Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління
(повна назва)

Кафедра Безпеки інформаційних технологій
(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

другий (магістерський)
(рівень вищої освіти)

Методи виявлення та протидії HID-атакам у зображеннях формату BMP

(тема)

Виконав:
студент 2 курсу, групи БІКСм-19-1
Гриньов Р.С.
(прізвище, ініціали)
Спеціальність
125 Кібербезпека
(коді і повна назва)
Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)
Освітня програма Безпека інформаційних і комунікаційних систем
(освітньо-професійна або освітньо-наукова)
Керівник доцент кафедри БІТ Сєвєрінов О.В
(посада, прізвище, ініціали)

Допускається до захисту

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

2020 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління

Кафедра Безпеки інформаційних технологій

Рівень вищої освіти другий (магістерський)

Спеціальність 125 Кібербезпека

(коді і повна назва)

Тип програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма Безпека інформаційних і комунікаційних систем

(повна назва)

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

ЗАВДАННЯ

НА АТЕСТАЦІЙНУ РОБОТУ

студентові Гриньову Ростиславу Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Методи виявлення та протидії HID-атакам у зображеннях формату BMP.

затверджена наказом по університету від 21 10 2020 р. № 1412 Ст.

2. Термін подання студентом роботи до екзаменаційної комісії 13 12 2020 р.

3. Вихідні дані до роботи HID-атаки. Формат зображень BMP.Мова програмування – Python. Сучасні засоби захисту.

4. Перелік питань, що потрібно опрацювати в роботі Аналіз основних типів вірусів та їх особливостей, методів, що використовуються під час злому систем та розповсюдження комп'ютерних вірусів. Вивчення особливостей HID атак та вірусів, що використовують для свого розповсюдження файли зображень. Розробка методів протидії данним атакам, зокрема атаці що була розроблена під час написання атестаційної роботи за напрямком бакалавра. Аналіз отриманих результатів.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) Презентаційний матеріал у вигляді слайдів

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Отримання завдання	6.05.20	Виконано
2	Робота з джерелами за тематикою роботи	7.07.20-25.08.20	Виконано
3	Вивчення принципів функціонування засобів захисту	26.08.20-7.09.20	Виконано
4	Аналіз особливостей HID-атак та вірусів, що використовують файли зображень для подолання засобів захисту	08.09.20-26.09.20	Виконано
5	Розробка методів протидії HID-атакам та виявлення вірусів у зображеннях формату BMP	27.10.20-26.10.20	Виконано
6	Аналіз отриманих результатів	27.10.20-31.10.20	Виконано
7	Публікація тез конференцій за результатами досліджень	1.11.20-1.12.20	Виконано

Дата видачі завдання 06 05 2020 р.

Студент _____
(підпис)

Керівник роботи _____ _____
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка містить 123 сторінки, 5 таблиць, 108 рисунків, 13 джерел.

Об’єкт дослідження – процес впровадження вірусів в зображення та подолання засобів захисту з використанням HID-атак.

Предметом дослідження є – методи подолання засобів захисту, що використовують HID-атаки та файли зображень для приховування вірусів.

Мета роботи – розробка методів протидії HID-атакам та виявлення вірусів у графічних файлах формату BMP.

Метод дослідження – вивчення літератури, аналіз структури та особливостей зображень формату BMP, аналіз особливостей HID-атак, розробка методів для виявлення та протидії, написання програми для виявлення та протидії HID-атакам та виявлення вірусів у графічних файлах формату BMP, аналіз отриманих результатів та написання висновків.

HID-АТАКИ, IDS, IPS, АНТИВІРУС, ВРАЗЛИВІСТЬ, ЕКСПЛОЙТ, КОМП’ЮТЕРНІ ВІРУСИ, МЕРЕЖЕВИЙ ЕКРАН, ОПЕРАЦІЙНА СИСТЕМА, УТИЛІТА, ШЕЛЛ-КОД.

ABSTRACT

The explanatory note contains 123 pages, 5 tables, 108 figures, 13 sources.

The object of the research is – the process of introducing viruses into images and overcoming protection means using HID attacks.

The subject of research is – methods of overcoming security measures that use HID attacks and image files to hide viruses.

The aim of the work is – to develop methods for countering HID attacks and detecting viruses in BMP graphic files.

The research method is – the study of literature, analysis of the structure and features of BMP images, analysis of the features of HID attacks, development methods for detecting and countering, writing a program for detecting and countering HID attacks and invoking viruses in graphic files in BMP format, analyzing the results and writing conclusions.

ANTI-VIRUS, COMPUTER VIRUSES, EXPLOIT, IDS, IPS, NETWORK SCREEN, OPERATIONAL SYSTEM, SHELL-CODE, VULNERABILITY, UTILITY.

ЗМІСТ

УМОВНІ ПОЗНАЧЕННЯ

HID – Human interface device – Тип комп'ютерного пристрою для прямої взаємодії з людиною.

IDS – Intrusion Detection System – Система виявлення атак (вторгнень).

IPS – Intrusion Prevention System — Система запобігання вторгнень.

NAT – Network Address Translation — Перетворення (трансляція) мережних адрес.

ІТС – Інформаційно-телекомунікаційна система.

ПЕОМ – Персональна електронно-обчислювальна машина.

ПЗ – Програмне забезпечення.

ВСТУП

В умовах сучасного розвитку технологій, поширеного використання мобільних пристроїв та ПЕОМ із доступом до глобальної мережі, росту популярності інтернет-ресурсів, де зазвичай обробляються персональні дані, задача захисту інформації виходить на перший план. Основними завданнями захисту є забезпечення цілісності, конфіденційності та доступності інформації.

В останні роки комп’ютерні віруси несуть серйозну загрозу, оскільки вони набули широкого поширення. Зловмисники все частіше вирисовують їх для грабунку банків, викрадення конференційної інформації та інтелектуальної власності.

Сьогодні персональні комп’ютери застосовуються масово, на жаль, це зумовило чисельну появу та розповсюдження вірусів. Ці шкідливі програми руйнують файлову структуру дисків, перешкоджають нормальній роботі і завдають шкоди інформації, що зберігається в комп'ютері, та викрадають персональні та конфіденційні дані [1].

Останнім часом в новинах все частіше з'являється інформація про різні комп'ютерні віруси. Зовсім нещодавно зараження текстових файлів вважалося неможливим - зараз цим вже нікого не здивуєш. Досить згадати появу вірусу WinWord.Concept, що перший почав вражати документи в форматі текстового процесора Microsoft Word for Windows 6.0 і 7.0. Сучасні віруси досить часто заражають текстові файли, файли формату PDF [1]. Новою тенденцією є використання штучного інтелекту як для створення вірусів і нових методів атак, так і для використання в антивірусних програмних рішеннях.

Крім того, незабаром нас чекатиме нова хвиля вірусів, яка буде використовувати для поширення файли зображень. Сьогодні дана методика поширення шкідливих програм застосовується не так часто, але її представники всім відомі (прикладом є шифрувальник Locky).

Кількість нових комп'ютерних вірусів постійно зростає, незважаючи різноманітні закони, що були прийняті в багатьох країнах для протидії комп'ютерними злочинами та активній розробці спеціальних програмних засобів захисту від вірусів [2].

Таким чином актуальність даної роботи зумовлена постійною появою нових видів вірусів, розвитком методів їх приховування, розробкою нових атак, та безперервним розвитком засобів захисту. В таких умовах важливо не тільки вчасно реагувати на появу нових загроз та розробляти методи захисту від них. Адже для виявлення нової загрози, її дослідження та розробки захисту від неї необхідний час. За цей час зловмисники використовуючи нову вразливість або атаку можуть завдати багатьом значних збитків. Тому не менш важливою є задача виявлення нових та потенційних вразливостей, методик приховування вірусного коду та подолання засобів захисту, факту злому та присутності в системі та завчасної розробки засобів захисту до того часу, як зловмисники зможуть скористатися цим та завдати шкоди.

Об’єкт дослідження – методи подолання засобів захисту, що використовують HID-атаки та файли зображень для приховування вірусів.

Предметом дослідження є – процес впровадження вірусів в зображення та подолання засобів захисту з використанням HID-атак.

Мета роботи – розробка методів протидії HID-атакам та виявлення вірусів у графічних файлах формату BMP.

Задачами роботи є:

- дослідження можливості впровадження комп’ютерних вірусів в файли зображень з метою подолання систем захисту, приховання факту проникнення в систему та ускладнення аналізу інциденту інформаційної безпеки

- розробка варіантів захисту від запропонованої атаки.

В якості цілі виступатиме ПЕОМ на базі операційної системи Windows.

Для проведення атаки використовувати мінімальний набір сторонніх програмних засобів.

1 АНАЛІЗ ОСНОВНИХ ТИПІВ ВІРУСІВ ТА ІХ ОСОБЛИВОСТЕЙ

1.1 Загальні відомості про віруси

Комп'ютерний вірус – вид шкідливої програми, що здатна створювати копії самої себе, поширюватися за допомогою різноманітних каналів зв'язку, впроваджуватися в код інших програм, завантажувальні сектори або системні області пам'яті [1]. Для маскування віруси використовують різноманітні методи. Найпростіші з них - зараження інших програм та виконання інших шкідливих дій лише при настанні певних умов. Після виконання шкідливих дій, вірус передає управління легітимній програмі, в якій знаходиться, і вона продовжує роботу у звичайному режимі. За допомогою цього досягається маскування зараженої програми під звичайну, оскільки зовні їх робота виглядає однаково.

Основна ціль вірусу – його поширення. Крім того, досить часто його супутньою функцією є пошкодження або видалення файлів, порушення роботи операційних систем або програмно-апаратних комплексів, приведення в непридатність структури розміщення даних або їх шифрування, блокування роботи користувачів і т.п.

Більшість різновидів вірусів влаштовані наступним чином, після запуску зараженої програми вірус залишається резидентно. Тобто він повертає управління оболонці операційної системи, або надбудові над операційною системою, але залишається в оперативній ПЕОМ. Завдяки цьому вірус може час від часу заражати програми та виконує шкідливі дії на комп'ютері.

Комп'ютерний вірус може зіпсувати, видалити або зашифрувати, будь-який файл що знаходиться на комп'ютері. Проте вірус здатний "заразити" деякі види файлів. Він впроваджує себе в ці файли таким чином, що вони будуть містити шкідливий код, який може почати свою роботу при деяких обставинах.

Шкідливі програми можна розділити на два основних види: віруси та хробаки [1].

Віруси – поширюються через інфікований файл, який може бути завантажений в мережі Інтернет, або можуть опинитися в неліцензійному програмному забезпеченні, часто їх поширюють за допомогою використання з використанням різноманітних методів соціальної інженерії через соціальні мережі, месенджери та інші програми для зв'язку. Вірус впроваджує шкідливий код в легітимну програму, або може замаскуватися під окрему програму. Зазвичай віруси намагаються розміститися в місцях, в які користувач рідко заходить, найчастіше це папки з компонентами операційної системи, приховані системні папки. Вірус не буде запущений, доки не буде запущена заражена програма, оскільки він не може запуститися сам.

Хробаки заражають безліч файлів на комп'ютері, наприклад всі виконувані файли, завантажувальні сектори, системні файли, і т.п. Одна з ключових особливостей хробаків те, що вони здатні самостійно проникнути в систему. Для цього вони використовують вразливості ОС, браузера або інших програм. Хробаки можуть проникати в системи та комп'ютери за допомогою чатів, електронної пошти або програм для спілкування. Крім того, вони можуть заходитися на заражених сайтах та проникати в систему використовуючи вразливості браузера. Хробаки можуть поширюватися по локальній мережі, якщо один з комп'ютерів в мережі виявиться зараженим [3].

1.2 Класифікація вірусів

Існує безліч видів і різновидів комп'ютерних вірусів. Ми розглянемо найпоширеніші останнім часом та найнебезпечніші.

Через велике різноманіття дуже важко створити єдину класифікацію комп'ютерних вірусів, проте їх можна класифікувати за такими ознаками:

- за деструктивним впливом;
- за способом зараження;
- за середовищем існування;
- за особливостями алгоритму.

За деструктивним впливом комп’ютерні віруси можна поділити на:

- "нешкідливі" віруси. Вони не несуть серйозної загрози, оскільки не перешкоджають роботі комп'ютера та не пошкоджують файли або компоненти ОС. Однак такі віруси можуть бути причиною зменшення обсяг доступної оперативної пам'яті та пам'яті на дисках [1];

- небезпечні віруси. Можуть викликати певні збої в роботі окремих програм або некритичні збої в роботі операційної системи вцілому;

- дуже небезпечні віруси. Віруси, що можуть знищити або зашифрувати певні або всі дані. Змінити або знищити системну інформацію та вивести з ладу ОС і т.п.

За способом зараження всі віруси можна поділити на:

- резидентні віруси. Зазвичай це один з різновидів файлових або завантажувальних вірусів. Причому, це – найнебезпечніший різновид. Резидентний вірус складається з двох частин. Це зроблено для того, щоб при зараженні (інфікуванні) комп'ютера резидентний вірус залишав резидентну частину функціонувати в оперативній пам'яті. Завдяки цьому вона може перехоплювати звернення ОС до різноманітних об'єктів (файлів, завантажувальних секторів і т.п.) і впроваджуватися в них. Резидентні віруси залишаються в оперативній пам'яті ПЕОМ, а отже є активними, до перезавантаження або вимикання;

- нерезидентні віруси заражають пам'ять комп'ютера і, на відміну від резидентних, є активними обмежений час.

За середовищем існування всі віруси можна поділити на:

- файлові віруси. Вони були найпоширенішими до появи мережі Інтернет. Сьогодні вони можуть заражати будь-які види виконуваних файлів різних операційних систем (так, наприклад, для Windows це виконувані файли (.elf, .exe, .com), драйвера (.sys), командні файли (.bat), динамічні бібліотеки (.dll) і т.п.).

Зараження відбувається в такий спосіб. Вірус модифікує файл, дописуючи свій код в нього та змінює його певним чином. Це необхідно, щоб при зверненні операційної системи до такого файлу (наприклад, виклик

програми з іншої програми або запуск користувачем, і т.п.) управління спочатку передавалося шкідливому коду, що може виконати будь-які дії. Після завершення роботи вірусу керування передається програмі, що виконується звичайним чином [3];

- завантажувальні віруси. Їх основна ціль - зараження завантажувальних секторів жорстких дисків. Принцип дії таких вірусів наступний. Вірус діє за схожим з файловими вірусами принципом, проте він додає свій код до спеціальної програми, що починають виконуватися до завантаження операційної системи одразу після включення комп'ютера. Основна задача такого ПЗ - підготовка і запуск операційної системи. Завдяки цьому завантажувальні віруси отримують керування та можливість виконувати шкідливі дії, до завантаження операційної системи. Такі віруси можуть записати себе в оперативну пам'ять до завантаження ОС, а отже можуть контролювати її роботу.

Часто такою функцією володіють віруси шифрувальники, вони заражають виконувані файли та головний завантажувальний сектор. Їх основне завдання – шифрування секторів диску. Від наслідків зараження таким вірусом дуже складно позбутися, недостатньо просто видалити вірус з головного завантажувального сектору і файлів, основна проблема полягає у необхідності розшифровки зашифрованих даних;

- макровіруси. Вони написані на мовах, що вбудовані в різні програмні системи. Найчастіше об'єктами зараження стають файли, що були створені різними компонентами Microsoft Office (Word, Excel і т.п.). Макровірус записує себе в DOC-файл та підміняє частину глобальних макросів собою. Після подібних маніпуляцій всі файли, що були створені та збережені в цій програмі, будуть містити макровіруси [4]. Віруси подібного типу не обмежені у функціоналі, вони можуть виконувати безліч різних деструктивних дій (наприклад, видалення всіх документів);

- мережеві віруси. Основна особливість вірусів даного типу - можливість використання різних мережевих протоколів для розповсюдження. Мережеві

віруси можуть записувати свій код на віддаленому комп'ютері різноманітними шляхами. Інтернет-хробаки - найпоширеніший тип подібних вірусів в наш час.

За особливостям алгоритмів всі віруси можна поділити на:

- найпростіші віруси – паразитичні, вони не несуть серйозної шкоди (наприклад, можуть змінювати вміст файлів). Зазвичай їх можна легко виявити і знищити [5];

- стелс-віруси – використовують різноманітні методики для приховування своєї присутності в системі. Саме через це їх дуже складно виявити. Найбільш поширений варіант забезпечення "невидимості" вірусу полягає в тому, що стелс-вірус складається з двох частин. Перша резидентна, вона постійно знаходиться в оперативній пам'яті комп'ютера. Її основне завдання - перехоплення звернення операційної системи до зараженого файлу. Резидентна частина видаляє шкідливий код з файлу, завдяки цьому додаток виявляється незараженим. Після завершення роботи з файлом ОС, резидентна частина знову заражає файл;

- поліморфні віруси. На сьогоднішній день цей вид комп'ютерних вірусів вважається найбільш небезпечним.

Поліморфні віруси – здатні модифікувати свій код. Завдяки цьому два примірники одного і того ж вірусу можуть не збігатися ні в одному біті. Це дозволяє вірусу подолати антивірусні програми, які часто використовують "маски" (уривки коду, типові для вірусів).

Поліморфні віруси бувають двох типів. Перший тип простіший, подібні віруси шифрують власний код з використанням випадкового ключа та випадкових команд дешифратора. Друга група складніша, оскільки подібні віруси здатні переписувати свій код.

