**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**

НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: **«ДОСЛІДЖЕННЯ МЕТОДІВ АНАЛІЗУ ТА ВІЗУАЛІЗАЦІЇ СКЛАДНИХ МОДЕЛЕЙ МЕРЕЖЕВОГО ТРАФІКУ З МЕТОЮ ПОКРАЩЕННЯ УПРАВЛІННЯ МЕРЕЖЕЮ, УСУНЕННЯ НЕСПРАВНОСТЕЙ ТА ВИЯВЛЕННЯ АНОМАЛІЙ»**

на здобуття освітнього ступеня магістр
за спеціальності 123 Комп'ютерна інженерія
(код,найменування спеціальності)
освітньо-професійної програми Комп'ютерні системи та мережі
(назва)

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

_____ Євген ДУРНЄВ
(підпис) *(ім'я, ПРІЗВИЩЕ здобувача)*

Виконав: здобувач вищої освіти гр.КСДМ-61
Євген ДУРНЄВ
(ім'я, ПРІЗВИЩЕ)

Керівник: Наталія ЛАЩЕВСЬКА
к.т.н., доцент *(ім'я, ПРІЗВИЩЕ)*

Рецензент: _____
науковий ступінь, вчене звання *(ім'я, ПРІЗВИЩЕ)*

Київ 2023

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут інформаційних технологій

Кафедра Комп'ютерної інженерії
Ступінь вищої освіти «Магістр»

Спеціальність 123 Комп'ютерна інженерія
Освітньо-професійна прорама Комп'ютерні системи та мережі

ЗАТВЕРДЖУЮ
Завідувач кафедрою Комп'ютерної інженерії
Наталія ЛАЩЕВСЬКА
(ім'я, ПРІЗВИЩЕ)
"____" _____2023 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дурнєву Євгену Станіславовичу
(прізвище, ім'я, по батькові здобувача)

1.Тема кваліфікаційної роботи: Дослідження методів аналізу та візуалізації складних моделей мережевого трафіку з метою покращення управління мережею, усунення несправностей та виявлення аномалій
керівник роботи Наталія ЛАЩЕВСЬКА к.т.н., доцент
(ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)
затверджені наказом Державного університету інформаційно-комунікаційних технологій від "19" 10 2023 р. №145
2. Строк подання кваліфікаційної роботи
3. Вихідні дані кваліфікаційної роботи:
3.1. Інтернет ресурси стосовно засобів моніторингу мережевого трафіку.
3.2. Методи аналізу та візуалізації моделей мережевого трафіку.
3.3. Науково-технічна література.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):
4.1. Аналіз та моніторинг мережевого трафіку.
4.2. Засоби моніторингу мережевого трафіку.
4.3. Візуалізація даних Netflow на основі графіків.
4.4. Camnep: система виявлення мережевих вторгнень на основі агентів.
5.Перелік ілюстраційного матеріалу: *презентація*

6. Дата видачі завдання “19” жовтня 2023р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	**Примітка**
1.	Підбір технічної літератури	19.10-28.10.2023р.	Виконано
2.	Аналіз та моніторинг мережевого трафіку	29.10-08.11.2023р.	Виконано
3.	Засоби моніторингу мережевого трафіку	09.11-21.11.2023р.	Виконано
4.	Візуалізація даних Netflow на основі графіків	22.11-30.11.2023р.	Виконано
5.	Camnep: система виявлення мережевих вторгнень на основі агентів	01.12-16.12.2023р.	Виконано
6.	Оформлення роботи, висновки	17.12-20.12.2023р.	Виконано
7.	Розробка демонстраційного матеріалу, доповідь	21.12-28.12.2023р.	Виконано

Здобувач вищої освіти Євген ДУРНЄВ

(підпис) *(ім’я, ПРІЗВИЩЕ)*

Керівник кваліфікаційної роботи Наталія ЛАЩЕВСЬКА

(підпис) *(ім’я, ПРІЗВИЩЕ)*

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття ступеня магістр: 75 стор., 23 рис., 2 табл., 25 джерел.

Мета роботи – Покращення управління мережею, усунення несправностей та виявлення аномалій.

Об'єкт дослідження – Методи аналізу та візуалізації моделей мережевого трафіку.

Предмет дослідження – Мережевий трафік.

Короткий зміст роботи Досліджено процес аналізу трафіку. На основі цього дослідження проведено порівняльний аналіз існуючих засобів аналізу трафіку. Реалізовано базовий функціонал пакетного аналізатора для деяких мережевих протоколів з унікальною реалізацією методів створення відбитків. В даній дипломній роботі представлено інноваційний підхід до обробки та візуалізації даних NetFlow, розроблено інструмент підтвердження концепції під назвою NetFlow Visualizer, який зараз пропонується як плагін для датчиків NetFlow. Також у даній роботі представлено мультиагентний фреймворк, який дозволяє інтегрувати декілька існуючих методів аналізу поведінки в мережі.

КЛЮЧОВІ СЛОВА: АНАЛІЗ, ВІЗУАЛІЗАЦІЯ, МЕРЕЖЕВИЙ ТРАФІК, УПРАВЛІННЯ МЕРЕЖЕЮ, НЕСПРАВНОСТІ, АНОМАЛІЇ, ЗАСОБИ МОНІТОРИНГУ МЕРЕЖЕВОГО ТРАФІКУ

ABSTRACT

The text part of the qualification work for obtaining a master's degree: 75 pages, 2 table, 23 figures, 25 sources.

The purpose of the work is improving network management, troubleshooting, and anomaly detection.

The object of research is Methods of analysis and visualization of network traffic patterns.

The subject of research is Network traffic.

Summary of the work: The process of traffic analysis has been studied. On the basis of this study, a comparative analysis of existing traffic analysis tools was carried out. The basic functionality of a packet analyzer for some network protocols with a unique implementation of fingerprinting methods has been implemented. This thesis presents an innovative approach to NetFlow data processing and visualization, developed a proof-of-concept tool called NetFlow Visualizer, which is now offered as a plug-in for NetFlow sensors. Also, this work presents a multi-agent framework that allows for the integration of several existing methods of network behavior analysis.

