МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНЕ НЕКОМЕРЦІЙНЕ ПІДПРИЄМСТВО
«ДЕРЖАВНИЙ УНІВЕРСИТЕТ
«КИЇВСЬКИЙ АВІАЦІЙНИЙ ІНСТИТУТ»
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА ТЕХНОЛОГІЙ
КАФЕДРА КІБЕРБЕЗПЕКИ

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри кібербезпеки

_____Анна ІЛЬЄНКО
"_____"_____ 20__ р.

КВАЛІФІКАЦІЙНА РОБОТА

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ЗДОБУВАЧА ОСВІТНЬОГО СТУПЕНЯ "МАГІСТР"

Тема: Система оцінки рівня небезпек в кібероточенні на основі спеціалізованих баз даних

Виконавець: Олеся КОТЧЕНКО

Керівник: к.т.н. Олена ВИСОЦЬКА

Нормоконтролер: к.т.н., доцент Андрій ПЕТРЕНКО

Київ 2024

ДЕРЖАВНЕ НЕКОМЕРЦІЙНЕ ПІДПРИЄМСТВО
«ДЕРЖАВНИЙ УНІВЕРСИТЕТ
«КИЇВСЬКИЙ АВІАЦІЙНИЙ ІНСТИТУТ»

Факультет комп'ютерних наук та технологій
Кафедра кібербезпеки
Освітній ступінь магістр
Спеціальність 125 «Кібербезпека та захист інформації»
Освітньо-професійна програма «Безпека інформаційних і комунікаційних систем»

ЗАТВЕРДЖУЮ
Завідувач кафедри кібербезпеки

_____Анна ІЛЬЄНКО
«30»___08_____2024 р.

ЗАВДАННЯ
на виконання кваліфікаційної роботи
КОТЧЕНКО Олесі Віталіївни

1. Тема кваліфікаційної роботи: «Система оцінки рівня небезпек в кібероточенні на основі спеціалізованих баз даних» затверджена наказом ректора від 30.08.2024 р. №1696/ст.
2. Термін виконання роботи: з 30.08.2024 по 15.12.2024
3. Вихідні дані до роботи: проаналізувати існуючі системи та методики оцінки рівня небезпек; на основі аналізу виділити вхідні і вихідні параметри, завдяки яким можливо провести порівняння існуючих систем, виявлення їх переваг і недоліків; розробити методику, алгоритм та програмне забезпечення системи оцінки рівня небезпек, провести тестування розробленого програмного застосунку оцінки рівня небезпек критичних даних.
4. Зміст пояснювальної записки: аналіз існуючих систем та методик аналізу і оцінки рівня небезпек; розробка методики системи аналізу та оцінки рівня

небезпек на основі спеціалізованих баз даних; розробка програмного забезпечення запропонованої системи, верифікація отриманих результатів.

5. Перелік обов'язкового графічного (ілюстративного) матеріалу: презентація

6. Календарний план-графік

№ з/п	Завдання	Термін виконання	Підпис керівника
1.	Дослідити сучасні системі і методики аналізу та оцінки рівня небезпек.	30.08.2024 – 05.09.2024	*Виконано*
2.	Обґрунтувати вибір рішення.	06.09.2024 – 15.09.2024	*Виконано*
3.	Розробити методики та структури системи аналізу та оцінки рівня небезпек.	16.09.2024 – 25.09.2024	*Виконано*
4.	Розробити алгоритм та програмне забезпечення системи аналізу та оцінки рівня небезпек.	26.09.2024 – 02.10.2024	*Виконано*
5.	Апробація роботи на науково-технічній конференції.	03.10.2024 – 25.10.2024	*Виконано*

7. Дата видачі завдання: «30» __08__ 2024 р.

Керівник кваліфікаційної роботи: _____ Олена ВИСОЦЬКА
(підпис керівника) (П.І.Б.)

Завдання прийняв до виконання: _____ Олеся КОТЧЕНКО
(підпис здобувача вищої освіти) (П.І.Б.)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи «Система оцінки рівня небезпек в кібероточенні на основі спеціалізованих баз даних»: 125 с., 29 рис., 5 табл., 29 літературних джерела.

Об’єкт дослідження: процес оцінки рівня небезпек.

Предмет дослідження: методи оцінки рівня небезпек для користувачів нетехнічних спеціальностей.

Мета кваліфікаційної роботи: розробити систему оцінки рівня небезпек в кібероточенні на основі спеціалізованих баз даних.

Методи дослідження: комп’ютерне моделювання, статичний та багатофакторний аналіз, теорія вірогідності.

Практична цінність: розроблено систему для оцінки рівня небезпек критичних даних корпоративних застосунків, яка за рахунок фіксації фокусної бази вразливостей, окрім основної функції також надає доказову базу на випадок інциденту кібербезпеки. Рівень розробки системи дозволяє її застосування фахівцями нетехнічних спеціальностей.

Наукова новизна: вдосконалено метод оцінки рівня небезпек на основі використання попереднього відбору небезпек для аналізу, що дозволило покращити метрику часу для цільового аналізу пропорційному обмеженню кількості вразливостей.

Практичне значення отриманих результатів. Розроблено систему для оцінки рівня небезпек критичних даних корпоративних застосунків, який може бути застосований фахівцями нетехнічних спеціальностей.

Результати кваліфікаційної роботи рекомендується використовувати користувачами нетехнічних спеціальностей, які не мають необхідних технічних знань та/або технічних ресурсів, для здійснення оцінки рівня небезпек.

ОЦІНКА РІВНЯ НЕБЕЗПЕК, КОНТРОЛЬ КРИТИЧНИХ ДАНИХ, КІБЕРЗАХИСТ.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

OWASP – Open Web Application Security Project

NIST – Національній інститут стандартів і технологій США

CWE – Common Weakness Enumeration

CVE – Common Vulnerabilities and Exposures

NVD – National Vulnerability Database

ВСТУП

Актуальність теми. Розвиток інформаційних технологій забезпечує прогрес в житті та виробничій діяльності. Але крім переваг він має і недоліки. Інформація в сучасному світі - це дорогий та критично важливий товар. Втрата інформації, її спотворення або несанкціоноване розповсюдження може привести до значних фінансових, а інколи людських втрат. Тому постійне вдосконалення систем захисту є постійним неперервним процесом, але виникає питання напряму цього процесу. Тому актуальним є оцінка рівня небезпек критичних даних корпоративних застосунків та вдосконалення методів, які дозволяють її отримати.

Мета і завдання виконання кваліфікаційної роботи. Мета роботи: розробити систему оцінки рівня небезпек в кібероточенні на основі спеціалізованих баз даних.

Для досягнення поставленої мети необхідно розв'язати наступні **задачі**:

1 Проаналізувати існуючі методи оцінки рівня небезпек критичних даних корпоративних застосунків та, на основі результату проведеного аналізу, обрати методи для вирішення задачі оцінки рівня небезпек для критичних інформаційних ресурсів;
2 Розробити систему оцінки рівня небезпек критичних даних корпоративних застосунків на основі обраних методів, що забезпечить організацію вчасного кіберзахисту відповідних ресурсів;
3 Провести тестування розробленої системи оцінки рівня небезпек критичних даних корпоративних застосунків, що дасть змогу дослідити доцільність використання розробленої системи для вирішення поставленої задачі.

Об’єкт дослідження – процес оцінки рівня небезпек.

Предмет дослідження – методи оцінки рівня небезпек для користувачів нетехнічних спеціальностей.

Новизна одержаних результатів полягає в наступному: вдосконалено метод оцінки рівня небезпек на основі використання попереднього відбору

небезпек для аналізу, що дозволило покращити метрику часу для цільового аналізу пропорційному обмеженню кількості вразливостей.

Практичне значення отриманих результатів. Розроблено систему для оцінки рівня небезпек критичних даних корпоративних застосунків, яка за рахунок фіксації фокусної бази вразливостей, окрім основної функції також надає доказову базу на випадок інциденту кібербезпеки. Рівень розробки системи дозволяє її застосування фахівцями нетехнічних спеціальностей.

Особистий внесок здобувача вищої освіти. Всі результати отримані в ході виконання кваліфікаційної роботи отримані автором самостійно.

Апробація отриманих результатів. Основні положення роботи доповідалися та обговорювалися на міжнародній науково-практичній конференції: Innovations and New Directions in Scientific Research: Proceedings of the International Scientific Conference (2024, October 14). Manchester, UK: Bookmundo.

Публікації. Одні матеріали конференції :

Olesya Kotchenko, Olena Vysotska. Analysis of advantages and disadvantages of harm assessment tools.Innovations and New Directions in Scientific Research: Proceedings of the International Scientific Conference (2024, October 14). Manchester, UK: Bookmundo...p.p. 113-114.

РОЗДІЛ 1
ОЦІНКА РІВНЯ НЕБЕЗПЕК КРИТИЧНИХ ДАНИХ КОРПОРАТИВНИХ ЗАСТОСУНКІВ

Методи захисту критичних даних в паралельних високопродуктивних середовищах мають складну розгалужену архітектуру [1]. До переліку методів захисту відносяться:

1) методи захисту та адміністрування операційної системи:

методи ідентифікації та автентифікації;

методи керування доступом;

методи моніторингу та аудиту;

методи контролю цілісності;

методи мережевої безпеки;

методи виконавчої системи;

методи конфігурування;

методи само тестування;

2) методи захисту (сервіси безпеки) проміжного програмного забезпечення:

методи ідентифікації та автентифікації;

методи керування доступом;

методи моніторингу та аудиту;

методи мережевої безпеки;

методи виконавчої системи;

Методи захисту операційної системи реалізують:

ідентифікацію та автентифікацію на рівні ідентифікатора користувача та пароля;

настроювання парольної політики;

розмежування доступу до об’єктів файлової системи на рівні власника, груп користувачів та списку доступу об’єктів;

забезпечення можливості доступу до захищених ресурсів тільки авторизованим користувачам;

забезпечення автоматичного захисту ресурсу при його створенні;

реєстрацію подій, що відбуваються в системі;

резервне копіювання та відновлення у випадку збоїв;

фільтрацію пакетів;

адміністрування системи тільки суперкористувачем;

контроль роботи системи з реєстрацією помилок, що виникають, та сповіщення адміністратора у разі їх виникнення;

перевірка цілісності критичних компонентів системи.

Методи захисту (сервіси безпеки) проміжного програмного забезпечення забезпечують:

управління доступом до ресурсів;

автентифікацію користувачів;

однократну реєстрацію користувачів з делегуванням повноважень ґрід-сервісам шляхом створення користувачем проксі-сертифікату, яким підписується завдання;

авторизацію;

автентифікацію ґрід-сайту та його сервісів (кожен сервіс ґрід-сайту має свій сертифікат);

конфіденційність та цілісність при обміні.

На рис.1.1 наведена таксономія методів захисту критичних даних в паралельних високопродуктивних середовищах.

Рис. 1.1 Таксономія методів захисту критичних даних в паралельних високопродуктивних середовищах

1.1.Інформаційні ризики та шкода

Інформаційні ризики мають зв'язок із процесами створення, передачі, зберігання і використання інформації в коп’ютерних та телекомунікаційних системах, а також в каналах зв'язку [1]. Інформаційні ризики - це наявність небезпеки виникнення непередбачених втрат в наслідок використання підприємством інформаційних технологій.

Класифікувати інформаційні ризики можна за ознаками:

-порушень при реалізації ризику властивостями інформаційних ресурсів (конфіденційність, цілісність, доступність, спостережність, автентичність);

-порушень контрактів/законодавства;

-загроз здоров’ю та життю;

-шкод продуктивності;

- збитку внаслідок виповнення ризику, а саме:

-фінансові збитки;

-репутаційні збитки.

З приведених вище видів збитку більш меньш точно можуть бути оцінені лише фінансові втрати. У чисельному вимірі можуть бути оцінені збитки від шкоди продуктивності персоналу та порушення контрактів/законодавства. Це визначено навностью крим прямих фінансових непрями наприклад втрата ліцензії або посади воні зрозуміли але важко монетуються. В очевидь репутаційні збитки та загроза здоров'ю і життю не мають прямого фінансового еквіваленту, вони більші, але наскільки, це питання індівидуально.

Також не може бути оцінена точно ймовірність реалізації загроз, вона носить як зрозуміло ймовірностний характер. Першоджерелами інформації про ймовірність реалізації ризиків можуть бути дані щодо таких самих або схожих, випадків від команд реагування на комп'ютерні надзвичайні події, державних органів, галузевих асоціацій. Проте всі ці дані мають недоліки та носять більш рекомендательний характер, тощо; вони по-перше, є неповними, по-друге неточними , по-третє не зовсім підходять, та на останнє невчасні. Саме тому, при оцінці ризиків статистичні дані про інциденти ІБ, які вже відбулись, приймаються до відома, але використовуються із обережністю.

Відсутність точних чисельних оцінок ймовірністі та збитків унеможливлює чисельну оцінку ризику. Тому, замість, в цьому випадку, використовують методології оцінювання ризиків, які ґрунтуються на якісних показниках. Це може бути методи, які передбачають оцінювання рівня збитку та ймовірності реалізації ризику за якісною шкалою (малий/середній/великий), а на їх основі — рівня ризику за таблицею. Приклади таких методів приведені у

методології оцінки ризиків OWASP та Керівництві з проведення оцінювання ризиків NIST SP 800-30 [4]. Ці методі досі схожі відрізняється тільки за кількостью рівнів оцінювання та визначеннях правил віднесення рівня збитку.

Існують більш складні методології [2-3], які додатково враховують вплив взаємозв'язків інформаційних ресурсів, вже наявні застосунки захисту, використовують певні бази даних вразливостей [6-8].

Головним результатом застосування цих методологій є перелік ризиків та, інколі, рекомендації з їх обробки з видповідної таксономієй.

Приклад переліку методів обробки загроз [8] :

1. протидія загрозам;
2. уникнення ризиків;
3. перенесення відповідних бізнес-ризиків на інші сторони;
4. прийняття ризиків.

Це класична четвірка поводження з ризиками.

Отже, в результаті оцінивши ризики інформаційної безпеки можна отримати оцінку шкоди інформаційній безпеці і визначити пріоритетні напрямки розвитку та покращення інформаційної безпеки підприємства.

Оцінка шкоди від реалізації ризику інформаційної безпеки є важливим етапом у керуванні безпекою та прийнятті рішень щодо заходів для мінімізації цього ризику. Більша частина методологій та стандартів притримується наступних кроків:

1. Ідентифікація потенційних загроз і ризиків: розгляньте всі можливі загрози та ризики, які можуть виникнути в контексті конкретної ситуації. Це може бути втрата конфіденційності, цілісності, доступності даних або порушення їх достовірності.
2. Визначення потенційних вразливостей та слабких місць: оцініть, які аспекти системи, процесів чи інфраструктури можуть бути уразливі до цих загроз.

3. Оцінка ймовірності виникнення ризику: спробуйте визначити ймовірність того, що загроза матиме негативні наслідки. Використовуйте наявні дані, статистику та експертні оцінки.
4. Визначення потенційних наслідків: розгляньте можливі наслідки реалізації ризику. Це може включати фінансові втрати, втрату репутації, порушення законодавства, втрату клієнтів тощо.
5. Кількісна або якісна оцінка шкоди: Цей етап може виконуватися у форматікількісних оцінок (у грошовому еквіваленті) або якісних оцінок (наприклад, низький, середній, високий рівень шкоди).
6. Прийняття рішення про прийняття ризику або запобігання: на основі оцінки шкоди, оцінки ймовірності та інших факторів, необхідно вирішити, чи прийняти ризик чи прийняти заходи для його мінімізації. Це може включати впровадження додаткових заходів безпеки, перенесення ризику (наприклад, за допомогою страхування) або прийняття ризику з відомими обмеженнями.
7. Моніторинг та аналіз результатів: після прийняття рішення, важливослідкувати за ситуацією, аналізувати наслідки та, за необхідності, коригувати стратегії безпеки.

Оцінка шкоди від реалізації ризику є складним завданням, яке вимагає аналізу багатьох факторів та врахування специфіки конкретної ситуації. Важливо мати уважний підхід та використовувати наявну інформацію та експертні оцінки для прийняття обґрунтованих рішень. Також слід пам'ятати, що можливість точно визначити всі можливі наслідки та їхній кількісний вимір може бути обмеженою, і у цьому випадку застосовуються якісні оцінки.

1.2. Теоретичні основи оцінки експертом

Експертну оцінку здійснюють експерти, тобто перекладаючі з латині досвідчені особи. А сама експертна оцінка це процедура визначення кількісних або якісних характеристик процесу чи явища. Експертна оцінка будується

шляхом використання знань і досвіду та надають у вигляді висновку, описової інтерпретації, ранжирування, рейтингу. Сферою застосування експертної оцінки є невізначиность процесу чи явища через методи точних наук [16] .

При аналізі ризиків кібербезпеки, експертна оцінка дозволяє враховувати досвід та знання експертів у процесі визначення ймовірності та ранжування можливих загроз.

Основні принципи експертної оцінки: визначення експертів; завдання; збір даних та відомостей; оцінка експертів; агрегація думок експертів; аналіз та використання результатів.

1. Визначення експертів: Експертами можуть бути фахівці з кібербезпеки, аудитори, адміністратори мереж та інші особи з відповідними знаннями та досвідом у сфері кібербезпеки. Приорітетом вибору є фахові знання в фокусної сфері застосування аналізу.
2. Визначення завдання: Процес формулювання формальноповного завдання для експертної оцінки. Визначення фокусних аспектів аналізу є половіною успиху аналізу.
3. Збір даних та відомостей: Інформація про систему, процес або ситуацію, що аналізується має мати обектівний неупереджений характер. По можлівости вона має бути обезличена.
4. Оцінка експертів: Експерти використовують різні методи оцінки, такі як анкетування, сценарійні аналізи, дельфійський метод тощо. Кожному із них притоманні свої риси, але узагалі експертів частіше просять визначити ймовірність та важливість різних аспектів ризику.
5. Агрегація думок експертів: Зазвичай, результати оцінки експертів об’єднуються, можливо за допомогою математичних методів, формуючі сукупну оцінку.
6. Аналіз та використання результатів: Прийняття рішень щодо управління ризиками кібербезпеки є головним результатом аналізу. Визначення пріоритетів у впровадженні заходів безпеки та розробка стратегії

реагування на потенційні інциденти - це головні шляхи використання результатів.

Експертна оцінка в першу чергу дозволяє не тільки враховувати кількісні аспекти ризику, а й кваліфіковані думки фахівців, що є важливим в аналізі кібербезпеки, оскільки вона часто має суб'єктивний характер і рішення може бути ефективним тільки при наявности відповідного досвіду експерта.

Експертна оцінка є важливим інструментом в прийнятті рішень, особливо коли інформація є неповною або суб'єктивною. Відомий факт, що експертна оцінка має свої переваги і недоліки. Розглянемо їх більш детально:

Переваги експертної оцінки:

1. Експерти часто мають глибокі та специфічні знання у своїй галузі, що може бути важливим у процесі прийняття рішень.
2. Експертна оцінка дозволяє враховувати суб'єктивні фактори та експертні думки, які можуть бути важливими в умовах невизначеності.
3. Експертна оцінка може бути виконана швидше, якщо порівнювати з іншими методами оцінки, що особливо важливо в ситуаціях, коли потрібно швидко приймати рішення.
4. Методи експертної оцінки можна адаптувати до різних ситуацій та галузей, що робить їх універсальними.

Недоліки експертної оцінки:

1. Результати експертної оцінки можуть бути суб'єктивними та залежати від особистих думок та досвіду експертів що може привести до помилкових висновків.
2. Експерти можуть недооцінювати або переоцінювати ризики через обмеженість свого досвіду або вплив особистих упереджень що веде до размитости результатів аналізу.
3. Експерти можуть мати особисті або професійні інтереси, які можуть впливати на їхні оцінки. Наприклад експерт може лобіювати використання свого методу захисту.

4. Витрата часу і ресурсів на визначення та залучення відповідних експертів для оцінки. Добрий експерт добре коштує та має шильний графік, що може бути критічно в деяких ситуаціях.
5. Експерти можуть мати обмежену здатність передбачити деякі надзвичайні ситуації або нові загрози. Нове це нове і це об’єктивно, для нього досвіду немає.

Три складові визначають успіх використання експертної оцінки: це вибір експертів саме того фаху та квалифікації, коректне визначення мети та проміжних завдань для оцінки та останнє, але не менш важливе - інтерпретація та використання отриманих результатів. Комплексна стратегії оцінки ризиків є добрий шлях використання експертної оцінкі.

1.3. Управління ризиками основні поняття та тенденції

Існує поняття ри́зик-ме́неджменту — це система управління ризиками, яка включає тактику та стратегію управління, спрямовану на виконання основних бізнес-цілей підприємства.

Ризик-менеджмент включає:

- систему управління;
- систему ідентифікації і калькуляції;
- систему моніторингу.

Сучасна наука визначає ризик як вірогідну подію, в результаті реализації якої можуть виникати, як позитивні, нейтральні та негативні наслідки. Поняття спекулятивного ризику допускає існування як позитивних, так і негативних наслідків. В випадку негативних наслідків, або їх відсутні взагалі, такий ризик іменується чистим.

Інструментарій [9] ризик-менеджменту включає правові, політичні, організаційні, економічні, соціальні інструменти. Ризик-менеджмент як система дозволяє одночасно застосовувати декілька методів та інструментів ризик-управління.

Найбільш відомі це метод відмови, метод зниження, метод ухвалення, метод передачі та страхування. Це відповидає найбільш сприятлівим стратегіям поводження з ризиками. Та сприянню відмови від надмірно ризикової діяльності, профілактики або диверсифікації, формуванню резервів або запасів , аутсорсингу витратних ризикових методів. Розглянемо методи оцінки ризиків.

1.3.1. Метод кількісної оцінки ризиків

Є базою для отримання кількісної оцінки ризиків. Є дійсним, якщо існує можливість виразити ризик в будь-якій одиниці виміру, наприклад, інформаційних ресурсах, відсотках або коштах. Використання методу дає змогу отримати певне значення об’єктові, якому присвоюють кількісний показник [10]. Роль оцінюваного об’єкта часто грає в потенційна вразлівость чи втрата з огляду на значимість активу. Є також інші варіанти.

1. Встановлюється цінність інформації, де завжди наявна реальна вартість, що оцінена в коштах.
2. Дається оцінка ймовірним втратам від можливої загрози, пов'язаним з певними інформаційними активами. За наявності множини загроз, окремо дається оцінка кожній з них відповідно.
3. Відсоток ймовірності настання окремої загрози визначається за рекомендованою інформацією.
4. Визначення можливої шкоди через ризик ІБ протягом певного періоду часу. В даній ситуації найбільш розповсюджено просте визначення: одинична втрата, помножена на частоту настання ризику [11].

Останнім кроком є аналіз отриманих даних з приводу оцінки можливої шкоди від кожного ризику. Ухвалюється одне з трьох рішень:

1. Погодитись з ризиком. Це рішення діє тільки тоді колі вітрати на інші рішення перевищують можливі втрати. Частіше це випадок, якщо можливість ризику прагне до нуля, а шкода мінімальна.

2. Зменшення загрози. Створити сукупність засобів і знайти способи здійснення додаткового захисту. Це не саме оптимальне рішення, компанії точно матимуть збитки, так як є значна необхідність закупки та розміщення методів захисту. Якщо ціна ІзОД вища за потенційну шкоду, необхідно враховувати іншу пропозицію.
3. Переміщення ризику. Тобто передати його третім особам, роль яких часто займають страхові компанії.

Перелік методів оцінки інформаційних активів з вираженням в коштах наведено нижче:

- із застосуванням табличних даних – експертний метод;
- метод шляхом проведення розрахувань ймовірних втрат ALE (Annualised Loss Expectency);

Ймовірні грошові збитки містять у собі такі фактори:

- втрачена вигода (чи отримана не в повній мірі);
- ціна додаткового обслуговування обладнання;
- зниження результативності.

1.3.2. Метод якісної оцінки ризиків

Якісний аналіз ризику виконує 3 функції:

1. Розставляє пріоритети ризиків відповідно до ймовірності та впливу.
2. Визначає основні зони ризику.
3. Покращує розуміння ризиків підприємства.

Кількісний аналіз ризиків не завжди можливий або оптимальний з точки зору грошей. Доволі поширена ситуація, в яких ресурси, витрачені на зменшення ризику, насправді переважають сам ризик. Тому поширений якісний аналіз, однією з основних цілей якісного аналізу ризиків є визначення пріоритетів ризиків на основі їх ймовірності та впливу. Або якісний аналіз ризиків також може покращити розуміння ризиків керівництву підприємства. Для цього не

завжди потрібна кількостна оцінка. Якісний аналіз ризику може складатися з 4х кроків:

Ідентифікація ризику; Аналіз впливу; Усунення ризику; Огляд і моніторинг.

1. Перший крок - Ідентифікація ризику - найважливіша частина якісного аналізу ризику. Це найбільш важлива стадія не ідентифікований риск невидимий для системи та не отримує ніякої противодії до моменту осознання його існування. Якщо не визначити ризики завчасно, керувати ними стане надзвичайно складно або взагалі неможливо. Коректно побудована таксономія ризиків дозволяє розподілити ресурси протідії та сконцентроватися на більш значимих.
2. Другий - Аналіз впливу – наприклад експертно оцінюють вплив кожного ризику за шкалою (1-5 або низький/середній/високий/екстремальний). Далі оцінюють ймовірність виникнення кожного ризику, використовуючи аналогічну шкалу. Отримані бали об’єднують, щоб створити загальний рейтинг ризику.
3. Третій - Усунення ризику - процес вчинення заходів щодо ризику не завжди можливе, але важливе адекватне реагування на загрозу.
4. Огляд і моніторинг — процес повторення попередних кроків для повторної оцінки ризиків та виявлення помилок минулого оцінювання [12].

1.3.3 Комбінований метод

Комбіновані методи пов’язують кількісну на якісну оцінку також можуть використовуюти шкали оцінки впливу, ймовірність, та кількісний показник загроз.

Основою таких методів є оцінка інформаційного активу. Інформаційним активом може бути база даних, державний реєстр, електронна бібліотека даже архів повідомлень. Головна особливость інформаційного активу – його не матеріальність, тому його для аналізу потрібно переформувати для преставлення

у вигляді об’єкту зі своїми правилами збору та зберегання. Обсяг та зміст визначається з потреб та цілей аналізу з усього спектру інформації, залученої в компанії. В процесі оцінки мають приймати широке коло експертів, до складу якого входять особи, які беруть участь бізнес-процесах - менеджери та висококваліфіковані фахівці, які можуть визначити, на якому етапі інформація використовується як цінний ресурс. Якість результатів залежіть від компетентності та професійного досвіду фахівців.

Іншою більш складною проблемою є визначення цінності активу інформації та його вираження в кількостному вираженні. Оцінити його числову оцінку може лише власник інформаційного активу або інша особа, за умовою що вона отримує від інформаційного активу дохід.

Для оцінки використовуються різні подходи. Найочевидним є варіант розрахуноку трудовитрат на генерацію, ідентифікацію, обробку та збереження кожної одиниці отриманої цінної інформації для визначення її вартості. Це може бути добуток часу, який працівник витрачає на отримання цієї інформації, помножений на середню годинну ставку. Однак цей підхід не працює для поточних або придбаних іншим шляхом активів. Більш перспективен підход це комплексна оцінка витрат, що враховує багато факторів, зокрема вартість отримання інформації, вартість обробки та зберігання інформації за допомогою інформаційних технологій.

Інша проблема, інформаційні активи дуже динамічні, термін корисного використання інформації дуже невизначений через швидку зміну вартості з втратою актуальності, прикладом швідкої зміни є вартісь інформації про переможця до та після змагання. Перераховані вище фактори вимагають періодичної переоцінки інформаційних активів, та оцінка інформаційних активів, створена тільки на початку та в кінці року, не обов’язково відображають реальну ситуацію.

1.4 Аналіз засобів оцінки шкоди

1.4.1 Матриця ризиків

Матриця ризиків часто використовується під час оцінки ризику для вимірювання рівня ризику, враховуючи наслідки/важкість та ймовірність реалізації ризику. Ці два показники можуть допомогти визначити загальний рейтинг ризику небезпеки. Під час використання матриці ризику слід поставити два ключових питання: наслідки: які воні та ймовірність: чі буде він реализований?

Найпоширенішими типами є матриця ризиків розміром 3 × 3, матриця ризиків 4 × 4 і матриця ризиків 5 × 5, що наведено в табл. 1.1. [13].

Таблиця 1.1

Матриця ризиків

Ймовірність		**Дуже ймовірно**	**Ймовірно**	**Малоймовірно**	**Майже неймовірно**
Наслідки	Фатальний вплив	Висока	Висока	Висока	Середня
	Великий вплив	Висока	Висока	Середня	Середня
	Малий вплив	Висока	Середня	Середня	Мала

1.4.2 Аналіз властівостей NIST Cybersecurity Framework

Класичне визначення NIST Cybersecurity Framework — це комплексний набір інструкцій і практик, розроблених, щоб допомогти організаціям керувати ризиками кібербезпеки та зменшувати їх [31]. Цей набір був розроблений Національним інститутом стандартів і технологій (NIST) та федеральним

агентством Міністерства торгівлі США. Інформаційний ресурс забезпечує структурований підхід для організацій, щоб підвищити рівень кібербезпеки та підвищити стійкість до кіберзагроз.

Методологія включає сім кроків:

1. Перший крок Підготовка : визначення організаційного контексту, включаючи бізнес-цілі, нормативні вимоги. Визначення завдань, пріоритетів та ресурсів доступу для виконання завдань у сфері кібербезпеки.

Підкрок – Пріоритезація та межі: визначення критичних активів, систем та операцій. Визначення, які функції та послуги є найбільш важливими для бізнес-процесів та завдань організації.

