Київський столичний університет імені Бориса Грінченка Факультет інформаційних технологій та математики
Кафедра інформаційної та кібернетичної безпеки імені професора Володимира Бурячка

«Допущено до захисту»
Завідувач кафедри інформаційної та кібернетичної безпеки імені професора Володимира Бурячка
кандидат технічних наук, доцент
Складанний П.М.

_____
(підпис)
« ___ » _____ 20__ р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття другого (магістерського)
рівня вищої освіти

Спеціальність 125 Кібербезпека та захист інформації

Тема роботи:
Дослідження методів та розробка рекомендацій щодо застосування методів та засобів захисту від RAT (Remote Access Trojan)

Виконав

студент групи БІКСм-1-24-1.4.д

Есаулов Олександр Дмитрович
(прізвище, ім’я, по батькові)

(підпис)

Науковий керівник

_____
(науковий ступінь, наукове звання)
Аносов А. О.
(прізвище, ініціали)

_____
(підпис)

Київ – 2025

Київський столичний університет імені Бориса Грінченка
Факультет інформаційних технологій та математики
Кафедра інформаційної та кібернетичної безпеки
імені професора Володимира Бурячка

Освітньо-кваліфікаційних рівень – магістр
Спеціальність 125 Кібербезпека та захист інформації
Освітня програма 125.00.01 Безпека інформаційних і комунікаційних систем

«Затверджую»
Завідувач кафедри інформаційної та
кібернетичної безпеки імені
професора Володимира Бурячка
кандидат технічних наук, доцент
Складанний П.М.

_____
(підпис)
« ___ » _____ 20__ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Есаулову Олександру Дмитровичу
(прізвище, ім'я, по батькові)

1. Тема роботи: **Дослідження методів та розробка рекомендацій щодо застосування методів та засобів захисту від RAT (Remote Access Trojan)**; керівник Аносов А. О., затверджені наказом ректора від «___»_____ 20___ року №__.
2. Термін подання студентом роботи «___»_____ 20___ р.
3. Вихідні дані до роботи:
   3.1 науково-технічна та нормативна література з теми дослідження
   3.2 технології: .NET (для аналізу AsyncRAT, QuasarRAT, NJRat forks); Python; GitHub;
4. Зміст текстової частини роботи (перелік питань, які потрібно розробити):
   4.1 Аналіз архітектури сучасних RAT-програм та їх функціональних можливостей.
   4.2 Дослідження особливостей роботи AsyncRAT, QuasarRAT, Pupy та NJRat та порівняння їхніх механізмів шифрування, транспорту та модульності.
   4.3 Розробка та демонстрація інструмента поведінкового виявлення RAT у Windows-системі.
5. Перелік графічного матеріалу:
   5.1 Презентація доповіді, виконана в Microsoft PowerPoint.
   5.2 Типові схеми
   -архітектури RAT-програм (AsyncRAT, QuasarRAT, Pupy, NJRat);
   -схема типового C2-з'єднання;
   -схема роботи інструмента виявлення RAT у системі;

-UML-діаграми модулів та компонентів.

6. Дата видачі завдання «___»_____ 20___ р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів підготовки роботи	Термін виконання	Примітка
1.	Уточнення постановки завдання		Виконано
2.	Аналіз літератури		Виконано
3.	Обґрунтування вибору рішення		Виконано
4.	Збір даних		Виконано
5.	Виконання та оформлення розділу 1.		Виконано
6.	Виконання та оформлення розділу 2.		Виконано
7.	Виконання та оформлення розділу 3.		Виконано
8.	Вступ, висновки, реферат		Виконано
9.	Апробація роботи на науково методичному семінарі та/або науково-технічній конференції		Виконано
10.	Оформлення та друк текстової частини роботи		Виконано
11.	Оформлення презентацій		Виконано
12.	Отримання рецензій		Виконано
13.	Попередній захист роботи		Виконано
14.	Захист в ЕК		

Студент Есаулов Олександр Дмитрович

(підпис) (прізвище, ім’я, по батькові)

Науковий керівник _____Аносов Андрій

Олександрович

(підпис) (прізвище, ім’я, по батькові)

РЕФЕРАТ

Кваліфікаційна робота присвячена технологіям використання систем виявлення та протидії шкідливому програмному забезпеченню типу Remote Access Trojan (RAT) в інформаційно-комунікаційних системах корпоративного рівня. Робота складається зі вступу, трьох розділів, що містять __ рисунків та __ таблиць, висновків та списку використаних джерел, що містить __ найменування. Загальний обсяг роботи становить __ сторінок, з яких сторінки займають ілюстрації і таблиці на окремих аркушах, а також додатки, перелік умовних скорочень та список використаних джерел.

Об’єктом дослідження в роботі є процес виявлення, аналізу та нейтралізації RAT у корпоративних інформаційних системах.

Предметом дослідження є методи та моделі виявлення RAT на основі поведінкових, мережевих та аналітичних підходів.

Метою роботи є підвищення ефективності систем захисту інформації шляхом удосконалення методів виявлення та блокування RAT у корпоративних мережах, а також формування рекомендацій щодо впровадження комплексної захисної інфраструктури.

Наукова новизна одержаних результатів полягає в тому, що в роботі запропоновано удосконалену концептуальну модель багаторівневої системи виявлення RAT, яка базується на поєднанні поведінкового аналізу, мережевих методів контролю та індикаторів компрометації. Галузь застосування. Запропоновані підходи можуть бути використані для створення корпоративних систем інформаційної безпеки, центрів моніторингу SOC, систем виявлення вторгнень та платформ автоматизованого реагування на кіберінциденти.

Ключові слова: БЕЗПЕКА, ЗАГРОЗА, ІНФОРМАЦІЯ, ІНФОРМАЦІЙНО-АНАЛІТИЧНА СИСТЕМА, ОБ’ЄКТ БЕЗПЕКИ, ПОРУШНИК, СИСТЕМА

ЗАХИСТУ, RAT, ВИЯВЛЕННЯ ЗАГРОЗ.

ЗМІСТ

СПИСОК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

RAT (Remote Access Trojan) – троян віддаленого доступу

C2 (Command and Control) – сервер керування та контролю

IoC (Indicators of Compromise) – індикатори компрометації

AV (Antivirus) – антивірусне програмне забезпечення

EDR (Endpoint Detection and Response) – система виявлення та реагування на атаки

MITM (Man-in-the-Middle) – атака “людина посередині”

API (Application Programming Interface) – інтерфейс прикладного програмування

TLS (Transport Layer Security) – протокол захищеного транспорту

HTTP/HTTPS – протоколи передавання гіпертексту

TCP/UDP – транспортні протоколи мережевого рівня

VM (Virtual Machine) – віртуальна машина

DNS (Domain Name System) – система доменних імен

OSINT (Open Source Intelligence) – розвідка з відкритих джерел

ВСТУП

Актуальність теми. За останні роки масштаби кіберзагроз стрімко зростають, а однією з найнебезпечніших категорій шкідливого програмного забезпечення стали трояни віддаленого доступу (Remote Access Trojan, RAT). За даними аналітичних звітів провідних компаній у сфері кібербезпеки [1], частка атак із використанням RAT зросла більш ніж на 40% у порівнянні з попередніми періодами. Цей тип шкідливих програм дозволяє зловмиснику непомітно отримати повний контроль над системою жертви, викрадати конфіденційні дані, здійснювати шпигунські операції, запускати додаткові модулі або розгортати інші види шкідливого ПЗ.

Основною ціллю таких атак стають державні установи, фінансові організації, підприємства малого та середнього бізнесу, які часто не мають достатнього рівня захисту та ресурсів для комплексного моніторингу інцидентів. За даними спеціалістів, лише у 2023 році було скомпрометовано понад 15 мільярдів записів персональних даних та корпоративної інформації внаслідок використання RAT та споріднених загроз [2, 3].

Особливу небезпеку становить і той факт, що в умовах зростання кількості віддалених робочих місць, гібридних форм зайнятості та активного використання

хмарних середовищ атаки на основі RAT набувають ще більшої ефективності. Низький рівень цифрової гігієни співробітників, брак сучасних методів контролю доступу та моніторингу, а також складність виявлення RAT у системах створюють передумови для суттєвого зростання кількості кіберінцидентів.

Проблема захисту від RAT є надзвичайно актуальною, оскільки такі атаки здатні завдати критичної шкоди: від крадіжки конфіденційних даних і блокування систем до організації DDoS-атак, фінансових втрат, простою бізнес-процесів або повного зупинення діяльності організації. Відсутність комплексного підходу до виявлення та протидії RAT у багатьох установах спричиняє суттєві ризики для інформаційної безпеки.

В сучасних умовах важливою складовою системи кіберзахисту є використання багатоетапних методів виявлення та протидії RAT, зокрема поведінкових аналізаторів, систем виявлення вторгнень, мережевих сенсорів, технологій машинного навчання, а також ефективних процедур інцидент-менеджменту та безперервного моніторингу [4].

У зв’язку з цим дослідження методів захисту від RAT, оцінювання їх ефективності та розробка практичних рекомендацій щодо впровадження засобів протидії є важливими завданнями сучасної інформаційної безпеки.

Вище перелічене підтверджує актуальність даного дослідження.

Мета роботи полягає у дослідженні методів та засобів виявлення, запобігання і протидії троянам віддаленого доступу (RAT), а також у розробці рекомендацій щодо їх ефективного застосування в інформаційних системах.

Для досягнення поставленої мети необхідно вирішити такі завдання:

1. Проаналізувати природу та класифікацію RAT, механізми їх роботи та основні вектори атак;
2. Дослідити сучасні методи виявлення RAT та програмні інструменти протидії;
3. Здійснити порівняльний аналіз ефективності існуючих методів і засобів захисту;

4. Розробити рекомендації щодо впровадження комплексної системи захисту від RAT у корпоративних мережах;
5. Сформувати модель багаторівневого захисту інформаційної системи від атак із використанням RAT.

Об’єкт дослідження – процес виявлення та протидії шкідливому програмному забезпеченню типу RAT.

Предмет дослідження – методи та засоби захисту інформаційних систем від троянів віддаленого доступу.

Методи дослідження. У роботі використано методи аналізу та синтезу, порівняльний аналіз, методи моделювання, системний підхід, статистичні методи, а також методи дослідження мережевого та поведінкового трафіку з метою виявлення ознак RAT.

Наукова новизна полягає у систематизації методів захисту від RAT, оцінці їх ефективності та формуванні комплексних рекомендацій щодо протидії цим загрозам у різних типах інформаційних систем. Робота пропонує структурований підхід до побудови багаторівневого захисту з урахуванням сучасних кіберзагроз та технік обходу захисних систем.

Теоретичне та практичне значення отриманих результатів полягає в обґрунтуванні необхідності використання комплексної системи протидії RAT, а також у можливості практичного застосування розроблених рекомендацій у корпоративних мережах, IT-компаніях, державних установах та підприємствах малого і середнього бізнесу.

Галузь застосування. Результати роботи можуть бути використані IT-підрозділами організацій для впровадження засобів захисту від RAT, під час побудови корпоративних політик безпеки, навчання фахівців та у рамках освітніх програм з кібербезпеки.

1 АНАЛІЗ СУЧАСНИХ ПІДХОДІВ ДО ПОБУДОВИ СИСТЕМ ЗАХИСТУ ВІД RAT-АТАК

1.1 Аналіз природи та класифікації шкідливого ПЗ типу RAT

Remote Access Trojan (RAT) є однією з найбільш поширених і небезпечних категорій шкідливого програмного забезпечення, призначеною для непомітного віддаленого доступу до комп'ютерної системи та повного контролю над нею. RAT поєднує у собі властивості як класичних троянів, так і засобів адміністрування, однак основна відмінність полягає у прихованому характері роботи та використанні шкідливих цілей.

На відміну від легітимного програмного забезпечення для дистанційного керування, RAT встановлюється без згоди користувача, часто маскується під нешкідливі файли, використовує приховані канали комунікації та може модифікувати свою поведінку з метою уникнення виявлення. Саме тому цей тип шкідливого ПЗ становить значну загрозу для корпоративних мереж, державних структур і приватних користувачів.

1.1.1 Механізми роботи Remote Access Trojan

Більшість RAT реалізують подібну архітектуру, яка складається з двох основних компонентів:

1. клієнтської частини, що встановлюється на комп’ютер жертви;
2. керуючого модуля (серверної частини), який використовується зловмисником для віддаленого контролю.

Ключові механізми роботи RAT включають (рис 1.1):

1) Встановлення та закріплення в системі

Після проникнення RAT намагається забезпечити свою присутність у системі, використовуючи техніки персистентності:

- автозапуск через реєстр;
- створення служб або планувальників завдань;
- копіювання себе у системні директорії;
- модифікацію налаштувань політик безпеки.

2) Встановлення каналу управління (Command and Control, C2)

RAT потребує стабільного зв’язку з командним сервером для отримання інструкцій. Канал комунікації може реалізовуватися через:

- TCP/UDP-з’єднання;
- HTTP(S)-тунелі;
- захищені шифровані канали;
- комунікацію через проксі або анонімізуючі мережі.

3) Збір інформації про систему

Після активації RAT зазвичай проводить розвідку:

- збір системної інформації;
- визначення привілеїв користувача;
- аналіз мережевих підключень;
- збір списку процесів і програм.

4) Виконання команд зловмисника

Коло можливостей RAT дуже широке і може включати:

- віддалене виконання команд;
- перегляд або зміну файлів;

- кейлоггінг, аудіо- та відеозапис;
- перехоплення трафіку;
- завантаження та запуск додаткових модулів;
- керування пристроями периферії.

5) Самозахист та приховування

RAT часто включають модулі антианалізу, які блокують спроби дослідження або видалення.

Таким чином, механізм роботи RAT є комплексним і включає як технічні, так і організаційні аспекти управління системою.

1.1.2 Канали розповсюдження та вектори атак

RAT активно використовуються у фішингових кампаніях, APT-атаках та масових зараженнях. Їх поширення здійснюється через різноманітні вектори:

1) Соціальна інженерія

Найпоширеніший спосіб інфікування — завантаження жертвою шкідливого файлу.

