Хмельницький національний університет
Факультет інформаційних технологій
Кафедра кібербезпеки

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему
Метод та система виявлення ботів в публічній мережі

Галузь знань 12 – Інформаційні технології

Спеціальність 125 – Кібербезпека

КРМКБ.220177.22.01.03 ПЗ

Виконав: студент 2 курсу, група КБм-22-1 _____ Білик Д.С.
Підпис

Керівник доц., к.т.н, доцент _____ Кльоц Ю.П.
Підпис

Нормоконтролер старший викладач _____ Мостовий С.В.
Підпис

До захисту допускаю:
Зав. кафедри кібербезпеки, к.т.н., доц _____ Кльоц Ю.П.
Підпис

8 12 2023 р.

Хмельницький, 2023

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій

Кафедра Кібербезпеки

Освітній рівень Магістр

Галузь знань 12 Інформаційних технологій

Спеціальність 125 Кібербезпека

Освітня програма Кібербезпека

ЗАТВЕРДЖУЮ

Зав. кафедри Ю.П. Кльоц

"30" 08 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Білику Денису Сергійовичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Метод та система виявлення ботів в публічній мережі

Керівник роботи Кльоц Юрій Павлович

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

кандидат технічних наук, доцент

Затверджена наказом № 30 ректора університету, додаток №25 від 15.08.2023

2. Строк подання студентом проєкту (роботи) на кафедру 01.12.2023

3. Вихідні дані до проекту (роботи) Розробити метод розподіленого виявлення керуючих компонентів ботнета, прототип мультиагентної системи виявлення ботнетів.

4 Зміст пояснювальної записки (перелік питань, які потрібно розробити) Вступ. Постановка задачі. Розробка архітектури мультиагентної системи, методу розподіленого виявлення керуючих компонентів ботнета. Проведення експерименту. Формування вимог до мультиагентної системи. Тестування системи. Висновки.

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали і посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Мостовий С.В. Старший викладач кафедри кібербезпеки		

7. Дата видачі завдання «01» вересня 2023р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Вибір напряму дослідження і узгодження тематики КРМ з керівником	01.06.2023	
2	Ознайомлення з предметною областю; формулювання мети і задач дослідження; визначення об'єкта і предмета дослідження	04.09.2023	
3	Робота над розділом 1 – огляд життєвого циклу ботнетів; аналіз методів їх виявлення	18.09.2023	
4	Робота над розділом 2 – розробка архітектури системи та алгоритмів для виконання поставленого завдання	02.10.2023	
5	Робота над розділом 3 – реалізація методів та схем для проведення експерименту	16.10.2023	
6	Робота над розділом 4 – формування вимог до системи, проектування та тестування системи	06.11.2023	
7	Робота над науковою публікацією	10.11.2023	
8	Узгодження отриманих результатів, оформлення пояснювальної записки згідно вимог	15.11.2023	
9	Попередній захист роботи	17.11.2023	
10	Захист роботи на засіданні ЕК	06.12.2023	

Студент _____ Д.С. Білик
Підпис Ініціали, прізвище

Керівник проекту (роботи) _____ Ю.П. Кльоц
Підпис Ініціали, прізвище

АНОТАЦІЯ

Тема кваліфікаційної роботи: Метод та система виявлення ботів в публічній мережі.

Автор роботи: Білик Денис Сергійович

Керівник роботи: к.т.н., доц. Кльоц Юрій Палович

Загальний обсяг роботи: 84 сторінок, 25 рисунків, 13 таблиць, 1 додаток, 60 посилань.

Ключові слова: публічна мережа, ботнети, система виявлення.

Для досягнення мети в роботі були сформульовані та вирішені наступні завдання: проведено аналіз існуючих підходів до виявлення ботнетів; розроблено алгоритм для визначення керуючого трафіку ботнетів у глобальних мережах; представлено структуру мультиагентної системи для виявлення та блокування ботнетів, проведено аналіз ефективності запропонованих у дослідженні алгоритмів; розроблено метод для розподіленого виявлення керуючих компонентів ботнету, який дозволяє виявляти керуючі сервери та вузли мережі, використовуючи сигнатури керуючого трафіку.

При вирішенні поставлених у роботі завдань були використані наступні методи дослідження: теоретико-множинні методи для представлення моделей, агентно-орієнтовані методи для розробки програмних систем, об'єктно-орієнтовані методології для проектування, і методи інтелектуального аналізу даних. Для оцінки результатів запропонованих рішень використовувалися методи функціонального та інформаційного моделювання.

Здобуті результати мають практичне значення в контексті використання мультиагентної системи для виявлення та блокування ботнетів. Ця система базується на виявленні керуючого трафіку ботнету за допомогою інтелектуального аналізу даних і може бути успішно впроваджена для забезпечення безпеки інформаційних систем.

08.12.2023р.

ANNOTATION

Theme of qualification work: Method and system for detecting bots in a public network.

Author of the work: Bilyk Denys Serhiyovych

Mentor: Ph.D., Assoc. Yurii Palovich Klots

Total volume of work: 84 pages, 25 figures, 13 tables, 1 appendix, 60 references.

Keywords: public network, botnets, detection system.

To achieve the goal, the following tasks were formulated and solved in the work: an analysis of existing approaches to detecting botnets was carried out; an algorithm was developed to determine the control traffic of botnets in global networks; the structure of the multi-agent system for detecting and blocking botnets is presented, the effectiveness of the algorithms proposed in the study is analyzed; developed a method for distributed detection of control components of a botnet, which allows detection of control servers and network nodes using signatures of control traffic.

The following research methods were used to solve the tasks set in the work: multiple-theoretic methods for presenting models, agent-oriented methods for developing software systems, object-oriented methodologies for design, and methods of intellectual data analysis. Functional and informational modeling methods were used to evaluate the results of the proposed solutions.

The obtained results are of practical importance in the context of using a multi-agent system to detect and block botnets. This system is based on the detection of botnet control traffic using intelligent data analysis and can be successfully implemented to ensure the security of information systems.

08.12.2023р.

ЗМІСТ

ВСТУП

Протягом останніх десятиліть у всьому світі відбувся значний розквіт Інтернету та програм, що базуються на ньому. Використання Інтернету стало необхідною складовою нашого життя. Він, безсумнівно, приніс великий комфорт, але з ростом нашої залежності з’явилося чимало важливих проблем у сфері інформаційної безпеки. Слід наголосити, що питання безпеки в Інтернеті набуває все більшого значення для тих, хто використовує Інтернет у роботі, бізнесі або освіті.

Більшість атак та шахрайських дій в Інтернеті виконуються за допомогою шкідливого програмного забезпечення, такого як віруси, трояни, черв'яки, шпигунські програми та ботнети. Шкідливе програмне забезпечення є основним джерелом більшості злочинних дій в Інтернеті, таких як цільові атаки, розподілені атаки типу "відмова в обслуговуванні", шахрайські схеми та інші. Серед різних видів шкідливого програмного забезпечення ботнети є основною платформою, яку зловмисники використовують як масштабний інструмент для підтримки своєї злочинної діяльності, такої як атаки типу DDoS, відсилання спаму, фішинг та крадіжка інформації.

Дані свідчать, що ботнети стали основною загрозою для безпеки в Інтернеті. Зазвичай ботнети виявляються за допомогою пасивного спостереження та аналізу мережевого трафіку. Для їх виявлення використовуються підходи, які базуються на пошуку сигнатур або виявленні аномалій у трафіку. Менш популярні методи включають аналіз DNS-трафіку та використання пасток (honeypots). Головним недоліком існуючих систем виявлення ботнетів є те, що вони не враховують взаємозв'язку між різними агентами ботнету та різними етапами їх життєвого циклу. Це призводить до часткового виявлення і неможливості повністю блокувати діяльність ботнету. Отже, актуальним завданням є розробка мультиагентної системи для виявлення та блокування ботнетів, в якій основний аналіз базується на визначенні керуючого трафіку в мережі з використанням методів інтелектуального аналізу мережевого трафіку.

Об'єктом дослідження у цьому проекті є системи захисту від ботнетів у різних відкритих комп'ютерних мережах, включаючи Інтернет.

Предметом дослідження є розгляд методів і алгоритмів для виявлення ботнетів шляхом використання інтелектуального аналізу даних у межах мультиагентного підходу.

Основною метою цієї роботи є підвищення рівня захищеності інформаційних систем від атак, які використовують ботнети, завдяки розробці та впровадженню мультиагентної системи для виявлення та блокування ботнетів з використанням алгоритмів інтелектуального аналізу даних.

Для досягнення мети в роботі були сформульовані та вирішені наступні завдання: провести аналіз існуючих підходів до виявлення ботнетів; розробити алгоритм для визначення керуючого трафіку ботнетів у глобальних мережах, використовуючи технології інтелектуального аналізу даних; представити структуру мультиагентної системи для виявлення та блокування ботнетів, провести аналіз ефективності запропонованих у дослідженні алгоритмів; розробити метод для розподіленого виявлення керуючих компонентів ботнету, який дозволяє виявляти керуючі сервери та вузли мережі, використовуючи сигнатури керуючого трафіку; створити експериментальний прототип мультиагентної системи для виявлення та блокування ботнетів у рамках наукових досліджень.

Методи дослідження. При вирішенні поставлених у роботі завдань були використані наступні методи дослідження: теоретико-множинні методи для представлення моделей, агентно-орієнтовані методи для розробки програмних систем, об'єктно-орієнтовані методології для проектування, і методи інтелектуального аналізу даних. Для оцінки результатів запропонованих рішень використовувалися методи функціонального та інформаційного моделювання.

Здобуті результати мають практичне значення в контексті використання мультиагентної системи для виявлення та блокування ботнетів. Ця система базується на виявленні керуючого трафіку ботнету за допомогою інтелектуального аналізу даних і може бути успішно впроваджена для забезпечення безпеки інформаційних систем.

1 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ВИЯВЛЕННЯ БОТНЕТІВ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Огляд ботнетів та їх життєвого циклу

Однією з найбільш серйозних загроз у сфері Інтернет-безпеки є ботнети. Бот - це програма-робот, яка функціонує на зараженому комп'ютері автономно і автоматично, без належного повідомлення користувача. Зазвичай, код бота розробляється зловмисними групами і включає в себе різноманітні функціональні можливості, призначені для виконання різних шкідливих операцій [1]. У окремих випадках термін "бот" може вказувати на комп'ютер, який був інфікований ботом. Ботнет - це мережа з таких інфікованих ботів, які перебувають під віддаленим контролем зловмисника. Зловмисника, який керує ботнетом, зазвичай називають "бот-майстром". Бот-майстер здійснює керування ботнетом за допомогою спеціальних каналів комунікації та управління. [2].

В даний час ботнети є однією з основних причин зловмисної діяльності в Інтернеті, що включає:

– Децентралізовані атаки, такі як "атаки типу відмова в обслуговуванні" (DDoS), можуть бути ініційовані ботнетом з метою спрямованого перерозподілу ресурсів, таких як пропускна здатність системи в Інтернеті. Ця атака призводить до того, що система не може надавати належне обслуговування своїм легітимним користувачам через використання ресурсів ботнету. В сучасний час майже всі DDoS-атаки виконуються за допомогою ботнет-платформ. Незважаючи на відносну простоту техніки DDoS-атак, вона виявляється надзвичайно ефективною для вимірювання масштабів ботнету та загальної пропускної здатності ботів [3].

– Розсилання спаму. Приблизно половина всіх електронних листів представляє собою надмірну пошту (спам), що складається з кількох мільярдів спам-повідомлень, які щодня з'являються в інтернет-трафіку. Більшість цього спаму насправді надсилається ботнетами. Точний відсоток спаму, який генерують ботнети, може коливатися в залежності від різних статистичних даних. Ряд відомих

ботнетів був використаний для розсилки спаму, у тому числі Bobax, ранній спам-бот, що використовує HTTP як C&C, і Storm Worm, P2P ботнет який агресивно проводить розсилку спаму [4,5].

– Фішинг. Ботнети активно використовуються для створення шкідливих підроблених веб-сайтів. Зазвичай, зловмисники розсилають спам-повідомлення (часто використовуючи ботнети) з метою обману користувачів та змусити їх відвідати ці підроблені сайти.

– Клікфрод. Бот-майстер може одержувати прибуток від керування ботами та отримання їхніх кліків на онлайн-рекламні оголошення, що дозволяє надсилати HTTP-запити на веб-сторінки рекламодавців, з метою особистого чи комерційного зиску. Це може бути частиною схеми підвищення рейтингу веб-сайтів у пошукових системах. Наприклад, ботнет Clickbot.A використовується для виконання малозамітних атак шахрайського кліку імітуючи активність великої кількості звичайних користувачів.

– Крадіжка інформації. Боти активно використовуються для викрадення конфіденційної інформації, такої як номери кредитних карток, паролі або авторизаційні ключі, зберігаються на локальному комп'ютері користувача. За допомогою кейлогерів та функцій захоплення екрану, бот легко може вкрасти паролі до облікового запису системи дистанційного банкінгу.

– Поширення інших небажаних програм, наприклад, рекламного чи шкідливого програмного забезпечення (ШПЗ).

– Хостинг "Bulletproof". На цьому сервері інфіковані комп'ютери можуть розміщувати незаконний вміст.

Для отримання глибшого розуміння природи ботнетів, важливо розглянути їх життєвий цикл, який, як правило, складається з кількох етапів: концепція (початок), поширення, взаємодія, виконання атак, оцінка результатів атаки (завершення) [6].

На першому етапі життєвого циклу будь-якого ботнету визначається його концепція. Для створення ботнету важливо мати мотивацію, оскільки вона визначатиме архітектуру і розробку ботнету. На цій стадії визначаються

характеристики, які обумовлені конкретною метою створення ботнету. Етап складається з двох основних процесів: проектування і розробки. На етапі проектування конкретизується організаційна структура ботнету, яка може бути трьох типів:

1. Централізована. Всі заражені комп’ютери знаходяться під централізованим управлінням, тобто координуються за допомогою виділеного керуючого сервера.

2. Децентралізована. В цьому випадку всі комп'ютери передають команди управління між собою. Структура побудована за технологією P2P.

3. Гібридна включає в себе декілька рівноправних підмереж ботнету, кожна із яких керується виділеним вузлом.

Етап поширення. Після створення шкідливого коду бота виникає необхідність в його поширенні. Зазвичай це досягається шляхом використання вразливостей в системах та введення ботів на вразливі вузли. Існує багато шляхів для поширення шкідливого програмного забезпечення. Наприклад, можна заразити віддалені вразливі вузли, використовуючи безпосередню експлуатацію вразливостей, або розповсюджувати його через засоби соціальної інженерії, такі як електронна пошта та миттєві повідомлення. Останнім часом бот-майстри використовують компрометовані веб-сервіси для зараження тих, хто відвідує веб-сайти. Шляхом впровадження кількох методів розповсюдження шкідливого коду бот-майстер може заражати багато жертв [7].

Етап взаємодії описує взаємодію між бот-майстром і ботами, включаючи два різних процеси. Перший процес включає в себе реєстрацію скомпрометованого вузла як функціонуючої частини ботнету. Другий процес полягає в забезпеченні керуючого зв'язку для ботнету. Бот-майстер повинен підтримувати зв'язок з ботами через керуючий канал. Цей обмін інформацією включає передачу команд від бот-майстра до ботів і виконання операцій з технічного обслуговування [8].

На етапі виконання атаки бот-майстер видає команди ботам щодо проведення атак, і ботнет займається зловмисною діяльністю.

Кінцевою метою будь-якого ботнету є успішне виконання атаки. Для визначення, чи досягнута ця мета, важливо провести оцінку результатів атаки. Якщо результати оцінки свідчать про неуспішність функціонування, то ботнет стає неефективним, і всі ресурси, витрачені на попередні етапи, залишаються марними.

1.2 Аналіз методів виявлення ботнетів

1.2.1 Виявлення вторгнень та шкідливого програмного забезпечення

Існуючі методи виявлення вторгнень та шкідливого програмного забезпечення можна розподілити на дві категорії: мережеві рішення та рішення, що працюють на самостійних мережевих вузлах. Техніки, які використовуються на вузлах, мають важливе значення для виявлення виконуваних файлів шкідливого програмного забезпечення і аномалій у поведінці на рівні окремих вузлів. [9-14]. Антивірусні засоби є ефективними для довготривалого виявлення традиційних вірусів. Ще одним стандартним методом виявлення вторгнень на рівні окремого вузла є моніторинг системних викликів [15-19].

Але при пошуку ботнетів ці методи мають деякі недоліки. По-перше, традиційні антивірусні інструменти, які базуються на сигнатурному аналізі, вимагають об'ємної, точної і регулярно оновлюваної бази сигнатур. Ботнети можуть легко ухилятися від виявлення за сигнатурою, оновлюючи свій код частіше, ніж користувачі оновлюють свої антивірусні бази. По-друге, системи виявлення на рівні вузла мають ті ж привілеї, що й боти на конкретному вузлі. Таким чином, боти можуть вимикати антивірусні заходи системи або використовувати технології руткіта для захисту себе від виявлення на локальному вузлі [20-22]. Частота виявлення ботнетів в порівнянні з традиційним шкідливим програмним забезпеченням є відносно низькою. Наприклад, шкідливе програмне забезпечення, як Кракен, не було виявлено в 80% комерційних антивірусних продуктах. Таким чином, можна стверджувати, що мільйони вузлів у мережі Інтернет пов'язані з діяльністю ботнетів, і фактичний відсоток може бути ще вищим. Крім того, моніторинг вузла в режимі реального часу на основі поведінки,

зазвичай, супроводжується значними системними витратами, і тому такі рішення можуть бути менш привабливими для кінцевих користувачів.

Досягнені дослідження в області виявлення вторгнень в мережах вже відображають багато методів та систем виявлення вторгнень. До таких систем належать Snort [23] і Zeek [24], які базуються на сигнатурах для визначення спроб вторгнень в мережевому трафіку. Головним обмеженням сигнатурних систем виявлення вторгнень є їх неспроможність впізнати нові атаки, які раніше не були відомі і, отже, не мають сигнатур. Системи виявлення вторгнень, що базуються на аналізі аномалій, можуть подолати це обмеження, описуючи нормальний трафік та визначаючи будь-яке відхилення від цього опису як аномалію. Проте основним недоліком таких рішень є велика кількість хибних спрацювань [25-29].

Перед появою ботнетів характерними були хробаки. Хробаки - це зловмисна програма, яка поширюється шляхом саморозмноження через мережеві технології. Основною відмінністю ботнетів від хробаків є наявність керуючого каналу [30-32]. Тому ботнети є більш гнучкими, ніж хробаки. Гнучкість полягає в можливості керувати діяльністю ботів, в той час як діяльність хробаків програмується розробниками напередодні і відсутній вибір функціональності хробака після зараження.

Певні методи виявлення вторгнень і системи виявлення вторгнень можуть бути корисні для виявлення певних аномалій у ботнетах, але самі по собі не є відповідними для виявлення всіх видів ботнетів з наступних причин:

– Більшість систем виявлення вторгнень спеціалізуються на аналізі вхідного трафіку для виявлення ознак можливих точкових атак. Зазвичай вони успішно виявляють початкові спроби вторгнення, але це може супроводжуватися великою кількістю сповіщень про підозрілий мережевий трафік. Однак визначення, чи є успішним зараженням локального хоста від звичайного сканування і спроб вторгнення, завдається надзвичайно важко. Також важко визначити, коли само шкідливе програмне забезпечення починає активну діяльність в контрольованій мережі. Бо комп'ютери можуть бути заражені ботами не лише за допомогою віддалених атак, але і через інші механізми. Наприклад, користувач може

випадково завантажити шкідливий вкладення в електронній пошті, або ж може стати жертвою атаки типу "Drive-by download", відвідавши певний веб-сайт. Також вже заражений ноутбук може підключатися до контрольованої мережі.

– Ботнети демонструють високий рівень гнучкості в своєму життєвому циклі інфікування, який може включати різні етапи та дії. Проте існуючі підходи до виявлення зазвичай обмежені в аналізі лише конкретного етапу, наприклад, сканування, і, відповідно, можуть бути менш ефективними у виявленні ботнетів. Ці підходи можуть призводити до невірних результатів, якщо звичайний хост або хост, заражений іншим видом шкідливого програмного забезпечення, проявляє активність, що схожа на сканування. Або ж вони можуть не виявляти ботнети, якщо боти використовують нестандартні методи сканування, відмінні від тих, що визначені системою виявлення.

Отже, є потреба у нових методах, які були б більш придатними для ефективного виявлення ботнетів.

1.2.2 Взаємозв'язок сповіщень та взаємодія систем виявлення вторгнень

Основне призначення взаємозв'язку сповіщень – зменшення обсягу журналу подій, виявлення складних атак, які складаються з кількох етапів, і визначення спроб атаки. Зокрема, методи, які використовуються для виявлення складних атак, можуть мати схожість із технікою вертикальної кореляції.

Один з підходів до виявлення складних і багатоетапних атак включає в себе явне визначення етапів і послідовності різних компонентів атаки. Наприклад, системи USTAT і NetSTAT, які базуються на методах аналізу змін стану, розглядають комп'ютерну атаку як послідовність дій, що призводять до переходу системи безпеки в інший стан. Крім того, вони встановлюють зв'язки між різними етапами складних атак, використовуючи STATL - мову для представлення атак у вигляді станів та переходів [34-35]. Ще одна подібна система - JIGSAW, використовує поняття концепції і можливості для моделювання складних атак [36-38]. CAML - це мова, призначена для визначення та виявлення сценаріїв багатоетапних атак [39-40]. Усі ці методи ґрунтуються на чітких причинно-наслідкових зв'язках, таких як передумови та наслідки або точна часова

послідовність атак. Однак очевидним обмеженням цих методів є необхідність вручну вказувати всі залежності та послідовності для всіх атак, а такі залежності та послідовності часто бувають невідомими або дуже неточними. Навіть відсутність певних подій у залежностях або послідовностях може призвести до невдачі всієї кореляції. Якщо йдеться про виявлення ботнетів, то навіть якщо вузли, що інфіковані ботами, регулярно виконують послідовність загальних дій (етапів), дуже важко точно визначити всі ці етапи та виявити їх, так само важко, як передбачити порядок та часовий інтервал, протягом якого вони відбудуться. Таким чином, зазначені вище методи кореляції сповіщень не є ефективними для виявлення ботнетів.

Головною метою взаємодії між системами виявлення вторгнень (IDS) є збір інформації з різних джерел для виявлення розподільних і координованих атак. Ці методи виявлення мають певні спільні риси з методами виявлення керуючого трафіку ботнету.

Деякі автори пропонують застосовувати розподілену архітектуру, яка об'єднує кілька агентів для виявлення вторгнень та можливостей реагування. Ці методи надають багаторівневе розподілене рішення для виявлення розподільних і скоординованих атак на різних хостах. У інших дослідженнях розглядають співвідношення даних з різних джерел (наприклад, журналів реєстрації Syslog, Firewall, Netflow) для підвищення точності виявлення вторгнень. Всі ці методи пропонують абстрактну високорівневу архітектуру для гібридних систем виявлення вторгнень, замість надання конкретних алгоритмів і методів виявлення для певних атак, таких як ботнети [41].

Система Seurat розроблена для виявлення агрегованих аномальних подій, таких як поширення шкідливого програмного забезпечення, шляхом аналізу змін у файловій системі вузла відносно часу. Для виявлення поширення шкідливого програмного забезпечення в мережі, запропоновано використовувати спільні групи машин для обміну інформацією про останні системні виклики, які були виконані.

1.2.3 Відстеження на основі Honeypot

Для ефективного збору інформації про ботнети та їх відстеження, дослідники

часто використовують методи приманки, відомі як honeypot. Зокрема, honeypot використовувалися для моніторингу та відстеження ботнетів, і зібрані дані та звіти сприяли кращому розумінню діяльності цих ботнетів. [42-43]. Nepenthes - це система низькорівневої інтеракції, яка імітує декілька потенційних вразливостей та автоматизує збір бінарних файлів шкідливого програмного забезпечення. Використовуючи цей комплексний підхід для збору екземплярів ботів і відстеження ботнетів, дослідники можуть глибше вивчати поточну активність ботнетів. Після збору примірників ботів, аналіз бінарних файлів дозволяє створювати сигнатури для конкретних ботнетів або отримувати інформацію про сервери управління та контролю (C&C).

