Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Виявлення аномалій у мережевому трафіку SCADA

(тема)

Виконав:
здобувач 2 року навчання,
групи СПм-23-4
Даниїл ІВАНЧЕНКО
(власне ім’я, прізвище)

Спеціальність
123 «Комп’ютерна інженерія»
(код і повна назва спеціальності)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма
Системне програмування
(повна назва освітньої програми)

Керівник: проф. Ігор РУБАН
(посада, власне ім’я, прізвище)

Допускається до захисту

Завідувач кафедри ЕОМ _____ Андрій КОВАЛЕНКО
(підпис) (власне ім’я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління

Кафедра електронних обчислювальних машин

Рівень вищої освіти другий (магістерський)

Спеціальність 123 «Комп’ютерна інженерія»
(код і повна назва)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне програмування
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
“____” _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Іванченку Даниїлу Ігоровичу
(прізвище, ім’я, по батькові)

1. Тема роботи Виявлення аномалій у мережевому трафіку SCADA

затверджена наказом по університету від “ 21 ” квітня 2025 р. № 296 Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 16 червня 2025 р.

3. Вхідні дані до роботи Приклад аномального трафіку

4. Перелік питань, що потрібно опрацювати у роботі

Аналіз архітектури та функціонування SCADA-систем

Огляд потенційних загроз і типів атак на SCADA-системи

Огляд існуючих методів виявлення аномалій у мережевому трафіку

Побудова системи виявлення аномалій для SCADA-трафіку

Розробка та навчання моделей виявлення аномалій

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій 19 слайдів

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Огляд методів виявлення аномального трафіка	22.04.25-29.04.25	
	в промислових сстемах		
2	Вибір та обгрунтування методики дослідження	30.04.25-05.05.25	
3	Вибір інструментальних засобів	06.05.25-09.05.25	
4	Розробка моделей	10.05.25-20.05.25	
5	Проведення експериментів	21.05.25-02.06.25	
6	Оформлення матеріалів кваліфікаційної роботи	03.06.25-05.06.25	
7	Подання кваліфікаційної роботи керівникові та	06.06.25-09.06.25	
	її попередній захист		
8	Подання кваліфікаційної роботи на рецензу-	10.06.25-12.06.25	
	вання		

Дата видачі завдання " 21 " квітня 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____ проф. Ігор РУБАН
(підпис) (посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 59 с., 23 рис., 1 дод., 7 джерел.

SCADA-СИСТЕМА, АНОМАЛІЇ, МЕРЕЖЕВИЙ ТРАФІК, КІБЕРБЕЗПЕКА, КРИТИЧНА ІНФРАСТРУКТУРА, ВИЯВЛЕННЯ ВТОРГНЕНЬ, МАШИННЕ НАВЧАННЯ, НЕКОНТРОЛЬОВАНЕ НАВЧАННЯ, IDS (СИСТЕМА ВИЯВЛЕННЯ ВТОРГНЕНЬ), МОДЕЛІ КЛАСИФІКАЦІЇ, АНАЛІЗ ПОВЕДІНКИ, ЗАХИСТ МЕРЕЖ.

Метою кваліфікаційної роботи є розробка методу виявлення аномалій у мережевому трафіку SCADA-систем з метою підвищення рівня кібербезпеки об’єктів критичної інфраструктури.

У ході виконання кваліфікаційної роботи було проаналізовано архітектуру та принципи функціонування SCADA-систем, досліджено типи можливих атак на мережеву інфраструктуру SCADA, а також проведено огляд сучасних підходів до виявлення аномалій у трафіку. В якості основи для побудови системи виявлення були використані методи машинного навчання, зокрема алгоритми неконтрольованого навчання. Було реалізовано процес збору, попередньої обробки та аналізу трафіку, з використанням реальних або симульованих даних. Проведено експериментальну перевірку ефективності моделей для класифікації аномалій, оцінено якість за допомогою метрик (precision, recall, F1-score, AUC). Робота демонструє, що застосування алгоритмів машинного навчання дозволяє своєчасно виявляти потенційно небезпечну активність у SCADA-системах.

ABSTRACT

Master’s thesis: 59 pages, 23 figures, 1 appendices, 7 sources.

SCADA SYSTEM, ANOMALIES, NETWORK TRAFFIC, CYBERSECURITY, CRITICAL INFRASTRUCTURE, INTRUSION DETECTION, MACHINE LEARNING, UNSUPERVISED LEARNING, IDS (INTRUSION DETECTION SYSTEM), CLASSIFICATION MODELS, BEHAVIOR ANALYSIS, NETWORK PROTECTION.

The major goal of this thesis is to develop a method for detecting anomalies in SCADA system network traffic to enhance the cybersecurity of critical infrastructure.

During the research, the architecture and operational principles of SCADA systems were analyzed, along with an investigation into potential types of attacks on SCADA network infrastructure. A review of modern anomaly detection approaches in network traffic was also conducted. The proposed detection system is based on machine learning techniques, particularly unsupervised learning algorithms. The work involved the collection, preprocessing, and analysis of traffic data using both real and simulated datasets. An experimental evaluation of the models' effectiveness in anomaly classification was performed, and their performance was measured using standard metrics such as precision, recall, F1-score, and AUC. The results demonstrate that machine learning algorithms can effectively and promptly identify potentially malicious activities in SCADA environments.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

АСК – автоматизована система керування

ARP – протокол визначення фізичної адреси (англ., Address Resolution Protocol)

API – програмний інтерфейс прикладного програмування (англ., Application Programming Interface)

CNN – згорткова нейронна мережа (англ., Convolutional Neural Network)

FN – хибнонегативне спрацювання (англ., False Negative)

FP – хибнопозитивне спрацювання (англ., False Positive)

IDS – система виявлення вторгнень (англ., Intrusion Detection System)

IoT – Інтернет речей (англ., Internet of Things)

MITM – атака типу "людина посередині" (англ., Man-In-The-Middle)

PLC – програмований логічний контролер (англ., Programmable Logic Controller)

SCADA – система контролю та збору даних (англ., Supervisory Control and Data Acquisition)

TP – правильне позитивне спрацювання (англ., True Positive)

TN – правильне негативне спрацювання (англ., True Negative)

ВСТУП

У сучасному світі системи диспетчерського управління та збору даних (SCADA) відіграють ключову роль у забезпеченні стабільного функціонування критичної інфраструктури, зокрема в енергетиці, водопостачанні, транспорті та промисловості. На відміну від традиційних IT-систем, SCADA-системи не лише передають та зберігають інформацію, а й безпосередньо керують фізичними процесами в реальному світі – наприклад, відкривають або закривають клапани, регулюють роботу насосів і двигунів.

Зі зростанням потреби у віддаленому моніторингу та керуванні, SCADA-системи дедалі частіше інтегруються з IT-інфраструктурою та підключаються до Інтернету. Така інтеграція розширює можливості управління, але водночас значно збільшує площину атаки. Через обмежену початкову увагу до питань кібербезпеки в цих системах, вони стають вразливими до як традиційних атак з боку кіберзлочинців, так і до специфічних загроз, пов’язаних із промисловим середовищем.

Одним із протоколів, який широко використовується в SCADA-системах, є IEC 60870-5-104. Цей протокол, розроблений для передачі керуючих команд і телеметрії, працює поверх TCP/IP, що робить його потенційною ціллю для атак, характерних для класичних мереж.

У цій кваліфікаційній роботі розглядається підхід до виявлення аномалій у мережевому трафіку SCADA-систем, що використовують протокол IEC 60870-5-104. Метою роботи є створення інструменту виявлення атак шляхом аналізу відхилень у поведінці трафіку, зокрема через аналіз часових інтервалів між пакетами. Особливу увагу приділено використанню системи виявлення вторгнень Bro (нині відомої як Zeek), яка завдяки своїй гнучкій архітектурі дозволяє створювати як спеціалізовані парсери протоколів, так і механізми обробки подій у режимі реального часу.

У рамках дослідження було реалізовано повноцінну систему виявлення аномалій, яка працює у двох режимах – навчальному та детекції – та перевірено її ефективність на реальному трафіку лабораторної SCADA-системи. Результати експериментів демонструють здатність системи ефективно виявляти різні типи атак, включаючи сканування портів, MITM та атаки з підміною даних.

Таким чином, дана робота робить внесок у підвищення рівня безпеки SCADA-систем шляхом застосування адаптивних підходів до аналізу трафіку та демонструє перспективність використання аномалійного виявлення в індустріальних мережах.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 SCADA системи

Системи диспетчерського управління і збору даних (SCADA) використовуються для забезпечення деяких із найважливіших потреб сучасного суспільства. На відміну від звичайних IT-систем, SCADA-системи не лише зберігають і обробляють інформацію. Вони також керують фізичними процесами у реальному світі, наприклад, відкриттям і закриттям клапанів або регулюванням швидкості роботи насосів і двигунів.

Історично склалося так, що промислові системи керування були незалежними одна від одної, і кожна система використовувала власні пропрієтарні протоколи зв’язку. Наступним етапом еволюції стало з’єднання різних підсистем через локальні мережі. Проте зв’язок все ще забезпечувався за допомогою закритих протоколів, розроблених без урахування або з мінімальним урахуванням безпеки. Зі зростанням потреби у віддаленому моніторингу та керуванні з операційного центру SCADA-системи вже не можуть існувати ізольовано. Це створює нові виклики для безпеки таких систем. Поверхня атаки значно зростає, коли SCADA-системи підключаються до Інтернету.

Іноді SCADA-система може здаватися від’єднаною від Інтернету, хоча насправді вона має опосередковане з’єднання. На рисунку 1.1 зображено можливу конфігурацію корпоративної мережі, яка включає дві SCADA-мережі, що не мають прямого підключення до Інтернету, але можуть бути доступні через офісну мережу. Дуже важливо тримати окремі сегменти мережі ізольованими, щоб зменшити ризик поширення зараження на всі частини системи. Було зафіксовано кілька інцидентів, коли SCADA-система була скомпрометована через корпоративну мережу, що мала підключення до Інтернету. У 2014 році на сталеливарному заводі в Німеччині стався напад із

використанням складної соціотехнічної атаки. Хакери отримали доступ до корпоративної мережі, а потім пробралися до виробничої мережі, завдавши серйозної шкоди всій системі [1].

Існують також способи компрометації систем, які взагалі не мають з'єднання з Інтернетом. Наприклад, оператор може оновлювати прошивку системи за допомогою USB-накопичувача, попередньо завантаживши оновлення, і випадково перенести на нього шкідливе програмне забезпечення. Коли такий USB-накопичувач буде підключено до SCADA-хоста для оновлення, система теж буде заражена. У 2012 році вірус був виявлений у системі керування турбіною на електростанції у США. З'ясувалося, що вірус потрапив до системи через флешку, якою користувався технік [1].

Протокол, який було досліджено у цій роботі, – це IEC 60870-5-104, що використовується для мережевої комунікації в SCADA-системах. IEC 60870-5 – це набір стандартів, призначених для передачі команд і інформації в мережах SCADA. Спочатку стандарт IEC 60870-5 розроблявся переважно для потреб електроенергетичної галузі, хоча може бути застосований і в інших промислових сферах.

Стандарт було завершено у 1995 році з появою протоколу IEC 60870-5-101, який передбачав передавання даних через повільні послідовні канали зв'язку. У грудні 2000 року було опубліковано стандарт IEC 60870-5-104. Основна відмінність полягає в тому, що замість послідовних з'єднань цей протокол використовує стек TCP/IP для комунікації в мережах.

Огляд стеку протоколів IEC 60870-5-104 наведено на рисунку 1.2. Рівні User Process та Application є майже ідентичними до відповідних рівнів у стандарті IEC 60870-5-101.