KEY WORDS: ANALYSIS, VISUALIZATION, NETWORK TRAFFIC, NETWORK MANAGEMENT, FAULTS, ANOMALIES, NETWORK TRAFFIC MONITORING TOOLS

ЗМІСТ

ВСТУП

Активний розвиток інформаійних технологій робить їх невідємною частиною сьогодення. Інформаційні системи сьогодні широко експлуатуються як і в комерційних, так і в державних. Їх взаємодія між собою відбувається через мережі.

Спостерігається стрімкий зріст об'єму мережевого трафіку, що провокує ускладнення його структури. Аналіз трафіку стає все більш потрібним в областях контроля та управління, оптимізації, захисту від шкідливого втручання. Моніторинг трафіку також необхідний, щоб більш ефективно діагностувати і вирішувати проблеми, коли вони постають, таким чином не доводячи різні мережеві сервіси до простоя впродовж тривалого часу. Доступно багато різних інструментів, які дозволяють допомогти адміністраторам з моніторингом та аналізом мережевого трафіку. Дана робота висвітлює методи моніторингу мережевого трафіку та надає порівняльну хавактеристику засобів мережевого трафіку.

Моніторинг мережі — складне завдання, що вимагає великих сил та витрат, яке є невід’ємною складовою повсякденного життя мережевих адміністраторів. Адміністратори постійно прагнуть підтримати безперебійну роботу своєї мережі. Якщо мережа «впаде» хоча б на невеликий період часу, продуктивність в компанії скоротиться і (в разі організацій які надають державні послуги) сама можливість надання основних послуг буде поставлена під загрозу. У зв'язку з цим адміністраторам необхідно проводити постійний аналіз мережевого трафіку, стежити за його рухом і продуктивністю на всій мережі і перевіряти, чи з'явилися в ній проломи в безпеці. В випадках моніторингу при виявленні проблеми після її вирішення постає питання тестування коректності та цілісної роботи системи для цього в допомогу приходить моделювання трафіка, що дозволяє змоделювати трафік для здійснення аналізу, щоб зробити висновки можливих вразливостей

системи, її продуктивності, дослідженні різних протоколів, алгоритмів або топографій мереж.

В даній дипломній роботі представлено інноваційний підхід до обробки та візуалізації даних NetFlow. Даний метод візуалізації на основі графів заповнює прогалину між візуалізацією високоагрегованої інформації, представленої у вигляді діаграм, і детальною інформацією, представленою у вигляді лог-файлів. У представленому методі візуалізації вузли графа означають мережеві пристрої, а орієнтовані ребра представляють зв'язок між цими пристроями. В роботі також представлено утилізацію зовнішніх джерел даних (DNS, імена портів тощо), що допомагає представити дані NetFlow більш інтуїтивно зрозумілим чином. Таким чином, цей підхід є дуже природним як для мережевих адміністраторів, так і для неспеціалістів. На основі цих методів було розроблено інструмент підтвердження концепції під назвою NetFlow Visualizer, який зараз пропонується як плагін для датчиків NetFlow.

В роботі представлено прототип агентної системи виявлення вторгнень, призначеної для розгортання у високошвидкісних магістральних мережах. Основним внеском системи є інтеграція декількох методів виявлення аномалій за допомогою колективного моделювання довіри в групі агентів, що співпрацюють між собою, кожен з яких використовує певний алгоритм виявлення. Аномалії використовуються як вхідні дані для моделювання довіри. На цьому етапі кожен агент визначає довіру до потоку на основі агрегованих аномалій. Агрегація виконується за допомогою розширених моделей довіри, які моделюють довіру до узагальнених локалізованих ідентифікаторів, представлених набором спостережуваних ознак. Система базується на статистиці трафіку у форматі NetFlow, що збирається спеціальними мережевими картами з апаратним прискоренням, і здатна здійснювати спостереження за гігабітними мережами в режимі реального часу.

1 АНАЛІЗ ТА МОНІТОРИНГ МЕРЕЖЕВОГО ТРАФІКУ

Аналіз мережевого трафіку набуває все більшу актуальність у зв’язку з розвитком мережевих технологій, збільшенням обсягу даних, переданих по мережі, впровадженням великої кількості нових мережевих протоколів (в тому числі закритих). В якості основних областей практичного застосування можна виділити наступні:

- виявлення проблем в роботі мережі;
- тестування (налагодження мережевих протоколів);
- запобігання мережевих атак;
- класифікація трафіку.

Поглиблений аналіз трафіку в даний час є однією з основних технологій при вирішенні практичних завдань пов'язаних із забезпеченням безпеки мереж, оптимізацією пропускної здатності каналів передачі даних, контролем якості зв'язку та інших. Якісне рішення згаданих завдань вимагає максимальної глибини розбору мережевого трафіку і надання результатів розбору в зручній формі.

Можливість аналізу трафіку на рівні контенту додатків є у вузького кола закритих комерційних систем таких фірм, як Checkpoint і Cisco. Ці системи не дозволяють користувачам вільно розширювати перелік підтримуваних мережевих протоколів, їх закритість ускладнює інтеграцію. У той же час існуючі вільно поширювані інструменти не забезпечують належної якості проведеного аналізу, внаслідок чого з їх допомогою не завжди вдається отримати дані протоколів прикладного рівня.

1.1 Аналіз мережевого трафіку: управління через АРМ

Аналіз трафіку — необхідна область для управління, контролю і оптимізації, а також для захисту систем. Сучасні корпоративні мережі стають все

більш складними завдяки додаванню численних персональних пристроїв, безлічі хмарних додатків, точок доступу, віртуальних серверів тощо. Команди адміністраторів повинні ретельно керувати пропускною здатністю та якістю обслуговування, впроваджувати балансири навантажень для високої доступності, створювати надмірне середовище для резервного копіювання та забезпечувати синхронізовану роботу декількох компонентів у гібридному середовищі.

Аналіз можна здійснювати на рівні окремого пакету, в контексті роботи додатку або в діагностиці складних систем.

Для якісного управління додатком варто розібрати, що собою являє APM (Application Performance Management).

APM — це процес, який забезпечує зв’язок між роботою програми та всієї структури, в яку додаток входить. Також виміряє, оцінює і документує основні зв’язки додатку. Тобто в додатку ми повинні не лише підраховувати час роботи або використаний трафік, а у разі помилок системи повинні знайти їх причину та знайти методи їх усунення.