Підкрок – Створення стратегії управління ризиками: розробка стратегії управління ризиками кібербезпеки. Встановлення цілей, визначення рівнів толерантності до ризику та визначення пріоритетів для зменшення ризику.

2. Другий крок Визначення: підкроки ідетифікації та оцінки.

Підкрок - Ідентифікація активів: ідентифікація та документація всіх інформаційних активів, включаючи обладнання, програмне забезпечення, дані та персонал.

Підкрок - Оцінка загроз: аналіз потенційних загроз кібербезпеці та вразливостей, які можуть вплинути на активи організації.

Підкрок - Оцінка вразливості: оцінка слабких і вразливих місць, які існують в інфраструктурі, системах і програмах організації.

Підкрок - Оцінка ризику: об’єднання інформації, зібраної в результаті оцінки загроз і вразливостей, щоб оцінити ймовірність і потенційний вплив різних кіберзагроз.

3. Третій крок - Захист:

Підкрок – Контроль доступу: Запровадження заходів для контролю того, хто має доступ до критично важливих систем, даних і ресурсів.

Підкрок – Безпека даних: застосування шифрування, маскування та інших безпекових заходів для захисту конфіденційної інформації від несанкціонованого доступу або розголошення.

Підкрок – Навчання та підвищення обізнаності: забезпечення навчання та програм підвищення обізнаності з кібербезпеки.

Підкрок – Планування реагування на інциденти: розробка та впровадження планів реагування на інциденти.

4. Четвертий крок Виявлення:

Підкрок – Безперервний моніторинг: запровадження інструментів і процесів для безперервного моніторингу IT-інфраструктури, мереж і додатків організації на наявність ознак ненормальної активності або інцидентів безпеки.

Підкрок – Виявлення аномалій: використовуйте технології та методи для виявлення незвичайної або підозрілої поведінки в системах і мережах організації.

5. П’ятий крок Реагування:

Підкрок – Реагування на інциденти та звітування: встановлення процедури для швидкого реагування на інциденти кібербезпеки, включаючи звітування, розслідування, стримування, викорінення та відновлення.

Підкрок – Координація та комунікація: визначення ролі та обов’язків членів групи реагування на інциденти та встановлення каналів для ефективного спілкування під час інциденту.

6. Шостий крок Відновлення:

Підкрок – Планування відновлення: розробка та реалізація планів відновлення критичних систем і операцій після інциденту кібербезпеки.

Підкрок – Отримання уроків: Проведення оглядів після інцидентів, щоб проаналізувати реагування на інциденти та зусилля з відновлення, визначити сфери, які потрібно покращити, і відповідно оновити плани реагування на інциденти.

7. Останній крок – Перегляд і оновлення: Періодично перегляд та оновлення програм кібербезпеки організації з урахуванням змін у технології, загроз і бізнес-пріоритетів.

До переваг NIST Cybersecurity Framework можна віднести:

1. Стандартизація та надійність: Фреймворк надає загальний набір стандартів та керівництв, які допомагають організаціям впоратися з кіберзагрозами.
2. Простота використання: Фреймворк надає простий та зрозумілий набір керівництв та інструментів для реалізації кібербезпекових заходів.
3. Адаптивність: Він дозволяє організаціям пристосовувати свій підхід до кібербезпеки відповідно до їхніх потреб та специфіки.
4. Фокус на бізнес-процесах: Фреймворк допомагає організаціям зорієнтувати свої безпекові заходи на захист бізнес-процесів та важливих активів.

Явними недоліками NIST Cybersecurity Framework є:

1. Не враховує специфічних потреб галузей: Фреймворк є загальним і може не враховувати специфічні потреби різних галузей.
2. Вимагає кваліфікації: Для успішного впровадження фреймворка може бути необхідними фахові знання та розуміння принципів кібербезпеки.
3. Може бути затратним по часу: Реалізація всіх складових фреймворка може вимагати значних зусиль та ресурсів.

Загалом NIST Cybersecurity Framework допомагає організаціям оцінити поточний стан кібербезпеки, визначити сфери, які потрібно покращити, і впровадити ефективні заходи безпеки для захисту своїх активів і операцій.

1.4.3 OWASP Risk Assessment Calculator

OWASP калькулятор оцінки ризиків є інструментом для оцінки та кількісного визначення ризиків безпеки, пов’язаних із програмним забезпеченням [30]. Цей калькулятор розроблено Open Web Application Security Project (OWASP) для забезпечення здійснення оцінки та визначення пріоритетів ризиків безпеки.

Головне призначення OWASP калькулятора оцінки ризиків — допомога у виявленні, аналізі та визначенні пріоритетів ризиків безпеки в програмних застосунках. Основна його функція підтримка прийняття рішень щодо розподілу ресурсів на заходи безпеки.

Компоненти: Фактори ризику; Оцінка ймовірності та впливу; Розрахунок оцінки ризику; Класифікація ризику.

Калькулятор враховує різні фактори ризику, які впливають на загальну оцінку ризику застосунків. Ці фактори включають рівень зловмисника, його навички та кількість потенційних зловмисників, а також наявність вразливостей, можлівість їх використання та втрати від них.

Оцінюється ймовірність реалізації загрози та потенційний вплив, який вона може мати на програму та її користувачів.

На основі попередньо визначеного підходу, калькулятор поєднує оцінки ймовірності та впливу, щоб створити числову оцінку ризику.

Потім оцінка ризику класифікується по рівнях ризику з метою побудові таксономії їх пріоритетів.

Виділяють переваги:

Об’єктивна оцінка - забезпечує стандартизований підхід до оцінки ризиків, зменшуючи суб’єктивні судження.

Розстановка пріоритетів – присвоєння числових значень ризикам допомагає визначити пріоритетність заходів безпеки на основі їх потенційного впливу.

Розподіл ресурсів – допомагає розподіляти ресурси, фокусуючись на аналізі найбільш критичних ризиків.

Комунікація – є основою для обговорення між сторонами процесу оцінки, надаючи загальну термінологію для обговорення ризиків безпеки.

Калькулятор використовує методологію, засновану на галузевому досвіду, практиках і стандартах. У ньому використовуються принципи систем управління ризиками, наприклад ISO 27005 і NIST SP 800-30.

Методологія складається з кількох основних етапів:

Етап 1. Ідентифікація активів і загроз:

Процес починається з визначення активів, пов’язаних із програмним забезпеченням. Це можуть бути конфіденційні дані, функції, апаратна архітектура, програмні компоненти тощо.

Наступний крок – визначення потенційних загроз притоманних цим активам. В першу чергу розглядаються різні векторі атак і сценаріїв, які потенційно можуть поставити під загрозу безпеку програми.

Етап 2. Оцінка ймовірності:

Цей крок – оцінка вірогідності або ймовірності реалізації кожної ідентифікованої загрози. Оцінка ґрунтується на властівостях аналізуємого застосунку, це уразливість застосунку потенційними зловмисниками, існуючі дані про ці вразливости та наявність заходів безпеки.

Етап 3. Оцінка впливу:

Оцінка впливу зосереджена на ідентифікації потенційних наслідків успішного порушення безпеки або використання вразливості. Це включає оцінку розміру збитків, а саме фінансових та репутаційних втрат, правових наслідків тощо.

Етап 4. Виявлення та аналіз вразливості:

Цей крок передбачає виявлення уразливих місць. Ці вразливості можуть бути пов’язані з кодом, конфігураціями, залежностями чи іншими аспектами життєвого циклу програмного забезпечення.

Етап 5. Фактори ризику та показники:

Калькулятор має враховувати різні фактори ризику, які впливають на загальний профіль ризику програми. Ці фактори можуть включати популярність застосунку, вимоги до конфіденційності даних, які вона обробляє, нормативно правові вимоги тощо.

Калькулятор використовує показники та формули для кількісної оцінки ризику, пов’язаного з кожною ідентифікованою загрозою.

6. Оцінка ризику:

На цьому кроку оцінки ймовірності та впливу поєднуються шляхом використання попередньо визначеного підходу для розрахунку числової оцінки

ризику для кожної ідентифікованої загрози. Ця оцінка являє собою кількісне відображення рівня ризику, пов’язаного з кожною загрозою.

7. Класифікація ризику:

Розраховані показники ризику класифікуються за рівнями ризику (наприклад, низький, середній, високий), щоб полегшити визначення пріоритетів. Це допомагає зацікавленим сторонам спершу зосередитися на усуненні найбільш критичних ризиків.

Рекомендації щодо використання:

Калькулятор оцінки ризиків OWASP призначено для фахівців із безпеки, розробників, керівників проектів та інших зацікавлених сторін, залучених до життєвого циклу розробки програмного забезпечення.

Калькулятор оцінки ризиків OWASP надає інструмент для організацій, який дозволяє оцінювати й усувати ризики безпеки у своїх програмних застосунках, узгоджуючи їх функціонування із актуальними практиками управління ризиками та забезпечення безпеки.

Попередньо були показані части функціональної методології. Але треба згадати, що після визначення об’єктів, що оцінюються відбувається оцінювання за допомогою Факторів ризику. Кожний з факторів має декілька визначених методологією опцій вибору зі свої коефіцієнтом, який використовується у формулі підрахунку рівня ризику. Фактори ризику розподіляються на 2 основні групи факторі вірогідності (Likelihood Factors) та факторі впливу (Impact Factors), наведено на рис. 1.2 [14].

Спочатку вираховуються фактори вірогідності, які в свою чергу розподіляються на дві підгрупи: фактори зловмисника (Threat Agent Factors) та фактори вразливості (Vulnerability Factors).

На останньому етапі вираховується загальна оцінка величини ризику, за допомогою множення Оцінки пливу на оцінку ймовірності, як показано в табл. 1.2, табл. 1.3.

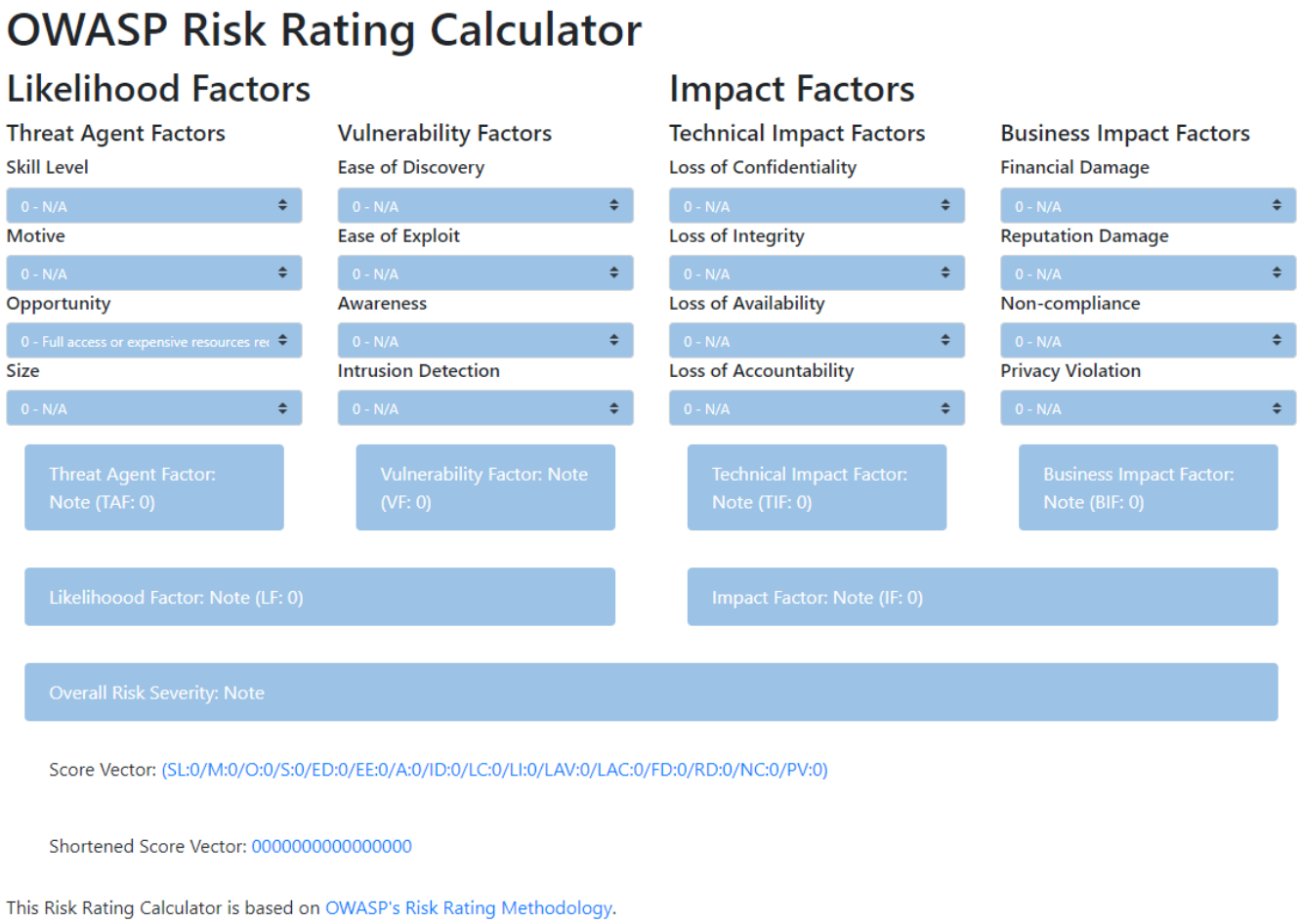


Рис. 1.2 Приклад функціонального додатку оцінки ризиків

Таблиця 1.2

Оцінка відносно коефіцієнтів

Коефіцієнт	Оцінка
<3	Низький
3<=, <6	Середній
6<=	Високий

Також для розуміння шкалі оцінки існує матриця ризиків

Таблиця 1.3

Матриця ризиків OWASP

Вірогідність	Вплив		
	Низький	Середній	Високий
Низька	Низький	Низький	Середній
Середня	Низький	Середній	Високий
Висока	Середній	Високий	Критичний

Переваги:

1. Безкоштовність та доступність: OWASP Risk Assessment Calculator є безкоштовним і відкритим інструментом, доступним для всіх зацікавлених сторін.

2. Використання універсальних метрик: Інструмент використовує загальновизнані метрики ризиків, такі як ймовірність виникнення загрози та потенційний вплив, що дозволяє проводити оцінку на базі об'єктивних критеріїв.

Недоліки:

1. Обмежена універсальність: OWASP Risk Assessment Calculator може бути обмеженим, оскільки він спеціалізується на веб-додатках.

2.Не завжди докладний: Калькулятор може не надавати докладного опису всіх можливих ризиків і вразливостей, що вимагають окремого аналізу.

3. Не підходить для всіх сценаріїв: Інструмент може не підходити для складних або незвичайних сценаріїв, які вимагають індивідуального аналізу.

4. Залежність від якості вхідних даних: Точність результатів визначення ризиків залежить від якості та точності вхідних даних, які вводяться в інструмент.

Загалом, OWASP Risk Assessment Calculator є корисним інструментом для оцінки ризиків веб-додатків, але важливо розуміти його обмеженості.

1.4.4 Factor Analysis of Information Risk (FAIR)

Факторний аналіз інформаційних ризиків (FAIR) — це кількісна фінансоорентована оцінка ризиків, яка дозволяє організаціям ідетифікувати, аналізувати та кількісно оцінювати ризики інформаційної безпеки у фінансовому плані. FAIR забезпечує структурований і систематичний підхід до оцінки та встановлення пріоритетів ризиків, пов'язаних з інформаційними активами [15].

Метою та завданнями є кількісна оцінка ризику, ідетифікація впливу на бізнес пріоритезація ризиків.

Кількісна оцінка ризику має на меті забезпечити кількісний підхід до оцінки ризику, який базується на даних. Це дозволяє організаціям виражати та повідомляти про ризики в грошовому еквіваленті, що полегшує прийняття кращих рішень.

Ідетифікація впливу на бізнес допомагає зрозуміти потенційний фінансовий вплив інцидентів безпеки, дозволяючи приймати обґрунтовані рішення щодо розподілу ресурсів і стратегій обробки ризиків.

Пріоритезація ризиків: шляхом кількісної оцінки ризиків організації можуть визначити їх пріоритетність на основі їх потенційного фінансового впливу. Це гарантує, що ресурси розподіляються насамперед для вирішення найбільш критичних ризиків.

Ключові компоненти розрахунків:

Частота випадків втрати (LEF); Величина втрати (LM); Ризик; Вартість активів (AV); Частота подій (TEF).

LEF це розрахункова частота, з якою очікується певний тип події збитків протягом певного періоду. Вона кількісно визначає, як часто певна загроза може відбуватися.

LM представляє оцінений фінансовий вплив конкретного типу збиткової події. Він кількісно визначає потенційні грошові втрати, пов'язані з конкретною загрозою.

Ризик = LEF x LM. Ризик, пов’язаний із певною загрозою, розраховується як добуток частоти подій втрати та величини втрати. Ця формула забезпечує кількісне представлення ризику.

AV представляє вартість інформаційного активу, який знаходиться під загрозою. Це стосується даних, систем, програм або будь-якого іншого активу, на який може вплинути інцидент безпеки.

TEF оцінює частоту, з якою певна загроза може статися.

Етапи виконання FAIR аналізу:

Етап 1 Визначення обсягу: визначення меж аналізу, включаючи конкретні активи, загрози та сценарії, які будуть оцінюватися.Збір і аналіз даних: збір даних та інформації, пов’язаної з цінністю активів, частотою загроз і величиною збитків. Аналіз цих даних, для оцінки значення для кожного компонента.

Етап 2 Розрахунок ризику: використання формули LEF x LM для розрахунку ризиків, пов’язаних із конкретними загрозами.

Етап 3 Аналіз чутливості: проведення аналізу чутливості, щоб зрозуміти, як зміни вхідних значень (наприклад, вартості активів, величини збитків) впливають на загальну оцінку ризику.

Етап 4 Звітування та комунікація: процес комунікації між зацікавленим сторонам про результати аналізу, включаючи оцінені ризики, потенційні фінансові наслідки та рекомендовані стратегії зменшення ризиків.

Переваги аналізу:

Об’єктивне прийняття рішень: FAIR забезпечує кількісну основу для прийняття рішень, зменшуючи залежність від суб’єктивних оцінок ризику.

Розподіл ресурсів: допомагає визначити пріоритети ресурсів для заходів із зменшення ризиків на основі потенційного фінансового впливу конкретних загроз.

Покращена комунікація: забезпечує чітке та стисле повідомлення про ризики зацікавленим сторонам, у тому числі керівникам, що дозволяє більш обґрунтовано обговорювати питання управління ризиками.

Переваги FAIR:

1. Кількісна оцінка ризиків: Однією з основних переваг FAIR є можливість кількісно оцінювати ризики інформаційної безпеки, що дозволяє більш точно розуміти потенційні загрози та їх вплив на організацію.
2. Урахування ймовірностей та важливості: FAIR дозволяє враховувати ймовірності виникнення ризиків та важливість їх наслідків, що дозволяє краще оцінити реальний вплив.
3. Гнучкість та адаптивність: Методологія FAIR дозволяє адаптуватися до різних типів організацій, галузей та різноманітних сценаріїв ризиків.
4. Можливість порівнювати ризики: Завдяки кількісній оцінці, FAIR дозволяє порівнювати ризики між собою та приймати більш обґрунтовані рішення щодо їх управління.

Таблиця 1.4

Порівняльна характеристика засобів оцінювання, що вже існують

Характеристика	OWASP Risk Assessment Calculator	NIST Cybersecurity Framework	FAIR	SIEM Tools
Доступність	Висока (відкрите джерело)	Висока (відкритий доступ)	Середня (вимагає підготовки)	Висока (широко розповсюджені)
Точність Оцінки	Висока для веб-застосунків	Загальна, адаптивна	Висока, кількісна	Залежить від інструменту
Масштабованість	Обмежена	Висока	Висока	Висока

Закінчення таблиці 1.4

Характеристика	OWASP Risk Assessment Calculator	NIST Cybersecurity Framework	FAIR	SIEM Tools
Автономність	Ручне використання	Частково автоматизована	Потребує експертного аналізу	Висока (автоматизація)
Гнучкість	Обмежена	Висока	Висока	Висока
Ефективність	Залежить від користувача	Залежить від реалізації	Висока в кількісному аналізі	Залежить від використання

До недоліків FAIR відносять:

1. Складність в використанні: FAIR може бути складним для впровадження та використання без належного навчання та експертизи.

2. Вимоги до даних: Для точної оцінки ризиків за допомогою FAIR, часто потрібна значна кількість даних.

3. Можливість суб'єктивних оцінок: Як і в будь-якій кількісній оцінці, FAIR може піддаватися впливу суб'єктивних оцінок, що можуть спотворити результати.

4. Часова витратність: Оцінка ризиків за допомогою FAIR може вимагати значної кількості часу, особливо при великих обсягах даних та складних сценаріях.

FAIR — це інструмент для покращення практики управління ризиками шляхом кількісного визначення та визначення пріоритетів ризиків інформаційної безпеки. Він забезпечує структурований підхід, який добре узгоджується з фінансовими та бізнес-орієнтованими поглядами на ризик.

Висновок до розділу 1

В результаті виконання розділу 1 отримані наступні результати:

В першому роздилі наведено оцінку рівня небезпек критичних даних корпоративних застосунків. А саме, наведено таксономію методів захисту критичних даних. Описано інформаційні ризики та можлива шкода від них. Проаналізовано метод експертної оцінки. Показано, що на практиці, ефективне використання експертної оцінки вимагає уважності до вибору експертів, чітко визначених завдань для оцінки та грамотного аналізу отриманих результатів. Крім того, може бути корисним використовувати експертну оцінку як один із елементів комплексної стратегії оцінки ризиків. А саме, якісний аналіз ризику має складатися минімум з 4х кроків:

Ідентифікація ризику - найважливіша частина якісного аналізу ризику. Якщо не вдасться визначити ризики завчасно, керувати ними стане надзвичайно складно. Спосіб для ідентифікації ризику полягає в тому, щоб він був простим а саме все, що може мати непевний вплив на підприємство. Виявлення очевидних ризиків допоможе глибше зануритися в більш хибні. Ідентифікація ризику пов’язана з кількістю, тому потрібно задіяти якомога більшу кількість людей, щоб отримати широкий діапазон поглядів;

Аналіз впливу - розділення ризиків на загрози та можливості. Використовуючи якісний аналіз ризику, оцінюють вплив кожного ризику за шкалою (1-5 або низький/середній/високий/екстремальний). Далі оцінюють ймовірність виникнення кожного ризику, використовуючи аналогічну шкалу. Отримані бали об’єднують, щоб створити загальний рейтинг ризику;

Усунення ризику - процес вчинення заходів щодо ризику;

Огляд і моніторинг — процес повторення останніх кроків для повторної оцінки ризиків та виявлення помилок минулого оцінювання.

При аналізі менеджменту ризиків розглянуто методи кількісної та якісної оцінки ризиків. А також підхід, який поєднує їх.

В ході аналізу засобів оцінки шкоди показана матриця ризиків, яка часто використовується під час оцінки ризику для вимірювання рівня ризику, враховуючи наслідки/важкість та ймовірність реалізації ризику.

На останнє в першому розділі розглянуто структуру, що забезпечує структурований підхід для організацій, щоб підвищити рівень кібербезпеки та підвищити стійкість до кіберзагроз. Це NIST Cybersecurity Framework — комплексний набір інструкцій і практик, розроблених, щоб допомогти організаціям керувати ризиками кібербезпеки та зменшувати їх. Він був розроблений Національним інститутом стандартів і технологій (NIST), федеральним агентством Міністерства торгівлі США.

Також в якості майбутньої базі для порівняння обрано калькулятор, розроблений Open Web Application Security Project (OWASP), який забезпечує структурований підхід до оцінки та визначення пріоритетів ризиків безпеки та застосунок для факторного аналізу інформаційних ризиків (FAIR) — який призначено для кількісної оцінки ризиків, яка в свою чергу дозволяє організаціям розуміти, аналізувати та кількісно оцінювати ризики інформаційної безпеки у фінансовому плані. Він забезпечує структурований і систематичний підхід до оцінки та встановлення пріоритетів ризиків, пов’язаних з інформаційними активами.

Проаналізовано існуючі методи оцінки рівня небезпек критичних даних корпоративних застосунків та на основі результату проведеного аналізу було обрано методи для вирішення задачі.

РОЗДІЛ 2
ВДОСКОНАЛЕННЯ МЕТОДІВ ОЦІНКИ РІВНЯ НЕБЕЗПЕК НА ОСНОВІ АНАЛІЗУ СПЕЦІАЛІЗОВАНИХ БАЗ ДАНИХ

2.1. Методи оцінки рівня небезпек на основі Common Weakness Enumeration (CWE)™

Common Weakness Enumeration (CWE)™ складна для розуміння. Але вона може запропонувати досить багато, тому розглянемо CWE більш детально [30].

Для початку слід описати, що таке CWE. CWE – це розроблений спільнотою список поширених типів слабких місць програмного та апаратного забезпечення, які можуть мати наслідки для безпеки.

«Слабкість» — це стан у програмному, мікро програмному, апаратному або сервісному компоненті, який за певних обставин може сприяти впровадженню вразливостей. Слабкі умови в багатьох випадках вводяться розробником під час розробки продукту. Незважаючи на те, що розробники можуть мати абсолютно різні методи кодування, всі вони здатні впроваджувати один і той же загальний тип слабких місць, що призводить до вразливостей у їхніх власних продуктах. Список CWE та пов'язані з ним таксономії та схеми класифікації слугують мовою, яка може бути використана для виявлення та опису цих слабких сторін у термінах «CWE». Найкраща новина полягає в тому, що CWE може вільно використовуватися будь-якою організацією або особою для будь-яких досліджень, розробок та/або комерційних цілей відповідно до Умов використання CWE.

Для будь якої вразливості CWE присвоюється ідентифікатор у формі CWE-<ID>, де це <ID> просто унікальний номер, вибраний під час присвоєння (наприклад, «CWE-888»). Після ідентифікатора CWE-ID йде описова назва слабкого місця (наприклад, "CWE-798: Використання жорстко закодованих облікових даних").

Щоб вразливості було присвоєно CWE-ID для опублікування на веб-сайті CWE, її опис повинен містити набір необхідної інформації, включаючи:

Назву – Назва включає передбачувану поведінку, помилку (тобто слабкість), проблемний ресурс (якщо це доречно) і порушену технологію (якщо це доречно);

Резюме – Резюме являє собою одне або два речення, які описують слабкість, акцентуючи увагу на допущеній помилці;

Розширений опис – Розширений опис складається з одного або двох абзаців, які додатково описують, як слабкість може бути проблемою. Він розрахований на аудиторію, яка може не розуміти, як слабкість може бути проблемою.

Способи впровадження - Режим впровадження визначає, як і коли може бути виявлено слабкі місця (наприклад, за етапами життєвого циклу продукту).

Потенційні пом'якшення – Потенційні пом'якшення наслідків – це один або декілька методів, які усунуть та/або зменшать частоту або вплив слабкості.

Загальні наслідки – Загальні наслідки – це типовий негативний вплив на безпеку (або вплив), який виникає, якщо зловмисник може використати цю слабкість.

Застосовані платформи – Застосовані платформи це мови програмування, операційні системи, архітектури та технології, в яких зазвичай виявляється ця слабкість.

Демонстраційні приклади – Демонстраційні приклади ілюструють слабкість за допомогою коду, пояснювального тексту та/або діаграм.

Спостережувані приклади – Спостережувані приклади – це публічно повідомлені вразливості (наприклад, записи CVE) у реальних продуктах, які демонструють слабкі сторони.

Відносини – Відносини – це інші CWE, пов'язані зі слабкістю.

Посилання – Посилання включають одну або кілька цитат з URL–адресами для наукових робіт, білих книг, дописів у блогах, презентацій слайдів або відео, які описують слабкі сторони.

Таким чином вхідну інформацію в базу CWE можна описати наступним вектором:

$$CWE = \begin{pmatrix} Nam \\ Rez \\ Op \\ Vn \\ s \\ Pos \\ Wer \\ Prd \\ Prs \\ Rel \\ Ref \end{pmatrix} \quad (2.1),$$

де Nam - назва; Rez - Резюме; Op - Розширений опис; Vn - Способи впровадження; S - Потенційні пом'якшення; Pos - Загальні наслідки; Wer - Застосовані платформи; Prd - Демонстраційні приклади; Prs - Спостережувані приклади; Rel – Відносини; Ref - Посилання.

Розглянемо приклад. Наведені нижче скріншоти дають уявлення про приклад, представлений у посібнику [https://cwe.mitre.org/data/definitions/1000.html], "CWE-798: Використання жорстко закодованих облікових даних". Цей CWE описує ситуацію, коли облікові дані, такі як паролі або криптографічні ключі, були жорстко закодовані в апаратному або програмному продукті.

Для генерації наступних форм було переглянуто CWE, шляхом відвідання https://cwe.mitre.org/, ввівши «CWE-798» (без лапок) у поле пошуку ідентифікатора у верхньому правому куті сторінки та натиснувши кнопку «Перейти».

На рис. 2.1 показані деякі з описових текстів CWE-798, в той час як на малюнку нижче показані розділи «Розширеного опису» та «Поширені наслідки». Останній, наприклад, включає очікувані наслідки відповідно до тріади безпеки: Цілісності, Конфіденційності, Доступності, а також специфічні в сфери контролю доступу та інсталяції.

Common Weakness Enumeration
A community-developed list of SW & HW weaknesses that can become vulnerabilities

Home > CWE List > CWE- Individual Dictionary Definition (4.15) ID Lookup:

Home | About | CWE List | Mapping | Top-N Lists | Community | News | Search

CWE-798: Use of Hard-coded Credentials

Weakness ID: 798
Vulnerability Mapping: ALLOWED
Abstraction: Base

View customized information: Conceptual | Operational | Mapping Friendly | Complete | Custom

Description

The product contains hard-coded credentials, such as a password or cryptographic key.