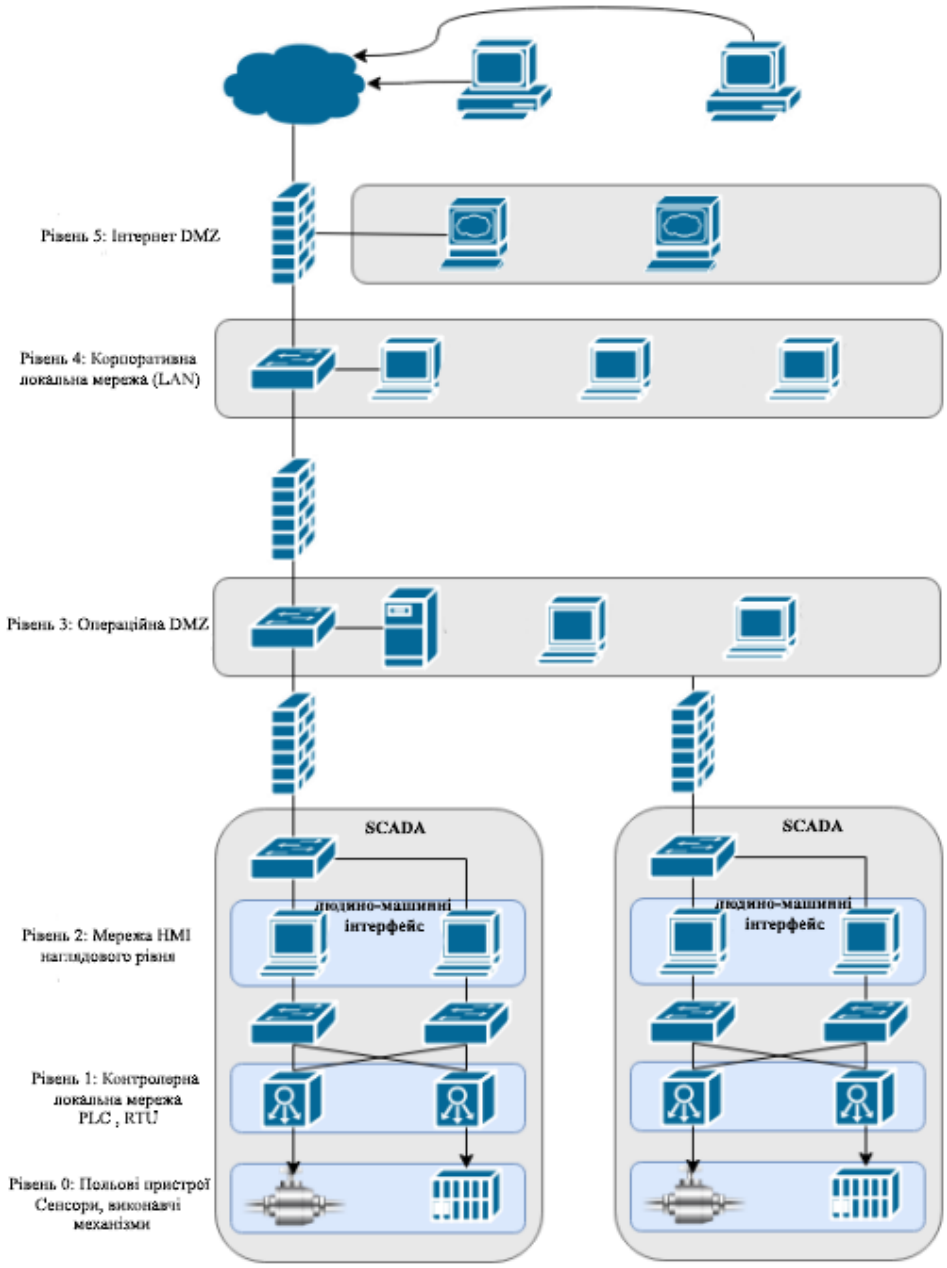

Рисунок 1.1 – Огляд можливого налаштування корпоративної мережі

User Process	IEC 60870-5-101
Application	
Transport	TCP/IP Transport and network protocols
Network	
Link	
Physical	

Рисунок 1.2 – Огляд стеку протоколів

На рисунку 1.3 показано формат кадру Application Protocol Data Unit (APDU), як його визначено в протоколі IEC 60870-5-104. Компонент Application Protocol Control Information (APCI) не був необхідним у протоколі IEC 60870-5-101, але є обов'язковим у IEC 60870-5-104. APCI використовується для керування передачею даних. Компонент Application Service Data Unit (ASDU) є необов'язковим. [2]

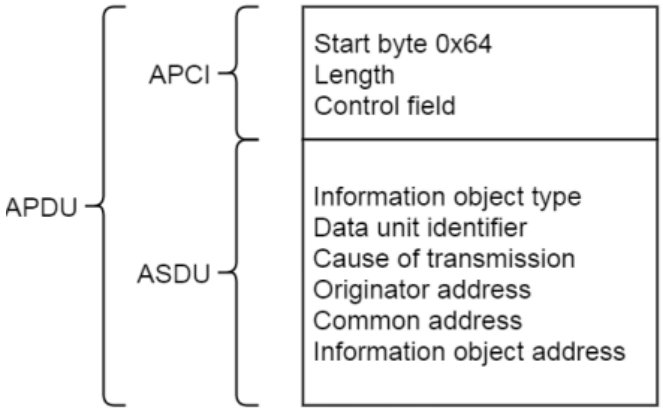

Рисунок 1.3 – Формат блоку даних протоколу додатку (APDU)

Існує три різні формати поля керування в APCI, які називаються форматами I, S та U. Формат інформаційної інструкції (I) використовується для APDU, що містять ASDU. Це єдиний формат, який передає інформацію. Формат безнумерованого керування (U) використовується лише в APDU, які містять лише APCI. Формат U застосовується як механізм запуску та зупинки потоку комунікацій. Формат нагляду (S) відповідає за контроль транспортування APDU. APDU, які використовують формат S, не включають ASDU, подібно до формату U.

Протокол IEC 60870-5-104 розроблений спеціально для транспортування інформації та команд у середовищах SCADA. Надзвичайно важливо, щоб пакети передавалися надійним способом.

Перехід до використання TCP/IP як транспортного протоколу також відкриває можливість застосування атак, характерних для звичайних IT-мереж. У цьому звіті було розглянуто три типи атак. Перший тип – це

розвідувальна атака, відома як сканування портів. Другий тип – атака "людина посередині" (MITM, Man-in-the-Middle). Третій тип – атака, що називається атакою передбачення (prediction attack).

Сканування портів може використовуватись для з’ясування, які порти відкриті та які служби працюють на цільовому хості. Ця інформація згодом може бути використана для атаки на хост за допомогою експлойтів, що націлені на конкретні служби. Атаку можна здійснити шляхом спроб підключення до різних портів і перевірки, чи вдалося встановити з’єднання. Такий тип атаки можна виявити, коли протягом короткого часу виконується багато спроб підключення. Щоб зробити атаку менш помітною, можливе неповне встановлення з’єднання. Це знижує ризик того, що система створить журнали, які можуть викрити атакуючого. Такий підхід полягає у надсиланні лише першого пакета запиту, а після відповіді хоста з’єднання негайно закривається. Для цього потрібні привілеї на роботу з сирими пакетами (raw packets). У цьому звіті використовується саме цей варіант атаки.

У звіті розглядаються дві версії атаки. Перша – коли атаку здійснює хост, новий для мережі, або такий, що зазвичай не спілкується з цільовим пристроєм. Схематичне зображення можливої сценарної атаки наведено на рисунку 1.4.

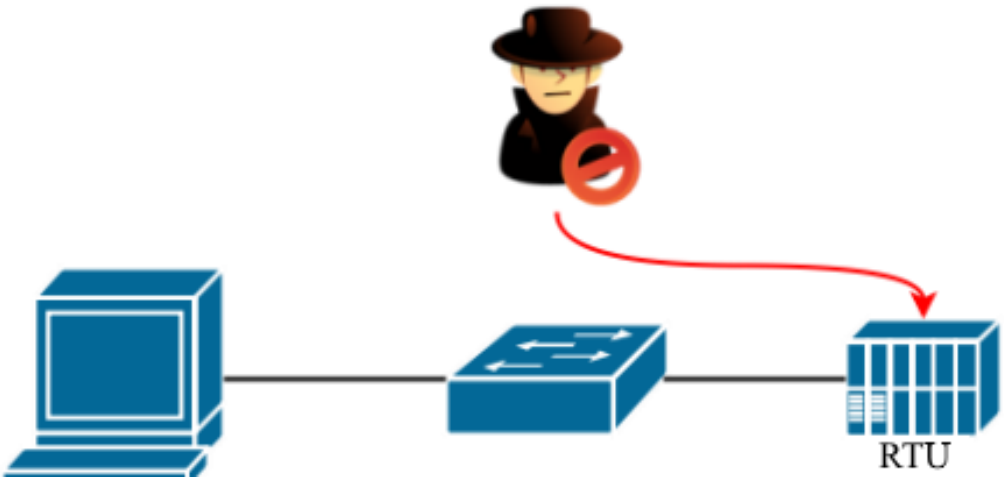

Рисунок 1.4 – Огляд атаки сканування портів

Інша версія атаки передбачає, що її джерелом є вузол, який уже є частиною нормально функціонуючої мережі. Це може статися у випадку, якщо хост було скомпрометовано або його використовує внутрішній зловмисник для збору інформації про систему. Така атака є складнішою для виявлення, оскільки надходить від хоста, який раніше поводився нормально та не викликав підозр.

Сценарій такої атаки зображено на рисунку 1.5.

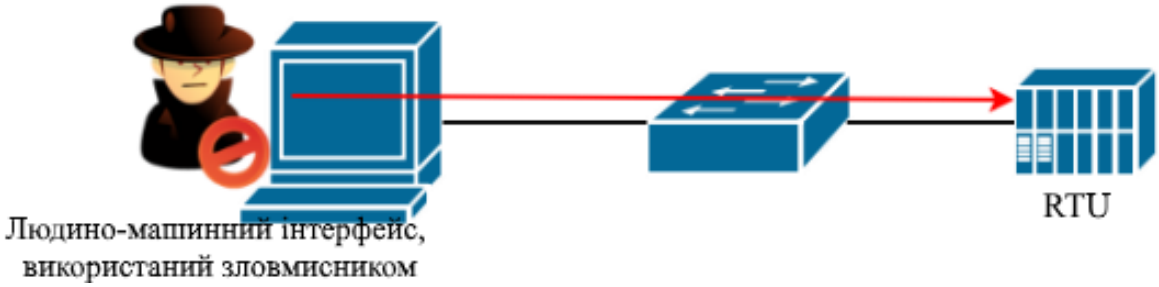

Рисунок 1.5 – Огляд сканування портів, що надходять з відомого хоста

Атака типу «людина посередині» (MITM), як підказує назва, полягає в тому, що зловмисник намагається втрутитися між двома хостами, які обмінюються даними, з метою перехоплення трафіку. Існує кілька способів реалізації такого типу атаки залежно від типу зв’язку між хостами.

У мережі, побудованій на основі TCP/IP, кожен мережевий інтерфейс має власну MAC-адресу, а кожен хост – свою IP-адресу. Протокол ARP (Address Resolution Protocol) використовується в IP-мережах для встановлення відповідності між MAC-адресою інтерфейсу та IP-адресою хоста. Саме ARP-протокол може бути використаний для реалізації MITM-атаки.

Атака починається з того, що зловмисник надсилає підроблені ARP-пакети. Ці пакети створюються таким чином, щоб здавалося, що єдиний шлях для двох хостів обмінюватися даними — це надсилати пакети на інтерфейс зловмисника. Огляд цієї атаки зображено на рисунку 1.6. Зелена

стрілка позначає звичайний обмін даними між двома хостами, а червоні стрілки – це підроблені ARP-пакети, надіслані зловмисником.

