priate model is proposed that provides a formalized description and criteria for evaluating the effectiveness of each of the operations and the recognition procedure as a whole. At the same time, for the first time, the list of criteria for assessing the quality of pre-processing of images, subject to neural network analysis in the biometric authentication system, has been substantiated, and for the first time, approaches to determining the parameters of interference and recognizing attacks using dummies have been proposed. The approach to determining the parameters of obstacles involves comparing the parameters of obstacles with the location and number of key and control faces that they overlap. Recognition of attacks is proposed to be implemented based on the analysis of the dynamics of basic emotions, the dynamics of eye movement parameters and the environment. The results of this study are important in the context of the development of effective biometric authentication tools, as they provide a formalized description of the requirements for the functionality of the main components of this procedure for recognizing the identity and emotions of personnel of critical infrastructure facilities.

Keywords: model, critical infrastructure, face image, iris, neural network, biometric authentication, information protection, information security.

Корченко Олександр Григорович, лауреат Державної премії України в галузі науки і техніки, Заслужений діяч науки і техніки України, доктор технічних наук, професор, перший проректор Державний університет інформаційно-комунікаційних технологій, професор Університету Комісії Народної Освіти (Краків, Польща).

Oleksandr Korchenko, laureate of the State Prize of Ukraine in the field of Science and Technology, Honored Worker of Science and Technology of Ukraine, Dr Hub. (Eng), Professor, Vice-Rector for Research, National Aviation University, Professor of the National Education Commission of the University, Krakow, Poland.
E-mail: icaocentre@nau.edu.ua.
Orcid ID: 0000-0003-3376-0631.

Терейковський Олег Ігорович, аспірант, Національний авіаційний університет.

Oleh Tereikovskyi, PhD student, National Aviation University.
E-mail: tereikovskyio@gmail.com.
Orcid ID: 0000-0001-5045-0163.

DOI: 10.18372/2410-7840.26.18840
УДК 004.056.5

ОСОБЛИВОСТІ ВИКОРИСТАННЯ AMAZON INSPECTOR ДЛЯ ВИЯВЛЕННЯ ВРАЗЛИВОСТЕЙ ХМАРНИХ ДОДАТКІВ

Андрій Партика, Богдан Недодус

Основними проблемами з якими може зіштовхнутись бізнес можуть слугувати вразливості до різноманітних кібератак, втрати конфіденційності даних, збільшення кількості збоїв і зменшення стабільності інформаційної інфраструктури, збільшення капітальних витрат, нові вимоги до незалежності даних, проблеми з масштабуванням інформаційної інфраструктури бізнесу. Зазначені вище проблеми можуть слугувати підґрунтям для міграції на хмарні технології, що в свою чергу забезпечить зменшення видатків на підтримку інфраструктури, підвищить ефективність управління інформаційної інфраструктури в порівнянні з роботою в локальному середовищі, збільшить гнучкість організації. Актуальність дослідження полягає в покращенні інформаційної безпеки, забезпечення конфіденційності, цілісності і доступності, виявленню вразливостей додатку і середовища завдяки використання вбудованих служб AWS. Метою даної роботи є впровадження оцінки і покращення безпеки робочого середовища і додатку, розгорнутому на базі хмарних сервісів, шляхом автоматизації сканування і аналізу робочого навантаження AWS.

***Ключові слова**: Amazon Web Services, AWS, Amazon Inspector, IAM, хмарні технології, вразливість, інфраструктура, моніторинг.*

ВСТУП

За умов стрімкого і безперервного розвитку інформаційних технологій різні мотиви можуть стимулювати бізнес-перетворення, які підштовхують власників підприємств до переходу на хмарні технології. Хмарні обчислення – це технологія, яка розширює сфери комп'ютерної мережі, створюючи середовище, яке пропонує масштабованість, краще використання апаратного забезпечення, програми на вимогу та сховище, а також нижчі витрати в довгостроковій перспективі завдяки створенню віртуальних серверів, клонованих із існуючих екземплярів, кожна з яких пропонує майже миттєве підвищення продуктивності, що дозволяє компаніям швидко й динамічно реагувати на нові вимоги [1]. «Хмара» може розміщуватися на території компанії або за її межами. Однак із зростанням попиту потреба в конфіденційності, цілісності та доступності даних стала однією з найважливіших проблем у хмарних об-

численнях. Щоб гарантувати задоволення цих потреб, у хмарі використовуються різні криптографічні підходи, моделі розподілу ролей, налаштування спеціальних механізмів авторизації та аутентифікації [2]. Amazon Web Services (AWS) є світовим лідером у сфері хмарних провайдерів і надає широкий спектр хмарних послуг, зокрема: обчислення, зберігання даних, бази даних, мережі, інструменти розробки програмного забезпечення, штучний інтелект, аналітику, безпеку тощо [3]. У цій роботі ми досліджуємо деякі вбудовані інструменти безпеки та аналізу хмарних ресурсів та розглядаємо питання безпеки хмарних додатків розгорнутих у AWS.

АНАЛІЗ ОСТАННІХ ДОСЛІДЖЕНЬ І ПУБЛІКАЦІЙ

Хмарні обчислення широко застосовуються компаніями для розміщення додатків із покращеною продуктивністю за невеликих операційних витрат і складності. Розвиток хмарних додатків поєднується зі збільшенням векторів загроз безпеці та вразливостей. Авторами [4] пропонується новий інструмент оцінки та забезпечення безпеки для хмари під назвою CloudSafe, який забезпечує автоматизовану оцінку безпеки та забезпечує найкращий контроль безпеки для хмари шляхом зіставлення різних інструментів безпеки. Але варто зазначити, що цей інструмент вимагає додаткових налаштувань та впровадження стороннього програмного забезпечення у хмарну інфраструктуру вашої організації, що не завжди є дозволеним з точки зору безпеки самої організації.