Типові сценарії:

- фішингові листи з вкладеннями;
- завантаження фальшивих програм або оновлень;
- інфіковані документи з макросами (Office, PDF).

2) Експлуатація вразливостей

Зловмисники можуть використовувати уразливості у:

- веб-браузерах;
- операційних системах;
- службах віддаленого доступу;
- популярних офісних програмах.

3) Легітимні канали поширення

До таких належать:

- компрометовані веб-сайти;
- зламані оновлення програмного забезпечення;
- безкоштовні інсталятори або зламані версії ПЗ.

4) Сторонні інструменти адміністрування

Іноді RAT маскується під:

- засоби віддаленої підтримки;
- інструменти тестування безпеки;
- діагностичні утиліти.

5) Поширення через мережеву інфраструктуру

Це стосується локальних заражень у корпоративних мережах:

- через спільні папки;
- через вразливі протоколи типу SMB;
- шляхом зараження мережевих пристроїв.

Таким чином, широкий спектр векторів атаки робить RAT особливо небезпечним та складним для виявлення.

1.1.3 Техніки приховування та обходу захисних систем

Сучасні RAT активно застосовують техніки ухилення від інструментів безпеки, що дозволяє їм тривалий час залишатися непоміченими.

До найважливіших технік належать (таблиця 1.1):

1) Обфускація та пакування коду

RAT маскують свій код, щоб ускладнити статичний аналіз:

- шифрування або стискання виконуваного файлу;
- динамічне завантаження модулів;
- приховані імпорти бібліотек.

2) Антианаліз та антивідлагодження

RAT можуть виявляти пісочниці або відладчики:

- перевірка наявності процесів аналізу;
- аналіз характеристик системи (наприклад, мінімальний обсяг RAM);
- уповільнення роботи у віртуальних середовищах.

3) Маскування мережевого трафіку

Застосовуються такі техніки:

- шифрування всього C2-трафіку;

- використання HTTPS або DNS-тунелювання;
- імітація легітимного трафіку (Google, Microsoft тощо).

4) Приховування процесів

RAT можуть:

- інжектувати свій код у легітимні системні процеси;
- запускатися як служби з типовими назвами;
- приховувати свої вікна та взаємодію з системою.

5) Уникнення антивірусів та EDR

Сучасні RAT намагаються:

- блокувати служби безпеки;
- вимикати журнали подій;
- змінювати права доступу до системних компонентів;
- підмінювати власні компоненти як «безпечні».

Техніки обходу захисту є ключовою складовою стійкості RAT і значно ускладнюють процес їх виявлення традиційними засобами.

Рис. 1.1 Схема класифікації RAT

Таблиця 1.1

Порівняння технік приховування та обходу захисних систем

Категорія методу	**Техніка**	**Опис**	**Типові прояви**	**Складність реалізації**
Уникнення аналізу	Anti-VM	Перевірка артефактів віртуальних середовищ	Відсутність запуску в VM	Середня
	Anti-Sandbox	Перевірка затримок, активності користувача	Нетипові таймінги виконання	Висока
	Anti-Debug	Перевірка debugger flags	Збої під відлагодженням	Середня
Уникнення моніторингу	API Unhooking	Відновлення оригінальних API-функцій	Аномалії в WinAPI-патернах	Висока
	Direct Syscalls	Обхід EDR шляхом прямих системних викликів	Raw syscalls у поведінці	Висока
	Obfuscated Syscalls	Шифрування або randomization системних викликів	Noisy trace	Дуже висока
Fileless техніки	Reflective DLL Injection	Завантаження DLL у пам’яті	Відсутність файлів на диску	Висока
	Shellcode Injection	Shellcode виконання в пам’яті	Suspicious RWX memory	Висока
Обхід AV	Packing/Obfuscation	Пакувальники, криптори	Поліморфізм екзешників	Середня

	Signature Evasion	Модифікація рядків, структур	Відсутність сигнатур	Легка
Persistence evasion	Living-off-the-land	Використання PowerShell/WMI	Anomalous LOLBins activity	Середня

1.2. Порівняння існуючих підходів до виявлення та протидії RAT

Системи захисту від шкідливого програмного забезпечення застосовують різні методи для виявлення та нейтралізації загроз типу Remote Access Trojan. Кожен підхід має власні переваги, недоліки та сфери ефективності. В умовах зростання складності RAT, їх здатності до приховування та динамічної зміни поведінки, жоден метод не є універсальним. Саме тому сучасні системи захисту комбінують сигнатурний аналіз, поведінкові методи, мережеву аналітику та технології машинного навчання.

Нижче розглянуто ключові підходи, які застосовуються в антивірусних рішеннях, EDR/EDR-платформах, системах моніторингу мереж та комплексних системах безпеки (рис 1.2).

1.2.1. Сигнатурні методи

Сигнатурне виявлення є класичним підходом, який використовується антивірусними системами протягом багатьох років. Метод ґрунтується на пошуку у файлах, пам’яті або мережевому трафіку унікальних фрагментів коду — **сигнатур**, притаманних конкретним зразкам шкідливого ПЗ.

Переваги сигнатурних методів:

- **Висока точність** при виявленні вже відомих RAT.
- **Мінімум хибних спрацювань** за наявності добре сформованих сигнатур.
- **Низькі обчислювальні витрати**, що дозволяє їх застосовувати на будь-яких пристроях.

Недоліки:

- **Неефективність проти нових або модифікованих зразків**

RAT, які змінюють сигнатури за допомогою обфускації, поліморфізму або пакування.

- **Залежність від регулярного оновлення бази сигнатур**.
- Неможливість виявлення **поведінкових аномалій**, характерних для складних атак.

Сигнатурні методи залишаються базовим інструментом, однак їх ефективність суттєво знижується у боротьбі з сучасними RAT, що активно використовують методи маскування.

1.2.2 Поведінковий аналіз

Поведінкові методи виявлення базуються на моніторингу дій програм у реальному часі та аналізі їх відповідності типовим моделям шкідливої активності. Замість пошуку унікального коду, система оцінює **послідовності операцій**, **аномальні дії** та **патерни поведінки**.

Основні поведінкові індикатори RAT:

- Нестандартні запити до мережі.
- Створення прихованих процесів.
- Модифікація реєстру для встановлення персистентності.
- Доступ до камер, мікрофонів та кейлоггінг-модулі.
- Різкі зміни системних привілеїв.

Переваги поведінкового аналізу:

- **Виявлення невідомих або обфускованих RAT**, навіть якщо їх сигнатури відсутні.
- Оцінка **реальної активності**, а не лише статичних характеристик.
- Можливість виявлення складних багатомодульних шкідливих програм.

Недоліки:

- **Ризик хибнопозитивних спрацювань**, особливо в системах з великою кількістю активних процесів.
- Високі обчислювальні витрати.

- Необхідність ретельного налаштування профілів поведінки.

Поведінковий аналіз є одним з найсильніших підходів у боротьбі з RAT, особливо якщо поєднується з іншими методами.

1.2.3 Аналіз мережевого трафіку

Оскільки RAT потребують зв’язку з командними серверами (C2), аналіз мережевого трафіку є одним із найрезультативніших підходів, особливо у корпоративних мережах.

Ключові напрямки мережевого аналізу:

- Виявлення **аномальних шаблонів трафіку** (часті звернення, нестандартні порти).
- Аналіз **DNS-запитів** на предмет підозрілих доменів.
- Кореляція обсягів переданих даних із типовою поведінкою пристрою.
- Виявлення **C2-комунікацій**, включно з зашифрованими каналами.
- Аналіз **поведінкових характеристик SSL/TLS-сесій**.

Переваги такого підходу:

- Можливість виявлення RAT навіть без доступу до зараженої системи.
- Ефективність проти складних, добре прихованих модулів, що працюють у пам’яті.
- Використання у SIEM, IDS/IPS та мережевих сенсорах.

Недоліки:

- Сучасні RAT активно використовують шифрування, що ускладнює аналіз.
- Великий обсяг трафіку вимагає складної інфраструктури обробки.
- Деякі RAT можуть маскуватися під легітимні сервіси.

Аналіз мережевого трафіку є критично важливим для виявлення RAT, особливо у масштабних мережах, однак потребує додаткових методів для аналізу

зашифрованих каналів.

1.2.4 Методи на основі машинного навчання

З розвитком штучного інтелекту з’явилася можливість аналізувати великі обсяги даних та визначати складні аномалії, які важко ідентифікувати традиційними методами. Машинне навчання (ML) дозволяє будувати моделі, що визначають ознаки RAT на основі статистичних та поведінкових характеристик.

Основні напрямки застосування ML:

- класифікація шкідливих і легітимних файлів;
- кластеризація подібних зразків RAT;
- виявлення аномалій у поведінці процесів;
- аналіз мережевого трафіку і формування профілів нормальної активності;
- прогнозування нових варіантів атак на основі історичних даних.

Переваги методів ML:

- **Висока ефективність проти нових та модифікованих RAT**;
- можливість працювати з великими наборами даних;
- здатність автоматично адаптуватися до змін у поведінці атак.

Недоліки:

- необхідність великих обсягів достовірних навчальних даних;
- складність інтерпретації рішень моделей «чорного ящика»;
- ризик впливу шкідливих даних (data poisoning).

Методи машинного навчання демонструють найвищу ефективність у сучасних системах виявлення RAT, проте їх застосування потребує комплексного підходу, значних ресурсів та експертизи.

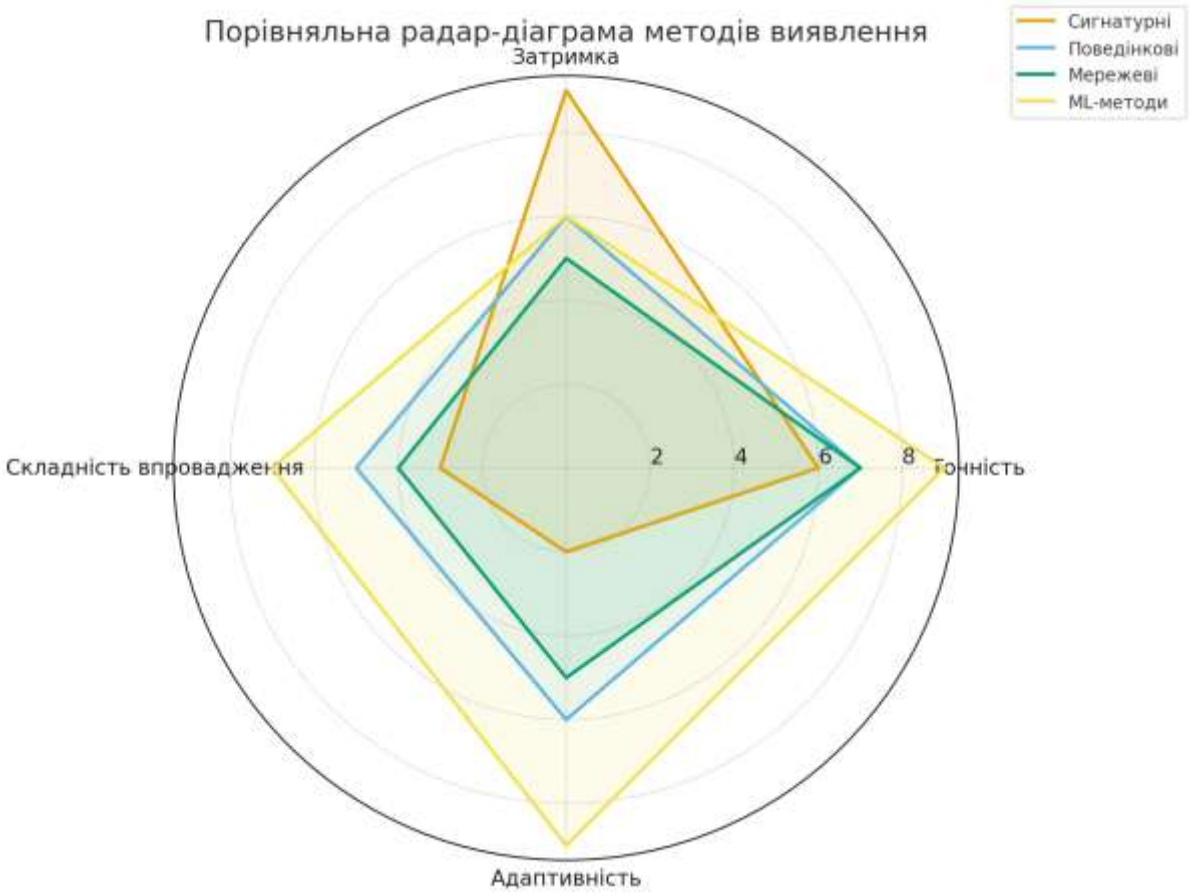


Рис. 1.2 Порівняння існуючих підходів до виявлення та протидії RAT

Таблиця 1.2

Порівняння патернів маскування

Категорія	**Метод**	**Призначення**	**Типові патерни**	**Складність**
Маскування виконання	Process Injection	Приховане виконання в легітимних процесах	Аномалії w/ svchost.exe, explorer.exe	Висока
	Process Hollowing	Замінена область пам’яті процесу	Hollowed дитина-процес	Дуже висока
	Thread Hijacking	Використання потоків легітимного	Неочікувані thread-start events	Висока

		процесу		
Маскування трафіку	Domain Fronting	Маскування трафіку під CDN	TLS SNI anomalies	Висока
	TLS Obfuscation	Модифікація TLS fingerprint	Нетипові JA3-хеші	Середня
	HTTP Mimicry	Маскування під браузер	User-Agent spoofing	Середня
Приховування коду	Code Obfuscation	Ускладнення реверсу	Заплутана логіка, random names	Легка
	String Encryption	Шифрування рядків	Decryption loops	Легка
	Control-flow flattening	Зміна структури виконання	Аномальна CFG	Середня
Уникнення OS visibility	Rootkit-техніки	Приховування процесів/файлів	Невидимі об’єкти в інструментах моніторингу	Дуже висока

1.3 Формалізація задачі виявлення та нейтралізації RAT у корпоративних мережах

Ефективний захист корпоративної інфраструктури від Remote Access Trojan вимагає чітко визначеної моделі загроз, формалізованих вимог до системи безпеки та критеріїв оцінювання результативності застосованих засобів. Формалізація

задачі дозволяє визначити межі дослідження, структурувати процес виявлення RAT та встановити об’єктивні показники, що характеризують рівень захищеності компанії.

