Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління
(повна назва)

Кафедра Безпеки інформаційних технологій
(повна назва)

АТЕСТАЦІЙНА РОБОТА

Пояснювальна записка

рівень вищої освіти другий

(магістерський)

Критичні вразливості захисту інформації в ОС
(тема)

Виконав: Ревізорова К.І.
(прізвище, ініціали)

студент 2 курсу, групи БІКСм-18-1

Спеціальність 125 Кібербезпека
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма «Безпека інформаційних і комунікаційних систем»
(повна назва освітньої програми)

Керівник доцент Гріненко Т.О.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ Халімов Г.З.
(підпис) (прізвище, ініціали)

2019 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління
(повна назва)
Кафедра Безпеки інформаційних технологій
(повна назва)
Рівень вищої освіти другий (магістерський)
Спеціальність 125 Кібербезпека
(код і повна назва)
Тип програми освітньо-професійна
(освітньо-професійна, або освітньо-наукова)
Освітня програма «Безпека інформаційних і комунікаційних систем»
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри
_____
(підпис)
«____»_____ 20 ___ р.

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ

студентові *Ревізоровій Карині Ігорівні*
(прізвище, ім'я, по батькові)

1. Тема роботи *Критичні вразливості захисту інформації в ОС*
затверджена наказом по університету від "04" листопада 2019 р. № 1649Ст
2. Термін подання студентом роботи до екзаменаційної комісії
3. Вихідні дані до роботи
1. *Бази даних вразливостей. Операційні системи. Вразливості. Метрики оцінки вразливостей.*
2.The Red book. ISO/IEC 27000. Life-cycle of a Security Vulnerabilit. Threat and Vulnerability Management Standard. Experience Report: Study of Vulnerabilities of Enterprise Operating Systems
3. База даних вразливостей NVD
4. Перелік питань, що потрібно опрацювати в роботі
1. Аналіз вразливостей в інформаційному середовищі
2. Аналіз баз даних вразливостей
3. Вибірка метрик для оцінки критичності вразливостей
4. Програмна реалізація розробленої методики оцінки критичності вразливостей
5. Результати досліджень
5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій *Презентаційний матеріал у вигляді слайдів*

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	*Отримання завдання*	*9.09.18*	
2	*Пошук інформації та літератури*	*10.09.18-10.02.19*	
3	*Обробка зібраних даних*	*11.02.19-19.04.19*	
4	*Програмна реалізація метидики оцінки критичності вразливостей в ОС*	*20.04.19-25.07.19*	
5	*Висновки над виконаною роботою*	*26.07.19-26.08.19*	
6	*Оформлення пояснювальної записки*	*27.08.19-31.10.19*	

Дата видачі завдання _____ _____20__ р.

Студент _____
(підпис)

Керівник роботи (проекту) _____ доцент Гріненко Т.О._
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка до атестаційної роботи магістра: 88 с., 24 рис., 9 табл., 48 джерел, 1 додаток.

БАЗА ДАННИХ ВРАЗЛИВОСТЕЙ, ВРАЗЛИВІСТЬ, МЕТРИКА ОЦІНКИ КРИТИЧНОСТІ, ОПЕРАЦІЙНА СИСТЕМА, ОЦІНКА КРИТИЧНОСТІ ВРАЗЛИВОСТЕЙ

Об'єкт дослідження – захист інформації в ОС.

Предмет дослідження – вразливості захисту інформації в ОС.

Мета роботи – аналіз вразливостей в інформаційному середовищі, методик оцінки критичності вразливостей і існуючих програмних рішень на світовому ринку; обґрунтування вибору метрик та параметрів при розробці програмного продукту для оцінки критичності вразливостей.

Методи дослідження – аналіз баз даних вразливостей, методик оцінки критичності вразливостей.

Проведений аналіз вразливостей в інформаційному середовищі, методик оцінки критичності вразливостей і існуючих програмних рішень на світовому ринку. Обґрунтовано вибір метрик та параметрів при розробці програмного продукту для оцінки критичності вразливостей. Результатом роботи є програмна реалізація засобу для оцінки рівня критичності вразливостей в ОС.

За допомогою розробленої програми можна отримати експертну характеристику вразливостей для облегшення та пришвидшення майбутнього аналізу ризиків. Також цей засіб можна використовувати як допоміжну утиліту для тестування безпеки, планування і розробки патчів безпеки. Крім того, вона надає достатньо інформації для вибору оптимальної за рівнем безпеки (за показниками користувачів) операційної системи для реалізації своїх бізнес рішень.

ABSTRACT

Explanatory note consists of 88 pages, 24 images, 9 tables, 48 links, 1 addition.

VULNERABILITY DATABASE, VULNERABILITY, CRITICAL ASSESSMENT METRIC, OPERATING SYSTEM, CRITICAL ASSESSMENT OF VULNERABILITIES.

The object of study - protection of information in the OS.

The subject of the study is information security vulnerabilities in the operating system.

Purpose of the work - analysis of vulnerabilities in the information environment, methods of assessing the criticality of vulnerabilities and existing software solutions in the world market; substantiation of the choice of metrics and parameters when developing a software product to assess the vulnerability criticality.

Research methods - analysis of vulnerability databases, methods of vulnerability criticality assessment.

The analysis of vulnerabilities in the information environment, methods of assessing the criticality of vulnerabilities and existing software solutions in the world market. The choice of metrics and parameters when developing a software product to evaluate the vulnerability of the vulnerability is substantiated. The result of the work is a software implementation of a tool for assessing the level of criticality of vulnerabilities in the OS.

The developed program helps you get vulnerability expertise to facilitate and accelerate future risk analysis. You can also use this tool as an auxiliary utility for security testing, planning, and development of security patches. In addition, it provides enough information to select the best operating system security level for your business decisions.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАК, ОДИНИЦЬ І ТЕРМІНІВ

ІБ інформаційна безпека;

ІТ інформаційні технології;

СУБД система управління базами даних;

БД база даних;

ОС операційна система;

ПЗ програмне забезпечення;

КС комп’ютерні системи;

КЗЗ комплекси засобів захисту;

НСД несанкціонований доступ.

ВСТУП

Проблема обробки й аналізу інформації є однією знай актуальніших. Невпинне зростання інформаційних масивів і кількісних даних змушує замислюватись над тим, як виконати її обробку й аналіз. Існує досить великий клас систем обробки інформації, під час розробки яких фактор безпеки відіграє першорядну роль (наприклад, банківські, медичні, економічні системи) [2].

Одним з важливих заходів захисту інформації в комп'ютеризованих системах є визначення переліку загроз інформації. Одна або декілька загроз можуть використовувати ряд вразливостей інформації [2].

Інформаційна безпека (ІБ) – властивість системи протягом заданого часу протистояти несанкціонованому зняттю і модифікації інформації [3]. Під несанкціонованим зняттям розуміється отримання інформації, до якої у абонента немає доступу, тобто порушення правил доступу. Під несанкціонованою модифікацією розуміється зміна інформації, яка призводить до порушення її цілісності [4].

ІБ може містити процес управління ризиками. Ризик – можливість виникнення та вірогідні масштаби наслідків негативного впливу протягом певного періоду часу [5]. Ризик інформаційної безпеки – добуток втрат від порушення конфіденційності, цілісності, автентичності або доступності інформаційних ресурсів на імовірність такого порушення [6]. Загроза – можлива причина небажаного інциденту, який може завдати шкоди системі або організації [6].

Існує багато методик оцінювання ризиків (NIST 800-30, методика CRAMM, OCTAVE та ін.) [7]. У процесі оцінювання ризиків бере участь оцінка загроз, їх наслідків, вразливостей інформації і засобів їх обробки, а також ймовірності їх виникнення [8]. Вразливість в інформаційних технологіях (ІТ) – недолік або слабке місце, яке може бути використане для реалізації загрози [9].

Більшість хакерських атак стає можливими через наявність вразливостей в існуючих операційних систем (ОС) та програмного забезпечення(ПЗ) [10].

У мережі Internet з'являється все більше шкідливого коду, який використовує вразливості для проникнення в комп'ютери, виконання запрограмованих дій і подальшого свого поширення [10].

У кожній ОС є вразливості, але не для всіх є відповідний засіб експлуатування вразливостей. Вразливості усуваються за допомогою оновлень чи патчів, що випускаються розробниками. Але часто оновлення для ПЗ теж мають вразливості, а інколи, навіть більш шкідливі ніж ті, які були до оновлення [10].

Операційна система є найважливішим програмним компонентом будь-якої обчислювальної машини, тому від рівня реалізації політики безпеки в кожній конкретній операційній системі залежить і загальна безпека інформаційної системи [11]. Організація ефективного та надійного захисту ОС неможлива без попереднього аналізу можливих загроз її безпеки. Загрози безпеки ОС істотно залежать від умов експлуатації системи, від того, яка інформація зберігається і обробляється в системі, тощо. Наприклад, якщо ОС використовується для організації електронного документообігу, найбільш небезпечними загрозами є такі, що пов'язані з порушенням цілісності та доступності файлів. [12].

Мета роботи – проаналізувати проблеми вразливостей у операційних системах. Дослідити та порівняти бази даних вразливостей. Визначити та порівняти існуючі методики оцінки критичності вразливостей. Провести огляд існуючих програмних рішень на світовому ринку. Здійснити вибірку метрик для оцінки критичності вразливостей.

1 ПРИНЦИПИ СТВОРЕННЯ ЗАХИЩЕНИХ СИСТЕМ

1.1 Поняття та функції ОС

Операційна система (ОС) позбавляє програмістів і користувачів не тільки від необхідності безпосередньо працювати з апаратурою дискового накопичувача, надаючи їм простий файловий інтерфейс, але і бере на себе всі інші рутинні операції, пов'язані з управлінням іншими апаратними пристроями. За допомогою ОС комп'ютер, здатний виконувати лише дії, які визначаються системою команд, перетворюється в віртуальну машину, що виконує широкий набір набагато складніших функцій. Віртуальна машина також управляється командами, однак вони відносяться до більш високого рівня: для відкриття файлу з відомим ім'ям досить командою запустити деяку прикладну програму. Таку машину легше програмувати, з нею легше працювати, ніж безпосередньо з апаратурою реального комп'ютера або реальної мережі [11].

Операційна система (ОС) ¬ це сукупність програмних засобів, що призначені для автоматизованого керування виконанням програми та надання користувачам певних послуг [2].

Операційну систему можна представити як засіб управління ресурсами обчислювальної системи з метою найбільш ефективного їх використання. До числа основних ресурсів сучасних обчислювальних систем відносяться процесори, основна пам'ять, дискові накопичувачі, принтери, таймери, набори даних, мережеві і інші пристрої. Ресурси розподіляються між процесами. Процес (або завдання) є базовим поняттям більшості сучасних ОС і часто коротко визначається як виконується програма. Якщо програма являє собою статичний об'єкт (у вигляді файлу з кодами і даними), то процес - динамічний об'єкт, який виникає в ОС після того, як користувач або вона сама запускає програму на виконання [11].

Перша функція ОС – керування ресурсами комп'ютера та їх розподіл. Ресурси – це логічні й фізичні компоненти комп'ютера: оперативна пам'ять, місце на диску, периферійні пристрої, процесорний час, тощо. Управління ресурсами включає рішення таких задач, як визначення, якому процесу, коли і в якій кількості слід виділити даний ресурс; відстеження стану і облік використання ресурсу; постачання оперативну інформацію про те, зайнятий або вільний ресурс і яка частка ресурсу вже розподілена. Управління ресурсами складає важливу частину функцій будь-якої ОС, які реалізуються за допомогою підсистем управління ресурсами.

Наступна функція ОС – керування обчислювальними процесами. Обчислювальним процесом (або завданням) називається послідовність дій, яка задається програмою. У теорії, функції керування процесами можна було б передати кожній прикладній програмі, але тоді програми були б набагато більшими та складнішими. Тому існування однієї основної програми комп'ютері, що здійснює керування процесами інших програм – логічне технічне рішення.

Для виконання третьої функції ОС – забезпечення взаємодії користувача з апаратурою – служить інтерфейс користувача ОС. До складу інтерфейсу користувача входить також набір сервісних програм – утиліт. Утиліта – це невелика програма, що виконує конкретну сервісну функцію. Утиліти звільняють користувача від виконання рутинних і часом досить складних операцій [12].

Сучасні ОС надають користувачеві широкий спектр сервісних послуг. Чим досконалішою є ОС, тим зручніше у ній працювати користувачу.

1.2 Комплекс засобів захисту операційних систем

Комп'ютерна система, як правило, складається з безлічі компонентів. Деякі з компонентів можуть бути спеціально призначені для реалізації політики безпеки (наприклад, засоби ізоляції процесів або керування потоками інформації). Інші можуть впливати на безпеку опосередковано, наприклад, забезпечувати функціонування компонентів першого типу. І, нарешті, треті можуть взагалі не бути задіяні під час вирішення завдань забезпечення безпеки. Множина всіх компонентів перших двох типів називається комплексом засобів захисту (КЗЗ). Іншими словами, КЗЗ це сукупність всіх програмно-апаратних засобів, в тому числі програмних забезпечень (ПЗ), задіяних під час реалізації політики безпеки. Частина комп’ютерних систем (КС), що складає КЗЗ, визначається розробником. Будь-який компонент КС, який внаслідок якого-небудь впливу здатний спричинити порушення політики безпеки, повинен розглядатись як частина КЗЗ [1].

Комплекс засобів захисту розглядає ресурси КС як об'єкти і керує взаємодією цих об'єктів відповідно до політики безпеки інформації, що реалізується. Як об'єкти ресурси характеризуються двома аспектами: логічне подання (зміст, семантика, значення) і фізичне (форма, синтаксис). Об'єкт характеризується своїм станом, що в свою чергу характеризується атрибутами і поводженням, яке визначає способи зміни стану. Для різних КС об'єкти можуть бути різні. Наприклад, для системи управління базами даних (СУБД) в якості об'єктів можна розглядати записи баз даних, а для операційної системи процеси, файли, кластери, сектори дисків, сегменти пам'яті і т. ін. Все, що підлягає захисту відповідно до політики безпеки, має бути визначено як об'єкт [12].

У сучасних умовах застосовуються дві технології створення захищених ОС: розроблення захищених систем «з нуля» і побудова так званих «довірених» версій шляхом модернізації наявних систем.

Під час розроблення захищених систем «з нуля» всі їхні функціональні можливості та архітектурні рішення (які мають бути сертифіковані за встановленим класом вимог) закладаються на етапі проектування. Характерною рисою такого підходу є розроблення методів гарантованої реалізації встановлених вимог. У цьому випадку застосовують класичну схему проектування захищених систем [12]:

- розробка моделі безпеки;
- визначення об'єктів взаємодії;
- визначення правил керування доступом;
- вибір механізмів керування доступом;
- вибір методів ідентифікації й автентифікації взаємодіючих сторін;
- визначення множини подій, що підлягають аудиту;
- реалізація системи.

Хоча прикладів застосування такого підходу небагато через складність його реалізації та велику вартість (системи Trusted Xenix, Trusted Mach, Harris CX/SX) слід зазначити, що лише таким чином можна було створити системи, сертифіковані відповідно до найвищих вимог [12].

Під час побудови «довірених» версій шляхом модернізації наявних систем, як правило, до останніх додають функції шифрування, цифрового підпису, підсилюють у них керування доступом через впровадження мандатного керування, розподіляють обов'язки адміністратора системи між різними обліковими записами чи «ролями», впроваджують додаткові засоби ідентифікації й автентифікації, аудита та моніторингу. Цей підхід до створення захищених систем переважає насамперед завдяки своїй економічності, яку зумовлюють менший обсяг робіт зі створення й реалізації системи та можливість збереження сумісності наявними рішеннями. Крім того, модернізовані системи успадковують імідж систем-прототипів (створений популярністю фірм-розробників), що підвищує довіру до них і дає змогу

вико-ристовувати досвід їх експлуатації й супроводження. Типові приклади реалізації такого підходу ОС Trusted Solaris, СКБД Trusted Oracle [12].

Слід зазначити, що обидва підходи не суперечать один одному, а є рівноправними складовими технологіями для створення захищених ОС.

Одним із способів оцінки безпеки ОС є стандарти, дотримання яких дають певну гарантію захищеності. Основна проблема українських стандартів безпеки ОС це відсутність свого стандарту безпеки ОС. Стандарти захисту даних, утворюють базис понять, на якому будуються всі дії щодо забезпечення інформаційної безпеки (ІБ). У той же час стандарти орієнтовані в першу чергу на виробників ПЗ і експертів в області безпеки і в набагато меншому ступені на користувачів і адміністраторів [14].

Будь-який міжнародний стандарт безпеки передбачає наявність наступних етапів [14]:

– визначення цілей забезпечення ІБ комп'ютерних систем;

– створення ефективної системи управління ІБ;

– розробка критеріїв і показників ефективності використання ОС з точки зору забезпечення ІБ;

– розрахунок комплексного показника ІБ для оцінки відповідної ОС;

– застосування інструментарію забезпечення ІБ і оцінки її поточного стану;

– використання методик управління безпекою з обґрунтованою системою метрик і заходів забезпечення ІБ, що дозволяють об'єктивно оцінити захищеність інформаційних активів.