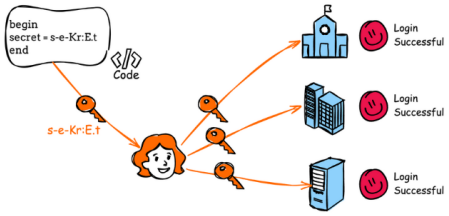


Рис. 2.1. Приклад опису вразливості № 798 Використання жорстко закодованих облікових даних

Extended Description

There are two main variations:

- Inbound: the product contains an authentication mechanism that checks the input credentials against a hard-coded set of credentials. In this variant, a default administration account is created, and a simple password is hard-coded into the product and associated with that account. This hard-coded password is the same for each installation of the product, and it usually cannot be changed or disabled by system administrators without manually modifying the program, or otherwise patching the product. It can also be difficult for the administrator to detect.
- Outbound: the product connects to another system or component, and it contains hard-coded credentials for connecting to that component. This variant applies to front-end systems that authenticate with a back-end service. The back-end service may require a fixed password that can be easily discovered. The programmer may simply hard-code those back-end credentials into the front-end product.

Common Consequences

Scope	Impact	Likelihood
Access Control	**Technical Impact:** *Bypass Protection Mechanism* If hard-coded passwords are used, it is almost certain that malicious users will gain access to the account in question. Any user of the product that hard-codes passwords may be able to extract the password. Client-side systems with hard-coded passwords pose even more of a threat, since the extraction of a password from a binary is usually very simple.	
Integrity Confidentiality Availability Access Control Other	**Technical Impact:** *Read Application Data; Gain Privileges or Assume Identity; Execute Unauthorized Code or Commands; Other* This weakness can lead to the exposure of resources or functionality to unintended actors, possibly providing attackers with sensitive information or even execute arbitrary code. If the password is ever discovered or published (a common occurrence on the Internet), then anybody with knowledge of this password can access the product. Finally, since all installations of the product will have the same password, even across different organizations, this enables massive attacks such as worms to take place.	

Рис. 2.2. Розділи «Розширеного опису» та «Поширені наслідки».

Potential Mitigations

Phase: Architecture and Design
For outbound authentication: store passwords, keys, and other credentials outside of the code in a strongly-protected, encrypted configuration file or database that is protected from access by all outsiders, including other local users on the same system. Properly protect the key (CWE-320). If you cannot use encryption to protect the file, then make sure that the permissions are as restrictive as possible [REF-7].
In Windows environments, the Encrypted File System (EFS) may provide some protection.

Phase: Architecture and Design
For inbound authentication: Rather than hard-code a default username and password, key, or other authentication credentials for first time logins, utilize a "first login" mode that requires the user to enter a unique strong password or key.

Phase: Architecture and Design
If the product must contain hard-coded credentials or they cannot be removed, perform access control checks and limit which entities can access the feature that requires the hard-coded credentials. For example, a feature might only be enabled through the system console instead of through a network connection.

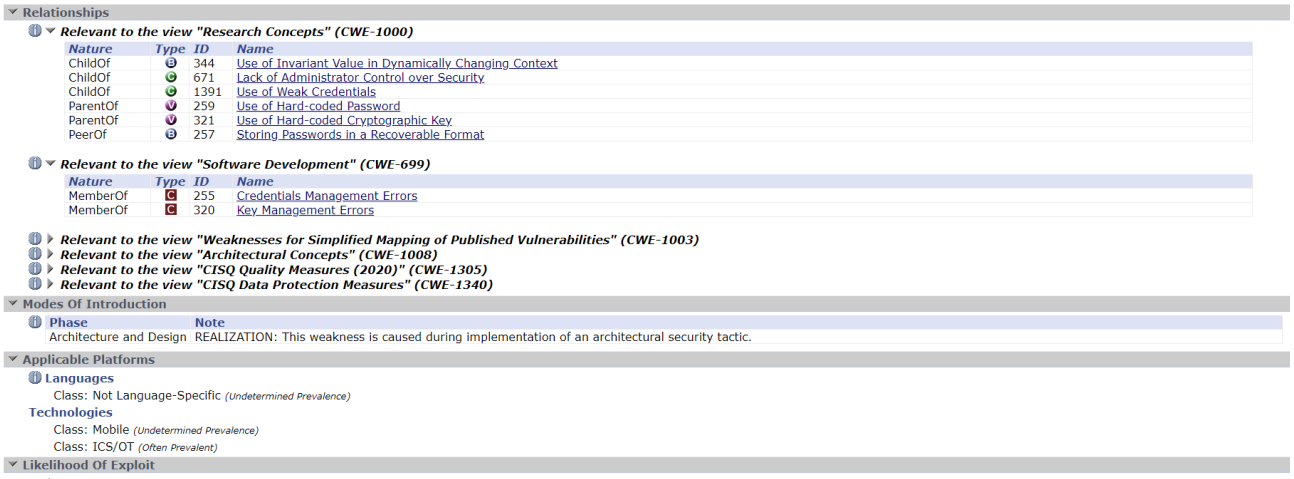
Phase: Architecture and Design
For inbound authentication using passwords: apply strong one-way hashes to passwords and store those hashes in a configuration file or database with appropriate access control. That way, theft of the file/database still requires the attacker to try to crack the password. When handling an incoming password during authentication, take the hash of the password and compare it to the saved hash.
Use randomly assigned salts for each separate hash that is generated. This increases the amount of computation that an attacker needs to conduct a brute-force attack, possibly limiting the effectiveness of the rainbow table method.

Phase: Architecture and Design
For front-end to back-end connections: Three solutions are possible, although none are complete.

- The first suggestion involves the use of generated passwords or keys that are changed automatically and must be entered at given time intervals by a system administrator. These passwords will be held in memory and only be valid for the time intervals.
- Next, the passwords or keys should be limited at the back end to only performing actions valid for the front end, as opposed to having full access.
- Finally, the messages sent should be tagged and checksummed with time sensitive values so as to prevent replay-style attacks.

Рис. 2.3. Розділ «Потенційні пом'якшення наслідків».

На наведеному вище малюнку показані можливі шляхи зменшення наслідків від використання вразливості. Особливістю є розподіл їх по фазі застосування.



Relationships

Relevant to the view "Research Concepts" (CWE-1000)

Nature	Type	ID	Name
ChildOf	B	344	Use of Invariant Value in Dynamically Changing Context
ChildOf	C	671	Lack of Administrator Control over Security
ChildOf	C	1391	Use of Weak Credentials
ParentOf	V	259	Use of Hard-coded Password
ParentOf	V	321	Use of Hard-coded Cryptographic Key
PeerOf	B	257	Storing Passwords in a Recoverable Format

Relevant to the view "Software Development" (CWE-699)

Nature	Type	ID	Name
MemberOf	C	255	Credentials Management Errors
MemberOf	C	320	Key Management Errors

Relevant to the view "Weaknesses for Simplified Mapping of Published Vulnerabilities" (CWE-1003)

Relevant to the view "Architectural Concepts" (CWE-1008)

Relevant to the view "CISQ Quality Measures (2020)" (CWE-1305)

Relevant to the view "CISQ Data Protection Measures" (CWE-1340)

Modes Of Introduction

Phase	Note
Architecture and Design	REALIZATION: This weakness is caused during implementation of an architectural security tactic.

Applicable Platforms

Languages

Class: Not Language-Specific *(Undetermined Prevalence)*

Technologies

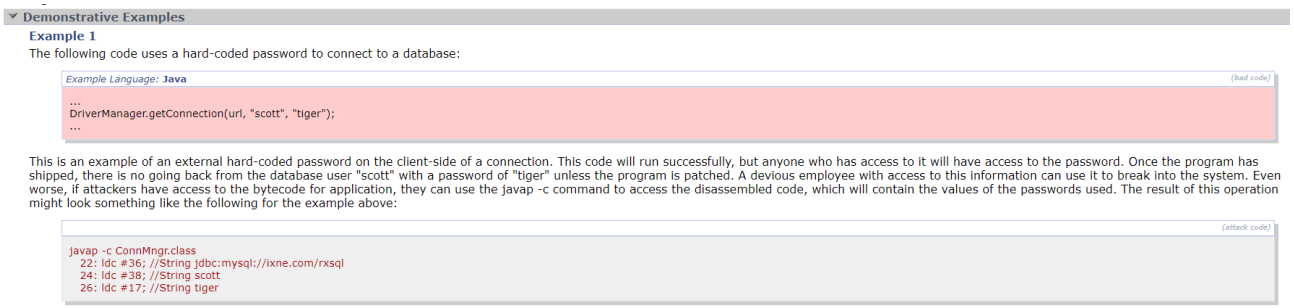
Class: Mobile *(Undetermined Prevalence)*

Class: ICS/OT *(Often Prevalent)*

Likelihood Of Exploit

High

Рис. 2.4. Розділи «Взаємодія», «Способи впровадження», «Застосовані платформи».



Demonstrative Examples

Example 1

The following code uses a hard-coded password to connect to a database:

Example Language: **Java** *(bad code)*

```
...
DriverManager.getConnection(url, "scott", "tiger");
...
```

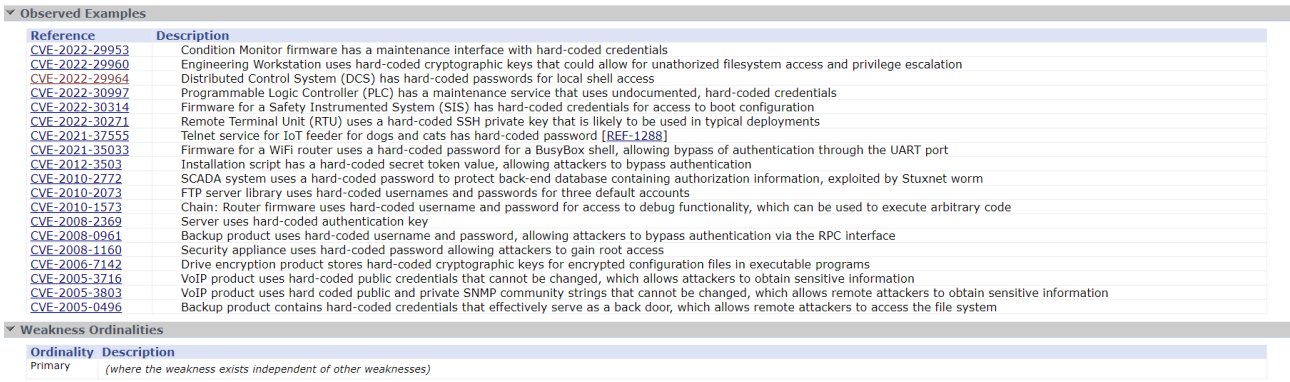
This is an example of an external hard-coded password on the client-side of a connection. This code will run successfully, but anyone who has access to it will have access to the password. Once the program has shipped, there is no going back from the database user "scott" with a password of "tiger" unless the program is patched. A devious employee with access to this information can use it to break into the system. Even worse, if attackers have access to the bytecode for application, they can use the javap -c command to access the disassembled code, which will contain the values of the passwords used. The result of this operation might look something like the following for the example above:

(attack code)

```
javap -c ConnMngr.class
  22: ldc #36; //String jdbc:mysql://ixne.com/rxsql
  24: ldc #38; //String scott
  26: ldc #17; //String tiger
```

Рис. 2.5. Розділ «Демонстраційні приклади».

Наведеній вище розділ демонструє приклади застосування вразливостей.



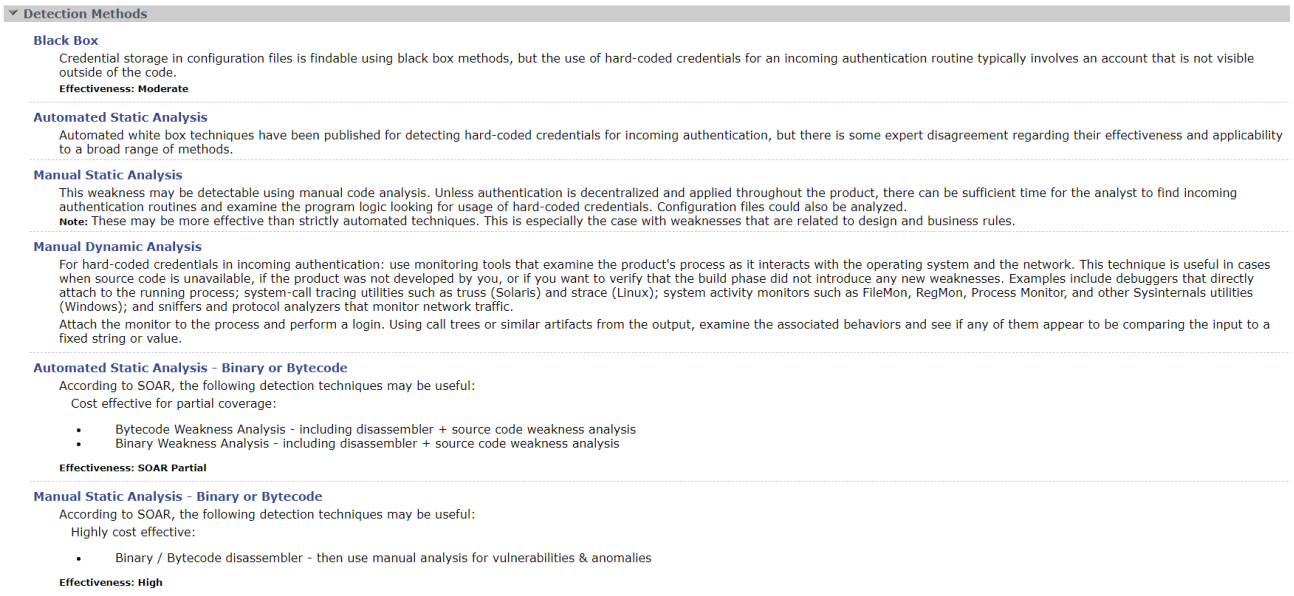
Observed Examples

Reference	Description
CVE-2022-29953	Condition Monitor firmware has a maintenance interface with hard-coded credentials
CVE-2022-29960	Engineering Workstation uses hard-coded cryptographic keys that could allow for unathorized filesystem access and privilege escalation
CVE-2022-29964	Distributed Control System (DCS) has hard-coded passwords for local shell access
CVE-2022-30997	Programmable Logic Controller (PLC) has a maintenance service that uses undocumented, hard-coded credentials
CVE-2022-30314	Firmware for a Safety Instrumented System (SIS) has hard-coded credentials for access to boot configuration
CVE-2022-30271	Remote Terminal Unit (RTU) uses a hard-coded SSH private key that is likely to be used in typical deployments
CVE-2021-37555	Telnet service for IoT feeder for dogs and cats has hard-coded password [REF-1288]
CVE-2021-35033	Firmware for a WiFi router uses a hard-coded password for a BusyBox shell, allowing bypass of authentication through the UART port
CVE-2012-3503	Installation script has a hard-coded secret token value, allowing attackers to bypass authentication
CVE-2010-2772	SCADA system uses a hard-coded password to protect back-end database containing authorization information, exploited by Stuxnet worm
CVE-2010-2073	FTP server library uses hard-coded usernames and passwords for three default accounts
CVE-2010-1573	Chain: Router firmware uses hard-coded username and password for access to debug functionality, which can be used to execute arbitrary code
CVE-2008-2369	Server uses hard-coded authentication key
CVE-2008-0961	Backup product uses hard-coded username and password, allowing attackers to bypass authentication via the RPC interface
CVE-2008-1160	Security appliance uses hard-coded password allowing attackers to gain root access
CVE-2006-7142	Drive encryption product stores hard-coded cryptographic keys for encrypted configuration files in executable programs
CVE-2005-3716	VoIP product uses hard-coded public credentials that cannot be changed, which allows attackers to obtain sensitive information
CVE-2005-3803	VoIP product uses hard coded public and private SNMP community strings that cannot be changed, which allows remote attackers to obtain sensitive information
CVE-2005-0496	Backup product contains hard-coded credentials that effectively serve as a back door, which allows remote attackers to access the file system

Weakness Ordinalities

Ordinality	Description
Primary	*(where the weakness exists independent of other weaknesses)*

Рис. 2.6. Розділ «Спостережувані приклади».



Detection Methods

Black Box

Credential storage in configuration files is findable using black box methods, but the use of hard-coded credentials for an incoming authentication routine typically involves an account that is not visible outside of the code.

Effectiveness: Moderate

Automated Static Analysis

Automated white box techniques have been published for detecting hard-coded credentials for incoming authentication, but there is some expert disagreement regarding their effectiveness and applicability to a broad range of methods.

Manual Static Analysis

This weakness may be detectable using manual code analysis. Unless authentication is decentralized and applied throughout the product, there can be sufficient time for the analyst to find incoming authentication routines and examine the program logic looking for usage of hard-coded credentials. Configuration files could also be analyzed.

Note: These may be more effective than strictly automated techniques. This is especially the case with weaknesses that are related to design and business rules.

Manual Dynamic Analysis

For hard-coded credentials in incoming authentication: use monitoring tools that examine the product's process as it interacts with the operating system and the network. This technique is useful in cases when source code is unavailable, if the product was not developed by you, or if you want to verify that the build phase did not introduce any new weaknesses. Examples include debuggers that directly attach to the running process; system-call tracing utilities such as truss (Solaris) and strace (Linux); system activity monitors such as FileMon, RegMon, Process Monitor, and other Sysinternals utilities (Windows); and sniffers and protocol analyzers that monitor network traffic.

Attach the monitor to the process and perform a login. Using call trees or similar artifacts from the output, examine the associated behaviors and see if any of them appear to be comparing the input to a fixed string or value.

Automated Static Analysis - Binary or Bytecode

According to SOAR, the following detection techniques may be useful:

Cost effective for partial coverage:

- Bytecode Weakness Analysis - including disassembler + source code weakness analysis
- Binary Weakness Analysis - including disassembler + source code weakness analysis

Effectiveness: SOAR Partial

Manual Static Analysis - Binary or Bytecode

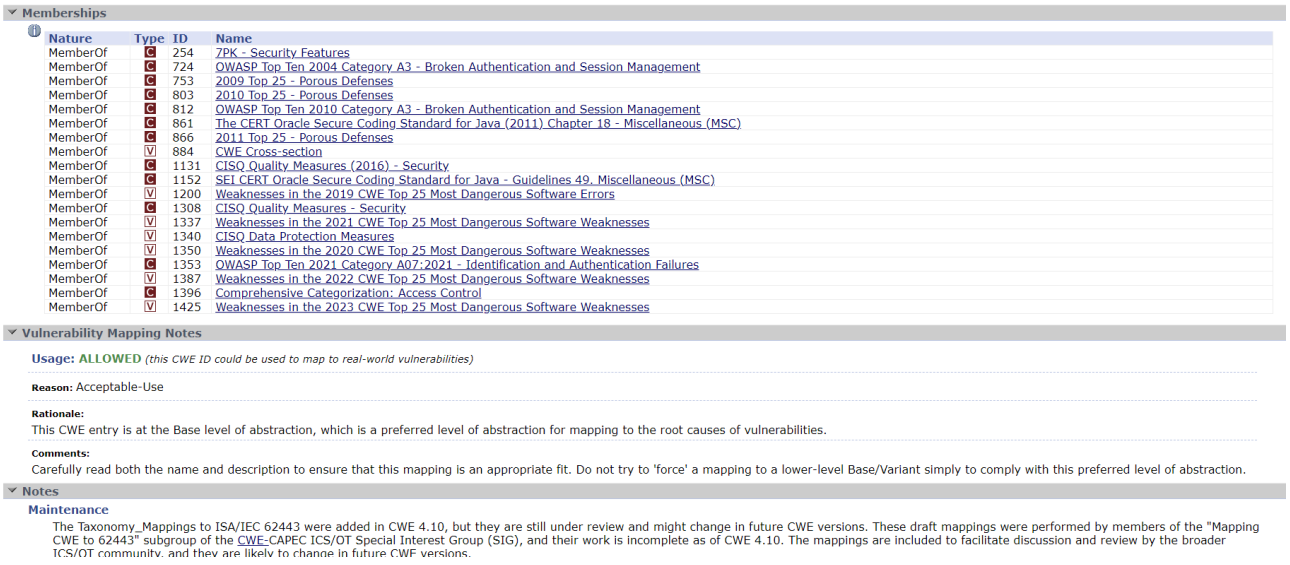
According to SOAR, the following detection techniques may be useful:

Highly cost effective:

- Binary / Bytecode disassembler - then use manual analysis for vulnerabilities & anomalies

Effectiveness: High

Рис. 2.7. Розділ «Методи виявлення».



Memberships

Nature	Type	ID	Name
MemberOf	C	254	7PK - Security Features
MemberOf	C	724	OWASP Top Ten 2004 Category A3 - Broken Authentication and Session Management
MemberOf	C	753	2009 Top 25 - Porous Defenses
MemberOf	C	803	2010 Top 25 - Porous Defenses
MemberOf	C	812	OWASP Top Ten 2010 Category A3 - Broken Authentication and Session Management
MemberOf	C	861	The CERT Oracle Secure Coding Standard for Java (2011) Chapter 18 - Miscellaneous (MSC)
MemberOf	C	866	2011 Top 25 - Porous Defenses
MemberOf	V	884	CWE Cross-section
MemberOf	C	1131	CISQ Quality Measures (2016) - Security
MemberOf	C	1152	SEI CERT Oracle Secure Coding Standard for Java - Guidelines 49. Miscellaneous (MSC)
MemberOf	V	1200	Weaknesses in the 2019 CWE Top 25 Most Dangerous Software Errors
MemberOf	C	1308	CISQ Quality Measures - Security
MemberOf	V	1337	Weaknesses in the 2021 CWE Top 25 Most Dangerous Software Weaknesses
MemberOf	V	1340	CISQ Data Protection Measures
MemberOf	V	1350	Weaknesses in the 2020 CWE Top 25 Most Dangerous Software Weaknesses
MemberOf	C	1353	OWASP Top Ten 2021 Category A07:2021 - Identification and Authentication Failures
MemberOf	V	1387	Weaknesses in the 2022 CWE Top 25 Most Dangerous Software Weaknesses
MemberOf	C	1396	Comprehensive Categorization: Access Control
MemberOf	V	1425	Weaknesses in the 2023 CWE Top 25 Most Dangerous Software Weaknesses

Vulnerability Mapping Notes

Usage: ALLOWED *(this CWE ID could be used to map to real-world vulnerabilities)*

Reason: Acceptable-Use

Rationale:

This CWE entry is at the Base level of abstraction, which is a preferred level of abstraction for mapping to the root causes of vulnerabilities.

Comments:

Carefully read both the name and description to ensure that this mapping is an appropriate fit. Do not try to 'force' a mapping to a lower-level Base/Variant simply to comply with this preferred level of abstraction.

Notes

Maintenance

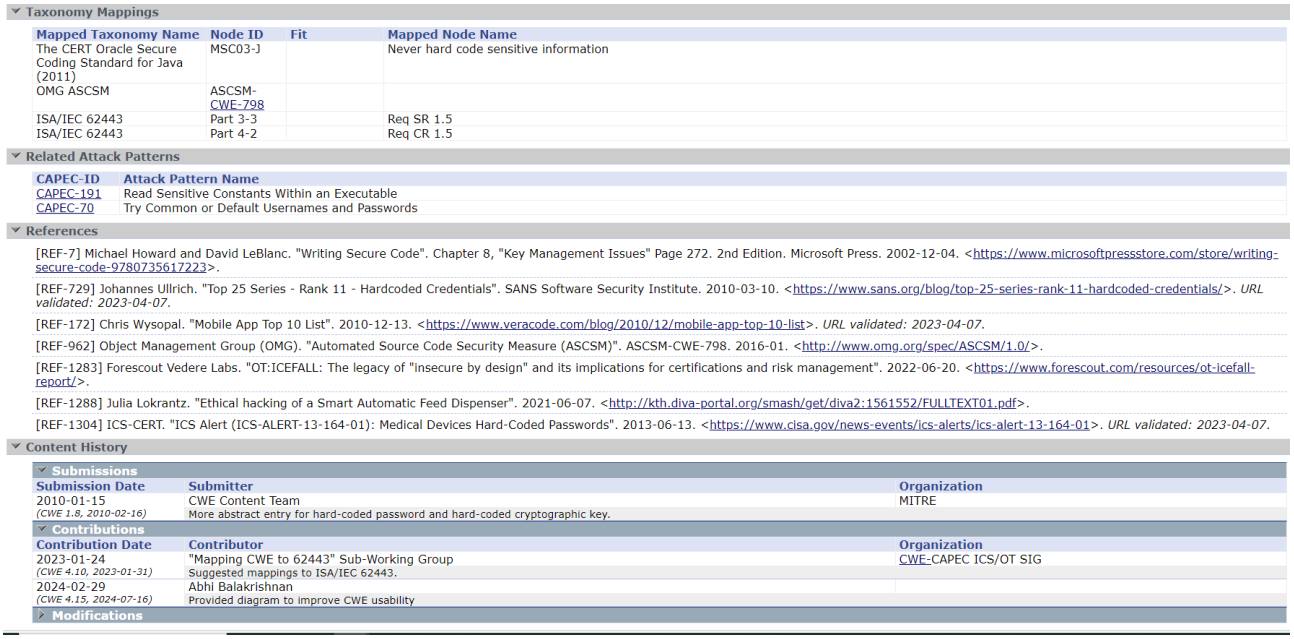
The Taxonomy_Mappings to ISA/IEC 62443 were added in CWE 4.10, but they are still under review and might change in future CWE versions. These draft mappings were performed by members of the "Mapping CWE to 62443" subgroup of the CWE-CAPEC ICS/OT Special Interest Group (SIG), and their work is incomplete as of CWE 4.10. The mappings are included to facilitate discussion and review by the broader ICS/OT community, and they are likely to change in future CWE versions.

Рис. 2.8. Розділі «Членство», «Мапа вразливостей», «Примітки».

На основі вище згаданого можна зробити висновок CWE – зручна для вчасного інформування власників автоматизованих систем, але вона має нерегулярну структуру і потребує побудову інформаційних систем для її швидкої інтерпретації та обробки. Тому для її вдосконалення пропонується двоетапна схема аналізу:

На першому кроці визначається фокус уваги аналізу, з бази вразливостей відокремлюється підмножина загроз, які є фокусними для даного аналізу далі будемо називати цю підмножину класом вразливостей.

На другому кроці проводиться більш детальний аналіз класу загроз. За рахунок однорідності підмножини вразливостей підвішується ефективність аналізу.



Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
The CERT Oracle Secure Coding Standard for Java (2011)	MSC03-J		Never hard code sensitive information
OMG ASCSM	ASCSM-CWE-798		
ISA/IEC 62443	Part 3-3		Req SR 1.5
ISA/IEC 62443	Part 4-2		Req CR 1.5

Related Attack Patterns

CAPEC-ID	Attack Pattern Name
CAPEC-191	Read Sensitive Constants Within an Executable
CAPEC-70	Try Common or Default Usernames and Passwords

References

[REF-7] Michael Howard and David LeBlanc. "Writing Secure Code". Chapter 8, "Key Management Issues" Page 272. 2nd Edition. Microsoft Press. 2002-12-04. <https://www.microsoftpressstore.com/store/writing-secure-code-9780735617223>.

[REF-729] Johannes Ullrich. "Top 25 Series - Rank 11 - Hardcoded Credentials". SANS Software Security Institute. 2010-03-10. <https://www.sans.org/blog/top-25-series-rank-11-hardcoded-credentials/>. *URL validated: 2023-04-07.*

[REF-172] Chris Wysopal. "Mobile App Top 10 List". 2010-12-13. <https://www.veracode.com/blog/2010/12/mobile-app-top-10-list>. *URL validated: 2023-04-07.*

[REF-962] Object Management Group (OMG). "Automated Source Code Security Measure (ASCSM)". ASCSM-CWE-798. 2016-01. <http://www.omg.org/spec/ASCSM/1.0/>.

[REF-1283] Forescout Vedere Labs. "OT:ICEFALL: The legacy of "insecure by design" and its implications for certifications and risk management". 2022-06-20. <https://www.forescout.com/resources/ot-icefall-report/>.

[REF-1288] Julia Lokrantz. "Ethical hacking of a Smart Automatic Feed Dispenser". 2021-06-07. <http://kth.diva-portal.org/smash/get/diva2:1561552/FULLTEXT01.pdf>.

[REF-1304] ICS-CERT. "ICS Alert (ICS-ALERT-13-164-01): Medical Devices Hard-Coded Passwords". 2013-06-13. <https://www.cisa.gov/news-events/ics-alerts/ics-alert-13-164-01>. *URL validated: 2023-04-07.*

Content History

Submissions

Submission Date	Submitter	Organization
2010-01-15 *(CWE 1.8, 2010-02-16)*	CWE Content Team More abstract entry for hard-coded password and hard-coded cryptographic key.	MITRE

Contributions

Contribution Date	Contributor	Organization
2023-01-24 *(CWE 4.10, 2023-01-31)*	"Mapping CWE to 62443" Sub-Working Group Suggested mappings to ISA/IEC 62443.	CWE-CAPEC ICS/OT SIG
2024-02-29 *(CWE 4.15, 2024-07-16)*	Abhi Balakrishnan Provided diagram to improve CWE usability	

Modifications

Рис. 2.9. Розділи «Таксономія», «Схожі шаблоні», «Посилання», «Журнал».