1.3.1 Модель загроз для інформаційної системи

Модель загроз описує потенційні вектори атак, можливі наслідки компрометації та здатності зловмисника. Для RAT, які орієнтовані на встановлення прихованого віддаленого контролю над системою, ключовими загрозами є:

Основні загрози:

1. **Несанкціонований віддалений доступ**
Зловмисник отримує можливість управляти системою, виконувати команди, встановлювати інше ПЗ.
2. **Витік конфіденційної інформації**
RAT часто включають модулі збирання даних: паролі, документи, мережеві конфіги, скріншоти, відео, аудіо тощо.
3. **Порушення цілісності та функціонування системи**
Атакувальник може модифікувати файли, вносити зміни до ОС, встановлювати бекдори.
4. **Переміщення всередині мережі (lateral movement)**
Через заражену машину зловмисник поширює атаку в інфраструктурі.
5. **Ескалація привілеїв**
Більшість RAT намагаються отримати привілеї адміністратора.
6. **Встановлення персистентності**
RAT закріплюються у системі через автозапуск, служби або завдання планувальника.

Актори загроз:

- індивідуальні хакери;
- організовані кібергрупи (APT);
- внутрішні порушники;
- конкуренти;

- державні структури (кібершпигунство).

Оцінка можливостей зловмисника:

- середній — використання готових RAT із GitHub або даркнет-маркетів;
- високий — здатність модифікувати RAT, вбудовувати обфускацію;
- дуже високий — створення унікального RAT під конкретну ціль.

Таким чином, модель загроз формує основу для подальшого визначення рівня захисту, який повинна забезпечувати корпоративна система.

1.3.2 Вимоги до системи захисту

Вимоги формуються відповідно до моделі загроз та визначають функціональні і нефункціональні характеристики системи, що повинні забезпечити виявлення, аналіз та нейтралізацію RAT.

Функціональні вимоги:

1. **Виявлення шкідливих файлів та процесів**
   – на основі сигнатур;
   – аналізу поведінки;
   – статичного та динамічного аналізу.
2. **Моніторинг мережевого трафіку**
   – виявлення підозрілих з’єднань;
   – фіксація аномальних DNS-запитів;
   – виявлення командно-керуючих каналів (C2).
3. **Автоматичне блокування активності RAT**
   – завершення процесів;
   – блокування мережевих з’єднань;
   – ізоляція зараженого хоста.
4. **Встановлення персистентності та спроб її виявлення**
   – аналіз автозапуску;
   – аналіз реєстру;
   – контроль системних служб.

5. **Збір та кореляція телеметрії**

– обробка логів;

– інтеграція з SIEM/EDR.

6. **Створення звітності та аудит**

– фіксація інцидентів;

– хронологія подій;

– рекомендації щодо усунення.

Нефункціональні вимоги:

- **Низький рівень хибнопозитивних спрацювань** — рекомендаційне значення <5%.
- **Мінімальний вплив на продуктивність системи** — споживання CPU < 5–7%.
- **Стійкість до обходу** — захист від руткітів, тамперінгу.
- **Масштабованість** — можливість впровадження у мережах >1000 вузлів.
- **Сумісність** — підтримка Windows, Linux, macOS (за потреби).

1.3.3 Формалізація критеріїв ефективності

Для оцінювання результативності системи захисту використовуються формальні критерії, які дозволяють порівнювати різні методи та рішення.

1. Точність виявлення (Detection Rate, DR)

$$DR = \frac{TP}{TP + FN} \times 100\%$$

де:

TP — правильно виявлені RAT,

FN — невиявлені зразки RAT.

Показник характеризує здатність системи не пропускати загрози.

---

2. Рівень хибнопозитивних спрацювань (False Positive Rate, FPR)

$$FPR = \frac{FP}{FP + TN} \times 100\%$$

де:

FP — легітимні події, помилково визначені як RAT,

TN — коректно ідентифіковані нормальні події.

Надмірні хибнопозитивні спрацювання знижують якість роботи SOC та EDR.

3. Час реакції системи (Response Time, RT)

Показник включає:

- час виявлення,
- час блокування процесу,
- час ізоляції хоста.

Чим менше RT, тим менше шансів успішного виконання RAT своїх функцій.

4. Покриття векторів атак (Coverage, COV)

Визначається кількістю типів атак, які система здатна виявляти:

- встановлення персистентності,
- мережевий C2,
- кейлоггінг,
- ексфільтрація файлів,
- privilege escalation тощо.

5. Стійкість до обходу (Bypass Resistance, BR)

Оцінюється здатність системи протидіяти:

- обфускації,
- поліморфізму,
- модифікаціям RAT,
- rootkit-технікам,
- tampering-атакам.

6. Продуктивність (Performance Impact, PI)

Оцінює навантаження на:

- RAM,
- CPU,

- мережевий трафік.

Висновки до першого розділу

У першому розділі досліджено сучасні підходи до побудови систем захисту від RAT-атак, проаналізовано природу та класифікацію Remote Access Trojan, механізми їх роботи, канали розповсюдження та техніки приховування. Проведене порівняння сигнатурних, поведінкових, мережевих та машинних методів виявлення показало, що для боротьби з сучасними RAT необхідний комплексний багаторівневий підхід.

Формалізовано модель загроз для корпоративної інформаційної системи, визначено вимоги до сучасної системи захисту, а також сформульовано критерії ефективності, що дозволяють об’єктивно оцінювати результати виявлення та нейтралізації RAT у корпоративному середовищі.

Отримані результати створюють теоретичну основу для подальших досліджень, викладених у наступних розділах дипломної роботи, зокрема щодо реалізації методів пошуку, аналізу та розробки рекомендацій із застосування засобів захисту від RAT.

Розділ 2 ОСОБЛИВОСТІ ПОШУКУ, ВИЯВЛЕННЯ ТА АНАЛІЗУ

RAT У ІНФОРМАЦІЙНИХ СИСТЕМАХ

2.1 Методи дослідження поведінкових ознак RAT

Поведінкові ознаки є ключовим елементом для виявлення Remote Access Trojan, оскільки сучасні RAT активно використовують техніки обходу сигнатурних систем, шифрування та обфускацію. На відміну від статичного аналізу, поведінкові методи дозволяють виявити шкідливу активність на основі реальних дій шкідливого ПЗ у системі: змін у файловій структурі, системних викликів, мережевих операцій, модифікації реєстру та служб.

У цьому підрозділі розглянуто основні методи дослідження поведінкових ознак RAT, що застосовуються у системах моніторингу безпеки, EDR-рішеннях, пісочницях та інструментах динамічного аналізу.

2.1.1 Аналіз системних викликів

Аналіз системних викликів (system calls) є одним із найефективніших методів виявлення прихованої активності RAT. Оскільки будь-яка програма взаємодіє з ядром операційної системи через набір системних функцій, їх послідовність та параметри дозволяють визначити справжнє призначення процесу.

Основні аспекти аналізу системних викликів при дослідженні RAT:

1. Виявлення підозрілих патернів викликів

Типові RAT активно використовують виклики:

- для створення прихованих процесів (CreateProcess, NtCreateThreadEx);
- доступу до пам’яті інших процесів (OpenProcess, WriteProcessMemory);
- перехоплення подій клавіатури та миші (GetAsyncKeyState, SetWindowsHookEx);
- ініціації мережевих з’єднань (connect, send, recv).

Наявність нетипових або повторюваних послідовностей системних викликів може свідчити про шкідливу активність.

2. Кореляція системних викликів із поведінковими ланцюжками

RAT зазвичай виконують операції в певній послідовності (рис 1.2):

1. ініціалізація →
2. встановлення з’єднання →
3. збір даних →
4. передавання інформації.

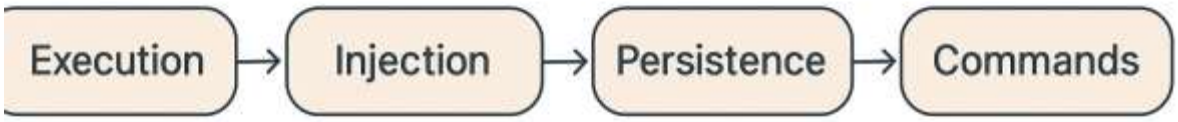


Рис. 1.2 Послідовність ініціалізації

Кожен етап має характерний набір системних викликів, що дозволяє будувати поведінкові ланцюги (behavior graphs).

3. Побудова профілю нормальної поведінки

У корпоративних мережах можливо формувати моделі «нормальної» активності для окремих сервісів. Будь-яке відхилення від профілю може бути ознакою RAT.

Переваги аналізу системних викликів:

- надзвичайно висока точність;
- можливість виявлення невідомих або модифікованих RAT;
- складність обходу з боку зловмисника.

Недоліки:

- високі обчислювальні витрати;

- потреба у спеціальному програмному забезпеченні (пісочниці, EDR).

2.1.2 Моніторинг змін у файловій системі

Файлова активність є ще одним важливим індикатором роботи RAT. Сучасні RAT намагаються створювати або модифікувати файли, що забезпечують персистентність, збір даних або приховування шкідливого ПЗ.

Основні напрями аналізу файлової активності:

1. Виявлення створення та модифікації підозрілих файлів

До типових ознак належать:

- створення виконуваних файлів у системних директоріях (C:\Windows\System32, AppData);
- розміщення тимчасових модулів у прихованих папках;
- регулярне оновлення файлів конфігурації RAT.

2. Аналіз діяльності щодо персистентності

RAT часто створюють:

- файли ярликів у папках автозапуску;
- DLL-бібліотеки;
- приховані файли, які автоматично запускаються при старті системи.

Виявлення таких дій є важливим для ранньої детекції.

3. Моніторинг операцій читання та копіювання файлів

Багато RAT включають модулі ексфільтрації даних. Підозрілими вважаються:

- масове читання файлів користувача;
- копіювання документів у нестандартні директорії;
- створення архівів із підозрілими атрибутами.

4. Аналіз прав доступу

Надання файлам або папкам нетипових прав доступу може свідчити про підготовку до прихованого керування.

Переваги підходу:

- дозволяє визначити механізми персистентності;
- ефективний проти широкого спектра RAT;
- простий у реалізації.

Недоліки:

- можливість хибнопозитивних спрацювань через легітимні процеси;
- залежність від глибини та частоти моніторингу.

2.1.3 Аналіз реєстру та служб

Реєстр Windows та служби операційної системи є одними з головних цілей RAT для закріплення у системі та контрольованого запуску. Тому їх аналіз є важливою частиною поведінкового моніторингу.

1. Аналіз ключів реєстру, пов'язаних із автозапуском

RAT найчастіше змінюють такі гілки реєстру:

- HKCU\Software\Microsoft\Windows\CurrentVersion\Run
- HKLM\...\RunServices
- HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon

Ключові ознаки:

- поява нових записів із підозрілими шляхами;
- заміна значень для системних процесів (наприклад, Shell, Userinit);
- посилання на файли у прихованих директоріях.

2. Моніторинг служб Windows

RAT можуть:

- створювати нові служби;
- змінювати конфігурацію існуючих;
- маскувати шкідливі служби під системні.

Аномалії включають:

- служби з випадковими назвами;
- служби, що запускають незнайомі виконувані файли;
- запуск служб у нетиповому режимі.

3. Виявлення нелегітимних драйверів та rootkit-компонентів

Складні RAT можуть завантажувати драйвери ядра для приховування:

- процесів,
- файлів,
- мережевих з’єднань.

Аналіз драйверів дозволяє виявити глибші рівні компрометації.

4. Кореляція реєстру, служб та файлових змін

Зміна реєстру + створення файлу + запуск процесу часто означають спробу персистентності. Кореляція таких подій значно підвищує точність виявлення RAT.

Переваги підходу:

- високий рівень детекції персистентних компонентів RAT;
- важко обійти навіть модифікованим шкідливим ПЗ;
- надає критичну інформацію для форензики.

Недоліки:

- орієнтованість на Windows-системи;
- потреба у глибокій аналітичній обробці подій.

2.2 Мережеві методи виявлення RAT

Мережеві методи відіграють ключову роль у виявленні RAT, оскільки більшість таких шкідливих програм потребують стабільного каналу зв’язку з сервером керування (Command and Control, C2). Відстеження та аналіз мережевого трафіку дозволяє виявити як активні фази роботи RAT, так і приховані спроби ініціалізації зв’язку, передачі даних, тунелювання або обходу мережевих політик.

У цьому підрозділі розглядаються основні підходи до мережевого моніторингу, що застосовуються у корпоративних системах безпеки, зокрема в IDS/IPS, SIEM, NDR-платформах та інструментах глибокого аналізу трафіку.

Таблиця 2.1

Порівняльна таблиця мережевих методів маскування та виявлення

Метод	Призначення	Опис	Стійкість	Виявлення

Domain Fronting	Маскування C2 через CDN	Трафік йде через Cloudflare/Akamai	Висока	JA3/HTTP аномалії
Reverse Connection	Приховування ініціації	Сервер підключається до клієнта	Середня	Незвичні вхідні підключення
Fast-Flux	Часте оновлення IP	Постійна зміна DNS	Висока	DNS anomalies
DGA (domain generation algorithms)	Генерація доменів	Рандомні домени	Дуже висока	ML-детектори
Proxy chaining	Багаторівневі проксі	C2 не видно напряму	Середня	Аномальна маршрутизація
Protocol Mimicry	Прикидання під HTTPS/SSH/SMB	Імітація легітимних протоколів	Середня	Титанічний DPI

2.2.1 Виявлення підозрілої C2-комунікації

C2-комунікація — це основний механізм, через який RAT отримує команди від зловмисника та передає викрадені дані. Тому виявлення аномалій у структурі та характері цього зв’язку є важливим завданням мережевої безпеки.

