**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ**

**КАФЕДРАКОМП’ЮТЕРИЗОВАНИХ СИСТЕМ ЗАХИСТУ ІНФОРМАЦІЇ**

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач кафедри

А.В. Ільєнко

« » 20 р.

На правах рукопису

УДК 004.056.5:510.22(043.3)

**МАГІСТЕРСЬКА АТЕСТАЦІЙНА РОБОТА**

**ВИПКСКНИКА ОСВІТНЬОГО СТУПЕНЯ**

**«МАГІСТР»**

**Тема:** Метод тестування і оцінки захищеності комп’ютерних систем

|  |  |
| --- | --- |
| **Автор:** | Т.Ю. Загазей |
| **Науковий керівник:** д.ф-м.н.проф. | О.В. Азаренко |
| **Нормоконтролер:** асистент | С.В. Єгоров |
|  |  |

**Київ 2020**

**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ**

**Факультет:**Кібербезпеки, комп’ютерної та програмної інженерії

**Кафедра:** Комп’ютеризованих систем захисту інформації

**Освітній ступінь:** Магістр

**Спеціальність:** 125 «Кібербезпека»

**Освітньо-професійна програма:** «Безпека інформаційних і комунікаційних систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_\_\_\_\_\_\_С.В. Казмірчук

«\_\_\_»\_\_\_\_\_\_\_\_\_\_20\_\_\_ р.

**ЗАВДАННЯ**

**на виконання магістерської атестаційної роботи**

**магістрант Загазей Тарас Юрійович**

1. Тема: *Метод тестування і оцінки захищеності комп’ютерних систем*

затверджена наказом ректора від 02.04.2019 р. № 2265/ст

1. Термін виконання роботи**:** з 14.10.2019 р. по 09.02.2020 р.
2. Вихідні дані роботи: дані про інформаційно-комп'ютерну мережу, її склад, та види тестування.
3. Зміст пояснювальної записки: проведення аналізу нормативно-правового забезпечення України та міжнародних стандартів в сфері захисту інформації; дослідження методики тестування на проникнення; формування методу оцінки захищеності на основі тестування на проникнення; програмна реалізація методу; аналіз компонентів інформаційно-комунікаційної системи та визначення її рівня захищеності; надання рекомендацій щодо посилення захисту системи.
4. Перелік обов'язкового ілюстративного матеріалу: рисунки, таблиці.

**КАЛЕНДАРНИЙ ПЛАН**

**виконання магістерської роботи**

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п/п** | **Етапи виконання магістерської роботи** | **Термін виконання етапів** | **Примітка** |
| 1 | Уточнення постановки задачі | 14.10.2019-20.10.2019 | Виконано |
| 2 | Аналіз літературних джерел | 21.10.2019-21.11.2019 | Виконано |
| 3 | Обґрунтування вибору рішення | 22.11.2019 | Виконано |
| 4 | Збір інформації | 23.11.2019-30.11.2019 | Виконано |
| 5 | Аналіз даних та їх класифікація | 1.12.2019 | Виконано |
| 6 | Ознайомлення із загальними положеннями хакінгу та тестування на проникнення. | 2.12.2019-  12.12.2019 | Виконано |
| 7 | Створення полігону для проведеня тестування | 13.12.2019-14.12.2019 | Виконано |
| 8 | Створення алгоритму оцінювання захищеності на основі тестування на проникнення. | 15.12.2019-28.12.2019 | Виконано |
| 9 | Проведення тестування на проникнення та аналіз результатів | 29.12.2019-31.12.2019 | Виконано |
| 10 | Оформлення і друк пояснювальної записки | 1.01.2020-8.01.2020 | Виконано |
| 11 | Оформлення презентації | 9.01.2020 | Виконано |
| 12 | Захист роботи | 06.02.2020 | Виконано |

Магістрант Т. Загазей

(підпис, дата)

Науковий керівник О. Азаренко

(підпис, дата)

**РЕФЕРАТ**

Магістерська атестаційна робота складається зі вступу, основної частини, що містить 3 розділи, загальних висновків і списку використаних джерел, додатків і має 97 сторіноки основного тексту, 20 рисунків, 10 сторінок додатків. Список використаних джерел містить 31 найменування і займає 4 сторінки. Загальний обсяг роботи 97 сторінок.

Предмет дослідження: інформаційно-комп'ютерна мережа.

Об’єкт дослідження: метод оцінки захищеності мережі.

Мета дипломної роботи: показати ефективність тестування на проникнення як методу оцінки захищеності інформаційних систем та розробка алгоритму оцінювання знайдених вразливостей.

Дослідження по даній роботі проводились за допомогою системного аналізу методів та засобів оцінки захищеності мереж. Даний метод спирається на комплекс загальнонаукових, експериментальних, практичних, статистичних, математичних [методів](https://uk.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4).Дослідження проводились на віртуальній машині вже існуючими програмними засобами.

Наукова новизна дослідження полягає в наступному: вперше розроблено і реалізовано алгоритм та програму визначення рівня небезпеки вразливостей, що дозволило ефективно та наглядно визначати загальний рівень захищеності системи за рахунок оцінювання кожної знайденої вразливості.

Практична значимість дослідження полягає в тому, що розроблений метод та програмне забезпечення для оцінки вразливостей ефективно може застосовуватися для наочної демонстрації результатів тестування на проникненя та знайдених вразливостей. Також результати використовуються для покращення захищеності комп’ютерної системи.

Ключові слова: ТЕСТУВАННЯ НА ПРОНИКНЕННЯ, ОЦІНКА ЗАХИЩЕНОСТІ, ЕКСПЛУАТАЦІЯ ВРАЗЛИВОСТЕЙ, АУДИТ БЕЗПЕКИ, ВНУТРІШНЄ ТЕСТУВАННЯ НА ПРОНИКНЕННЯ, БЕЗПЕКА МЕРЕЖІ, ОЦІНКА РИЗИКІВ.

**ЗМІСТ**

[НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ 1](#_Toc30353149)

[ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ 7](#_Toc30353150)

[ВСТУП 8](#_Toc30353151)

[РОЗДІЛ 1 11](#_Toc30353152)

[АНАЛІЗ НОРМАТИВНИХ ДОКУМЕНТІВ У СФЕРІ КІБЕРБЕЗПЕКИ ТА ТЕСТУВАННЯ НА ПРОНИКНЕННЯ В УКРАЇНІ І СВІТІ 11](#_Toc30353153)

[1.1. Аналіз українського законодавства в сфері захисту інформації 11](#_Toc30353154)

[1.1.1. Нормативні документи в питанні побудови і вимог до ІКМ 13](#_Toc30353155)

[1.1.2. Нормативні документи в питанні оцінки захищеності і відповідності вимогам 16](#_Toc30353156)

[1.2. Недоліки законодавства в питанні тестування на проникнення 19](#_Toc30353157)

[1.3. Міжнародне законодавство та стандарти в питанні тестування на проникнення 23](#_Toc30353158)

[1.4. Висновки по розділу 27](#_Toc30353159)

[РОЗДІЛ 2 29](#_Toc30353160)

[ПОСТАНОВКА ЗАВДАННЯ ТА ТЕОРЕТИЧНІ АСПЕКТИ ТЕСТУВАННЯ НА ПРОНИКНЕННЯ ІКМ 29](#_Toc30353161)

[2.1. Аналіз необхідності тестування на проникнення 29](#_Toc30353162)

[2.2. Вразливості, які може виявити тестування на проникнення 32](#_Toc30353163)

[2.3. Основні поняття і терміни хакінгу та тестування на проникнення 35](#_Toc30353164)

[2.4. Аналіз загальної методики хакінгу 37](#_Toc30353165)

[2.4.1. Footprinting and Reconnaissance 38](#_Toc30353166)

[2.4.2. Scanning 41](#_Toc30353167)

[2.4.3. Enumeration 43](#_Toc30353168)

[2.4.4. System hacking 44](#_Toc30353169)

[2.4.5. Escalation of privilege 46](#_Toc30353170)

[2.4.6. Covering tracks 47](#_Toc30353171)

[2.4.7. Planting of backdoors 47](#_Toc30353172)

[2.5. Методика тестування на проникнення 48](#_Toc30353173)

[2.6. Аналіз відомих методик оцінки захищеності 50](#_Toc30353174)

[2.7. Аналіз алгоритму оцінювання вразливостей CVSS 51](#_Toc30353175)

[2.8. Висновки по розділу 54](#_Toc30353176)

[РОЗДІЛ 3 56](#_Toc30353177)

[ПРАКТИЧНА РЕАЛІЗАЦІЯ МЕТОДУ ОЦІНКИ ЗАХИЩЕНОСТІ СИСТЕМИ НА ОСНОВІ ТЕСТУВАННЯ НА ПРОНИКНЕННЯ 56](#_Toc30353178)

[3.1. Розробка алгоритму оцінки рівня небезпеки вразливостей та захищеності системи. 56](#_Toc30353179)

[3.1.1. Імовірні збитки від експлуатації вразливості 57](#_Toc30353180)

[3.1.2. Імовірність експлуатації вразливості 59](#_Toc30353181)

[3.1.3. Визначення сумарного рівня небезпеки вразливості 61](#_Toc30353182)

[3.2. Програмна реалізація алгоритму оцінки рівня небезпеки вразливостей та захищеності системи 64](#_Toc30353183)

[3.3. Аналіз причин походження та складності ліквідації вразливостей. 66](#_Toc30353184)

[3.4. Проведення демонстраційного тестування на проникнення 67](#_Toc30353185)

[3.5. Аналіз результатів 73](#_Toc30353186)

[3.6. Рекомендації щодо покращення захищеності 78](#_Toc30353187)

[3.7. Висновки по розділу 79](#_Toc30353188)

[ВИСНОВКИ 80](#_Toc30353189)

[СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ 82](#_Toc30353190)

[Додаток А. Програмний код калькулятора оцінювання рівня загрози вразливостей 86](#_Toc30353191)

# ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

ІКМ – Інформаційно-комп’ютерна мережа

CERT-UA - Computer Emergency Response Team of Ukraine

FIRST - Forum of Incident Response and Security Teams

ОС – Операційна система

КЗЗ – Комплекс засобів захисту

АС – Автоматизована система

КС – Комп’ютерна система

СУБД - Систе́ма управлі́ння ба́зами да́них

ІSО - International Organization for Standardization

DoS – Denial-of-service

SQL - Structured query language

DNS - Domain Name System

HTML - HyperText Markup Language

ICMP - Internet Control Message Protocol

SAM - Security Account Manager

DLL - Dynamic-link library

AES - Advanced Encryption Standard

FTP – File Transfer Protocol

SMB – Server Message Block

HTTP – Hypertext Transfer Protocol

XSS - Cross-site scripting

# ВСТУП

Діяльність сучасного суспільства|товариства| в значній мірі|значною мірою| залежить від надійних інформаційно-комунікаційних систем, в яких циркулюють гігантські обсяги|обсяги| інформації. З метою якісного та ефективного функціонування цих систем|коштів| розробляється та впроваджується широкий спектр сучасних технічних засобів і|устро програмно-апаратних комплексів. Таким чином кількість нових технологій, програмного та апаратного забезпечення з кожним роком невпинно зростає, а разом з цим і зростає кількість вразливостей та можливостей для зловмисників заподіяти певну шкоду.

На сьогоднішній день в розвинутих країнах сфера захисту інформації не відстає в темпах розвитку від інформаційних технологій, проте покищо не існує абсолютно безпечної інформаційно-комп’ютерної мережі, технології, програми чи апаратного пристрою. Тому постає питання щоб якнайкраще захистити мережу, тобто звести імовірність злому практично до нуля. В цій справі ніяк не обійтися без різноманітних методів оцінки захищеності, таких як аудит інформаційної безпеки, тестування на проникнення, експертизи, передексплуатаційні випробування тощо.

Тестування на проникнення набирає все більшої популярності і це не дивно, адже воно симулює дії хакера і найбільш точно може оцінити захищеність системи в реальних умовах. Суть даного метода полягає в проведені безлічі атак на систему з метою виявлення вразливостей і їх експлуатації. Далі всі вразливості описуються і у формі звіту разом із рекомендаціями щодо їх закриття подаються замовнику, який на основі цього звіту може оцінити рівень захисту своєї системи і скористатися рекомендаціями для його підвищення.

Таким чином тестування на проникнення виступає дуже потужним інструментом в сфері кібербезпеки, а саме в способах оцінки захищеності.

Мета|ціль| та завдання роботи – показати ефективність тестування на проникнення як методу оцінки захищеності інформаційних систем та удосконалити його, шляхом додавання алгоритму виставлення рейтингової оцінки системі що тестується. Така оцінка допоможе замовнику та, в разі розголошення результатів, іншим особам чи організаціям краще зрозуміти на якому рівні знаходиться його система безпеки.

Відповідно до мети роботи необхідно вирішити наступні завдання:

* Визначити актуальність тестування на проникнення в сучасному світі.
* Дослідити методику проведення тестування на проникнення;
* Розробити алгоритм оцінювання результатів тестування ;
* Практично провести демонстраційне тестування на проникнення;
* Проаналізувати отримані результати та оцінити їх.

Об’єктом дослідження є тестування на проникнення як метод оцінки захищеності комп’ютерної системи.

Практична значимість дослідження полягає в наступному: в роботі був розроблений алгоритм виставленя рівня загрози вразливостей та оцінки захищеності системи на основі проведення тестування на проникнення.

Розроблений алгоритм базується на основних характеристиках інформації та таких метриках як складність атаки, рівень нанесених збитків, та умови, в яких знаходиться атакуючий. В результаті оцінювання виставляється оцінка загрозі вразливості та оцінка захищеності системи.

Проведений аналіз методики оцінювання захищеності – тестування на проникнення показав її актуальність в сучасному світі та необхідність практично для всіх видів організацій, в яких відбувається електронний обіг важливої інформації, незалежно від масштабів мережі.

Були опубліковані тези, присвячених темі, що розглядалася в дипломній роботі:

Мелешко О.О., Загазей Т.Ю. Метод оцінки захищеності інформаційно комп'ютерної мережі на основі тестування на проникнення // XV Міжнародна науково-практична конференція «Динаміка сучасної науки - 2019» - 2019.

Корнієнко Б.Я., Загазей Т.Ю. Метод оцінки захищеності інформаційно-комп'ютерної мережі на основі тесту на проникнення //ПОЛІТ. СУЧАСНІ ПРОБЛЕМИ НАУКИ – 2018. – с.5

# РОЗДІЛ 1

# АНАЛІЗ НОРМАТИВНИХ ДОКУМЕНТІВ У СФЕРІ КІБЕРБЕЗПЕКИ ТА ТЕСТУВАННЯ НА ПРОНИКНЕННЯ В УКРАЇНІ І СВІТІ

## Аналіз українського законодавства в сфері захисту інформації

Правову основу забезпечення кібербезпеки України становлять [Конституція України](http://zakon5.rada.gov.ua/laws/show/254%D0%BA/96-%D0%B2%D1%80), закони України щодо основ національної безпеки, засад внутрішньої і зовнішньої політики, електронних комунікацій, захисту державних інформаційних ресурсів та інформації, вимога щодо захисту якої встановлена законом, цей та інші закони України, [Конвенція про кіберзлочинність](http://zakon5.rada.gov.ua/laws/show/994_575), інші міжнародні договори, згода на обов’язковість яких надана Верховною Радою України, укази Президента України, акти Кабінету Міністрів України, а також інші нормативно-правові акти, що приймаються на виконання законів України [1].

Згідно з Законом Укрїни «Про основні засади забезпечення кібербезпеки України» об’єктами кіберзахисту є: комунікаційні системи всіх форм власності, в яких обробляються національні інформаційні ресурси та/або які використовуються в інтересах органів державної влади, органів місцевого самоврядування, правоохоронних органів та військових формувань, утворених відповідно до закону; об’єкти критичної інформаційної інфраструктури; комунікаційні системи, які використовуються для задоволення суспільних потреб та/або реалізації правовідносин у сферах електронного урядування, електронних державних послуг, електронної комерції, електронного документообігу [1].

Також в цьому Законі вказано, що суб’єкти забезпечення кібербезпеки у межах своєї компетенції:

* здійснюють заходи щодо запобігання використанню кіберпростору у воєнних, розвідувально-підривних, терористичних та інших протиправних і злочинних цілях [1];
* здійснюють виявлення і реагування на кіберінциденти та кібератаки, усунення їх наслідків [1];
* здійснюють інформаційний обмін щодо реалізованих та потенційних кіберзагроз [1];
* розробляють і реалізують запобіжні, організаційні, освітні та інші заходи у сфері кібербезпеки, кібероборони та кіберзахисту [1];
* забезпечують проведення аудиту інформаційної безпеки, у тому числі на підпорядкованих об’єктах та об’єктах, що належать до сфери їх управління [1];
* здійснюють інші заходи із забезпечення розвитку та безпеки кіберпростору [1].

Далі бачимо визначення щодо державно-приватної взаємодії у сфері кібербезпеки, яка здійснюється шляхом створення системи своєчасного виявлення, запобігання та нейтралізації кіберзагроз, у тому числі із залученням волонтерських організацій тощо.

Також, згідно до ЗУ «Про основні засади забезпечення кібербезпеки України» створена урядова команда реагування на комп’ютерні надзвичайні події України CERT-UA, завданнями якої є:

* накопичення та проведення аналізу даних про кіберінциденти, ведення державного реєстру кіберінцидентів [1];
* надання власникам об’єктів кіберзахисту практичної допомоги з питань запобігання, виявлення та усунення наслідків кіберінцидентів щодо цих об’єктів [1];
* організація та проведення практичних семінарів з питань кіберзахисту для суб’єктів національної системи кібербезпеки та власників об’єктів кіберзахисту [1];
* підготовка та розміщення на своєму офіційному веб-сайті рекомендацій щодо протидії сучасним видам кібератак та кіберзагроз [1];
* взаємодія з правоохоронними органами, забезпечення їх своєчасного інформування про кібератаки [1];
* взаємодія з іноземними та міжнародними організаціями з питань реагування на кіберінциденти, зокрема в рамках участі у Форумі команд реагування на інциденти безпеки FIRST із сплатою щорічних членських внесків [1];
* взаємодія з українськими командами реагування на комп’ютерні надзвичайні події, а також іншими підприємствами, установами та організаціями незалежно від форми власності, які провадять діяльність, пов’язану із забезпеченням безпеки кіберпростору [1];
* опрацювання отриманої від громадян інформації про кіберінциденти щодо об’єктів кіберзахисту [1];
* сприяння державним органам, органам місцевого самоврядування, військовим формуванням, утвореним відповідно до закону, підприємствам, установам та організаціям незалежно від форми власності, а також громадянам України у вирішенні питань кіберзахисту та протидії кіберзагрозам [1].

Отже, сьогодні вже на державному рівні аудит, складовою якого може бути тестування на проникнення, є одною з обов’язнових дій в державній сфері. Проте, обов’язковість аудиту і зокрема пентесту не розповсюджюється на приватний сектор.

## Нормативні документи в питанні побудови і вимог до ІКМ

Першим документом буде НД ТЗІ 2.5-005-99 Класифікація автоматизованих систем і стандартні функціональні профілі захищеності оброблюваної інформації від несанкціонованого доступу (зі зміною № 1).

Цей документ установлює принципи класифікації автоматизованих систем і утворення стандартних функціональних профілів захищеності оброблюваної інформації від несанкціонованого доступу [2].

Цей документ призначений для постачальників (розробників), споживачів (замовників, користувачів) автоматизованих систем, які використовуються для обробки (в тому числі збирання, зберігання, передачі і т. ін.) критичної інформації (інформації, яка потребує захисту), а також для державних органів, які здійснюють функції контролю за обробкою такої інформації [2].

Згідно з цим нормативним документом Автоматизована система являє собою організаційно-технічну систему, що об’єднує ОС, фізичне середовище, персонал і оброблювану інформацію. Вимоги до функціонального складу КЗЗ залежать від характеристик оброблюваної інформації, самої ОС, фізичного середовища, персоналу і організаційної підсистеми. Вимоги до гарантій визначаються насамперед характером (важливістю) оброблюваної інформації і призначенням АС [2].

Всі АС поділяються на 3 класи. Розглянемо 2-й і 3-й Класи, до яких відносяться практично всі сучасні діючі ІКМ, на які зазвичай замовляють аудит чи тестування на проникнення.

Клас «2» — локалізований багатомашинний багатокористувачевий комплекс, який обробляє інформацію різних категорій конфіденційності [2].

Клас «3» — розподілений багатомашинний багатокористувачевий комплекс, який обробляє інформацію різних категорій конфіденційності [2].

Далі описуються функціональні профілі захищеності, які в процесі аудиту допоможуть зрозуміти рівень захисту ІКМ та необхідний рівень впливу на неї, якщо в межах аудиту проводиться тестування на проникнення.

Стандартний функціональний профіль захищеності являє собою перелік мінімально необхідних рівнів послуг, які повинен реалізовувати КЗЗ обчислювальної системи АС, щоб задовольняти певні вимоги щодо захищеності інформації, яка обробляється в даній АС.

Стандартні функціональні профілі будуються на підставі існуючих вимог щодо захисту певної інформації від певних загроз і відомих на сьогоднішній день функціональних послуг, що дозволяють протистояти даним загрозам і забезпечувати виконання вимог, які пред’являються.

