї ролі було прикріплено політику (policy), що обмежує її дії в рамках певних управлінських функцій, виключаючи можливість зміни конфігурацій IAM.

IAM група: З метою керування доступом для різних типів користувачів і автоматизації прав доступу були створені тестові групи. Ці групи допомагають організувати користувачів згідно з їх ролями та відповідними політиками безпеки.

IAM користувач: Для тестування та демонстрації правил безпеки було створено тестового користувача з обмеженим доступом. Цей користувач був доданий до раніше створених груп, що дозволило налаштувати його права доступу відповідно до корпоративних політик безпеки.

Після цього було створено EC2 instance, що є по суті сервером для вебдодатку. На цьому сервері було розгорнуто Wordpress вебдодаток. Далі було розгорнуто наступні елементи інфраструктури:

Elastic IP: Для забезпечення стабільного та постійного зовнішнього IP-адресу для EC2 інстансу було надано Elastic IP.

Load Balancer (ELB): Для розподілу вхідного веб-трафіку між інстансами та забезпечення високої доступності та надійності веб-додатку було розгорнуто Load Balancer. ELB допомагає управляти навантаженням, перерозподіляючи запити до веб-серверів на основі їх поточного навантаження та стану.

Security Group: Контроює мережевий доступ до нашого вебдодатку, що по суті брандмауером. Було описано 3 правила, які надають доступ до 443(HTTPS) та 80(HTTP) порту з світу, та до 22(SSH) порт з мережі VPN.

Після успішної побудови тестової інфраструктури, ми зосередили увагу на розробці комплексної системи реагування на інциденти, що забезпечує високий рівень безпеки конфігурацій доступу в AWS. Основна мета системи полягала в автоматизації процесів виявлення, реагування та відновлення від інцидентів, що стосуються неправомірних змін політик доступу та ролей IAM, що складається з наступних компонентів

AWS CloudTrail: його було налаштовано для надсилання журналів аудиту в AWS S3, де зберігається детальна інформація про всі API-виклики, здійснені в обліковому записі AWS. Це дозволяє моніторити та аналізувати будь-які зміни в політиках доступу та інші критичні дії.

Amazon S3: використовується для зберігання резервних копій політик доступу та логів CloudTrail, що забезпечує надійність зберігання та легкий доступ до історії конфігурацій при потребі в аналізі чи відновленні.

Amazon EventBridge: налаштовано для моніторингу специфічних подій, зареєстрованих CloudTrail, таких як видалення або зміна політик IAM. EventBridge спрямовує ці події до відповідних Lambda-функцій для швидкого реагування.

Amazon SNS (Simple Notification Service): використовується для надсилання автоматизованих сповіщень в реальному часі про інциденти або відновлення політик, що значно підвищує прозорість процесів безпеки.

AWS Lambda: використовується для роботи функцій автоматизованого бекапування та відновлення політик доступу в AWS, що є однією зі складових автоматизованої рішення реагування, на інциденти в інфраструктурі.

Також було використано тегування, ресурсів AWS для створення тегів для критично важливих політик доступу, які не повинні зазнавати будь яких змін.

Логіка роботи:

Рішення використовує AWS CloudTrail для постійного моніторингу та логування всіх API-викликів в AWS, зосереджуючись на критичних змінах, таких як модифікація або видалення політик IAM, а саме iam:DeletePolicy та iam:UpdatePolicy, що забезпечує постійний моніторинг на підозрілу активність.

AWS Event Bridge, використовується для, налаштування трігерів, які будуть запускати функції бекапу та відновлення політик доступу. Налаштований він наступним чином:

Для функції бекапу налаштовано правило, що запускає автоматизацію кожен тиждень

Для реагування на інциденти було налаштовано AWS EventBridge для запуску функцій та створено 2 Lambda функції, а саме функція бекапу AWS Policy та відновлення політик.

AWS Event Bridge, налаштований він наступним чином:

Для функції бекапу налаштовано правило, що запускає автоматизацію кожен тиждень

Для функції було створено правило, яке реагує API запити iam:UpdatePollicy та iam:DeletePolicy від CloudTrail, які запускають функцію відновлення AWS Policy

Функція бекапу:

AWS IAM політики доступу — це елемент управління безпекою хмарної інфраструктури, який використовується для визначення дозволених та заборонених API запитів до ресурсів AWS. Ці політики представлені у форматі JSON, що дозволяє детально налаштувати правила доступу.

Lambda-функція виконує критичну роль у цьому процесі, використовуючи API для отримання вмісту відповідного JSON файлу політики. Після отримання файла, функція створює точну копію цієї політики та зберігає її у попередньо визначений S3 bucket. Це перетворює S3 bucket в сховище для резервних копій політик,

забезпечуючи можливість відновлення в разі їх випадкового або навмисного змінення.

Крім основної функції збереження, Lambda-функція також перевіряє наявність тега 'Critical' у політиці. Якщо цей тег присутній, вона дублює його у створений файл, що підсилює важливість таких файлів для організації. Після успішного збереження політики, функція відправляє повідомлення на електронну пошту, інформуючи відповідних адміністраторів про успішне виконання завдання.