2.2. Методи оцінки рівня небезпек на основі National Vulnerability Database (NVD)

National Vulnerability Database [31] Національна база даних вразливостей. NVD – це репозиторій уряду США даних управління вразливостями на основі стандартів, представлених за допомогою протоколу автоматизації контенту безпеки (SCAP). Ці дані дозволяють автоматизувати управління вразливостями, вимірювання безпеки та відповідність. NVD включає бази даних із посиланнями на контрольні списки безпеки, недоліки програмного забезпечення, пов'язані з безпекою, назви продуктів і показники впливу. Всі вразливості в NVD мають ідентифікатор CVE.

2.2.1. Метрики вразливостей по NVD.

Загальна система оцінки вразливостей (CVSS) – це метод, який використовується для забезпечення якісного вимірювання серйозності. CVSS не є мірилом ризику. CVSS v2.0 і CVSS v3.x складаються з трьох метричних груп: базових, тимчасових і екологічних. CVSS v4.0 дещо відрізняється і складається з груп Базова, Загроза, Навколишнє середовище та Додаткова. За допомогою метрик отримується числова оцінка в діапазоні від 0 до 10. Оцінка CVSS також представлена у вигляді векторного рядка, стисненого текстового представлення значень, які використовуються для отримання оцінки. Таким чином, CVSS добре підходить як стандартна система вимірювання для галузей, організацій та урядів, яким потрібні точні та стабільні оцінки серйозності вразливостей. Два поширених використання CVSS – це розрахунок серйозності вразливостей, виявлених у системах, і як фактор визначення пріоритетності дій з усунення вразливостей. Національна база даних вразливостей (NVD) забезпечує збагачення CVSS для всіх опублікованих записів CVE.

NVD підтримує стандарти Common Vulnerability Scoring System (CVSS) v2.0, v3.x і v4.0. NVD надає CVSS-оцінку базових метрик, вроджених характеристик кожної вразливості. NVD надає калькулятор CVSS для кожної версії CVSS, щоб дозволити користувачам оцінювати показники, не пов'язані з базою.

Специфікації CVSS належать і управляються FIRST.Org, Inc. (FIRST), американською некомерційною організацією, місія якої полягає в допомозі командам реагування на інциденти комп'ютерної безпеки в усьому світі. Офіційну документацію CVSS можна знайти за посиланням [https://www.first.org/cvss/].

CVSS складається з чотирьох метричних груп: Базова, Загроза, Навколишнє середовище та Додаткова, кожна з яких складається з набору показників, як показано на рисунку 2.10

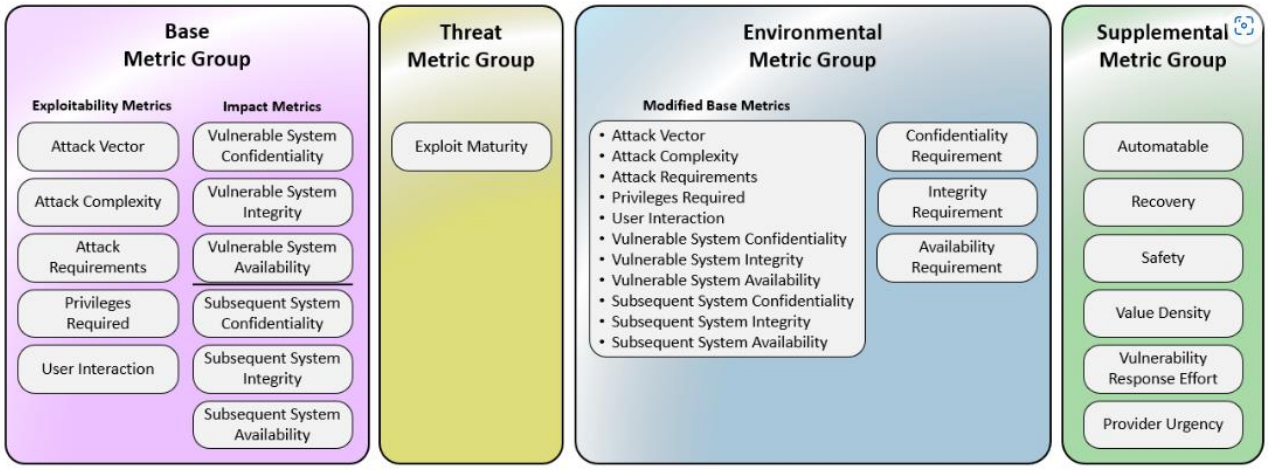


Рис. 2.10. Загальна структура метрик CVSS

Група «базових» показників представляє внутрішні характеристики вразливості, які є постійними в часі та в різних середовищах користувача. Вона складається з двох наборів показників: метрики експлуатаційної придатності та метрики впливу.

Група показників «загроз» відображає характеристики вразливості, пов'язаної із загрозою, яка може змінюватися з часом, але не обов'язково в різних середовищах користувача. Наприклад, підтвердження того, що вразливість не була використана, а також не було відкрито доступного коду або інструкцій експлойта, що підтверджує концепцію, знизить результуючий бал CVSS. Значення в цій групі показників можуть змінюватися з часом.

Група показників «Навколишнє середовище» відображає характеристики вразливості, які є актуальними та унікальними для конкретного середовища споживачів. Міркування включають наявність засобів контролю безпеки, які можуть пом'якшити деякі або всі наслідки успішної атаки, а також відносну важливість вразливої системи в технологічній інфраструктурі.

Група «додаткових» показників включає показники, які надають контекст, а також описують і вимірюють додаткові зовнішні атрибути вразливості. Відповідь на кожен показник у групі додаткових показників має визначати споживач CVSS, що дозволяє використовувати систему аналізу ризиків кінцевого користувача для застосування локально значущої серйозності до показників і значень. Жодна

метрика, в межах своєї специфікації, не матиме жодного впливу на остаточний бал CVSS (наприклад, CVSS-BTE). Потім організації-споживачі можуть призначати важливість та/або ефективний вплив кожного показника або набору/комбінації показників, надаючи їм більший, менший або зовсім відсутній вплив на категоризацію, пріоритезацію та оцінку вразливості. Метрики та значення просто передадуть додаткові зовнішні характеристики самої вразливості.

Розглянемо приклад опису конкретних загроз.

CVE-2020-3549

Уразливість у функціональності програмного забезпечення Cisco Firepower Management Center (FMC) і програмного забезпечення Cisco Firepower Threat Defense (FTD) може дозволити неавтентифікованому віддаленому зловмиснику отримати хеш реєстрації пристрою.

Вразливість пов'язана з недостатнім захистом від переговорів під час первинної реєстрації пристрою. Зловмисник, який займає позицію "людина посередині", може скористатися цією вразливістю, перехопивши певний потік зв'язку sftunnel між пристроєм FMC і пристроєм FTD. Успішний експлойт може дозволити зловмиснику розшифрувати та змінити зв'язок sftunnel між пристроями FMC та FTD, дозволяючи зловмиснику змінювати дані конфігурації, надіслані з пристрою FMC на пристрій FTD, або дані сповіщень, надіслані з пристрою FTD на пристрій FMC.

Для CVSS v4 Score: Base + Threat 5.2 маємо опис в вигляди тріад.

{показника, значення, коментар}

Attack Vector	Network	The vulnerable system is accessible from remote networks.
Attack Complexity	Low	No specialized conditions or advanced knowledge are required.
Attack Requirements	Present	An attacker must be on-path to be able to intercept communications between affected systems.
Privileges Required	None	No privileges are required for an attacker to successfully exploit the vulnerability.
User Interaction	Passive	A user must be logged in and using the application for traffic to be generated that an attacker could capture.
Vulnerable System Confidentiality	High	An attacker could gain access to the system with a highly privileged user account.
Vulnerable System Integrity	High	An attacker could gain access to the system with a highly privileged user account.
Vulnerable System Availability	High	An attacker could gain access to the system with a highly privileged user account.
Subsequent System Confidentiality	None	There is no impact to the vulnerable system confidentiality.
Subsequent System Integrity	None	There is no impact to the vulnerable system integrity.
Subsequent System Availability	None	There is no impact to the vulnerable system availability.
Exploit Maturity	Unreported	There is no known proof-of-concept code or malicious exploitation of this vulnerability.

Рис. 2.11. Опис показників CVE-2020-3549

Для збереження даних використовуються формати JSON та XML завдяки чому спрощується експорт та імпорт даних . Формат JSON та XML доступні в версіях CVSS v2.0, v3.0, v3.1, and v4.0. JSON Schemas:

- https://www.first.org/cvss/cvss-v2.0.json - JSON Формат для CVSS v2.0.
- https://www.first.org/cvss/cvss-v3.0.json - JSON Формат для CVSS v3.0.
- https://www.first.org/cvss/cvss-v3.1.json - JSON Формат для CVSS v3.1.
- https://www.first.org/cvss/cvss-v4.0.json - JSON Формат для CVSS v4.0.

- https://nvd.nist.gov/schema/cvss-v2_0.2.xsd - XSD Формат для CVSS v2.0 підтримується на сайті NIST's.
- https://www.first.org/cvss/cvss-v3.0.xsd - XSD Формат для CVSS v3.0.
- https://www.first.org/cvss/cvss-v3.1.xsd - XSD Формат для CVSS v3.1.
- https://www.first.org/cvss/cvss-v4.0.xsd - XSD Формат для CVSS v4.0.

На основі вище згаданого можна зробити висновок NVD – зорієнтована на оцінку вразливостей але не має власного опису загроз і тому залежить від недоліків баз вразливостей, з яких вона отримує інформацію.

2.3. Методи оцінки рівня небезпек на основі GitHub Advisory

Консультативна база даних GitHub [32] містить список відомих вразливостей безпеки та шкідливого програмного забезпечення, згрупованих у три категорії: рекомендації, перевірені GitHub, неперевірені рекомендації та рекомендації щодо шкідливого програмного забезпечення.

Існує обмеження - кожна порада в базі даних GitHub Advisory Database стосується вразливостей у проектах з відкритим вихідним кодом або шкідливого програмного забезпечення з відкритим вихідним кодом.

Вони мають своє відзначення вразливості: вразливість — це проблема в коді проєкту, яка може бути використана для заподіяння шкоди конфіденційності, цілісності або доступності проєкту або інших проєктів, які використовують його код. Вразливості розрізняються за типом, серйозністю та методом атаки. Вразливості в коді зазвичай вводяться випадково і виправляються незабаром після їх виявлення.

На відміну від вразливості, зловмисне програмне забезпечення або зловмисне програмне забезпечення — це код, який навмисно розроблений для виконання небажаних або шкідливих функцій. Зловмисне програмне забезпечення може бути націлене на апаратне чи програмне забезпечення, конфіденційні дані або на користувачів будь-якої програми, яка використовує зловмисне програмне забезпечення.

GitHub розглядає рекомендації, якщо вони стосуються вразливостей у пакеті, який надходить із підтримуваного реєстру.

- Композитор (реєстр: https://packagist.org/)
- Erlang (реєстр: https://hex.pm/)
- Go (реєстр: https://pkg.go.dev/)
- Дії на GitHub (https://github.com/marketplace?type=actions/)
- Maven (реєстр: https://repo.maven.apache.org/maven2)
- Npm (реєстр: https://www.npmjs.com/)
- NuGet (реєстр: https://www.nuget.org/)

- Піп (реєстр: https://pypi.org/)
- Паб (реєстр: https://pub.dev/packages/registry)
- RubyGems (реєстр: https://rubygems.org/)
- Rust (реєстр: https://crates.io/)
- Swift (реєстр: Н/Д)

GitHub це доволі спеціалізоване але професійне середовище.

Кожна рекомендація з безпеки, незалежно від її типу, має унікальний ідентифікатор, який називається ідентифікатором GHSA. Кваліфікатор призначається, коли нова порада створюється на GitHub або додається до бази даних GitHub Advisory Database з будь-якого з підтримуваних джерел GHSA-ID

Синтаксис ідентифікаторів GHSA має такий формат: де: GHSA-xxxx-xxxx-xxxx

- x це буква або цифра з наступного набору: .23456789cfghjmpqrvwx
- Поза частиною назви: GHSA
- Цифри і букви призначаються випадковим чином.
- Всі букви написані маленькими літерами.

Консультативна база даних GitHub підтримує як CVSS версії 3.1, так і CVSS версії 4.0.

Кожна рекомендація щодо безпеки містить інформацію про вразливість або зловмисне програмне забезпечення, яка може включати опис, серйозність, уражений пакет, екосистему пакетів, уражені версії та виправлені версії, вплив, а також необов'язкову інформацію, таку як посилання, обхідні шляхи та кредити. Крім того, рекомендації зі списку Національної бази даних вразливостей містять посилання на запис CVE, де ви можете прочитати більш детальну інформацію про вразливість, її оцінки CVSS та рівень якісної серйозності. Для отримання додаткової інформації зверніться до «Національної бази даних вразливостей» від Національного інституту стандартів і технологій.

Рівень серйозності – це один із чотирьох можливих рівнів, визначених у розділі 5 «Загальної системи оцінювання вразливостей (CVSS)».

- Низький
- Середній/Помірний
- Високий
- Критичний

Консультативна база даних GitHub використовує рівні CVSS, описані вище. Якщо GitHub отримує CVE, консультативна база даних GitHub використовує версію CVSS, призначену супроводжувачем, яка може бути версії 3.1 або 4.0. Якщо CVE імпортовано, консультативна база даних GitHub підтримує версії CVSS 4.0, 3.1 та 3.0.

Система оцінки прогнозування експлойтів, або EPSS, – це система, розроблена глобальним Форумом команд реагування на інциденти та безпеки (FIRST) для кількісної оцінки ймовірності експлойту вразливості. Модель видає оцінку ймовірності від 0 до 1 (від 0 до 100%), де чим вищий бал, тим більша ймовірність того, що вразливість буде використана.

Консультативна база даних GitHub включає оцінки EPSS від FIRST для рекомендацій, що містять CVE, з відповідними даними EPSS. GitHub також відображає процентиль оцінки EPSS, який є часткою всіх оцінених вразливостей з однаковим або нижчим показником EPSS.

Наприклад, якщо консультативна особа мала показник EPSS у відсотках 90,534% на рівні 95-го процентиля, це означає, що:

- Існує 90,534% ймовірності того, що ця вразливість буде використана в дикій природі протягом наступних 30 днів.
- 95% від загальної кількості змодельованих вразливостей вважаються менш імовірними для використання в найближчі 30 днів, ніж ця вразливість.

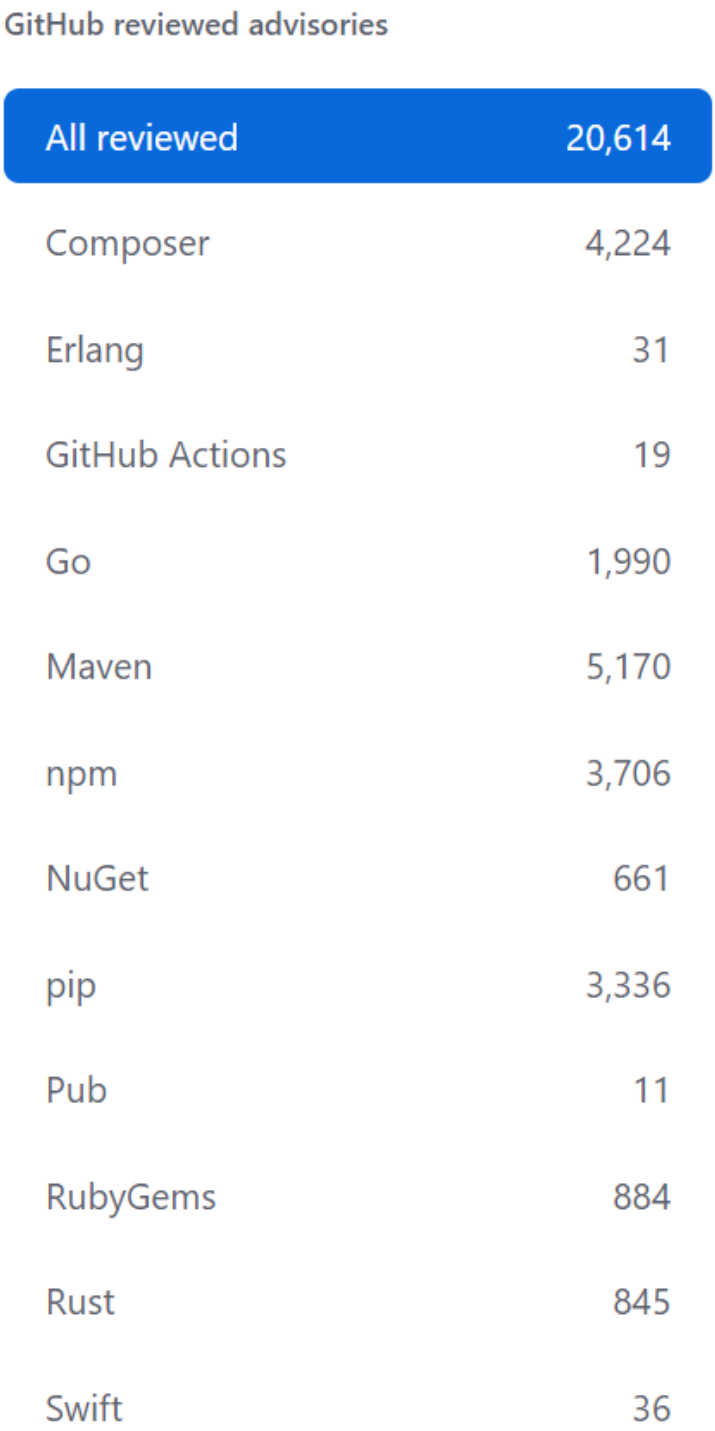


Рис. 2.12. Опис тематик GitHub

2.4. Методи оцінки рівня небезпек на основі FIRST

FIRST прагне об'єднати команди реагування на інциденти та безпеки з кожної країни світу, щоб забезпечити безпечний Інтернет для всіх [32].

Ефективне реагування є глобальним завданням, що відображає глобальний характер Інтернету. Ґрунтуючись на моделі однорангового управління мережею,

групи реагування на інциденти комп'ютерної безпеки (CSIRTs), групи реагування на інциденти безпеки продуктів (PSIRTs) і незалежні дослідники безпеки працюють разом, щоб обмежити шкоду від інцидентів безпеки. Для цього потрібен високий рівень довіри. Інциденти не обмежуються одним культурним чи політичним куточком Інтернету, і вони не поважають кордони чи кордони. Таким чином, FIRST сприяє інклюзивності.

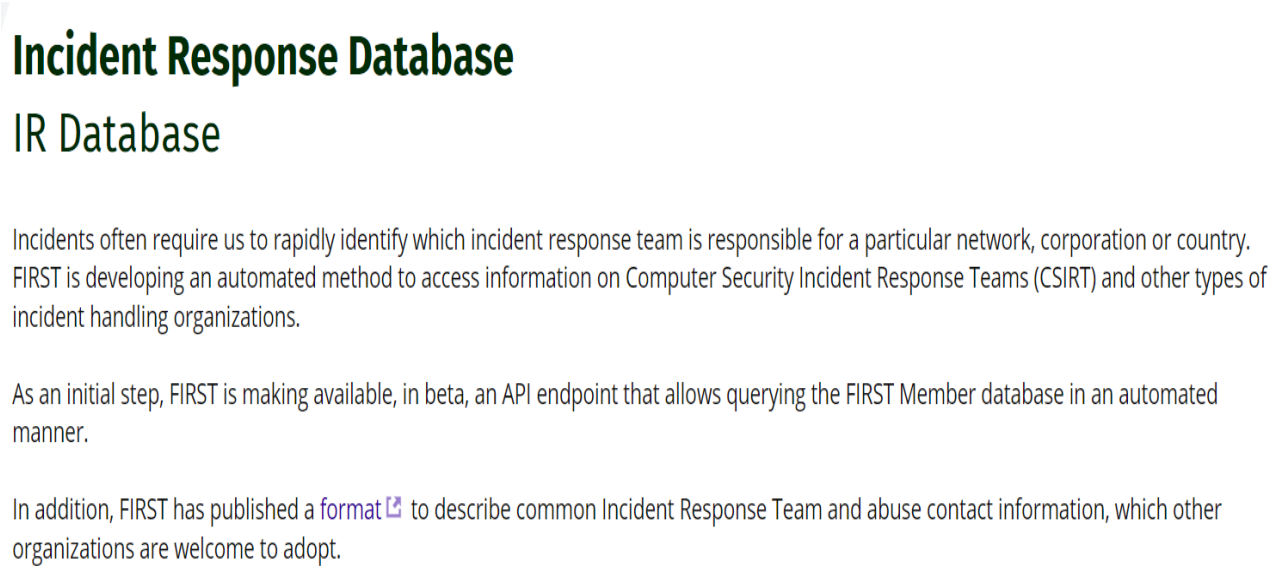


Рис. 2.13. Опис IR Database

2.5. Визначення загальних вимог

Аналіз показує, що в разі як ми бажаємо розробити конкурентне спроможну систему для оцінки рівня небезпек то маємо зробити систему з наступними властивостями:

• Вона має забезпечити імпорт різнорідних баз даних;

• Вона має фокусуватися на окремих вразливостях;

• Вона повинна мати відкритий код.

•Також необхідно створити компонент для перегляду файлів на локальному комп'ютері, мережі.

Висновок до розділу 2

В результаті виконання розділу 2 отримані наступні результати:

Зроблено аналіз особливостей методів оцінювання баз даних вразливостей Common Weakness Enumeration, National Vulnerability Database, GitHub Advisory, FIRST.

На основі цього аналізу було визначено загальні вимоги до системи, які будуть враховані при подальшій розробці.

В другому розділі проведено аналіз особливостей методів оцінювання баз даних вразливостей Common Weakness Enumeration, National Vulnerability Database, GitHub Advisory, FIRST. Головнм результатом цього аналізу було запропоновано надалі використовувати двоетапну схему аналізу. А саме, на основі аналізу було зроблено висновок CWE – зручна для вчасного інформування власників автоматизованих систем, але вона має нерегулярну структуру і потребує побудову інформаційних систем для її швидкої інтерпретації та обробки. Тому для її вдосконалення пропонується двоетапна схема аналізу:

На першому кроці визначається фокус уваги аналізу, з базі вразливостей відокремлюється підмножина загроз, які є фокусними для даного аналізу, далі будемо називати цю підмножину класом вразливостей.

На другому кроці проводиться більш детальний аналіз класу загроз. За рахунок однорідності підмножині вразливостей підвищується ефективність аналізу.

Важливість National Vulnerability Database Національної бази даних вразливостей для нашей роботи обумовлена рядом факторів. NVD – це репозиторій уряду США даних управління вразливостями на основі стандартів, представлених за допомогою протоколу автоматизації контенту безпеки (SCAP). Ці дані дозволяють автоматизувати управління вразливостями, вимірювання безпеки та відповідність. NVD включає бази даних із посиланнями на контрольні списки безпеки, недоліки програмного забезпечення, пов'язані з безпекою, назви продуктів і показники впливу. Всі вразливості в NVD мають ідентифікатор CVE.

Консультативна база даних GitHub містить, критічній для нашої розробки, список відомих вразливостей безпеки та шкідливого програмного забезпечення, згрупованих у три категорії: рекомендації, перевірені GitHub, неперевірені рекомендації та рекомендації щодо шкідливого програмного забезпечення.

Остання частка анализу стосується FIRST прагне об'єднати команди реагування на інциденти та безпеки з кожної країни світу, щоб забезпечити безпечний Інтернет для всіх.

Ефективне реагування є глобальним завданням, що відображає глобальний характер Інтернету. Ґрунтуючись на моделі однорангового управління мережею, групи реагування на інциденти комп'ютерної безпеки (CSIRTs), групи реагування на інциденти безпеки продуктів (PSIRTs) і незалежні дослідники безпеки працюють разом, щоб обмежити шкоду від інцидентів безпеки. Для цього потрібен високий рівень довіри. Інциденти не обмежуються одним культурним чи політичним куточком Інтернету, і вони не поважають кордони чи кордони. Таким чином, FIRST сприяє інклюзивності нашого аналізу.

РОЗДІЛ 3
РОЗРОБКА АЛГОРИТМІЧНОГО ЗАБЕЗПЕЧЕННЯ ПОШУКУ ОКРЕМИХ КЛАСІВ ВРАЗЛИВОСТЕЙ В СПЕЦІАЛІЗОВАНИХ БАЗАХ ДАНИХ

3.1 Вибір середовища реалізації

Для розробки системи було обрано середовище реалізації:

PyCharm 2024.2.1 (Community Edition)

Build #PC-242.21829.153, built on August 29, 2024

Runtime version: 21.0.3+13-b509.11 amd64 (JCEF 122.1.9)

VM: OpenJDK 64-Bit Server VM by JetBrains s.r.o.

Toolkit: sun.awt.windows.WToolkit

Windows 11.0

GC: G1 Young Generation, G1 Concurrent GC, G1 Old Generation

Memory: 1500M

Cores: 12

Registry:

ide.experimental.ui=true

i18n.locale=

Переваги:

- Гнучкість — це, основна перевага мови, так як завдяки своїй гнучкості мова отримала популярність серед багатьох розробників.
- Можливість розширення — один із слоганів мови звучить як — Just Import! — що повністю пояснює, наскільки мова розширюється і була розширена за останні роки. Існують бібліотеки і фреймворки під будь-який тип завдань і потреб. Також величезним плюсом є те, що ми можемо використовувати C код з Python.
- Простота синтаксису. Синтаксис – з синтаксису було прибрано все зайве, код чистий і зрозумілий без зайвих дужок і виразів.

- Інтерпретованість. Інтерпретатор Python існує для всіх популярних платформ і за замовчуванням входить в більшість дистрибутивів Linux, а значить є на більшості серверів «з коробки».
- PEP – єдиний стандарт для написання коду, що робить код підтримуваним і читабельним навіть при переході від одного програміста до іншого. Це підтримує популярність Python.
- Open Source – код інтерпретатора Python є відкритим, що дозволяє будь-кому, хто зацікавлений у розвитку мови взяти участь в його розробці і поліпшити його. Якщо дивитися деталі релізу однією з версій мови, то можна помітити, що величезні частини нового функціоналу реалізовані сторонніми розробниками.
- Ком'юніті – навколо Python утворилося досить дружнє і приємне ком'юніті, яке готове прийти на допомогу будь-якому починаючому або вже вмілому розробнику і розібратися в його проблемі.

Всі ці переваги цього середовище програмування визначили його вибір для реалізації коду на першому та другому кроках побудови системи.

Тестування здійснювалось на робочий станції конфігурації яка наведена на рис. 3.1.:

Обов’язковим є наявність широкого каналу зв’язку.

Можливим варіантом може бути Линукс машина критичним є наявність швидкого диску достатньо обсягу.

Операційні системи сучасних версії є необов’язковим але бажаними.

Вимоги по процесору демократичні. Вимоги до оперативної пам’яті відео карти мінімальні.

Версія USB бажана свіжіша.

Підключення через WiFi не бажане але цілком можливе.

Елемент	Значення
Назва ОС	Майкрософт Windows 10 Pro
Версія	10.0.19045 Збірка 19045
Інший опис ОС	Недоступно
Виробник ОС	Microsoft Corporation
Назва системи	DESKTOP-5GLMTGK
Виробник	Micro-Star International Co., Ltd
Модель	MS-7B86
Тип	x64-based PC
Обліковий номер системи	To be filled by O.E.M.
Процесор	AMD Ryzen 5 3600 6-Core Processor, 3600 МГц, ядер 6, логічних процесорі...
Версія BIOS/Дата	American Megatrends Inc. M.70, 17.06.2020
Версія SMBIOS	2.8
версія вбудованого контроле...	255.255
Модель BIOS	UEFI
Виробник системної плати	Micro-Star International Co., Ltd
Тип системної плати	B450-A PRO MAX (MS-7B86)
Версія системної плати	4.0
Роль платформи	Робочий стіл
Стан безпечного завантаження	Вимкнути
Конфігурація PCR7	Прив'язка неможлива
Папка Windows	C:\Windows
Системна папка	C:\Windows\system32
Пристрій завантаження	\Device\HarddiskVolume1
Мова	Росія
Апаратнозалежний рівень (HAL)	Версія = "10.0.19041.3636"
Ім'я користувача	DESKTOP-5GLMTGK\Анатолий
Часовий пояс	Фінляндія (зима)
Установлена фізична пам'ять (...	16,0 ГБ
Загальний обсяг фізичної пам'...	16,0 ГБ
Доступно фізичної пам'яті	8,69 ГБ
Усього віртуальної пам'яті	18,3 ГБ
Доступно віртуальної пам'яті	8,07 ГБ
Розмір файлу довантаження	2,38 ГБ
Файл довантаження	C:\pagefile.sys
Захист ПДП ядра	Вимкнути
Безпека на основі віртуалізації	Не ввімкнуто
Підтримка шифрування прист...	Можливі причини збою автоматичного шифрування пристрою: Модуль TP...
Hyper-V – розширення режим	

Рис 3.1 Апаратна конфігурація тестового комп’ютера

3.2 Ідентифікація одиничної вразливості

Для ідентифікації одиничної вразливості було написано тестовий фрагмент якій дозволяє отримати опис вразливості з бази даних.

```
import requests
import json

def get_cve_record(cve_id):
    url = f"https://cveawg.mitre.org/api/cve/{cve_id}"
    response = requests.get(url)

    if response.status_code != 200:
        print(f"Помилка при отриманні даних для {cve_id}")
        return None

    data = response.json()
    return data

def print_cve_details(cve_data):
    cve_id = cve_data.get('cveMetadata', {}).get('cveId', 'N/A')
    assigner = cve_data.get('cveMetadata', {}).get('assignerOrgId', 'N/A')
    state = cve_data.get('cveMetadata', {}).get('state', 'N/A')

    # Опис
    descriptions = cve_data.get('containers', {}).get('cna', {}).get('descriptions', [])
    description_text = descriptions[0].get('value', 'N/A') if descriptions else 'N/A'

    # Посилання
    references = cve_data.get('containers', {}).get('cna', {}).get('references', [])
    reference_urls = [ref.get('url', '') for ref in references]

    # Метрики CVSS
    metrics = cve_data.get('containers', {}).get('cna', {}).get('metrics', [])
    cvss_scores = {}

    for version in ['cvssV2_0', 'cvssV3_0', 'cvssV3_1', 'cvssV4_0']:
        for metric in metrics:
            if version in metric:
                cvss_data = metric[version]
                cvss_scores[version] = {
                    'score': cvss_data.get('baseScore', 'N/A'),
                    'severity': cvss_data.get('baseSeverity', 'N/A'),
                    'version': cvss_data.get('version', 'N/A'),
                    'vectorString': cvss_data.get('vectorString', 'N/A')
                }

    # Виводимо всі поля
    print(f"CVE ID: {cve_id}")
    print(f"Assigner: {assigner}")
    print(f"State: {state}")
```

Рис 3.2 Текст коду підпрограм

Він включає дві підпрограми, які будуть використовуватися далі, це Get_cve_record та Print_cve_details. Get_cve_record — отримує інформацію безпосередньо з сервера Митре адреса https://cveawg.mitre.org/api/cve/{cve_id} та перевіряє безпомилковість цієї операції. Print_cve_details — дешифрує інформацію та виводу її на екран.

