Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ДОПУСТИТИ ДО ЗАХИСТУ:
В. о. завідувач кафедри
кібербезпеки та захисту
інформації
_____ Іван ПАРХОМЕНКО
«__» червня 2025 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА

кваліфікаційної роботи

галузь знань 12 Інформаційні технології
(шифр і назва галузі знань)

спеціальність 125 «Кібербезпека»
(код і назва спеціальності)

освітній ступень бакалавр

освітня програма Кібербезпека
(назва освітньо-професійної програми)

на тему: «Автоматизована система кіберрозвідки об’єкта інформаційної діяльності»

Виконавець: студент IV курсу, групи КБ-42

_____ **Олександр ГОРБАТЮК**
(підпис) (ім’я, прізвище)

	Підпис	**Ім’я ПРІЗВИЩЕ**
Керівник		Яніна ШЕСТАК
Нормоконтроль		Іван БІЛОКОНЬ

Київ 2025

Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ЗАТВЕРДЖЕНО:
В. о. завідувача кафедри
кібербезпеки
та захисту інформації
_____ Іван ПАРХОМЕНКО
«29» листопада 2024 р.

ЗАВДАННЯ
на виконання кваліфікаційної роботи

спеціальності 125 Кібербезпека
(код і назва спеціальності)

освітньої програми Кібербезпека
(назва освітньої-професійної програми)

Студенту **КБ-42** (група) **Горбатюку Олександру Юрійовичу** (прізвище ім'я по-батькові)

Тема кваліфікаційної роботи Автоматизована система кіберрозвідки об'єкта інформаційної діяльності

1. ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Тема кваліфікаційної роботи затверджена на засіданні кафедри кібербезпеки та захисту інформації протокол №6 від 28.11.2024 р.

2. ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Методи пасивної та активної кіберрозвідки, структура доменної інфраструктури, набір інструментів інформаційного збору, основи Bash-програмування, принципи формування автоматизованих звітів.

3. ЗМІСТ РОЗРАХУНКОВО-ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

Аналіз актуальних кіберзагроз для організаційної інфраструктури, огляд технік та інструментів кіберрозвідки, проєктування архітектури автоматизованої системи збору інформації, розробка Bash-скрипта для виявлення активів.

4. ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Практична цінність Інструмент автоматизованої кіберрозвідки дозволяє швидко й ефективно виявляти вразливості інформаційної інфраструктури

5. ДАТА ВИДАЧІ ЗАВДАННЯ

Дата видачі завдання: 29 листопада 2024 року

Завдання видала _____ (підпис) Яніна ШЕСТАК (ім'я, прізвище)

Завдання прийняв до виконання _____ (підпис) Олександр ГОРБАТЮК (ім'я, прізвище)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Найменування етапів робіт	Строки виконання робіт (початок-кінець)	Відмітка про виконання
1	Уточнення постановки задачі	29.11.2024 – 22.01.2025	*виконано*
2	Аналіз літератури	29.01.2025 – 11.02.2025	*виконано*
3	Аналіз кіберзагроз, методів рекону та правових аспектів	12.02.2025 – 15.02.2025	*виконано*
4	Огляд інструментів кіберрозвідки та методів збору інформації	16.02.2025 – 04.03.2025	*виконано*
5	Розробка Bash-скрипта для автоматизованої кіберрозвідки	05.03.2025 – 21.03.2025	*виконано*
6	Тестування скрипта, збір результатів, генерування HTML-звіту	22.03.2025 – 08.04.2025	*виконано*
7	Аналіз скрипта, функції, обґрунтування вибору утиліт	09.04.2025 – 10.05.2025	*виконано*
8	Оформлення пояснювальної записки	11.05.2025 – 27.05.2025	*виконано*
9	Підготовка до захисту кваліфікаційної роботи	28.05.2025 – 13.06.2025	*виконано*

Завдання видала _____ (підпис) Яніна ШЕСТАК (ім'я, прізвище)

Завдання прийняв до виконання _____ (підпис) Олександр ГОРБАТЮК (ім'я, прізвище)

Термін подання кваліфікаційної роботи до ЕК 13 червня 2025 року

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи складається зі вступу, трьох розділів, загальних висновків, списку використаних джерел, додатків, має 79 сторінок основного тексту, 2 таблиці та 3 рисунка. Список використаних джерел містить 30 найменувань і займає 4 сторінки.

Метою роботи є теоретичне обґрунтування та практична реалізація засобу автоматизованого збору технічної інформації про IT-інфраструктуру підприємства з метою виявлення потенційних векторів атак.

Для досягнення зазначеної мети поставлено наступні завдання:

- Дослідити методи активної та пасивної кіберрозвідки в контексті захисту цифрової інфраструктури
- Проаналізувати сучасні утиліти для збору технічної інформації та виявлення поверхні атаки веб-ресурсів і мережевих сервісів
- Реалізувати Bash-скрипт для автоматизації процесу кіберрозвідки з врахуванням активних і пасивних методів збору даних
- Протестувати та верифікувати ефективність розробленого інструмента на прикладі типового IT-середовища підприємства

Об'єктом дослідження є процес кіберрозвідки в контексті захисту цифрової інфраструктури організації.

Предметом дослідження є методи, інструменти та підходи до активної й пасивної кіберрозвідки, зокрема з використанням утиліт для аналізу поверхні атаки та виявлення вразливостей.

Практичною цінністю отриманих результатів є програмна реалізація автоматизованого засобу збору відкритої інформації про IT-інфраструктуру підприємства з метою раннього виявлення потенційних векторів атак

Ключові слова: кіберрозвідка, інформаційна безпека, рекон, поверхня атаки, автоматизація, інфраструктура, рекомендації.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

API	–	Application Programming Interface
AXFR	–	Authoritative Zone Transfer
CDN	–	Content Delivery Network
CLI	–	Command-Line Interface
CMS	–	Content Management System
CVE	–	Common Vulnerabilities and Exposures
DDoS	–	Distributed Denial of Service
DLP	–	Data Loss Prevention
DNS	–	Domain Name System
DoS	–	Denial of Service
HTTP	–	HyperText Transfer Protocol
IDS	–	Intrusion Detection System
IOT	–	Internet of Things
IP	–	Internet Protocol
NSE	–	Nmap Scripting Engine
OSINT	–	Open Source Intelligence
ОС	–	Операційна Система
ПЗ	–	Програмне Забезпечення
SPF	–	Sender Policy Framework
SSL	–	Secure Sockets Layer
SYN	–	Synchronize
TCP	–	Transmission Control Protocol
TLS	–	Transport Layer Security
UDP	–	User Datagram Protocol
URI	–	Uniform Resource Identifier
URL	–	Uniform Resource Locator
VPN	–	Virtual Private Network
WAF	–	Web Application Firewall

ВСТУП

Інформаційна безпека є невіддільною частиною стабільної роботи підприємств, адже сучасні інформаційні системи все частіше потрапляють у поле зору зловмисників. Щоб не стати легкою мішенню, організаціям потрібно не лише реагувати на атаки, а й завчасно виявляти слабкі місця в інфраструктурі. Один із ключових напрямів такої превентивної діяльності — кіберрозвідка (Reconnaissance), що передбачає систематичний збір технічної й структурної інформації про ресурси компанії з метою виявлення потенційних векторів компрометації.

Кіберрозвідка, що застосовується як на етапі підготовки атак, так і для захисних цілей (Red/Blue teaming, Threat Hunting), допомагає виявляти слабкі місця у веб-додатках, мережевих сервісах і доменних зв’язках. Автоматизовані засоби кіберрозвідки підвищують ефективність і швидкість аналізу, зменшують вплив людського чинника та сприяють проактивному захисту.

Кваліфікаційна робота присвячена аналізу методів кіберрозвідки та розробці Bash-скрипту для автоматизованого збору інформації про інфраструктуру організації. Розглянуто пасивні й активні техніки, виконано порівняльний аналіз інструментів (Subfinder, Katana, WhatWeb, Nmap, Dirsearch), реалізовано власне рішення для автоматизації збору, обробки та візуалізації даних.

Актуальність роботи зумовлена тим, що кіберзлочинці дедалі частіше використовують інструменти Recon для підготовки атак, тому підприємства повинні володіти такими ж чи кращими засобами, щоб передбачити та запобігти загрозам. Створення ефективної, гнучкої та придатної для модифікацій системи кіберрозвідки дозволяє покращити стратегії захисту, вчасно виявляти витоки інформації та скорочувати площу атаки.

Метою цієї роботи є створення автоматизованого інструменту для кіберрозвідки організаційної інфраструктури, що забезпечить можливість своєчасного виявлення вразливих компонентів системи та підвищить обізнаність організації щодо її зовнішньої видимості в інтернеті.

Для досягнення поставленої мети були сформульовані наступні завдання:

- дослідити основи інформаційної безпеки в контексті кіберрозвідки;
- охарактеризувати інфраструктуру організації як об’єкт Recon-аналізу;
- розглянути активні та пасивні методи розвідки;
- провести аналіз існуючих інструментів автоматизації Recon-етапу;
- реалізувати Bash-скрипт для збору та обробки інформації;
- протестувати інструмент на етичних платформах та оцінити ефективність;
- надати рекомендації щодо зменшення експозиції та захисту інфраструктури;

Структура роботи складається з трьох основних розділів.

У першому розділі розглянуто теоретичні основи кіберрозвідки, поняття інформаційної безпеки, типи Recon-методів (активних і пасивних), а також проаналізовано інфраструктуру організації як об'єкт кіберрозвідки.

Другий розділ присвячено практичним інструментам і технікам: тут здійснено аналіз та порівняння інструментів збору інформації, описано конкретні методи брутфорсу, сканування портів і фазингу директорій, а також розглянуто приклади атак, де Recon був вирішальним етапом.

У третьому розділі представлено реалізацію автоматизованої системи кіберрозвідки у вигляді Bash-скрипту, проведено його тестування, аналіз результатів, обґрунтовано вибір інструментів, а також надано рекомендації щодо підвищення рівня кіберстійкості підприємства.

У завершальній частині роботи підбито підсумки проведеного дослідження та сформульовано практичні поради щодо вдосконалення

кіберзахисту організацій шляхом активного моніторингу та автоматизованої розвідки.

Результатом виконаної дипломної роботи стало формування цілісного підходу до аналізу моделі кіберзагроз через інструменти автоматизованої кіберрозвідки. Було розглянуто й апробовано на практиці комплекс методів кіберрозвідки, які дозволяють ідентифікувати активи цільової інформаційної інфраструктури, потенційні вектори атак, публічно доступні сервіси та приховані вразливості. Результати проведеного дослідження можуть бути корисними для спеціалістів, які займаються аудитом безпеки, тестуванням на проникнення та побудовою засобів активного виявлення загроз на ранніх етапах.

РОЗДІЛ 1
КІБЕРЗАГРОЗИ ТА ІНФОРМАЦІЙНІ АКТИВИ ОРГАНІЗАЦІЇ ЯК ОБ'ЄКТ ДОСЛІДЖЕННЯ

1.1 Основи інформаційної безпеки в організаційному середовищі

Інформаційна безпека в умовах корпоративного середовища розглядається як сукупність взаємопов’язаних технічних, організаційних і процедурних заходів, спрямованих на гарантування цілісності, конфіденційності та доступності інформаційних ресурсів підприємства. Оскільки сучасні організації функціонують у рамках складної ІТ-інфраструктури, що об’єднує сервери з різними операційними системами, локальні мережі, термінали, мобільні пристрої та хмарні сервіси, необхідно формувати систему управління ризиками, інтегровану в архітектуру цієї інфраструктури та адаптовану до її динамічних змін. Особливу увагу слід приділити регламентованим політикам обробки інформації й стандартизованим процедурам контролю доступу, оскільки саме вони виступають основними компонентами, що забезпечують належну структурованість та повторюваність заходів із захисту. Водночас механізми моніторингу інцидентів, що виявляють ознаки несанкціонованого втручання, а також регулярне підвищення рівня кібергігієни персоналу, що, своєю чергою, впливає на ефективність реагування на потенційні загрози, мають бути безперервно підтримувані в актуальному стані.

Натомість гетерогенність сучасного ІТ-пейзажу призводить до того, що в експлуатації нерідко залишаються морально застарілі компоненти — сервери, пристрої або програмне забезпечення, оновлення яких або виведення з експлуатації може спричиняти ризики порушення стабільності бізнес-процесів. Ураховуючи це, традиційні статичні методи захисту виявляються малоефективними, оскільки не здатні своєчасно адаптуватися до появи нових

векторів атак, що формуються під впливом як зовнішніх чинників (зміни регуляторних вимог чи зростання складності загроз), так і внутрішніх (реорганізація IT-структури чи впровадження нових технологій). У результаті необхідно впроваджувати багаторівневу модель захисту, яка в контексті організації охоплює IDS, багатофакторної аутентифікації, сегментації мережевих середовищ і забезпечення процедур аудиту доступу до критичних інформаційних об’єктів.

Розширення цифрової взаємодії з партнерами, клієнтами та постачальниками призводить до активного використання зовнішніх API, віддаленого доступу, інтеграції з Customer Relationship Management (далі CRM), ERP та іншими бізнес-системами. Як наслідок, інформаційні потоки все частіше залишають межі локальної мережі, проходячи через зовнішні канали, рівень захищеності яких складно контролювати або оцінити з боку внутрішніх IT-відділів. Це відкриває додаткові вектори ризику, зокрема можливість стороннього втручання, компрометацію даних під час їх передачі або недотримання вимог щодо конфіденційності, і ці помилки є критичними у рамках стандартів на кшталт General Data Protection Regulation (далі GDPR) чи ISO/IEC 27001. З огляду на це, забезпечення інформаційної безпеки не може обмежуватися окремими інструментами, навіть якщо вони є технічно досконалими. Необхідно впроваджувати багаторівневу модель захисту, що орієнтована на контекст роботи підприємства. Такий підхід включає в себе IDS, багатофакторну аутентифікацію, сегментацію мереж, аудит доступу, регулярне оновлення критично важливих компонентів, а також підтримку планів безперервності бізнесу та аварійного відновлення. Водночас, із розвитком хмарних технологій виникають нові питання щодо розподілу відповідальності між постачальником послуги та замовником. Останній нерідко недооцінює обсяг своїх зобов'язань у частині налаштування, шифрування, ведення журналів подій та контролю доступу. Варто також враховувати, що навіть за наявності належної технічної реалізації, основна загроза може виходити саме від людського фактора. Нестача обізнаності співробітників щодо базових

принципів інформаційної безпеки, нехтування внутрішніми політиками або недбале ставлення до створення надійних паролів — усе це здатне значно знизити рівень захищеності [1]. До цього додаються інсайдерські ризики, коли витік інформації або саботаж здійснюється зсередини компанії. Сучасні підходи до інформаційної безпеки дедалі частіше концентруються на управлінні поведінкою користувачів: впроваджуються регулярні навчальні тренінги, проводяться моделювання фішингових атак, використовуються засоби DLP, а також автоматизовані системи контролю доступу до критичних активів.

Не менш важливим аспектом є постійне тестування діючої моделі безпеки на предмет її відповідності новим загрозам. Це включає як ручні аудити, так і застосування спеціалізованих інструментів для сканування вразливостей, моделювання атак, аналізу поведінки користувачів (UEBA) та оцінки дієвості застосованих політик. Без належного процесу перевірки ефективність системи безпеки залишається умовною, оскільки загрози еволюціонують швидше, ніж оновлюються внутрішні регламенти та процедури захисту. У зв’язку з цим, на перший план виходить необхідність своєчасного виявлення потенційних вразливостей ще на етапі попереднього аналізу - так званої reconnaissance (далі рекон), яка дозволяє отримати уявлення про інфраструктуру організації. Цей процес дуже важливий як для забезпечення власної захищеності в рамках захисного сценарію, так і для ідентифікації можливих векторів атак у контексті оцінки ефективності існуючих заходів безпеки.

Наведене викликано тим, що сучасне цифрове середовище формує надзвичайно складний ландшафт загроз, котрі швидко еволюціонують і охоплюють як технічні, так і людські чинники. Для того щоб проводити всебічний аналіз ризиків інформаційної безпеки на підприємстві, необхідно використовувати системний підхід до їхньої класифікації; у стандартному поділі виокремлюють три головні категорії: технічні загрози, організаційні загрози та соціоінженерні атаки. Завдяки такому розподілу можливо чіткіше ідентифікувати джерела ризику, розробити відповідну стратегію протидії та

апріорі визначити найбільш уразливі вектори атак - як з огляду на інфраструктурні елементи, так і з урахуванням людського фактору.

Технічні загрози охоплюють уразливості в апаратному та програмному забезпеченні, помилки в конфігурації мереж, а також активні дії зловмисників, спрямовані на використання цих вразливостей. Серед класичних прикладів технічних атак можна виділити DoS та DDoS - спроби перевантаження серверів через масові запити, що призводить до відмови в обслуговуванні [2]. Часто вони виступають не самостійною атакою, а частиною більшої операції, наприклад, як відволікальний маневр перед проникненням у внутрішню мережу. До технічних загроз також належать атаки типу Man-In-The-Middle, які дозволяють перехоплювати, змінювати або скеровувати трафік, що проходить між користувачем і сервером. Такі атаки зазвичай використовують протокольні недоліки або слабкі налаштування шифрування, як приклад відсутність TLS або використання недійсних SSL-сертифікатів, а також техніки Address Resolution Protocol (ARP)- або DNS-spoofing. Окрему загрозу становить шкідливе програмне забезпечення: програми типу ransomware шифрують дані користувача і вимагають викуп за їх розблокування, тоді як spyware діє приховано, передаючи інформацію третім особам. Rootkit-и та backdoor-и, своєю чергою, забезпечують непомітний і стійкий несанкціонований доступ до системи. Загроза ускладнюється тим, що такі інструменти можуть залишатися активними в інфраструктурі роками - особливо якщо системи не оновлюються, а журналювання та моніторинг здійснюються формально або взагалі відсутні.

Організаційні загрози виникають через людський фактор і слабке адміністрування. Часто вони мають нетехнічний характер, але наслідки можуть бути критичними. Наприклад, недбалість персоналу або відсутність внутрішніх політик може призвести до витоку конфіденційних даних - наприклад, при використанні слабких паролів, збереженні документів у незахищених спільних папках або пересиланні важливої інформації через особисті поштові скриньки. Частою проблемою є ситуація, коли після звільнення співробітника його обліковий запис залишається активним або має доступ до корпоративних

сервісів. Подібні прорахунки в управлінні доступом можуть стати першою точкою проникнення для зловмисника.

Соціоінженерні загрози базуються на психологічному маніпулюванні людиною. Найбільш поширеним видом є фішинг - коли зловмисник імітує відомі структури (наприклад, банк або ІТ-відділ), щоб виманити логін або пароль. Інші техніки включають pretexting - ситуації, коли атакуючий видає себе за керівника або колегу, щоб отримати потрібну інформацію [3]. Ускладнені варіанти передбачають поєднання цифрових і фізичних засобів: наприклад, телефонний дзвінок у відділ підтримки одночасно з надсиланням фальшивого листа. Такі загрози особливо небезпечні, оскільки об’єктом атаки є не система, а людина, яку набагато важче "захистити" технічними засобами.

Для ефективної протидії загрозам недостатньо просто їх класифікувати - необхідно також розуміти, через які саме точки доступу зловмисник може здійснити атаку. У цьому контексті ключовим поняттям є поверхня атаки - це сукупність усіх потенційних точок входу до інформаційної системи. Аналіз поверхні атаки дозволяє структурувати ризики і виявити ті ділянки інфраструктури, які найбільш вразливі до зовнішнього або внутрішнього впливу [4].