Такі віруси на додачу до шифрування власного коду, можуть змінювати код модулів для шифрування та розшифрування, оскільки містять код генерації модулю для шифрування та модулю для розшифрування. Завдяки цьому віруси другого типу мають більш складну структуру, що відрізняє їх від звичайних

шифрувальних вірусів, що можуть шифрувати ділянки свого коду, проте мають при цьому незмінний код модулів шифрування і розшифрування;

- троянські програми – програми, що містять в собі деяку незадокументовану руйнівну функцію. Такі програми спрацьовують при настанні певної події або умови. Зазвичай такі програми маскуються під корисні утиліти;

- руткіти представляють собою більш складний варіант троянських програм. Однак руткіти для маскування проникають глибоко в систему;

- ботнет – це комп'ютерна мережа, що об'єднує велику кількості хостів, які називаються ботами. Найчастіше бот у складі ботнета це спеціальна програма, що приховано встановлюється на пристрій жертви. Бот дозволяє зловмиснику використовувати ресурси зараженого комп'ютера та виконувати з нього різноманітні дії;

- хробаки - віруси, що здатні вражаючи цілі системи, а не окремі програми завдяки тому, що для свого розповсюдження використовують глобальні мережі. Основна ціль таких атак - інформаційні системи великого масштабу. З появою глобальної мережі Інтернет цей вид вірусів представляє найбільшу загрозу;

- бекдор – програми, які встановлює зловмисник на скомпрометованій комп'ютерній системі після отримання початкового доступу з метою закріплення та повторного отримання доступу [4]. Основне призначення такого типу вірусу - потайне керування комп'ютером. Бекдор дозволяє передавати на уражений комп'ютер файли та програми, інші віруси для продовження атаки або завантажувати файли з ураженого комп'ютера. Крім того, зазвичай бекдор дозволяє виконувати системні операції (створення нових мережевих ресурсів, перезавантаження ПК, модифікація паролів і т.п.) або отримати віддалений доступ до реєстру [5]. Цей тип вірус по суті відкриває атакуючому "чорний хід" на комп'ютер користувача, звідси і походить назва. Небезпека даного типу вірусу збільшилася у зв'язку з тим, що багато сучасних мережевих хробаків або містять в собі бекдор-компоненту, або встановлюють її після зараження ПК. Другою особливістю багатьох подібних програм є те, що вони дозволяють

використовувати комп'ютер користувача для сканування мережі, проведення мережевих атак злому мереж - при цьому спроби злому ведуться з комп’ютера користувача, що нічого не підозрює;

- кейлогер – програмне забезпечення, що дозволяє приховано моніторити натискання клавіш та веде журнал таких натискань. Програмні кейлогери набагато розповсюдженіші, ніж апаратні, проте апаратні кейлогери не можуть бути ідентифіковані за допомогою засобів захисту, тому при захисті важливої інформації про них обов’язково потрібно пам’ятати. Варто розуміти, що наявність у програм функціоналу, що дозволяє перехоплювати натискання клавіш зустрічається значно частіше, ніж думають користувачі. Такий функціонал часто використовується для виклику функцій програми з іншої програми за допомогою "гарячих клавіш" (hotkeys). Крім того, існує багато легального ПЗ, що використовується в компаніях для моніторингу роботи працівників. Так адміністратори можуть використовувати подібні програми для того, щоб стежити за роботою працівника протягом дня. Крім того звичайний користувач може використовувати таке ПЗ для спостереження за активністю сторонніх людей на своєму комп'ютері. Однак, ті ж самі легальні програми можуть бути використані для крадіжки конфіденційної інформації. Кейлогер, на відміну від інших вірусів, представляє небезпеку тільки для користувача, а для системи він абсолютно безпечний. Небезпека для користувача зумовлена тим, що програма може перехоплювати всі дані, що вводяться з клавіатуру, а отже зловмисник отримає доступ до номерів рахунків в електронних платіжних системах, авторизаційних даних, адрес і т.п.. Для звичайного користувача все може закінчитися крадіжкою всіх грошей з банківських рахунків або втратою облікових записів. Однак використання таких вірусів може призвести до більш серйозних наслідків, ніж втрата грошей конкретною людиною. Такі програми успішно використовуються для політичного та економічного шпигунства, вони дозволяють отримати доступ до відомостей, що становлять не тільки комерційну, а й державну таємницю, а також компрометувати системи безпеки,

що використовуються комерційними та державними структурами (наприклад, за допомогою крадіжки закритих ключів в криптографічних системах);

- шелл-код (shellcode) — це двійковий виконуваний код, що передає управління командному процесору. В Unix shell це "/bin/sh", в операційних системах сімейства Microsoft Windows - "cmd.exe". Зазвичай шелл-код використовується як корисне навантаження експлойта. За допомогою такого корисного навантаження атакуючий отримує доступ до командної оболонки ПЕОМ.

Існує два типи шелл-кодів. Шелл-код першого типу – з прив'язкою до порту (port binding shellcode) відкриває заздалегідь заданий порт TCP на віддаленому ПЕОМ. Через цей порт зловмисник зможе під’єднатися до командної оболонки. Шелл-код другого типу – зворотна оболонка (reverse shell shellcode) він підключається до комп'ютера атакуючого через заданий порт. Такі шелл-коди використовуються набагато частіше, оскільки вони дозволяють подолати брандмауера або NAT (Network Address Translation).

Проведений аналіз показує велику різноманітність типів комп’ютерних вірусів та істотні відмінності у їх функціонуванні. Варто зазначити, що сучасні комп’ютерні віруси можуть не підпадати явно під цю класифікацію та поєднувати у собі особливості функціонування та розповсюдження різних типів. Це зумовлено безперервним розвитком та появою нових вірусів, методів подолання захисту та приховування факту присутності вірусу в системі. Тому необхідно розглянути основні методи, що використовуються зловмисниками для злому систем захисту, збору інформації, проведення атак та розповсюдження шкідливих програм.

2 МЕТОДИ, ЩО ВИКОРИСТОВУЮТЬСЯ ПІД ЧАС ЗЛОМУ СИСТЕМ ТА РОЗПОВСЮДЖЕННЯ КОП'ЮТЕРНИХ ВІРУСІВ

2.1 Соціальна інженерія

Соціальна інженерія - метод отримання необхідного доступу до інформації, заснований на особливостях психології людей. Основною метою атакуючого є змусити жертву виконати певні дії та/або отримати доступ до конфіденційної інформації, авторизаційних даних, банківських даних та інших захищених систем. Хоча термін соціальної інженерії з'явився не так давно, такі методи отримання інформації використовується досить довго.

Всі техніки соціальної інженерії засновані на когнітивних викривлення. Зловмисники використовують ці помилки в поведінці для створення атак, спрямованих на отримання конфіденційної інформації, часто за згодою жертви. Кіберзлочинці використовують страх, жадібність, надію та інші емоції для підвищення ефективності своїх атак.

Один з найяскравіших прикладів – ситуація, коли людина входить і вішає в будівлі компанії оголошення з інформацією про зміну телефону довідкової служби інтернет-провайдера. Таке оголошення не викличе підозри, оскільки що воно виглядає як офіційне. Коли хтось з співробітників компанії подзвонить за цим номером, для вирішення проблем, зловмисник може запитувати особисті дані для "ідентифікації" користувача. Так атакуючий зможе отримати авторизаційні дані, ідентифікатори та іншу конфіденційну інформацію.

Для того, щоб убезпечити себе від впливу соціальної інженерії, необхідно зрозуміти, як вона працює. Розглянемо основні методики соціальної інженерії і методи захисту від них.

Претекстінг - це відпрацьований набір дій за певним сценарієм, який був складений заздалегідь. Він дозволяє змусити жертву вчинити певну дію або видати будь-яку корисну інформацію. Подібна атака найчастіше вимагає

використання месенджерів або програм, що дозволяють робити відео- та аудіодзвінки, таких як Skype, телефон і т.п.

Для успішного використання цієї техніки зловмиснику спочатку необхідно зібрати інформацію про жертву (ім'я співробітника, дата народження, посада, назву проектів, з якими він працює). Зловмисник може використовувати реальні запити з ім'ям співробітників компанії для того, щоб ввійти в довіру, після чого він зможе отримати необхідну інформацію.

Фішинг - це вид інтернет-шахрайства, який спрямований на отримання конфіденційних даних користувачів, найчастіше авторизаційних або платіжних. На сьогоднішній день цей метод є найпопулярнішим методом соціальної інженерії. Перед кожним великим витоком інформації проходить хвиля фішингових розсилок. Найяскравішим прикладом такої атаки може бути повідомлення від банку або платіжної системи, що вимагає здійснення певних дій або перевірки певної інформації, відправлене електронною поштою. Зазвичай такі листи роблять максимально схожими на офіційні. Причини термінової необхідності проходження такої перевірки можуть називатися найрізноманітніші збій в системі, втрата даних та інше. Зазвичай в таких листах міститься посилання на фальшиву веб-сторінку, в точності схожу на офіційну. При переході по ньому користувач побачить форму, яка вимагає ввести конфіденційну інформацію.

Троянський кінь - ця техніка використовує емоції користувачів такі, як цікавість, страх або інші. Зловмисник відправляє електронний лист з вкладенням жертві, в якому знаходиться чек з банку з виграшем, "оновлення" антивіруса або компромат на друзів. Насправді у вкладенні шкідлива програма, завдання якої збір та викрадення або модифікації інформації [2].

Послуга за послугу - зловмисник звертається до користувача електронною поштою або корпоративним телефоном та представляється, наприклад, системним адміністратором. Після чого інформує користувача про виникнення технічних проблем на його робочому місці. Та повідомляє, що їх необхідно усунути. Зловмисник може підштовхувати жертву на вчинення певних

дій у процесі "вирішення" такої проблеми. Це дозволяє атакуючому встановити вірусне програмне забезпечення або виконати певні команди.

Дорожнє яблуко - це видозмінений метод троянського коня з використанням фізичних носіїв (флеш-накопичувачів або CD-дисків). Зловмисник "губить" портативні носій в загальнодоступних місцях на території компанії (робочі місця співробітників, парковка, їдальня, туалети). Додатково зловмисник може нанести на носій логотип компанії або провокативний підпис для того, щоб зацікавити та спровокувати співробітника. Наприклад, "зарплата співробітників", "дані про продажі", "звіт в податкову" та інше.

Зворотна соціальна інженерія. Зловмисник створює таку ситуацію, щоб жертва сама була змушена звернутися по "допомогу" до нього. Наприклад, зловмисник може наклеїти оголошення з контактами "служби підтримки" або вислати схожий електронний лист. Після чого порушник створює несправності в обладнанні жертви. В такому випадку користувач сам звернеться до порушника, який зможе отримати необхідні йому дані в процесі "вирішення" проблеми зловмисником.

Плечовий серфінг (англ. Shoulder surfing) спостереження особистої інформації жертви через її плече. Дана атака найбільш поширена та успішна в громадських місцях (торговельні центри, кафе, аеропорти, вокзали) а також в громадському транспорті.

Опитування IT-фахівців показало, що:

85% опитаних спостерігали конфіденційну інформацію, яку їм не належало знати;

82% розповіли, що інформацію на їх екранах, могли побачити сторонні;

82% не впевнені в тому, що в їх організації будь-хто буде захищати свій екран від сторонніх осіб.

Застосування технік соціальної інженерії вимагає від зловмисника не тільки глибоких знань психології. Для успішного проведення атаки також необхідне уміння збирати інформацію про людину. В наш час основним джерелом для збору такої інформації стали відкриті джерел, насамперед

соціальні мережі. Наприклад, такі сайти як "Facebook", livejournal, "Instagram", "Одноклассники", "Twitter", "ВКонтакте", "LinkedIn" містять величезну кількість корисних даних, яку люди виклали у вільний доступ. Такі дані можуть бути використані зловмисником, оскільки переважна більшість користувачів залишаючи у вільному доступі вичерпні дані та відомості про себе та не приділяють належної уваги питанням кібербезпеки.

Так наприклад, можна згадати історію про викрадення сина Євгена Касперського. Злочинці використали соціальні мережі для отримання інформації про розклад дня та маршрути руху підлітка.

Обмеження доступу до своєї сторінки в соціальній мережі, не може захистити користувача від того, що його персональна інформація ніколи не потрапить до рук шахраїв. Так бразильський дослідник з питань кібербезпеки доказав, що можна за 24 години стати другом будь-якого користувача Facebook. Дослідник використовував для цього методи соціальної інженерії. Після того, як Нельсон Новаєс Нето обрав жертву, він створив фальшивий акаунт людини з її оточення - її начальника. Впершу чергу Нето відправив запити на дружбу друзям друзів начальника жертви. На другому етапі дослідник направляв запити безпосередньо друзям жертви. Нельсон Новаєс Нето зміг за 7,5 години потрапити в друзі до жертви. Це дозволило досліднику отримати доступ до особистої інформації жертви незважаючи на те, що начальник ділився нею тільки зі своїми друзями.

Найбільш ефективними та небезпечним є методики зворотньої соціальної інженерії, рука допомоги, малих прохань, дорожнього яблука з використанням таких емоцій як страх і жадібність.

2.2 Способи розповсюдження комп’ютерних вірусів

Існує велика різноманітність способів, за допомогою яких поширюються комп’ютерні віруси. Найбільш поширені:

- електронна пошта з вкладеннями. Зловмисники створюють лист, який не викличе у жертви підозр. Наприклад, співробітнику прийшов лист і повідомляє про деякі документи. У діловому листуванні, особливо при великому потоці листів існує велика ймовірність, що людина відкриє документ та потрапить на фішингову сторінку. Також нерідко люди з цікавості відкривають вхідне повідомлення, після чого починаються проблеми. Запобіжний захід простий: якщо вам невідомо, від кого прийшов цей лист, і що може бути у вкладенні, просто не відкривайте його, а відразу видаліть. Такі заходи можуть здатися радикальними, але насправді вони є найбільш ефективними в подібній ситуації.

- Перегляд сайтів. При перегляді і використанні сайтів з неліцензійним програмним забезпеченням є велика ймовірність заразити свій комп'ютер вірусом. Шахраї спеціально створюють такі сайти для отримання доступу до комп'ютерів користувачів. Однак іноді сайт стає розповсюджувачем вірусного програмного забезпечення в результаті його злому. Щоб уникнути зараження, необхідно уникати використання таких сайтів та змінити налаштування антивірусної програми так, щоб жодне зовнішнє з'єднання не могло бути встановлено без вашої участі, а також щоб жодна програма не була встановлена на ваш комп’ютер без вашого відома.

- Мережа. У разі, якщо комп'ютер підключений до домашньої мережі, або ж є частиною великої мережі, можна легко отримати вірус не через свою провину. Один з учасників мережі отримує вірус і вже через кілька хвилин є розповсюджувачем цього вірусу.

- Фішинг. Втрата даних в результаті фішингу це нерідке явище, але при цьому легко заразити комп'ютер вірусами. При відвідуванні фальшивих сайтів, програми шпигуни часто автоматично встановлюють віруси на комп'ютери [2]. У разі виникнення підозр найкраще відразу зателефонувати в банк і перевірити, чи все в порядку, а не відкривати сумнівні посилання.

- ПЗ з вірусами. Схема проста - користувач завантажує ПЗ, яке заражене вірусом. Таким чином шкідливий код потрапляє на комп'ютер. Завдяки

широкому розповсюдженню неліцензійного програмного забезпечення віруси активно розповсюджуються мережею Інтернет, оскільки користувачі не хочуть платити за ліцензійне програмне забезпечення.

- Хакери.

- Лжеантивірусне програмне забезпечення. Воно є найпоширенішим шляхом інфікування. Завантажуючи антивірусну програму з неперевіреного джерела, ви піддаєтеся ризику інфікувати свій комп'ютер. Нерідко це безкоштовні антивіруси від невідомих виробників. Оптимальним рішенням є покупка ліцензійного антивіруса, який захистить ваш комп'ютер.

- Переносні носії. При відсутності антивірусної програми, що здатна на льоту перевіряти флеш накопичувачі дуже високий ризик зараження через такі носії. Розповсюдження вірусів за допомогою флеш-накопичувача є одним з найбільш масових та ефективних способів поширення комп’ютерних вірусів, через високе поширення з'ємних носіїв. Нерідко зловмисники застосовують методики соціальної інженерії, такі як метод дорожнього яблука.

Розглянемо особливості функціонування та розповсюдження шифрувальника Locky, який використовує для свого розповсюдження файли зображень, а саме зображення формату SVG.

2.3 Аналіз особливостей розповсюдження шифрувальника Locky

Широке використання та всесвітню відомість шифрувальник Locky отримав у лютому 2016 році. Він поширювався через зображення формату SVG в соціальній мережі "Facebook". Було зафіксовано спроби зараження користувачів цим шифрувальником у 114 країнах світу.

Даний вірус з поміж усіх інших гучних спалахів заражень виділяють особливості функціонування. На відміну від більшості вірусів в тому числі і шифрувальників, що розповсюджуються за допомогою поштового спаму, набору експлойтів (exploit kit) та шкідливої реклами, розробники шифрувальника Locky вигадали новий метод.

Спочатку спам-листи (рисунок 2.1) містили у додатках файли формату DOC з макросом, що завантажував з віддаленого сервера та запускав шифрувальник.

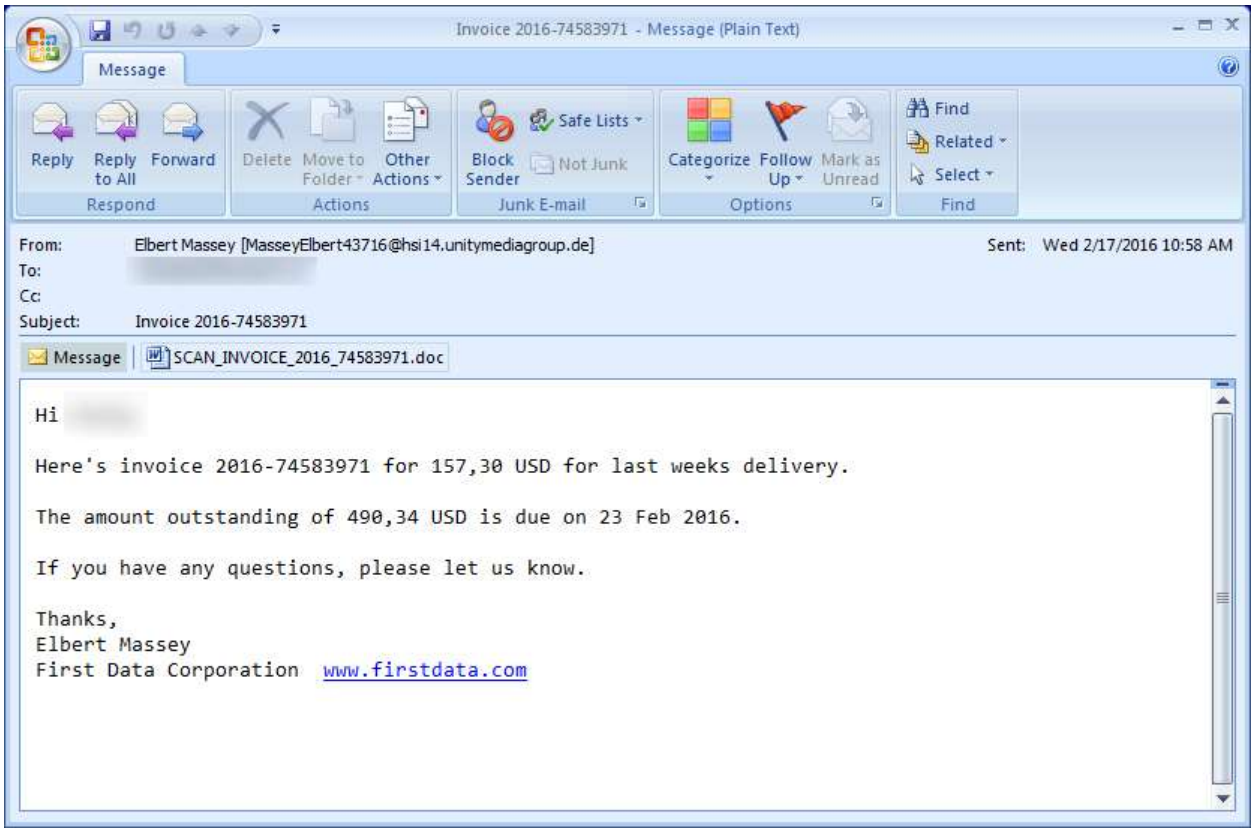

Рисунок 2.1 – Лист перших спам-розсилок із прикріпленим шкідливим документом

З часом антивірусні продукти почали визначати файли зі шкідливим макросом, як Trojan-Downloader.MSWord.Agent та HEUR:Trojan-Downloader.Script.Generic.