Також необхідно враховувати, що всі процеси повинні відбуватись одночасно для мережевого обладнання, каналів зв’язку, серверів тощо.

1.1.1 Види моніторингу APM

На базі процесу APM можна виділити види моніторингу:

- на основі метрик додатків - кілька інструментів використовують різні показники сервера та додатків і називають це APM. У кращому випадку вони можуть повідомити, скільки запитів отримує програма та потенційно, які URL-адреси можуть бути повільними. Оскільки вони не займаються профілюванням рівня коду, вони не можуть сказати чому.

- ефективність на рівні коду - Stackify Retrace, New Relic, AppDynamics і Dynatrace - типовий тип APM, на основі кодування профілю та відстеження транзакцій.

- на основі мережі - Extrahop використовує термін APM стосовно їх здатності вимірювати продуктивність додатків на основі мережевого трафіку. Існує ціла категорія товарів під назвою NPM, яка зосереджена на цьому типі рішень.

1.1.2 Критичні особливості APM

Ось деякі ключові особливості, яких притримуються більшість розробників при обробці даних:

- виконання кожного веб-запиту та транзакції - в основі APM можна оцінювати ефективність кожного веб-запиту та транзакції у програмі. Потім можна скористатися цим, щоб зрозуміти, до яких запитів програма звертається найбільше, які найповільніші та які саме слід додати для покращення роботи;

- профілювання продуктивності на рівні коду - якщо є необхідність зрозуміти, чому програма повільна, викидає помилки або є в ній дивні помилки, доведеться перейти до рівня коду. Знаючи, що певний веб-запит не працює, важливо з'ясувати, чому, іноді це дуже складно;

- використання та ефективність усіх залежностей програми, таких як бази даних, веб-сервіси, кешування тощо - чому додаток повільний, зазвичай зводиться до швидкого трафіку або до проблеми з однією із програм. Дуже часто зустрічаються такі проблеми: конкретний запит SQL відбувається повільно; сервер баз даних SQL не працює; сусіди в хмарі викликають проблеми;

- деталізація окремих веб-запитів чи транзакцій- вирішувати проблеми на виробництві дуже важко. Сліди транзакцій значно полегшують цю роботу, оскільки можна бачити деталі про те, що саме відбувається у коді та як це впливає на користувачів;

- основний моніторинг сервера та показники, такі як процесор, пам'ять тощо - проблеми із додатком можуть виникати з багатьох причин. Завдяки віртуалізації та хмарі сервер, який опускається, не є настільки поширеним в ці дні. Однак це

все ж таки відбувається і це те, за чим потрібно стежити. Також важливо стежити за такими речами, як CPU сервера та пам'ять. Багато сучасних веб-додатків зазвичай не пов'язані з CPU, але вони все ще можуть використовувати багато CPU, і це корисний показник для автоматичного масштабування вашої програми в хмарі;

- показники рамки програми, такі як лічильники продуктивності, JMX MBeans тощо - такі показники сервера, як CPU та пам'ять, цікаві, але для розробників показники додатків можуть бути набагато більш цінними для моніторингу справжньої роботи програми. Розробникам необхідно відстежувати показники навколо таких предметів, як збирання сміття, чергування запитів, обсягів транзакцій, часу завантаження сторінки та багато іншого. Також вони можуть стежити за різноманітними лічильниками продуктивності Windows та JMX MBeans. Це також може бути критично важливим для моніторингу таких речей, як Redis, Elasticsearch, SQL та інших сервісів за ключовими показниками;

- показники спеціальних програм, створені командою розробників або бізнесом - стандартні показники сервера та додатків можуть бути дуже корисними для моніторингу програм. Однак можна отримати набагато більшу цінність, створивши та відстежуючи власні спеціальні показники;

- дані журналу додатків-дані журналу - це важлива частина роботи розробників для відслідковування запитів, коли їх програми розгорнуті. Розробникам потрібен доступ до своїх журналів через централізоване рішення для ведення журналів, як продукт управління журналом. На щастя, управління журналом - це включена функція APM в Retrace;

- помилки програми - як розробники, необхідно знати, коли у користувача відбувається помилка на сторінці, і важливо також знати, яка саме. Помилки - це перша лінія захисту для пошуку проблем із застосуванням. Необхідно знайти та виправити помилки або принаймні знати про них, перш ніж клієнти повідомлять нам про це самі, або перестануть бути користувачами.

Відмінне відстеження помилок, звітування та оповіщення є надзвичайно важливими для розробників в системі управління продуктивністю додатків. Необхідно налаштувати сповіщення про нові винятки, а також для моніторингу загальної кількості помилок. Швидше за все, можна знайти якісь нові помилки, які можна швидко визначити та виправити;

- реальний моніторинг користувачів (RUM) - розуміння продуктивності програм на стороні сервера важливо. Однак сьогоднішні програми використовують стільки javascript, що важливо також відстежувати, скільки часу їх браузеру потрібно повністю завантажувати та відтворювати веб-сторінки. Проста помилка JavaScript або повільне завантаження файлу javascript може повністю зіпсувати програму. Справжній моніторинг користувачів або RUM - ще одна важлива особливість APM, що розробникам потрібно повністю контролювати свої програми.

1.2 «Глибина» аналізу мережевого трафіку

Розглядаються мережі з комутацією пакетів. Рішення практичних завдань аналізу полягає в розборі заголовків мережевих протоколів в пакетах і відновленні потоків даних, що передаються. Можна виділити 3 групи при проведення аналізу трафіку (рисунок 1.1):

- поверхневий (SPI — Shallow Packet Inspection);
- помірний (MPI — Medium Packet Inspection);
- поглиблений (DPI — Deep Packet Inspection).

1.2.1 Поверхневий аналіз пакетів (SPI)

Технологія аналізу трафіку, яка грунтується виключно на заголовках пакету рівнів L1-L3 по моделі OSI. Потребує низькі вимоги до обчислювальних ресурсів, що дозволяє аналізувати великі обсяги трафіку. Технологія широко поширена, на

її основі працює більшість міжмережевих екранів операційних систем (зокрема в ОС Windows XP/Vista і OS X), маршрутизаторів і інших мережевих пристроїв. На її основі реалізовані мережеві списки контролю доступу на рівні IP адрес і портів (Access Control List, ACL). Таким чином, дана технологія добре підходить для розмежування доступу ззовні до окремих комп'ютерів (IP) і сервісів (порти) внутрішньої мережі.