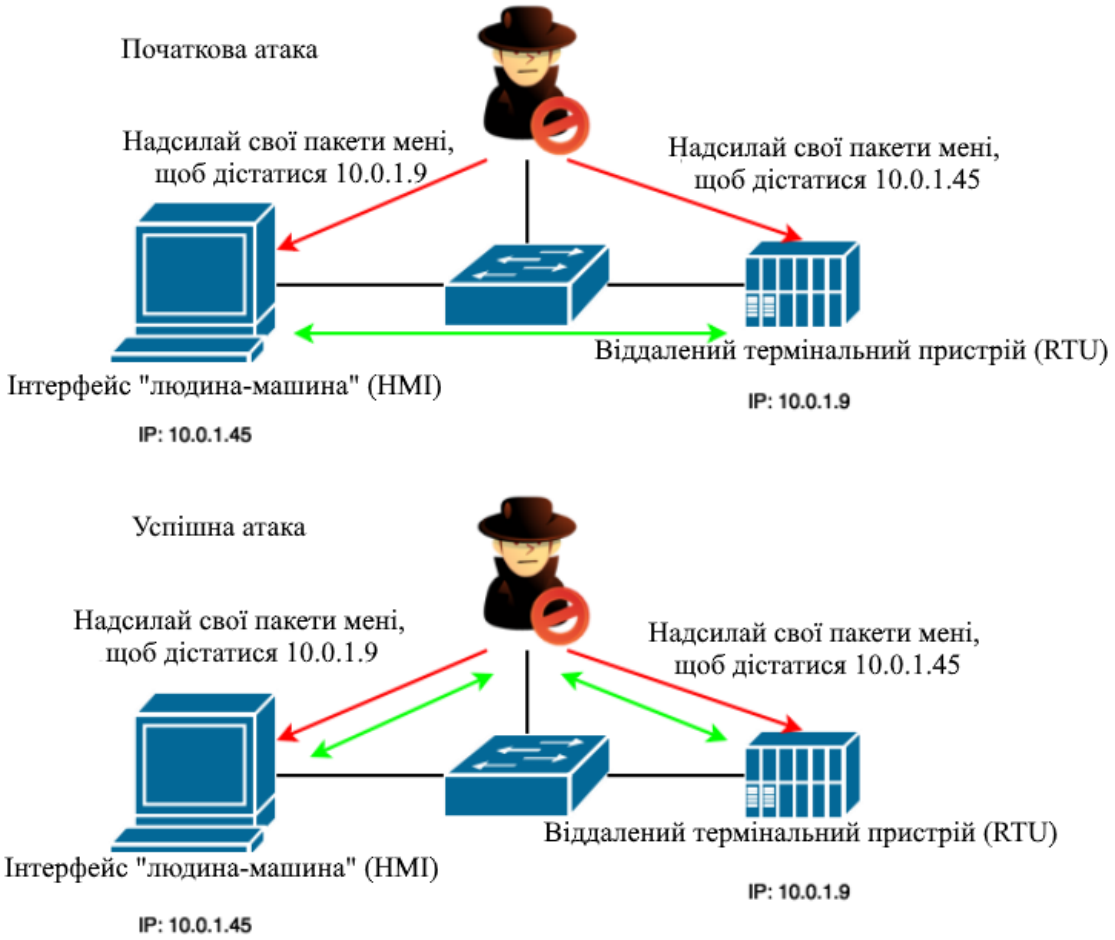

Рисунок 1.6 – Огляд двох етапів MITM-атаки

Коли атака типу «людина посередині» (MITM) проходить успішно, маршрут комунікації змінюється, і весь трафік перехоплюється зловмисником. У такому випадку зловмисник отримує повний контроль над усіма даними, що передаються від одного хоста до іншого. Оскільки протокол IEC 60870-5-104 не використовує шифрування, атакуючий має змогу вільно як читати, так і змінювати передавані пакети.

Атака типу «передбачення» (prediction attack) – це атака, при якій зловмисник намагається передбачити значення, які будуть використані хостом. Якщо передбачення виявляється успішним, це дозволяє зловмиснику вставляти підроблені пакети в уже встановлене з’єднання.

Протокол керування передачею даних (TCP) забезпечує надійне транспортування інформації. TCP використовує комбінацію номерів послідовності (SEQ) і підтвердження (ACK) для відстеження пакетів та відновлення їх порядку після прибуття. Номери SEQ/ACK також служать захистом від повторних атак (replay attacks), коли перехоплений пакет надсилається зловмисником повторно, щоб спричинити небажану дію. Початковий номер SEQ має бути випадковим, щоб унеможливити передбачення. Проте після встановлення з'єднання номери SEQ/ACK збільшуються детерміновано – залежно від обсягу переданих даних у пакеті.

Основна відмінність між протоколом IEC 60870-5-104 та звичайними IT-протоколами, такими як FTP, полягає в тривалості з'єднання. У звичайних IT-середовищах TCP-з'єднання зазвичай короткочасні – тривають кілька секунд або хвилин. У SCADA-системах з'єднання можуть залишатися відкритими протягом тижнів або навіть довше. Така тривалість створює можливість для зловмисника передбачити, які SEQ/ACK номери буде використано у наступному пакеті.

Ще один важливий аспект полягає в тому, що довжина пакетів IEC 60870-5-104 визначається типом команди. Це дозволяє зловмиснику заздалегідь підготувати підроблений пакет і надіслати його незадовго до того, як має прибути справжній пакет. Оскільки обидва пакети надходять до хоста, останній може сприйняти справжній пакет як повторну передачу через втрату пакета й відкинути його, що дає змогу атаці пройти успішно.

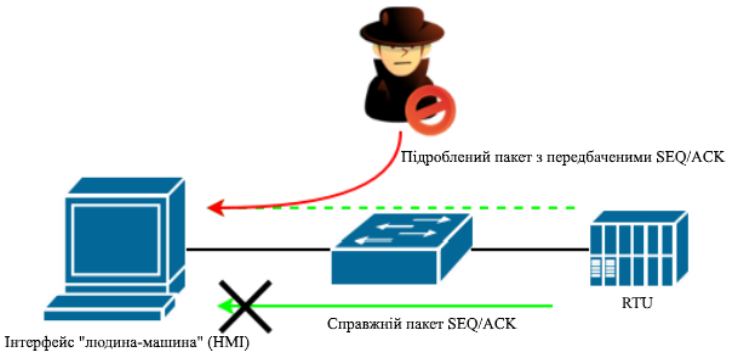

Рисунок 1.7 – Огляд атаки на передбачення

На рисунку 1.7 показано приклад успішної атаки типу передбачення. Підроблений пакет, позначений червоною стрілкою, використовує адресу джерела RTU (віддаленого термінального пристрою), щоб виглядати так, ніби саме RTU надіслав цей пакет. Підроблений пакет також зображено пунктирною зеленою стрілкою. Реальний пакет, який надсилається з RTU, буде відкинутий, оскільки інший пакет із такими ж значеннями SEQ/ACK уже надійшов першим.

Для успішного виконання такої атаки зловмисник має добре розуміти протоколи, що використовуються в мережі. Крім того, йому потрібно мати змогу прослуховувати мережевий трафік, щоб коректно передбачити значення SEQ/ACK. Це можливо, якщо передача даних здійснюється через мережевий хаб, а не комутатор (switch), оскільки хаб передає трафік усім пристроям у мережі. Проте, навіть якщо використовується комутатор, зловмисник може атакувати його й змусити працювати як хаб. Це можна зробити, перевантаживши таблицю MAC-адрес комутатора, щоб той більше не міг коректно визначати, куди пересилати пакети.

Також зловмиснику потрібно спостерігати за частотою надсилання пакетів, щоб мати змогу точно вставити свій підроблений пакет у правильний момент потоку, що підвищує ймовірність успішної атаки.

1.2 Системи виявлення вторгнень

Системи виявлення вторгнень (IDS) можна поділити на два типи залежно від методу виявлення потенційних загроз: на основі зловживань (misuse-based) та на основі аномалій (anomaly-based).

Виявлення на основі зловживань спирається на базу даних раніше проаналізованих атак. Для кожної з атак на основі її характеристик створюється сигнатура, яка додається до бази даних. Такий підхід повністю залежить від актуальності й повноти бази сигнатур. Якщо буде застосовано

нову атаку, яка ще не має сигнатури, система її не виявить. Навіть незначна модифікація відомої атаки може дозволити їй уникнути виявлення.

Натомість виявлення на основі аномалій ґрунтується на моделі нормальної поведінки системи. Порівнюючи поточний трафік із цією моделлю, можна виявити підозрілу або зловмисну активність, що не відповідає очікуваній поведінці. Такий підхід дозволяє також виявляти так звані атаки нульового дня (zero-day), тобто нові типи атак, які ще не були проаналізовані.

Огляд основних компонентів виявлення аномалій зображено на рисунку 1.8. Попередньо створена модель порівнюється з даними в реальному часі. Якщо спостерігається значне відхилення, система виявлення аномалій генерує попередження.

Недоліком виявлення аномалій є висока кількість хибнопозитивних спрацьовувань (false positives). Це зумовлено тим, що модель ніколи не буває ідеальною. Вона має бути достатньо суворою, щоб не пропускати атаки, але водночас досить гнучкою, щоб не створювати надмірну кількість помилкових попереджень [3].

Рисунок 1.8 – Огляд системи виявлення аномалій

Метою впровадження автоматичної системи виявлення є зменшення навантаження на оператора та скорочення часу між здійсненням атаки та її виявленням. Захисні заходи можуть бути реалізовані лише після виявлення атаки, тому своєчасне виявлення є критично важливим.

У порівнянні з офісним IT-середовищем, трафік у SCADA-системах є більш детермінованим. Це означає, що легше побудувати модель нормально функціонуючої системи. Проте для створення такої моделі під час роботи SCADA-системи необхідно переконатися, що навчальні дані очищені (санітовані) та не містять уже наявних атак. Якщо в навчальному наборі присутні атаки, то вони будуть включені до моделі як частина нормальної поведінки, що призведе до неправильного функціонування системи виявлення.

Огляд фази навчання такої системи зображено на рисунку 1.9.

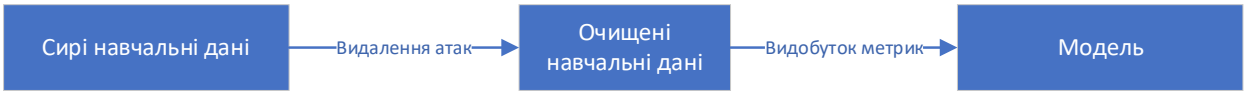

Рисунок 1.9 – Огляд процесу створення моделі системи

Системи виявлення вторгнень (IDS) також можна класифікувати за місцем їх розміщення. Хостова IDS (HIDS) встановлюється безпосередньо на хості, який вона моніторить. Така система спостерігає за станом операційної системи та встановлених програм. Наприклад, HIDS може виявити, якщо програма раптово почала змінювати файли за межами своїх звичних директорій.

Мережева IDS (NIDS), навпаки, аналізує трафік у межах мережі. Її слід встановлювати в ключових точках мережі для ефективного моніторингу. Щоб NIDS могла «слухати» трафік, використовуються спеціальні технічні засоби. Наприклад, апаратні TAP-и – це пристрої, які дублюють усі передавання по кабелю і забезпечують повне захоплення кадрів без втрат. Але такі пристрої можуть бути дорогими. Альтернативою є дзеркалювання портів (mirrored ports), яке підтримується багатьма сучасними комутаторами і не потребує додаткових витрат. Проте оскільки ця функція реалізована програмно, вона має найнижчий пріоритет і може призводити до втрати пакетів у завантажених мережах. Якщо частина комунікації проходить через

бездротову мережу, можливо здійснити перехоплення трафіку за допомогою пасивного прослуховування.

У цьому дослідженні розглядається винятково мережева IDS, яка використовує виявлення аномалій.