Однак, варто зазначити, що в сучасних версіях пакету Microsoft Office вимкнене автоматичне виконання макросів з міркувань безпеки. Проте статистика свідчить, що користувачі дуже часто вмикають макроси вручну, навіть в документах невідомого походження, що призводить до поганих наслідків.

Після цього під час спам-розсилок (рисунок 2.2) зловмисники почали використовувати замість текстових документів формату DOC ZIP-архіви, що містили декілька обфускованих скриптів написаних на мові JavaScript.

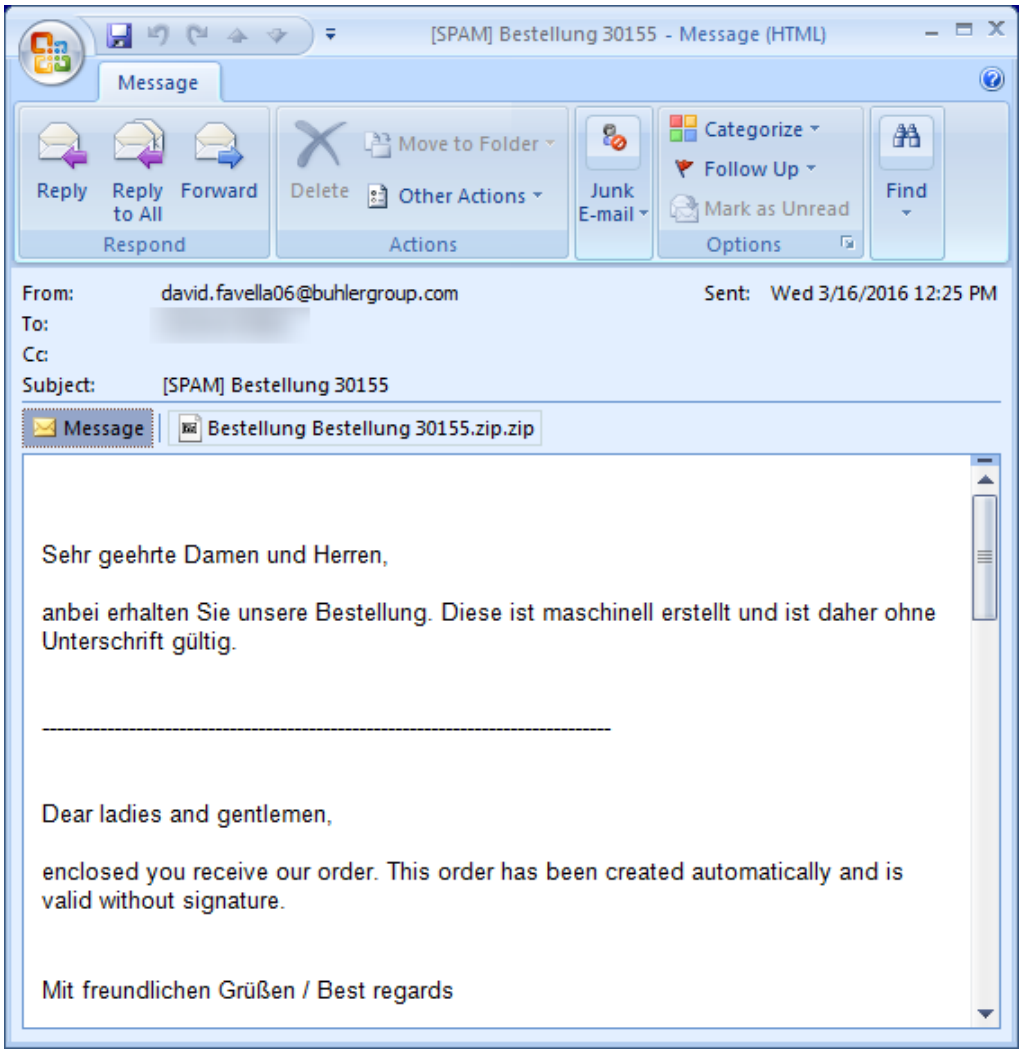

Рисунок 2.2 – Приклад листа з другої спам-компанії

Зловмисники спонукали жертв запустити скрипти вручну за допомогою шахрайських методик в тому числі і соціальної інженерії. Вміст прикріпленого архіву можна побачити на рисунку 2.3.

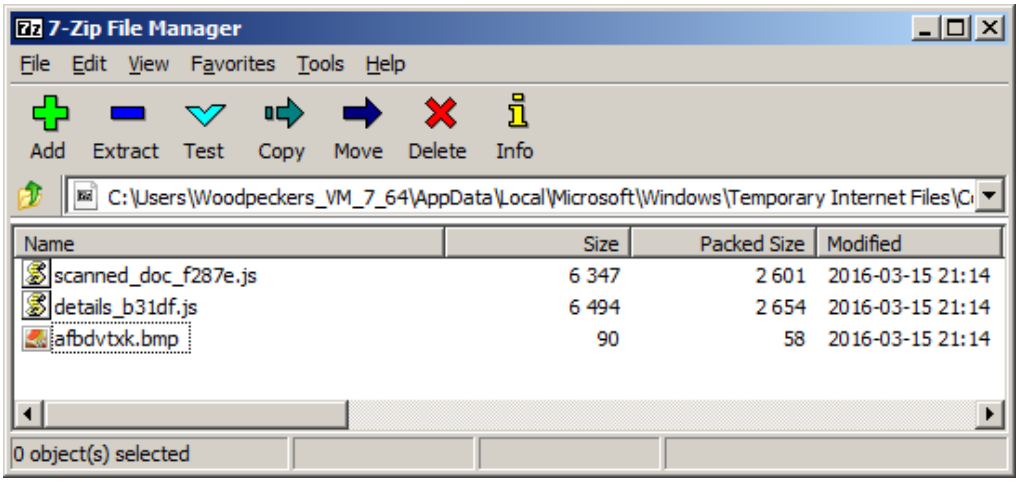

Рисунок 2.3 – Вміст прикріпленого до листа архіву

Після запуску скрипт завантажував з віддаленого сервера та виконував шифрувальник Locky. Такі скрипти з часом почали визначатися антивірусами, як Trojan-Downloader.JS.Agent и HEUR:Trojan-Downloader.Script.Generic.

У 2016 році дослідники Барт Блейз (Bart Blaze) і Пітер Круз (Peter Kruse) виявили в соціальній мережі "Facebook" нову спам-кампанію. Зловмисники поширювали завантажувач Nemucod, який завантажував на персональний комп'ютер жертви шифрувальника вимагача Locky. Все це відбувалося за допомогою SVG-зображень.

Розберемо детальніше формат SVG. Це порівняно молодий формат, що використовується для векторних зображень. Він підтримує нерухому (декларативну) та анімовану інтерактивну (скриптову) графіку. Підтримка форматом ECMAScript та JavaScript дозволяє описувати складні сценарії. Зловмисників формат SVG зацікавив саме завдяки своїм особливостям, а саме, через те, що в його основі лежить XML, формат допускає використання динамічного контенту та має підтримку JavaScript. Шахраї додають шкідливий JavaScript-код безпосередньо в код зображення. Це може бути посилання на зовнішній файл.

Після того, як користувач отримає таке зображення в месенджері та натисне на нього, він потрапить на замаскований під YouTube сайт. Після цього спливаюче вікно проінформує жертву про те, що необхідно встановити спеціальне розширення для перегляду відео, яке називається Ubo або One. Через те, що у розширення відсутня іконка, воно здається невидимим.

Саме за рахунок цього розширення і поширюється спам, оскільки воно отримує доступ до аканту "Facebook" жертви через браузер. Після цього розширення починає масово надсилати друзям SVG зображення з вірусом. Крім того на комп'ютер користувача також завантажується вірус Nemucod. Який в свою чергу дозволяє шифрувальник Locky проникнути в заражену систему.

За даними KSN, атаки Locky були зафіксовані в 114 країнах.

Таблиця 2.1 – Країни, що найбільше постраждали

Країна	Кількість атакованих користувачів
Франція	469
Германія	340
Індія	267
США	224
ЮАР	182
Італія	171
Мексика	159
Бразилія	156
Китай	126
В’єтнам	107

В свій час шифрувальник Locky отримав всесвітню відомість та зміг заразити велику кількість користувачів саме завдяки нестандартному підходу зловмисників. Проте використання зображень формату SVG одночасно з перевагами (підтримка JavaScript, та відносно простий спосіб запуску шкідливого скрипта) має свої суттєві недоліки. Ця атака є спорідненою з атаками, в яких використовуються файли формату PDF з використанням JavaScript та файли, що є результатами роботи продукту Microsoft Office (використання макросів для атаки). Всі ці атаки мають суттєвий недолік. Переважна більшість засобів захисту в наш час обов’язково перевіряють файли, що мають підтримку мов сценаріїв (в тому числі JavaScript, VBS) та можуть виявляти шкідливі скрипти. Крім того багато засобів захисту можуть просто блокувати передачу та роботу в таких файлах. Крім того в багатьох програмах передбачений режим роботи з відключеною підтримкою скриптів. Тому в роботі

розглянутий формат зображень BMP, оскільки він не має підтримки мов сценаріїв, не вирізняється з поміж інших файлів та не привертає уваги адже має просту структуру, проте особливості будови цього формату дозволяє оминути засоби захисту, в тому числі системи виявлення вторгнень (IDS), системи запобігання вторгнень (IPS), "пісочниці", антивіруси та брандмауер.

3 КОНЦЕПЦІЯ АТАКИ З ВИКОРИСТАННЯМ ФАЙЛІВ ЗОБРАЖЕНЬ

3.1 Аналіз структури та особливостей файлів зображень формату BMP

Комп'ютерні віруси представляють серйозну загрозу як для великих фірм і компаній, так і для простих користувачів. Сьогодні існує величезна різноманітність комп'ютерних вірусів. Створюються нові методики їх поширення та приховування [6]. Останнім часом все популярнішими стають методи, що використовують стеганографію для приховування вірусного програмного забезпечення в файлах.

У наш час організацій використовуються різноманітні засоби захисту для забезпечення конфіденційності даних. До таких засобів можна віднести антивірусне програмне забезпечення, брандмауери, системи виявлення вторгнень (IDS) та системи запобігання вторгнень (IPS). Проте навіть вони не можуть гарантувати захисту [7].

Зловмисники можуть використовувати різноманітні методики для обходу антивірусів, пісочниць, IDS/IPS, наприклад, впроваджувати комп'ютерні віруси в зображення. До того, як вірус запуститься на комп'ютері співробітника, його встигнуть проаналізувати різноманітні засоби захисту (рисунок 3.1).

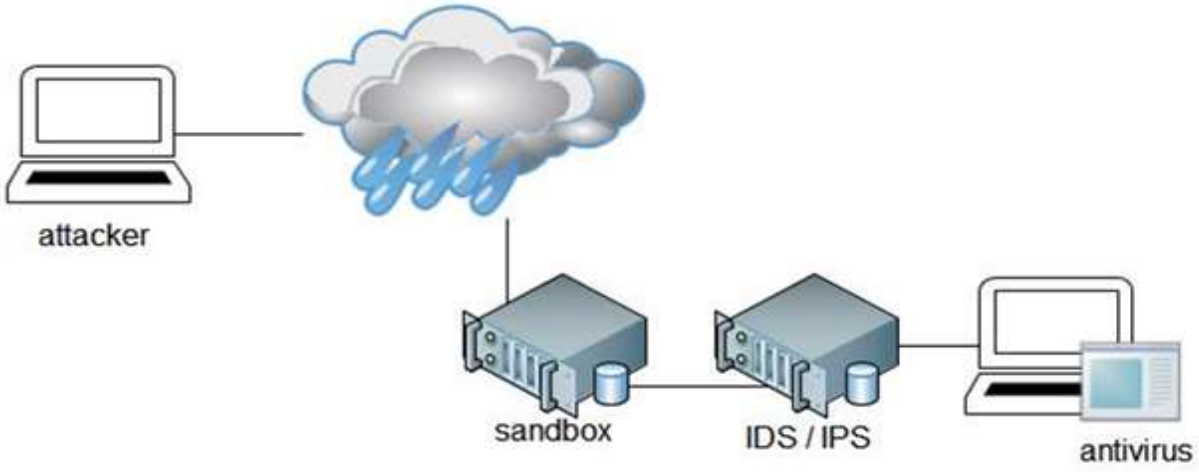

Рисунок 3.1 – Візуалізація шляху вірусу до цільового комп'ютера

Більшість методів аналізу включають використання відбитків і аналіз поведінки в пісочниці, а саме:

- перевірка обсягу вільної пам’яті диска,
- перевірка працюючих процесів,
- перевірка поточного домену,
- перевірка обсягу вільної оперативної,
- перевірка часу безвідмовної роботи.

Зловмисники постійно розробляють нові функції та методики для приховування шкідливого коду. Пісочниці зазвичай будуть аналізувати тільки файли певних потенційно небезпечних типів виконувані файли, документи, що були створені в пакеті Microsoft Office та подібних, бібліотеки динамічних посилань (DLL), аплети Java. Зазвичай переважна більшість із засобів захисту не перевіряють зображення або інші безпечні типи файлів. Це зроблено з метою підвищення швидкодії та економії ресурсів, вони вважають, що немає причин витрачати процесорний цикл на аналіз зображення [8].

Проаналізуємо можливість використання зображення в якості контейнеру для приховування комп’ютерного вірусу та подолання засобів захисту. Крім того даний метод дозволить суттєво ускладнити розбір інциденту інформаційної безпеки, оскільки буде прихований канал та сам факт проникнення в систему.

Розглянемо формат зображень BMP. Кожне зображення такого формату має заголовок файлу, заголовок зображення, растрові дані та карту кольорів (крім зображень з 24-бітним кольором) [8]. Детально проаналізуємо структуру заголовку файлу формату BMP (таблиця 3.1). Це 14-байтна структуру, що розташовується на початку файлу та містить інформацію про тип та розмір файлу та розташування даних. Далі знаходиться структура заголовку зображення (таблиця 3.2), в якій містяться дані про розмір, колір та стиск зображення.

Поле Compression відповідає за визначення типу стиску, що використовується. Якщо його значення 0, то стиск відсутній. Значення RLE-4 або RLE-8 вказує на те, що використовується метод стиску груповими координатами із 4-бітами/піксель та 8-бітами/піксель відповідно.

Таблиця 3.1 – Структура заголовку файлу формату BMP

Зміщення	Розмір (байт)	Ім’я	Опис
0	2	Type	Сигнатура формату. Використовується для ідентифікації формату. Має бути 4D42(hex)/424D(hex) (little-endian/big-endian). Після приведення до системи ASCII-символів має вигляд “BM”.
2	4	Size	Розмір файлу в байтах.
6	2	Reserved 1	Зарезервоване поле має містити 0.
8	2	Reserved 2	Зарезервоване поле має містити 0.
10	4	OffsetBits	Положення піксельних даних відносно початку файлу (в байтах).

Таблиця 3.2 – Структура заголовку зображення

Зміщення	Розмір (байт)	Ім’я	Опис
14	4	Size	Довжина заголовку.
18	4	Width	Ширина зображення в пікселях.
22	4	Height	Висота зображення в пікселях.
26	2	Planes	Кількість площин.
28	2	BitCount	Глибина кольору, біт на піксель (1, 4, 8, 24).
30	4	Compression	Тип компресії (0 – відсутня, 1 – RLE-8, 2 – RLE-4).

Продовження таблиці 3.2

34	4	SizeImage	Розмір зображення, байт (включно з доповненням).
38	4	XpelsPerMeter	Горизонтальна роздільна здатність, пікселів на метр.
42	4	YpelsPerMeter	Вертикальна роздільна здатність, пікселів на метр.
46	4	ColorsUsed	Число кольорів, що використовується (0 – максимально можливе для даної глибини кольору).
50	4	ColorTable	Кількість основних кольорів.

Найбільшу увагу привертають поля Size (розмір файлу BMP в байтах), XpelsPerMeter та YpelsPerMeter (горизонтальна та вертикальна роздільна здатність, пікселів на метр) та зарезервовані поля, адже вони є ненадійними [8]. Звичайний заголовок зображення формату BMP починається з рядка 42 4D XX XX XX XX 00 00 00 00. Так наприклад відкривши зображення чорного прямокутника (рисунок 3.2) в hex-редакторі ми побачимо наступний заголовок (рисунок 3.3).

Рисунок 3.2 – Приклад малюнку в форматі BMP

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Текст декодиров
00000000	42	4D	66	D6	00	00	00	00	00	00	36	00	00	00	28	00	BMfЦ.....6...(
00000010	00	00	C6	00	00	00	5C	00	00	00	01	00	18	00	00	00	..Ж...\.....
00000020	00	00	30	D6	00	00	00	00	00	00	00	00	00	00	00	00	..0Ц.....
00000030	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000040	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....

Рисунок 3.3 – Початок заголовку зображення

Як видно з попереднього рисунку, початок заголовку файлу має вигляд (рисунок 3.4).

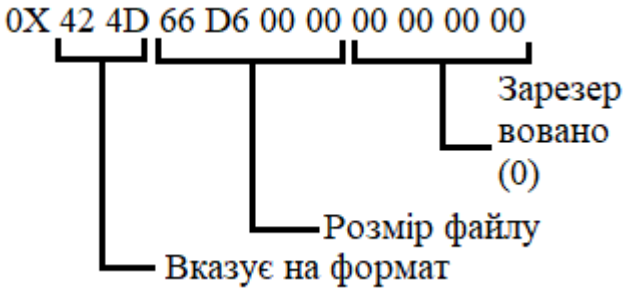

Рисунок 3.4 – Заголовок зображення

Відкривши будь-який hex-редактор, можна змінити зарезервовані поля та поля, що містять інформацію про розмір файлу. Запишемо в ці ділянки слово “testtest” (рисунок 3.5).