У роботі [5] зазначено, що рівень безпеки існуючих додатків повинен містити вдосконалені рішення безпеки для підвищення рівня безпеки, а також продуктивності запропонованої архітектури. Запропонована архітектура безпеки містить рішення для досягнення конфіденційності, цілісності з сильною політикою автентифікації для розробки хмарних програм у веб-службі Amazon. Проте навіть найкращі рішення вимагають програмних та безпекових оновлень в зв'язку з появою нових вразливостей та потенційних загроз. Саме тому існує необхідність постійного моніторингу та аналізу працюючих хмарних додатків на наявність такого роду загроз [6-8].

AWS надає багато різних інструментів безпеки, щоб допомогти клієнтам захистити свої облікові записи та програми AWS. В даній роботі ми дослідимо один з сервісів, що відповідає за безпеку саме програм і додатків. Amazon Inspector – це служба оцінки безпеки програм, розгорнутих на віртуальних машинах (EC2). Ці оцінки включають доступ до мережі, загальні вразливості та ризики (CVE), контрольні показники Center for Internet Security (CIS), а також загальні найкращі практики, такі як вимкнення входу для root користувача по протоколу SSH і перевірка дозволів системного каталогу на ваших примірниках EC2 [9]. На основі даних, наданих у програмі агента, Inspector створює звіт із детальним списком результатів безпеки, упорядкованих за ступенем небезпеки. Метою даної роботи є впровадження оцінки і покращення безпеки хмарного додатку шляхом автоматизації сканування і аналізу робочого навантаження AWS за допомогою Amazon Inspector.

ОСНОВНА ЧАСТИНА

Перш за все виділимо найбільш поширені вразливості, які виникають при роботі з сервісами AWS:

1. Загальнодоступне сховище зберігання даних S3. Сервіс S3 дозволяє створювати так звані бакети (bucket) – організоване місце для статичного зберігання файлів. Служба дозволяє налаштувати відро S3 таким чином, що інші користувачі зможуть отримати доступ до нього, тобто, зробити публічним. Це означає, що будь-хто з відповідним URL посиланням або дозволом може отримати доступ для читання, інколи – запису. Загальнодоступний бакет може бути зручним для обміну загальнодоступними або для розміщення статичних веб-сайтів, проте це створює низку ризиків безпеки. Наслідками цієї проблеми можуть бути різними, в залежності від того чи користувач ззовні має право на запис і чи зберігається конфіденційна інформація у бакеті. Анонімний користувач може прочитати збережені дані і перезаписати їх, що еквівалентно видаленню. Якщо у бакеті розміщений статичний контент, тоді зловмисник отримує можливість змінити будь-що всередині та, наприклад, встановити скрипт XSS [10]. Amazon рекомендує періодично перевіряти конфігурацію і налаштувань дозволів бакетів. S3 також пропонує механізм захисту шляхом шифрування на стороні сервера. Застосовуючи ці заходи безпеки, користувачі можуть мінімізувати ризики і забезпечити конфіденційність, цілісність і доступність своїх даних [11, 12];

2. Ескалація привілеїв. Типова вразливість, яка дозволяє отримати більше привілеїв, ніж може мати один обліковий запис, що в свою чергу дасть зловмиснику неавторизований доступ до більшої частини ресурсів середовища. Прикладом реалізації такої вразливості може слугувати надання привілеїв співробітнику, щоб він міг

створювати нові віртуальні сервера. Далі користувач передає існуючу роль в EC2 машину. Якщо існує роль із вищими привілеями, яка може бути надана віртуальному серверу, тоді цей користувач може створити новий віртуальний сервер з цією роллю. Додатковою проблемою може стати керування середовищам із декількома обліковими записами. Це створює надмірну залежність від політик контролю доступу, які надає AWS. Вони можуть надавати занадто багато привілеїв і створювати ситуації, в яких можливе зловживання ними. Реалізовуючи дану вразливість зловмисник може отримати несанкціонований доступ до ресурсів, наприклад, доступ до бази даних, даних облікових записів користувачів або вихідні файли програм, розгорнутих на віртуальних серверах. В свою чергу зловмисник може модифікувати або видаляти важливі дані, що порушує цілісність інформації. Завдяки привілеям порушник спокійно може розповсюдити зловмисне програмне забезпечення або розгорнути середовище для реалізації атак на інші системи [13];

3. Підробка запитів на стороні сервера. SSRF (Server-Side Request Forgery) – це вразливість вебдодатків, яка дозволяє зловмиснику ініціювати серверні запити до інших ресурсів у мережі. Це може призвести до розкриття конфіденційної інформації, втручання у роботи внутрішніх систем або впливу на зовнішні сервіси. Під час реалізації цієї атаки зловмисник намагається переконати сервер виконувати запити до внутрішньої або зовнішніх систем своїми інструкціями. Цього можна досягти, контролюючи вхідні дані, наприклад, URL-адресу, що передається на віртуальний сервер для виконання запитів. Хакер може спрямувати ці запити на важливі ресурси. Наслідки успішної атаки можуть бути доволі серйозними. Якщо зловмиснику вдасться скомпрометувати внутрішні системи або служби, до яких він отримав доступ через SSRF, він може користуватися середовищем для розповсюдження шкідливого середовища, отримання доступу до кредитних даних користувачів або баз даних, що може призвести до паралічу систем та інфраструктури, втрати контролю над системами [14, 15];

4. Ненадійні конфігурації за замовчуванням. Часто служби AWS пропонують користувачам типовий шаблон налаштувань сервісу. Налаштування за замовчуванням часто розроблені, щоб забезпечити зручність або легкість користування. Проте акцент надається саме на аспекті захищеності, що в свою чергу може призвести до порушенню даних, несанкціонованого доступу або порушенню відповідності [16];

5. Ненадійні конфігурації мережі. Під час налаштовування мережевих компонентів AWS можна допуститися помилок і піддати мережеву інфраструктуру різним ризикам. Існує багато факторів, які потрібно брати до уваги при створенні, наприклад, віртуального серверу AWS, щоб зловмисник мав якнайменше вікон для атаки. Найчастішою помилкою є відкриття непотрібних портів для публічного Інтернету, що може дати можливість отримати хакеру напряму доступ до ресурсів всередині мережі. Погано налаштовані або слабкі правила брандмауера також слугують ризиком, бо можуть дозволити неавторизованому трафіку входити і виходити з мережі. Зловмисники використовують ці прогалини, щоб обійти заходи безпеки, розпочату атаку і отримати конфіденційні дані [17].