Хоча використання honeypot є ефективним інструментом для збору інформації про ботнети, вони мають ряд обмежень. По-перше, низькорівневі honeypot, такі як Nepenthes, здатні відловлювати атаки лише за умови, що вони спеціально імітують обмежене число відомих експлойтів, тобто вони не можуть виявити атаки, для яких не існують чітко визначених експлойтів. З іншого боку, високорівневі honeypot можуть не вміти точно імітувати всі сервіси, і вони також можуть виявити проблеми зі шкалою. По-друге, honeypot зазвичай призначені для захоплення шкідливих програм, які поширюються через сканування віддалених вразливостей, і, отже, вони можуть не бути ефективними для захоплення шкідливих програм, які використовують інші методи поширення, такі як атаки через електронну пошту або методи типу «Web drive-by download», які є двома найбільш поширеними способами розповсюдження. По-третє, немає гарантії, щодо частоти або обсягу захоплення шкідливих програм за допомогою honeypot, оскільки ці системи можуть лише очікувати і сподіватися, що шкідливі програми самі звернуться до них. По-четверте, шкідливі програми можуть уникнути виявлення через сканування мережі з використанням "відомих" honeypot, визначити віртуальне оточення, яке часто використовується для розгортання honeypot, і змінити свою поведінку, щоб уникнути виявлення та аналізу. Нарешті, honeypot повідомляють про зараження лише на системах-пастках і не можуть виявити зараження на реальних машинах, які не є частиною honeypot і працюють в

корпоративній мережі. Всі ці недоліки обмежують ефективність використання honeypot як засобу виявлення ботнетів [44-47].

1.2.4 Існуючі підходи виявлення ботнетів

Питання виявлення ботнетів вже давно цікавить науковців, і було запропоновано багато різних методів для їх виявлення. Один із запропонованих підходів - це логіко-імовірнісний метод, який дозволяє розробляти моделі бот-атак, що враховують найкращі практики в боротьбі з атаками ботнетів. У цьому методі використовується механізм прийняття рішень, який базується на формалізації експертного досвіду з бот-атак в нечіткій базі нечітких продукційних правил.

Автори запропонували структуру інтелектуальної системи, яка включає наступні рівні:

– система нечітких продукційних правил, що описує роботу ідентифікатора характеристик бот-атак з урахуванням експертних оцінок;

– нейро-нечітка мережа, у структурі якої відображена система нечітких продукційних правил;

– чітка самонавчальна нейронна мережа для вирішення завдання кластеризації (класифікації) вхідних даних бот-активності з веб-простору.

Для виявлення IRC-ботнетів запропоновано об'єднати IRC-статистику з даними про робоче навантаження TCP, зокрема аномальною активністю сканування. Цей підхід дозволяє визначати певні типи ботнетів, зокрема IRC-ботнети, які виконують атаки сканування [48-50].

Також використовується метод, який включає аналіз DNSBL (чорних списків DNS) для виявлення членів ботнетів, які поширюють спам. Основним припущенням цього методу є те, що бот-керівник може використовувати DNSBL для визначення статусів своїх ботів, іншими словами, це відзначається як підозріле, коли машина генерує багато запитів щодо інших, але сама рідко отримує запити від інших машин. Ця евристика може бути корисною у деяких випадках, але загалом не завжди ефективна і може призводити до багатьох помилкових спрацьовувань. Отже, цей підхід працює лише в обмежених ситуаціях для виявлення спам-ботнетів. [51-52].

Система Rishi - це система виявлення IRC-ботів, яка ґрунтується на методі сигнатур і виявляє їх за співпадінням з іменами відомих IRC-ботів. Подібно до всіх інших сигнатурних засобів, таких як антивіруси або системи виявлення вторгнень, цей підхід є точним лише в разі наявності великої та актуальної бази даних сигнатур. Однак ця система також має свої недоліки, які є характерними для сигнатурних рішень: вона не може виявити ботів, для яких відсутні шаблони з відомими іменами. [53-54].

Традиційні методи виявлення шкідливого програмного забезпечення, які використовують сигнатури та евристичні підходи, зазвичай не можуть надати ефективний рівень виявлення нових і невідомих варіантів шкідливого ПЗ. Тому для вирішення цієї задачі широко використовуються методи інтелектуального аналізу даних, зокрема, методи машинного навчання. Для виявлення ботнетів використовуються різні характеристики мережевого трафіку, зокрема трафік протоколів обміну повідомленнями, таких як IRC. Ці характеристики можуть бути аналізовані на мережевому рівні, наприклад, шляхом вимірювання обсягу передачі даних у байтах на секунду. Також проводиться аналіз потоків мережевого рівня для виявлення та контролю IRC-ботнетів у великих мережах.

Система BotSwat є вузловою системою виявлення шкідливого програмного забезпечення, яка ідентифікує програми, що використовують мережеві дані як аргументи для системних викликів, і ці дані не вводяться користувачем і не мають явного дозволу на використання у якості параметрів системних викликів. Система намагається визначити поведінку потенційно віддаленого керування ботом за допомогою цього підходу. Важливо враховувати, що цей метод може призводити до помилкових виявлень (оскільки багато легітимних програм також використовують мережеві дані в аргументах своїх системних викликів), а також до зниження продуктивності (оскільки аналіз поширення шкідливих програм через цей метод є складним, він, як правило, використовується для аналізу, а не для активного виявлення) [55].

TAMD - це система, яка служить для виявлення потенційно шкідливого програмного забезпечення, включаючи ботнети. Вона обробляє трафік, який має

схожий зовнішній адреси призначення, а також подібні корисні дані у пакетах і включає в себе внутрішні вузли, які працюють на однакових операційних системах. Метод агрегування, що застосовується в TADM, базується на спостереженні за обсягами трафіку в мережі призначення, зокрема, на виявленні збільшення трафіку порівняно з попередніми значеннями. Важливо враховувати, що цей метод обмежений виявленням ботів, які використовують централізовану структуру керування C&C, і може не ефективно працювати з іншими структурами (наприклад, P2P). TADM намагається виявити широкий спектр потенційно підозрілих хостів, які мають спільні мережі призначення, однакові корисні дані та подібні операційні системи. Тому TADM може викликати більший відсоток помилкових спрацьовувань [56-57].

1.3 Суть мультиагентної системи виявлення та блокування ботнетів

Існує потреба в розробці нових методів, які краще підходять для виявлення ботнетів. Зазвичай традиційні техніки виявлення вторгнень та шкідливого програмного забезпечення корисні для розпізнавання конкретних характеристик ботнетів. Деякі з наявних методів можуть стати частиною нової системи, яка об'єднує їх зі свіжими підходами до виявлення. Варто зазначити, що ботнети є складними мультиагентними системами з елементами інтелектуальної діяльності. Під час розповсюдження ботнетів автоматично відбувається аналіз програмного забезпечення користувачів на предмет вразливостей, які можуть бути використані для інфікування. Після інфікування боти повинні вирішувати складні завдання, такі як обхід захисних систем користувачів, маскування своєї діяльності та спілкування з керуючими серверами. Для надійного захисту від ботнетів необхідно використовувати систему захисту, що має такий же рівень складності, як і самі ботнети. Це означає, що система має здатність аналізувати мережеві дані в різних мережах, виявляти мережеві атаки, впливати на фільтрацію трафіку, розпізнавати сигнатури шкідливої поведінки та співпрацювати з іншими компонентами для ефективного виконання цих завдань. [58].

Мультиагентну систему виявлення та блокування ботнетів NET.BOTNET можна розглядати з семи різних кутів, що допоможуть зрозуміти її структуру та функціональність.

Перший аспект стосується типу рішення: вузлового або мережевого. Серед систем виявлення ботнетів було представлено BotSwat, який є прикладом вузлового рішення, так само як і NET.BOTNET, який є прикладом мережевого рішення.

Другим аспектом є метод роботи, який може бути заснований на сигнатурах, поведінці або виявленні аномалій. В даному контексті, Rishi є прикладом рішення, що використовує сигнатури. Інші рішення оперують методами аналізу поведінки або виявлення аномалій. Важливо відзначити, що представлене рішення включає в себе різні методи роботи, які використовуються для різних завдань. Наприклад, для виявлення атаки можуть застосовуватися два методи, для виявлення керуючого трафіку ботнету - метод аналізу поведінки, а для виявлення ботів - сигнатурний метод.

Третім аспектом є режим роботи системи, який може бути пасивним або активним. Всі згадані вище методи та системи є пасивними. Це означає, що вони спостерігають за мережевим трафіком та поведінкою системи, не втручаючись у її активність. Перевагою пасивної стратегії є безпека, оскільки вона не втручається в роботу ботнету. Важливо відзначити, що багатьом пасивним системам потрібно більше часу для налагодження моніторингу.

Четвертим аспектом є етап виявлення. Життєвий цикл ботнету, описаний раніше, може бути спрощено до двох етапів - підготовка та експлуатація, як показано на рисунку 1.1. В етапі підготовки нешкідливий вузол заражається ботом шляхом віддаленого зараження або виконання шкідливого файлу, і стає готовим до керування через C&C. Вузол починає фазу підготовки після атаки і завершує її, коли бот набуває повну функціональність. Фаза експлуатації починається, коли бот намагається підключитися до керуючого каналу і готовий виконувати шкідливі дії за директивами бот-майстра. Таким чином, методи виявлення ботнетів можна класифікувати в залежності від моменту сповіщення про виявлення - це може бути

на етапі підготовки або на етапі експлуатації. Усі розглянуті системи виявлення, за винятком BotHunter, спеціалізуються на етапі експлуатації, тобто вони виявляють наявність ботів, незалежно від того, яким способом вони проникли в мережу.

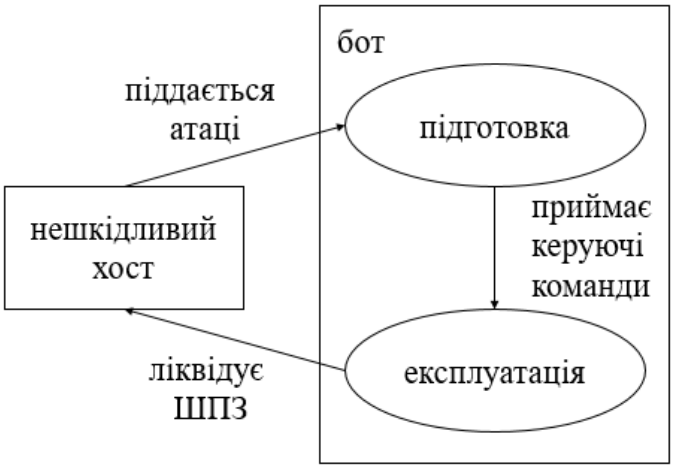

Рисунок 1.1 - Спрощений життєвий цикл зараження ботом

П'ятим аспектом можна виокремити ціль виявлення. TAMD спрямовані на виявлення групи ботів, у той час як інші системи фокусуються на виявленні окремих ботів. Фактично, це два різних підходи, які доповнюють один одного важливим чином. Метод, спрямований на виявлення груп, потребує спостереження за багатьма ботами (принаймні двома) для успішного виявлення. Він дозволяє виявити аномалії, які не можуть бути помічені на рівні окремих вузлів.

Шостим аспектом є передумова виявлення. Ці рішення потребують додаткової зовнішньої інформації (наприклад, DNSBL). З серед розглянутих рішень деякі вимагають інформації з інших джерел, таких як DNSBL, та деякі потребують попереднього аналізу кластерів даних (наприклад, результатів сканування), отриманих від інших систем.

Останнім аспектом є залежність системи від конкретної техніки управління ботнетом, такої як протокол чи структура. Багато існуючих рішень працюють лише з певним протоколом або структурною C&C, і призначені тільки для ботнетів, які базуються на IRC. Проте ці методи можуть бути використані для виявлення ботнетів, що використовують інші типи C&C, такі як ті, які ґрунтуються на засобах

миттєвого обміну повідомленнями. Незважаючи на це, вони, як і раніше, обмежені централізованою структурою і вимагають знання профілю трафіку. TAMD, з іншого боку, не обмежена конкретним протоколом взаємодії з C&C, але використовує агрегацію за адресою призначення, що обмежує її застосування тільки до централізованої структури ботнету.

1.4 Постановка задачі

Мета кваліфікаційної роботи магістра – розробка методу та системи виявлення ботів в публічній мережі. Для досягнення поставленої мети слід виконати наступне:

1. спроектувати архітектуру системи, що включатиме в себе агентів для виявлення ботів, та описати взаємодію цих агентів між собою для прийняття рішень, щодо виявлення та реагування на ботнети;
2. розробити метод виявлення керуючих компонентів ботнета, за допомогою яких здійснюється управління, базуючись на алгоритмі виявлення керуючого трафіку, який допоможе ідентифікувати активність ботнета;
3. представити метод розподіленого виявлення керуючих компонентів ботнета, з яких здійснюється контроль атаки ботнету;
4. провести експериментальне дослідження задля доведення ефективності розробленого методу, описавши середовище проведення експерименту;
5. розробити прототип мультиагентної системи виявлення ботнетів шляхом формування вимог до системи, проектуванням та безпосередньо проведенням тестування роботи системи.

2 РОЗРОБКА АРХІТЕКТУРИ ТА АЛГОРИТМІВ СИСТЕМИ ВИЯВЛЕННЯ БОТНЕТІВ

2.1 Архітектура мультиагентної системи

Використання мультиагентного підходу надає основну перевагу - можливість динамічного розв'язання задачі виявлення ботнетів. Цей підхід дозволяє створити гнучкий і масштабований метод виявлення.

Перш за все, для виявлення ботнета необхідно виявити розподілену атаку типу "відмова в обслуговуванні", яку найчастіше використовують ботнети. Після виявлення такої атаки, важливо заблокувати її на стороні джерела атаки і взяти атакуючий інструмент під спостереження для виявлення характерних ознак роботи бота. Далі потрібно спробувати ідентифікувати інших учасників ботнета, шукаючи у різних мережах раніше виявлені ознаки роботи бота.

При розробці методу виявлення ботнетів була використана типова структура мережі Інтернет, яка базується на взаємодії автономних систем. Пропонований метод виявлення ботнетів ґрунтується на засобі захисту від розподілених атак типу "відмова в обслуговуванні". Цей метод дозволяє виявити атаку в мережі, що є ціллю атаки, і запобігти генерації атаки в мережі, яка є джерелом атаки. Для відображення структури та функцій системи виявлення ботнетів була створена функціональна модель, яка була побудована за допомогою методології функціонального моделювання IDEF0. Цей метод дозволяє створювати описові графічні моделі, що показують функції системи та інформацію, яка пов'язана з цими функціями. Модель організована за принципом декомпозиції, де елементи на кожному рівні деталізації виконують дії щодо обробки інформації за визначеними умовами і використовують задані механізми. SADT-модель, яка використовується, поєднує діаграми в ієрархічні структури, де діаграми верхніх рівнів менш деталізовані, ніж діаграми нижніх рівнів. Ця методологія створена спеціально для побудови моделей складних систем та визначення системних шляхів їх побудови.

Контекстний блок функціональної моделі, згідно з рисунком 2.1, ілюструє,

що на вхід мультиагентної системи виявлення та блокування ботнетів надходить мережевий трафік. За допомогою різних методів та алгоритмів, спрямованих на забезпечення безпеки інформації, на виході отримується інформація про сигнатуру бота, заблоковані ботнети, візуалізацію кібератак та інформацію про керуючі компоненти ботнету. Для розуміння, які функції повинні бути включені до процесу захисту Інтернету від атак ботнетів та як ці функції взаємодіють одна з одною, була проведена декомпозиція цього процесу.

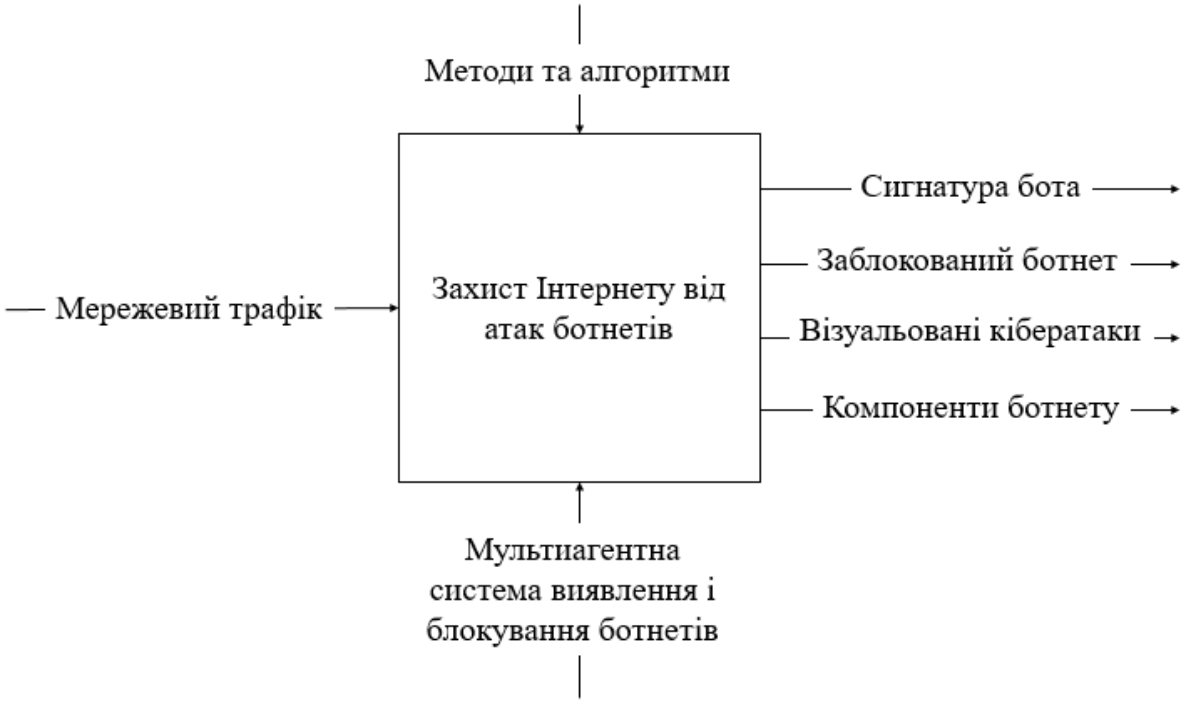

Рисунок 2.1 – Контекстна діаграма функціонування мультиагентної системи виявлення та блокування ботнетів

Внаслідок цього було створено низку процесів, які відображають функціональні аспекти системи, як показано на рисунку 2.2. Ці процеси вирішують різноманітні відомі завдання забезпечення інформаційної безпеки, більш того, в основному ці завдання мають вже визначені рішення:

- завдання виявлення атаки типу «розподілу відмову в обслуговуванні»;
- завдання блокування атаки;
- завдання виявлення характерних ознак роботи бота (вияв керуючого

трафіку та формування сигнатури);

– завдання виявлення бота;
– завдання координації агентів системи;
– завдання контролю та моніторингу роботи агентів;
– завдання накопичення інформації про кіберзагрози;
– завдання візуалізації кібератак і ботнетів.

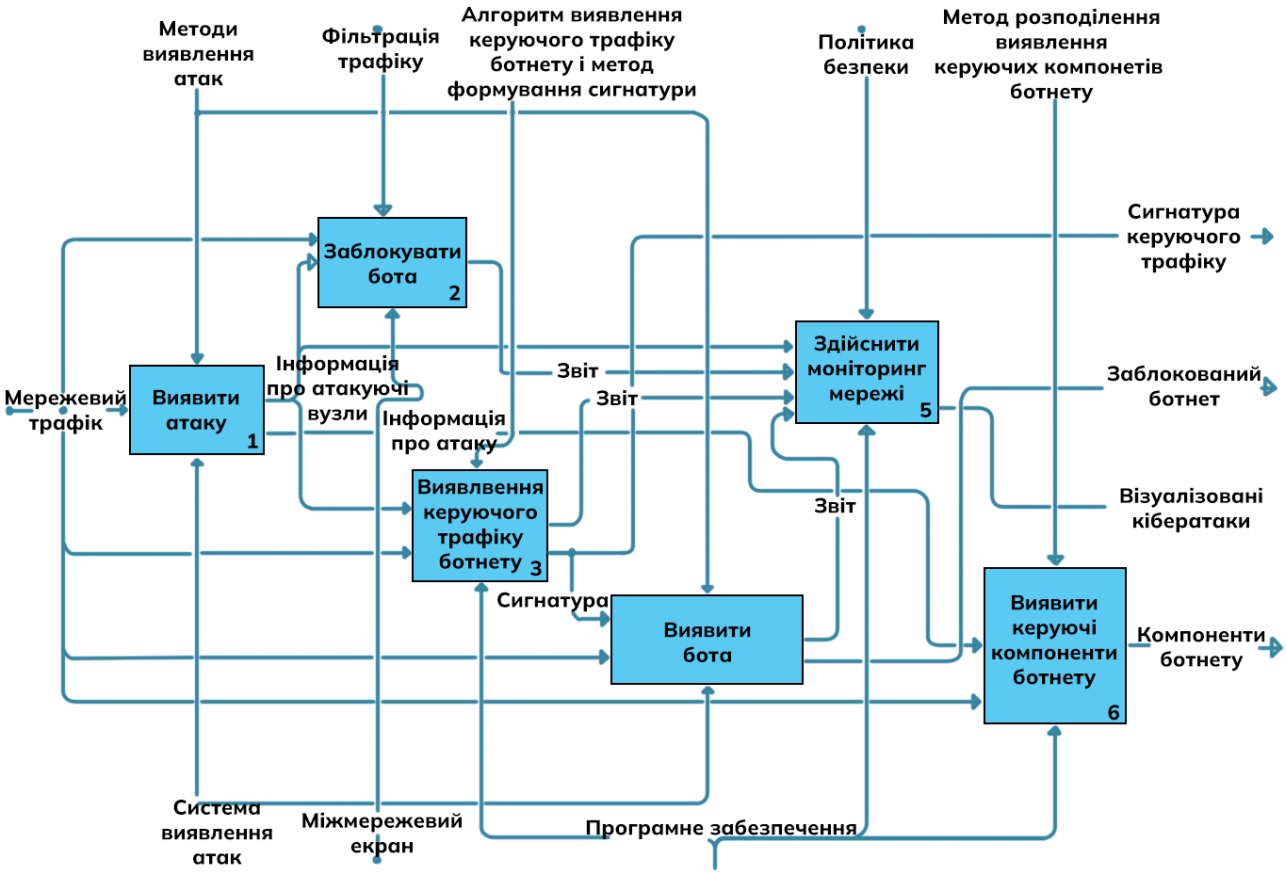

Рисунок 2.2 – Функціональна модель мультиагентної системи виявлення та блокування ботнетів

Для оптимальної функціональності розроблюваної системи необхідно, щоб процеси були розподілені на різних локаціях відносно цілі та джерела атак. Наприклад, процеси "Виявлення атаки" і "Відстеження керуючих вузлів, з яких здійснюється керування атакою", мають бути реалізовані в мережах, де знаходиться цільовий об'єкт атаки. З метою підвищення ефективності процесу відстеження керуючих вузлів атаки, цей процес також може бути реалізований в

мережах, що досліджуються. Процеси "Заблокувати бота" і "Створити сигнатуру" повинні бути реалізовані в мережах джерела атаки. Процес "Створення мережі" має бути реалізований в нейтральних мережах, відносно мети та джерела атаки, хоча він може працювати і в умовах потрапляння в різні мережі. Процес "Ідентифікація бота" має бути реалізований у максимально можливій кількості мереж для більш широкого охоплення процесу виявлення ботів.

Більшість процесів використовують мережевий трафік як вхідні дані. На виході з цих процесів формуються різноманітні дані, які наведено у таблиці 2.1.

Таблиця 2.1 – Результати роботи процесів мультиагентної системи

Процес	Дані на виході
«Виявити атаку»	інформація про атакуючі вузли, інформація про атаку
«Заблокувати бота»	статус блокування трафіка атаки бота
«Виявлення керуючого трафіку»	статус формування сигнатури і після завершення процесу – сигнатура
«Виявити бота»	звіт про виявлення бота за відомою сигнатурою
«Здійснити моніторинг мережі»	представлення всіх даних, отриманих під час роботи мультиагентної системи
«Визначити керуючі компоненти ботнету»	розширена інформація про вузли, що підозрюються в керуванні атакою

Після аналізу функціональної моделі було виявлено, що ефективним рішенням для боротьби з ботнетами може стати система, архітектурно схожа на ботнети. За сутністю, ботнет є мультиагентною системою, де спільна робота агентів призводить до ефективних кібератак. Таким чином, система захисту також має бути мультиагентною, що дозволить досягти високої ефективності в боротьбі як з атаками від продуктивних ботнетів, так і як загальним інструментом для протидії зловмисним мережам. Мультиагентний підхід, фактично, розв'язує проблеми масштабування при збільшенні розмірів системи виявлення. Виявлені характерні

ознаки взаємодії ботів з контролерами ботнетів використовуються для динамічної генерації сигнатур ботів. Ці сигнатури дозволяють виявляти наявність ботів в інших мережах. Такий підхід вирішує проблему автоматизації виявлення ботів.

В якості загальних ознак ботів можна виділити:

– IP-адреси або доменне ім'я контролюючого центру ботнету;

– характеристики HTTP або IRC-пакетів із визначеними командами управління;

– розмірність пакетів;

– часові інтервали мережевих взаємодій;

– трафік зловмисної активності, наприклад, сканування, розсилання спаму, завантаження бінарних файлів;

– інформацію протоколів DNS, SMTP;

– протокол обміну даними і порти транспортного рівня, що використовується.