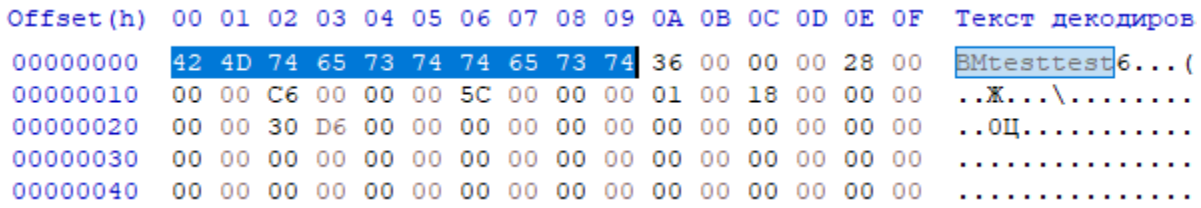

Рисунок 3.5 – Зображення зі зміненим заголовком

Тепер заголовок зображення матиме вигляд (рисунок 3.6).

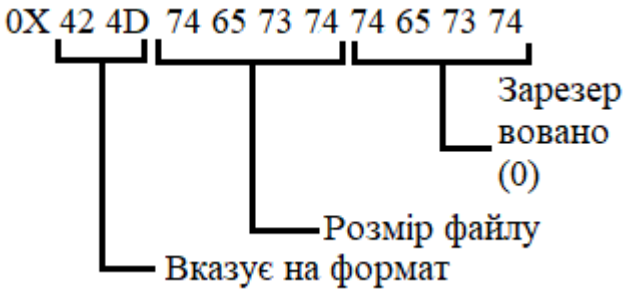

Рисунок 3.6 – Змінений заголовок зображення

Проте на самому зображенні не буде ніяких спотворень (рисунок 3.7). Це зумовлено тим, що ці поля службові і не використовуються для відображення інформації. Варто зазначити, що розмір файлу також залишиться без змін.

Рисунок 3.7 – Зображення зі зміненими полями

Перше на що звертається увага при розборі формату файлу BMP - це заголовок, що починається з 0x 42 4D. Він вказує на тип файлу (BM), його ключова особливість полягає в тому, що при конвертації в інструкції асемблера ми отримаємо, що 42 – це inc edx, а 4D – dec ebp. Це означає, що якщо ці інструкції будуть записані перед кодом основної програми, то вони не викличуть збоїв, крім того вони не мають команд переходів [8].

Якщо змінити значення декількох останніх рядків в hex-редакторі, то на зображенні виникнуть спотворення (рисунок 3.8). Значення дорівнювали 00, оскільки зображення було повністю чорним.

Offset(h)	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Текст декодирован
0000D560	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0000D570	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0000D580	00	00	54	68	69	73	20	70	6C	61	63	65	20	69	73	...This place is
0000D590	65	6E	6F	75	67	68	20	74	6F	20	61	63	63	6F	6D	enough to accom
0000D5A0	6F	64	61	74	65	20	6D	61	6C	69	63	69	6F	75	73	modate malicious
0000D5B0	63	6F	64	65	2C	20	77	68	69	63	68	20	77	69	6C	code, which wil
0000D5C0	20	62	65	20	68	69	64	64	65	6E	20	69	6E	20	74	l be hidden in t
0000D5D0	65	20	69	6D	61	67	65	2E	20	42	75	74	20	64	75	he image. But du
0000D5E0	20	74	6F	20	74	68	65	20	63	68	61	6E	67	65	73	e to the changes
0000D5F0	74	68	65	72	65	20	77	69	6C	6C	20	62	65	20	6E	there will be n
0000D600	74	69	63	65	61	62	6C	65	20	76	69	73	75	61	6C	oticeable visual
0000D610	64	69	73	74	6F	72	74	69	6F	6E	73	2E	20	49	66	distortions. If
0000D620	61	72	74	69	66	69	63	69	61	6C	6C	79	20	72	65	artificially re
0000D630	75	63	65	20	74	68	65	20	68	65	69	67	68	74	20	duce the height
0000D640	66	20	74	68	65	20	69	6D	61	67	65	2C	20	74	68	of the image, th
0000D650	6E	20	74	68	65	79	20	63	61	6E	20	62	65	20	68	en they can be h
0000D660	64	64	65	6E	2E											idden.

Рисунок 3.8 – Змінене зображення

При відкритті такого зображення, можна побачити, спотворені пікселі, що з’явилися у правому верхньому куті (рисунок 3.9).

Рисунок 3.9 – Збільшений правий кут зміненого зображення

Проте ми можемо скористатися hex-редактором, та штучно зменшити висоту зображення, що дозволить приховати спотворені пікселі (рисунок 3.10).

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Текст декодиров
00000000	42	4D	74	65	73	74	74	65	73	74	36	00	00	00	28	00	BMtesttest6...(
00000010	00	00	C6	00	00	00	5B	00	00	00	01	00	18	00	00	00	..Ж...[.....
00000020	00	00	30	D6	00	CC	00	00	00	00	00	00	00	00	00	00	..0Ц.М.....
00000030	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000040	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....

Рисунок 3.10 – Штучне зменшення висоти зображення на 1 піксель

Після цих маніпуляцій відкривши зображення ми не помітимо нічого дивного (рисунок 3.11).

Рисунок 3.11 – Зображення зі штучно зменшеною висотою

Отже можливо впровадити вірус в зображення формату BMP так, щоб користувач не помітив змін у файлі, він не побачить нічого підозрілого. Оскільки ми зможемо приховати біті пікселі на зображенні, завдяки особливостям зображень формату BMP. Як було продемонстровано вище, це легко можна

зробити модифікувавши відповідні поля в заголовку, тим самим штучно зменшивши висоту зображення на декілька пікселів. Ін'єкція можлива через те, що байти, які вказують на тип файлу, з яких і починається файл, BM в ASCII, в шістнадцятковому вигляді – 42 4D, при конвертації в інструкції асемблера не призводять до помилки виконання, а подальші 8 байт заголовка ніяк не впливають на інтерпретацію зображення. Ці 8 байт можна заповнити будь-якими інструкціями асемблера, наприклад, записати в них jmp-інструкцію, яка вкаже на вірус, що зберігається в зображенні [8]. Наприклад, функція jmp могла вказувати на адресу 0000D583, де починається текст.

За допомогою простих маніпуляцій ми змогли отримати зображення, що містить вірус. Така можливість використовування зображень формату BMP в якості контейнерів для подолання засобів захисту обумовлена особливостями даного формату. Цей метод приховування шкідливого коду дозволить нам уникнути аналізу та виявлення пісочницями. Однак варто пам’ятати, що антивіруси та IPS/IDS можуть виконати статичний аналіз файлу та виявити вірус. Ми будемо використовувати обфускацію для приховування вірусу від цих засобів захисту [8]. Найпростіші варіанти обфускації використовують найпростіших логічних операцій для кодування. Ми будемо використовувати операцію XOR. Обфускація буде досягатися за рахунок сили ключа в якості якого буде використовуватися 32-х бітне значення в діапазоні від 0x11111111 до 0xffffffff, це дозволить уникнути жорстко закодованого коду.

Наприклад, припустимо, що ключ дорівнює 0x39643964 маємо (рисунок 3.12).

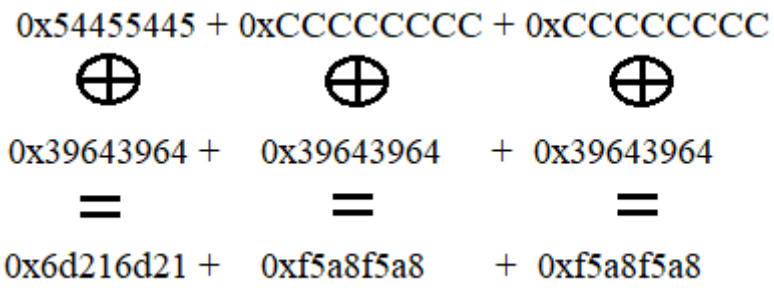

Рисунок 3.12 – Приклад результату операції обфускації

Для виконання коду, що буде прихований в зображенні, будемо використовувати готове корисне навантаження PowerShell від Cobalt Strike. Це дозволить нам завантажити зображення та виконати код в пам’яті. Цей скрипт використовує System.Net.WebClient для завантаження файлу, зображення, та VirtualAlloc і CreateThread для зчитування та виконання коду.

Найбільша небезпека розробленого методу подолання засобів захисту полягає в тому, що для протидії необхідно використовувати нестандартні методи [8]. Найпростіший варіант - зміна налаштувань засобів захисту таким чином, щоб вони перевіряли файли будь-яких типів. Однак такий метод буде вкрай неефективним та крім того це суттєво сповільнить або навіть повністю паралізує роботу всієї ІТС. Крім того, використання подібних методів зловмисниками дозволить сильно ускладнити розбір інциденту інформаційної безпеки в організації. По-перше, засоби захисту не відреагують на вірус, що унеможливить виявлення факту проникнення на початкових етапах. По-друге, якщо такий факт буде встановлений, то буде майже не можливо з’ясувати яким саме чином воно відбулося. Це обумовлено тим, що працівникам з відділу безпеки буде необхідно обробити величезний обсяг даних, тому, в першу чергу вони будуть з’ясовувати, які виконувані файли, документи Microsoft Office, файли PDF, бібліотеки DLL потрапили в систему та використовувалися за цей період. Обсяг інформації, яку треба обробити буде дуже великим через те, через те, що невідома навіть приблизна дата проникнення. В цьому випадку ніхто з працівників не буде досліджувати файли зображень.

Важливо пам'ятати, що віруси можуть заражати не тільки виконувані файли і динамічні бібліотеки, а й файли зображень, аудіо та відео.

Оскільки зображення не можна запустити, як виконуваний файл, то засоби захисту не звернуть на нього уваги, а технічні фахівці можуть знехтувати цією загрозою, оскільки легковажно ставляться до вмісту таких файлів. Файли зображень можуть нести серйозну загрозу. Тому необхідно більш ретельно проводити розслідування інцидентів інформаційної безпеки та уважно ставитися до налаштування систем IDS/IPS.

Проведений в роботі аналіз підтвердив, що зображення формату BMP можливо використовувати для приховування комп'ютерних вірусів та подолання засобів захисту. Існує багато методів для розповсюдження інфікованого зображення та виконання вірусного коду. Впровадити прихований вірусний код у вже існуючий виконуваний файл можна за допомогою будь-якого скрипта або виконуваного файлу. Цей метод дозволить оминути засоби захисту та залишитися непоміченим. Адже під час сканування засоби захисту не виявлять у них вірус. Проте більш надійним та перспективним є використання апаратних закладок у поєднанні з цим методом.

3.2 Аналіз можливості використання апаратних закладних пристроїв на базі запрограмованих мікроконтролерів для проведення атаки

Змінні носії даних часто стають джерелами поширення комп'ютерних вірусів. Якщо раніше для цих цілей використовувався файл autorun.inf в корені з'ємного носія даних, то останнім часом почастішали атаки в яких шкідливий код записується безпосередньо в мікроконтролер.

Сфери застосування подібного мікроконтролера, який в результаті стає шкідливим пристроєм, можуть бути різні. Подібні пристрої можуть використовуватися системними адміністраторами, інженерами, під час проведення прихованого пентеста в компанії фахівцями з безпеки , а також використовуватися зловмисниками [9].

Замаскувавши такий пристрій під периферійне обладнання, наприклад, клавіатура, маніпулятора “миша” або флеш накопичувача, можна суттєво підвищити шанси на те, що ним скористається співробітник організації. Подібні пристрої можуть надсилатися поштою в якості “сувенірів” або просто “губилися” поблизу організації, наприклад, на парковці. Як правило, досить високий відсоток користувачів, які з впевненістю підключать таке обладнання до комп’ютера, оскільки вони не замислюються про справжнє призначення

пристрою. При цьому, ні система, ні, засоби захисту не помітять нічого підозрілого, оскільки пристрій буде визначений як звичайна клавіатура [9].

У подібних ситуаціях вектор атаки лежить на стику технології та соціальної інженерії. Для цього зловмиснику необхідно самостійно або за допомогою користувача підключити схожий пристрій до комп'ютера, він визначиться в системі як пристрій введення і самостійно виконає запрограмовані дії. Таким пристроєм, наприклад, може бути Arduino-подібний мікроконтролер, що емулює клавіатуру, з подальшим виконанням коду на системі, що атакується.

HID, або Human Interface Device – тип комп’ютерного пристрою, який взаємодіє безпосередньо з людиною. Зазвичай такі пристрої приймають від оператора вхідні дані і надає йому результат (вихідні дані) [9]. Найпоширеніші типи HID-пристрої – це маніпулятор “миша”, клавіатура, джойстики. З точки зору ПЕОМ HID-пристрої розглядаються як простий інтерфейс взаємодії користувача та машини та є повністю довіреними. Тому, при під’єднанні до комп’ютера нового маніпулятора “миша” або клавіатури система, не запитуючи дозволу, автоматично встановлює драйвери. Така безмежна довіра до HID пристроїв може поставити під удар безпеку всієї ІТС.

Шкідливі HID пристрої можуть бути запрограмовані на виконання різноманітних операцій: від простого збору інформації про систему, отримання авторизаційних даних до повного контролю всієї системи і процесів з подальшим поширенням і зараженням нових пристроїв [9].

Через неправильну утилізацію обладнання або під час ремонту після якого були встановлені апаратні закладні пристрої відбувається велика кількість витоків інформації з організації. Наприклад, комп’ютер, що потрапив до ремонту на жорсткому диску якого знаходились конфіденційні документи. Проте правильна утилізація і виключення схожих ситуацій не гарантує безпеку. З часом виходить з ладу устаткування в будь-якій організації. Це може бути клавіатура або мережевий пристрій. Після заміни або ремонту звичайного маніпулятора “миша” ніхто не буде перевіряти таке обладнання на наявність зайвого мікроконтролера, що може виконувати шкідливі дії. Незважаючи на те що такі атаки досить специфічні і мало

розповсюджені вони дуже небезпечні [10]. Ця проблема може стати досить поширеною в Україні в найближчі декілька років, оскільки в нашій державі дуже мала кількість сертифікованих сервісних центрів. Оскільки їх послуги дорого коштують. З точки зору звичайного користувача периферійне обладнання не може становити небезпеки для ПЕОМ або організації. Крім того, схожий пристрій може встановити звичайний майстер, якого викликали виправити несправності.

Подібні пристрої можуть бути досить різноманітними. Складні пристрої можуть мати підтримку бездротових інтерфейсів, або можливість доступу до них через мережу Інтернет, що дозволить зловмиснику взаємодіяти з ними інтерактивно під'єднавшись до них дистанційно [11]. Більш прості та компактні варіанти запрограмовані на виконання конкретних дій. Такі пристрої можуть бути легко приховані наприклад, в маршрутизаторі, комутаторі, системному блоці персонального комп'ютера, та іншому обладнанні. Основна небезпека подібних атак в тому, що такі пристрої непомітні, а отже важко виявити сам факт компрометації. Такі пристрої можуть бути зловмисниками в якості плацдарму для проведення багатьох атак, оскільки їх основна перевага в тому, що вони можуть залишатися непоміченими роками.

Arduino Leonardo Micro був обраний в якості подібного шкідливого USB-пристрою (рисунок 3.13), оскільки він має невеликі габарити, достатню кількість пам'яті, підтримку USB та низку вартість [9]. Прошивка може створена в середовищі розробки Arduino Development Environment. За допомогою цього середовища для розробки можна скорегувати та записати програму в мікроконтролер. Розробка коду здійснюється за допомогою С-подібного синтаксису.

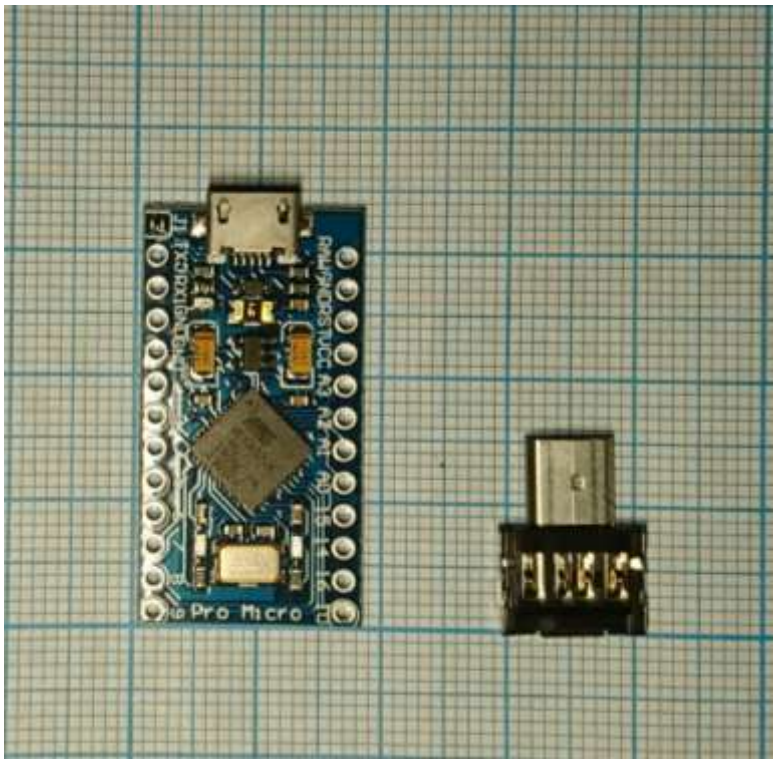

Рисунок 3.13 – мікроконтролер Arduino Leonardo Micro

Даний мікроконтролер може під’єднуватися до порту USB комп’ютера. Arduino Leonardo Micro має наступні характеристики:

- Мікроконтролер: ATmega32u4
- Гранична напруга живлення: 5-20 В
- Рекомендована напруга живлення: 7-12 В
- 18 цифрових вводів/виводів
- Максимальна сила струму: 40 mAh з одного виводу і 500 mAh з усіх виводів.
- Flash пам'ять: 32 КБ
- SRAM: 2,5 КБ
- EEPROM: 1 КБ
- Тактова частота: 16 МГц

Приховувати мікроконтролер будемо в маніпуляторі “миша” Real-El RM-207 (рисунок 3.14). Для цього нам необхідний USB хаб, в нашому випадку це Lapara LA-UH4372 (рисунок 3.15).