Для виконання поставлених задач були вибрані наступні сервіси платформи AWS: Elastic Beanstalk (EB), IAM, Amazon Inspector [18]. При виборі кожної з служб враховувалися фактори доцільності, функціональності, можливості масштабування і інтеграції з іншими сервісами, а також ціни за використання послуги. Робота деяких сервісів AWS може бути тісно пов'язана із взаємодією з іншими службами. Так, наприклад, сервіс Elastic Beanstalk використовує EC2 для створення та керування віртуальними серверами, S3 для зберігання статичних файлів (зображень, файлів CSS або скриптів JavaScript), Elastic Load Balancer для розподілу трафіку між різними екземплярами віртуальних серверів, IAM для управління доступом до ресурсів та обмежень привілеїв користувачів [19].

З метою отримання більш повного функціоналу, можливостей розширеного налаштування сервісів, створення та керування ресурсами, налаштувань безпеки AWS надає інструменти командного рядка. В рамках виконання роботи були використані інструменти AWS CLI і EB CLI. AWS CLI дозволяє взаємодіяти з AWS за допомогою команд в командному рядку або створювати сценарії на основі цих команд. Він підтримує різні операційні системи і дозволяє виконувати багато функціональних дій з AWS. В свою чергу EB CLI надає зручні команди для створення, налаштування, розгортання середовищ Elastic Beanstalk. Дані пакети програмного забезпечення були завантажені та встановлені на операційну систему Windows 10 [20].

Під час виконання дослідної частини були використані можливості сервісу IAM для забезпечення належного функціонування розгорнутого веб-додатку на базі EB. Для відмежування облікового запису головного (root) користувача з робочим середовищем і процесом розбудови інфраструктури були створені наступні сутності IAM: користувач з повноваженнями адміністратора, група адміністраторів, ролі адміністраторів, сервісні ролі для EB, політики доступів. Також, деякі з сутностей, наприклад, політики доступів або ролі, були згенеровані сервісом EB. Ролі в службі IAM бувають двох типів: роль і сервісна роль. Роль може бути призначена для користувачів або груп користувачів. В свою чергу сервісна роль призначається службам AWS (EC2, Lambda тощо) для надання їм можливості виконання певних операцій. Сервісна роль надає службам повноваження, щоб вони могли взаємодіяти з іншими сервісами AWS та ресурсами в межах аккаунту. Під час створення сервісної ролі для EC2 також створюється профіль екземпляру з еквівалентним іменем ролі, що дають можливість віртуальним серверам виконувати операції від імені інших служб AWS. За профілем екземпляру також автоматично закріплюється сама роль [21].

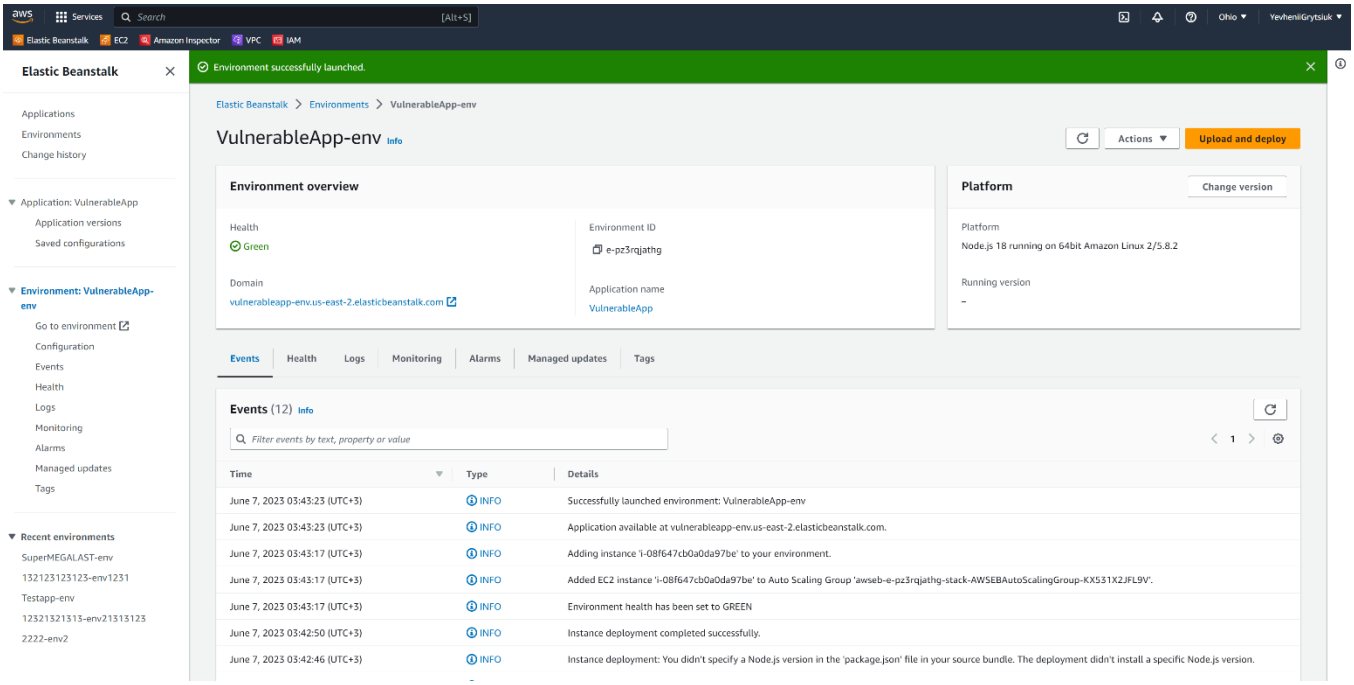

Рис. 1 Інтерфейс сторінки налаштувань і моніторингу середовища після завершення процедури розгортання додатку

В якості тестового додатку був обраний типовий веб-сайт, створений за допомогою Elastic Beanstalk на платформі Node.js. Такого типу додатки зручно використовувати якщо на меті стоїть тестування і перевірка безпеки додатку, мережі, інфраструктури, ролей і політик IAM тощо.