Як ми бачимо код добре структуровано. Він включає блоки:

Опис;

Посилання;

Метрики CVSS;

Виводу всіх полів;

Використання функцій.

Опис — описує ;) дескриптори.

Посилання — визначає references.

Метрики CVSS задає відповідні метрики, та аналізує іх версію.

Блок Виводу всіх полів визначає послідовність та формат друку.

Використання функцій. Задає діалог вводу номеру вразливості.

Проілюструвати її працездатність на прикладі CVE-2022-20809. Для цього запустимо в компілятору Pyton проект CVE-ID. Та виконаємо його тестування.

C:\Users\kot\AppData\Local\Programs\Python\Python312\python.exe

C:\KOT\CVE-ID.py

Введіть CVE ID (наприклад, CVE-2022-20809): CVE-2022-20809

CVE ID: CVE-2022-20809

Assigner: d1c1063e-7a18-46af-9102-31f8928bc633

State: PUBLISHED

Description: Multiple vulnerabilities in the API and web-based management interfaces of Cisco Expressway Series and Cisco TelePresence Video Communication Server (VCS) could allow an authenticated, remote attacker to write files or disclose

sensitive information on an affected device. For more information about these vulnerabilities, see the Details section of this advisory.

References:

https://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-expressway-filewrite-bsFVwueV

Metrics:

cvssV3_1:

Score: 4.3

Severity: MEDIUM

Version: 3.1

Vector String: CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:N/A:N

Process finished with exit code 0

Як ми бачимо в тестовому фрагменті здійснюється інтерактивний запит номеру вразливості, перевіряється доступність бази даних загроз та здійснюється зчитування, потім , в випадку безпомилкового зчитування, вивід інформації на екран.

Тестування довело повну працездатність розроблених підпрограм.

3.3 Побудова розширеного запиту на визначеному класі вразливостей

База даних загроз динамічно змінюється, тому є сенс фіксувати множину вразливостей окремого класу, формуючи для цього підмножини вразливостей на момент запиту, за рахунок цього спрощується подальша обробка та знижується складність подальшого аналізу. Виразимо це через алгоритм, циклічне виконання якого є базою системи аналізу.

В першу чергу визначаємо, який клас вразливостей будемо аналізувати. Наступним кроком сформуємо відповідний запит. На цьому кроку є технологічні особливості, які ми розглянемо далі при аналізі коду.

Далі виконуємо GET – запит та зберігаємо інформацію що отримано в результати його виконання.

Версії формату JSON залежать від версії CVSS це більш детально було описано в параграфі 2.2. стор. 50.

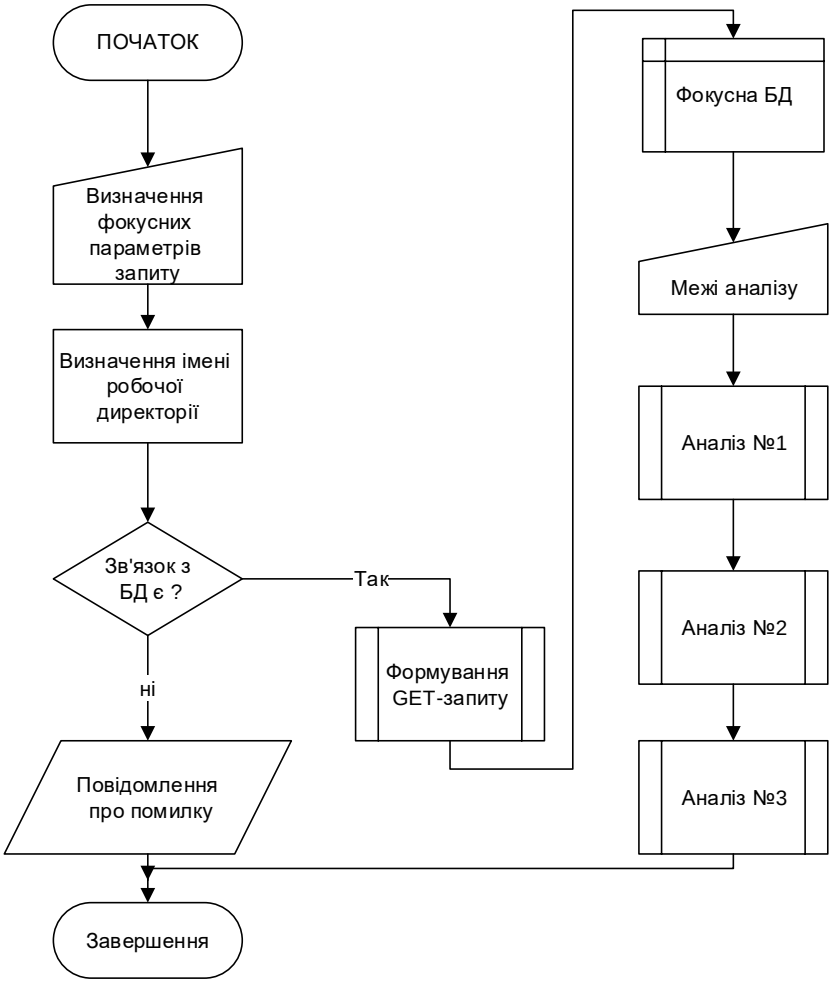


Рис. 3.3. Базовий алгоритм виділення класу вразливостей.

Розглянемо фрагмент коду. Спочатку визначимося з URL

```
# URL для запиту
url = "https://services.nvd.nist.gov/rest/json/cves/2.0?cpeName=cpe:2.3:o:microsoft:windows_10:19045"
```

Рис. 3.4. Визначення URL

Потім знайдемо шлях до активного каталогу.

```
# Отримання директорії, в якій знаходиться скрипт
script_dir = os.path.dirname(os.path.abspath(__file__))
file_path = os.path.join(script_dir, 'cve_data.json')  # повний шлях до файлу в тому ж каталозі, що і скрипт

try:
    # Видображення поточної директорії
    print(f"Поточна директорія скрипта: {script_dir}")
```

Рис. 3.5. Пошук шляху до активного каталогу.

Далі виконаємо запит та збережемо дані в формати JSON в активному каталозі. В фрагменті показані також деякі технологічні аспекти а саме обробка помилок та деякі нюанси форматування даних для відображення.

Фрагмент коду виконання GET – запиту зображено на рис. 3.6. та рис. 3.7

```
    # Виконання GET-запиту
    response = requests.get(url)

    # Перевірка успішності виконання запиту
    response.raise_for_status()  # Викликає помилку для невдалих відповідей (ні 200)

    # Отримання даних у форматі JSON
    data = response.json()
    # Збереження даних у файл
    with open(file_path, 'w') as json_file:
        json.dump(data, json_file, indent=4)  # indent для красивого форматування

    print(f"Дані збережено {file_path}")

except requests.exceptions.RequestException as e:
    print(f"Помилка при виконанні запиту: {e}")
except IOError as e:
    print(f"Помилка запису: {e}")



    # Збереження даних у файл
    with open(file_path, 'w') as json_file:
        json.dump(data, json_file, indent=4)  # indent для красивого форматування

    print(f"Дані збережено {file_path}")

except requests.exceptions.RequestException as e:
    print(f"Помилка при виконанні запиту: {e}")
except IOError as e:
    print(f"Помилка запису: {e}")
```

Рис. 3.6. Фрагмент коду виконання GET – запиту

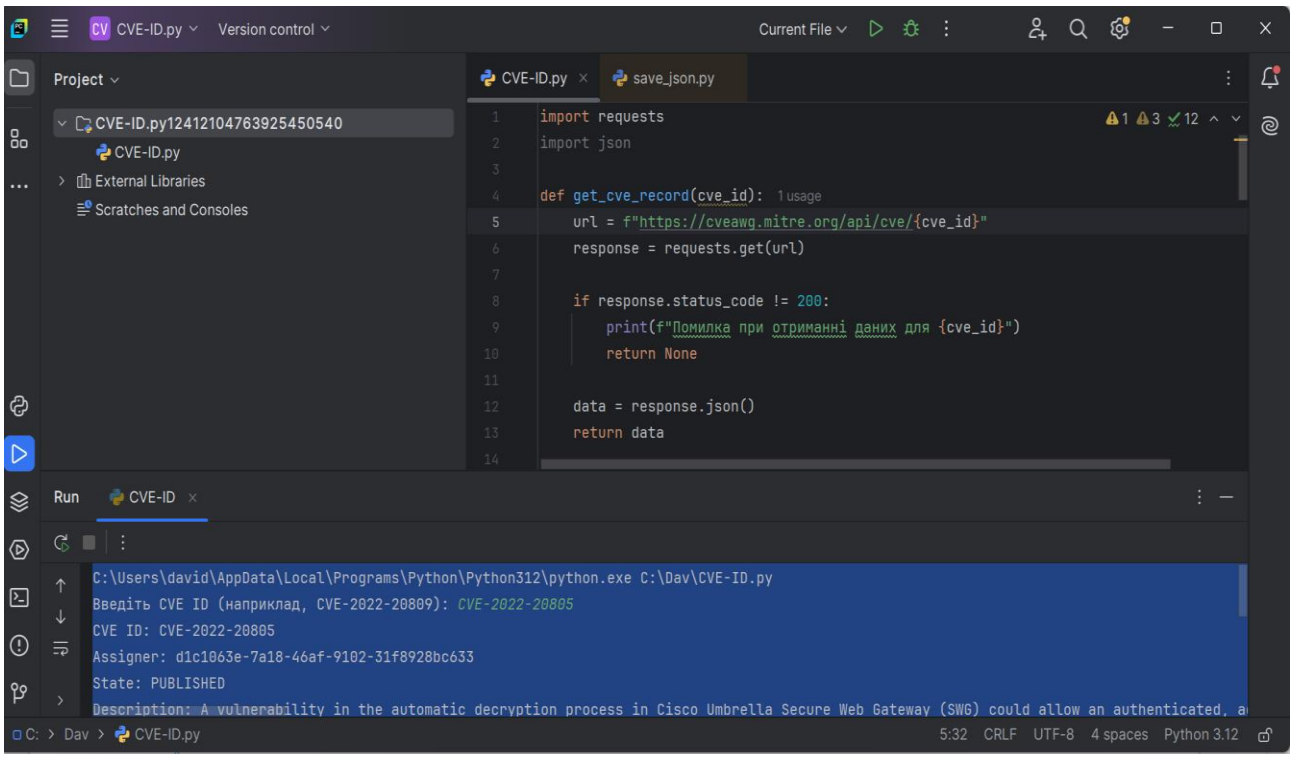


Рис. 3.7. Фрагмент коду виконання GET – запиту

3.4 Оцінювання вразливостей для моделі безпеки

Оцінювання вразливостей залежить від характеристик та бізнес спрямованості підприємства, яке підлягає оцінці. Побудуємо такую залежність:

a) Критичність інформаційного процесу для профільного бізнес процесу підприємства. Експертна оцінка рівня значення для загальної діяльності підприємства інформаційних технологій.

b) Процеси та активи оцінки ризиків при їх взаємодії. Існують різні топології взаємодії між процесами, активами та їх власниками. Це допомагає зрозуміти та моделювати погодження з існуючими ризками.

c) Матеріальне забезпечення процесу оцінки ресурсами людськими, фінансовими, часовими. Достатність підтримки ресурсами процесу оцінки ризиків від існуючих вразлівостей є головним фактором його успіху.

d) Правові вимоги законодавства, регуляторів та інших зацікавлених сторін до процесу керування ризиками інформаційної безпеки також потребують оцінки.

На основі залежності, що пропонується, будується модель оцінювання вразливостей підприємства, вона має дві частини орієнтовані на специфіку діяльності та потреби конкретного підприємства та незалежну. Обидві обираються виходячи з вище перелічених умов, існує оптимальний вибір різних методологій оцінки вразливостей. Незалежно від специфіки підприємства, частина має:

1. Забезпечувати відтворюваність результатів при повторном застосуванні методіки оцінювання, забезпечуючи однозначність та консистентність у оцінці ризиків.
2. Бути зрозумілою і прозорою. Недопускати багатозначної інтерпретації результатів оцінки, що забезпечує високу ступінь довіри та сприяє ефективному управлінню ризиками.
3. Відповідати потребам підприємства, бути адаптованою до його специфіки.
4. Відображати внутрішню структуру підприємства та наявних ресурсів, що включає адекватність до наявності необхідного обладнання та рівня технічної підготовки персоналу.
5. Віддзеркалювати реальну ситуацію з переліком небезпек, актуальних для підприємства, з урахуванням його унікальних ризиків.

Наступний крок - це фокусування точок зору для визначення моделі оцінювання вразливостей, існує класіфікація по категоріям Власник, Процес, Актив, Вразлівість.

1. Власники процесів: Ця категорія включає всі відділення, департаменти, а також окремих співробітників, що відповідають за виконання, моніторинг та підтримку різних бізнес-процесів.
2. Процеси: У цій категорії розглядаються всі процеси, у яких циркулює критична інформація (ІзОД).
3. Активи: Оцінюються всі активи підприємства, які мають цінність.
4. Вразливости: Розробляється або адаптується база даних вразливостей, яка містить інформацію про кожну загрозу, її вплив на

конфіденційність, цілісність та доступність активів, а також оцінку ймовірності її реалізації.

На основі цього розподілу можна розробити формули оцінки для кожної критичної точки зору з урахуванням раніше визначених аспектів оцінювання. Це дозволить підприємству краще поводитися з вразливостями.

Окрім оцінки, не меньш важливим є моніторинг вразливостей. Тому задача дослідження її доповнює та розширює.

Висновок до розділу 3

В результаті виконання розділу 3 отримані наступні результати: обрано середовище розробки для формування запитів до баз даних; протестовано обмін з базою даних вразливостей та визначено формати обміну та збереження інформації; запропоновано базовий алгоритм виділення класу вразливостей та розроблені підпрограми для його реалізації.

В третьому розділі здійснено розробку алгоритмічного забезпечення пошуку окремих класів вразливостей в спеціалізованих базах даних. Яка виконана в чотири кроки:

Обрано середовище реалізації, а саме прийнято рішення про двоетапну реализацію перша PyCharm 2024.2.1 та друга для інтерактівної частині VisualStudio.17.Release/17.11.3+35303.130. Це дозволило використати переваги обох підходів.

Для ідентифікації одиничної вразливості було написано тестовий фрагмент, який дозволив отримати опис вразливості з бази даних. Він включає дві підпрограми, які будуть використовуватися далі, це Get_cve_record та Print_cve_details. Get_cve_record — отримує інформацію безпосередньо з сервера Митре адреса https://cveawg.mitre.org/api/cve/{cve_id} та перевіряє безпомилковість цієї операції. Print_cve_details — дешифрує інформацію та виводу її на екран.

База даних загроз динамічно змінюється, тому є сенс фіксувати множину вразливостей окремого класу формуючи для цього підмножини вразливостей на момент запиту за рахунок цього спрощується подальша обробка та знижується складність подальшого аналізу. Для цього було запропоновано базовий алгоритм виділення класу вразливостей, циклічне виконання якого є базою системи аналізу.

В алгоритмі в першу чергу визначаємо, який клас вразливостей будемо аналізувати. Наступним кроком сформуємо відповідний запит. На цьому кроку є технологічні особливості, які ми розглянемо далі при аналізі коду. Далі виконуємо GET – запит та зберігаємо інформацію що отримано в результати його виконання. Версії формату JSON залежать від версії CVSS це більш детально було описано в параграфі 2.2. стор. 50. Далі виконаємо запит та збережемо дані в формати JSON в активному каталозі. В фрагменті показані також деякі технологічні аспекти, а саме обробка помилок та деякі нюанси форматування даних для відображення.

Оцінювання вразливостей залежить від характеристик та бізнес спрямованості підприємства, яке підлегає оцінці. Побудувано такую залежність: критичність інформаційного процеса для профільного бізнес процесу підприємства. Експертна оцінка рівня значення для загальної діяльності підприємства інформаційних технологій. Процеси та активи оцінка ризиків при їх взаїмодії. Існують різні топології взаїмодії між процесами, активами та їх власниками. Це допомагає зрозуміти та моделювати погодження з існуючими ризками. Матеріальне забезпечення процесу оцінки ресурсами людськими, фінансовими, часовими. Достатність підтримки ресурсами процесу оцінки ризиків від існуючих вразлівостей є головним фактором його успіху. Правові вимоги законодавства, регуляторів та інших зацікавлених сторін до процесу керування ризиками інформаційної безпеки також потребують оцінки.

На основі залежності, що пропонується будується модель оцінювання вразливостей підприємства, вона має дві частини орієнтовані на специфіку діяльності та потреби конкретного підприємства та незалежну. Обидві

обираються виходячи з вище перелічених умов, існує оптимальний вибір різних методологій оцінки вразливостей.

РОЗДІЛ 4
РОЗРОБКА ТА ТЕСТУВАННЯ СИСТЕМИ ОЦІНКИ ВРАЗЛИВОСТЕЙ НА ОСНОВІ СПЕЦІАЛІЗОВАНИХ БАЗАХ ДАНИХ

4.1 Вибір середовища реалізації

Інтерактивну частині системи оцінки вразливостей доцільно робити в програмному середовищі типу VisualStudio. Було обрано -

Microsoft Visual Studio Community 2022 Version 17.11.3. Або повністю VisualStudio.17.Release/17.11.3+35303.130.

Також необхідні компоненти:

Microsoft .NET Framework

Version 4.8.09032

Installed Version: Community

Visual C++ 2022 00482-90000-00000-AA273

Microsoft Visual C++ 2022

ASP.NET and Web Tools 17.11.231.19466

ASP.NET and Web Tools

Azure App Service Tools v3.0.0 17.11.231.19466

Azure App Service Tools v3.0.0

Azure Functions and Web Jobs Tools 17.11.231.19466

Azure Functions and Web Jobs Tools

C# Tools 4.11.0-3.24428.4+1ea9c390a5bb6815fdff2137ee155e23e78d6ff3

Для C# компонент, що використовуються в IDE, може використовуватися різна версія компілятора. Це залежить від типу проєкту та налаштувань.

Common Azure Tools 1.10 - Надає загальні послуги для використання Azure Mobile Services and Microsoft Azure Tools.

GitHub Copilot 0.2.1648.49400

GitHub Copilot це сервіс AI, який допомагає писати код швидше та з меншими витратами зусиль.

Microsoft JVM Debugger 1.0 - Забезпечує підтримку підключення налагоджувача Visual Studio до віртуальних машин Java, сумісних з JDWP

NuGet Package Manager 6.11.0

Менеджер пакетів NuGet у Visual Studio. Для отримання додаткової інформації про NuGet відвідайте https://docs.nuget.org/

Razor (ASP.NET Core) 17.11.3.2442001+68650a7d94261bc56a1f4bc522c2ee35314b1abb

Надає мовні послуги для ASP.NET Core Razor.

SQL Server Data Tools 17.11.38.0

Microsoft SQL Server Data Tools

Test Adapter for Boost.Test 1.0 -

Включає інструменти тестування Visual Studio з модульними тестами, написаними для Boost.Test.

Test Adapter for Google Test 1.0

Включає інструменти тестування Visual Studio з модульними тестами, написаними для Google Test.

TypeScript Tools 17.0.30715.2002

TypeScript Tools for Microsoft Visual Studio

Visual Basic Tools 4.11.0-3.24428.4+1ea9c390a5bb6815fdff2137ee155e23e78d6ff3

Visual F# Tools 17.11.0-beta.24421.7+af2f522de602281d4ef5a7b71507c428e814c5c1

Microsoft Visual F# Tools

Visual Studio IntelliCode 2.2

AI- помічник для розробки в Visual Studio.

Враховуючи технологічні обмеження визначені вище згаданим вибором розробимо структурну схему системи.

4.2 Вдосконалення методу оцінки рівня небезпек

В параграфі 3.3 запропоновано базовий алгоритм виділення класу вразливостей див. Рис. 3.3 . Побудуємо на його основі вдосконалений метод оцінки. В першому розділи було розглянуто декілька методів оцінки. Всі вони основані на повноформатному пошуку в базі загроз. Великій обсяг даних веде до складностей при аналізі вразливостей. Більш того зміни в базі даних можуть вести до помилок в процесі аналізу. Тому пропонується робити скорочену репліку бази даних на основі якої подальший аналіз буде більш простий.

Враховуючи вище згадане сформулюємо метод:

Крок1 Формування фокусного вектору аналізу.

Крок2 Формування меж множини вразливостей на основі проєкції фокусного вектору на повноформатну базу даних.

Крок3 Формування фокусної спеціалізованої бази вразливостей шляхом підготовки та виконання GET — запиту.

Крок4 Попередній аналіз фокусної спеціалізованої бази вразливостей та прийняття рішення про скорочення або розширення та типу аналізу.

Крок5 Безпосередній аналіз на фокусної спеціалізованої бази вразливостей

Крок6 Формування звітів.

На основі простої оцінки час проведення аналізу при такому підході скорочується приблизно в n/m раз де n розмір повноформатної базі даних, m розмір фокусної спеціалізованої бази вразливостей.

В деяких ситуаціях - це може дати критичну перевагу.

Для перевірки реалізуємо метод програмно.

4.3 Програмна реалізація методу оцінки рівня небезпек

Розроблюваний програмний модуль буде ґрунтуватися на принципах експертної оцінки, акцентуючи на важливості залучення користувача з глибокими знаннями про особливості підприємства для оцінки .

Необхідно створити зручні механізми введення даних, щоб користувачі могли легко і точно вносити інформацію. Це включає інтерфейси користувача, шаблони для даних, інтеграцію з базами даних, що вже існують, а також функції аналізу і візуалізації даних.

Система також має бути гнучкою та налаштованою, щоб враховувати зміни в бізнес-процесах, технологіях та регуляторних вимогах, забезпечуючи актуальність та відповідність оцінок реальному стану підприємства.

Загальний обсяг коду 26920 рядків, в додатку А наведено обрані фрагменти коду приблизно 1300 рядків це приблизно 5 процентів коду. Обмеження на кількість сторінок, на жаль, не дає прокоментувати все, тому обмежимося окремими ілюстраціями.

Для прискорення пошуку та ідентифікації логічним критеріям поставлені шістнадцятирічні сигнатури. Наприклад :

"vulnerable": true,
"criteria": "cpe:2.3:o:microsoft:windows_10:*:*:*:*:*:*:*:*",
"matchCriteriaId": "FBC814B4-7DEC-4EFC-ABFF-08FFD9FD16AA"

Тобто для CPE версії 2.3. Операційної системи - windows_10:*:*:*:*:*:*:*:* відповідає сигнатура FBC814B4-7DEC-4EFC-ABFF-08FFD9FD16AA.

Для опису вразливостей будемо використовувати структуру cve

"cve": {

яка включає:

"id": "CVE-2016-0089",

ім’я вразливості

"sourceIdentifier": "secure@microsoft.com",

джерело ім’я

"published": "2016-04-12T23:59:01.583",

дату видання

"lastModified": "2018-10-12T22:11:00.910",

дату останього корегування

"vulnStatus": "Modified",

статус

"cveTags": [],

посилання

"descriptions": [

опис

{

"lang": "en",

мова опису зазвичай англійська

"value": "Hyper-V in Microsoft Windows 8.1, Windows Server 2012 Gold and R2, and Windows 10 allows guest OS users to obtain sensitive information from host OS memory via a crafted application, aka \"Hyper-V Information Disclosure Vulnerability.\""

відповідно - Hyper-V в Microsoft Windows 8.1, Windows Server 2012 Gold і R2, а також Windows 10 дозволяє гостьовим користувачам ОС отримувати конфіденційну інформацію з пам'яті хост-ОС за допомогою створеної програми, відомої як «вразливість розкриття інформації Hyper-V».

},

{

"lang": "es",

"value": "Hyper-V en Microsoft Windows 8.1, Windows Server 2012 Gold y R2 y Windows 10 permite a usuarios del SO invitado obtener informaci\u00f3n sensible de la memoria del SO anfitri\u00f3n a trav\u00e9s de una aplicaci\u00f3n manipulada, tambi\u00e9n conocida como \"Hyper-V Information Disclosure Vulnerability\"."

```
        }
      ],
```


Далі іде опис метрик:


```
      "metrics": {
        "cvssMetricV30": [
          {
            "source": "nvd@nist.gov",
            "type": "Primary",
```


Цікавим для користувача є набір cvss - даних


```
            "cvssData": {
              "version": "3.0",
```


Опис в вигляду вектору -


```
              "vectorString":
"CVSS:3.0/AV:L/AC:L/PR:N/UI:N/S:C/C:H/I:N/A:N",
```


Напрям атаки


```
              "attackVector": "LOCAL",
```


Складність атаки


```
              "attackComplexity": "LOW",
```


Наявність привілей


```
              "privilegesRequired": "NONE",
```


Інтерактивна участь користувача


```
              "userInteraction": "NONE",

             "scope": "CHANGED",
```


Низка імпактів - конфіденційність


```
              "confidentialityImpact": "HIGH",
```


цілісність


```
              "integrityImpact": "NONE",
```


доступність


```
            "availabilityImpact": "NONE",
```

Кількісна та якісна оцінка

```
                "baseScore": 7.1,
                "baseSeverity": "HIGH"
            },
```

Експлойт оцінка

```
            "exploitabilityScore": 2.5,
```

Загальна імпакт оцінка

```
            "impactScore": 4.0
        }
    ],
```

Цієї частки опису достатньо для ілюстрації користі даної структурі, повний опис набагато більше та включає різні версії cvss — даних.

Розглянемо частку коду системи, яка реалізує запропонованій на рис 3.3 підхід. А саме, зчитує обраний файл відображає інформацію у відповідній екранній формі та виконує обраний аналіз.

Опис використовуваних модулів

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using Spire.Xls;

namespace vulnerability_analysis
{
```

Ініціація головної екранної формі

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

    private void Form1_Load(object sender, EventArgs e)
    {
```

Очищення множини вразливостей що аналізується

```
        cve_list.Rows.Clear();
```

Ініціація робочої кнізі

```
        Workbook workbook1 = new Workbook();
        workbook1.LoadFromFile(@"cve.xls");
        Worksheet worksheet1 = workbook1.Worksheets[0];
        DataTable dt1 = worksheet1.ExportDataTable();
        cve_list.DataSource = dt1;

//cve_list.AutoResizeColumns(DataGridViewAutoSizeColumnsMode.Fill);

cve_list.AutoResizeColumns(DataGridViewAutoSizeColumnsMode.AllCells);
```

Очищення та завантаження таблиці аналізу

```
        analis_table.Rows.Clear();
        Workbook workbook2 = new Workbook();
        workbook2.LoadFromFile(@"anz.xls");
        Worksheet worksheet2 = workbook2.Worksheets[0];
        DataTable dt2 = worksheet2.ExportDataTable();
        analis_table.DataSource = dt2;
```

```
        }

        private void button2_Click(object sender, EventArgs e)
        {

            BindingSource bs = new BindingSource();

            // Призначаємо джерело даних, яке, ймовірно, вже було призначено
DataGridView
            bs.DataSource = analis_table.DataSource;

            // Застосувати фільтр для BindingSource
            bs.Filter = "[CVE ID] = 'CVE-2015-6184'";

            // Встановлюємо відфільтрований BindingSource  в DataGridView
            analis_table.DataSource = bs;

        }

        private void button3_Click(object sender, EventArgs e)
        {
            try
            {
                analis_table.Rows.Clear();
                Workbook workbook2 = new Workbook();
                workbook2.LoadFromFile(@"anz.xls");
                Worksheet worksheet2 = workbook2.Worksheets[0];
                DataTable dt2 = worksheet2.ExportDataTable();
```

```
                if (dt2.Rows.Count > 0)
                {
                    analis_table.DataSource = dt2;

analis_table.AutoResizeColumns(DataGridViewAutoSizeColumnsMode.AllCells);  //
Автоматичний розмір за вмістом
                }
                else
                {
                    MessageBox.Show("Дани не знайдені в файлу Excel.");
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show("Помилка: " + ex.Message);
            }

        }

        private void button4_Click(object sender, EventArgs e)
        {

        }

        private void button4_Click_1(object sender, EventArgs e)
        {
         //
webBrowser1.Navigate("https://services.nvd.nist.gov/rest/json/cves/2.0?cpeName=cp
e:2.3:o:microsoft:windows_10:19045.4894");
        }
```

```
        private void пошукЗагрозToolStripMenuItem_Click(object sender,
EventArgs e)
        {

        }

        private void вихідToolStripMenuItem_Click(object sender, EventArgs e)
        {
```

Таким чином ми розглянули реалізацію функцій завантаження та декодування інформації. А також формування множини вразливостей для подальшого аналізу.

