ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «**Аналіз мережевого трафіку у IoT(Інтернет-речей) з використанням Wireshark**»

на здобуття освітнього ступеня бакалавра
зі спеціальності 123 Комп'ютерна інженерія

освітньо-професійної програми Комп'ютерна інженерія

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ Юрій МОТОРІН
(підпис)

Виконав: здобувач вищої освіти групи КІД-42
Юрій МОТОРІН

Керівник: Юлія БОРОВА

Рецензент: _____

Київ 2024

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут Інформаційних технологій

Кафедра 123 Комп'ютерної інженерії

Ступінь вищої освіти бакалавр

Спеціальність 123 Комп'ютерної інженерії

Освітньо-професійна програма Комп'ютерна інженерія

ЗАТВЕРДЖУЮ
Завідувач кафедрою Комп'ютерної інженерії
_____Наталія ЛАЩЕВСЬКА

«_____» _____ 2024р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Моторіну Юрію Олександровичу

1. Тема кваліфікаційної роботи: **«Аналіз мережевого трафіку у IoT(Інтернет-речей) з використанням Wireshark»**
Керівник кваліфікаційної роботи Юлія БОРОВА
затверджені наказом Державного університету інформаційно-комунікаційних технологій від «27» лютого 2024р. № 36.

2.Строк подання кваліфікаційної роботи «30» травня 2024р.

3. Вихідні дані до кваліфікаційної роботи:
3.1 Вимоги до кваліфікаційної роботи бакалавра з актуальних завдань спеціальності.
3.2 Нормативні матеріали (стандарти).
3.3 Технічні вимоги.
3.4 Наукова-технічна література з питань, пов'язаних з темою роботи.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Вивчення та аналіз типового мережевого трафіку, що виникає у зв'язку з підключеними до Інтернету речами
2. Виявлення шаблонів споживання ресурсів, ідентифікацію вразливостей
3. Пропозиції щодо оптимізації протоколів спілкування.

5. Перелік ілюстративного матеріалу: *презентація*

6. Дата видачі завдання «27» лютого 2024р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	15.03.2024	
2	Аналіз завдання	1.04.2024	
3	Пошук підходів до вирішення завдання	15.04.2024	
4	Написання розділів роботи	1.05.2024	
5	Формування висновків	10.05.2024	
6	Оформлення пояснювальної записки	13.05.2024	
7	Підготовка демонстраційних матеріалів	30.05.2024	
8	Попередній захист роботи	13.05.2024	
9	Пред’явлення роботи в деканат	3.06.2024	

Здобувач вищої освіти _____ Юрій МОТОРІН

(підпис)

Керівник кваліфікаційної роботи _____ Юлія БОРОВА

(підпис)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: 41 стор., 22 рис., 0 табл., 30 джерел.

Мета роботи – провести глибокий аналіз мережевого трафіку в середовищі Інтернету речей (IoT) з використанням аналізатора Wireshark для забезпечення безпеки та оптимізації мереж IoT.

Об’єкт дослідження – мережевий трафік у системах Інтернету речей (IoT).

Предмет дослідження – аналіз мережевого трафіку в IoT з використанням інструменту Wireshark для виявлення особливостей, патернів та можливих загроз.

Короткий зміст роботи: У роботі було розглянуто концепцію Інтернету речей (IoT) та її еволюцію - від зародження ідеї до сучасного стану розвитку. Окрему увагу приділено вимогам безпеки для систем IoT і значенню аналізу мережевого трафіку для забезпечення їх захисту.

Розглянуто протоколи, що використовуються в IoT, типи трафіку та методи його аналізу. Порівняно різноманітні популярні інструменти для моніторингу трафіку.

У роботі детально описано процес встановлення та налаштування аналізатора Wireshark, методику захоплення даних. На практиці проведено захоплення та аналіз різних типів пакетів: broadcast, ICMP, DNS, TCP. Окремо досліджено специфічні для IoT протоколи: MQTT, CoAP, LoRaWAN, ZigBee.

КЛЮЧОВІ СЛОВА: аналіз трафіку, Wireshark, IoT, інтернет речей

ЗМІСТ

ВСТУП

За 30 років свого існування Інтернет став невід'ємною частиною людського життя, принісши як значні переваги, так і низку викликів. Безсумнівними плюсами стали зручність комунікації, розвиток електронної комерції та покращення доступу до інформації. Однак серед негативних наслідків - ризики поширення дезінформації, формування залежності від цифрових технологій та загрози приватності користувачів.

Первісно створений як віртуальний простір, Інтернет нині з'єднує мільярди пристроїв у фізичному світі, формуючи концепцію Інтернету речей (IoT). У сучасній реальності IoT стрімко розвивається, ставши невід'ємним елементом повсякденного життя. Ця технологія об'єднує широкий спектр гаджетів - від смартфонів і розумних годинників до датчиків і промислових систем, які підключені до глобальної мережі для обміну даними. Така інтеграція відкриває значний потенціал для трансформації багатьох сфер - від охорони здоров'я й транспорту до виробництва та енергетики.

Метою цієї кваліфікаційної роботи є провести аналіз мережевого трафіку в середовищі Інтернету речей (IoT) з використанням аналізатора Wireshark для забезпечення безпеки та оптимізації мережі. Для досягнення цієї мети будуть вирішені наступні завдання:

1. Вивчення основних концепцій IoT, архітектури IoT-систем, протоколів та стандартів IoT.
2. Ознайомлення з методами аналізу мережевого трафіку в IoT-системах.
3. Оволодіння навичками роботи з Wireshark для аналізу мережевого трафіку IoT-систем.
4. Проведення практичного аналізу мережевого трафіку з використанням Wireshark.
5. Розробка рекомендацій щодо забезпечення кібербезпеки IoT-систем на основі результатів аналізу мережевого трафіку.

1 ТЕОРЕТИЧНІ ВІДОМОСТІ

1.1 Визначення IoT та його ключових характеристик

Після чотирьох десятиліть з моменту зародження Інтернету за допомогою ARPANET, термін "Інтернет" відноситься до широкої категорії додатків та протоколів, побудованих на основі складних та взаємопов'язаних комп'ютерних мереж, які обслуговують мільярди користувачів по всьому світу у режимі 24/7. Фактично, ми перебуваємо на початку нової ери, де всезагальне спілкування та зв'язок більше не є мрією або викликом. Пізніше увага змінилася на безшовну інтеграцію людей та пристроїв для злиття фізичного світу з віртуальним, що призвело до створення так званої утопії Інтернету речей (IoT).

Докладне вивчення цього явища розкриває дві важливі складові IoT: "Інтернет" та "Речі". Здається, що кожен об'єкт, здатний підключатися до Інтернету, потрапляє до категорії "Речі", але цей термін використовується для охоплення більш загального набору сутностей, включаючи розумні пристрої, сенсори, людей та будь-який інший об'єкт, який усвідомлює свій контекст і може спілкуватися з іншими сутностями, забезпечуючи доступність в будь-який час і в будь-якому місці. Це означає, що об'єкти повинні бути доступні без обмежень у часі або місці.

Всезагальний зв'язок є ключовою вимогою IoT, і для її задоволення додатки повинні підтримувати різноманітний набір пристроїв та протоколів зв'язку, від мікросенсорів, здатних виявляти та повідомляти необхідний фактор, до потужних серверів зворотного кінця, які використовуються для аналізу даних та отримання знань. Це також передбачає інтеграцію мобільних пристроїв, краєвих пристроїв, таких як маршрутизатори та розумні хаби, та людей як контролерів[1].

1.2 Еволюція визначення інтернету речей

1.2.1 Появлення IoT

Кевін Ештон вважається автором терміну "Інтернет речей", який вперше був використаний під час презентації 1999 року з управління ланцюгом поставок. Він вважає, що аспекти "речей" у способі, яким ми взаємодіємо та живемо в фізичному світі навколо нас, потребують серйозного переосмислення через досягнення в області обчислювальної техніки, Інтернету та швидкості генерації даних від розумних пристроїв. Тоді він був виконавчим директором Центру Auto-ID MIT, де вніс внесок у розширення застосувань RFID в більш широкі сфери, що стало основою для сучасної візії IoT[1].

1.2.2 Інтернет речей всього

З того часу було запропоновано багато визначень для IoT, включаючи визначення, яке в основному зосереджене на вимогах до зв'язку та сенсорних вимог для сутностей, що беруть участь у типових середовищах IoT. Хоча ці визначення відображають основні вимоги IoT, нові визначення IoT надають більше значення потребі у всезагальних та автономних мережах об'єктів, де ідентифікація та інтеграція сервісів мають важливу та необхідну роль. Наприклад, Cisco використовує термін "Інтернет усього" для посилання на людей, речі та місця, які можуть надавати свої сервіси іншим сутностям[1].

1.2.3 Індустріальне IoT

Також відоме як Промисловий Інтернет , Промисловий IoT (IoT) - це ще одна форма застосування IoT, яка користується популярністю серед великих високотехнологічних компаній. Те, що машини можуть виконувати певні завдання, такі як збір даних та комунікація, точніше, ніж люди, значно підвищило прийняття

IIoT. Комунікація від машини до машини (M2M), аналіз великих даних та методи навчання машин - це основні будівельні блоки, коли йдеться про визначення IIoT[1].

1.2.4 Інтелект В IoT

Ще одна характеристика IoT, яка виокремлюється в останніх визначеннях, - це "розумність". Це відрізняє IoT від схожих концепцій, таких як мережі сенсорів, і може бути подальшим розділеним на "розумність об'єктів" і "розумність мережі". Розумна мережа - це комунікаційна інфраструктура, яка характеризується наступними функціональними можливостями:

а) стандартизація та відкритість використовуваних комунікаційних стандартів, від шарів, які взаємодіють з фізичним світом (тобто мітки та сенсори), до комунікаційних шарів між вузлами та з Інтернетом;

б) адресованість об'єктів (прямі IP-адреси) та багатофункціональність (тобто можливість, що мережа, побудована для одного застосування (наприклад, моніторинг дорожнього руху), буде доступна для інших цілей (наприклад, моніторинг забруднення навколишнього середовища або безпеки дорожнього руху)[1];

1.2.5 Оцінка ринку IoT

Крім того, визначення приділяють особливу увагу потенційному ринку IoT з швидким темпом зростання, маючи ринкову вартість у 44,0 мільярди доларів у 2011 році. Згідно з комплексним дослідженням ринку, проведеним RnRMarketResearch, яке включає поточний розмір ринку і прогнози на майбутнє, ринок IoT та M2M буде мати вартість приблизно 498,92 мільярдів доларів до 2019 року. За словами того ж дослідження, вартість ринку IoT очікується досягти 1423,09 мільярдів доларів до 2020 року, причому Інтернет речей нано (IoNT) візьме ключову роль на майбутньому ринку та матиме вартість приблизно 9,69

мільярдів доларів до 2020 року.[1] Окрім всіх цих фантастичних та оптимістичних можливостей, для поточного IoT для досягнення передбаченого ринку потрібні різноманітні інновації та прогрес в різних галузях. Крім того, співпраця та обмін інформацією між провідними компаніями в галузі IoT, такими як Microsoft, IBM, Google, Samsung, Cisco, Intel, ARM, Fujitsu, Ecobee Inc., нарівні з меншими підприємствами та стартапами, сприятимуть прийняттю IoT та зростанню ринку.

1.3 Безпека мережевого трафіку в Інтернеті речей

Існує три широкі категорії загроз: перехоплення, порушення та маніпулювання.

Загрози перехоплення пов'язані із захопленням системи або інформації. Загрози порушення пов'язані із відмовою в обслуговуванні, знищенням та порушенням роботи системи. Загрози маніпулювання пов'язані із маніпулюванням даними, ідентифікацією, часовими рядами даних тощо.

Найпростішим пасивним видом загроз в Інтернеті речей (IoT) є підслуховування або моніторинг передач з метою отримання інформації, що передається. Це також називається атаками перехоплення. Атаки перехоплення призначені для отримання контролю над фізичними або логічними системами або для отримання доступу до інформації чи даних із цих систем.

Поширеність і фізичне розподілення об'єктів та систем IoT надають зловмисникам чудову можливість отримати контроль над цими системами. Розповсюдження розумних об'єктів, датчиків та систем призводить до самореклами, маяків та мережевого зв'язку, що надає зловмисникам більше можливостей для перехоплення або втручання в передачу інформації в цьому середовищі. Більше того, частота передачі даних, моделі та формати даних допомагають зловмисникам у криптоаналізі.