Внутрішня поверхня атаки включає всі ті елементи, які перебувають під контролем організації. Сюди належать внутрішні сервери, мережеві пристрої, робочі станції, сервіси аутентифікації, IP-адреси, маршрутизатори, локальні БД, а також корпоративні вебдодатки. Внутрішня поверхня атаки зазвичай охоплюється регулярними перевірками ІБ-відділу, проте тут часто трапляються критичні помилки - наприклад, відкриті порти, наявність служб з “дефолтними” налаштуваннями або застаріле ПЗ без оновлень. Якщо зловмисник потрапляє всередину (наприклад, через той самий фішинг), саме внутрішня поверхня відкриває йому шлях до розгортання горизонтального переміщення та ескалації привілеїв.

Зовнішня поверхня атаки охоплює всі ті ресурси, які доступні через Інтернет і взаємодіють із системами підприємства. Це можуть бути хостинги,

зовнішні API, сторонні постачальники SaaS-рішень, відкриті вебпортали, CRM-системи, DNS, поштові шлюзи, або CDN. Часто організація делегує безпеку цих компонентів третім особам, однак саме через них здійснюються масові сканування та експлуатація вразливостей. Приклад - випадкове відкриття адміністративного інтерфейсу додатку через незахищений порт або помилка конфігурації reverse proxy.

Невідома або прихована поверхня атаки є найбільш небезпечною, оскільки включає компоненти, про які організація або не знає, або не відстежує. Це може бути тіньова інфраструктура, залишки старих проєктів, облікові записи колишніх співробітників, відкриті bucket-и в хмарі, або навіть публічні архіви з конфіденційними файлами, що з’явилися через людську помилку. До цього типу також належать випадки витоку паролів у даркнеті, скомпрометовані токени, тощо. Аналіз і зменшення такої поверхні можливий лише через активні дії за допомогою рекону, моніторингу цифрового сліду компанії або використання інструментів зовнішнього сканування. На (рис. 1.1) представлена класична схема класифікації поверхонь атаки за трьома основними векторами, зокрема внутрішня, зовнішня та невідома поверхня атаки [5].

Рисунок 1.1 – Класифікація поверхонь атаки за трьома основними векторами

Формування надійної системи інформаційної безпеки, здатної адекватно реагувати на актуальні загрози та адаптуватися до змін середовища, неможливе без опори на фундаментальні принципи, що закладають концептуальну основу архітектури захисту. У цьому контексті ключову роль відіграють постулати моделі **CIA** (**C**onfidentiality, **I**ntegrity, **A**vailability), PoLP, а також практики сегментації мережевих середовищ і реалізації системного аудиту доступу до інформаційних ресурсів. Незважаючи на широке розповсюдження зазначених підходів і їхнє визнання як базових, вони потребують певного рівня адаптації до умов конкретної організаційної моделі, структури IT-інфраструктури і рівня зрілості процесів підприємства [6].

Модель CIA традиційно слугує вихідною точкою для побудови політик безпеки, передбачає збалансоване забезпечення трьох нерозривно пов'язаних аспектів - конфіденційності, цілісності та доступності інформації. Захист конфіденційності, у цьому випадку, досягається шляхом застосування сучасних криптографічних алгоритмів, побудови сегрегованих зон доступу, реалізації багатофакторної автентифікації та політик розмежування доступу на основі ролей. Цілісність інформаційних потоків і цифрових артефактів підтримується за рахунок впровадження механізмів контролю версій, цифрових підписів, перевірки геш-сум, а також моніторингу змін у системах зберігання даних. Що ж стосується доступності - тут критичним є впровадження рішень з балансування навантаження, організація резервного копіювання з географічною відмовостійкістю, а також захист від атак типу DoS/DDoS, зокрема через вбудовані у фаєрволи механізми rate-limiting і reverse proxy.

Окремо варто зупинитися на принципі найменших привілеїв, який передбачає надання користувачеві або сервісу лише тих повноважень, які є критично необхідними для виконання покладених на нього функцій. Реалізація цього принципу, як показує практика, суттєво знижує ризики як ненавмисних помилок з боку персоналу, так і цілеспрямованих деструктивних дій з боку зловмисника, який потенційно міг би отримати контроль над скомпрометованим обліковим записом. Більше того, суворе дотримання PoLP у поєднанні з

регулярним переглядом прав доступу, сегментацією адміністративних ролей і контролем над сервісними акаунтами створює середовище, у якому вертикальна ескалація привілеїв стає значно складнішою з технічного погляду, а отже менш ймовірною на практиці.

Ще одним невід’ємним елементом архітектури безпеки є логічна сегментація мережі, яка дозволяє ізолювати важливі компоненти від менш захищених або зовнішніх сегментів інфраструктури [7]. Цей підхід значно ускладнює горизонтальне переміщення зловмисника всередині корпоративної мережі після первинного проникнення, що відповідно знижує потенційну шкоду від інциденту. Водночас, ефективність сегментації суттєво зростає при її інтеграції з інструментами обліку та моніторингу доступів.

1.2 Інфраструктура організації як об'єкт вивчення кіберрозвідкою

Цифрова трансформація бізнес-середовища IT-інфраструктури організації вже давно перестала сприйматися виключно як сукупність технічних засобів для забезпечення функціонування інформаційних систем. Дедалі організації більше набуває ознак стратегічного ресурсу, від ступеня захищеності якого прямо залежить стабільність критичних бізнес-процесів. В наслідок цього підвищується актуальність системного вивчення IT-інфраструктури як потенційного об'єкта кіберрозвідки - тобто дій, які спрямовані на збирання технічної інформації про об'єкти інтересу з метою ідентифікації вразливих компонентів і розбудови векторів атаки.

Інформаційно-технологічна інфраструктура сучасного підприємства, навіть у найтиповішій конфігурації - становить багаторівневу систему апаратних, програмних та мережевих компонентів, кожен із яких - у тій чи іншій мірі - може бути джерелом потенційно цінної інформації з погляду розвідки. Серед ключових передумов для ефективного аналізу такої інфраструктури – це необхідність у комплексному підході, що враховує і

логічну структуру взаємозв'язків між сервісами, і специфіку їхньої зовнішньої репрезентації, зокрема у відкритих мережах.

На етапі первинного рекону особливий інтерес становлять саме ті компоненти, які безпосередньо взаємодіють із зовнішнім середовищем. З огляду на це, першочерговим об'єктом розвідки зазвичай стають веб-сервери - такі як Apache HTTP Server, Nginx або Microsoft IIS. Вони часто виявляються доступними через стандартні порти (80/443), і через це можуть бути швидко виявлені за допомогою інструментів масового сканування. Крім того, під час взаємодії із цими сервісами зловмисник здатен ідентифікувати заголовки відповіді, що містять версію програмного забезпечення, специфіку CMS, наявність плагінів або навіть особливості структури URL-адрес. Усе це, зрештою, формує картину потенційної атаки, однак на цьому етапі відіграє роль індикаторів для подальшого фазингу.

Не менш цінним джерелом інформації виступають DNS-сервери, що забезпечують перетворення імен у IP-адреси. З позиції розвідки вони дозволяють побудувати логічну карту доменного простору організації. За допомогою методів або активного збору субдоменів можливо реконструювати внутрішню або тестову структуру сервісів, що не відображається у публічному просторі. У разі наявності неправильно сконфігурованих зон, доступних для запиту AXFR, зловмисник без додаткового навантаження отримує повну карту DNS-записів, що значно прискорює наступні етапи дослідження.

Поштові сервери, зокрема ті, що використовують протоколи SMTP, POP3 або IMAP, також легко виявляються шляхом перевірки MX-записів або через банальний метод банер-грабінгу. Зібрані дані щодо конфігурації, типу поштового шлюзу та навіть підтримуваних механізмів аутентифікації можуть бути використані як для підготовки фішингових кампаній, так і для реконструкції організаційної структури - наприклад, через патерни іменування адрес.

Сегмент користувацьких пристроїв - особливо в умовах політик Bring Your Own Device (BYOD) або віддаленої роботи - складно піддається

безпосередньому аналізу, проте через побічні індикатори, такі як залишки користувацьких агентів у веб-заголовках, можна робити висновки про тип операційної системи, використовувані браузери, мобільні клієнти тощо. Ці дані не є цільовими в контексті першочергового рекону, проте можуть слугувати допоміжним орієнтиром для побудови соціотехнічних сценаріїв або атак на слабко захищені периферійні вузли.

Найбільшу увагу в процесі розвідки заслуговує доменна інфраструктура, наприклад яка реалізована на базі Active Directory. Хоча й без прямого доступу до внутрішньої мережі витягти її структуру є проблематично, деякі мета-дані все ж можуть бути отримані опосередковано - через відкриті Lightweight Directory Access Protocol (LDAP)-порти, службові доменні імена в сертифікатах, витоки конфігурацій з Jenkins, Git або навіть через витяг .git-директорій при неправильному налаштуванні веб-сервера. Навіть такі, здавалося б фрагментарні дані можуть допомогти визначити внутрішні імена хостів, структуру OU або існування сервісних облікових записів.

Транспортна інфраструктура, що включає в себе VPN-шлюзи, точки доступу, маршрутизатори і фаєрволи, зазвичай менш видима на рівні відкритого сканування. Водночас, за умови неправильної сегментації їх присутність може бути виявлена через типові сигнатури відповіді або SSL-сертифікати. Зібрані відомості (наприклад, специфікації прошивок, відкриті панелі керування тощо) не завжди мають миттєву практичну цінність, однак дозволяють зрозуміти загальний рівень технічної зрілості IT-департаменту цільової організації.

Інструменти спостереження, такі як системи логування, моніторингу як от Zabbix чи Prometheus, або резервного копіювання, рідко потрапляють у зону прямої розвідки, однак можуть бути ідентифіковані через непрямі ознаки – наприклад, залишки JavaScript у веб-інтерфейсах, несанкціоновані відповіді API або відсутність автентифікації на тестових хостах. Їх виявлення зазвичай не є пріоритетом, але в сукупності зі зібраною інформацією може підсилити загальну карту інфраструктури.

Гібридна інфраструктура - це інфраструктура яка поєднує в собі традиційні локальні компоненти та хмарні об'єкти. В контексті цього - основною ціллю розвідки стають облікові дані доступу до API, відкриті S3-бакети, конфігураційні файли з .env-структурою, що зберігаються у відкритих репозиторіях. У подібному середовищі акцент зміщується у бік OSINT-практик: сканування GitHub, використання інструментів типу GitLeaks, пошук ключів в архівах Wayback Machine.

Також не слід нехтувати промисловими системами, а також IoT-пристроями, які часто мають фіксовані IP-адреси й спрощені інтерфейси доступу. Їх присутність, при певному досвіді, може бути визначена навіть за специфічними банерами або HTTP-заголовками. В реальних умовах інженерні вузли рідко використовують стандартні заходи маскування, що суттєво полегшує їхню ідентифікацію.

1.3 Рекон як підхід до аналізу кіберзагроз

Успішність сучасних кібератак багато в чому залежить від якості попередньої розвідки - тобто збирання технічної інформації про цільову IT-інфраструктуру. У сфері безпеки цей процес називають реконом. Залежно від методу доступу до даних, рекон поділяється на пасивний та активний. Кожен із підходів має власні переваги й обмеження, а також впливає на рівень виявлення з боку систем моніторингу. Далі буде розглянуто особливості пасивного й активного рекону окремо.

1.3.1 Методи пасивного збору інформації

Пасивний рекон передбачає отримання інформації без безпосереднього контакту з цільовими системами. Зловмисник, умовно "спостерігає" за системою, не взаємодіючи з інфраструктурою напряму. Це мінімізує ризик виявлення, оскільки не створюються записи в логах і не активуються захисні

механізми. Основу таких дій складають техніки OSINT, а також аналіз публічних артефактів: DNS-записів, історії WHOIS, метаданих документів, згадок у зламаних базах даних тощо. Пасивний підхід особливо корисний на ранніх етапах атаки або під час побудови соціотехнічних сценаріїв.

Однією з основних складових пасивної розвідки - використання протоколу WHOIS, який забезпечує доступ до інформації про реєстрацію доменних імен та відповідних їм ресурсів у мережі Інтернет [8]. Застосування WHOIS-запитів дозволяє отримати розширені відомості, як от адміністративні та технічні контакти, дати створення та закінчення терміну дії домену. Така інформація, що збирається без прямої взаємодії з цільовою системою, відкриває можливість ідентифікувати власника вебресурсу, відстежити потенційні зв’язки між субдоменами, що нерідко вказує на приховані внутрішні структури інформаційної системи або ж навіть приналежність до певної організаційної групи. Вивчення WHOIS-записів дозволяє формувати цілісне уявлення про архітектуру мережевої інфраструктури, визначаючи потенційно вразливі сегменти, що в подальшому стають об'єктами подальшого аналізу. Особливо це стосується ситуацій, коли виявляються історичні зміни в реєстраційних даних або налаштуваннях серверів імен, які свідчать про міграцію сервісів, зміну хостинг-провайдерів чи модифікацію політик безпеки. Зокрема, спостереження за динамікою змін у WHOIS-інформації дозволяє визначити часові інтервали, в які відбувались оновлення домену, що може слугувати індикатором запланованих оновлень або потенційних вразливостей під час перехідних етапів.

Окрім WHOIS-запитів, вагоме значення у процесі пасивної розвідки відіграє дослідження DNS-записів, що дозволяють відтворити мережеву топологію об’єкта дослідження. Такі записи містять критичні дані щодо відповідності доменних імен IP-адресам (A-записи), маршрутизації електронної пошти (MX-записи), а також специфічних налаштувань безпеки, таких як Sender Policy Framework (SPF) та DomainKeys Identified Mail (DKIM), що визначають політики перевірки автентичності електронної пошти [9].

A-записи дозволяють визначити фізичне розташування серверів у мережі та їхню взаємодію з іншими компонентами інформаційної системи. MX-записи - відображають шляхи маршрутизації поштових повідомлень, що за певних обставин, може надати можливість для подальшого аналізу безпеки поштових серверів. TXT-записи, які часто використовуються для зберігання конфігураційних даних, зокрема SPF або DKIM, можуть слугувати індикатором політик аутентифікації електронної пошти, а також забезпечувати додаткову інформацію щодо налаштувань безпеки, що реалізовані на рівні поштових серверів. Аналіз DNS-записів здатен розкрити не лише поточну конфігурацію мережевої інфраструктури, але ще й її історичний стан за допомогою запитів до кешованих або архівних даних. Це вже відкриває перспективи для виявлення старих або неналежним чином захищених вузлів, які могли залишитися непоміченими після проведення оновлень або міграції сервісів.

Аналіз витоків даних, опублікованих у відкритому доступі - забезпечує отримання важливої інформації про можливі вразливості організації. Відповідні ресурси, такі як "Have I Been Pwned?", дозволяють відстежувати факти компрометації облікових записів, пов'язаних зі співробітниками цільової інфраструктури [10]. Зазвичай зловмисники та тестери на проникнення використовують ці дані для подальшого здійснення спрямованих атак, зокрема фішингових кампаній або підбору автентифікаційних даних для доступу до внутрішніх ресурсів. Здобута інформація може включати як облікові дані (логіни, паролі), так і конфіденційну інформацію (адреси електронної пошти, номери телефонів), що може істотно підвищити ризик несанкціонованого доступу до сегментів інформаційної системи.

Особливе місце в контексті пасивної розвідки займає використання сервісу Wayback Machine, який являє собою своєрідний архів вебсторінок, що зберігає їхні попередні версії. Можливість відновлення історичних версій вебресурсів відкриває перспективи для виявлення раніше доступних розділів або функцій, які на момент проведення розвідки вже могли бути видалені або приховані. Це дозволяє дослідити службові URL, тестові розділи, застарілі

інтерфейси або панелі адміністрування, які на етапі експлуатації могли містити вразливості, доступ до яких зловмисник може використати для несанкціонованого проникнення в систему. Вивчення попередніх версій вебсайтів також дозволяє виявити потенційні конфігураційні помилки або залишені без належного захисту ресурси, що здатні слугувати точками входу для атак.

В межах пасивного рекону особливу цінність становлять спеціалізовані сервіси, такі як Shodan, Censys, FOFA та crt.sh, які індексують технічні характеристики мережевих пристроїв і сервісів замість вмісту вебсторінок. Shodan здійснює сканування IP-простору для виявлення відкритих портів, банерів та IoT-пристроїв, що дозволяє оцінити загальну експозицію інфраструктури. Censys зосереджується на аналізі TLS/SSL-сертифікатів і зв'язках між доменами, IP-адресами та криптографічними атрибутами. FOFA - китайський аналог, що забезпечує доступ до великої бази даних мережевих активів і може слугувати додатковим джерелом перевірки. crt.sh - надає історію сертифікатів доменів у рамках ініціативи Certificate Transparency, що дозволяє виявляти приховані піддомени та відстежувати зміну структури інфраструктури без прямого контакту з ціллю [11].

Методика Google Dorking являє собою один із найбільш концептуально витончених підходів до збору відкритої інформації, заснований на використанні розширених синтаксичних конструкцій пошукових запитів [12]. У цьому контексті мова йде про спеціалізовані оператори, що дозволяють надзвичайно точно фільтрувати індексований контент, відсікаючи при цьому малозначущу або тривіальну інформацію. Попри свою формальну простоту, цей метод створює передумови для глибокого аналітичного занурення у цифровий ландшафт цільової організації без необхідності прямого контакту з її інформаційними ресурсами — що, власне, і є ключовою перевагою в межах пасивного етапу OSINT-дослідження. Слід зазначити, що Google Dorking, попри його технічну пасивність, дозволяє побічно оцінити рівень кібергігієни організації - насамперед у розрізі налаштувань доступу до публічних

директорій, конфігураційних файлів і документації. Недбале або несвоєчасне внесення обмежень у файл robots.txt, відсутність правил індексації для службових сторінок, а також ігнорування базових політик control access - усе це виявляється через елементарні дорки, що, однак, демонструє серйозні прорахунки в моделі публікування цифрового контенту. Концептуальне значення даного методу полягає не лише у виявленні відкритих вразливих точок, але і в побудові гіпотетичної моделі поведінки організації у публічному інтернет-просторі: які дані вона готова демонструвати, а які - потрапляють у доступ випадково, внаслідок неуважності або недосконалих автоматичних процедур розгортання.

Основим напрямком у структурному аналізі мережевої інфраструктури є виявлення піддоменів, оскільки вони зазвичай відображають внутрішню логіку організаційного поділу ресурсів. Піддомени можуть виступати точками доступу до середовищ тестування, обслуговування, архівування чи внутрішніх адміністративних панелей - компонентів, що вказують на специфіку функціонального зонування цифрового простору організації. З аналітичної точки зору, піддомени є своєрідними маркерами, які дозволяють відтворити загальну топологію інформаційної структури. Їх наявність часто вказує на внутрішню класифікацію сервісів (наприклад, api., dev., mail., legacy.), що дає змогу умовно реконструювати IT-ландшафт компанії. Особливої уваги потребують застарілі або нефункціональні піддомени: навіть такі фрагменти, як стара версія CMS чи відкритий порт, можуть містити важливу технічну інформацію й стати відправною точкою для побудови потенційних векторів атаки.