Отримані у процесі декомпозиції завдання можна віднести до різних класів функціональності: {Виявлення, Блокування, Дослідження, Ідентифікація, Координація, Інтерфейс}. Кожному класу може відповідати свій тип агента, який вирішує завдання класу. Таким чином, мультиагентна система ідентифікації ботнету має вигляд:

$$MAS = \{A_{detection}, A_{blocking}, A_{discrovery}, A_{identification}, A_{coordination}, A_{interface}\},$$

де $A_{detection} = \{A^1_{detection}, \dots, A^n_{detection}\}$ – множина агентів виявлення атаки типу «розподілена відмова в обслуговуванні». Агенти даного класу вирішують завдання виявлення атак і реагують на неї певним у сценарії реагуванні чином. У кожній автономній системі мережі Інтернет розташовується щонайменше один агент даного класу $A^i_{detection}$, де i=1..n - номер автономної системи Інтернету.

$A_{blocking} = \{A^1_{blocking}, \dots, A^n_{blocking}\}$- множина агентів, що вирішують завдання блокування виявленої атаки. У кожній автономній системі мережі Інтернет розташовується щонайменше один агент даного класу $A^i_{blocking}$, де i=1..n - номер автономної системи Інтернету.

$A_{discrovery} = \{ A^1_{discrovery}, \dots, A^n_{discrovery}\}$ – множина агентів виявлення ознак робота. Клас агентів вирішує завдання визначення характерних ознак роботи робота. У кожній автономній системі мережі Інтернет розташовується щонайменше один агент даного класу $A^i_{discrovery}$, де i=1..n - номер автономної системи Інтернету.

$A_{identification} = \{A^1_{identification}, \dots, A^n_{identification}\}$ - множина агентів ідентифікації роботи робота в рамках автономної системи. Агенти цього класу аналізують трафік мережі наявність ознак функціонування ботів. У кожній автономній системі мережі Інтернет розташовується щонайменше один агент даного класу $A^i_{identification}$, де i=1..n – номер автономної системи Інтернету.

$A_{coordination}$ - множина агентів мережі, що вирішують завдання поширення інформації про активних агентів.

$A_{interface}$ - множина агентів мережі, вирішують такі завдання: контроль та моніторинг роботи мережі агентів, візуалізація атак, зберігання інформації.

Таким чином, структура мультиагентної системи ідентифікації ботів складається з наступних елементів, представлених нижче відповідно до рисунка 2.3:

– агент виявлення атаки типу «розподілена відмова в обслуговуванні»;

– агент виявлення ознак робота;

– агент ідентифікації роботів;

– агент блокування атак. Функціонує, коли його розташування є мережею джерела атаки. Зокрема, здійснює реагування на основі інформації отриманої від агентів виявлення атак згідно з профілем мережевої безпеки (блокування систем задіяних у реалізації атаки, оповіщення відповідальних осіб по електронній пошті, SMS);

– агент координації. Поширює інформацію про місцезнаходження різних агентів з метою взаємодії між ними;

– інтерфейсний агент. Встановлюється у будь-якій точці глобальної мережі Інтернет. Призначений для контролю та моніторингу роботи мережі агентів,

надання графічного інтерфейсу візуалізації виявлених атак, зберігання та забезпечення доступу до історії виявлених атак.

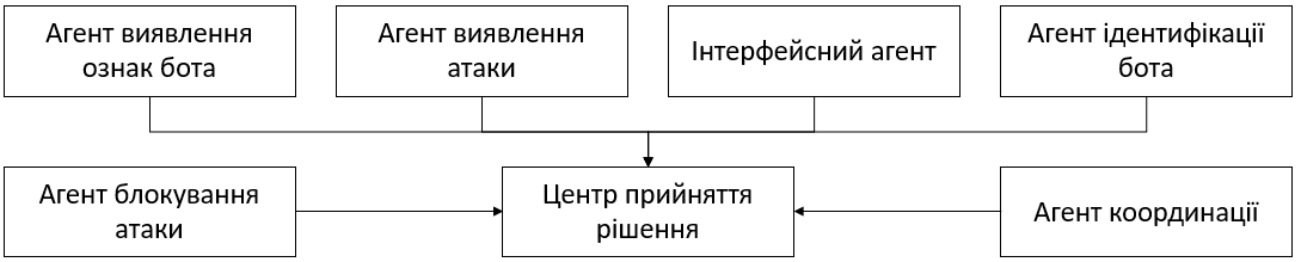

Рисунок 2.3 – Структура мультиагентної системи виявлення ботнетів

Концептуальний алгоритм функціонування системи можна описати наступним чином:

– Агент, який виявляє атаки типу "розподілена відмова в обслуговуванні", виявляє атаку в мережі, яка підконтрольна йому.

– Агент виявлення атаки повідомляє агенту координації інформацію про мережі, з яких походить атака.

– Агент координації передає інформацію про атаку агентам блокування, які знаходяться в автономних системах, які є джерелом атаки. Ця інформація стосується вузла, що атакує.

– Агент координації передає інформацію про виявлення ознак бота, який контролює мережу, з якої здійснюється атака, агенту виявлення ознак бота.

– Агент координації також повідомляє інтерфейсному агенту про атаку.

– Агент блокування припиняє зловмисну активність вузлів, які знаходяться в контрольованій мережі.

– Агент виявлення ознак бота аналізує активність вузлів, що взяли участь у атаку, та виявляє характерні ознаки роботи бота.

– Агент виявлення ознак бота передає характерні ознаки агенту координації.

– Агент координації розсилає інформацію про роботу ботів агентам ідентифікації ботів.

– Агенти ідентифікації аналізують трафік в своїй мережі, намагаючись виявити ознаки роботи бота. У разі успішної ідентифікації вони передають інформацію про бота агенту координації, який направляє її інтерфейсному агенту для подальшого прийняття рішення.

Моделі агентів (показані на рисунку 2.4) створені для представлення процесів, які виконують агенти завдання. Вони включають базові та спеціальні функції агентів, сценарії їх поведінки та протоколи взаємодії.

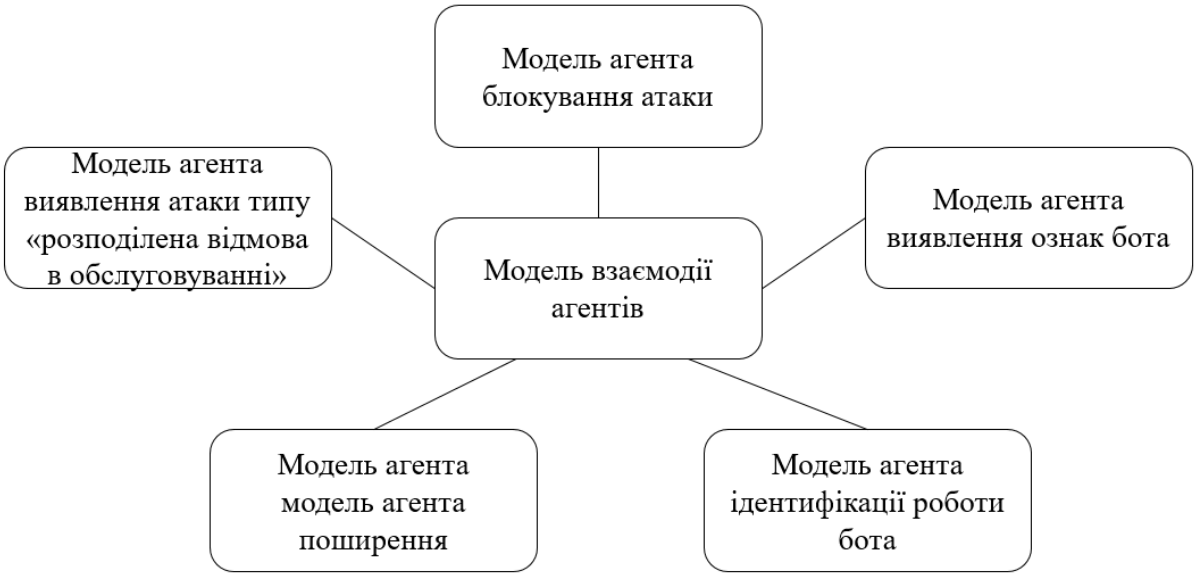

Рисунок 2.4 – Представлення основних моделей системи виявлення ботнетів

Основні функції агентів включають такі завдання: ініціалізація та завершення роботи, ведення списків активних агентів і взаємодія з модулями транспортного рівня. Крім базових функцій, деякі агенти можуть мати спеціалізовані функції, які засновані на основних функціях. Наприклад, для агентів виявлення атаки реалізація буде залежати від використаного методу виявлення, для агентів виявлення ознак бота - від методів аналізу діяльності бота, а для агентів блокування - від політики реагування на атаку. Для агентів виявлення метод роботи буде залежати від методу аналізу мережевого трафіку.

Протоколи взаємодії агентів представлені у вигляді послідовності команд з конкретними параметрами. У роботі для обміну повідомленнями між агентами

використовуються протоколи TCP і UDP. Вибір конкретного протоколу базується на доступних ресурсах комунікації.

Поведінка агентів може варіюватися залежно від сценаріїв. У деяких випадках сценарії конкретних агентів можуть бути обумовлені політикою безпеки, що приймається в системі виявлення.

Існує безліч моделей мультиагентних систем, кожна з яких акцентує увагу на певному аспекті системи. Для поставленої задачі найбільш доцільною є модель, яка базується на концепції алгебраїчної системи. Ця модель є успішною завдяки наступним аспектам:

– доступність: здатність агентів інтегруватися в системи з метою вирішення складних завдань разом;

– дозволяє розширити опис окремих агентів та уявлення про мультиагентну систему в цілому;

– фокусується на описі обмеженої множини можливих дій;

– спрямована на створення моделей штучних агентів.

Таким чином, MAS можна виразити таким чином: $\mathrm{MAS} = (\mathrm{A}, \mathrm{ENV}, \mathrm{INT}, \mathrm{ORG})$, де A – множина агентів; ENV – середовище, в якому знаходиться MAC; INT – типи взаємодій між множиною агентів; ORG – множина базових організаційних структур, відповідно конкретним функціям (ролям) агентів і зв’язками, що між ними встановлено.

Для опису введено множину INT, яка представляє взаємодії агентів між собою та агентів з навколишнім середовищем ENV. У цілях цього опису використовано три мови на різних рівнях. Ці мови виконують різні функції комунікації: мова взаємодії з іншими агентами (L2), мова локального планування (L1), мова рівня виконання (L0). Ця архітектура дозволяє створити багаторівневий агент, розподіляючи його функціональні можливості на кілька ієрархічних рівнів. Кожен рівень взаємодіє з іншими відповідно до ієрархії. Прикладом такої архітектури є архітектура InterRRaP (об'єднання реактивної поведінки та раціонального планування). Взаємодія за допомогою мови Lx позначається як int(Lx). Отже, $\mathrm{INT} = (\{\mathrm{int(L2)}\}, \{\mathrm{int(L0)}\})$. Мова L1 використовується для

планування дій агента в межах організаційних структур ORG.

Окремий агент, в рамках обраної моделі, може бути описаний як четвірка $A_i = (E_i, I_i, ORG_i, C)$, де E_i – елементи комунікаційної середовища, включаючи джерела інформації $E_i \subseteq ENV$; I_i – підмножина зв'язків даного агента з іншими $I_i \subseteq INT$; ORG_i – підмножина, що містить функції агента, що виконуються в загальній структурі мультиагентної системи; C – внутрішня функціональна структура агента.

Внутрішню функціональну структуру окремого агента можна представити таким чином $C = (K, F, I, G, B)$, де К - підсистема - ядро, відповідальне за динамічну реалізацію ORG; F – підсистема, відповідальна за виконання конкретних функцій агента; I – підсистема, що відповідає за взаємодію з джерелами інформації; G – підсистема, що відповідає за взаємодію з іншими агентами; B – база знань агента. Метамодель агента представлена на рисунку 2.5.

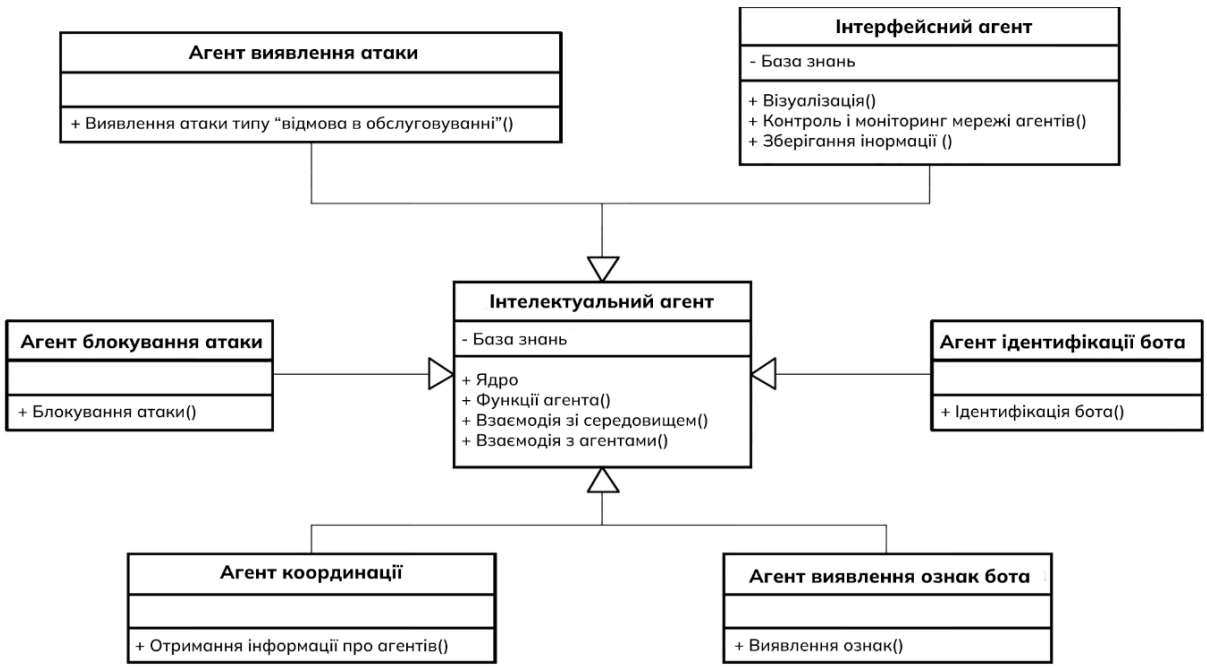

Рисунок 2.5 – Метамодель агента

Центральний блок метамоделі надає опис базового агента, який є основою для побудови основних агентів у системі. Додаткові блоки розкривають структуру цих основних агентів і відображають спеціальні функції, які прямо залежать від ролі агента в системі.

2.2 Поєднання агентів

Описана архітектура мультиагентної системи базується на архітектурі InterRRaP. Ця архітектура відповідає концепції змішаної архітектури, що поєднує в собі реактивність та можливість планування і відображає агента як багаторівневу сутність. Основний метод взаємодії між агентами - це кооперація. Кооперація полягає в об'єднанні зусиль агентів для досягнення загальної мети, і при цьому ролі, функції та обов'язки розподіляються між агентами.

Для опису взаємодії агентів один з одним і з навколишнім середовищем використовуються три рівні мов. Ці мови мають наступні комунікаційні функції: взаємодія з іншими агентами (L2), локальне планування (L1) і виконавчі функції (L0). Основна структура мультиагентної системи представлена на рисунку 2.6.

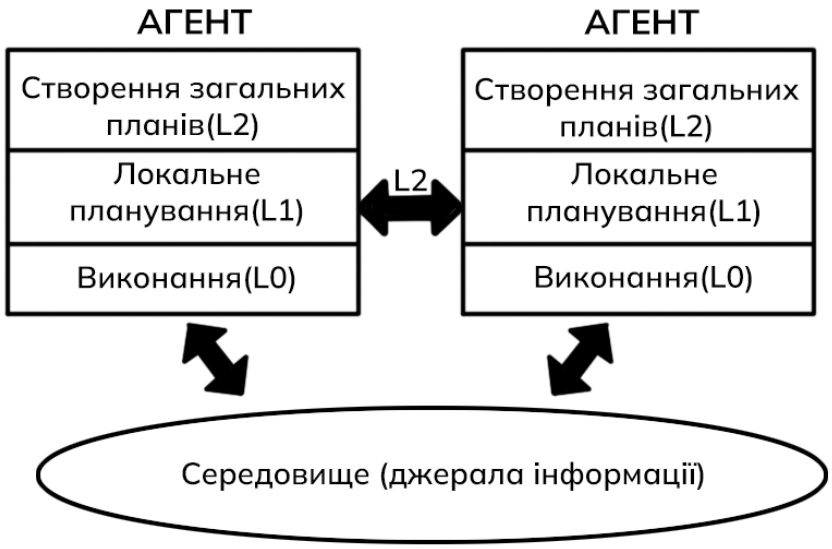

Рисунок 2.6 - Базова архітектура мультиагентної системи

Функціональний опис ролей в мультиагентній системі включає такі основні ролі агентів:

- виявлення розподільної атаки типу «відмова в обслуговуванні»;
- координатор;
- блокування атаки;

– дослідження трафіку скомпрометованого вузла;
– формування сигнатури;
– ідентифікація бота;
– інтерфейс.

Модель взаємодій між агентами може бути описана наступною взаємодією між агентами:

– агент виявлення розподільної атаки – агент-координатор;
– агент-координатор – агент блокування атаки;
– агент-координатор – агент дослідження трафіку скомпрометованого вузла;
– агент дослідження трафіку скомпрометованого вузла – агент-координатор;
– агент-координатор – агент формування сигнатури;
– агент формування сигнатури – агент-координатор;
– агент-координатор – агент ідентифікації;
– агент ідентифікації – агент-координатор;
– агент-координатор – агент-інтерфейс;
– агент-інтерфейс – агент-координатор.

Найпоширенішими спеціальними моделями співпраці агентів у розподільному штучному інтелекті є: модель аукціонів, протокол монотонних мінімальних поступок, модель договірних мереж, модель соціальних залежностей.

Враховуючи опис мультиагентної системи виявлення ботнетів, найбільш відповідною моделлю для співпраці агентів є модель договірних мереж, відома як модель Сміта. Ця модель забезпечує координацію дій агентів у розподільних системах. Агенти можуть виконувати певні завдання. У випадку, коли один агент намагається вирішити завдання самостійно, він може звернутися до іншого агента для допомоги. У такому випадку агент, який шукає допомогу, виступає замовником, а інші агенти можуть виконувати роль виконавців. Зазвичай замовник не шукає виконавця самостійно, а користується послугами посередника, який

виступає брокером. У ролі брокера виступає агент-менеджер. Агент-менеджер може бути реалізований як мобільний агент, який переміщується по мережі. Інші агенти, зазвичай, розташовані на своїх вузлах мережі.

Стандартна модель Сміта передбачає, що для вибору найбільш відповідного виконавця заявки організовується конкурс між усіма агентами, які відгукнулися на запит замовника. Пропонується, що агент-менеджер здійснює вибір виконавця за заданим критерієм. Це дозволяє уникнути основних недоліків даної моделі, таких як високе навантаження на канали передачі даних, що має особливе значення для використання цієї моделі, а також відсутність продуманого методу вибору виконавця.

У вирішуваній задачі ролі агентів розподілені наступним чином: агент-менеджер виступає агентом-координатором; агентами-замовниками є агент виявлення розподільної атаки, агент виявлення характеристик роботи бота, агент-ідентифікатор та агент-інтерфейс; агентами-виконавцями можуть бути агенти блокування атак, агент виявлення характеристик роботи бота, агент-ідентифікатор і агент-інтерфейс. Виходячи із моделі, процес взаємодії агентів можна описати набором $\mathrm{INT} = (\mathrm{A}, \mathrm{RR}, \mathrm{p}, \mathrm{P})$, де A – множина агентів, RR – множина ролей агентів, p:A →RR – функція розподілу ролей, P – загальний протокол взаємодії між агентами.

В цьому випадку RR={«виявлення розподільної атаки», «координатор», «блокування атаки», «дослідження трафіка бота», «формування сигнатури», «виявлення бота», «інтерфейс»}. Функція розподілу ролей визначається наступним способом:

$p(A^i_{detection})$ = «виявлення розподіленої атаки»,

$p(A^i_{blocking})$ = «блокування атаки»,

$p(A^i_{discovery})$ = «дослідження трафіку бота»,

$p(A^i_{feature})$ = «формування сигнатури»,

$p(A^i_{identification})$ = «виявлення бота»,

$p(A^i_{coordination})$ = «координатор»,

$p(A^i_{interface})$ = «інтерфейс».

Протокол взаємодії встановлює набір правил і угод, які регулюють спосіб спілкування. Цей протокол описується спеціальною мовою, яка використовується для створення спільних планів і взаємодії між агентами - ця мова відома як L2. Основна мета мови L2 в межах мультиагентної системи полягає в наступному:

– складання запитів агентів між собою;

– складання запитів до мультиагентної системи;

– складання відповідей на запити.

Можна розглядати мову L2 як сукупність компонентів, де L2 = (Bnmar, Response, Shell).

Bnmar - це основні типи взаємодії, які визначають конкретні завдання агента в мультиагентній системі. Ці примітиви визначають організаційну структуру агента.

Response - це результати обробки запитів, які формуються у доступній формі, придатній як для сприйняття людиною, так і для інших агентів.

Shell - це мова запитів оператора до мультиагентної системи, яка призначена для виконання оператором дій з моніторингу та контролю роботи системи.

Отже, можна виділити кілька режимів кооперативної взаємодії агентів у системі: режим незалежної взаємодії агентів, коли кожен агент діє незалежно; режим управління для агента Ai, коли він видає команди агенту Aj; режим підпорядкування для агента Ai, коли він виконує завдання агента Aj.

2.3 Розробка алгоритму виявлення керуючого трафіку

Розроблений алгоритм спроможний виявляти трафік ботнету незалежно від використовуваного протоколу чи організаційної структури ботнета.

Для створення загального підходу до виявлення керуючого трафіку, що здатен протистояти еволюції та змінам методів управління ботнетами, необхідно ретельно вивчити внутрішні характеристики комунікації ботнетів, які можна

використовувати у алгоритмах та функціях виявлення. Ми можемо визначити термін "ботнет" наступним чином: це "скоординована група шкідливих об'єктів (ботів), які контролюються за допомогою бот-майстра через командно-контролюючий канал". У цьому контексті "шкідливі" вказує на те, що ці боти використовуються для здійснення шкідливих дій. Наприклад, зібрані дані показують, що близько 53% шкідливих дій, які зафіксовані у тисячах реальних ботнетів, пов'язані із скануванням (поширенням або DDoS-атаками), тоді як близько 14% пов'язані із завантаженням бінарних файлів (для оновлення шкідливого ПЗ). Більшість ботнетів, що базуються на протоколі HTTP або P2P, використовуються для розсилки спаму. Термін "контрольовані" вказує на те, що боти повинні отримати команди для своєї діяльності, зв'язавшись із серверами C&C (командно-контролюючими). Іншими словами, існує зв'язок між ботами та керуючими серверами, які можуть бути централізованими або розподіленими. Термін "скоординована група" означає, що кілька (не менше двох) ботів у межах одного ботнету взаємодіють через мережу та виконують аналогічні та пов'язані зловмисні дії під керівництвом керуючих серверів.

Алгоритм відстежує комунікаційні зв'язки у термінах "хто спілкується з ким" на кожному етапі роботи. Він об'єднує цю інформацію із різних точок, проводить аналіз, порівнюючи кластери мережевого трафіку. Припускаючи наявність активності в комунікаціях з керуючим центром, алгоритм визначає шаблони скоординованих груп. Іншими словами, алгоритм виконує кластерний аналіз схожих комунікаційних активностей у керуючому трафіку C&C і використовує кросс-кластерну кореляцію для визначення кластерів, що ділять спільну модель спілкування. Це дозволяє створювати сигнатури для конкретних ботнетів, аналізуючи їх керуючий трафік.

Алгоритм використовує два типи агентів: агентів дослідження трафіку та агентів формування сигнатур. Агенти дослідження трафіку відстежують трафік з компрометованих вузлів і групують його за подібністю. Агенти формування сигнатур виконують кросс-кластерну кореляцію між усіма кластерами, отриманими від агентів дослідження трафіку з усіх компрометованих вузлів. Це

дозволяє визначити кластер трафіку зі схожими шаблонами комунікацій, який використовується для створення сигнатур. Структура роботи агентів дослідження трафіку та формування сигнатур описана нижче, згідно до рисунку 2.7.