Ось деякі з відомих активних загроз:

— маскарад: сутність видає себе за іншу сутність. Це включає маскування інших об'єктів, датчиків і користувачів;

— посередник (Man-in-the-middle): коли зловмисник приховано пересилає та можливо змінює зв'язок між двома сутностями, які вважають, що вони безпосередньо спілкуються один з одним;

— атаки повторного відтворення: коли зловмисник надсилає деякі старі (аутентичні) повідомлення одержувачу. У випадку каналу мовлення або маяка, доступ до попередньо переданих даних є легким;

— відмова в обслуговуванні (DoS-атаки): коли сутність не може виконувати свою належну функцію або діє таким чином, що перешкоджає іншим сутностям виконувати свої належні функції;

Активні загрози, такі як маскування, атаки повторного відтворення, DoS-атаки, загалом є порівняно легкими в середовищі IoT. Одним із прикладів є реалізація клонованих маяків із ненадійного джерела. Маяки – це маленькі бездротові пристрої, які постійно передають простий радіосигнал, що говорить: "Я тут, це мій ідентифікатор". У більшості випадків сигнал сприймається смартфонами поблизу за допомогою технології Bluetooth Low Energy (BLE). Коли мобільний пристрій виявляє сигнал маяка, він зчитує ідентифікаційний номер (ідентифікатор) маяка, розраховує відстань до маяка та на основі цих даних запускає дію в сумісному з маяком мобільному додатку[1].

В літературі [1] загрози IoT перераховуються як клонування розумних об'єктів ненадійними виробниками, підробка/заміна пристроїв IoT третіми сторонами, заміна шкідливого програмного забезпечення та атаки на відносно незахищені пристрої шляхом підслуховування або викрадення облікових даних або властивостей безпеки, на додаток до стандартних векторів загроз, таких як атаки типу "людина посередині" та DoS-атаки. Вимоги до безпеки та конфіденційності визначаються характером атак у середовищі IoT.

1.3.1 Вимоги безпеки Інтернету Речей

Нижче перелічені основні властивості безпеки, які необхідно реалізувати в IoT:

— конфіденційність: Передані дані повинні бути прочитані лише кінцевими точками зв'язку;

— доступність: Кінцеві точки зв'язку завжди доступні та не можуть бути зроблені недоступними;

— цілісність: Отримані дані не піддаються змінам під час передачі, а їх точність і повнота гарантуються протягом усього життєвого циклу;

— аутентифікація: Відправника даних завжди можна перевірити, а одержувачів даних неможливо підробити;

— авторизація: Доступ до даних можуть отримати лише ті, кому це дозволено, і вони повинні бути недоступні для інших;

Забезпечення безпеки IoT є складним завданням, яке потребує поєднання підходів із мобільних та хмарних архітектур з промисловим управлінням, автоматизацією та фізичною безпекою. Багато вимог безпеки для IoT аналогічні вимогам для Інтернету, заснованого на протоколі IP. Технології та послуги, які використовувалися для забезпечення безпеки Інтернету, застосовуються в більшості випадків з необхідною адаптацією на кожному рівні еталонної моделі IoT.

Використання технологій IP в Інтернеті речей має ряд основних переваг, таких як єдиний і однорідний набір протоколів та перевірена архітектура безпеки. Це також спрощує механізми розробки та розгортання інноваційних послуг за рахунок розширення перевірених фреймворків на базі IP.

Однак, це призводить до явища, яке називається "збільшенням поверхні атаки". Це означає, що коли ми підключаємо раніше не підключені пристрої, що передають конфіденційні дані, розміщуємо дані в мобільній хмарі або переносимо обчислення на периферійні пристрої, неминуче з'являються нові точки входу для загроз безпеці.

Оскільки мережі розумних об'єктів та IP об'єднуються, існує висока ймовірність виникнення вразливостей безпеки через трансляцію протоколів, несумісні інфраструктури безпеки тощо. Модель безпеки підприємства традиційно базувалася на двох головних принципах:

1. Найкращих у своєму класі програмах та пристроях: рішення для брандмауерів, мережевої безпеки, безпеки даних, безпеки контенту тощо.
2. Периметрі, тобто організації захищали кінцевий пристрій та сервер, і реагували на виявлені вторгнення або загрози, такі як віруси чи атаки DoS.

В контексті Інтернету речей механізми безпеки, що базуються на периметрі, сьогодні мають малу актуальність. Поверхня атаки значно ширша, часто безмежна, і включає різнорідні системи.

Варто зазначити, що одним із ключових елементів сучасної безпеки в Інтернеті є використання складних криптографічних алгоритмів, що потребують значної потужності обробки. Багато, якщо не більшість, пристроїв IoT базуються на низькопродуктивних процесорах або мікроконтролерах, що мають низьку потужність обробки та пам'ять, і не розроблені з безпекою як пріоритетною метою.

Шифрування для забезпечення конфіденційності, автентифікація для підтвердження особи та автентифікація інформації за допомогою цифрових підписаних сертифікатів є ключовими механізмами безпеки в Інтернеті сьогодні. Ці механізми базуються на наступному:

— криптографічні шифри такі як Advanced Encryption Standard (AES), Secure Hash Algorithm (SHA2), та відкриті ключі RSA та криптографія еліптичних кривих (ECC);

— протокол Transport Layer Security (TLS) та його попередник Secure Sockets Layer (SSL), які забезпечують автентифікацію та шифрування інформації за допомогою згаданих шифрів;

— інфраструктура відкритих ключів (PKI) забезпечує будівельні блоки для автентифікації та довіри за допомогою стандарту цифрових сертифікатів та Центрів сертифікації (CA);

1.3.2 Значення аналізу мережевого трафіку IoT

Аналіз мережевого трафіку відіграє ключову роль у забезпеченні стійкості, продуктивності та безпеки мережі. Основні моменти, які роблять аналіз мережевого трафіку важливим, включають:

а) виявлення кіберзагроз та злочинної активності: Аналіз мережевого трафіку допомагає виявляти кіберзагрози, такі як вторгнення, спам, фішинг та інші, що допомагає запобігти можливим кібератакам та захистити мережу від шкідливих дій;

б) оптимізація використання ресурсів мережі: Аналіз мережевого трафіку допомагає виявити проблеми з пропускною здатністю та забезпечує оптимальне використання ресурсів мережі;

в) відновлення роботи мережі: Аналіз мережевого трафіку може допомогти швидко відновити роботу мережі після виникнення проблем або відмов обладнання;

г) підвищення рівня безпеки: Аналіз мережевого трафіку дозволяє виявляти вразливості в мережі та вживати заходів для їх усунення, що сприяє запобіганню кібератак;

д) підтримка прийняття рішень: Аналіз мережевого трафіку надає необхідну інформацію для прийняття обґрунтованих рішень щодо конфігурації мережі та вибору обладнання, що відповідає вимогам безпеки;

У цілому, аналіз мережевого трафіку відіграє важливу роль у забезпеченні безпеки, ефективності та стійкості мережі, а також у виявленні проблем та прийнятті обґрунтованих рішень[1].

2 ТЕОРЕТИЧНІ ОСНОВИ АНАЛІЗУ МЕРЕЖЕВОГО ТРАФІКУ У IOT

2.1 Огляд протоколів, що використовуються в IoT

TCP/IP — це в першу чергу стек протоколів. То що ж це таке? Протоколом називається набір правил для однієї з комунікаційних функцій. Наприклад, протокол IP представляє собою набір правил для маршрутизації даних, а TCP визначає правила для надійної та послідовної доставки даних. А стек протоколів — набір організованих за рівнями протоколів, які, працюючи разом, дозволяють програмам обмінюватися даними.

Потреба у створенні TCP/IP виникла у 1970-х роках, коли були підключені перші хости до мережі та було виявлено, що тодішні протоколи ARPANET(Advanced Research Project Agency Network) працювали повільно і часто приводили до краху мережевих комунікацій. Отож, у статті Вінтона Г.Серфа “A Protocol for Packet Network Interconnections” від 1974 року, фактично було покладено зародження TCP/IP і вже починаючи з 1980 року - потребувалося 3 роки, щоб системи перейшли на нові протоколи(на той рік вже налічувалося 100 хостів).

Протоколи TCP/IP були розроблені з урахуванням незалежності від апаратного забезпечення хостів і їх операційних систем, а також від використовуваного середовища передачі даних і технологій зв'язку. Ці протоколи забезпечують високу надійність, зберігаючи працездатність навіть при високому рівні мережевих помилок, і, крім того, підтримують прозору перебудову маршрутів при втраті мережевих вузлів або рядків даних.

Крім простоти при об'єднанні декількох мереж, протоколи TCP/IP відкрили шлях для об'єднання мереж ARPANET університетів та дослідницьких організацій, що нарешті призвело до створення супермережі - Інтернету. Протягом 80-х років магістральні з'єднання Інтернету забезпечувалися за допомогою ARPANET.

Характеристики протоколів TCP/IP забезпечили стабільне і однорідне розширення мережі Інтернет, яка стала найбільшою глобальною мережею, об'єднуючи урядові та військові мережі, а також мережі університетів та комерційних організацій (кожна з яких може складатися з сотень підмереж). У 1985 році була створена нова магістральна мережа National Science Foundation Net (NSFNET), яка забезпечила швидкі з'єднання для дослідницьких центрів та суперкомп'ютерів.

Завдяки урядовій підтримці була сформована інфраструктура регіональних постачальників послуг (регіональних Service Provider), що охопила всю територію США. Університети та дослідницькі лабораторії підключалися до найближчого регіонального постачальника, який забезпечував магістральні з'єднання всередині загальної мережі. Це посприяло стрімкому розширенню Інтернету та формування постачальників послуг в десятках країн по всьому світу[11].

MQTT(message queuing telemetry transport) — це стандартний протокол обміну повідомленнями для Інтернету Речей, який працює поверх TCP/IP за принципом публікації/підписки. Він ідеально підходить для підключення віддалених пристроїв з невеликим об’ємом коду та мінімальною пропускною здатністю мережі. Зачастую це є різноманітні датчики/роботи/різноманітні битові прилади або-ж пристрої. MQTT працює за принципом радіо — йде трансляція по деякому каналу зв’язку, а підписники отримують ту інформацію, на розсилку його вони підписалися. Для цієї задумки MQTT потрібен центральний брокер, який буде сортувати повідомлення і передавати інформацію на ті теми(topics), на які підписалися пристрої[2]. Схема зображена на рисунку 2.1.

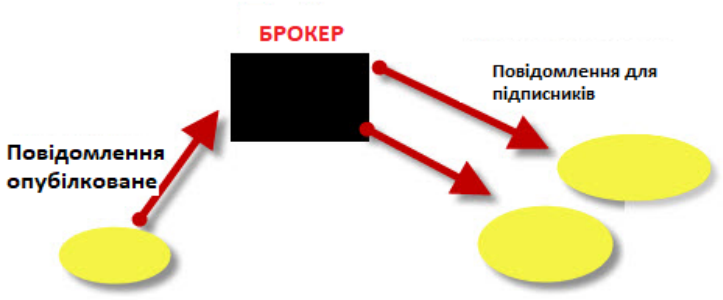

Рисунок 2.1 — Схема передачі повідомлення

Деякі важливі моменти, що треба знати про MQTT:

а) відсутність адрес: На відміну від систем електронної пошти, де клієнти мають адреси, в MQTT клієнти не мають адрес;

б) публікація повідомлень: Клієнти публікують повідомлення на теми, а не надсилають їх безпосередньо іншим клієнтам;

в) фільтрація та розповсюдження: Брокер MQTT відповідає за фільтрацію повідомлень за темами та розповсюдження їх підписникам;

г) підписка*:* Клієнти підписуються на теми, щоб отримувати пов'язані з ними повідомлення;

д) відсутність прямого зв'язку: Між публікатором та підписчиком немає прямого зв'язку;

е) публікація та підписка*:* Всі клієнти можуть публікувати (транслювати) та підписуватися (отримувати) повідомлення;

є) зберігання повідомлень: Брокери MQTT зазвичай не зберігають повідомлення;

Оскільки протокол працює за допомогою TCP/IP, це дозволяє пристроям тримати з’єднання дуже довгий час. А щоб перевірити, чи підключений пристрій до брокеру, кожні 60 секунд надсилаються повідомлення підтримки активності, щоб зрозуміти - чи залишається ще на зв’язку абонент чи ні. Також, кожен, хто хоче під’єднатися до брокера - потрібен мати унікальний ідентифікатор клієнта. У випадку, якщо його не будет — з’єднання буде неможливо, це справедливо і для ситуації, коли клієнт матиме ідентифікатор, але до брокеру вже під'єднаний такий пристрій с таким самим ID.

Отже, MQTT грає важливу роль як протокол у IoT за рахунок своєї універсальності та потужності, що робить його ідеальним вибором для широкого спектру IoT-застосунків.

CoAP(Constrained Application Protocol) — це спеціалізований протокол передачі даних в Інтернеті речей (IoT), який призначений для використання в пристроях та мережах з обмеженими ресурсами, такими як обмежена пропускна здатність чи потужність.

Він ґрунтується на архітектурі клієнт-сервер, подібній до HTTP, з особливим акцентом на підвищенні продуктивності в умовах з обмеженими ресурсами. Замість TCP (протокол передачі управління) в ньому використовується UDP (протокол дейтаграмних пакетів), щоб зменшити навантаження. Це робить CoAP ідеальним для пристроїв IoT, які живляться від батарей або працюють у сценаріях з нестабільним мережевим з'єднанням.

Вцілому, CoAP — простий протокол із методами, подібними до HTTP, такими як GET, POST, PUT та DELETE, для взаємодії з ресурсами. Він також підтримує асинхронні сповіщення та використовує компактні бінарні заголовки для ефективності пакетів та уніфіковані ідентифікатори ресурсів для спрощення адресації. Клієнти надсилають запити, а сервери відповідають, забезпечуючи надійну доставку даних у нестабільних мережах. Функція "спостереження" дозволяє отримувати оновлення ресурсів без постійного опитування[3].