У структурі OSINT-розвідки важливе місце посідає аналіз професійних соціальних мереж - таких як LinkedIn, GitHub, Stack Overflow, Xing, ResearchGate тощо [13]. Ці ресурси забезпечують доступ до персоніфікованої та контекстуалізованої інформації, яка за інших умов залишалася б недоступною. На відміну від технічних джерел, що переважно відображають інфраструктурні або протокольні характеристики систем, профілі співробітників організації

розкривають її соціотехнічний аспект. Аналіз змісту таких профілів дає змогу не лише реконструювати організаційну структуру, а й виявити технологічні пріоритети компанії, напрями реалізованих або поточних проєктів, рівень технічної кваліфікації персоналу та його професійні інтереси. Особливо цінними є згадки про стек технологій, середовища розробки, корпоративні стандарти або навіть випадково оприлюднені IP-адреси, фрагменти конфігурацій чи скриншоти з внутрішніх систем. У сукупності такі дані формують уявлення про рівень обізнаності співробітників у сфері безпеки та дозволяють оцінити потенційні ризики витоку інформації через людський фактор.

Окрему увагу слід приділяти ідентифікації ключових фігур у структурі компанії - осіб, що володіють адміністративними правами, приймають технічні рішення або безпосередньо впливають на політику кіберзахисту. Аналіз зв’язків між цими особами дозволяє відтворити схеми внутрішньої взаємодії та виявити потенційні вектори для соціотехнічного впливу.

1.3.2 Методи активного збору інформації про цільову систему

На відміну від пасивного, активний рекон передбачає пряму взаємодію з елементами цільової інфраструктури. Це може бути сканування відкритих портів, запити до веб-додатків, виявлення сервісів за допомогою банер-граббінгу, визначення ОС тощо. Такі дії є більш «шумними» і часто фіксуються засобами захисту, однак дають точніші й глибші технічні дані. Активний підхід доцільний тоді, коли необхідна детальна карта сервісів або перевірка конкретних векторів атаки.

Сканування портів є одним із найбільш розповсюджених методів активного рекону, оскільки дозволяє не тільки ідентифікувати відкриті порти на сервері, а й зробити певні припущення про типи доступних сервісів та потенційні вразливості. Серед інструментів для сканування портів найбільш відомим є Nmap [14]. Цей потужний сканер дає можливість проводити глибоке

дослідження системи, включаючи виявлення операційних систем, версій програмного забезпечення та можливих вразливостей за допомогою NSE. Вочевидь, Nmap є незамінним для детального аналізу, оскільки підтримує різноманітні методи сканування, дозволяючи налаштовувати діапазони портів, протоколи та інші параметри. Проте інколи виникає потреба в більш швидкому скануванні великих ділянок мережі, що робить Masscan, ще один інструмент для сканування портів, значно кориснішим. Masscan працює значно швидше за Nmap завдяки оптимізації для великих обсягів трафіку, але обмежується відсутністю підтримки скриптів NSE. Тому в залежності від конкретних цілей завдання, варто вибирати між глибоким та детальним аналізом, для чого краще підходить Nmap, або ж необхідністю в швидкому масштабному скануванні, для чого Masscan є оптимальним варіантом.

Фазинг є методом подачі випадкових або навмисно некоректних даних до вебсервера з метою виявлення потенційних вразливостей, які можуть бути використані зловмисниками для реалізації атак. Цей підхід є важливою складовою тестування безпеки вебресурсів, оскільки дозволяє виявити слабкі місця у формах введення, параметрах URL-запитів, cookie-файлах та інших елементах взаємодії з вебінтерфейсом. Одним із найбільш ефективних інструментів для реалізації фазингу є FFUF - високошвидкісний fuzz-сканер, що підтримує проведення словникових атак на шляхи, параметри, заголовки HTTP-запитів та інші ключові елементи вебдодатку. Завдяки своїй гнучкості та продуктивності, FFUF дозволяє ефективно виявляти приховані директорії, сторінки, API-ендпойнти, а також потенційно вразливі параметри. Іншим сучасним інструментом, що демонструє високу ефективність у контексті активного збору інформації, є Katana - сканер нового покоління, який поєднує функціонал краулінгу, фазингу та пасивного рекону. Katana автоматично виявляє URL-адреси, параметри, скрипти та інші ресурси вебсторінок, формуючи карту потенційної поверхні атаки. Особливістю Katana є її здатність до паралельної обробки великої кількості запитів, що робить її придатною для аналізу сучасних динамічних вебдодатків. Обидва інструменти FFUF і Katana - можуть

ефективно використовуватись в автоматизованих recon-ланцюжках для реалізації комплексного підходу до фазингу та попереднього аналізу безпеки цільової інфраструктури.

1.4 Етичні, правові та моральні аспекти кіберрозвідки

Попри технологічну доцільність використання засобів кіберрозвідки у контексті зміцнення інформаційної безпеки, постає важливе питання щодо легітимності, етичної допустимості та моральної обґрунтованості подібної діяльності. У випадках, коли розвідка виконується з метою аудиту безпеки, тестування на проникнення або в рамках службового розслідування - її правовий статус зазвичай чітко врегульований контрактними угодами, регламентами або внутрішніми політиками організації. Проте навіть у таких умовах важливо враховувати принципи етичної взаємодії з інформаційною системою.

Суттєвим фактором є обмеження на збір, зберігання та обробку персональних даних, а також заборона на порушення приватності при здійсненні навіть формально дозволеного аналізу. Будь-які дії, пов'язані із взаємодією з мережевою інфраструктурою, повинні супроводжуватись збереженням принципу мінімального впливу - тобто зводити до мінімуму ризики переривання сервісів, викривлення логів або компрометації конфіденційних даних. У правовій площині слід брати до уваги положення національного законодавства, зокрема Закон України «Про основні засади забезпечення кібербезпеки України» [15], а також нормами Кримінального кодексу (ст. 361–363-1), які прямо забороняють доступ до систем без відповідного дозволу, навіть з дослідницькою метою [16]. З моральної точки зору, здійснення кіберрозвідки - навіть за відсутності прямого втручання - вимагає дотримання загальноприйнятих норм цифрової етики. Збирання інформації з відкритих джерел або дослідження професійних профілів співробітників компанії, повинно ґрунтуватися на принципах поваги до приватності, доброчесності та обмеженості мети. Будь-які дії, що виходять за межі цих принципів, можуть не лише поставити під загрозу репутацію фахівця або організації, а й спричинити юридичні наслідки, особливо у випадках, коли

зібрані дані використовуються або зберігаються з порушенням вимог GDPR або національного законодавства щодо захисту персональної інформації.

Висновки за розділом 1

У першому розділі було проведено теоретичний аналіз об’єкта захисту - інформаційної інфраструктури організації як складної, багаторівневої та динамічної системи, що постійно зазнає впливу внутрішніх і зовнішніх факторів. Розглянуто основи інформаційної безпеки в корпоративному середовищі, класифікацію загроз, типи поверхонь атаки та принципи побудови захищених інформаційних систем, включно з концепціями CIA, PoLP і сегментації мереж.

Особливу увагу приділено аналізу кіберрозвідки як одного з ключових етапів у циклі інформаційної атаки. Було розмежовано поняття пасивного та активного рекону, описано їх характерні особливості, методи збору інформації, а також можливі джерела відкритих технічних даних. Показано, що саме на цьому етапі зловмисники формують розуміння структури IT-інфраструктури цільової організації та виявляють її слабкі ланки.

Розділ закладає теоретичне підґрунтя для подальшого вивчення практичних аспектів кіберрозвідки та побудови інструментів, які дозволяють реалізувати цей процес автоматизовано. У наступному розділі буде зосереджено увагу на практичній реалізації концепцій, розглянутих у цьому теоретичному блоці.

РОЗДІЛ 2
МЕТОДИ ТА ІНСТРУМЕНТИ КІБЕРРОЗВІДКИ ДЛЯ АНАЛІЗУ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ

Кіберрозвідка є одним із ключових етапів у процесі аналізу безпеки цифрової інфраструктури. Саме з неї починається побудова моделі загроз та формування уявлення про технічні і конфігураційні особливості цільового середовища. На практиці цей процес охоплює широкий спектр дій - від збору публічних технічних артефактів до цілеспрямованого аналізу активних сервісів і компонентів програмного забезпечення. Основною метою є створення максимально повної та релевантної картини цифрової поверхні атаки, що дозволяє як оцінити рівень ризику, так і виявити потенційно вразливі елементи на ранніх етапах. Методи, які застосовуються в рамках кіберрозвідки, істотно відрізняються за рівнем інтерактивності з цільовою інфраструктурою. У цьому контексті важливо не стільки повторювати класифікацію на пасивні чи активні підходи, скільки зосередитись на практичному застосуванні відповідних механізмів, ефективності інструментів та характері здобутої інформації.

У підсумку, даний розділ формує цілісну методологічну базу, необхідну для реалізації ефективного розвідного етапу у межах оцінки безпеки інформаційних систем. Крім того, розділ містить обґрунтування необхідності створення власного Bash-скрипта для автоматизації процесу збору даних. Такий підхід розглядається як практична альтернатива до використання наявних рішень, що, незважаючи на їхню універсальність, не завжди відповідають специфічним вимогам окремих сценаріїв або структур організаційної інфраструктури. Реалізація цього підходу буде докладно розглянута у наступному розділі.

2.1 Використання пасивних технік для збору OSINT-інформації

На етапі прикладної реалізації пасивного рекону, WHOIS-аналіз виступає інструментом поглибленого профілювання доменних ресурсів, що дозволяє здійснювати попередню атрибуцію інфраструктурних компонентів без потреби у взаємодії з цільовими вузлами. На відміну від загального опису протоколу WHOIS, що розглядається в розділі 1.3, на цьому етапі дослідження особливу увагу слід зосередити на обробці виводу інструментальних засобів, інтеграції результатів у загальний профіль цільової організації та кореляції отриманих ідентифікаторів.

З технічної точки зору, базовий запит до WHOIS-сервера може бути здійснений за допомогою командного рядка:

whois example.com

У відповіді, залежно від політики приватності доменного реєстратора, можуть бути виявлені: ім’я реєстратора, дати створення/закінчення домену, імена серверів імен (NS), контактні електронні адреси та додаткові адміністративні поля. Зібрані email-адреси можуть бути використані для наступного етапу OSINT-аналізу - перевірки на предмет появи у відкритих витоках даних або ідентифікації акаунтів у публічних сервісах. Окрім того, деякі доменні реєстратори надають непрямі індикатори про розміщення сервісів - наприклад, вивід, що вказує на використання DNS-сервісів від Cloudflare або Google Domains, дозволяє зробити припущення про присутність проксі-захисту або CDN.

Застосування WHOIS може бути автоматизоване, зокрема через обгортки на Python (наприклад, бібліотека python-whois) або скрипти на Bash, що ітерують список доменів та агрегують результати для подальшого аналізу. Це відкриває можливість не лише для однократного отримання даних, а й для моніторингу змін у реєстраційних полях з плином часу - наприклад, для

виявлення переходу між хостинг-провайдерами або змін у політиці делегування домену.

Ще одним дієвим способом пасивно-активного цифрового профілювання є застосування утиліти WhatWeb, яка дозволяє визначити технології, що лежать в основі вебресурсу. Вона аналізує HTTP-заголовки, код сторінки, відповіді сервера, наявність JavaScript-фреймворків та CMS-ідентифікаторів, автоматично встановлюючи, зокрема, використання WordPress, Joomla, nginx, Apache, Google Analytics, jQuery тощо [17]. Типовий запит виглядає наступним чином:

whatweb -v https://example.com

У відповідь користувач отримує структурований звіт, у якому відображено всі розпізнані компоненти, версії програмного забезпечення (за наявності) та сигнатури, на яких базується розпізнавання. Подібний підхід дозволяє оперативно скласти уявлення про технологічний стек сайту та потенційні вектори атак - наприклад, через відомі уразливості CMS або вебсерверів. WhatWeb також підтримує режими масового сканування, що дозволяє інтегрувати його в більш складні recon-ланцюжки або Bash-скрипти. Така інтеграція корисна у випадках, коли необхідно провести автоматизовану інвентаризацію великої кількості вебресурсів в межах однієї організації або доменного простору.

Поглиблений аналіз DNS-записів на етапі практичної реалізації пасивної розвідки дозволяє не лише відтворити логічну топологію об'єкта дослідження, а й виявити потенційні вектори для наступних етапів сканування або вторгнення. У контексті прикладного застосування, акцент робиться на інструментальних підходах до запиту різних типів записів, автоматизації збору даних та їхньої подальшої класифікації. Типові запити до DNS-серверів реалізуються за допомогою утиліт dig або host [18]. Наприклад:

dig example.com any

або цільовий запит до MX-записів:

host -t mx example.com

Отримані значення A-записів дозволяють зв’язати домен із конкретними IP-адресами, після чого можна здійснити геолокаційний аналіз, перевірку за чорними списками, а також уточнення факту наявності прямого доступу до серверів без проксі-шарів. У випадку присутності Canonical Name (CNAME)-записів, доцільно здійснювати трасування до оригінального ресурсу, що дозволяє виявити реальних провайдерів послуг або залежності від сторонніх платформ (наприклад, хмарних сервісів).

Окрему аналітичну цінність мають TXT-записи, які часто містять політики SPF, DKIM або DMARC. Їхнє існування та правильність налаштування свідчать про рівень зрілості інформаційної безпеки в контексті поштової інфраструктури. Наприклад, запис SPF виду:

"v=spf1 include:_spf.google.com ~all"

Вказує на делегування функцій поштової доставки сторонньому сервісу (у цьому випадку - Google), а модифікатор ~all дозволяє ідентифікувати м’який режим політики, що потенційно відкриває простір для спуфінгу.

Крім використання CLI-утиліт, доцільним є застосування спеціалізованих агрегаторів, зокрема DNSDumpster, SecurityTrails або PassiveTotal, які об’єднують кілька джерел і дозволяють швидко отримати карту піддоменів, історичні зміни записів, а також часові мітки, які дають змогу виявляти взаємозв'язки між об'єктами однієї організаційної структури.

У сучасних умовах інформаційної безпеки виявлення відкритих сервісів та уразливих мережевих компонентів є одним з найважливіших етапів як для тестування на проникнення, так і для побудови кіберзахисних стратегій. Пошукові системи, що спеціалізуються на індексації мережевої телеметрії - зокрема Shodan, Censys, FOFA та crt.sh - надають можливість отримувати розширені дані про інфраструктуру організацій, використовуючи вже зібрану інформацію без необхідності активного сканування з боку дослідника.

Одним із найвідоміших інструментів у цій галузі є Shodan - пошукова система, яка індексує інформацію про відкриті порти, версії сервісів, операційні системи та інші характеристики мережевих пристроїв на основі аналізу банерів.

Перевагою Shodan є також відображення відомих уразливостей, пов’язаних із виявленими сервісами, зокрема на основі CVE-ідентифікаторів. Це дозволяє швидко оцінити рівень ризиків. Взаємодія з Shodan можлива через веб-інтерфейс, який підтримує як прості, так і складні запити з фільтрами. Прикладом є запит:

org:"Example Corp" country:"UA"

Цей запит шукає всі пристрої, які мають відкриті порти, і в чиїх банерах вказано, що вони належать до організації "Example Corp" та знаходяться в Україні. Корисно для виявлення всієї зовнішньої інфраструктури компанії в межах конкретної країни.

Censys - це сервіс, орієнтований на аналіз TLS/SSL-сертифікатів, криптографічних параметрів, відкритих портів та протоколів. Він дозволяє ідентифікувати не лише тип і версію сервісу, а й рівень його криптографічної стійкості. Censys особливо корисний при оцінці застарілих конфігурацій або слабких алгоритмів шифрування.

Платформа надає зручний веб-інтерфейс, де можна виконувати запити за різними атрибутами, наприклад запит:

services.http.response.headers.server: "Apache/2.4"

Цей запит знаходить всі хости, у відповідях яких сервер HTTP повертає заголовок Server: Apache/2.4. Дає змогу визначити, які системи працюють на конкретній версії веб-сервера - це важливо для оцінки уразливостей.

FOFA - потужна китайська платформа для OSINT-досліджень, яка охоплює класичні сервери, IoT-пристрої, специфічні конфігурації вебзастосунків та банери. FOFA підтримує багаторівневу фільтрацію, наприклад:

app="nginx"

ip="192.168.1.1" && port="80"

Спочатку перший рядок шукає всі пристрої з веб-сервером nginx, незалежно від версії. Другий - фільтрує за конкретною IP-адресою та портом.

Саме така деталізація допомагає швидко знаходити цільові пристрої з відкритими службами.

crt.sh - це онлайн-сервіс, що здійснює пошук у загальнодоступних журналах Certificate Transparency. Завдяки цьому можна виявити домени та піддомени, які фігурували в історії SSL/TLS-сертифікатів, навіть якщо ці ресурси наразі неактивні або захищені фаєрволами.Для базового використання достатньо перейти за адресою:

https://crt.sh/?q=example.com

Дає список усіх SSL/TLS-сертифікатів, які колись видавалися для домену example.com, включно з піддоменами. Навіть якщо піддомен зараз неактивний або заборонений фаєрволом - він все одно буде видно у списку. Це дозволяє виявити внутрішні чи тестові ресурси компанії.

Важливо зазначити, що усі описані сервіси підтримують API для автоматизації процесів збору та обробки даних, однак в межах даної дипломної роботи інтеграція з API не реалізовувалась. Метою даного підрозділу є представлення методологічної цінності інструментів та демонстрація їхнього практичного значення для побудови OSINT-профілю організації.

Методика Google Dorking на прикладному рівні перетворюється з інструмента ручного пошуку в повноцінну платформу для інтелектуального цифрового профілювання, що дозволяє виявити несанкціоновані витоки, службові ресурси, конфігураційні помилки або дані внутрішньої структури організації.

На практиці оператори комбінуються в розширені запити для отримання конкретних типів файлів, прихованих директорій або адміністративних панелей. Приклади типових конструкцій:

site:example.com intitle:"index of" "backup"

site:example.com filetype:sql

site:example.com ext:env | ext:ini | ext:bak

У цих запитах оператор site: обмежує результати до вказаного домену, intitle: дозволяє шукати сторінки з конкретними словами в заголовку

(наприклад, "index of" часто свідчить про відкриту директорію), filetype: і ext: використовуються для виявлення файлів з певним розширенням, таких як .sql, .env, .ini або .bak, що можуть містити конфіденційні дані - резервні копії, дампи баз даних, налаштування доступу або середовищні змінні з API-ключами. Такі запити часто виявляють інформацію, яка випадково опинилася у відкритому доступі внаслідок помилок конфігурації. Також пошук можна масштабувати за допомогою Dorks-платформ на зразок ExploitDB Google Hacking Database, яка містить понад тисячу шаблонів запитів, що асоціюються з відомими витоками

Для напівавтоматизованого збору результатів використовуються Python-скрипти на базі бібліотеки googlesearch-python, які дозволяють сформувати масив URL-адрес за заданими дорками:

```
from googlesearch import search
for url in search('site:example.com filetype:pdf confidential', num_results=10):
print(url)
```

Результати можуть бути перевірені вручну або оброблені парсерами для витягування конкретної інформації. Наприклад, регулярні вирази дозволяють знаходити API-ключі, JWT-токени або паролі в знайдених документах.