Існує кілька популярних відкритих мережевих IDS:

- Snort (належить компанії Cisco) – одна з найвідоміших IDS, що базується на правилах. Користувачі можуть створювати власні правила або підписатися на офіційні оновлення. Завантаження доступне з www.snort.org.

- Suricata – ще одна потужна NIDS, розроблена Open Information Security Foundation, яку можна завантажити з www.suricata-ids.org. Вона підтримує багатопоточність (на відміну від Snort), може використовувати графічні процесори для прискорення обробки трафіку, а також має власний скриптовий рушій для опису складних атак.

- Bro (тепер відома як Zeek) – це не просто IDS, а велика платформа для аналізу трафіку та виявлення загроз, що використовує власну мову програмування BNPL (Bro Network Programming Language). Bro не спирається на правила чи сигнатури, як Snort або Suricata, а формує високорівневі журнали активності, що зручно для подальшого аналізу.

Ці системи мають різні підходи до виявлення вторгнень, що дає змогу адаптувати їх під специфіку SCADA-середовища.

Усі три згадані відкриті мережеві IDS мають функціональність для аналізу та виявлення протоколів TCP/IP. Проте жодна з них не має вбудованої підтримки для аналізу протоколу IEC 60870-5-104 на рівні прикладного шару. Тому, щоб використовувати один із цих інструментів для моніторингу SCADA-систем, необхідно розробити парсер (аналітичний модуль) цього протоколу для вибраної IDS.

Оскільки всі три системи IDS – відкриті (open-source), їхній вихідний код можна завантажити, модифікувати та розширити відповідно до потреб. Це відкриває можливість самостійно реалізувати підтримку протоколу IEC 60870-5-104.

Для Snort і Suricata необхідно створити спеціальний препроцесор – модуль, який попередньо обробляє трафік і аналізує специфіку протоколу.

У Bro (тепер Zeek) для цього використовується система плагінів, які реалізовують парсери для нових протоколів.

Таким чином, вибір IDS передбачає додаткову роботу зі створення власного модуля для аналізу специфічного промислового протоколу, що є ключовим для повноцінного контролю SCADA-середовищ.

1.3 Правило трьох сигм

Правило трьох сигм – це емпіричне правило, яке застосовується до нормального розподілу і стверджує, що 99.7% значень знаходяться на відстані не більше трьох стандартних відхилень від середнього значення. Це правило використовувалося в дослідженнях для класифікації даних, наприклад, у [4]. Завдяки цьому правилу можна визначити порогові значення, які надалі використовуються для виявлення аномалій.

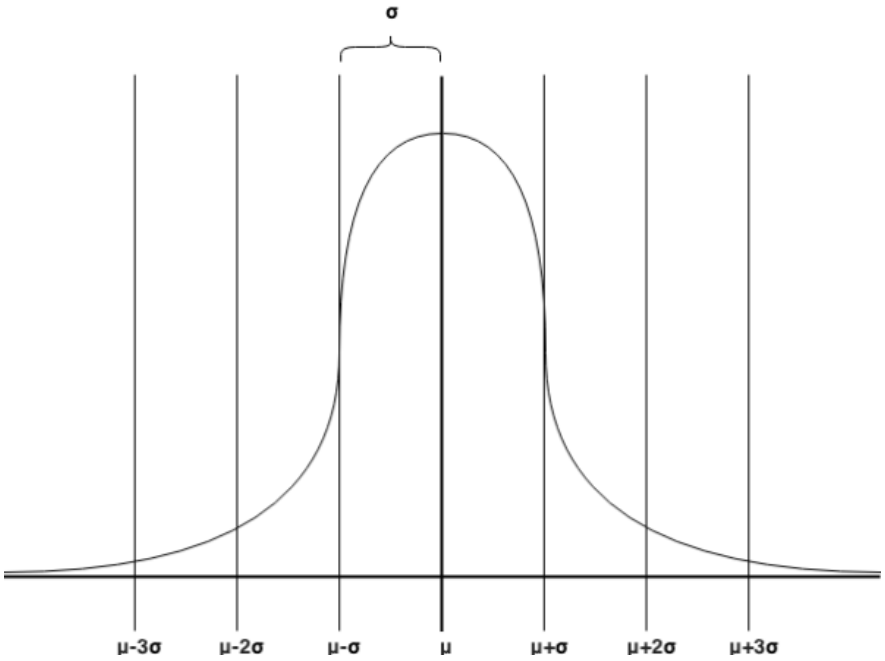

Рисунок 1.10 – Показано нормальний розподіл із середнім значенням та стандартним відхиленням

На рисунку 1.10 показано нормальний розподіл із середнім значенням у центрі (μ). У межах одного стандартного відхилення (σ) від середнього очікується приблизно 68% значень. Відповідно, у межах двох стандартних відхилень міститься близько 95% значень, а в межах трьох – приблизно 99.7%. Значення, що виходять за межі трьох сигм, становлять лише 0.3% і можуть вважатися рідкісними.

2 ВИБІР ІНСТРУМЕНТІВ І ПІДХОДІВ

2.1 Датчик вторгнення

Для реалізації детектора аномалій необхідно задовольнити кілька базових вимог. По-перше, повинна існувати можливість парсингу протоколу IEC 60870-5-104 – або з підтримкою «з коробки», або шляхом розширення IDS власним парсером. По-друге, система має бути здатною до самонавчання та створення моделі комунікації.

У якості базової платформи для створення детектора аномалій розглядалися три системи виявлення вторгнень. Як уже зазначалося, Snort і Suricata значною мірою покладаються на сигнатури та правила для виявлення атак. Натомість Bro (тепер відомий як Zeek) у меншій мірі залежить від правил. Основний метод виявлення атак у Bro полягає у використанні скриптів, написаних мовою Bro Network Programming Language (BNPL), що дозволяє створювати більш складні логіки, ніж традиційні сигнатури.

Використовуючи Bro, можна реалізувати всі частини системи в межах самого фреймворку Bro та його компонентів, без необхідності у сторонніх ресурсах. Крім того, Bro має довгий перелік підтримуваних протоколів, зокрема інший SCADA-протокол — DNP3, схожий на IEC 60870-5-104. Це свідчить про те, що Bro уже застосовувався для аналізу SCADA-комунікацій.

Сценарії Bro є подієво-орієнтованими, що добре підходить для аналізу мережевого трафіку. Коли виникає подія, можна виконати різні дії залежно від фази, в якій перебуває система виявлення. Якщо система ще знаходиться на етапі навчання – подія додається до білого списку. Якщо ж іде фаза виявлення – подія порівнюється з білим списком і вживаються відповідні заходи.

Жодна з розглянутих IDS-систем не має вбудованого парсера для протоколу IEC 60870-5-104. Проте, оскільки всі IDS є open-source, можливо створити власний парсер для будь-якої з них.

Оскільки Bro (тепер відомий як Zeek) не має вбудованого парсера для протоколу IEC 60870-5-104, його необхідно створити вручну. Стандартним способом додавання парсера до Bro є використання генератора парсерів BinPAC. BinPAC генерує C++ код із високорівневого опису протоколу.

Окрім BinPAC, існує його розширена версія – Spicy (раніше відома як BinPAC++). На відміну від BinPAC, який описує лише синтаксис, Spicy підтримує семантичні конструкції, що робить його значно гнучкішим. Ще одна ключова відмінність полягає в тому, що Spicy компілює в інструкції для абстрактної машини HILTI – проміжного рівня, що дозволяє ефективно реалізовувати складну логіку парсингу.

Таким чином, щоб інтегрувати підтримку протоколу IEC 60870-5-104 у Bro, потрібно або створити парсер за допомогою BinPAC, або скористатися сучаснішою альтернативою – Spicy.

2.2 Особливості виявлення аномалій

Багато полів і характеристик протоколу можуть бути використані для виявлення аномалій. Однак деякі з цих полів присутні лише в частині можливих інструкцій, що містяться в протоколі IEC 60870-5-104. Щоб створити більш універсальний детектор аномалій, було обрано іншу ознаку, яка є спільною для всіх пакетів.

Оскільки SCADA-системи керують фізичними процесами, час надходження інструкцій має вирішальне значення. Реальні вимоги до роботи системи передбачають необхідність виконання інструкцій у більш передбачуваному та регулярному ритмі, ніж у звичайних IT-системах. Тому для виявлення аномалій було обрано саме час надходження пакетів.

Передбачається, що різницю в часі між подібними пакетами можна апроксимувати нормально розподіленою величиною. Щоб мати змогу порівнювати ці значення, було обрано інтервал часу між подібними пакетами, наприклад, між пакетами IEC 60870-5-104 з однаковою інструкцією.

На рисунку 2.1 показано кілька подібних пакетів і проміжки часу між ними у ідеальному випадку. Середній інтервал між пакетами позначено як Δt. На рисунку 2.2 показано випадок, коли один із пакетів був затриманий з невідомої причини. Якщо ця затримка перевищує три стандартних відхилення від середнього значення нормального інтервалу, то детектор аномалій позначить цей пакет як аномальний.

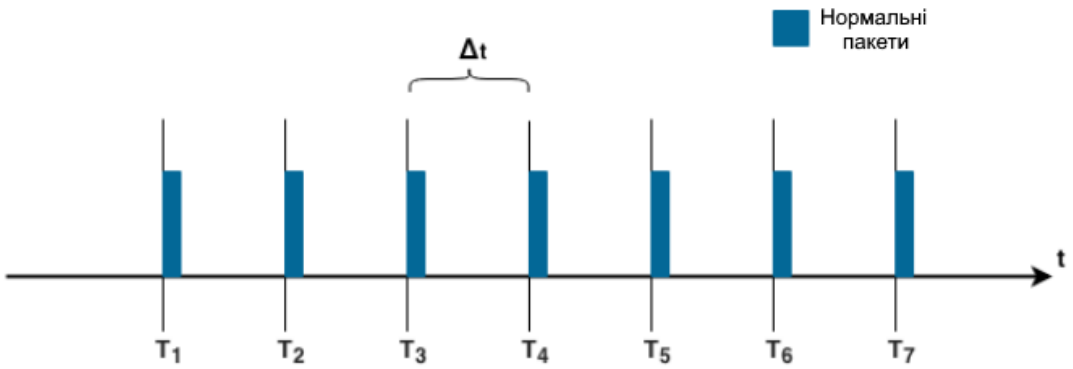

Рисунок 2.1 – Діаграма, що показує нормальний час прибуття подібних пакетів

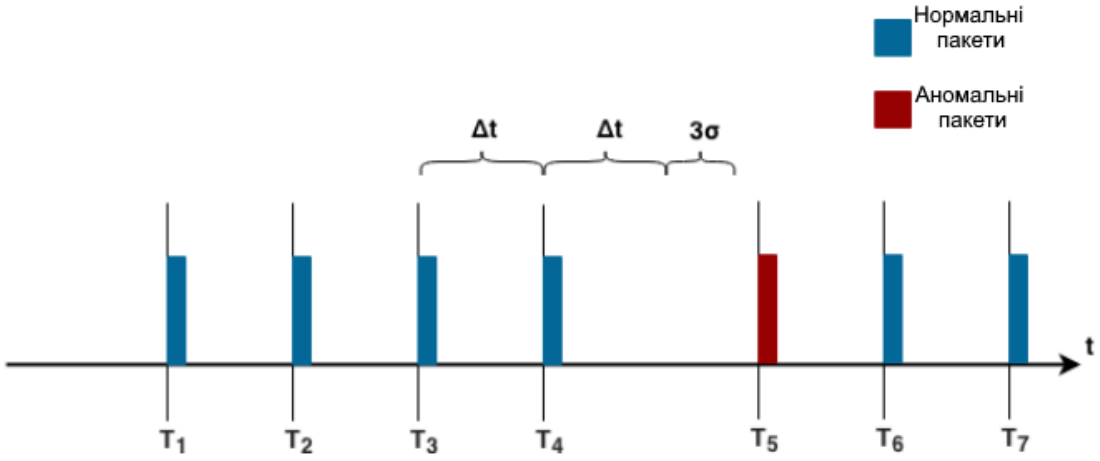

Рисунок 2.2 – Діаграма, яка показує, що пакет надійшов пізніше, ніж очікувалося

Щоб створити самонавчальну систему на основі Snort або Suricata, необхідно постійно генерувати правила вручну або за допомогою окремого інструменту чи скрипта, що знаходиться поза межами самого IDS. Такий підхід може бути застосований для реалізації «білих списків» (whitelisting), однак не дозволяє реалізувати складніші механізми виявлення, зокрема виявлення аномалій у часі надходження пакетів.