Для стандартних функціональних профілів захищеності не вимагається ні зв’язаної з ними політики безпеки, ні рівня гарантій, хоч їх наявність і допускається в разі необхідності. Політика безпеки КС, що реалізує певний стандартний профіль, має бути «успадкована» з відповідних документів, що встановлюють вимоги до порядку обробки певної інформації в АС. Так, один і той же профіль захищеності може використовуватись для опису функціональних вимог з захисту оброблюваної інформації і для ОС, і для СУБД, в той час, як їх політика безпеки, зокрема визначення об’єктів, буде різною [2].

Єдина вимога, якої слід дотримуватися при утворенні нових профілів, — це додержання описаних в НД ТЗІ 2.5-004-99 «Критерії оцінки захищеності інформації в комп’ютерних системах від несанкціонованого доступу» необхідних умов для кожної із послуг, що включаються до профілю.

Функціональні профілі можуть використовуватись також для зрівняння оцінки функціональності КС за критеріями інших держав з оцінкою за національними критеріями [2].

Захист інформації, що обробляється в АС, полягає в створенні і підтримці в дієздатному стані системи заходів, як технічних (інженерних, програмно-апаратних), так і нетехнічних (правових, організаційних), що дозволяють запобігти або ускладнити можливість реалізації загроз, а також знизити потенційні збитки. Іншими словами, захист інформації спрямовано на забезпечення безпеки оброблюваної інформації і АС в цілому, тобто такого стану, який забезпечує збереження заданих властивостей інформації і АС, що її обробляє. Система зазначених заходів, що забезпечує захист інформації в АС, називається комплексною системою захисту інформації [2].

Істотна частина проблем забезпечення захисту інформації в АС може бути вирішена організаційними заходами. Проте, з розвитком інформаційних технологій спостерігається тенденція зростання потреби застосування технічних заходів і засобів захисту.

Правовою основою забезпечення технічного захисту інформації в Україні є Конституція України, Закони України "Про інформацію", "Про захист інформації в автоматизованих системах", "Про державну таємницю", "Про науково-технічну інформацію", Концепція (основи державної політики) національної безпеки України, Концепція технічного захисту інформації в Україні, інші нормативно-правові акти, а також міжнародні договори України, що стосуються сфери інформаційних відносин. [3]

НД ТЗІ 1.1-002-99 «Загальні положення щодо захисту інформації в комп'ютерних системах від несанкціонованого доступу» і комплект НД, що базується на ньому, присвячений питанням організації захисту від НСД і побудови засобів захисту від НСД, що функціонують у складі обчислювальної системи АС. Організаційні і фізичні заходи захисту, включаючи захист від фізичного НСД до компонентів ОС, як і захист від витоку технічними каналами не є предметом розгляду. Незважаючи на це, при викладі увага приділяється також і деяким нетехнічним аспектам, але тільки там, де це впливає на оцінку технічної захищеності.

### 1.1.2. Нормативні документи в питанні оцінки захищеності і відповідності вимогам

2.5-004-99 «Критерії оцінки захищеності інформації в комп’ютерних системах від несанкціонованого доступу» (далі — Критерії) — установлює критерії оцінки захищеності інформації, оброблюваної в комп'ютерних системах, від несанкціонованого доступу.

Критерії є методологічною базою для визначення вимог з захисту інформації в комп'ютерних системах від несанкціонованого доступу; створення захищених комп'ютерних систем і засобів захисту від несанкціонованого доступу; оцінки захищеності інформації в комп'ютерних системах і їх придатності для обробки критичної інформації (інформації, що вимагає захисту). [4]

Критерії надають:

* + - Порівняльну шкалу для оцінки надійності механізмів захисту інформації від несанкціонованого доступу, реалізованих в комп'ютерних системах. [4]
    - Базу (орієнтири) для розробки комп'ютерних систем, в яких мають бути реалізовані функції захисту інформації. [4]

Критерії можуть застосовуватися до всього спектра комп'ютерних систем, включаючи однорідні системи, багатопроцесорні системи, бази даних, вбудовані системи, розподілені системи, мережі, об'єктно-орієнтовані системи та ін.

Цей документ призначено для постачальників (розробників), споживачів (замовників, користувачів) комп'ютерних систем, які використовуються для обробки (в тому числі збирання, зберігання, передачі і т. ін.) критичної інформації, а також для органів, що здійснюють функції оцінювання захищеності такої інформації та контролю за її обробкою. [4]

Цей документ відображає сучасний стан проблеми і підходів до її розв'язання. З розвитком нових тенденцій в галузі і за умови достатньої обгрунтованості документ є відкритим для включення до його складу Департаментом спеціальних телекомунікаційних систем та захисту інформації Служби безпеки України нових послуг. [4]

НД ТЗІ 2.7-009-09 «Методичні вказівки з оцінювання функціональних послуг безпеки в засобах захисту інформації від несанкціонованого доступу» містить методичні вказівки та рекомендації щодо здійснення експертного оцінювання відповідності засобів захисту інформації в комп'ютерних системах від несанкціонованого доступу (НСД) та комплексів засобів захисту (КЗЗ) комплексних систем захисту інформації (КСЗІ) в інформаційно-телекомунікаційних системах (ІТС) вимогам до технічного захисту інформації (ТЗІ) в частині оцінювання функціональних послуг безпеки. У частині, що стосується методології проведення оцінювання, наведені рекомендації можуть також використовуватися при проведенні експертизи засобів захисту інформації, в яких сукупність реалізованих функцій захисту не в повному обсязі відповідає встановленим НД ТЗІ вимогам щодо реалізації функціональних послуг безпеки [5].

НД призначений для державних органів, органів місцевого самоврядування, органів управління Збройних Сил України, інших військових формувань, а також підприємств, установ і організацій всіх форм власності, які виконують роботи зі створення та проведення експертизи засобів захисту інформації в комп'ютерних системах від НСД на відповідність вимогам НД системи ТЗІ в Україні[5].

Як зазначається в НД ТЗІ 1.1-002-99, з погляду методології в проблемі захисту від НСД інформації, оброблюваної в ІТС, можна виділити два напрями:

* забезпечення захищеності інформації у функціонуючих та/або створюваних ІТС [3];
* створення засобів технічного захисту інформації (ЗТЗІ) від НСД або захищених від НСД компонентів обчислювальної системи поза конкретним середовищем експлуатації [3].

При цьому, як у першому, так і в другому випадку, доцільним, а якщо в ІТС планується оброблення інформації, порядок захисту якої регламентується законами України або іншими нормативно-правовими актами (наприклад, інформації яка становить державну таємницю або вимоги до захисту якої встановлено законодавством), обов'язковим є незалежне підтвердження (оцінювання) відповідності реалізованих засобів та заходів захисту встановленим вимогам та нормам. Результатами проведеного оцінювання має бути відповідний висновок, на підставі якого власники ІТС та оброблюваних у них інформаційних ресурсів можуть приймати рішення щодо прийнятності та достатності вжитих заходів і реалізованих засобів.

У процесі проведення оцінювання, окрім сукупності показників, що характеризують конкретну ІТС або засіб захисту, необхідними також є:

* критерії оцінки, під якими слід розуміти сукупність вимог (шкала оцінки), яка використовується для оцінювання ефективності функцій захисту інформації та рівня гарантій коректності їх реалізації [3];
* система оцінювання, під якою слід розуміти адміністративно-правову структуру, у рамках якої організації, що проводять оцінювання, застосовують критерії оцінки [3];
* методологія оцінювання, яка визначає послідовність (алгоритм) дій, що виконуються експертами при оцінюванні ефективності функцій захисту інформації та рівня гарантій коректності їх реалізації, а також форму подання результатів оцінювання [3].

В Україні як критерії оцінки використовуються критерії, встановлені НД ТЗІ 2.5-004-99, а також вимоги інших НД ТЗІ щодо забезпечення захисту інформації в ІТС різного призначення. Вони надають:

* порівняльну шкалу для оцінювання ефективності функцій і механізмів захисту інформації від НСД, реалізованих в ІТС, а також рівня гарантій коректності їх реалізації [4];
* базу (орієнтири) для розроблення засобів захисту інформації, оброблюваної в ІТС, від НСД [4].

Згідно з вимогами НД ТЗІ 2.5-004-99, окремо оцінюються реалізовані функції захисту (функціональні послуги безпеки, ФПБ) та рівень гарантій коректності їх реалізації (рівень гарантій).

### 1.2. Недоліки законодавства в питанні тестування на проникнення

Українське законодавство в питанні аудиту і пентесту, або ж взагалі кібербезпеки ще не достатньо розвинене і часто можна трактувати тестування на проникнення як зловмисні дії.

Хоча останнім часом спостерігається рух в бік приведення чинного законодавства в сфері ІБ до норм і стандартів міжнародного.

Так, з метою забезпечення розвитку інформаційного суспільства в Україні, враховуючи низку наявних проблем в інформаційній сфері, зокрема необхідність розвитку інформаційного законодавства, проведення системної кореляції з вирішення питань захисту та забезпечення основних прав, цінностей і соціально-економічних інтересів людини, суспільства і держави, побудови в умовах відкритого суспільства ефективної системи інформаційної безпеки України, а також стан виконання [Рекомендацій парламентських слухань з питань розвитку інформаційного суспільства в Україні](http://zakon5.rada.gov.ua/laws/show/ru/3175-15), що відбулися 21 вересня 2005 року, схвалених Постановою Верховної Ради України від 1 грудня 2005 року № 3175-IV, були висунуті рекомендації, одна з яких: «адаптувати системи захисту державних інформаційних ресурсів до вимог та стандартів Європейського Союзу з проведенням тестів на проникнення критично важливих об’єктів національної інфраструктури».

Не зважаючи на все, неузгоджений пентест є протизаконним, навіть якщо всі результати будуть передані організації, що тестувалася, і всі дії будуть здійсненні виключно в корисних для організації цілях.

Сьогодні в Україні спостерігається помітне зростання ролі волонтерських та громадських структур, які допомагали державі в різних аспектах національної безпеки. Не стала виключенням і кіберсфера. Проукраїнські хактивісти (зокрема, «Українські кібер війська», «Кіберсотня», «FalconsFlame», «Trinity», «RUH8» та інші) здійснювали:

* злами комп’ютерних систем як терористів, так і їх російських кураторів;
* отримували доступ до поштових скриньок осіб, які задіяні в організації та реалізації російської агресії проти України;
* відслідковували аккаунти в соціальних мережах та мережі інтернет через які терористичні угрупування вели свою агітацію та збирали кошти;
* блокували окремі електронні гаманці фінансистів терористів;

Вочевидь для класифікації їх діяльності найбільш доцільно використовувати концепцію кольорових «капелюхів» (hat) , що вже стала звичною практикою при характеристиці дій хакерських (чи просто ІТ- експертних) угрупувань. «Білі капелюхи» (White hat) зазвичай є повністю легальними ІТ-фахівцями, що здійснюють свою діяльність законно і частіше за все – на комерційній основі. Основна їх мета – вдосконалення систем кібербезпеки тих чи інших структур (державних чи приватних), в тому числі через пошук недоліків коду тих інформаційних систем, які використовують ці структури. Основний інструмент – пентест. Важливим є те, що ці дії вони застосовують на запит самої організації та за чіткою попередньою домовленістю з нею.

Останнім часом їх безпосередня активність все частіше спрямовується на внутрішні інформаційні системи (передусім – державних органів та об’єктів критичної інфраструктури) щодо яких вони проводять несанкціоновані пентести, скачуючи при цьому окремі документи (які не мають грифів обмеження доступу) для підтвердження наявності уразливості. Незважаючи на те, що ці дії здійснюються ними в інтересах забезпечення більшої кібербезпеки держави в умовах агресії (принаймні така мета ними публічно декларується), але відповідно до чинного законодавства вони частіше за все є порушенням українського законодавства.

Показовим в цьому сенсі був випадок із українським програмістом О. Моховим, який у 2013 році помітив уразливість у системі «Приват-24 [6]», та здійснивши декілька тестів уразливості звернувся до Служби безпеки Приват-банку. Однак фактично експлуатація цієї уразливості можна було кваліфікувати як порушення законодавства (про що і заявила тоді прес- служба банку [7]), в т.ч. – статті 361 КК України. В подальшому ситуація була вирішена, однак при цьому принципова проблема не зникла. Ще більш виразно проблема своєрідної «негнучкості» українського законодавства щодо діяльності у кіберпросторі проявилась у історії навколо акції [8] (флешмобу) «Ukrainian Cyber Alliance» під загальним хештегом #F\*ckResponsibleDisclosure який розпочався у листопаді 2017 року. За словами організаторів його мета - «громадська акція для підвищення рівня ITгігієни» (за суттю акція є формою краудсорсингового пентесту державних інформаційних систем). В межах «флешмобу» були знайдені уразливості в комп’ютерних системах щонайменше 10 ЦОВВ (в т.ч. – правоохоронних органів), декількох облрад, мінімум 5 об’єктів, які можна віднести до критичної інфраструктури держави, низки наукових та комунальних закладів. Щонайменше одна з атакованих структур звернулась до поліції із заявою щодо незаконного вторгнення в локальну комп’ютерну мережу облради.

Слід визнати, що з погляду чинного законодавства дії «Ukrainian Cyber Alliance» дійсно порушують тією чи іншою мірою статтю 361 Кримінального Кодексу України: «несанкціоноване втручання в роботу електронно- обчислювальних машин (комп'ютерів), автоматизованих систем, комп'ютерних мереж чи мереж електрозв'язку» [9]. Фактично згадана стаття розглядає будь-яке втручання (в т.ч. те, яке не нанесло безпосередньої шкоди) до автоматизованої системі чи комп’ютерної мережі як злочин, адже внаслідок дій «Ukrainian Cyber Alliance» відбувся витік інформації. Формулювання зазначеної статті КК фактично унеможливлює діяльність некомерційних пентестерів, якщо ці тести заздалегідь не погоджені із об’єктами атаки. Вочевидь діяльність груп на кшталт «Ukrainian Cyber Alliance» із виявлення вразливостей в інформаційних системах (в т.ч. – органів державної влади) є важливою і цілком підпадає під сутнісні ознаки державно-приватного партнерства, однак потребує свого повноцінного законодавчого вирішення. Передусім - задля виведення таких хактивістів з-під дії статей КК України та здійснення їх діяльності у легальний спосіб, який би не наносив шкоду державним ресурсам і не створював передумови для застосування правоохоронними органами статті 361 [10].

Крім того, слід визнати, що діяльність «Ukrainian Cyber Alliance» та схожих на них хактивістів (які відрізняються від комерційних пентестерів які узгоджують свої дії із структурою, щодо якої проводиться пентест), вочевидь частково підпадає під формулювання «негласна перевірка готовності об’єктів критичної інфраструктури до можливих кібератак та кіберінцидентів», що відповідно до Закону України «Про основні засади забезпечення кібербезпеки України». покладено на Службу безпеки України. При цьому стаття 361 КК України підвідомча Національній поліції, відтак дії таких хактивістів мають бути узгоджені як з Національною поліцією (профільним департаментом), так і зі Службою безпеки України. Важливо відмітити, що «негласна перевірка…» є складовою оперативно-розшукової діяльності, яка у Законі України «Про оперативно- розшукову діяльність» визначена як «система гласних і негласних пошукових, розвідувальних та контррозвідувальних заходів, що здійснюються із застосуванням оперативних та оперативно-технічних засобів», а відповідно до українського законодавства і Служба безпеки України, і Національна поліція України є суб’єктами проведення оперативно-розшукової діяльності [10].

Важливо відмітити ще один аспект: відповідно до Закону України «Про основні засади забезпечення кібербезпеки України» суб’єкти кібербезпеки реалізують свої повноваження передусім щодо об’єктів критичної інфраструктури. Сайти ж органів державної влади, їх внутрішні комп’ютерні мережі швидше за все не підпадуть під ознаки об’єктів критичної інфраструктури. Відтак потрібно або уточнити положення зазначеного Закону, або ж передбачити можливість використання механізму співпраці з активістами і для цієї сфери в інших законах України [10].

### 1.3. Міжнародне законодавство та стандарти в питанні тестування на проникнення

Як було сказано вище, тестування на проникнення є частиною аудиту інформаційної безпеки (ІБ), отже пентест підпадає під стандартизацію та зоконодавство щодо проведення аудиту ІБ.

Опираючись на зарубіжні публікації, я провів огляд нормативних та законодавчих документів, які регулюють технології аудиту інформаційної безпеки організацій. Сьогодні актуальними питаннями є вимоги до практичних дій зі сторони організацій щодо контролю за дотриманням вимог до інформаційної безпеки та порядок визначення, чи виконують організації всі необхідні заходи з інформаційної безпеки.

Корисним інструментом, який може використовуватися для вдосконалювання системи інформаційної безпеки, є бібліотека інфраструктури ІТ (ІT Іnfrastructure Library – ІTІL) – набір оптимальних методів та принципів, які визначають інтегрований, заснований на процесах підхід з управління ІТ. Зацікавленість у застосуванні ІTІL постійно росте по усьому світу.

ІTІL рекомендує впровадження ефективних заходів в області ІБ на стратегічному, тактичному та операційному рівнях. Забезпечення ІБ розглядається як циклічний процес з фазами планування, впровадження, оцінки та підтримки. ІTІL оперує такими поняттями в області ІБ, як політики, процеси, процедури та інструкції. З деякими особливостями аналогічні підходи прослідковуються в CОBІT, а також у вітчизняних нормативних та законодавчих актах. Хоча в ІTІL відсутні безпосередні спеціалізовані стандарти оцінки відповідності, він є близьким до британського стандарту BS 15000, який присвячений управлінню ІТ-сервісами та методам їх оцінки.

Ще одним широко обговорюваним стандартом в області безпеки, є стандарт ІSО/ІЕC 15408 «Загальні критерії ІБ ІТ», який був гармонізований, наприклад, у Росії, як державний стандарт Р ИСО/МЭК 15408. Цей стандарт технічний і іноді важкий для сприйняття бізнесом. Він корисний для постачальників та покупців продукції ІБ, для того щоб визначити, наскільки достатнім є механізм захисту в продукції, що випробовується. На жаль, він не допомагає керівництву розібратися, чи правильно воно діє, забезпечуючи той чи інший рівень ІБ [11].

Найбільш відомими та широко використовуваними стандартами управління ІБ, а також доказом дотримання нормативних актів та законодавства, є міжнародні стандарти серії ІSО/ІЕC 2700X по управлінню ІТ. Беручи свій початок від первісних Британських стандартів серії 7799 (далі – ІSО/ІЕC 17799 та ІSО/ІЕC 27001), ці стандарти конкретно та чітко визначають технології ефективного впровадження систем управління ІТ. Є кілька причин, чому ці стандарти настільки популярні. Не останньою з них є та, що в них чіткі вказані методи проведення аудиторських перевірок на відповідність, а також можливість сертифікації по ІSО/ІЕC 27001 [11].

ІSО/ІЕC 17799 та ІSО/ІЕC 27001 допомагають відповістити на запитання: «як довести, що в організації забезпечений необхідний рівень безпеки?» і переконати регулювальні органі, що «усе виконується правильно» та «належним чином» [11].

ІSО/ІЕC 17799 та ІSО/ІЕC 27001 охоплюють всі основні сфери вимог, які пропонуються законодавством та нормативними актами, згаданими вище. Наріжним каменем відповідності стандартам є розуміння того, які інформаційні активи має організація, і впровадження необхідного рівня заходів контролю, заснованого на оцінці ризиків [11].

ІSО/ІЕC 17799 та ІSО/ІЕC 27001 – це просто та доступно написані стандарти, що надають корисні методики з заходів контролю, які організація захоче впровадити. При цьому стандарти зрозумілі як фахівцям в області ІБ, так і керівництву. Вони допомагають подолати комунікаційний бар’єр між обома сторонами, забезпечивши тим самим розуміння керівництвом, що робиться та чому. Керівництво розглядається стандартом як ключова ланка при постановці цілей в області ІБ [11].

Для того щоб бути сертифікованою по ІSО/ІЕC 27001, організація повинна довести, що в неї існують процедури по ідентифікації законів та нормативних актів, що стосуються її з погляду захисту інформації. Крім того, у неї повинна існувати програма по дотриманню цих нормативних вимог. Тоді сертифікація по ІSО/ІЕC 27001, якщо вона проведена належним чином, гарантувала б, що організація на ділі дотримує всі законодавчі та нормативні вимоги, які регулюють її діяльність [11].

Додаток А стандарту ІSО/ІЕC 27001 містить перелік заходів контролю, які повинні бути впроваджені в організації, яка бажає пройти сертифікацію. Однак, слід зазначити, не всі заходи контролю з даного списку обов’язково повинні бути впроваджені, якщо з цього питання існує документально підтверджене рішення керівництва, засноване на оцінці ризиків. Багато компаній використовують ІSО/ІЕC 27001 як засіб самооцінки, оскільки методик по проведенню оцінки безпеки недостатньо. Деякі компанії прагнуть пройти офіційний сертифікаційний аудит в акредитованих незалежних аудиторських організаціях [11].