Оцінка результатів Google Dorking дає змогу побічно виявити низку слабких місць у безпеці організації, зокрема наявність надлишкових привілеїв на публічних вебсерверах, що може призвести до несанкціонованого доступу до внутрішніх ресурсів. Крім того, за допомогою дорків можна виявити недбалість у розгортанні та тестуванні вебдодатків, що може свідчити про неналежний контроль за конфігураціями та даними, доступними в інтернеті. Виявлення відсутності правил індексації, таких як неправильні налаштування в файлі robots.txt або відсутність мета-тегів noindex, дозволяє зробити висновки про недостатній контроль за конфіденційністю вебресурсів. Часті запити, що націлені на виявлення помилок у конфігураціях, можуть вказати на невикористання WAF або сканерів DLP, що підвищує ймовірність витоку або компрометації чутливої інформації через слабкі місця в захисті вебсервісів.

2.2 Застосування активних технік для аналізу цільової системи

У цьому підрозділі буде розглядено методи та інструменти, що використовуються для активного збору інформації про інфраструктуру організації. Основна увага приділятиметься технікам сканування мереж, виявленню відкритих портів, визначенню версій служб і операційних систем, а також методам взаємодії з цільовими системами з метою отримання розширених відомостей. Також буде детально описано підходи до ідентифікації вразливостей на основі отриманих даних, що в подальшому використовується для проведення аналізу безпеки та розробки заходів з кіберзахисту.

2.2.1 Сканування відкритих портів та сервісів

Як вже зазначалось, сканування портів займає особливе місце, оскільки саме цей метод дозволяє отримати розширену інформацію про конфігурацію мережевих вузлів, доступні сервіси, їх версії, а також потенційні вразливості. Порт — це логічна точка завершення з'єднання, через яку програми обмінюються даними в мережі. Кожен порт асоціюється з певним мережевим сервісом (наприклад, HTTP, SSH або FTP), і знання про відкриті порти дає змогу зрозуміти, які саме служби працюють на хості. Основною метою сканування є визначення стану портів на віддаленій системі, що може бути класифіковано як відкриті, закриті або відфільтровані порти. Виявлення таких портів дозволяє зрозуміти, які сервіси функціонують на цільовому вузлі, які версії програмного забезпечення використовуються та наскільки вони вразливі до відомих атак. Процес сканування реалізується через надсилання специфічно сформованих мережевих запитів до певних діапазонів портів із подальшим аналізом відповідей від цільового хоста. У залежності від типу протоколу, що використовується, розрізняють сканування TCP та UDP-портів, кожне з яких має свої особливості та вимоги до реалізації.

TCP-сканування ґрунтується на можливостях протоколу Transmission Control Protocol, який забезпечує надійну передачу даних між вузлами мережі. Для дослідження відкритих TCP-портів найчастіше застосовуються такі методи, як TCP Connect Scan та SYN Scan, що відрізняються за рівнем взаємодії з цільовим сервером.

TCP Connect Scan передбачає повне тристороннє рукостискання (three-way handshake) між ініціатором сканування та віддаленим вузлом. Зокрема, під час цього процесу на цільовий порт надсилається SYN-пакет, у разі доступності порту сервер відповідає SYN/ACK, після чого здійснюється відправлення ACK для завершення встановлення з'єднання. Цей метод є доволі інформативним, оскільки дозволяє точно визначити, чи порт відкритий, закритий або відфільтрований міжмережевим екраном. Водночас, він залишає сліди в логах сервера, що робить його легко виявленим IDS. На (рис. 2.1) Показана схема TCP сканування [19].

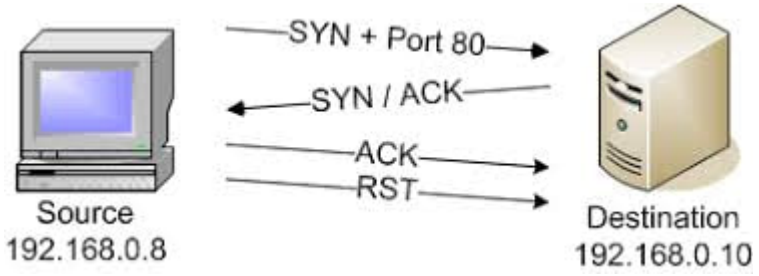


Рисунок 2.1 – Схема TCP сканування

SYN Scan, або "напіввідкрите сканування" (half-open scan), діє дещо інакше. У цьому випадку надсилається лише SYN-пакет, після чого, залежно від стану порту, повертається SYN/ACK (порт відкритий) або RST/ACK (порт закритий). Відмінність методу полягає в тому, що тристороннє рукостискання не завершується відправкою ACK, унаслідок чого з'єднання офіційно не вважається встановленим, а логування події на сервері, як правило, обмежується фіксацією невдалої спроби підключення. Це ускладнює виявлення

сканування засобами IDS, що підвищує ефективність цього підходу при проведенні прихованого рекону.

Класичним прикладом сервісів, що працюють поверх TCP, є SSH (порт 22), де забезпечення цілісності та надійності з'єднання є вкрай важливим для захисту даних під час віддаленого адміністрування систем.

Для реалізації зазначених методів сканування найчастіше використовується утиліта Nmap, яка дозволяє виконувати різноманітні типи запитів, налаштовувати діапазони портів, визначати версії сервісів та операційних систем. Команда для стандартного SYN-сканування виглядає наступним чином:

nmap -sS -p 1-1000 -T4 192.168.1.1

Вона ініціалізує швидке SYN-сканування перших 1000 портів за адресою 192.168.1.1, визначаючи їхній статус.

Окрім TCP-портів, важливою частиною процесу активного рекону є дослідження UDP-портів, що реалізується через метод UDP Scan. Протокол UDP суттєво відрізняється від TCP тим, що не забезпечує встановлення попереднього з'єднання перед передачею даних. Це робить його менш захищеним, але водночас більш швидким і зручним для сервісів, де важлива мінімальна затримка, наприклад, для DNS (порт 53).

У процесі UDP-сканування на цільовий порт надсилається спеціально сформований UDP-пакет. Якщо порт закритий, система, як правило, відповідає ICMP-повідомленням "Port Unreachable", що вказує на відсутність сервісу на цьому порту. Якщо ж порт відкритий, відповідь може бути відсутньою або сформованою залежно від налаштувань цільового сервісу. Така поведінка ускладнює визначення відкритих портів, оскільки відсутність відповіді не завжди означає доступність. Крім того, багато міжмережевих екранів можуть блокувати ICMP-пакети, що ще більше ускладнює процес сканування. Для підвищення надійності результатів сканування застосовують адаптивні таймаути та множинні спроби повторної відправки пакетів, щоб відокремити втрачені від мережі пакети від реальних “мовчазних” портів. Часто

використовують направлені “UDP ping” запити до відомих, які здатні спровокувати відповідь навіть на портах з мінімальною активністю.

На (рис. 2.2) схематично показано приклад UDP-сканування у випадках коли порт закритий та відкритий відповідно [20].

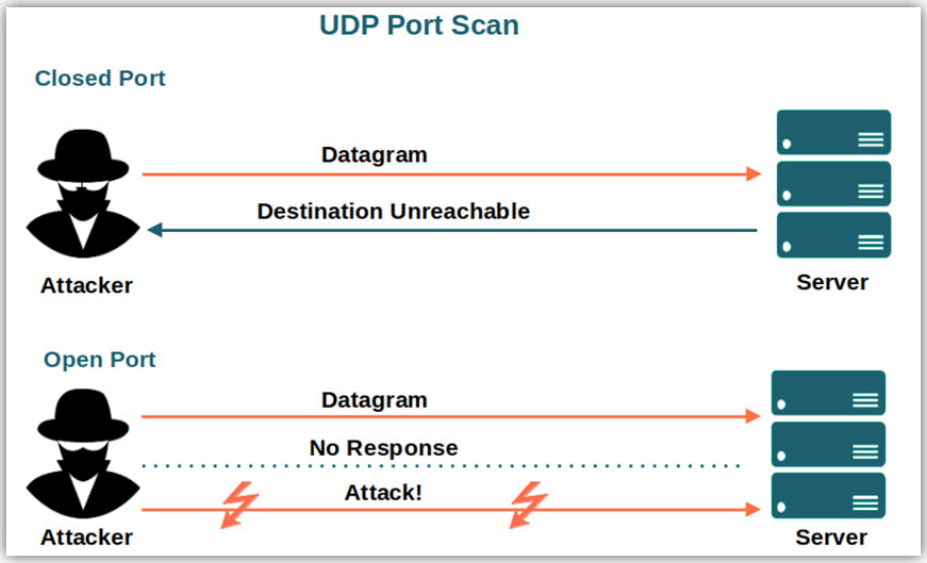


Рисунок 2.2 – Схематичне відображення UDP-сканування

Для реалізації UDP-сканування також використовується Nmap, який підтримує цей режим через опцію -sU. Приклад команди для сканування основних UDP-портів:

nmap -sU -p 53,123 192.168.1.1

Цей запит перевіряє доступність UDP-портів 53 (DNS) та 123 (NTP) на зазначеній IP-адресі.

Для більш детального аналізу Nmap надає можливість виконувати визначення версій сервісів, що працюють на виявлених портах, за допомогою опції -sV, а також дозволяє проводити OS Fingerprinting (-O), що дає змогу встановити тип операційної системи віддаленого хоста. Додатковою перевагою інструменту є підтримка NSE, який забезпечує можливість запуску скриптів для автоматизованого тестування вразливостей, аналізу мережевих протоколів та виявлення небезпечних конфігурацій. Наприклад, для виконання сканування з метою пошуку вразливостей використовується наступна команда:

nmap --script=vuln 192.168.1.1

Ця команда дозволяє Nmap здійснити серію запитів із використанням вбудованих скриптів, орієнтованих на виявлення найпоширеніших вразливостей, що суттєво підвищує рівень інформативності дослідження.

На відміну від Nmap, утиліта Masscan орієнтована на високошвидкісне сканування великих IP-діапазонів і здатна обробляти до 10 мільйонів портів за секунду. Ця продуктивність досягається завдяки низькорівневому доступу до мережевих інтерфейсів і використанню асинхронних запитів, що суттєво знижує затримки при обміні пакетами. Архітектура Masscan, по суті, є реалізацією модифікованого SYN-сканування, аналогічного тому, що використовується в Nmap, однак оптимізованого для паралельного виконання запитів на рівні ядра операційної системи. Втім, висока швидкість сканування супроводжується деякими обмеженнями: Masscan не підтримує NSE та інші розширені функції аналізу, зосереджуючи увагу винятково на виявленні відкритих портів. Команда для масового сканування всіх портів на вказаній IP-адресі виглядає наступним чином:

masscan -p1-65535 192.168.1.1 --rate=10000

Цей запит здійснює швидкісне сканування із зазначеною швидкістю в 10 тисяч пакетів на секунду, що дозволяє максимально швидко отримати карту відкритих портів на цільовому хості. Використання Masscan є доцільним на попередніх етапах розвідки, коли основною метою є швидке охоплення значних мережевих сегментів. Після виявлення активних портів, подальший аналіз конфігурації сервісів та пошук вразливостей може бути делегований Nmap, який надає розширені можливості для глибокого дослідження. Masscan також підтримує різні формати виводу (XML, JSON, binary), що дозволяє легко інтегрувати його результати в автоматизовані конвеєри обробки даних або власні скрипти для подальшого аналізу. За потреби можна використовувати опції фільтрації та виключення діапазонів, а також настроювати час очікування для оптимізації взаємодії з повільнішими мережами.

2.2.2 Визначення субдоменів методом брутфорсу

У процесі кіберрозвідки виявлення додаткових субдоменів цільового ресурсу є важливим етапом, тому що він дозволяє значно розширити обсяг інформації про інфраструктуру організації. Методика брутфорсу субдоменів ґрунтується на ідеї систематичного перебору можливих піддоменів з подальшою перевіркою їхньої доступності шляхом надсилання запитів до DNS-серверів. Такий підхід дозволяє ідентифікувати приховані або маловідомі елементи мережевої інфраструктури, які можуть містити критично важливі сервіси або інформаційні ресурси, що зазвичай залишаються поза увагою під час пасивного збору даних. В основі реалізації цього методу лежить формування словника, що включає найбільш вірогідні назви піддоменів, які часто використовуються в корпоративних та загальнодоступних мережах. Зазвичай до такого словника входять стандартні імена на кшталт www, mail, ftp, admin, portal, dev, test та інші, що відображають класичну структуру більшості корпоративних мереж. Використання розширених словників, сформованих на основі попереднього досвіду або витоків даних - збільшує ймовірність виявлення маловідомих піддоменів, що в свою чергу, може відкрити доступ до специфічних частин внутрішньої інфраструктури. Після підготовки словника здійснюється послідовне надсилання DNS-запитів до цільового домену для кожного з потенційних піддоменів. Відповідь сервера визначає, чи існує запис у DNS, що вказує на відповідний ресурс. Якщо сервер повертає валідну IP-адресу, це означає, що піддомен активний та функціонує як окремий вузол у межах корпоративної мережі. У випадку відсутності відповіді або отримання помилки типу "NXDOMAIN", можна стверджувати, що даний піддомен недоступний або не існує.

Для автоматизації процесу брутфорсу субдоменів розроблено низку спеціалізованих інструментів, серед яких особливе місце займають Gobuster, Subfinder та Sublist3r. Кожен з них має свої особливості реалізації, що визначають ефективність та швидкість сканування.

Gobuster - це інструмент, що орієнтований на швидкий та паралельний перебір піддоменів з використанням словників. Його основною перевагою є швидкість обробки DNS-запитів, оскільки застосовується механізм паралельного опитування серверів, що суттєво прискорює процес у порівнянні зі звичайними сканерами [21]. Для виконання стандартного брутфорсу субдоменів за допомогою Gobuster використовується наступна команда:

gobuster dns -d example.com -w subdomains.txt

У цьому випадку параметр -d вказує цільовий домен, тоді як -w визначає шлях до файлу словника, з якого беруться потенційні імена субдоменів. Інструмент відправляє запити до DNS-серверів і, за наявності позитивної відповіді - виводить список знайдених піддоменів.

Іншим поширеним інструментом для брутфорсу є Subfinder - легковаговий, але водночас потужний засіб для виявлення субдоменів, який підтримує інтеграцію з великим спектром зовнішніх API, таких як VirusTotal, crt.sh, ThreatCrowd та інші. Це дозволяє значно розширити область пошуку за рахунок звернень до зовнішніх баз даних [22]. Стандартний запит на виявлення субдоменів виглядає так:

subfinder -d example.com

На відміну від Gobuster, який зосереджується на переборі словників, Subfinder активно використовує методи пасивного рекону, що знижує рівень помітності сканування. Це є особливо корисним у випадках, коли важливо залишитися непоміченим для IDS.

Окрім зазначених, важливим інструментом є Sublist3r, який поєднує в собі методи брутфорсу та пасивного збору даних шляхом звернення до пошукових систем (Google, Bing, Yahoo), а також спеціалізованих платформ для відстеження доменних імен (Netcraft, DNSdumpster). Основна перевага цього рішення полягає в можливості залучення додаткових джерел інформації, що підвищує ймовірність виявлення рідкісних або нестандартних піддоменів. Приклад запуску виглядає наступним чином:

sublist3r -d example.com

2.2.3 Виявлення директорій шляхом словникових атак

У межах дослідження веборієнтованих ресурсів одним із ключових методів активного рекону є брутфорс директорій, що спрямовані на виявлення прихованих або маловідомих шляхів у файловій структурі вебсервера. Ці методи дозволяють ідентифікувати панелі адміністрування, тестові сторінки, резервні копії файлів або інші закриті ресурси, доступ до яких потенційно відкриває нові вектори атаки. Для досягнення цієї мети застосовуються різноманітні інструменти, що реалізують автоматизований перебір шляхів із використанням попередньо сформованих словників. Брутфорс директорій передбачає послідовне надсилання HTTP-запитів до серверу з метою перевірки наявності певних шляхів, визначених у словнику. Методика ґрунтується на використанні великої кількості типових і потенційно вразливих шляхів, що зазвичай присутні у вебдодатках або є залишками після тестування. Це можуть бути директорії на кшталт /admin/, /backup/, /test/, /dev/ та інші. Аналіз відповіді сервера дозволяє визначити, чи існує зазначена директорія, а статус-коди, що повертаються (див. Додаток А), вказують на доступність, обмеження доступу або відсутність ресурсу відповідно. Для реалізації брутфорсу директорій використовуються утиліти, що оптимізують процес перебору шляхів і дозволяють здійснювати його в паралельних потоках, підвищуючи ефективність сканування. Найбільш поширеними з них є DirBuster, Dirsearch та OKAdminFinder.

DirBuster - один із класичних інструментів для брутфорс-сканування директорій, що реалізує багатоетапний перебір шляхів з урахуванням вкладених структур. Основною його перевагою є можливість одночасного пошуку як директорій, так і окремих файлів, що можуть бути прихованими від стандартного індексування. Паралельне опитування дозволяє суттєво знизити час сканування великих обсягів даних.

Dirsearch - це легковаговий, але надзвичайно ефективний інструмент, що забезпечує швидкий перебір директорій завдяки оптимізованому механізму

HTTP-запитів. Підтримка технологій багатопотоковості дозволяє одночасно здійснювати велику кількість запитів, що підвищує продуктивність процесу. Приклад команди для брутфорсу директорій за допомогою Dirsearch виглядає наступним чином:

dirsearch -u http://example.com -e php,html -w wordlist.txt

Тут параметр -u визначає цільову URL-адресу, -e вказує розширення файлів для пошуку, а -w вказує шлях до словника.

OKAdminFinder - спеціалізована утиліта, орієнтована на пошук адміністративних панелей у вебдодатках. Її робота ґрунтується на використанні спеціально сформованих словників, що містять типові адреси адмін-панелей, характерні для популярних CMS. Інструмент здійснює перебір адрес із фіксацією отриманих статус-кодів, що дозволяє виявити адміністративні інтерфейси навіть у випадках, коли вони приховані від індексації пошуковими системами.

2.2.4 Тестування цільових ресурсів методом фазингу

Фазинг директорій є методикою активного рекону, спрямованою на виявлення прихованих або невідомих раніше ресурсів у файловій структурі вебсервера. На відміну від брутфорсу, який ґрунтується на переборі попередньо визначених шляхів зі словника, фазинг використовує більш динамічний підхід, що включає генерування HTTP-запитів із варіативними параметрами. Це дозволяє знаходити не лише відомі, але й випадково приховані або непередбачувані директорії, доступ до яких може надавати розширені можливості для атаки або подальшого сканування.

Процес фазингу директорій передбачає автоматизоване надсилання HTTP-запитів до вебсервера з використанням змінних шляхів у структурі URL. Суть підходу полягає у підстановці різних значень в URI-запити, що формуються на основі шаблону. Наприклад, типовий запит може виглядати наступним чином:

```
GET /FUZZ HTTP/1.1
Host: example.com
User-Agent: Mozilla/5.0
Accept: */*
```

Тут змінна FUZZ слугує маркером, що під час фазингу замінюється на різноманітні шляхи зі словника, або генерується автоматично на основі визначених правил. У відповідь вебсервер повертає статус-коди, що дозволяє досліднику визначити наявність або відсутність відповідної директорії. Для реалізації цієї методики застосовуються спеціалізовані утиліти, серед яких варто виділити FFUF та Katana.