Рисунок 3.14 - Маніпуляторі “миша” Real-El RM-207

Рисунок 3.15 – USB хаб Lapara LA-UH4372

Всі пристрої мають інтерфейси USB 2.0. Контакти інтерфейсу наведені на рисунку 3.16.

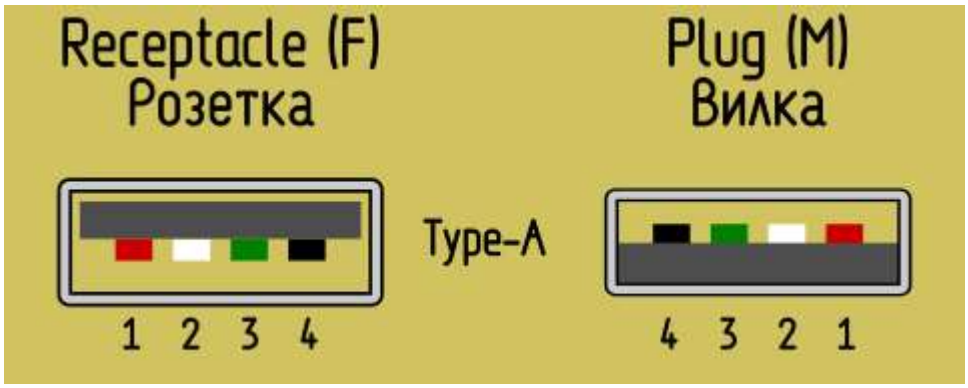

Рисунок 3.16 – Контакти USB

Як видно з рисунку, інтерфейс USB має 4 контакти, 2 з яких інформаційні. Червоний VBUS (+ 5V, Vcc - Voltage Collector Collector) +5 вольт постійної напруги відносно GND. Для USB 2.0 максимальний струм - 500 mA. Білий D- (-Data). Зелений D+ (+Data). GND – “земля”.

Необхідно розібрати маніпулятор “миша” (рисунок 3.17) та USB хаб (рисунок 3.18) та з’єднати відповідні контакти, керуючись маркуваннями на платах.

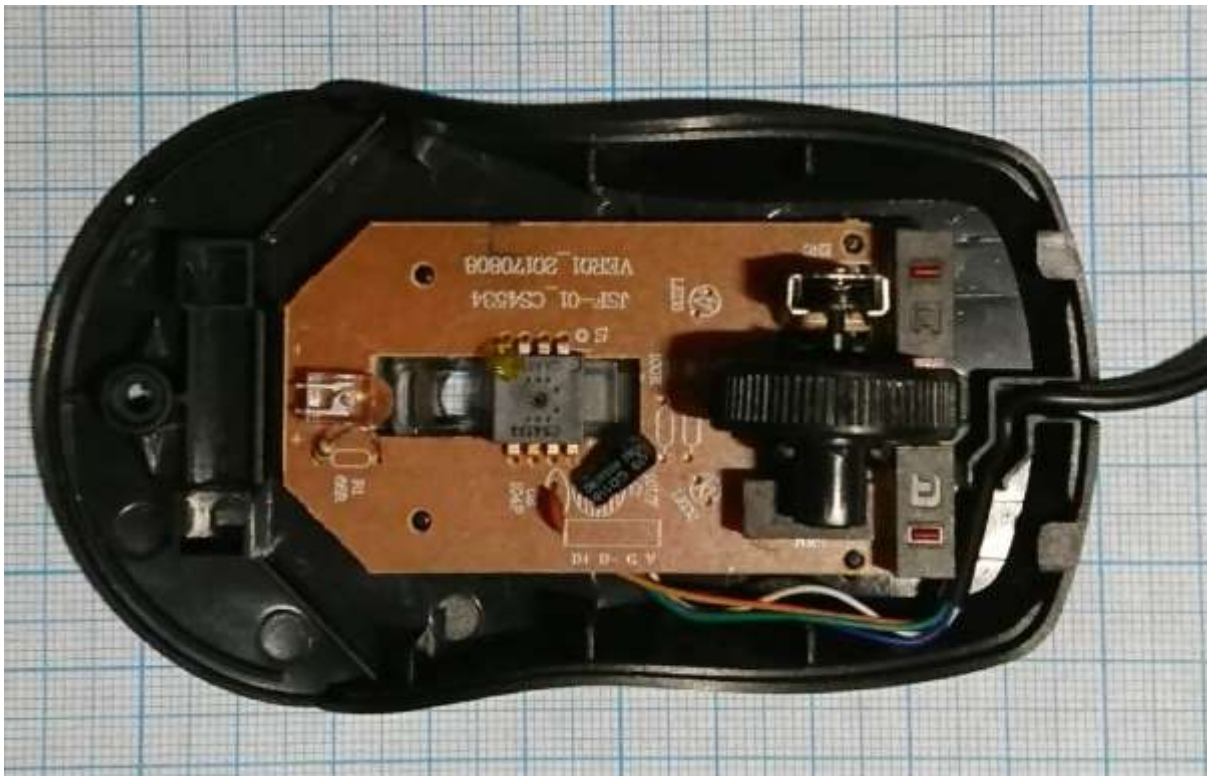

Рисунок 3.17 – Маркування на контактах USB інтерфейсу маніпулятора “миша”

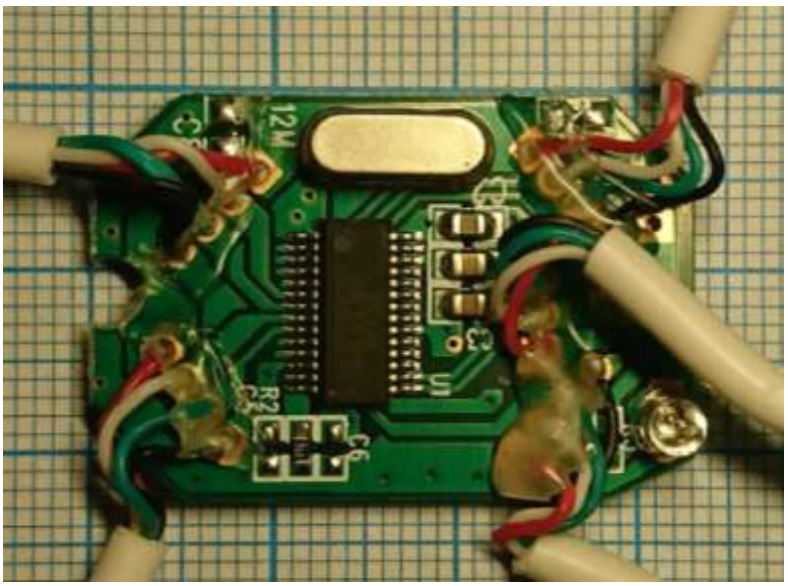

Рисунок 3.18 – Контакти USB хабу

Спочатку необхідно відокремити дріт “миші” та під’єднати його до входу USB хабу. Після цього необхідно під’єднати один з виходів хабу до плати маніпулятора та видалити 2 USB інтерфейси хабу. До інтерфейсу, що залишився

під’єднаємо Arduino. Після всіх маніпуляцій модифікований маніпулятор “миша” виглядає наступним чином (рисунок 3.19).

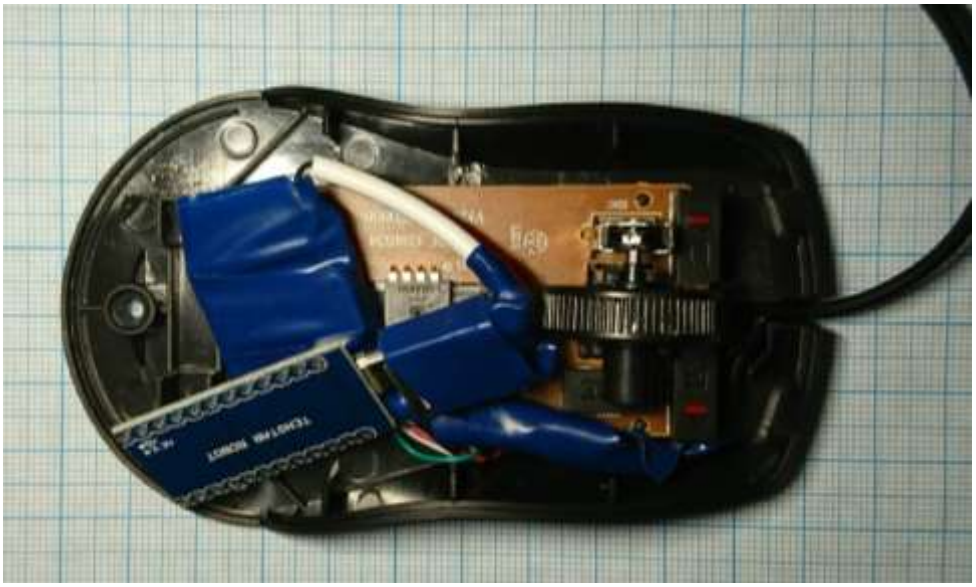

Рисунок 3.19 – Модифікований маніпулятор “миша”

Вага маніпулятора “миші” майже не змінилась, а всі нові комплектуючі вміщуються в корпусі. Після проведених маніпуляцій ми отримали повністю робочий маніпулятор з вбудованим мікроконтролером Arduino. Фактично ми отримали апаратну закладку, що зможе виконувати будь-які запрограмовані дії.

Небезпека подібних пристроїв у їх непомітності та “безумовній безпечності”. Маючи необхідні знання, навички та набір звичайних пристроїв (рисунок 3.20), ми створили модифікований маніпулятор “миша”, що здатний завдати серйозної шкоди будь-якій організації та разом з використання шелкоду, що прихований в зображенні формату BMP, обійти засоби захисту.

Рисунок 3.20 – Апаратні засоби, що використовувались

Замість Arduino Leonardo Micro можна використати ATtiny85 (рисунок 3.21), що має набагато менші розміри та вбудований USB інтерфейс.

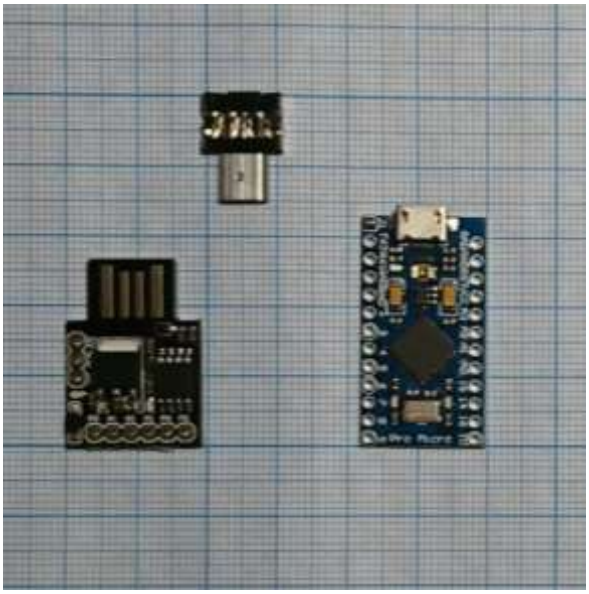

Рисунок 3.21 – ATtiny85 в порівнянні з Arduino Leonardo Micro

ATtiny85 має наступні технічні характеристики:

- Ширина шини даних 8-біт
- Тактова частота 20 мегагерц

- Кількість входів/виходів 6
- Обсяг пам'яті програм (flash пам’ять) 8 КБ
- Обсяг EEPROM 512 КБ
- Обсяг RAM 512 КБ
- Вбудований інтерфейс USB
- Напруга живлення 2.7 ... 5.5 В.

Варто зазначити, що атака з використанням зображення формату BMP може бути здійснена і без подібного обладнання, що робить її ще більше небезпечною.

4 ІСНУЮЧІ ПРИСТРОЇ ДЛЯ ПРОВЕДЕННЯ HID-АТАК

4.1 USB Rubber Ducky

USB Rubber Ducky (рисунок 4.1) - пристрій, що зовні нагадує звичайний флеш-накопичувач, який прикидається клавіатурою (яка являє собою HID-пристрій) і при підключенні до комп'ютера швидко набирає команди, що були записані в ньому. Це один з найпопулярніших пристроїв для проведення HID-атак від бренду Hak5. Його можна придбати в інтернеті у вільному доступі за 40$. Основні характеристики пристрою зазначені на рисунку 4.2.

Рисунок 4.1 – USB Rubber Ducky

Рисунок 4.2 – Основні характеристики USB Rubber Ducky

USB Rubber Ducky був винайдений засновником Hak5 Дарреном Кітченом під час роботи системним адміністратором. Пристрій розвивалося

через лінь, щоб не вводити одні й ті самі команд для виправлення помилок роботи принтерів і мережевих ресурсів.

З таким пристроєм людина може підійти до комп'ютера і підключити, здавалося б, звичайний USB флеш-накопичувач, який встановить бекдор, знайде та викраде необхідний документ, паролі або зробить будь-яку кількість завдань підчас проведення пентесту.

Всі ці речі можуть бути виконані за допомогою багатьох добре продуманих команд. Людина може зробити такі речі просто сидячи за комп’ютером та набираючи команди, якщо робити це без помилок, то знадобиться декілька хвилин.

USB Rubber Ducky робить це за лічені секунди. З точки зору засобів захисту та комп’ютерної системи HID-пристрої є повністю довіреними і в основному розглядаються як простий інтерфейс між користувачем і машиною, тому є довіреними.

З 2010 року USB Rubber Ducky є фаворитом серед хакерів, пентестерів та IT-фахівців. З його появою були винайдені HID-атаки, проте засоби захисту від них не розроблялись тривалий час. Крім того вони досі не є складовими комплексних засобів захисту, через це звичайні користувачі та великі компанії є досі вразливими для HID-атак.

Майже кожен обчислювальний пристрій має інтерфейс введення даних. HID, або Human Interface Device – тип комп’ютерного пристрою, який взаємодіє безпосередньо з людиною. При підключенні клавіатури або маніпулятора “миша” до комп'ютера, вони автоматично розпізнаються в системі як HID пристрої.

Розробники USB Rubber Ducky створили просту скриптову мову Ducky Script, за допомогою якої можна створювати сценарії атаки не володіючи іншими мовами програмування. В таблиці 4.1 наведені команди мови Ducky Script.

Таблиця 4.1 – Команди мови Ducky Script

Команда	Опис команди
REM	REM - коментар. Рядки, що починаються з REM, не обробляються.
DEFAULT_DELAY або DEFAULTDELAY	Використовуються для визначення часу (мілісекунд) для очікування між кожною наступною командою.
DELAY	Створює миттєву паузу в сценарії.
STRING	Друкує рядок. Може приймати один або кілька символів
WINDOWS або GUI	Емулює натиск клавіши Windows.
MENU або APP	Емулює клавішу програми, яку іноді називають клавішею меню або клавішею контекстного меню. У системах Windows це схоже на комбінацію клавіш SHIFT F10, створюючи меню, схоже на клацання правою кнопкою миші.
SHIFT	Емулює натиск клавіши SHIFT. Команда SHIFT може використовуватися під час навігації полів для вибору тексту, серед інших функцій.
ALT	Емулює натиск клавіши ALT.
CONTROL або CTRL	Емулює натиск клавіши CTRL.
DOWNARROW або DOWN LEFTARROW або LEFT RIGHTARROW або RIGHT UPARROW або UP	Емулює натиск клавіш зі стрілками.
BREAK або PAUSE	Емулює натиск клавіши CTRL BREAK.
CAPSLOCK	Емулює натиск клавіши CAPSLOCK.
DELETE	Емулює натиск клавіши DELETE .
END	Емулює натиск клавіши END.
ESC або ESCAPE	Емулює натиск клавіши ESC.
HOME	Емулює натиск клавіши HOME.
INSERT	Емулює натиск клавіши INSERT.
NUMLOCK	Емулює натиск клавіши NUMLOCK.
PAGEUP	Емулює натиск клавіши PAGEUP.
PAGEDOWN	Емулює натиск клавіши PAGEDOWN.
PRINTSCREEN	Емулює натиск клавіши PRINTSCREEN. Робить знімок екрану.
SCROLLLOCK	Емулює натиск клавіши SCROLLLOCK.
SPACE	Емулює натиск клавіши SPACE.
TAB	Емулює натиск клавіши TAB.

Приклад найпростішого скрипта, що відкриває командний рядок, викликає блокнот та друкує в ньому рядок наведений на рисунку 4.3.

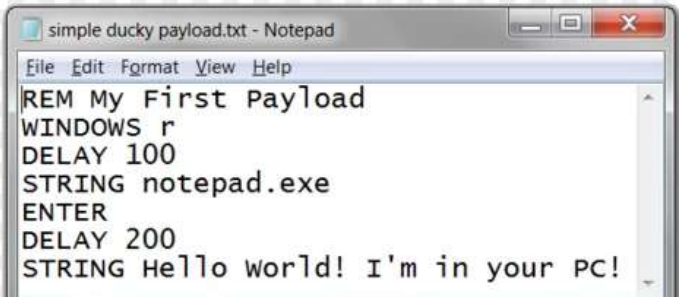

Рисунок 4.3 – Приклад найпростішого скрипта на мові Ducky Script

4.2 Bash bunny

Bash Bunny (рисунок 4.4) – це ще один пристрій від Hak5, проста і потужна багатофункціональна платформа для проведення USB-атак та автоматизації різноманітних завдань пентестерів і системних адміністраторів вартістю 99$. Bash Bunny підтримує просту скриптову мову Bunny Script, яка схожа на мову Ducky Script, має багатопозиційний перемикач для вибору сценарію атаки та централізоване сховище корисних навантажень.

Рисунок 4.4 – Bash Bunny

Пристрій має схоже з USB Rubber Ducky призначення, але також має додатковий функціонал. Bash Bunny можна використовувати для різних векторів атак, включаючи HID-атаки, USB Ethernet атаки. Завдяки цьому пристрій може виконувати декілька атак одночасно.

Bash Bunny може зберігати в пам’яті декілька сценаріїв атак, вибір між якими можна здійснити за допомогою перемикача. Багатокольоровий світлодіод

сигналізує, про завершення певних атаки. Завдяки 4-х ядерному процесору та швидкому флеш-накопичувачу пристрою необхідно 5-7 секунд для успішного виконання атаки.

Bash Bunny здатний емулювати комбінації довірених USB-пристроїв, таких як гігабітний Ethernet, послідовний порт, флеш-накопичувач і клавіатура.

Крім того, Bash Bunny - це повнофункціональний Linux-бокс з доступом до оболонки з виділеної послідовної консолі, це дозволяє використовувати будь-які інструменти для проведення тестування на проникнення.

4.3 USBHarpoon

Ідея створити звичайний USB-кабель, який експлуатує HID-атаки, виникла в дослідників досить давно. Адже кабель викличе у потенційної жертви менше підозр, ніж периферія. Однак успішних спроб до недавнього часу практично не було.