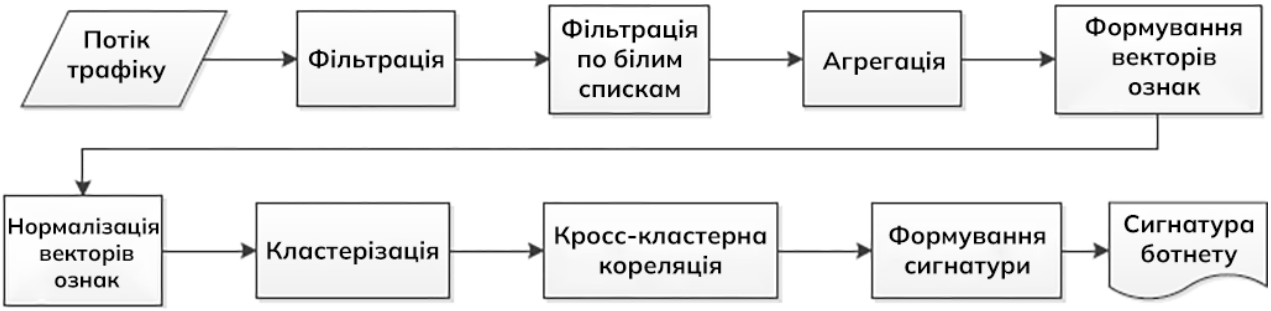

Рисунок 2.7 – Структура роботи агентів дослідження трафіку та формування сигнатури

Перш за все, потрібно провести фільтрацію непотрібних потоків трафіку. Це виконується у два етапи: базова фільтрація та фільтрація за допомогою білих списків. Важливо відзначити, що ці етапи не критичні для нормальної роботи процесу кластеризації, але вони корисні для зменшення навантаження на трафік і підвищення ефективності процесу кластеризації, а також фільтрації трафіку легітимних користувачів. На етапі базової фільтрації відкидаються всі потоки, які не мають напряму від внутрішніх хостів до зовнішніх хостів. Це означає, що алгоритм ігнорує потоки, які виникають між внутрішніми вузлами мережі, а також ті потоки, які ініціюються зовнішніми вузлами та спрямовані до внутрішніх. Також відфільтровуються потоки, які не перебувають у стані «ESTABLISHED». Фільтрація за білими списками відсіює всі потоки, що спрямовані на відомі легітимні сервери (наприклад, Google). Припускається, що такі сервери не можуть бути серверами C&C. Список цих серверів формується на основі їхньої популярності. Для кожної країни або регіону також може бути створений індивідуальний список відомих ресурсів як доповнення до загального списку.

Після фільтрації наступним етапом є агрегація пов'язаних комунікаційних потоків з метою зменшення навантаження. За часовим інтервалом E (зазвичай

кілька годин) всі TCP/UDP потоки, які відносяться до одного і того ж протоколу (TCP або UDP), мають однакові джерело, призначення і порт, об'єднуються в один комунікаційний потік ci={fj}j=1..m, де кожне значення fj – це окремий потік TCP/UDP. Множина {ci}i=1..n об'єднує всі n комунікаційних потоків, які спостерігалися протягом інтервалу E, і відображає комунікацію хоста, за яким ведеться спостереження.

Завданням побудови моделі комунікації є визначення комунікаційних потоків, які є спільними для всіх спостережуваних вузлів. Це можна досягти шляхом групування аналізованих потоків у кластери. Для того, щоб застосувати алгоритми групування до комунікаційних потоків, спершу необхідно подати ці потоки у векторному вигляді. Для цього в алгоритмі використовуються різноманітні статистичні ознаки з кожного комунікаційного потоку ci, які перетворюються у d-вимірний вектор pi <Rd. Можна описати цей процес як функцію F:C→ Rd. Функція F визначається таким чином: для кожного комунікаційного потоку ci обчислюються дискретні розподіли чотирьох випадкових величин:

– потоків за годину (flows per hour, FPH): FPH обчислюється шляхом підрахунку кількості TCP/IP-потоків в ci, які присутні кожну годину протягом часового інтервалу E;

– кількість пакетів в потоці (packets per flow, PPF): PPF обчислюється шляхом додавання загальної кількості пакетів, які надсилаються в кожному TCP/UDP потоці ci ;

– середнє число байт в пакетах (bytes per packets, bpp). Для кожного TCP/UDP потоку fj<ci обчислюється загальна кількість байт, переданих в fj, на кількість пакетів, відправлених в fj;

– середня кількість байт в секунду (bytes per second, bps).). BPS обраховується як загальна кількість байт, переданих у кожному потоці fj<ci, поділене на тривалість fj.

Приклад результату цього процесу показано на рисунку 2.8, на якому відображено особливості відвідувань соціальної мережі «Facebook» користувачем,

випадково вибраним із журналу реального мережевого потоку.

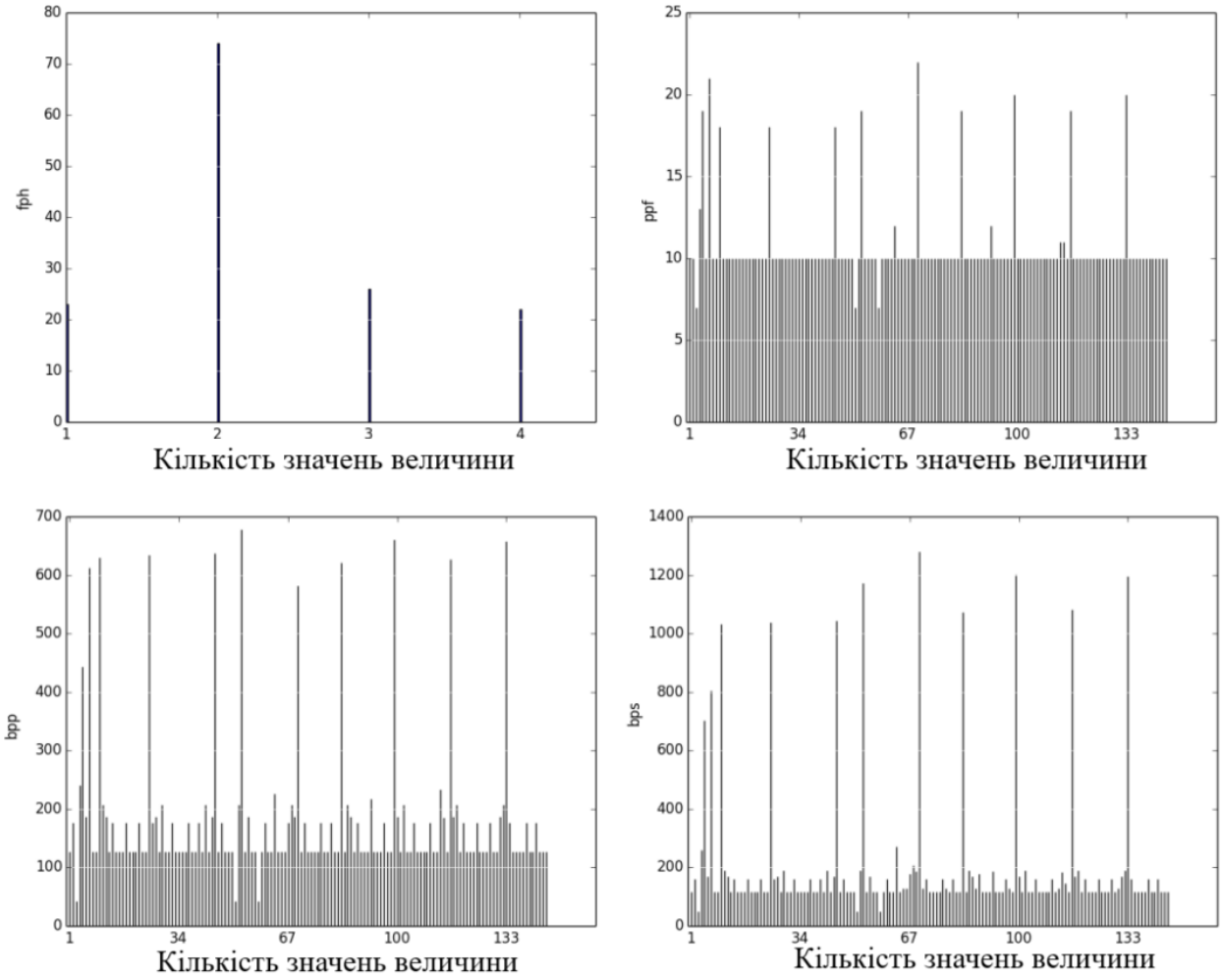

Рисунок 2.8 – Особливості відвідувань соціальної мережі «Facebook» користувачем, випадково вибраним із журналу реального мережевого потоку

Враховуючи дискретний характер розподілу вибірки кожної з цих чотирьох випадкових величин, обчислюється наближена варіація за допомогою техніки групування даних (бінінгу). Наприклад, для того, щоб наблизити розподіл fph, розіб'ємо вісь x на 8 інтервалів у такий спосіб: $[0, k_1]$, $(k_1, k_2]$, …, $(k7, \infty)$. Значення $k_1,\ldots,k_7$ обчислюються наступним чином. Спочатку розраховується дискретний розподіл вибірки fph на основі всіх комунікаційних потоків трафіку на інтервалі E. Потім обчислюються квантилі $q_{10\%}$, $q_{20\%}$, $q_{30\%}$, $q_{40\%}$, $q_{50\%}$, $q_{70\%}$, $q_{90\%}$ отриманого розподілу і встановлюємо $k_1=q_{10\%}$, $k_2=q_{20\%}$, $k_3=q_{30\%}$ і так далі. Тепер для кожного комунікаційного потоку можна описати його fph-розподіл у якості вектора із 8

елементів, де кожен елемент і представляє число разів fph прийнятих значення в межах відповідного інтервалу (k_{i-1} , k_i]. Даний алгоритм застосовується також і для ppf, bpp, bps, і тому можна відмалювати кожен комунікаційний потік c_i у вектор-шаблон p_i із d=32 елементів.

Масштабований шаблон відвідувань, витягнутий із комунікаційного потоку взаємодій із соціальною мережею «Facebook», показано на рисунку 2.9.

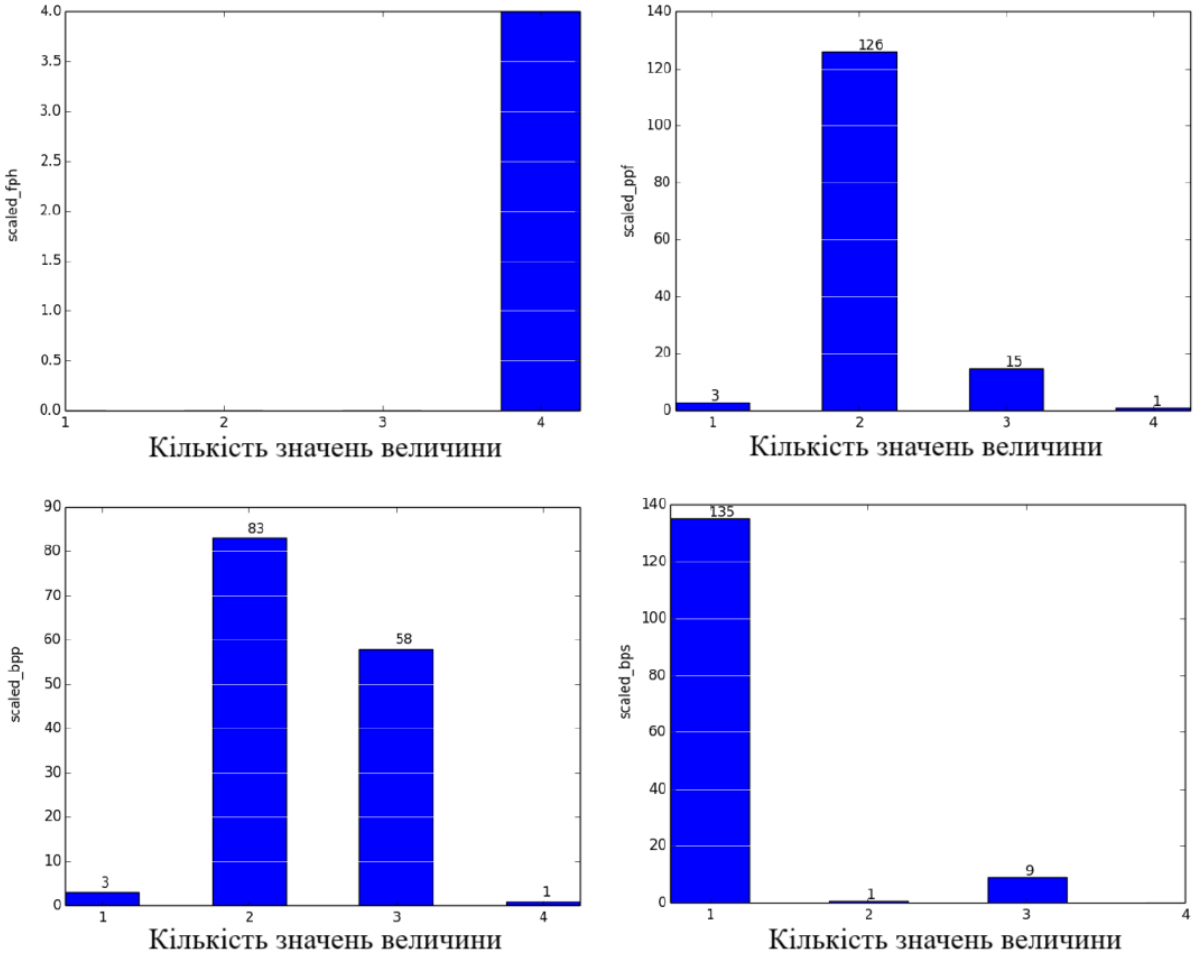

Рисунок 2.9 – Масштабований шаблон відвідувань

Наступним етапом є проведення кластеризації комунікаційних потоків з метою виявлення груп потоків, які мають схожі характеристики, і можуть вказувати на трафік до серверів C&C. За допомогою цієї техніки, ми намагаємося знайти комунікаційні потоки, які мають близькі векторні представлення в просторі Rd, що свідчить про схожу комунікаційну структуру. Наприклад, якщо два боти з

одного ботнету підключені до різних C&C-серверів (деякі ботнети можуть використовувати кілька серверів управління), їхні характеристики трафіку повинні бути схожими. Іншими словами, векторні представлення цих комунікаційних потоків в просторі Rd мають бути дуже схожими. Для виявлення керуючого трафіку групи вузлів, які мають схожі шаблони зв'язку, застосовується техніка кластеризації на наборі даних D, який представляє векторне уявлення комунікаційних потоків. Кластеризація проводиться методами навчання без учителя, спрямованими на пошук значущих груп у даних у просторі ознак F. Оцінка "значущих кластерів" залежить від конкретної області застосування. Головною метою є групування даних у кластери, які є одночасно компактними та добре відокремлюються один від одного, і вибір метрики подібності зазвичай здійснюється у просторі ознак F.

Кластеризація комунікаційних потоків становить виклик, оскільки набір даних D часто є об'ємним, навіть для середніх мереж, і розмірність d-простору ознак також значно зростає. Крім того, враховуючи те, що відсоток комп'ютерів у мережі, що заражені ботами, як правило, дуже невеликий, потрібно розрізняти кілька ботнетів, які можуть використовувати однакові комунікаційні потоки, від великої кількості нормальних комунікаційних потоків. Усе це ускладнює завдання кластеризації мережевих потоків.

Кластеризація мережевих потоків виконується у два етапи, як це показано на рисунку 2.10. Такий підхід допомагає вирішити проблеми складності, пов'язані з кластеризацією.

На першому етапі проводитиметься первинна кластеризація у зменшеному просторі ознак $R^{d'}$, з d'<d, використовуючи простий алгоритм кластеризації. Зменшення розмірності простору ознак з d=32 ознаки до d'=8 ознак шляхом обчислення середнього і дисперсії розподілу fph, ppf, bpp і bps для кожного комунікаційного потоку. Потім застосовується алгоритм кластеризації X-means на отриманому поданні комунікаційного потоку для надходження великих кластерів $\{C_i'\}_{i=1..y_1}$. Результатом першого кроку кластеризації є множина $\{C_i'\}_{i=1..y_1}$, де y_1 – відносно великі кластери. Таким чином, набір даних D розбивається на менші

набори даних (кластери C'_i), що містять набори точок, не дуже віддалених одна від одної. Оскільки кластери $\{C'_i\}_{i=1..y_1}$. генеруються на першому кроці кластеризації, то їх кількість не дуже велика, далі можна провести більш трудомісткий аналіз для кожного C'_i).

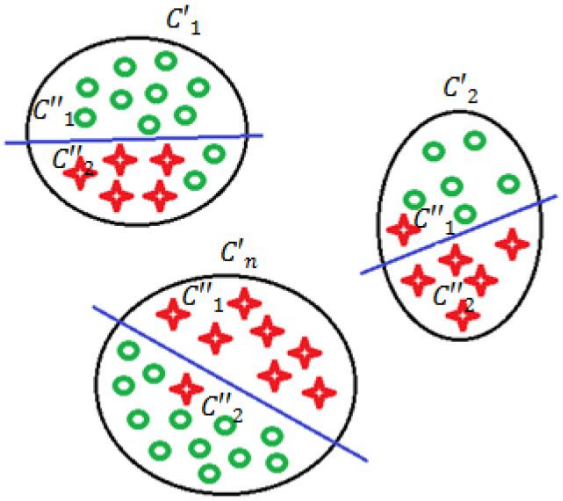

Рисунок 2.10 – Двоетапна кластеризація

Для подальшого вдосконалення результатів першого етапу кластеризації проводиться другий етап кластеризації на окремих наборах даних і, використовуючи простий алгоритм кластеризації, що враховує повний опис комунікаційних потоків у Rd. Тобто на другому етапі кластеризації використовуються всі d=32 ознаки для представлення комунікаційного потоку. Другий крок генерує набір y_2 невеликих і більш точних кластерів $\{C''_i\}_{i=1..y_2}$.

Кластеризація на першому та другому етапах була здійснена за допомогою алгоритму XMeans. XMeans, як ефективний алгоритм, базується на популярному методі кластеризації K-means. На відміну від K-means, XMeans не потребує від користувача заздалегідь визначати кількість кінцевих кластерів. Він виконує кілька ітераційних проходів K-means та ефективно оцінює кластеризацію, використовуючи критерій Байєса для визначення найкращої кількості кластерів. XMeans масштабується легко та швидко в залежності від розміру набору даних.

Оскільки завдання "навчання без учителя" є складним, результати двоетапного алгоритму кластеризації можуть бути не завжди ідеальними. Внаслідок цього комунікаційні потоки, пов'язані з ботнетом, можуть розділятися між декількома різними кластерами.

Після отримання результатів кластеризації, алгоритм виконує крос-кластерну кореляцію. Це полягає в виділенні кластерів, які мають найбільший перетин серед усіх кластерів, що були сформовані на скомпрометованих вузлах. Іншими словами, ці кластери повинні бути найбільш близькими один до одного. У процесі кластерного аналізу для кількісної оцінки ступеня близькості вводиться поняття метрики. Визначення схожості і відмінності між об'єктами визначається на основі ступеня перетину двох кластерів. Для цього використовується наступний вираз в якості міри близькості: $sim(C_i'', C_j'') = \max\limits_{i \neq j}{}_{i,j=1,y_2} \left(\left|C_i'' \cap C_j''\right|\right)$.

2.4 Метод розподіленого виявлення керуючих компонентів ботнета, з яких здійснюється контроль атаки ботнету

У рамках реалізації певної атаки можна виділити дві основні ролі серед зловмисників: ініціатора атаки та виконавця атаки. Більшість існуючих систем захисту від ботнетів акцентують увагу на виявленні та блокуванні керуючих серверів ботнетів та самих ботів. Однак це не вирішує основної проблеми - ідентифікації зловмисників, які мають свої мотивації. Навіть якщо існуючі захисні методи можуть виявити та зупинити роботу ботнету, вони не можуть встановити особистість та місцезнаходження зловмисників. Проблему ботнетів не можна остаточно вирішити, поки не буде розроблено методику виявлення зловмисників в Інтернеті. У світлі цього усвідомлення своєї відповідальності, зловмисники можуть намагатися створювати та керувати іншими ботнетами з більшою обережністю. Якщо ризик бути впізнаним збільшується, то частина зловмисників може утримуватися від створення та використання ботнетів. Таким чином, навіть недосконалий метод ідентифікації зловмисників може ефективно стримувати їхню

активність.

Завдання відстеження зловмисника, який використовує ботнет, який було виявлено, є досить складним. Зловмисник може бути онлайн лише протягом обмеженого періоду часу: для надсилання команд ботнету або перевірки статусу ботів. Тому відстеження повинно відбуватися в режимі реального часу. Крім того, зловмисники часто не підключаються безпосередньо до керуючих серверів. Додатково, вони можуть зашифровувати свій керуючий трафік. Наприклад, ботнет Agobot підтримує протоколи SSL/TLS для забезпечення шифрування комунікації. Отже, ефективний метод відстеження зловмисника повинен працювати ефективно з невеликими обсягами зашифрованого трафіку. Жоден із існуючих методів відстеження зловмисника не здатен виконувати це завдання ефективно в режимі реального часу. Наприклад, деякі методи потребують великих обсягів трафіку, щоб мати можливість аналізувати зашифрований трафік, який проходить через ланцюг проксі-серверів. Протягом сесії боти обмінюються лише обмеженою кількістю пакетів з бот-майстром. Зважаючи на невеликий обсяг трафіку, зазначені методи стають неефективними для виявлення бот-майстра.

Існують методи виявлення бот-майстра, які базуються на аналізі трафіку або заражених вузлів, або на аналізі трафіку керуючого центру.

Один із ключових кроків при виконанні розподільних атак, таких як атаки "відмова в обслуговуванні", - це контроль за виконанням атаки. Процес цього контролю зображений на діаграмі прецедентів організації атаки, яка використовує ботнет (див. рисунок 2.11). Учасники атаки, організатори включно, мають інтерес в тому, щоб атакований сервіс залишався недоступним протягом всього періоду атаки. Для досягнення цієї мети вони використовують різне програмне забезпечення.

Оскільки багато подібних атак спрямовані на веб-ресурси, контроль доступності виконується за допомогою стандартних браузерів або утиліт, що використовують протокол ICMP. Ці утиліти, наприклад, "ping", надсилають ICMP-запити "echo request" і обробляють отримані пакети "echo reply", щоб визначити затримки та частоту втрат пакетів шляхом вимірювання часу між відправкою

запиту і отриманням відповіді. Це дозволяє визначити стан зайнятості каналів передачі даних і доступність віддаленого вузла. У разі успішної атаки, віддалений вузол стає недоступним.

З боку атакованого вузла активність цього роду може бути поміченою завдяки великій кількості ICMP-запитів "echo request". Це може навести на підозру стосовно джерел цих запитів і надати список адрес джерел. Оскільки однією з мет мультиагентної системи є підтримка процесу кіберрозслідувань, цей список адрес, разом із додатково зібраною інформацією, може стати корисним для ідентифікації злочинців та притягнення їх до відповідальності.

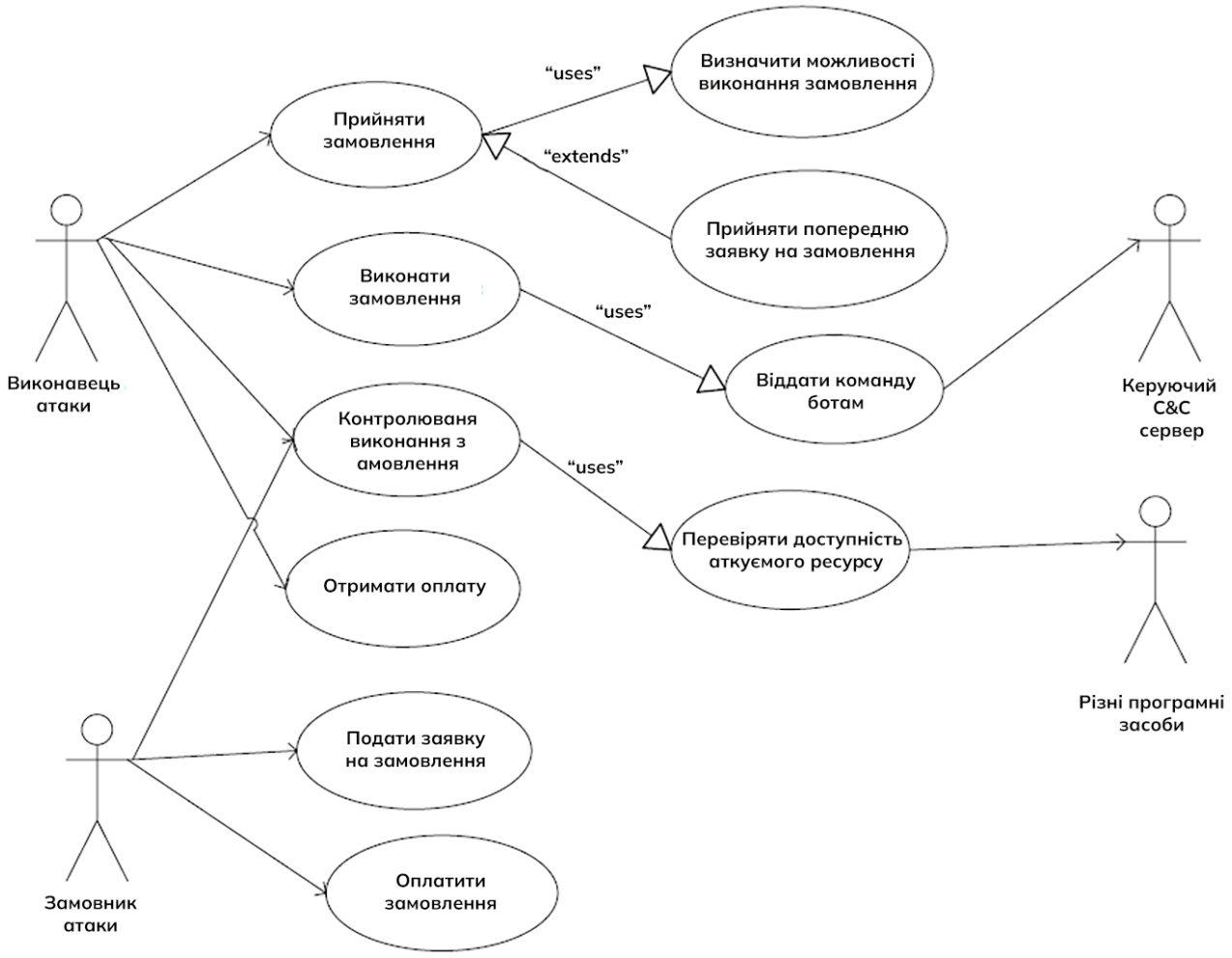

Рисунок 2.11 – Діаграма прецедентів процесу організації атаки, що здійснюється за допомогою ботнету

Функції, які були описані у цьому розділі, втілені у модулі, який відстежує

вузли, що керують атакою. Цей модуль інтегрований в склад агента для виявлення атаки. Структура цього модуля представлена на діаграмі, зображеній на рисунку 2.12.