Рисунок 1.1 - Відповідність рівнів моделі «OSI» «глибині» аналізу мережевого трафіку

1.2.2 Помірний аналіз пакетів (MPI)

Технологія аналізу трафіку, яка ґрунтується на інспектуванні сесій і сеансів зв'язку, ініційованих додатком, але встановлюваних шлюзом-посередником (рисунок 1.2). Також застосовується термін «проксі додатків» (application proxy). В рамках даної технології вміст пакетів аналізується частково і по визначеним правилам. Не використовуються складні методи аналізу типу сигнатурного. Пристрої, що реалізують даний функціонал розміщуються між провайдером

інтернету і кінцевим користувачем. Дані пристрої розбирають заголовки аж до транспортного рівня і невелику частину даних пакета для зіставлення розібраної частини з деяким списком розбору (parse list), з подальшою реакцією в разі їх виявлення. Дані списки зазвичай коротше списків ACL і надають більш широкий діапазон дій на відміну від «дозволити/заборонити» в разі ACL. Ці списки також більш виразні, так як дозволяють прив'язуватися не до IP- адреса, а до формату даних пакетів і даними деяких протоколів рівня додатки, наприклад, URL-адресів в разі протоколу HTTP. За допомогою MPI можна, наприклад, заблокувати можливість отримання flash-файлів або картинок з певних інтернет сервісів (на рівні уявлення OSI) або заблокувати частину команд (на рівні додатку OSI) в окремих протоколах. Набір протоколів, як правило, дуже обмежений. Наприклад, в перших версіях CheckPoint FireWall-1 (CheckPoint FW-1) підтримувалися протоколи Telnet, FTP, HTTP, а в Cisco Private Internet Exchange (Cisco PIX) - FTP, HTTP, H.323, RSH, SMTP і SQLNET. Згодом дані набори незначно розширювалися. Також відомо, що дана технологія використовується в продуктах компаній McAfee і Symantec. Міжмережеві екрани, що використовують цю технологію, відносяться до другого покоління. Дана технологія більш гнучка в порівнянні з SPI і, крім розмежування доступу, підходить для більшого числа завдань - кешування вмісту, аналіз стисненого/шифрованого трафіку, обмеження функціонала окремих протоколів шляхом заборони окремих команд. Завдяки підключенню в режимі проксі, може служити в якості Wan Optimizer’a.

Основний недолік MPI - погана масштабованість: кожна команда і протокол вимагають окремого «шлюзу» (вхідний-вихідний порти). Крім того, робота в режимі проксі сильно знижує швидкість обробки. Для зниження навантаження на проксі-сервер був розроблений протокол ICAP, що дозволяє проксі-серверам відправляти через них дані для проведення аналізу стороннім серверам на предмет безпеки або аналізу вмісту. Ця схема реалізована в антивірусному продукті ClamAV, який може підключатися до проксі-серверів Squid і NetCache. Ці фактори сильно обмежують застосування даної технології на рівні провайдерів

інтернету внаслідок необхідності аналізу великого числа протоколів і команд на широких каналах зв’язку.

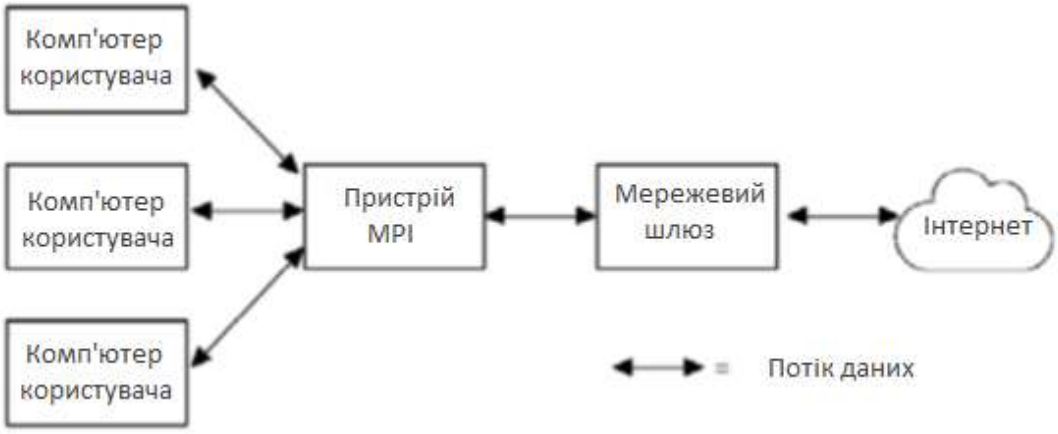

Рисунок 1.2 - Схема застосування пристроїв аналізу на основі технології MPI

1.2.3 Поглиблений аналіз пакетів (DPI)

Іноді вживають більш вузький термін - DPP (Deep Packet Processing), який має на увазі такі дії над пакетами, як модифікація, фільтрація або перенаправлення. Сьогодні обидва терміни часто використовуються як взаємозамінні. Дана технологія є логічним розвитком MPI. В рамках даного підходу аналізатор переглядає вміст кожного пакета повністю. Одним з важливих відмінностей від попередніх технологій є те, що системи на базі DPI можуть приймати рішення не тільки по вмісту пакетів, але і за непрямими ознаками, властивим якимось певним мережевим програмами і протоколам. Для цього може використовуватися статистичний аналіз.

Наприклад, аналіз частоти зустрічі певних символів, довжин пакетів, відстань між мітками часу послідовних пакетів тощо. Також, в порівнянні з попередніми підходами, значно розширено список застосувань технології: класифікація, обмеження смуги, пріоритезація, маркування, кешування тощо. Технологія DPI отримала розвиток, перш за все, через стрімке зростання

обчислювальних здібностей процесорів, їх швидкодії і, відповідно, можливостей для більш повного і точного аналізу мережевих даних.