Функція відновлення:

Функція відновлення в AWS Lambda призначена для швидкого реагування на інциденти, пов'язані з невідповідними змінами або видаленням політик доступу. Ця функція автоматично активується відповідно до подій, визначених у EventBridge, які вказують на потенційні інциденти безпеки. Вона працює наступним чином:

Функція спершу дістає JSON файл з S3 бакета, де зберігається копія політики. Це виконується через API-виклик до S3, щоб отримати вміст файлу, який містить усі правила доступу. Перед відновленням функція перевіряє тег 'Critical' для кожного файла. Якщо файл має цей тег, це вказує на високу важливість політики для організації, і функція продовжує процес відновлення.

Функція використовує вміст завантаженого файлу для відновлення політики в IAM. Якщо політика була видалена або змінена, вона створює нову версію політики або оновлює існуючу на основі збережених даних у файлі. Це здійснюється через API-виклики IAM, які реімплементують політику відповідно до її оригінального стану. Після успішного відновлення політики, функція через сервіс AWS SNS надсилає повідомлення на пошту про успішне відновлення.

3.3 Практична реалізація удосконаленого методу реагування на інциденти

В цьому підрозділі було зосереджено увагу на практичній реалізації удосконаленого методу реагування на інциденти, розробленого в попередніх розділах. Основною метою цього етапу є перевірка автоматизації в реальних умовах

для оцінки її здатності ефективно ідентифікувати, реагувати та відновлювати політики доступу в AWS після їх несанкціонованого видалення або зміни.

Етапи перевірки рішення:

Для початку було зроблено перевірку, функції бекапу політик доступу.

На рисунку 3.4 зображені політики доступу AWS

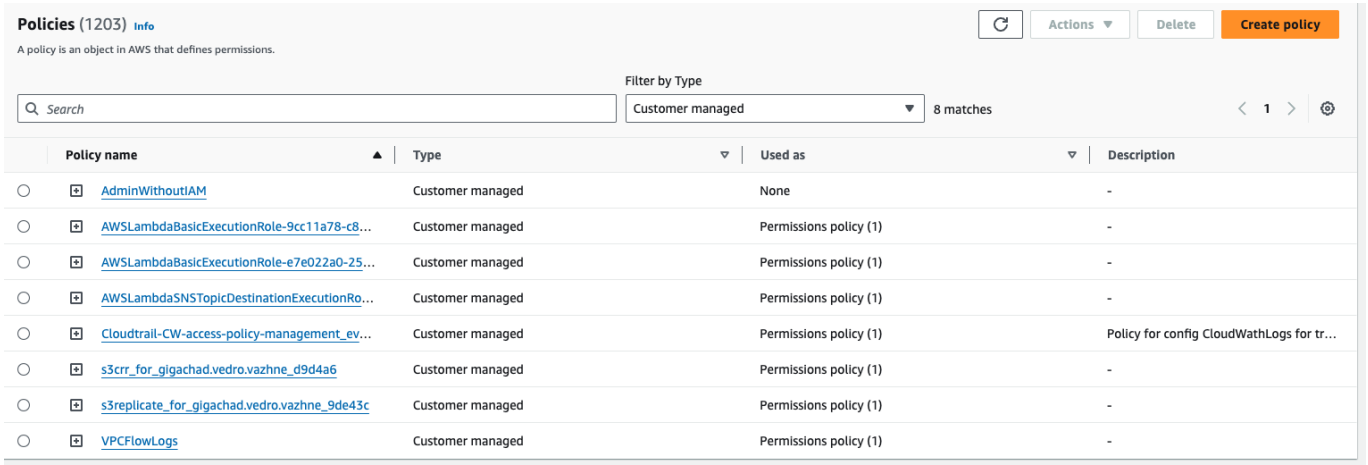

Рисунок 3.4 - Політики доступу AWS

Для тесту встановлено тег Critical на політику AdminWithoutIAM для запуску Lambda функції відновлення політики доступу, цей тег є сигналом для системи, що будь-які зміни або видалення цієї політики мають бути негайно виявлені та оброблені з високим пріоритетом.

Після цього було проведено перевірку реагування на зміни в політиці AdminWithoutIAM, для цього спочатку було перевірено вміст політики

На рисунку 3.5 зображено частину політики доступу AdminWithoutIAM

Рисунок 3.5 - Частина політики доступу AdminWithoutIAM

Після ініціації тесту, було здійснено кероване видалення політики доступу AdminWithoutIAM, аби активувати систему відновлення. Наступним кроком було детальне переглядання логів в AWS CloudTrail, яке дозволило переконатися у коректному захопленні події видалення і відповідній реакції системи, що підтвердило активацію механізмів відновлення.