Прикладами таких стандартів є міжнародний стандарт ISO/IEC 15408 та ISO/IEC 17799:2002 (BS 7799:2000).

Використання методик першого стандарту дозволяє визначити для компанії ті критерії, які можуть бути використані в якості основи для вироблення оцінок захисних властивостей продуктів і систем інформаційної технології. Крім того, ці методики дозволяють проводити найбільш повне

порівняння результатів оцінки захисних властивостей корпоративних інформаційних систем за допомогою загального переліку (набору) вимог для функцій захисту продуктів і систем, а також методів точних вимірювань, які проводяться під час отримання оцінок захисту. Головний недолік стандарту в тому, що за умовами використання його інструкцій не можна вирішити проблему несанкціонованого доступу, модифікації або втрати доступу до інформації в результаті випадкових або навмисних дій і ряд інших аспектів ІБ [14].

Другий стандарт ISO/IEC 17799:2002 (BS 7799:2000) «Управління інформаційною безпекою – Інформаційні технології» розглядає такі питання як:

– необхідність забезпечення ІБ;

– основні поняття і визначення ІБ;

– управління доступом;

– вимоги з безпеки до корпоративних інформаційних систем в ході їх розробки, експлуатації і супроводу.

Зазначені стандарти мають рекомендаційну сутність та підходять для побудови «довірених» версій шляхом модернізації наявних систем та для систем, розробляємих з «нуля».

При розробці захищених систем, крім стандартів, у основу покладені п’ять принципів: інтегрованості, інваріантності, уніфікації, адекватності.

Принцип інтегрованості. Засоби захисту слід вбудовувати в систему таким чином, щоб вони мали змогу контролювати всі без винятку механізми взаємодії. Найпростіший метод реалізації цього принципу під час створення ОС максимальне обмеження кількості механізмів взаємодії та інтеграція засобів захисту безпосередньо в ці механізми. Принцип інваріантності засоби захисту мають бути не залежними від особливостей реалізації утиліт і прикладних програм, а також від логіки їх функціонування, крім того, вони мають бути універсальними для взаємодій усіх типів. Інваріантності засобів захисту для ОС можна досягти шляхом застосування суворо регламентованої парадигми

функціонування програм, що обмежує способи їх взаємодій. Принцип уніфікації має існувати однозначна відповідність між взаємодіями суб'єктів і об'єктів, що контролюються, та операціями доступу, керування якими описано в моделях безпеки. Це дає змогу зробити засоби захисту уні-версальними і використовувати їх без змін для реалізації різних моделей без-пеки та для контролю доступу до об'єктів, що мають різну природу. Під час створення ОС дотримання цього принципу приводить до необхідності розроблення універсального інтерфейсу доступу (що об'єднує всі способи взає-модій між суб'єктами й об'єктами), всі функції якого однозначно відображаються на множину операцій, що описує модель безпеки [3].

Принцип адекватності. Для забезпечення реальної здатності протидіяти атакам необхідно виключити всі чинники, що спричиняють виникнення вразливостей, адже саме на їх використанні базуються механізми реалізації атак. Так як одною із причин появи вразливостей є непослідовність у реалізації контролю доступу. Наявні системи містять привілейовані засоби та служби, які передають користувачам частину своїх повноважень, оминаючи засоби контролю. Один з прикладів механізм SUID/SGID у системі UNIX. Будь-яка програмна помилка в таких засобах призводить до появи вразливостей. Відтак переважну більшість причин появи вразливостей можна усунути, реалізувавши в системі контроль доступу на основі універсального інтерфейсу та єдиного механізму взаємодії. Також необхідно мінімізувати об'єм довіреного коду самих засобів захисту задля зменшення ймовір-ності появи в них помилок. Принцип коректності засоби захисту мають реалізовувати керування доступом відповідно до формальних моделей. За наявності несуперечливої моделі безпеки можна формально обґрунтувати безпеку системи та отримати об'єк-тивний критерій оцінювання коректності її роботи. На основі такої моделі можна створювати вичерпні тести, що перевіряють правильність роботи засобів захисту в усіх режимах і за будь-яких обставин [3].

1.3 Основні підсистеми комплексу засобів захисту ОС

Структура підсистем комплексу засобів захисту (КЗЗ) ОС загалом відповідає структурі типової підсистеми захисту (рисунок 1.1).

Рисунок 1.1 Підсистеми захисту ОС

Однією з функцій підсистеми захисту ОС є керування політикою безпеки. Захищена ОС має надавати інтерфейси, які дають змогу адміністраторам ефективно вирішувати завдання з підтримки адекватної політики безпеки (зокрема, інтерфейси для настроювання підсистем розмежування доступу, ідентифікації й автентифікації, аудита). Наступна функція ідентифікація й автентифікація. Жодний користувач не може розпочати роботу в середовищі захищеної ОС, не надавши системі свій ідентифікатор і не підтвердивши його справжність за допомогою додаткової інформації, що автентифікує цього користувача. Підсистема розмежування доступу безпосередньо реалізовує політику безпеки. Кожному користувачу надається доступ лише до тих захищених об'єктів, до яких цей доступ дозволено політикою безпеки [3].

Реєстрація й облік (аудит). У захищеній ОС здійснюється реєстрація всіх потенційно небезпечних подій. Підсистема аудита здійснює захист журналів реєстрації від НСД. Також ця підсистема може надавати засоби для аналізу

журналів і відстеження джерел тих чи інших подій. Криптографічні функції застосовують для захисту конфіденційності та цілісності інформації, для автентифікації й забезпечення унеможливлення відмови від авторства. Таким чином, криптографічні функції можуть бути використані як самостійні засоби захисту або як допоміжні механізми в інших засобах. Також важливо зазначити, що будь-яка сучасна ОС надає додаткові засоби для захисту цілісності даних не лише від НСД, але й від випадкових помилок, а також від аварійних ситуацій і збоїв у системі. Насамперед це стосується даних у файлових системах. Такі засоби реалізують можливості відкоту, а також автоматизацію процесу створення резервних копій і відновлення з них.

Як правило, під підсистемою антивірусного захисту розуміють сукупність програм, які надають можливість виявляти і знешкоджувати відомі шкідливі програми, які належать до вірусів («троянські коні», мережні хробаки, шпигунські програми) і до засобів здійснення атак. Слід за-значити, що без антивірусного захисту в наш час неможливо підтримувати ОС у безпечному стані, особливо якщо її встановлено на підключеному до мережі комп'ютері. Однак антивірусні засоби не входять до складу ОС, а поста-чаються окремо, що є наслідком не технічних, а економічних рішень, зокрема, антимонопольного законодавства. Також К33 ОС спирається на функції захисту, реалізовані в апаратних засобах [3].

Можна зробити висновок, що К33 ОС це сукупність великої кількості програмних модулів, частина з яких виконується у режимі ядра, а частина у режимі користувача, тобто як прикладні програми. У деяких ОС, де використовують більше рівнів привілеїв процесів (кілець захисту), будують ієрархію засобів захисту. У сучасних ОС К33 чітко виокремлюється в архітектурі системи (наприклад, у Windows), але є й такі захищені системи, в яких підсистеми і окремі компоненти К33 розпорошені по всій системі (наприклад, традиційні системи UNIX і Linux). Як правило, підсистема захисту дозволяє додавати додаткові модулі, які реалізують підсилені функції захисту за допомогою відповідних інтерфейсів.

2 АНАЛІЗ ВРАЗЛИВОСТЕЙ В ІНФОРМАЦІЙНОМУ СЕРЕДОВИЩІ

1.1 Поняття вразливості

Виникнення потенційних загроз безпеки пов'язано з наявністю вразливостей. Вразливість недолік або слабке місце, яке може бути використане для реалізації загрози [13].

Причинами виникнення вразливостей можуть бути [13]:

– помилки при проектуванні та розробці програмного (програмно-апаратного) забезпечення;

– навмисні дії по внесенню вразливостей в ході проектування і розробки програмного (програмно-апаратного) забезпечення;

– неправильні налаштування програмного забезпечення, неправомірна зміна режимів роботи пристроїв і програм;

– несанкціоновані ненавмисні дії користувачів, що призводять до виникнення вразливостей;

– збої в роботі апаратного і програмного забезпечення (викликані збоями в електроживленні, виходом з ладу апаратних елементів в результаті старіння і зниження надійності, зовнішніми впливами електромагнітних полів технічних пристроїв і ін.).

Розглянемо основну класифікацію вразливостей, яка приведена на рисунку 2.1.

Рисунок 2.1 Класифікація основних вразливостей

Як зображено на рисунку, по типу ПЗ розрізняють вразливості системного програмного забезпечення та вразливості прикладного програмного забезпечення [13]. Розглянемо їх більш детальніше.

Системне ПЗ призначено для управління роботою комп’ютера та комп’ютерних мереж, розподілу його ресурсів, підтримки діалогу з користувачами, надання їм допомоги в обслуговуванні комп’ютера. Вразливості системного програмного забезпечення необхідно розглядати з прив'язкою до архітектури побудови обчислювальних систем.

При цьому можливі вразливості [13]:

– у засобах операційної системи, призначених для управління локальними ресурсами інформаційної системи (які забезпечують виконання функцій управління процесами, пам'яттю, пристроями вводу / виводу, інтерфейсом з користувачем і т.п.), драйвери, утиліти;

– у засобах операційної системи, призначених для виконання допоміжних функцій, утиліти (архівування, дефрагментації і ін.), системних

обробляючих програмах (компіляторах, компонувальниках, відладчиках і т.п.), програмах надання користувачу додаткових послуг (спеціальних варіантів інтерфейсу, калькуляторів, ігр і т.п.), бібліотеках процедур різного призначення (бібліотеках математичних функцій, функцій вводу / виводу і т.д.);

– у засобах комунікаційної взаємодії (мережевих засобах) операційної системи.

Крім програмного забезпечення, вразливості можуть існувати і в інших аспектах системи, включаючи проектування протоколів, апаратне забезпечення, конфігурацію системи і робочі процедури. Складність сучасного програмного забезпечення є однією з основних причин існування недоліків, які можуть призвести до компрометації системи [15].

Типи недоліків програмного забезпечення, які можуть привести до вразливостей [15]:

– помилки пам’яті: переповнення буфера, помилки динамічної пам'яті (висячі покажчики, подвійні або неприпустимі значення, нульові покажчики), неініціалізовані змінні;

– вхідні помилки перевірки: введення коду або команди, SQL-ін'єкції, міжсайтовий скриптинг (XSS), обхід каталогу;

– невизначеність паралелізму (race condition) помилка проектування багатопотокової системи або додатки, при якій робота системи або додатки залежить від того, в якому порядку виконуються частини коду [16];

– привілейована плутанина: підробка запитів на міжсайтовий запит (CSRF), clickjacking (тип атак на веб-сайти. Користувач, здійснюючи клік на спеціально сформованої сторінці зловмисника, насправді потрапляє інший сайт [17]).

До прикладного програмного забезпечення відносяться прикладні програми загального користування та спеціальні прикладні програми. Прикладні програми загального користування текстові та графічні редактори, медіа-програми, системи управління базами даних, програмні платформи

загального користування для розробки програмних продуктів, засоби захисту інформації загального користування, тощо. Спеціальні прикладні програми це програми, які розробляються в інтересах вирішення конкретних прикладних задач в даній інформаційній системі (в тому числі програмні засоби захисту інформації, розроблені для конкретної інформаційної системи) [14].

Вразливості прикладного програмного забезпечення можуть бути [14]:

– функції і процедури, що відносяться до різних прикладних програм і несумісні між собою (не функціонують в одному операційному середовищі) через конфлікти, пов'язаних з розподілом ресурсів системи;

– функції, процедури, зміна певним чином параметрів яких дозволяє використовувати їх для проникнення в операційну середу інформаційної системи і виклику штатних функцій операційної системи, виконання несанкціонованого доступу без виявлення таких змін операційною системою;

– фрагменти коду програм («дірки», «люки»), введені розробником, що дозволяють обходити процедури ідентифікації, автентифікації, перевірки цілісності та ін., передбачені в операційній системі;

– відсутність необхідних засобів захисту (автентифікації, перевірки цілісності, перевірки форматів повідомлень, блокування несанкціоновано модифікованих функцій, тощо);

– помилки у програмах (в оголошенні змінних, функцій і процедур, у кодах програм), які за певних умов (наприклад, при виконанні логічних переходів) призводять до збоїв, в тому числі до збоїв функціонування засобів і систем захисту інформації, до можливості несанкціонованого доступу до інформації.

Вразливість, яка не має відповідної загрози, може не вимагати патчів, але повинна усвідомлюватися розробниками і піддаватися моніторингу на предмет змін. Слід зазначити, що невірно реалізоване або неправильно функціонуючий засіб контролю або засіб контролю, який неправильно використовується, теж може бути вразливістю. Важливо оцінювати, наскільки серйозними є

вразливості, іншими словами. Вразливість слід оцінювати, розглядаючи всі загрози, які можуть використовувати її у конкретній ситуації [8].

Розглянемо взаємодію загроз та вразливостей більш детально. Загроза може стати причиною небажаного інциденту, в результаті якого організації може бути завдано шкоди. Цей збиток може виникнути внаслідок атаки на інформацію, що належить організації і приводить до її несанкціонованого розкриття, модифікації, пошкодження, знищення, недоступності або втрати. Загрози можуть виходити від випадкових або навмисних джерел або подій. При реалізації загрози використовується одна або більше вразливостей систем, додатків або сервісів. Загрози можуть виходити як зсередини організації, так і ззовні [18].

Взаємодія загроз і вразливостей представлена на рисунку 2.2.

Рисунок 2.2 Взаємодія загроз та вразливостей

Як показано на рисунку, дія зловмисника полягає в використанні тієї або іншої вразливості, для впливу на інформаційний об'єкт та комплексний захист ПЗ, який теж може мати вразливості. А також впливу на властивості об'єкта, що

робить можливим виникнення і реалізацію загрози [8]. Слід зазначити, що зловмисник найчастіше використовує вже знайдені іншими людьми вразливості (наприклад, White Hat хакери публікують знайдені вразливості). Але деякі хакери спеціально приховують цю інформацію для власної вигоди.

Методи пошуку вразливостей [19]:

– імітація атак. Дані перевірки відносяться до механізму «зондування» і засновані на експлуатації різних дефектів в програмному забезпеченні. Деякі вразливості не виявляють себе, поки їх не чіпати. Для цього проти підозрілого сервісу або вузла запускаються реальні атаки;

– аналіз коду;

– сканування. Механізм пасивного аналізу, за допомогою якого сканер вразливостей намагається визначити наявність вразливості без фактичного підтвердження її наявності за непрямими ознаками. Цей метод є найбільш швидким і простим для реалізації.

Одним з варіантів пошуку вразливостей є аналіз коду. Безпосередньо, машинний код можуть розуміти тільки висококваліфіковані спеціалісти, більшість користуються спеціальною програмою дизассемблер програма яка «розшифровує» машинний код і переводить його в код асемблера - найбільш близького до машинного коду мови програмування. Після аналізу коди і виявлення вразливості, складається спеціальна програма (експлойт), яка експлуатує вразливість. З моменту виявлення і до випуску патча або оновлення, вразливість називається вразливістю нульового дня, якщо вона публічно відома. Програма з такою вразливістю може бути зламана в будь-який момент, і відомості про такі вразливості так само є товаром. Після закриття вразливості (випуску патча) цінність вразливості падає [20].

На рисунку 2.3 наведено життєвий цикл вразливостей [21].

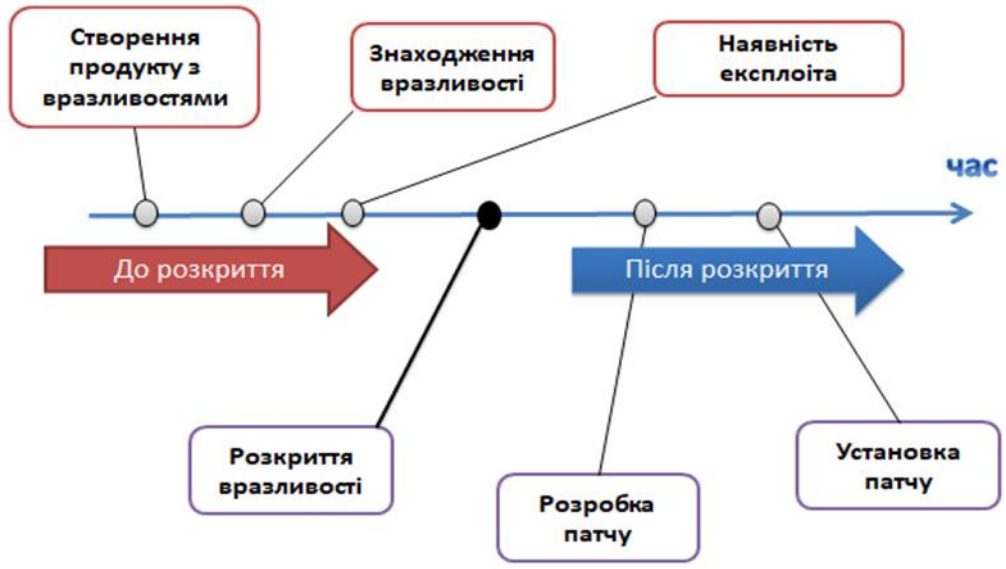

Рисунок 2.3 Життєвий цикл вразливостей

Як показано на рисунку 2.3, першим етапом є створення і випуск у реліз продукту, який може мати вразливості. Коли виявлена вразливість, розробники вивчають і оцінюють її, щоб визначити її загальний вплив на систему. Це важливий крок, оскільки неправильне визначення ризику може привести до часткового виправлення і залишити системи вразливими. Це також дозволяє пріоритезувати виправлення, щоб спочатку обробляти ці проблеми з найбільшим ризиком для клієнтів, а проблеми з низьким або мінімальним ризиком не передаються клієнтам. Іноді вивчення вразливості призводить до виявлення інших вразливостей, які потребують виправлення або реорганізації. Але поки розробники вивчають природу вразливості і розробляють для неї патч, зловмисники продовжують експлуатувати їх [21].