FFUF є інструментом для швидкісного фазингу директорій, що підтримує багатопотоковість та забезпечує високий рівень продуктивності під час сканування великих ділянок файлової структури вебсервера. Його основною перевагою є можливість динамічного підбору шляхів шляхом підстановки значень у шаблони URL-запитів. FFUF дозволяє визначати різні типи запитів, вказувати заголовки, використовувати проксі для маскування трафіку та налаштовувати рівень паралелізації запитів. Приклад виконання базового фазингу директорій:

```
ffuf -u http://example.com/FUZZ -w directories.txt
```

У цій команді використовується параметр -u для визначення цільової URL-адресу для фазингу; FUZZ — спеціальний маркер, що замінюється на значення зі словника; -w directories.txt — вказує шлях до файлу зі списком типових директорій.

FFUF підтримує також фазинг параметрів URL, що дозволяє тестувати змінні запити. Наприклад:

```
ffuf -u http://example.com/page.php?id=FUZZ -w params.txt
```

Ця команда перевіряє параметр id на можливі приховані значення, підставляючи їх зі словника params.txt. Аналіз статус-кодів, отриманих у відповідь, надає змогу ідентифікувати, чи є відповідний ресурс доступним.

Окрім цього, FFUF дозволяє здійснювати багатопотокове сканування, що значно підвищує швидкість обробки великих обсягів запитів. Наприклад, за допомогою ключа -t можна встановити кількість одночасних потоків:

ffuf -u http://example.com/FUZZ -w directories.txt -t 100

Тут встановлено 100 паралельних запитів, що суттєво знижує час на повний обхід словника.

У свою чергу, katana є більш сучасним інструментом для фазингу, орієнтованим на асинхронне надсилання запитів. Це дозволяє мінімізувати затримки під час очікування відповідей від сервера, що суттєво підвищує ефективність сканування. Відмінною рисою Katana є її інтеграція з іншими засобами збору інформації, такими як Amass або Subfinder, що дозволяє у процесі фазингу динамічно оновлювати список піддоменів і директорій для подальшої перевірки. Приклад базового фазингу директорій з використанням Katana:

katana -u http://example.com -w paths.txt --rate-limit 500

У цій команді використовується параметр -u для визначення цільової URL-адресу для фазингу; -w paths.txt - словник із потенційними шляхами; --rate-limit 500 - обмеження швидкості до 500 запитів на секунду.

Katana також підтримує багатопотоковість і асинхронність у запитах, що дозволяє їй обходити великий перелік ресурсів значно швидше у порівнянні з класичними фазерами. Важливою особливістю є можливість зчитування результатів попередніх сканувань, що можуть бути використані для формування нових шаблонів фазингу.

2.3 Порівняльний аналіз інструментів для кіберрозвідки

Рано чи пізно постає запитання: який з наявних інструментів є найбільш ефективним для конкретного етапу розвідки або типу цілі? З огляду на різноманіття утиліт - від високошвидкісних сканерів портів до

вузькоспеціалізованих засобів аналізу субдоменів чи метаданих - вибір часто зводиться до компромісу між швидкістю, глибиною аналізу та рівнем маскування активності. Відсутність універсального рішення спонукає аналітика порівнювати інструменти за контекстом завдання, доступним ресурсам і технічним обмеженням цільового середовища. Власне, практична цінність проведення такого порівняльного аналізу полягає у можливості визначити оптимальний набір утиліт для різних етапів рекону, мінімізуючи час та обчислювальні ресурси. Зокрема, у межах автоматизованих систем кіберрозвідки, що орієнтовані на моніторинг відкритих сервісів, пошук піддоменів, виявлення вебтехнологій, а також фазинг директорій, коректний підбір інструментів дозволяє забезпечити максимальне покриття цільової площини атаки при мінімальних ризиках виявлення.

Для цього у таблиці нижче представлено детальний порівняльний аналіз одинадцяти ключових утиліт, що широко застосовуються в галузі кібербезпеки. Наведена інформація включає в себе основні можливості кожного інструмента, його цільове призначення, тип рекону а також переваги та недоліки (табл. 2.3). Така структура дозволяє не тільки досить швидко зорієнтуватися в можливостях кожної з утиліт, а й також раціонально сформувати подальшу стратегію тестування безпеки або моніторингу інфраструктури, базуючись на конкретних завданнях розвідки.

Таблиця 2.1

Порівняння можливостей інструментів рекону

Інструмент	*Ціль*	*Тип рекону*	*Можливості*
Subfinder	Пошук піддоменів шляхом збору даних з пасивних джерел	Пасивний	Збір даних із відкритих джерел (VirusTotal,

			Shodan, DNSdumpster та ін.)
Sublist3r	Пошук піддоменів через пасивні та активні джерела	Пасивний/ Активний	Виявлення піддоменів через пошукові системи, DNS-запити та OSINT

продовження табл. 2.1

Katana	Фазинг директорій та шляхів на вебсерверах	Активний	Асинхронний фазинг директорій та URL, підтримка sitemap, robots.txt
Dirsearch	Брутфорс директорій та прихованих ресурсів	Активний	Перебір директорій на основі словників, підтримка проксі та авторизації
DirBuster	Брутфорс директорій та файлів на вебсерверах	Активний	GUI-орієнтована утиліта для перебору шляхів та файлів
Okadminfinder	Пошук адміністративних панелей на вебресурсах	Активний	Визначення URL панелей адміністрування через перебір типових шляхів
Gobuster	Фазинг піддоменів та директорій	Активний	Високошвидкісний фазинг шляхів та піддоменів на основі словників
Nikto	Сканування вебсерверів на вразливості	Активний	Виявлення відомих вразливостей, помилок конфігурації та небезпечних налаштувань

WhatWeb	Ідентифікація вебтехнологій	Пасивний/ Активний	Визначення серверних технологій, CMS, версій ПЗ та плагінів

продовження табл. 2.1

Nmap	Сканування портів, визначення версій ПЗ та ОС	Активний	Різноманітні типи сканування (SYN, UDP, OS detection, скрипти NSE)
Masscan	Масове сканування відкритих портів	Активний	Високошвидкісне сканування (10 млн портів за секунду), ідеальне для великих діапазонів IP

Таблиця 2.2

Переваги та недоліки інструментів рекону

Інструмент	*Переваги*	*Недоліки*
Subfinder	Висока швидкість збору піддоменів завдяки паралельним запитам; мінімальне навантаження на цільовий сервер, відсутність ризику бути виявленим.	Залежність від актуальності даних у відкритих джерелах; обмежена ефективність у випадку прихованих піддоменів або ізольованих інфраструктур.
Sublist3r	Підтримка як пасивного та активного сканування; можливість виявлення піддоменів	Низька швидкість сканування в порівнянні з іншими інструментами; ризик бути

		заблокованим при активному скануванні великих діапазонів.

продовження табл. 2.2

Katana	Висока швидкість за рахунок асинхронної обробки; можливість обхідних маневрів через sitemap та robots.txt.	Вимагає налаштування для коректної роботи з деякими специфічними вебсерверами; потребує значних ресурсів при глибинному фазингу великих сайтів.
Dirsearch	Гнучкість налаштувань, підтримка протоколів HTTPS та HTTP; можливість роботи через проксі-сервер.	Високе навантаження на сервер під час інтенсивного сканування; обмежена ефективність проти сучасних WAF-рішень.
DirBuster	Зручний графічний інтерфейс, що полегшує конфігурацію; підтримка різних словників та глибокого перебору.	Порівняно низька швидкість у порівнянні з консольними аналогами; потребує значних ресурсів при глибокому скануванні.
Okadminfinder	Швидкий пошук відомих адміністративних панелей; проста інтеграція у	Лише типовий перебір шляхів; неефективний проти нестандартних URL та захищених панелей з багатофакторною автентифікацією.

	Bash-скрипти для автоматизації.	

продовження табл. 2.2

Gobuster	Максимальна продуктивність під час фазингу за рахунок багатопотоковості; можливість сканування як DNS, так і HTTP.	Високе навантаження на ціль при скануванні великих словників; ризик блокування за рахунок аномальної кількості запитів.
Nikto	Широке покриття відомих вразливостей та помилок конфігурації; підтримка протоколів HTTP та HTTPS.	Високий рівень детектування засобами захисту; порівняно довгий час сканування великих сайтів.
WhatWeb	Висока швидкість сканування; можливість розширення функціоналу за допомогою плагінів.	Не завжди коректно ідентифікує нові або кастомізовані технології; обмежена підтримка складних вебзастосунків.
Nmap	Висока точність визначення ОС та версій ПЗ; підтримка скриптів NSE для глибокого аналізу.	Значне навантаження на ціль при інтенсивному скануванні; висока ймовірність детектування фаєрволами та IDS.
Masscan	Найшвидший інструмент у своєму класі; можливість сканування величезних діапазонів за лічені хвилини.	Низька точність у визначенні версій сервісів; потребує додаткового аналізу після виявлення відкритих портів.

2.4 Відомі кіберінциденти, де розвідка зіграла ключову роль

У 2020 році було виявлено цілеспрямовану атаку на інфраструктуру компанії SolarWinds, у ході якої зловмисники скомпрометували процес CI/CD платформи Orion. Попередній етап розвідки включав глибокий аналіз архітектури програмного середовища SolarWinds, ідентифікацію модуля, що регулярно оновлюється клієнтами, а також визначення механізмів перевірки цілісності. На основі зібраних даних було впроваджено бекдор, який мав вигляд легітимної .NET DLL (SolarWinds.Orion.Core.BusinessLayer.dll), підписаної цифровим сертифікатом компанії. Після розповсюдження інфікованого оновлення серед понад 18 000 клієнтів, подальший процес рекону проводився вже зсередини заражених систем: зловмисники використовували DNS-тунелювання для зворотного зв’язку з C2-інфраструктурою, виконували доменний аналіз за допомогою команд типу whoami, nltest /dclist - вивчали внутрішню Active Directory-структуру. Доступ до облікових даних було отримано через викрадення токенів та маніпуляції з SAML-токенами, що дало можливість для подальшого горизонтального переміщення у високозахищених середовищах, таких як ті, що належали урядовим установам США [23].

У листопаді 2020 року компанія Capcom зазнала однієї з найбільших атак у своїй історії, у результаті якої було викрадено понад 1 ТБ внутрішніх даних, зокрема особисту інформацію співробітників, документи з комерційною таємницею, фінансові звіти та вихідні коди ігор. Ключову роль у цій атаці відіграв попередній етап грамотно організованої кіберрозвідки, під час якого зловмисники здійснили глибокий аналіз зовнішньої інфраструктури компанії.

Проведення активного сканування відкритих портів, зокрема 443 (HTTPS) і 1194 (OpenVPN), дозволило виявити VPN-сервер, який залишався доступним з Інтернету внаслідок переходу компанії на віддалений режим роботи під час пандемії COVID-19. Зловмисники проаналізували тип і версію серверного ПЗ, що дало змогу точно ідентифікувати використання вразливого продукту Pulse

Secure VPN, який мав відому критичну уразливість CVE-2019-11510 — експлуатуючи яку, можливо було зчитувати довільні файли з файлової системи без проходження автентифікації. Отримавши у такий спосіб конфігураційні файли, кеш паролів та токени, атакувальники успішно автентифікувалися у внутрішню мережу компанії, де виконали додаткове мережеве зондування, аналіз доменної структури, виявлення контролерів Active Directory та облікових записів із підвищеними привілеями. Здобувши повний контроль над критичною частиною мережі, вони розгорнули вимагача Ragnar Locker, який здійснив шифрування систем і паралельне викрадення чутливих даних [24].

Цей інцидент чітко демонструє, що ефективно виконаний рекон - навіть без використання уразливостей нульового дня - може стати вирішальним етапом у багаторівневій цільовій атаці. Саме виявлення неправильно сконфігурованого VPN та експлуатація відомої уразливості стали прямим наслідком ретельного попереднього аналізу поверхні атаки компанії.

У 2021 році було виявлено кілька інцидентів витоку даних через некоректну конфігурацію S3-бакетів, які не були захищені політиками доступу (ACL або Bucket Policy) і залишались відкритими для загального доступу.

- Slickwraps: Відкритий бакет містив файли з особистими фотографіями клієнтів, email-адресами, деталями замовлень. Доступ до бакету було отримано через пошук по bucket enumeration із використанням таких сервісів, як GreyhatWarfare [25].
- Decathlon: Понад 123 мільйони записів журналів тестових CI/CD-середовищ, включаючи токени доступу, логіни, хешовані паролі, були знайдені у відкритому S3-бакеті. Пошук здійснювався за ключовими словами, пов’язаними з назвою компанії, з використанням API AWS SDK.
- FedEx (Bongo International): У результаті витоку з бакету стороннього підрядника, що обробляв документи для митного оформлення, було оприлюднено понад 119 тисяч PDF-документів, включаючи скани паспортів і водійських посвідчень. Доступ був отриманий через неправильно реалізований CORS-запит, що дозволив виконувати запити без автентифікації.

У всіх випадках рекон на основі брутфорсу імен бакетів, використання відкритих репозиторіїв і пасивного моніторингу DNS-записів дозволили зловмисникам виявити неправильні конфігурації, які відкривали прямий доступ до конфіденційних даних без необхідності використання уразливостей у програмному забезпеченні.

2.5 Автоматизовані комплекси для проведення кіберрозвідки

У професійному середовищі фахівців з кібербезпеки дедалі гостріше постає потреба в автоматизації процесів первинної розвідки, адже зростаючі обсяги даних, складність мережевих структур і обмеження часу вимагають ефективних, масштабованих рішень. Саме тому автоматизовані інструменти кіберрозвідки стають невід'ємною частиною сучасного підходу до виявлення потенційних вразливостей. Водночас, незважаючи на безумовні переваги автоматизації, важливо усвідомлювати й відмінності між автоматизованими та ручними методами збору даних. Кожен із цих підходів має свої переваги та недоліки залежно від контексту застосування, цілей дослідження та технічних умов. З огляду на це, доцільно здійснити порівняння обох підходів, щоб чітко окреслити їх можливості та сфери найефективнішого використання.

Хоча і ручні методи проведення рекону традиційно вважаються більш гнучкими та адаптивними - вони демонструють певні серйозні обмеження, які стають надалі критичнішими в умовах зростаючої складності корпоративних IT-систем. Для більш ефективної роботи аналітика за таким підходом потрібна не лише висока технічна обізнаність, а й здатність оперувати багатьма утилітами та інтерпретувати гетерогенні дані в реальному часі. Водночас значна кількість однотипних дій, як-от запити до DNS, сканування портів, аналіз HTTP-відповідей, призводить до великих витрат часу та людських ресурсів, що в свою чергу негативно впливає на оперативність виявлення критичних уразливостей. До того ж ризики, пов'язані з людським фактором, такі як

помилки у синтаксисі команд, неправильне тлумачення результатів сканування чи випадкове пропущення важливих ознак вразливості залишаються суттєвими навіть у разі високої кваліфікації фахівця. На противагу цьому, автоматизовані інструменти кіберрозвідки дозволяють значно підвищити ефективність, точність та масштабованість відповідних операцій, формуючи передумови для більш системного та повторюваного підходу. Основною ідеєю таких інструментів є можливість автоматичного виконання послідовностей дій, які імітують типову логіку роботи аналітика, але без прямої участі людини в кожному з етапів. Це забезпечується через скрипти, API або спеціалізовані програмні модулі, які, з одного боку, уніфікують процес розвідки, а з іншого - дозволяють інтегрувати його в більш широкі системи неперервного аналізу.

Серед актуальних рішень, які вже зарекомендували себе як дієві засоби автоматизації етапу рекону, можна виділити як комерційні, так і відкриті інструменти. Наприклад, Acunetix, що позиціонується як повнофункціональний сканер вебзастосунків, підтримує глибокий аналіз HTML-структури, виявлення понад семи тисяч відомих уразливостей та інтеграцію із системами управління ризиками. Важливо, що цей інструмент також автоматично визначає використовувані CMS, JavaScript-бібліотеки та типи бекенд-серверів, що напряму впливає на якість вхідних даних для подальшого тестування [26].

Іншим поширеним рішенням є Burp Suite Pro, що відзначається великою гнучкістю та орієнтованістю на інтерактивний аналіз, включаючи можливість часткового ручного втручання. Його функціональність включає проксі-модуль, активний сканер, генератор запитів, а також розширюваність через модулі на базі Burp Extender. Завдяки такій архітектурі Burp дозволяє комбінувати автоматичний та ручний підходи, проте все ще потребує значних зусиль з боку користувача для повноцінного налаштування кожного етапу.

Серед open-source рішень особливу увагу слід приділити утиліті Nuclei, яка реалізує концепцію шаблонного сканування та дозволяє з високою швидкістю перевіряти тисячі доменів за типовими сценаріями уразливостей. Завдяки відкритому коду та регулярному оновленню шаблонів, Nuclei зручний у

використанні як базовий елемент власної системи автоматизації. У поєднанні з такими інструментами, як httpx, Katana, Naabu або Subfinder, він формує ефективне середовище для швидкого та масштабованого аналізу публічно доступних ресурсів. Користувачі можуть створювати власні шаблони у форматі YAML, що дозволяє адаптувати сканування під специфічні вимоги проєкту та оперативно реагувати на нові типи загроз. Інтеграція Nuclei в CI/CD-пайплайни забезпечує автоматичне оновлення інформації про вразливості на етапі розгортання, а вбудована підтримка JSON- і CSV-експорту спрощує подальший аналіз результатів та генерацію звітів [27].

Зважаючи на переваги та обмеження описаних рішень, доцільним виглядає створення власного автоматизованого інструменту для проведення первинної розвідки. Поєднання кількох інструментів у єдиному скрипті дозволяє реалізувати мультипоточний підхід: паралельно запускати WhatWeb для збору технологічних даних, Nmap для виявлення відкритих портів, Dirsearch для пошуку каталогів та Katana для фазингового аналізу JavaScript-ресурсів. Зокрема, розробка Bash-скрипта, що об'єднує у єдиний ланцюг такі утиліти, як Nmap, WhatWeb, Dirsearch та Katana, дозволяє сформувати гнучкий, прозорий та контрольований процес сканування. На відміну від деяких комерційних рішень, такий підхід усуває залежність від сторонніх платформ, забезпечує можливість адаптації до конкретних технічних умов об'єкта дослідження. Таким чином, створення власного Bash-інструменту може розглядатись як практично обґрунтоване рішення, яке логічно поєднує ефективність автоматизації з гнучкістю аналітичного контролю. Детальний розгляд архітектури, логіки роботи та особливостей реалізації такого скрипта наведено в розділі 3.

2.6 Методи захисту від розвідки та зменшення площі атаки

Важливим компонентом захисту IT-інфраструктури стає не лише виявлення та нейтралізація спроб проникнення, а й активна протидія самій можливості збору таких даних. Інакше кажучи, організація має також забезпечити приховування критичних елементів, або ж навіть навмисно викривлювати доступну для зловмисника картину, тим самим ускладнюючи побудову достовірної моделі атаки.

Одним із базових технічних напрямків протидії рекону є виключення можливостей для пасивного збирання інформації з вебсайтів та прилеглих до них інфраструктурних елементів. Наприклад, заборона листингу директорій на вебсерверах (опція Options -Indexes для Apache або autoindex off для Nginx) запобігає випадковому чи навмисному перегляду вмісту публічних каталогів, які часто містять резервні копії, конфігураційні файли чи службові скрипти. Нерідко тестові версії сайтів, файли з розширенням .bak, .old або .zip залишаються у відкритому доступі, ігноруючи вимоги до інформаційної гігієни [28]. Для підвищення безпеки слід регулярно переглядати журнали доступу до цих файлів та автоматично видаляти застарілі або небажані записи.