Для розробки USBHarpoon об'єдналися фахівці RFID Research Group, SYON Security і Кевін Митник, який і запропонував дослідникам цю ідею. Сам Митник надихнувся роботами дослідника, відомого в Twitter під ніком MG. Ще на початку 2018 року той демонстрував у своєму мікроблозі атаку через USB-кабель, проте зв'язатися з ним самим і обговорити співпрацю Митник, на жаль, не зміг.

В результаті був створений USBHarpoon. Об'єднаній групі дослідників вдалося вирішити всі проблеми і спроектувати працюючий USB-кабель, який також є HID-сумісним пристроєм.

USBHarpoon спрацює на будь-якому розблокованому комп’ютеру, до якого буде під’єднаний. Після підключення кабель виконає ряд команд, що були завчасно запрограмовані, завантажить і запустить пейлоад. У Windows це здійснюється безпосередньо через Run, а у випадку з Linux і macOS може знадобитися запуск терміналу. Зазвичай таку активність добре видно на екрані пристрою, однак якщо потрібно, атакуючий може приховати її.

Автор оригінального дослідження, Карстен Нол, нагадує, що проблема BadUSB, по суті, не була усунена і небезпечна донині.

4.4 O.MG Cable

Нова розробка MG - шкідливий кабель під назвою O.MG Cable (рисунок 4.5), був розроблений за підтримки ще кількох талановитих інженерів. MG вперше продемонстрував його на початку 2019 року. Такий кабель зовні не відрізняється від звичайного, і його можна підключити до комп'ютера під управлінням Linux, Mac або Windows.

Рисунок 4.5 – O.MG Cable

При підключенні O.MG Cable визначається системою як HID пристрій. Завдяки тому, що MG додав до своєї розробки підтримку бездротових з'єднань, після підключення O.MG Cable до цільової машині, зловмисник отримує можливість виконувати через Wi-Fi будь-які команди.

Після демонстрації прототипу в Twitter, розробник привіз O.MG Cable на конференцію з інформаційної безпеки DEFCON, де їх можна було придбати за ціною 200$.

O.MG Cable дозволяє створювати, зберігати і передавати нові корисні навантаження повністю віддалено. Кабель розроблений з урахуванням вимог Red Teams, включаючи такі функції, як додаткові завантажувальні навантаження та

можливість примусового видалення прошивки, що призводить до того, що кабель повністю повертається в нешкідливий стан.

В даний час (при підключенні до кабелю) атакуючий може перебувати на відстані 90 метрів від цілі, однак цю відстань можна суттєво збільшити налаштувавши кабель так, щоб він робив у якості клієнта в найближчої бездротової мережі. У такому випадку відстань атаки вже нічим не обмежена якщо ця бездротова мережа має вихід в інтернет.

Дослідник розповів, що набагато легше робити кабелі з нуля, ніж переробляти оригінальні кабелі Apple вручну. Проект O.MG Cable ось-ось переросте в повноцінне виробництво, так як з MG вже погодилася співпрацювати Hak5, і виробництво O.MG Cable планують поставити на потік, продаючи пристрої як легітимний інструмент для пентестерів та ІБ-фахівців.

Згідно блогу MG, вартість серійних O.MG Cable повинна скласти близько 100$.

4.5 Звичайний USB флеш-накопичувач

Насправді, операційна система нічого не знає про пристрій, який підключається до комп’ютера. Їй доводиться чекати, доки пристрій сам не повідомить, до якого класу обладнання він належить. Якщо взяти найпростіший приклад, коли ми під’єднуємо флеш-накопичувач за допомогою USB інтерфейсу, то він повідомляє операційній системі не тільки що є накопичувачем, а й свій обсяг.

Розглянемо алгоритм ініціалізації USB пристроїв.

Вони повідомляють USB-хосту кодами класів для завантаження необхідних драйверів. за допомогою кодів класів можна уніфікувати роботу з однотипними пристроями різних виробників. Кількість, які підтримує пристрій, визначається кількістю кінцевих точок (USB endpoints). Хост у момент підключення запитує у пристрою ряд стандартизованих відомостей (дескрипторів), Після цього на основі цієї інформації він приймає рішення, як

працювати з цим пристроєм. Дескриптори містять відомості про тип пристрою та виробника, на підставі яких хост підбирає програмний драйвер.

Звичайний флеш-накопичувач буде мати код класу 08h (Mass Storage Device - MSD), в той час як веб-камера, забезпечена мікрофоном, буде характеризуватися вже двома: 01h (Audio) і 0Eh (Video Device Class).

При підключенні USB-пристрою він реєструється, отримує адресу і відправляє свій дескриптор або дескриптори, щоб операційна система завантажила необхідні драйвера і відправила назад необхідну конфігурацію. Після цього починається безпосередня взаємодія з пристроєм. По завершенню роботи відбувається дерегістрація пристрою. Варто відзначити, що пристрої можуть мати кілька дескрипторів, а також можуть дерегіструватися і реєструватися в якості іншого пристрою.

Якщо розкрити корпус флеш-накопичувача (рисунок 4.6), то крім пам'яті (Mass Storage), яка видна користувачеві, на платі ще буде контролер, який відповідає за описані вище дії.

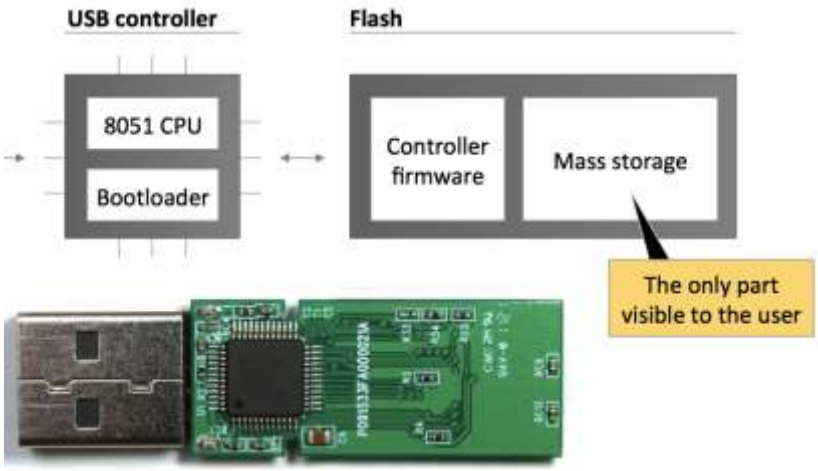

Рисунок 4.6 – USB флеш-накопичувач без корпусу

На конференції Black Hat ще у 2014 році двоє дослідників (Karsten Nohl і Jakob Lell) поділилися досвідом, як перепрограмувати контролер флеш-накопичувача своєю прошивкою. Після цього такий пристрій реєструвався в системі як клавіатура і набирав задані команди. Через серйозність проблеми дослідники не викладали код експлойта. Однак, через деякий час, двоє інших

дослідників (Adam Caudill і Brandon Wilson) вже на конференції Derbycon представили світу працездатний PoC, для мікроконтролерів Phison 2251-03.

Ця прошивка дозволяла перетворити звичайний флеш-накопичувач на пристрій для проведення HID-атак. Список відомих моделей USB флеш-накопичувачів, для яких підходить дана прошивка включає пристрої з чіпами 2303, 2307 та 2309:

- Kingston DataTraveler 3.0 T111 8GB (2303 Chip)
- Silicon power marvel M60 64GB (2303 Chip)
- Toshiba TransMemory-MX™ Black 16 GB (2303 Chip)
- Patriot Stellar 64 Gb Phison (2303 Chip)
- Silicon Power32G 2307 (2303 Chip)
- Kingston DataTraveler 100 G3 8GB (2307 Chip)
- Kingston DataTraveler 100G3 16GB (2307 Chip)
- Kingstone DataTraveler DTM30 16GB (2307 Chip)
- Silicon Power M01 32GB (2307 Chip)
- Silicon Power M01 8GB Y (2307 Chip)
- Silicon Power M01 8GB X 2309 (2307 Chip)
- Silicon Power BLAZE B30 32GB USB 3.1 (2309 Chip)

Існує велика кількість комерціалізованих рішень для проведення HID-атак. Одні з них мають вигляд звичайного флеш-накопичувача, інші виглядають як звичайні кабелі. Одні пристрої призначені лише для проведення HID-атак (можуть виконувати лише заздалегідь запрограмовані сценарії або мати підтримку бездротових мереж і можливість створення і виконання атак в реальному часі), інші можуть виконувати атаки різних векторів одночасно. Все це створює серйозну небезпеку як для користувачів, так і для великих компаній. Незважаючи на те, що такі атаки відомі відносно давно, а їх кількість збільшується з кожним роком, засоби захисту не можуть їх виявити та протидіяти їм.

5 МЕТОДИ ЗАХИСТУ ВІД РОЗГЛЯНУТИХ АТАК

5.1 Методи виявлення вірусів у файлах зображень формату BMP

Найпростіший варіант – це зміна налаштувань антивірусних засобів. Це дозволить перевіряти файли зображень на наявність вірусів, в тому числі і за допомогою “масок”. Однак варто розуміти, що такий метод має недоліки, так це призведе до детального аналізу всіх файлів зображень, що потрапляють в мережу. Оскільки подібні мережі можуть бути дуже великими, а обсяг даних значним це буде потребувати використання значних обчислювальних ресурсів системи. Що може призвести до серйозних втрат працездатності та відмові в обслуговуванні.

Тому в першу чергу для протидії описаній в роботі атаці, необхідно розроблювати нові засоби захисту та методи виявлення вірусів. Так доречніше буде проводити детальний аналіз файлу після перевірки та виявлення ключових ознак, що вказує на використання даної атаки. А саме:

- перевірка зарезервованих полів. Дані в цих полях для зображень формату BMP мають бути нульовими. Отже, відмінні значення в цих полях, будуть вказувати на те, що у файл був впроваджений вірус;

- зчитування розміру файлу з поля “Size” в заголовку файлу та перевірка цього значення зі справжнім розміром файлу. Відмінності у цих даних теж можуть свідчити про прихований шкідливий код;

- отримання з заголовку зображення даних, що вказують на горизонтальну та вертикальну кількість пікселів, після чого це значення необхідно порівняти зі справжньою кількістю пікселів. Відмінність цих даних вкаже на штучне зменшення зображення, а отже на наявність вбудованого вірусу;

- крім того можна використовувати методи, що дозволяють знаходити аномалії у зображеннях. Так, при наявності великої кількості спотворених

пікселів можна стверджувати про наявність в зображенні вірусу. Однак ці методи будуть обчислювально складнішими за попередні.

Крім того можна використати додаткові методи, що дозволять попередити подібні атаки:

- для унеможливлення розміщення подібних інфікованих зображень формату BMP на web-ресурсах якими володіють зловмисники або до яких вони мають доступ, необхідно чітко визначити перелік доступних інтернет ресурсів для користувачів використовуючи для цього “білий список”;

- фільтрувати та контролювати трафік організації. При можливості співробітникам потрібно обмежити можливість завантаження виконуваних файлів, бібліотеки динамічних посилань (DLL), скриптів, документів з невідомих джерел, що були створені в пакеті Microsoft Office, а також графічних файлів формату BMP.

5.2 Методи захисту від HID-атак

HID-атака є складовою розглянутої атаки, а отже для комплексного захисту необхідно розглянути методи протидії для неї:

- найпростіший варіант – це заборона на встановлення з’ємних пристроїв. Це досить легко зробити як для локальної машини, так і для робочих станцій в домені. Для цього необхідно використати групову політику безпеки. Однак, при цьому буде не доступний Plug'n'Play;

- використовувати список довірених пристроїв. Використання "білого списку" не є абсолютним захистом. Зловмисник може запрограмувати значення Vendor ID і Product ID, що будуть повністю відповідати вже зареєстрованим в системі. Саме за допомогою цих значень пристрій ідентифікується системою;

- найбільш радикальне рішення – заборона фізичного доступу до USB-портів. Проте таке рішення не є вдалим і може призвести до додаткових труднощів;

- найбільш вдалим та ефективним є використання евристичних методів для виявлення і блокування HID-атак. Наприклад, методи, що ґрунтуються на аналізі зміни швидкості введення.

Крім того для попередження подібних атак треба:

- робити перевірку на наявність закладних пристроїв придбаного обладнання та обладнання після ремонту. Та проводити періодичну перевірку існуючого обладнання;
- обмежити доступ до обладнання сторонніх осіб та працівників, які з ним не взаємодіють.

Враховуючи те, що розглянута атака включає в себе використання двох небезпечних векторів, що можуть використовуватися окремо, було прийняте рішення розробити окремі засоби захисту для виявлення вірусів у зображеннях формату BMP та протидії HID-атакам.

6 ОПИС РОЗРОБЛЕНИХ ПРОГРАМ, ДЛЯ ПРОТИДІЇ HID-АТАКАМ ТА ВИВЛЕННЯ ВІРУСІВ ВПРОВАДЖЕНИХ В ФАЙЛ ЗОБРАЖЕННЯ ФОРМАТУ BMP

6.1 Загальні відомості

В ході роботи були розроблені дві програми на мові Python. Перша виявляє впроваджений вірус у файли зображень формату BMP. Вона складається з одного модуля:

BMP_CHECK.py – основний модуль програми, що перевіряє зарезервовані поля зображення BMP, поле SIZE а також визначає справжню кількість пікселів та перевіряє зі значенням кількості пікселів по горизонталі та вертикалі, що зазначені в заголовку зображення.

Дана програма функціонує тільки на операційній системі Linux.

Програма написана на мові Python.

Друга програма виявляє та протидіє HID-атакам аналізуючи швидкість введення тексту та її зміну. Вона складається з одного модуля:

HID.py – основний модуль програми, протидіє HID-атакам аналізуючи швидкість введення тексту та її зміну. Після запуску програма функціонує як фоновий процес.

Дана програма функціонує тільки на операційній системі Windows.

Програма написана на мові Python.

6.2 Функціональне призначення

Програми призначені для демонстрації ефективності розроблених методів протидії розглянутим атакам. А саме, дозволяють виявити вірус у файлі зображень формату BMP та виявити та протидіяти HID-атакам.

Вхідними даними для програми “BMP_CHECK.py” є файл формату BMP.

Програма працює тільки з файлами зображень формату BMP.

Вхідними даними для програми “HID.py” є конфігураційний файл з налаштуваннями.

6.3 Опис логічної структури

Програма “BMP_CHECK.py” складається з одного модуля, що перевіряє зарезервовані поля зображення BMP, поле SIZE а також визначає справжню кількість пікселів та перевіряє зі значенням кількості пікселів по горизонталі та вертикалі, що зазначені в заголовку зображення.

Приклад функцій:

- функція, що перевіряє файл зображення:

def Check(vars):

де

vars – адреса файлу зображення;

- функція, що перевіряє доступність файлу зображення та відкриває його для зчитування:

def get_file_data(data):

де

data – адреса файлу зображення;

- функція, що зчитує файл зображення:

def parse_image(data):

де

data – адреса файлу зображення;

- функція, що кодує строку у шістнадцятковому форматі:

def toHex(var):

де

var – рядок, яку необхідно закодувати.

Програма “HID.py” складається з одного модуля, що функціонує як фоновий процес та протидіє HID-атакам аналізуючи швидкість введення тексту та її зміну.

Приклад функцій:

- функція, що спрацьовує при кожному натисканні клавіши та виявляє вторгнення:

def KeyStroke(event):

де

event – об’єкт події;

- функція, що виконує задану політику після виявлення вторгнення:

def caught(event):

де

event – об’єкт події;

- функція, що записує дані про атаку у лог файл:

def log(event):

де

event – об’єкт події.

6.4 Технічні засоби, що використовуються

Для проведення досліджень необхідно мати ПЕОМ під керуванням операційної системи Windows та ПЕОМ під керуванням операційної системи Linux та встановлений Metasploit. Бажано використовувати Parrot Security OS або Kali Linux. Це спеціальні дистрибутиви, призначений для тестування системи на проникнення, оцінки вразливостей і ліквідації їх наслідків та комп'ютерної криміналістики. Обидві ПЕОМ повинні бути підключені до однієї точки доступу WiFi.

Немає обмежень за обсягом пам’яті, специфіці додаткових пристроїв. Використання мікроконтролерів, що можна запрограмувати не є обов’язковим.

Наприклад, була використана наступна конфігурація:

Процесор: Intel(R)Celeron(R)CPU N3350@1.10GHz 1.10GHz

ОЗУ: 4,00Гб

Тип системи: 64-розрядна операційна система

6.5 Виклик та завантаження

Спочатку необхідно використати програму, що впроваджує вірус у файли зображень формату BMP. Вона була розроблена під час написання бакалаврської атестаційної роботи. Після чого інфіковане та/або оригінальне зображення подається на вхід програми “BMP_CHECK.py”, яка визначає наявність вірусу.

Програма “HID.py” запускається користувачем і функціонує у якості фонового процесу.

6.6 Вхідні дані

Вхідними даними для програми “BMP_CHECK.py” є зображення формату BMP.

Вхідними даними для програми “HID.py” є спеціальний конфігураційний файл (рис. 6.1).