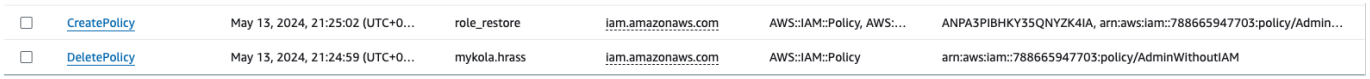

Рисунок 3.6 - Логи успішної роботи Lambda функції DeletePolicy

На рисунку 3.7 зображено відновлену політику AdminWithoutIAM та переконаємось в успішній роботі

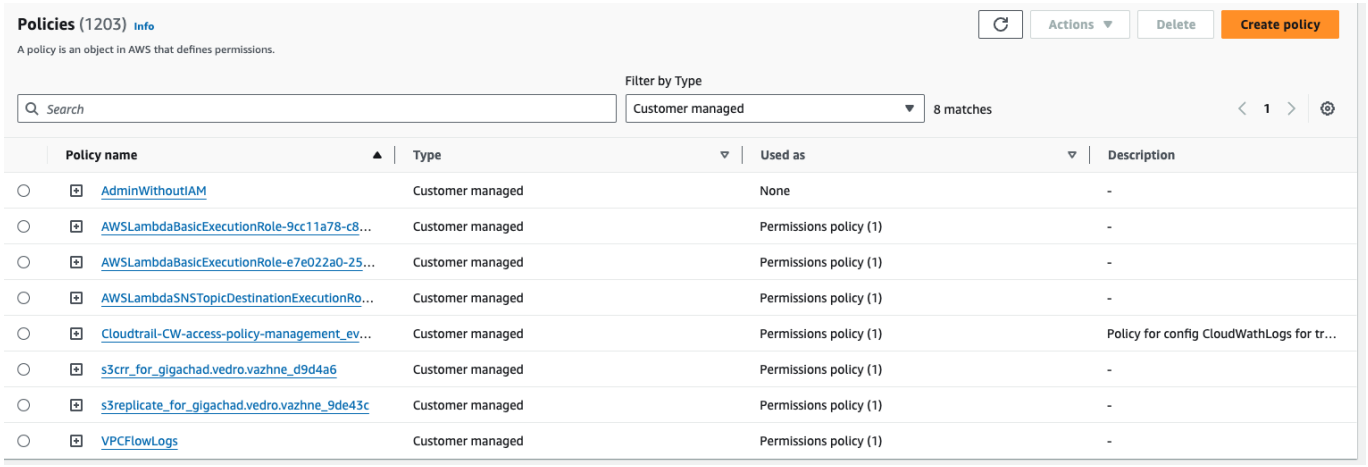

Рисунок 3.7 - Відновлена політика доступу AdminWithoutIAM

Крім того на рисунку 3.8 зображено повідомлення на пошті, про успішне відновлення

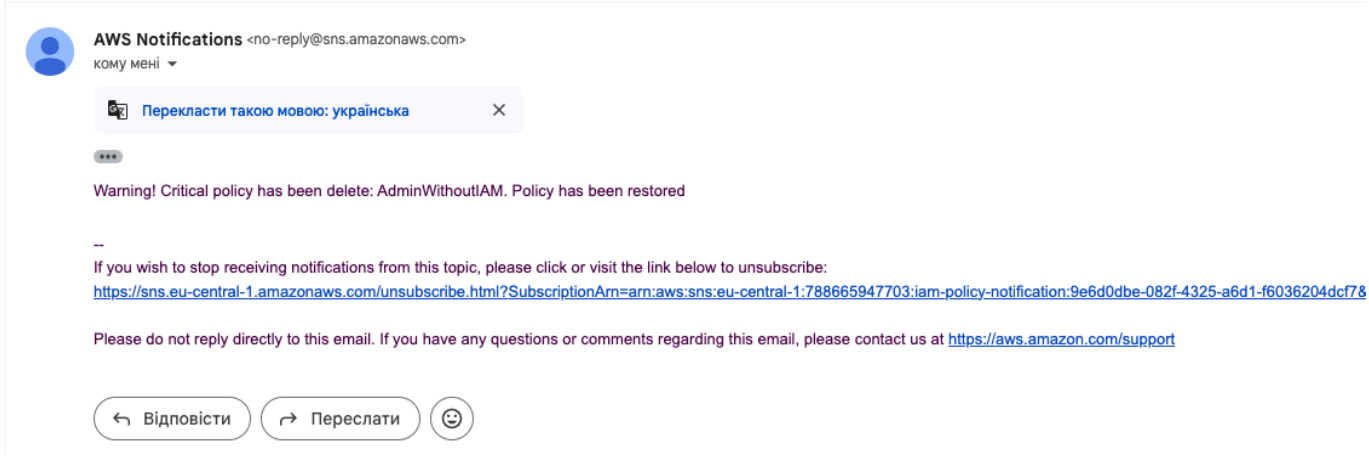

Рисунок 3.8 - Повідомлення про успішну роботу функції відновлення

Тестування, відновлення політики доступу було успішно виконано, наступним кроком було зміна в політики , шляхом видалення певних API запитів в політиці, а саме iam:UpdateAssumeRolePolicy.