Далі здійснюється розробка патча. Виправлення повинно повністю усунути вразливість, не вносячи інших проблем на цьому шляху. Після того як виправлення пройшло через процеси проектування і перевірки, його відправляють клієнтам. У той же час виправлення стають доступними в базах даних вразливостей, які надають інформацію про неї [21].

Спеціалісти з Cisco стверджують, що крім забезпечення безпеки протягом всього природного життєвого циклу продуктів, розробники технологій повинні повідомляти IT спільноту про те як і чому необхідно контролювати вразливості. Однак, коли розглядається розкриття вразливості, питання про те, скільки інформації надавати і коли робити це публічно, є суперечливим питанням [22].

Деякі фахівці виступають за повне і негайне розкриття інформації, включаючи конкретну інформацію, яка може бути використана для використання вразливості; інші вважають, що інформація про вразливість не повинна публікуватися взагалі, оскільки вона може використовуватися зловмисником. Наприклад, публікація експлойта з вразливостями нульового дня відбувається, як тільки вразливість стає загальновідомою. Для зменшення ризику багато експертів вважають, що обмежена інформація повинна бути доступна для обраної групи осіб після закінчення певного періоду часу з моменту виявлення [22].

В наш час інформація щодо вразливостей продовжує зберігатися у відкритому виді. І спеціалісти розроблюють різні методи їх оцінки. Слід зазначити, що термін «управління вразливостями» часто плутають зі скануванням вразливостей. Незважаючи на те, що вони зв'язані один з одним, між ними існує суттєва відмінність. Сканування вразливостей складається з використання комп'ютерної програми для виявлення вразливостей в мережах, комп'ютерної інфраструктури або додатках [5]. Управління вразливостями це ідентифікація, оцінка, класифікація і вибір рішення для усунення вразливостей. [23].

Сканування вразливостей призначене для тестування та аналізу систем і служб для відомих вразливостей. Сканування містить список параметрів сканування (порти, протоколи та поведінкові характеристики мережевого пакету, використовувані для сканування) і активів. У звіті міститься пріоритетний список вразливостей, опис вразливості, розрахунковий ризик і заходи по відновленню. Це дуже важливо, тому що будь-яка вразливість у

операційній системі може поставити під загрозу безпеку додатків. Забезпечуючи безпеку операційної системи, середу стає більш стабільною, контролюється доступ до ресурсів і контролюється зовнішній доступ до середовища. Фізична безпека системи має важливе значення. Загрози можуть проходити через Інтернет, але вони також можуть надходити з фізичного терміналу. Навіть якщо веб-доступ дуже безпечний, якщо зловмисник отримує фізичний доступ до сервера, зламати систему набагато простіше [5].

2.2 Вразливості у прикладному програмному забезпечені

Прикладна програма, або додаток програма, призначена для виконання певних завдань і розрахована на безпосередню взаємодію з користувачем. У більшості операційних систем прикладні програми не можуть звертатися до ресурсів комп'ютера безпосередньо, а взаємодіють з обладнанням і іншими програмами за допомогою операційної системи [25].

Для організації атак часто використовуються вразливості, виявлені в веб-додатку та іншому прикладному програмному забезпеченні. Хоча існує безліч різних інструментів і методів для використання вразливостей, є кілька способів, які використовуються для атак набагато частіше, ніж інші. До них відносяться [26]:

- скрипти для сайту;
- SQL-ін'єкція;
- ін'єкція LDAP;
- підробка запитів на міжсайтовий запит.

Домінують на діаграмах (рисунок. 2.4) у 2017-м році помилки, пов'язані з інкапсуляцією, середовищем, функціями безпеки і перевіркою введення. 72% веб-додатків і 93% мобільних додатків мали принаймні одну помилку інкапсуляції, таку як недолік ескалації привілеїв, в той час як 77% веб-додатків

і 88% мобільних додатків мали недоліки в середовищі, такі як помилки конфігурації [26].

Вразливості, пов'язані з функціями безпеки, такі як проблеми перевірки достовірності та контролю доступу та помилки шифрування, були присутні в 90% веб-додатків і в 99% мобільних додатків. Фактично, вразливості, пов'язані з функціями безпеки, домінували в усіх інших категоріях в таких як веб-додатки, так і у мобільних додатках. Ця тенденція суперечлива, враховуючи, що такі функції, як автентифікація і шифрування, повинні зробити програми більш сильними, а не слабкими [26].

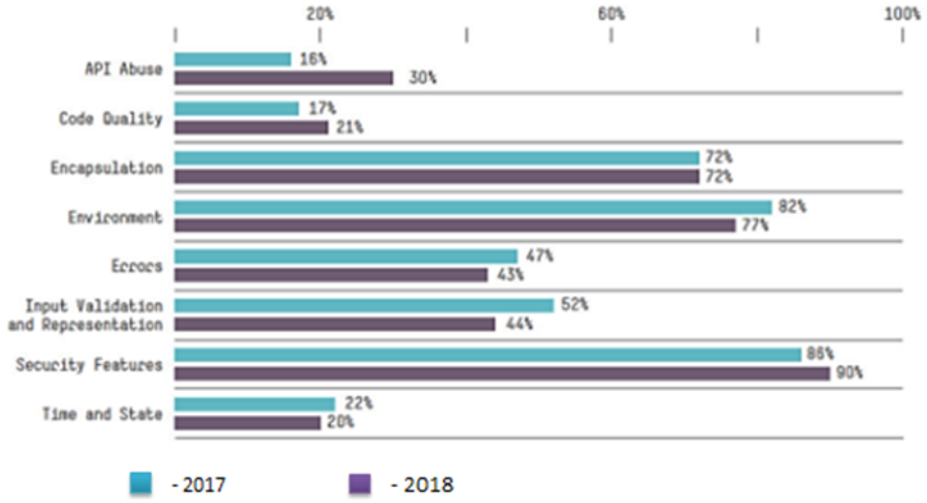

Рисунок 2.4 Порівняння помилок у ПЗ у 2017 та 2018 рр.

У списку десяти найбільш поширених критичних вразливостей у додатках були слабкі протоколи SSL, проблеми з міжсайтовими сценаріями і помилки нульового покажчика (рисунок 2.5).

В середньому мобільний пристрій підключається до більш ніж 100 різних IP-адрес протягом дня, і велика частина інформації, що надходить в і/з пристрою, незашифрована (SMS, електронна пошта, тощо). Тому мобільні

додатки схильні до того, що їх вразливості будуть під більшою увагою зловмисників.

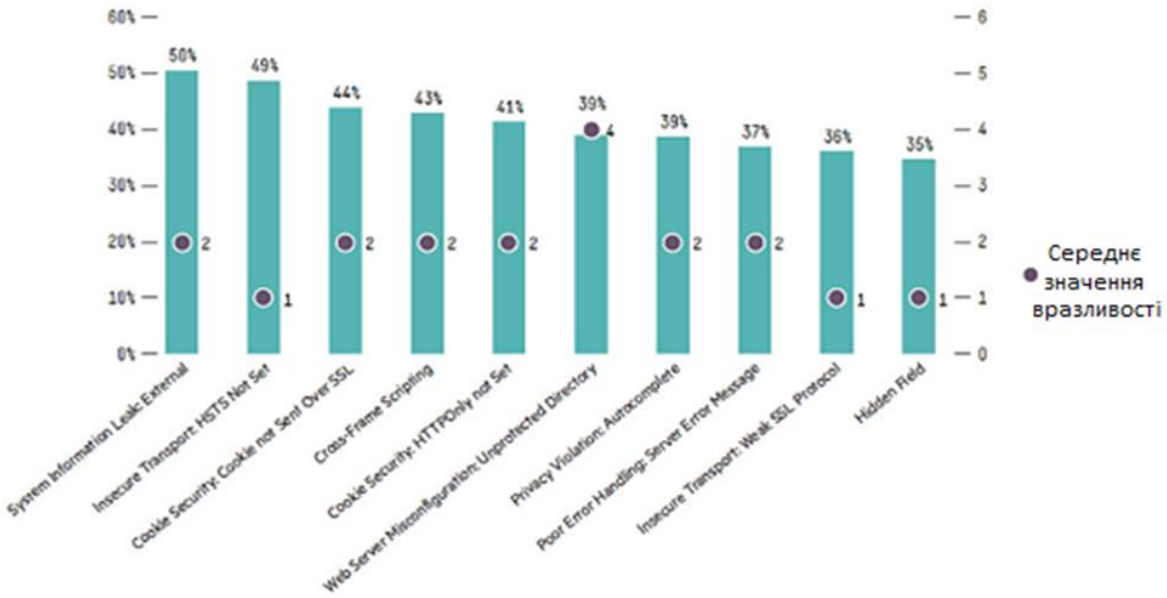

Рисунок 2.5 10 найбільш поширених критичних вразливостей у ПЗ

В середньому мобільний пристрій підключається до більш ніж 100 різних IP-адрес протягом дня, і велика частина інформації, що надходить в і/з пристрою, незашифрована (SMS, електронна пошта, тощо). Тому мобільні додатки схильні до того, що їх вразливості будуть під більшою увагою зловмисників. Командою білих хакерів був сформований топ десяти вразливостей мобільних та веб-додатків та рішень як захистити свій пристрій. Огляд деяких пунктів представлений нижче [27]:

– бінарні захист: Недостатній Jailbreak / Виявлення Root. Корекція або джейлбрейкінг пристрої обходять захист даних і схеми шифрування в системі. Коли пристрій було скомпрометовано, на пристрої може запускатися будь-який вид шкідливого коду, який може значно змінити передбачувану поведінку логіки додатка. Інструменти відновлення і даних, як правило, працюють на кореневих пристроях;

– недостатній захист транспортного рівня. Додатки часто не можуть шифрувати мережевий трафік, коли необхідно захищати конфіденційний

зв'язок. Шифрування (зазвичай TLS) має використовуватися для всіх автентифікованих підключень, особливо для веб-сторінок, доступних через Інтернет. Також необхідно зашифрувати з'єднання з бекендом або піддати ризику аутентифікацію або токен для зловмисників в тій же мережі, що й хост додатка. Ці бекенд з'єднання можуть представляти меншу ймовірність експлуатації, ніж підключення до зовнішнього Інтернету; проте їх вплив у разі експлуатації може як і раніше приводити до компрометації облікових записів користувачів або ще гірше. Шифрування слід використовувати, коли передаються конфіденційні дані, такі як кредитна карта або інформація про здоров'я. Додатки, які повертаються у відкритий текст або іншим чином витісняються з режиму шифрування, можуть бути атаковані зловмисниками;

– інформаційний витік версія сервера: інформація про сервер присутній у відповіді. Інформаційна витік слабке додаток, коли додаток виявляє конфіденційні дані, такі як технічні дані веб-додатки, середовища або призначених для користувача даних. Чутливі дані можуть використовуватися зловмисником для використання цільового програми, його хостингової мережі або її користувачів; витік конфіденційних даних повинна бути обмежена або відвернена по можливості;

– недостатня авторизація/аутентифікація: недостатньо способів авторизації, коли програма не виконує відповідні перевірки повноважень, щоб гарантувати, що користувач виконує функцію або отримує доступ до даних у відповідності з політикою безпеки. Процедури авторизації повинні забезпечувати дотримання прав користувача, послуги або програми. Коли користувач аутентифіцируваний на веб-сайті, це не обов'язково означає, що користувач повинен мати повний доступ до всього контенту і функцій;

– криптографія неправильна перевірка сертифіката. Ця програма або не перевіряє сертифікати SSL/TLS, або використовує систему перевірки сертифіката SSL/TLS, яка не буде правильно перевіряти, що довірений постачальник видав сертифікат. Клієнт повинен бути налаштований на

видалення з'єднання, якщо сертифікат не може бути перевірений або не вказано. Будь-які дані, якими обмінюються по з'єднанню, де сертифікат не правильно перевірений, можуть бути піддані несанкціонованого доступу або зміни;

– Brute Force перебір користувачів: існує безліч способів для зловмисника визначити, чи існує користувач в системі; атака грубої сили це метод визначення невідомого значення за допомогою автоматизованого процесу, щоб спробувати велику кількість можливих значень. Атака використовує той факт, що ентропія значень менше, ніж сприймається. Наприклад, хоча 8-значний буквено-цифровий пароль може мати 2,8 трильйона можливих значень, багато людей будуть вибирати свої паролі з набагато меншого підмножини, що складається із загальних слів і термінів. Якщо повідомлення про помилки змінюються при неправильному уявленні імені користувача та/або пароль, зловмисник може визначити наявність дійсного імені користувача/адреси електронної пошти на основі будь-яких відмінностей в повідомленнях про помилки. Якщо ідентифікатор користувача генерується послідовно передбачуваним чином (XXX102017, XXX112017 і т. д.), Зловмисник може перерахувати список користувачів шляхом збільшення ідентифікатора користувача;

– недостатнє завершення сеансу: після того, як користувач вийде з програми, ідентифікатори, які використовувалися під час сеансу, повинні бути визнані недійсними. Якщо сервер не може анулювати ідентифікатори сеансу, інші користувачі можуть використовувати ці ідентифікатори для уособлення цього користувача і виконання дій від його імені;

– інформаційна витік з кеш додатків. Чутливі дані можуть вийти за боковий вівтар кеша додатків або через основний код програми, або через сторонні структури. Мобільні пристрої являють собою унікальну проблему щодо безпечного зберігання даних. Пристрої можуть бути легко втрачені або вкрадені. Багато користувачів не блокують свої пристрої. Кешировані дані

можуть бути переглянуті зловмисником, який виконує криміналізацію даних на фізичному пристрої.

2.3 Вразливості в операційних системах

Вразливості можна виявити як в прикладному програмному забезпеченні, так і в операційних системах [28]. Хоча процес тестування може ізолювати більшу частину дефектів, їх неможливо повністю усунути.

Операційна система це програмне забезпечення, яке складається з програм і даних, які керують роботою апаратної частини пристроїв і забезпечують роботу користувача і системних додатків. Основними компонентами операційної системи є: ядро (основна частина, що керує апаратними засобами і виконанням програм), файлова структура (система зберігання файлів на запам'ятовуючих пристроях), інтерпретатор команд користувача (відповідає за взаємодію користувача з комп'ютером) і утиліти (програми, які виконують службові функції) [29].

Існують різні операційні системи. Вони відрізняються своїми функціональними можливостями і сферою застосування [29].

Вразливості операційної системи вважаються особливим випадком дефектів програмного забезпечення [30]. Їх використання може скомпрометувати всі процеси і служби, запущені в операційній системі, і дозволити зловмисникам отримати доступ до всіх даних, що зберігаються на вразливому комп'ютері. Вони представляють загрозу безпеці та надійності системи [28].

Наприклад, вразливість CVE-2016-7256 відноситься до сімейств операційних систем Windows. Суть проблеми бібліотека шрифтів неправильно обробляє спеціально створені вбудовані шрифти OpenType. Зловмисник може успішно використовувати цю вразливість за допомогою сценаріїв атаки на основі Інтернету або обміну файлами. В результаті можна повністю контролювати вразливу систему, дозволяючи хакерам встановлювати програми,

переглядати, змінювати або видаляти дані, створювати нові облікові записи користувачів з правами адміністратора і т. д. Іншим прикладом є вразливість CVE-2014-0160 в бібліотеці криптографії OpenSSL, яка дозволяє віддаленим зловмисникам отримувати конфіденційну інформацію з пам'яті процесу і навіть скомпрометувати секретний ключ сервера через створені пакети, які викликають переповнення буфера. Це торкнулося півмільйона веб-сайтів і сервісів, включаючи Yahoo, Amazon Web Services, GitHub, Wikipedia, тощо [28].

Були проаналізовані наступні ОС:

- Ubuntu сервер Ubuntu 12.04;
- Red Hat Red Hat Enterprise Linux 6;
- Novell Novell Linux Enterprise Server 11 SP2;
- Windows Microsoft Windows Server 2012 R2;
- MacOS Apple MacOS Server 10.8;
- Solaris Oracle Solaris 11.

Основною причиною для вибору конкретних ОС і їх була популярність. Незважаючи на те, що обрані ОС тепер замінені більш новими версіями, аналіз [28] показує, що значна кількість нових вразливостей як і раніше розкривається в обраних версіях ОС. На жаль, більшість цих вразливостей існує і в самих останніх операційних системах.