4.4 Тестування системи

Експериментальне дослідження розробленої системи з оцінки вразливостей виконується з метою перевірки його ефективності, правильності реалізації блок-схеми та актуальності розробки .

Основна мета цього експерименту полягає у визначенні, наскільки адекватно розроблена система здатна оцінювати вразливості, з якими може зіштовхнутися комерційне підприємство. Це включає перевірку як загальної працездатності модуля, так і його спроможності виконувати специфічні завдання, пов'язані з оцінкою вразливостей.

Ключові завдання експерименту включають:

1. Перевірка коректності роботи форм користувача: Це включає оцінку інтуїтивності інтерфейсу, зручності введення даних та загальної функціональності користувацького інтерфейсу.
2. Можливість редагування інформації під час оцінювання.

3. Вивід результатів згідно розробленого алгоритму: Аналіз результатів, які генеруються модулем, з метою переконатися у їх відповідності запланованій логіці та оціночному алгоритму.

4. Адекватність оцінок вразливостей: Оцінка того, наскільки точно та об'єктивно програмний модуль визначає потенційні вразливості та їх наслідки для підприємства. Основною ціллю експерименту є підтвердження, що розроблена система відповідає всім встановленим вимогам та є надійним інструментом для оцінки вразливостей у комерційному середовищі. Це включає перевірку його здатності надавати користувачам точну, надійну та зрозумілу інформацію про вразливостей, які можуть вплинути на їхню діяльність.

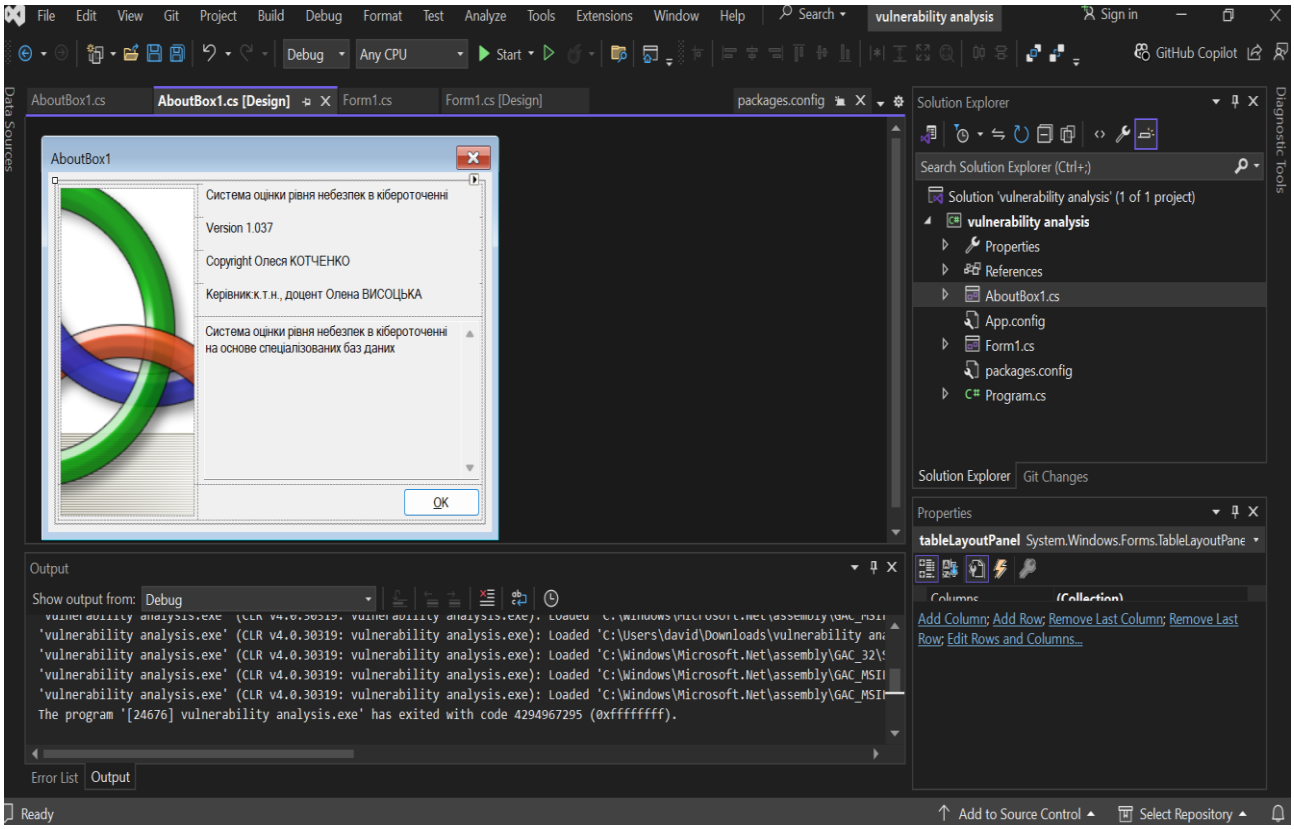


Рис. 4.1. Робоче вікно Microsoft Visual Studio під час тестування Системі оцінки рівня небезпек в кібероточенні на основі спеціалізованих баз даних.

Програмне забезпечення - «Система оцінки рівня небезпек в кібероточенні на основі спеціалізованих баз даних» розроблено в межах магістерської роботи.

Виконавець: Олеся КОТЧЕНКО. Керівник: к.т.н., доцент Олена ВИСОЦЬКА Версія 1.037 від 11.11.2024.

Основне призначення — відображає дані про вразливості в зручної для користувача формі. Формує множину вразливостей та виконує задані типи аналізу. Має відкриту структуру и може бути розширена за рахунок додавання інших видів аналізу.

Має зручне меню та три основних екрані формі.

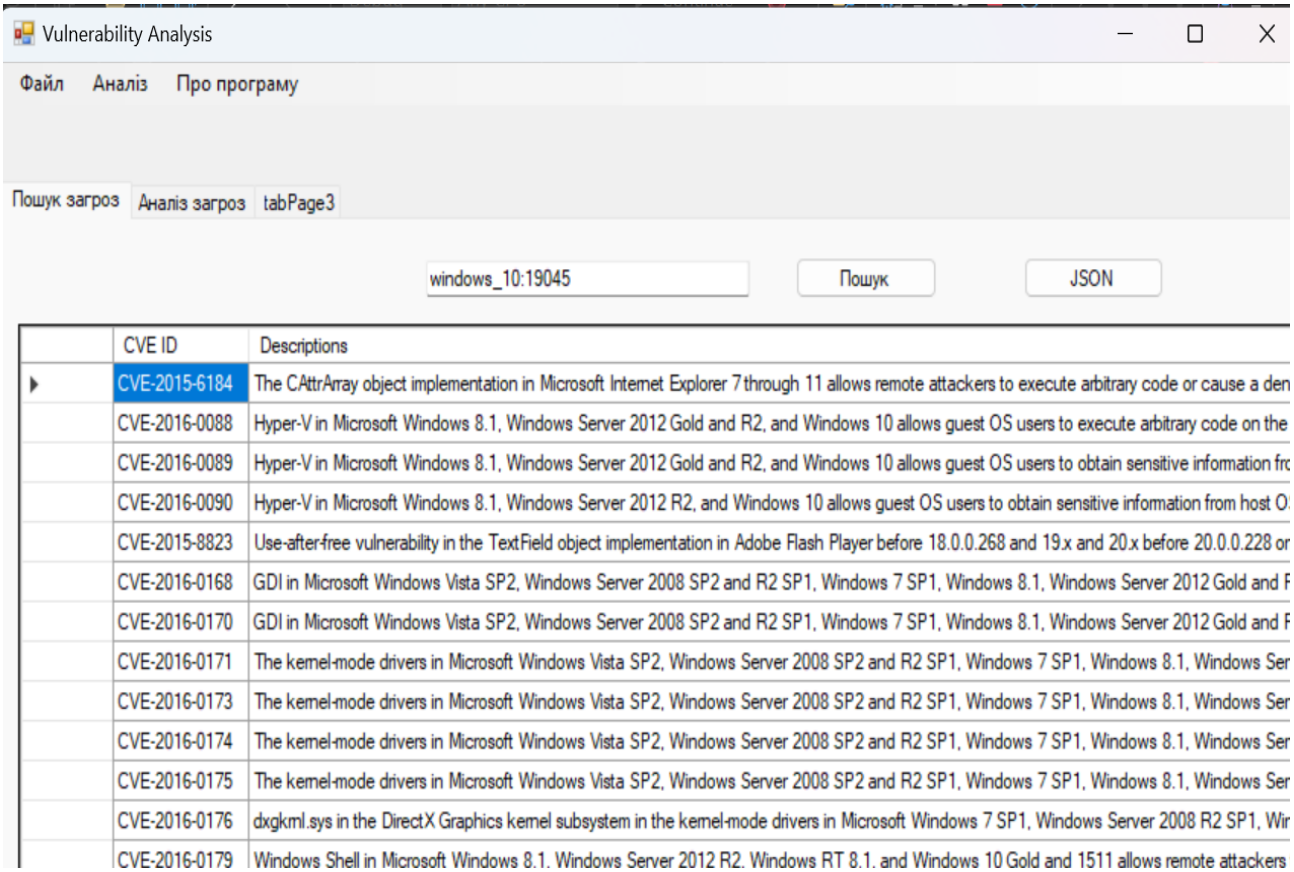


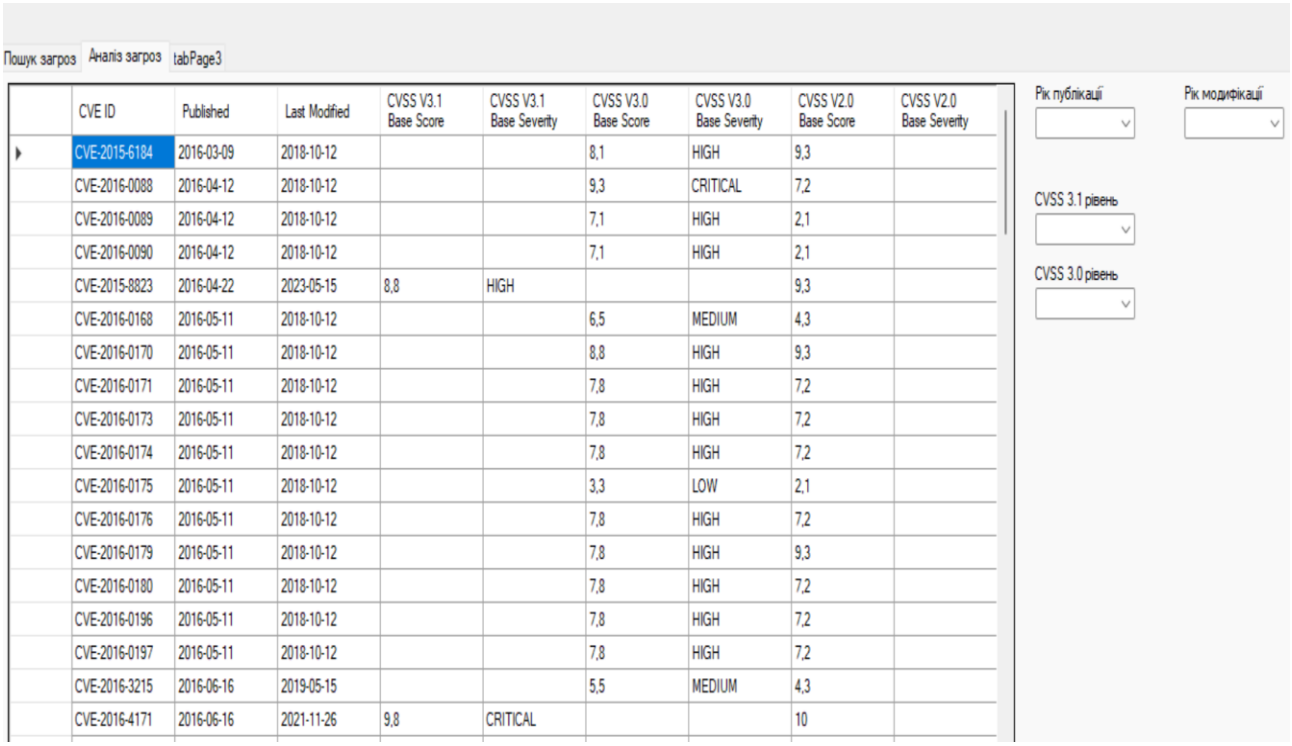
Рис. 4.2 Форма пошуку вразливостей

Форма пошуку вразливостей (Рис 4.2) відображає ідентифікатор та дескриптор вразливості.

Ідентифікатор це унікальній номер вразливості відповідно до базі CVE.

Дескриптор вразливості містить опис вразливості.

Наступна форма відбору вразливостей для аналізу. Містить ідентифікатор та критичну для аналізу інформацію. Також на цієї формі присутні вікна відбору, які дозволяють обмежити множину вразливостей для подальшого аналізу.



CVE ID	Published	Last Modified	CVSS V3.1 Base Score	CVSS V3.1 Base Severity	CVSS V3.0 Base Score	CVSS V3.0 Base Severity	CVSS V2.0 Base Score	CVSS V2.0 Base Severity
CVE-2015-6184	2016-03-09	2018-10-12			8,1	HIGH	9,3	
CVE-2016-0088	2016-04-12	2018-10-12			9,3	CRITICAL	7,2	
CVE-2016-0089	2016-04-12	2018-10-12			7,1	HIGH	2,1	
CVE-2016-0090	2016-04-12	2018-10-12			7,1	HIGH	2,1	
CVE-2015-8823	2016-04-22	2023-05-15	8,8	HIGH			9,3	
CVE-2016-0168	2016-05-11	2018-10-12			6,5	MEDIUM	4,3	
CVE-2016-0170	2016-05-11	2018-10-12			8,8	HIGH	9,3	
CVE-2016-0171	2016-05-11	2018-10-12			7,8	HIGH	7,2	
CVE-2016-0173	2016-05-11	2018-10-12			7,8	HIGH	7,2	
CVE-2016-0174	2016-05-11	2018-10-12			7,8	HIGH	7,2	
CVE-2016-0175	2016-05-11	2018-10-12			3,3	LOW	2,1	
CVE-2016-0176	2016-05-11	2018-10-12			7,8	HIGH	7,2	
CVE-2016-0179	2016-05-11	2018-10-12			7,8	HIGH	9,3	
CVE-2016-0180	2016-05-11	2018-10-12			7,8	HIGH	7,2	
CVE-2016-0196	2016-05-11	2018-10-12			7,8	HIGH	7,2	
CVE-2016-0197	2016-05-11	2018-10-12			7,8	HIGH	7,2	
CVE-2016-3215	2016-06-16	2019-05-15			5,5	MEDIUM	4,3	
CVE-2016-4171	2016-06-16	2021-11-26	9,8	CRITICAL			10	

Рис. 4.3 Форма відбору вразливостей для аналізу

Зовнішній вигляд та функціонал меню показано на рис 4.4-6

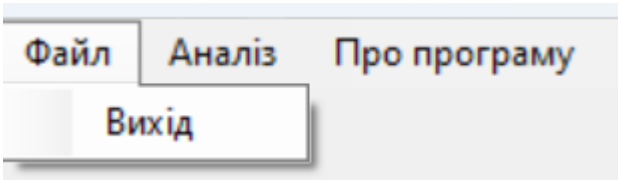


Рис. 4.4 Форма пошуку вразливостей

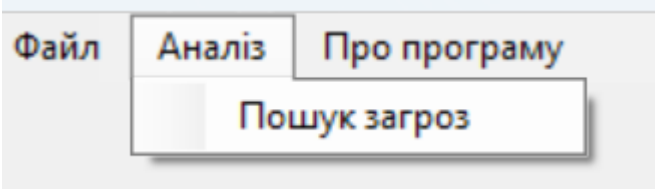


Рис. 4.5 Форма пошуку вразливостей

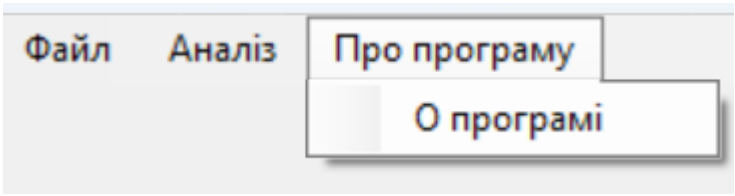


Рис. 4.6 Форма пошуку вразливостей

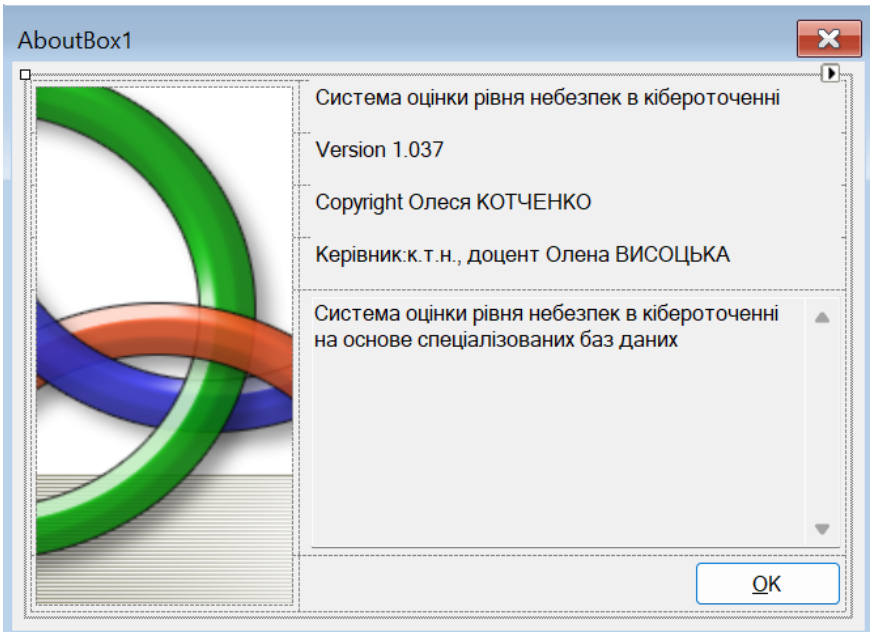


Рис. 4.7 Форма About

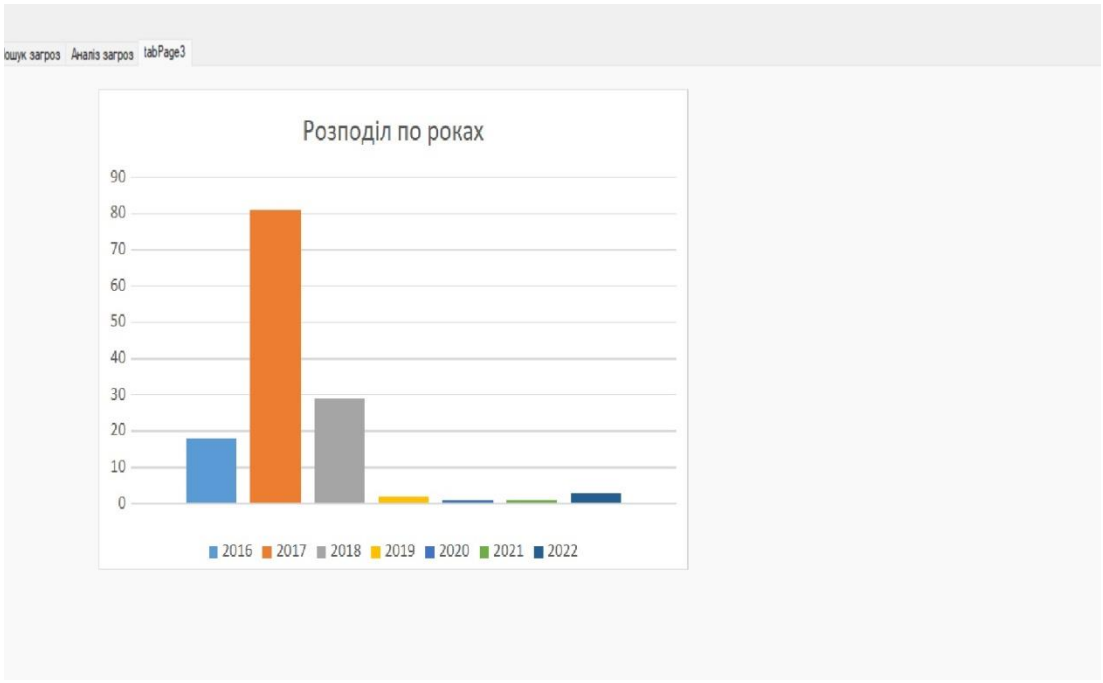


Рис. 4.8 Форма аналізу по роках

Форма About містить інформацію про автора та поточні характеристики програми. Версія , призначення, дескриптор, керівник

Перерахуємо вразливості в фокусній базі:

2016 18

2017 81

2018 29

2019 2

2020 1

2021 1

2022 3

2023 0

2024 0

Дані відповідають отриманому розподілу.

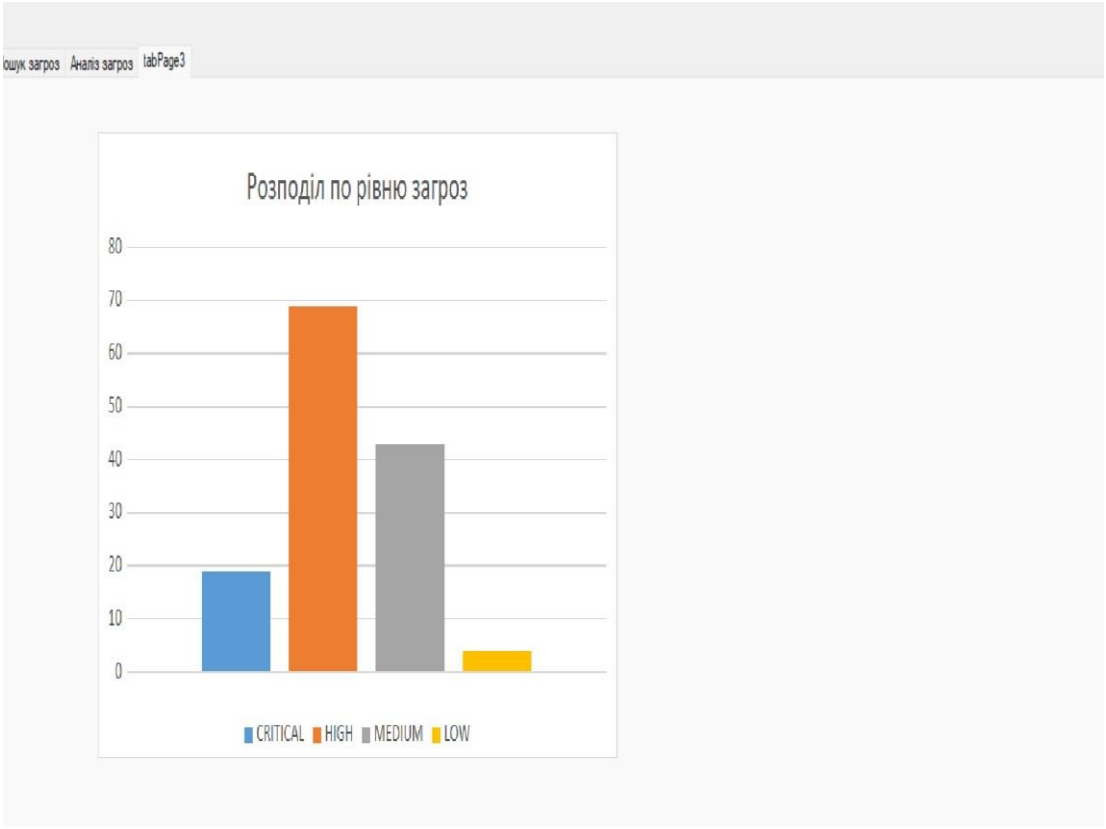


Рис. 4.9 Форма аналізу по рівню загрози

Перевіряємо :

CRITICAL 19

HIGH 69

MEDIUM 43

LOW 4

Дані відповідають отриманому розподілу.

Доступність коду дозволяє швидко додавати будь якій потрібний вид аналізу. Наприклад поєднаємо ці аналізи.

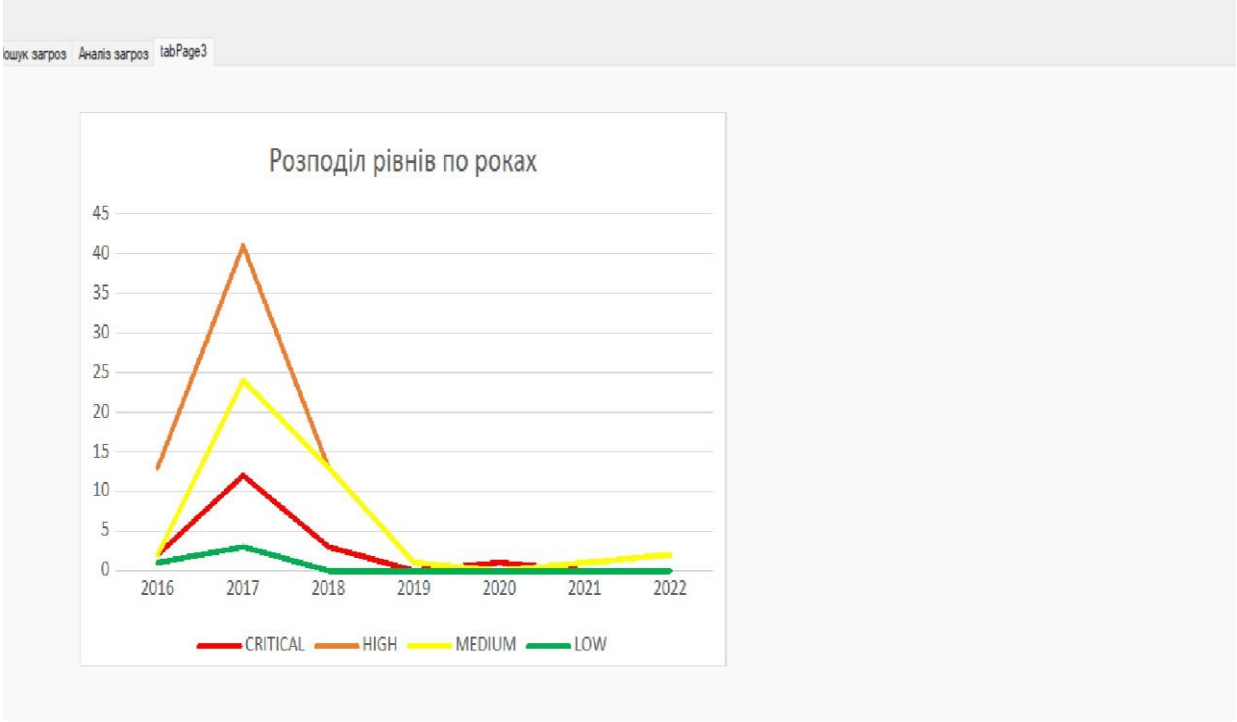


Рис. 4.10 Форма аналізу розподіл рівнів по роках

Наведена вище інформація доводить доцільність використання розробленої системи для вирішення поставленої задачі.

На основі аналізу із розділу 1, таблиці 1.4 порівняємо засоби оцінювання, що вже існують та розробленого додатку

На основі таблиці 4.1 можна зробити висновок про доцільність та ефективність використання системи.

Таблиця 4.1

Порівняльна характеристика засобів оцінювання, що вже існують та розробленого додатку

Характеристика	OWASP Risk Assessment Calculator	NIST Cybersecurity Framework	FAIR	SIEM Tools	Система оцінки вразливостей на основі спеціалізованих баз даних
Доступність	Висока (відкрите джерело)	Висока (відкритий доступ)	Середня (вимагає підготовки)	Висока (широко розповсюджені)	Висока (відкритий доступ)
Точність Оцінки	Висока для веб-застосунків	Загальна, адаптивна	Висока, кількісна	Залежить від інструменту	Залежить від виду аналізу
Масштабованість	Обмежена	Висока	Висока	Висока	Висока
Автономність	Ручне використання	Частково автоматизована	Потребує експертного аналізу	Висока (автоматизація)	Частково автоматизована
Гнучкість	Обмежена	Висока	Висока	Висока	Висока
Ефективність	Залежить від користувача	Залежить від реалізації	Висока в кількісному аналізі	Залежить від використання	Залежить від користувача

Висновок до розділу 4

В результаті виконання розділу 4 отримані наступні результати:

В четвертому розділі здійснено розробку та тестування системи оцінки вразливостей на основі спеціалізованих базах даних. Для цього було вдосконаленно метод оцінки рівня небезпек. А саме, на основі попередньо прийнятих рішень було прийнято двоетапний підхід до аналізу та побудовано на основі базового алгоритму виділення класу вразливостей, побудовано шести кроковий метод оцінки рівня небезпек.

Програмна реалізація методу оцінки рівня небезпек включає загальний обсяг коду 26920 рядків, в додатку А наведено обрані фрагменти коду приблизно 1300 рядків це приблизно 5 процентів коду.

Проведено тестування розробленої системи оцінки рівня небезпек, що дало змогу дослідити доцільність використання розробленої системи для вирішення поставленої задачі

ВИСНОВКИ

В ході виконання кваліфікаційної роботи було: розглянуто методи оцінки рівня небезпек критичних даних корпоративних застосунків та на основі результату проведеного аналізу було обрано методи для вирішення задачі, обрано середовище розробки для формування запитів до баз даних; зроблено аналіз особливостей методів оцінювання баз даних вразливостей Common Weakness Enumeration, National Vulnerability Database, GitHub Advisory, FIRST.