Не менш поширеним є використання файлу robots.txt для заборони індексації чутливих ресурсів пошуковими системами. Однак застосування цього методу потребує обережності: парадоксально, але сама наявність такого файлу, і зокрема згадки про закриті URI у його вмісті, може стати орієнтиром для зловмисника, який використовує методи Google Dorking. Ефективніше буде не фокусуватися на масовому приховуванні інформації через robots.txt, а радше контролювати її появу в індексованому просторі на рівні CMS, серверних відповідей та метатегів.

У сфері конфігурації вебсерверного програмного забезпечення ключовими техніками протидії є вимкнення будь-яких сервісних банерів, які ідентифікують версію вебсервера (наприклад, рядок Server: Apache/2.4.51), а

також приховування або модифікація заголовків HTTP-відповідей, які вказують на технологічний стек. Крім того, модифікація типових сторінок помилок (наприклад, заміна стандартної 404 Not Found на кастомізовані сторінки без діагностичної інформації) дозволяє приховати структуру сервісу та запобігти отриманню індикаторів щодо файлової системи чи логіки обробки запитів.

Комплексним засобом протидії як ручній, так і автоматизованій розвідці є WAF - спеціалізовані фільтрувальні системи, що аналізують HTTP/HTTPS-трафік і дозволяють блокувати підозрілу або аномальну активність на ранніх стадіях. Популярні WAF-рішення, такі як Cloudflare, ModSecurity або F5 Advanced WAF, забезпечують гнучкі механізми фільтрації, включаючи виявлення шаблонних запитів від сканерів, обмеження частоти доступів, аналіз поведінкових патернів та автоматичну реакцію на типові техніки типу fingerprinting. Варто розуміти, що більшість WAF працюють лише на рівні прикладного трафіку, і тому вони малоефективні проти пасивного рекону.

Поглиблена протидія кіберрозвідці також передбачає впровадження механізмів технологічного обману - наприклад, honeypots, які імітують уразливі системи, або honeytokens, що дозволяють відстежити спроби несанкціонованого доступу до підставних об'єктів. Вони не лише виявляють спроби сканування, а й створюють хибні вектори атаки, що можуть обманути навіть досвідченого противника. Такі системи ефективно інтегруються з SIEM-платформами для подальшої кореляції подій та формування звітів.

Однак необхідно розуміти, що ефективна стратегія протидії кіберрозвідці не може обмежуватися лише одним із зазначених напрямів. Вона повинна бути багаторівневою системою взаємопов'язаних заходів, які не лише зменшують обсяг доступної для збору інформації, а й спотворюють або підмінюють її таким чином, щоб атака на основі отриманих даних була технічно недоцільною або надто ризикованою для зловмисника. Водночас адаптивність цієї стратегії до змін у середовищі загроз повинна бути закладена на етапі її проектування,

оскільки відсутність оновлення підходів призводить до втрати їх ефективності навіть у середньостроковій перспективі.

Висновки за розділом 2

У другому розділі було здійснено детальний аналіз механізмів та методів кіберрозвідки як критично важливого етапу в дослідженні інформаційної безпеки. Послідовно розглянуто прикладну реалізацію пасивного та активного рекону, окрему увагу приділено технікам збору інформації про домени, піддомени, DNS-записи, структуру вебресурсів, відкриті порти, технології вебзастосунків і метадані, доступні в публічному просторі. Здійснено практичний огляд інструментів, що використовуються на кожному з етапів - з акцентом на їхні технічні особливості, функціональні можливості, продуктивність і ступінь виявлення з боку систем захисту.

Показано, що кіберрозвідка охоплює не лише технічний аналіз, але й системний підхід до виявлення цифрового сліду організації, формування attack surface і виявлення точок потенційної компрометації. Особливо підкреслено значення автоматизації розвідки - як засобу підвищення ефективності, масштабованості та повторюваності аналізу. Було обґрунтовано доцільність створення власного Bash-скрипта для об’єднання типових інструментів у єдиний ланцюг, що дозволяє адаптувати процес до специфіки цільового середовища, зменшити залежність від сторонніх сервісів і підвищити контроль над результатами.

Розділ також продемонстрував значущість розвідки на прикладах відомих інцидентів (SolarWinds, Norsk Hydro, S3-інциденти), де саме збір інформації про поверхню атаки став ключовим чинником успішного проникнення. Окремо висвітлено стратегії протидії рекону, зокрема методи приховування інфраструктурних ознак, впровадження WAF, контроль над метаданими, виключення індексації та застосування honeytoken-технік.

Другий розділ створює практичну основу для реалізації автоматизованої системи кіберрозвідки, яка буде представлена у наступному розділі дипломної роботи.

РОЗДІЛ 3
ПРАКТИЧНА РЕАЛІЗАЦІЯ МЕТОДІВ КІБЕРРОЗВІДКИ

3.1 Розробка автоматизованого Bash-скрипту

У процесі створення інструменту для автоматизованої технічної розвідки інфраструктури організації було здійснено критичний аналіз можливих засобів реалізації сценаріїв збору інформації, що працюють на рівні командного рядка. Основними критеріями оцінювання виступали сумісність із UNIX-подібними операційними системами, низький поріг входження в середовище розробки, мінімальні апаратні вимоги, ефективність при взаємодії з мережею та файловою системою, а також здатність до інтеграції з утилітами командного рядка - такими як nmap, subfinder, dirsearch, katana, whatweb тощо. Ураховуючи вищенаведені параметри, як базове середовище розробки було обґрунтовано обрано Bash - інтерпретатор командного рядка, що зазвичай попередньо встановлений у більшості сучасних дистрибутивів Linux, включно з контейнеризованими або мінімалістичними середовищами на кшталт Alpine чи Debian Slim.

Серед ключових переваг, які роблять Bash релевантним інструментом у даному контексті, варто, перш за все, відзначити його повну сумісність із переважною більшістю систем класу *nix, що, своєю чергою, дозволяє забезпечити універсальність розробленого скрипту без необхідності адаптації до специфіки окремих платформ. Крім того, Bash-сценарії вирізняються вкрай низькими вимогами до ресурсів, і це важливо при роботі у віддалених або

обмежених за обчислювальними потужностями середовищах як от VPS, Raspberry Pi або хмарних функціях.

Ще одним суттєвим фактором є можливість максимально гнучкої інтеграції з утилітами нижнього рівня, такими як curl або grep. Це забезпечує високу швидкість обробки потокових даних, як у локальному файловому просторі, так і під час взаємодії з мережею. Більш того, Bash-сценарії зберігають відкриту структуру - кожна команда є прозорою, а логіка виконання чітко прослідковується, що значно полегшує налагодження, аудит та подальшу модифікацію скрипту під індивідуальні потреби організації.

У ході обґрунтування доцільності реалізації recon-модуля саме на Bash було проведено порівняльний аналіз із аналогічними комерційними рішеннями, зокрема з Acunetix, Nuclei та Burp Suite Pro. Зазначені інструменти мають усталену репутацію у професійних колах, проте їх застосування у контексті повної автоматизації розвідки виявляється не завжди виправданим.

Хоч і Acunetix є одним з найпотужніших комерційних веб-сканерів - він має низку обмежень, які ускладнюють інтеграцію в автономні recon-сценарії. Його закрита архітектура та орієнтованість на графічний інтерфейс значно ускладнюють сценарне управління, особливо у headless-середовищах або під час CI/CD-запусків. Високі вимоги до ресурсів і залежність від Windows/GUI-платформ також обмежують можливість масштабованого та гнучкого розгортання.

Nuclei, попри відкриту архітектуру та зручну систему шаблонів, тісно інтегрований із рядом суміжних інструментів (зокрема httpx, dnsx, subfinder) і потребує певної кривої навчання щодо створення та кастомізації шаблонів. при масштабному скануванні великих пулів доменів відсутність вбудованої підтримки розподіленого виконання може стати вузьким місцем. З практичного боку, Nuclei не надає зручного механізму агрегації результатів у єдиний звіт без залучення зовнішніх скриптів, що в контексті автоматизації - є додатковим ускладненням.

Щодо Burp Suite Pro, то його функціонал орієнтований на ручний аналіз HTTP-трафіку, що, безперечно, дає змогу виявляти складні вразливості, проте водночас унеможливлює побудову повноцінного автоматизованого recon-ланцюжка. Навіть за умови використання API або кастомних розширень, застосування Burp залишається або обмеженим у безкоштовній версії, або пов’язаним з необхідністю придбання ліцензії.

На цьому тлі, Bash демонструє низку переваг щодо вищезгаданих альтернатив:

- повну автоматизованість процесу без необхідності втручання оператора на жодному етапі виконання;
- гнучку адаптацію до різнорідних середовищ та інфраструктур без обмежень за архітектурою чи ОС;
- відсутність потреби у комерційній ліцензії, а також зручну інтеграцію у хмарні або локальні пайплайни автоматизації.

3.2 Огляд функціоналу Bash-скрипта

Bash-скрипт (див. Додаток Б) виконує функцію автоматизованого збору інформації про веб-ресурс з використанням різних інструментів OSINT та сканування. Скрипт дозволяє проводити пошук піддоменів, виявлення прихованих директорій та адмін-панелей, фазинг URL-адрес, аналіз технологій вебсайту та сканування портів на наявність відомих вразливостей. Його особливістю є інтерактивність, а також підтримка переривання окремих підпроцесів без повної зупинки сценарію. Функціонал скрипта побудований на модульній основі, де кожен блок відповідає за окремий етап розвідки. Додатково було реалізовано автоматичну перевірку наявності необхідних утиліт та їх встановлення у разі відсутності.

3.2.1 Основні етапи роботи скрипта

Після ініціалізації скрипта відбувається запит на введення доменного імені цільового веб-ресурсу у форматі example.com. Унаслідок цього створюється відповідна директорія з назвою домену, що забезпечує логічне розмежування результатів кожної окремої сесії збору інформації. Такий підхід сприяє впорядкуванню даних і значно полегшує їх подальший аналіз. Логіка створення і переходу у відповідну папку реалізована у функції initialize_directory (domain).

Далі виконується процедура виявлення піддоменів, яка базується на застосуванні інструменту Subfinder. Отримані результати фіксуються у файл subdomains.txt, після чого вони трансформуються у формат URL, зручний для подальших етапів аналізу - у файл urls.txt. Варто підкреслити, що пасивний характер збору мінімізує ризик виявлення і фіксації досліджуваної активності з боку цілі. Цей процес інкапсульований у функції find_subdomains(domain), яка відповідає за запуск інструменту та подальшу конвертацію результатів.

На наступному етапі здійснюється ідентифікація доступних директорій веб-серверів за допомогою двох паралельних інструментів: Dirsearch та Dirb. Перший застосовує методику брутфорс-запитів, відфільтровуючи відповіді за HTTP-статусами (200, 301, 302, 403 тощо), що дає змогу виокремити актуальні ресурси. Другий інструмент функціонує за схожою логікою, проте відрізняється деталями реалізації обходу, що забезпечує додаткову глибину сканування. Результати кожного із засобів зберігаються у відповідних текстових файлах — dirsearch.txt та dirb.txt. Логіка запуску та збереження результатів цих інструментів реалізована у функціях run dirsearch (urls_file) та run dirb (urls_file) відповідно.

Паралельно із пошуком директорій виконується фазинг веб-застосунків, реалізований через інструмент Katana. Цей компонент автоматично обходить піддомени, зокрема аналізує динамічні елементи, у тому числі JavaScript-посилання, виявляючи таким чином приховані або незадокументовані ресурси, які часто є джерелом потенційних вразливостей. Результати

зберігаються у файлі katana.txt, що, в свою чергу, значно розширює розуміння структури цільового сайту. Ця логіка інкапсульована у функції run_katana (urls_file).

Подальший крок полягає у зборі метаданих і характеристик веб-сайтів, що здійснюється за допомогою WhatWeb — інструменту, який дозволяє ідентифікувати технологічний стек цільового ресурсу. При цьому аналіз проводиться з підвищеним рівнем агресивності (параметр -a 3), що дає змогу виявити CMS, тип вебсерверів, фреймворки JavaScript та інші компоненти. Результати цього етапу записуються у whatweb.txt. З отриманих даних додатково витягуються унікальні IP-адреси, які формують список для наступного глибокого сканування портів, що зберігається у файлі ip.txt. Цей процес організовано у функції run_whatweb(urls_file).

Завершальний етап реалізований через комплексне порт-сканування знайдених IP-адрес за допомогою утиліти Nmap. При цьому застосовується опція --script vuln, яка активує набір вбудованих NSE-скриптів, орієнтованих на виявлення відомих вразливостей на відкритих портах. Отримані результати акумулюються у файлі nmap.txt, формуючи підґрунтя для подальшого аудиту безпеки цільової інфраструктури. Логіка порт-сканування міститься у функції run_nmap (ip_file).

3.2.2 Автоматизоване створення звіту

На завершальному етапі виконання автоматизованої системи здійснюється генерація узагальненого звіту на основі результатів попередніх сканувань та виявлень. Для цього використовується окремий скрипт report.sh (див. Додаток В), логіка якого полягає у динамічному формуванні HTML-документа зі структурованим відображенням вмісту основних файлів результатів (див. Додаток Д).

Після запуску скрипта користувач вводить шлях до директорії, де зберігаються скан-файли, сформовані під час попередніх етапів. Скрипт

перевіряє наявність зазначеної директорії, після чого переходить до її обробки. У процесі створення звіту формується HTML-файл recon_report.html, який містить стилізовані блоки з інтерактивними заголовками для кожного з типів даних (піддомени, URL-адреси, IP, тощо). Це забезпечує зручну навігацію в межах одного документа та дозволяє швидко згортати чи розгортати відповідні розділи. Кожен лог-файл зчитується по черзі, після чого вміст додається у відповідний HTML-блок у вигляді тегу <pre>, що гарантує збереження форматування. У випадку, якщо файл відсутній або порожній, замість вмісту виводиться повідомлення про це. Для покращення безпеки скрипт також реалізує екранізацію спеціальних HTML-символів (таких як <, >, &) з метою запобігання потенційним вразливостям при відображенні вмісту. У звіт також інтегрована кнопка "Розгорнути всі", що дозволяє миттєво переглянути повний вміст усіх секцій, або навпаки - згорнути їх для зручнішого перегляду. Така інтерактивна реалізація забезпечує користувачу гнучкість у роботі з великими обсягами даних.

У результаті виконання скрипта формується єдиний HTML-документ, який може бути відкритий у будь-якому веб-браузері, що значно спрощує процес аналізу отриманої інформації, зберігає її у візуально структурованому вигляді та є зручним для подальшої передачі.

3.2.3 Обґрунтування вибору утиліт

При розробці системи автоматизованої кіберрозвідки найголовнішим критерієм вибору інструментів була ефективність в умовах обмеженого часу, точність результатів та зручність інтеграції в скриптову автоматизацію. Порівняльний аналіз утиліт дозволив визначити найбільш доцільні варіанти.

Під час розробки системи автоматизованої кіберрозвідки вибір утиліт здійснювався на основі їхньої ефективності, точності, швидкості та зручності інтеграції в скриптове середовище. Наприклад, для виявлення піддоменів було обрано Subfinder замість Amass, оскільки він забезпечує вищу швидкість при

пасивному зборі, простий у використанні, легше масштабується та не потребує складної конфігурації. На відміну від нього, Amass часто створює надлишковий трафік та виконує активне сканування, що може бути недоцільним у пасивному сценарії розвідки.

Для пошуку директорій застосовано Dirsearch, який перевершує Gobuster завдяки розширеній підтримці HTTP-параметрів - таких як заголовки, cookies, user-agent - і гнучкішій обробці нестандартних відповідей сервера. Хоча Gobuster демонструє кращу швидкість, його функціонал виявляється обмеженим у випадках з більш складними конфігураціями веб-додатків за наявності WAF або нетипової логіки доступу.

Katana було обрано як основний засіб фазингу через її можливість витягувати динамічні посилання, вбудовані в JavaScript-код. Це суттєво відрізняє її від Hakrawler чи Waybackurls, які фокусуються або на поверхневому HTML-аналізі, або на витягу URL-адрес із архівних джерел, що зовсім не гарантує актуальність або повноту зібраної інформації.

Для збору технологічної інформації про цільові сайти я віддав перевагу WhatWeb, оскільки він має широку базу плагінів, підтримує агресивне сканування та зручно автоматизується через CLI, забезпечуючи стабільніші результати при масовому скануванні. Альтернатива у вигляді Wappalyzer хоч і популярна, але більше орієнтована на ручну роботу через браузерні розширення, що унеможливлює її повноцінне застосування в автоматизованих сценаріях. При виборі враховано активність спільноти та регулярність оновлення інструментів, що гарантує своєчасне отримання нових шаблонів і виправлень.

Останій етап сканування здійснювався за допомогою Nmap, який хоч і поступається Masscan у швидкості, компенсує це глибиною аналізу. Завдяки NSE-скриптам, Nmap дозволяє не лише виявити активні порти, а й одразу оцінити наявні вразливості та сервіси, що критично важливо для глибокої технічної оцінки безпеки в рамках тестування проникнення.

3.3 Тестування і аналіз результатів

У якості цільового об’єкта для перевірки функціональності скрипта було обрано демонстраційний веб-ресурс Acunetix для етичного тестування - vulnweb.com. Тестування проводилося в середовищі Kali Linux 2023.4 (ядро 6.5.0), із попередньо встановленими утилітами та повноцінним доступом до мережі Інтернет через VPN. Bash-скрипт запускали в терміналі з правами на створення директорій та запис файлів.

Результати першого етапу показали, що інструмент subfinder виявив понад 20 піддоменів, серед яких працювали тільки 5. Також були виявлені менш очевидні варіанти з ускладненими іменами, що містили префікси та артефакти. Це свідчить про високий рівень охоплення скриптом пасивних джерел та словникового аналізу.

На етапі виявлення директорій за допомогою dirsearch були отримані два потоки результатів. Перший стосувався стандартних директорій, таких як /admin/, /docs/, /images/, /cart/, де деякі з них (наприклад, /admin/ на testphp.vulnweb.com) повертали код 200 та відкритий доступ до форми входу. Другий - через brute-force на rest.vulnweb.com - продемонстрував серію помилок 403 на низці чутливих файлів (.htaccess.bak1, .htpasswd_test тощо) та успішне виявлення конфігураційних файлів config.php і db.sql з кодом 200. Аналогічно, dirb підтвердив понад 20 директорій, зокрема потрібність перевірки /info.php і /files/.

Фазинг динамічних посилань з Katana згенерував близько 40 URL, серед яких є параметризовані запити із id-параметром, потенційно вразливі до SQL-ін’єкцій або XSS. Це значно розширює картину атаки порівняно зі статичними списками.

Метадані WhatWeb свідчили про використання Apache 2.4.25 (Debian) та PHP 5.4.45, а також відсутність атрибута HttpOnly у cookies, що полегшує

викрадення сесій. Динамічна генерація JS-фреймворків не була виявлена, але низька версія PHP підвищує ризик експлуатації старих вразливостей.