Як етичний хакер, потрібно бути в курсі законів і як вони регулюють вашу діяльність. Незнання або відсутність розуміння закону не звільняють від відповідальності. Ви також повинні дотримуватися Кодексу етики, який визначено Радою ЄС. Проте недотриманя цього Кодексу не веде до кримінальної чи іншої відповідальності.

Нижче наведено короткий опис законів, нормативних актів та директив, які тим чи іншим чином пов’язані з тестуванням на проникнення:

* 1973 рік – Кодекс Сполучених Штатів справедливої ​​інформаційної практики регулює технічне обслуговування та зберігання персональної інформації базами даних, такими як медичні та кредитні бюро.
* 1974 рік - Закон Сполучених Штатів про конфіденційність регулює обробку особистих даних в уряді США.
* 1984 рік - Медичний закон Сполучених Штатів про комп'ютерні злочини спрямований на незаконне отримання або зміну даних про лікарські засоби.
* 1986 рік (З поправками в 1996 р.) -. Закон Сполучених Штатів про комп'ютерні шахрайства та зловживання, що включає такі питання, як зміна, пошкодження або знищення інформації на федеральному рівні та торгівля нею, якщо це впливає на міждержавну або зовнішню торгівлю.
* 1986 рік - Закон про захист електронних комунікацій, що забороняє прослуховування або перехоплення вмісту повідомлення без розмежування між приватним або публічним системи
* 1996 рік - Закон Кеннеді-Кассебаума про медичне страхування та підзвітність щодо мобільності (HIPAA) (Health Insurance and Portability Accountability Act) (з додатковими вимогами, доданими в грудні 2000 р.) Розглядаються питання конфіденційності особистої медичної інформації в США.
* 1996 рік - Національний закон про захист інформаційної інфраструктури був прийнятий у жовтні 1996 року в рамках публічного права 104-294; Цей акт спрямований на захист конфіденційності, цілісності і доступності даних та систем. Цей акт призначений щоб заохотити інші країни прийняти подібні рамки, тим самим створивши єдиний підхід до вирішення комп'ютерних злочинів в існуючій глобальній інформаційній інфраструктурі.
* Закон 2002 року Сарбейнса-Окслі (SOX або SarBox) - закон, що стосується відповідальності публічних компаній, що мають справу з публічною інформацією.
* 2002 рік - Закон про управління федеральною інформаційною безпекою (FISMA) - закон, призначений щоб захистити безпеку інформації, яка зберігається або обробляється державними системами на федеральному рівні [9].

Далі в Таблиці 1.1 наведені Закони та акти інших держав в інформаційній сфері

Таблиця 1.1

Закони та акти інших держав в інформаційній сфері

|  |  |
| --- | --- |
| Країна | Закон\акт |
| Австралія | The Trade Marks Act 1995  The Copyright Act 1968  Cybercrime Act 2001 |
| Великобританія | The Copyright, Etc. and Trademarks Act 2002  Computer Misuse Act 1990 |
| Країна | Закон\акт |
| Китай | Copyright Law of People`s Republiv of China 2001 |
| Німеччина | Section 202a, Data Espionage,Section 303a. Alternation of Data, Section 303b. Computer Sabotage |
| Італія | Penal Code Article 615 ter |
| Японія | The Trademark Law 1957, Copyright Management Business Law 2000 |
| Канада | Copyright Act 1985, c. C-42, Canadian Criminal Code Section 342.1 |

## 1.4. Висновки по розділу

В даному розділі були розглянуті закони та нормативні документи в сфері кібербезпеки України і світу. На основі проаналізованої правової бази можна сказати, що українське законодавство відстає від законодавства провідних країн світу, проте останнім часом прогрес у даному питанні.

Тестування на проникнення описується лише як рекомендовані дії для державних органів та служб, в той час як в деяких країнах світу пентест є обов’язковою процедурою для всіх організацій.

Також недоліком є те що будь-яке несанкціоноване тестування на проникнення несе за собою кримінальну відповідальність, навіть якщо воно зроблено в корисних цілях та всі результати були передані власникам системи. Незважаючи на недосконалість законів і українських норм, вітчизняні організації звертаються до міжнародних стандартів чи проектів з відкритим кодом в сфері кібербезпеки.

# РОЗДІЛ 2

# ПОСТАНОВКА ЗАВДАННЯ ТА ТЕОРЕТИЧНІ АСПЕКТИ ТЕСТУВАННЯ НА ПРОНИКНЕННЯ ІКМ

### 2.1. Аналіз необхідності тестування на проникнення

Цілком природно, що бізнес сьогодні в першу чергу вкладає ресурси в практичну частину інформаційної безпеки на противагу формального виконання вимог щодо мінімальних засобів захисту. І так само природно його бажання перевірити, наскільки ефективно збудована система безпеки захищає від різноманітних атак і від витоку цінної інформації.

Тестування на проникнення, здійснюване за допомогою тестів на подолання захисту ІКМ (penetration testing — pentest, пентест),— доволі популярна в усьому світі послуга аудиту, що дозволяє [12] :

* виявляти недоліки у сфері інформаційної безпеки (ІБ) з погляду стороннього спостерігача, не враховані при розробці політики безпеки;
* розкривати внутрішні і зовнішні спроби проникнення в інформаційну систему (ІС) й запобігати їм.

Тестування на проникнення зводиться до реалізації санкціонованої спроби обійти наявний комплекс засобів захисту ІС. У процесі його проведення аудитор відіграє роль зловмисника, мотивованого на порушення ІБ мережі замовника. Як правило, інтенсивній перевірці підлягають технічні засоби захисту корпоративної мережі. Проте залежно від поставлених умов об’єктом оцінювання можуть бути й інші, наприклад соціотехнічні аспекти безпеки (рівень поінформованості користувачів тощо).

Тестування на проникнення має допомогти користувачеві з’ясувати, по- перше, чи всі положення політики безпеки досягають своїх цілей і використовуються згідно з попереднім задумом, а по-друге, чи існують прогалини в політиці безпеки, якими може скористатись зловмисник для досягнення своїх цілей. Таке тестування може проводитись як у складі аудиту на відповідність стандартам, так і у вигляді самостійної роботи. Наприклад, під час проведення аудиту на відповідність стандарту ISO 17799 елементи pentest можуть використовуватися для оцінювання ефективності реалізації таких захисних механізмів, як захист від шкідливого коду, мережна безпека і т.д. У вигляді самостійної роботи тести можуть мати на меті:

* обґрунтування необхідності проведення робіт із підвищення захищеності;
* визначення незалежної оцінки рівня безпеки інформаційної системи.

У першому випадку замовником тестування виступають, як правило, далекоглядні керівники ІТ підрозділів або підрозділів ІБ. Здобуті результати демонструють вищому керівництву недоліки існуючої системи управління інформаційною безпекою (СУІБ). Оскільки тестування на проникнення — порівняно недорогий вид послуг, то його найчастіше можна провести за рахунок бюджету відповідного підрозділу. У другому випадку роботи проводяться або після впровадження комплексу засобів захисту (КЗЗ), або перед уведенням тестованої системи в промислову експлуатацію. У цьому разі результати тестування дозволяють реально оцінити залишкові ризики, а можливо і виявити приховані недоліки в системі.

Типові тести такого роду спираються на використання:

* методів соціальної інженерії, які дають змогу виявляти рівень поінформованості користувачів;
* технічних засобів (наприклад, нового web-інтерфейсу), що зазнають випробування на злам.

Плануючи тестування на проникнення, необхідно визначити межі та режим його проведення. Відповідні роботи можуть виконуватися з попереднім повідомленням персоналу (системних і мережних адміністраторів) або без такого попередження. Якщо користувачі й адміністратори не знають про «злам», що готується, керівництво має добру нагоду оцінити ефективність чинних механізмів виявлення та розслідування комп’ютерних інцидентів. Проте «прихований» тест підвищує ймовірність виникнення відмови через помилку експерта або не зовсім коректне настроювання серверів і мережного обладнання.

Найбільш близький до реальних дій зловмисників так званий комплексний тест на проникнення. Використовуючи різні технічні й соціоінженерні прийоми, аудитори в ході його проведення намагатимуться обійти наявні захисні механізми, аби виконати поставлені Замовником завдання (підвищення привілеїв, отримання доступу до конфіденційної інформації, модифікація даних із СУБД тощо) [13-16].

Усі спроби проникнення мають контролювати обидві сторони — як «зловмисник», так і «клієнт». Це допоможе протестувати систему набагато ефективніше.

За результатами роботи складається звіт, що містить:

* методику проведення тестування;
* висновки для керівництва, що включають у себе загальну оцінку рівня захищеності;
* опис виявлених недоліків СУІБ;
* опис ходу тестування з інформацією про всі виявлені уразливості та результати їх експлуатації;
* рекомендації з усунення виявлених уразливостей.

Логічним продовженням тестування на проникнення можуть бути роботи з побудови комплексної системи управління рівнем захищеності, проведення моніторингу захищеності периметра корпоративної мережі, розробка програми підвищення поінформованості з питань ІБ та впровадження системи управління ІБ.

Отже, завдання тестування на проникнення в тому, щоб відповісти на такі питання:

* Що може побачити зловмисник у цільовій системі?
* Що зловмисник може зробити з цією інформацією?
* Хто-небудь може помітити спроби вторгнення?
* Чи всі компоненти інформаційної системи адекватно захищені та оновлені?
* Скільки зусиль, часу і грошей потрібно, щоб побудувати адекватний захист?
* Чи заходи щодо інформаційної безпеки відповідають галузевим та юридичним стандартам?

Тестування на проникнення, що імітує дії зловмисника згідно з поставленими цілями й неодмінно досвідченим фахівцем, практично завжди результативне. Адже саме воно дає змогу виявляти реальні проблеми інформаційної безпеки в компанії й привертати до них увагу керівництва. Адже про наявну якість захисту найпереконливіше може засвідчити демонстрація успішного доступу до інформації, що вважається добре захищеною, або демонстрація повного контролю над особистими комп’ютерами відповідальних співробітників.

### 2.2. Вразливості, які може виявити тестування на проникнення

В реальному світі не все як в теорії і вразливості розподіляються далеко не рівномірно по всім вузлам і аспектам комп’ютерної мережі. Найчастіше вразливості знаходяться в місцях, ігнорувати які, здавалося б, ніхто не буде, але все ж таки. [17]

Загальне визначення вразливості - нездатність системи протистояти реалізації певної загрози або сукупності загроз. Тобто, це певні недоліки в комп'ютерній системі, завдяки яким можна навмисно порушити її цілісність і викликати неправильну роботу. [18]

Уразливість може виникати в результаті допущених помилок програмування, недоліків, допущених при проектуванні системи, ненадійних паролів, вірусів та інших шкідливих програм, скриптових і SQL-ін'єкцій. Деякі уразливості відомі тільки теоретично, інші ж активно використовуються і мають відомі експлойти.

Для мережевого периметра характерним недоліком є надмірне і безконтрольне кількість відкритих сервісів (портів). Це веде до збільшення площі атаки, іншими словами хакеру-зловмисникові буде легше знайти слабку ланку в периметрі захисту. Коли багато сервісів, персонал просто не встигає стежити за ними, вчасно встановлювати оновлення і патчі безпеки, правильно настроювати сервера і програми. У мережі постійно з'являється інформація про свіжі вразливості, якщо хакер скористається нею раніше, ніж адміністратор «накотить» оновлення - то злом гарантований.

Також часто «пісочниці» (тестові сервери і додатки з мінімальними налаштуваннями безпеки) розміщують в тому ж сегменті мережі, де працюють критичні сервіси, що дає змогу легко отримати над ними контроль.

Ще один типовий недолік - відсутність системи управління інформаційною безпекою. Для надійного захисту ІТ інфраструктури повинні функціонувати такі процеси як - установка оновлень, моніторинг подій безпеки, управління і контроль за змінами, реакція на інциденти безпеки, резервування та інші. Без них дуже складно контролювати ситуацію. Також важливо зрозуміти, що 100% захисту комп'ютера, підключеного до мережі Інтернет не існує, компанія повинна знати як реагувати і як можна знизити шкоду у випадку, якщо злом системи вже стався.

Тест на проникнення виявляє як технічні «діри в захисті», так і неорганізованість самої системи захисту.

Для додатків типовим недоліком є відсутність стандартизації при розробці коду і вимог до безпеки. Будь-який програміст - в першу чергу людина, а людям властиві помилки, і коли немає єдиних вимог, ці помилки обов'язково з'являться у вашому додатку. Серед «популярних» помилок - відсутність перевірки даних, які користувач посилає в додаток, неправильне розмежування прав доступу користувачів, логічні недоліки в архітектурі коду та інші. Все це веде до можливості провести атаки типу переповнення буфера, обхід автентифікації, обхід обмежень додатків (авторизації), вивід сервера з ладу (DoS) та інших.

Для Web сайту уразливості ідентичні тим, що були згадані вище для додатків, так як сайт також є в певному сенсі додатком. У той же час атаки на сайти мають свою специфіку, характерну саме для Веб додатків. Найпопулярніші види атак - SQL injection, Cross Site Scripting, Cross-Site Request Forgery, обхід автентифікації та авторизації, вставки виконуваного коду або файлів на сервер тощо.

Соціальна інженерія - найлегший спосіб злому на сьогодні, адже найбільша вразливість – це людина, а саме: її зайва довірливість, страх і лінь. Вміло використовуючи ці «людські недоліки», можна отримати доступ до дуже критичних даних. Найпоширенішою атакою тут є фішинг, він проводяться у вигляді поштової розсилки або смс, а ошукані люди здійснюють діяння, необхідне для злому. Зловмиснику досить, щоб співробітник:

* завантажив і запустив файл або документ відкрив у листі
* перейшов по посиланню
* ввів облікові дані в підставну систему як дві краплі води схожу на реальну, що використовується в компанії. Наприклад, корпоративну пошту.

Тест на проникнення дозволяє визначити наскільки співробітники компанії можуть ці атаки розпізнати і адекватно на них зреагувати. А також наскільки швидко і правильно реагує служба безпеки, якщо такий злом вже відбувся.

Залежно від поточних потреб і завдань клієнта можливі три різні за глибиною і складністю виконуваних перевірок рівні тестування на проникнення.

Тестування 1-го рівня виконується за допомогою спеціальних програмних засобів (сканерів вразливостей, типових наборів експлойтів і т. ін.). Тестування 2-го і 3-го рівнів здійснюється вручну, дозволяючи виявити уразливості, які не можуть бути знайдені в ході тестування 1-го рівня. На цьому етапі можуть бути виявлені, зокрема, уразливості, пов’язані з конфігурацією конкретних систем клієнта, а також нові уразливості в додатках, наприклад так звані уразливості нульового дня— zero day. Після проведення тестування вручну виконавець аналізує результати, порівнюючи їх з результатами тестування 1-го рівня. Можна також провести аналіз пропущених вразливостей (FalseNegatives — FN), тобто вразливостей, які не вдалося знайти. Після виявлення таких вразливостей вибірково (за домовленістю із Замовником) здійснюються заходи з їх експлуатації (аби продемонструвати Замовникові можливі наслідки). Окремо слід обговорити процедуру виконання атак, які можуть істотно вплинути на функціонування систем і процесів Замовника, наприклад DoS атак.

### 2.3. Основні поняття і терміни хакінгу та тестування на проникнення

Етичні хакери беруть участь у санкціонованому зломі, тобто зломі з дозволу власника системи. У світі етичного хакінгу, частіше використовують слово pentester, що є скороченням від penetration tester. Робота пентестера заключається у проходженні в систему але з корисними цілями.

Далі розглядаються терміни, які використовуються в сфері етичного хакінгу.

Hack Value (Цінність злому). Цей термін описує те, наскільки певна ціль (організація, мережа, веб-сайт тощо) є цінним і привабливим для хакерів. Тобто, наскільки цінна інформація там зберігається чи обробляється.

Target of Evaluation (TOE) (Ціль оцінки) - це система чи ресурс, що має оцінюватися на вразливості. TOE повинна бути описана в договорі з клієнтом.

Attack (Атака) - це процес активного та цілеспрямованого впливу на ціль.

Exploit - це чітко визначений спосіб порушити безпеку системи.

Payload – це частина Exploit, яка безпосередньо виконує шкідливі дії.

Атака нульового дня (Zero Day). Цей термін описує загрозу або вразливість, яка досі не була відома розробникам. У багатьох випадках така вразливість вважається серйозною проблемою.

Захищеність (Security). Це стан в середовищі, в якому відбуваються лише дозволені дії і процеси.

Загроза (Threat) – це потенційне порушення безпеки.

Вразливість (Vulnerability) – це слабке (незахищене) місце у системі, яке може бути використане в поганих цілях.

Daisy Chaining - це процес виконання декількох атак одну за одною таким чином, що результати попередньї використовуються в наступній. [19]

Пентестери використовують навички і стратегію дій як у хакерів, одночасно володіючи знаннями в сфери захисту інформації. В цьому полягає основна ідея тестування на проникнення – оцінити систему з обох сторін, і допомогти усунути вразливості.

Першим і основним пунктом тестування на проникнення є підписання юридично цінного договору між замовником (власником цілі оцінки) і виконавцем (пентестером). В цьому договорі повинні описуватися всі бажані з боку замовника дії, ресурси, які можна перевіряти та типи атак, які дозволено проводити. Також описується тип тестування, який може бути одним з наступних:

Чорний ящик (Black Box) - Тип випробувань, в якому пентестер володіє мінімальними відомостями про ціль, або взагалі не володіє ними. Таке тестування найбільш добре імітує реальну ситуацію злому системи ззовні [20].

Сірий ящик (Gray box) - Форма тестування, де знання, надані учасникам тестування, обмежені. В цьому типі тесту, тестер отримує такі знання, як IP-адреси, інформацію про операційні системи та мережеве середовище, але ця інформація обмежена. Цей тип тесту буде симулювати дії інсайдера [21].

Білий ящик (White box) - Форма тестування, в якій інформація, надана тестувальнику, повністю описує систему. Цей тип тесту як правило, здійснюється всередині компанії, власними силами (тобто командою тестерів) [22].

В сфері оцінювання захищеності систем існують такі послуги як: Аудит безпеки, Оцінка вразливостей і Тестування на проникнення. Аудит безпеки лише перевіряє чи дотримується організація певного набору стандартів, політик безпеки і процедур. Оцінка вразлвостей фокусується на визначенні вразливостей в інформаційній системі, але не забезпечує визначення можливості використання цієї вразливості і оцінки можливих збитків. В свою чергу тестування на проникнення це методологічний підхід до оцінки захищеності системи, який показує як вразливості можуть бути використані атакуючою стороною

### 2.4. Аналіз загальної методики хакінгу

Методика злому відноситься до покрокового підходу, який використовується агресором для атаки на комп'ютерні мережі. Не існує конкретного підходу, який використовується усіма хакерами. Зазвичай хакери відходять від правил і стандартних методик, щоб уникнути виявлення злому або щоб напевне дістатися до потрібної інформації невідомими для служби безпеки методами і шляхами. Проте, незважаючи на те, які будуть використані техніки та атаки, на кожному з етапів, самі етапи злому є стандартними.

Пошук слідів, збір інформації. (Footprinting and Reconnaissance). На цьому етапі відбувається пасивний збір інформації без безпосереднього контакту з цільовою системою. Під пасивним збором інформції розуміється пошук в Google, Whois запити, підслуховування працівників компанії тощо.

Сканування (scanning) - це етап, в якому на основі отриманої інформації на попередньому етапі збираються дані вже з впливом на систему. На даному етапі можна отримати інформацію про ip-адреси, включені хости, сервіси, топологію мережі тощо. Сканування включає в себе виконання технік, таких як ping sweeps, сканування портів та спостереження за об'єктами.

Перебір зібраної інформації (Enumeration) - це наступна фаза, де визначається більш детальна інформація про ціль, а саме: типи і версії операційних систем, встановлені патчі, імена користувачів, груп тощо.

Злам системи (System hacking)слідує за перебором. На цьому етапі планується і здійснюється атака на систему на основі зібраної на попередніх етапах інформації.

Ескалація привілеїв (Escalation of privilege) - це фаза хакінгу, де відбувається перехід до користувачів і процесів з різними рівнями прав доступу. Тобто перехід від звичайного користувача до адміністратора чи рута.

Прикриття слідів (Covering tracks)- фаза, в якій зачищаються сліди перебування зломщика в системі: очищуються логи, видаляються файли, процеси тощо.

Встановлення бекдорів (Planting of backdoors) означає встановлення таємного з’єднання з системою, так званих задніх дверей. Робиться це для того щоб можна було в будь-який момент повернутися в систему для крадіжки інформації або запуску атаки.

### 2.4.1. Footprinting and Reconnaissance

Пошук слідів - це процес дослідження і збору інформації про потенційну жертву з інтенсивним пошуком шляхів атаки. На цьому етапі потрібно зібрати якнайбільше важливої інформації з загальнодоступних ресурсів, яку пізніше можна використати для проведення атаки на мережу чи соціальної інженерії.