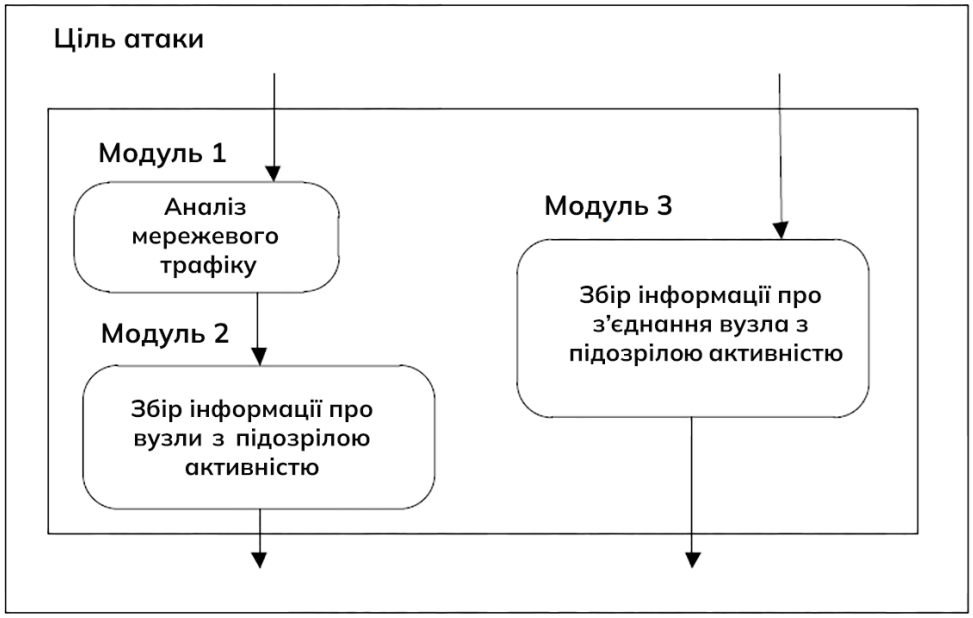

Рисунок 2.12 – Модуль відображення вузлів, що контролюють атаку

Модуль складається з трьох підсистем: аналіз мережевого трафіку, збір інформації про вузли з підозрілою активністю та збір інформації про з'єднання вузла з підозрілою активністю. Цей модуль активується, коли агент виявлення атаки реєструє атаку. Під час аналізу мережевого трафіку здійснюється підрахунок всіх вхідних ICMP-запитів типу «echo request» з унікальних IP-адрес. Ця інформація складає таблицю, де кожній IP-адресі відповідає кількість отриманих ICMP-запитів. Таблиця сортується за зменшенням кількості отриманих ICMP-запитів. У верхній частині таблиці зазвичай розташовані IP-адреси вузлів, які контролюють атаку типу «відмова в обслуговуванні». Ця таблиця передається на модуль збору інформації про вузли з підозрілою активністю. Робота цього модуля полягає в зборі інформації про виявлені вузли, включаючи:

– тип і версія операційної системи;
– тип пристрої;

- запущені служби;
- порти;
- маршрут слідування даних;
- DNS інформація;
- реєстраційні дані whois;
- географічне розташування вузла;
- вразливість вузла.

Підсистеми аналізу мережевого трафіку та збору інформації про вузли з підозрілою активністю спрямовані на визначення цілей атаки. Підсистема збору інформації про з'єднання вузлів працює на агентах, які розташовані якомога ближче до виявлених вузлів з підозрілою активністю. Ця підсистема активується на завершальному етапі роботи всієї мультиагентної системи, коли вже відомі адреси керуючих серверів. Це дозволить відслідковувати взаємодію бот-майстра з керуючим сервером. Під час роботи цього модуля збір інформації про вузли, які контролюють атаку, вся інформація збирається агентом моніторингу.

Слід зауважити, що запропонований метод застосовується в основному при розподільних атаках типу «відмова в обслуговуванні». Хоча цей метод дуже ефективний для таких атак, він може бути менш ефективним для інших видів кібератак.

2.5 Висновки до розділу

У цьому розділі представлена функціональна модель системи виявлення та блокування ботнетів, яка наголошує на необхідності розробки системи, що в деякій мірі еквівалентна за складністю самим ботнетам. Така система відображає структуру і концептуальний алгоритм роботи запропонованої мультиагентної системи.

У параграфі 2.1 було проведено аналіз функціональної моделі. У результаті аналізу функціональної моделі стало очевидним, що оптимальним рішенням для

протидії ботнетам може стати створення системи, яка має схожу архітектуру з самими ботнетами. Така система захисту повинна також бути мультиагентною, оскільки це дозволить досягти високої ефективності у боротьбі як з атаками від продуктивних ботнетів, так і стати загальним інструментом для протидії зловмисним мережам.

У параграфі 2.2 здійснено опис взаємодії агентів між собою та із навколишнім середовищем, визначено ролі агентів. Слід зазначити, що правила взаємодії визначаються протоколом, який регулює даний аспект.

У третьому параграфі даного розділу було розроблено алгоритм виявлення керуючого трафіку ботнету, який базується на інтелектуальному аналізі даних. Цей алгоритм може виявляти трафік ботнету незалежно від протоколу чи організаційної структури ботнета. Алгоритм базується на використанні двох категорій агентів: агентів, що аналізують трафік, та агентів формування сигнатур, що виконують кросс-кластерну кореляцію між усіма кластерами. Під час проведення кластерного аналізу для оцінки ступеня схожості об'єктів вводиться поняття метрики. Визначення ступеня схожості та різниці між об'єктами визначається на основі обчислення перетину між двома кластерами.

У параграфі 2.4 був запропонований метод розподіленого виявлення керуючих компонентів ботнету, що використовується для контролю над атаками ботнету. Цей метод включає збір різноманітної інформації з передбачуваних адрес зловмисників, такої як інформація про операційну систему, тип пристрою, служби, порти, маршрути передачі даних, DNS-інформація, реєстраційні дані WHOIS, географічне розташування вузла та вразливості вузла. Ця інформація може бути дуже корисною під час розслідування інцидентів.

3 ДОВЕДЕННЯ ЕФЕКТИВНОСТІ РОЗРОБЛЕНИ АЛГОРИТМІВ

3.1 Метод автоматичного формування бази ботів

У контексті розробки системи виявлення ботнетів на основі мультиагентного підходу, важливим кроком є етап тестування, а також подальший аналіз продуктивності цієї розробки. У зв'язку з використанням мультиагентного підходу у побудові системи виявлення, процес тестування може бути поділений на наступні етапи:

– тестування працездатності кожного окремого агента;
– тестування розроблених алгоритмів;
– комплексне тестування системи виявлення в цілому.

Для вирішення завдань кожного з цих етапів, необхідно використовувати розгорнутий кіберполігон з працюючим ботнетом. Таким чином, виникає важлива задача автоматичного формування бази ботів, які діють у складі існуючих ботнетів.

Існують різні методи для отримання екземплярів шкідливого програмного забезпечення. До них входять проведення комп'ютерно-технічної експертизи, використання систем виявлення атак і honeypot системи. Один із найпростіших і поширених методів - це використання honeypot систем.

Honeypot-системи це об'єкти інформаційної системи, які призначені для спеціальної мети - привернення незаконної або несанкціонованої діяльності. Це свого роду пастки, розташовані в мережі з метою привернення уваги зловмисників. Часто honeypot-системи представляють собою віртуальні машини, які імітують роботу реальних систем і привертають увагу до себе, симулюючи роботу різних програм та сервісів.

Більшість ботів, що поширюються через веб-додатки, використовують автоматизовані методи. Програмовані боти автоматично сканують простір IP-адрес в пошуках активних серверів, які надають веб-сервіси. Після знаходження веб-сервера боти надсилають запити на основі попередньо створеної бази з надією знайти вразливий веб-сервіс, який можна атакувати. Якщо веб-сервіс виявляється

вразливим, то на сервер завантажується код бота, який потім може бути використаний для проведення атак.

Застосована концепція honeypot-системи полягає у направленні всіх вхідних запитів до веб-сервера на певний скрипт. Завдання цього скрипта - протоколювати всі запити, що надходять.

Після цього скрипт для аналізу запротокольованих запитів визначає спроби завантаження на сервер сторонніх скриптів через протоколи HTTP або FTP. Якщо такі спроби виявляються, скрипт аналізу самостійно завантажує копії ботів і зберігає їх в базі даних.

Ця процедура дозволяє створити базу даних з екземплярами ботів, які розповсюджуються через веб-додатки. Всю цю діяльність можна представити у вигляді діаграми компонентів, що ілюструє процес формування такої бази ботів. Діаграма цього процесу подана на рисунку 3.1. Також можна відобразити взаємодію компонентів під час виконання цього процесу, як показано на рисунку 3.2.

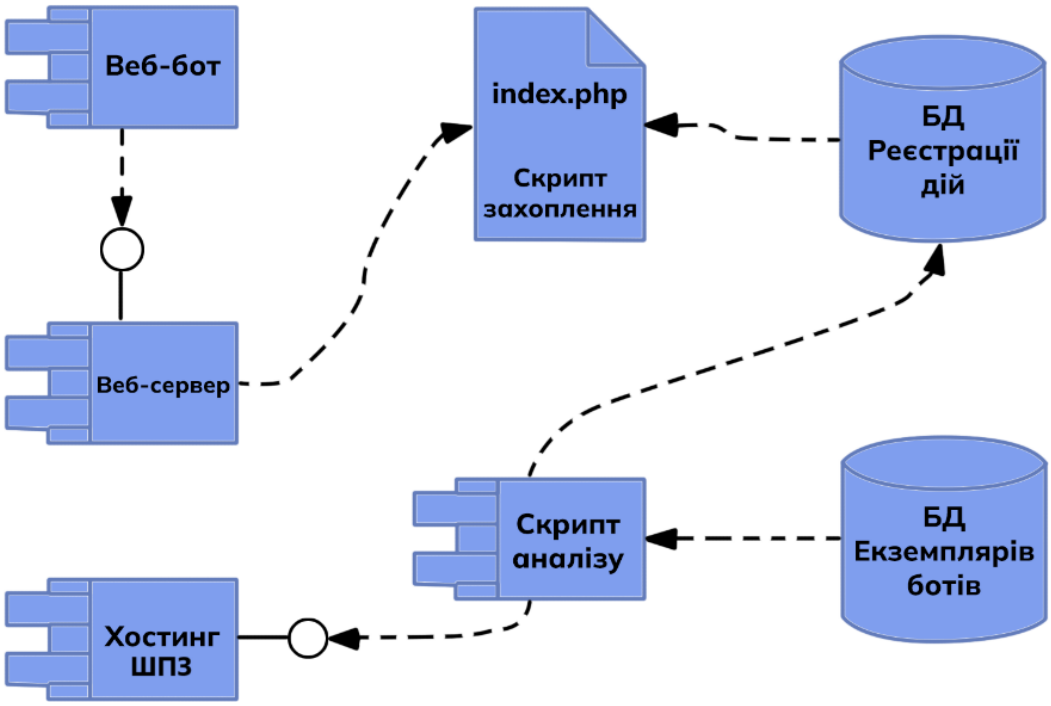

Рисунок 3.1 – Діаграма компонентів системи формування бази ботів, поширюваних через веб-застосунки

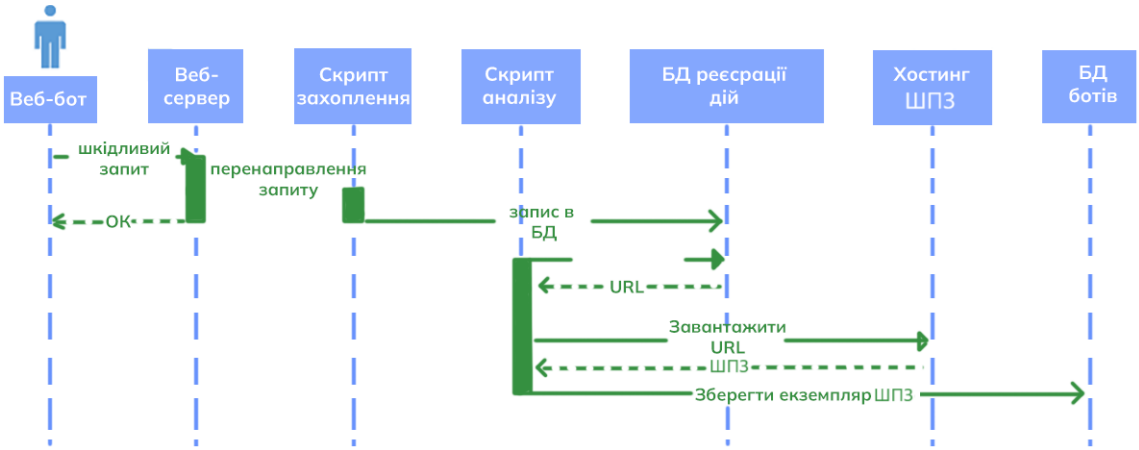

Рисунок 3.2 – Діаграма взаємодій системи формування бази ботів,поширюваних через веб-застосунки

Для проведення дослідження веб-сервер був налаштований на переспрямування всіх вхідних запитів HTTP на сценарій sniff.php (рисунок 3.3).

```php
<?php
    $loginfo['date']=date('c');
    $loginfo['env']=var_export($_ENV,true);
    $loginfo['get']=var_export($_GET,true);
    $loginfo['post']=var_export($_POST,true);
    file_put_contents('/tmp/log.txt',var_export($loginfo,true),FILE_APPEND);
?>
```

Рисунок 3.3 – Скрипт реєстрації вхідних запитів

Цей скрипт отримує HTTP-запити і записує їх до файлу "/tmp/log.txt". Потім скрипт аналізує ці записи, щоб виявити спроби завантаження зовнішніх скриптів. Приклад спроби завантаження шкідливого програмного забезпечення представлено на рисунку 3.4.

```
)array (
  'date' => '2015-03-16T09:13:54+04:00',
  'env' => 'array (
)',
  'get' => 'array (
)',
  'post' => 'array (
  \'action\' => \'lay_navigation\',
  \'eoltype\' => \'unix\',
  \'token\' => \'\',
  \'configuration\' => \'a:1:{i:0;O:10:"PMA_Config":1:{s:6:"source";s:29:"ftp://178.32.7.44/pub/124.php";}}\',
)',
```

Рисунок 3.4 – Приклад посилання на завантаження коду бота

3.2 Опис схеми проведення експериментального дослідження

Метою проведення експериментального дослідження була перевірка працездатності розробленого алгоритму виявлення керуючого трафік ботнету у реальній мережі. Експеримент проводився у кампусній мережі Хмельницького національного університету. Це дозволило створити умови, що максимально близькі до реальних. Зокрема, боти функціонували на реальних робочих станціях, що забезпечує наявність не одноманітного користувальницького трафіку. Використовувані в експерименті агенти розташовувалися на підконтрольних віртуальних серверах. Використовувалися два основних агенти, які беруть участь у визначенні керуючого трафіку бонета: агент дослідження трафіку скомпрометованих вузлів та агент формування сигнатури. Інформацію для обробки агенти дослідження трафіку одержували за протоколом Netflow. Для цього комутатори в мережі скомпрометованих вузлів реалізовували функцію сенсора Netflow і передавали дані колектору. Колектор Netflow працює як модуль агента дослідження трафіку. Для кожного скомпрометованого вузла дані колектором збирається окремо один від одного. Структура мережі експерименту представлена на рисунку 3.5.

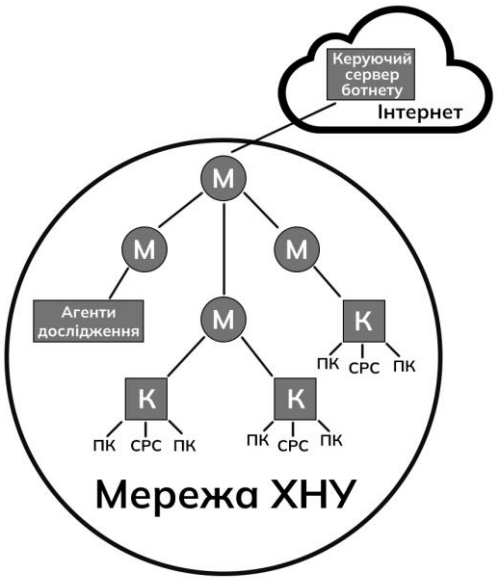

Рисунок 3.5 – Топологія експериментальної мережі

Для цього експерименту був використаний модифікований код одного з ботів, який був отриманий за допомогою системи, описаної у розділі 3.1. Модифікація коду включала в себе обмеження функціоналу бота та налаштування параметрів для взаємодії з керуючим сервером. Оскільки програмного забезпечення керуючого сервера не існувало, ми створили його з мінімальним функціоналом, який дозволяв нам відслідковувати статус ботів:

– показування списку активних ботів;

– відображення інформації про кожного бота, таку як час їх останнього входу в онлайн, мережеву адресу та операційну систему.

Методологія проведення експерименту включала кілька етапів, які були представлені на рисунку 3.6.

Рисунок 3.6 – Методологія проведення експерименту

Підготовка фізичної та віртуальної мережі інфраструктури включала в себе кілька етапів. Спочатку була налаштована технологія Netflow на всіх комутаторах, до яких були підключені скомпрометовані робочі станції. Далі проводилося завантаження та запуск програмного забезпечення бота на цих робочих станціях. Ці скомпрометовані робочі станції працювали під управлінням операційної системи Windows 10 та мали наступні мережеві адреси: 10.10.90.11/24, 10.10.90.17/24, 10.10.90.164/24.

На наступному етапі проводився збір даних сенсорами Netflow протягом тижня. Зібрані дані були передаватися агенту дослідження трафіку, який був розміщений на сервері в досліджуваному сегменті мережі ХНУ.

Агент дослідження даних відповідно до алгоритму поетапно виконував свої функції. У результаті його роботи для кожного скомпрометованого вузла формувалися набори кластерів, відповідно до трафіку, що проходив через ці вузли. Набори цих кластерів передавалися агенту формування сигнатур, однією з основних функцій якого було порівняння кластерів із наборів з метою виявлення найбільш схожих кластерів. Кластери, які співпадали, містили дані про керуючий трафік ботнету.

З метою оцінки ефективності розробленого алгоритму на завершальному етапі проводилась оцінка результатів його роботи. Оцінка полягала в тому, наскільки точно і повно кластери, знайдені алгоритмом, відображають керуючий трафік ботнету. Оскільки в рамках експериментального дослідження була доступна інформація про управляючий сервер ботнету (мережева адреса, порт, протокол взаємодії), то було можливо обчислити класичні метрики для оцінки алгоритмів вилучення інформації, такі як точність і повнота, і використовувати їх для розрахунку метрики - F-міра.

Метрика F-міра об'єднує інформацію про точність і повноту алгоритму, забезпечуючи баланс між цими двома метриками. Точність алгоритму в межах кластера визначається як відношення трафіку, який належить керуючому трафіку ботнету і фактично входить до цього кластера, до загального обсягу трафіку, призначеного для цього кластера. Повнота алгоритму визначається як частка

керуючого трафіку ботнету в кластері відносно всього керуючого трафіку ботнету, що міститься в зібраних даних. Ці значення обчислюються для кожного кластера на основі таблиці об’єднання, представленої в таблиці 3.1.

Таблиця 3.1 – Таблиця об’єднання

		Експертна оцінка	
		Позитивна	Негативна
Оцінка системи	Позитивна	TP	FP
	Негативна	FN	TN

В таблиці об’єднання міститься інформація про те скільки разів система прийняла правильне рішення і скільки разів хибне рішення, де:

– TP (true positives) – істинно-позитивне рішення, тобто результат, який віднесено до кластеру трафіка ботнету і дійсно такими являються;

– TN (true negatives) – істинно-хибне рішення, тобто результат, який віднесено до кластеру трафіка відмінному від трафіка ботнету і дійсно такими являються;

– FP (false positives) – хибно-позитивне рішення, тобто рішення, які віднесено до кластеру трафіка ботнету, але такими не являються;

– FN (false negatives) – хибно-негативні рішення, тобто рішення, які віднесено до трафіку що не являється ботнетом, проте таким являється.

Точність (Precision) кластеризації розраховується за наступною формулою:

$$Precision = \frac{TP}{TP + FP} \tag{3.1}$$

Повнота (Recall) кластеризації розраховується за наступною формулою:

$$Recall = \frac{TP}{TP + FN} \tag{3.2}$$

Метрика F-міра представляє собою гармонічне середнє між точністю і повнотою. Вона близька до нуля, якщо точність або повнота наближається до нуля. Розрахунок даної метрики здійснюється наступним чином:

$$F = 2 * \frac{Recall * Precision}{Recall + Precision} \quad (3.3)$$

Метрика F-міра зводить до одного числа дві інші основні метрики: точність і повноту. Що, в свою чергу, прискорює процес прийняття рішення при оцінка зміни алгоритму в кращу чи гіршу сторону.

3.3 Опис результатів експерименту

В таблиці 3.2 наведено статистичні дані аналізованого трафіка за 5 днів для спостережуваногоскомпрометованого вузла. Впродовж першого дня було передано 984534 пакети (TCP, UDP) і 12456 потоків відповідно. Показано, що фільтрація ефективна з точки зору зменшення обсягу даних. Агрегування також змінює кінцеві дані, в результаті отримуємо 3457 агрегованих потоків за добу. Кількість аналізованих потоків прямо залежить від активності користувачів скомпрометованих вузлів.

Таблиця 3.2 – Статистичні дані аналізованого трафіка вузла 10.10.90.11/24

День	До фільтрації		Після фільтрації	
	потоки	пакети	потоки	пакети
1	12 456	984 534	3 457	894 651
2	10 785	832 215	2 042	689 725
3	11 377	912 453	2 755	758 963
4	13 721	1 124 348	3 724	1 012 984
5	12 024	932 756	3 181	820 142

Після цього була проведена кластеризація аналізованого трафіку, і однією з вирішуваних задач було вибрати найбільш підходящий алгоритм для цієї кластеризації. Для цього було здійснено порівняння трьох алгоритмів кластеризації: DBSCAN (алгоритм кластеризації щільним методом), XMeans (алгоритм, який автоматично оцінює кількість кластерів) і EM (алгоритм максимальної правдоподібності Expectation-Maximization). Ці алгоритми самостійно визначають кількість кластерів. Оцінено їх ефективність з точки зору їх здатності створювати кластери так, щоб керуючий трафік ботнета переважно розміщувався в одному або кількох кластерах. Для оцінки якості роботи алгоритму виявлення керуючого трафіку використано метрику Ф-міра, яка об'єднує в собі метрики точності і повноти. Для вимірювання подібності між векторами ознак x та y у просторі розмірності n ми використано Евклідову відстань:

$$dist(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{3.4}$$

У результаті експерименту було створено набір кластерів для кожного скомпрометованого вузла. Важливо відзначити, що експеримент був проведений з використанням різних тимчасових інтервалів, які використовувалися під час агрегації потоків трафіку. Результати цього експерименту наведені у таблицях 3.3-3.5.