У зв’язку з цим було прийнято рішення використовувати Bro (тепер відомий як Zeek) як основну платформу для розробки детектора аномалій. Як автоматичне створення «білих списків», так і виявлення аномалій можуть бути реалізовані безпосередньо у скриптовій мові Bro, що значно спрощує інтеграцію та дозволяє побудувати самодостатню систему.

Для реалізації парсера було обрано генератор Spicy, який є розширеною версією BinPAC. Spicy забезпечує більш високий рівень абстракції та підтримує семантичні конструкції, що спрощує створення парсерів. Основна причина вибору Spicy – простота розробки порівняно з BinPAC.

Недоліком використання Spicy є обмежена документація, оскільки проєкт перебуває ще на ранньому етапі розвитку. Також у документації зазначено, що Spicy та його проміжна віртуальна машина HILTI поки що не готові до промислового використання.

Водночас Spicy та HILTI мають великий потенціал, адже вони не обмежуються використанням лише з Bro. Це відкриває можливість використання розробленого парсера IEC 60870-5-104 у майбутньому і в інших системах, таких як міжмережеві екрани, аналізатори трафіку тощо.

3 РОЗРОБКА ТА РЕАЛІЗАЦІЯ

3.1 Проектування системи виявлення вторгнень

У цьому розділі описано проєктування системи виявлення вторгнень, а також представлено її реалізацію. Робота системи виявлення вторгнень поділяється на дві окремі фази:

- фаза навчання (learning phase) – під час якої створюється модель нормальної поведінки системи;
- фаза виявлення (detection phase) – під час якої система вже працює у звичайному режимі та виявляє потенційні аномалії.

Принцип роботи базується на самонавчальній архітектурі, де дані з мережі спочатку збираються і аналізуються з метою формування еталонної моделі «нормального» трафіку. Надалі всі події порівнюються з цією моделлю, і при виявленні значних відхилень система сигналізує про можливе вторгнення.

Спрощене представлення етапів роботи системи наведено на рисунку 3.1, який ілюструє загальну послідовність процесу – від збору даних до виявлення порушень.

Огляд системи представлено на рисунку 3.2.

Парсери протоколів у Bro реалізуються у вигляді плагінів. Проєкт Spicy розробив плагін, який дозволяє прозоро інтегрувати Spicy-парсери з Bro. У результаті Bro обробляє парсери протоколів, написані на Spicy, так само, як і власні вбудовані парсери.

Вхідними даними цього плагіна є події Bro (Bro events). Саме ці події використовуються в системі виявлення аномалій для отримання інформації з пакетів протоколу. Такий підхід дозволяє коду для виявлення аномалій читати поля з пакетів, які були б недоступними без повноцінного парсера — наприклад, код інструкції (instruction code), що використовується в пакеті.

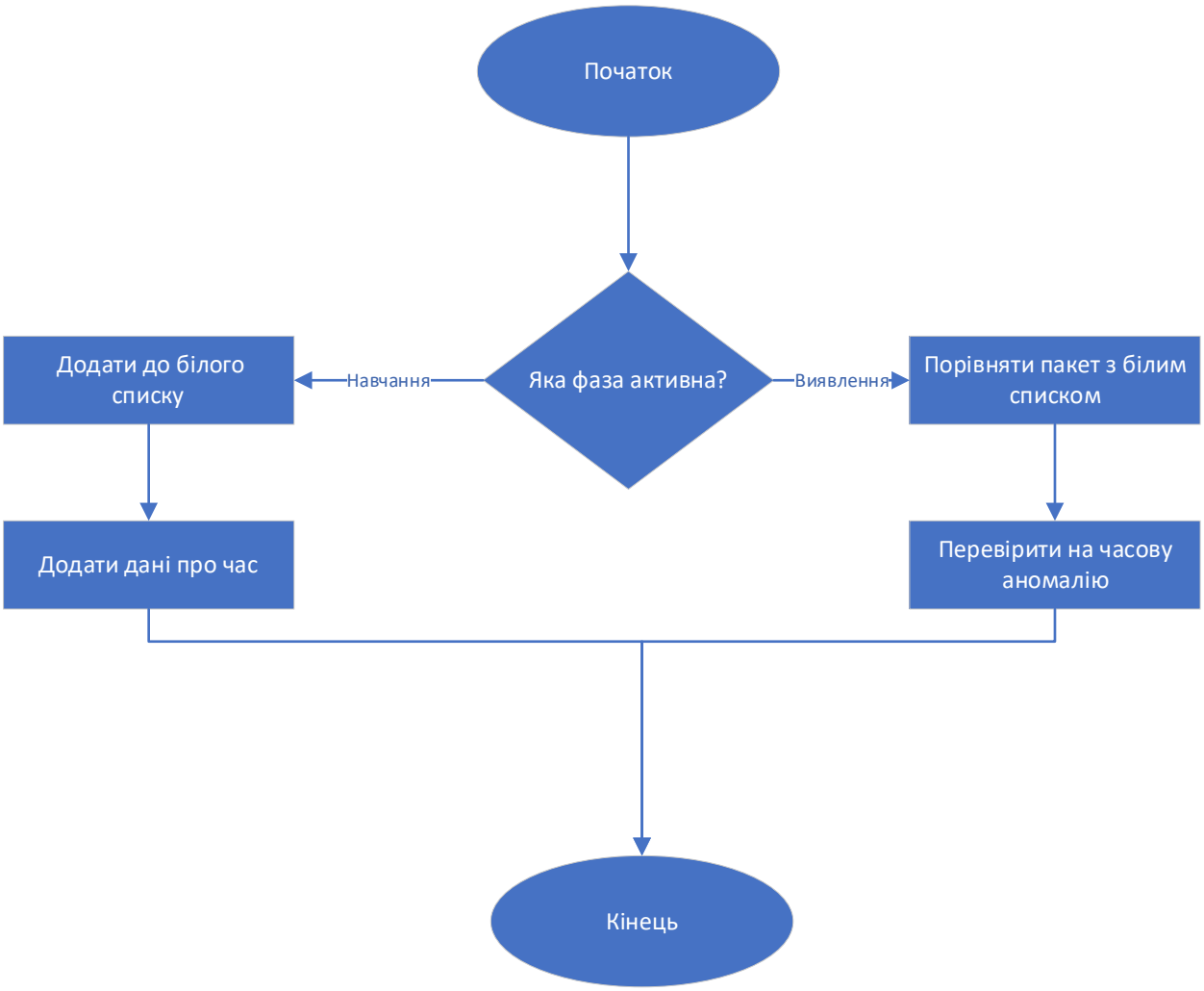

Рисунок 3.1 – Блок-схема обробки новоприбулого пакета

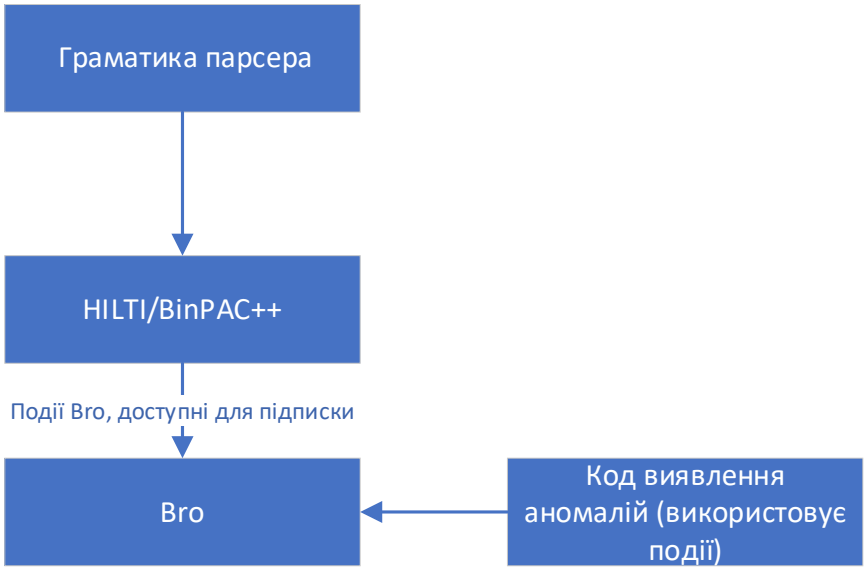

Рисунок 3.2 – Огляд компонентів системи

3.2 Архітектура парсеру

Парсер протоколу IEC 60870-5-104 було розроблено спеціально для цього проєкту та реалізовано за допомогою Spicy. Парсер має ієрархічну структуру, де APCI (Application Protocol Control Information) виступає як найвищий рівень.

Блок APCI містить інформацію про довжину інструкції та контрольне поле. Перші два біти контрольного поля визначають, який формат інструкції буде використано. Обраний формат інструкції задає спосіб інтерпретації решти контрольного поля.

Якщо використовується наглядовий формат (S) або ненумерований формат (U), то контрольне поле є останнім елементом у пакеті. Приклад пакета з форматом керування S або U наведено на рисунку 3.3.

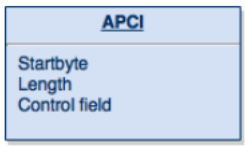

Рисунок 3.3 – Вміст блоку APCI

Однак якщо використовується інформаційний формат (I), то блок міститиме ASDU (Application Service Data Unit). Загальний вигляд структури даних, що використовується для об’єкта нормалізованої вимірювальної інформації, наведено на рисунку 3.4.

Блок ASDU включає інформацію про:

- тип інструкції;
- причину передачі (cause of transmission);
- спільну адресу переданих даних (common address).

Залежно від значення в ідентифікаторі одиниці даних (data unit identifier), ASDU може містити одну одиницю даних або список одиниць даних одного типу. Кожен блок на рисунках 3.3 і 3.4 представляє окремий

модуль Spicy unit. Кожну одиницю даних у складі ASDU реалізовано окремо як Spicy-модуль, що утворює ієрархічну структуру з кількох менших модулів.

Перевагою такої архітектури є те, що при використанні форматів керування U або S, модуль ASDU просто не додається. Для представлення різних інструкцій формату I змінюється лише тип інформаційного об’єкта, що включається.

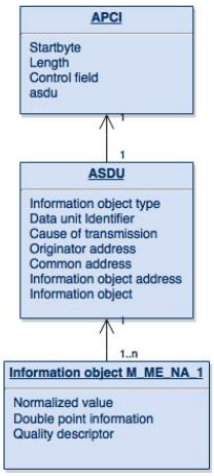

Рисунок 3.4 – Ієрархія інформаційного поля управління, зокрема нормованого об'єкта

3.3 Етап навчання

Першим кроком у виявленні загрози є вивчення нормальної поведінки системи. Протягом цього періоду здійснюється аналіз мережевого трафіку та формування моделі на основі зібраних даних. Передбачається, що під час фази навчання жодних атак не відбувається.

Модель, що використовується у цьому проєкті, складається з двох основних частин:

- білий список (whitelist) – насправді складається з кількох окремих списків, кожен із яких відповідає за різні аспекти комунікації;
- таблиця часових характеристик – містить інформацію про різницю у часі прибуття подібних пакетів.