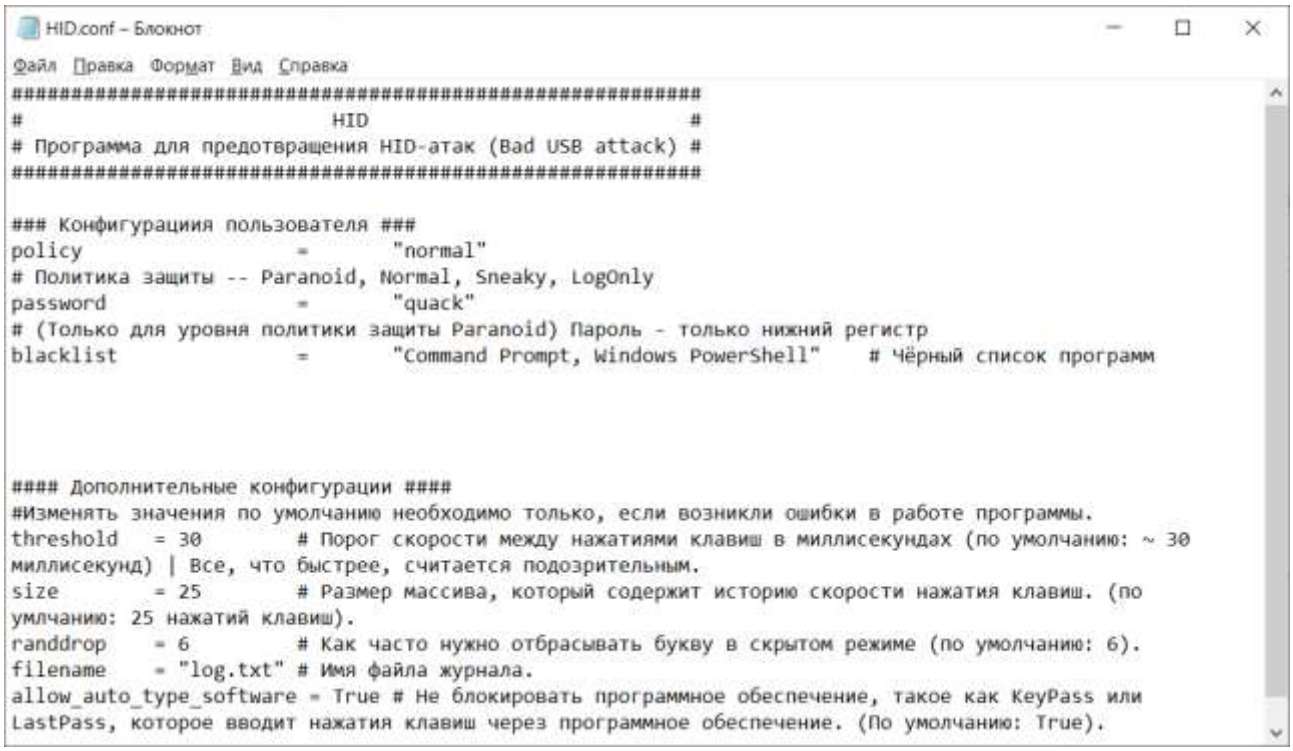

Рисунок 6.1 – Конфігураційний файл для програми HID.py

6.7 Вихідні дані

Вихідними даними для програми “BMP_CHECK.py” є результат перевірки суттєвих ознак наявності вірусу у зображеннях формату BMP.

Вихідними даними для програми “HID.py” є запис відомостей про перервану атаку у лог файл.

7 АНАЛІЗ РЕЗУЛЬТАТІВ ВИПРОБУВАНЬ

7.1 Підготовка вірусних зразків

В ході виконання атестаційної роботи були розроблені дві програми для виявлення та протидії розглянутій атаці. Перша програма “BMP_CHECK.py” для виявлення вірусів у зображеннях формату BMP, друга “HID.py” – для протидії HID-атакам. Особливості функціонування програм представлені в додатку А. Тестування програми “BMP_CHECK.py” будемо проводити на персональному комп’ютері під керуванням операційної системи Parrot Security OS. Спочатку необхідно створити інфікований файл. Metasploit – це інструмент, призначений для створення та використання експлойтів, а також виконувати експлуатацію та постексплуатацію вразливостей [12]. Для створення вхідних даних також можна скористатися утилітою Msfvenom, яка є складовою Metasploit. Для впровадження вірусу у зображення скористаємося програмою, що була розроблена під час бакалаврської атестаційної роботи. Вона впровадить в зображення вірус, змінить висоту зображення та запише в заголовок jump-функцію.

Скориставшись утилітою Msfvenom, згенеруємо шелл-код (рисунок 7.1).

```
┌──[root@parrot]─[~/SCAM]
└──╼ # msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.0.104 LPORT=4334 -f raw > shell
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 341 bytes
```

Рисунок 7.1 – Створення шелл-коду

Розберемо команду. Ключ -p відповідає за вибір корисного навантаження. В даному випадку структура означає: операційна система (windows), корисне навантаження (meterpreter), тип підключення (reverse_tcp). Підключення може бути двох типів: шелл-код може відкрити порт TCP, що був заданий заздалегідь, через який можна отримати доступ до командної оболонки, такий код має назву

прив’язаний до порту (port binding shellcode). Іншим варіантом є підключення до порту зловмисника, це дозволяє обійти брандмауер та NAT, такий варіант називається зворотною оболонкою (reverse shell shellcode). Параметр LPORT вказує номер порту до якого підключиться вірус, LHOST – IP адреса атакуючого, ключ -f вказує на тип формату файлу на виході. Raw означає, що на виході буде отримане необроблене корисне навантаження.

Після цього необхідно запустити програму, що впроваджує вірус у файл зображення формату BMP (рисунок 7.2). Ця програма була розроблнна під час написання бакалаврської атестаційної роботи.

Рисунок 7.2 – Запуск програми, що впроваджує вірус у файл зображення формату BMP

Після запуску відкриється головне меню програми. Спочатку необхідно обрати опцію “sc”, для того щоб обробити файл з корисним навантаженням та підготувати шелл-код. В якості вхідних даних необхідно задати шлях до згенерованого шелл-коду. Результатом роботи модулю є підготовлений шелл-код (рисунок 7.3).

Рисунок 7.3 – Результат роботи модулю, що відповідає за підготовку шелл-коду

Після цього необхідно повернутися у головне меню за допомогою команди “exit”, та ввести команду “gen”, для запуску наступного модуля. В цьому модулі необхідно задати шелл-код (результат попереднього модуля), при необхідності можна змінити параметри. Результатом роботи модулю буде інфіковане зображення (рисунок 7.4).

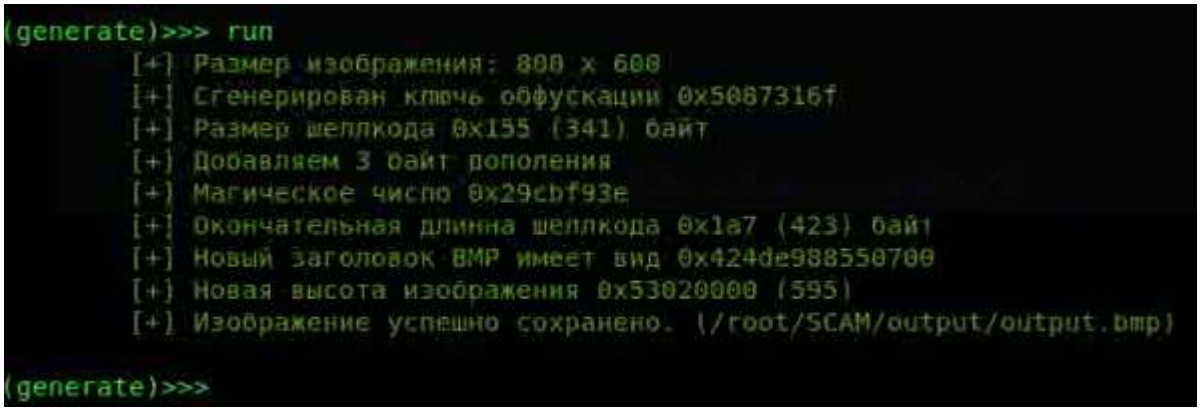

Рисунок 7.4 – Результат впровадження вірусу в зображення формату BMP

7.2 Аналіз виявлення вірусних зразків різними антивірусними засобами

Для перевірки файлів використаємо web-сайт www.virustotal.com. Він надає можливість аналізу файлу по базам даних усіх популярних антивірусних засобів та безпосереднього аналізу файлу антивірусами [8]. Спочатку перевіримо шелл-код у форматі raw, що був згенерований за допомогою msfvenom. Файл, що був завантажений у 2019 (рисунок 7.5) та той самий файл, що був завантажений у 2020 році (рисунок 7.6). Результати сканування вірусних зразків різними антивірусними засобами представлені в додатку Б.

Рисунок 7.5 – Результат сканування шелл-коду в форматі raw 2019 рік

Рисунок 7.6 – Результат сканування шелл-коду в форматі raw 2020 рік

29 з 60 антивірусних засобів визначили файл, як вірусний. Той самий вірусний код, але у вигляді виконуваного файлу, визначили вірусним вже 63 з 70 антивірусів (рисунок 7.8). Різна кількість антивірусних засобів, що перевіряла файл, зумовлена типом файлу. Усі антивірусні програми перевіряють різноманітні виконувані файли, але не всі перевіряють файли інших типів.

Рисунок 7.7 – Результати сканування шелл-коду в форматі exe 2019 рік

Рисунок 7.8 – Результати сканування шелл-коду в форматі exe 2020 рік

Впроваджувати вірус будемо в програму PuTTY. Це безкоштовний клієнт для протоколів Telnet, SSH та інших. Перевіримо оригінальний файл, що був завантажений у 2019 році (рисунок 7.9).

Рисунок 7.9 – Результати сканування оригінального PuTTY 2019 рік

Та той самий файл, що був завантажений у 2020 році (рисунок 7.10)

Рисунок 7.10 – Результати сканування оригінального PuTTY 2020 рік

Як видно з результатів, лише один антивірус зробив припущення про наявність вірусу (рисунок 7.11).

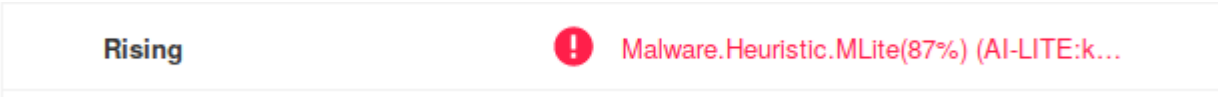

Рисунок 7.11 – Припущення про наявність вірусу

Будемо використовувати дві найпоширеніші методики для впровадження вірусу у виконуваний файл, що дозволяють зберегти його працездатність: створення нової секції та впровадження у code cave [8]. Code cave – це послідовність нульових байтів в пам'яті процесу. Для впровадження вірусу скористаємося утилітою Backdoor Factory. Це спеціальна утиліта для впровадження шелл-коду у виконувані файли та динамічні бібліотеки. Спочатку видалимо цифровий підпис файлу. Та перевіримо результати 2019 рік (рисунок 7.12) та 2020 рік (рисунок 7.13).

Рисунок 7.12 – Перевірка PuTTY з видаленим цифровим підписом 2019 рік

Рисунок 7.13 – Перевірка PuTTY з видаленим цифровим підписом 2020 рік

Створимо нову порожню секцію в оригінальному файлі та проведемо сканування 2019 рік (рисунок 7.14) 2020 рік (рисунок 7.15).

Була створена нова порожня секція, проте вірус туди не записався, оскільки замість нього був вказаний порожній файл.

Рисунок 7.14 – Перевірка PuTTY з новою порожньою секцією 2019 рік

Рисунок 7.15 – Перевірка PuTTY з новою порожньою секцією 2020 рік

Як видно з результатів, після створення нової порожньої секції без впровадження вірусного коду файл визначається 31 антивірусними засобами як вірус. Створимо нову секцію в оригінальному файлі та впровадимо в неї вірус. Перевіримо результати. 31 з 72 антивірусних засобів виявили вірус. Результати скагування 2019 рік (рисунок 7.16) 2020 рік (рисунок 7.17).

Рисунок 7.16 – Результати приховування вірусу в новій секції

Рисунок 7.17 – Результати приховування вірусу в новій секції

Приховаємо вірус у code cave (рисунок 7.18).

```
└─$ backdoor-factory -f PuTTY.exe -s user_supplied_shellcode_threaded -U sect
ion  -o PuTTY+cave.exe -Z
```

Рисунок 7.18 – Впровадження вірусу в code cave

За результатами сканування лише 18 антивірусів виявили вірус. Результати сканування 2019 рік (рисунок 7.19) 2020 рік (рисунок 7.20).

Рисунок 7.19 – Результати приховування вірусу в code cave 2019 рік

Рисунок 7.20 – Результати приховування вірусу в code cave 2020 рік

Перевіримо оригінальний файл BMP результати сканування 2019 рік (рисунок 7.21) 2020 рік (рисунок 7.22). Після цього впровадимо шелл-код за допомогою розробленої програми в зображення Результати сканування 2019 рік (рисунок 7.23) 2020 рік (рисунок 7.24).

Рисунок 7.21 – Оригінальне зображення 2019 рік

Рисунок 7.22 – Оригінальне зображення 2020 рік

Рисунок 7.23 – Інфіковане зображення 2019 рік

Рисунок 7.24 – Інфіковане зображення 2020 рік

Як видно з результату, жоден антивірусний засіб не зміг знайти вірус у зображенні. Що означає, що метод приховування вірусів від засобів захисту, в тому числі й антивірусів, розроблений в цій роботі є найбільш ефективним.

Якщо відкрити детальну інформацію про оригінальне (Додаток В) та інфіковане зображення (Додаток Г), можна побачити, що вони відрізняються лише значеннями висоти в пікселях, кількістю мега пікселів та значеннями хеш функцій вирахуваних від файлів.

Узагальнені результати проведених перевірок приведені у таблиці 7.1. Систематизований результат за 2019 рік у вигляді діаграми наведений на рисунку 7.25, за 2020 рік на рисунку 7.26. Знаком “-” позначається, якщо антивірус не виявив вірус, знаком “+” , якщо вірус був виявлений, знаком “*”, якщо антивірус не оброблює цей тип файлу.

Таблиця 7.1- Результати сканування зразків

Назва антивірусу	Шелл-код у форматі raw	Шелл-код у форматі exe	PuTTY	PuTTY без цифрового підпису	PuTTY з порожньою новою секцією	PuTTY з шелл-кодом в новій секції	PuTTY з шелл-кодом в code cave	Оригінальне BMP зображення	Інфіковане BMP зображення
Acronis	*	+	-	-	-	-	-	*	*
Ad-Aware	+	+	-	-	-	-	-	-	-
AegisLab	+	+	-	-	+	+	+	-	-
AhnLab-V3	+	+	-	-	-	-	-	-	-
Alibaba	*	+	-	-	-	-	-	*	*
ALYac	+	+	-	-	-	-	-	-	-
Antiy-AVL	-	-	-	-	-	-	-	-	-
Arcabit	+	+	-	-	-	-	-	-	-
Avast	+	+	-	-	+	+	-	-	-
Avast Mobile Security	-	-	-	-	-	-	-	-	-
AVG	+	+	-	-	+	+	-	-	-
Avira	-	+	-	-	+	+	-	-	-
Babable	-	-	-	-	-	-	-	-	-
Baidu	-	-	-	-	-	-	-	-	-
BitDefender	+	+	-	-	-	-	-	-	-
Bkav	-	+	-	+	+	+	+	-	-
CAT-QuickHeal	-	+	-	-	-	-	-	-	-
ClamAV	+	+	-	-	+	+	-	-	-
CMC	-	-	-	-	-	-	-	-	-
Comodo	-	+	-	-	+	+	-	-	-
Crowd Strike Falcon	*	+	-	-	+	+	+	*	*
Comodo	-	+	-	-	+	+	-	-	-
Cybereason	*	+	-	-	-	-	-	*	*
Cylance	*	+	-	-	+	+	+	*	*
Cyren	-	+	-	-	+	+	-	-	-
DrWeb	+	+	-	-	+	+	-	-	-
eGambit	*	+	-	-	+	+	+	*	*
Emsisoft	+	+	-	-	-	-	-	-	-

Продовження таблиці 7.1

Назва антивірусу	Шелл-код у форма-ті raw	Шелл-код у форма-ті exe	PuTTY	PuTTY без циф-рового підпису	PuTTY з поро-жньою новою сек-цією	PuTTY з шелл-кодом в новій секції	PuTTY з шелл-кодом в code cave	Оригіна-льне BMP зобра-ження	Інфіко-ване BMP зобра-ження
Endgame	*	+	-	-	+	+	-	*	*
eScan	+	+	-	-	-	-	-	-	-
ESET-NOD32	-	+	-	-	+	+	-	-	-
F-Prot	-	+	-	-	-	+	-	-	-
F-Secure	-	+	-	-	+	+	-	-	-
FireEye	+	+	-	-	+	+	-	-	-
Fortinet	-	+	-	-	+	+	-	-	-
Gdata	+	+	-	-	-	-	-	-	-
Ikarus	-	+	-	-	-	+	+	-	-
Jiangmin	-	-	-	-	-	-	-	-	-
K7AntiVirus	-	+	-	-	-	-	-	-	-
K7GW	-	+	-	-	-	-	-	-	-
Kaspersky	+	+	-	-	+	+	-	-	-
Kingsoft	-	-	-	-	-	-	-	-	-
Malware-Bytes	-	-	-	-	-	-	-	-	-
MAX	+	+	-	-	-	-	-	-	-
MaxSecure	*	-	-	-	-	-	-	*	*
McAfee	-	+	-	-	+	+	-	-	-
McAfee-GW-Edition	-	+	-	+	+	+	-	-	-
Microsoft	+	+	-	-	+	+	+	-	-
Gdata	+	+	-	-	-	-	-	-	-
Ikarus	-	+	-	-	-	+	+	-	-
Jiangmin	-	-	-	-	-	-	-	-	-
K7AntiVirus	-	+	-	-	-	-	-	-	-
K7GW	-	+	-	-	-	-	-	-	-
Kaspersky	+	+	-	-	+	+	-	-	-
Kingsoft	-	-	-	-	-	-	-	-	-
Malware-Bytes	-	-	-	-	-	-	-	-	-
MAX	+	+	-	-	-	-	-	-	-
MaxSecure	*	-	-	-	-	-	-	*	*
McAfee	-	+	-	-	+	+	-	-	-
McAfee-GW-Edition	-	+	-	+	+	+	-	-	-
Microsoft	+	+	-	-	+	+	+	-	-
NANO-Antivirus	+	+	-	-	+	+	-	-	-

Продовження таблиці 7.1

Назва антивірусу	Шелл-код у форма-ті raw	Шелл-код у форма-ті exe	PuTTY	PuTTY без циф-рового підпису	PuTTY з поро-жньою новою сек-цією	PuTTY з шелл-кодом в новій секції	PuTTY з шелл-кодом в code cave	Оригіна-льне BMP зобра-ження	Інфіко-ване BMP зобра-ження
Palo Alto Networks	*	-	-	-	-	-	-	*	*
Panda	-	-	-	-	-	-	-	-	-
Symantec	+	-	-	-	-	-	-	*	*
Symantec Mobile Insight	*	-	-	-	-	-	-	*	*
Qihoo-360	-	+	-	-	+	+	-	-	-
Rising	-	+	+	+	+	+	+	-	-
Sentinel One	*	+	-	-	+	+	-	*	*
Sophos AV	-	+	-	-	+	+	-	-	-
Sophos ML	*	+	-	-	-	-	-	*	*
SUPER Anti Spyware	-	+	-	-	-	-	-	-	-
Symantec	*	+	-	-	+	+	+	-	-
TAC-HYON	-	-	-	-	-	-	-	-	-
Tencent	+	-	-	-	-	-	-	-	-
The Hacker	-	-	-	-	-	-	-	-	-
Total Defense	-	-	-	-	-	-	-	*	*
Trapmine	*	+	-	-	-	-	-	*	*
Trend Micro	-	+	-	-	-	-	-	-	-
Trend Micro-House Call	-	+	-	-	-	-	-	-	-
Trustlook	*	-	-	-	-	-	-	*	*
VBA32	-	-	-	-	-	-	-	-	-
VIPRE	-	-	-	-	+	-	-	-	-
ViRobot	-	+	-	-	-	-	-	-	-
Webroot	*	+	-	-	-	-	-	*	*
Yandex	-	+	-	-	+	+	-	-	-
Zillya	-	-	-	-	-	+	-	-	-
ZoneAlarm	+	+	-	-	+	+	-	-	-
Zoner	-	-	-	-	-	-	-	-	-