Методологія збору інформації - це покроковий процес, який охоплює всі можливі джерела важливої інформації. Вона включає в себе такі кроки: збір інформації за допомогою пошукових систем, збір інформації використовуючи розширені Google Hacking техніки, збір інформації з соціальних мереж, збір інформації з вебсайтів, збір інформації за допомогою електронної пошти, збір інформації з сервісів WHOIS та DNS, збір інформації про мережу, збір інформації за допомогою соціальної інженерії. Далі розберемо детальніше кожен крок.

Збір інформації за допомогою пошукових систем. На цьому кроці атакуючий використовує пошукові системи для знаходження інформації про ціль, такої як:

* відомості про працівників, керівництво, структуру організації, номера телефонів, фотографії, адреси електронної пошти тощо, використовуючи сервіси Wikipedia, Googl, Яндекс, Bing, соціальні мережі Facebook, Twitter, Вконтакті, Instagram, сервіси пошуку людей AnyWho, PeopleSmart, PrivateEye, сайти пошуку роботи LinkedIn, robota ua, head hunter, Work.ua, фінансові сервіси Google Finance, Yahoo! Finance, форуми, блоги та інші.
* публічні і закриті веб-сайти за допомогою Google search engine, Bing, Netcraft тощо.
* інформація про операційні системи використовуючи сервіс Netcraft, SHODAN.
* географічні дані з сервісів Google Earth, Google Maps, Wikimapia, National Geographic Maps, Yahoo Maps, Bing Maps, Яндекс карти.

Збір інформації використовуючи розширені Google Hacking техніки. Синтаксис пошуку Google допомагає створювати складні запити для більш точного і детальнішого пошуку. Для цього використовуються спеціальні оператори [23] :

* [cache:] Показує веб-сторінки, збережені в кеші Google.
* [link:] Показує список веб-сайтів, що мають посилання на визначені веб-сторінки.
* [site:] Обмежує результати на ці веб-сайти в даному домені.
* [intitle:] Обмежує результат до документів, що включають шукані слова в заголовку.
* [inurl:] Обмежує результат до документів, що включають шукані слова в посиланні.

Збір інформації з соціальних мереж. На цьому кроці вже деяка інформація з соціальних мереж, але тут проводиться більш глибокий пошук у всіх соціальних мережах. Часто використовуються фальшиві профілі, застосовується соціальна інженерія, атакуючий вступає у спільноти та групи, в яких є жертва, для отримання більш важливої і конфеденційної інформації.

Збір інформації з вебсайтів проводиться шляхом моніторингу і аналізу веб-сайту жертви. Дослідження сайту може дати таку інформацію як: ПЗ, що використовується та їх версія, інформацію про операційну систему, суб дерикторії, ім’я файлів, баз даних, запитів, скриптових платформ.

Тут використовуються такі програми: Burp Suite, Zaproxy, Paros Proxy, Firebug або різноманітні веб павуки. Також застосовуються техніки дослідження HTML коду та кукі, дзеркалювання веб-сайтів за допомогою програм BlackWidow, GNU Wget тощо.

Збір інформації за допомогою електронної пошти. Основним джерелом інформації в електронних листах є їх заголовки, які містять такі дані як: ip-адрес відправника, поштовий сервер відправника, дата і час, яка була видана поштовим сервером відправника, систему ідентифікації поштового сервера, повне ім’я відправника. Також використовуються сервіси гео-відслідковування листів eMailTrackerPro, PoliteMail, Email Lookup тощо.

Збір інформації за допомогою WHOIS та DNS сервісів. Бази даних WHOIS підтримуються регіональними Інтернет регістрами та містять персональну інформацію про власника домену. Запити WHOIS повертають такі дані: доменне ім’я, контактні дані власника домену, доменне ім’я сервера, мережевий діапазон, час і дата створення та модифікацій. Сервіси та програми для створення WHOIS запитів: MX Toolbox , SmartWhois, Whois online, LanWhois, CallerIP, Whois тощо.

DNS записи містять окрім імені та ip-адреси важливу інформацію про розміщення і тип сервера. Основними сервісами для вилучення інформації з DNS записів є Domain Dossier, DNS Lookup, MX Toolbox.

Збір інформації про мережу означає побудову топології. Це можна зробити за допомогою ARIN whois database search або інших регіональних Інтернет реєстрів. Не менш важливою є технологія Traceroute, яка працює на концепті ICMP протоколу і використовує TTL поле заголовку ICMP пакета щоб визначити маршрутизатори на шляху до цілі. За допомогою traceroute можна дізнатися інформацію про топологію мережі, довірені маршрутизатори та розміщення фаєрволів. Програми, які для цього використовуються: Path Analyzer Pro, VisualRoute.

Збір інформації за допомогою соціальної інженерії відбувається шляхом використання людської натури, якій притаманна довірливість, страх та байдужість. Для отримання важливої інформації використовуються такі техніки соціальної інженерії: підслуховування (Eavesdropping), підглядання (Shoulder Surfing) та аналіз сміття (Dumpster Diving).

Для збору інформації вцілому використовуються такі широкоспеціалізовані програми як: Maltego, Recon-ng, FOCA.

### 2.4.2. Scanning

Сканування мережі включає в себе набір процедур виявлення хостів, портів і сервісів в мережі. Сканування є одним з компонентів збору інформації, який допомагає створити профіль цільової мережі.

Цілями сканування мережі є: дослідження включених хостів, ip-адрес і відкритих портів; дослідження операційних систем та архітектур системи; дослідження сервісів на хостах; виявлення вразливостей на хостах.

Для керування TCP з’єднанням використовуються Прапори TCP комунікації: URG (пакет повинен виконатися без черги), PSH (негайно надіслати всі пакети з буфера), FIN (закінчити з’єднання), ACK (підтверджує отримання пакету), RST (розірвати з’єднання), SYN (ініціює з’єднання між хостами) [24].

Процес комутації відбувається шляхом так званого трьохразового рукостискання.

Більшість технік сканування базується на надсиланні різноманітних модифікованих пакетів та аналізі відповідей від хостів (програма Colasoft Packet Builder, Nmap).

Процес сканування складається з декількох основних частин: сканування «живих» систем, перевірка відкритих портів, збирання банерів, сканування вразливостей. Найпопулярнішою мультизадачною програмую для сканування є Nmap або її адаптація з графічним інтерфейсом ZenMap.

Сканування «живих» систем. Основними техніками на цьому кроці є:

* ICMP Scanning, яка заключається у відправлені ICMP ECHO запиту до хоста і якщо він «живий», то повернеться ICMP ECHO відповідь. Цей тип сканування використовується для визначення місця активних пристроїв та чи проходить ICMP запит через фаєрволи, тобто детектування фаєрволів. Програмою, за допомогою якої це можна зробити є Nmap, Hping2/Hping3 [25].
* Ping Sweep, що використовує техніку ICMP Scanning, проте не для 1 пристрою, а одразу для цілої підмережі, або вибраного списку ip-адрес. Тобто одночасно відсилає ICMP ECHO запит до багатьох девайсів. Програмою, за допомогою якої це можна зробити є Nmap, Engry IP Scanner, SolarWinds Engineer`s Toolset, Colasoft Ping Tool тощо.

Перевірка відкритих портів.

Першим на цьому етапі є SSDP Scanning, яке дозволяє отримати інформацію про plug and play пристрої, що доступні в мережі. Протокол SSDP працює в зв’язку з UPnP, вразливості якого дозволяють проводити атаки типу переповнення буферу чи DoS.

Наступні техніки можна розділити на сканування мережних сервісів TCP і сканування мережних сервісів UDP. Детальніше зображено на Рисунку 2.1 .

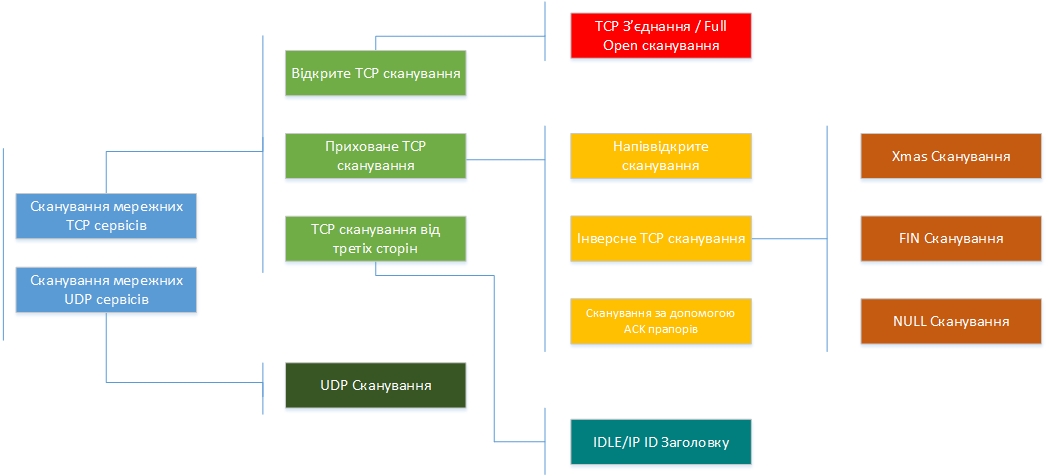


Рис. 2.1 Техніки сканування

Сканування TCP З’єднанням. Ця техніка базується на проведенні повного з’єднання з хостом шляхом трьохразового рукостискання, після чого посилається RST пакет, який розриває з'єднання. Якщо цей процей пройшов успішно, значить що порт відкритий.

Приховане, напіввідкрите сканування. Приховане сканування використовує розірвання TCP з’єднання до того як трьохразове рукостискання повністю завершиться. Процес прихованого сканування включає: надсилання SYN пакету, і якщо порт відкритий у відповідь прийде SYN/ACK , на що надсилається RST.

Інверсне сканування. Атакуючий надсилає пакети з прапорами FIN, URG, PSH або без прапорів. Якщо відповідь не приходить, це означає, що порт відкритий, якщо ж прийде RST/ACK – порт закритий.

При UDP скануванні, немаєтрьох разового рукостискання, тому, якщо не приходить відповіді на надісланий пакет, це означає, що порт відкритий.

Збирання банерів. Це метод призначений для того, щоб визначити які операційні системи встановлені на цільовій системі. Є два типи збору банерів:

* активний – спеціальні модифіковані пакети надсилаються до віддаленої ОС і записується відповідь. Потім ця відповідь порівнюється з базою даних і визначається встановлена операційна система.
* пасивний – це збирання інформації, без активної дії на системи. Робити це можна з повідомлень про помилку, перехвату мережевого трафіку тощо.

Сканування на вразливості. Таке сканування визначає вразливості чи слабкі місця в системі і мережі. Сканування на вразливості проводиться автоматичними програмами, в яких міститься база даних усіх відомих вразливостей. Такими програмами є: Nessus, GFI LanGuadr, Qualys FreeScan.

Після процесу сканування малюється мережева діаграма, яка містить логічну і фізичну топологію мережі, розміщення на ній пристроїв, фаєрволів, відомості про операційні системи, сервіси і т.д.

### 2.4.3. Enumeration

На етапі перебору атакуючий створює активне з’єднання з системою і здійснює спрямовані запити, щоб отримати більше інформації. Видобута інформація використовується для ідентифікації точок атаки на систему. Перебір проводиться безпосередньо в середині мережі цілі.

Перебір проводиться з такою інформацією як: мережеві ресурси, ресурси з спільним доступом, таблиці маршрутизації, настройки аудиту і сервісів, SNMP і DNS деталі, імена хостів, користувачі і групи, додатки.

Деякі техніки перебору: вилучити імена користувачів, використовуючи ідентифікатори електронної пошти; спробувати використати паролі за замовчуванням; вилучити імена користувачів, використовуючи SNMP; провести повний перебір Active Directory; вилучити імена груп з Windows;

Найбільш популярними є NetBIOS, Net View, SNMP та інші перебори.

Ім’я NetBIOS це унікальний шістнадцяти символьний ASCII рядок, що використовується для ідентифікації мережних пристроїв поверх стеку TCP/IP. П’ятнадцять символів це ім’я, а шістнадцятий зарезервований для назви типу запису. Отже, використовуючи NetBIOS перебір можна дізнатися список комп’ютерів, що належать домену, список спільних папок або пристроїв, політики і паролі. Для цього використовується утиліта Nbtstat, програми SuperScan, Hyena, Winfingerprint, NetBIOS Enumerator тощо [26].

Утиліта Net View використовується для отримання списку всіх спільних ресурсів віддалених хостів або робочих груп.

SNMP перебір – це процес перебору профілів користувачів або пристроїв на цільовій системі використовуючи протокол SNMP. SNMP отримує два паролі для доступу і конфігурування агентів: публічний і приватний. Ці паролі видаються хостам в мережі за замовчуванням і, якщо користувач не змінив їх, атакуючий може використати відомі паролі для отримання інформації такої як: імена хостів, інформація про маршрутизатори, девайси, спільні ресурси, ARP таблиці, таблиці маршрутизації тощо.

Іншими типами перебору які можна використати є: LDAP, NTP, SMTP та DNS перебори.

### 2.4.4. System hacking

В широкому значені злам системи означає несанкціонований вхід в неї, ескалація привілеїв, виконання зловмисних дій, запуск шкідливого програмного забезпечення, та прикриття слідів. Проте в значені однієї з фаз хакінгу розглядатиметься лише злам паролів, тобто вхід в систему та виконання шкідливого ПЗ.

Для несанкціонованого входу в систему використовуються різноманітні техніки атак на паролі. Ці атаки проводяться з метою розкриття паролю і подальшого доступу до системи.

Типи атак на паролі:

* не електронні атаки (підглядання, соціальна інженерія, перебір сміття).
* активні онлайн атаки (перебір по словнику або повний перебір, хеш ін’єкції, фішинг, трояни, шпигуни, кейлогери, вгадування паролів).
* пасивні онлайн атаки (перехоплення трафіку, Man-in-the-Middle)
* офлайн атаки (наперед обчислені хеші (Rainbow tables), розподілена мережа).

Атака повного перебору означає перебір усіх можливих значень пароля, комбінацій символів і спроба використання кожної комбінації. Така атака вимагає дуже високих обчислювальних потужностей, тому доцільніше створювати словники можливих паролів на основі зібраної на попередніх етапах інформації, або задавати певні правила для перебору паролів, такі як ключові слова, порядок слів тощо. Також великою вразливістю є паролі за замовчуванням, бази яких є в інтернеті, тому не складає проблем, знаючи модель маршрутизатора, чи інформацію про систему, підібрати пароль.

Іншим шляхом як дізнатися пароль є зараження комп’ютера жертви трояном або кейлогером. Ці утиліти працюють в фоновому режимі, невидимо для користувача і перехоплюють усю інформацію, а сае таку як логіни і паролі.

Також для отримання паролю використовуються техніки перехоплення трафіку, тобто «слухання» усього трафіку в мережі, серед якого можуть бути логіни і паролі. Після чого атакуючий використовує ці дані для входу в систему.

Досить популярними є різноманітні атаки з використанням хешів паролів. Однією з таких є атака з використанням Rainbow Table – це таблиця наперед обчислених хешів, подібно до словників, але яка містить співвідношення хешо і відкритого пароля. Оскільки в системі паролі зберігаються і передаються в хешованому виді, то отриманий хеш порівнюється зі значенями в таблиці і таким чином знаходиться необхідний пароль. Такі таблиці можна згенерувати за допомогою програм rtgen, Winrtgen.

Ще одним способом злому паролю є зчитування SAM файлу з Windows [27] , в якому паролі зберігаються в виді, зображеному на Рисунку 2.2 .

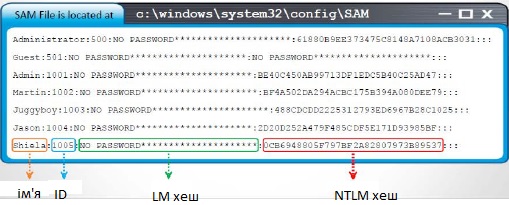


Рис. 2.2 Вміст SAM файлу

Для зчитування інформації з цих файлів та для злому паролів можна використати такі програми: pwdump7, fgdump, L0phtCrack, Ophcrack, Cain & Abel, RainbowCrack, John the Ripper.

Після того як хакер зламує пароль і входить в систему він має права на виконання програм, що означає що ціль досягнута і може бути вкрадена інформація або заподіяна різноманітна шкода.

### 2.4.5. Escalation of privilege

У випадку, якщо атакуючий увійшов в систему використовуючи профіль без прав адміністратора і не може виконати деякі програми або немає доступу до певної інформації, наступним кроком злому буде ескалація привілеїв. Під ескалацією привілеїв розуміється отримання адміністраторських прав або прав інших користувачів, використовуючи недоліки в розробці, помилки, баги чи неправильну конфігурацію програмного забезпечення або операційної системи.

Ескалація привілеїв шляхом використання викрадення (підроблення) DLL.

Більшість додатків Windows не використовують повноцінний шлях під час завантаження зовнішньої бібліотеки DLL замість того вони шукають каталог, з якого вони були завантажені спочатку. Тому якщо атакуючий підставив свою DLL в дерикторію програми вона замінить реальну бібліотеку.

Для ескалації привілеїв можна використовувати програму Active@ Password Changer, що дає змогу локально скидати адміністративні та користувацькі паролі.

### 2.4.6. Covering tracks

Після зламу системи і отримання прав адміністратора наступним кроком буде прикриття слідів для того щоб уникнути виявлення несанкціонованого доступу.

Для цього використовуються такі техніки:

* Виключення аудиту. Як тільки хакер отримав права адміністратора він виключає аудит, а після виконання всіх запланованих дій, аудит включається знову. Зробити це можна за допомогою програми Auditpol.
* Очищення Log файлів. Для того щоб зловмисні дії не були помічені, атакуючий очищує логи програмою clearlogs для Windows або через meterpreter shell в Linux. Також це можна зробити вручну, знайшовши ці файли в системі.
* Очищення онлайн слідів. Для цього потрібно очистити «найбільш часто використовувані», кукі, кеш, історію та вимкнути автозаповнення. Це можна зробити вручну, або використати програми CCleaner, MRU-Blaster, Wipe.

### 2.4.7. Planting of backdoors

Існує багато способів встановлення бекдорів на систему. Навіть в самі системі є такі можливості для зловмисника, а саме: PsTools suite. Цей пакет включає змішаний пакет утиліт, призначений для полегшення адміністрування системи.

Серед цих інструментів є PsExec, який призначений для запуску команд в інтерактивному режимі або інтерактивно на віддаленій системі. Для роботи, PsExec потрібно лише скопіювати в папку в локальній системі та запускати з відповідними налаштуваннями.

Наступний варіант полягає в тому, що зловмисник може запустити на віддаленій системі трояни, руткіти та блакитні двері, що також дає змогу віддаленого керування.

Іншими утилітами, які можуть виявитися корисними для приєднання до системи віддалено, є: PDQ Deploy, Active Directory, RemoteExec, DameWare, Netcat

### 2.5. Методика тестування на проникнення

Тестування на проникнення складається з трьох фаз:

* Передатакувальна фаза: підготовка і планування, розробка методології, збір інформації.
* Атакувальна фаза: тестування периметру, атака цілі, ескалація привілеїв, виконання, імплантація.
* Постатакувальна фаза: звітування, очищення слідів, закриття проекту.

Методика тестування на проникнення має такі складові:

* Збір інформації
* Аналіз вразливостей
* Зовнішнє тестування на проникнення
* Внутрішнє тестування на проникнення
* Тестування на проникнення фаєрволів
* Тестування на проникнення IDS
* Тестування злому паролів
* Соціальна інженерія
* Тестування на проникнення веб додатків
* SQL тестування
* Тестування маршрутизаторів і комутаторів
* Тестування безпровідних мереж
* Тестування на DoS
* Тестування можливості крадіжки техніки з важливою інформацією
* Тестування захищеності коду
* Тестування фізичної безпеки
* Тестування баз даних
* Тестування VoIP
* Тестування VPN
* Тестування хмарних технологій
* Тестування віртуальних машин
* Детектування вірусів і троянів
* Тестування системи керування логами і аудитом
* Перевірка цілісності файлів
* Тестування мобільних пристроїв
* Тестування на проникнення телекомунікацій
* Тестування електронної пошти
* Тестування оновлень безпеки в системі
* Тестування на предмет витоку інформації
* SAP тестування на проникнення.

Всі ці компоненти разом складають комплексне тестування на проникнення. Проте мало хто замовляє саме комплексне тестування, найчастіше вибираються лише деякі види тестування, які потрібні для тої чи іншої організації.

### 2.6. Аналіз відомих методик оцінки захищеності

На сьогоднішній день існує велика кількість готових покрокових методик тестування захищеності. Деякі з них безплатні, деякі комерційні, проте всі вони розроблені з метою певної стандартизації процедури тестування на проникнення і відповідно встановлення вимог до захищеності. [28]

OWASP. Відкритий проект з безпеки веб-застосунків (Open Web Application Security Project) (OWASP) — онлайн-спільнота, яка створює вільно доступні статті, методології, документацію, інструменти та технології у галузі безпеки веб-застосунків. Їхнім продуктом є проект з відкритим кодом по безпеці додатків, який допомагає організаціям створювати і покращувати програми, додатки і розробляє документальну базу по безпеці веб додатків. [OWASP](https://uk.wikipedia.org/wiki/OWASP) Testing Guide є методологією перевірки на наявність вразливостей [веб-застосунків](https://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%B7%D0%B0%D1%81%D1%82%D0%BE%D1%81%D1%83%D0%BD%D0%BE%D0%BA), яка в тому числі може бути використана при тестуванні на проникнення веб-сайтів та систем, які використовують [веб-технології](https://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%82%D0%B5%D1%85%D0%BD%D0%BE%D0%BB%D0%BE%D0%B3%D1%96%D1%97).