Таблиця 3.3 – Результати метрик якості після експерименту

	Інтервал	TP	TN	FP	FN
	1	2	3	4	5
DBSCAN	1	312237	101476	162473	318465
	2	155048	129821	89456	315400
	3	192467	115497	123487	327512
	4	225682	221231	167849	398222
	5	297427	98786	152882	271047

Кінець таблиці 3.3 – Результати метрик якості після експерименту

	1	2	3	4	5
EM	1	334556	99557	174242	286296
	2	205127	149478	74565	260555
	3	196784	186073	164242	211864
	4	294622	218997	197054	302311
	5	299523	97544	180531	242544
XMEANS	1	415425	137435	99679	242112
	2	244465	165789	63212	216259
	3	234563	211662	147658	165080
	4	371687	237546	179541	224210
	5	350872	101643	165339	202288

Таблиця 3.4 – Показники параметрів точності та повноти ефективності виділення керуючого трафіку ботнету

Алгоритм	Інтервал	P	R	F
DBSCAN	1	0,66	0,5	0,58
	2	0,63	0,33	0,48
	3	0,61	0,37	0,48
	4	0,57	0,36	0,47
	5	0,66	0,52	0,59
EM	1	0,66	0,54	0,6
	2	0,73	0,44	0,59
	3	0,55	0,48	0,51
	4	0,6	0,49	0,55
	5	0,62	0,55	0,59
XMEANS	1	0,81	0,63	0,72
	2	0,79	0,53	0,66
	3	0,61	0,59	0,6
	4	0,67	0,62	0,65
	5	0,68	0,63	0,66

Таблиця 3.5 – Показники параметрів акуратності та помилки ефективності виділення керуючого трафіку ботнету

Алгоритм	Інтервал	A (акуратність)	S (помилка)
DBSCAN	1	0,46	0,54
	2	0,41	0,59
	3	0,41	0,59
	4	0,44	0,56
	5	0,48	0,52
EM	1	0,49	0,51
	2	0,51	0,49
	3	0,5	0,5
	4	0,51	0,49
	5	0,48	0,52
XMEANS	1	0,62	0,38
	2	0,59	0,41
	3	0,59	0,41
	4	0,6	0,4
	5	0,55	0,45

Результати дослідження вказують на те, що найбільш ефективним є використання алгоритму XMeans. Відмінною особливістю алгоритму DBSCAN є те, що він визначає частину бот-трафіку, яка проявляє високу повноту, але не виділяє цей трафік в окремий кластер, а замість цього об'єднує його з користувацьким трафіком. У порівнянні з алгоритмом DBSCAN, алгоритм XMeans пропонує швидший процес кластеризації, що відображено у середньому часі, витраченому на кластеризацію.

Для покращення показників виявлення керуючого трафіку ботнету був проведений другий етап кластеризації. На цьому етапі використовувалися кластери, які були сформовані на першому етапі кластеризації. Результати цього

етапу наведено в таблицях 3.6-3.7.

Таблиця 3.6 – Показники ефективності для вузла 10.10.90.11/24

Алгоритм	Інтервал	P	R	F
XMEANS	1	0,77	0,7	0,73
	2	0,88	0,6	0,74
	3	0,61	0,67	0,64
	4	0,65	0,69	0,67
	5	0,73	0,7	0,71

Таблиця 3.7 – Показники акуратності та помилки

Алгоритм	Інтервал	A (акуратність)	S (помилка)
XMEANS	1	0,65	0,35
	2	0,67	0,33
	3	0,63	0,37
	4	0,63	0,37
	5	0,62	0,38

3.4 Висновки до розділу

Був розроблений метод для автоматичного формування бази ботів. Цей метод включає створення низько інтерактивної honeypot-системи, спрямованої на веб-сервіси, та збільшення обсягу бази шкідливих програмних засобів, необхідної для проведення експериментів. У застосованій концепції honeypot-системи, всі вхідні запити до веб-сервера спрямовуються на виконання певного скрипта. Головною функцією цього скрипта є реєстрація та збереження всіх вхідних запитів, які надходять до системи. Всього було отримано 18 прикладів ботів.

Описано методологію проведення експерименту та розгорнуто експериментальну мережу на базі Хмельницького національного університету. Для

обчислювальних експериментів був змінений код бота, який було отримано за допомогою розробленої honeypot-системи. Було внесені зміни в наступних аспектах бота:

– використання протоколу HTTP як керуючого каналу.
– обмеження функціоналу шкідливого програмного засобу.
– функціональність для взаємодії з керуючим сервером.

Проведені обчислювальні експерименти з метою оцінки ефективності запропонованого алгоритму виявлення керуючого трафіку ботнету. У рамках експериментального дослідження, був доступ до інформації про сервер управління ботнету, включаючи мережеву адресу, порт та протокол взаємодії. Це надало можливість застосувати стандартні метрики для оцінки алгоритмів вилучення інформації, такі як точність і повнота, та використовувати їх для розрахунку комплексної метрики - F-міри.

В результаті експериментів для запропонованого алгоритму виявлення керуючого трафіку ботнету був вибраний алгоритм кластеризації - XMEANS. Результати цього алгоритму виявилися найкращими для всіх скомпрометованих вузлів. У порівнянні з іншими розглянутими алгоритмами, в середньому результат був кращим на 15%. Крім того, цей алгоритм має найшвидший час виконання.

Показники ефективності розробленого алгоритму були наступними:

– середнє значення метрики Ф-міра становило 0,7;
– середнє значення помилкових спрацювань складало 0,36;
– середнє значення точності становило 0,73.

4 РОЗРОБКА ДОСЛІДНОГО ПРОТОТИПУ МУЛЬТИАГЕНТНОЇ СИСТЕМИ ВИЯВЛЕННЯ БОТНЕТІВ

4.1 Формування вимог до мультиагентної системи виявлення ботнетів

У проведеному дослідженні була розглянута мультиагентна система виявлення ботнетів, яка представляє собою комплексне рішення, включаючи різні види програмного забезпечення. Використовуване програмне забезпечення може бути розділене на наступні категорії:

– система запобігання вторгненням;
– міжмережевий екран;
– система моніторингу мережевої безпеки.

Зазначена система є системою виявлення, тому до неї можуть бути висунуті вимоги, які повинна задовольняти система виявлення атак. Структура вимог до систем виявлення вторгнень наведена на рисунку 4.1. З метою розрізнення вимог до функцій безпеки систем виявлення вторгнень було визначено шість класів захисту даних системи.

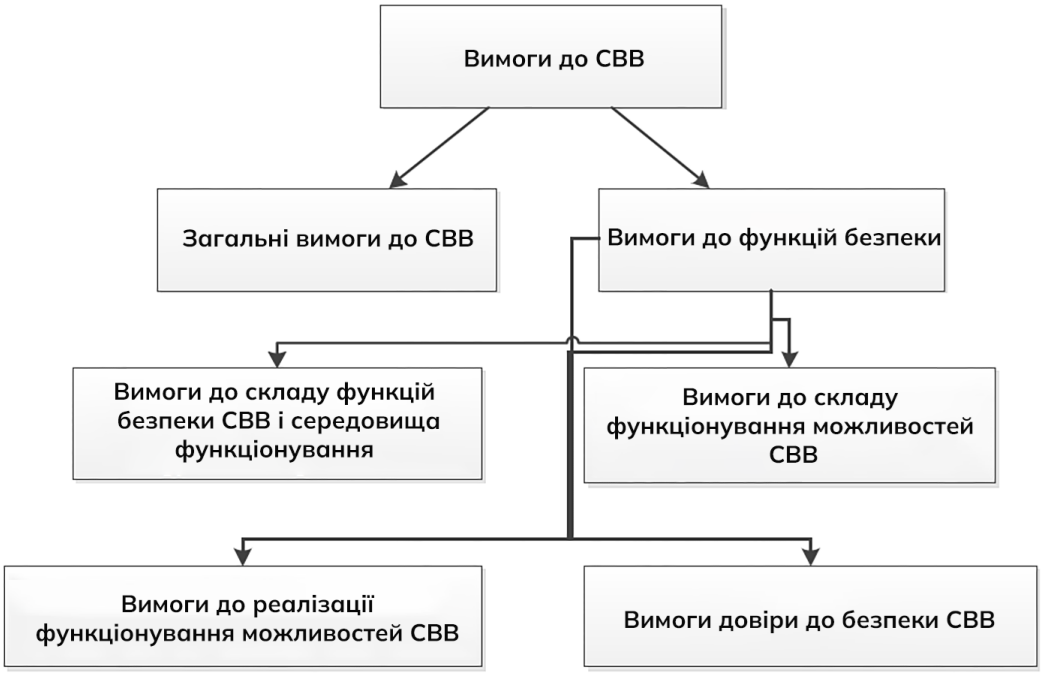

Рисунок 4.1 – Структура вимог до СВВ

Рівень класифікації систем виявлення вторгнень може бути визначений наступним чином, де перший клас відповідає найвищому рівню, а шостий клас - найнижчому:

– Системи виявлення вторгнень, які відповідають 6-му класу захисту, застосовуються у інформаційних системах, які містять персональні дані, що належать до 3 та 4 класів.

– Системи виявлення вторгнень, відповідно до 5-го класу захисту, використовуються у інформаційних системах, які обробляють персональні дані 2 класу.

– Системи виявлення вторгнень, що відповідають 4-му класу захисту, використовуються у інформаційних системах, де дані є обмеженим доступом і не містять конфіденційної інформації, яка стосується державної таємниці. Також ці системи можуть застосовуватися у інформаційних системах, що містять персональні дані 1 класу, а також у загальнодоступних інформаційних системах 2 класу.

– Системи виявлення вторгнень, відповідно до 3, 2 і 1 класів захисту, використовуються у інформаційних системах, де обробляється інформація, яка містить відомості, що стосуються державної таємниці.

Тип системи виявлення вторгнень та клас захисту визначає сукупність вимог до функцій безпеки СВВ. Ці вимоги можна поділити на 4 набори:

– вимоги до складу функцій безпеки СВВ і середовища, в яких ці СВВ функціонують;

– вимоги до складу функціональних можливостей СВВ, що забезпечують реалізацію функцій СВВ;

– вимоги до реалізації функціональних можливостей СВВ;

– вимоги довіри до безпеки СВВ.

Для СВВ рівня мережі 4 класу виділяють десять функцій безпеки, які повинні бути реалізованими:

– розмежування доступу до управління СВВ;

– управління роботою СВВ;
– управління параметрами СВВ;
– управління встановленням оновлень бази правил СВВ;
– аналіз даних СВВ;
– аудит безпеки СВВ;
– контроль цілісності СВВ;
– збір даних про події та активності в контрольованій ІС;
– реагування СВВ;
– маскування СВВ.

В якості функцій безпеки середовища функціонування виділяють:

– забезпечення довіреного маршруту з адміністраторами СВВ (автентифікація і захищений канал управління);
– забезпечення довірчого каналу оновлень бази правил;
– забезпечення умов безпечного функціонування;
– управління атрибутами безпеки.

Функціональні вимоги безпеки СВВ рівня мережі 4 класу включаючи:

– вимоги по здійсненню збору даних СВВ;
– вимоги до аналізу даних СВВ;
– вимоги до реагування СВВ;
– вимоги до засобами оновлення бази вирішальних правил СВВ;
– вимоги до захисту СВВ;
– вимоги до управління режимами виконання функцій безпеки;
– вимоги до управління даними функцій безпеки;
– вимоги до управління ролями суб'єктів;
– вимоги до засобів адміністрування СВВ;
– вимоги до аудиту і функціонування СВВ.

Перелік функціональних вимог безпеки забезпечує наступні функціональні можливості СВВ рівня мережі 4 класу:

– можливість збору інформації про мережу трафіка;

– можливість виконання аналізу зібраних даних СВВ про мережу трафіку в режимі, близькому до реального масштабу, та за результатами аналізу фіксувати інформацію про дату і годину, результат аналізу, ідентифікатор джерела даних, протокол, який використовується для проведення вторгнення;

– можливість виконання аналізу зібраних даних з метою виявлення вторгнень за допомогою сигнатурного та евристичних методів;

– можливість виконання аналізу зібраних даних з метою виявлення вторгнень з евристичних методів, заснованих на методах виявлення аномалій мережевого трафіку на заданому рівні евристичного аналізу;

– можливість виявлення вторгнень на основі аналізу службової інформації протоколів мережевого рівня базової еталонної моделі взаємозв'язку відкритих систем;

– можливість фіксації факту виявлення вторгнень чи порушень безпеки у журналах аудиту;

– контроль адміністратора СВВ про виявлені вторгнення щодо ставлення до контрольованого вузла ІС та порушень безпеки за допомогою відображення відповідного повідомлення на консолі управління;

– можливість автоматизованого оновлення бази вирішальних правил;

– можливість тестування (самотестування) функцій безпеки СВВ;

– можливість зі сторони уповноважених адміністраторів (ролей) керувати режимом виконання функцій безпеки СВВ;

– можливість зі сторони уповноважених адміністраторів (ролей) керувати даними СВВ;

– підтримка визначених ролей для СВВ та їх асоціації з конкретними адміністраторами СВВ та користувачами ІС;

– можливість адміністрування СВВ;

– можливість генерації записів аудиту для подій, потенційно підданих аудиту;

– можливість асоціації кожної події аудиту з ідентифікатором суб'єкта, що

його ініціював;

- можливість надавати можливість читати інформацію із записів аудиту;
- обмеження доступу до читання записів аудиту;
- пошук, сортування, упорядкування даних аудиту.

Вимоги до безпеки СВВ охоплюють аспекти управління конфігураціями, постачанням та експлуатацією системи, розробкою, управлінням, підтримкою життєвого циклу, тестуванням, оцінкою чутливості та оновленням бази вирішальних правил.

Основними компонентами системи виявлення вторгнень є сенсори та аналізатори. Сенсори збирають інформацію про пакети даних, які обмінюються в межах інформаційної системи (ІС), де вони розташовані. Сенсори рівня мережі СВВ можуть бути реалізовані як програмне забезпечення, встановлене на стандартних програмно-технічних платформах, або як програмно-технічні пристрої, підключені до ІС. Аналізатори виконують обробку зібраної даними датчиками інформації, генерують звіти на основі результатів аналізу та управляють процесами реагування на виявлені вторгнення. Рішення про виявлення вторгнень у СВВ приймаються на підставі результатів аналізу інформації, зібраної датчиками СВВ, і використанням бази вирішальних правил СВВ.

4.2 Проектування структури системи

На рисунку 2.3 наведена структура дослідницького прототипу системи виявлення ботнетів на основі мультиагентної архітектури.

Кожен агент обладнаний модулем кооперації, що використовується для спільної роботи між агентами. Цей модуль дозволяє агентам обмінюватися даними та надсилати команди один одному, коли це необхідно.

Агент виявлення атаки включає в себе модуль для виявлення атак, який сприяє виявленню потенційних атак, формуванню списку атакуючих вузлів і передачі цієї інформації іншим агентам.

Агент блокування атаки, отримавши список вузлів, які брали участь у

проведенні атаки, формує правила блокування трафіку для зупинки атак та накладає їх, використовуючи модуль блокування атак.

Агент дослідження трафіку атакуючої машини агрегує весь трафік, що надходить протягом певного часового періоду, і після цього кластеризує його за допомогою модуля кластеризації. Потім він передає ці кластери агенту формування сигнатур.

Агент формування сигнатур використовує модуль кросс-кластерної кореляції для аналізу кластерів, отриманих в результаті аналізу трафіку всіх вузлів, які були помічені під час однієї атаки. Він створює сигнатури для розпізнавання ботів за допомогою модуля формування сигнатур.

Агент виявлення ботів має свій власний модуль для цього процесу.

Агент моніторингу складається з модуля обробки даних, модуля надання інтерфейсу управління та моніторингу, а також бази даних.

Агент координування відповідає за забезпечення кооперації між агентами та містить модуль обміну повідомленнями.

Розробка прототипу системи з нуля не є необхідною, оскільки багато завдань агентів можуть бути вирішені за допомогою вже наявних засобів з відкритим вихідним кодом. Відповідність модулів агентів та класів системи наведено в Таблиці 4.1.

Таблиця 4.1 – Відповідності модулів агентів та класів систем

Модуль агентів	Клас системи
Модуль виявлення атаки	Система виявлення атак
Модуль блокування атаки	Межмережевий екран
Модуль виявлення ботів	Система виявлення ботів
Модуль обміну повідомленнями	Сполучне програмне забезпечення
Модуль обробки даних	Диспетчер бінарних файлів події СОВ
Модуль наданого інтерфейсу управління і моніторингу	Система моніторингу безпеки мережі

Отже, для більшості функцій агентів можна використовувати наявні системи з відкритим вихідним кодом. Для реалізації вимог до модуляції було створено модуль кооперації, модуль відображення зловмисника, модуль кластеризації трафіку, модуль кросс-кластерної кореляції та модуль формування сигнатур ботнета.

Як систему виявлення атак та систему виявлення ботів було вибрано Suricata, що є системою виявлення/запобігання атакам на основі правил, яка використовується для моніторингу мережевого трафіку та сповіщення системного адміністратора при виникненні підозрілих подій. Ця система поширюється під ліцензією GPLv2.

Базовою операційною системою при розробці дослідницького прототипу стала Ubuntu 22.04 – дистрибутив Linux, ліцензія GPL. D якості міжмережевого екрану використовувався Netfilter – міжмережевий екран, вбудований в ядро Linux, управляється утилітою iptables.

Платформа для обміну повідомленнями між компонентами програмної системи була реалізована на основі стандарту AMQP (Advanced Message Queuing Protocol) та використовувала RabbitMQ, який поширюється під ліцензією Mozilla Public License.

Для отримання файлів подій від агентів виявлення атак та ботів використовувався диспетчер бінарних файлів подій під назвою Barnyard2, який розповсюджується під ліцензією GPLv2. Завдання моніторингу та візуалізації було вирішено за допомогою веб-системи моніторингу мережевої безпеки Snorby, яка розповсюджується під ліцензією GPLv3. Структуру системи розгортання можна побачити на рисунку 4.2.

Під час розробки дослідницького прототипу було використано різні мови програмування. Для створення модулів використовувалися агенти, написані на мовах програмування Perl та Python. Ці мови підтримують найбільш визнані та використовувані парадигми програмування. Для реалізації алгоритмів інтелектуального аналізу даних використовувалася мова програмування R, яка призначена для статистичної обробки даних та роботи з графікою, і є вільним

програмним середовищем обчислень з відкритим вихідним кодом в рамках проекту GNU. Крім цього, для вирішення різних системних завдань були розроблені скрипти, оброблені командним процесором Unix - Bash.

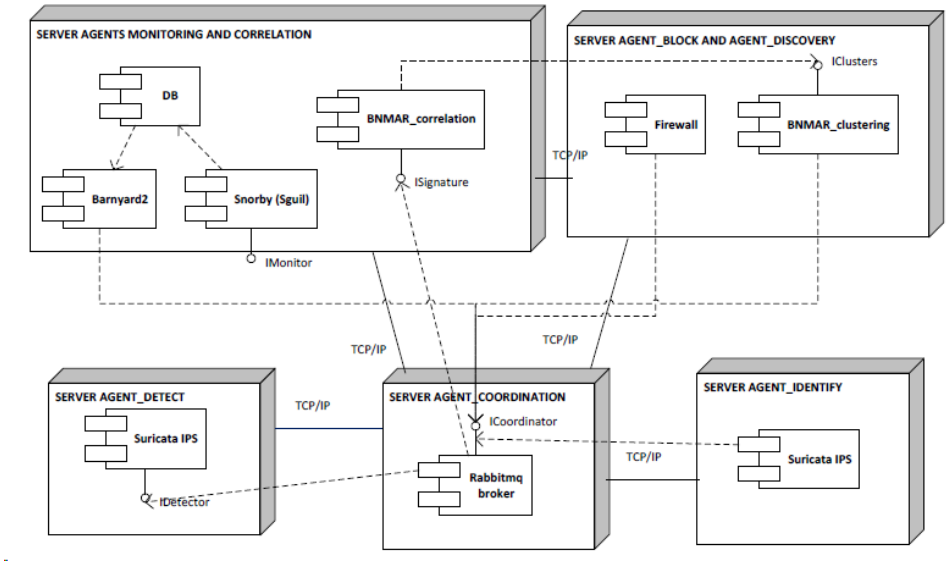

Рисунок 4.2 – Діаграма розгортання системи виявлення ботнетів

Для агрегування даних, які генеруються різними модулями системи, було створено базу даних. Структура цієї бази даних відображена в інформаційній моделі даних, яка була побудована за методикою IDEF1X. Методика IDEF1X використовує умовний синтаксис для створення універсального представлення структури даних у реляційних базах даних. Отримана модель даних не залежить від конкретної реалізації бази даних, програмного чи апаратного забезпечення. Структуру цієї бази даних можна побачити на рисунку 4.3.

Розроблена база даних дозволяє збирати важливу інформацію про інциденти в мережі Інтернет, включаючи дані про ботнети та їх керуючі компоненти, такі як сервери управління та вузли мережі, які використовуються для координації атак. Зібрана інформація має велике значення для забезпечення централізованого моніторингу кіберзагроз та проведення розслідувань в галузі кіберзлочинності.

Ця інформація зберігається для трьох основних об'єктів у системі виявлення ботнетів: агентів, подій безпеки та ботнетів. Агенти системи виявлення пов'язані з подіями безпеки, оскільки вони виявляють активні атаки та також мають

взаємозв'язок з ботнетами, формуючи сигнатури ботнетів. Ботнети через свою шкідливу активність спричиняють події безпеки.

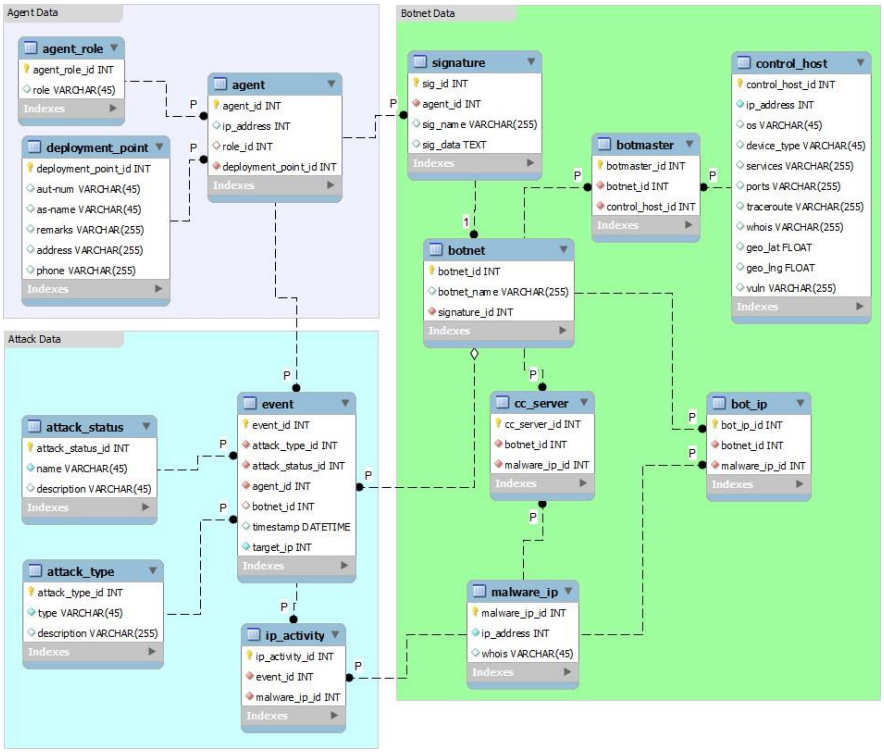

Рисунок 4.3 – Інформаційна модель даних системи

Для моделювання фізичного розташування агента, що вивчає трафік, використовувалася діаграма розташування, яку можна побачити на рисунку 4.4.

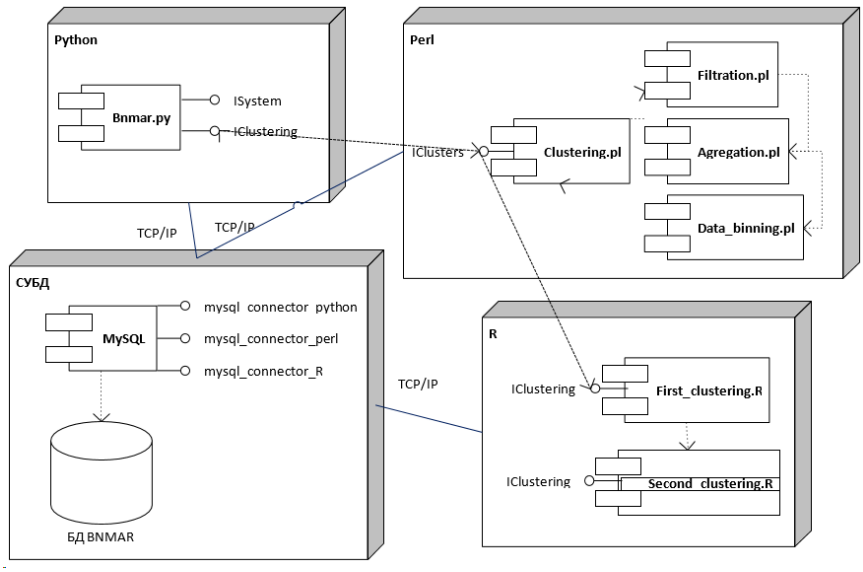

Рисунок 4.4 – Діаграма розгортання агента дослідження трафіка

Були створені програмні модулі для забезпечення функціональності агентів, які вивчають трафік та формують сигнатури.