Одразу після запуску процедури розгортання додатку користувача переадресовують на сторінку, яка містить інформацію та налаштування для керування та моніторингу створеного середовища (рис. 1). Вкладка "Events" на сторінці середовища Elastic Beanstalk надає інформацію про події, пов'язані з цим середовищем. Вона відображає список останніх подій, які сталися або відбуваються у середовищі. Кожна подія відображається з відповідними деталями, наприклад, час і тип події, повідомлення і опис. Ця вкладка є корисним інструментом для моніторингу та відстеження подій, які відбуваються в середині середовища Elastic Beanstalk. Процес створення середовища включає наступні етапи: підготовка середовища, конфігурування середовища, запуск віртуального сервера EC2 та відповідної інфраструктури, завантаження та розгортання додатку і валідація їх роботи. Підтвердженням успішності запуску середовища слугує подія Successfully launched environment: VulnerableApp-env.

Після успішного розгортання додатку наступним кроком є активація Amazon Inspector в межах акаунту, де потрібно здійснити сканування. Після підтвердження активації, користувача переадресовує на сторінку панелі керування. На цій вкладці доступний загальний огляд стану безпеки ресурсів аккаунту. Користувач може переглянути

основні показники безпеки, такі як виявлені вразливості, рівень загрози, статус сканування EC2 сервісів. Inspector вимагає попередньо визначених налаштувань наданих користувачем у вигляді політики IAM. Служба надає можливість згенерувати типовий шаблон налаштувань поведінки роботи. Політика дозволів для ролі, яка називається AmazonInspector2ServiceRolePolicy, дозволяє Amazon Inspector виконувати наступні завдання: виконувати дії для отримання віртуальними серверами мережевих шляхів; виконувати операції з CloudWatch; взаємодіяти з IAM, щоб отримати інформацію про політики, які можуть створювати вразливі місця безпеки тощо. Одразу після ввімкнення служби і налаштувань політик, Amazon Inspector починає автоматизоване безперервне сканування вразливостей EC2 віртуальних серверів.

Amazon Inspector надає дані про загальні вразливості та ризики (Common Vulnerabilities and Exposures – CVE) для екземплярів віртуальних серверів EC2, лише за умови встановленого та активованого агента Amazon EC2 System Manager (SMM). Amazon EC2 Systems Manager надає централізоване керування та автоматизацію EC2 ресурсам, що дозволяє спростити управління та забезпечити ефективність вашого хмарного середовища AWS. До функціональних особливостей цієї служби відносяться: створювання логічних груп ресурсів, наприклад додатків, компонент додатків або робочі середовища; зміни конфігурацій ресурсів, відповідні сповіщення, операційні повідомлення, інвентаризацію програмного забезпечення та стан відповідності виправлень; дозволяє використовувати ресурси AWS для автоматизації типових і повторюваних операцій. Для завантаження та встановлення цього агента потрібно підключитися до віртуального сервера і виконати інсталяцію програмного пакету.

Аналіз результатів сканування

Сканування віртуальних серверів може зайнятий тривалий час. Процес перевірки залежить від кількості ресурсів, які потрібно перевірити, та обсягу даних, які треба проаналізувати. Для невеликих розгорнутих середовищ або окремих ресурсів сканування може зайняти лише кілька хвилин. Однак, для великих та складних інфраструктур, що включають багато ресурсів, сканування може зайняти години або навіть більше.

На вкладці "Findings" в Amazon Inspector можна побачити результати сканування на вразливості та проблеми безпеки, виявлені в розгорнутій інфраструктурі. Тут відображаються знайдені проблеми, які можуть включати вразливості, потенційні атаки, неправильну конфігурацію, відсутність необхідних заходів безпеки та інші проблеми. Користувач може фільтрувати результати за різними категоріями, такими як вразливості, конфігураційні проблеми або загрози безпеки. Також можна переглядати статус виявлень, включаючи активні, виправлені або відхилені проблеми (рис. 2).

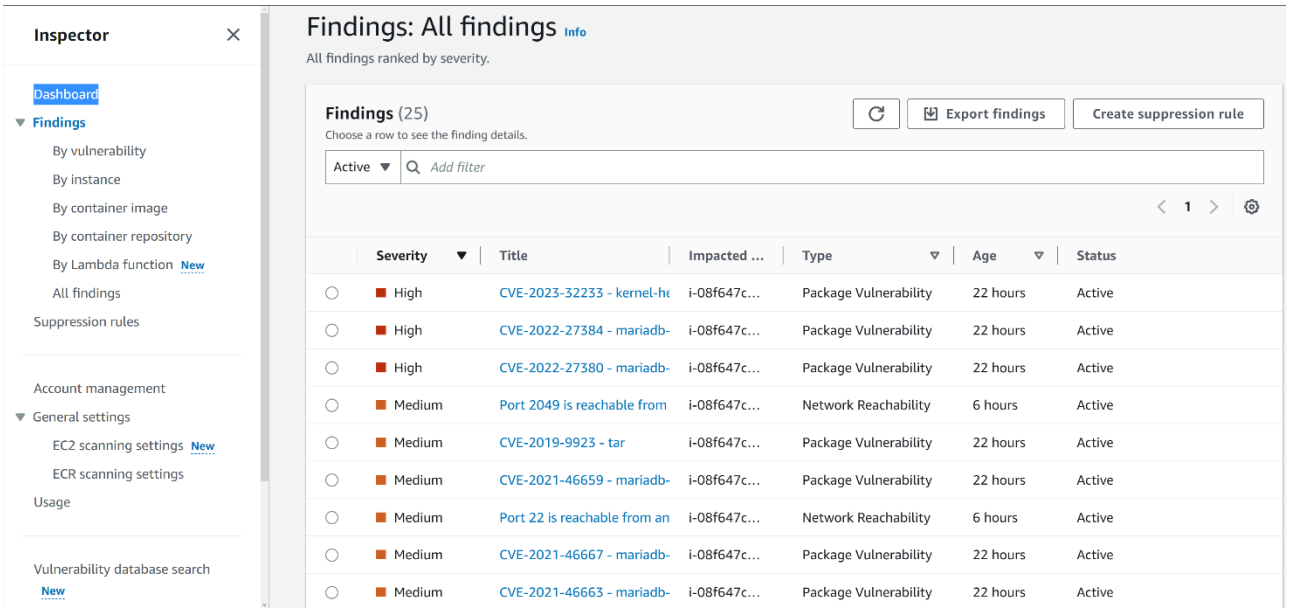

Рис. 2. Список виявлених загроз

Натиснувши на ідентифікатор загрози можна отримати детальний опис цієї загрози. Опис містить інформацію про тип загрози, її потенційні наслідки та можливі способи використання. Опис загрози також може включати технічні деталі про вразливість або проблему безпеки, включаючи