На рисунку 3.9 зображено вміст зміненої політики AdminWithoutIAM

```
{
    "Statement": [
        {
            "Action": [
                "iam:DeactivateMFADevice",
                "iam:CreateServiceSpecificCredential",
                "iam:DeleteAccessKey",
                "iam:DeleteGroup",
                "iam:UpdateOpenIDConnectProviderThumbprint",
                "iam:UpdateGroup",
                "iam:CreateRole",
                "iam:CreateLoginProfile",
                "iam:DeleteServerCertificate",
                "iam:UploadSSHPublicKey",
                "iam:DetachGroupPolicy",
                "iam:DetachUserPolicy",
                "iam:DeleteOpenIDConnectProvider",
                "iam:ChangePassword",
                "iam:PutGroupPolicy",
                "iam:UpdateLoginProfile",
                "iam:UpdateServiceSpecificCredential",
                "iam:CreateGroup",
                "iam:RemoveClientIDFromOpenIDConnectProvider",
                "iam:UpdateUser",
                "iam:DeleteUserPolicy",
                "iam:AttachUserPolicy",
                "iam:DeleteRole",
                "iam:UpdateRoleDescription",
```


Рисунок 3.9 - Вміст зміненої політики AdminWithoutIAM

Після внесення змін в політику, було здійснено детальний перегляд логів у AWS CloudTrail, що дозволило підтвердити успішну реакцію Lambda функції на внесені зміни. Логи чітко відображали активацію функції відновлення, вказуючи на її коректну роботу та здатність ефективно реагувати на інциденти, пов’язані з модифікацією політик. Ця інформація свідчила про надійність системи у виявленні та відновленні політики

☐	CreatePolicyVersion	May 13, 2024, 21:31:22 (UTC+0...	role_restore	iam.amazonaws.com	AWS::IAM::Policy	arn:aws:iam::788665947703:policy/AdminWithoutIAM
☐	DeletePolicyVersion	May 13, 2024, 21:31:22 (UTC+0...	role_restore	iam.amazonaws.com	AWS::IAM::Policy	arn:aws:iam::788665947703:policy/AdminWithoutIAM
☐	CreatePolicyVersion	May 13, 2024, 21:31:15 (UTC+0...	mykola.hrass	iam.amazonaws.com	AWS::IAM::Policy	arn:aws:iam::788665947703:policy/AdminWithoutIAM

Рисунок 3.10 - Логи успішної роботи Lambda функції відновлення

Додатково було здійснено аналіз вмісту відновленої політики, де було виявлено наявність доданого дозволу iam:UpdateAssumeRolePolicy. Це дозвіл, що було інтегровано у політику в ході її автоматичного відновлення, свідчить про ефективність розробленої функції. Наявність цього дозволу є важливим індикатором того, що функція не лише успішно відновила політику до її первісного стану, але й коректно застосувала всі необхідні зміни, що могли бути внесені до моменту інциденту.

Під час тестування також було перевірено нотифікації, що надходили на електронну пошту. Виявлено, що функція коректно відправляла повідомлення про внесені зміни в політиці, а також інформувала про успішну роботу функції відновлення. Ці електронні повідомлення містили деталі змін, включно з вказівкою на конкретні дії, які були виконані функцією, забезпечуючи чітке розуміння процесів, що відбувалися в системі. Надійність нотифікацій підтверджувала високий рівень прозорості та ефективність моніторингу в рамках розробленої системи.