OSSTMM (Open Source Security Testing Methodology Manual) являє собою рецензовану методологію проведення тестів безпеки. OSSTMM випадки тестів діляться на п'ять каналів, які в сукупності тестують: управління інформацією та даними, рівні безпекової обізнаності персоналу, рівні управління шахрайством та соціальною інженерією, комп'ютерні та телекомунікаційні мережі, бездротові пристрої, мобільні пристрої, контроль безпеки фізичного доступу, фізичні місця розташування, такі як будівлі, периметри і військові бази.

OSSTMM зосереджується на технічних деталях: які саме елементи повинні бути перевірені, що робити до, під час і після тесту безпеки, як виміряти результат. OSSTMM також відомий своїми правилами ведення “бойових дій” ([англ.](https://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) Rules of Engagement), які визначають для тестувальника і клієнта, як тест повинен правильно протікати, починаючи з заперечення помилкових повідомлень від тестувальників, закінчуючи тим як клієнт може очікувати на отримання звіту. Нові тести для кращої міжнародної практики, закони, правила і етичні проблеми регулярно додаються та оновлюються.

NIST (National Institute of Standards and Technology) обговорює тестування на проникнення у документі SP800-115. Методологія NIST є менш вичерпною ніж OSSTMM, проте вона є кращою для прийняття регулюючими органами. З цієї причини NIST посилається на OSSTMM.

ISSAF (Information Systems Security Assessment Framework) – це проект з відкритим кодом, покликаний допомагати професіоналам з безпеки. Метою ISSAF є знайти, розробити, опублікувати і поширити повну та практичну загальноприйняту систему оцінки безпеки інформаційної системи.

## 2.7. Аналіз алгоритму оцінювання вразливостей CVSS

Одним з найпопулярніших алгоритмом оцінювання є CVSS версії 2 і 3.

Common Vulnerability Scoring System (CVSS) – безплатний і відкритий стандарт промисловості призначений для оцінки вразливостей безпеки комп’ютерних систем. Цей стандарт дозволяє вибирати і пріоретизувати метрики вразливості для найточнішого для даного випадку обрахунку. Вразливіть отримує бал по шкалі від 0 до 10. Поточна версія цього стандарту – 3.1, реліз якої відбувнся в червні 2019 року. Проте дуже популярною залишається версія 2. Ця версія і була взята за основу при розробці мого алгоритму.

Дослідження Національної консультативної ради з питань інфраструктури (NIAC) у 2003/2004 р. Призвели до запуску CVSS версії 1 у лютому 2005 року, метою якої було "розроблено для забезпечення відкритих та загальновизнаних стандартів оцінки вразливості програмного забезпечення". Цей початковий проект не підлягав експертній експертизі чи перегляду з боку інших організацій. У квітні 2005 року NIAC сформував Форум команд реагування на інциденти та безпеку , щоб розвивати CVSS і надалі.

Зворотній зв'язок від постачальників, які використовують CVSS версії 1 у виробництві, припустив, що існували суттєві проблеми з початковою чернетою CVSS версії 1. Робота над CVSS версії 2 розпочалася у квітні 2005 року, остаточна специфікація була запущена у червні 2007 року.

Подальші відгуки призвели до того, що у 2012 році почалася робота над CVSS версії 3.

CVSS версії 2 базується на двох групах мерик. Перша груа – базові (обов’язкові) метрики, друга група – метрики середовища і третя група – часові метрики.

Всі метрики оцінюються балами і в кінці рейтинг виводиться за формулою.

Базова група метрик відображає невід'ємні і фундаментальні характеристики уразливості, які є постійними з плином часу і призначеними для користувача середовищами. До неї входять вектор атаки, складність атаки, аутентифікація, конфіденційність, цілісність, доступність. [29]

Вектор атаки може бути локальним, віддаленим через внутрішню мережу або через глобальну мережу. При цьому чим дальше від цілі зловмисник тим більший бал. [29]

Складність атаки може бути високою, середньою і низькою. Передбачається, що у атакуючого вже є готовий експлойт, і під «складністю» мається на увазі складність його використання. Якщо для успішної експлуатації потрібно лише запустити програму, то очевидно, що складність буде низька. Однак іноді зловмисникові доводиться вдаватися до додаткових дій. Наприклад, якщо він має намір провести фішинг-атаку (атаку, при якій жертва переходить по помилкової посиланням і самостійно вводить конфіденційну інформацію на ресурсі зловмисника, візуально схожий на основний ресурс), то він може використовувати метод соціальної інженерії, і в такому випадку складність експлуатації вважається середньої. Однак соціальна інженерія не завжди приводить до успіху, тому зловмисник може вдатися до перехоплення DNS, атакувавши DNS-сервер. Тоді складність експлуатації буде високою. Чим нижче складність, тим вище числова оцінка. [29]

Показник аутентифікації відображає складності для атаки, пов'язані з необхідністю надання зловмисником облікових даних. У моделі CVSS аутентифікація може зовсім не вимагатися, вимагатися один раз або декілька разів. При цьому, якщо вразливість виявлена ​​в самій системі аутентифікації, то вважається, що аутентифікація не вимагається. Якщо потрібно аутентифікація для доступу в локальну мережу, з якої є вразливе додаток, а в самому додатку аутентифікація не вимагається, то вважається, що для експлуатації уразливості аутентифікація не вимагається. [29]

Наступні три метрики базової групи - вплив на конфіденційність, вплив на цілісність і вплив на доступність - визначають можливі наслідки експлуатації уразливості. У кожній з цих метрик вплив може не надаватися, бути частковим, і бути повним. Вплив на актив вважається частковим, порушена конфіденційність / цілісність тільки певної частини файлів або переривається доступність лише деяких компонент системи. Якщо до системи є фізичний доступ або доступ з root-правами, то вона вважається повністю скомпрометованої. [29]

Часова група метрик відображає характеристики уразливості, які змінюються з часом. До неї входять vожливість використання уразливості, рівень виправлення, ступінь достовірності джерела. [29]

Можливість використання уразливості відображає стан методів експлуатації і доступність коду. Розрізняють стадію теоретичного існування експлойта, стадію існування концепції експлуатації - коли можна провести демонстрацію, але на більшості реальних додатків вона не спрацює, стадію існування сценарію - коли доступний код експлойта і він працює в більшості випадків без змін, і стадію високою небезпеки, якщо експлойт доступний і працює автономно на безлічі пристроїв. Підсумкова оцінка тимчасової групи тим вище, чим простіше використовувати уразливість. [29]

Рівень виправлення відображає стан обробки уразливості. При цьому розрізняють стадію, коли будь-яке рішення недоступно, а також стадії існування рекомендацій, тимчасового рішення і офіційного виправлення. [29]

Ступінь достовірності джерела, який повідомив про уразливість - вона може бути не підтверджена, не доведена і підтверджена вендором ПО. [29]

Група метрик середовища відображає характеристики уразливості, які є релевантними і унікальними для конкретного середовища користувача. До неї входять імовірність нанесення непрямих збитків, щільність цілей

Імовірність нанесення непрямих збитків відображає наскільки сильно можуть постраждати активи, не пов'язані з вразливим додатком. Ця ймовірність може бути низькою, середньо-низькою, середньо-високої і високої. [29]

Щільність цілей відображає наскільки сильно постраждає система в цілому від експлуатації уразливості. Щільність цілей може бути низькою, середньої або високої. [29]

Вимоги до конфіденційності, вимоги до цілісності та вимоги до доступності - можуть бути низькими, середніми або високими. Це визначається вимогами, що пред'являються до уразливої ​​системи. [29]

Ідеальну систему оцінки вразливостей створити швидше за все неможливо. Навіть в актуальній версії стандарту CVSS 3.0 можуть бути неоднозначні трактування, наприклад, при визначенні важливості приватної інформації. Однак, використання будь-якої версії CVSS в більшості випадків дозволяє правильно розставити пріоритети в обробці вразливостей і оцінити можливі ризики. Найдетальнішу оцінку дозволяє проводити остання версія стандарту. Але якщо в організації вже існує велика база з оцінками, виконаними за попередньою версією CVSS, то перехід на більш сучасну версію може виявитися занадто трудомістким і не принести потрібного результату.

## 2.8. Висновки по розділу

В другому розділі були викладені теоретичні відомості, пов’язані з тестуванням на проникнення. Визначено актуальність тестування на проникнення в сучасному світі та досліджено методику його проведення. Також з’ясовано, які вразливості може виявити пентест.

Процедура тестування на проникнення стає все більш популярною через її наближеність до реальних атак зловмисників, що на практиці демонструє рівень захищеності.

Щодо методики тестування, то вона відрізняється практично у кожного виконавця, проте базується на однакових етапах хакингу. Деякі команди проводять пентест згідно власноруч розробленої техніки, інші ж звертаються до методологій з відкритим кодом, які доступні безкоштовні і практично повністю описують весь порядок і аспекти тестування на проникнення.

# РОЗДІЛ 3

# ПРАКТИЧНА РЕАЛІЗАЦІЯ МЕТОДУ ОЦІНКИ ЗАХИЩЕНОСТІ СИСТЕМИ НА ОСНОВІ ТЕСТУВАННЯ НА ПРОНИКНЕННЯ

В цьому розділі на найпростіших прикладах злому з використанням найпростіших атак демонструється реалізація запропонованого алгоритму оцінки захищеності системи.

##### 3.1. Розробка алгоритму оцінки рівня небезпеки вразливостей та захищеності системи.

Тест на проникнення дозволяє отримати об'єктивну оцінку того, наскільки легко отримати  доступ до ресурсів корпоративної мережі і сайту компанії, яким способом і через які вразливості. Тест на проникнення є моделюванням дій зловмисника по проникненню в  інформаційну систему і дозволяє виявити найбільше вразливостей в захисті мережі. Тест на проникнення проводять для здобуття незалежної оцінки захищеності своєї корпоративної мережі [30]

У вигляді самостійної роботи тести можуть проводитися з двома основними цілями: необхідності проведення робіт по підвищенню захищеності або здобуття незалежної оцінки рівня безпеки інформаційної системи.

В цьому випадку результати тестування дозволяють реально оцінити залишкові ризики, а можливо і виявити приховані недоліки в системі.

В сучасному світі не існує системи яку неможливо зламати, тому основними показниками захищеності стають імовірність експлуатації вразливості та можливі нанесені збитки від цього. Чим ці показники нижчі тим система вважається більш захищеною. Імовірність експлуатації вразливості та можливі нанесені збитки є важливими і з точки зору бізнесу, адже якщо якась вразливість не несе за собою великих збитків, або малоймовірна до експлуатації, то не завжди керівництво організації захоче витрачати кошти на закриття цієї вразливості.

Отже алгоритм має базуватися на двох вищезгаданих критеріях. Повинно оцінювати як і окремо кожну вразливість так і дати загальну оцінку захищеності системи.

Також після проведення тестування оцінювання рівня небезпеки вразливостей повинне бути простим та зрозумілим для замовника і одночасно загальним та об’єктивним.

Можливі збитки та імовірність експлуатації вразливості є основними критеріями та оцінюються по шкалі «низька, середня, висока». Ці показники є комплексними і в свою чергу поділяються на більш прості, зрозумілі базові показники. Далі розглянемо детальніше кожен з них.

##### 3.1.1. Імовірні збитки від експлуатації вразливості

Основним активом будь-якої організації є інформація. Це можуть бути персональні дані клієнтів, об’єкти інтелектуальної діяльності, документи, що містять державну таємницю тощо. Тому всі збитки пов’язані з інформацією, а саме з порушенням конфіденційності, цілісності або доступності.

Конфіденційність, цілісність чи доступність може або порушуватись або ні, тому градація рівня небезпеки буде базуватися також на кількості інформації.

Градації кількості важливої інформації, на яку впливає експлуатація вразливості:

- Вся (більшість) конфіденційна, чутлива інформація, що зберігається на системі

- Обмежена частина конфіденційної, чутливої інформації, що зберігається на системі

- Конфіденційна інформація не страждає, а лише системна інформація.

На основі цього була визначена трьохрівнева градація в форматі: «Повне порушення, часткове, не порушується». Детальний опис далі в таблицях.

Таблиця 3.1

Порушення конфіденційності

|  |  |
| --- | --- |
| Порушення конфіденційності (VC) | Опис |
| Повне (VC1) | Якщо експлуатація вразливості дозволяє читати (отримувати доступ) більшість конфіденційної, чутливої інформації, що зберігається на системі. |
| Часткове (VC2) | Якщо експлуатація вразливості дозволяє читати (отримувати доступ) обмежену кількість конфіденційної, чутливої інформації, що зберігається на системі. |
| Не порушується (VC3) | Якщо експлуатація вразливості не дозволяє читати (отримувати доступ) конфіденційну, чутливої інформації, що зберігається на системі, а лише відбувається розкриття системної інформації. |

Таблиця 3.2

Порушення цілісності

|  |  |
| --- | --- |
| Порушення цілісності (VI) | Опис |
| Повне (VI1) | Якщо експлуатація вразливості дозволяє змінювати більшість конфіденційної, чутливої інформації, що зберігається на системі. |
| Часткове (VI2) | Якщо експлуатація вразливості дозволяє змінювати обмежену частину конфіденційної, чутливої інформації, що зберігається на системі. |
| Не порушується (VI3) | Якщо експлуатація вразливості не дозволяє змінювати конфіденційну, чутливу інформацію, що зберігається на системі. |

Таблиця 3.3

Порушення доступності

|  |  |
| --- | --- |
| Порушення доступності (VA) | Опис |
| Повне (VA1) | Якщо експлуатація вразливості призводить до повної втрати доступу до системи. |
| Порушення доступності (VA) | Опис |
| Часткове (VA2) | Якщо експлуатація вразливості призводить до часткової втрати доступу до системи (втрати доступності лише до деяких функцій). |
| Не порушується (VA3) | Якщо експлуатація вразливості не призводить до втрати доступу до системи. |

Ще одним важливим фактором в питанні нанесених збитків системі чи організації після експлуатації вразливості є отриманий доступ до системи в результаті цієї експлуатації.

Таблиця 3.3

Доступ, отриманий після експлуатації вразливості

|  |  |
| --- | --- |
| Доступ до системи (SA) | Опис |
| Повний (SA1) | Якщо експлуатація вразливості призводить до отримання можливостей віддаленого виконання коду з правами адміністратора. |
| Частковий (SA2) | Якщо експлуатація вразливості призводить до отримання можливостей віддаленого виконання коду через пристрої користувачів (принтери, IP телефони, персональні ПК від імені користувача тощо) або від імені акаунтів з правами користувача. |
| Без доступу (SA3) | Якщо експлуатація вразливості не призводить до отримання можливостей віддаленого виконання коду.(Атакуючий не отримує жодного доступу). |

##### 3.1.2. Імовірність експлуатації вразливості

Імовірність експлуатації вразливості – це не менш важливий показник для визначення загального рівня небезпеки цієї вразливості і захищеності системи в цілому. Цей показник базується на тому, що необхідно зловмиснику для експлуатації. Основними складовими є складність пошуку та реалізації вразливості (чи це загально відома вразливість, експлойти на яку є в загальному доступі, чи на пошук і експлуатацію вразливості необхідно затратити велику кількість часу та ресурсів), необхідний рівень доступу (анонімний чи самостворений акаунт, чи акаунт з правами адміністратора тощо) та вектор атаки (через інтернет чи експлуатація можлива лише за специфічних умов підключення). Детальніший опис далі в таблицях.

Таблиця 3.4

Складність виявлення та експлуатації вразливості

|  |  |
| --- | --- |
| Індекс складності експлуатації (VS) | Опис |
| Високий (VS1) | Вразливість може бути виявлена більшістю сканерів вразливостей, та/або існують експлойти у відкритому виді. |
| Середній (VS2) | Вразливість виявляється за допомогою загальних технік, або через помилки на сервері. |
| Низький (VS3) | Виявлення вразливості вимагає додаткових відомостей про внутрішню структуру системи або тривалого, глибокого тестування. |

Таблиця 3.5

Доступ до системи перед експлуатацією

|  |  |
| --- | --- |
| Індекс існуючого доступу до системи (CA) | Опис |
| Високий (CA1) | Вразливість може бути проексплуатована анонімно (без аутентифікації), або через самостійно створений обліковий запис. |
| Низький (CA2) | Вразливість може бути проексплуатована лише від імені адміністратора. |

Таблиця 3.6

Вектор експлуатації

|  |  |
| --- | --- |
| Індекс вектору атаки (VV) | Опис |
| Високий (VV1) | Вразливість може бути проексплуатована з будь-якого хоста в інтернеті або у внутрішній мережі. |
| Низький (VV2) | Для експлуатації вразливості атакуючому необхідно фізичний доступ, специфічна логічна маршрутизація до цілі, проведення MitM атаки або успішне проведення атак соціальної інженерії. |

##### 3.1.3. Визначення сумарного рівня небезпеки вразливості

Далі зведемо всі показники та виведемо логічні формули для визначення загального рівня збитків (ризика від експлуатації) та імовірності експлуатації вразливості.

Таблиця 3.7

Ризик від експлуатації

|  |  |  |
| --- | --- | --- |
| Ризик від експлуатації | Опис | Логічна Формула |
| Високий | Якщо експлуатація вразливості призводить до отримання можливостей віддаленого виконання коду з правами адміністратора.  Або експлуатація вразливості дозволяє читати (отримувати доступ) або змінювати більшість конфіденційної, чутливої інформації, що зберігається на системі. | (SA1) OR ((VC1) OR (VI1)) |
| Середній | Якщо експлуатація вразливості призводить до отримання можливостей віддаленого виконання коду через пристрої користувачів або від імені акаунтів з правами користувача.  Або експлуатація вразливості дозволяє читати (отримувати доступ) або змінювати обмежену кількість конфіденційної, чутливої інформації, що зберігається на системі. | (SA2) OR ((VC2) OR (VI2)) |
| Низький | Якщо експлуатація вразливості не призводить до отримання можливостей віддаленого виконання коду (Атакуючий не отримує жодного доступу), а лише можливе обмеження доступності до системи або деяких її функцій. Експлуатація вразливості не дозволяє читати або змінювати (отримувати доступ) конфіденційну, чутливої інформації, що зберігається на системі, а лише відбувається розкриття системної інформації. | (SA3) OR (VC3) OR (VI3) OR (VA1) OR (VA2) OR (VC3) |

Таблиця 3.8

Імовірність експлуатації

|  |  |  |
| --- | --- | --- |
| Імовірність експлуатації | Опис | Логічна Формула |
| Високий | Якщо вразливість може бути виявлена більшістю сканерів вразливостей, та/або існують експлойти у відкритому виді. Та вразливість може бути проексплуатована анонімно (без аутентифікації), або через самостійно створений обліковий запис, з будь-якого хоста з інтернету або внутрішньої мережі. | (VS1) AND (CA1) AND (VV1) |
| Середній | Якщо вразливість виявляється за допомогою загальних технік, або через помилки на сервері, та вразливість може бути проексплуатована анонімно (без аутентифікації), або через самостійно створений обліковий запис, з будь-якого хоста з інтернету або внутрішньої мережі. | (VS2) AND (CA1) AND (VV1) |
| Низький | Якщо виявлення вразливості вимагає додаткових відомостей про внутрішню структуру системи або тривалого, глибокого тестування. Або вразливість може бути проексплуатована лише від імені адміністратора. Або, якщо для експлуатації вразливості атакуючому необхідно фізичний доступ, специфічна логічна маршрутизація до цілі, проведення MitM атаки або успішне проведення атак соціальної інженерії. | (VS3) OR (CA2) OR (VV2) |

Далі потрібно визначити рівень небезпеки вразливості, і робитимемо це за зведеною матрицею.

Таблиця 3.9

Матриця рівня небезпеки вразливості

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ризик від експлуатації | Високий | Середній | Високий | Критичний |
| Середній | Низький | Середній | Високий |
| Низький | Інформаційний | Низький | Середній |
|  | Низький | Середній | Високий |
|  | Імовірність експлуатації | | | |

Такий підхід дозволяє максимально просто і зрозуміло оцінити небезпеку кожної знайденої вразливості.

На основі всіх вразливостей визначається рівень захищеності системи. Тут відбувається поділ на 5 рівнів: «Дуже низький», «Низький», «Середній», «Високий», «Дуже високий». Алгоритм цього дуже простий: якщо є хоча би одна вразливість з рівнем «Критичний», то рівень захищеності буде відповідно «Дуже низький», якщо є хоча би одна вразливість з рівнем «Високий» і цей рівень найвищий, то рівень захищеності буде «Низький» і так далі. Визначення рівня захищеності системи наведено далі в таблиці.