1. Аналіз доменів та IP-адрес, пов’язаних із C2

C2-сервери часто використовують:

- динамічні DNS-сервіси;
- новостворені або малорейтингові домени;

- IP-адреси, що належать хостинг-провайдерам із поганою репутацією;
- одноразові домени (burner domains).

Системи безпеки порівнюють домени з:

- відомими списками IOC (Indicators of Compromise);
- базами загроз (Threat Intelligence);
- моделями доменної ентропії (виявлення DGA — Domain Generation Algorithm).

2. Аналіз шаблонів взаємодії з C2

RAT зазвичай використовують нетипові для звичайного ПЗ шаблони запитів:

- регулярна періодичність звернень (beaconing),
- дуже низький або дуже високий інтервал між пакетами,
- стабільний розмір пакетів,
- нестандартні заголовки HTTP/HTTPS.

Виявлення «маячків» (beacons) є одним із найточніших методів детекції RAT у мережі.

3. Виявлення шифрованої або тунельованої комунікації

Багато RAT приховують трафік через:

- TLS/SSL-шифрування,
- HTTPS-тунелювання,
- використання нестандартних портів (наприклад, 443 для нефайлових протоколів),
- проксі-сервери або Tor-мережу.

Ознакою підозрілого з’єднання може бути:

- шифрування без SNI,
- вручну створений самопідписаний сертифікат,
- відсутність нормальної структури TLS-рукостискання.

4. Кореляція C2-трафіку з поведінкою хостів

Поєднання інформації про трафік і поведінку кінцевої системи (EDR + NDR) дозволяє значно підвищити точність виявлення RAT.

2.2.2 Виявлення аномалій у поведінці трафіку

На відміну від сигнатурних підходів, методи поведінкового аналізу мережевого трафіку орієнтуються на відхилення від нормальної активності в корпоративній мережі. Цей підхід є ефективним проти невідомих RAT та Zero-day кампаній.

1. Статистичний аналіз мережевих потоків

Основними параметрами, що підлягають моніторингу, є:

- обсяг переданих даних;
- кількість сесій;
- частота вихідних підключень;
- використання нетипових портувань;
- співвідношення вхідного та вихідного трафіку.

RAT часто формують:

- невеликі, але регулярні вихідні пакети,
- сплески активності під час передачі даних.

2. Виявлення аномальних патернів у часових рядах

Побудова моделей «нормальної» поведінки (наприклад, за допомогою EWMA, ARIMA або класифікації кластерів) дозволяє виявити:

- раптові піки активності,
- аномальні періоди тиші,
- нетипові цикли звернень.

3. Аналіз взаємодії між внутрішніми вузлами

RAT, що поширюються по мережі або виконують б lateral movement, залишають характерні сліди:

- нетипові запити через SMB, WMI, RDP;
- ініціація сесій між хостами, які ніколи не взаємодіяли раніше;
- підвищена активність внутрішнього сканування портів.

4. Методи моделювання графів мережевої взаємодії

Представлення трафіку у вигляді графа (вузли — хости, ребра — з’єднання) дозволяє виявити:

- аномальні з’єднання,
- нелегітимні маршрути,
- вузли, що ведуть підозрілу активність.

2.2.3 Deep Packet Inspection у боротьбі з RAT

Deep Packet Inspection (DPI) — це метод глибокого аналізу пакетів, який дозволяє вивчати не тільки заголовки, але й вміст переданих даних. DPI є ефективним інструментом для виявлення RAT-трафіку, особливо коли зловмисники намагаються маскувати комунікацію під легітимні протоколи.

1. Ідентифікація протоколів незалежно від порту

RAT часто працюють на «заглушених» портах, наприклад:

- HTTP через порт 8080,
- SSH через порт 80,
- користувацькі протоколи на порту 443.

DPI аналізує структуру пакета і визначає справжній протокол, навіть якщо порт нетиповий.

2. Виявлення ознак шкідливого протоколу або формату даних

DPI може аналізувати:

- структуру команд RAT,
- формат протоколів C2,
- характерні поля, включаючи довжину та ентропію.

Наприклад, багато RAT мають унікальні послідовності байтів у пакетах.

3. Виявлення прихованих каналів зв’язку

DPI дозволяє ідентифікувати:

- стеганографічні методи (дані, приховані у зображеннях або HTTP-полях),
- нестандартні методи тунелювання даних,
- шкідливі payload’и всередині легітимних пакетів.

4. Детектування шкідливого шифрування

Хоча DPI не завжди може розшифрувати TLS-трафік, воно здатне виявити:

- аномальні параметри шифрування,
- нестандартні обміни ключами,
- підозрілі сертифікати.

Переваги DPI:

- високий рівень точності;
- здатність виявляти складні RAT, що маскують трафік;
- виявлення протоколів незалежно від порту.

Недоліки:

- велике навантаження на мережеве обладнання;
- можливі конфіденційні ризики через аналіз вмісту пакетів;
- складність обробки великого обсягу шифрованого трафіку.

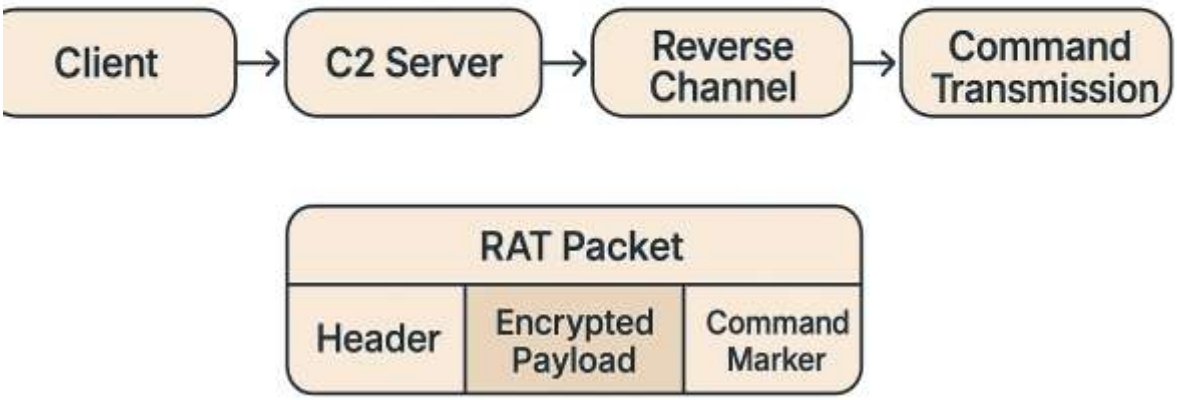


Рис. 2.2 Структура передачі пакетів RAT

2.3 Огляд та аналіз існуючих open-source RAT, доступних на GitHub

Open-source RAT (Remote Access Trojan), які доступні у відкритих репозиторіях GitHub, найчастіше використовуються дослідниками кібербезпеки

для аналізу шкідливих методів, тестування захисних систем та розробки сигнатур і поведінкових правил.

Завдяки відкритому вихідному коду такі інструменти дозволяють глибше зрозуміти механізми, якими користуються реальні шкідливі програми, але при цьому важливо наголосити: **аналіз програмного коду здійснюється виключно в освітніх та дослідницьких цілях**, без створення або використання шкідливого ПЗ.

У цьому підрозділі розглядаються чотири найвідоміші open-source RAT-інструменти:

AsyncRAT, QuasarRAT, Pupy, NJRat-open-source forks, — які часто згадуються у дослідницьких роботах та звітах Threat Intelligence.

2.3.1 Аналіз функціональних можливостей та архітектури (AsyncRAT, QuasarRAT, Pupy, NJRat forks)

1. AsyncRAT

Мова програмування: C# (.NET).
Типова архітектура: клієнт-сервер із TLS-шифруванням для захищеного каналу комунікації.

Функціональні можливості:

- віддалене керування файловою системою на уражених хостах;
- збір натискань клавіш (кейлогінг) та моніторинг активності користувача;
- виконання віддалених команд і скриптів;
- робота через зашифровані WebSocket-з’єднання, що забезпечує стелс-передачу даних;
- модульність архітектури та можливість підключення додаткових плагінів для розширення функціоналу.

Особливості:

- широке застосування шифрування для захисту переданих даних та конфігураційних файлів;
- гнучкі та складні механізми налаштування і взаємодії модулів, що підвищує стійкість до базових сигнатурних методів виявлення;

- можливість приховувати активність у системі та адаптувати поведінку під конкретні умови роботи мережі, що ускладнює її детектування традиційними антивірусними рішеннями.

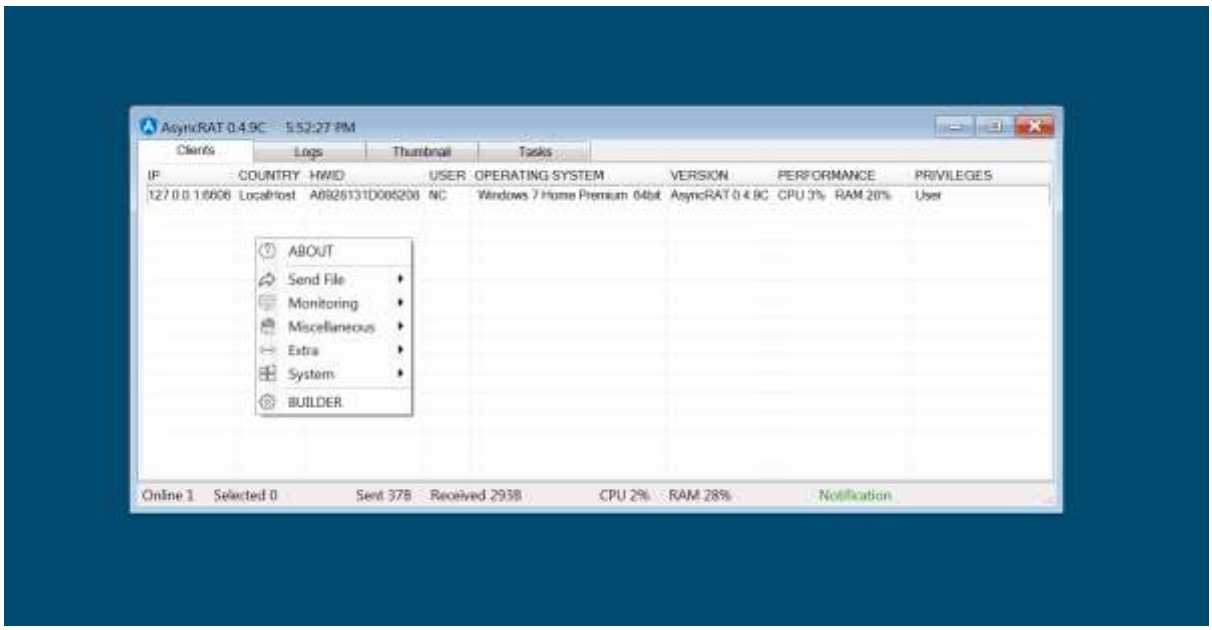


Рис. 2.3 Інтерфейс AsyncRAT

2. QuasarRAT

Мова програмування: C#. **Архітектура:** легкий багатопотоковий клієнт із використанням TCP-комунікації для взаємодії між серверною та клієнтською частинами. Архітектура оптимізована під високу швидкість обробки команд, низьке споживання ресурсів і стабільну роботу у мережах з мінливою пропускною здатністю.

Функціональні можливості:

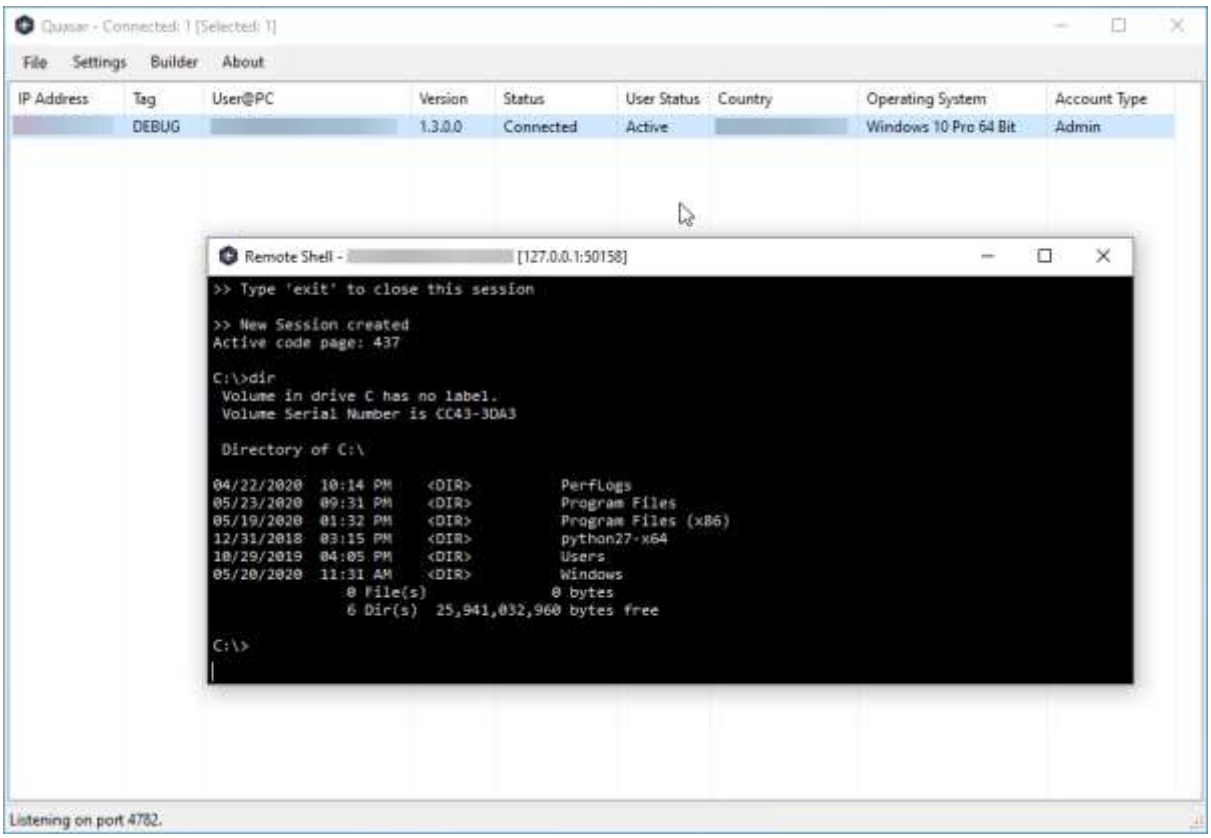
- **збір інформації про систему**, включаючи загальні дані про ОС, мережеву конфігурацію та перелік активних процесів;
- **керування процесами** з можливістю дистанційного завершення або запуску нових задач;
- **перегляд екрана в реальному часі**, що забезпечує повний моніторинг активності користувача;
- **передача файлів у двосторонньому режимі**, що дозволяє завантажувати й отримувати матеріали з ураженої системи;