На рисунку 3.5 показано, як обробляється щойно отриманий пакет у фазі навчання:

- якщо пакет не відповідає жодному запису у білому списку, то створюється новий запис і додається до відповідного списку;

- якщо пакет є ARP або IEC 60870-5-104, виконується обчислення часових характеристик (інтервалів між подібними пакетами).

Після цього система готова до обробки наступного пакета.

Цей підхід дозволяє поступово сформувати модель нормальної поведінки без втручання людини.

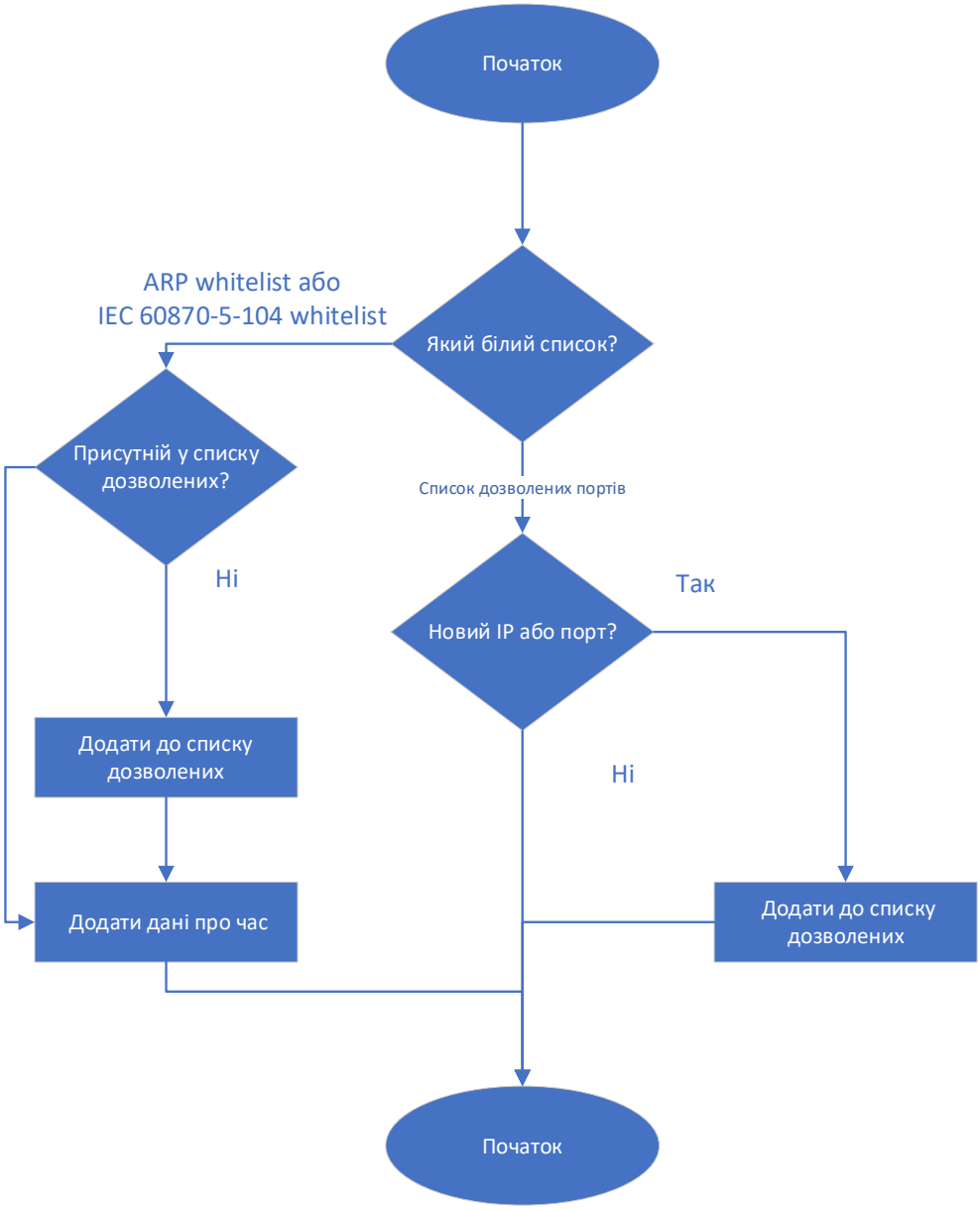

Рисунок 3.5 – Блок-схема для етапу навчання

Реалізований білий список (whitelist) складається з кількох таблиць, де кожна таблиця відповідає за певний аспект комунікації. Ці списки наповнюються під час фази навчання, формуючи структури, що зберігають інформацію про те, які пари хостів взаємодіяли між собою та які протоколи при цьому використовувались.

У системі реалізовано три окремі whitelist-и, кожен із яких створений у вигляді таблиць у Bro. Першим є whitelist ARP, який представляє таблицю, що зіставляє MAC-адресу хоста з відповідною IP-адресою.

На рисунку 3.6 показано, як таблиця ARP індексується та як виглядає запис у білому списку.

Обробка ARP-пакета відбувається за таким принципом:

- якщо MAC-адреса є новою, вона додається до таблиці разом із відповідною IP-адресою;
- якщо MAC-адреса вже відома, але IP-адреса відрізняється від тієї, що вже в таблиці, тоді IP-адреса оновлюється новим значенням.

Цей підхід дозволяє підтримувати актуальну відповідність MAC- і IP-адрес у мережі, що є критично важливим для виявлення потенційно підроблених ARP-повідомлень.

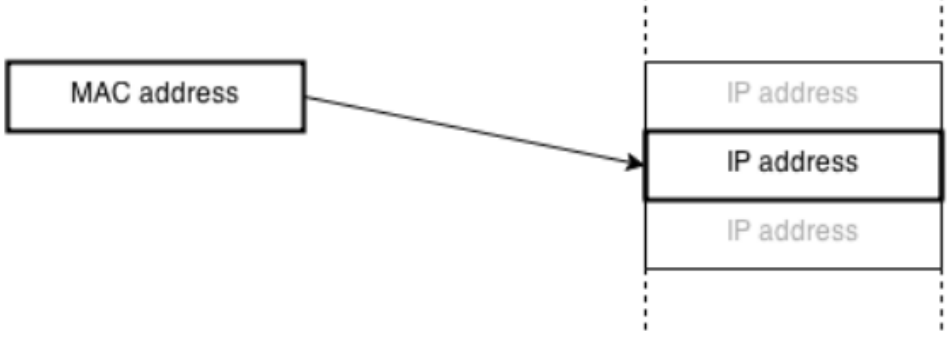

Рисунок 3.6 – Структура білого списку ARP

Список дозволених портів (port whitelist) є таблицею, індексованою за кортежем IP-адрес ініціатора та відповідача. Елемент таблиці містить два списки, в яких зберігаються порти, що використовувалися з кожного боку під

час комунікації. На рисунку 3.7 показано, як індексується список дозволів і яка структура запису в таблиці. Якщо кортеж IP-адрес відсутній у таблиці, він буде доданий разом із портами, що використовуються. Якщо запис уже існує, списки будуть витягнуті. Якщо порт ініціатора відсутній у відповідному списку – він буде доданий. Якщо порт відповідача відсутній у його списку – він також буде доданий.

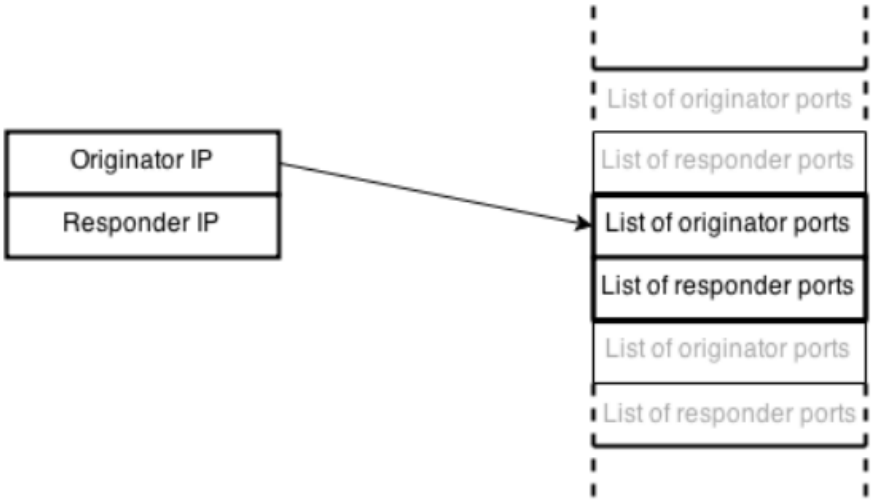

Рисунок 3.7 – Представлення білого списку портів

Список дозволів на основі часу (timing whitelist) використовує кортеж для індексації таблиці часу (timing table). Якщо будь-яке з його значень не відповідає жодному запису в таблиці, буде створено новий запис і додано до таблиці часу. Саме цей список дозволів здатен повідомляти про зміни в трафіку IEC 60870-5-104. Таблиця часу також включає інформацію про затримки в трафіку ARP.

На рисунку 3.8 показано структуру індексного кортежу та запису в таблиці. Поле Mode може мати одне з чотирьох значень: ARP, I, S або U. Значення I, S і U використовуються для різних форматів керування протоколу IEC 60870-5-104, тоді як ARP застосовується для зберігання часових даних щодо ARP-трафіку. Поле Instruction є необов’язковим, оскільки воно необхідне лише для режиму I.

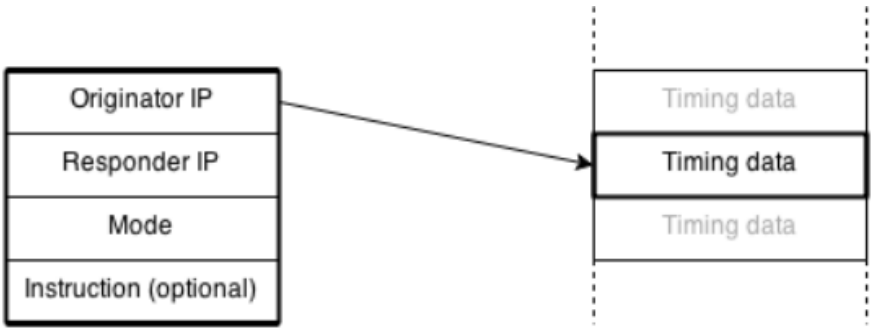

Рисунок 3.8 – Представлення білого списку IEC 60870-5-104

Запис у списку дозволених (whitelist) для протоколу IEC 60870-5-104 містить дані, що стосуються часових характеристик (timing). Цей запис є специфічним для пари хостів, а також для точного режиму (Mode) та інструкції (Instruction), які використовуються.

Дані, що містяться в цьому записі, включають:

- мінімальна різниця часу прибуття ($\Delta t\_\min$) – найменший зафіксований інтервал між двома подібними пакетами;

- максимальна різниця часу прибуття ($\Delta t\_\max$) – найбільший зафіксований інтервал;

- середнє значення різниці часу прибуття (μ) – математичне очікування інтервалів між подібними пакетами;

- змінна для обчислення дисперсії (S_n) – проміжне значення, потрібне для розрахунку дисперсії;

- дисперсія різниці часу прибуття (σ^2) – показує ступінь розкиду інтервалів навколо середнього значення;

- час прибуття попереднього пакета (t_{n-1}) – використовується для розрахунку поточної різниці у часі;

- кількість проаналізованих пакетів (n) – загальна кількість подібних пакетів, використаних для обчислення статистики.

Ці параметри дозволяють IDS-системі аналізувати регулярність комунікацій і виявляти аномалії, якщо час прибуття нових пакетів значно відрізняється від очікуваного.

Щоб уникнути зберігання всіх пакетів у пам’яті, всі обчислення

виконуються інкрементно для кожного нового пакета. Це дозволяє мати лише один запис у таблиці для кожного індексного кортежу.