Наприклад, 100% вразливостей Apple MacOS Server, зареєстрованих базою даних NVD протягом 2016 року, були виявлені у версіях 10.8 і 10.11. Останній був випущений 21.03.2016. Відсоткова частка вразливостей, поширених у 2012 і 2016 роках (випущених 26.09.2016 р) версіями Microsoft Windows Server, 6.x і 7.x (випущена 10.06.2014, остання оновлена версія 01.08.2017) версій Red Hat Enterprise Linux і 12.4 і 16.4 (випущені 21.04.2016) версії Ubuntu Server відповідно 85%, 75% і 70%. Також варто відзначити, що Oracle Solaris 11.3 розділяє близько 30 відсотків вразливостей, виявлених протягом 2016 року, з Oracle Solaris 10.0, але не з версією 11.0, проаналізованої

у документі. Це можна пояснити, якщо припустити, що значна частина коду останньої версії 11.3 була повторно використана з 10-ї версії замість 11-го [31].

На рисунку 2.6 показана вище подана статистика вразливостей поширених у цих ОС у проміжку 2012 і 2018 рр.

Red Hat Enterprise Linux 6 і Oracle Solaris 11 були випущені до спостережуваного періоду (2010 та 2012 рр.). Інші операційні системи (Ubuntu Server 12.04, Novell Linux Enterprise Server 11 SP2, Microsoft Windows Server 2012 R2 і Apple Macintosh Server 10.8) були випущені на початку 2012 року. Слід зазначити, що на дату офіційного випуску деякі з них вже мали вразливості, які раніше були виявлені в попередніх версіях ОС. Зокрема, Ubuntu Server 12.04 успадкував 15 таких вразливостей, Red Hat Enterprise Linux 6 - 46, Novell Linux Enterprise Server 11 SP2 - 26 і вразливості Oracle Solaris 11 – 13 [28].

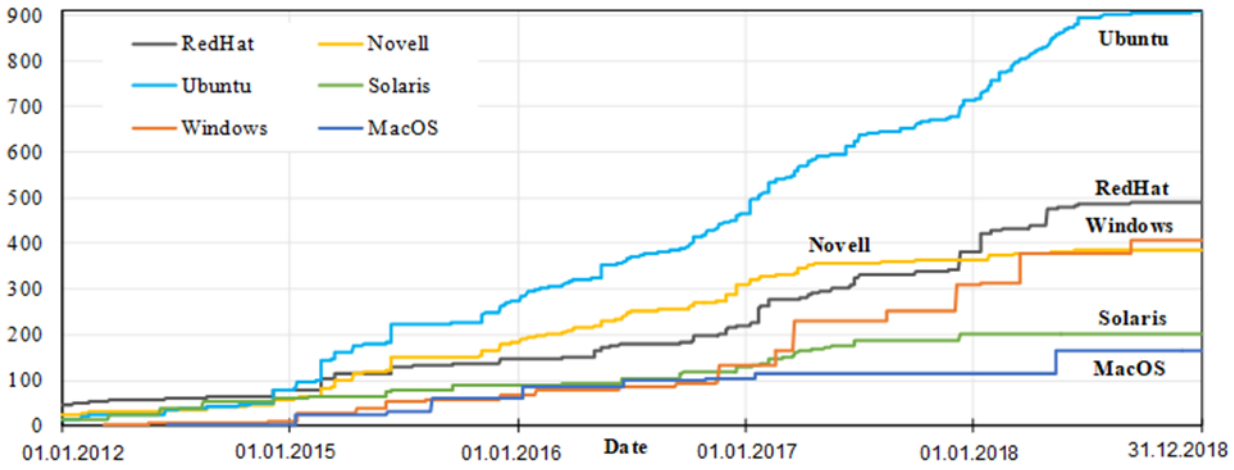

Рисунок 2.6 Статистика вразливостей у ОС

Значне зростання загальної кількості вразливостей, виявлених в сучасних операційних системах, а також загальна тенденція до збільшення їх критичності демонструють серйозні проблеми безпеки і ризики, з якими стикаються розробники ОС і користувачі. Причиною такої тенденції скоріш за все є розширення використання цих ОС у корпоративному середовищі і, як наслідок, підвищення уваги до них з боку дослідників і кіберзлочинців [32].

2.4 Модель загроз

Вразливості системного програмного забезпечення необхідно розглядати з прив'язкою до архітектури побудови обчислювальних систем.

При цьому можливі уразливості [15]:

- в засобах операційної системи, призначених для управління локальними ресурсами (які забезпечують виконання функцій управління процесами, пам'яттю, пристроями введення / виводу, інтерфейсом з користувачем і т.п.), драйвери, утиліти;
- в засобах операційної системи, призначених для виконання допоміжних функцій, – утиліти (архівування, дефрагментації і ін.), системних обробних програмах (компіляторах, компонувальниках, відладчиках і т.п.), програмах надання користувачу додаткових послуг (спеціальних варіантах інтерфейсу, калькуляторах, іграх і т.п.), бібліотеках процедур різного призначення (бібліотеках математичних функцій, функцій введення / виводу і т.д.);
- у мікропрограмах;
- в засобах комунікаційної взаємодії (мережевих засобах) операційної системи.

Загрози, що спрямовані на ОС є [3]:

сканування файлової системи;

викрадення ключової інформації;

добирання паролів;

збирання сміття;

перевищення повноважень;

тощо.

Сканування файлової системи – це насамперед атака на політику безпеки. Потрібно, щоб усі користувачі в системі мали доступ лише до тих файлів, до

яких він дозволений згідно з політикою безпеки. Якщо порушник після сканування системи отримує доступ до інших файлів, він порушує політику безпеки, тобто здійснює несанкціонований доступ (НСД). Порушник отримує можливість несанкціоновано ознайомитися з інформацією (порушення конфіденційності) або модифікувати чи видаляти її (порушення цілісності).

Прикладом ключової інформації може бути пароль, електронно-цифровий підпис, тощо, які можуть зберігатися на технічному приладі: текстові документи, файли, пристрої флеш-пам'яті, токени. В деяких випадках у вигляді носія інформації може розглядатися людина. Втрата інформацією своєї цінності (порушення безпеки інформації) може статися внаслідок переміщення інформації або зміни фізичних властивостей носія [1].

На відміну від викрадення ключової інформації, добирання паролів здійснюють за допомогою засобів автентифікації для того, щоб знайти пароль, який сприйматиметься як правильний. У результаті порушник отримує несанкціонований доступ до певного ресурсу, якщо пароль використовують для обмеження доступу до цього ресурсу, або можливість працювати в системі від імені іншого користувача, якщо пароль використовують для автентифікації.

Збирання сміття – це отримання даних, які залишаються в об'єктах, що операційна система звільняє після їх використання. Найтиповішими з таких об'єктів є файли на дисках і ділянки оперативної пам'яті.

Загроза перевищення повноважень полягає у тому, що порушник якимось чином отримує повноваження, що перевищують ті, які йому було надано згідно з політикою безпеки. Перевищення повноважень можливе або через помилки, яких припустилися під час розроблення і реалізації політики безпеки (наприклад, некоректні настроювання системи розмежування доступу), або через наявність вразливостей в програмному забезпеченні, яке входить до складу ОС [3].

Якщо описані вище загрози не присутні у ОС, то цю ОС можна вважати захищеною.

3 АНАЛІЗ БД ВРАЗЛИВОСТЕЙ

3.1 Огляд і аналіз БД вразливостей

База даних вразливостей це база даних, призначена для збору, зберігання і поширення інформації про виявлені вразливості, призначених для комп'ютерних систем. База даних, зазвичай, описує виявлену вразливість, може оцінювати потенційний збиток для комп'ютерних систем і обхідний шлях, необхідний для уникнення атаки [26].

Існує багато баз даних вразливостей. Після того, як CERT (the Computer Emergency Response Team) публічно видав у 1989 році інформацію про вразливості, всі ці вразливості з'явилися в багатьох нових базах даних, опублікувавши їх у іншому інформаційному форматі. У декількох базах даних існує багато типів вразливостей, але у них вона по різному представлена. Наприклад, у базі даних MS є інформація про вразливості у продуктах Microsoft, BugTraq, яка запитує рішення про вразливість, як її вирішити, NVD має рівень оцінки вразливості, OSVDB велика БД, що містить більше звітів про вразливість [33].

Перелік найвідоміших БД [33]:

– OSVDB: метою БД є надання точної та об'єктивної інформації про вразливість безпеки в комп'ютеризованому обладнанні. У блозі OSVDB обговорюються різні теми, пов'язані з вразливостями, включаючи розкриття інформації, запуск бази даних вразливостей (VDB), тощо [34]. Але ця БД повідомляє недостатнє статистичної інформації ;

– Security Focus Bugtraq: містить докладну інформацію про вразливості, якщо інформацію про експлойт (якщо він є). Але вона містить заплутану систему пошуку вразливостей;

– CVE: база даних загальновідомих вразливостей інформаційної безпеки. Кожній вразливості присвоюється ідентифікаційний номер, опис і ряд загальнодоступних посилань з описом. Підтримкою CVE займається організація MITRE;

– NVD: підтримка з боку уряду США і спільна робота з БД CVE;

– VULDB: велика база даних вразливостей, с додатковим функціоналом у вигляди статистики та оглядом існуючих експлоітів;

– NIPC: китайська БД вразливостей. Не має перекладу на інші мови;

– Microsoft Security bulletin: містить інформацію тільки про вразливості Microsoft продуктів;

– Exploit Database: база даних містить інформацію про вразливості які вже застосовуються для атак. У своїй роботі я не використовую цю базу даних, оскільки вона містить інформацію про вразливості, тільки для яких є експлойти;

– MFSA: розміщена інформація про вразливості в продуктах Mozilla. У своєму дослідженні я не розглядую цю базу даних, оскільки її вразливості призначені тільки для продуктів Mozilla і не охоплюють всі Сервіси.

Кожна база даних має свій префікс. Наприклад: Security Focus префікса Bugtraq BID, Microsoft Security Bulletin має префікс MS, Common Vulnerabilities і Exposures - CVE. Ці джерела для збору інформації про вразливість і публікації її мають різні правила і керівництва, і для цього вони мають різну форматову інформацію. Для зв'язку з інформацією про вразливість у двох або більше джерелах вони реєструють вразливість в базі даних CVE. Приклад розшифровка CVE номеру: BID це префікс, підтримуваний Security Focus Bugtraq (приклад: BID 8829) і інформація про вразливість. У деяких випадках дані про вразливість реєструються у різних базах даних іншим ідентифікатором, таким чином, для пошуку інформації про усі дірки у безпеці з інших джерел, використовуючи ідентифікатор CVE у якості первинного ключа для зв'язку різних баз даних вразливостей. Наприклад: CVE-2003-1523 в інших

джерелах, зареєстрован різними ідентифікаторами SECUNIA: 10001 (SA10001), BID: 8829 (Bugtraq ID: 8829), XF: dbmailmultiple-sql-injection (13416) [35].

На основі переліку, представленого вище, далі проаналізовано три основні бази даних, які збирають більше 90000 вразливостей [26]. У цьому аналізі порівняно бази даних за кількістю вразливостей і впровадження CVE, щоб знайти кращі бази. З цих обраних баз даних вразливостей була отримана інформація про вразливості і застосована до моєї розробки.

Перша БД Common Vulnerabilities and Exposures (CVE). CVE працює з 1999 року і є відкритою БД. Кожна вразливість має унікальний номер - це унікальні загальні ідентифікатори для загальновідомих вразливостей у кібербезпеки. Ідентифікатори CVE призначені для використання в відношенні виявлення вразливостей [35].

CVE це словник загальновживаних імен (наприклад, ідентифікатори CVE) для загальнодоступних вразливостей інформаційної безпеки. Загальні ідентифікатори CVE полегшують обмін даними між окремими базами даних та інструментами мережевої безпеки і забезпечують основу для оцінки охоплення інструментів безпеки організації. Якщо в звіті з одного з ваших коштів безпеки містяться ідентифікатори CVE, ви можете швидко і точно отримати доступ до інформації про виправлення в одній або декількох окремих CVE-сумісних базах даних, щоб усунути проблему [35].

Для завантаження і роботи з базою даних CVE представляє дані в багатьох різних форматах у вигляді XML, HTML, CSV. Файл CVE з форматуванням HTML CVE містить ідентифікатор CVE і інформацію про вразливості з 1999 року до останньої дати завантаження [35].

БД CVE включає в себе такі поля [36]:

– опис: це стандартизоване текстовий опис проблеми у вигляді одного загального запису. Це означає, що номер запису вразливості зарезервований компанією Mitre для пошуку проблеми;

– посилання: це список URL-адресов та іншої інформації (наприклад, рекомендаційні номера постачальників) для цієї проблеми;

– створення запису. Для CVE, її призначає безпосередньо Miter, це дата, в якій Mitre створила запис CVE. Для CVE, призначених CNA (наприклад, Microsoft, Oracle, HP, Red Hat і т. Д.), це також дата, яку запис створила Mitre, а не CNA. Таким чином, в разі, коли CNA запитує блок номерів CVE заздалегідь, дата введення буде тоді, коли це CVE призначається CNA. Сам CVE не може використовуватися протягом кількох днів, тижнів, місяців або навіть років (наприклад, Red Hat підтримує блоки CVE для старіших проблем безпеки в програмному забезпеченні з відкритим вихідним кодом, яким ще не призначені CVE);

CVE намагається призначити один CVE для кожної проблеми безпеки, однак у багатьох випадках це призведе до надзвичайно великої кількості CVE (наприклад, якщо в додатку PHP виявлено кілька десятків вразливостей для міжсайтового скриптинга). Щоб впоратися з цим, існують керівні принципи, які охоплюють розділення і злиття питань з окремими номерами CVE. Спочатку розглядуються питання, які необхідно об'єднати, потім слід розділити проблеми на тип вразливості (наприклад, переповнення буфера або переповнення стека), а потім на версію програмного забезпечення (наприклад, якщо одна проблема зачіпає версію 1.3.4 через 2.5. 4, а інші з 1.3.4 до 2.5.8, їх будуть розділяти), а потім об’єднують проблеми [30].

Варто відзначити, що статистику по кількості нових вразливастей у БД CVE можна подивитися тільки у вигляді статей. У БД CVE не має функції такої, тому неможливо слідкувати за новою інформацією у любий момент тільки за допомогою сайту.

Друга БД U.S. National Vulnerability Database (NVD). Її підтримує уряд США і вона співпрацює з базою даних уразливості CVE. Також NVD має загальну систему оцінки вразливостей (CVSS), яка обчислює значення від 0,0 до 10,0 у залежності від рівня безпеки уразливості. NVD використовує формат

XML, у результаті представлений XML-файл, упорядкований за датою уразливості. nvdcve-2.0- [YEAR] .xml 2002 <= [РІК] <= Поточний рік. Де «Поточний рік» – це змінна, яка збільшується щороку [38].

У таблиці 3.1 представлена кількісна характеристика вразливостей за останні три роки.

Таблиця 3.1 – Вразливості з 2015-2018 рр.

Рік	Кількість за в рік	Відсоток від усіх
2015	6,487	66.38%
2016	6,447	100.00%
2017	14,647	100.00%
2018	6,566	100.00%

На рисунку 3.1 проілюстровані показники кількості вразливостей, подані БД NVD.

Рисунок 3.1 Показники кількості вразливостей у NVD

На травень база даних NVD налічує 101180 вразливостей [38]. Значне зростання кількості вразливостей по сьогоднішній день зображене на рисунку 3.2.

Рисунок 3.2 – Загальна кількість вразливостей за даними сайту NVD

Третя БД VULDB. Ця БД почала документування вразливостей з 1970 року. Крім технічних подробиць є додаткова інформація про загрозу безпеки застосування вразливостей, наприклад, поточні рівні ризику. Раніш ця БД мала назву Bugbase та була зорієнтована на німецькомовних спеціалістів. Кожен запис містив не тільки загальне резюме проблеми, але також конкретний аналіз залежностей і можливого впливу. Це було щось нове в області баз даних вразливостей. І було високо оцінено IT-адміністраторами та розробниками. Тому у 2013 році БД була переведена на англійську мову. Розробники додали функції пошуку та статистики. У базу даних була додана додаткова підтримка

мов, таких як французька, іспанська, італійська та польська. Завдяки цій інтернаціоналізації, прийняття в галузі інформаційної безпеки стало набагато швидше. За минулі роки підтримка відкритих стандартів стала дуже важливою. Ось чому всі записи VULDB підтримують CVSSv3, CVE, CWE, CPE, OVAL і IAVM. У 2016 році була також реалізована унікальна особливість прогнозу експлойтів, яка допомагає користувачам оцінювати ступінь вразливості [39].

VulDB безкоштовна база даних, але додаткові комерційні послуги роблять БД привабливою для великих корпоративних клієнтів. Додаткова інформація, індивідуальний статистичний аналіз і поглиблений технічний огляд експлойтів - це лише деякі з можливостей. Користувачі можуть створювати обліковий запис і використовувати комерційні послуги. Або приєднайтеся до видання спільноти, яка дозволяє редагувати і переглядати записи. Багато дослідників і адміністратори вразливостей використовують ці функції для внесення змін в існуючі записи або пропонують нові матеріали для додавання в базу даних. Також БД має огляд різних призначень ризиків різних джерел документованих вразливостей (рисунок. 3.3) [39].