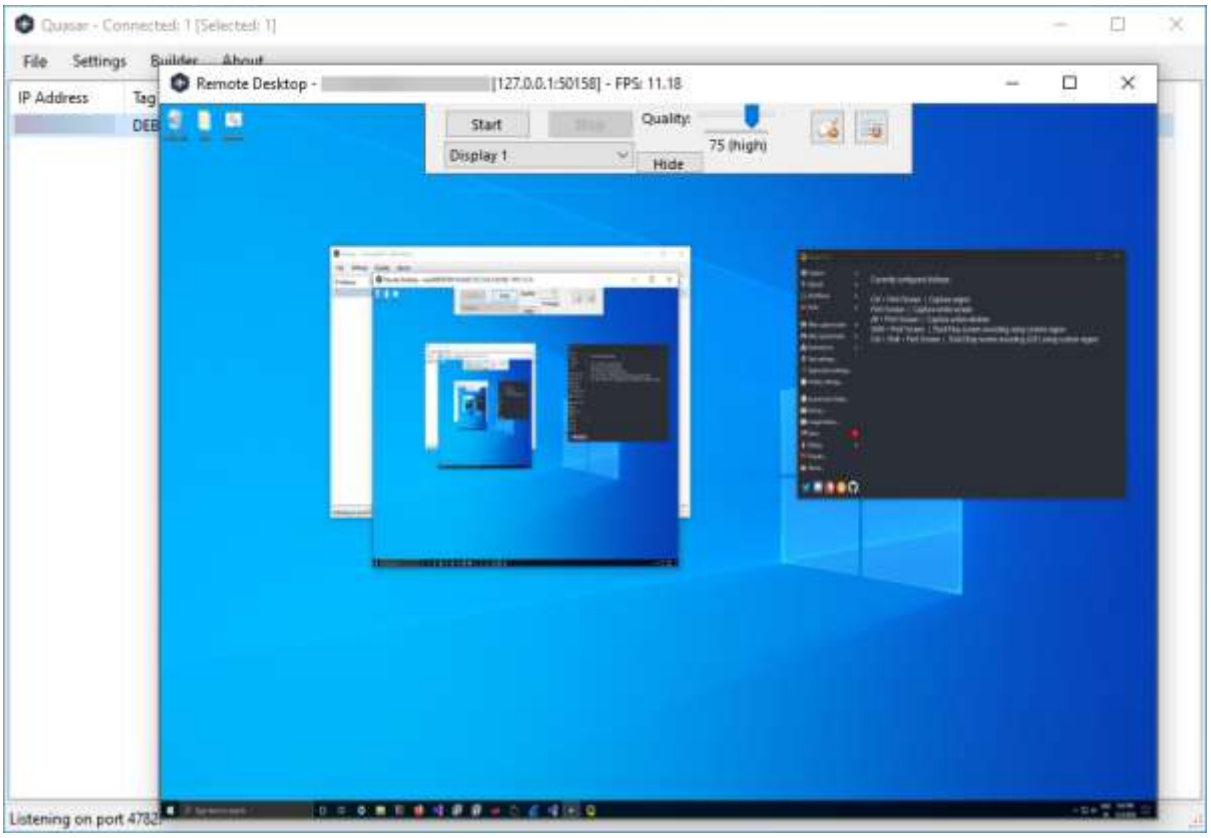
- **обхід базових антивірусних механізмів** за рахунок мінімальної кількості зовнішніх залежностей і простої структури виконуваного файлу, що ускладнює сигнатурне виявлення.

Особливості:

QuasarRAT вирізняється **простотою архітектури**, чітким розподілом модулів та максимальною прозорістю реалізації. Це робить його одним із найпопулярніших інструментів для академічних досліджень та аналізу принципів роботи RAT-класу. У порівнянні з більш складними проєктами він демонструє мінімалістичний, але ефективний підхід до реалізації базових функцій віддаленого доступу.

Такі характеристики дозволяють використовувати QuasarRAT як показовий приклад у дослідженнях поведінкових ознак RAT, а також при формуванні критеріїв детектування на рівні трафіку, процесів та взаємодії з ОС.


Quasar - Connected: 1 [Selected: 1]
File Settings Builder About
IP Address Tag User@PC Version Status User Status Country Operating System Account Type
DEBUG 1.3.0.0 Connected Active Windows 10 Pro 64 Bit Admin
Remote Shell - [127.0.0.1:50158]
>> Type 'exit' to close this session
>> New Session created
Active code page: 437
C:\>dir
Volume in drive C has no label.
Volume Serial Number is CC43-3DA3
Directory of C:\
04/22/2020 10:14 PM <DIR> PerfLogs
05/23/2020 09:31 PM <DIR> Program Files
05/19/2020 01:32 PM <DIR> Program Files (x86)
12/31/2018 03:15 PM <DIR> python27-x64
10/29/2019 04:05 PM <DIR> Users
05/20/2020 11:31 AM <DIR> Windows
0 File(s) 0 bytes
6 Dir(s) 25,941,032,960 bytes free
C:\>
Listening on port 4782.


Quasar - Connected: 1 [Selected: 1]
File Settings Builder About
IP Address
Remote Desktop - [127.0.0.1:50158] - FPS: 11.18
Start
Quality:
75 (high)
Display 1
Hide
Listening on port 4782

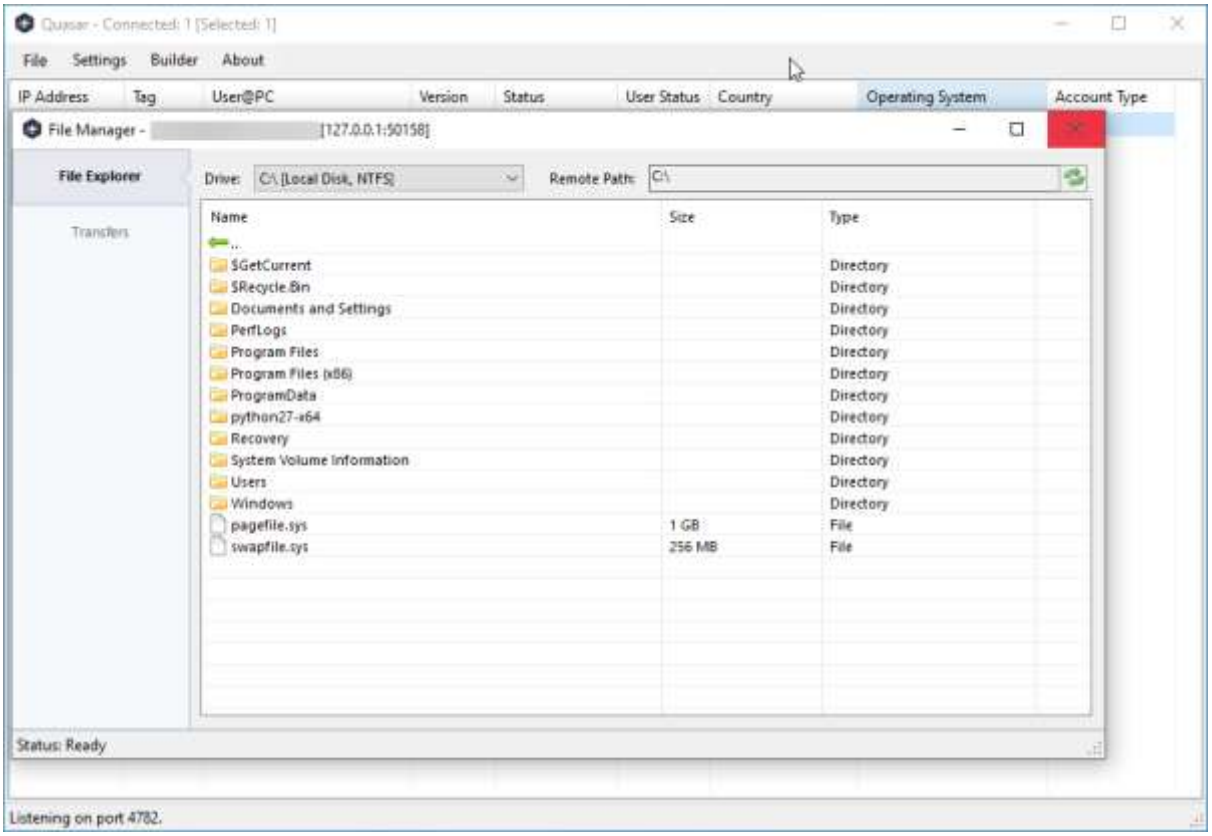


Рис. 2.4 Інтерфейс QuasarRAT

3. Pupy (Python/C)

Мова програмування: Python із використанням окремих високопродуктивних компонентів, написаних на C. Така комбінація забезпечує гнучкість у розробці та оптимізовану роботу критичних модулів.

Архітектура:

Pupy являє собою **повноцінний багатофункціональний фреймворк**, орієнтований на розширення та динамічну інтеграцію додаткових можливостей. Його модульна структура дозволяє розробникам та дослідникам легко адаптувати функціонал під різні сценарії використання. Відмінною рисою є підтримка бездискових механізмів виконання — модулі можуть завантажуватися та працювати повністю в оперативній пам’яті.

Функціональні можливості:

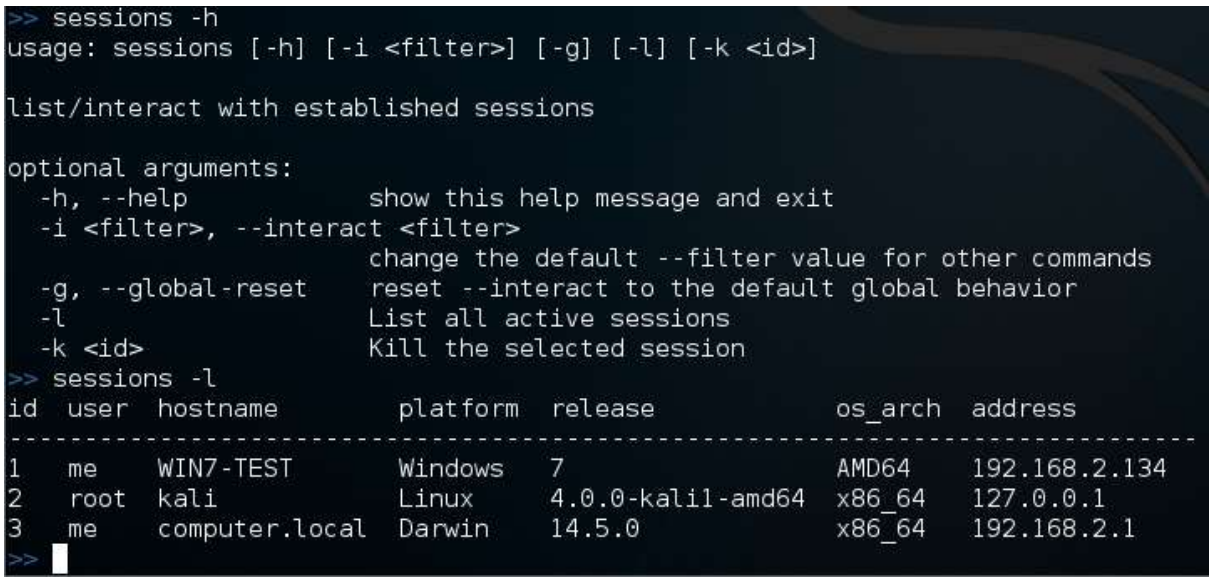
- **мультиплатформенність:** підтримка Windows, Linux, macOS, Android, що робить Pupy одним із найбільш універсальних інструментів у своєму класі;

- **підтримка різних транспортів:** HTTP/HTTPS, SSH, TCP, WebSocket, що забезпечує широкі можливості для побудови каналів зв’язку;
- **динамічне завантаження модулів у пам’ять**, без створення файлів на диску, що дозволяє проводити дослідження без зміни файлової системи;
- **інтеграція з інструментами постексплуатації**, що включає засоби збору інформації, аналізу середовища та взаємодії з ОС на розширеному рівні.

Особливості:

Завдяки своїй архітектурі та гнучкості Pupy часто застосовується в академічних і лабораторних дослідженнях, присвячених аналізу шкідливого ПЗ, механізмів приховування та методів побудови мережевих каналів керування. Його плагінна природа та відкритість коду дозволяють вивчити сучасні практики побудови складних RAT-рішень, включно з реалізацією stealth-механізмів та мультиплатформенного інструментарію.