назву, CVE-ідентифікатор (якщо відомий) та опис вразливості. Користувач може дізнатись більше про цю загрозу, її вплив на безпеку системи та можливі наслідки, які вона може мати на інфраструктуру. Опис загрози також може містити рекомендації щодо виправлення проблеми. Ці рекомендації можуть включати конкретні кроки або рекомендовані заходи безпеки для вирішення виявленої проблеми. Це може включати підказки щодо конфігурації, оновлення програмного забезпечення або встановлення заходів безпеки для запобігання атакам.

Загалом, Amazon Inspector при скануванні додатку, розгорнутому за допомогою Elastic Beanstalk при скануванні віртуального сервера виявив 25 вразливостей. Більшість загроз складають вразливості програмних пакетів, використаних для створення веб-сайту – 17. Найбільше загроз середньої ступені тяжкості – 19 (табл. 1).

Кожна зі знайдених вразливостей може потенційно привести до витоку даних та порушити роботу додатку загалом. Розглянувши опис і деталі кожної з них можна оцінити суть проблеми та рівень її загрози:

- *CVE-2023-32233 – kernel-headers, kernel-tools and 1 more.* У ядрі Linux до версії 6.3.1 під час обробки пакетних запитів може бути використано для виконання довільних операцій читання та запису в пам'яті ядра. Непривілейовані локальні користувачі можуть отримати привілеї root. Це відбувається через неправильне використання анонімних наборів. Загроза стосується пакетів налаштування оперативної системи Linux;

Таблиця 1

Список знайдених вразливостей Amazon Inspector при скануванні додатку VulnerableApp

Назва загрози	**Тип загрози**	**Ступінь тяжкості**
CVE-2023-32233 – kernel-headers, kernel-tools and 1 more	Вразливість програмного пакету	Висока
CVE-2022-27384 – mariadb-libs	Вразливість програмного пакету	Висока
CVE-2022-27380 – mariadb-libs	Вразливість програмного пакету	Висока
Port 2049 is reachable from an Internet Gateway – TCP	Доступність мережі	Середня
CVE-2019-9923 – tar	Вразливість програмного пакету	Середня
CVE-2021-46659 – mariadb-libs	Вразливість програмного пакету	Середня
Port 22 is reachable from an Internet Gateway – TCP	Доступність мережі	Середня
CVE-2021-46667 – mariadb-libs	Вразливість програмного пакету	Середня
CVE-2021-46663 – mariadb-libs	Вразливість програмного пакету	Середня
CVE-2022-38090 – microcode_ctl	Вразливість програмного пакету	Середня
CVE-2022-31624 – mariadb-libs	Вразливість програмного пакету	Середня
CVE-2021-46666 – mariadb-libs	Вразливість програмного пакету	Середня
Port 5432 is reachable from an Internet Gateway – TCP	Доступність мережі	Середня
CVE-2021-46668 – mariadb-libs	Вразливість програмного пакету	Середня
CVE-2022-21216 – microcode_ctl	Вразливість програмного пакету	Середня
CVE-2021-46657 – mariadb-libs	Вразливість програмного пакету	Середня
Port 3389 is reachable from an Internet Gateway – TCP	Доступність мережі	Середня
CVE-2022-2521 – libtiff	Вразливість програмного пакету	Середня
Port 1433 is reachable from an Internet Gateway – TCP	Доступність мережі	Середня
CVE-2021-46661 – mariadb-libs	Вразливість програмного пакету	Середня
CVE-2021-3800 – glib2	Вразливість програмного пакету	Середня
Port 443 is reachable from an Internet Gateway – TCP	Доступність мережі	Низька
Port 80 is reachable from an Internet Gateway – TCP	Доступність мережі	Низька
Port 25 is reachable from an Internet Gateway – TCP	Доступність мережі	Інформаційна

- *CVE-2022-27384 – mariadb-libs.* Вразливість, пов'язана з бібліотекою mariadb-libs. Ця вразливість може мати потенційний вплив на системи, які використовують цю бібліотеку для роботи з базами даних MariaDB. Вразливість CVE-2022-27384 може дозволити зловмисникам виконати атаку віддаленого виконання коду або отримати несанкціонований доступ до системи, якщо недостатньо захищена або застаріла версія mariadb-libs використовується. Це може призвести до втрати конфіденційності, цілісності та доступності даних. Загроза стосується пакетів програмного забезпечення для роботи з базою даних MariaDB;

• *CVE-2022-27380 – mariadb-libs.* Вразливість стосується проблеми в компоненті програмного пакету mariadb-libs my_decimal::operator = MariaDB Server версії 10.6.3 і нижче, яка дозволяє зловмисникам викликати відмову в обслуговуванні (DoS) за допомогою спеціально створених інструкцій SQL. Загроза стосується пакетів програмного забезпечення для роботи з базою даних MariaDB;

• *port 2049 is reachable from an Internet Gateway – TCP.* Порт 2049, який використовує протокол TCP, доступний для з'єднання з Інтернет-шлюзом. Це може мати наслідком те, що сервіс або програма, яка використовує цей порт, може бути доступна з Інтернету. Важливо враховувати, що доступність порту 2049 з Інтернету може мати потенційні ризики з точки зору безпеки. Зловмисники можуть використовувати цей порт для сканування, атак або несанкціонованого доступу до системи. Загроза стосується налаштувань мережі сервера;