LoRaWAN та Lora — це два ключових елементи у сфері бездротових комунікацій. LoRa - це модуляційна техніка, яка базується на технології Chirp Spread Spectrum (CSS) і дозволяє передавати дані на великі відстані. Ця технологія ідеально підходить для передачі невеликих обсягів даних з низькою швидкістю передачі. LoRa працює на ліцензійних бездротових підгігагерцевих діапазонах, таких як 915 МГц, 868 МГц та 433 МГц, а також на 2,4 ГГц.

LoRaWAN - це протокол керування доступом до носія середнього рівня, побудований на основі модуляції LoRa. Це програмний шар, який визначає, як пристрої використовують апаратне забезпечення LoRa, наприклад, коли вони передають дані, та формат повідомлень. Цей протокол розробляється та підтримується LoRa Alliance і має важливе значення для побудови мереж Інтернету речей.

LoRaWAN вражає своєю ефективністю та потужністю в декількох аспектах:

— низьке енергоспоживання: Пристрої LoRaWAN оптимізовані для роботи у режимі низького споживання енергії та можуть працювати до 10 років від одного елемента живлення типу "монетка";

— великий радіус дії: Ворота LoRaWAN можуть передавати та отримувати сигнали на відстань понад 10 кілометрів у сільських районах та до 3 кілометрів у густонаселених міських районах;

— глибока проникність у приміщення*:* Мережі LoRaWAN можуть забезпечувати глибоке внутрішнє покриття та легко охоплюють будівлі з кількома поверхами;

— спектр без ліцензії: Вам не потрібно платити високі плати за ліцензію на частотний спектр для розгортання мережі LoRaWAN;

— геолокація: Мережа LoRaWAN може визначати місцезнаходження кінцевих пристроїв за допомогою тріангуляції без необхідності в GPS;

— високий обсяг*:* Сервери мережі LoRaWAN обробляють мільйони повідомлень від тисяч воріт;

— захист кінцевих точок*:* LoRaWAN забезпечує безпечну комунікацію між кінцевим пристроєм та сервером за допомогою шифрування AES-128;

— дистанційне новлення програмного забезпечення: Ви можете віддалено оновлювати програмне забезпечення для кінцевих пристроїв через повітря;

— роумінг: Кінцеві пристрої LoRaWAN можуть виконувати плавний перехід з однієї мережі в іншу;

— низькі витрати: Мінімальна інфраструктура, економічні кінцеві вузли та відкрите програмне забезпечення;

Завдяки цьому, LoRaWAN має застосування у таких секторах як: моніторинг температури вакцин, ведення обліку диких тварин, відстеження пацієнтів з деменцією, аграрні справи, водозбереження, контроль якості їжі, управління відходами, відстеження велосипедів, аеропортове відстеження та багато іншого[4,25].

2.2 Типи мережевого трафіку в IoT

В IoT існує багато типів трафіку, але їх усіх можна об’єднати в три окремі групи:

а) командний трафік: Це тип трафіку, який включає в себе команди або інструкції, які відправляються від контролера або центрального пристрою до вузлів IoT для виконання певних дій. Наприклад, команди на управління вмикачами, реле або іншими пристроями. Командний трафік може бути використаний для включення або вимкнення пристроїв, зміни режимів роботи тощо;

б) сенсорний трафік: Це тип трафіку, який включає в себе дані, що збираються від сенсорів або датчиків, розташованих на IoT-пристроях. Ці дані можуть включати інформацію про температуру, вологість, освітленість, рівень води, рух та інші параметри навколишнього середовища. Сенсорний трафік є основою для збору даних та аналізу в IoT системах;

в) керуючий трафік: Це тип трафіку, який включає в себе дані, що відправляються від центральних пристроїв або контролерів до IoT-пристроїв для керування їхньою роботою або параметрами. Наприклад, налаштування параметрів роботи, віддалене оновлення програмного забезпечення, управління енергозбереженням тощо;

Розділення трафіку в IoT - це ключовий фактор для оптимізації продуктивності, надійності, безпеки, масштабованості та аналітики мереж IoT.

Це дозволяє ефективно управляти трафіком, забезпечувати безперебійну роботу, захищати дані та приймати обґрунтовані рішення щодо розвитку IoT-систем.

2.3 Методи аналізу мережевого трафіку

Аналіз трафіку поділяють на активний та пасивний. Ці методи суттєво відрізняються та мають свої недоліки та плюси. Щодо пасивного — це метод, при якому виконується збір даних без втручання у процес передачі даних. Він полягає у моніторингу трафіку, який протікає через мережу, та аналізі цієї інформації для виявлення шаблонів, аномалій тощо. Може виявляти широкий спектр проблем, пов’язаний з продуктивністю і безпекою, а також не суттєво впливає на роботу мережі і пристроїв. Такий аналіз проводиться за допомогою сніферів, що записують усі мережеві пакети, що проходять через інтерфейс, та систем виявлення/запобігання вторгнень.

Активний аналіз відрізняється тим, що потребує активної діяльності спеціаліста: аналітик створює та відправляє спеціально сформовані пакети даних або запити для збору інформації або тестування реакції мережі на певні події. Такий аналіз потребує знань, може уповільнити роботу мережі на час тестування, але незважаючи на це, може предоставити більше даних про мережу та її характеристики, виявити інші скриті загрози, які неможливо виявити під час пасивного аналізу.

Окрім аналізу саме трафіку мережі, можна проводити аналіз різноманітних протоколів, як-от TCP, UDP, ICMP тощо може допомогти визначити, які типи трафіку домінують у мережі, ідентифікувати потенційні проблеми з продуктивністю або безпекою, пов'язані з певними протоколами. Окремо можна зробити DNS аналіз, який надасть таку інформацію:

DNS-запити з підозрілими характеристиками

— запити з підробленими адресами джерела або адресами, на використання яких ви не давали дозволу: це може свідчити про участь заражених вузлів вашої мережі в DDoS-атаці;

— незвично велика кількість DNS-запитів: ознака бот-активності або сканування мережі;

DNS-запити, які використовують протокол TCP замість UDP: може бути атака з використанням уразливості.

Спотворені DNS-запити

— запити, спрямовані на IP-адреси, відомі як джерела шкідливого ПО: свідчить про наявність заражених вузлів у вашій мережі;

— запити, які використовують алгоритми генерації доменів (DGA): ознака бот-активності;

— несанкціоновані запити до іменних серверів: інформує про спробу перенаправити трафік на шкідливі ресурси;

DNS-відповіді з підозрілими характеристиками

а) Аномально великі відповідні повідомлення — атака з посиленням (amplification attack).

б) Сегменти відповідного повідомлення, які викликають підозру — ознака "отруєння" кешу або прихованого каналу.

в) DNS-відповіді для ваших доменів, які повертають IP-адреси, що відрізняються від опублікованих кимось — можливо взлом доменного імені або облікового запису реєстрації.

г) відповіді від іменних серверів, на яких не давали дозволу розміщувати свою зону — несанкціонована зміну конфігурації DNS;

д) позитивні відповіді при визначенні адрес тих імен у вашій зоні, які повинні повертати повідомлення про помилку (NXDOMAIN) — злом DNS;

е) DNS-відповіді з підозрілих IP-адрес — ознака бот-активності або сканування мережі;

є) надмірна кількість відповідних повідомлень на запити з коротким часом життя (TTL) — можлива бот-активність;

ж) надмірна кількість відповідей, які містять "помилку розпізнавання імені" (NXDOMAIN) — бот-активність або сканування мережі[];

2.4 Інструменти для аналізу мережевого трафіку

Для виконання цих завдань існує ряд спеціалізованих інструментів, які дозволяють детально аналізувати трафік, виявляти патерни, ідентифікувати проблеми та покращувати загальну продуктивність мережі. До таких інструментів належать Wireshark, NetworkX, FlowMon та інші. Кожен з них має свої особливості та переваги, що дозволяє вибирати найбільш підходящий варіант в залежності від конкретних потреб і завдань.

2.4.1 Wireshark

Wireshark, відомий спочатку як Ethereal, має цікаву історію, що почалася в середині 90-х років:

— 1998 рік: Геральд Комер розпочав проект Ethereal, що став одним з перших інструментів для аналізу мережевого трафіку з відкритим кодом;

— 2006 рік: Через юридичні питання щодо назви "Ethereal", проект був перейменований на Wireshark після розділення між розробниками та Фондом безпеки мережі;

— 2008 рік: Вийшла перша стабільна версія Wireshark 1.0.0, яка включала численні покращення функцій та розширену підтримку протоколів;

— 2013 рік: Версія Wireshark 1.10.0 представила важливі оновлення, такі як підтримка нових протоколів і покращення швидкодії;

Подальший розвиток Wireshark включає регулярні оновлення, що забезпечують підтримку нових мережевих технологій, покращення інтерфейсу користувача та підвищення продуктивності.

Існує можливість роботи в графічному інтерфейсі (див. рисунок 2.2), що дозволяє використовувати весь інструментарій безпосередньо на робочому місці. При роботі на віддаленому сервері на через консоль можна зберігати дані в файли з розширенням .pcap, та проводити аналіз вже на локальній машині. При запуску

додатка можна завантажити файл з існуючими даними або ж почати перехоплення пакетів. Також існує можливість налаштування фільтрів.

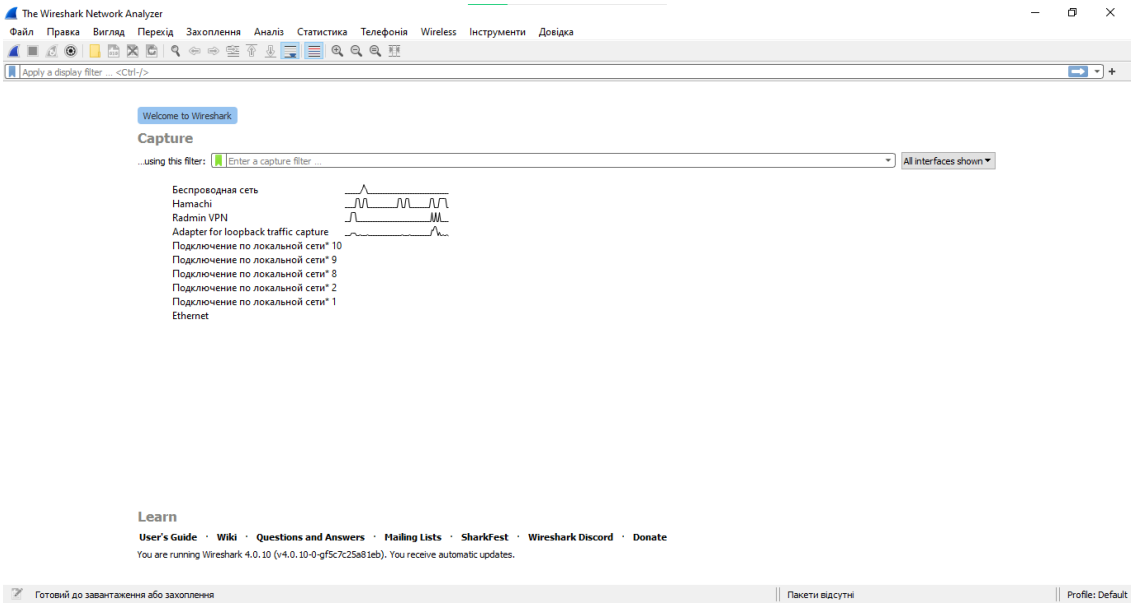

Рисунок 2.2 — Приклад програми Wireshark

До недоліків аналізатору Wireshark можна віднести:

— високий рівень складності налаштування фільтрів;

— перевантажений інтерфейс;

— менша продуктивність в порівнянні з комерційними продуктами;

2.4.2 Microsoft Network Monitor

Network Monitor, створений компаніями Windows та Microsoft Systems Management Server, є ефективним інструментом для моніторингу мережевого трафіку. Він дозволяє здійснювати як реальний час, так і зберігати та аналізувати трафік для подальшого дослідження. Ці можливості важливі для виявлення та вирішення проблем у локальних та розподілених мережах, а також для роботи з різноманітними мережевими пристроями, які використовують стек протоколів TCP/IP.

Основні переваги Network Monitor включають можливість діагностики проблем з мережевими з'єднаннями. Інструмент дозволяє переглядати пакети протоколів TCP/IP, надаючи детальну інформацію про дані, що передаються між пристроями в мережі.

Засоби програми також дозволяють оцінювати продуктивність мережі, надаючи інформацію про навантаження, джерела трафіку та слабкі місця. Це сприяє створенню об’єктивної оцінки роботи локальних комп'ютерних мереж (ЛКМ).

Network Monitor здатен знаходити обладнання, яке передає так звані "маяки" – повідомлення про помилки у мережах з кільцевою топологією. Хоча більшість сучасних мереж є комутованими, інструмент все ще корисний для пошуку фрагментованих пакетів.

Приклад графічного інтерфейсу Network Monitor зображений на рис. 2.3.