Pupy є важливим прикладом для аналітиків, оскільки демонструє, як багатокомпонентна архітектура може забезпечити високу адаптивність, розширюваність та стійкість до традиційних методів виявлення.



```
>> help
Available commands :

- clients        List connected clients
- exit           Quit Pupy Shell
- help           show this help
- jobs           manage jobs
- list_modules   List available modules with a brief description
- python         start interacting with the server local python interpreter (for debugging
- read           execute a list of commands from a file
- run            run a module on one or multiple clients
- info           get some informations about one or multiple clients
- ps             list processes
- contest
- migrate        Migrate pupy into another process using reflective DLL injection
- exec           execute shell commands on a remote system
- pyexec         execute python code on a remote system
```

```
>> run pyexec
usage: pyexec [-h] (--file <path> | -c <code string>)
pyexec: error: one of the arguments --file -c/--code is required
>> run pyexec -c 'import platform; print platform.uname()[0:3]'
#>#>  PupyClient(id=1, user=me, hostname=computer.local, platform=Darwin)  <#<#
('Darwin', 'computer.local', '14.5.0')

#>#>  PupyClient(id=2, user=me, hostname=WIN-TEST, platform=Windows)  <#<#
('Windows', 'WIN-TEST', '7')

#>#>  PupyClient(id=3, user=root, hostname=kali, platform=Linux)  <#<#
('Linux', 'kali', '4.0.0-kali1-amd64')
>>
```

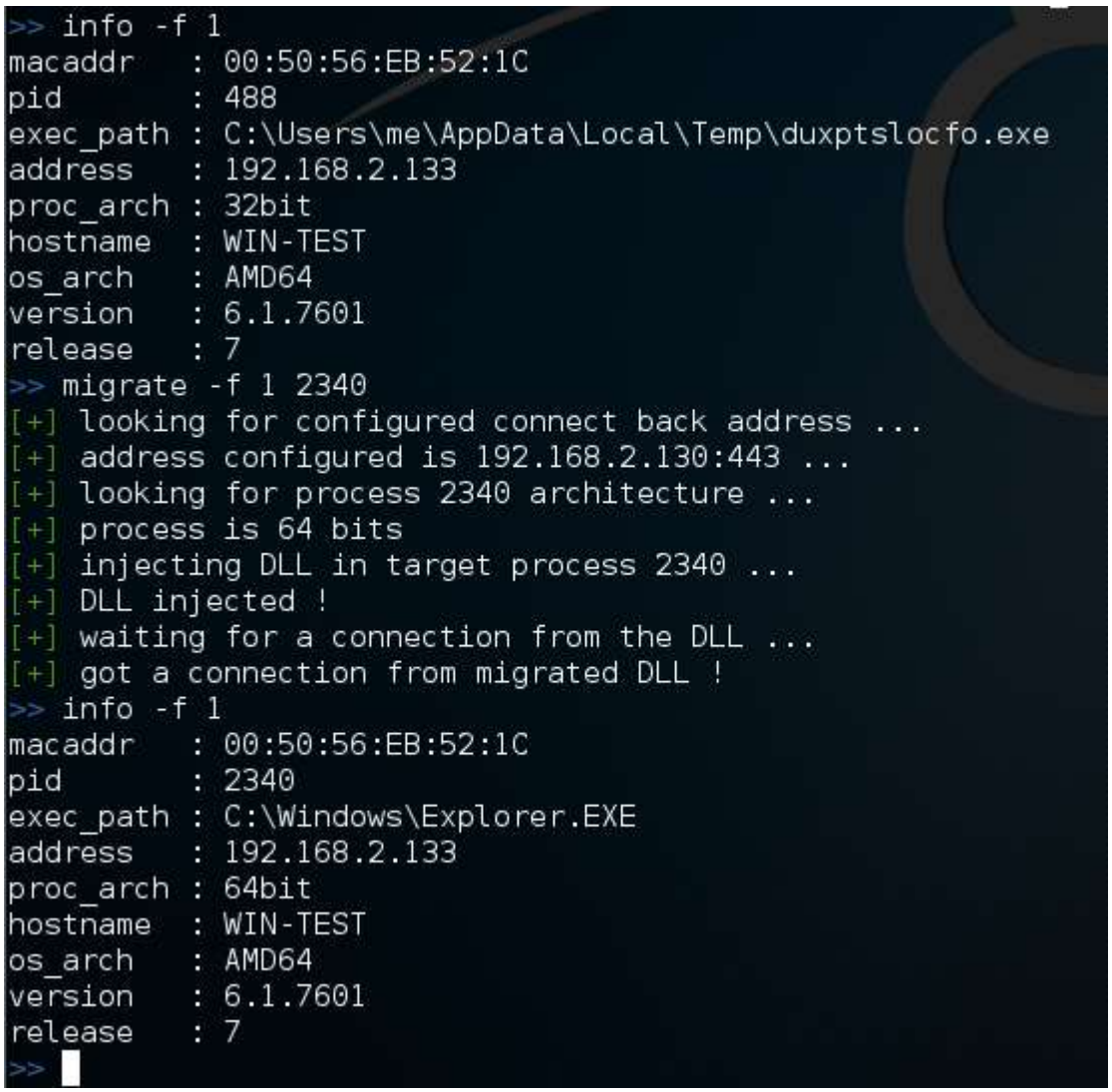


Рис. 2.5 Інтерфейс Pupy

4. NJRat (open-source forks)

Мова програмування: VB.NET

Використання цієї мови зумовило появу значної кількості модифікованих збірок, оскільки VB.NET є доступною для початківців та легко піддається реверс-інжинірингу й подальшим змінам.

Архітектура:

NJRat реалізує **класичну модель "клієнт–С2-сервер"**, у якій клієнтська частина встановлює постійний канал зв’язку із сервером через простий механізм обміну повідомленнями. Архітектура відносно лінійна, без складної модульності, що робить форки легко модифікованими. Канал зв’язку зазвичай використовує TCP-протокол, з мінімальним шифруванням або його відсутністю, що відображає характерну простоту цього сімейства RAT.

Функціональні можливості:

NJRat та його численні форки зазвичай включають набір базових, але поширених функцій, притаманних класичним RAT-інструментам:

- **кейлогінг** — фіксація натискань клавіш;
- **запис відео та аудіо** через доступ до вебкамери та мікрофона;
- **керування файлами та процесами**, що охоплює перегляд, копіювання, видалення, запуск програм;
- **мережевий сканер**, що дозволяє здійснювати базове зондування локальної мережі;
- **використання різних типів проксі-підключень**, що забезпечує гнучкість у встановленні каналу зв'язку та обході простих мережевих обмежень.

Особливості:

Однією з ключових рис NJRat є те, що навколо нього сформувалась велика екосистема **модифікованих форків**. Багато з них з'явилися у відкритому доступі, що дозволяє:

- простежити **еволюцію шкідливого ПЗ** у межах одного сімейства;
- порівняти різні реалізації механізмів C2-комунікацій;
- дослідити, як автори додають або прибирають функціонал;
- вивчити способи спрощеного обходу сигнатурних антивірусів;
- оцінити, як змінюються прийоми обфускації в залежності від генерації форків.

З аналітичної точки зору NJRat є цінним прикладом для досліджень, оскільки демонструє **мінімалістичний, але робочий підхід до побудови RAT-архітектур**, а його численні модифікації — корисний матеріал для аналізу тенденцій розвитку шкідливого ПЗ у відкритому середовищі.

Рис. 2.6 Інтерфейс NJRat

Таблиця 2.1

Порівняльна таблиця Open Source RAT

Характеристика	**AsyncRAT**	**QuasarRAT**	**Pupy (Python/C)**	**NJRat (open-source forks)**
Мова розробки	C# (.NET)	C# (.NET)	Python + C	VB.NET / C#
Платформи	Windows	Windows	Windows, Linux, macOS, Android	Windows
Архітектура	Client–Server, асинхронний C2	Client–Server	Модульна, децентралізована, memory-resident	Client–Server

Тип C2 з’єднання	TCP + WebSocket	TCP	TLS/SSL, reverse-connection	TCP
Шифрування трафіку	AES + RSA	AES	TLS (OpenSSL)	Просте XOR або AES (залежно від форку)
Механізми приховування	Обфускація, злиття з легітимними процесами	Мінімальні (легше детектується)	Fileless режим, виконання у пам’яті	Примітивні, легко виявляється
Персистентність	Реєстр, планувальник, служби	Реєстр	Складні сценарії автозапуску	Реєстр
Модулі	кейлогер, файловий менеджер, RDP/скріншот, криптор	кейлогер, скріншоти, файлові операції	shell-доступ, privilege escalation, тунелювання	кейлогер, скріншоти, файлові операції
P2P можливості	Ні	Ні	Частково (через тунелювання)	Ні
Складність аналізу	Середня (через обфускацію)	Низька	Висока (fileless, Python-скрипти,	Низька

			шифрування)	
Ідентифікаційні ознаки	Специфічні WebSocket-патерни, сигнатури AES-ключів	Характерні .NET збірки	TLS-трафік нетипового формату, Python-процеси	Унікальні IP-C2, часті XOR-рядки
Стійкість до EDR/AV	Середня	Низька	Висока	Дуже низька
Активність у GitHub	Висока	Висока	Висока	Низька
Типові сценарії використання	Стелс-керування Windows	Легке адміністрування та атаки	Пентестинг, APT-операції	Масові ботнет-атакувальні кампанії
Рівень загрози	Високий	Середній	Дуже високий	Середній–низький

2.3.2 Порівняльний аналіз підходів до приховування та стійкості RAT

1. Техніки ухилення від виявлення

У проаналізованих RAT найчастіше зустрічаються такі підходи:

- **Обфускація коду** (AsyncRAT, NJRat forks) Використання .NET-обфускаторів приховує логіку виконання та ускладнює реверс-інжиніринг.

- **Динамічне завантаження модулів** (Pupy)

Модулі завантажуються в оперативну пам’ять без створення файлів — це дозволяє обходити сигнатурні сканери.

- **Шифрування C2-трафіку** (AsyncRAT, Pupy)

RAT використовують TLS/SSL, нестандартні алгоритми шифрування або власні протоколи.

- **Маскування під легітимні процеси**

Деякі форки NJRat та QuasarRAT намагаються імітувати системні процеси Windows.

2. Мережеві техніки стійкості

- **Періодичне маячкування (beaconing)** з нестандартними інтервалами.
- **Використання резервних C2-серверів** (fallback-infrastructure).
- **Динамічні DNS-адреси**, що дозволяють швидко змінювати місцезнаходження сервера керування.
- **Маскування під легітимні протоколи**: HTTP/HTTPS-тунелі, WebSocket-трафік.

3. Архітектурні елементи, що підвищують стійкість до аналізу

- модульність, що дозволяє швидко змінювати функціонал;
- використання пам’яткоорієнтованих механізмів (Pupy);
- мінімалізм та низька кількість залежностей (QuasarRAT) — ускладнює детектування за статичними ознаками.

2.3.3. Ідентифікація експлуатаційних ознак RAT, корисних для побудови системи захисту

На основі дослідження відкритих проєктів можна виокремити низку **стійких експлуатаційних ознак**, що допомагають будувати системи виявлення у корпоративних мережах.

1. Типові сесії C2-зв’язку

- регулярні запити малої довжини;
- повторювані пакети;

- використання TLS без SNI;
- аномальні сертифікати.

2. Типові артефакти на рівні ОС

(актуально для більшості розглянутих RAT):

- зміни у Run/RunOnce-гілках реєстру;
- поява специфічних служб або Scheduled Tasks;
- нетипові DLL-інжекції в легітимні процеси;
- створення файлів конфігурації у тимчасових каталогах.

3. Поведінкові індикатори

- систематичне опитування процесів;
- запити на скринінг дисплея;
- нетипове використання API для роботи з камерою/мікрофоном;
- виклики криптографічних бібліотек у нетипових контекстах.

4. Ознаки мережевої активності

- доступ до підозрілих або новостворених доменів;
- багатопоточні outbound-з’єднання;
- невідповідність між User-Agent та характером програмної активності;
- трафік, маскований під легітимні служби.

Висновки до другого розділу

У другому розділі було досліджено ключові підходи до виявлення RAT у корпоративних інформаційних системах, включно з поведінковими, системними, мережевими та глибокими методами аналізу трафіку.

Особлива увага приділена огляду open-source RAT, оскільки саме вони дозволяють проаналізувати найпоширеніші техніки приховування та стійкості, що використовуються зловмисниками. У результаті дослідження сформовано перелік характерних експлуатаційних ознак, які доцільно застосовувати під час побудови систем детектування, включаючи EDR/IDS/NDR-рішення.

Зібрані дані стануть основою для формування вимог і розроблення комплексної системи захисту від RAT, що буде розглянуто в третьому розділі.

Розділ 3 ОБГРУНТУВАННЯ ВИБОРУ ТА РОЗРОБКА РЕКОМЕНДАЦІЙ І МОДЕЛІ ЗАХИСТУ ВІД RAT

3.1 Вибір оптимальних методів виявлення та протидії RAT

Розробка ефективної системи захисту від Remote Access Trojan у корпоративному середовищі потребує обґрунтованого добору методів детектування, аналізу та локалізації шкідливої активності. На підставі результатів попередніх розділів було визначено, що найбільш ефективним є поєднання мережевих та поведінкових підходів, доповнених механізмами автоматизованого аналізу загроз.

Вибір технологій здійснюється з урахуванням особливостей роботи сучасних RAT, їх здатності обходити статичні та сигнатурні методи захисту, а також вимог до масштабованості та сумісності у корпоративній інфраструктурі.

3.1.1 Обґрунтування вибору індикаторів компрометації (IoC)

Індикатори компрометації (Indicators of Compromise) становлять основу процесу виявлення початкових проявів RAT у корпоративній мережі. Аналіз відкритих репозиторіїв RAT та типових випадків компрометації дозволив виділити такі групи IoC:

1. Файлові та системні IoC

Виявлення змін у критичних ділянках файлової системи та реєстру є одним із первинних джерел інформації про можливу активність RAT. До типових індикаторів належать:

- створення прихованих виконуваних файлів у каталозі AppData, Temp, Roaming;
- поява конфігураційних файлів RAT;
- додавання ключів автозапуску у Run/RunOnce;
- реєстрація нових служб або планувальників завдань.