Year	Low	Medium	High	Total
2018	39%	59.2%	1.8%	*7053*
2017	40.2%	58%	1.8%	*16291*
2016	41.2%	55.3%	3.5%	*8322*

Рисунок 3.3 – Огляд критичності ризиків у БД VulDB

Група модераторів шукає інформацію про нові або існуючі вразливості. Якщо буде визначена нова проблема, будуть зібрані додаткові дані з інших джерел і буде створений новий запис VulDB. Потім цей запис передається клієнтам, на веб-сайт і також стає доступним через API і облікові записи в соціальних мережах [39].

На основі аналізу представлених вище баз даних вразливостей, буде побудована таблиця 3.2 та графік за кількістю вразливостей по кожній з них – рисунок. 3.4

Таблиця 3.2 – Порівняння БД вразливостей

Назва	Синтаксис	Кількість вразливостей	Додаткові функції	Статистика
CVE	Префікс CVE + Рік + Довільні цифри	101 338	Корисні посилання	-
NVD	Префікс CVE + Рік + Довільні цифри	101 180	Загальна система оцінки вразливостей	+
VULDB	Дата+критичнsсть вразливості+короткий опис+[Префікс CVE]	116 973	Інфомація про експоіти	+

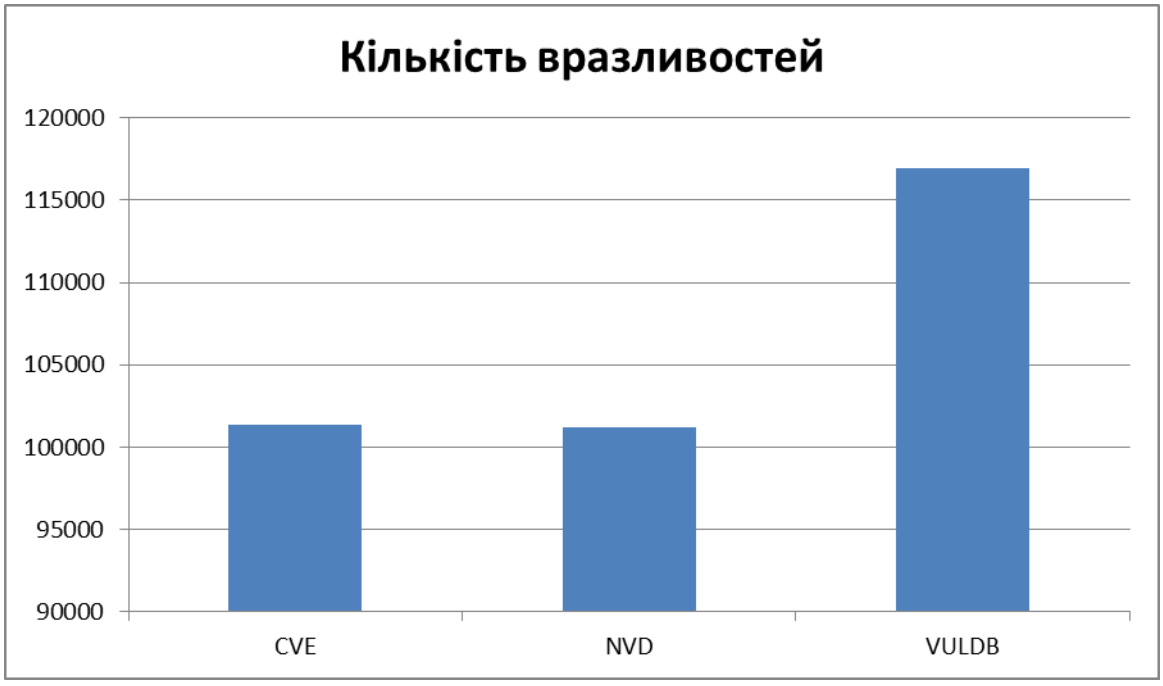

Рисунок 3.4 – Порівняння БД за кількістю вразливостей

Проаналізувавши ці бази даних, хочеться відзначити, що БД CVE має тільки оглядовий характер опису вразливостей. Для аналізу вразливостей на критичність, цієї інформації дуже мало. БД VULDB, як показано на рисунку 2.6 займає перше місце за кількістю вразливостей, але вони не проходять через аналіз спеціалістів, а просто публікуються користувачами. У декількох джерелах, написано, що спеціалісти сумніваються у достовірності інформації щодо вразливостей. Наприклад, цитата з статті компанії Login Plugin Vulnerabilities: «Попередження про вразливість, яка не існує, не корисна, особливо якщо публікують інформацію, що вразливість знаходиться в поточній версії плагіна, що часто буває. Приклад такої ситуації з плагіном WP Markdown Editor. БД VulDB включає цю вразливість в своїх даних» [40].

Що стосується БД NVD, як було вже написано вище, вона співпрацює з БД CVE. Як тільки з’являється нова вразливість і публікується у БД CVE, експерти аналізують її і публікують розгорнутий опис вразливості, включаючи вразливі продукти, та види можливих атак. Тому в цій БД менша кількість вразливостей ніж у двох інших приведених БД, але її інформації можна довіряти. Крім того, на сайті є можливість завантажити БД на свій пристрій у різних форматах. Тому в якості довірливого і зручного джерела була обрана база даних NVD.

3.2 Методи оцінки критичності вразливостей

Термінологія, яка використовується в галузі аналізу загроз, ризиків і вразливостей, не є однорідною і багато позначень мають більше одного визначення або використовуються різними шляхами. Це пов'язано з особливостями використання за допомогою існуючих моделей і методів. Деякі моделі і методи не відрізняють різницю між загрозою і ризиком і використовують ці поняття як взаємозамінні, наприклад, модель Swedish Rescue Services Agency [41].

Існує ряд інших систем оцінки вразливостей, які створені комерційними та некомерційними організаціями. Ось деякі з них:

- CERT/CC;
- Система аналізу вразливостей SANS;
- Система оцінки від Microsoft;
- CVSS.

Кожна з них має свої переваги, але всі вони відрізняються по тому, яка буде ознака вимірювання. Наприклад, CERT / CC використовує значення оцінок від 0 до 180 і враховує такі фактори, як наприклад, чи схильна Інтернет-інфраструктура до ризику і який тип передумов потрібен для експлуатації вразливості. Система аналізу вразливостей SANS враховує, в якій конфігурації знайдена вразливість - стандартної чи ні, чи є система клієнтом або сервером. Система оцінки від Microsoft намагається відобразити складність експлуатації та загальний вплив від експлуатації вразливості. Ці системи оцінки є корисними, але вони представляють підхід, при якому вважається, що наслідки експлуатації вразливості однакові для приватної особи і для компанії [42].

SANS при аналізі вразливостей (SANS Critical Vulnerability Analysis) присвоює критичний рівень тим вразливостям, для яких існує загальнодоступна програма, яка використовує ці вразливості, або використання яких не потребує спеціальних навичок. В іншому випадку навіть потенційно дуже небезпечна проблема буде мати високий, а не критичний рівень ризику. Крім простоти експлуатації при оцінці вразливості за методикою SANS враховується поширеність вразливих систем [43].

Група PSS компанії Microsoft застосовує методику оцінки ризику пов'язаного з шкідливим програмним забезпеченням (мається на увазі з атаками, які використовують вразливість), що враховує наявність оновлення, що усуває помилку, кількість векторів, які може застосовувати атакуючий, поширеність вразливих систем. Наприклад, «критично небезпечний черв'як» повинен поширюватися через діру в безпеці у програмному забезпеченні

Microsoft, для якої відсутня оновлення, використовуючи два і більше вектора атаки в широко поширених системах [43].

Методика, використовувана CERT, розрахунок ступеня ризику вразливості в залежності від наведених нижче критеріїв [43]:

- наскільки доступна інформація про вразливість?
- чи зареєстровані випадки використання вразливості?
- піддаються небезпеки критичні для мережі Internet-вузли?
- яка кількість вузлів мережі вразлива?
- які наслідки використання вразливості?
- наскільки легко скористатися вразливістю?
- які умови використання вразливості?

На жаль, зв'язок між умовами, їх можливими вагами і результуючої ступенем ризику формально не визначена, що залишає великий простір для розбіжностей в оцінках однієї і тієї ж вразливості. Крім того, перераховані методики дають значення ризику для Internet в цілому, а не для конкретної інформаційної системи або корпоративної мережі [43].

Загальна система оцінки вразливостей (CVSS) це відкрита схема, яка дозволяє обмінюватися інформацією про IT-вразливості. Система оцінки CVSS складається з трьох метрик: базова метрика, тимчасова метрика і контекстна метрика. Кожна метрика являє собою число (оцінку) в інтервалі від 0 до 10 і вектор - короткий текстовий опис зі значеннями, які використовуються для виведення оцінки. Базова метрика відображає основні характеристики вразливості. Тимчасова метрика відповідає таким характеристикам вразливості, які змінюються з часом, а контекстна метрика характеристикам, які є унікальними для середовища користувача. CVSS є зрозумілим, прозорим і загальноприйнятим способом оцінки IT-вразливостей для керівників, виробників додатків і засобів підтримки інформаційної безпеки, дослідників та ін. [42].

Кількісна оцінка безпеки має різні області. Залежно від того, як ми розглядаємо системи, застосовна метрика може відрізнятися. Потрібно визначити метрики, які є практичними, корисними і значущими.

Резюмуючи сказане, можна сформулювати вимоги до методики оцінки вразливості в такий спосіб:

– має бути присутня можливість оцінки ступеня ризику вразливості в залежності від можливості її експлуатації;

– результатом застосування методики має бути числове значення, що підходить для використання при аналізі ризиків;

– методика повинна мати можливість адаптації до конкретної інформаційної системи;

– параметри, які використовуються при розрахунку, повинні допускати мінімум різночитань;

– механізм розрахунку результуючого значення має бути простий і зрозумілий.

3.3 Порівняльна характеристика програмних засобів на ринку

Незважаючи на зростаючий інтерес до галузі захисту інформації, програм для оцінки вразливостей не так багато. Незважаючи на це, завдання вибору оптимального продукту подібного класу є непростою, оскільки при аналізі потрібно враховувати багато факторів.

Крім того, існує велика кількість сканерів вразливостей. Сканери вразливостей - це програмні або апаратні засоби, що служать для здійснення діагностики та моніторингу мережевих комп'ютерів, що дозволяють сканувати мережі, комп'ютери та програми на предмет виявлення можливих проблем у системі безпеки, оцінювати і усувати вразливості [44].

Сканери вразливостей дозволяють перевірити різні додатки в системі на предмет наявності «дірок», якими можуть скористатися зловмисники. Також

можуть бути використані низькорівневі засоби, такі як сканер портів, для виявлення та аналізу можливих додатків і протоколів, що виконуються в системі [44].

Більшість сучасних сканерів безпеки мережі працює за принципами [44]:

– збір інформації про мережу, ідентифікація всіх активних пристроїв і сервісів, запущених на них;

– виявлення потенційних вразливостей;

– підтвердження обраних вразливостей, для чого використовуються специфічні методи і моделюються атаки;

– автоматичне усунення вразливостей. Не завжди цей етап реалізується в мережевих сканерах безпеки, але часто зустрічається в сканерах системних.

Таким чином, сканери спрямовані на вирішення наступних завдань [44]:

– ідентифікація та аналіз вразливостей;

– інвентаризація ресурсів, таких як операційна система, програмне забезпечення та пристрої мережі;

– формування звітів, що містять опис вразливостей і варіанти їх усунення.

Але під «аналізом вразливостей» мається на увазі механізм, який запускає імітації атак, тим самим перевіряючи вразливість. При зондуванні застосовуються методи реалізації атак, які допомагають підтвердити наявність проблем [44]. Таким чином, за допомогою сканерів ми аналізуємо вразливості, які має наш пристрій. Застосовуючи сканери вразливостей, ми не зможемо виявити проблеми, наприклад, Linux Kernel, якщо ми маємо ОС Windows (не беручи до уваги застосування віртуальної машини).

Під час аналізу ринку програм, які могли б допомогти аналізувати рівень критичності вразливостей, були знайдені дві утіліти, які приблизно б могли ініціювати оцінку критичності вразливостей

Класифікація програмного забезпечення для оцінки критичності вразливостей:

- утиліта PT Exploit Explorer;
- калькулятор CVSS V2;
- Nessus Vulnerability Scanner.

Безкоштовна утиліта PT Exploit Explorer (PT EE) допомогає спеціалістам з безпеки звернути увагу на вразливості, для використання яких були створені та опубліковані спеціальні хакерські програми. Наявність публічного експлойта значно підвищує ймовірність інциденту: подібний хакерський інструмент дозволяє будь-якому недосвідченому зломщикові скористатися вразливістю і автоматизувати атаку навіть просто з хуліганських спонукань. Програма PT Exploit Explorer дозволяє шукати посилання на експлойти в загальнодоступних базах даних, включаючи Rapid7 і exploit-db. Утиліта повністю сумісна со сканером вразливостей XSpider і системою контролю захищеності та відповідності стандартам MaxPatrol, а також зі звітами інших систем виявлення вразливостей в будь-яких незжатих форматах. Крім того, в програмі доступна обробка текстових файлів, що містять довільний список вразливостей з бази CVE [45]. Скріншот роботи утіліти представлено на рисунок. 3.5

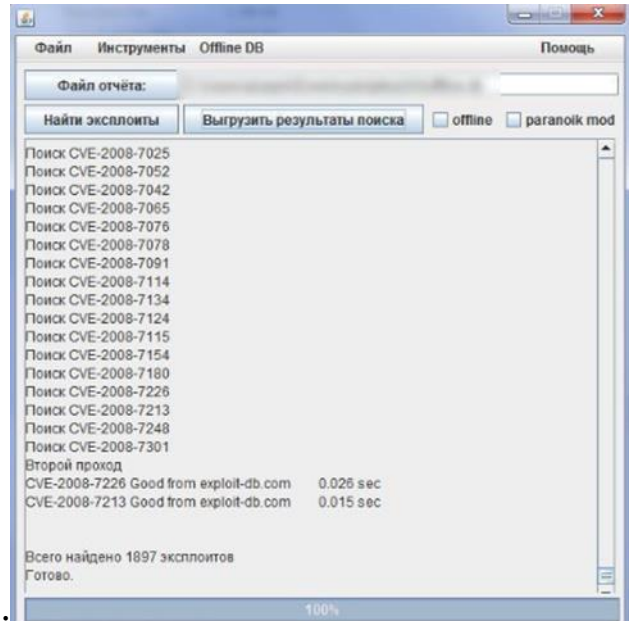

Рисунок 3.5 Скріншот роботи утіліти

Калькулятор CVSS V2 побудований на підставі методики загальної системи оцінки вразливостей (CVSS). В якості базових метрик він набуває таких значень [46]:

– спосіб отримання доступу: локальний, мережний, суміжна мережа;

– складність отримання доступу: висока, середня, низька;

– автентифікація: множинна, єдина, не потрібно;

– вплив на конфіденційність: не робить, часткове, повне;

– вплив на цілісність: не робить, часткове, повне;

– вплив на доступність: не робить, часткове, повне.

В якості тимчасових метрик калькулятор приймає наступні значення [46]:

– можливість використання: не визначено, теоретично, є концепція, є сценарій, висока;

– рівень виправлення: не визначено, офіційне, тимчасове, рекомендації, недоступно;

– ступінь достовірності джерела: не визначено, не підтверджена, не доведена, підтверджена.

В якості контекстних метрик калькулятор приймає такі значення [46]:

– можливість нанесення непрямих збитків: не визначено, відсутнє, низька, середня, підвищена, висока;

– щільність мети: не визначено, відсутнє, низька, середня, висока;

– вимоги до конфіденційності: не визначено, низька, середня, висока;

– вимоги до цілісності: не визначено, низька, середня, висока;

– вимоги до доступності: не визначено, низька, середня, висока.

Чисельне значення базового вектора вразливості (базова оцінка) змінюється від 0 до 10. На основі чисельного значення базового вектора V вразливості (базової оцінки) присвоюються один з чотирьох рівнів небезпеки [46]:

– низький рівень небезпеки, якщо $0{,}0 \leq V \leq 3{,}9$;

– середній рівень небезпеки, якщо $4{,}0 \leq V \leq 6{,}9$;

– високий рівень небезпеки, якщо 7,0 ≤ V ≤ 9,9;
– критичний рівень небезпеки, якщо V = 10,0.