Опис розроблених програмних засобів:

– основний модуль. Відповідає за ініціалізацію алгоритму виявлення керуючого трафіку, узгодження та координація функціональних модулів. Виконує функції крос-кластерної кореляції та подальше формування підписи керуючого трафіку (Bnmar.py);

– модуль фільтрації трафіка. Реалізує описану процедуру фільтрації даних про трафік. Вхідними даними є інформація про потоки трафіка, такі як час з'єднання, мережеві адреси, порти джерела і призначення, протокол взаємодії (Filtration.pl);

– модуль агрегації даних. Вхідні дані є результатом роботи утиліти фільтрації і часового інтервалу, в рамках якого буде здійснюватися агрегування потоків. Результатом роботи є файл з агрегованими схожими потоками (Aggregation.pl);

– модуль векторизації агрегованих даних. Вхідними даними є файл агрегованих потоків і часового інтервалу, з яким проводилась агрегація потоків. Якщо тимчасовий інтервал не заданий, утиліта буде очікувати інтервал ручного введення для векторизації критеріїв. Результатом роботи є файл з векторизованими критеріями (Data_binning.pl);

– модуль контролю виконання алгоритму (Clustering.pl). Основними функціями є: підготовка отриманих даних до кластеризації, запуск процедури двоетапної кластеризації. Перший етап кластеризації здійснюється за допомогою скрипта First_clustering.R. Другий етап виконується за допомогою скрипта Second_clustering.R. Результатом роботи є інформація про розподіл потоків трафіка за кластерам;

– модулі, що виконують кластеризацію трафіка (First_clustering.R і Second_clusternig.R).

Налаштування системи реалізується шляхом налаштування конфігураційних

файлів агентів. Моніторинг подій безпеки та ботнетів здійснюється через веб-інтерфейс і файли журналів.

4.3 Тестування роботи системи

Для перевірки ефективності розробленого прототипу було здійснено тестування в реальних умовах. Тестування проводилося на базі сегментів мережі та кіберполігону Хмельницького національного університету. У декількох сегментах мережі постачальника послуг Інтернету були встановлені агенти мультиагентної системи для виявлення та блокування ботнетів. Розгортання агентів проводилося відповідно до діаграми розгортання системи виявлення ботнетів, яку було описано раніше. Структура мережі під час проведення тестування мультиагентної системи була проілюстрована на рисунку 4.5.

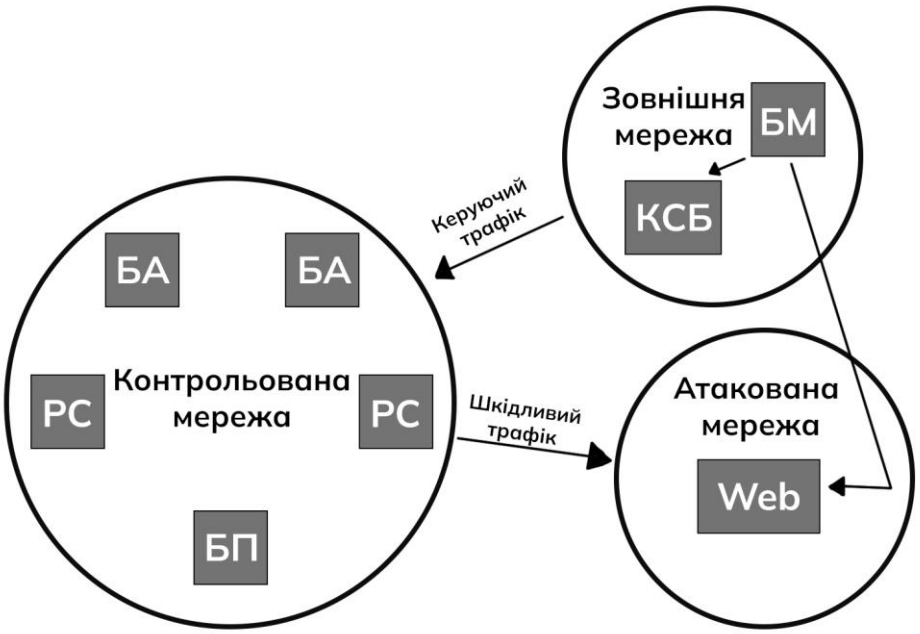

Рисунок 4.5 – Структура мережі при проведенні тестування мультиагентної системи виявлення та блокування ботнетів

Використаний ботнет складався з комп'ютерів, які були скомпрометовані і знаходилися в контрольованих мережевих сегментах. Було створено два таких

контрольованих мережевих сегменти. У одному з цих сегментів розміщені засоби захисту, які були піддані тестуванню, тоді як у другому сегменті засоби захисту не були встановлені. Керуючий сервер ботнету був розгорнутий на зовнішньому хостингу в Інтернеті. Атакований комп'ютер, який виконував функцію веб-сервера, знаходився в одному з мережевих сегментів провайдера. У цьому ж сегменті була розгорнута точка мультиагентної системи, включаючи наступні агенти: агент виявлення атак, агент блокування атак, агент дослідження трафіку і агент координації. Ця точка мультиагентної системи була розгорнута в повністю контрольованому мережевому сегменті, який включав в себе весь набір агентів.

У таблиці 4.2 наведено кількість активних і пасивних ботів разом із вказівкою місць їх розгортання. Активними вважаються боти, які брали участь у атаках, тоді як пасивні боти – ті, які не приймали участь у цих атаках.

Таблиця 4.2 – Об’єм ботнету

	Кількість ботів в сегменте с розвернутими засобами захисту	Кількість ботів в сегменте без засобів захисту	Всього ботів
Всього	18	2	20
Активних	15	1	16
Пасивних	3	1	4

Створений ботнет був керований через спеціальну панель управління, що надавала можливість слідкувати за поточним статусом ботів і віддавати команди конкретним скомпрометованим комп'ютерам.

Згідно з тестовим планом дослідницького прототипу, група ботів атакувала цільовий комп'ютер, використовуючи атаку "HTTP POST flood". Однак для перевірки можливості виявлення пасивних ботів під час атаки не всі боти були задіяні. Для контролю за виконанням атаки з комп'ютера ботмайстра була запущена утиліта ping з мережевою адресою цільового вузла в якості параметра.

Для порівняння ефективності розробленого прототипу були використані інші

відомі захисні системи, які були схожі за своїми функціями. Серед них були мережева система запобігання вторгнення Snort і мережева система виявлення активності ботів BotHunter. Snort використовує сигнатурний підхід, і без конкретно заданої сигнатури керуючого трафіку система не може виявити пасивного бота. З іншого боку, BotHunter використовує підхід до аналізу аномалій трафіку, і він спроможний виявляти ботів, аналізуючи ознаки різних етапів життєвого циклу бота. BotHunter має більшу ймовірність виявлення бота, який потрапив до мережі після встановлення цієї системи.

Після проведення тестування системи отримано такі результати: система запобігання вторгненням Snort правильно виявила 15 скомпрометованих вузлів. Система BotHunter правильно виявила 11 скомпрометованих вузлів і один виявила неправильно. Розроблений прототип виявив 18 скомпрометованих вузлів. Подані результати тестування в таблиці 4.3. Важливо відзначити, що перші дві системи виявили лише активних ботів, тоді як у випадку прототипу з 18 виявлених ботів 3 знаходилися в пасивному стані.

Таблиця 4.3 – Результати тестування мультиагентної системи

	Повнота	Точність	Ф-міра
NET.BOTNET	0,90	1,00	0,95
Snort	0,75	1,00	0,86
BotHunter	0,55	0,92	0,69

Точність показників в усіх системах висока. У перших двох системах це досягнуто завдяки використанню сигнатур. Для загальної оцінки ефективності системи важливим є показник повноти виявлення, який змінюється від 15% до 35%. Це пояснюється кількома факторами. По-перше, на відміну від інших систем, розроблений прототип виявив частину пасивних ботів, які були в зоні дії мультиагентної системи. По-друге, систему BotHunter встановлювали в сегменті, де вже існували боти, і це вплинуло на її результати. Загалом показник Ф-міри розробленого прототипу на 9% і 26% вищий в порівнянні з Snort та BotHunter,

відповідно. Розроблений прототип здатний визначити мережеву адресу вузла, що контролював атаку. Функціональне порівняння систем наведено в таблиці 4.4.

Таблиця 4.4 – Результати тестування функціональності мультиагентної системи

	Виявлення атаки	Виявлення ботів	Формування сигнатури	Виявлення ботмайстра
NET.BOTNET	так	так	так	так
Snort	так	так	ні	ні
BotHunter	ні	так	ні	ні

Мультиагентна система, яку було розроблено, може досягти максимальної ефективності в разі всезагального впровадження. Іншими словами, чим більше точок розгортання буде наявно, тим кращі результати можна очікувати від її функціонування. З цією метою була розроблена концепція практичного впровадження мультиагентної системи, яка представлена на рисунку 4.6.

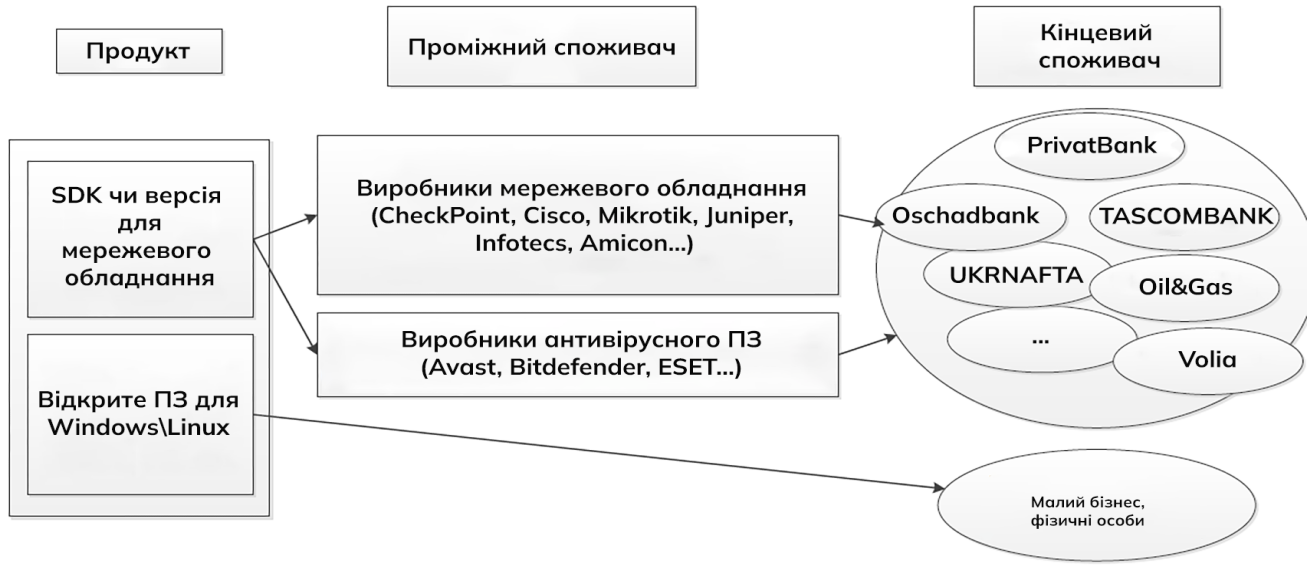

Рисунок 4.6 – Концепція практичного впровадження мультиагентної системи

Систему можна впроваджувати по двом основним напрямкам: як SDK-

платформу та у вигляді відкритого програмного забезпечення. Розповсюдження у формі відкритого програмного забезпечення дозволить впроваджувати систему в секторі малого та середнього бізнесу, де власники виявляють підвищений інтерес до питань інформаційної безпеки. У цьому випадку, систему можна поширювати у вигляді інсталяційного файлу програмного забезпечення для агентів.

Це сприятиме розширенню системи і відповідно поліпшить її загальну ефективність. У разі впровадження у вигляді SDK-платформи, систему можуть інтегрувати антивірусні компанії та розробники мережевого обладнання у свої продукти. Це дозволить використовувати систему не лише фізичним особам та малим бізнесом, але і великим телекомунікаційним, фінансовим і промисловим компаніям.

4.4 Висновки до розділу

У параграфі 4.1 були сформульовані вимоги до мультиагентної системи виявлення і блокування ботнетів. Висунуто вимоги до СВВ, які охоплюють аспекти управління конфігураціями, постачанням та експлуатацією системи, розробкою, управлінням, підтримкою життєвого циклу, тестуванням, оцінкою чутливості та оновленням бази вирішальних правил. Визначено, що до основних компонентів системи виявлення вторгнень слід відносити сенсори та аналізатори.

В параграфі 4.2 представлено опис призначення агентів відповідно до структури, яку зображено на рисунку 2.3. Також була розроблена структура прототипу системи та програмне забезпечення для трьох видів агентів: агента дослідження трафіку, агента формування сигнатури та агента координатора. Для виконання функцій інших типів агентів було використано вільно поширюване програмне забезпечення. Крім того, була розроблена інфраструктура для тестування дослідницького прототипу системи. Ця інфраструктура включала в себе використання мережевих сегментів і кіберполігону Хмельницького національного університету для проведення тестових випробувань та оцінки функціональності та ефективності розробленого прототипу системи.

Цей етап розробки та тестування прототипу демонструє роботу, яку було проведено в напрямку вдосконалення системи виявлення та блокування ботнетів, як інструмента боротьби з кіберзагрозами.

У параграфі 4.3 проведено тестування дослідницького прототипу, побудованого на основі запропонованого алгоритму виявлення керуючого трафіку ботнетів. Розгортання агентів проводилося відповідно до діаграми розгортання системи виявлення ботнетів. Прототип виявив 90% ботів у використовуваному ботнеті, включаючи тих, що не брали участі в атаках. Також була виявлена мережева адреса, яка використовувалася для керування розподільною атакою типу "відмова в обслуговуванні".

Проведено порівняння ефективності розробленого прототипу системи виявлення та блокування ботнетів з безкоштовними захисними системами, які мають схожі функціональні можливості: мережевою системою запобігання вторгнення Snort та мережевою системою виявлення активності ботів BotHunter. Показник Ф-міри розробленого прототипу перевищує аналогічні показники Snort та BotHunter на 9 і 26% відповідно. Порівняння демонструє переваги використання розробленого прототипу у функціях формування сигнатури та виявлення компонентів ботнету.

Подано концепцію практичного впровадження мультиагентної системи виявлення та блокування ботнетів. Ця концепція описує можливість всебічного впровадження мультиагентної системи для підвищення ефективності її використання.

ВИСНОВКИ

В даній роботі була вирішена задача підвищення безпеки інформаційних систем від атак, що виконуються ботнетами, завдяки розробці та впровадженню мультиагентної системи для виявлення та блокування ботнетів за допомогою алгоритмів інтелектуального аналізу даних.

Був проведений аналіз стану проблеми виявлення ботнетів в відкритих комп'ютерних мережах, включаючи Інтернет, існуючих методів захисту від розподільних атак, таких як "відмова в обслуговуванні", і методів виявлення ботнетів. Проведений аналіз виявив відсутність повноцінного захисту від зловмисної активності ботнетів.

Був запропонований алгоритм виявлення керуючого трафіку ботнету на основі інтелектуального аналізу даних, з можливістю автоматичного створення сигнатур для керуючого трафіку. Цей алгоритм відрізняється від існуючих методів, оскільки він дозволяє автоматично виявляти ботнети незалежно від протоколу їх управління, включаючи як централізовані, так і децентралізовані структури, і не залежить від конкретного типу шкідливої діяльності ботів.

Була розроблена архітектура інтелектуальної мультиагентної системи для виявлення та блокування ботнетів, яка відповідає типовій архітектурі ботнету. Ця система дозволяє блокувати атаки на стороні джерела, що допомагає розвантажити канали передачі від зловмисного трафіку. Також за допомогою розробленого алгоритму виявлення керуючого трафіку система може виявляти і блокувати пасивних учасників ботнету.

Був запропонований метод розподіленого виявлення керуючих компонентів ботнету, який ґрунтується на сигнатурі керуючого трафіку. Завдяки використанню мультиагентного підходу цей метод дозволяє виявляти керуючі сервери та вузли мережі, з яких здійснюється контроль атаки.

Був розроблений прототип мультиагентної системи для виявлення та блокування ботнетів, і його ефективність була підтверджена за допомогою комп'ютерного моделювання ботнету і його зловмисної діяльності.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Alaa Obeidat, Rola Yaqbeh. Smart Approach for Botnet Detection Based on Network Traffic Analysis. *Journal of Electrical and Computer Engineering*. 2022. Vol. 2022. DOI: https://doi.org/10.1155/2022/3073932

2. Ying Xing, Hui Shu, Hao Zhao, Dannong Li, Li Guo. Survey on Botnet Detection Techniques: Classification, Methods, and Evaluation. *Mathematical Problems in Engineering*. 2021. Vol. 2021. DOI: https://doi.org/10.1155/2021/6640499

3. Sandip Sonawane. A Survey of Botnet and Botnet Detection Methods. *International Journal Of Engineering Research & Technology (IJERT).* 2018. Vol. 7, No. 12. URL: https://www.ijert.org/a-survey-of-botnet-and-botnet-detection-methods

4. Devendra Sharma, Aman Sharma. Botnet and Botnet Detection Techniques. *Academic Journal of Information Security*. 2019. Vol. 01, No. 01. URL: https://www.xournals.com/assets/publications/AJIS_V01_I01_P06-10_01-2019.pdf

5. H. Dhayal, J. Kumar. Botnet and P2P Botnet Detection Strategies: A Review. *International Conference on Communication and Signal Processing (ICCSP)*. 2018. PP. 1077-1082. DOI: 10.1109/ICCSP.2018.8524529.

6. Satya Ranjan Das, Sanjay K Jena. AN OVERVIEW OF BOTNET DETECTION TECHNIQUES. *International Journal of Advanced Research in Engineering and Technology (IJARET)*. 2020. Vol. 11, No. 12. PP. 204-212. DOI: 10.34218/IJARET.11.12.2020.024

7. Foram Suthar, Nimisha Patel, Samarat V.O. Khanna. A Signature-Based Botnet (Emotet) Detection Mechanism. *International Journal of Engineering Trends and Technology*. 2022. Vol. 70, No. 5. PP. 185-193. DOI: https://doi.org/10.14445/22315381/IJETT-V70I5P220

8. Rimsha Malik, Bhavya Alankar. Botnet and Botnet Detection Techniques. *International Journal of Computer Applications.* 2019. Vol. 178, No. 17.

9. Zhihua Cui, Lei Du, Penghong Wang, Xingjuan Cai, Wensheng Zhang. Malicious code detection based on CNNs and multi-objective algorithm. *Journal of*

Parallel and Distributed Computing. 2019. Vol. 129. PP. 50-58. DOI: https://doi.org/10.1016/j.jpdc.2019.03.010.

10. Z. Cui, F. Xue, X. Cai, Y. Cao, G. Wang, J. Chen. Detection of Malicious Code Variants Based on Deep Learning. *IEEE Transactions on Industrial Informatics*. 2018. Vol. 14, No. 7. PP. 3187-3196. DOI: 10.1109/TII.2018.2822680.

11. R. U. Khan, X. Zhang, R. Kumar. Analysis of ResNet and GoogleNet models for malware detection. *J Comput Virol Hack Tech*. 2019. Vol. 15. PP. 29–37. DOI: https://doi.org/10.1007/s11416-018-0324-z

12. H. Yang, S. Li, X. Wu, H. Lu, W. Han. A Novel Solutions for Malicious Code Detection and Family Clustering Based on Machine Learning. *IEEE Access*. 2019. Vol. 7. PP. 148853-148860. DOI: 10.1109/ACCESS.2019.2946482.

13. M. Ashik, A. Jyothish, S. Anandaram, P. Vinod, F. Mercaldo, F. Martinelli, A. Santone. Detection of Malicious Software by Analyzing Distinct Artifacts Using Machine Learning and Deep Learning Algorithms. *Electronics*. 2021. Vol. 10, No. 14. P. 1694. DOI: https://doi.org/10.3390/electronics10141694

14. A. Gautam, R. Gopakumar, G. Deepa. Artificial Neural Network and Partial Pattern Recognition to Detect Malware. *Advances in Cybernetics, Cognition, and Machine Learning for Communication Technologies. Lecture Notes in Electrical Engineering*. *Springer*. 2020. Vol. 643. DOI: https://doi.org/10.1007/978-981-15-3125-5_1

15. X. Xiao, S. Zhang, F. Mercaldo. Android malware detection based on system call sequences and LSTM. *Multimed Tools*. 2019. Vol. 78. PP. 3979–3999. DOI: https://doi.org/10.1007/s11042-017-5104-0

16. Chih-Hung Lin, Hsing-Kuo Pao, Jian-Wei Liao. Efficient dynamic malware analysis using virtual time control mechanics. *Computers & Security*. 2018. Vol. 73. PP. 359-373. DOI: https://doi.org/10.1016/j.cose.2017.11.010.

17. P. Vinod, Akka Zemmari, Mauro Conti. A machine learning based approach to detect malicious android apps using discriminant system calls. *Future Generation Computer Systems*. 2019. Vol. 94. PP. 333-350. DOI: https://doi.org/10.1016/j.future.2018.11.021.

18. Ikram Ben Abdel Ouahab, Yasser Alluhaidan, Lotfi Elaachak, Mohammed Bouhorma. Detection Malware Using RGB Images and CNN Model Subclassing. *Advances in Cybersecurity, Cybercrimes, and Smart Emerging Technologies*. 2023. PP. 3-13. DOI: 10.1007/978-3-031-21101-0_1.

19. Jagsir Singh, Jaswinder Singh. A survey on machine learning-based malware detection in executable files. *Journal of Systems Architecture*. 2020. Vol. 112. DOI: https://doi.org/10.1016/j.sysarc.2020.101861.

20. I. Kuzminykh, M. Yevdokymenko. Analysis of Security of Rootkit Detection Methods. *IEEE International Conference on Advanced Trends in Information Theory (ATIT)*. 2019. PP. 196-199. DOI: 10.1109/ATIT49449.2019.9030428.

21. A. A. Awad, S. G. Sayed, S. A. Salem. Collaborative Framework for Early Detection of RAT-Bots Attacks. *IEEE Access*. 2019. Vol. 7. PP. 71780-71790. DOI: 10.1109/ACCESS.2019.2919680.

22. Ö. Aslan, S.S. Aktuğ, M. Ozkan-Okay, A. A. Yilmaz, E. Akin. A Comprehensive Review of Cyber Security Vulnerabilities. *Threats, Attacks, and Solutions. Electronics.* 2023. Vol. 12, No. 6. DOI: https://doi.org/10.3390/electronics12061333

23. Snort – Network Intrusion Detection & Prevention System. URL: https://www.snort.org/ (дата звернення 15.08.2023).

24. The Zeek Network Security Monitor. URL: https://zeek.org/ (дата звернення 17.08.2023).

25. Ramesh Singh Rawat, S. Pilli Emmanuel, R. C. Joshi. Survey of Peer-to-Peer Botnets and Detection Frameworks. *International Journal of Network Security*. 2018. Vol. 20, No. 3. PP. 547-557. DOI: 10.6633/IJNS.201805.20(3).18

26. T. Radivilova, L. Kirichenko, A. S. Alghawli, A. Ilkov, M. Tawalbeh, P. Zinchenko. The Complex Method of Intrusion Detection Based on Anomaly Detection and Misuse Detection. *IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)*. 2020. PP. 133-137. DOI: 10.1109/DESSERT50317.2020.9125051.