Ці IoC формуються на основі аналізу змін, характерних для AsyncRAT, QuasarRAT, NJRat та інших представників цього класу ПЗ.

2. Мережеві IoC

Оскільки RAT потребує постійного зв’язку з командно-контрольним (C2) сервером, ключовими є:

- нетипові outbound-з’єднання у нестандартні часи;
- періодичні пакети малої довжини (beaconing);
- звернення до динамічних або щойно зареєстрованих доменів;
- використання TLS-трафіку без SNI або з аномальними сертифікатами;
- тривалі постійні TCP-з’єднання.

Мережеві IoC є одними з надійніших ознак, оскільки більшість RAT не можуть функціонувати без C2-з’єднання.

3. Поведінкові IoC

Вони включають аномальні дії на рівні ОС:

- нетипове створення процесів та потоків;
- регулярні виклики API, пов’язаних з кейлоггінгом, доступом до камери або скриншотів;
- різке збільшення використання ресурсів без відповідної активності користувача;
- інжекція DLL у системні процеси.

Поведінкові індикатори є критичними для виявлення нових або модифікованих версій RAT, які маскують свій код, але не можуть приховати характерні моделі поведінки.

4. Комбіновані IoC (кореляційні)

Поєднання кількох джерел (файлових, мережевих, поведінкових) дозволяє зменшити кількість хибних спрацювань та сформувати точніші правила детектування.

Такі IoC є основою для сучасних EDR/NDR-рішень, що використовують кореляційні механізми.

3.1.2 Вибір методів аналізу трафіку та поведінки

З огляду на складність сучасних RAT та їхню здатність використовувати шифрування, традиційні підходи (наприклад, сигнатурні) є недостатними. Тому для корпоративної системи захисту було обрано комбінований підхід, що включає:

1. Аналіз мережевого трафіку (NDR-підхід)

Технологічні компоненти:

- аналіз метаданих трафіку (flow-based detection);
- визначення нетипових шаблонів beaconing;
- DPI (Deep Packet Inspection) для виявлення аномалій у протоколах;
- застосування сигнатур для відомих C2-протоколів.

Цей метод дозволяє виявити RAT навіть при шифруванні пакету, оскільки метадані, таймінги та патерни залишаються характерними.

2. Поведінковий аналіз процесів (Endpoint-підхід)

Виявлення RAT на рівні кінцевих точок ґрунтується на:

- аналізі API-викликів;
- моніторингу файлової системи;
- контролі змін у реєстрі;
- відстеженні підозрілої активності процесів.

Саме цей підхід ефективний проти шкідливих програм, які модифікують свій код або використовують поліморфізм.

3. Кореляційний аналіз у SIEM

SIEM-системи дозволяють:

- поєднувати журнали подій із кількох джерел;
- створювати складні правила детектування на основі IoC та поведінкових патернів;
- виявляти активність RAT навіть при відсутності явних ознак компрометації.

Кореляція подій є критично важливою для швидкого виявлення та ескалації інцидентів.

4. Застосування алгоритмів машинного навчання

ML-підходи додаються як додатковий рівень детектування та дозволяють:

- виявляти невідомі RAT-навантаження на основі аномальної поведінки;
- класифікувати мережевий трафік за характерними патернами;
- аналізувати багатовимірні поведінкові ознаки.

ML-моделі особливо ефективні у виявленні RAT зі зміненою архітектурою чи новими механізмами приховування трафіку.

3.1.3 Формування вимог до автоматизованої системи захисту

Для побудови ефективної системи протидії RAT у корпоративному середовищі сформульовано такі ключові вимоги:

1. Функціональні вимоги

Система має забезпечувати:

- автоматизоване виявлення файлових, мережевих та поведінкових IoC;
- класифікацію та пріоритизацію інцидентів;
- можливість блокування або ізоляції заражених хостів;
- журналювання та централізований збір логів;
- автоматичний аналіз C2-активності.

2. Вимоги до продуктивності

- мінімальний вплив на роботу кінцевих станцій;
- здатність обробляти великі обсяги журналів у реальному часі;
- масштабованість під корпоративну мережу різного розміру.

3. Вимоги до інтеграції

Система має підтримувати інтеграцію з:

- SIEM;
- EDR/NDR;
- мережевими засобами моніторингу;
- Active Directory та інфраструктурою сертифікатів;
- хмарними сервісами (за потреби).

4. Вимоги до безпеки

- надійний контроль доступу;
- збереження цілісності журналів;
- захист телеметрії від підробки;
- можливість аудиту дій адміністратора.

5. Вимоги до аналітики та звітності

- візуалізація поведінкових моделей;
- побудова графів взаємодії процесів та хостів;
- автоматичне формування звітів про потенційні компрометації.

3.2 Розробка моделі комплексної системи захисту від RAT

Розробка ефективної моделі захисту від Remote Access Trojan (RAT) у корпоративних мережах ґрунтується на інтеграції мережевого, поведінкового та аналітичного підходів. Мета моделі — забезпечити раннє виявлення, локалізацію, нейтралізацію та подальший аналіз загроз RAT, мінімізуючи ризики для бізнес-процесів та інформаційної безпеки організації.

Система побудована за модульним принципом і включає три основні компоненти: виявлення, моніторинг та реагування.

3.2.1 Архітектура системи

Архітектура комплексної системи захисту від RAT передбачає багаторівневу структуру, що включає:

1. Рівень кінцевих точок (Endpoint layer)

- встановлення агентів EDR на робочих станціях та серверах;
- збір поведінкових даних про процеси, файлову систему та реєстр;
- відправка телеметрії на центральний сервер.

2. Мережевий рівень (Network layer)

- розгортання NDR-сенсорів для аналізу мережевого трафіку;
- детектування аномальних з’єднань та підозрілої C2-активності;
- використання DPI для виявлення прихованих каналів передачі даних.

3. Центральний аналітичний рівень (Analytics layer)

- кореляція даних з EDR, NDR, SIEM;

- застосування правил та алгоритмів машинного навчання для виявлення аномалій;
- формування пріоритетів інцидентів та рекомендацій для реагування.

4. Інтерфейс керування та звітності (Management & Reporting)

- панель адміністрування та моніторингу;
- генерація звітів про підозрілу активність;
- інтеграція з системами оповіщення та внутрішньої комунікації.

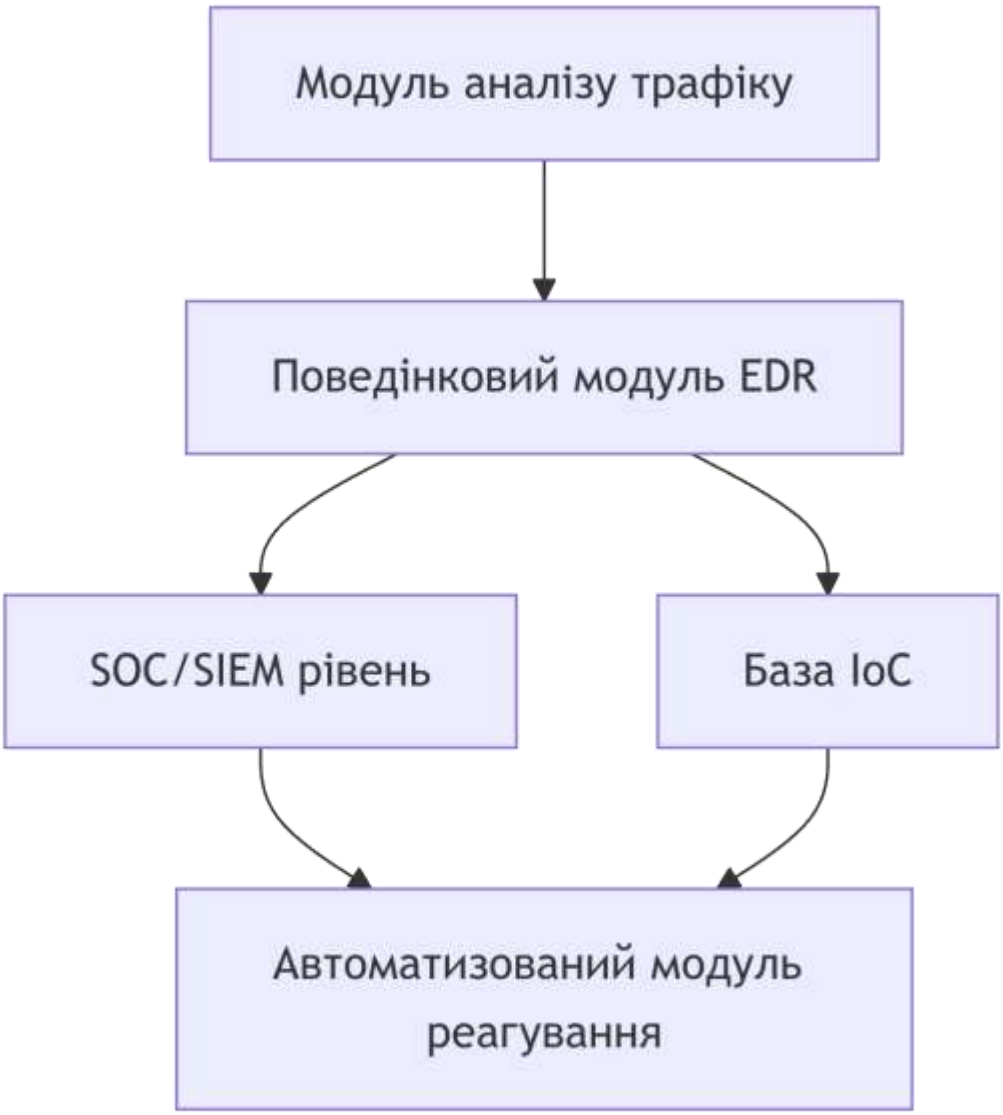


Рис. 3.1 Архітектура моделі комплексної системи захисту від RAT

3.2.2 Механізми взаємодії модулів (виявлення, моніторинг, реагування)

Для ефективної роботи модульна система використовує наступні механізми взаємодії:

1. Виявлення

- агенти EDR та сенсори NDR збирають первинні дані про поведінку процесів, мережеву активність, зміни у файловій системі та реєстрі;

- сигнатурні та поведінкові алгоритми обробляють дані для первинної класифікації інцидентів;
- кореляційний механізм на аналітичному рівні визначає комплексні патерни RAT.

2. Моніторинг

- постійний збір та аналіз логів у режимі реального часу;
- побудова графів взаємодії процесів та мережевих потоків;
- інтеграція з SIEM для централізованого відстеження подій та аномалій;
- адаптивне навчання моделей машинного навчання на основі накопиченої телеметрії.

3. Реагування

- автоматичне ізолювання ураженого хоста від корпоративної мережі;
- блокування підозрілих мережевих з’єднань;
- повідомлення адміністратора про критичні інциденти;
- запуск скриптів або процедур ліквідації шкідливих компонентів (remediation);
- документування подій для подальшого аналізу та навчання системи.

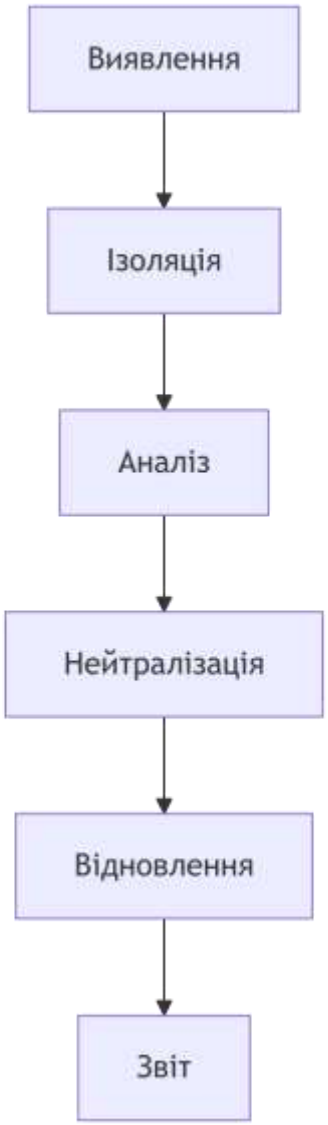


Рис. 3.2 Механізми взаємодії модулів

3.2.3 Регламент реагування на інциденти, пов’язані з RAT

Регламент визначає послідовність дій у разі виявлення RAT:

1. Ідентифікація

- автоматичне сповіщення про підозрілу активність;
- первинна оцінка ризику та підтвердження IoC.

2. Локалізація та ізоляція

- визначення уражених хостів та сегментів мережі;

- тимчасове відключення або ізоляція для обмеження поширення;
- блокування C2-з’єднань.

3. Нейтралізація

- завершення підозрілих процесів;
- видалення шкідливих файлів і служб;
- відновлення змінених налаштувань реєстру.

4. Аналіз та звітність

- детальний форензичний аналіз інциденту;
- формування звітів для керівництва та служби безпеки;
- оновлення правил детектування та IoC для запобігання повторних атак.

5. Постінцидентні дії

- перевірка суміжних систем на наявність заражень;
- оцінка ефективності системи реагування;
- навчання персоналу та оптимізація процесів моніторингу.

Модель комплексної системи забезпечує **багаторівневий захист**, що поєднує проактивне виявлення, моніторинг поведінки, автоматизоване реагування та аналітичний контроль інцидентів RAT. Вона є масштабованою та може інтегруватися з існуючими корпоративними інструментами безпеки.

3.3 Практичні рекомендації щодо впровадження системи захисту від RAT

Впровадження комплексної системи захисту від Remote Access Trojan (RAT) у корпоративних інформаційних системах потребує поєднання **організаційних, технічних та навчально-освітніх заходів**. Такий підхід дозволяє не тільки ефективно виявляти та нейтралізувати шкідливі програми, а й підвищувати загальний рівень інформаційної безпеки організації.

3.3.1 Організаційні заходи

Організаційні заходи спрямовані на формування внутрішніх політик та процедур, які забезпечують комплексний контроль за інформаційною безпекою:

1. **Впровадження політики безпеки**

- визначення відповідальних осіб за безпеку інформаційних систем;
- регламентування правил використання корпоративних пристроїв та мережі;
- встановлення процедури реагування на інциденти.