На відміну від MPI, дана технологія спочатку розроблялася для високошвидкісної обробки та ідентифікації великої кількості додатків в реальному часі. Таким чином, рішення на основі DPI добре масштабуються як по ширині мережевого каналу (відомі рішення, що працюють на каналах близько 100 гбіт/сек), так і за кількістю ідентифікованих додатків (в існуючих рішеннях - порядку декількох тисяч). З точки зору реалізації, основний компонент будь-якого рішення DPI - модуль класифікації, що відповідає за класифікацію мережевих потоків. При цьому в залежності від цілей застосування DPI, класифікація може виконуватися з різною точністю:

- тип протоколу або додатка (наприклад, Web, P2P, VoIP);
- конкретний протокол рівня додатка (HTTP, BitTorrent, SIP);
- додаток, що використовує протокол (Google Chrome, µTorrent,Skype).

Важливо відзначити, що відповідність між класами різних рівнів точності неоднозначно, що показано на рисунку 1.3.

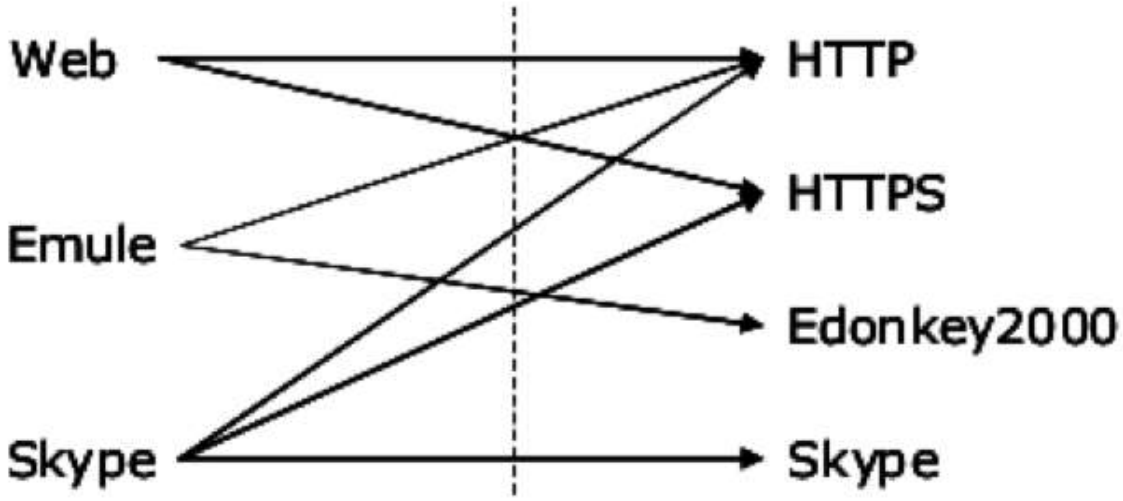

Рисунок 1.3 - Різниця між ідентифікацією додатків (зліва) і протоколів (праворуч)

1.3 Особливості передачі мережевого трафіку

Кожен мережевий пакет складається з керуючої інформації і корисного навантаження. Тут і далі термін «пакет» застосовується в якості універсального для узагальнення таких понять, як фрейм, дейтаграма, сегмент відповідних мережевих протоколів. В процесі розбору в пакеті виділяються заголовки протоколів, аналізуються значення полів в них.

Структура заголовка визначається специфікацією, тоді як корисне навантаження може містити довільним чином організовані дані, хоча зазвичай являє собою пакет протоколу наступного, більш високого рівня: для продовження розбору необхідно визначати, який це протокол (рисунок 1.4).

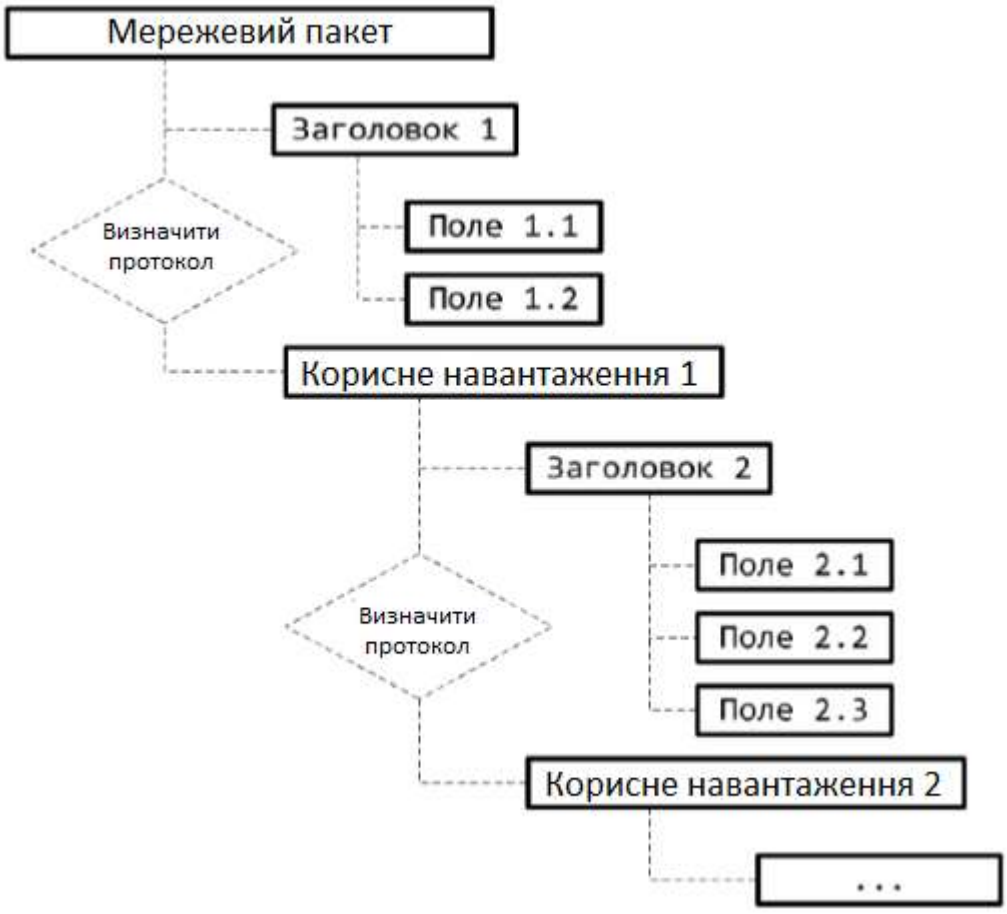

Рисунок 1.4 - Виділення та розбір заголовків протоколів в пакеті

Відповідно до моделі OSI заголовки мережевих протоколів пакета утворюють стек і, як правило, слідують один за одним у природному порядку - від низького рівня до високого. Однак при організації тунельних з'єднань цей порядок може бути порушений - наприклад, при передачі IPv4-пакетів (мережевий рівень) в рамках пакетів протоколу UDP (транспортний рівень). Тунельні протоколи в даний час набули широкого поширення: зокрема, вони використовуються при організації віртуальних приватних мереж. У загальному випадку можлива побудова тунелю довільної конфігурації: зокрема, один тунель може бути вкладений в інший. Розбір тунельного трафіку повинен підтримуватися мережевим аналізатором.