• *CVE-2019-9923 – tar.* Загроза пов'язана з уразливістю, знайденою в утиліті "tar", яка використовується для створення та обробки архівних файлів. Ця вразливість потенційно може дозволити зловмиснику перезаписати файли, що призведе до виконання довільного коду або несанкціонованого доступу до конфіденційної інформації. Уразливість виникає через недолік у тому, як утиліта «tar» обробляє символічні посилання під час розархівації. Створюючи спеціально створений архівний файл, зловмисник може маніпулювати символьними посиланнями, щоб перезаписувати довільні файли в системі. Ця вразливість має наслідки для безпеки, оскільки нею може скористатися зловмисник, маючи доступ до зловмисно створеного архівного файлу. Успішне використання цієї вразливості може призвести до зламу цільової системи, несанкціонованого доступу до конфіденційних даних або виконання шкідливого коду. Загроза стосується пакетів програмного забезпечення для роботи з архівуванням даних;

• *CVE-2021-46659 – mariadb-libs.* Вразливість виявлена в пакеті "mariadb-libs". Ця вразливість потенційно може дозволити зловмиснику виконати довільний код або спричинити стан відмови в обслуговуванні (DoS) у вразливій системі. Уразливість виникає через недолік у тому, як бібліотека "mariadb-libs" обробляє певні типи введених даних. Використовуючи цю вразливість, зловмисник може маніпулювати введенням таким чином, що викликає переповнення буфера або інші проблеми з пошкодженням пам'яті, що призводить до виконання довільного коду або збою програми. Вплив цієї вразливості може бути серйозним, оскільки вона може дозволити зловмиснику отримати неавторизований доступ до ураженої системи, порушити цілісність бази даних або порушити доступність служби. Загроза стосується пакетів програмного забезпечення для роботи з базою даних MariaDB;

• *port 22 is reachable from an Internet Gateway – TCP.* Вразливість вказує на те, що порт 22, який зазвичай асоціюється з протоколом Secure Shell (SSH), доступний через Інтернет-шлюз. Якщо порт 22 доступний через Інтернет-шлюз, це означає, що служба SSH, яка працює у відповідній системі, доступна з Інтернету. Порт 22 є стандартним портом, на якому служби SSH прослуховують вхідні з'єднання. Якщо порт 22 доступний через Інтернет-шлюз, це означає, що служба SSH у пов'язаній системі доступна для Інтернету, що потенційно робить її мішенню для несанкціонованого доступу або зловмисних дій. Загроза стосується налаштувань мережі сервера;

• *CVE-2021-46667 – mariadb-libs.* У програмних пакетах MariaDB виявлено вразливість цілочисельного переповнення, де виділено недійсний розмір ref_pointer_array. Ця проблема призводить до відмови в обслуговуванні. Загроза стосується пакетів програмного забезпечення для роботи з базою даних MariaDB;

• *CVE-2022-21216 – microcode_ctl.* Ця вразливість, пов'язана з пакетом «microcode_ctl». Мікрокод – це мікропрограма, яка встановлюється на певні процесори комп'ютера для керування їх роботою та підвищення продуктивності. У цьому конкретному CVE виявлено проблему безпеки в пакеті microcode_ctl, який використовується для оновлення мікрокоду в системах Linux. Недолік може дозволити привілейованому користувачеві потенційно дозволити розкриття інформації через локальний доступ. Загроза стосується процесу оновлень програмного пакету "microcode";

• *CVE-2022-31624 – mariadb-libs.* Сервер бази даних MariaDB до версії 10.7 вразливий до відмови в обслуговуванні. Під час виконання методу plugin/server_audit/server_audit.c log_statement_ex утримуване блокування lock_bigbuffer не знімається належним чином, що дозволяє локальним користувачам ініціювати відмову в обслуговуванні через взаємоблокування. Загроза стосується пакетів програмного забезпечення для роботи з базою даних MariaDB;

• *CVE-2022-33972 – microcode_ctl.* Неправильний розрахунок у механізмі ключа мікрокоду для деяких масштабованих процесорів Intel(R) Xeon(R) 3-го покоління може дозволити привілейованому користувачеві потенційно ввімкнути розкриття інформації через локальний доступ. Загроза стосується процесу оновлень програмного пакету "microcode";

• *CVE-2021-46666 – mariadb-libs.* Сервер бази даних MariaDB до версії 10.6.2 допускає аварійне завершення роботи програми через неправильну обробку переходу від виразу HAVING до виразу-WHERE. Загроза стосується пакетів програмного забезпечення для роботи з базою даних MariaDB;

• *port 5432 is reachable from an Internet Gateway – TCP.* Порт 5432 зазвичай асоціюється з системою керування базами даних PostgreSQL. Відкритий порт, особливо пов'язаний із системою керування базами даних PostgreSQL, може становити загрозу безпеці, якщо він доступний в Інтернеті без належних заходів безпеки. Це потенційно може призвести до несанкціонованого доступу, витоку даних або зловмисних атак, спрямованих на базу даних. Загроза стосується налаштувань мережі сервера;

• *CVE-2021-46668 – mariadb-libs.* Сервер бази даних MariaDB до версії 10.5.9 допускає збій програми через певні довгі оператори SELECT DISTINCT, які неналежним чином взаємодіють з обмеженнями ресурсів системи зберігання для тимчасових структур даних. Загроза стосується пакетів програмного забезпечення для роботи з базою даних MariaDB;

• *port 3389 is reachable from an Internet Gateway – TCP.* Порт 3389 зазвичай асоціюється з протоколом віддаленого робочого стола (RDP), який використовується для віддаленого доступу та керування системами на базі Windows. В повідомленні зазначено, що «Порт 3389 доступний через Інтернет-шлюз – TCP», це означає, що вказаний порт відкритий і доступний з Інтернету через протокол TCP. Відкритий і доступний порт потенційно може піддати систему несанкціонованому доступу або зловмисним діям. Доступ до порту 3389 через Інтернет-шлюз означає, що в системі ввімкнено RDP і доступ до неї здійснюється з Інтернету. Загроза стосується налаштувань мережі сервера;

• *port 1433 is reachable from an Internet Gateway – TCP.* Порт 1433 зазвичай асоціюється з системою керування базами даних Microsoft SQL Server. Коли в повідомленні зазначено, що вказаний порт відкритий і доступний з Інтернету через протокол TCP. З точки зору безпеки, це потенційно може піддати систему, на якій працює Microsoft SQL Server, несанкціонованому доступу або зловмисним діям, оскільки порт 1433 відкритий і доступний з Інтернету. Загроза стосується налаштувань мережі сервера.