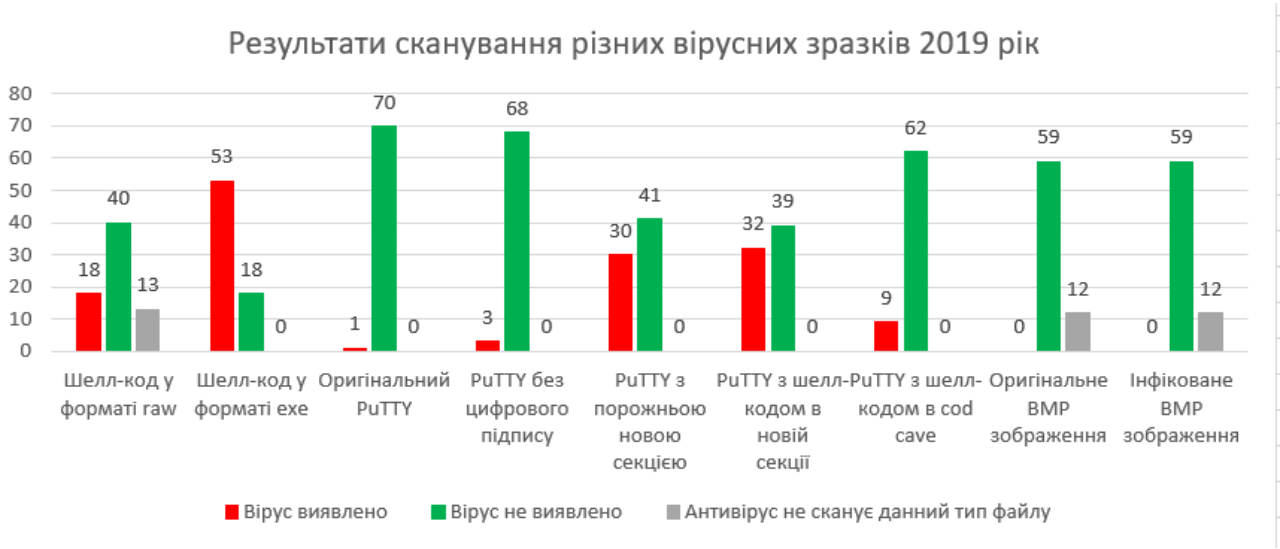

Рисунок 7.25 – Систематизовані результати сканування вірусних зразків 2019 рік

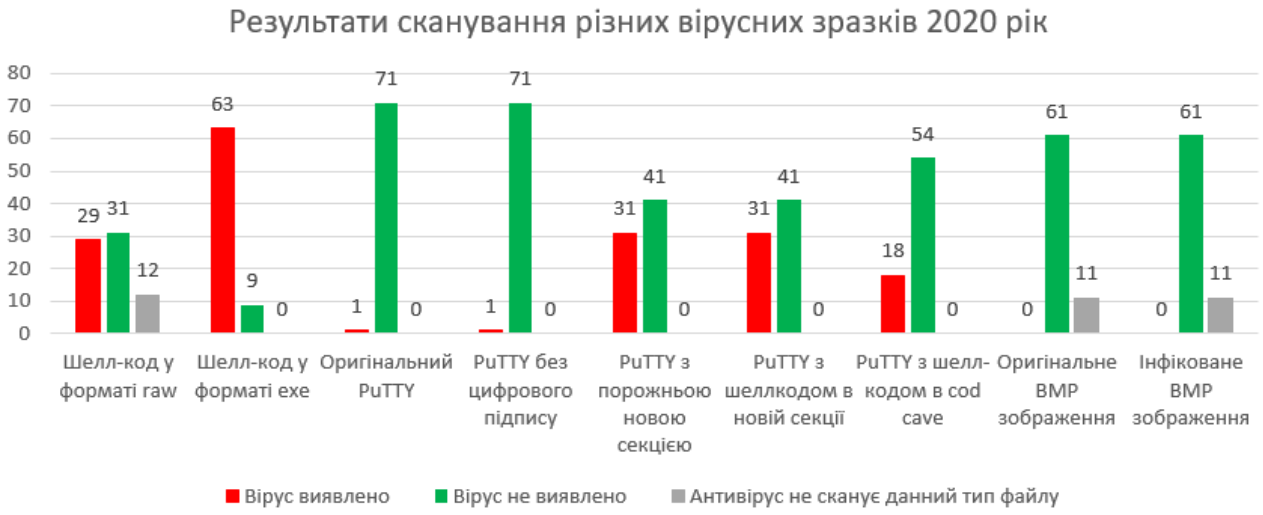

Рисунок 7.26 – Систематизовані результати сканування вірусних зразків 2020 рік

7.3 Демонстрація роботи програми “BMP_CHECK.py”

В ході дослідження була розроблена програма на мові Python, що перевіряє зарезервовані поля зображення BMP, поле SIZE а також визначає справжню кількість пікселів та перевіряє зі значенням кількості пікселів по горизонталі та вертикалі, що зазначені в заголовку зображення. Вона приймає в якості вхідних

даних зображення формату BMP та робить висновок щодо наявності суттєвих ознак, що властиві інфікованим зображенням [13].

При перевірці оригінального зображення не було виявлено жодних аномалій (рисунок 7.27).

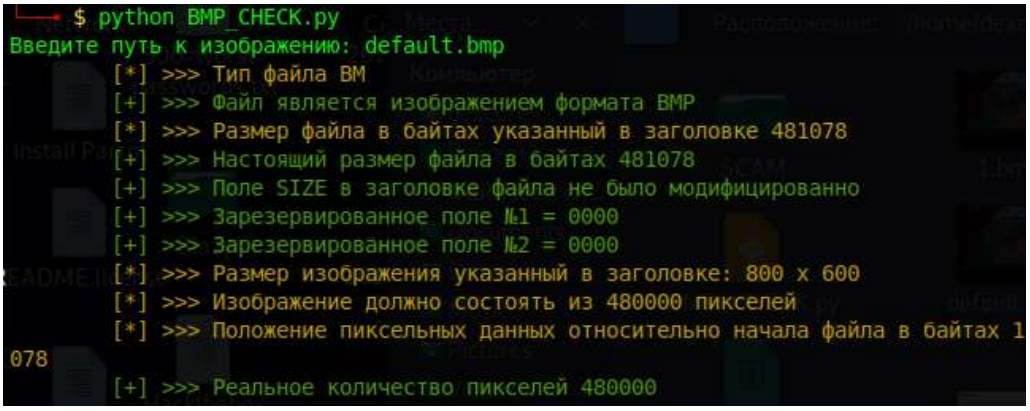

Рисунок 7.27 – Перевірка оригінального зображення

При перевірці інфікованого зображення були виявлені аномалії за ключовими ознаками (рисунок 7.28).

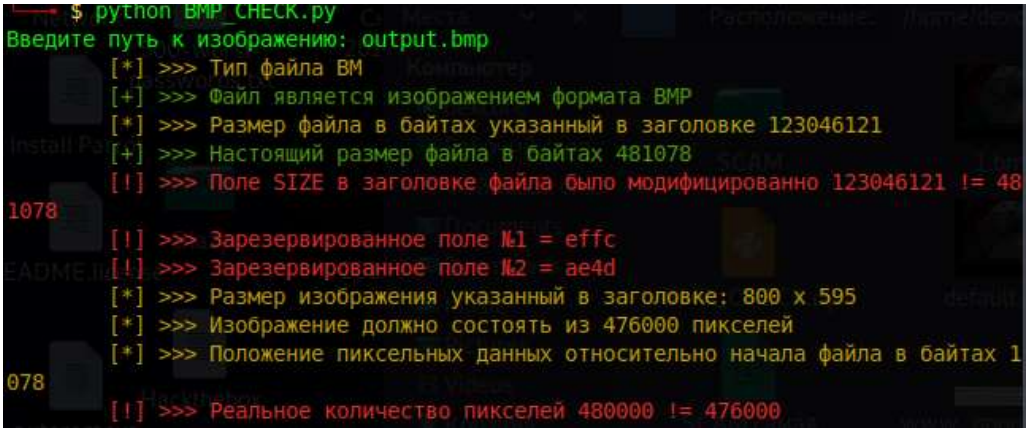

Рисунок 7.28 – Перевірка інфікованого зображення

7.4 Демонстрація роботи програми “HID.py”

В ході дослідження також була розроблена програма на мові Python, що аналізує швидкість введення тексту та її зміну, за допомогою цього виявляє та протидіє HID-атакам. При різкому суттєвому збільшенні швидкості, що перевищує людські можливості, програма виявить атаку [13]. Програма має 4 режими:

- реєстрація атаки в журналі при виявленні без протидії;

- переривання кількох натискань клавіш після виявлення атаки. Цього буде досить, щоб перервати будь-яку атаку. Крім того відповідні записи будуть занесені в журнал;

- тимчасове відключення введеня з клавіатури при виявленні атаки. Введення знову буде дозволено після закінчення атаки. Атака також буде зареєстрована в журналі;

- блокування подальшого введення до тих пір, поки не буде введений правильний пароль, який можна задати в конфігураційному файлі файлі (.conf) при виявленні атаки. Атака також буде зареєстрована в журналі.

Програма не має графічного інтерфейсу. Після запуску програма функціонує як фоновий процес. Оскільки програма була розроблена для операційних систем сімейства Windows і не має особливих вимог для функціонування, то вона буде функціонувати навіть на старих версіяї операційних систем. Для підтвердження цього тестування програми “HID.py” будемо проводити на персональному комп’ютері під керуванням операційної системи Windows XP. Атака показана на рисунку 7.29. Варто зазначити, що до цієї атаки вразливі всі версії операційних системи сімейства Windows, Linux. MacOS також вразлива до цих атак.

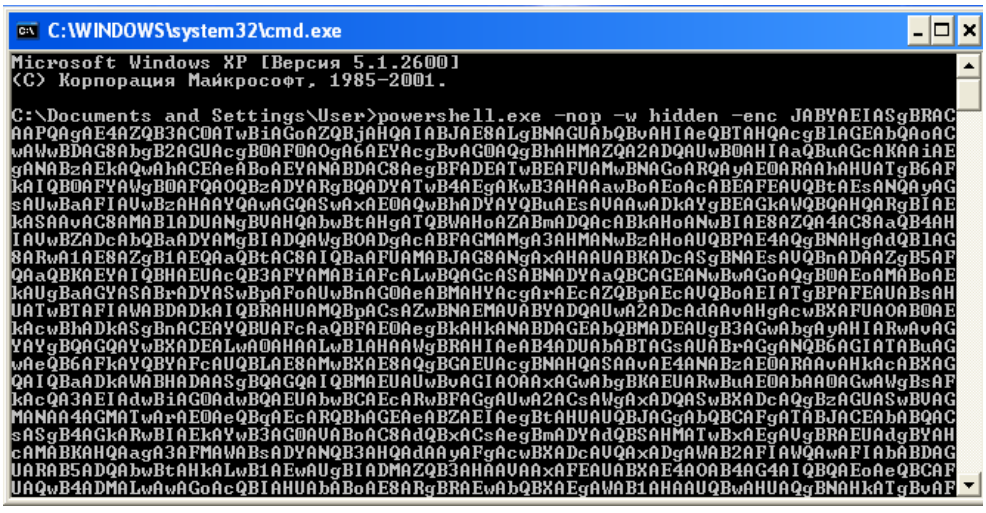

Рисунок 7.29 - Успішна атака на незахищену систему.

Після запуску програми, HID-атакам була виявлена та знешкоджена.

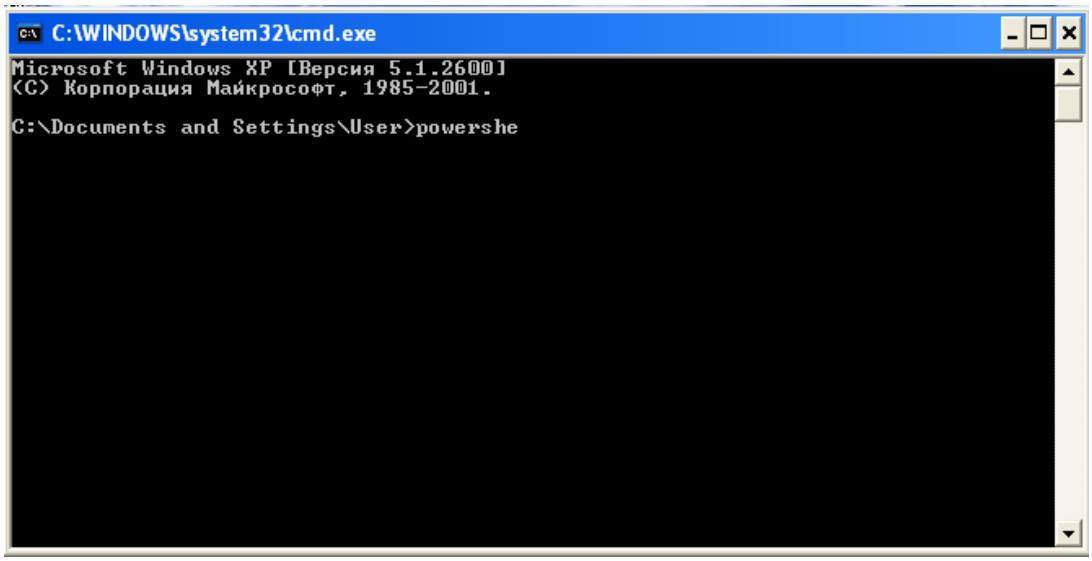

Рисунок 7.30 — Виявлена та знешкоджена атака за допомогою розробленої програми

Як видно з результатів, розроблені методи змогли успішно протидіяти розглянутій атаці. Вони показали суттєво більшу ефективність, ніж сучасні засоби захисту.

ВИСНОВКИ

У ході виконання атестаційної роботи були розроблені дві програми для виявлення та протидії розглянутій атаці. Перша програма для виявлення вірусів у зображеннях формату BMP, друга – для протидії HID-атакам.

Проаналізувавши отримані результати, можна стверджувати, що дані методи протидії виявилися набагато ефективнішими за сучасні засоби захисту. Це обумовлено тим, що більшість із засобів захисту просто не звертають уваги на зображення або інший безпечний тип файлу. Оскільки вважають, що немає причин витрачати процесорний цикл на аналіз зображення. Антивірусні засоби захисту під час сканування не виявили вірус у файлі формату BMP. Крім того аналіз поведінки зображення в “пісочниці” також не виявить вірус, оскільки при відкритті зображення шкідливий код не почне виконуватися. Враховуючи вище сказане, IDS, IPS також не зможуть виявити вірус. Це свідчить про неефективність виявлення та протидії існуючих засобів захисту розглянутій атаці. Крім того HID пристрої сприймаються, як звичайний інтерфейс взаємодії користувача та персонального комп’ютера, тому сучасні засоби захисту не можуть виявити та протидіяти HID-атакам.

За результатами випробувань можна стверджувати, що розроблені методи протидії розглянутій атаці є найбільш ефективними. Результати даної роботи можна використовувати під час розробки засобів антивірусного захисту та комплексних засобів захисту ІТС та для їх модернізації з метою попередження подібних атак.

Результати роботи були оприлюднені на 6 міжнародних конференціях. Крім того результати роботи та висновки щодо ефективності розроблених методів для протидії даним атакам були оприлюднені в науковому журналі, який потім був процитований в SCOPUS.

ПЕРЕЛІК ПОСИЛАНЬ

1. Безруков М.М. Класифікація комп'ютерних вірусів та методи захисту від них. – Москва, СП "ІСЕ", 2009 р. – 148 с.

2 Ф.Файтс, П.Джонстон, М.Кратц. Комп'ютерний вірус: проблеми і прогноз. – Москва, "Мир ", 2013 р. – 152 с.

3 Мостовий Д.Ю. Сучасні технології боротьби з вірусами. – Москва, 2012 р. – 146 с.

4 Моісеєнков І. Безпека комп'ютерних систем. – "Эксмо-Пресс", 2015 р. – 156 с.

5 А.В.Міхайлов Комп'ютерні віруси і боротьба з ними. – Москва, "Мир ", 2012 р. – 153 с.

6 Гриньов Р.С., Сєвєрінов О.В. Шкідливий USB HID-емулятор. Радіоелектроніка та молодь у XXI столітті: між. форум. Харків, 2018. с. 120-121.

7 Гриньов Р.С., Сєвєрінов О.В. Апаратний закладний пристрій з підтримкою Wi-FI. Радіоелектроніка та молодь у XXI столітті: каталог виставки між. форум. Харків, 2018. с. 32.

8 Гриньов Р.С., Сєвєрінов О.В. Аналіз небезпеки апаратних закладних пристроїв. Радіоелектроніка та молодь у XXI столітті: між. форум. Харків, 2019. с. 93-94.

9 Гриньов Р.С., Сєвєрінов О.В. Аналіз тенденцій вірусних загроз в Україні. Сучасні напрямки розвитку інформаційно-комунікаційних технологій та засобів управління: між. конф. Харків, 2019. с. 100.

10 Гриньов Р.С., Севєрінов О.В. Аналіз небезпеки впровадження вірусного програмного забезпечення в зображення. Комп’ютерні та інформаційні системи і технології: між. науково-технічна конф. Харків, 2019. с. 75.

11 Гриньов Р.С., Севєрінов О.В. Аналіз статистики та особливостей розповсюдження вірусів в Україні. Сучасні напрямки розвитку інформаційно-комунікаційних технологій та засобів управління: між. конф. Харків, 2019. с. 100.

12 Офіційна документація Parrot Security OS. URL: https://docs.parrotsec.org/doku.php (дата звернення: 22.10.2018).

13 Grynov Rostyslav, Vitalii Martovytskyi, Oleksandr Sievierinov, Vladislav Sukhoteplyj, Olha Soloviova, Yelizaveta Kortyak. A Method for Identifying and Countering HID Attacks - Virus Detection in BMP Images. Volume 8. No. 7, July 2020 International Journal of Emerging Trends in Engineering Research