Виділяють протоколи зі збереженням і без збереження стану. Обов'язковою частиною специфікації протоколу зі збереженням стану є відповідний автомат станів (Protocol State Machine). При проведенні аналізу в режимі реального часу кількість з'єднань, для яких необхідне збереження характеристик поточного стану, може необмежено рости. Тому аналізатор повинен гнучко управляти розподілом доступних йому ресурсів.

Важливою характеристикою мережевого протоколу є MTU (Maximum Transmission Unit) - максимальний розмір даних, які можуть бути передані в рамках одного пакету. Для протоколу IPv4 значення MTU складає 65535. Оскільки на практиці IPv 4 - пакети зазвичай інкапсулюється в Ethernet-фрейми, результуюче значення MTU визначається відповідно до конкретної версією стандарту Ethernet, підтримуваного мережевим обладнанням. Для блоків даних з розміром, що перевищує MTU, проводиться фрагментація: відправник розбиває блок на порції допустимого розміру, після чого кожна порція передається в рамках окремого пакета. Одержувач, таким чином, повинен виконати дефрагментацію: відновити вихідний блок з отриманих окремо порцій. Для протоколу IPv4 останній фрагмент визначається скинутим прапором MF (More Fragments): при цьому в ньому не містяться (рисунок 1.5.) дані наступної PDU (Protocol Data Unit - одиниця передачі).

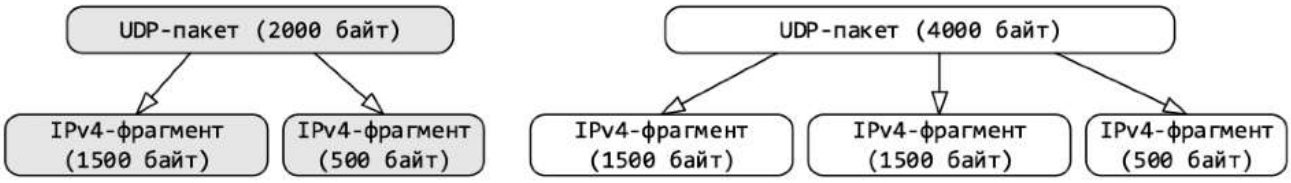

Рисунок 1.5 - Приклад фрагментації IPv4

У випадку протоколу TCP (рисунок 1.6) неформальною ознакою «останнього» для заданого PDU сегмента є PSH-прапор, однак цей сегмент в загальному випадку містить дані наступної одиниці передачі звідси виникає завдання визначення меж.

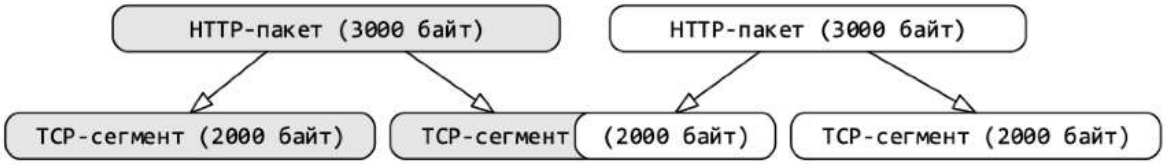

Рисунок 1.6 - Приклад сегментації TCP

Таким чином для проведення глибокого аналізу потрібно:

- відновлювати початковий порядок пакетів;
- визначати границі вище за списком PDU.

Для забезпечення безпеки з’єднань деякі протоколи передбачають передачу даних в зашифрованому виді наприклад сімейство протоколів TLS. Щоб проаналізувати зашифровані дані, необхідно до того їх розшифрувати, використав наданий користувачем ключ (рисунок 1.7): аналізатор повинен мати інтерфейс для додавання відсутньої для проведення розбору інформації.

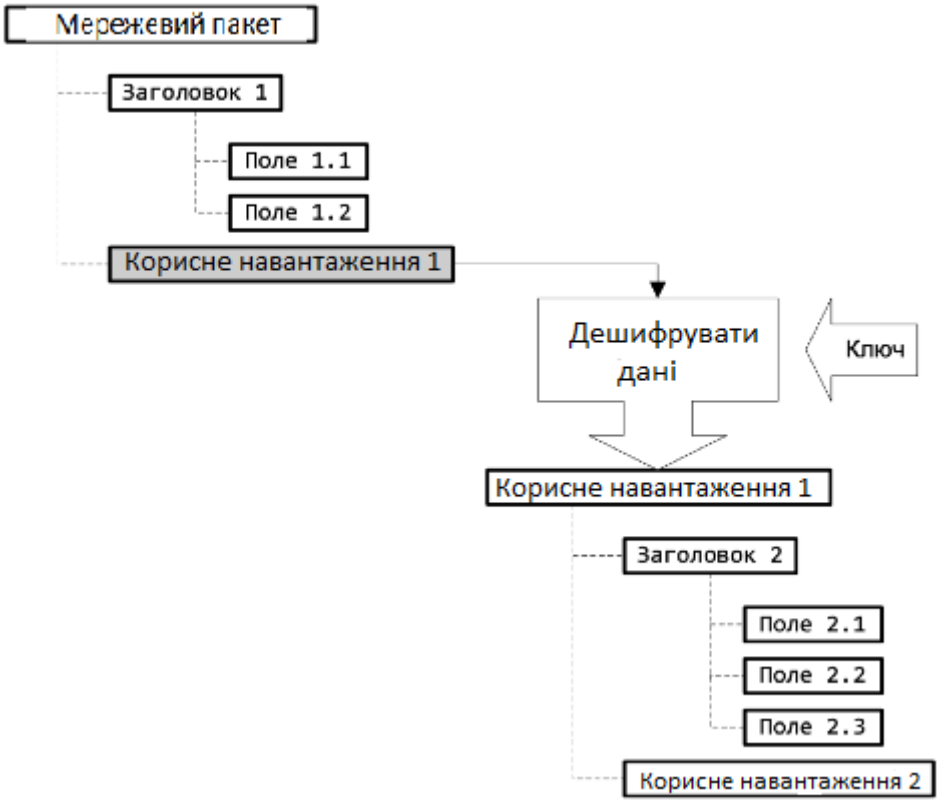

Рисунок 1.7 - Розшифровка даних з використанням зовнішнього ключа

При аналізі трафіку неминуче виникають помилки розбору. Під цією помилкою розуміється невідповідність між специфікацією протоколу (кодом який виконує розбір) та даними, розбір яких проводиться згідно цієї специфікації. Причини виникнення помилок розбору різняться:

- недокументовані можливості протоколу;
- викривлення даних при транспортуванні мережею;
- помилки в коді аналізатора;

Помилки розбору повинні легко локалізуватися та відтворюватися. Якщо помилка, яка виникла не є критичною, аналіз повинен продовжуватися.

Аналізатори мережевого трафіку, як правило, мають модульну архітектуру: з часом з'являються нові протоколи, і їх необхідно підтримувати. Розширювати систему, в якій функції розбору даних всіх протоколів зосереджені в одному функціональному модулі, важко. У разі модульної архітектури для кожного протоколу створюється окремий модуль, в якому визначаються методи та

структури даних для роботи з цим протоколом. Виникає додатковий питання про залежності: при додаванні нового модуля необхідно «повідомити» про його існування іншим. Вносити зміни в код існуючих модулів неефективно: може бути порушена логіка їх роботи або внесені помилки, налагодження яких важке. До того ж потрібна повторна збірка змінених модулів. Тому необхідно мінімізувати кількість внесених в існуючу розробку змін, необхідних для додавання підтримки нового протоколу.

Слід відзначити, що деякі практичні завдання вирішуються за допомогою аналізу файлу зі збереженим трафіком (мережева траса):

- відтворення помилок розбору;
- розробка (налагодження) розбирачів;
- розслідування інцидентів порушення інформаційної безпеки. Тому вкрай важливо забезпечити можливість використання «результатів» offline-аналізу для роботи в режимі online, тобто перенесення модулів розбору між інструментами offline- і online-аналізу.

Необхідність в проведенні розбору заголовків мережевих пакетів, як вже було зазначено, виникає при вирішенні багатьох практичних завдань. Важливо розуміти, що фахівець в області мережевої безпеки в загальному випадку може не мати високі навичками в програмуванні. Тому необхідно надати високорівневий інтерфейс (API), що дозволяє користувачеві підтримувати в рамках системи нові (зокрема закриті) мережеві протоколи.

2 ЗАСОБИ МОНІТОРИНГУ МЕРЕЖЕВОГО ТРАФІКУ

Основним інструментом для спостереження повідомлень, якими обмінюються хости є сніфер пакетів. Як випливає з назви, сніфер пакетів захоплює ("нюхає") повідомлення, передані з комп'ютера або отримані ним. Крім цього програма-сніфер, як правило зберігає та відображає вміст різних полів протоколів у цих захоплених повідомленнях. Сніфер пакетів – це пасивна програма. Вона досліджує повідомлення, що відправляються і одержуються програмами і протоколами, які працюють на комп'ютері, але ніколи не відправляє пакети сама. Отримані пакети ніколи не адресуються сніферу. Сніфер пакетів отримує копію пакетів, переданих/отриманих додатками або протоколами, які виконуються на комп'ютері.

2.1 Пакетні сніфери

Пакетний сніфер — це або програмний, або апаратний інструмент для перехоплення, реєстрації та аналізу мережевого трафіку та даних. Ці інструменти допомагають визначити, класифікувати та усунути неполадки мережевого трафіку за типом програми, джерелом та пунктом призначення. На ринку є різноманітні інструменти, більшість з яких покладаються на інтерфейси прикладних програм (API), відомі як pcap (для Unix-подібних систем) або libcap (для систем Windows) для захоплення мережевого трафіку. Тоді найкращі сніфери пакетів аналізують ці дані, що дозволяє точно визначити джерело проблеми та не допустити її в майбутньому.

Щоб по-справжньому зрозуміти важливість сніферів , важливо розуміти як відбувається маршрутизація в Інтернеті. Кожен електронний лист, який надсилається, відкрита веб-сторінку та файл поширюється в Інтернеті як тисячі маленьких керованих фрагментів, відомих як пакети даних. Ці пакети

передаються через стек протоколів, відомий як протокол управління передачею /протокол Інтернету (TCP/IP). TCP/IP розбивається на чотири шари:

- рівень протоколу додатків, рівень протоколу управління передачею (TCP),
- рівень інтернет-протоколу (IP) та апаратний рівень.

Кожен пакет переміщується через рівень програми мережі до рівня TCP, де йому присвоєний номер порту. Далі, пакет переходить на IP-рівень і отримує свою цільову IP-адресу. Як тільки пакет має номер порту та IP-адресу, він може бути відправлений через Інтернет. Надсилання здійснюється через апаратний рівень, який перетворює пакетні дані в мережеві сигнали. Коли пакет прибуває до місця призначення, дані, які використовуються для маршрутизації пакету (номер порту, IP-адреса тощо), видаляються, і пакет рухається далі через стек протоколів нової мережі. Після досягнення вершини він збирається в первісну форму.

2.1.1 Опис роботи пакетних сніферів

Пакетні сніфери працюють, перехоплюючи дані про трафік під час проходження по дротовій або бездротовій мережі та копіюючи їх у файл. Це відомо як захоплення пакетів. Хоча комп'ютери, як правило, розроблені для того, щоб ігнорувати ступінь трафікової активності від інших комп’ютерів, пакетні сніфери це перетворюють. При встановленні програмного забезпечення, мережева карта інтерфейсу (NIC) - інтерфейс між комп’ютером та мережею - повинна бути встановлена в розрядний режим. Це дає змогу комп’ютеру зафіксувати та обробити через sniffer пакет все, що потрапляє в мережу.