Скріншот роботи калькулятора наведений на рисунку 3.6

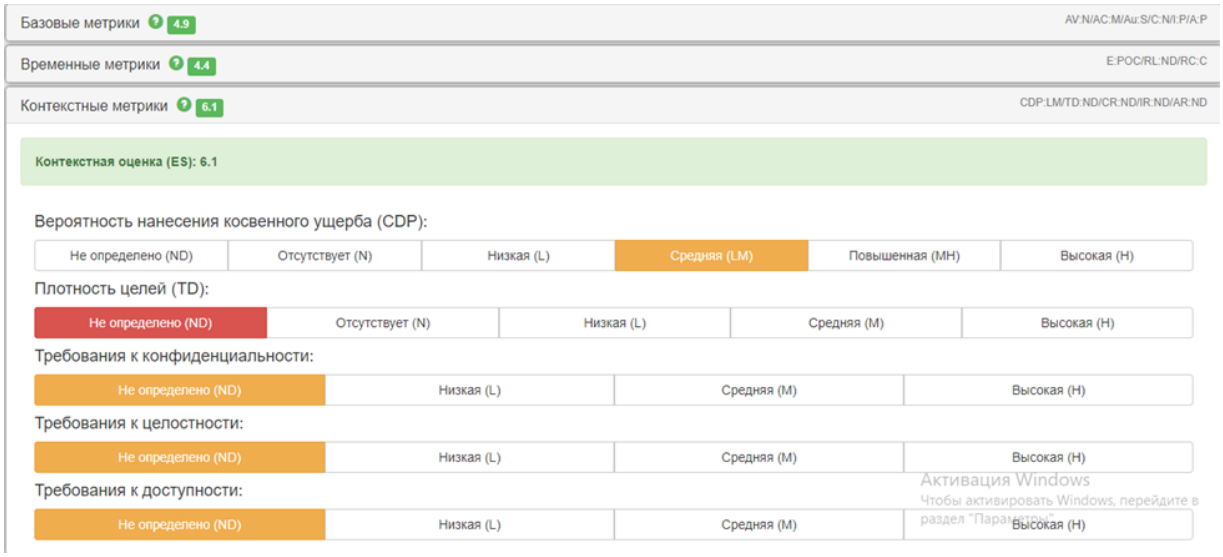

Рисунок 3.6 Скріншот роботи калькулятора CVSS V2

Одним з найбільш популярних сканерів вразливостей на ринку є Nessus Vulnerability Scanner. Він став свого роду стандартом для сканерів вразливостей. Спочатку він стартував як проект з відкритим кодом. Далі його придбала компанія Tenable, і тепер він є комерційним продуктом (версія Professional). Незважаючи на це, у Nessus Scanner як і раніше є «Home» версія, яка розповсюджується безкоштовно, але має обмеження у 16 IP адрес. Далі буде розглянута саме ця версія [47].

Nessus що складається з серверної і клієнтської частин. Клієнтська програма існує як для Unix систем, так і для win32, у той час як сервер може бути запущений тільки у системі Unix. Спілкування між клієнтом і сервером відбувається за власним протоколу Nessus NTP, який працює поверх TCP. Сканер у змозі одночасно досліджувати декілька систем. Досліджуючи систему, сканер опитує всі порти заданого діапазону на наявність мережевих сервісів. Визначивши доступний сервіс, сканер починає аналізувати його стійкість до злому, імітуючи задані у модулях типи атак [47].

За результатами сканування ми отримуємо список з IP адресами та пов'язані з ними ризики. Ризики мають колірне кодування. Приклад роботи програми наведено на рисунку 3.7.

У верхньому меню програми, можна відобразити всі вразливості, виявлені у мережі. При виборі конкретної вразливості, можна отримати більш детальну інформацію. Важливо відзначити, що крім опису вразливості, у звіті присутній також і спосіб її виправлення. Як показано на рисунку 3.7, Nessus оцінює знайдені вразливості по наступним категоріям: Info, Low, Medium, Hight. Інформує про їх кількість та будує діаграму для кращої візуалізації.

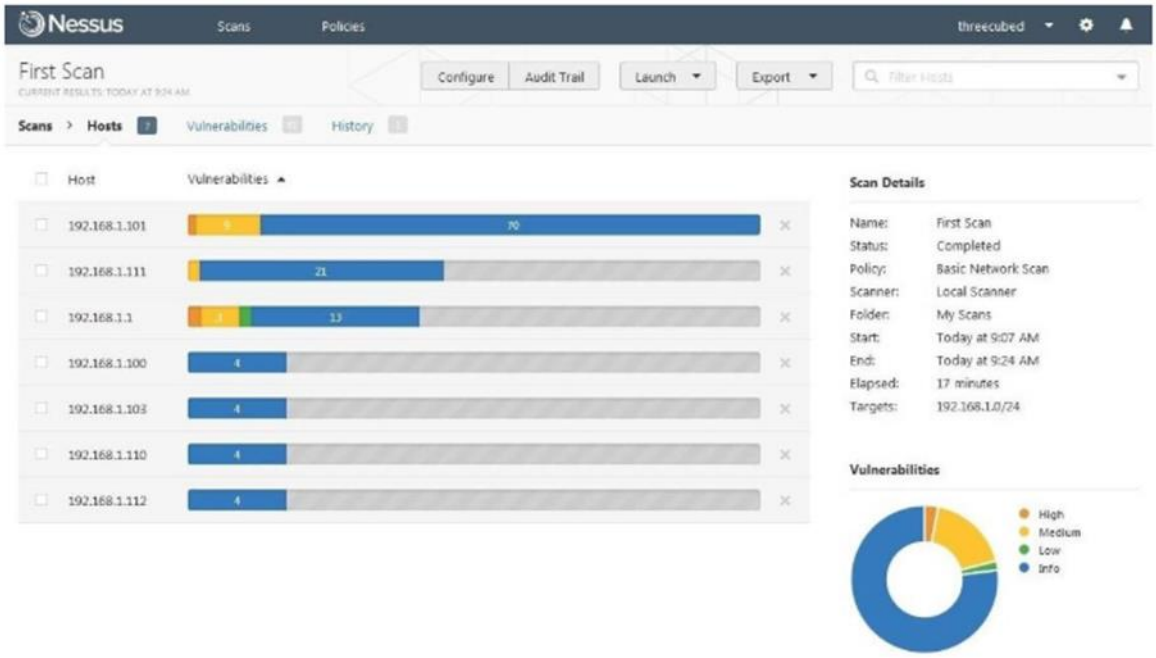

Рисунок 3.7 - Приклад роботи Nessus Vulnerability Scanner

На підставі наведеної інформації була побудована порівняльна таблиця 3.3

Таблиця 3.3 - Порівняльна характеристика програм оцінки вразливостей

Назва	Плата	Плюси	Мінуси
PT Exploit Explorer	+	-можливість завантажувати текстові файли; -має корисні посилання на експлоіти;	-джерело БД CVE, яка дає мало інформації для оцінки; -застосування тільки для вразливостей для яких є експлоіти; -неявно описано, як здійснюється оцінка критичності вразливостей.

Продовження Таблиці 3.3

Калькулятор CVSS V2	+	-детальний огляд метрик	-заплутані критерії відбору; -незрозуміла логіка оцінювання.
Nessus	-/+ (обмежена free версія)	-містить актуальні моделі загроз, на можливість експлуатації яких він швидко і якісно перевіряє клієнтську мережу.	-демонструє критичність вразливостей тільки знайдений при скануванні.

Проаналізувавши ці програми, можна зробити висновок, що на даний момент не існує програмного забезпечення, яке могло б:

– аналізувати вразливості без допоміжних втручань користувача;

– аналізувати не тільки вразливості для яких існують експлоіти;

– демонструвати вибірку вразливостей для кількох ОС у зручному форматі;

– проводити інтеграцію з БД вразливостей NVD.

Така експертна характеристика вразливостей може у майбутньому використовуватися користувачами для оцінки показників, які можуть визначати розподіл ресурсів для тестування безпеки, планування і розробки патчів безпеки. Крім того, вона може використовуватися кінцевими користувачами як допоміжний додаток з метою оцінки ризиків та оцінки необхідної надмірності в ресурсах і процедурах для усунення потенційних порушень безпеки. Мета полягає в тому, щоб бути впевненим, що модель, яку ми можемо мати, буде дозволяти розробникам і користувачам краще оцінювати безпеку операційних систем. З огляду на критичність помилок безпеки, способи експлуатації яких знаходяться у відкритому доступі, утиліта буде корисна спеціалістам інформаційної безпеки в оперативному управлінні IT-ризиками – наприклад, для розстановки пріоритетів при інсталяції оновлень.

4 МЕТРИКИ ОЦІНКИ ВРАЗЛИВОСТЕЙ

4.1 Вибірка метрик для оцінки критичності вразливостей

На підставі аналізу методів у розділі 3.2 «Методи оцінки критичності вразливостей» були вибрані наступні метрики впливу вразливостей, що дозволяють зловмисно:

- впливати на конфіденційність (є влив/ не визначено);
- впливати на цілісність (є влив/ не визначено);
- впливати на доступність (є влив/ не визначено);
- дії направлені на пошкодження пам'яті (використовується/ не визначено);
- підбір паролів (використовується/ не визначено);
- спричинення до відмови в обслуговуванні (є влив/ не визначено);
- несанкціонована зміна прав доступу (використовується/ не визначено);
- злонамірне виконання довільного коду (використовується/ не визначено).

Також перевірка на:

- необхідність автентифікації для реалізації атаки (легко/ не визначено);
- складність експлуатації вразливості (легко/ не визначено);
- локальна або зовнішня взаємодія (локальна/зовнішня/ не визначено).

Вибрані метрики, на відміну калькулятору CVSS, є більш конкретизовані. Наприклад, при аналізу впливу на доступність у CVSS оцінюються наступні варіанти: не вливає, цілком, частково. Це суперечне припущення, бо не зрозуміло, як можна частково здійснити вплив на доступність, бо якщо зловмисник має вплив то вже можлива злонамірна дія, не зважаючи частковим чи повним був вплив. Також не використовувався параметр «щільність цілей зловмисника», тому що з опису вразливостей, це зрозуміти неможливо.

Також ці метрики були вибрані на підставі аналізу опису вразливостей у ОС Windows за останні три місяці у БД вразливостей NVD.

Параметри, які використовуються при розрахунку, повинні допускати мінімум різночитань, тому із можливих метрик були вибрані саме ці параметри.

4.2 Обгрунтування вибраних метрик та порівняння оцінок

Щоб уникнути великої розбіжності результатів, була вибрана шкала від 0 до 10. Та для зручності користувача умовні позначення Low, Medium, High та Critical як у інших відомих методиках [37]. Пояснення умовних позначень:

- low низький рівень. Вразливість має незначний вплив на систему. Наприклад, вразливість, що дозволяє отримати деяку аналітику;
- medium помірний рівень. Вразливості можуть мати № вплив на систему, але мають складність реалізації або не серйозні наслідки. Наприклад, вразливість, що дозволяє управляти гостевім доступом у систему, тим саме не маючи влив на основні її компоненти;
- high значний рівень. Вразливості, які мають істотний вплив на систему. Наприклад, спроможність зловмиснику виконувати довільний код;
- critical критичний рівень. Вразливості, наслідок експлуатації яких має серйозний вплив на систему. Наприклад, отримання зловмисником повного доступу до системи на привілейованому рівні.

Порівняння ранжування у різних компаніях продемонстровано у таблиці 4.1 [37].

Для засобу, що розробляється, будуть використовуватися наступні умовні позначення: Low - 0.1-3.9, Medium - 4.0-6.9, High - 7.0-9.4, Critical - 9.5 - 10.0.

Для кожного параметру вибраних метрик на основі моєї експертної оцінки та порівняння показників у калькуляторі CVSS [40] були виставлені коефіцієнти (таблиця 4.2-4.5).

Таблиця 4.1 Підходи до виставлення якісного рівня небезпеки

Назва	Low	Medium	High	Critical
Nvd.nist.gov	0-3.9	4.0-6.9	7.0-10.0	-
Tenable (Nessus)	0-3.9	4.0-6.9	7.0-9.9	10.0
Rapid 7	0.1-3.9	4.0-7.9	8.0-10.0	-
CVSSv3	0.1-3.9	4.0-6.9	7.0-8.9	9.0 - 10.0

Таблиця 4.2 Вагові коефіцієнти

Метрика	Має вплив	Не визначено
Впливає на конфіденційність	1	0,4
Впливає на цілісність	1,5	0,4
Впливає на доступність	2	0,4
Дії направлені на пошкодження пам'яті	2	0,1
Підбір паролів	1,5	0,1
Спричинення до відмови в обслуговуванні	2	0,2
Несанкціонована зміна прав доступу	2	0,2
Злонамірне виконання довільного коду	1,5	0,5

Таблиця 4.3 – Вагові коефіцієнти

Метрика	Легко	Не визначено
Складність експлуатації вразливості	1	0,2

Таблиця 4.4 - Вагові коефіцієнти

Метрика	Присутня	Не визначено
Необхідність автентифікації для реалізації атаки	1	0,2

Таблиця 4.5 - Вагові коефіцієнти

Метрика	Локальна	Зовнішня	Не визначено
Тип взаємодії	1	1	0,2

Для порівняння у таблиці 4.6 представлені розрахунки, які надає БД NVD та розрахунки на підставі вибраних метрик та вагових коефіцієнтів (у таблиці позначається як «*».

Таблиця 4.6 – Порівняння оцінок

Vuln ID	CVSSv3	Nvd.nist.gov	*
CVE-2018-4127	8,8	6,8	8,1
CVE-2017-7172	7,8	9,3	9
CVE-2017-2493	6,5	4,3	4,5
CVE-2018-6251	7,2	7,8	7,5
CVE-2018-1234	5,5	2,1	4,3
CVE-2018-0929	4,3	4,3	3
CVE-2018-0886	7	7,6	7,3
CVE-2018-0888	5,7	4,7	4
CVE-2018-6016	7,8	4,6	6,3
CVE-2018-10190	7,8	7,2	8,1
CVE-2018-1000117	6,7	7,2	6,3
CVE-2018-10204	8,8	9	7,6

На підставі таблиці був побудован графік розбіжності результатів (рисунок. 4.1). Виходячи с того, що (за результатами таблиці 4.6) відмінність між показниками CVSSv3 та Nvd.nist.gov становить 12,6%, а відмінність між CVSSv3 та «*» - 10,1 % та між Nvd.nist.gov та «*» - лише 2,5, можна зробити висновок, що параметри та коефіцієнти для оцінки були вибрані оптимально.

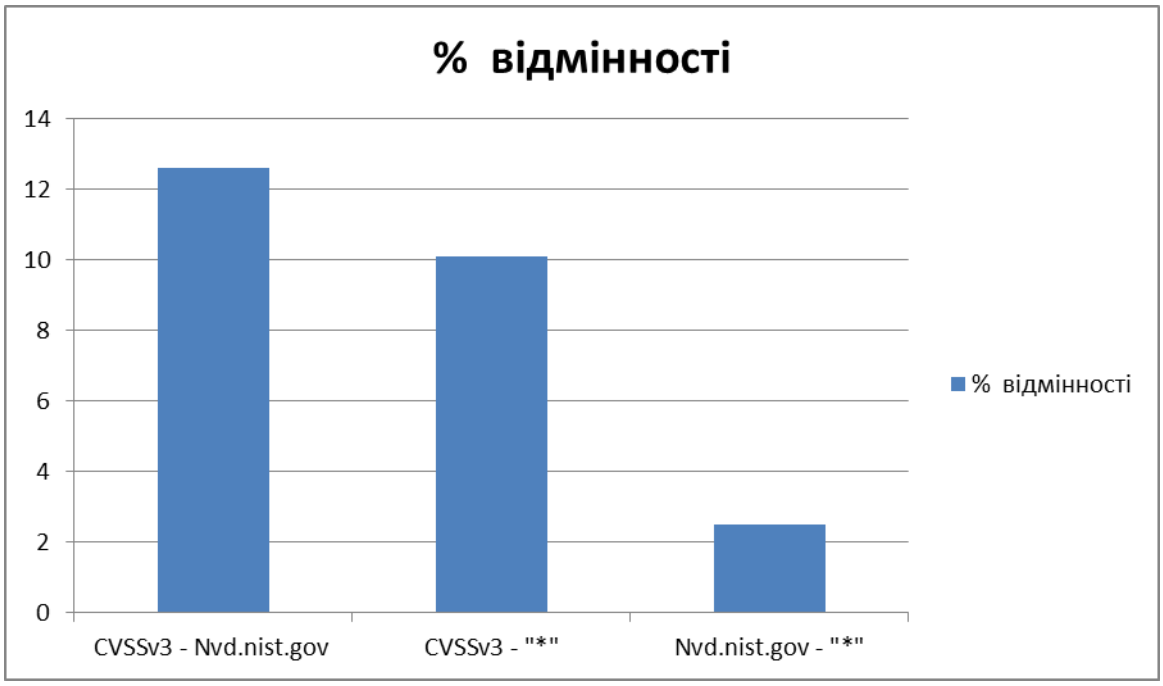

Рисунок 4.1 Графік розбіжності результатів

5 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

5.1 Загальні відомості

Основним призначенням розробленої програми є оцінка критичності вразливостей ОС. Оцінюються операційні системи: Windows, iOS, RedHat, Ubuntu, Novell, Solaris, MacOS, Android.

5.2 Вибір мови програмування

Для вирішення поставленого завдання необхідна мова програмування, яка має відкриті бібліотеки і є об'єктно-орієнтованою. Нетехнічний критерій – наявність досвіду роботи з мовою. Під вище перелічені критерії потрапляють мови програмування C# і C++. Найбільш краще підходить мова програмування C#, так як відповідає представленим критеріям, в свою чергу мова програмування C ++ не відповідає висунутому нетехнічному критерію.

Для програмування на мові C # необхідне середовище розробки з наявністю відладчика IntelliSense і аналізатора коду. Під ці вимоги відповідає VisualStudio Community 2017 з огляду на те, що розробка програмного забезпечення буде розроблятися під управлінням операційної системи Windows 10.