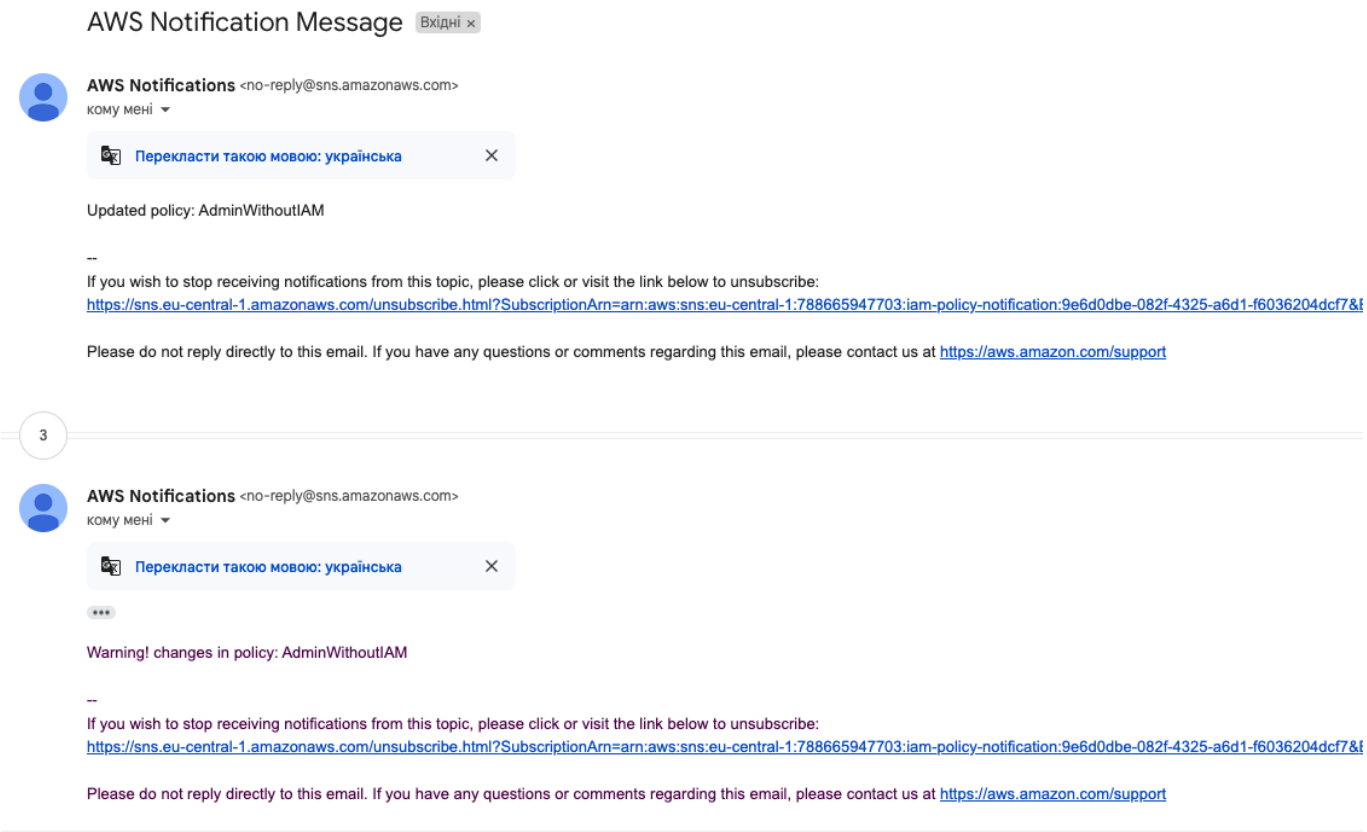

Рисунок 3.11 - Повідомлення про несанкціоновану зміну в політиці та її успішне відновлення

Результати тестування підтвердили високу ефективність і надійність розробленої системи реагування на інциденти. Під час тестів було продемонстровано, що система може швидко відновлювати критично важливі політики доступу після їхнього випадкового чи навмисного видалення.

Основні моменти, які було підтверджено під час тестування, включали:

- **Швидке Виявлення Змін:** Система здатна оперативно виявляти зміни в політиках, завдяки інтеграції з AWS CloudTrail, що дозволяє миттєво реагувати на будь-які модифікації.
- **Ефективне Відновлення:** Відновлення політик відбувається автоматично з використанням резервних копій, збережених у S3 бакетах. Функція відновлення демонструє здатність не тільки відновлювати втрачені дані, але й оновлювати політики до їхніх останніх стабільних версій.
- **Надійність Нотифікацій:** Система забезпечує відправлення нотифікацій про всі дії, пов’язані з відновленням політик, що дозволяє користувачам бути в курсі змін і стану відновлення.
- **Забезпечення Безпеки:** Через використання критичних тегів та стратегічне управління версіями політик, система запобігає потенційним порушенням безпеки, які можуть виникнути через неправомірні зміни або видалення важливих політик.

Ці результати вказують на те, що розроблена система є комплексним і надійним рішенням для управління інцидентами безпеки у хмарних середовищах, забезпечуючи стабільність, безпеку та відмінний контроль над конфігураціями доступу.

Висновки до розділом 3

У третьому розділі було детально описано процес розробки та програмну реалізацію удосконаленого методу реагування на інциденти доступу в хмарній інфраструктурі. Було виконано аналіз існуючих проблем безпеки, пов'язаних з недосконалостями у системах управління доступом, що вимагають вдосконалення для забезпечення більш високого рівня захисту даних.

Завдяки розробленій системі, яка автоматизує процеси моніторингу та відновлення критичних політик доступу, було досягнуто збільшення швидкості реагування на несанкціоновані зміни, забезпечивши таким чином більшу стійкість системи до потенційних загроз. Система демонструє високу ефективність у виявленні

та відновленні важливих політик доступу, що є ключовим аспектом для забезпечення неперервності бізнес-процесів і захисту конфіденційної інформації.

В результаті впровадження системи було проведено ряд тестів, які підтвердили її ефективність та надійність. Тестування показало, що система здатна оперативно реагувати на зміни, відновлювати політики до їх первісного стану і відправляти сповіщення про зміни, забезпечуючи таким чином високий рівень прозорості та контролю за безпекою хмарних ресурсів.

Ці результати демонструють значний прогрес у покращенні захисту хмарної інфраструктури та можуть бути застосовані для подальшого розширення та удосконалення системи безпеки в масштабах організації або навіть галузі в цілому.

ВИСНОВКИ

У кваліфікаційній роботі розв’язано актуальне завдання щодо удосконалення методів реагування на інциденти безпеки в хмарній інфраструктурі. Результати дослідження включають наступне:

У першому розділі було проведено глибокий аналіз існуючих загроз та вразливостей хмарних інфраструктур. Розглянуто основні канали витоку даних і методи несанкціонованого доступу до ресурсів хмари, включаючи атаки на рівні API і маніпуляції з доступом користувачів.

У другому розділі описані існуючі методи та засоби захисту інфраструктури вебдодатків, розглянуто та порівняно сервіси забезпеченню безпеки інфраструктури вебдодатків від різних клаудпровайдерів Azure, GCP та AWS.

У третьому розділі було розроблено удосконалення методу реагування на інциденти доступу в хмарному сердовищі, здійснено практичне тестування розробленої системи, яке підтвердило її ефективність. Застосування системи на тестовій інфраструктурі демонструє високу точність і швидкість реагування на інциденти, що забезпечує значне зниження часу до відновлення нормального функціонування системи після атак.