Для кожної з виявлених вразливостей пропонуються шляхи та рекомендації щодо їх усунення:

• *програмні пакети "mariadb-libs".* Для зменшення ризиків пов'язаних з пакетом MariaDB Amazon Inspector надає рекомендації щодо оновлення встановлених пакетів програмного забезпечення до запропонованої виправленої версії кожного пакета, щоб усунути виявлені вразливості;

• *програмні пакети "microcode-ctl".* Для усунення ризиків пов'язаних з мікрокодом потібно оновити BIOS і встановлені програмні пакети програми до останньої версії;

• *відкриті порти 2049, 22, 5432, 1433, 443, 80, 25.* Для запобігання виникнення потенційних загроз від відкритих портів потрібно оцінити необхідність відкриття кожного для зовнішнього доступу до системи. Якщо виконання завдання не передбачає надання портам публічного доступу, то можливо краще обмежити їхню доступність. Робота цих портів асоціюється з програмним забезпеченням, наприклад, PostgreSQL, MSSQL, SSH тому, важливо оновити пакети програмного забезпечення цих програм для виправлення відомих вразливостей. У будь якому разі потрібно налаштувати файрвол або мережеві політики для потенційно небезпечного трафіку на цих портах. Також рекомендується встановлення додаткових заходів безпеки, такі як інтрафейсний фільтр.

ВИСНОВКИ

В процесі дослідження був проведений аналіз функціональних можливостей, особливостей конфігурації та специфіки інтеграції сервісів AWS. На основі служб Elastic Beanstalk і Amazon Inspector був розгорнутий веб-додаток і проведений процес оцінки вразливостей. Основні результати роботи можна сформулювати наступним чином:

1. Amazon Inspector надає ефективні можливості оцінки безпеки для програм, розгорнутих як на базі Elastic Beanstalk, так і на базі звичайних віртуальних серверів EC2. Його автоматизоване сканування та аналіз допомагають виявити загальні вразливості та потенційні ризики безпеки в програмі та її базовій інфраструктурі;

2. Інтеграція Amazon Inspector у життєвий цикл розробки програмного забезпечення про-

грам може значно підвищити рівень безпеки. Включивши сканування та оцінку вразливостей у процес розробки та розгортання, організації можуть завчасно виявляти та вирішувати проблеми безпеки;

3. Amazon Inspector доповнює інші підходи до тестування безпеки, надаючи додаткову інформацію та покриття, специфічне для конкретно взятого середовища. Він може виявляти вразливості, які можуть бути пропущені під час тестування вручну або при застосуванні традиційних інструментів безпеки, забезпечуючи більш комплексний захист;

4. Своєчасне усунення виявлених вразливостей має вирішальне значення для підтримки безпеки програм. Amazon Inspector надає детальні висновки та рекомендації, що дозволяє розробникам і командам DevOps визначати пріоритети та оперативно усунути вразливості.

На підставі цих висновків можна сформувати наступні рекомендації:

1. Організаціям, які використовують хмарні сервіси AWS, слід розглянути можливість використання Amazon Inspector у своїй стратегії тестування безпеки. Він пропонує цінний рівень виявлення вразливостей, який доповнює існуючі заходи безпеки;

2. Рекомендується регулярне та систематичне сканування програм за допомогою Amazon Inspector, щоб забезпечити постійний моніторинг потенційних вразливостей. Цей проактивний підхід допомагає виявити та вирішити проблеми безпеки, перш ніж їх зможе використати зловмисник;

3. Безперервне навчання та оновлення найновіших методів безпеки та рекомендацій є надзвичайно важливими. Регулярний перегляд і застосування виправлень і оновлень для стека програм допомагає усунути відомі та нові вразливості.

Підсумовуючи, використання Amazon Inspector, як нативного хмарного інструменту безпеки AWS, у поєднанні з іншими сервісами, що використовуються для розробки може значно підвищити рівень безпеки програм. Інтегруючи виявлення та оцінку вразливостей у процес розробки та розгортання, організації можуть зменшити ризики, захистити конфіденційні дані та забезпечити загальну безпеку та стійкість своїх програм.

ЛІТЕРАТУРА

[1]. Kizza, Joseph. (2024). Cloud Computing Technology and Security. 10.1007/978-3-031-47549-8_23.

[2]. Shevchuk, D., Harasymchuk, O., Partyka, A., Korshun N. Designing Secured Services for Authentication, Authorization, and Accounting of Users, Workshop on Cybersecurity Providing in Information and Telecommunication Systems II, vol. 3550, (2023), pp. 217-225.

[3]. Rayapati, Siri & Muttavarapu, Sravya & Nagasuri, Navya & Singhal, Sunita. (2023). Security in Cloud Technologies: A Brief Overview. pp. 683-695. 10. 4028/p-4pq758.

[4]. An, Seongmo & Leung, Asher & Hong, Jin & Eom, Taehoon & Park, Jong. (2022). Toward Automated Security Analysis and Enforcement for Cloud Computing Using Graphical Models for Security. IEEE Access. 10. 1-1. 10.1109/ACCESS.2022.3190545.

[5]. Solanki, Madan & Tokekar, Vrinda. (2022). Design and Implementation of Strong Security Architecture for Amazon Web Service based on Cloud Applications. International Journal of Innovative Technology and Exploring Engineering. 11. pp. 17-22. 10. 35940/ ijitee.L9324.11111222.

[6]. Dubey, Parul & Tiwari, Arvind & Raja, Rohit. (2023). Cloud Monitoring. 10.2174/9789815165821-123010011.