Порт-сканування Nmap підтвердило два відкриті порти - 80 (HTTP) та 443 (HTTPS). Застосування скрипту vuln виявило низку висококритичних вразливостей Apache, підтверджених CVE з рейтингом 9.8. Окрім того, виявлено потенційні резервні файли (/db.sql, /info.php, /docs/) та незахищений HTTP TRACE, що створює ризик Cross Site Tracing. Крім того, не було виявлено налаштувань заголовка HSTS, що підвищує ймовірність атак зі зниженням рівня захисту

Загальний час виконання повного циклу сканування становив приблизно 10 хвилини, причому скрипт не викликав зависань або критичних збоїв. Точність результатів була в межах очікуваних; лише дублікатні URL із Katana потребували додаткової фільтрації.

Отримані дані мають значну цінність для зловмисника: список піддоменів і відкриті директорії дозволяють здійснювати фішинг через піддомени, XSS-атаки через параметризовані посилання й перелічення API-методів. Наявність резервних архівів і конфігураційних файлів відкриває шляхи для ін’єкції коду, а відсутність належної cookie-безпеки - для викрадення сесій.

Попри успішну і швидку інвентаризацію піддоменів, під час тестування було виявлено низку слабких місць скрипта. Насамперед, він залежить від стабільного інтернет-з’єднання. У випадках втрати доступу до цих сервісів точність та повнота результатів значно знижується.

Особливої уваги заслуговує поведінка скрипта у випадку наявності WAF або інших захисних механізмів, таких як rate-limiting, CAPTCHA, блокування IP-адрес або геофільтрація. У подібних умовах скрипт працює нестабільно - при спробі виконати DNS-брутфорс або HTTP-запити через whatweb чи katana, WAF може заблокувати підозрілу активність, що може спровокувати розрив з’єднання. У результаті частина запитів не виконується або повертає помилку тайм-ауту, що знижує ефективність збору даних. Таким чином, для

використання скрипта в реальному середовищі необхідно враховувати наявність захисної інфраструктури цільових ресурсів. Доцільним буде розширення скрипта за рахунок підтримки:

- таймінгових затримок між запитами;
- обфускації User-Agent;
- використання proxy;
- рандомізації підходів до DNS-брутфорсу та API-запитів.

3.4 Захист від автоматизованих recon-інструментів

У цьому підрозділі наведено рекомендації та технічні заходи для зменшення ефективності автоматизованих інструментів кіберрозвідки, що використовуються для збору інформації про інфраструктуру. Метою є ускладнення OSINT-процедур та обмеження активного сканування вебресурсів.

3.4.1 Рекомендації щодо захисту від OSINT-розвідки

Передусім слід мінімізувати доступність метаданих про домен і сервер. Наприклад, використовувати WHOIS-privasy для приховування контактної інформації власника та публікувати мінімальний набір DNS-записів, потрібних для роботи сервісів. Повністю сховати піддомени практично неможливо (їх оголошують публічні сертифікати й відкриті DNS), тому непотрібні середовища краще переводити в офлайн чи захищати доступ (наприклад, за допомогою окремих облікових даних) або ж використовувати wildcard-сертифікат, щоб у публічних базах не фігурували окремі записи. Не менш важливо захищати службові файли і директорії. Необхідно забороняти доступ до прихованих файлів конфігурації, таких як .env, .git та інші. Це досягається через конфігурацію сервера - наприклад, директиви location ~ /\. в Nginx чи правила <FilesMatch "^\."> Deny from all в Apache [29]. Навіть тимчасове потрапляння .env на сервер може призвести до витоку секретів (облікових даних тощо). Аналогічно варто обмежити доступ до інших чутливих каталогів (admin/, конфігурацій тощо) – замість розміщення адмін-інтерфейсів у публічній зоні можна вимагати авторизації або доступу лише з довірених IP. Уникнення додаткового освітлення прихованих ресурсів: файл robots.txt слугує інструкцією для пошукових роботів, але він відкритий публічно. Не варто додавати в нього секретні шляхи для приховування - навпаки, це видає потенційним зловмисникам місцезнаходження цих сторінок [30]. Тому варто взагалі

обмежити вміст robots.txt лише загальними інструкціями, а не дізнаватись для атаки шляхи через цей файл. Для обмеження автоматичного сканування сайта рекомендується впровадити механізми відстеження і блокування підозрілої активності. Системи типу WAF чи IDS можуть аналізувати шаблони запитів та виявляти “поведінку сканера”: множинні послідовні запити з різними параметрами або часті звернення до існуючих сторінок за короткий час. У таких системах можна налаштувати фільтрацію за User-Agent - наприклад, блокувати відомі сигнатури сканерів (WhatWeb, Katana тощо), а також обмежувати частоту запитів з одного IP (rate limiting). Наприклад, стаття про блокування бота WhatWeb рекомендує відкидати запити з User-Agent «WhatWeb» і вводити rate limiting для автоматизованих сканувань. Крім того, WAF може містити спеціальні правила, що ідентифікують характерні шаблони роботи сканерів (наприклад, незвичайні заголовки або послідовності URL). Додатково до згаданих заходів, при підозрі сканування можна вводити CAPTCHA на певні форми або сторінки, щоб відрізняти боти від реальних користувачів. Рекомендується використовувати CDN і проксі, які візуально маскують вихідні сервери: наприклад, приховувати сервіс-підпис («ServerTokens») у HTTP-заголовках Apache та Nginx, виключати версії ПЗ із заголовків відповіді. Також фільтрація User-Agent і HTTP-заголовків на рівні CDN/WAF здатна ускладнити збір OSINT-даних - наприклад, додавання довільних чи некоректних названь серверів замість реальних значень у заголовках може збивати деякі боти з пантелику.

3.4.2 Технічні поради з конфігурування серверів

По-перше, потрібно вимкнути у веб-сервері оприлюднення версій програмного забезпечення. У Apache це робиться через ServerTokens Prod та ServerSignature Off, у Nginx - server_tokens off. Так само не слід передавати відомості про ОС хоста в заголовках. Це ускладнює автоматичну ідентифікацію серверного софту під час сканування. По-друге, слід відключити індексацію

директорій. За замовчуванням деякі сервери повертають список файлів папки, якщо в ній немає index.html. Активований «directory listing» може ненавмисно викрити приховані файли (наприклад, резервні копії, старі скрипти тощо). Як наголошується в галузевих рекомендаціях, вимкнення індексації директорій - це проста краща практика безпеки. Її реалізують на рівні конфігурації сервера (або створюючи порожній index у кожній папці, але краще загальною директивою серверу, щоб не забути це). Аудит URL-адрес сприяє зменшенню поверхні атаки. Рекомендується переглянути структуру шляхів і прибрати або перемістити непотрібні кінцеві точки (наприклад, інтерфейси розробки, тесту чи конфігурації). Наявність незахищених API-ендпоінтів чи тестових сторінок слід виявити і виключити. Приклад: перенести конфіденційні API на окремий субдомен з додатковим захистом або забезпечити їхню аутентифікацію. Також варто переглянути шаблони генерації URL (наприклад, випадкові довгі імена ресурсів замість очевидних), щоб ускладнити їхнє вгадування сканерами. Ще один напрям - протидія порт-скануванню. Сервер чи мережеві пристрої можуть запровадити фільтрацію TCP: наприклад, використання SYN-cookie і обмеження швидкості обробки SYN-запитів захищає від надмірних спроб встановити з’єднання. Також розгорнення систем IDS/IPS дозволяє виявляти аномальну активність на мережевому рівні - нетипові серії пакетів, SYN-флуд тощо. Це дозволяє блокувати IP, які намагаються масово сканувати порти. Спеціально для REST/API-серверів слід обмежити права доступу: вимагати API-ключі чи токени, впроваджувати контролі аутентифікації й авторизації на рівні запитів. Також налаштовуйте логування активності API, щоб можна було реагувати на незвичайні шаблони звернень, наприклад численні невдалі виклики або звернення до неіснуючих ендпоінтів. Варто також застосовувати rate limiting для API щоб обмежити число викликів за одиницю часу, та перевірку правильності запитів (фільтрація за заголовками, параметрами). На рівні веб-серверів і додатків встановіть сучасні заголовки безпеки. До них належать щонайменше: X-Content-Type-Options: nosniff (перешкоджає MIME-сніффінгу), X-Frame-Options: DENY (захищає від clickjacking) або

відповідний директивам CSP frame-ancestors, Content-Security-Policy (CSP – для жорсткого визначення дозволених джерел скриптів, стилів тощо). Правильно налаштовані CSP, HSTS, X-Content-Type-Options та інші заголовки закривають цілі класи атак (XSS, зміна MIME-типів, підміна контенту). Наприклад, як зазначає стаття з Invicti, найважливішими серед них є Content-Security-Policy і Strict-Transport-Security, а також звичайно X-Content-Type-Options: nosniff для запобігання несподіваній інтерпретації контенту браузерами.

Висновок до розділу 3

У третьому розділі було реалізовано та всебічно описано прототип автоматизованої системи кіберрозвідки на базі Bash-скрипта. Спочатку було обґрунтовано вибір Bash як платформи: його сумісність із UNIX-системами, мінімальні залежності і прозорість інтеграції з CLI-утилітами дозволили створити гнучкий і легкий у підтримці інструмент.

Було детально розглянуто функціонал скрипта: від сегментованої ініціалізації робочого каталогу, пасивного збору піддоменів, активного перебору директорій, фазингу динамічних URL та агрегування метаданих, до виявлення вразливих портів з використанням NSE-скриптів Nmap і генерації інтерактивного HTML-звіта.

У наступному підрозділі було проведено тестування на демонстраційному ресурсі vulnweb.com, що підтвердило працездатність скрипта: за 10 хвилини були виявлені понад 20 піддоменів, критичні конфігураційні файли, динамічні точки входу й низка високорейтингових вразливостей Apache. Одночасно фіксовано обмеження - чутливість до WAF, дублювання даних і складнощі з обробкою великих обсягів результатів.

Наостанок було запропоновано низку рекомендацій щодо підвищення стійкості до OSINT-роков: мінімізацію метаданих, обмеження доступу до

службових ресурсів, впровадження WAF/CDN, приховування банерів серверів та жорсткіше налаштування HTTP-заголовків.

ВИСНОВОК

У процесі виконання кваліфікаційної роботи на тему «Автоматизована система кіберрозвідки об'єкта інформаційної діяльності» було розроблено ефективний і гнучкий інструмент автоматизованої кіберрозвідки, здатний уніфікувати процес збору інформації про інфраструктуру організації та сформувати попередню картину потенційних векторів атаки. У першому розділі дослідження було закладено фундаментальне розуміння об’єкта захисту - інформаційної інфраструктури як складної та динамічної системи, що перебуває під безперервним пресингом численних внутрішніх і зовнішніх загроз. Було розглянуто основи інформаційної безпеки в корпоративному середовищі, ключові концепції CIA, PoLP та роль мережевої сегментації у зменшенні поверхні атаки. Окрему увагу приділено аналізу кіберрозвідки як формуючого етапу будь-якої атаки, де пасивні та активні методи дозволяють злочинцям мапувати цільові системи та виявляти їх слабкі місця.

Другий розділ зосередився на методології проведення рекону: детально описано техніки OSINT та активного сканування, включно з WHOIS-та DNS-аналізом, Google Dorking, OSINT-сервісами, пасивними та активними сканерами (WhatWeb, Shodan, Nmap), а також інструментами для перебору директорій і піддоменів. Проведений огляд основних CLI-утиліт показав, що поєднання різних джерел інформації в єдиному ланцюгу аналізу значно підвищує повноту зібраних даних і дає змогу оперативно оцінити поверхню атаки.

Третій розділ практично підтвердив обґрунтованість обраного підходу: розроблений на Bash-скрипт об’єднав Subfinder, Dirsearch, Dirb, Katana, WhatWeb і Nmap у єдину автоматизовану команду, де кожен етап передає свої

результати далі в логічній послідовності. Тестування на демонстраційному ресурсі vulnweb.com довело, що цей скрипт здатний швидко зібрати понад 20 піддоменів, виявити критичні конфігураційні файли, динамічні точки входу і навіть знайти серйозні вразливості Apache із рейтингом CVSS 9.8. Результати також вказали на слабкі місця - залежність від стабільності зовнішніх сервісів, підвищену чутливість до WAF та rate-limiting і потребу в обробці дублікатів та помилок.

Викладені рекомендації щодо мінімізації OSINT-движків та поради з налаштування серверів створюють основу для посилення стійкості організаційних ресурсів. Особливо важливим є впровадження систем виявлення аномалій і deceptive-технологій, які дозволяють реагувати на активність сканування у режимі реального часу.

Виконана робота не лише довела ефективність поєднання простих, відкритих CLI-утиліт у скриптовому рішенні, а й окреслила перспективи подальшого розвитку: інтеграцію в CI/CD, розширення бази даних результатів, підтримку різноманітних API та проксі-маршрутизацію для обходу захисних механізмів. Запропоновані удосконалення та рекомендації формують дорожню карту для створення масштабованої, надійної та автоматизованої системи кіберрозвідки, здатної адаптуватися до змінних умов кіберзагроз.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. SecurityScorecard. The human factor in cybersecurity [Electronic resource] // SecurityScorecard. – 2024. – Access: https://securityscorecard.com/blog/the-human-factor-in-cybersecurity/

2. Baker K. Common types of cyberattacks [Electronic resource] // CrowdStrike. – 2024. – Access: https://www.crowdstrike.com/en-us/cybersecurity-101/cyberattacks/common-cyberattacks/

3. Proofpoint Threat Reference. Pretexting [Electronic resource] // Proofpoint Threat Reference. – Access: https://www.proofpoint.com/us/threat-reference/pretexting

4. OWASP Cheat Sheet Series. OWASP Attack Surface Analysis Cheat Sheet [Electronic resource] // OWASP Cheat Sheet Series. – Access: https://cheatsheetseries.owasp.org/cheatsheets/Attack_Surface_Analysis_Cheat_Sheet.html

5. Sloan R. Minimizing your company’s attack surface: key cyber protection [Electronic resource] // Zscaler. – 2024. – Access: https://www.zscaler.com/cxorevolutionaries/insights/minimizing-your-companys-attack-surface-key-cyber-protection

6. Barker E., Branstad D., Chokhani S., Smid M. An Ontology of Identity Credentials [Electronic resource] // NIST. – 2009. – Access: https://nvlpubs.nist.gov/nistpubs/Legacy/IR/nistir7609.pdf

7. OWASP. Security Misconfiguration [Electronic resource] // OWASP Top Ten 2017. – Access: https://owasp.org/www-project-top-ten/2017/A6_2017-Security_Misconfiguration.html

8. American Registry for Internet Numbers (ARIN). Whois Database [Electronic resource] // ARIN. – Access: https://www.arin.net/resources/registry/whois/

9. Vande Casteele S. The Art of Subdomain Enumeration [Electronic resource] // Outpost24. – 2025. – Access: https://outpost24.com/blog/art-of-subdomain-enumeration/

10. OSINT Ambition. 5 Basic Techniques for Automating Investigations Using the Wayback Machine [Electronic resource] // OSINT Ambition. – 2023. – Access: https://publication.osintambition.org/5-basic-techniques-for-automating-investigations-using-the-wayback-machine-archive-org-3d1f2b8247d2

11. Borges E. OSINT Tools [Electronic resource] // Recorded Future. – 2024. – Access: https://www.recordedfuture.com/threat-intelligence-101/tools-and-technologies/osint-tools

12. Imperva. Google Dorking [Electronic resource] // Imperva. – Access: https://www.imperva.com/learn/application-security/google-dorking-hacking/

13. Secjuice. LinkedIn OSINT Techniques Part II [Electronic resource] // Secjuice. – 2020. – Access: https://www.secjuice.com/linkedin-osint-techniques-part-ii/

14. Nmap Documentation. Port Scanning Techniques [Electronic resource] // Nmap Documentation. – Access: https://nmap.org/book/man-port-scanning-techniques.html

15. Верховна Рада України. Закон України “Про основні засади забезпечення кібербезпеки України” [Електронний ресурс] // Верховна Рада України. – Режим доступу: https://zakon.rada.gov.ua/laws/show/2163-19#Text

16. Верховна Рада України. Кримінальний кодекс України [Електронний ресурс] // Верховна Рада України. – Режим доступу: https://zakon.rada.gov.ua/laws/show/2341-14#Text

17. MorningStar Security Research. WhatWeb [Electronic resource] // MorningStar Security Research. – Access: https://morningstarsecurity.com/research/whatweb

18. BIND9 Documentation. Dig Command Manual [Electronic resource] // BIND9 Documentation. – Access: https://bind9.readthedocs.io/en/latest/manpages.html#dig

19. Messer P. Optimizing Your Nmap Scan [Electronic resource] // Professor Messer. – Access: https://www.professormesser.com/nmap/optimizing-your-nmap-scan-nmap-scanning-methods/

20. CheapSSL Security. What is the UDP Protocol? [Electronic resource] // CheapSSL Security. – 2022. – Access: https://cheapsslsecurity.com/blog/what-is-the-udp-protocol-a-user-datagram-protocol-definition/

21. Meyer K. Gobuster Directory Enumerator Cheat Sheet [Electronic resource] // Abricto Security. – 2022. – Access: https://abrictosecurity.com/gobuster-directory-enumerator-cheat-sheet/

22. YesWeHack. Subdomain Enumeration – Expand Attack Surface [Electronic resource] // YesWeHack. – 2025. – Access: https://www.yeswehack.com/learn-bug-bounty/subdomain-enumeration-expand-attack-surface

23. Oladimeji S., Kerner S.M. SolarWinds Hack Explained [Electronic resource] // TechTarget. – 2023. – Access: https://www.techtarget.com/whatis/feature/SolarWinds-hack-explained-Everything-you-need-to-know

24. Ilascu I. Capcom ransomware gang used old VPN device [Electronic resource] // BleepingComputer. – 2021. – Access: https://www.bleepingcomputer.com/news/security/capcom-ransomware-gang-used-old-vpn-device-to-breach-the-network/

25. Sophos News. Slickwraps data breach earns scorn [Electronic resource] // Sophos News. – 2020. – Access: https://news.sophos.com/en-us/2020/02/27/slickwraps-data-breach-earns-scorn-for-all/

26. Kime C. Acunetix Web Application Security Review [Electronic resource] // Datamation. – 2022. – Access: https://www.datamation.com/security/acunetix-web-application-security-review/

27. ProjectDiscovery. Nuclei – Overview [Electronic resource] // ProjectDiscovery. – Access: https://docs.projectdiscovery.io/tools/nuclei/overview

28. Banach Z. Disable Directory Listing on Web Servers [Electronic resource] // Invicti. – 2024. – Access: https://www.invicti.com/blog/web-security/disable-directory-listing-web-servers/

29. Essam I. Developers: Hide Your Hidden Files [Electronic resource] // Medium. – 2018. – Access: https://medium.com/@zoxta/developers-hide-your-hidden-files-7de8132c5d44

30. MDN Web Docs. Practical Implementation of robots.txt [Electronic resource] // MDN Web Docs. – Access: https://developer.mozilla.org/en-US/docs/Web/Security/Practical_implementation_guides/Robots_txt

ДОДАТОК А

Таблиця кодів веб-браузера

Код	*Опис*
1xx	Запит прийнято
100	Клієнт може продовжити запит
101	Сервер прийняв запит на зміну протоколу
2xx	Запит оброблено успішно
200	Успішна відповідь
201	Ресурс успішно створено
3xx	Подальші дії клієнта необхідні
301	Постійне перенаправлення
302	Тимчасове перенаправлення
304	Кешована версія актуальна
4xx	Запит містить помилку
400	Неправильний синтаксис запиту
401	Потрібна авторизація
403	Доступ заборонено
404	Ресурс не знайдено
5xx	Сервер не зміг виконати запит
500	Внутрішня помилка сервера
502	Сервер отримав неправильну відповідь сервера