Ці результати вказують на значний прогрес у сфері захисту хмарних ресурсів та можуть бути використані для подальшого вдосконалення політик безпеки на рівні організації та галузі загалом.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Звіт Verizon Data Breach Investigations Report 2021 [Електронний ресурс]. – Режим доступу: https://www.verizon.com/about/news/verizon-2021-data-breach-investigations-report.
2. "User Behavior Analytics for Cyber Security: A Survey" (2019). – Режим доступу: https://scholarlykitchen.sspnet.org/2020/10/13/elsevier-has-deployed-an-end-user-tracking-tool-for-security/.
3. "A Comprehensive Study on User Behavior Analytics for Network Security" (2018) [Електронний ресурс]. – Режим доступу: https://ieeexplore.ieee.org/document/9498875.
4. "What is User Behavior Analytics (UBA)?" (Splunk). – Режим доступу: https://www.splunk.com/en_us/products/user-behavior-analytics.html.
5. "The Ultimate Guide to User Behavior Analytics" (LogRhythm) [Електронний ресурс]. – Режим доступу: https://logrhythm.com/a-guide-to-user-and-entity-behavior-analytics-ueba/.
6. "How User Behavior Analytics Can Improve Your Cybersecurity Posture" (Forrester) [Електронний ресурс]. – Режим доступу: https://www.forrester.com/blogs/category/cybersecurity/.
7. ISO 27001 [Електронний ресурс]. – Режим доступу: https://www.iso.org/standard/72642.html.
8. Національний інститут стандартів і технологій (NIST) [Електронний ресурс]. – Режим доступу: https://www.nist.gov/cyberframework.
9. "Аномальне виявлення в комп'ютерних системах" Дугласа М. [Електронний ресурс]. – Режим доступу: https://link.springer.com/chapter/10.1007/978-3-031-11438-0_32.
10. "Огляд методів виявлення аномалій" Чарльза Ц. [Електронний ресурс]. – Режим доступу: https://ieeexplore.ieee.org/document/9033532.
11. "Виявлення аномалій у мережевій поведінці" С. Чаудри [Електронний ресурс]. – Режим доступу:https://dergipark.org.tr/tr/download/article-file/3240049

12. "Огляд методів виявлення аномалій у мережевій поведінці" А. Синха, Д. Рао [Електронний ресурс]. – Режим доступу: https://www.sciencedirect.com/science/article/abs/pii/S0167404820302170

13. "Виявлення аномалій для кібербезпеки: ринковий огляд" Gartnet [Електронний ресурс]. – Режим доступу: https://www.gartner.com/en/documents/3342317.

14. "Виявлення аномалій за допомогою глибоких нейронних мереж" І. Махмуда [Електронний ресурс]. – Режим доступу: https://arxiv.org/abs/1901.03407

15. "Виявлення аномалій у великих обсягах даних за допомогою методів на основі графів" М. Салех [Електронний ресурс]. – Режим доступу: https://arxiv.org/abs/2302.00058

16. "Стан виявлення аномалій у кібербезпеці" М. Вілл [Електронний ресурс]. – Режим доступу: https://www.sciencedirect.com/science/article/abs/pii/S0167404820302170.

17. "Anomaly Detection for Cybersecurity" [Електронний ресурс]. – Режим доступу: https://www.xenonstack.com/insights/cyber-network-security.

18. "CERT Coordination Center: Incident Response Guide" [Електронний ресурс]. – Режим доступу: https://www.sei.cmu.edu/education-outreach/courses/course.cfm?coursecode=P28

19. "The State of Cybersecurity in 2023" by Cybersecurity Insights [Електронний ресурс]. – Режим доступу: https://www.isaca.org/resources/news-and-trends/isaca-now-blog/2023/state-of-cybersecurity-2023-navigating-current-and-emerging-threats

20. Гейс, Р. "Машинне навчання для кібербезпеки: Не панацея" [Електронний ресурс]. – Режим доступу: https://ieeexplore.ieee.org/document/9998305.

21. Шьолькопф, Б. "Ефективні методи виявлення аномалій". Режим доступу: https://www.sciencedirect.com/topics/computer-science/anomaly-detection.

22. Сміт, Дж. "Нейронні мережі для виявлення аномалій у кібербезпеці" [Електронний ресурс]. – Режим доступу: https://www.researchgate.net/publication/343343019_Anomaly_Detection_for_Cyber-Security_Based_on_Convolution_Neural_Network_A_survey.

23. О'Рейлі, Т. "Використання Python для реалізації протоколів безпеки" [Електронний ресурс]. – Режим доступу: https://www.oreilly.com/library/view/mastering-python-for/9781788992510/.

24. Лю, І. "Роль підготовки даних у моделях кібербезпеки" [Електронний ресурс]. – Режим доступу: https://ieeexplore.ieee.org/document/9791666.

25. Рафаель Герцог, інші. "Kali Linux розкрито: Освоєння розподілу для тестування проникнення" [Електронний ресурс]. – Режим доступу: https://www.amazon.com/Kali-Linux-Revealed-Penetration-Distribution/dp/0997615605.

26. Себастьян Рашка. "Машинне навчання з Python" [Електронний ресурс]. – Режим доступу: https://www.amazon.com/Introduction-Machine-Learning-Python-Scientists-ebook/dp/B01M0LNE8C.