Першим кроком є обчислення інтервалу часу між оброблюваним пакетом та попереднім пакетом того самого типу. Це показано у формулі (3.1). Варто зазначити, що для обчислення різниці часу потрібні принаймні два пакети, тому ця формула застосовується лише за умови n ≥ 2.

$$\Delta t_n = t_n - t_{n-1}, \quad для\ n \geq 2. \tag{3.1}$$

Значення Δt_n (різниця часу між поточним і попереднім пакетом) далі використовується для обчислення мінімального та максимального значення серед усіх отриманих пакетів, як показано у формулах (3.2) та (3.3):

$$\Delta tmin_n = \min(\Delta t_n, \Delta tmin_{n-1}), \tag{3.2}$$

$$\Delta tmax_n = \max(\Delta t_n, \Delta tmax_{n-1}). \tag{3.3}$$

Середнє значення обчислюється інкрементно, щоб уникнути збереження всіх значень і не витрачати пам’ять. Інкрементні формули наведені у рівняннях (3.4) та (3.5):

$$\mu_2 = t_2 - t_1, \tag{3.4}$$

$$\mu_n = \mu_{n-1} + \frac{\Delta t_n - \mu_{n-1}}{n-1}, \quad для\ n > 2. \tag{3.5}$$

Змінна Sn обчислюється з використанням середнього значення, як показано у формулах (3.6) та (3.7):

$$S_2 = 0, \tag{3.6}$$

$$S_n = S_{n-1} + (\Delta t_n - \mu_{n-1})(\Delta t_n - \mu_n), \quad для\, n > 2\,. \tag{3.7}$$

Нарешті, дисперсія обчислюється відповідно до формул (3.8) та (3.9):

$$\sigma_2^2 = 0\,, \tag{3.8}$$

$$\sigma_n^2 = \frac{S_n}{n-2}, \quad для\ n > 2\,. \tag{3.9}$$

3.4 Етап виявлення

У цьому розділі детально пояснюються дві основні частини, що беруть участь у фазі виявлення. Коли навчальна фаза завершена, модель системи вважається сформованою. Наступною є фаза виявлення – це основний робочий стан системи, під час якого створюються сповіщення про аномалії. Першим кроком у цьому процесі є перевірка інформації з пакета на відповідність відповідному білому списку. Білий список обирається залежно від використовуваного протоколу. На рисунку 3.9 показано, як обробляється новоприбулий пакет під час фази виявлення. Якщо пакет не відповідає нормальній поведінці, створюється сповіщення.

Під час фази виявлення всі ARP-пакети порівнюються з ARP-білим списком. Якщо ARP-пакет містить MAC-адресу, якої немає в таблиці, буде створене сповіщення. Також буде згенеровано сповіщення, якщо MAC-адреса присутня, але IP-адреса в записі не відповідає IP-адресі в оброблюваному пакеті. Це дозволяє виявляти хости, які намагаються підмінити свої адреси, а також нові, потенційно небезпечні, пристрої, що підключаються до мережі.

Білий список портів використовується для виявлення аномальних потоків між відомими хостами, які вже були додані до списку. Сповіщення буде створено, якщо пара хостів відсутня в таблиці. Також буде створене

сповіщення, якщо пара хостів уже існує, але використовується новий порт. Якщо хост ініціює з'єднання на новий порт, це може свідчити про те, що його було скомпрометовано й він використовує сервіси, які зазвичай не використовує.

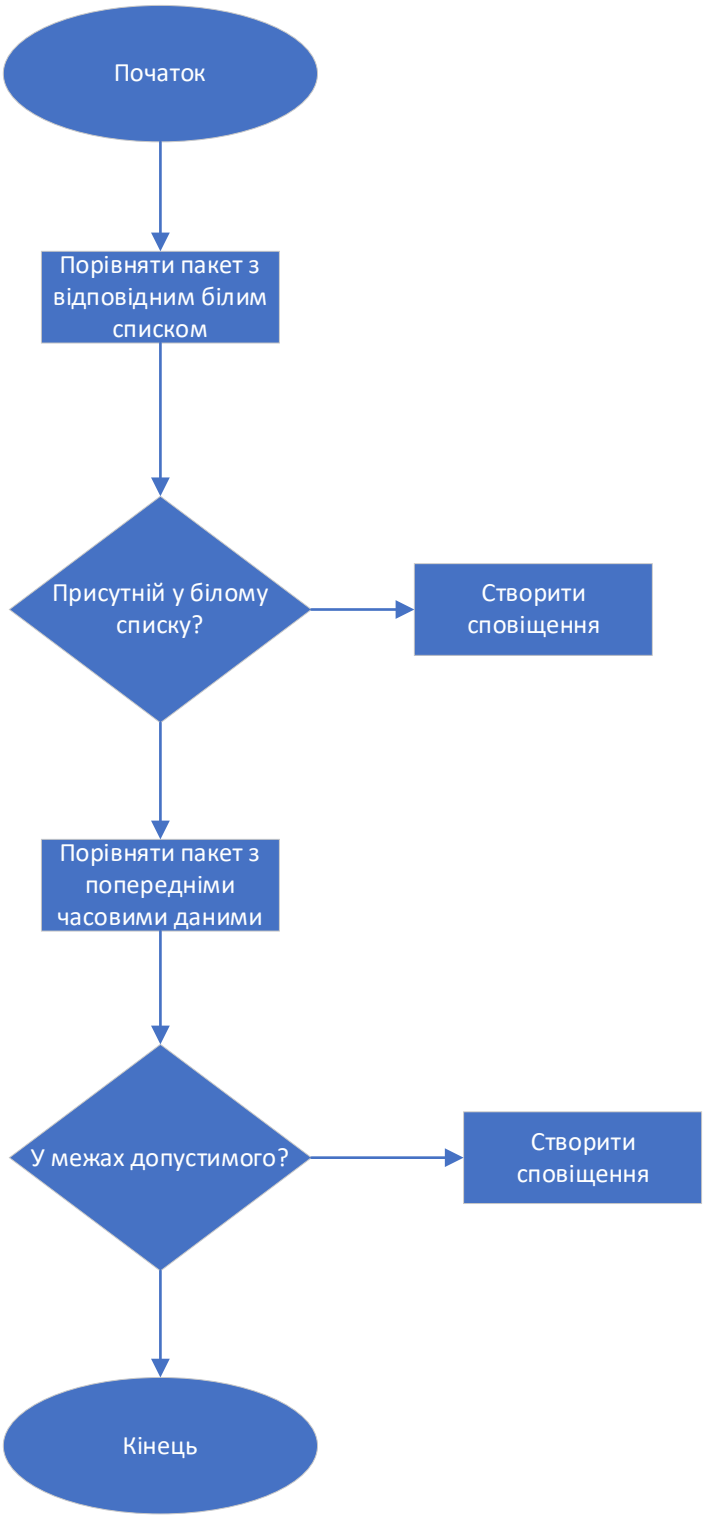

Рисунок 3.9 – Блок-схема етапу виявлення

Третій білий список – це таблиця таймінгу (timing table), яка індексує дані про час прибуття пакетів. Якщо не існує даних про таймінг для конкретної пари хостів із певним режимом (Mode) та інструкцією (Instruction), буде створено сповіщення. Якщо надсилається інструкція, яка раніше не з'являлась, це також викличе сповіщення — навіть якщо ця інструкція дозволена за стандартом IEC 60870-5-104. Інструкції, які додаються до білого списку, є специфічними для цього протоколу. Малоймовірно, що пара хостів буде використовувати всі можливі інструкції з IEC 60870-5-104. Так само малоймовірно, що всі пари хостів використовуватимуть однаковий набір інструкцій. Тому кожна пара хостів матиме власний дозволений набір інструкцій.

Тільки ті пакети ARP і IEC 60870-5-104, які не були визнані аномальними за білими списками, передаються до модуля виявлення аномалій на основі часу.

Кожен ARP- або IEC 60870-5-104-пакет, який проходить перевірку білим списком, порівнюється з записом у таблиці таймінгу для цього конкретного типу пакету. Наприклад, пакет між хостами A і B у режимі I з інструкцією 100 буде порівнюватися лише з іншими подібними пакетами (від тієї ж пари хостів, з тим же режимом та інструкцією). Поле інструкції є необов'язковим, оскільки тільки пакети з полем керування, встановленим у режим I, мають різні інструкції.

Першим кроком є обчислення інтервалу часу між поточним пакетом і попереднім аналогічним пакетом. Цей інтервал обчислюється за формулою (3.1).

Наступним кроком є порівняння інтервалу часу з мінімальним і максимальним значенням, які були розраховані під час фази навчання. Якщо інтервал менший за мінімальне значення або більший за максимальне, буде згенеровано сповіщення.

Останній етап виявлення аномалій – перевірка, чи знаходиться інтервал у межах трьох стандартних відхилень від середнього значення. Формула для

цього подана в рівнянні (3.10). У подальших формулах l використовуватиметься для позначення пакетів, проаналізованих у фазі навчання, а n – для всіх проаналізованих пакетів.

$$3\sigma_l > |\Delta t_n - \mu_l|. \tag{3.10}$$

Якщо інтервал виходить за межі трьох стандартних відхилень від середнього значення, він також буде порівнюватися з межами чотирьох, п’яти та шести стандартних відхилень. Це робиться для визначення рівня варіації (змінності) в мережевому трафіку.

4 ОЦІНКА РОБОТИ

4.1 Середовище оцінювання

Запис охоплює дані за цілий тиждень, зібрані у віртуальній SCADA-лабораторії. Тестове середовище складалося з чотирьох віддалених термінальних пристроїв (RTU) моделі NETCON RTU 28-IP з номерами від 1 до 4, комутатора HP ProCurve 1800-8G, Raspberry Pi моделі B+ та комп'ютера з програмним забезпеченням Zenon. Zenon використовується для моніторингу та управління SCADA-системами. Схема налаштування представлена на рисунку 4.1.

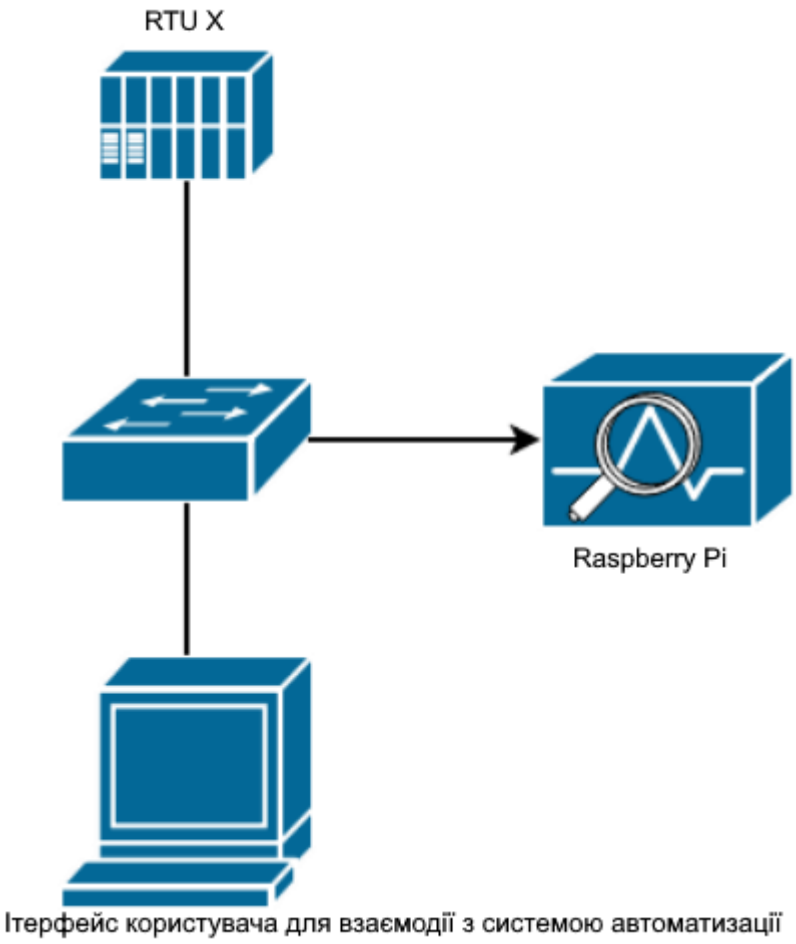

Рисунок 4.1 – Тестове середовище

Raspberry Pi здійснював запис усього мережевого трафіку між RTU та HMI. Запис здійснювався за допомогою Tcpdump через дзеркальний порт на комутаторі. Отриманий запис було використано як вхідні дані для системи виявлення вторгнень.