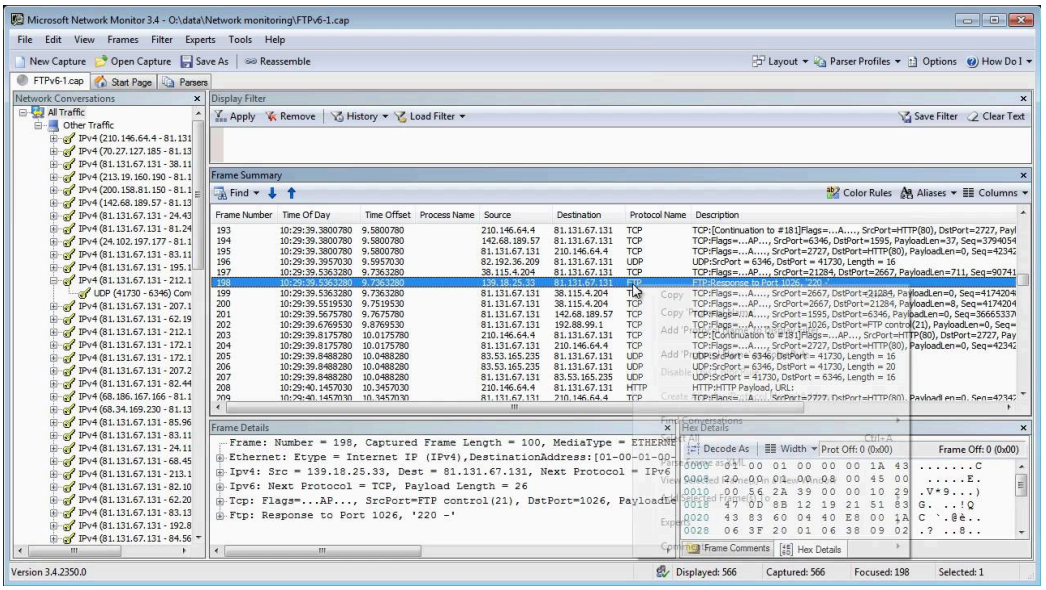

Рисунок 2.3 — Інтерфейс Network Monitor

Скориставшись цим мережевим аналізатором, можна помітити, що він має ті ж самі недоліки, що й Wireshark: громіздкий навантажений інтерфейс, підтримку тільки Windows систем, високе навантаження на комп’ютер та менш детальне декодування протоколів в порівнянні з комерційними продуктами.

2.4.2 LanHound

LanHound складається з двох основних компонентів: адміністративної консолі та віддаленого агента для перехоплення пакетів (де консоль виступає як агент). Цей мережевий аналізатор функціонує виключно на операційних системах Windows.

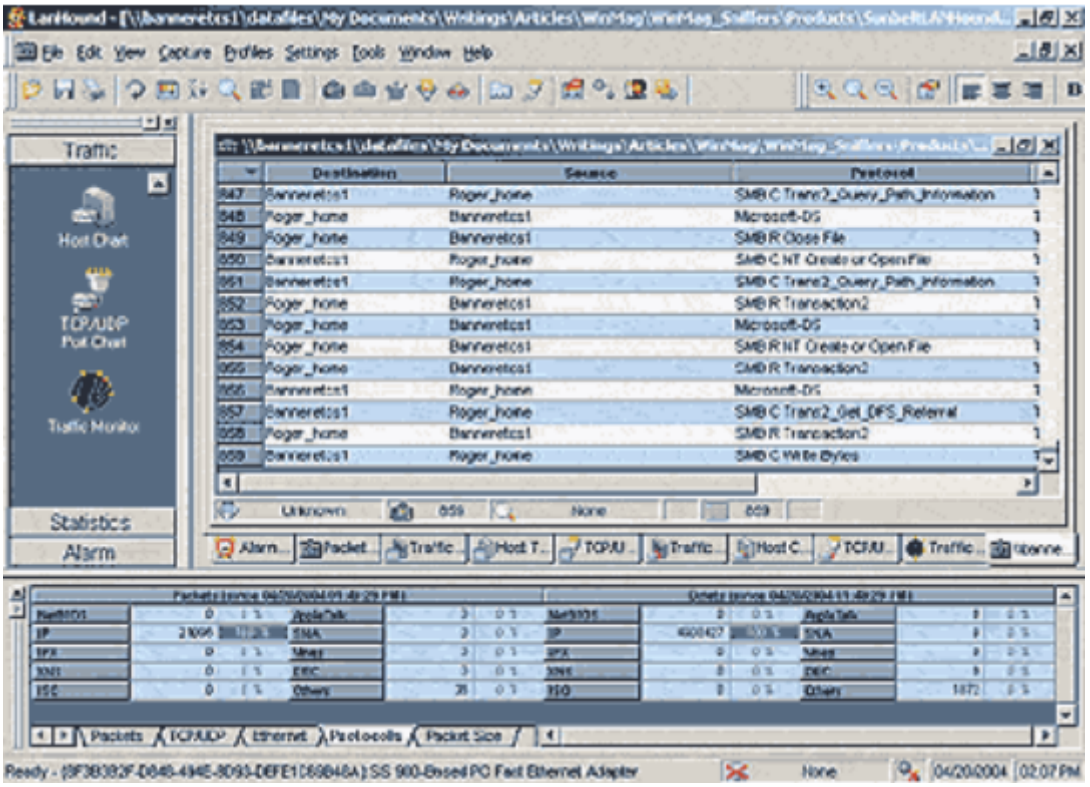

Рисунок 2.4 — Інтерфейс LanHound

LanHound вирізняється простим графічним інтерфейсом (див. рисунок 2.4) та набором стандартних для подібних програм функцій, таких як фільтрація перехопленого трафіку, пошук по імені, попередження і створення звітів. Програма включає модуль експертного аналізу, який дещо розширює можливості стандартних попереджень. Звіти, які генерує LanHound, містять таблиці активних пристроїв, гістограми, матриці трафіку та підсумкові звіти по пакетах.

Зібрані LanHound дані можна обробляти різними способами, наприклад, створювати гістограми і діаграми. Одна з цікавих особливостей цієї програми –

можливість маніпулювати збереженим трафіком і навіть повторно транслювати його по мережі, що рідко зустрічається у продуктах цього класу.

До недоліків програми відноситься:

— висока ціна;

— слабка дешифрація багатьох протоколів;

— не працює під управлінням Windows 2003, 2008;

— відсутній модуль експертного аналізу;

2.4.3 CommView

CommView є комерційним продуктом, але для ознайомлення доступна демонстраційна версія з обмеженими функціями. Цей пакетний аналізатор призначений для моніторингу як локальних, так і глобальних мереж. Він може захоплювати мережеві пакети, декодувати їх і надавати детальну інформацію у зручному форматі. Відмінною рисою CommView є те, що він не вимагає попереднього встановлення WinPcap, на відміну від багатьох інших подібних інструментів.

Програма підтримує операційні системи Windows і має великий перелік підтримуваних протоколів, що дозволяє отримувати детальну інформацію про перехоплені пакети.

CommView також дозволяє створювати фільтри для захоплення пакетів, однак ці фільтри не можуть бути застосовані до вже зібраних даних; вони працюють лише під час процесу захоплення. Інтерфейс програми складається з трьох вікон: у першому відображаються захоплені пакети, у другому – декодована інформація про окремий пакет, а у третьому – вміст самого пакета (див. рисунок 2.5).

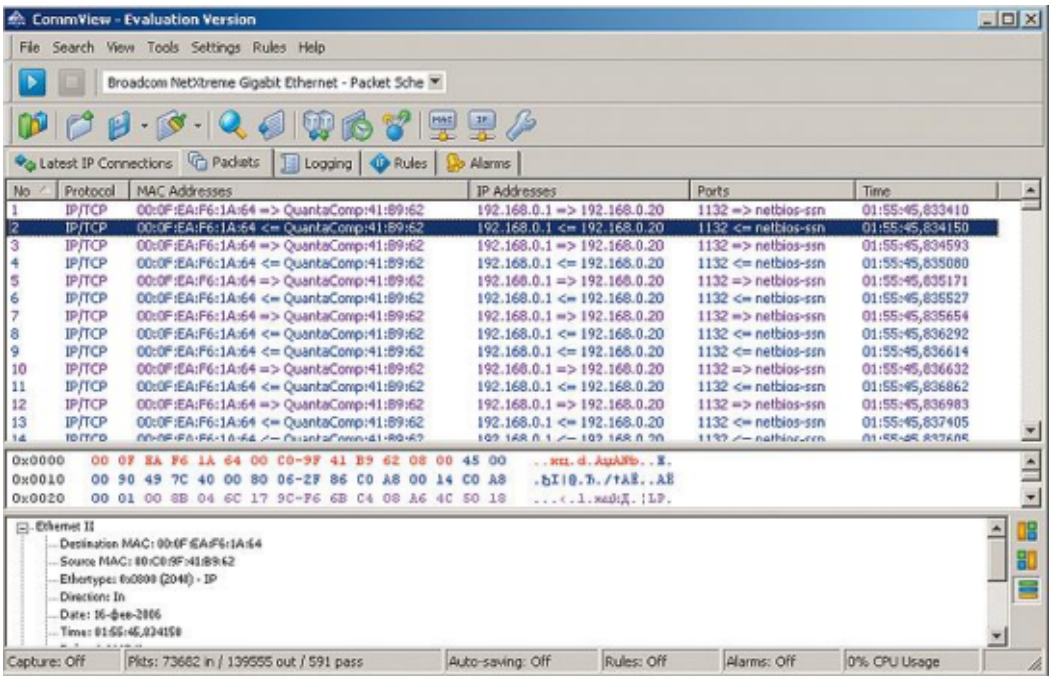

Рисунок 2.5 — Інтерфейс програми CommView зі змістом пакету

З недоліків можна виділити:

— висока ціна;

— відсутність деяких Windows-декодерів, таких як Kerberos і RDP;

— роздільні версії для локальної і бездротової мережі;

— для деяких мережевих адаптерів підтримується тільки змішаний режим;

Отже, розібравши усі наявні популярні програми для аналізу мережевого трафіку, було обрано для роботи Wireshark саме через доступність, надійність та некомерційність цієї програми. Для більш серйозних завдань краще обрати інший платний продукт з потрібними можливостями. Зважаючи, що це лише кваліфікаційна робота, був зроблений саме такий вибір.

3 АНАЛІЗ МЕРЕЖЕВОГО ТРАФІКУ З ВИКОРИСТАННЯМ WIRESHARK

3.1 Вибір платформи для встановлення Wireshark

Wireshark може бути встановлений на різні операційні системи, включаючи Windows, macOS та Linux. Кожна з цих платформ має свої переваги та недоліки, які можуть вплинути на досвід роботи з аналізом мережевого трафіку.

Windows є популярною платформою для встановлення Wireshark завдяки своїй широкій доступності та простоті у використанні. Однак, при роботі з Wireshark на Windows можуть виникнути деякі проблеми, які слід враховувати:

— проблеми з продуктивністю: wireshark може працювати повільніше на Windows, особливо при обробці великих обсягів трафіку;

— обмеження прав доступу: для захоплення трафіку на низькому рівні (наприклад, для захоплення всіх пакетів у бездротовій мережі) можуть бути потрібні додаткові права адміністратора або спеціальні драйвери (наприклад, Npcap);

— сумісність драйверів: існує ймовірність виникнення помилок через несумісність драйверів мережевих адаптерів, особливо при роботі з новітніми або специфічними моделями;

Linux, з іншого боку, часто є кращою платформою для роботи з Wireshark, особливо для професіоналів з мережевого аналізу та кібербезпеки:

— вища продуктивність: Лінукс, як правило, забезпечує кращу продуктивність і ефективність при роботі з мережевим трафіком;

— повний доступ до мережевих інтерфейсів: у Linux можна легко отримати повний доступ до мережевих інтерфейсів без необхідності встановлення додаткових драйверів або прав адміністратора;

— широка підтримка мережевих утиліт: linux надає доступ до широкого спектру мережевих утиліт та інструментів, які можуть бути інтегровані з Wireshark для покращення аналізу;

3.1.1 Встановлення на Windows

1) Потрібно завантажити останню версію з офіційного сайту https://www.wireshark.org/download.html
2) Запустити інсталятор та притримуватися інструкцій майстра встановлення. Рекомендується встановити Npcap, який необхідний для захоплення трафіку.
3) Після інсталяції переконатися, що отримано права адміністратора

3.1.2 Встановлення на Linux

1) Потрібно ввести наступні команди в термінал:
   sudo apt update
   sudo apt install wireshark
2) Додайте свого користувача до групи wireshark для дозволу захоплення трафіку без привілеїв root:
   sudo usermod -aG wireshark $(whoami)
3) Перезавантажити пристрій щоб зміни набули чинності.