27. Андреас Мюллер і Сара Гвідо. "Вступ до машинного навчання з Python" [Електронний ресурс]. – Режим доступу: https://www.amazon.com/Introduction-Machine-Learning-Python-Scientists-ebook/dp/B01M0LNE8C.

28. Еліс Чженг. "Оцінка моделей машинного навчання" [Електронний ресурс]. – Режим доступу: https://docs.aws.amazon.com/machine-learning/latest/dg/evaluating_models.html.

29. Пітер Брюс і Ендрю Брюс. "Практична статистика для дата-сайєнтистів" [Електронний ресурс]. – Режим доступу: https://www.amazon.com/Practical-Statistics-Data-Scientists-Essential/dp/1491952962.

30. Джейсон Кеннон. "Автоматизація безпеки за допомогою Python" [Електронний ресурс]. – Режим доступу: https://www.amazon.com/Python-Cybersecurity-Automated-beginner/dp/B098GL3WW7.

31. Жульєн Вегент. "Забезпечення DevOps: Безпека у хмарі" [Електронний ресурс]. – Режим доступу: https://www.amazon.com/Securing-DevOps-Security-Julien-Vehent/dp/1617294136.

32. Webroot. "Майбутні тренди у кібербезпеці" [Електронний ресурс]. – Режим доступу: https://www.webroot.com/us/en/business/integrated-solutions.

33. Маккейб, Дж. "Сучасні системи кібербезпеки: Підходи та виклики" [Електронний ресурс]. – Режим доступу: https://link.springer.com/chapter/10.1007/978-3-031-24673-9_9.

34. Нортон, С. "Аналіз безпеки комп’ютерних мереж" [Електронний ресурс]. – Режим доступу: https://www.amazon.com/Security-Analysis-Principles-Technique-2nd/dp/B00L2BRQHQ.

35. Фішер, Р. "Технології захисту інформації в мережі" [Електронний ресурс]. – Режим доступу: https://www.amazon.com/Best-Sellers-Network-Security/zgbs/digital-text/16977293011.

36. Хілл, К. "Архітектура безпеки для розробників" [Електронний ресурс]. – Режим доступу: https://www.amazon.com/Best-Sellers-Books-Security-Design/zgbs/books/7743006011.

37. Ванг, Ф. "Комплексні системи безпеки на основі машинного навчання" [Електронний ресурс]. – Режим доступу: https://ieeexplore.ieee.org/document/9888151.

38. Чен, Л. "Хмарні обчислення і безпека" [Електронний ресурс]. – Режим доступу: https://www.amazon.com/Cloud-Security-Comprehensive-Secure-Computing/dp/0470589876.

39. О’Ніл, Л. "Захист веб-додатків: Практичні аспекти" [Електронний ресурс]. – Режим доступу: https://www.amazon.com/Web-Application-Security/s?k=Web+Application+Security.

40. "Статистичний аналіз у кібербезпеці" Мартінес, Л. [Електронний ресурс]. – Режим доступу: https://www.tandfonline.com/journals/tsec20

41. "Розробка рекомендацій для систем безпеки" Томпсон, Р. [Електронний ресурс]. – Режим доступу: https://link.springer.com/chapter/10.1007/978-3-031-05237-8_54

42. "Безпека персональних даних: виклики та рішення" Фостер, Дж. [Електронний ресурс]. – Режим доступу: https://www.routledge.com/9781317245544

43. "Системи штучного інтелекту в кібербезпеці" Харт, С. [Електронний ресурс]. – Режим доступу: https://link.springer.com/book/10.1007/978-3-319-98842-9

44. "Стандарти безпеки даних" - Кук, Д. [Електронний ресурс]. – Режим доступу: https://www.qualityassurancemag.com/article/delivering-food-safety/

45. "Захист ідентифікаційних даних у мережі" Вудс, Е. [Електронний ресурс]. – Режим доступу: https://www.amazon.com/Network-Security-Books/b?ie=UTF8&node=3746

46. "Мережеві заходи безпеки і їх ефективність" Бенсон, Т. [Електронний ресурс]. – Режим доступу: https://www.amazon.com/Network-Security-Books/b?ie=UTF8&node=3746

47. "Мережеві протоколи та безпека" Грант, Р. [Електронний ресурс]. – Режим доступу: https://www.amazon.com/Network-Security-Books/b?ie=UTF8&node=3746

48. "Стандарти безпеки для захисту даних" Ньюман, К. [Електронний ресурс]. – Режим доступу: https://www.amazon.com/Data-Protection-Guidebook-Notification-Cybersecurity/dp/B09PMFWVGC

49. "Протоколи безпеки для корпоративних мереж" Шульц, М. [Електронний ресурс]. – Режим доступу: https://www.amazon.com/Network-Protocols-Security-Professionals-vulnerabilities/dp/1789953480

50. "Виклики та можливості для машинного навчання у безпеці" Пірс, Ж. [Електронний ресурс]. – Режим доступу: https://arxiv.org/abs/2002.09254

ДОДАТКИ

ДОДАТОК А

1. Грасс Микола, Микола Браіловський. Удосконалення захищеної інфраструктури веб додатку за допомогою клаудпровайдера Amazon Web Services , PCSITS 2024, Київ – матеріали конференції, ст. 97 – 99.