Таблиця 3.10

Рівень захищеності системи на основі вразливостей

|  |  |
| --- | --- |
| Найвищий рівень небезпеки знайдених вразливостей | Результуючий рівень захищеності системи |
| Інформаційний | Дуже високий |
| Низький | Високий |
| Середній | Середній |
| Високий | Низький |
| Критичний | Дуже низький |

##### 3.2. Програмна реалізація алгоритму оцінки рівня небезпеки вразливостей та захищеності системи

Програмна реалізація алгоритму являє собою просту програму (Windows forms application), написану на мові програмування C#, для платформи Windows у вигляді калькулятора з можливістю вибору значень всіх вищезгаданих метрик та назви вразливості. Дана програма обчислює рівень небезпеки кожної добавленої вразливості і виводить результат в текстовому полі та у вигляді кругової діаграми. А також вона обчислює загальний рівень захищеності системи на основі всіх введених вразливостей.

Програма розроблена в середовищі програмування Microsoft Visual Studio - серія продуктів фірми Майкрософт, які включають інтегроване середовище розробки програмного забезпечення та ряд інших інструментальних засобів. Ці продукти дозволяють розробляти як консольні програми, так і програми з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms, а також веб-сайти, веб-застосунки, веб-служби як в рідному, так і в керованому кодах для всіх платформ, що підтримуються Microsoft Windows, Windows Mobile, Windows Phone, Windows CE, .NET Framework, .NET Compact Framework та Microsoft Silverlight. [31]

Графічно програма являє собою 1 форму з випадаючими списками, кнопками та текстовими полями.

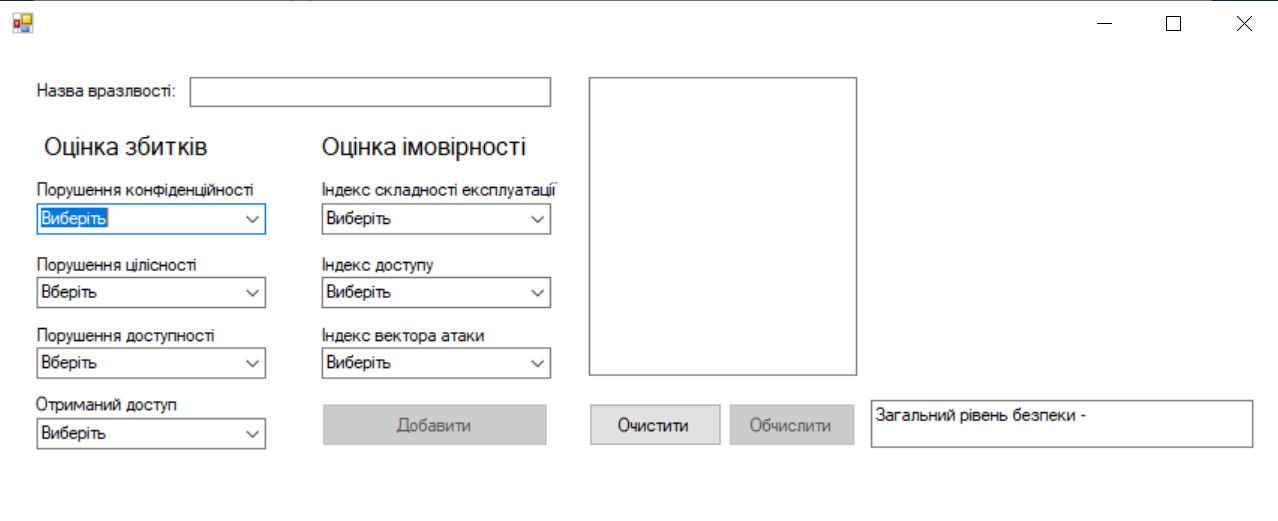


Рис. 3.1 Вид програми.

Довідка з описом усіх метрик реалізована у вигляді спливаючої підказки при наведені курсора мишки на випадаючий список. Після прочитання підказки потрібно вибрати необхідний пункт зі списку.

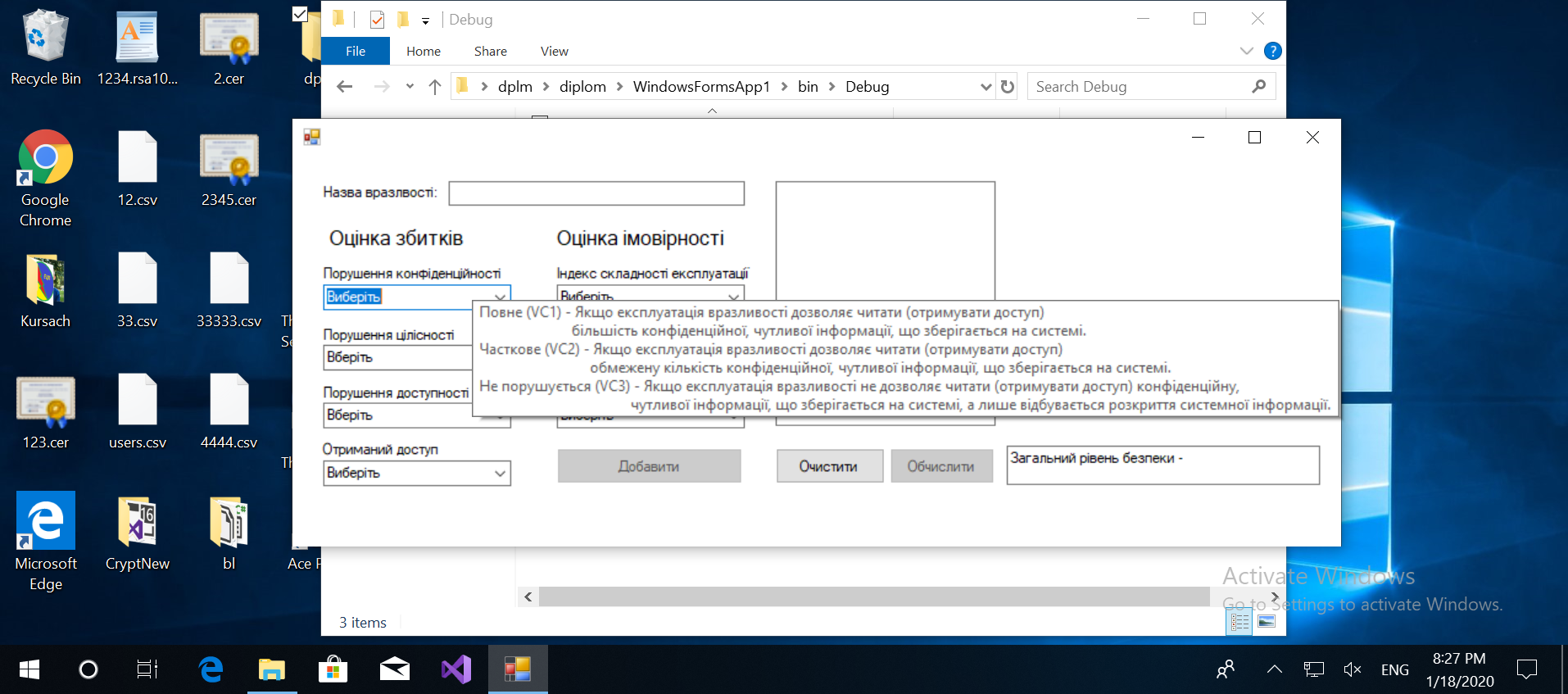


Рис. 3.2 Спливаючі підказки.

Лише коли вибрані всі метрики і вказана назва вразливості програма дозволить добавити її в список і приступити до наступної. Також обчислити загальний рівень безпеки та побудувати діаграму можливо, лише коли в списку є хоча би одна вразливість. Це зроблено для коректної роботи обчислень і запобігання помилок і зависань програми.

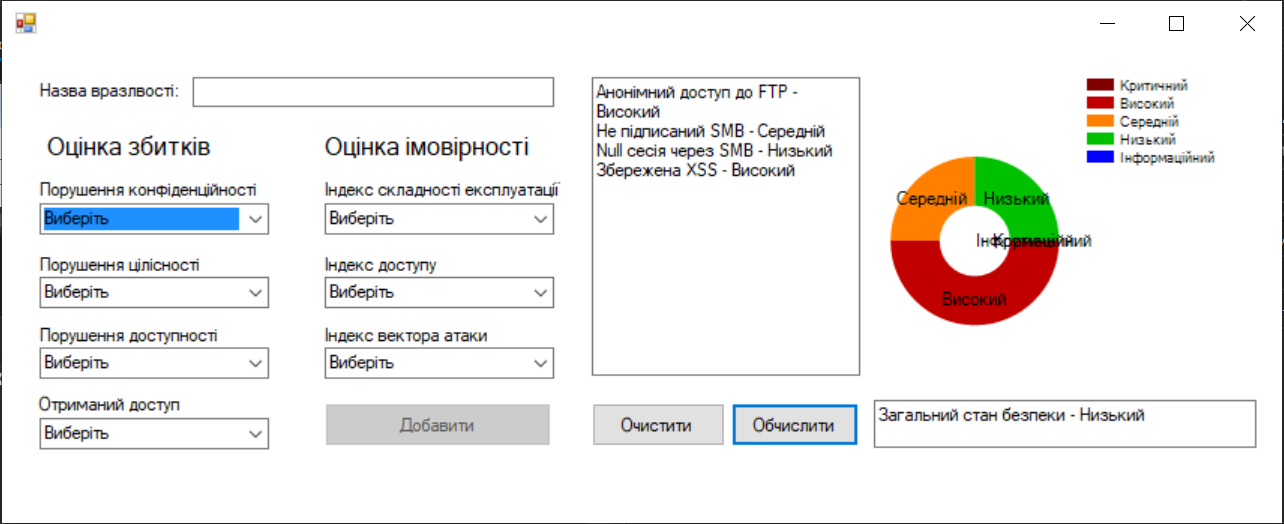


Рис. 3.3 Приклад роботи програми.

##### 3.3. Аналіз причин походження та складності ліквідації вразливостей.

Для оцінки захищеності системи і для подальших усунень проблем необхідно також визначити, що саме спричинило виникнення вразливості. Цей показник, напевне, найважливіший на рівні з ризиком втрати конфіденційних даних, тому що він чітко показує де потрібно проводити зміни та на яку ланку забезпечення інформаційної безпеки потрібно виділити більше ресурсів.

Ще одним важливим показником є те, наскільки легко усунути знайдену вразливість. Проте цей показник є дуже абстрактним і базується лише на загальних відомостях, тому що виконавець тестування не має можливості оцінити фінансові, людські та інші внутрішні ресурси організації замовника. Тому ця метрика носить ймовірний характер.

Для характеристики вразливості з точки зору походження були визначені такі рівні:

Неправильна конфігурація – будь-які помилки в налаштуваннях роботи програм, архітектури мережі, ролей, прав доступу тощо.

Неправильне керування обновленнями – пропущені оновлення ПО та патчів ОС.

Не безпечне програмування – некоректно написане програмне забезпечення, що призводить до виникнення вразливостей.

Вплив людського фактору – будь які вразливості, які спричинені байдужістю та неуважністю людей.

Метрика складності усунення вразливості оцінюється по таких рівнях:

Просте усунення – якщо для виправлення вразливості необхідно лише оновити ПО, закрити порт тощо.

Складне усунення – якщо для виправлення вразливості необхідне переконфігурування системи чи мережі, переписання частини коду тощо з залученням власних ресурсів.

Дуже складне усунення – якщо для виправлення вразливості необхідне переналаштування всієї системи, глобальне переписання коду з залученням сторонніх ресурсів тощо.

Ці характеристики ніяк не впливають на рівень ризику вразливості, а лише допомагають отримати більше інформації після проведення тестування на проникнення. Адже тестування на проникнення не є дешевою послугою, тому потрібно отримати якомога більше інформації.

##### 3.4. Проведення демонстраційного тестування на проникнення

Для тестування алгоритму, а саме для знаходження вразливостей був вибраний навчальний полігон bWAPP bee-box v.1.6.7 що являє собою віртуальну машину, спеціально призначену для тренування навичок взлому та пошуку вразливостей. Ця віртуальна машина містить безліч сервісів з практично усіма типовими вразливостями.

У цій роботі для прикладу було вибрано лише декілька вразливостей, які дуже часто зустрічаються в реаліях українських організацій.

Далі наведена послідовність знаходження вразливостей та їх опис з поясненням, що саме порушується при експлуатації.

Анонімний доступ до FTP

При скануванні віртуальної машини був виявлений сервіс FTP. Далі була спроба анонімного підключення, яка стала успішною. Був отриманий доступ до файлів, які можна переглядати. Це означає що порушується конфіденційність частини даних, які зберігаються на системі, адже логічно, що файли, доступні по FTP – це не всі файли на цій віртуальні машині.

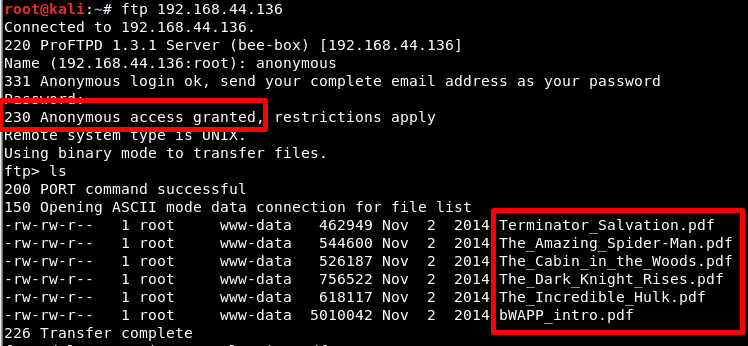


Рис. 3.4 Анонімний доступ до файлів через FTP.

Далі була вдала спроба видалення одного файлу і вона була успішна, це означає, що порушується цілісність а також доступність. Як у випадку конфіденційності також лише частини інформації.

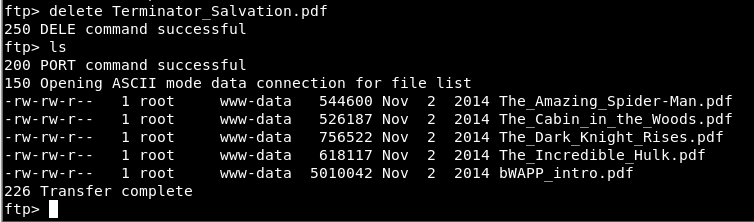


Рис. 3.5 Анонімний доступ до файлів через FTP.

Вразливість анонімного доступу до FTP дозволяє користувачу без ніякого облікового запису, не знаючи логіна і паролю отримати доступ до файлів, які містять чутливу інформацію. Також зловмисник може спробувати загрузити велику кількість даних, що спричинить DOS системи.

Підписання SMB не вимагається

На даній віртуальній машині також був SMB сервіс. Цей сервіс призначений для «розшарення» папок та віддаленого доступу до файлів. Даний сервіс є дуже популярним, але відомий також великою кількістю вразливостей.

Одна з них дуже просто перевіряється за допомогою сканера nmap. На наступному скріншоті видно, що підписання SMB вимкнене.

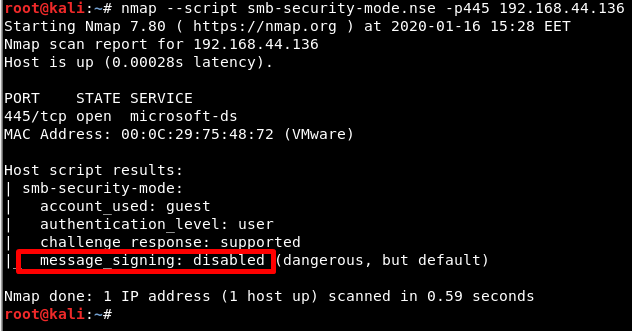


Рис. 3.6 Підписання SMB не вимагається.

Підписання SMB відбувається попакетно і захищає трафік від атаки типу SMB Relay. Якщо підписання не вимагається, то атакуючий може отримати доступ до чутливої інформації такої як хеші паролів або перехватити з’єднання іншого користувача.

В даному випадку, не порушується ні конфіденційність ні цілісність ні доступність, також не був отриманий доступ. Ця вразливість є потенційно небезпечна як базис для інших атак.

Перебір нульової сесії SMB

В сервісі SMB існує ще одна дуже поширена вразливість – це перебір нульової сесії SMB. Перевіряється наявність цієї вразливості за допомогою спеціальної програми «enum4linux».

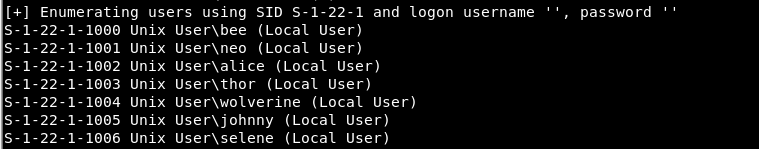


Рис. 3.7 Дозволений перебір нульової сесії SMB.

Якщо перебір дозволений, це означає, що користувач може підключитися до SMB з пустими логіном і паролем. Це дозволяє отримати список та вміст «розшарених» папок, користувачів на системі, список прав на читання чи запис, особливості парольної політики та іншу системну інформацію. Тут також не порушується ні конфіденційність ні цілісність ні доступність, також не був отриманий доступ.

Збережена XSS

Тестування веб додатків переважно проходить як окрема процедура, тому що веб час доступний на весь інтернет і повинен бути найбільш захищеним.

Одною з найпоширеніших вразливостей в вебі є XSS. Вона переважно виявляється дуже просто, більшістю сканерів або звичайними пейлоадами. Так і було знайдено дану вразливість, використавши пейлоад <img src=x onerror=alert(document.location)>

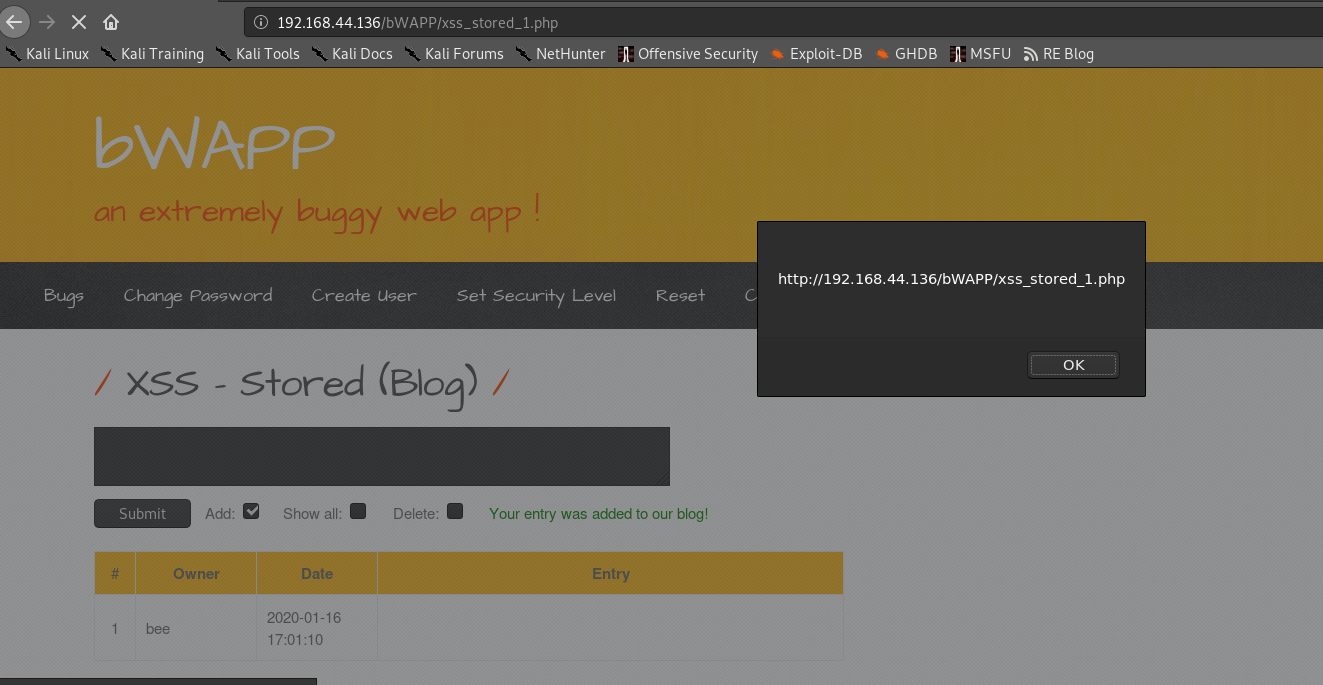


Рис. 3.8 Збережена XSS.

Така вразливість може дозволити перехватити сесію (файли кукі), та отрианти доступ до веб додатка від імені адміністратора.

Робиться це за допомогою пейлоада <img src=x onerror=this.src='http://yourserver/?c='+document.cookie>

В цей час піднімається сервер python -m SimpleHTTPServer 80.



Рис. 3.9 Перехвачені кукі.

Далі ці кукі вставляються в браузері та отримується сесія.