3.1.3 Технологія захоплення трафіку WinPCAP

Архітектурний дизайн WinPCAP розширює стандартні можливості операційних систем сімейства Win32, дозволяючи приймати та передавати дані по мережі без використання стеку протоколів операційної системи. Він взаємодіє безпосередньо з мережевим адаптером комп'ютера. Крім того, ця архітектура забезпечує високорівневе API для керування низькорівневими процесами. WinPCAP складається з трьох ключових компонентів: драйверу пристрою захоплення пакетів (packet.vxd), динамічної бібліотеки низького рівня (packet.dll) та статичної бібліотеки високого рівня (libpcap). Ця архітектура знаходить застосування в розробці додатків обробки пакетів для операційних систем

Windows і UNIX[]. На рисунку 3.1 наведено структуру стека захоплення пакетів, що описує шлях від мережевого адаптера до додатка верхнього рівня.

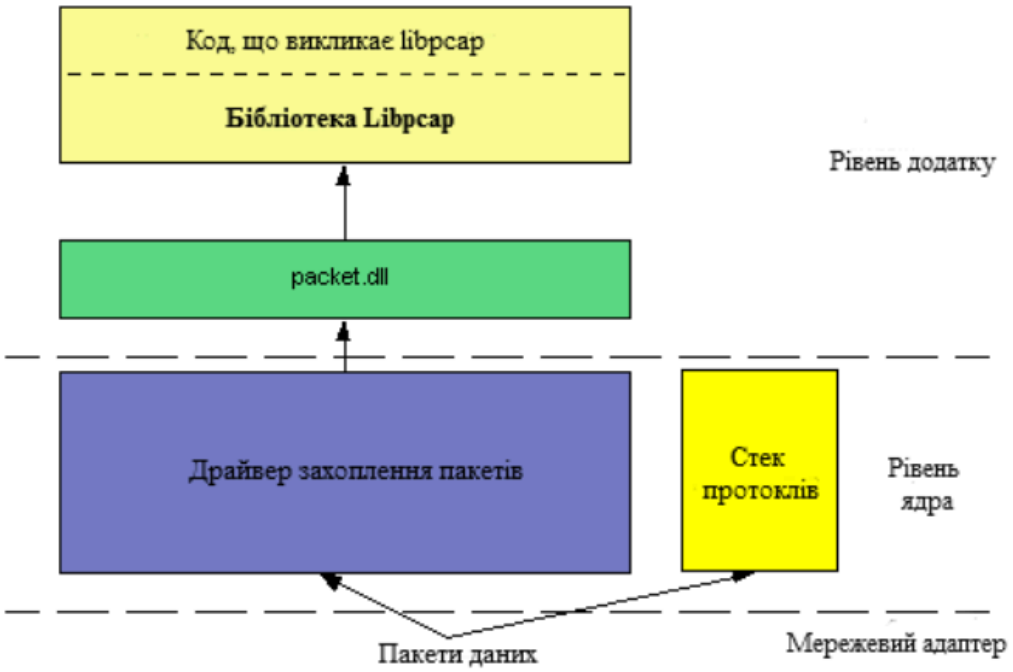

Рисунок 3.1 – Структура стека захоплення пакетів

На найнижчому рівні розташований мережевий адаптер, який приймає всі пакети, що проходять через мережу. Драйвер packet.vxd є низькорівневим програмним компонентом, що функціонує на рівні ядра операційної системи та безпосередньо взаємодіє з драйвером мережевого адаптера. Pcap-драйвер надає набір функцій низького рівня для прийому і передачі даних на канальному рівні через NDIS. Він відповідає за керування різними типами адаптерів та забезпечує зв'язок між адаптером і програмним забезпеченням, що формує пакети різної структури.

Динамічна бібліотека packet.dll відокремлює додаток користувача від драйвера, надаючи незалежний від операційної системи інтерфейс. Це дозволяє додатку працювати на різних версіях Windows без необхідності перекомпіляції. Бібліотека packet.dll працює на рівні користувача, але функціонує окремо від основної програми.

Статична бібліотека libpcap використовується в програмному забезпеченні користувача для перехоплення і фільтрації пакетів. Вона використовує функції, що надаються бібліотекою packet.dll, забезпечуючи управління процесами прийому і фільтрації даних на високому рівні. Бібліотека libpcap статично пов'язана з програмою користувача і є її невід'ємною частиною.

Специфікація NDIS визначає три типи мережевих драйверів:

а) Драйвери протоколів верхнього рівня;

б) Драйвери протоколів проміжного рівня;

в) Драйвери мережевих карт (NIC), включаючи драйвери мережевих карт локальних мереж (NDIS LAN Miniport NIC) та драйвери мережевих карт глобальних мереж (NDIS WAN Miniport);

На основі цієї технології побудован Wireshark. Архітектура програми зображена на рисунку 3.1.2.

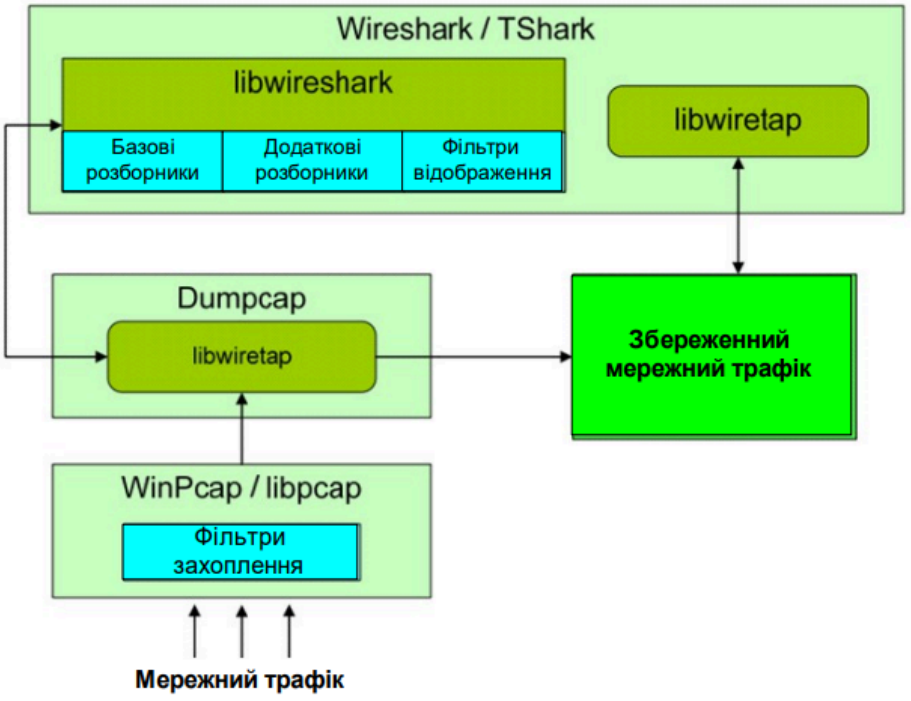

Рисунок 3.1.2 — Архітектура програми

3.1.4 Інтерфейс Wireshark

Інтерфейс програми досить простий, зустрічне вікно зображене на рис. 3.2.

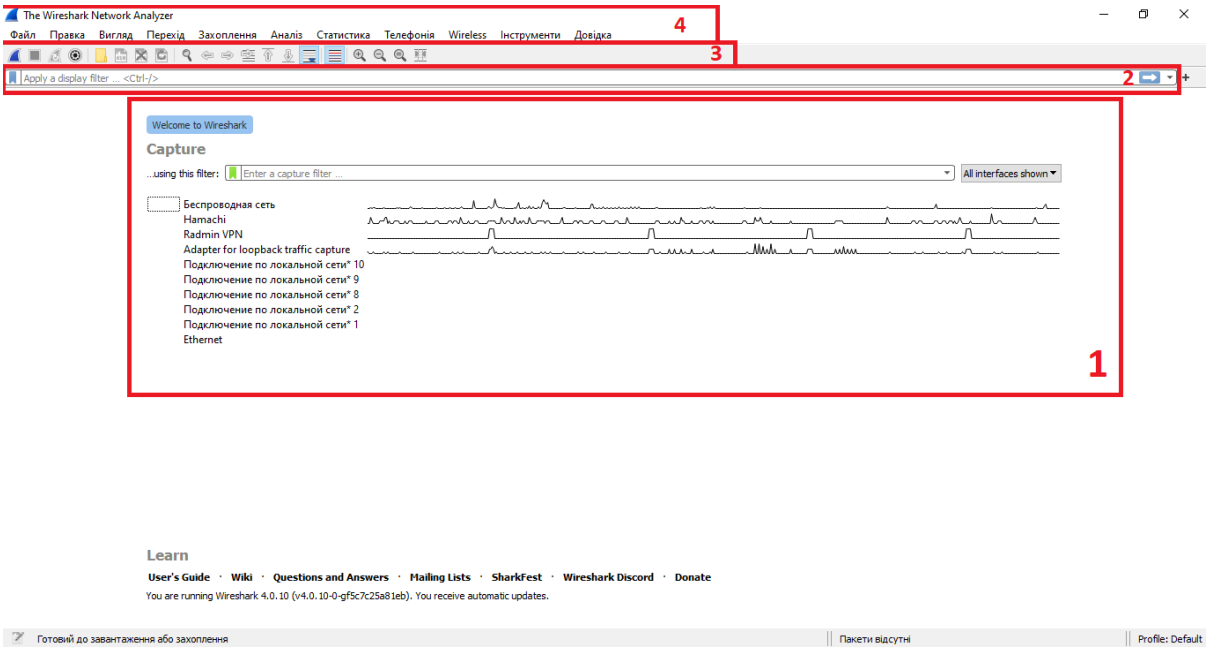

Рисунок 3.2 — Зустрічний інтерфейс Wireshark

1 — Наявні мережеві адаптери для прослуховування та статистикою використання у реальному часі

2 — Строка для вводу фільтрів

3 — Панель інструментів, яка включає в себе керування запуском сканування, збереженням файлів захоплення, пошуку пакетів, збільшенням/зменшенням відображуваного тексту.

4 — Головне меню, включає в себе такі пункти як Файл(управління відкриттям файлів/зберіганням), Правка(пошук пакетів, налаштування параметрів та профілів конфігурації), Вид(налаштування відображення елементів інтерфейсу та масштабування), Перехід(перемикання між пакетами), Захоплення(запуск, зупинка та налаштування параметрів захоплення трафіку), Аналіз(робота з фільтрами, статистикою та експертною інформацією), Телефонія(аналіз VoIP дзвінків, RTP та SIP потоків), Інструменти(додаткові інструменти для генерації

правил фаєрволу та інших задач), Довідка(доступ до документації та інформації про програму)

Після вибору мережевого адаптеру для аналізу, Wireshark відозміниться та відобразить інший інтерфейс, який зображений на рисунку 3.2.1.

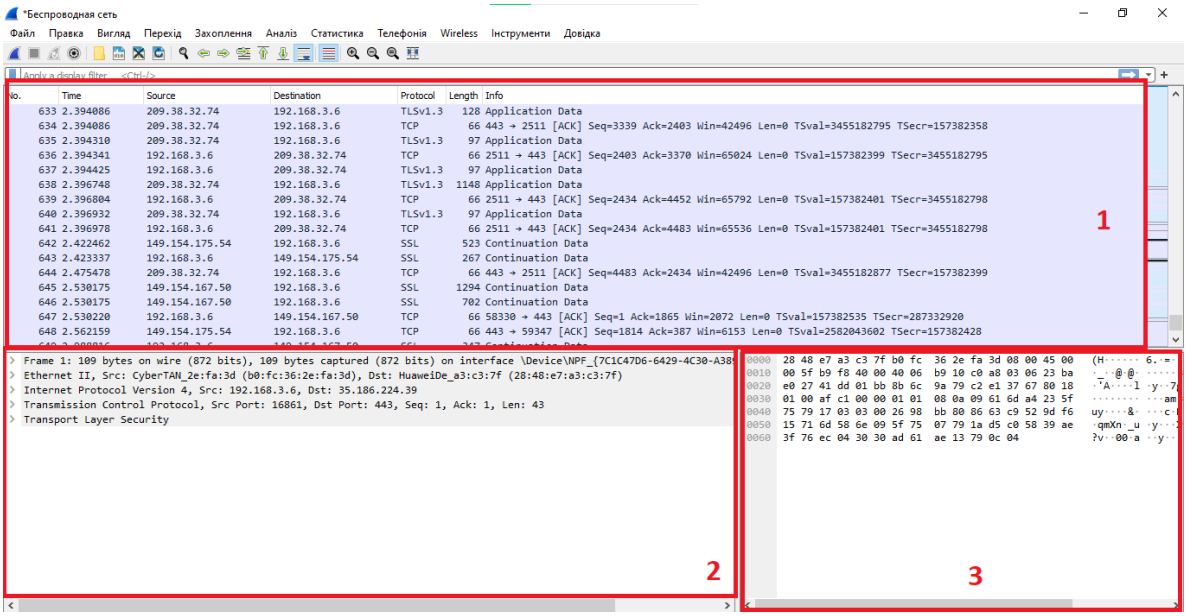

Рисунок 3.2.1 — Робочий інтерфейс Wireshark

1 — Список усіх захоплених пакетів

2 — Область деталей пакету: область, що відображає детальну інформація про вибраний пакет. Тут можна побачити всі рівні протоколів та поля заголовків.

3 — Область байтів пакету: область, де відображаються сирі дані вибраного пакета в шістнадцятковому та ASCII форматах. Вона показує, як виглядає пакет на рівні байтів.