ДОДАТОК Б

Код Lambda функції бекапу політик доступу

```
import json
import boto3

def lambda_handler(event, context):
    iam = boto3.client('iam')
    s3 = boto3.client('s3')
    sns = boto3.client('sns')
    bucket_name = 'univ-iam-role-backup-bucket-228'
    sns_topic_arn = 'arn:aws:sns:us-east-1:788665947703:iam-policy-notification'

    # Paginator to handle multiple policies if they exceed the API response limit
    paginator = iam.get_paginator('list_policies')
    message_details = []  # Collect details for SNS message

    for page in paginator.paginate(Scope='Local'):
        for policy in page['Policies']:
            policy_arn = policy['Arn']
            policy_name = policy['PolicyName']
            default_version_id = policy['DefaultVersionId']

            # Fetch the default version of the policy
            policy_version_response = iam.get_policy_version(PolicyArn=policy_arn,
VersionId=default_version_id)
            document = policy_version_response['PolicyVersion']['Document']

            # Add a wrapper to match the desired format
            wrapped_policy_document = {
                "Version": document['Version'],
                "Statement": document['Statement']
            }

            # Serialize the wrapped policy data to JSON
            json_data = json.dumps(wrapped_policy_document, indent=4)

            # Check for the 'Critical' tag
            tags = iam.list_policy_tags(PolicyArn=policy_arn)
            is_critical = any(tag['Key'] == 'Critical' and tag['Value'].lower() == 'true' for tag in tags['Tags'])
            tagging = 'Critical=true' if is_critical else 'Critical=false'

            # Save JSON data to an S3 bucket with tagging
            file_key = f"policies/{policy_name}.json"
            s3.put_object(Body=json_data, Bucket=bucket_name, Key=file_key, Tagging=tagging)
            print(f"Successfully uploaded {file_key} to {bucket_name} with tag '{tagging}'.")
            message_details.append(f"{policy_name}.json")
```

```
    # Send SNS notification
    if message_details:
        sns_message = f"Backup was made and moved to bucket {bucket_name}:\n" +
"\n".join(message_details)
        sns.publish(
            TopicArn=sns_topic_arn,
            Message=sns_message,
            Subject='Backup Complete Notification'
        )

    return {
        'statusCode': 200,
        'body': json.dumps('Successfully processed all policies.')
    }
```

Код Lambda функції відновлення політик доступу

```
import json
import boto3

def lambda_handler(event, context):
    s3 = boto3.client('s3')
    iam = boto3.client('iam')
    sns = boto3.client('sns')
    bucket_name = 'univ-iam-role-backup-bucket-228'  # Replace with your actual S3 bucket name
    topic_arn = 'arn:aws:sns:us-east-1:788665947703:iam-policy-notification'  # Replace with your actual
SNS Topic ARN

    response = s3.list_objects_v2(Bucket=bucket_name, Prefix='policies/')
    if 'Contents' in response:
        for obj in response['Contents']:
            file_key = obj['Key']
            policy_name = file_key.split('/')[-1].replace('.json', '')
            policy_arn = f'arn:aws:iam::788665947703:policy/{policy_name}'

            # Load JSON file from S3
            obj = s3.get_object(Bucket=bucket_name, Key=file_key)
            policy_document = obj['Body'].read().decode('utf-8')

            # Check "Critical" tag on S3 object
            tags_response = s3.get_object_tagging(Bucket=bucket_name, Key=file_key)
            tags = {tag['Key']: tag['Value'] for tag in tags_response['TagSet']}
            if tags.get('Critical', 'false').lower() == 'true':
                # Check if the policy exists
                try:
                    existing_policy = iam.get_policy(PolicyArn=policy_arn)
                    is_policy_existing = True
                except iam.exceptions.NoSuchEntityException:
                    is_policy_existing = False

                if is_policy_existing:
```

```
            # Remove old versions of the policy
            versions = iam.list_policy_versions(PolicyArn=policy_arn)
            for version in versions['Versions']:
                if not version['IsDefaultVersion']:
                    iam.delete_policy_version(PolicyArn=policy_arn, VersionId=version['VersionId'])

            # Update the existing policy
            iam.create_policy_version(
                PolicyArn=policy_arn,
                PolicyDocument=policy_document,
                SetAsDefault=True
            )
            message = f"Updated policy: {policy_name}"
            print(message)
            sns.publish(TopicArn=topic_arn, Message=message)
        else:
            # Create a new policy with the "Critical" tag
            try:
                new_policy = iam.create_policy(
                    PolicyName=policy_name,
                    PolicyDocument=policy_document,
                    Tags=[{'Key': 'Critical', 'Value': 'true'}]
                )
                message = f"Created new policy: {policy_name} with Critical tag"
                print(message)
                sns.publish(TopicArn=topic_arn, Message=message)
            except iam.exceptions.EntityAlreadyExistsException:
                print(f"Policy with name {policy_name} already exists. Skipping creation.")
                continue
    else:
        print(f"Skipping non-critical policy: {policy_name}")

return {
    'statusCode': 200,
    'body': json.dumps('All policies processed successfully!')
}
```