[7]. Bihari, Vipin & Kumar, Asutosh & Sattar, Arif & Ranjan, Mritunjay & Scholar, Research. (2023). Fortifying the Cloud: Unveiling the Next- Generation Security Model of AWS. International Journal of Innovative Research in Engineering & Multidisciplinary Physical Sciences. Volume 11. pp. 1-12. 10.37082/ IJIRMPS.v11.i3.230230.

[8]. Saiteja, Kommuri. (2022). AWS CLOUD SECURITY.

[9]. Features of Amazon Inspector [Електронний ресурс]. Режим доступу до ресурсу: https://docs.aws.amazon.com/inspector/latest/user/what-is-inspector.html.

[10]. Jayawardana, Hansaka & Uyanahewa, Madhavi & Hapugala, Venura & Thilakarathne, Thiyuni. (2023). An Analysis of XSS Vulnerabilities and Prevention of XSS Attacks in Web Applications. 10.13140/ RG.2.2.21854.00321.

[11]. Amazon: Security Vulnerabilities, CVEs [Електронний ресурс]. Режим доступу до ресурсу: https://www.cvedetails.com/vulnerability-list/vendor_id-12-126/Amazon.html.

[12]. AWS S3 Security Best Practices [Електронний ресурс]. Режим доступу до ресурсу: https://www.-wiz.io/academy/s3-bucket-security-risks-and-best-practices.

[13]. Intro: AWS Privilege Escalation Vulnerabilities [Електронний ресурс]. Режим доступу до ресурсу: https://rhinosecuritylabs.com/aws/aws-privilege-escalation-methods-mitigation.

[14]. Exploit SSRF to gain AWS Credentials [Електронний ресурс]. Режим доступу до ресурсу: https: //scalesec.com/blog/exploit-ssrf-to-gain-aws-credentials.

[15]. Server Side Request Forgery (SSRF) and AWS EC2 instances after Instance Meta Data Service version 2(IMDSv2) [Електронний ресурс]. Режим доступу

до ресурсу: https://blog.appsecco.com/server-side-request-forgery-ssrf-and-aws-ec2-instances-after-instance-meta-data-service-version-38fc1ba1a28a.
[16]. Vakhula, O., Opirskyy, I., Mykhaylova, O. Research on Security Challenges in Cloud Environments and Solutions based on the security-as-Code Approach, Workshop on Cybersecurity Providing in Information and Telecommunication Systems II, vol. 3550, (2023) 55-69.
[17]. Kolb, Tobias. (2023). Development and evaluation of a cloud security testing framework for penetration testing and red teaming of AWS cloud environments.
[18]. Dubey, Parul & Tiwari, Arvind & Raja, Rohit. (2023). Introduction To Cloud Computing and AWS. 10.2174/9789815165821123010002.
[19]. Gandhi, Raj & Shahji, Vivek & Kamble, Nitin. (2021). Access Control Model Based on AWS IAM. International Journal of Innovative Research in Computer and Communication Engineering. 9. 14508. 10.15680/IJIRCCE.2021.0911024.
[20]. Dubey, Parul & Tiwari, Arvind & Raja, Rohit. (2023). Cloud Integrations. 10.2174/9789815165821-123010010.
[21]. Dubey, Parul & Tiwari, Arvind & Raja, Rohit. (2023). Identity and Access Management in AWS. 10.2174/9789815165821123010003.

FEATURES OF USING AMAZON INSPECTOR TO IDENTIFY VULNERABILITIES OF CLOUD APPLICATIONS

Vulnerability to various cyber-attacks, loss of data confidentiality, increased number of failures and reduced stability of information infrastructure, increased capital costs, new requirements for data independence, problems with scaling business information infrastructure can be the main problems that a business may face. The above-mentioned problems can serve as a basis for migration to cloud technologies, which in turn will ensure a reduction in expenses for infrastructure support, increase the efficiency of information infrastructure management compared to work in a local environment, and increase the flexibility of the organization. The relevance of the research lies in improving information security, ensuring confidentiality, integrity and availability, identifying application and environment vulnerabilities through the use of built-in AWS services. The purpose of this work is to implement the evaluation and improvement of the security of the working environment and the application deployed on the basis of cloud services by automating the scanning and analysis of the AWS workload.

Keywords: Amazon Web Services, AWS, Amazon Inspector, IAM, cloud technologies, vulnerability, infrastructure, monitoring.

Партика Андрій Ігорович, к.т.н., старший викладач кафедри захисту інформації Національного університету «Львівська політехніка».

Andrii Partyka, Ph.D., Senior Lecturer the Department of Information Security, Lviv Polytechnic National University.
E-mail: andrijp14@gmail.com.
Orcid ID: 0000-0003-3037-8373.

Недодус Богдан Ігорович, студент, спеціальності 125 Кібербезпека Національного університету «Львівська політехніка».

Bohdan Nedodus, student the Department of Information Security, Lviv Polytechnic National University.
E-mail: nedodusbohdan@gmail.com.
Orcid ID: 0009-0007-3822-5829.

DOI: 10.18372/2410-7840.26.18841
УДК 004.77

METHOD OF ANALYSIS OF OUTGOING TRAFFIC PACKAGE SIGNATURES

Nataliia Petliak, Yuliia Khokhlachova

To detect outgoing malicious traffic, a method based on fuzzy logical inference has been developed to analyze signatures of outgoing traffic. The study results indicate that continuing activities in this direction are worthwhile to unload network resources during peak loads. The method verifies the signature of the outgoing traffic packet against a set of rules. The key tasks of the method are connection permission, if the packet signature is defined as permitted during classification; blocking the connection, if it is determined that the signature of the package is prohibited; and adding new signatures to existing dictionaries. During the experiment, the method confirmed its effectiveness. Having a method based on fuzzy logic for signature analysis of outgoing traffic packets has several advantages, including the detection of previously unknown attacks, reduction of the total number of cyber-attacks, prevention of overloading of network equipment, and reduction of the probability of compromise. current network.

***Keywords**: fuzzy logic, signature analysis, outgoing traffic, signature classification.*

RELEVANCE AND PROBLEM STATEMENT

The rapid growth of the number of users of digital technologies leads to an increase in the number of cyber incidents and cyber-attacks on various spheres of activity. Any attack can cause not only significant financial losses but also reputational losses for a certain person, company, or even the state.