На основі цього аналізу було визначено загальні вимоги до системи, які будуть враховані при подальшій розробці; протестовано обмін з базою даних вразливостей та визначено формати обміну та збереження інформації; запропоновано базовий алгоритм виділення класу вразливостей та розроблені підпрограми для його реалізації.

В результаті виконання кваліфікаційної роботи отримано наступні результати:

1. Проаналізовано існуючі методи оцінки рівня небезпек для критичних даних корпоративних застосунків та на основі результатів проведеного аналізу обрано методи оцінки рівня небезпек для критичних інформаційних ресурсів.

2. Розроблено систему оцінки рівня небезпек на основі вдосконаленого методу оцінки рівня небезпек за рахунок використання попереднього відбору небезпек для аналізу, що дозволило покращити метрику часу для цільового аналізу пропорційному обмеженню кількості вразливостей.

3. Проведено тестування розробленого програмного застосунку оцінки рівня небезпек, що дало змогу дослідити доцільність використання розробленої системи для вирішення поставленої задачі.

Робота включає в собі чотири розділи. В першому розділі наведено оцінку рівня небезпек критичних даних корпоративних застосунків. А саме, наведено таксономію методів захисту критичних даних. Описано інформаційні ризики та можливу шкоду. Проаналізовано метод експертної оцінки. Показано, що у практиці, ефективне використання експертної оцінки вимагає уважності до

вибору експертів, чітко визначених завдань для оцінки та грамотного аналізу отриманих результатів. Крім того, може бути корисним використовувати експертну оцінку, як один із елементів комплексної стратегії оцінки ризиків. А саме, якісний аналіз ризику має складатися мінімум з 4х кроків:

Ідентифікація ризику - найважливіша частина якісного аналізу ризику. Якщо не вдасться визначити ризики завчасно, керувати ними стане надзвичайно складно. Спосіб для ідентифікації ризику полягає в тому, щоб він був простим а саме, все, що може мати непевний вплив на підприємство. Виявлення очевидних ризиків допоможе глибше зануритися в більш хибні. Ідентифікація ризику пов'язана з кількістю, тому потрібно задіяти якомога більшу кількість людей, щоб отримати широкий діапазон поглядів;

Аналіз впливу - розділення ризиків на загрози та можливості. Використовуючи якісний аналіз ризику, оцінюють вплив кожного ризику за шкалою (1-5 або низький/середній/високий/екстремальний). Далі оцінюють ймовірність виникнення кожного ризику, використовуючи аналогічну шкалу. Отримані бали об'єднують, щоб створити загальний рейтинг ризику;

Усунення ризику - процес вчинення заходів щодо ризику;

Огляд і моніторинг — процес повторення останніх кроків для повторної оцінки ризиків та виявлення помилок минулого оцінювання.

При аналізі менеджменту ризиків розглянуто методи кількісної та якісної оцінки ризиків. А також підхід, який поєднує їх.

В ході аналізу засобів оцінки шкоди показана матриця ризиків, яка часто використовується під час оцінки ризику для вимірювання рівня ризику, враховуючи наслідки/важкість та ймовірність реалізації ризику.

Наостаннє, в першому розділі розглянуто структуру, що забезпечує структурований підхід для організацій, щоб підвищити рівень кібербезпеки та підвищити стійкість до кіберзагроз. Це NIST Cybersecurity Framework — комплексний набір інструкцій і практик, розроблених, щоб допомогти організаціям керувати ризиками кібербезпеки та зменшувати їх. Він був

розроблений Національним інститутом стандартів і технологій (NIST), федеральним агентством Міністерства торгівлі США.

Також, в якості майбутньої бази для порівняння обрано калькулятор, розроблений Open Web Application Security Project (OWASP), який забезпечує структурований підхід до оцінки та визначення пріоритетів ризиків безпеки та застосунок для факторного аналізу інформаційних ризиків (FAIR) — який призначено для кількісної оцінки ризиків, яка в свою чергу дозволяє організаціям розуміти, аналізувати та кількісно оцінювати ризики інформаційної безпеки у фінансовому плані. Він забезпечує структурований і систематичний підхід до оцінки та встановлення пріоритетів ризиків, пов'язаних з інформаційними активами.

В другому розділі проведено аналіз особливостей методів оцінювання баз даних вразливостей Common Weakness Enumeration, National Vulnerability Database, GitHub Advisory, FIRST. Головним результатом цього аналізу було запропоновано надалі використовувати двоетапну схему аналізу. А саме, на основі аналізу було зроблено висновок CWE – зручний для вчасного інформування власників автоматизованих систем, але він має нерегулярну структуру і потребує побудову інформаційних систем для її швидкої інтерпретації та обробки. Тому для його вдосконалення пропонується двоетапна схема аналізу:

На першому кроці визначається фокус уваги аналізу, з бази вразливостей відокремлюється підмножина загроз, які є фокусними для даного аналізу, далі будемо називати цю підмножину класом вразливостей;

На другому кроці проводиться більш детальний аналіз класу загроз. За рахунок однорідності підмножини вразливостей підвішується ефективність аналізу.

Важлівость National Vulnerability Database Національної бази даних вразливостей для нашей роботи обумовлена рядом факторів. NVD – це репозиторій уряду США даних управління вразливостями на основі стандартів, представлених за допомогою протоколу автоматизації контенту безпеки (SCAP).

Ці дані дозволяють автоматизувати управління вразливостями, вимірювання безпеки та відповідність. NVD включає бази даних із посиланнями на контрольні списки безпеки, недоліки програмного забезпечення, пов'язані з безпекою, назви продуктів і показники впливу. Всі вразливості в NVD мають ідентифікатор CVE.

Консультативна база даних GitHub містить критичний для нашої розробки список відомих вразливостей безпеки та шкідливого програмного забезпечення, згрупованих у три категорії: рекомендації, перевірені GitHub, неперевірені рекомендації та рекомендації щодо шкідливого програмного забезпечення.

Остання частка анализу стосується FIRST, прагне об'єднати команди реагування на інциденти та безпеки з кожної країни світу, щоб забезпечити безпечний Інтернет для всіх.

Ефективне реагування є глобальним завданням, що відображає глобальний характер Інтернету. Ґрунтуючись на моделі однорангового управління мережею, групи реагування на інциденти комп'ютерної безпеки (CSIRTs), групи реагування на інциденти безпеки продуктів (PSIRTs) і незалежні дослідники безпеки працюють разом, щоб обмежити шкоду від інцидентів безпеки. Для цього потрібен високий рівень довіри. Інциденти не обмежуються одним культурним чи політичним куточком Інтернету, і вони не поважають кордони. Таким чином, FIRST сприяє інклюзивності нашого аналізу.

В третьому розділі здійснено розробку алгоритмічного забезпечення пошуку окремих класів вразливостей в спеціалізованих базах даних. Яка виконана в чотири кроки:

Обрано середовище реалізації, а саме прийнято рішення про двоетапну реалізацію перша PyCharm 2024.2.1 та друга для інтерактівної частини VisualStudio.17.Release/17.11.3+35303.130. Це дозволило використати переваги обох підходів.

Для ідентифікації одиничної вразливості було написано тестовий фрагмент, який дозволив отримати опис вразливості з бази даних. Він включає дві підпрограми, які будуть використовуватися далі, це Get_cve_record та Print_cve_details. Get_cve_record – отримує інформацію безпосередньо з сервера

Митре адреса https://cveawg.mitre.org/api/cve/{cve_id} та перевіряє безпомилковість цієї операції. Print_cve_details — дешифрує інформацію та виводить її на екран.

База даних загроз динамічно змінюється тому є сенс фіксувати множину вразливостей окремого класу формуючи для цього підмножини вразливостей на момент запиту, за рахунок цього спрощується подальша обробка та знижується складність подальшого аналізу. Для цього було запропоновано базовий алгоритм виділення класу вразливостей, циклічне виконання якого є базою системи аналізу.

В алгоритмі в першу чергу визначаємо, який клас вразливостей будемо аналізувати. Наступним кроком сформуємо відповідний запит. На цьому кроці є технологічні особливості, які ми розглянемо далі при аналізі коду. Далі виконуємо GET – запит та зберігаємо інформацію, що отримано в результаті його виконання. Версії формату JSON залежать від версії CVSS, це більш детально було описано в параграфі 2.2. стор. 50. Далі виконаємо запит та збережемо дані в формати JSON в активному каталозі. У фрагменті показані також деякі технологічні аспекти, а саме обробка помилок та деякі нюанси форматування даних для відображення.

Процес вибору моделі оцінювання вразливостей тісно пов'язаний із специфікою та характеристиками конкретного підприємства. Аспекти, що впливають на цей вибір, включають рівень залежності підприємства від інформаційних технологій та їх значення для загальної діяльності. Важливо визначити, наскільки критичні IT-системи та інформаційні процеси для бізнесу; Аналіз залежностей між власниками процесів, процесами та активами. Це допомагає зрозуміти, як взаємодія між різними елементами впливає на ризики та їхнє управління; наявність необхідних ресурсів - людських, фінансових, часових - для ефективної реалізації процесів оцінки та управління ризиками. Забезпечення достатньої підтримки та ресурсів є ключовим для успішної оцінки ризиків; вимоги законодавства, регуляторів та інших заінтересованих сторін до

процесу керування ризиками інформаційної безпеки. Необхідно забезпечити відповідність діяльності підприємства чинним нормативам і стандартам.

Виходячи з перелічених умов, для кожного підприємства може бути оптимальним вибір різних методологій оцінки вразливостей.

В четвертому розділі здійснено розробку та тестування системи оцінки вразливостей на основі спеціалізованих базах даних. Для цього було вдосконаленно метод оцінки рівня небезпек. А саме, на основі попередньо прийнятих рішень було прийнято двоетапний підхід до аналізу та побудовано, на основі базового алгоритму виділення класу вразливостей, шести кроковий метод оцінки рівня небезпек .

Програмна реалізація методу оцінки рівня небезпек включає загальний обсяг коду 26920 рядків, в додатку А наведено обрані фрагменти коду приблизно 1300 рядків, це приблизно 5 процентів коду.

Проведено тестування розробленої системи оцінки рівня небезпек, що дало змогу дослідити доцільність використання розробленої системи для вирішення поставленої задачі.

Наукова новизна: вдосконалено метод оцінки рівня небезпек на основі використання попереднього відбору небезпек для аналізу, що дозволило покращити метрику часу для цільового аналізу пропорційному обмеженню кількості вразливостей.

Практичне значення отриманих результатів. Розроблено систему для оцінки рівня небезпек критичних даних корпоративних застосунків, яка за рахунок фіксації фокусної бази вразливостей, окрім основної функції також надає доказову базу на випадок інциденту кібербезпеки. Рівень розробки системи дозволяє її застосування фахівцями нетехнічних спеціальностей.

Особистий внесок здобувача вищої освіти. Всі результати отримані в ході виконання кваліфікаційної роботи отримані автором самостійно.

Апробація отриманих результатів. Основні положення роботи доповідалися та обговорювалися на міжнародної науково-практичної конференції: Innovations and New Directions in Scientific Research: Proceedings of

the International Scientific Conference (2024, October 14). Manchester, UK: Bookmundo.

Публікації. Одні матеріали конференції :

Olesya Kotchenko, Olena Vysotska. Analysis of advantages and disadvantages of harm assessment tools.Innovations and New Directions in Scientific Research: Proceedings of the International Scientific Conference (2024, October 14). Manchester, UK: Bookmundo...p.p. 113-114.

СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Bell L. LaPadula. Secure Computer System: Mathematical Foundation, ESDTR-73-278, V 1, MITRE Corportation.

2. LaPadula D. Bell. Secure Computer Systems: A Mathematical Model, ESD TR-73-278, V. II, MITRE Corportaion.

3. Nyanchama M. Modeling mandatory access control in role-based security systems / M. Nyanchama, S. Osborn // InDBSec. — 1995. — P. 129–144.

4. Anisimov A. Variable-length prefix codes with multiple delimiters / A. Anisimov, I. Zavadskyi // IEEE Transactions on Information Theory. — 2017. — Vol. 63, № 5. — P. 2885–2895.

5. Cotrini C. Analyzing first-order role based access control / C. Cotrini, T. Weghorn, D. Basin, M. Clavel. — 2015.

6. J.Rushby, Formal methods and their role in the certification of critical systems, 1995.

7. Daniel Servos, Sylvia L. Osborn. Current Research and Open Problems in Attribute-Based Access Control. — University of Western Ontario.

8. Vincent C. Hu, etc. Guide to Attribute Based Access Control (ABAC) Definition and Considerations. — National Institute of Standards and Technology, 2014. — P. 45.

9. William Fisher. Attribute Based Access Control. — National Institute of Standards and Technology, 2015. — P. 22.

10. Alan H. Karp, etc. From ABAC to ZBAC: The Evolution of Access Control Models. — HP Laboratories, 2009. — P. 21.

11. Bill Parducci, Hal Lockhart. eXtensible Access Control Markup Language (XACML) Version 3.0 Plus Errata 01. — OASIS, 2017. — P. 154.

12. Xin Jin. ATTRIBUTE-BASED ACCESS CONTROL MODELS AND IMPLEMENTATION IN CLOUD INFRASTRUCTURE AS A SERVICE. — THE UNIVERSITY OF TEXAS AT SAN ANTONIO, 2014. — P. 144. 92

13. Yang K. Dac-macs: effective data access control for multi-authority cloud storage systems / K. Yang, X. Jia, K. Ren, B. Zhang // 2013 Proceedings IEEE INFOCOM. — 2013. — P. 2895–2903.

14. Line M. B. Examining the suitability of industrial safety management approaches for information security incident management / M. B. Line, E. Albrechtsen // Information and Computer Security. — 2016. — Vol. 24, № 1. — P. 20–37.

15. Al-Kahtani M. A. Rule-based rbac with negative authorization / M. A. AlKahtani, R. Sandhu. — 2004.

16. Marchenko O. Machine learning method for paraphrase identification / O. Marchenko, A. Anisimov, A. Nykonenko. — Springer, 2017.

17. Терейковський І. А. Формування політики безпеки комп'ютерних систем/ І. А. Терейковський // Захист інформації. — 2008. — Т. 10, № 1. — С. 12–22.

18. Андрєєв В. І. Основи інформаційної безпеки / В. І. Андрєєв, В. О. Хорошко, В. С. Чередніченко, М. Є. Шелест. — К. : ДУІКТ, 2009. — 292 с.

19. Давиденко А. М. Використання формальних засобів опису процесів надання повноважень / А. М. Давиденко, О. А. Суліма // Захист інформації. — 2016. — Т. 18. — С. 143–149.

20. Горбулін В. П. Інформаційні операції та безпека суспільства: загрози, протидія, моделювання / В. П. Горбулін, О. Г. Додонов, Д. В. Ланде. — К. : Інтертехнологія, 2009. — 164 с.

21. Bertino E. Geo-rbac: a spatially aware rbac / E. Bertino, B. Catania // Proceedings of the tenth ACM symposium on Access control models and technologies. — 2005. — P. 29–37.

22. Грайворонський М. В. Безпека інформаційно-комунікаційних систем / М. В. Грайворонський, О. М. Новіков. — К. : Видавнича група BHV, 2009. — 608 с.

23. Phillips C. E. Security assurance for an rbac/mac security model / C. E. Phillips, S. A. Demurjian, T. C. Ting. — 2003.

24. Архипов О. Є. Інформаційні ризики: методи та способи дослідження, моделі ризиків і методи їх ідентифікації / О. Є. Архипов, А. В. Скиба // Захист інформації. — 2013. — Т. 15, № 4. — С. 366–375.

25. Корченко О. Г. Системи захисту інформації / О.Г. Корченко. – К.: НАУ, 2004. – 264 с.

26. Критерії оцінки захищеності інформації в комп'ютерних системах від несанкціонованого доступу : НД ТЗИ 2.5-004-99. – [Введ. 01.07.99].– К.:ДСТСЗИ СБ України, 1999. – 38 с.

27. Ferraiolo D. F., Kuhn D. R. (October 1992). "Role Based Access Control". 15th National Computer Security Conference: 554—563.

28. Sandhu R., Coyne E. J., Feinstein H. L., Youman C. E. (August 1996). «RoleBased Access Control Models». IEEE Computer (IEEE Press) 29 (2): 38–47.

29. Editor, CSRC Content. Role Based Access Control - FAQs. csrc.nist.gov (EN-US). Процитовано 2024-10-16.

30. CWE - New to CWE (mitre.org) https://cwe.mitre.org/about/new_to_cwe.html

31. NVD - Home (nist.gov) https://nvd.nist.gov/VD – Home

32. GitHub Advisory Database · GitHub https://github.com/advisories

33. IR Database (first.org) https://www.first.org/global/irt-databaseabase (first.org)

ДОДАТКИ

Додаток А

Фрагмент коду інтерфейсу системи оцінки рівня небезпек

```
\\* Розроблено в межах магістерської роботи:  Система оцінки рівня
\\* небезпек в кібероточенні на основі спеціалізованих баз даних
\\*Керівник: к.т.н., доцент Олена ВИСОЦЬКА
\\*Виконавець: Олеся КОТЧЕНКО
\\* Версія 1.037 від 11.11.2024
\\* Конфігурація програмного забезпечення
\\*Microsoft Visual Studio Community 2022
\\*Version 17.11.3
\\*VisualStudio.17.Release/17.11.3+35303.130
\\*Microsoft .NET Framework
\\*Version 4.8.09032
\\*Installed Version: Community
\\*Visual C++ 2022   00482-90000-00000-AA273
\\*Microsoft Visual C++ 2022
\\*ASP.NET and Web Tools   17.11.231.19466
\\*ASP.NET and Web Tools
\\*Azure App Service Tools v3.0.0   17.11.231.19466
\\*Azure App Service Tools v3.0.0
\\*Azure Functions and Web Jobs Tools   17.11.231.19466
\\*Azure Functions and Web Jobs Tools
\\*C# Tools   4.11.0-3.24428.4+1ea9c390a5bb6815fdff2137ee155e23e78d6ff3
\\*Common Azure Tools   1.10
\\*GitHub Copilot   0.2.1648.49400
\\*Microsoft JVM Debugger   1.0
\\*NuGet Package Manager   6.11.0
\\*Razor (ASP.NET Core)
\\*17.11.3.2442001+68650a7d94261bc56a1f4bc522c2ee35314b1abb
\\*SQL Server Data Tools   17.11.38.0
\\*Microsoft SQL Server Data Tools
\\*Test Adapter for Boost.Test   1.0
\\*Test Adapter for Google Test   1.0
\\\*TypeScript Tools   17.0.30715.2002
\\*TypeScript Tools for Microsoft Visual Studio
\\*Visual Basic Tools   4.11.0-
\\*3.24428.4+1ea9c390a5bb6815fdff2137ee155e23e78d6ff3
\\*Visual F# Tools   17.11.0-
\\*beta.24421.7+af2f522de602281d4ef5a7b71507c428e814c5c1
\\*Microsoft Visual F# Tools
\\*Visual Studio IntelliCode   2.2
```

```
\\*
{
   "resultsPerPage": 135,
   "startIndex": 0,
   "totalResults": 135,
   "format": "NVD_CVE",
   "version": "2.0",
   "timestamp": "2024-09-27T11:22:45.387",
   "vulnerabilities": [
      {
         "cve": {
            "id": "CVE-2015-6184",
            "sourceIdentifier": "secure@microsoft.com",
            "published": "2016-03-09T23:59:00.163",
            "lastModified": "2018-10-12T22:10:41.470",
            "vulnStatus": "Modified",
            "cveTags": [],
            "descriptions": [
               {
                  "lang": "en",
                  "value": "The CAttrArray object implementation in Microsoft Internet Explorer 7 through 11 allows remote attackers to execute arbitrary code or cause a denial of service (type confusion and memory corruption) via a malformed Cascading Style Sheets (CSS) token sequence in conjunction with modifications to HTML elements, aka \"Internet Explorer Memory Corruption Vulnerability,\" a different vulnerability than CVE-2015-6048 and CVE-2015-6049."
               },
               {
                  "lang": "es",
                  "value": "La implementaci\u00f3n de objeto CAttrArray en Microsoft Internet Explorer 7 hasta la versi\u00f3n 11 permite a atacantes remotos ejecutar c\u00f3digo arbitrario o provocar una denegaci\u00f3n de servicio (confusi\u00f3n de tipo y corrupci\u00f3n de memoria) a trav\u00e9s de una secuencia de tokens Cascading Style Sheets (CSS) mal formada en conjunci\u00f3n con modificaciones a elementos HTML, tambi\u00e9n conocida como \"Internet Explorer Memory Corruption Vulnerability\", una vulnerabilidad diferente a CVE-2015-6048 y CVE-2015-6049."
               }
            ],
            "metrics": {
               "cvssMetricV30": [
                  {
                     "source": "nvd@nist.gov",
                     "type": "Primary",
                     "cvssData": {
```

```
                        "version": "3.0",
                        "vectorString":
"CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:H",
                        "attackVector": "NETWORK",
                        "attackComplexity": "HIGH",
                        "privilegesRequired": "NONE",
                        "userInteraction": "NONE",
                        "scope": "UNCHANGED",
                        "confidentialityImpact": "HIGH",
                        "integrityImpact": "HIGH",
                        "availabilityImpact": "HIGH",
                        "baseScore": 8.1,
                        "baseSeverity": "HIGH"
                     },
                     "exploitabilityScore": 2.2,
                     "impactScore": 5.9
                  }
               ],
               "cvssMetricV2": [
                  {
                     "source": "nvd@nist.gov",
                     "type": "Primary",
                     "cvssData": {
                        "version": "2.0",
                        "vectorString": "AV:N/AC:M/Au:N/C:C/I:C/A:C",
                        "accessVector": "NETWORK",
                        "accessComplexity": "MEDIUM",
                        "authentication": "NONE",
                        "confidentialityImpact": "COMPLETE",
                        "integrityImpact": "COMPLETE",
                        "availabilityImpact": "COMPLETE",
                        "baseScore": 9.3
                     },
                     "baseSeverity": "HIGH",
                     "exploitabilityScore": 8.6,
                     "impactScore": 10.0,
                     "acInsufInfo": false,
                     "obtainAllPrivilege": false,
                     "obtainUserPrivilege": false,
                     "obtainOtherPrivilege": false
                  }
               ]
            },
            "weaknesses": [
               {
```

```
                "source": "nvd@nist.gov",
                "type": "Primary",
                "description": [
                  {
                    "lang": "en",
                    "value": "NVD-CWE-Other"
                  }
                ]
              }
            ],
            "configurations": [
              {
                "nodes": [
                  {
                    "operator": "OR",
                    "negate": false,
                    "cpeMatch": [
                      {
                        "vulnerable": true,
                        "criteria":
"cpe:2.3:a:microsoft:internet_explorer:7:*:*:*:*:*:*:*",
                        "matchCriteriaId":    "1A33FA7F-BB2A-4C66-B608-
72997A2BD1DB"
                      },
                      {
                        "vulnerable": true,
                        "criteria":
"cpe:2.3:a:microsoft:internet_explorer:8:*:*:*:*:*:*:*",
                        "matchCriteriaId":     "A52E757F-9B41-43B4-9D67-
3FEDACA71283"
                      },
                      {
                        "vulnerable": true,
                        "criteria":
"cpe:2.3:a:microsoft:internet_explorer:9:*:*:*:*:*:*:*",
                        "matchCriteriaId":   "C043EDDD-41BF-4718-BDCF-
158BBBDB6360"
                      },
                      {
                        "vulnerable": true,
                        "criteria":
"cpe:2.3:a:microsoft:internet_explorer:10:*:*:*:*:*:*:*",
                        "matchCriteriaId":     "D5808661-A082-4CBE-808C-
B253972487B4"
                      },
```

```
                        {
                           "vulnerable": true,
                           "criteria":
"cpe:2.3:a:microsoft:internet_explorer:11:*:*:*:*:*:*:*",
                           "matchCriteriaId":  "15BAAA8C-7AF1-46CE-9FFB-
3A498508A1BF"
                        }
                     ]
                  }
               ]
            },
            {
               "operator": "AND",
               "nodes": [
                  {
                     "operator": "OR",
                     "negate": false,
                     "cpeMatch": [
                        {
                           "vulnerable": true,
                           "criteria":
"cpe:2.3:a:microsoft:internet_explorer:11:*:*:*:*:*:*:*",
                           "matchCriteriaId":  "15BAAA8C-7AF1-46CE-9FFB-
3A498508A1BF"
                        }
                     ]
                  },
                  {
                     "operator": "OR",
                     "negate": false,
                     "cpeMatch": [
                        {
                           "vulnerable": false,
                           "criteria":
"cpe:2.3:o:microsoft:windows_10:*:*:*:*:*:*:*:*",
                           "matchCriteriaId":  "FBC814B4-7DEC-4EFC-ABFF-
08FFD9FD16AA"
                        }
                     ]
                  }
               ]
            }
         ],
         "references": [
            {
```

```
                    "url": "https://docs.microsoft.com/en-us/security-updates/securitybulletins/2015/ms15-106",
                    "source": "secure@microsoft.com"
                  },
                  {
                    "url": "https://www.verisign.com/en_US/security-services/security-intelligence/vulnerability-reports/articles/index.xhtml?id=1218",
                    "source": "secure@microsoft.com"
                  }
                ]
              }
            },
            {
              "cve": {
                "id": "CVE-2016-0088",
                "sourceIdentifier": "secure@microsoft.com",
                "published": "2016-04-12T23:59:00.147",
                "lastModified": "2018-10-12T22:11:00.707",
                "vulnStatus": "Modified",
                "cveTags": [],
                "descriptions": [
                  {
                    "lang": "en",
                    "value": "Hyper-V in Microsoft Windows 8.1, Windows Server 2012 Gold and R2, and Windows 10 allows guest OS users to execute arbitrary code on the host OS via a crafted application, aka \"Hyper-V Remote Code Execution Vulnerability.\""
                  },
                  {
                    "lang": "es",
                    "value": "Hyper-V en Microsoft Windows 8.1, Windows Server 2012 Gold y R2 y Windows 10 permite a usuarios del SO invitado ejecutar c\u00f3digo arbitrario en el SO anfitri\u00f3n a trav\u00e9s de una aplicaci\u00f3n manipulada, tambi\u00e9n conocida como \"Hyper-V Remote Code Execution Vulnerability\"."
                  }
                ],
                "metrics": {
                  "cvssMetricV30": [
                    {
                      "source": "nvd@nist.gov",
                      "type": "Primary",
                      "cvssData": {
                        "version": "3.0",
                        "vectorString": "CVSS:3.0/AV:L/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H",
```

```
                "attackVector": "LOCAL",
                "attackComplexity": "LOW",
                "privilegesRequired": "NONE",
                "userInteraction": "NONE",
                "scope": "CHANGED",
                "confidentialityImpact": "HIGH",
                "integrityImpact": "HIGH",
                "availabilityImpact": "HIGH",
                "baseScore": 9.3,
                "baseSeverity": "CRITICAL"
            },
            "exploitabilityScore": 2.5,
            "impactScore": 6.0
        }
    ],
    "cvssMetricV2": [
        {
            "source": "nvd@nist.gov",
            "type": "Primary",
            "cvssData": {
                "version": "2.0",
                "vectorString": "AV:L/AC:L/Au:N/C:C/I:C/A:C",
                "accessVector": "LOCAL",
                "accessComplexity": "LOW",
                "authentication": "NONE",
                "confidentialityImpact": "COMPLETE",
                "integrityImpact": "COMPLETE",
                "availabilityImpact": "COMPLETE",
                "baseScore": 7.2
            },
            "baseSeverity": "HIGH",
            "exploitabilityScore": 3.9,
            "impactScore": 10.0,
            "acInsufInfo": false,
            "obtainAllPrivilege": false,
            "obtainUserPrivilege": false,
            "obtainOtherPrivilege": false
        }
    ]
},
"weaknesses": [
    {
        "source": "nvd@nist.gov",
        "type": "Primary",
        "description": [
```

```
                {
                  "lang": "en",
                  "value": "CWE-284"
                }
              ]
            }
          ],
          "configurations": [
            {
              "nodes": [
                {
                  "operator": "OR",
                  "negate": false,
                  "cpeMatch": [
                    {
                      "vulnerable": true,
                      "criteria":
"cpe:2.3:o:microsoft:windows_10:*:*:*:*:*:*:*:*",
                      "matchCriteriaId":  "FBC814B4-7DEC-4EFC-ABFF-
08FFD9FD16AA"
                    },
                    {
                      "vulnerable": true,
                      "criteria":
"cpe:2.3:o:microsoft:windows_8.1:*:*:*:*:*:*:*:*",
                      "matchCriteriaId":  "A7F51B5F-AA19-4D31-89FA-
6DFAC4BA8F0F"
                    },
                    {
                      "vulnerable": true,
                      "criteria":
"cpe:2.3:o:microsoft:windows_server_2012:-:*:*:*:*:*:*:*",
                      "matchCriteriaId":  "A7DF96F8-BA6A-4780-9CA3-
F719B3F81074"
                    },
                    {
                      "vulnerable": true,
                      "criteria":
"cpe:2.3:o:microsoft:windows_server_2012:r2:*:*:*:*:*:*:*",
                      "matchCriteriaId":  "DB18C4CE-5917-401E-ACF7-
2747084FD36E"
                    }
                  ]
                }
              ]
```

```
            }
          ],
          "references": [
            {
              "url": "http://www.securitytracker.com/id/1035538",
              "source": "secure@microsoft.com"
            },
            {
              "url": "https://docs.microsoft.com/en-us/security-updates/securitybulletins/2016/ms16-045",
              "source": "secure@microsoft.com"
            }
          ]
        }
      },
      {
        "cve": {
          "id": "CVE-2016-0089",
          "sourceIdentifier": "secure@microsoft.com",
          "published": "2016-04-12T23:59:01.583",
          "lastModified": "2018-10-12T22:11:00.910",
          "vulnStatus": "Modified",
          "cveTags": [],
          "descriptions": [
            {
              "lang": "en",
              "value": "Hyper-V in Microsoft Windows 8.1, Windows Server 2012 Gold and R2, and Windows 10 allows guest OS users to obtain sensitive information from host OS memory via a crafted application, aka \"Hyper-V Information Disclosure Vulnerability.\""
            },
            {
              "lang": "es",
              "value": "Hyper-V en Microsoft Windows 8.1, Windows Server 2012 Gold y R2 y Windows 10 permite a usuarios del SO invitado obtener informaci\u00f3n sensible de la memoria del SO anfitri\u00f3n a trav\u00e9s de una aplicaci\u00f3n manipulada, tambi\u00e9n conocida como \"Hyper-V Information Disclosure Vulnerability\"."
            }
          ],
          "metrics": {
            "cvssMetricV30": [
              {
                "source": "nvd@nist.gov",
                "type": "Primary",
```

```
                        "cvssData": {
                          "version": "3.0",
                          "vectorString":
"CVSS:3.0/AV:L/AC:L/PR:N/UI:N/S:C/C:H/I:N/A:N",
                          "attackVector": "LOCAL",
                          "attackComplexity": "LOW",
                          "privilegesRequired": "NONE",
                          "userInteraction": "NONE",
                          "scope": "CHANGED",
                          "confidentialityImpact": "HIGH",
                          "integrityImpact": "NONE",
                          "availabilityImpact": "NONE",
                          "baseScore": 7.1,
                          "baseSeverity": "HIGH"
                        },
                        "exploitabilityScore": 2.5,
                        "impactScore": 4.0
                      }
                    ],
                    "cvssMetricV2": [
                      {
                        "source": "nvd@nist.gov",
                        "type": "Primary",
                        "cvssData": {
                          "version": "2.0",
                          "vectorString": "AV:L/AC:L/Au:N/C:P/I:N/A:N",
                          "accessVector": "LOCAL",
                          "accessComplexity": "LOW",
                          "authentication": "NONE",
                          "confidentialityImpact": "PARTIAL",
                          "integrityImpact": "NONE",
                          "availabilityImpact": "NONE",
                          "baseScore": 2.1
                        },
                        "baseSeverity": "LOW",
                        "exploitabilityScore": 3.9,
                        "impactScore": 2.9,
                        "acInsufInfo": false,
                        "obtainAllPrivilege": false,
                        "obtainUserPrivilege": false,
                        "obtainOtherPrivilege": false
                      }
                    ]
                  },
                  "weaknesses": [
```

```
                {
                  "source": "nvd@nist.gov",
                  "type": "Primary",
                  "description": [
                    {
                      "lang": "en",
                      "value": "CWE-200"
                    }
                  ]
                }
              ],
              "configurations": [
                {
                  "nodes": [
                    {
                      "operator": "OR",
                      "negate": false,
                      "cpeMatch": [
                        {
                          "vulnerable": true,
                          "criteria":
"cpe:2.3:o:microsoft:windows_10:*:*:*:*:*:*:*:*",
                          "matchCriteriaId":  "FBC814B4-7DEC-4EFC-ABFF-
08FFD9FD16AA"
                        },
                        {
                          "vulnerable": true,
                          "criteria":
"cpe:2.3:o:microsoft:windows_8.1:*:*:*:*:*:*:*:*",
                          "matchCriteriaId":  "A7F51B5F-AA19-4D31-89FA-
6DFAC4BA8F0F"
                        },
                        {
                          "vulnerable": true,
                          "criteria":
"cpe:2.3:o:microsoft:windows_server_2012:-:*:*:*:*:*:*:*",
                          "matchCriteriaId":  "A7DF96F8-BA6A-4780-9CA3-
F719B3F81074"
                        },
                        {
                          "vulnerable": true,
                          "criteria":
"cpe:2.3:o:microsoft:windows_server_2012:r2:*:*:*:*:*:*:*",
                          "matchCriteriaId":  "DB18C4CE-5917-401E-ACF7-
2747084FD36E"
```

```
                        }
                     ]
                  }
               ]
            }
         ],
         "references": [
            {
               "url": "http://www.securitytracker.com/id/1035538",
               "source": "secure@microsoft.com"
            },
            {
               "url": "https://docs.microsoft.com/en-us/security-updates/securitybulletins/2016/ms16-045",
               "source": "secure@microsoft.com"
            }
         ]
      }
   },
   {
      "cve": {
         "id": "CVE-2016-0090",
         "sourceIdentifier": "secure@microsoft.com",
         "published": "2016-04-12T23:59:02.977",
         "lastModified": "2018-10-12T22:11:01.160",
         "vulnStatus": "Modified",
         "cveTags": [],
         "descriptions": [
            {
               "lang": "en",
               "value": "Hyper-V in Microsoft Windows 8.1, Windows Server 2012 R2, and Windows 10 allows guest OS users to obtain sensitive information from host OS memory via a crafted application, aka \"Hyper-V Information Disclosure Vulnerability.\""
            },
            {
               "lang": "es",
               "value": "Hyper-V en Microsoft Windows 8.1, Windows Server 2012 R2 y Windows 10 permite a usuarios del SO invitado obtener informaci\u00f3n sensible de la memoria del SO anfitri\u00f3n a trav\u00e9s de una aplicaci\u00f3n manipulada, tambi\u00e9n conocida como \"Hyper-V Information Disclosure Vulnerability\"."
            }
         ],
         "metrics": {
```

```
                "cvssMetricV30": [
                    {
                        "source": "nvd@nist.gov",
                        "type": "Primary",
                        "cvssData": {
                            "version": "3.0",
                            "vectorString":
"CVSS:3.0/AV:L/AC:L/PR:N/UI:N/S:C/C:H/I:N/A:N",
                            "attackVector": "LOCAL",
                            "attackComplexity": "LOW",
                            "privilegesRequired": "NONE",
                            "userInteraction": "NONE",
                            "scope": "CHANGED",
                            "confidentialityImpact": "HIGH",
                            "integrityImpact": "NONE",
                            "availabilityImpact": "NONE",
                            "baseScore": 7.1,
                            "baseSeverity": "HIGH"
                        },
                        "exploitabilityScore": 2.5,
                        "impactScore": 4.0
                    }
                ],
                "cvssMetricV2": [
                    {
                        "source": "nvd@nist.gov",
                        "type": "Primary",
                        "cvssData": {
                            "version": "2.0",
                            "vectorString": "AV:L/AC:L/Au:N/C:P/I:N/A:N",
                            "accessVector": "LOCAL",
                            "accessComplexity": "LOW",
                            "authentication": "NONE",
                            "confidentialityImpact": "PARTIAL",
                            "integrityImpact": "NONE",
                            "availabilityImpact": "NONE",
                            "baseScore": 2.1
                        },
                        "baseSeverity": "LOW",
                        "exploitabilityScore": 3.9,
                        "impactScore": 2.9,
                        "acInsufInfo": false,
                        "obtainAllPrivilege": false,
                        "obtainUserPrivilege": false,
                        "obtainOtherPrivilege": false
```

```
            }
          ]
        },
        "weaknesses": [
          {
            "source": "nvd@nist.gov",
            "type": "Primary",
            "description": [
              {
                "lang": "en",
                "value": "CWE-200"
              }
            ]
          }
        ],
        "configurations": [
          {
            "nodes": [
              {
                "operator": "OR",
                "negate": false,
                "cpeMatch": [
                  {
                    "vulnerable": true,
                    "criteria":
"cpe:2.3:o:microsoft:windows_10:*:*:*:*:*:*:*:*",
                    "matchCriteriaId":   "FBC814B4-7DEC-4EFC-ABFF-
08FFD9FD16AA"
                  },
                  {
                    "vulnerable": true,
                    "criteria":
"cpe:2.3:o:microsoft:windows_8.1:*:*:*:*:*:*:*:*",
                    "matchCriteriaId":    "A7F51B5F-AA19-4D31-89FA-
6DFAC4BA8F0F"
                  },
                  {
                    "vulnerable": true,
                    "criteria":
"cpe:2.3:o:microsoft:windows_server_2012:r2:*:*:*:*:*:*:*",
                    "matchCriteriaId":    "DB18C4CE-5917-401E-ACF7-
2747084FD36E"
                  }
                ]
              }
```

```
                ]
              }
            ],
            "references": [
              {
                "url": "http://www.securitytracker.com/id/1035538",
                "source": "secure@microsoft.com"
              },
              {
                "url": "https://docs.microsoft.com/en-us/security-updates/securitybulletins/2016/ms16-045",
                "source": "secure@microsoft.com"
              }
            ]
          }
        },
        {
          "cve": {
            "id": "CVE-2015-8823",
            "sourceIdentifier": "psirt@adobe.com",
            "published": "2016-04-22T18:59:00.110",
            "lastModified": "2023-05-15T18:57:00.297",
            "vulnStatus": "Analyzed",
            "cveTags": [],
            "evaluatorComment": "CWE-416: Use After Free",
            "descriptions": [
              {
                "lang": "en",
                "value": "Use-after-free vulnerability in the TextField object implementation in Adobe Flash Player before 18.0.0.268 and 19.x and 20.x before 20.0.0.228 on Windows and OS X and before 11.2.202.554 on Linux, Adobe AIR before 20.0.0.204, Adobe AIR SDK before 20.0.0.204, and Adobe AIR SDK & Compiler before 20.0.0.204 allows attackers to execute arbitrary code via crafted text property, a different vulnerability than CVE-2015-8048, CVE-2015-8049, CVE-2015-8050, CVE-2015-8055, CVE-2015-8056, CVE-2015-8057, CVE-2015-8058, CVE-2015-8059, CVE-2015-8061, CVE-2015-8062, CVE-2015-8063, CVE-2015-8064, CVE-2015-8065, CVE-2015-8066, CVE-2015-8067, CVE-2015-8068, CVE-2015-8069, CVE-2015-8070, CVE-2015-8071, CVE-2015-8401, CVE-2015-8402, CVE-2015-8403, CVE-2015-8404, CVE-2015-8405, CVE-2015-8406, CVE-2015-8410, CVE-2015-8411, CVE-2015-8412, CVE-2015-8413, CVE-2015-8414, CVE-2015-8420, CVE-2015-8421, CVE-2015-8422, CVE-2015-8423, CVE-2015-8424, CVE-2015-8425, CVE-2015-8426, CVE-2015-8427, CVE-2015-8428, CVE-2015-8429, CVE-2015-8430, CVE-2015-8431, CVE-2015-8432, CVE-2015-8433, CVE-2015-8434, CVE-2015-8435, CVE-2015-8436, CVE-2015-8437, CVE-2015-8441, CVE-2015-8442, CVE-2015-8447, CVE-2015-8448, CVE-2015-8449, CVE-2015-8450,
```