Зробивши огляд програми та проаналізувавши інтерфейс, можна зробити висновок, що Wireshark був розроблений з урахуванням потреб різних категорій користувачів, від початківців до досвідчених професіоналів. Його логічна структура та зручні інструменти дозволяють швидко і ефективно аналізувати мережевий трафік.

Глибокий рівень деталізації при перегляді пакетів, розширені можливості фільтрації та сортування даних дозволяють проводити ретельний аналіз поведінки мережевих протоколів та розбиратися у складних проблемах. Експертна система

програми також є неоціненним помічником для виявлення потенційних вразливостей, аномалій та зловмисної активності в мережевому трафіку.

3.2 Підготовка до захоплення трафіку

Перед початком захоплення мережевого трафіку в IoT необхідно ретельно підготуватися, щоб гарантувати ефективність та точність процесу. Перед усім потрібно визначити цілі аналізу, це може включати як і просто моніторинг продуктивності мережі, так і виявлення потенційних загроз безпеці, усунення несправностей тощо. Окрім цього, важливим кроком буде ознайомлення з топологією мережі та визначити пристрої, які вже підключені, та протоколами, які найчастіше використовувалися. Базуючись на топології мережі та цілях захоплення, необхідно вибрати оптимальні точки для встановлення Wireshark або інших інструментів захоплення трафіку. Це можуть бути шлюзи, які з'єднують IoT мережу з іншими мережами, або стратегічно розташовані вузли, через які проходить більшість трафіку.

У цій роботі буде використовуватися та досліджуватися звичайна домашня мережа. Вона складається з восьми пристроїв та діапазоном IPv4 адрес з 192.168.1.3(маршрутизатор) по 192.168.1.50(найбільш відомий високий адрес) та об’єднані в топологію Зірка. Вона має мінуси, але простота організації мережі завдяки цій топології — приваблює.

Оскільки аналізуватися мережа буде через безпровідне підключення, варто внести зміни у налаштуваннях шифрування, додавши ключі доступу для дешифровки трафіку. Оскільки трафік шифрується алгоритмом WPA/WPA2-PSK, потрібно ввести пароль/ssid мережі у налаштуваннях. Щоб змінити це, потрібно перейти у Правку -> Налаштування -> Протоколи -> IEEE 802.11 -> Decryption keys -> Edit та ввести дані. Приклад внесених змін зображений на рисунку 3.3.

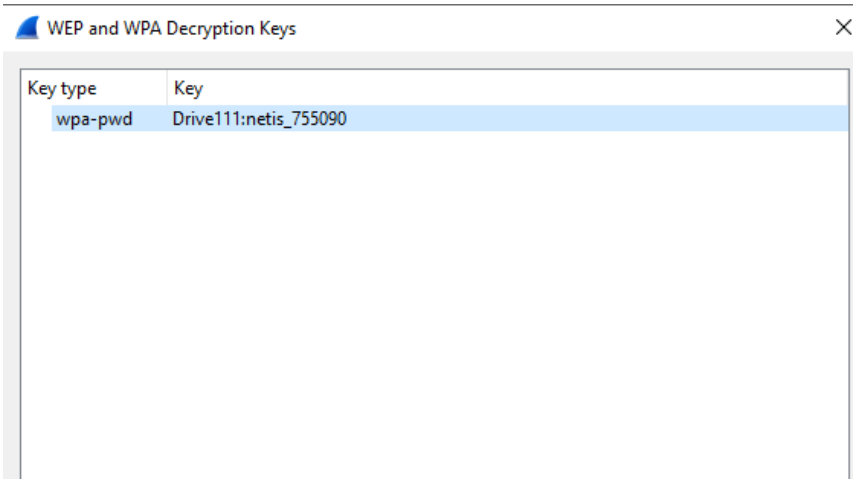

Рисунок 3.3 — Впровадження змін до протоколу IEEE 802.11

Впроваджені зміни дали свої результати і пакети стали доступні для більше детального аналізу, для прикладу, до введення ключів шифрування більш дательну інформацію не надавалася, замість неї повідомлялося наступне: Encrypted data (WPA2/CCMP) : 84c4a92d0b60eaf29a0344a561ae71c3ef8fc58d28a98c51. Тепер же інформації стало значно більше і вона включає усі необхідні дані про пакет, накшталт встановлених флагів.

3.2.1 Захоплення трафіку

Перед проведенням тестування на продуктивність мережі чи виявлення підозрілої активності, необхідно зібрати базовий мережевий трафік, що буде використано як контрольну точку для подальшого аналізу. Тому слід захоплювати трафік протягом щонайменше 15 хвилин в період типового навантаження на систему. Це дасть змогу зафіксувати звичайну поведінку мережі та додатків, що працюють на ній.

Після 15 хвилин захоплення трафіку, в результаті було передано по мережі 25 тисяч пакетів. Результат можна побачити на рисунку 3.4.

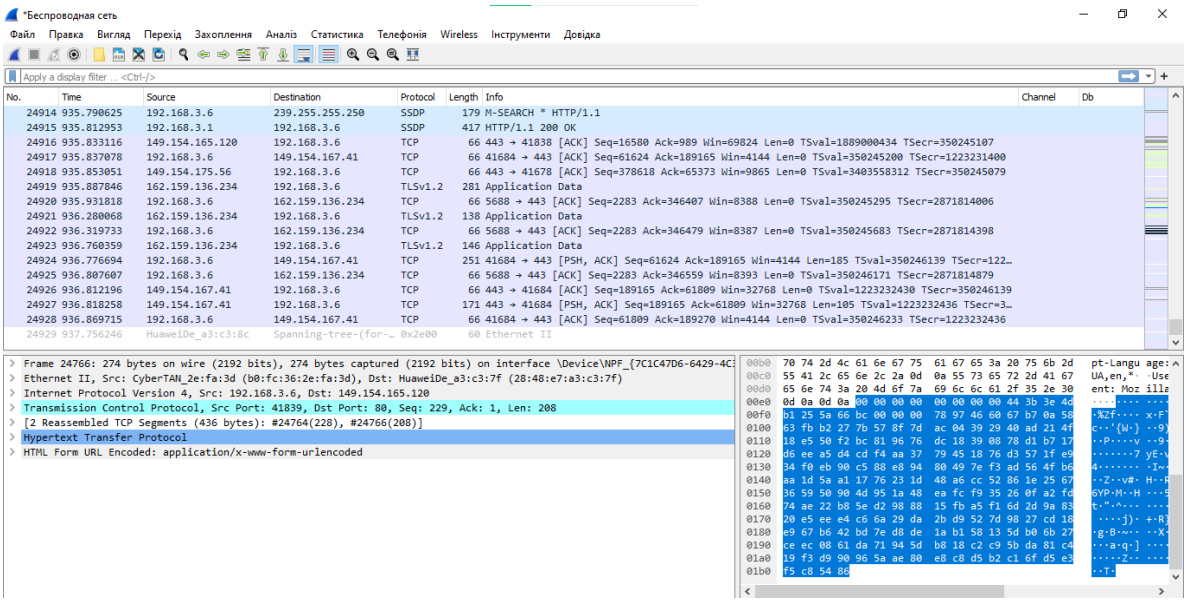

Рисунок 3.4 — Результати захоплення

Наступним кроком буде збереження цих пакеті в окремий файл аби при незапланованих ситуаціях не втратити результати. Щоб зберегти файл потрібно натиснути на іконку зберження, що зображена на рисунку 3.4.1.

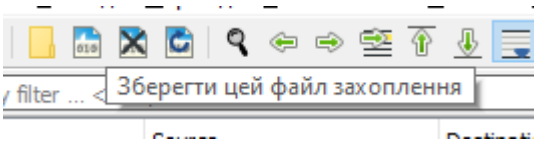

Рисунок 3.4.1 — Збереження сесії в окремий файл захоплення

Збережені файли мають формат .pcapng, він є більш новим і покращеним форматом, розробленим для заміни старого формату .pcap.На відміну від старого, pcapng отримав покращення

а) підтримка необмеженого розміру файлів, на відміну від обмеженого 32-бітним значенням у форматі .pcap;

б) кращий спосіб зберігання даних завдяки секційованому дизайну. Це полегшує додавання, видалення та перегляд окремих секцій без необхідності читати весь файл;

в) покращена підтримка метаданих, таких як коментарі, назви інтерфейсів, геолокаційні дані тощо;[9,14,22,23]

г) кращі можливості стискання даних для економії дискового простору;

3.3 Аналіз broadcast пакетів у захопленому трафіку

Широкомовні (broadcast) запити є невід'ємною частиною роботи багатьох мережевих протоколів і служб. Вони використовуються для виявлення пристроїв у мережі, розподілу IP-адрес, оновлення кешу ARP та виконання інших важливих функцій. Однак через свою природу broadcast-трафік може стати джерелом різноманітних атак і загроз безпеці мережі.

В сесії, що була захоплена, був зроблений такий аналіз. Для цього, в поле фільтрів, потрібно було ввести наступне: eth.dst == ff-ff-ff-ff-ff-ff. Це дало змогу відфільтрувати серед усіх пакетів ті, які мали адрес призначення - broadcast. Результат фільтрування зображений на рисунку 3.5.

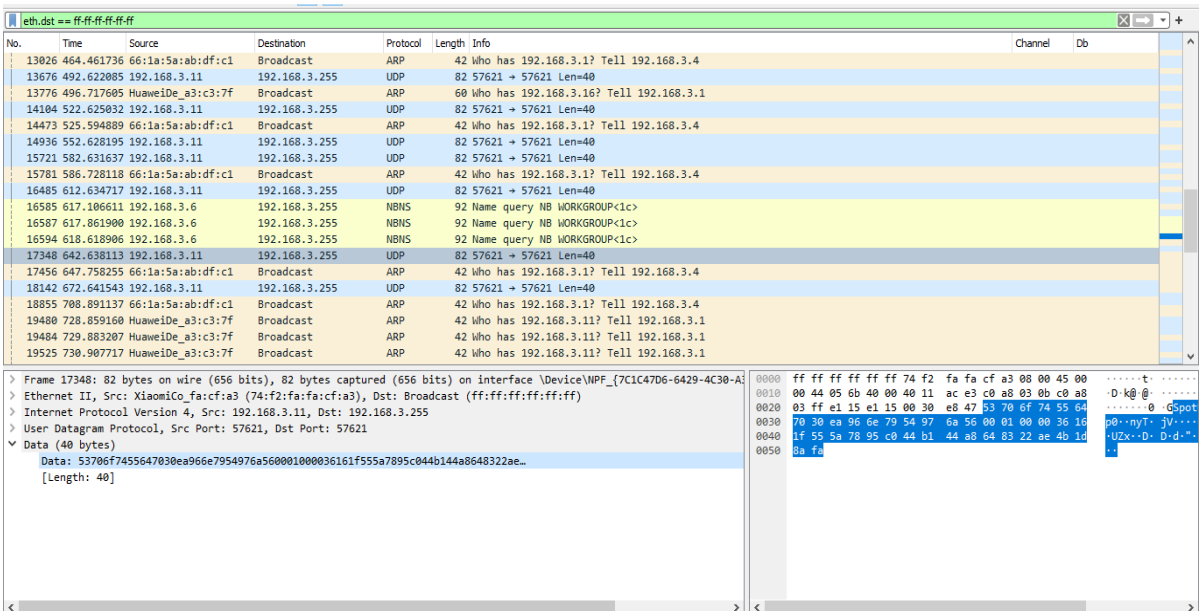

Рисунок 3.5 — Відфільтрований трафік

Базуючись на отриманому наборі даних, можна зробити висновок, що в цілому робота широкомовного (broadcast) трафіку є нормальною і не виявлено жодних явних аномалій. Більшість спостережуваних широкомовних запитів пов'язані зі стандартними мережевими операціями та протоколами. Зокрема, було помічено наступне:

— значна частина широкомовних запитів генерувалася мережевим маршрутизатором для періодичного оновлення записів ARP-кешу. Це типова і очікувана поведінка, необхідна для коректного функціонування мережі;

— окремі пристрої, такі як 192.168.3.11, відправляли широкомовні UDP-запити, ймовірно, для взаємодії з певними мережевими додатками або службами. Такі запити є нормальними і не свідчать про наявність загроз;

— було виявлено використання застарілого протоколу NetBIOS Name Service (NBNS) пристроєм з IP-адресою 192.168.3.6. Це може бути пов'язано з роботою старих додатків, які все ще покладаються на NBNS для розподілу імен у локальній мережі;

Спостережувана широкомовна активність відповідає звичайному функціонуванню мережевих протоколів та додатків в даному середовищі.

3.4 Аналіз icmp протоколу захопленого трафіку

ICMP(Internet Control Message Protocol) — це допоміжний протокол в стеку TCP/IP, який використовується для відправки повідомлень про помилки та інших службових даних про комунікацію в IP-мережах. Він один з найтовариськіших протоколів, який може розказати дуже багато. У захопленому трафіку був проведений аналіз цього протоколу, увівши у строку фільтрів icmp, отримано результати, що зображено на рисунку 3.6.