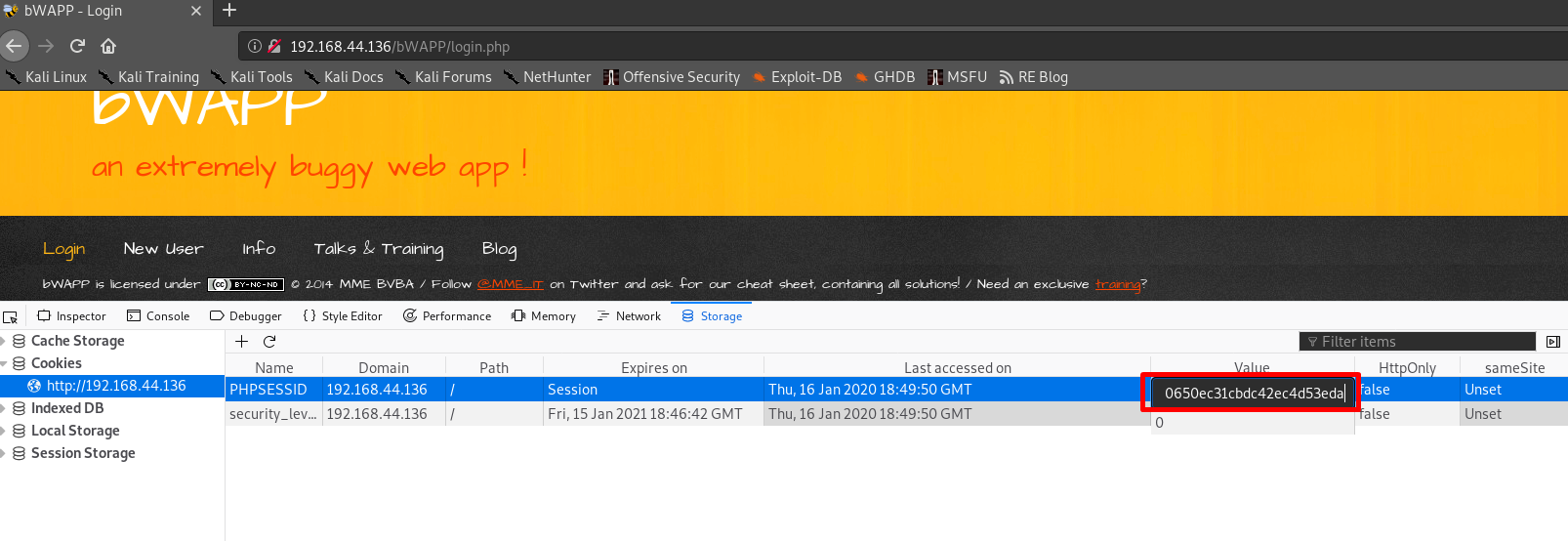


Рис. 3.10. Використання кукі в браузері.

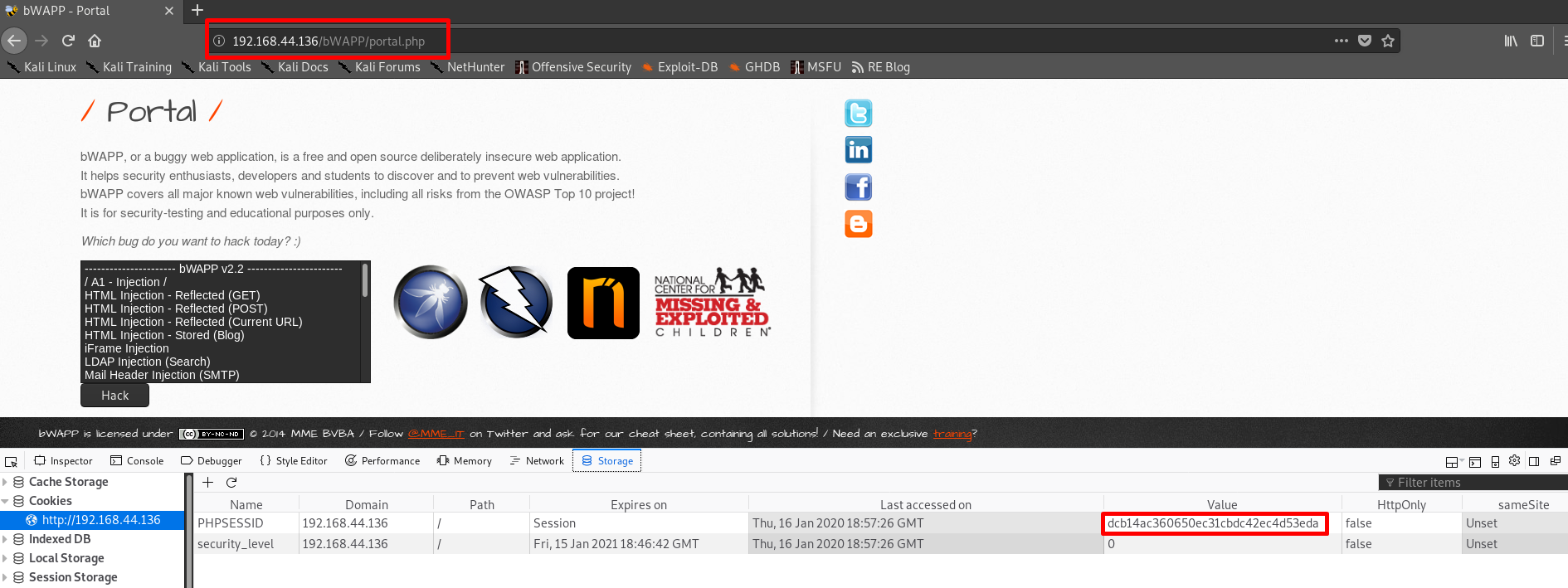


Рис. 3.11. Отримання сесії.

Витік інформації через HTTP заголовки

При досліджені веб додатку було виявлено, що в заголовку розголошується системна інформація про версію програмного забезпечення.

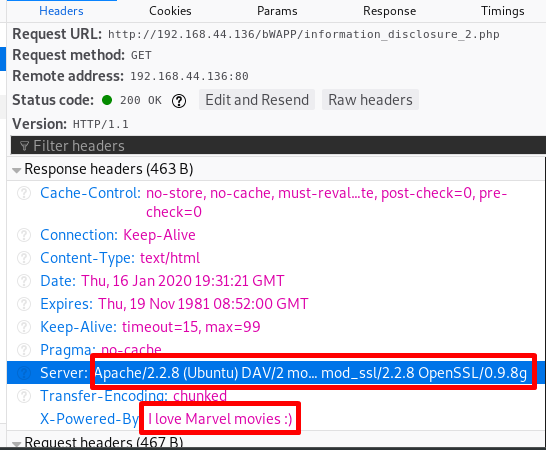


Рис. 3.12. Розкриття інформації в заголовках.

Ця вразливість не є небезпечною, проте розкрита інформація може слугувати як база для подальших атак і розширити площу атаки.

Проведене тестування було лише вибірковим і показовим, для демонстрації обчислень згідно алгоритму та роботи програмної реалізації цього алгоритму.

##### 3.5. Аналіз результатів |

Далі потрібно оцінити всі знайдені вразливості за створеними критеріями.

Анонімний доступ до FTP

В цій вразливості порушується конфіденційність, доступність та цілісність. А також отримується частковий доступ до системи. Щодо імовірності – цю вразливість дуже легко знайти простим написанням словом Anonymous при спробі входу. Даний хост доступний з будь-якої точки мережі а також не потрібен жодний акаунт для її експлуатації.

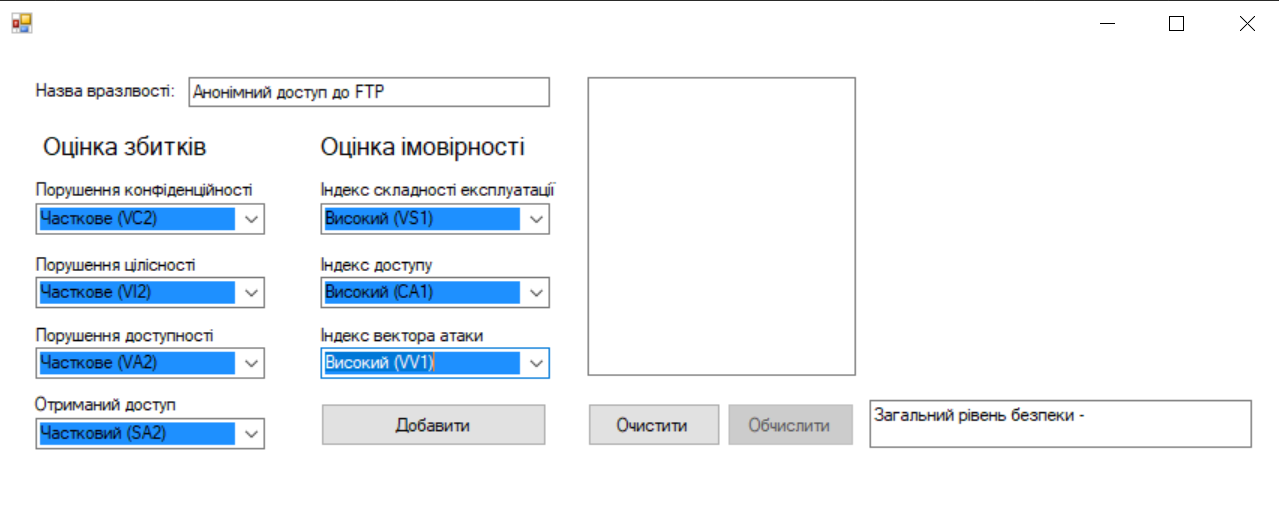


Рис. 3.13. Оцінка Анонімного доступу до FTP.

Результуюча рівень вразливості – високий.

Підписання SMB не вимагається

В даному випадку, не порушується ні конфіденційність ні цілісність ні доступність, також не був отриманий доступ. Проте виявлення вразливості не потребує складних дій.

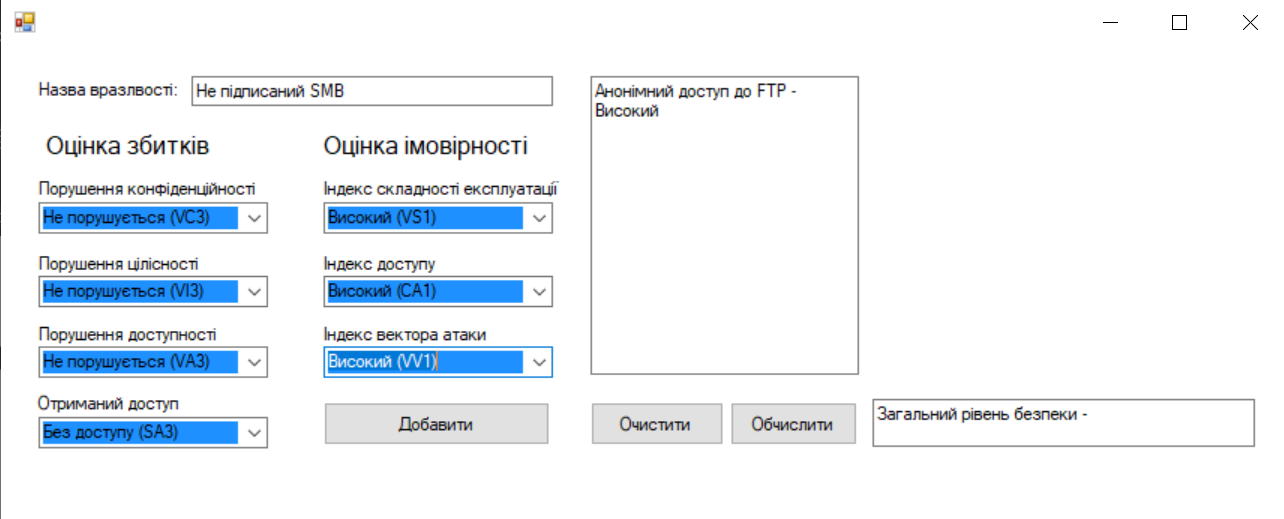


Рис. 3.14. Оцінка не обов’язкового підписання SMB.

Результуючий рівень вразливості – середній.

Перебір нульової сесії SMB

В даному випадку, не порушується ні конфіденційність ні цілісність ні доступність, також не був отриманий доступ. Проте виявлення вразливості не потребує складних дій, окрім знання додаткових скріптів в nmap.

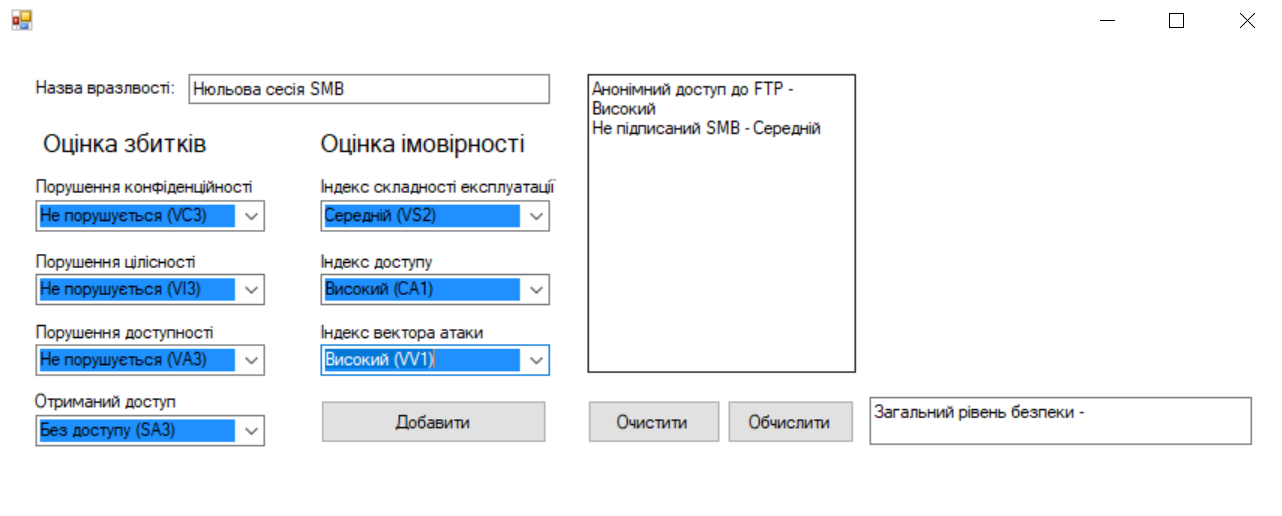


Рис. 3.15. Оцінка нульової сесії SMB.

Рівень небезпеки даної вразливості – низький.

Збережена XSS

В цій вразливості порушується конфіденційність, доступність та цілісність. А також отримується частковий доступ до системи. Щодо імовірності – вразливість може бути виявлена будь-яким сканером з будь-якого хоста в мережі.

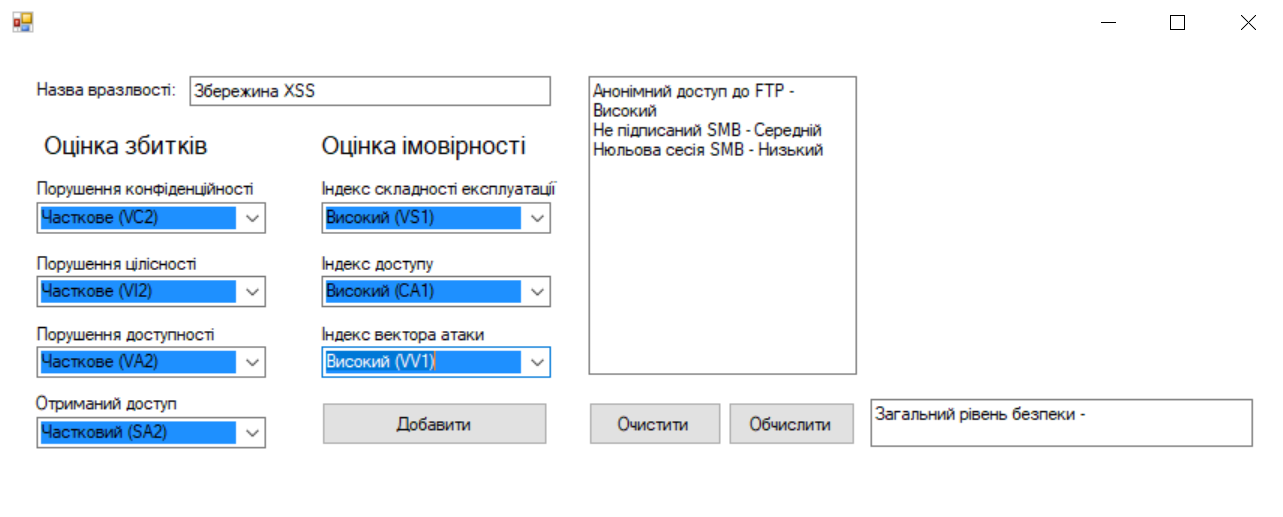


Рис. 3.16. Оцінка XSS.

Результуючий рівень – високий.

Витік інформації через HTTP заголовки

В цій вразливості не порушується конфіденційність, цілісність чи доступність інформації, проте простота виявлення встановлює рівень небезпеки – середній.

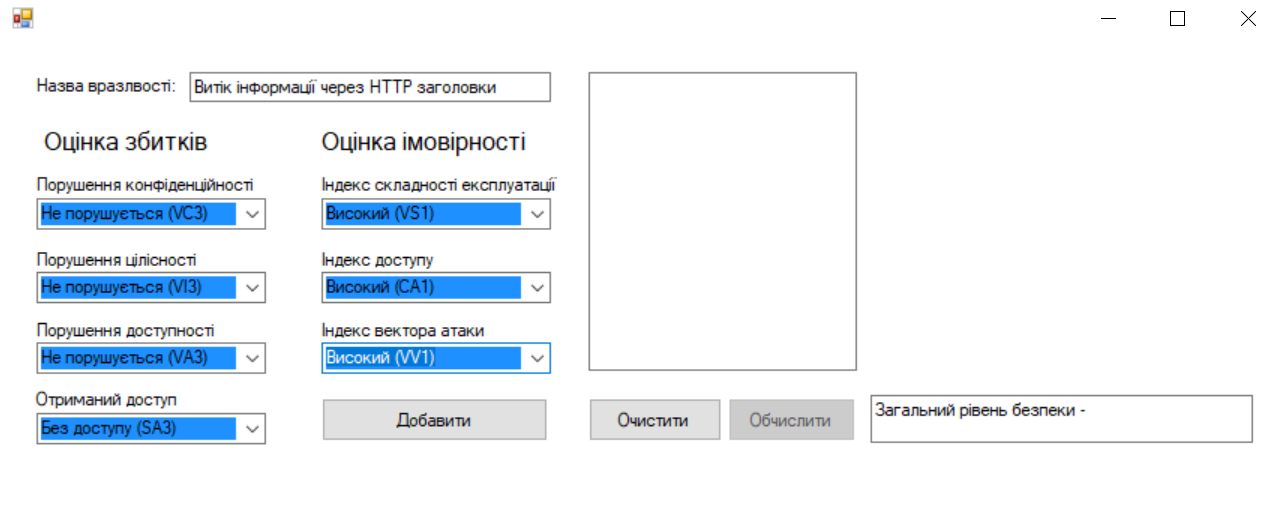


Рис. 3.17. Оцінка витоку інформації через заголовки HTTP.

Далі йде обчислення результуючого стану захищеності системи, та будується діаграма рівнів вразливостей.

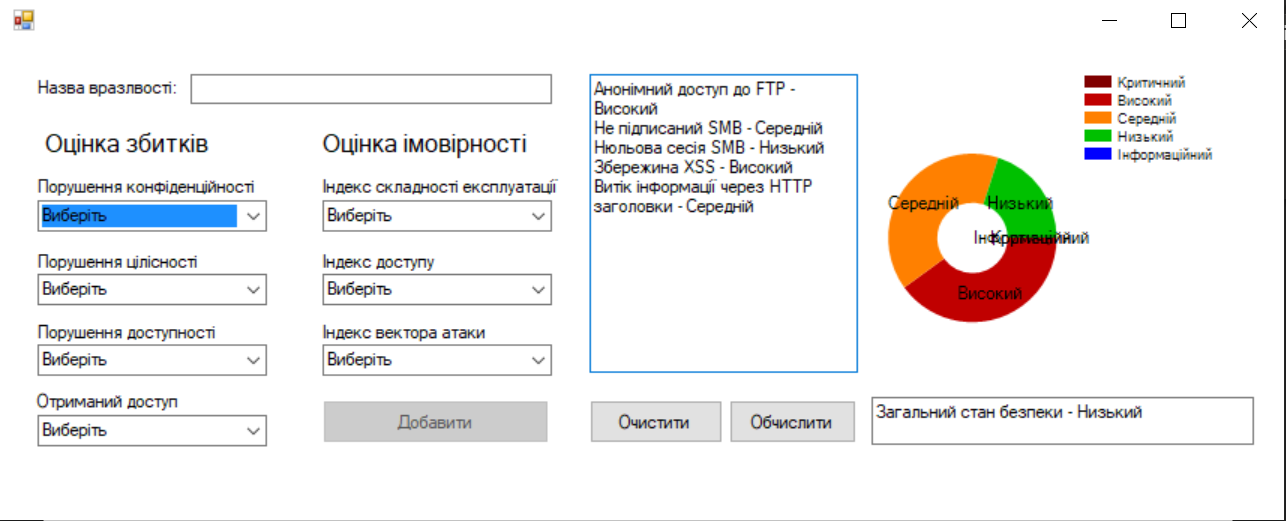


Рис. 3.18. результуюча оцінка.