5.3 Розробка алгоритму

Робота алгоритма роботи програмного засобу для оцінки рівня критичності вразливостей ОС представлена на рисунках 5.1–5.3.

На рисунку 5.1 зображений перший етап роботи користувача з програмним забезпеченням. Користувач здійснює вибір параметрів для пошуку:

знайти всі вразливості або для заданої операційної системи та кількість знайдених вразливості на сторінці результату.

На рисунку 5.2 зображений другий етап взаємодії користувача з програмою – пошук вразливості за її унікальним ідентифікатором – CVE кодом. На кінці цього етапу користувач отримує детальну інформацію про вразливість та список продуктів у яких вона є. На рисунку 5.3 останній етап роботи програми –оцінка критичності вразливості.

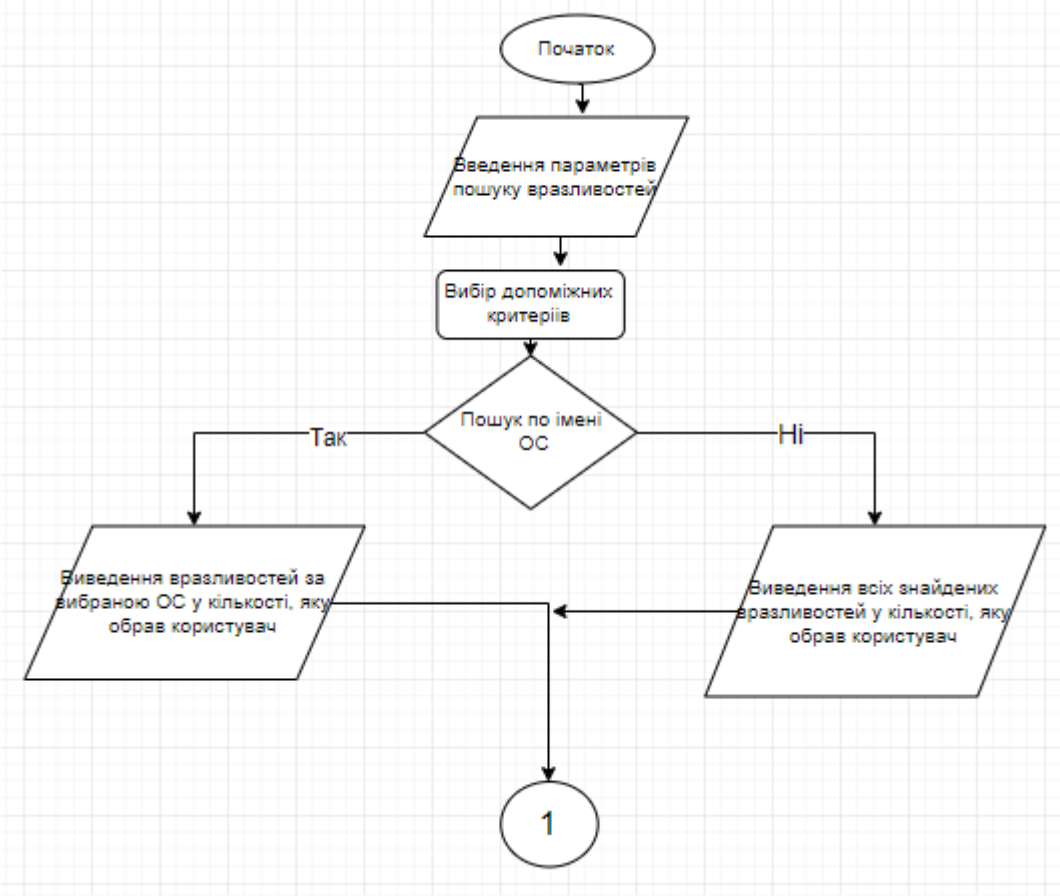

Рисунок 5.1 – Алгоритм першого етапу роботи програми

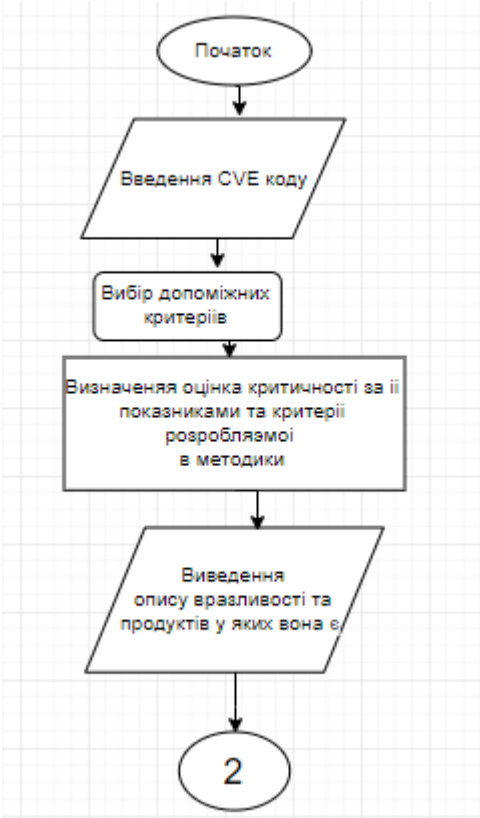

Рисунок 5.2 – Алгоритм другого етапу роботи програми

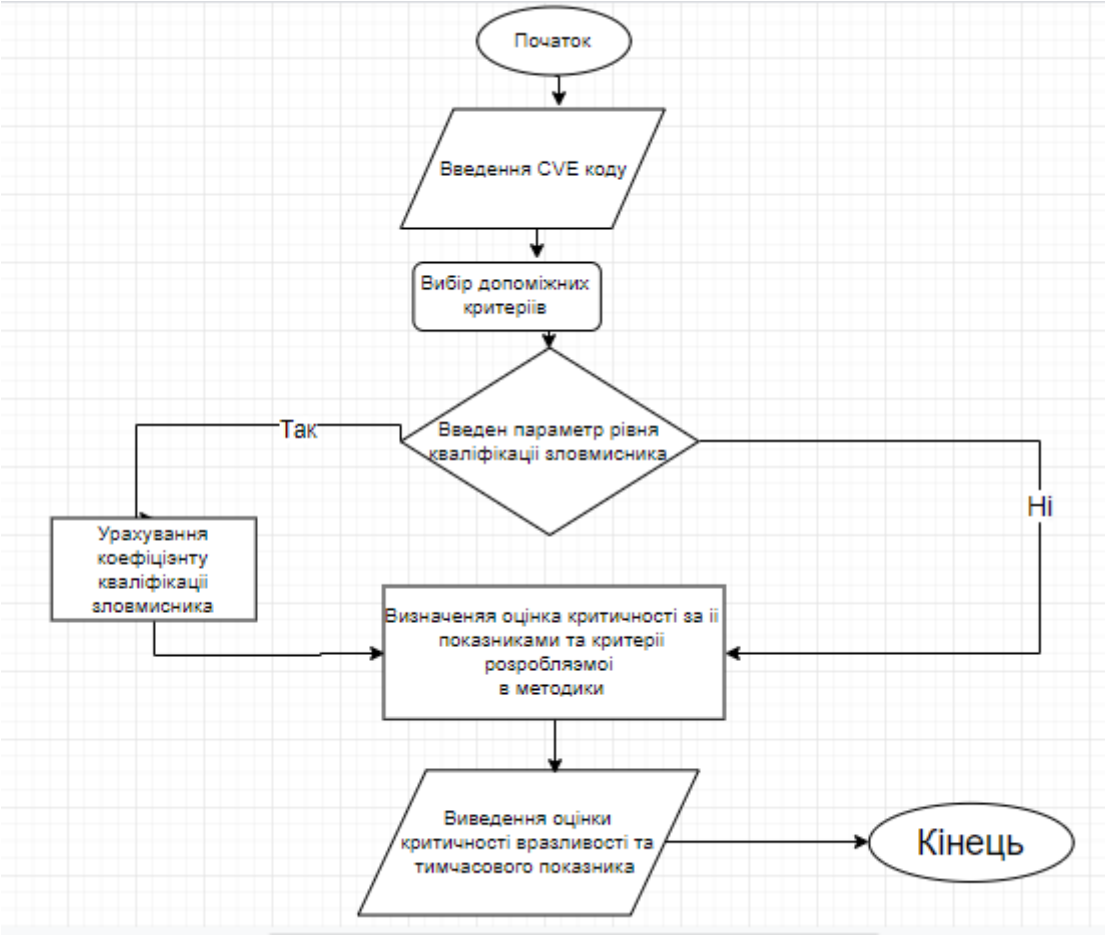

Рисунок 5.3 – Алгоритм останнього етапу роботи програми

5.4 Опис роботи програми

Програма представляє собою application programming interface (API). На рисунку 5.4 представлен зовнішній вигляд за допомогою Swagger.

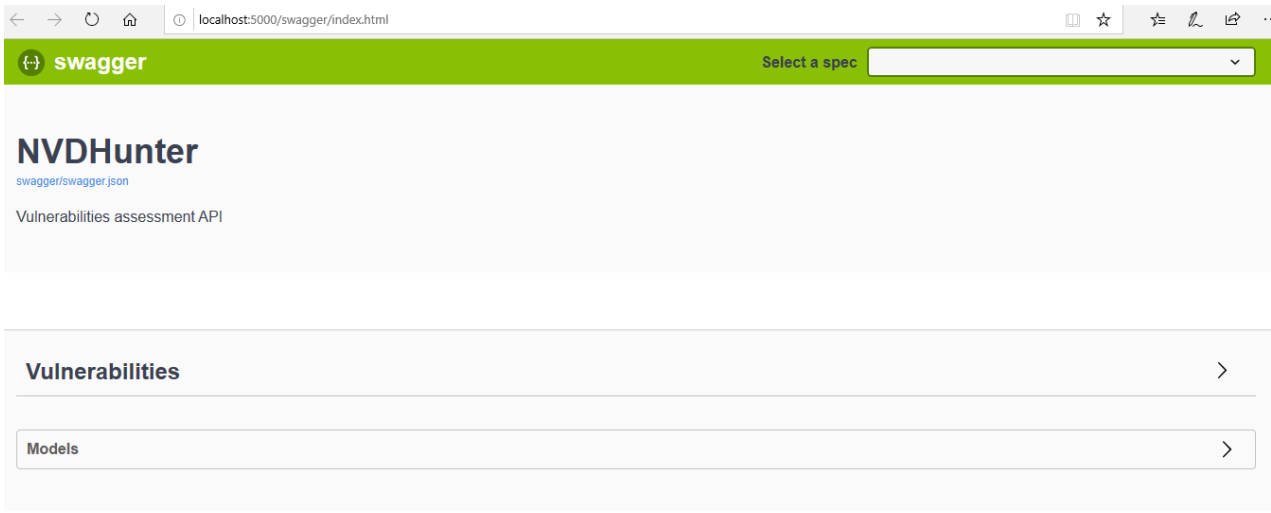

Рисунок 5.4 – Зовнішній вигляд

Далі на рисунку 5.5 представлен перелік ендпоінтів (endpoint – точка прийому запиту на стороні сервера).

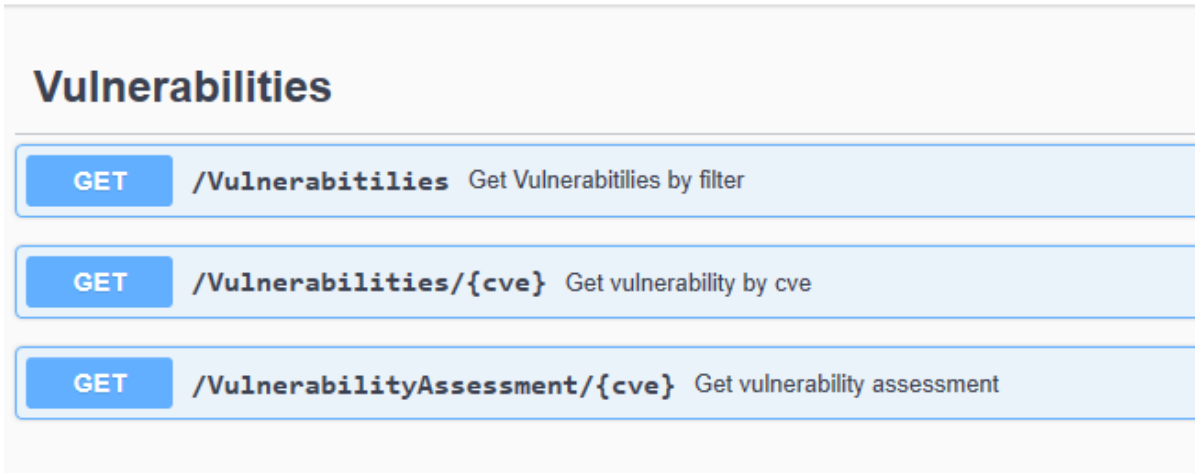

Рисунок 5.5 – Доступні ендпоінти

Перший ендпоінт «/Vulnerabilities» дає змогу користувачу отримати список вразливостей за параметрами, які ввів користувач. Приклад роботи ендпоінта представлений на рисунках 5.6-5.7.

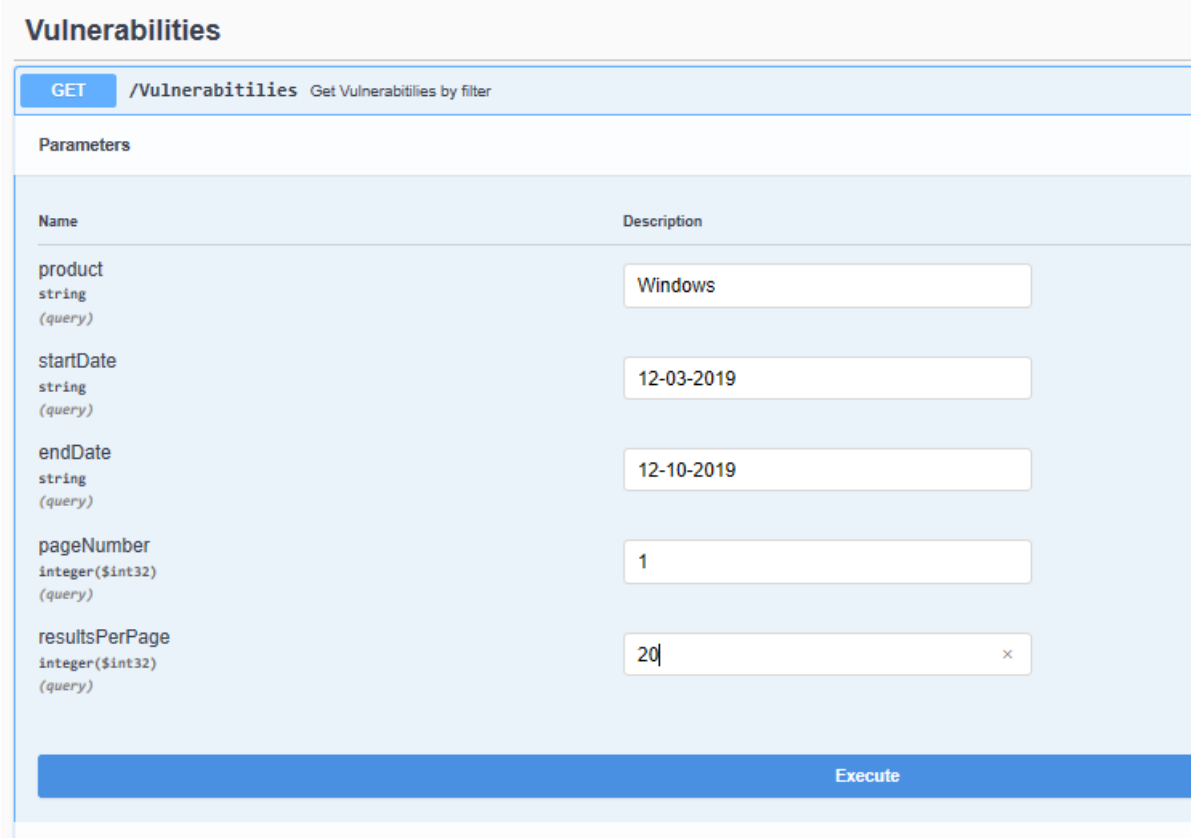

Рисунок 5.6 – Приклад вхідних даних для ендпоінта «/Vulnerabilities»

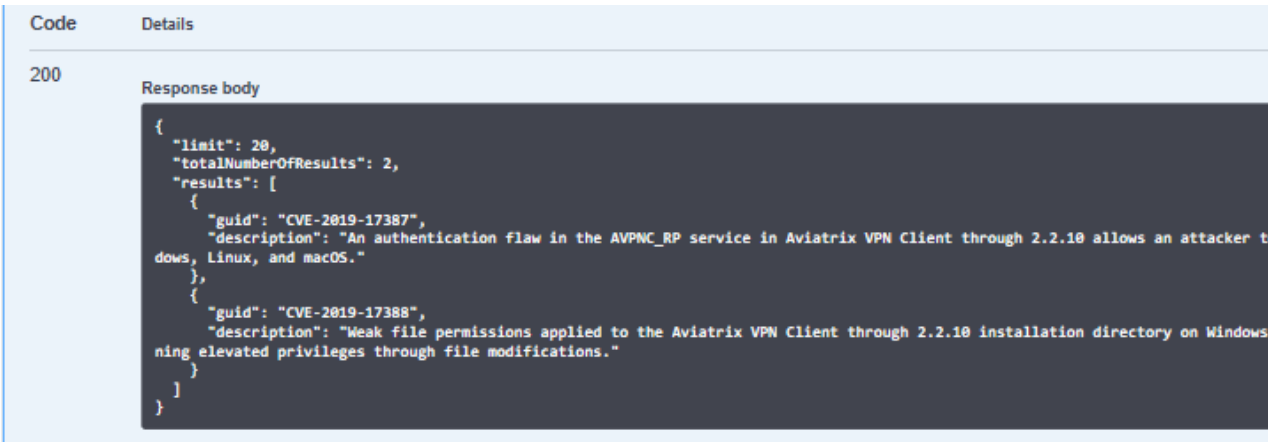

Рисунок 5.7 – Приклад вихідних даних для ендпоінта «/Vulnerabilities»

Другий ендпоінт «/Vulnerability/{cve}» дає змогу користувачу отримати детальну інформацію про вразливість за її унікальним ідентифікатором – CVE кодом. Приклад роботи ендпоінта представлений на рисунках 5.8-5.9.