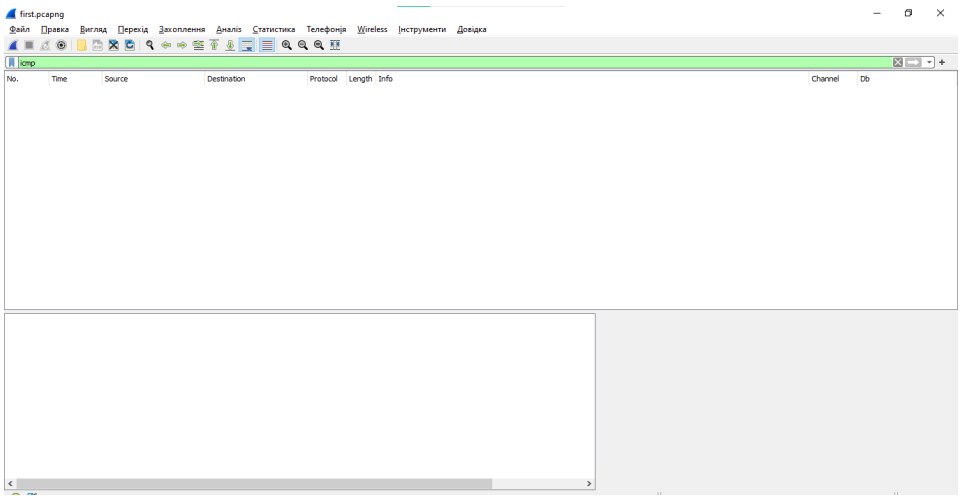

Рисунок 3.6 — Відфільтровані пакети icmp протоколу

Можна побачити, що icmp пакети не були захоплені. Відсутність ICMP пакетів у захопленому трафіку може вказувати на кілька можливих сценаріїв:

а) не проводилося сканування мережі зовнішніми зловмисниками. ICMP запити, такі як "ping", часто використовуються для виявлення активних вузлів перед потенційними атаками. Відсутність цих пакетів може свідчити про те, що зовнішнє сканування не здійснювалося під час захоплення трафіку;

б) не проводилося тестування досяжності пристроїв командою ping. Адміністратори зазвичай використовують утиліту ping для перевірки зв'язності з хостами в мережі. Брак ICMP Echo Request/Reply пакетів може означати, що такі перевірки не виконувалися в проміжок часу захоплення трафіку;

Це хороший результат. Мережа не сканувалася зловмисником та не повідомляла про помилки під час передачі пакетів. При появі ICMP пакетів варто звертати увагу на код, що передається протоколом — він розкаже дуже багато інформації. Приклад коду, що передає icmp, зображений на рисунку 3.6.1.

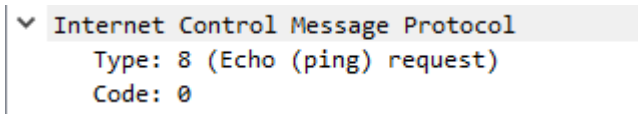

Рисунок 3.6.1 — ICMP повідомляє тип та код пакету

В цьому випадку, тип 8 — операція exo(пінг). Код 0 — exo запит. Така сама ситуація і для типу 0 - операція exo(пінг), код 0 - exo відповідь.

Інші значення типів та їх кодів:

Типи 1,2,7 — не використовуються.

Типи 4, 6, 15-18, 30-39 — застаріли.

Типи 19,20-29, 42-252, 255 — зарезервовані.

Типи 41, 253,254 — експериментальні.

Тип 3 — Пункт призначення недоступний. Коди:

— 0: Мережа недосяжна;

— 1: Хост недосяжний;

— 2: Протокол недосяжний;

— 3: Порт недосяжний;

— 4: Необхідна фрагментація, але встановлений прапорець її заборони (DF);

— 5: Неправильний маршрут від джерела;

— 6: Мережа призначення невідома;

— 7: Хост призначення невідомий;

— 8: Хост джерела ізольований;

— 9: Мережу адміністративно заборонено;

— 10: Хост адміністративно заборонений;

— 11: Мережа недоступна для ToS;

— 12: Хост недоступний для ToS;

— 13: Комунікації адміністративно заборонені;

— 14: Порушення порядку переваги вузлів;

— 15: Активно відсікання порядку пріоритету ;

Тип 5 — Перенаправлення. Коди:

— 0: Перенаправлення пакетів у мережу;

— 1: Перенаправлення пакетів до вузла;

— 2: Перенаправлення для кожного типу обслуговування (ToS);

— 3: Перенаправлення пакета до вузла для кожного типу обслуговування;

Тип 9. Код 0 — оголошення маршрутизатора.

Тип 10. Код 0 — запит маршрутизатора.

Тип 11 — Час життя дейтаграми сплив. Коди:

— 0: Час життя пакета (TTL) закінчився під час транспортування;

— 1: Час життя пакета закінчився під час збирання фрагментів;

Тип 12 — Неправильний параметр (проблема з параметрами дейтаграми: помилка в IP-заголовку або відсутня необхідна опція). Коди :

— 0: Покажчик вказує на помилку;

— 1: Відсутня необхідна опція;

— 2: Некоректна довжина;

Тип 13 Код 0 — Запит мітки часу.

Тип 14 Код 0 — Відповідь із міткою часу.

Тип 40 — Photuris. Коди:

— 0: Зарезервовано;

— 1: Невідомий індекс параметрів безпеки;

— 2: Параметри безпеки правильні, але сталася помилка аутентифікації;

— 3: Потрібна перевірка автентичності;

— 4: Потрібна перевірка автентичності;

— 5: Потрібна авторизація;[5]

3.5 Аналіз DNS протоколу захопленого трафіку

DNS(Domain Name System) — протокол, що грає важливу роль, забезпечуючи перетворення доменних імен на IP адреси, без нього неможливий доступ до більшості веб ресурсів та робота багатьох мережевих додатків.

Ретельний моніторинг DNS трафіку допомагає ідентифікувати проблеми продуктивності, такі як надмірна кількість DNS запитів, повторні запити до одного домену чи збої в резолвінгу DNS. Це дозволяє своєчасно виявляти й усувати подібні проблеми, що можуть негативно позначатися на роботі додатків та користувацькому досвіді.

Окрім цього, аналіз DNS відіграє важливу роль у забезпеченні дотримання політик безпеки організації. Багато компаній обмежують або забороняють доступ до певних категорій веб-сайтів, і DNS трафік є ключовим індикатором для моніторингу виконання таких обмежень.

У захопленій сесії був зроблений також і цей аналіз. Для цього було введено в строку фільтрів слово dns і Wireshark показав усі пакети dns запитів. Це зображено на рисунку 3.7.

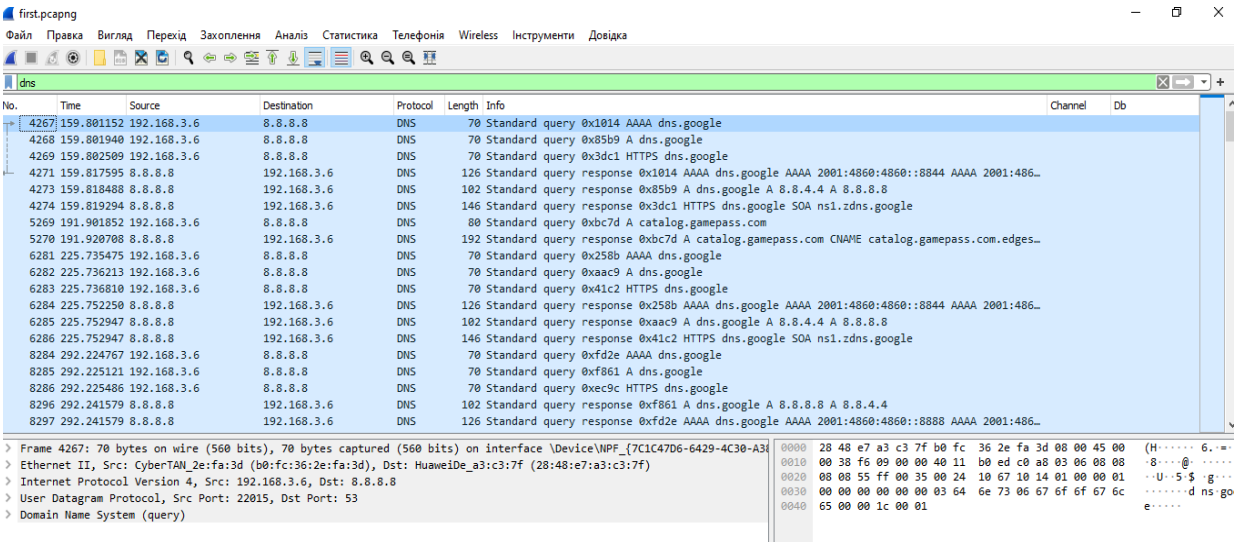

Рисунок 3.7 — Пакети DNS аналізуємого трафіку

Провівши аналіз, було зроблено висновок, що адрес 192.168.3.6 є єдиним у мережі, хто надслиає DNS пакети. Вони включаються в себе легітимні звернення до 8.8.8.8 — це адреса публічного рекурсивного резолвера Google. Це досить поширена практика, особливо в домашніх мережах чи невеликих офісах. Публічні резолвери зазвичай забезпечують швидкі та надійні послуги з розв'язання DNS запитів.

Іншою підозрілою активності не було помічено. Але слід буде звернути увагу, якшо з’являться пакети, в яких присутні запити до DNS серверів, що мають дуже довгий та незрозумілий адрес накшталт zxyaskbot1.com або afd7e3tdf8ry6qwdcx9zbyp3o .com — це може бути ознакою DNS тунелювання, особливо якщо передані пакети на такі адреса мають великий розмір. Ця атака має на меті обійти заборони firewall’а комп’ютера жертви та таємно виводити дані, віддалено управляти пристроєм за допомогою RAT(Remote Access Trojan).

Також, уваги потребує активність адресів, які надсилають запити до DNS серверу і отримують відповіді з кодом 2(помилка сервера), це може свідчити про намагання отруїти кеш DNS сервера. Зловмисник намагатиметься підробити DNS-відповідь таким чином, щоб DNS-сервер сприйняв її як легітимну і закешував хибну відповідь. Після успішної підробки закешована хибна відповідь

буде повертатись легітимним клієнтам замість справжньої IP-адреси, ефективно перенаправляючи їх на шкідливий ресурс.

Підсумовуючи, аналіз DNS трафіку є критично важливим для забезпечення безпеки та стабільної роботи мережі. В досліджуваній мережі не було виявлено підозрілої активності з протоколом DNS.

3.6 Аналіз TCP протоколу захопленого трафіку

TCP(Transmission Control Protocol) — є одним з основних протоколів стека TCP/IP, який забезпечує надійну передачу даних у мережі. На відміну від User Datagram Protocol (UDP), TCP гарантує доставку пакетів до місця призначення у правильному порядку, без втрат і дублікатів. Це досягається завдяки низці механізмів, вбудованих у протокол.

Щоб встановити TCP з'єднання, відправник та одержувач проходять через процес "рукостискання" із обміном спеціальними пакетами. Після цього дані передаються через мережу у вигляді сегментів, кожен з яких містить порядковий номер, номер підтвердження, розмір вікна та інші поля заголовка TCP. Одержувач відправляє пакети підтвердження (ACK), сповіщаючи відправника про успішний прийом даних. [9,11,14,15]

Для отримання пакетів tcp потрібно ввести це слово в строку фільтрів у Wireshark, результат введення зображений на рисунку 3.8.

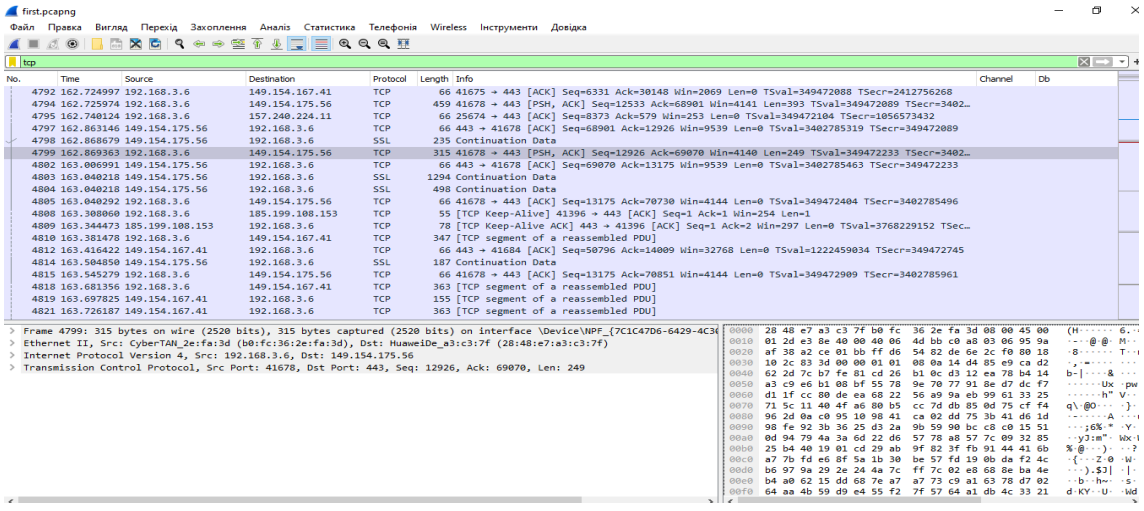

Рисунок 3.8 — Відфільтровані пакети TCP

На основі отриманих даних, можна зробити висновок, що TCP протокол працює без нарікань — пакети доставляються без затримок та великої кількості помилок. Окрім цієї перевірки, слід зробити наступну: перевірити, чи не надсилає якийсь вузол в мережі пакети з флагом SYN(synchronize) на різні порти одного користувача. Це може свідчити про те, що перший вузол намагається знайти відкриті порти другого вузла, можливо, задля атаки на відкритий порт. Вона може бути різною: від DDoS атаки для виснаження системи та відмови в обслуговуванні і до виконання віддаленого коду. З недавніх критичних вразливостей можна виділити наступні: Spring4Shell (CVE-2022-22965), яка була знайдена у фреймворку Spring для Java та зловмисники таким чином могли виконувати віддалений код; та Log4Shell (CVE-2021-44228), вразливість бібліотеки Apache Log4j, що також дозволяла виконувати віддалений код.