Також оцінюються визначені раніше додаткові компоненти – причини походження та складність усунення вразливостей.

Таблиця 3.11

Причини походження та складність усунення вразливостей

|  |  |  |
| --- | --- | --- |
| Назва вразливості | Походження | Складність усунення |
| Анонімний доступ до FTP | Неправильна конфігурація | Просте |
| Підписання SMB не вимагається | Неправильна конфігурація | Просте |
| Перебір нульової сесії SMB | Неправильна конфігурація | Просте |
| Збережена XSS | Не безпечне програмування | Складне |
| Витік інформації через HTTP заголовки | Неправильна конфігурація | Просте |

Результати зручно відображати у вигляді діаграм

При повноцінному тестуванні на проникнення виконавець описує у звіті усі результати тестування, додаючи скріншоти, вказує на ризики а також надає рекомендації щодо покращення захищеності системи шляхом виправлення знайдених вразливостей.

## 3.6. Рекомендації щодо покращення захищеності

Тестування на проникнення проводиться як з метою виявлення вразливостей так і з метою отримання рекомендацій по усуненю цих вразливостей та покращенню захищеності в цілому. Тому виконавець повинен надати рекомендації по виправленні кожної вразливості. Проте, це не детальні інструкції а лише поверхневі рекомендації.

Анонімний доступ до FTP

Для виправлення цієї вразливості потрібно забезпечити аутентифікацію, що базується на логіні і паролі. А також в конфігурації сервісу, заборонити анонімний вхід. Також хорошою практикою є відкриття доступу до FTP лише для користувачів, яким це необхідно відповідно до бізнес логіки.

Підписання SMB не вимагається

Щоб виправити цю вразливість а також мінімізувати збитки необхідно поміняти конфігурацію та застосувати підписання SMB. Також потрібно обмежити права користувачів на системі.

Перебір нульової сесії SMB

Слід заборонити підключення з пустим логіном і паролем, тобто в конфігурації заборонити нульову сесію.

Збережена XSS

В основі XSS вразливості лежить неправильне програмування. Тому, потрібно запровадити санітизацію та валідацію на формах користувацького вводу.

Витік інформації через HTTP заголовки

Це одна з найпоширеніших витоків інформації, і усувається вона дуже легко – потрібно сконфігурувати іншу відповідь сервера.

## 3.7. Висновки по розділу

В цьому розділі був побудований полігон для тестування в програмі GNS3, також був розроблений алгоритм оцінювання захищеності на основі пентесту. Проведене саме тестування на проникнення та проаналізовані отримані дані. Виконавши дану роботу можна сказати, що розроблений алгоритм справляється зі своїм завданням – покращити та спростити сприйняття результату тестування замовником, а також наглядніше показати рівень захисту системи. В проведеній роботі є також і мінуси, такі як невелика кількість видів проведених тестувань та обмеженість в апаратних потужностях, що не дозволило розгорнути масштабніший полігон тестування.

В загальному проведена практична робота є успішною, адже були досягнуті поставлені цілі.

# ВИСНОВКИ

Буквально словосполучення тестування на проникнення походить від англійського penetration test. Даний процес дозволяє моделювати несанкціонований доступ в інформаційні системи, а також інші дії, що дозволяють порушити нормальне функціонування інформаційних систем і бізнес-процесів. По суті, це метод оцінки захищеності інформаційних систем та/або інформації, та об'єктів, де вона зберігається або обробляється від несанкціонованого використання. Ця послуга переважно замовляється компаніями, чия ІТ інфраструктура є досить великою, в якій функціонують критичні сервіси, а бізнес-процеси складними, тому що це не дешева процедура, проте дуже ефективна.

В роботі була розглянута проблема актуальності пентесту та його удосконалення для кращого сприйняття результатів. Спершу була проаналізована нормативна база України і світу в сфері захисту інформації і тестування на проникнення. Було визначено рівень українського законодавства та розглянуті основні міжнародні стандарти в кібербезпеці.

Наступним етапом було теоретичне дослідження тестування на проникнення. Були вирішенні такі завдання як визначення актуальності тестування на проникнення в сучасному світі та дослідження методики проведення пентесту. Отже, тестування на проникнення, здійснюване за допомогою тестів на подолання захисту ІКМ послуга аудиту, що дозволяє виявляти недоліки у сфері інформаційної безпеки з погляду стороннього спостерігача, не враховані при розробці політики безпеки та розкривати внутрішні і зовнішні спроби проникнення в інформаційну систему й запобігати їм. Визначено, що основні завдання тестування на проникнення в тому, щоб відповісти на такі питання:

* Що може побачити зловмисник у цільовій системі?
* Що зловмисник може зробити з цією інформацією?
* Хто-небудь може помітити спроби вторгнення?
* Чи всі компоненти інформаційної системи адекватно захищені та оновлені?
* Скільки зусиль, часу і грошей потрібно, щоб побудувати адекватний захист?
* Чи заходи щодо інформаційної безпеки відповідають галузевим та юридичним стандартам?

Щодо методики тестування, то вона в загальному випадку базується на основних фазах хакінгу, проте кожен виконавець може використовувати свої власні техніки проведення атак чи збору інформації. Проте початком пентесту завжди є підписаний договір, а результатом – звіт, де описані всі знайдені вразливості та шляхи їх експлуатації. Саме на основі цього звіту замовник може проводити дії з покращення захищеності своєї системи.

Далі завданням було розробити алгоритм оцінювання результатів тестування, практично провести тестування на проникнення та проаналізувати отримані результати. Запропонований алгоритм базується на способі оцінки ризиків інформаційної безпеки. Незважаючи на простоту, він забезпечує непогану точність оцінки захищеності мережі на основі результатів проведеного тестування. Проведене тестування системи складалося лише з пошуку деяких вразливостей, але в межах роботи цього було достатнь для демонстрації роботи методу оцінки захищеності та корисності пентесту.

Отже, можна сказати, що посталені завдання були вирішенні, а про актувальність пентесту найкраще скаже те, що неможливо забезпечити безкомпромісний захис системи, тому доцільно використовувати практичний захист, одним з методів якого є тестування на проникнення. Такий підхід є єдиним способом отримати реальну картину стану захищеності системи, і, отже, здобути контроль над ІТ-середовищем, що постійно зростає.

# СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Закон України «Про основні засади забезпечення кібербезпеки України». (Відомості Верховної Ради, 2017 р., № 45, ст.403), редакція закону від 05.10.2017 р. чинна з 05.10.2017 р.
2. НД ТЗІ 2.5-005-99 «Класифікація автоматизованих систем і стандартні функціональні профілі захищеності оброблюваної інформації від несанкціонованого доступу (зі зміною № 1)». 1999 р., №22
3. НД ТЗІ 1.1-002-99 «Загальні положення щодо захисту інформації в комп'ютерних системах від несанкціонованого доступу». 2012 р., №806
4. НД ТЗІ 2.5-004-99 «Критерії оцінки захищеності інформації в комп’ютерних системах від несанкціонованого доступу». 2012 р., №806.
5. НД ТЗІ 2.7-009-09 «Методичні вказівки з оцінювання функціональних послуг безпеки в засобах захисту інформації від несанкціонованого доступу» 2009р., №172
6. Как КПИшник Алексей Мохов взломал ПриватБанк [Електронний ресурс]. – Режим доступу: https://kpishnik.com/mohov-privat24/ – Дата доступу: 06.02.2020
7. Митько Д. ПриватБанк обвинил украинского программиста во взломе своего Android-приложения [Електронний ресурс]. – Режим доступу: https://habr.com/post/193204/ – Дата доступу: 06.02.2020
8. F\*ck responsible disclosure. Привет от очень злых хакеров [Електронний ресурс]. – Режим доступу: <https://petrimazepa.com/disclosure.html> – Дата доступу: 06.02.2020
9. Кримінальний кодекс України (Відомості Верховної Ради України (ВВР), 2001, № 25-26, ст.131)
10. Дубов Д.В. Проблеми застосування окремих статей кримінального кодексу україни в контексті хактивістської діяльності [Електронний ресурс]. – Режим доступу: <http://old2.niss.gov.ua/content/articles/files/CrimeCodex-4b8b5.pdf> – Дата доступу: 06.02.2020
11. Казакова Н.Ф., Плешко Е.А., Айвазова К.Б. Міжнародна регламентація правового регулювання та стандартизації аудиту інформаційної безпеки. Вісник східноукраїнського національного університету імені Володимира Даля № 15 (204) ч.1 2013.
12. Certified Ethical HackerVersion 9 Study Guide, 2016 by John Wiley & Sons, Inc., Indianapolis, Indiana
13. Тесты на проникновение [Електронний ресурс].— Режим доступу: <http://www.ptsecurity.ru/services/pen>test Дата доступу: 06.02.2020
14. Соколов А. Тестирование на проникновение: инструментальный анализ уязвимостей или имитация действий злоумышленника? [Електронний ресурс] /Андрей Соколов.— Режим доступу: https://forum.antichat.ru/threads/445279/ Дата доступу: 06.02.2020
15. Дорофеев А. Тестирование на проникновение: демонстрация одной уязвимости или объективная оценка защищенности? [Електронний ресурс] /А. Дорофеев.— Режим доступу: <http://www.npo-echelon.ru/doc/inside-dorofeev.pdf> Дата доступу: 06.02.2020
16. Соколов, А. Тесты на проникновение повысят интерес топ-менеджмента к ИБ [Електронний ресурс] /Андрей Соколов.— Режим доступу: <http://www.cnews.ru/reviews/free/security2009/articles/pentest.shtml> Дата доступу: 06.02.2020
17. Сердюк В. Тест на проникновение как эффективный инструмент для оценки реальной защищенности банка от внешних угроз [Електронний ресурс] /В. Сердюк.— Режим доступу: <http://www.abajour.ru/files/Serduk_04-2010.pdf> Дата доступу: 06.02.2020
18. Уразливість (інформаційні технології) [Електронний ресурс] .— Режим доступу:<https://uk.wikipedia.org/wiki/%D0%A3%D1%80%D0%B0%D0%B7%D0%BB%D0%B8%D0%B2%D1%96%D1%81%D1%82%D1%8C_(%D1%96%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D1%96%D0%B9%D0%BD%D1%96_%D1%82%D0%B5%D1%85%D0%BD%D0%BE%D0%BB%D0%BE%D0%B3%D1%96%D1%97)> Дата доступу: 06.02.2020
19. Ходорів Д., Естерле К., Що таке «тест на проникнення» або «пентест» [Електронний ресурс] — Режим доступу: <http://www.chernigov.ua/news/internet/2524-sho-take-test-na-proniknennia-abo-pentest.html> Дата доступу: 06.02.2020
20. [Daisy Chaining](https://trinity-hackers.wikispaces.com/Daisy+Chaining) [Електронний ресурс] — Режим доступу: <https://trinity-hackers.wikispaces.com/Daisy+Chaining> Дата доступу: 06.02.2020
21. Бейзер Б. Тестирование чёрного ящика. Технологии функционального тестирования программного обеспечения и систем. — Питер, 2004. — 320 с. — [ISBN 5-94723-698-2](https://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D1%83%D0%B6%D0%B5%D0%B1%D0%BD%D0%B0%D1%8F:%D0%98%D1%81%D1%82%D0%BE%D1%87%D0%BD%D0%B8%D0%BA%D0%B8_%D0%BA%D0%BD%D0%B8%D0%B3/5947236982).
22. Особенности тестирования «серого ящика» [Електронний ресурс] — Режим доступу: <http://quality-lab.ru/key-principles-of-gray-box-testing/> Дата доступу: 06.02.2020
23. ISO/IEC/IEEE 24765:2010 Systems and software engineering — Vocabulary
24. Операторы поисковой системы Google [Електронний ресурс] — Режим доступу: <https://blog.aweb.ua/operatory-poiskovoj-sistemy-google/> Дата доступу: 06.02.2020
25. TCP Flags [Електронний ресурс] — Режим доступу: <https://www.keycdn.com/support/tcp-flags/> Дата доступу: 06.02.2020
26. ping sweep (ICMP sweep) [Електронний ресурс] — Режим доступу: <https://searchnetworking.techtarget.com/definition/ping-sweep-ICMP-sweep> Дата доступу: 06.02.2020
27. What Is NetBIOS? [Електронний ресурс] — Режим доступу: <https://www.lifewire.com/netbios-software-protocol-818229> Дата доступу: 06.02.2020
28. Эффективное получение хеша паролей в Windows. Часть 1 [Електронний ресурс] — Режим доступу: https://www.securitylab.ru/analytics/425033.php Дата доступу: 06.02.2020
29. Павел Ханин, Денис Гамаюнов. Общий обзор систем оценки уязвимостей (CVSS 2.0/3.0) [Електронний ресурс] — Режим доступу: <https://safe-surf.ru/specialists/article/5211/596644/> Дата доступу: 06.02.2020
30. Кавун С.В. Інформаційна безпека. Навчальний посібник. Ч.1. – Харків: Вид. ХНЕУ, 2008. – 352 с.
31. Microsoft Visual Studio [Електронний ресурс] — Режим доступу: <https://uk.wikipedia.org/wiki/Microsoft_Visual_Studio> Дата доступу: 06.02.2020

# Додаток А. Програмний код калькулятора оцінювання рівня загрози вразливостей

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Windows.Forms.DataVisualization.Charting;

namespace WindowsFormsApp1

{

public partial class Form1 : Form

{

int VC = -1;

int VI = -1;

int VA = -1;

int SA = -1;

int VS = -1;

int CA = -1;

int VV = -1;

int impact = -1;

int likelihood = -1;

int severity = 1;

string severity2 = "";

string[] summary = new string[50];

**Продовження додатку А**

string[] xValues = { "Критичний", "Високий", "Середній", "Низький", "Інформаційний" };

double[] yValues = { 0, 0, 0, 0, 0 };

string[] finalstatearr = { "Дуже низький", "Низький", "Середній", "Високий", "Дуже високий" };

int i = 0;

string finalstate = "";

public Form1()

{

InitializeComponent();

}

private void RadioButton1\_CheckedChanged(object sender, EventArgs e)

{

}

private void ComboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

comboBox1.BackColor = Color.DodgerBlue;

}

private void ToolTip1\_Popup(object sender, PopupEventArgs e)

{

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void SplitContainer1\_Panel1\_Paint(object sender, PaintEventArgs e)

{

}

**Продовження додатку А**

private void Label1\_Click(object sender, EventArgs e)

{

}

private void Label2\_Click(object sender, EventArgs e)

{

}

private void Label5\_Click(object sender, EventArgs e)

{

}

private void ComboBox2\_SelectedIndexChanged(object sender, EventArgs e)

{

comboBox2.BackColor = Color.DodgerBlue;

}

private void ComboBox3\_SelectedIndexChanged(object sender, EventArgs e)

{

comboBox3.BackColor = Color.DodgerBlue;

}

private void ComboBox4\_SelectedIndexChanged(object sender, EventArgs e)

{

comboBox4.BackColor = Color.DodgerBlue;

}

private void ComboBox5\_SelectedIndexChanged(object sender, EventArgs e)

{

comboBox5.BackColor = Color.DodgerBlue;

}

private void ComboBox6\_SelectedIndexChanged(object sender, EventArgs e)

{

comboBox6.BackColor = Color.DodgerBlue;

}

**Продовження додатку А**

private void ComboBox7\_SelectedIndexChanged(object sender, EventArgs e)

{

comboBox7.BackColor = Color.DodgerBlue;

}

private void Label10\_Click(object sender, EventArgs e)

{

}

private void TextBox1\_TextChanged(object sender, EventArgs e)

{

}

private void TextBox2\_TextChanged(object sender, EventArgs e)

{

}

private void Button1\_Click(object sender, EventArgs e)

{

if (comboBox1.SelectedIndex >= 0 && comboBox2.SelectedIndex >= 0 && comboBox3.SelectedIndex >= 0 && comboBox4.SelectedIndex >= 0 && comboBox5.SelectedIndex >= 0 && comboBox6.SelectedIndex >= 0 && comboBox7.SelectedIndex >= 0)

{

VC = comboBox1.SelectedIndex + 1;

VI = comboBox2.SelectedIndex + 1;

VA = comboBox3.SelectedIndex + 1;

SA = comboBox4.SelectedIndex + 1;

VS = comboBox5.SelectedIndex + 1;

CA = comboBox6.SelectedIndex + 1;

VV = comboBox7.SelectedIndex + 1;

**Продовження додатку А**

if (VC == 1 || VI == 1)

{

if (SA == 1)

impact = 1;

}

if (VC == 2 || VI == 2)

{

if (SA == 2)

impact = 2;

}

if (VA == 1 || VA == 2 || VA == 3)

{

if (VC == 3 || VI == 3 || SA == 3)

impact = 3;

}

if (VS == 1 && CA == 1 && VV == 1)

{

likelihood = 1;

}

if(VS == 2 && CA == 1 && VV == 1)

{

likelihood = 2;

}

if(VS == 3 || CA == 2 || VV == 2)

{

likelihood = 3;

}

severity = impact + likelihood;

**Продовження додатку А**

if (severity == 6)

severity2 = "Інформаційний";

if (severity == 5)

severity2 = "Низький";

if (severity == 4)

severity2 = "Середній";

if (severity == 3)

severity2 = "Високий";

if (severity == 2)

severity2 = "Критичний";

if (severity < 2)

severity2 = "Невідомо";

}

textBox2.Text += textBox1.Text + " - " + severity2;

textBox2.Text += "\r\n";

textBox1.Text = "";

comboBox1.BackColor = Color.White;

comboBox1.SelectedIndex = -1;

comboBox1.Text = "Виберіть";

comboBox2.BackColor = Color.White;

comboBox2.SelectedIndex = -1;

comboBox2.Text = "Виберіть";

comboBox3.BackColor = Color.White;

comboBox3.SelectedIndex = -1;

comboBox3.Text = "Виберіть";

comboBox4.BackColor = Color.White;

comboBox4.SelectedIndex = -1;

comboBox4.Text = "Виберіть";

**Продовження додатку А**

comboBox5.BackColor = Color.White;

comboBox5.SelectedIndex = -1;

comboBox5.Text = "Виберіть";

comboBox6.BackColor = Color.White;

comboBox6.SelectedIndex = -1;

comboBox6.Text = "Виберіть";

comboBox7.BackColor = Color.White;

comboBox7.SelectedIndex = -1;

comboBox7.Text = "Виберіть";

VC = -1; VI = -1; VA = -1; SA = -1; VS = -1; CA = -1; VV = -1;

i++;

summary[i] = severity2;

button1.Enabled = false;

comboBox1.BackColor = Color.White;

comboBox2.BackColor = Color.White;

comboBox3.BackColor = Color.White;

comboBox4.BackColor = Color.White;

comboBox5.BackColor = Color.White;

comboBox6.BackColor = Color.White;

comboBox7.BackColor = Color.White;

}

private void Button1\_Enter(object sender, EventArgs e)

{

}

private void Form1\_MouseMove(object sender, MouseEventArgs e)

{

**Продовження додатку А**

if (textBox1.Text !="" && comboBox1.SelectedIndex >= 0 && comboBox2.SelectedIndex >= 0 && comboBox3.SelectedIndex >= 0 && comboBox4.SelectedIndex >= 0 && comboBox5.SelectedIndex >= 0 && comboBox6.SelectedIndex >= 0 && comboBox7.SelectedIndex >= 0)

{

button1.Enabled = true;

}

else

button1.Enabled = false;

if (textBox2.Text == "")

{

button3.Enabled = false;

}

else

button3.Enabled = true;

}

private void Chart1\_Click(object sender, EventArgs e)

{

}

private void Button3\_Click(object sender, EventArgs e)

{

for (int o=0;o<50;o++)

{

if (summary[o] == "Критичний")

yValues[0] += 1;

**Продовження додатку А**

if(summary[o] == "Високий")

yValues[1] += 1;

if (summary[o] == "Середній")

yValues[2] += 1;

if (summary[o] == "Низький")

yValues[3] += 1;

if (summary[o] == "Інформаційний")

yValues[4] += 1;

}

chart1.Series.Clear();

chart1.Series.Add(new Series("ColumnSeries")

{

ChartType = SeriesChartType.Doughnut

});

chart1.Series["ColumnSeries"].Points.DataBindXY(xValues, yValues);

for (int u=0;u<5;u++)

{

if (yValues[4] != 0)

{

finalstate = finalstatearr[u];

break;

}

}

textBox3.Text = "Загальний стан безпеки - " + finalstate;

}

**Продовження додатку А**

private void Button2\_Click(object sender, EventArgs e)

{

VC = -1; VI = -1; VA = -1; SA = -1; VS = -1; CA = -1; VV = -1;

impact = -1;

likelihood = -1;

severity = 1;

severity2 = "";

yValues[0] = 0; yValues[1] = 0; yValues[2] = 0; yValues[3] = 0; yValues[4] = 0;

i = 0;

textBox2.Text = "";

textBox1.Text = "";

textBox3.Text = "Загальний стан безпеки - ";

chart1.Series.Clear();

}

private void TextBox3\_TextChanged(object sender, EventArgs e)

{

}

}

}