Через певні проблеми, які докладніше розглядаються в тестування було розділене на чотири окремі потоки – кожен містив трафік до та від одного RTU. Проте лише два з цих потоків вдалося запустити без збоїв із парсером IEC 60870-5-104, згенерованим за допомогою Spicy. Усі подальші тренування та тестування проводилися лише для цих двох RTU – а саме RTU 1 та RTU 4.

Усі тести були сформовані на основі оригінального тижневого запису. Навчальні дані включали перші 24 години запису, якщо не зазначено інше.. Символ X після мітки RTU позначає номер RTU – або 1, або 4.

Оцінювання системи виявлення вторгнень (IDS) проводиться шляхом класифікації проаналізованих пакетів у чотири категорії.

Пакет позначається як позитивний, якщо IDS вважає його аномальним. Якщо IDS класифікує пакет як позитивний, і цей пакет справді є шкідливим, то це називається істинно позитивним (TP). Істинно негативний (TN) виникає, коли IDS позначає пакет як нормальний, і він дійсно є нормальним.

Хибно негативний (FN) – це коли IDS класифікує шкідливий пакет як нормальний. Хибно позитивний (FP) – це коли IDS повідомляє про загрозу, але насправді пакет є легітимним.

Ця класифікація є основою для розрахунку ключових метрик оцінки, таких як точність (precision), повнота (recall), F1-міра та інші.

4.2 Методи генерації атак

Для реалізації досліджених атак було використано низку інструментів, які є у вільному доступі в мережі Інтернет.

Wireshark – це провідний у світі аналізатор протоколів. Він був важливим інструментом для аналізу та розуміння протоколу IEC 60870-5-

104. Також Wireshark використовувався для перевірки правильності вставлення та маніпулювання атаками в записах трафіку.

Tshark – консольна версія Wireshark, яка використовує ті самі механізми обробки пакетів. Застосовувався у випадках, коли графічний інтерфейс Wireshark працював повільно через великі розміри файлів.

Tcprewrite – використовувався для видалення тегів VLAN, які додавалися комутатором, а також для зміни номерів портів в атакувальних файлах, щоб вони відповідали портам реальної системи.

Mergecap – інструмент командного рядка, що входить до Wireshark. Він дозволяє об'єднувати різні записи мережевого трафіку. Наприклад, для вставлення маніпульованих пакетів атак у запис нормального трафіку. Також застосовувався для конвертації між форматами .pcapng і .pcap.

Bittwiste – консольний інструмент для зміни IP- і MAC-адрес у файлах атак, щоб вони відповідали справжньому запису з SCADA-середовища.

Nmap – утиліта для аудиту безпеки та дослідження мереж. Використовувалася для реалізації атак сканування портів. Перше сканування імітує нового зловмисника в мережі, друге — атаку з компрометованого хоста.

Arpspoof (з пакету dsniff) – використовувався для створення MITM-атаки. Він надсилає ARP-відповіді обом сторонам, змушуючи їх надсилати пакети через атакується вузол, дозволяючи зловмиснику перехоплювати та змінювати трафік.

Scapy – найважливіший інструмент, що використовується у цьому дослідженні. Це Python-бібліотека для маніпуляцій з мережевими пакетами. Незважаючи на те, що протокол IEC 60870-5-104 не підтримується напряму, Scapy дозволяє змінювати дані вручну. Він інтенсивно використовувався для реалізації атаки на основі передбачення: пакети з реального запису копіювалися та вставлялися трохи раніше з модифікованими даними – імітуючи ін'єкцію фальшивої інформації.

4.3 Оцінювання

Атаки на виявлення портів було реалізовано у двох варіантах. Перший варіант імітував атаку з боку невідомого вузла, який раніше не був присутній у мережі. Другий варіант моделював атаку з боку вже відомого вузла, який раніше здійснював зв’язок із цільовим хостом. Такий сценарій можливий, якщо цей вузол був скомпрометований зловмисником.

Результати атак наведено на рисунку 4.2. Обидва варіанти дали однакові результати – усі пакети атак було виявлено, що забезпечило 100% рівень істинно позитивних спрацювань (true positive rate) для механізму білого списку. Проте були зафіксовані хибнопозитивні спрацювання з боку модуля виявлення аномалій за часом, які збігаються з результатами 24-годинного навчального тесту.

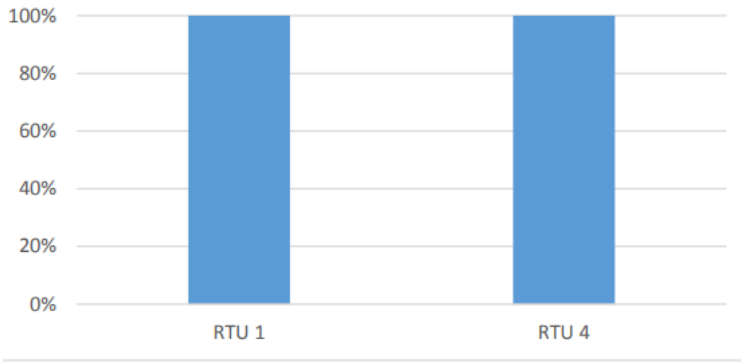

Рисунок 4.2 –Відсоток позитивних результатів сканування портів

Атаку типу «людина посередині» (Man-in-the-Middle, MITM) було реалізовано шляхом створення ARP-пакетів із зміненими адресами джерела та призначення. Надсилаючи ці пакети до HMI та RTU, зловмисник змушує обидві сторони передавати свої дані через нього. У такий спосіб атакуючий отримує повний доступ до трафіку, може читати та змінювати вміст пакетів перед їх подальшою передачею справжньому отримувачу.

Білий список за часом коректно виявляє приблизно половину пакетів атаки. Якщо пакет позначається цим модулем як аномальний, це означає, що

для цього типу пакету між конкретними вузлами ще не накопичено достатньо статистики часу прибуття. У таких випадках пакет не потрапляє до перевірки на відхилення від значень «три сигми». Лише ті пакети, що пройшли перевірку на відповідність білому списку, аналізуються механізмом виявлення аномалій на основі міжпакетного інтервалу. Саме тому лише половина атакуючих пакетів фіксується через перевищення меж «трьох сигм».

Загальна кількість істинно позитивних спрацьовувань, пов’язаних з міжпакетним інтервалом, визначається як сума істинно позитивних спрацьовувань за «три сигми» та з білого списку за часом. Таким чином, усі пакети, окрім одного, успішно виявлено на RTU 1. Останній пакет було виявлено на RTU 4 за допомогою методу «три сигми» або порівняння з мінімумом/максимумом.

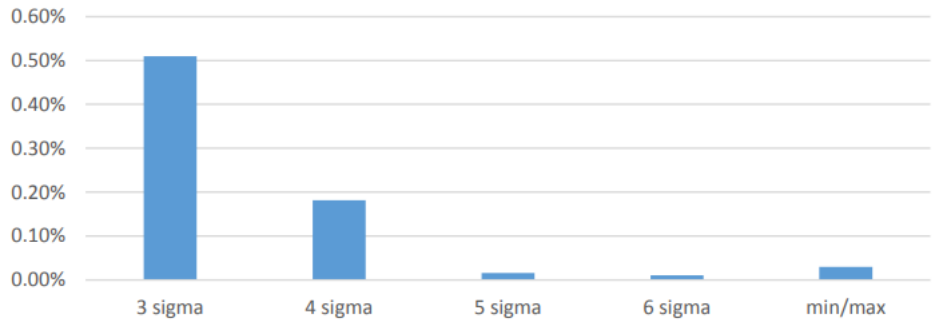

Рисунок 4.3 – Середній FPR для RTU 1 та RTU 4 атаки MITM

ARP-білий список виявив майже всі пакети атаки. Ті пакети, які не були виявлені, – це пакети, якими атакуючий відновлював безпосередній зв’язок між хостами після завершення атаки.

ВИСНОВКИ

У межах цієї кваліфікаційної роботи було реалізовано систему виявлення аномалій у трафіку протоколу IEC 60870-5-104, який широко використовується в SCADA-системах. З урахуванням специфіки таких систем, де передавання даних відбувається у передбачуваному та детермінованому режимі, основною ідеєю проєкту стало використання часових характеристик мережевого трафіку для виявлення потенційних атак.

Для реалізації підходу було обрано платформу Bro (тепер Zeek), яка дозволила побудувати повноцінну аномалійну систему виявлення завдяки підтримці сценаріїв на власній мові програмування та можливості інтеграції з власноруч створеними парсерами протоколів. Було створено спеціальний парсер для IEC 60870-5-104 за допомогою генератора Spicy, що забезпечив гнучкий аналіз трафіку на прикладному рівні.

Система працює у двох режимах: навчальному та режимі виявлення. Під час навчання формується модель нормальної поведінки системи – будуються «білі списки» допустимих взаємодій хостів, портів та інтервалів між пакетами. У режимі виявлення усі вхідні пакети перевіряються на відповідність цим спискам, а також аналізуються за допомогою правила трьох сигм для оцінки відхилення часового інтервалу.

Під час тестування були успішно виявлені три типи атак: сканування портів, атака типу «людина посередині» (MITM) та атака передбачення послідовностей (prediction attack). Особливо ефективним виявився підхід на основі аналізу часу прибуття пакетів – як показали результати, майже всі пакети атаки було виявлено або за допомогою часових відхилень, або через порушення попередньо визначених шаблонів.

Отже, результати дослідження підтверджують, що побудова системи виявлення аномалій для IEC 60870-5-104 на основі поведінкового аналізу трафіку є ефективною стратегією для захисту SCADA-середовищ. Подальше

розширення моделі, зокрема включення інших характеристик трафіку, а також удосконалення алгоритмів виявлення, може ще більше підвищити надійність і точність захисту таких критичних систем.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Repository of Industrial Security Incidents (RISI). – [Електронний ресурс]. – Режим доступу: https://risidata.com

2. Clarke, Gordon, Deon Reynders, and W. Edwin. "Practical modern SCADA protocols: DNP3." IEC60870-5 and related systems, Australia (2004).

3. Nweke, L. O. (2021). A survey of specification-based intrusion detection techniques for cyber-physical systems.

4. Tokgoz, Emre. Six Sigma for Continuous Improvement in Cybersecurity: A Guide for Students and Professionals. Springer Nature, 2025.

5. Ji, R., Padha, D., Singh, Y., & Sharma, S. (2024). Review of intrusion detection system in cyber-physical system based networks: Characteristics, industrial protocols, attacks, data sets and challenges. *Transactions on Emerging Telecommunications Technologies*, *35*(9), e5029.

6. Mittal, Himanshu, et al. "A new intrusion detection method for cyber–physical system in emerging industrial IoT." Computer Communications 190 (2022): 24-35.

7. Мартовицький В. О., Шеховцов О. В., Алєйник Д. С., Пахомова Є. В. та Іванченко Д. І. «ПІДХІД ДО ВИЯВЛЕННЯ ТА КЛАСИФІКАЦІЇ РАДІОКЕРОВАНИХ МОДЕЛЕЙ ЗА ЇХ РАДІОСИГНАЛОМ» Вісник Херсонського національного технічного університету» для розміщення у № 2 (2025)