503	Сервер тимчасово недоступний
504	Таймаут при очікуванні відповіді іншого сервера

ДОДАТОК Б

Програмна реалізація алгоритму

```
#!/bin/bash

# Функція для виконання команди з можливістю переривання (Ctrl+C)

run_with_interrupt() {

    "$@" &

    cmd_pid=$!

    # Встановлюємо обробник SIGINT для переривання лише поточного процесу

    trap "echo 'Перервано поточне завдання, переходимо до наступного...'; kill $cmd_pid 2>/dev/null; wait $cmd_pid; return 1" SIGINT

    wait $cmd_pid

    ret=$?

    trap - SIGINT

    return $ret

}

# Перевірка та встановлення необхідних утиліт

install_tools() {

    tools=("subfinder" "dirsearch" "dirb" "katana" "whatweb" "nmap")

    for tool in "${tools[@]}"; do

        if ! command -v "$tool" &>/dev/null; then

            echo "$tool не знайдено, встановлюю..."

            sudo apt-get install -y "$tool" || sudo snap install "$tool" || echo "Не вдалося встановити $tool"

        fi

    done

}
```

Продовження додатку Б

```
install_tools

# Запит доменного імені

read -p "Введіть ім'я сайту (наприклад, example.com): " domain

mkdir -p "$domain"

cd "$domain" || exit 1

# Етап 1: Пошук піддоменів

echo "Пошук піддоменів..."

run_with_interrupt subfinder -d "$domain" -o subdomains.txt || echo "Subfinder перервано або завершився з помилкою"

sed 's/^/http:\/\//' subdomains.txt > urls.txt

# Етап 2: Пошук директорій та адмін-панелей

rm -f dirsearch.txt dirb.txt okadminfinder.txt  # Очищення старих даних

while read -r url; do

    echo "Сканування $url..."

    echo "Запуск dirsearch для $url"

    run_with_interrupt dirsearch -u "$url" -o temp_dirsearch.txt

    if [ $? -eq 0 ]; then

        grep -E '^(200|301|302|403)' temp_dirsearch.txt >> dirsearch.txt

    fi

    echo "Запуск dirb для $url"

    run_with_interrupt dirb "$url" -o temp_dirb.txt

    if [ $? -eq 0 ]; then

        cat temp_dirb.txt >> dirb.txt

    fi

    echo "Запуск okadminfinder для $url"
```

Продовження додатку Б

```
    run_with_interrupt okadminfinder -u "$url" -o temp_okadminfinder.txt
    if [ $? -eq 0 ]; then
        cat temp_okadminfinder.txt >> okadminfinder.txt
    fi
done < urls.txt
# Етап 3: Katana та WhatWeb
echo "Запуск Katana..."
run_with_interrupt katana -list subdomains.txt -jc -o katana.txt || echo "Katana перервана або завершилась з помилкою"
rm -f whatweb.txt ip.txt  # Очищення старих даних
while read -r url; do
    echo "Аналіз $url..."
    run_with_interrupt whatweb -a 3 "$url" >> whatweb.txt
done < urls.txt
grep -oE '[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+' whatweb.txt | sort -u > ip.txt
# Етап 4: Сканування Nmap
rm -f nmap.txt  # Очищення старих даних
while read -r ip; do
    echo "Сканування $ip..."
    run_with_interrupt nmap -sV -Pn --script vuln "$ip" -o temp_nmap.txt
    if [ $? -eq 0 ]; then
        cat temp_nmap.txt >> nmap.txt
    fi
done < ip.txt
```

```
echo "Сканування завершено. Підсумковий звіт збережено у $report_file"
```

ДОДАТОК В

Створення звіту

```
#!/bin/bash
echo "Введіть шлях до директорії зі скан-файлами:"
read TARGET_DIR
if [[ ! -d "$TARGET_DIR" ]]; then
   echo "Помилка: Директорія '$TARGET_DIR' не існує."
   exit 1
fi
cd "$TARGET_DIR" || { echo "Не вдалося перейти в директорію"; exit 1; }
OUTPUT="recon_report.html"
cat <<EOF > $OUTPUT
<!DOCTYPE html>
<html lang="uk">
<head>
   <meta charset="UTF-8">
   <title>Звіт з кіберрозвідки</title>
   <style>
      body {
         font-family: monospace;
         background-color: #f4f4f4;
         padding: 20px;
      }
      h1, h2 {
         color: #333;
         user-select: none;
```

Продовження додатку В

```
      }
      .section {
         margin-bottom: 20px;
```

```
    padding: 10px;
    border: 2px solid #ccc;
    border-radius: 10px;
    background-color: #ffffff;
    box-shadow: 0 2px 5px rgba(0,0,0,0.1);
}
pre {
    background-color: #fefefe;
    padding: 15px;
    border: 1px solid #ddd;
    overflow-x: auto;
    white-space: pre-wrap;
    word-wrap: break-word;
    display: none; /* за замовчуванням сховано */
}
.expanded > pre {
    display: block;
}
h2 {
    cursor: pointer;
    margin: 0;
    padding: 10px;
}
```

Продовження додатку В

```
h2::before {
    content: "+ ";
    font-weight: bold;
    color: #666;
    margin-right: 6px;
}
.expanded > h2::before {
    content: "- ";
    color: #000;
}
```

```
    #toggleAll {
        padding: 10px 20px;
        margin-bottom: 20px;
        font-size: 16px;
        cursor: pointer;
        background-color: #007BFF;
        color: #fff;
        border: none;
        border-radius: 8px;
        box-shadow: 0 2px 5px rgba(0,0,0,0.2);
    }
</style>
<script>
    document.addEventListener('DOMContentLoaded', function() {
        const sections = document.querySelectorAll('.section');
        // Клік по кожному заголовку
```

Продовження додатку В

```
        sections.forEach(section => {
            const header = section.querySelector('h2');
            header.addEventListener('click', () => {
                section.classList.toggle('expanded');
            });
        });
        // Кнопка "Згорнути/Розгорнути всі"
        const toggleBtn = document.getElementById('toggleAll');
        let expanded = false;
        toggleBtn.addEventListener('click', () => {
            sections.forEach(section => {
                if (expanded) {
                    section.classList.remove('expanded');
                } else {
                    section.classList.add('expanded');
                }
            });
```

```
            expanded = !expanded;
            toggleBtn.textContent = expanded ? 'Згорнути всі' : 'Розгорнути всі';
        });
    });
  </script>
</head>
<body>
  <h1>Автоматизований звіт з кіберрозвідки</h1>
  <p>Дата генерації: $(date)</p>
```

Продовження додатку В

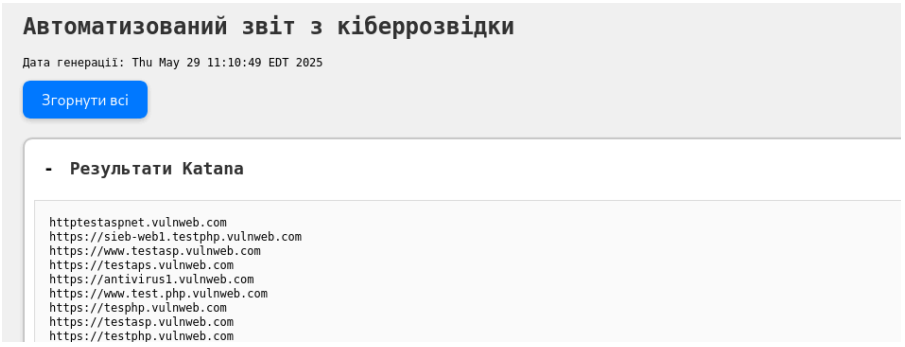
```
  <button id="toggleAll">Розгорнути всі</button>
EOF
declare -A files=(
  ["subdomains.txt"]="Знайдені піддомени"
  ["ip.txt"]="IP-адреси"
  ["urls.txt"]="Знайдені URL-адреси"
  ["whatweb.txt"]="Інформація з WhatWeb"
  ["nmap.txt"]="Результати сканування Nmap"
  ["temp_nmap.txt"]="Додаткові Nmap результати"
  ["dirsearch.txt"]="Результати Dirsearch"
  ["katana.txt"]="Результати Katana"
)
for file in "${!files[@]}"; do
  echo "<div class=\"section\">" >> $OUTPUT
  echo "<h2>${files[$file]}</h2>" >> $OUTPUT
  if [[ -s "$file" ]]; then
    echo "<pre>" >> $OUTPUT
    # Екранізація HTML символів
    cat "$file" | sed 's/&/\&amp;/g; s/</\&lt;/g; s/>/\&gt;/g' >> $OUTPUT
```

```
        echo "</pre>" >> $OUTPUT
    else
        echo "<p><i>Файл порожній або відсутній</i></p>" >> $OUTPUT
    fi
    echo "</div>" >> $OUTPUT
done
echo "</body></html>" >> $OUTPUT
echo "[+] HTML-звіт створено: $TARGET_DIR/$OUTPUT"
```

ДОДАТОК Д

Створений звіт

Результати фазингу



Результати брутфорсу директорій

- Результати Dirsearch

```
403   305B   http://rest.vulnweb.com/.htaccess.bak1
403   305B   http://rest.vulnweb.com/.htaccess.orig
403   305B   http://rest.vulnweb.com/.htaccess.save
403   307B   http://rest.vulnweb.com/.htaccess.sample
403   302B   http://rest.vulnweb.com/.ht_wsr.txt
403   306B   http://rest.vulnweb.com/.htaccess_extra
403   305B   http://rest.vulnweb.com/.htaccess_orig
403   303B   http://rest.vulnweb.com/.htaccessBAK
403   303B   http://rest.vulnweb.com/.htaccess_sc
403   303B   http://rest.vulnweb.com/.htaccessOLD
403   304B   http://rest.vulnweb.com/.htaccessOLD2
403   295B   http://rest.vulnweb.com/.htm
403   296B   http://rest.vulnweb.com/.html
403   301B   http://rest.vulnweb.com/.htpasswds
403   305B   http://rest.vulnweb.com/.htpasswd_test
403   302B   http://rest.vulnweb.com/.httr-oauth
200     0B   http://rest.vulnweb.com/config.php
200   317B   http://rest.vulnweb.com/db
200   317B   http://rest.vulnweb.com/db.sql
301   319B   http://rest.vulnweb.com/docs    -> REDIRECTS TO: http://rest.vulnweb.com/docs/
200     4KB  http://rest.vulnweb.com/docs/
301   320B   http://rest.vulnweb.com/files    -> REDIRECTS TO: http://rest.vulnweb.com/files/
301   321B   http://rest.vulnweb.com/images    -> REDIRECTS TO: http://rest.vulnweb.com/images/
200    23KB  http://rest.vulnweb.com/info.php
403   304B   http://rest.vulnweb.com/server-status
403   305B   http://rest.vulnweb.com/server-status/
403   305B   http://rest.vulnweb.com/.htaccess.bak1
403   305B   http://rest.vulnweb.com/.htaccess.orig
403   305B   http://rest.vulnweb.com/.htaccess.save
403   307B   http://rest.vulnweb.com/.htaccess.sample
```

Продовження додатку Д

- IP-адреси

```
18.215.71.186
44.228.249.3
44.238.29.244
```

Результати пошуку піддоменів

- Знайдені піддомени

```
www.vulnweb.com
testasp.vulnweb.com
www.testasp.vulnweb.com
www.test.php.vulnweb.com
tesphp.vulnweb.com
ttestphp.vulnweb.com
rest.vulnweb.com
restasp.vulnweb.com
edu-rost.ruwww.vulnweb.com
antivirus1.vulnweb.com
tetphp.vulnweb.com
testphp.vulnweb.com
testaps.vulnweb.com
testaspx.vulnweb.com
test.php.vulnweb.com
blogger.com.vulnweb.com
edu-rost.rutestasp.vulnweb.com
estphp.vulnweb.com
sieb-web1.testphp.vulnweb.com
testaspnet.vulnweb.com
testap.vulnweb.com
testapsnet.vulnweb.com
5cwww.vulnweb.com
virus.vulnweb.com
odincovo.vulnweb.com
viruswall.vulnweb.com
www.testphp.vulnweb.com
httptestaspnet.vulnweb.com
testhtml5.vulnweb.com
test.vulnweb.com
www.virus.vulnweb.com
```

Результати пошуку відкритих портів

- Результати сканування Nmap

```
# Nmap 7.95 scan initiated Thu May 29 10:53:48 2025 as: /usr/lib/nmap/nmap --privileged -sV -Pn --script vuln -o temp_nmap.txt 18.215.71.186
Nmap scan report for ec2-18-215-71-186.compute-1.amazonaws.com (18.215.71.186)
Host is up (0.13s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT     STATE SERVICE VERSION
80/tcp   open  http    Apache httpd 2.4.25 ((Debian))
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-server-header: Apache/2.4.25 (Debian)
|_http-dombased-xss: Couldn't find any DOM based XSS.
| http-fileupload-exploiter:
|
|_    Couldn't find a file-type field.
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
| http-enum:
|   /db.sql: Possible database backup
|   /info.php: Possible information file
|_  /docs/: Potentially interesting folder
| vulners:
|   cpe:/a:apache:http_server:2.4.25:
|       C94CBDE1-4CC5-5C06-9D18-23CAB216705E    10.0    https://vulners.com/githubexploit/C94CBDE1-4CC5-5C06-9D18-23CAB216705E  *EXPLOIT*
|       PACKETSTORM:181114      9.8     https://vulners.com/packetstorm/PACKETSTORM:181114      *EXPLOIT*
|       MSF:EXPLOIT-MULTI-HTTP-APACHE_NORMALIZE_PATH_RCE-       9.8     https://vulners.com/metasploit/MSF:EXPLOIT-MULTI-HTTP-APACHE_NORMALIZE_PATH_RCE-       *EXPLOIT*
|       MSF:AUXILIARY-SCANNER-HTTP-APACHE_NORMALIZE_PATH-       9.8     https://vulners.com/metasploit/MSF:AUXILIARY-SCANNER-HTTP-APACHE_NORMALIZE_PATH-       *EXPLOIT*
|       HTTPD:E8492EE5729E8FB514D3C0EE370C9BC6  9.8     https://vulners.com/httpd/HTTPD:E8492EE5729E8FB514D3C0EE370C9BC6
|       HTTPD:C072933AA965A86DA3E2C9172FFC1569  9.8     https://vulners.com/httpd/HTTPD:C072933AA965A86DA3E2C9172FFC1569
|       HTTPD:A1BBCE110E077FFBF4469D4F06DB9293  9.8     https://vulners.com/httpd/HTTPD:A1BBCE110E077FFBF4469D4F06DB9293
|       HTTPD:A09F9CEBE0B7C39EDA0480FEAEF4FE9D  9.8     https://vulners.com/httpd/HTTPD:A09F9CEBE0B7C39EDA0480FEAEF4FE9D
|       HTTPD:9BCBE3C14201AFC4B0F36F15CB40C0F8  9.8     https://vulners.com/httpd/HTTPD:9BCBE3C14201AFC4B0F36F15CB40C0F8
|       HTTPD:9AD76A782F4E666676719E36B64777A7A  9.8     https://vulners.com/httpd/HTTPD:9AD76A782F4E666676719E36B64777A7A
|       HTTPD:650C6B8A1FEAD1FBD1AF9746142659F9  9.8     https://vulners.com/httpd/HTTPD:650C6B8A1FEAD1FBD1AF9746142659F9
|       HTTPD:2BE0032A6ABE7CC52906DBAAFE0E448E  9.8     https://vulners.com/httpd/HTTPD:2BE0032A6ABE7CC52906DBAAFE0E448E
|       HTTPD:1F84410918227CC81FA7C000C4F999A3  9.8     https://vulners.com/httpd/HTTPD:1F84410918227CC81FA7C000C4F999A3
|       HTTPD:156974A46CA46AF26CC4140D00F7EB10  9.8     https://vulners.com/httpd/HTTPD:156974A46CA46AF26CC4140D00F7EB10
|       F9C0CD4B-3B60-5720-AE7A-7CC31DB839C5    9.8     https://vulners.com/githubexploit/F9C0CD4B-3B60-5720-AE7A-7CC31DB839C5  *EXPLOIT*
|       F8A7DE57-8F14-5B3C-A102-D546BDD8D2B8    9.8     https://vulners.com/githubexploit/F8A7DE57-8F14-5B3C-A102-D546BDD8D2B8  *EXPLOIT*
|       F607361B-6369-5DF5-9B29-E90FA29DC565    9.8     https://vulners.com/githubexploit/F607361B-6369-5DF5-9B29-E90FA29DC565  *EXPLOIT*
|       F41EE867-4E63-5259-9DF0-745881884D04    9.8     https://vulners.com/githubexploit/F41EE867-4E63-5259-9DF0-745881884D04  *EXPLOIT*
|       EDB-ID:51193    9.8     https://vulners.com/exploitdb/EDB-ID:51193      *EXPLOIT*
|       EDB-ID:50512    9.8     https://vulners.com/exploitdb/EDB-ID:50512      *EXPLOIT*
|       EDB-ID:50446    9.8     https://vulners.com/exploitdb/EDB-ID:50446      *EXPLOIT*
|       EDB-ID:50406    9.8     https://vulners.com/exploitdb/EDB-ID:50406      *EXPLOIT*
|       E81474F6-6DDC-5FC2-828A-812A8815E3B4    9.8     https://vulners.com/githubexploit/E81474F6-6DDC-5FC2-828A-812A8815E3B4  *EXPLOIT*
|       E796A40A-8A8E-59D1-93FB-78EF4D8B7FA6    9.8     https://vulners.com/githubexploit/E796A40A-8A8E-59D1-93FB-78EF4D8B7FA6  *EXPLOIT*
|       E59A01BE-8176-5F5E-BD32-D30B009CDBDA    9.8     https://vulners.com/githubexploit/E59A01BE-8176-5F5E-BD32-D30B009CDBDA  *EXPLOIT*
|       D7922C26-D431-5825-9897-B98478354289    9.8     https://vulners.com/githubexploit/D7922C26-D431-5825-9897-B98478354289  *EXPLOIT*
|       D5084D51-C8DF-5CBA-BC26-ACF2E33F8E52    9.8     https://vulners.com/githubexploit/D5084D51-C8DF-5CBA-BC26-ACF2E33F8E52  *EXPLOIT*
|       D10426F3-DF82-5439-AC3E-6CA0A1365A09    9.8     https://vulners.com/githubexploit/D10426F3-DF82-5439-AC3E-6CA0A1365A09  *EXPLOIT*
|       D0368327-F989-5557-A5C6-0D9ACDB4E72F    9.8     https://vulners.com/githubexploit/D0368327-F989-5557-A5C6-0D9ACDB4E72F  *EXPLOIT*
```

ДОДАТОК Е

СПИСОК ОПУБЛІКОВАНИХ ПРАЦЬ ЗА ТЕМОЮ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Тези наукових конференцій

Аналіз кіберрозвідки інфраструктури організації підприємства. / О. Горбатюк, Я. Шестак. // Проблеми кібербезпеки інформаційно-комунікаційних систем: Збірник матеріалів доповідей та тез; м. Київ, 11 квітня 2025 року; Київський національний університет імені Тараса Шевченка / та ін. – К.: ВПЦ "Київський університет", 2025. – 69 с.