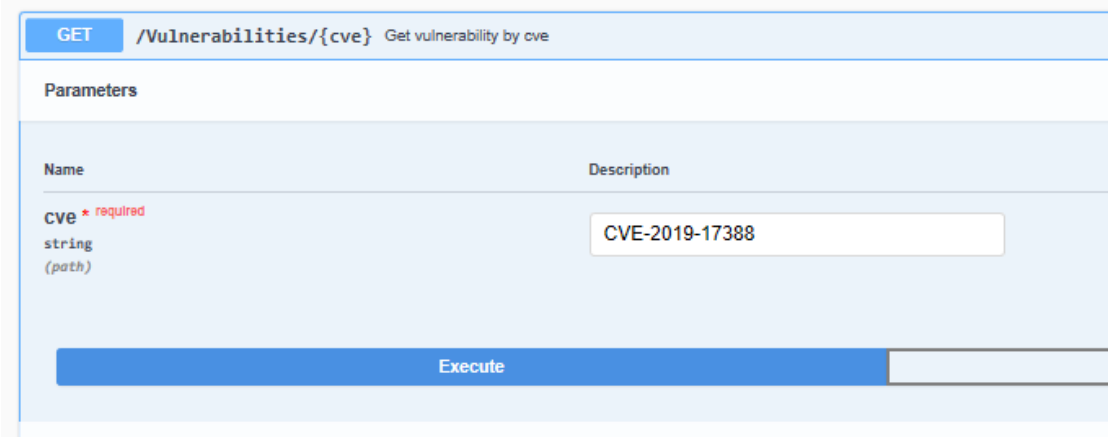

Рисунок 5.8 – Приклад вхідних даних для ендпоінта «/Vulnerability/{cve}»

Рисунок 5.9 – Приклад вихідних даних для ендпоінта «/Vulnerability/{cve}»

Останній ендпоінт «/VulnerabilityAssessment/{cve}» дає змогу користувачу отримати оцінку критичності вразливості за її унікальним ідентифікатором – CVE кодом та параметром, який характеризує навики зловмисника. Приклад роботи ендпоінта представлений на рисунках 5.10-5.11.

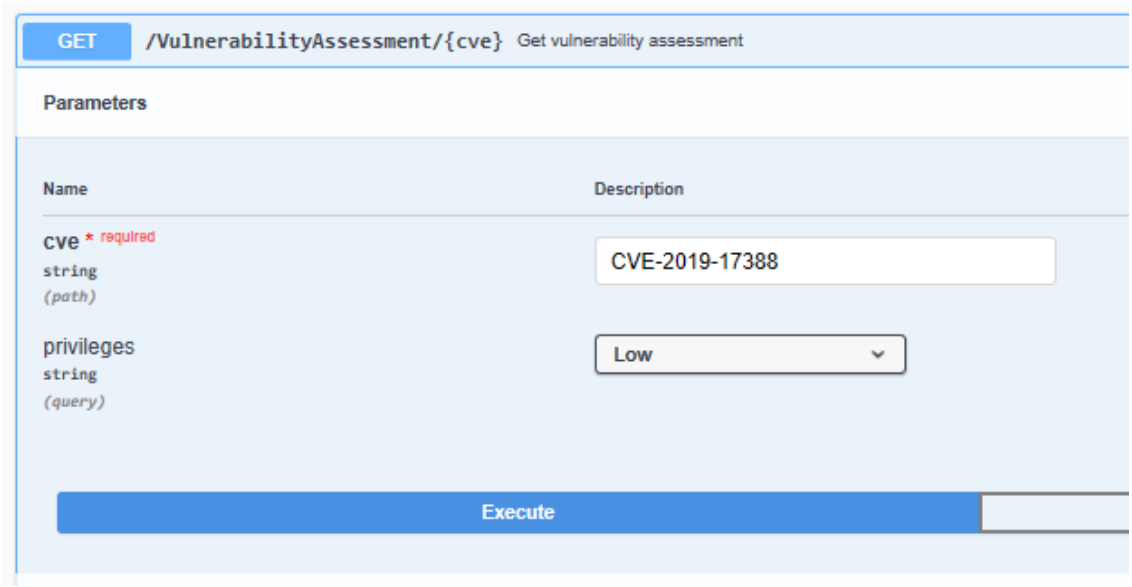

Рисунок 5.10 – Приклад вихідних даних для ендпоінта «/VulnerabilityAssessment/{cve}»

Рисунок 5.11 – Приклад вихідних даних для ендпоінта «/VulnerabilityAssessment/{cve}»

ВИСНОВКИ

На основі виконання атестаційної роботи, можна зробити висновок, що оцінка критичності вразливостей є складний і суперечний процес. Вона складається з великої системи метрик, параметрів так експертних оцінок.

Був проведений аналіз методик оцінки критичності вразливостей та існуючих програмних рішень на світовому ринку. На їх основі була здійснена вибірка метрик та параметрів для розробки власного продукту.

Також був розглянутий і проаналізований життєвий цикл вразливостей в операційних системах та прикладному програмному забезпеченні. На основі зробленого аналізу, можна вважати, що вразливості ОС є особливим випадком дефектів програмного забезпечення. Вони представляють загрозу безпеці та надійності системи більш ніж вразливості у прикладному програмному забезпеченні. Була призведена робота над аналізом і порівнянням баз даних вразливостей CVE, NVD та VULNDB. По показникам довірливості, зручності реалізації та великої бази інформації щодо вразливостей була обрана БД NVD.

Результатом атестаційної роботи є програмна реалізація засобу для оцінки рівня критичності вразливостей ОС. За допомогою розробленої програми можна отримати експертну характеристику вразливостей для облегшення та пришвидшення майбутнього аналізу ризиків. Також цей засіб можна використовувати як допоміжну утиліту для тестування безпеки, планування і розробки патчів безпеки. Крім того, вона надає достатньо інформації для вибору оптимальної за рівнем безпеки (за показниками користувачів) операційної системи, для реалізації своїх бізнес рішень. Подальшими етапами в розвитку цього проекту є:

- розробка додаткових метрик;
- аналіз факторів взаємодії середовища на вразливості;
- перехід з веб версії до мобільного додатку.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Слишком большие данные: сколько информации хранится в интернете? //[Електроний ресурс] //ain.ua– 27.05.2019. Режим доступу: https://ain.ua/special/skolko-vesit-internet/;

2. Бучик, С. С. ІНФОРМАЦІЙНА БЕЗПЕКА./ С. С. Бучик, В. О. Шалаєв – Ж.: Житомирський військовий інститут імені С. П. Корольова orcid.org/0000-0003-0892-3494;

3. Кузнецов, А.А. Безопасность информационных систем и технологий./ В.И. Есенин, А.А. Кузнецов, Л.С. Сорока - Х.: ООО «ЭДЭНА»,2010.-656с;

4. Певнев В. Я. Эффективность информационной безопасности замкнутых систем / В. Я. Певнев // Радіоелектронні і комп'ютерні системи. / 2009. - № 5(37). – С. 82-85;

5. Закон України «Про стандарти, технічні регламенти та процедури оцінки відповідності» від 01.12.2005 № 3164-IV;

6. ISO/IEC 27000 Информационные технологии - Методы и средства обеспечения безопасности - Системы менеджмента информационной безопасности – Общие сведения и словарь [Електроний ресурс] // ISO/IEC – 01.06.2019. Режим доступу: http://pqm-online.com/assets/files/pubs/translations/std/iso-mek-27000-2016.pdf;

7. АНАЛІЗ ПРОЦЕСУ УПРАВЛІННЯ РИЗИКАМИ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ В ЗАБЕЗПЕЧЕННІ ЖИВУЧОСТІ ІНФОРМАЦІЙНО-ТЕЛЕКОМУНІКАЦІЙНИХ СИСТЕМ [Електроний ресурс] // Академія сухопутних військ імені гетьмана Петра Сагайдачного – 27.05.2019. Режим доступу: file:///C:/Users/Shark%20Smart/Downloads/soi_2014_8_29.pdf;

8. Козлова, Е. А. Оценка рисков информационной безопасности с помощью метода нечеткой кластеризации и вычисления взаимной информации./ Е. А. Козлова - 2013. - №5. - С. 154-161;

9. Модель угроз ПД. Организационно-распорядительная документация по защите ПД) [Електроний ресурс] // Институт.ру – 10.11.2019. Режим доступу: http://www.intuit.ru/studies/courses/697/553/lecture/12447;

10. Дослідження вразливостей операційних систем з використанням програмних засобів Microsoft Basline Sequrity Analyzer [Електроний ресурс] // ukrbukva.net – 20.05.2019. Режим доступу: http://ukrbukva.net/page,3,95658-Issledovanie-uyazvimosteiy-operacionnyh-sistem-s-ispol-zovaniem-programmnyh-sredstv-Microsoft-Basline-Sequrity-Analyzer.html;

11. Безбогов А.А. Безопасность операционных систем : учебное пособие / А.А. Безбогов, А.В. Яковлев, Ю.Ф. Мартемьянов. – М. : "Издательство Машиностроение-1", 2007. – 220 с. – 400 экз. ISBN 978-5-94275-348-1;

12. Проблемы обеспечения безопасности ОС [Електроний ресурс] // Your Private Network– 01.06.2019. Режим доступу: http://ypn.ru/301/operating-systems-security-problems/;

13. Модель угроз ПД. Организационно-распорядительная документация по защите ПД) [Електроний ресурс] // Институт.ру – 10.11.2019. Режим доступу: http://www.intuit.ru/studies/courses/697/553/lecture/12447;

14. Цуранов М.В. Стандарти безпеки операційних систем : стаття / Цуранов М.В, Сліпченко О.В [Електроний ресурс] // hups.mil.gov.ua – 11.11.2019. Режим доступу: http://www.hups.mil.gov.ua/periodic-app/article/9774/ukr;

15. The Red Book: A Roadmap for Systems Security Research[Електроний ресурс] // redbook – 10.11.2019. Режим доступу: http://www.freetechbooks.com/the-syssec-red-book-a-roadmap-for-systems-security-research-t1195.html;

16. Race condition//[Електроний ресурс] //wikipedia– 24.05.2019. Режим доступу: https://en.wikipedia.org/wiki/Race_condition;

17. Боремся с Clickjacking //[Електроний ресурс] //Блог Ильи Барышева– 24.05.2019. Режим доступу: http://prophet.ru/2011/08/fighting-clickjacking/;

18. Анализ угроз и уязвимостей [Електроний ресурс] // ИБ – 02.06.2019. Режим доступу: https://goo.gl/VWYweQ;

19. Пошук вразливостей [Електроний ресурс] // SiteScanner – 01.06.2019. Режим доступу: http://bug.kpi.ua/web/index.php?r=site%2Farticle&id=37;

20. Ликбез по информационной безопасности// pikabu – 20.05.2019. Режимдоступу:https://pikabu.ru/story/likbez_po_informatsionnoy_bezopasnosti_5122856;

21. Life-cycle of a Security Vulnerability// RedHat – 21.05.2019. Режим доступу: https://access.redhat.com/blogs/766093/posts/1976453;

22. Vulnerability//[Електроний ресурс] //WhatIs.com – 23.05.2019. Режим доступу: https://whatis.techtarget.com/definition/vulnerability;

23. УПРАВЛЕНИЕ УЯЗВИМОСТЯМИ [Електроний ресурс] // Перспективный мониторинг – 03.06.2019. Режим доступу: https://amonitoring.ru/service/vulnerability-management/.

24. Threat and Vulnerability Management Standard [Електроний ресурс] // resolver.com – 28.05.2019. Режим доступу: https://www.resolver.com/trust/policies/threat-vulnerability-management-standard/;

25. Прикладное программное обеспечение [Електроний ресурс] // wikipedia – 01.06.2018. Режим доступу: https://goo.gl/kbXtzQ;

26. Vulnerability database [Електроний ресурс] // wikipedia– 01.06.2018. Режим доступу: https://goo.gl/yX6ttd;

27. Top 10 Vulnerabilities in Mobile Applications [Електроний ресурс] // whitehatsec – 01.06.2019. Режим доступу: https://www.whitehatsec.com/blog/top-10-vulnerabilities-in-mobile-applications/;

28. Experience Report: Study of Vulnerabilities of Enterprise Operating Systems [Електроний ресурс] // ieee – 02.06.2019. Режим доступу: https://ieeexplore.ieee.org/abstract/document/8109087/?part=1;

29. Операционная система [Електроний ресурс] // securitylab – 29.05.2019. Режим доступу: https://www.securitylab.ru/news/tags/%EE%EF%E5%F0%E0%F6%E8%EE%ED%ED%E0%FF+%F1%E8%F1%F2%E5%EC%E0/;

30. Vulnerabilities in Major Operating Systems Standard [Електроний ресурс] // researchgate – 29.05.2019. Режим доступу: https://www.researchgate.net/publication/2906333;

31. S. Frei, M. May, U. Fiedler and B. Plattner, “Large-scale vulnerability analysis,” in SIGCOMM Workshop on Large-Scale Attack Defense, 2006;

32. Common Vulnerabilities and Exposures [Електроний ресурс] // en.wikipedia– 30.05.2019. Режим доступу: https://en.wikipedia.org/wiki/Common_Vulnerabilities_and_Exposures;

33. Analyzing Vulnerability Databases [Електроний ресурс] // researchgate – 29.05.2019. Режим доступу: https://www.researchgate.net/publication/316971384_Analyzing_Vulnerability_Databases;

34. About OSVDB [Електроний ресурс] // blog.osvdb – 30.05.2019. Режим доступу: https://blog.osvdb.org/about/;

35. Cve reference key/maps [Електроний ресурс] // cve – 29.05.2019. Режим доступу: http://cve.mitre.org/data/refs/index.html;

36. Common Vulnerabilities and Exposures [Електроний ресурс] // en.wikipedia– 30.05.2019. Режим доступу: https://en.wikipedia.org/wiki/Common_Vulnerabilities_and_Exposures;

37. CVE Abstraction Content Decisions: Rationale and Application (Archived) [Електроний ресурс] // cve– 30.05.2019. Режим доступу: https://cve.mitre.org/cve/editorial_policies/cd_abstraction.html;

38. NVD Vulnerabilities [Електроний ресурс] // NATIONAL VULNERABILITY DATABASE – 29.05.2019. Режим доступу: https://nvd.nist.gov/vuln;

39. About VULDB [Електроний ресурс] // vuldb – 30.05.2019. Режим доступу: https://vuldb.com/?doc.about;

40. VulDB Includes False Report of Vulnerability in WordPress Plugin [Електроний ресурс] //Plugin Vulnerabilities– 31.05.2019. Режим доступу: https://www.pluginvulnerabilities.com/tag/vuldb/;

41. Andersson, O. Threat, risk, and vulnerability analyses during the development of IT systems in the Swedish Armed Forces./ Ola Andersson – Sweden: Umeå University Department of Computing Science SE-901 87 UMEÅ SWEDEN;

42. Полное руководство по общему стандарту оценки уязвимостей [Електроний ресурс] // securitylab – 31.05.2019. Режим доступу: https://www.securitylab.ru/analytics/355336.php;

43. Системы оценки уязвимостей [Електроний ресурс] // Windows IT Pro/RE – 31.05.2019. Режим доступу: https://www.osp.ru/winitpro/2006/02/1156304/;

44. Сканеры уязвимостей [Електроний ресурс] // it-black – 31.05.2019. Режим доступу: https://it-black.ru/skanery-uyazvimostey/;

45. Системы оценки уязвимостей [Електроний ресурс] // Windows IT Pro/RE – 31.05.2019. Режим доступу: https://www.osp.ru/winitpro/2006/02/1156304/;

46. Калькулятор CVSS V2 [Електроний ресурс] // Банк данных угроз безопасности информации– 31.05.2019. Режим доступу: https://bdu.fstec.ru/cvss2;

47. About OSVDB [Електроний ресурс] // blog.osvdb – 30.05.2019. Режим доступу: https://blog.osvdb.org/about/;

48. Оценка уязвимостей CVSS 3.0 [Електроний ресурс] //Хабр– 31.05.2019. Режим доступу: https://m.habr.com/company/pt/blog/266485/.