CVE-2015-8452, CVE-2015-8454, CVE-2015-8653, CVE-2015-8655, CVE-2015-8821, and CVE-2015-8822."
},
{
"lang": "es",
"value": "Vulnerabilidad de uso despu\u00e9s de liberaci\u00f3n de memoria en la implementaci\u00f3n de objeto TextField en Adobe Flash Player en versiones anteriores a 18.0.0.268 y 19.x y 20.x en versiones anteriores a 20.0.0.228 en Windows y OS X y en versiones anteriores a 11.2.202.554 en Linux, Adobe AIR en versiones anteriores a 20.0.0.204, Adobe AIR SDK en versiones anteriores a 20.0.0.204 y Adobe AIR SDK & Compiler en versiones anteriores a 20.0.0.204 permite a atacantes ejecutar c\u00f3digo arbitrario a trav\u00e9s de la propiedad text manipulada, una vulnerabilidad diferente a CVE-2015-8048, CVE-2015-8049, CVE-2015-8050, CVE-2015-8055, CVE-2015-8056, CVE-2015-8057, CVE-2015-8058, CVE-2015-8059, CVE-2015-8061, CVE-2015-8062, CVE-2015-8063, CVE-2015-8064, CVE-2015-8065, CVE-2015-8066, CVE-2015-8067, CVE-2015-8068, CVE-2015-8069, CVE-2015-8070, CVE-2015-8071, CVE-2015-8401, CVE-2015-8402, CVE-2015-8403, CVE-2015-8404, CVE-2015-8405, CVE-2015-8406, CVE-2015-8410, CVE-2015-8411, CVE-2015-8412, CVE-2015-8413, CVE-2015-8414, CVE-2015-8420, CVE-2015-8421, CVE-2015-8422, CVE-2015-8423, CVE-2015-8424, CVE-2015-8425, CVE-2015-8426, CVE-2015-8427, CVE-2015-8428, CVE-2015-8429, CVE-2015-8430, CVE-2015-8431, CVE-2015-8432, CVE-2015-8433, CVE-2015-8434, CVE-2015-8435, CVE-2015-8436, CVE-2015-8437, CVE-2015-8441, CVE-2015-8442, CVE-2015-8447, CVE-2015-8448, CVE-2015-8449, CVE-2015-8450, CVE-2015-8452, CVE-2015-8454, CVE-2015-8653, CVE-2015-8655, CVE-2015-8821 y CVE-2015-8822."
}
],
"metrics": {
"cvssMetricV31": [
{
"source": "nvd@nist.gov",
"type": "Primary",
"cvssData": {
"version": "3.1",
"vectorString": "CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H",
"attackVector": "NETWORK",
"attackComplexity": "LOW",
"privilegesRequired": "NONE",
"userInteraction": "REQUIRED",
"scope": "UNCHANGED",
"confidentialityImpact": "HIGH",
"integrityImpact": "HIGH",
"availabilityImpact": "HIGH",

```
            "baseScore": 8.8,
            "baseSeverity": "HIGH"
          },
          "exploitabilityScore": 2.8,
          "impactScore": 5.9
        }
      ],
      "cvssMetricV2": [
        {
          "source": "nvd@nist.gov",
          "type": "Primary",
          "cvssData": {
            "version": "2.0",
            "vectorString": "AV:N/AC:M/Au:N/C:C/I:C/A:C",
            "accessVector": "NETWORK",
            "accessComplexity": "MEDIUM",
            "authentication": "NONE",
            "confidentialityImpact": "COMPLETE",
            "integrityImpact": "COMPLETE",
            "availabilityImpact": "COMPLETE",
            "baseScore": 9.3
          },
          "baseSeverity": "HIGH",
          "exploitabilityScore": 8.6,
          "impactScore": 10.0,
          "acInsufInfo": false,
          "obtainAllPrivilege": false,
          "obtainUserPrivilege": false,
          "obtainOtherPrivilege": false,
          "userInteractionRequired": true
        }
      ]
    },
    "weaknesses": [
      {
        "source": "nvd@nist.gov",
        "type": "Primary",
        "description": [
          {
            "lang": "en",
            "value": "CWE-416"
          }
        ]
      }
    ],
```

```
                "configurations": [
                  {
                    "operator": "AND",
                    "nodes": [
                      {
                        "operator": "OR",
                        "negate": false,
                        "cpeMatch": [
                          {
                            "vulnerable": false,
                            "criteria":
"cpe:2.3:o:microsoft:windows_8.0:*:*:*:*:*:*:*:*",
                            "matchCriteriaId":    "461CBD40-CB18-4868-BAB4-
CCBD724B9E07"
                          },
                          {
                            "vulnerable": false,
                            "criteria":
"cpe:2.3:o:microsoft:windows_8.1:*:*:*:*:*:*:*:*",
                            "matchCriteriaId":    "A7F51B5F-AA19-4D31-89FA-
6DFAC4BA8F0F"
                          }
                        ]
                      },
                      {
                        "operator": "OR",
                        "negate": false,
                        "cpeMatch": [
                          {
                            "vulnerable": true,
                            "criteria":
"cpe:2.3:a:adobe:flash_player:*:*:*:*:*:internet_explorer:*:*",
                            "versionEndIncluding": "19.0.0.245",
                            "matchCriteriaId":     "9A6F84D7-62F0-45C0-962B-
5EC8946B67AA"
                          }
                        ]
                      }
                    ]
                  },
                  {
                    "operator": "AND",
                    "nodes": [
                      {
                        "operator": "OR",
```

```
                        "negate": false,
                        "cpeMatch": [
                          {
                            "vulnerable": true,
                            "criteria": "cpe:2.3:a:adobe:air:*:*:*:*:*:*:*:*",
                            "versionEndIncluding": "19.0.0.241",
                            "matchCriteriaId": "044936DC-41A9-407F-BE64-
B0D6FD7F501E"
                          }
                        ]
                      },
                      {
                        "operator": "OR",
                        "negate": false,
                        "cpeMatch": [
                          {
                            "vulnerable": false,
                            "criteria": "cpe:2.3:o:apple:mac_os_x:-:*:*:*:*:*:*:*",
                            "matchCriteriaId": "4781BF1E-8A4E-4AFF-9540-
23D523EE30DD"
                          },
                          {
                            "vulnerable": false,
                            "criteria": "cpe:2.3:o:google:android:-:*:*:*:*:*:*:*",
                            "matchCriteriaId": "F8B9FEC8-73B6-43B8-B24E-
1F7C20D91D26"
                          },
                          {
                            "vulnerable": false,
                            "criteria": "cpe:2.3:o:microsoft:windows:-
:*:*:*:*:*:*:*:*",
                            "matchCriteriaId": "A2572D17-1DE6-457B-99CC-
64AFD54487EA"
                          }
                        ]
                      }
                    ]
                  },
                  {
                    "operator": "AND",
                    "nodes": [
                      {
                        "operator": "OR",
                        "negate": false,
                        "cpeMatch": [
```

```
                        {
                          "vulnerable": true,
                          "criteria":
"cpe:2.3:a:adobe:flash_player:*:*:*:*:esr:*:*:*",
                          "versionEndIncluding": "18.0.0.261",
                          "matchCriteriaId":    "40DB16C2-C074-4BA5-BAC3-
7BF751724E01"
                        }
                      ]
                    },
                    {
                      "operator": "OR",
                      "negate": false,
                      "cpeMatch": [
                        {
                          "vulnerable": false,
                          "criteria": "cpe:2.3:o:apple:mac_os_x:-:*:*:*:*:*:*:*",
                          "matchCriteriaId":      "4781BF1E-8A4E-4AFF-9540-
23D523EE30DD"
                        },
                        {
                          "vulnerable": false,
                          "criteria":                "cpe:2.3:o:microsoft:windows:-
:*:*:*:*:*:*:*",
                          "matchCriteriaId":     "A2572D17-1DE6-457B-99CC-
64AFD54487EA"
                        }
                      ]
                    }
                  ]
                },
                {
                  "operator": "AND",
                  "nodes": [
                    {
                      "operator": "OR",
                      "negate": false,
                      "cpeMatch": [
                        {
                          "vulnerable": true,
                          "criteria":
"cpe:2.3:a:adobe:flash_player:*:*:*:*:*:chrome:*:*",
                          "versionEndIncluding": "19.0.0.245",
                          "matchCriteriaId":     "82EBB50D-BFC5-4165-A673-
D98BD23C79C8"
```

```
                        }
                    ]
                },
                {
                    "operator": "OR",
                    "negate": false,
                    "cpeMatch": [
                        {
                            "vulnerable": false,
                            "criteria": "cpe:2.3:o:apple:mac_os_x:-:*:*:*:*:*:*:*",
                            "matchCriteriaId": "4781BF1E-8A4E-4AFF-9540-
23D523EE30DD"
                        },
                        {
                            "vulnerable": false,
                            "criteria": "cpe:2.3:o:google:chrome_os:-
:*:*:*:*:*:*:*",
                            "matchCriteriaId": "D32ACF6F-5FF7-4815-8EAD-
4719F5FC9B79"
                        },
                        {
                            "vulnerable": false,
                            "criteria": "cpe:2.3:o:linux:linux_kernel:-
:*:*:*:*:*:*:*",
                            "matchCriteriaId": "703AF700-7A70-47E2-BC3A-
7FD03B3CA9C1"
                        },
                        {
                            "vulnerable": false,
                            "criteria": "cpe:2.3:o:microsoft:windows:-
:*:*:*:*:*:*:*",
                            "matchCriteriaId": "A2572D17-1DE6-457B-99CC-
64AFD54487EA"
                        }
                    ]
                }
            ]
        },
        {
            "operator": "AND",
            "nodes": [
                {
                    "operator": "OR",
                    "negate": false,
                    "cpeMatch": [
```

```
                        {
                          "vulnerable": true,
                          "criteria":
"cpe:2.3:a:adobe:flash_player:*:*:*:*:*:*:*:*",
                          "versionEndIncluding": "11.2.202.548",
                          "matchCriteriaId":       "A02328C4-D810-4774-8F28-
2B5FB6C7CDB5"
                        }
                      ]
                    },
                    {
                      "operator": "OR",
                      "negate": false,
                      "cpeMatch": [
                        {
                          "vulnerable": false,
                          "criteria":                "cpe:2.3:o:linux:linux_kernel:-
:*:*:*:*:*:*:*:*",
                          "matchCriteriaId":      "703AF700-7A70-47E2-BC3A-
7FD03B3CA9C1"
                        }
                      ]
                    }
                  ]
                },
                {
                  "operator": "AND",
                  "nodes": [
                    {
                      "operator": "OR",
                      "negate": false,
                      "cpeMatch": [
                        {
                          "vulnerable": false,
                          "criteria":            "cpe:2.3:o:microsoft:windows_10:-
:*:*:*:*:*:*:*:*",
                          "matchCriteriaId":       "21540673-614A-4D40-8BD7-
3F07723803B0"
                        }
                      ]
                    },
                    {
                      "operator": "OR",
                      "negate": false,
                      "cpeMatch": [
```

```
                        {
                          "vulnerable": true,
                          "criteria":
"cpe:2.3:a:adobe:flash_player:*:*:*:*:*:edge:*:*",
                          "versionEndIncluding": "19.0.0.245",
                          "matchCriteriaId":     "C5C96375-3919-417F-ADDC-
657F3676EF91"
                        }
                      ]
                    }
                  ]
                },
                {
                  "operator": "AND",
                  "nodes": [
                    {
                      "operator": "OR",
                      "negate": false,
                      "cpeMatch": [
                        {
                          "vulnerable": true,
                          "criteria": "cpe:2.3:a:adobe:air_sdk:*:*:*:*:*:*:*:*",
                          "versionEndIncluding": "19.0.0.241",
                          "matchCriteriaId":     "89A1DBA3-8B4E-4832-8D39-
6490CD99FE6B"
                        }
                      ]
                    },
                    {
                      "operator": "OR",
                      "negate": false,
                      "cpeMatch": [
                        {
                          "vulnerable": false,
                          "criteria": "cpe:2.3:o:apple:iphone_os:-:*:*:*:*:*:*:*",
                          "matchCriteriaId":      "B5415705-33E5-46D5-8E4D-
9EBADC8C5705"
                        },
                        {
                          "vulnerable": false,
                          "criteria": "cpe:2.3:o:apple:mac_os_x:-:*:*:*:*:*:*:*",
                          "matchCriteriaId":      "4781BF1E-8A4E-4AFF-9540-
23D523EE30DD"
                        },
                        {
```

```
                        "vulnerable": false,
                        "criteria": "cpe:2.3:o:google:android:-:*:*:*:*:*:*:*",
                        "matchCriteriaId": "F8B9FEC8-73B6-43B8-B24E-
1F7C20D91D26"
                     },
                     {
                        "vulnerable": false,
                        "criteria": "cpe:2.3:o:microsoft:windows:-
:*:*:*:*:*:*:*",
                        "matchCriteriaId": "A2572D17-1DE6-457B-99CC-
64AFD54487EA"
                     }
                  ]
               }
            ]
         },
         {
            "operator": "AND",
            "nodes": [
               {
                  "operator": "OR",
                  "negate": false,
                  "cpeMatch": [
                     {
                        "vulnerable": true,
                        "criteria":
"cpe:2.3:a:adobe:air_sdk_\\&_compiler:*:*:*:*:*:*:*:*",
                        "versionEndIncluding": "19.0.0.241",
                        "matchCriteriaId": "7C30B2BE-C291-495C-B7A8-
A27492BE7177"
                     }
                  ]
               },
               {
                  "operator": "OR",
                  "negate": false,
                  "cpeMatch": [
                     {
                        "vulnerable": false,
                        "criteria": "cpe:2.3:o:apple:iphone_os:-:*:*:*:*:*:*:*",
                        "matchCriteriaId": "B5415705-33E5-46D5-8E4D-
9EBADC8C5705"
                     },
                     {
                        "vulnerable": false,
```

```
                                    "criteria": "cpe:2.3:o:apple:mac_os_x:-:*:*:*:*:*:*:*",
                                    "matchCriteriaId": "4781BF1E-8A4E-4AFF-9540-
23D523EE30DD"
                                },
                                {
                                    "vulnerable": false,
                                    "criteria": "cpe:2.3:o:google:android:-:*:*:*:*:*:*:*",
                                    "matchCriteriaId": "F8B9FEC8-73B6-43B8-B24E-
1F7C20D91D26"
                                },
                                {
                                    "vulnerable": false,
                                    "criteria": "cpe:2.3:o:microsoft:windows:-
:*:*:*:*:*:*:*",
                                    "matchCriteriaId": "A2572D17-1DE6-457B-99CC-
64AFD54487EA"
                                }
                            ]
                        }
                    ]
                },
                {
                    "operator": "AND",
                    "nodes": [
                        {
                            "operator": "OR",
                            "negate": false,
                            "cpeMatch": [
                                {
                                    "vulnerable": true,
                                    "criteria":
"cpe:2.3:a:adobe:flash_player_desktop_runtime:*:*:*:*:*:*:*:*",
                                    "versionEndIncluding": "19.0.0.245",
                                    "matchCriteriaId": "60921187-5894-4500-A822-
02986DC497C9"
                                }
                            ]
                        },
                        {
                            "operator": "OR",
                            "negate": false,
                            "cpeMatch": [
                                {
                                    "vulnerable": false,
                                    "criteria": "cpe:2.3:o:apple:mac_os_x:-:*:*:*:*:*:*:*",
```

```
                                    "matchCriteriaId": "4781BF1E-8A4E-4AFF-9540-
23D523EE30DD"
                                 },
                                 {
                                    "vulnerable": false,
                                    "criteria": "cpe:2.3:o:microsoft:windows:-
:*:*:*:*:*:*:*",
                                    "matchCriteriaId": "A2572D17-1DE6-457B-99CC-
64AFD54487EA"
                                 }
                              ]
                           }
                        ]
                     },
                     {
                        "operator": "AND",
                        "nodes": [
                           {
                              "operator": "OR",
                              "negate": false,
                              "cpeMatch": [
                                 {
                                    "vulnerable": false,
                                    "criteria":
"cpe:2.3:o:microsoft:windows_10:*:*:*:*:*:*:*:*",
                                    "matchCriteriaId": "FBC814B4-7DEC-4EFC-ABFF-
08FFD9FD16AA"
                                 },
                                 {
                                    "vulnerable": false,
                                    "criteria": "cpe:2.3:o:microsoft:windows_8.0:-
:*:*:*:*:*:*:*",
                                    "matchCriteriaId": "F265782D-8CEC-4C97-83A3-
86404A1C09BE"
                                 },
                                 {
                                    "vulnerable": false,
                                    "criteria": "cpe:2.3:o:microsoft:windows_8.1:-
:*:*:*:*:*:*:*",
                                    "matchCriteriaId": "E93068DB-549B-45AB-8E5C-
00EB5D8B5CF8"
                                 }
                              ]
                           },
                           {
```

```
                        "operator": "OR",
                        "negate": false,
                        "cpeMatch": [
                           {
                              "vulnerable": true,
                              "criteria":
"cpe:2.3:a:adobe:flash_player:*:*:*:*:*:internet_explorer:*:*",
                              "versionEndIncluding": "19.0.0.245",
                              "matchCriteriaId":       "9A6F84D7-62F0-45C0-962B-
5EC8946B67AA"
                           }
                        ]
                     }
                  ]
               }
            ],
            "references": [
               {
                  "url":       "http://www.zerodayinitiative.com/advisories/ZDI-15-
665",
                  "source": "psirt@adobe.com",
                  "tags": [
                     "Third Party Advisory",
                     "VDB Entry"
                  ]
               },
               {
                  "url":              "https://helpx.adobe.com/security/products/flash-
player/apsb15-32.html",
                  "source": "psirt@adobe.com",
                  "tags": [
                     "Patch",
                     "Vendor Advisory"
                  ]
               }
            ]
         }
      },
      {
         "cve": {
            "id": "CVE-2016-0168",
            "sourceIdentifier": "secure@microsoft.com",
            "published": "2016-05-11T01:59:07.680",
            "lastModified": "2018-10-12T22:11:24.147",
            "vulnStatus": "Modified",
```

"cveTags": [],
"descriptions": [
{
"lang": "en",
"value": "GDI in Microsoft Windows Vista SP2, Windows Server 2008 SP2 and R2 SP1, Windows 7 SP1, Windows 8.1, Windows Server 2012 Gold and R2, Windows RT 8.1, and Windows 10 Gold and 1511 allows remote attackers to obtain sensitive information via a crafted document, aka \"Windows Graphics Component Information Disclosure Vulnerability,\" a different vulnerability than CVE-2016-0169."
},
{
"lang": "es",
"value": "GDI en Microsoft Windows Vista SP2, Windows Server 2008 SP2 y R2 SP1, Windows 7 SP1, Windows 8.1, Windows Server 2012 Gold y R2, Windows RT 8.1 y Windows 10 Gold y 1511 permite a atacantes remotos obtener informaci\u00f3n sensible a trav\u00e9s de un documento manipulado, tambi\u00e9n conocido como \"Windows Graphics Component Information Disclosure Vulnerability\", una vulnerabilidad diferente a CVE-2016-0169."