Отже, для того, щоб перевірити мережу на наявність такої підозрілою поведінки, потрібно у строку фільтрів Wireshark вставити таку строку: ip.src==192.168.1.0/24 && tcp.flags.syn ==1, вона покаже усі пакети пристроїв мережі 192.168.1.0/24, що були відправлені з флагом syn, Результат введеної команди зображений на рисунку 3.8.1.

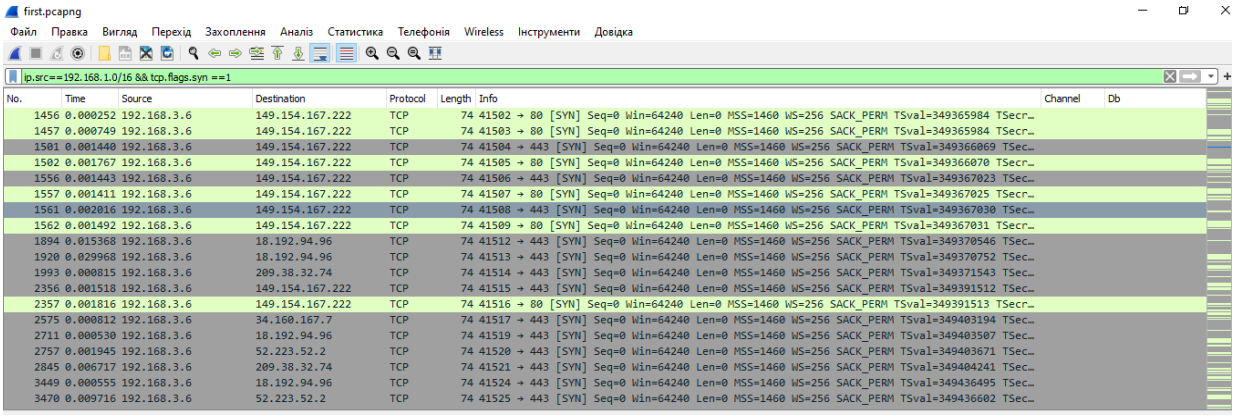

Рисунок 3.8.1 — TCP пакети з флагом SYN

Отримавши дані, не було помічено підозрілої активності вузлів. Єдиний хост що надсилав такі пакети, був 192.168.3.6. Але передача пакетів була з використанням destination port 443, що є цілком нормальною активністю.

Таким чином, аналіз TCP пакетів є важливим як і для продуктивності мережі, так і для її безпеки.

3.7 Аналіз IoT протоколів

В попередніх розділах було розглянуто аналіз протоколів, пов'язаних з персональними комп'ютерами. Можна сказати, що комп’ютер не відноситься до IoT, але він стосується однієї з ланок архітектури IoT - вона зображена на рисунку 3.9.

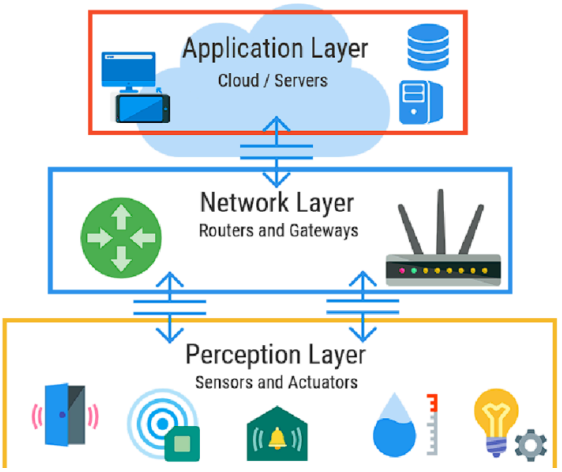

Рисунок 3.9 — Типова архітектура IoT

Оскільки усі ці слої архітектури пов’язані між собою, доцільно проводити аналіз мережевого трафіку як на рівня додатків, якщо на ПК встановлені програми для візуалізації даних, управління IoT-системою, налаштування пристроїв або інші користувацькі інтерфейси**,** так і на мережевому рівні якщо ПК виступає в ролі шлюзу або точки доступу для підключення IoT-пристроїв до мережі, або в ролі адміністратора. Аналіз саме цієї ланки із поєднанням SIEM систем дозволить попередити про підозрілу активність у організації із метою компроментування та захоплення вузлів, забезпечити продуктивність мережі.

Щодо популярних IoT протоколів таких як LoRaWAN, MQTT тощо, які відносяться до рівня сприймання(perception), їх аналіз може бути корисним під час налаштування датчиків та інших “речей‘ в реальному часі на предмет доставки

потрібних даних у пакеті і 100% доставки пакету загалом, налаштування корректного з’єднання. Wireshark підтримує більшість IoT протоколів, зокрема LoRaWAN, MQTT та CoAP. Це зображено на рисунках 3.9.1, 3.9.2, 3.9.3.

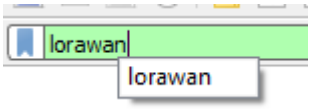

Рисунок 3.9.1 — Наявність підтримки протоколу LoRaWAN у Wireshark

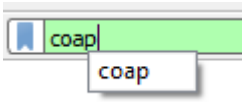

Рисунок 3.9.2 — Наявність підтримки протоколу CoAP у Wireshark

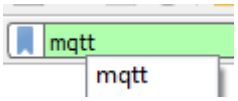

Рисунок 3.9.3 — Наявність підтримки протоколу CoAP у Wireshark

Отже, в IoT системах аналіз трафіку різних протоколів дасть детальну інформацію про стан мережі(доставку пакетів чи можливі проблеми із їх отриманням), можливі атаки. Оскільки IoT архітектура складається з трьох слоїв, варто, при наріканнях на проблеми у мережі, проводити аналіз усіх трьох рівнів.

ВИСНОВКИ

У цій кваліфікаційній роботі була затронута важлива тема аналізу трафіку в мережах Інтернету Речей (IoT). Хоча в практичній частині використовувалася звичайна домашня мережа, це дозволило продемонструвати переваги та широкі можливості аналізатора трафіку Wireshark для дослідження різноманітних мережевих протоколів та виявлення потенційних проблем чи загроз. Отриманий досвід може бути застосований при роботі з реальними IoT-системами.

Дослідження таких протоколів як ICMP, TCP, DNS та окремо — broadcast запитів, дозволило зрозуміти, що доставка пакетів була в цілому ефективною та стабільною в досліджуваній мережі, підозрілих активності пристроїв помічено не було. Щодо протоколів MQTT, CoAP, LoRaWAN, ZigBee, які стосуються виключно архітектури IoT, їх аналіз у межах цієї роботи був обмеженим. Але Wireshark підтримує ці протоколи, що може бути корисним при первинному налаштуванні IoT-пристроїв або при необхідності переглянути детальний зміст даних, що передаються цими протоколами.

Загалом, результати кваліфікаційної роботи продемонстрували ефективність Wireshark як універсального інструменту для всебічного моніторингу та аналізу мережевого трафіку в різноманітних середовищах, включно з Інтернетом Речей.

ПЕРЕЛІК ПОСИЛАНЬ

1. Buyya, R., Dastjerdi, A.V. Internet of Things: Principles and Paradigms. – Published by Todd Green, 2016. pp.5-15,183-190.
2. How MQTT Works [Електронний ресурс] : [Інтернет-сайт]. - Режим доступу: http://www.steves-internet-guide.com/mqtt-works/ (дата звернення 1.05.2024). - Назва з екрана.
3. CoAP: A complete guide [Електронний ресурс] : [Інтернет-сайт]. - Режим доступу: https://nonamesecurity.com/learn/what-is-constrained-application-protocol-coap/ (дата звернення 2.05.2024). - Назва з екрана
4. What are LoRa and LoRaWAN? [Електронний ресурс] : [Інтернет-сайт]. - Режим доступу: https://www.thethingsnetwork.org/docs/lorawan/what-is-lorawan/ (дата звернення 2.05.2024). - Назва з екрана
5. ICMP Parameters [Електронний ресурс] : [Інтернет-сайт]. - Режим доступу: https://www.iana.org/assignments/icmp-parameters/icmp-parameters.xhtml (дата звернення 10.05.2024). - Назва з екрана
6. Goyal, Piyush, and Anurag Goyal. "Comparative study of two most popular packet sniffing tools-Tcpdump and Wireshark."2017 9th International Conference on Computational Intelligence and Communication Networks (CICN). IEEE, 2017.
7. Hamid, Isredza Rahmi A., et al. "Network monitoring system to detect unauthorized connection." Acta Electronica Malaysia 1.2, 2017.
8. Bhandari, Aishwarya, et al. "Packet Sniffing and Network Traffic Analysis Using TCP—A New Approach." Advances in Electronics, Communication and Computing. Springer, Singapore, 2018
9. Sanders, Chris. Practical packet analysis: Using Wireshark to solve real-world network problems. No Starch Press, 2017. pp.12-29, 102-120.
10. Saxena, Praful, and Sandeep Kumar Sharma. "Analysis of Network Traffic by using Packet Sniffing Tool: Wireshark." International Journal of Advance Research, Ideas and Innovations in Technology 3.6, 2017.

11. Dr. Sidnie Feit. “TCP/IP: Architecture, Protocols, Implementation”. - Published by McGraw-Hill, 1996. pp.16-38.
12. Höller, Jan, et al. "From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence." Academic Press, 2014.
13. Minoli, Daniel. "Building the Internet of Things with IPv6 and MIPv6: The Evolving World of M2M Communications." John Wiley & Sons, 2013.
14. Orebaugh, Angela, et al. "Wireshark & Ethereal Network Protocol Analyzer Toolkit." Syngress, 2006. pp.65-95.
15. Kurose, James F., and Keith W. Ross. "Computer Networking: A Top-Down Approach." Pearson, 2017. pp.52-68, 601-627.
16. Bormann, Carsten, and Zach Shelby. "CoAP: An Application Protocol for Billions of Tiny Internet Nodes." IEEE Internet Computing 16.2, 2012.
17. Vucinic, Marjan, et al. "OSCAR: Object Security Architecture for the Internet of Things." Ad Hoc Networks, 2014.
18. Roman, Rodrigo, Jianying Zhou, and Javier Lopez. "On the features and challenges of security and privacy in distributed Internet of Things." Computer Networks 57.10, 2013.
19. Sicari, Sabrina, et al. "Security, privacy and trust in Internet of Things: The road ahead." Computer networks 76, 2015.
20. Kreibich, Christian, et al. "Network Security Monitoring." Addison-Wesley, 2014. pp.17-45, 122-149.
21. Lammle, Todd. "CompTIA Network+ Study Guide: Exam N10-007." John Wiley & Sons, 2018.
22. Chappell, Laura. "Wireshark Network Analysis: The Official Wireshark Certified Network Analyst Study Guide." Laura Chappell University, 2017. pp.35-62.
23. Bejtlich, Richard. "The Practice of Network Security Monitoring: Understanding Incident Detection and Response." No Starch Press, 2013. pp.89-117.
24. Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of things: A survey on enabling technologies, protocols, and applications. IEEE Communications Surveys & Tutorials, 17(4), 2347-2376.

25. Olsson, J. (2014). 6 LoWPAN demystified. Texas Instruments, 13, 6-13.
26. Shafiq, M. Z., Ji, L., Liu, A. X., Pang, J., & Wang, J. (2012). A first look at cellular machine-to-machine traffic: insights and analysis. In Proceedings of the 2012 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems (pp. 287-298).
27. Zillner, T., & Strobl, S. (2015). Zigbee exploited: The good, the bad and the ugly. Black Hat Europe, 14.
28. Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. Future generation computer systems, 29(7), 1645-1660.
29. Shelby, Z., & Bormann, C. (2009). 6LoWPAN: The wireless embedded Internet (Vol. 43). John Wiley & Sons.
30. Atzori, L., Iera, A., & Morabito, G. (2010). The internet of things: A survey. Computer networks, 54(15), 2787-2805.