27. N. F. Firoz, M. T. Arefin, M. R. Uddin. Performance Optimization of Layered Signature Based Intrusion Detection System Using Snort. *Cyber Security and Computer Science. ICONCS. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*. *Springer*. 2020. Vol 325. DOI: https://doi.org/10.1007/978-3-030-52856-0_2

28. Sajad Einy, Cemil Oz, Yahya Dorostkar Navaei. The Anomaly- and Signature-Based IDS for Network Security Using Hybrid Inference Systems. *Mathematical Problems in Engineering*. 2021. Vol. 2021. DOI: https://doi.org/10.1155/2021/6639714

29. A. Khraisat, I. Gondal, P. Vamplew. Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecur*. 2019. Vol. 2, No. 20. DOI: https://doi.org/10.1186/s42400-019-0038-7

30. R. Ball. Computer Viruses, Computer Worms, and the Self-Replication of Programs. *Viruses in all Dimensions. Springer*. 2023. DOI: https://doi.org/10.1007/978-3-658-38826-3_4

31. A. Sheikh. Trojans, Backdoors, Viruses, and Worms. *Certified Ethical Hacker (CEH) Preparation Guide*. 2021. DOI: https://doi.org/10.1007/978-1-4842-7258-9_5

32. K. Hamid, M. W. Iqbal, M. Aqeel, X. Liu, M. Arif. Analysis of Techniques for Detection and Removal of Zero-Day Attacks (ZDA). *Ubiquitous Security. Communications in Computer and Information Science*. 2020. Vol. 1768. DOI: https://doi.org/10.1007/978-981-99-0272-9_17

33. M. Vyawahare, M. Chatterjee. Survey on Detection and Prediction Techniques of Drive-by Download Attack in OSN. *Advanced Computing Technologies and Applications. Algorithms for Intelligent Systems. Springer*. 2020. DOI: https://doi.org/10.1007/978-981-15-3242-9_42

34. I. Kovačević, S. Groš, K. Slovenec. Systematic Review and Quantitative Comparison of Cyberattack Scenario Detection and Projection. *Electronics*. 2020. Vol. 9, No. 10. DOI: https://doi.org/10.3390/electronics9101722

35. L. N. Tidjon, M. Frappier, A. Mammar. Intrusion Detection Using ASTDs. *Advanced Information Networking and Applications. AINA 2020. Advances in Intelligent Systems and Computing*. 2020. Vol. 1151. DOI: https://doi.org/10.1007/978-3-030-44041-1_118

36. B. Vidgen, L. Derczynski. Directions in abusive language training data, a systematic review: Garbage in, garbage out. *PLoS ONE*. 2020. Vol. 15, No. 12. DOI: https://doi.org/10.1371/journal.pone.0243300

37. L. Alevizos, VT Ta, M. Hashem Eiza. Augmenting zero trust architecture to endpoints using blockchain: A state-of-the-art review. *Security and Privacy*. 2022. Vol. 5, No. 1. DOI:10.1002/spy2.191

38. N. Polatidis, E. Pimenidis, M. Pavlidis et al. From product recommendation to cyber-attack prediction: generating attack graphs and predicting future attacks. *Evolving Systems*. 2020. Vol. 11. PP. 479–490. DOI: https://doi.org/10.1007/s12530-018-9234-z

39. S. Seo, D. Kim. SOD2G: A Study on a Social-Engineering Organizational Defensive Deception Game Framework through Optimization of Spatiotemporal MTD and Decoy Conflict. *Electronics*. 2021. Vol. 10, No. 23. DOI: https://doi.org/10.3390/electronics10233012

40. J. -H. Cho et al.. Toward Proactive, Adaptive Defense: A Survey on Moving Target Defense. *IEEE Communications Surveys & Tutorials*. 2020. Vol. 22, No. 1. PP. 709-745. DOI: 10.1109/COMST.2019.2963791.

41. S.H. Mousavi, M. Khansari, R. Rahmani. A fully scalable big data framework for Botnet detection based on network traffic analysis. *Information Sciences*. 2020. Vol. 512. PP. 629-640. DOI: https://doi.org/10.1016/j.ins.2019.10.018.

42. N. P. Selvaraj, S. Paulraj, P. Ramadass, R. Kaluri, M. Shorfuzzaman, A. Alsufyani, M. Uddin. Exposure of Botnets in Cloud Environment by Expending Trust Model with CANFES Classification Approach. *Electronics*. 2022. Vol. 11, No. 15. DOI: https://doi.org/10.3390/electronics11152350

43. M. Wazzan, D. Algazzawi, O. Bamasaq, A. Albeshri, L. Cheng. Internet of Things Botnet Detection Approaches: Analysis and Recommendations for Future Research. *Appl. Sci*. 2021. Vol. 11. DOI: https://doi.org/10.3390/app11125713

44. F. Ja'fari, S. Mostafavi, K. Mizanian et al. An intelligent botnet blocking approach in software defined networks using honeypots. *J Ambient Intell Human Comput*. 2021. Vol. 12. PP. 2993–3016. DOI: https://doi.org/10.1007/s12652-020-02461-6

45. M. A. Al-Shareeda, S. Manickam, M. A. Saare, S. Karuppayah, M. A. Alazzawi. A Brief Review of Advanced Monitoring Mechanisms in Peer-to-Peer (P2P) Botnets. *8th International Conference on Contemporary Information Technology and Mathematics (ICCITM)*. 2022. PP. 312-317. DOI: 10.1109/ICCITM56309.2022.10031721.

46. F. Antonucci, M. M. Chowdhury. Botnets as the Modern Attack Vector. *IEEE World AI IoT Congress (AIIoT)*. 2022. PP. 585-590. DOI: 10.1109/AIIoT54504.2022.9817360.

47. V. S. D. Priya, S. S. Chakkaravarthy. Containerized cloud-based honeypot deception for tracking attackers. *Sci Rep*. 2023. Vol. 13. DOI: https://doi.org/10.1038/s41598-023-28613-0

48. W. C. Shi, H.M. Sun. DeepBot: a time-based botnet detection with deep learning. *Soft Comput*. 2020. Vol. 24. PP. 16605–16616. DOI: https://doi.org/10.1007/s00500-020-04963-z

49. K. Shinan, K. Alsubhi, A. Alzahrani, M. U. Ashraf. Machine Learning-Based Botnet Detection in Software-Defined Network: A Systematic Review. *Symmetry*. 2021. Vol. 13. DOI: https://doi.org/10.3390/sym13050866

50. A. Shafee. Botnets and their detection techniques. *International Symposium on Networks, Computers and Communications (ISNCC)*. 2020. PP. 1-6. DOI: 10.1109/ISNCC49221.2020.9297307.

51. G. Rosenthal, O. E. Kdosha, K. Cohen, A. Freund, A. Bartik, A. Ron. ARBA: Anomaly and Reputation Based Approach for Detecting Infected IoT Devices. *IEEE Access*. 2020. Vol. 8. PP. 145751-145767. DOI: 10.1109/ACCESS.2020.3014619.

52. Yong Jin, Masahiko Tomoishi, Nariyoshi Yamai. Anomaly Detection on User Terminals Based on Outbound Traffic Filtering by DNS Query Monitoring and Application Program Identification. *In Proceedings of the 2021 International Conference on Human-Machine Interaction (ICHMI '21)*. 2021. PP. 47–56. DOI: https://doi.org/10.1145/3478472.3478481

53. H. A. Sukhni, M. Ahmad Al-Khasawneh, F. H. Yusoff. A Systematic Analysis for Botnet Detection using Genetic Algorithm. *2nd International Conference on Smart Computing and Electronic Enterprise (ICSCEE)*. 2021. PP. 63-66. DOI: 10.1109/ICSCEE50312.2021.9498109.

54. S. P. Majhi, S. K. Swain, P. K. Pattnaik. Issues of Bot Network Detection and Protection. *Cognitive Informatics and Soft Computing. Advances in Intelligent Systems and Computing*. 2020. Vol. 1040. DOI: https://doi.org/10.1007/978-981-15-1451-7_34

55. Evanson Mwangi Karanja, Shedden Masupe, Mandu Gasennelwe Jeffrey. Analysis of internet of things malware using image texture features and machine learning techniques. *Internet of Things*. 2020. Vol. 9. DOI: https://doi.org/10.1016/j.iot.2019.100153.

56. P. J. Beslin Pajila, E. Golden Julie. Detection of DDoS Attack Using SDN in IoT: A Survey. *Intelligent Communication Technologies and Virtual Mobile Networks. ICICV 2019. Lecture Notes on Data Engineering and Communications Technologies*. 2020. Vol. 33. DOI: https://doi.org/10.1007/978-3-030-28364-3_44

57. C. Liu et al. IEdroid:Detecting Malicious Android Network Behavior Using Incremental Ensemble of Ensembles. *IEEE 27th International Conference on Parallel and Distributed Systems (ICPADS)*. 2021. PP. 788-795. DOI: 10.1109/ICPADS53394.2021.00104.

58. Umar Iftikhar, Kashif Asrar, Maria Waqas, Syed Abbas Ali. BOTNETs: A Network Security Issue. *International Journal of Advanced Computer Science and Applications(IJACSA)*. 2020. Vol. 11, No. 11. DOI: http://dx.doi.org/10.14569/IJACSA.2020.0111155

59. СОУ 207.01:2017. Текстові документи. Загальні вимоги. Хмельницький: ХНУ, 2017. 46 с. URL: https://msn.khnu.km.ua/pluginfile.php/466522/mod_resource/content/1/132_C%20T%20A%20H%20Д%20A%20P%20T%20чист%20.pdf

60. ДСТУ 8302:2015. Бібліографічне посилання. Загальні положення та правила складання. [Чинний від 2016-07-1]. Київ, 2016. 20 с. (Державна наукова установа ―Книжкова палата України імені Івана Федорова).

ДОДАТОК А ПЕРЕЛІК НАУКОВИХ ПРАЦЬ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЦЕНТРАЛЬНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
НАУКОВА АСОЦІАЦІЯ КІБЕРБЕЗПЕКИ УКРАЇНИ

УДК 004.4

Тези доповідей

VII Міжнародної науково-практичної конференції
до 30-ти річчя кафедри кібербезпеки та програмного забезпечення

"Інформаційна безпека та комп'ютерні технології"

1 листопада 2023 року

Кропивницький 2023

УДК 004.4

Матеріали VII Міжнародної науково-практичної конференції "Інформаційна безпека та комп'ютерні технології" до 30-ти річчя кафедри кібербезпеки та програмного забезпечення: тези доповідей, 1 листопада 2023 р. – Кропивницький: ЦНТУ, 2023. – 135 с.

Наведені тези пленарних та секційних доповідей за теоретичними та практичними результатами наукових досліджень і розробок. Представлені результати теоретичних досліджень в галузях проектування інформаційних систем, технологій захисту інформації, використання сучасних інформаційних технологій в управлінні системами за різними галузями народного господарства.

Матеріали публікуються в авторській редакції.

За достовірність викладених фактів, цитат та інших відомостей відповідальність несуть автори.

ЗМІСТ

СЕКЦІЯ 1. ІНФОРМАЦІЙНА БЕЗПЕКА ДЕРЖАВИ, СУСПІЛЬСТВА ТА ОСОБИСТОСТІ

СЕКЦІЯ 1. ІНФОРМАЦІЙНА БЕЗПЕКА ДЕРЖАВИ, СУСПІЛЬСТВА ТА ОСОБИСТОСТІ

УДК 004.056

Д.С. Білик[1], Ю.П.Кльоц[1], Н.С.Петляк[1]
bilykds@khmnu.edu.ua, klots@khmnu.edu.ua, npetlyak@khmnu.edu.ua
[1]Хмельницький національний університет, м. Хмельницький

МЕТОД ВИЯВЛЕННЯ БОТІВ В ПУБЛІЧНІЙ МЕРЕЖІ НА ОСНОВІ МУЛЬТИАГЕНТНОГО ПІДХОДУ

Ботнет – це мережа, що складається з скомпрометованих хостів, керованих деяким шкідливим програмним забезпеченням (бот). Бот є частково автономною частиною шкідливого програмного забезпечення, яке контролюється віддалено.

Традиційні антивірусні інструменти засновані на пошуку сигнатур, таким чином вимагають об'ємної, точної бази сигнатур, яка має регулярно оновлюватися [1]. Ботнети можуть легко уникнути сигнатурного виявлення, оновлюючи себе частіше, ніж користувачі оновлюють антивірусні бази. Деякі боти можуть відключити антивірусні засоби системи або використовувати руткіт-технології, щоб захистити себе від виявлення на локальному вузлі мережі. Саме тому частота виявлення ботів відносно низька в порівнянні з іншими шкідливими програмами.

Мультиагентний підхід фактично позбавляє проблем масштабування при зростанні системи ідентифікації [2]. Виявлення набору однакових ознак взаємодії ботів із контролерами ботнетів можуть вирішити проблему автоматизації виявлення ботів. Як загальні ознаки ботів можна виділити:

– IP-адреса або доменне ім'я контролюючого центру ботнету;

– характеристики HTTP або IRC пакетів із певними командами управління;

– розмірність мережних пакетів;

– часові інтервали мережевих взаємодій;

– трафік зловмисної активності, наприклад, сканування, розсилка спаму, завантаження бінарних файлів; інформацію про протоколи DNS, SMTP;

– протокол обміну даними та порти транспортного рівня.

За основу при побудові методу ідентифікації ботнетів було взято типову структуру мережі Інтернет, засновану на взаємодії між автономними системами. Пропонований метод ідентифікації ботів базується на засобі захисту від розподілених атак типу «відмова в обслуговуванні» з можливістю виявлення атаки в мережі жертви та запобігання генерації атаки в мережі джерела. Отримані у процесі декомпозиції завдання можна віднести до різних класів функціональності: {Виявлення, Блокування, Дослідження, Ідентифікація, Координація, Інтерфейс}. Кожному класу може відповідати свій тип агента, який вирішує завдання класу. Таким чином, мультиагентна система ідентифікації ботнета має вигляд

$$MAS = \left\{A_{\text{detection}}, A_{blocking}, A_{discrovery}, A_{identification}, A_{coordination}, A_{\text{interface}}\right\},$$

де $A_{\text{detection}} = \left\{A^1_{\text{detction}}, \ldots, A^n_{\text{detection}}\right\}$ – множина агентів виявлення атаки типу «розподілена відмова в обслуговуванні». Агенти даного класу вирішують завдання виявлення атак і реагують на неї певним у сценарії реагуванні чином. У кожній автономній системі мережі Інтернет розташовується щонайменше один агент даного класу, $A^i_{blocking}$ де i=1..n - номер автономної системи Інтернету.

$A_{blocking} = \left\{A^1_{blocking}, \ldots, A^n_{blocking}\right\}$ - множина агентів, що вирішують завдання блокування виявленої атаки. У кожній автономній системі мережі Інтернет розташовується щонайменше один агент даного класу $A^i_{blocking}$ де i=1..n - номер автономної системи Інтернету.

$A_{discovery} = \left\{A^1_{discovery}, \ldots, A^n_{discovery}\right\}$ – множина агентів виявлення ознак робота. Клас агентів вирішує завдання визначення характерних ознак роботи робота. У кожній автономній системі мережі Інтернет розташовується щонайменше один агент даного класу $A^i_{discovery}$ де i=1..n - номер автономної системи Інтернету.

$A_{identification} = \left\{A^1_{identification}, \ldots, A^n_{identification}\right\}$ - множина агентів ідентифікації роботи робота в рамках автономної системи. Агенти цього класу аналізують трафік мережі наявність ознак функціонування ботів. У кожній автономній системі мережі Інтернет розташовується щонайменше один агент даного класу $A^i_{identification}$ де i=1..n – номер автономної системи Інтернету.

$A_{coordinatsn}$ - множина агентів мережі вирішують завдання поширення інформації про активних агентів.

$A_{interface}$ - множина агентів мережі вирішують такі завдання: контроль та моніторинг роботи мережі агентів, візуалізація атак, зберігання інформації.

Таким чином, структура мультиагентної системи ідентифікації ботів складається з наступних елементів:

1. Агент виявлення атаки типу «розподілена відмова в обслуговуванні»
2. Агент виявлення ознак робота
3. Агент ідентифікації роботів
4. Агент блокування атак. Функціонує, коли його розташування є мережею джерела атаки. Зокрема, здійснює реагування на основі інформації отриманої від агентів виявлення атак згідно з профілем мережевої безпеки (блокування систем задіяних у реалізації атаки, оповіщення відповідальних осіб по електронній пошті, SMS).
5. Агент координації. Поширює інформацію про місцезнаходження різних агентів з метою взаємодії між ними.
6. Інтерфейсний агент. Встановлюється у будь-якій точці глобальної мережі Інтернет. Призначений для контролю та моніторингу роботи мережі агентів, надання графічного інтерфейсу візуалізації виявлених атак, зберігання та забезпечення доступу до історії виявлених атак.

Концептуальний алгоритм функціонування системи полягає у наступному:

1. Агент виявлення атаки типу «розподілена відмова в обслуговуванні» виявляє атаку на підконтрольну йому мережу.
2. Агент виявлення атаки повідомляє агенту координації інформацію про мережі джерела виявленої атаки.
3. Агент координації передає агентам блокування, що знаходяться у відповідних джерелах атаки автономних системах, інформацію про вузол, що атакує.
4. Агент координації передає агенту виявлення ознак бота, що контролює мережу джерела атаки, інформацію про атакуючий вузол.
5. Агент координації передає інтерфейсному агенту інформацію про атаку.
6. Агент блокування припиняє зловмисну активність вузлів, що знаходяться в контрольованій мережі.
7. Агент виявлення ознак бота аналізує активність вузлів помічених в атаці. Внаслідок чого виявляє характерні ознаки роботи бота.
8. Агент виявлення ознак бота повідомляє характерні ознаки роботи агенту координації.
9. Агент координації розсилає інформацію про роботу ботів агентам ідентифікації ботів.
10. Агенти ідентифікації аналізують трафік своєї мережі, пробуючи виявити отримані ознаки роботи бота. У разі вдалої ідентифікації передають інформацію про бота агенту координації, який направляє її інтерфейсному агенту для подальшого прийняття рішення.

Запропонований метод (рис.1) дозволяє виявити ботнетів на основі аналізу функціонування ботів, що беруть участь у конкретній атаці. Особливість методу полягає у можливості ідентифікувати ботів, які не брали участі в атаці за рахунок роботи розподіленої мережі інтелектуальних агентів.

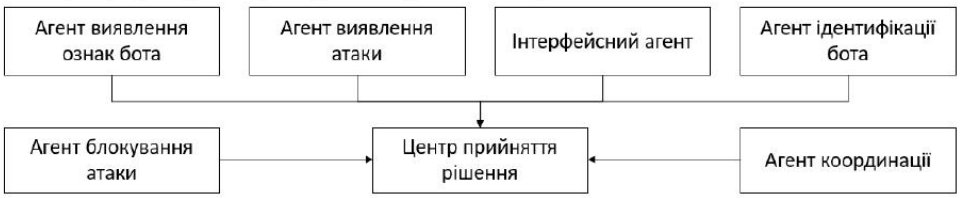

Рис. 1. Мультиагентий метод виявлення ботів

Список літератури

1. Козюра В. Д., Хорошко В. О., Шелест М. Є., Ткач Ю. М., Балюнов О.О. Захист інформації в комп'ютерних системах: підручник. – Ніжин: ФОП Лук'яненко В.В., ТПК «Орхідея», 2020. – 236с.

2. Lysenko, S.; Bobrovnikova, K.; Kharchenko, V.; Savenko, O. IoT Multi-Vector Cyberattack Detection Based on Machine Learning Algorithms: Traffic Features Analysis, Experiments, and Efficiency. Algorithms 2022, 15, 239.

Завідувачу кафедри кібербезпеки
к.т.н., доц. Кльоцу Ю.П.
Білика Дениса Сергійовича
ПІБ здобувача вищої освіти
студента ФІТ, 2 курсу, групи КБм-22-1

ЗАЯВА

З правилами чинного Положення «Про дотримання академічної доброчесності в Хмельницькому національному університеті» від 26.09.2020 (зі змінами від 26.11.2020), згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений (а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

5.12.2023
дата

підпис

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 0.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилок в документах: 11%**

	Документ		Сумарний збіг по Базі Даних	
ID: 121895 Назва: Метод та система виявлення ботів в публічній мережі Додано в БД: 2023-12-06 Автора: Білик Д.С. Керівники: Кльоц Ю.П. Консультанти: Опоненти:	Символи	Лексеми	Символи	Лексеми
	105525	896	449 (0%)	6 (1%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

UNICHECK
by Turnitin

Ім'я користувача:
Кафедра кібербезпеки

ID перевірки:
1015975242

Дата перевірки:
06.12.2023 10:20:47 EET

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
06.12.2023 10:45:16 EET

ID користувача:
100008300

Назва документа: **Білик_плагіат**

Кількість сторінок: **78** **Кількість слів:** **14900** **Кількість символів:** **118285** **Розмір файлу:** **2.86 MB** **ID файлу:** **1015654798**

2.11%
Схожість

Найбільша схожість: **0.46%** з Інтернет-джерелом (https://el-conf.com.ua/wp-content/uploads/2020/04/8%D1%87%D0%B...

1.82% Джерела з Інтернету	137	Сторінка 80
0.54% Джерела з Бібліотеки	47	Сторінка 80

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи	1

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ

КАФЕДРИ КІБЕРБЕЗПЕКИ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:
Назва: Метод та система виявлення ботів в публічній мережі

Автор: Білик Денис Сергійович
Спеціальність: 125 – Кібербезпека
Освітня програма: освітньо-професійна
Науковий керівник: Кльоц Юрій Павлович, к.т.н, доцент
Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Оригінальність тексту роботи за результатами перевірки системою Unichek складає 2,11%, оригінальність тексту роботи за результатами перевірки системою Anti-Plagiarism v-15.257 складає 0%.

Згідно з правилами чинного Положення «Про систему забезпечення академічної доброчесності у хмельницькому національному університеті» від 31.08.2023, авторська робота, обсяг оригінального тексту у відсотках до загального обсягу матеріалу в якій складає 90-100 %, визнається роботою з високою унікальністю тексту і допускається до захисту.

Керівник роботи _____ Ю.П. Кльоц

Гарант ОП _____ В.Ю. Тітова

Завідувач кафедри кібербезпеки _____ Ю.П. Кльоц

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

освітнього ступеня «магістр»

Студент Білик Денис Сергійович

Тема Метод та система виявлення ботів в публічній мережі

Спеціальність 125 – Кібербезпека

Обсяг кваліфікаційної роботи освітньо-кваліфікаційного рівня «магістр»:

кількість листів креслень - ; кількість сторінок записки 84

1. Короткий зміст роботи та прийнятих рішень В рамках роботи проведено дослідження проблем виявлення та блокування ботнетів у публічних мережах. В роботі поставлено та вирішено наступні задачі: провести аналіз існуючих підходів до виявлення ботнетів; розробити алгоритм для визначення керуючого трафіку, використовуючи технології інтелектуального аналізу даних; представити структуру мультиагентної системи для виявлення та блокування ботнетів, провести аналіз ефективності запропонованих у дослідженні алгоритмів; розробити метод для розподіленого виявлення керуючих компонентів ботнету, який дозволяє виявляти керуючі сервери та вузли мережі, використовуючи сигнатури керуючого трафіку; створити експериментальний прототип мультиагентної системи для виявлення та блокування ботнетів у рамках наукових досліджень.

2. Висновок про відповідність кваліфікаційної роботи завданню Кваліфікаційна робота відповідає поставленому завданню як в теоретичній, так і в практичній частині

3. Характеристика виконання кожного розділу роботи, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У вступі обґрунтовується актуальність теми дослідження; її зв'язок із науковими програмами, планами, темами та сформульовано мету та основні завдання дослідження. У першому розділі було досліджено життєвий цикл ботнетів, аналіз наявних методів їх виявлення, описано переваги мультиагентної системи. У другому розділі описано архітектуру мультиагентної системи та функціональну модель, розроблено алгоритм виявлення керуючого трафіку, представлено мето розподіленого виявлення керуючих компонентів ботнета. У третьому розділі розроблено метод автоматичного формування бази ботів, описано схему проведення дослідження та отримані результати. У четвертому розділі сформовано вимоги до мультиагентної системи виявлення ботнетів та представлено структуру системи.

4. Позитивні сторони роботи Кваліфікаційна робота має комплексну наукову і практичну цінність. Наукова цінність полягає у пропозиції способу виявлення ботнетів. Практична цінність полягає у підвищенні рівня виявлення та блокування роботи ботнетів у інформаційно-комунікаційних мережах.

5. Негативні сторони роботи Недоліком реалізованої системи є те, що максимальну ефективність можна досягти шляхом розгортання системи у різних публічних мережах

6. Оцінка графічного оформлення та пояснювальної записки роботи Оформлення всіх матеріалів кваліфікаційної роботи є якісним, здійснене з дотриманням актуальних стандартів та інституційних положень ХНУ. Пояснювальна записка відповідає нормам щодо її оформлення як за структурою, так і за представленням і форматуванням матеріалу.

7. Відгук про роботу в цілому Кваліфікаційна робота заслуговує позитивної оцінки. Весь матеріал кваліфікаційної роботи структурований, чіткий та наскрізно пов'язаний.

8. Інші зауваження

9. Оцінка кваліфікаційної роботи Враховуючи всі позитивні та негативні сторони представленої кваліфікаційної роботи, можна зробити висновок, що вона заслуговує оцінку «добре».

РЕЦЕНЗЕНТ (прізвище, ім'я, по батькові, посада, місце роботи) Завідувач кафедри автоматизації, комп'ютерно-інтегрованих технологій та робототехніки, доктор технічних наук, професор Мартинюк Валерій Володимирович.

« 8 » грудня 2023 року _____ (підпис)