2. **Процедури моніторингу та контролю**
   - регулярна перевірка систем на наявність IoC та поведінкових аномалій;
   - централізоване ведення журналів подій та контроль їхньої цілісності;
   - встановлення регулярних аудитів безпеки.
3. **Управління доступом та обліковими записами**
   - розмежування прав доступу до критичних систем;
   - контроль використання адміністративних облікових записів;
   - застосування принципу «найменших привілеїв» (least privilege).

3.3.2 Технічні заходи

Технічні заходи забезпечують практичне впровадження методів виявлення та нейтралізації RAT:

1. **Впровадження багаторівневої системи захисту**
   - інтеграція EDR (Endpoint Detection and Response) та NDR (Network Detection and Response);
   - використання SIEM для централізованого аналізу та кореляції подій;
   - налаштування IDS/IPS для виявлення мережевих аномалій.
2. **Контроль мережевого трафіку**
   - моніторинг вихідних та внутрішніх з’єднань;
   - застосування DPI для виявлення прихованих каналів передачі даних;

- блокування підозрілих доменів, IP та портів.

3. **Захист кінцевих точок та серверів**
   - встановлення антивірусного ПЗ з підтримкою поведінкових методів детектування;
   - обмеження можливості виконання підозрілих скриптів і процесів;
   - регулярне оновлення ОС та програмного забезпечення для усунення вразливостей.
4. **Резервування та відновлення**
   - регулярне створення резервних копій критичних даних;
   - тестування процедур відновлення після інцидентів;
   - ізоляція та очищення уражених систем.

3.3.3 Навчання персоналу та політики безпеки

Людський фактор є однією з основних причин успіху атак RAT. Для мінімізації ризиків необхідно:

1. **Навчання персоналу**
   - проведення тренінгів із безпечного використання корпоративних ресурсів;
   - інформування про сучасні методи соціальної інженерії та фішингу;
   - навчання ідентифікуванню підозрілих повідомлень та вкладень.
2. **Регулярні тестування та симуляції атак**
   - проведення навчальних сценаріїв RAT-атаки у контрольованому середовищі;
   - перевірка готовності персоналу та IT-систем реагувати на інциденти.
3. **Формування культури інформаційної безпеки**
   - регулярні оновлення внутрішніх політик та процедур;

- наголошення на важливості дотримання правил доступу та обліку;
- заохочення повідомлення про підозрілу активність без страху покарання.

Комплексне впровадження системи захисту від RAT має ґрунтуватися на **синергії організаційних, технічних та навчальних заходів**. Таке поєднання дозволяє:

- підвищити точність виявлення шкідливих програм;
- мінімізувати ризики поширення RAT у корпоративній мережі;
- сформувати культуру безпечної поведінки серед співробітників;
- забезпечити швидке реагування та відновлення після інцидентів.

Висновки до третього розділу

У третьому розділі було обґрунтовано вибір методів та розроблено модель комплексної системи захисту від Remote Access Trojan (RAT) у корпоративних мережах. На основі проведеного аналізу та дослідження open-source RAT сформовано наступні ключові висновки:

1. **Вибір індикаторів компрометації (IoC)**

Було визначено критичні групи IoC, включаючи файлові, поведінкові та мережеві ознаки. Застосування комбінації цих індикаторів забезпечує більш точне виявлення RAT і зменшує кількість хибних спрацювань.

2. **Оптимальні методи виявлення RAT**

Найбільш ефективним є поєднання:

- аналізу мережевого трафіку та аномалій C2-з’єднань;
- поведінкового аналізу кінцевих точок;
- кореляційного аналізу даних у SIEM та використання алгоритмів машинного навчання для виявлення нових або модифікованих RAT.

3. **Розроблена архітектура системи**

Модель комплексного захисту включає багаторівневу архітектуру:

- кінцеві точки (EDR);
- мережеві сенсори (NDR);
- аналітичний рівень для кореляції та обробки даних;
- інтерфейс управління та звітності.

Така структура забезпечує раннє виявлення, локалізацію та нейтралізацію загроз RAT.

4. **Механізми взаємодії модулів**

Виявлення, моніторинг та реагування інтегровані у єдину систему, що дозволяє автоматично ізолювати уражені хости, блокувати підозрілий трафік та документувати інциденти для подальшого аналізу.

5. **Регламент реагування на інциденти**

Розроблено послідовність дій при виявленні RAT: ідентифікація, локалізація, нейтралізація, аналіз та постінцидентні дії. Це забезпечує мінімізацію шкоди та швидке відновлення корпоративних систем.

6. **Практичні рекомендації щодо впровадження**

- організаційні заходи: політики безпеки, контроль доступу, аудит;
- технічні заходи: багаторівнева система захисту, контроль трафіку, захист кінцевих точок, резервування даних;
- навчання персоналу: тренінги, симуляції атак, формування культури безпеки.

Завдяки такому підходу розроблена модель забезпечує **синергійний ефект**, поєднуючи технічні та організаційні рішення, що дозволяє підвищити ефективність захисту від RAT, зменшити ризики поширення шкідливих програм та покращити готовність організації до кіберінцидентів.

ВИСНОВКИ

Проаналізувавши існуючі методи виявлення та протидії RAT, їх архітектурні особливості, поведінкові та мережеві ознаки, можна зробити висновок, що сучасні підходи мають значні обмеження при застосуванні у корпоративних мережах через високу адаптивність шкідливого ПЗ та використання складних механізмів приховування.

Було виявлено ряд недоліків існуючих рішень для виявлення RAT, а саме:

- обмежена ефективність сигнатурних методів проти модифікованих і поліморфних RAT;
- недостатнє використання поведінкового аналізу та кореляції подій у реальному часі;
- обмежені можливості мережевих систем моніторингу щодо виявлення C2-комунікацій через шифровані канали;
- відсутність єдиної моделі інтеграції різних методів виявлення у корпоративних системах.

Результатом виконаної роботи є розробка принципів побудови комплексної системи захисту від RAT та практичних рекомендацій щодо її впровадження. Під час виконання роботи було:

1. Проведено аналіз природи та класифікації RAT, що дозволило виділити основні механізми роботи, канали поширення та методи приховування, які використовуються сучасними шкідливими програмами.
2. Проаналізовано існуючі підходи до виявлення RAT, включаючи сигнатурні методи, поведінковий аналіз, аналіз мережевого трафіку та методи на основі машинного навчання, що дало змогу оцінити їх переваги та обмеження.
3. Визначено ключові поведінкові та мережеві ознаки RAT, які можуть слугувати індикаторами компрометації (IoC) для побудови системи виявлення.
4. Проведено огляд та порівняльний аналіз open-source RAT (AsyncRAT, QuasarRAT, Pupy, NJRat), що дозволило виділити експлуатаційні

особливості та способи приховування, які впливають на ефективність детектування.

5. Розроблено концептуальну модель комплексної системи захисту від RAT із модульною архітектурою для виявлення, моніторингу та реагування на інциденти, а також визначено організаційні, технічні та навчальні заходи для її впровадження.

6. За результатами моделювання та аналізу можна зробити висновок, що запропонована система забезпечує підвищення ефективності захисту корпоративних мереж та може слугувати основою для практичного впровадження та подальших досліджень у сфері кібербезпеки.

Оформлення результатів цього дослідження здійснювалося згідно з методичними рекомендаціями кафедри [21].

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. MITRE ATT&CK. Remote Access Tools (T1219). URL: https://attack.mitre.org/techniques/T1219/
2. MITRE ATT&CK. Command and Control Techniques. URL: https://attack.mitre.org/tactics/TA0011/
3. Соколов, В., & Складанний, П. (2023). Методика оцінки комплексних збитків від інциденту інформаційної безпеки. Кібербезпека: освіта, наука, техніка, 1(21), 99–120. https://doi.org/10.28925/2663-4023.2023.21.99120
4. Корнієць, В., & Складанний, П. (2024). Формування вимог до архітектури і функцій систем моніторингу кібербезпеки. Телекомунікаційні та інформаційні технології, 4(85), 90–96. https://doi.org/10.31673/2412-4338.2024.040224 5.
5. Cisco Talos. Remote Access Trojan Threat Analysis. URL: https://blog.talosintelligence.com
6. Symantec Enterprise. Analysis of Remote Access Trojans and Their Evolution. URL: https://symantec-enterprise-blogs.security.com
7. VirusTotal. Malware Families: RAT. URL: https://www.virustotal.com/gui/collections
8. Almukaynizi, M., Alasmary, W. "Network-Based Detection of Remote Access Trojans Using Behavioral Features." *IEEE Access*, 2022. DOI: https://doi.org/10.1109/ACCESS.2022.3151234
9. Sukaylo, I., & Korshun, N. (2022). The influence of NLU and generative AI on the development of cyber defense systems. Cybersecurity: Education, Science, Technique, 2(18), 187–196. https://doi.org/10.28925/2663-4023.2022.18.187196
10. Kumar, A., Singh, P. “Detection of RAT Malware Through Network Anomaly Analysis.” *Journal of Information Security and Applications*, 2020.
11. Bilge, L., Dumitras, T. “Before We Knew It: An Empirical Study of Zero-Day Attacks in the Real World.” *ACM CCS*.2012.
12. S. García, M. Grill. "An Empirical Comparison of Botnet Detection Methods." *Computers & Security*, 2014.

13. GitHub. AsyncRAT. URL: https://github.com/maaaaz/AsyncRAT
14. GitHub. QuasarRAT. URL: https://github.com/quasar/QuasarRAT
15. GitHub. Pupy. URL: https://github.com/n1nj4sec/pupy
16. GitHub. NJRat forks. URL: https://github.com/search?q=NJRat
17. Wireshark Documentation. Deep Packet Inspection Guide. URL: https://www.wireshark.org/docs
18. Suricata IDS Documentation. Network Threat Detection Engine. URL: https://docs.suricata.io
19. Zeek (Bro). Network Security Monitoring Platform Documentation. URL: https://docs.zeek.org
20. Romaniuk, O., Skladannyi, P., & Shevchenko, S. (2022). Comparative analysis of solutions to provide control and management of privileged access in the IT environment. Cybersecurity: Education, Science, Technique, 4(16), 98–112. https://doi.org/10.28925/2663-4023.2022.16.98112
21. Жданова, Ю. Д., Складанний, П. М., & Шевченко, С. М. (2023). Методичні рекомендації до виконання та захисту кваліфікаційної роботи магістра для студентів спеціальності 125 Кібербезпека та захист інформації. https://elibrary.kubg.edu.ua/id/eprint/46009/1/Y_Zhdanova_P_Skladannyi_S_Shevchenko_MR_Master_2023_FITM.pdf

Додаток А

Фрагмент коду для виявлення ознак активності RAT у Windows-системі

```
# Перевірка підозрілих мережевих підключень
Write-Host "=== Аналіз активних мережевих з'єднань ==="
$connections = Get-NetTCPConnection | Where-Object {
    $_.RemotePort -notin 80,443,53 -and
    $_.RemoteAddress -notlike "127.*" -and
    $_.RemoteAddress -notlike "::1"
}
$connections | Format-Table -AutoSize

# Перевірка автозапуску
Write-Host "`n=== Перевірка автозапуску ==="
$autoruns = Get-ItemProperty
HKLM:\Software\Microsoft\Windows\CurrentVersion\Run
$autoruns | Format-List

# Виявлення процесів без підпису (часта ознака RAT)
Write-Host "`n=== Аналіз підписів виконуваних файлів ==="
Get-Process | ForEach-Object {
    $path = $_.Path
    if ($path) {
        $sig = Get-AuthenticodeSignature $path
```

```
        if ($sig.Status -ne "Valid") {
            [PSCustomObject]@{
                Process = $_.Name
                Path = $path
                Signature = $sig.Status
            }
        }
    }
} | Format-Table -AutoSize

# Пошук змін у системних каталогах протягом останньої доби
Write-Host "`n=== Перевірка підозрілих змін у системних каталогах ==="
Get-ChildItem "C:\Users\*\AppData" -Recurse -ErrorAction SilentlyContinue |
Where-Object { $_.LastWriteTime -gt (Get-Date).AddDays(-1) } |
Select-Object FullName, LastWriteTime |
Format-Table -AutoSize
```

Додаток Б

Приклад структури клієнтської частини RAT

```
// *** ДЕМОНСТРАЦІЙНИЙ ФРАГМЕНТ ***
// Код не містить жодних шкідливих функцій і призначений лише для
// демонстрації типової архітектури RAT у навчальних цілях.

using System;
```

```
using System.Net.Sockets;
using System.Text;
using System.Threading.Tasks;

namespace DemoRAT
{
    // Базовий "клієнт", що демонструє логіку обміну повідомленнями
    class Client
    {
        private string serverAddress = "127.0.0.1";    // Локально, без зовнішніх підключень
        private int serverPort = 9000;

        public async Task StartAsync()
        {
            try
            {
                using (TcpClient client = new TcpClient())
                {
                    // У реальних RAT — тут відбувається C2-з’єднання
                    Console.WriteLine("Спроба підключення до навчального сервера...");
                    await client.ConnectAsync(serverAddress, serverPort);

                    NetworkStream stream = client.GetStream();
```

```
            byte[] buffer = Encoding.UTF8.GetBytes("Hello, server!");
            await stream.WriteAsync(buffer, 0, buffer.Length);

            byte[] response = new byte[1024];
            int bytes = await stream.ReadAsync(response, 0, response.Length);
            Console.WriteLine("Отримано відповідь: " +
Encoding.UTF8.GetString(response, 0, bytes));
        }
      }
      catch (Exception ex)
      {
        Console.WriteLine("Помилка: " + ex.Message);
      }
    }
  }

  // Точка входу
  class Program
  {
    static async Task Main(string[] args)
    {
      Console.WriteLine("Демонстраційний RAT-клієнт запущено.");
      Client client = new Client();
      await client.StartAsync();
```

```
        }
    }
}
```