Що можна захопити, залежить від типу мережі. Для дротових мереж конфігурація мережевих комутаторів, які відповідають за централізацію зв'язку з декількох підключених пристроїв, визначає, чи може мережевий сніфер бачити трафік у всій мережі або лише на її частині. У бездротових мережах інструменти збору пакетів зазвичай можуть захоплювати лише один канал одночасно, якщо хост-комп'ютер не має безлічі бездротових інтерфейсів.

2.1.2 Переваги пакетних сніферів

Отже, у чому сенс аналізаторів пакетів? Сніфер може допомогти орієнтуватися на нові ресурси при розширенні пропускної спроможності мережі, керуванні пропускною здатністю, підвищенням ефективності, забезпеченням ділових послуг, підвищенням безпеки та покращенням роботи кінцевих користувачів. Що стосується великих та малих компаній, щоденні завдання можуть бути негайно зірвані проблемами ефективності, пов'язаними з мережею, додатком чи обома. Щоб відновити роботу їхньої компанії, систематики повинні мати можливість швидко визначити першопричину. Оскільки sniffers пакетів переглядають та збирають інформацію для всього трафіку по всій мережі, вони можуть оцінювати критичні шляхи мережі, щоб допомогти адміністраторам визначити, що програма чи мережа є причиною поганого досвіду користувачів. З цією інформацією, адміністратори краще оснащені для визначення та вирішення походження проблеми.

Коли користувачі повідомляють про повільність, адміністратори можуть використовувати аналіз PCAP для вимірювання часу реакції в мережі - також відомий як затримка мережевого шляху - та визначати кількість часу, необхідного для переходу пакета через мережевий шлях від відправника до одержувача. Це дозволяє адміністраторам швидко визначити причину уповільнення та виявити постраждалі програми, щоб вжити заходів.

Аналіз трафіку за типом. Оцінюючи проблеми з мережею та додатками, першочергове значення має трафік у мережі. За допомогою правильного аналізатора IP-sniffer та пакетів трафік класифікується на типи на основі IP-адрес сервера призначення, використовуваних портів та вимірювання загального та відносного обсягу трафіку для кожного типу. Це дає вам змогу виявити надмірний рівень некомерційного трафіку (наприклад, соціальних медіа та зовнішнього веб-серфінгу). Можна також визначити трафік, що протікає через мережеве

посилання, а також трафік на конкретні сервери або програми для цілей управління ємністю.

Поліпшення пропускної здатності. Коли користувачі скаржаться на те, що «мережа повільна» або «Інтернет знижується», продуктивність припиняється, знижуючи рентабельність інвестицій . Щоб виправити цю помилку, потрібно зрозуміти, як та ким використовується пропускна здатність мережі. Сніфер пакету Wi-Fi може отримати показники продуктивності для автономних точок доступу, бездротових контролерів та клієнтів. Багато з них також пропонують моніторинг несправностей, продуктивності та доступності мережі, кореляцію даних між стековим стеком, аналіз мережевого шляху та багато іншого, щоб допомогти виявити потенційні проблеми та мінімізувати час простою мережі.

Поліпшення безпеки. Великий обсяг вихідного трафіку може означати, що хакер використовує програми, або для спілкування зовні, або для передачі великої кількості даних.

2.2 Огляд інструментів обробки пакетів

Сьогодні на ринку є незліченна кількість інструментів, як платних, так і безкоштовних. І хоча кожен інструмент побудований на основних принципах збору мережевого трафіку, вони значно відрізняються за своєю шириною та глибиною. Багато інструментів з відкритим кодом є надзвичайно простими у своїй конструкції, і в цьому справа: ці інструменти створені для забезпечення надійного, чистого збору даних, залишаючи як найменший слід. Якщо потрібні прості сніфери та швидка діагностика, безкоштовний інструмент з відкритим кодом може бути корисним. Багато - хоча і не всі - безкоштовні версії можна оновити, щоб надати додаткові аналітичні функції, якщо потрібна більша підтримка.

З такою кількістю продуктів на ринку важко дізнатися, який сніферний пакет вибрати. Незважаючи на те, що безкоштовних варіантів вистачає, покупка

сніфера пакета може гарантувати, що ви озброєні інструментом, який не тільки фіксує дані, але й пропонує інтуїтивний аналіз. Виходячи за рамки базових сніферів, яких налічується десятки, можна знайти більш надійні аналітичні інструменти для збору пакетів та мережевого збору даних. У багатьох випадках, що відрізняє ці інструменти, - це здатність проводити глибоку перевірку пакетів (DPI).

Ці великі інструменти на рівні підприємств часто оснащені для оповіщення про випадки виключення та створення інтуїтивно зрозумілих графіків та діаграм із відображенням детальних показників. Хоча вони йдуть високою ціною, вони варті своїх інвестицій.

2.2.1 TCPDUMP

Це популярний інтерфейс командного рядка (CLI) та інструмент дослідження пакетів з відкритим кодом, сумісний на платформах Unix та Linux. Він був винайдений у 1987 році в Національній лабораторії Лоуренса Берклі, а після цього опублікований через кілька років.

У ньому є бібліотека libpcap, розроблена мовою програмування на C, яка працює для збору інформації мережі. Libpcap забезпечує інтерфейс для всіх загальних платформ на базі Unix, включаючи FreeBSD та Linux. Інтерфейс libpcap в платформі Windows під назвою WinDump. Windump використовується WinPcap, який є портом Windows бібліотеки libpcap. Розробники розробляли бібліотеку libpcap як API незалежної платформи для роботи над різними програмами та усунення системної залежності для модулів збору даних у кожній програмі. TCPDump розглядається як інструмент розбору.

За замовчуванням він перехоплює та друкує пакет, який захопило з мережі; інші функції, такі як зберігання, виконуються заданими командами. TCPDump працює так:

- читання/запис захопленого файлу з мережі в пакеті CAPture (PCAP) за допомогою команд CLI;

- він фільтрує пакети за деякими заданими параметрами;

- він друкує на екрані захоплені дані відповідно до заданих параметрів.

Це більш легкий і портативний інструмент для сніфу пакетів, оскільки він залежить лише від CLI, і мережеві адміністратори використовують його для доступу до мережевих пристроїв з віддаленого місця. На рисунку 2.1 показаний трафік TCP/IP та його аналіз дослідження пакетів TCPDump, відображення адреси і вмісту трафіку даних.

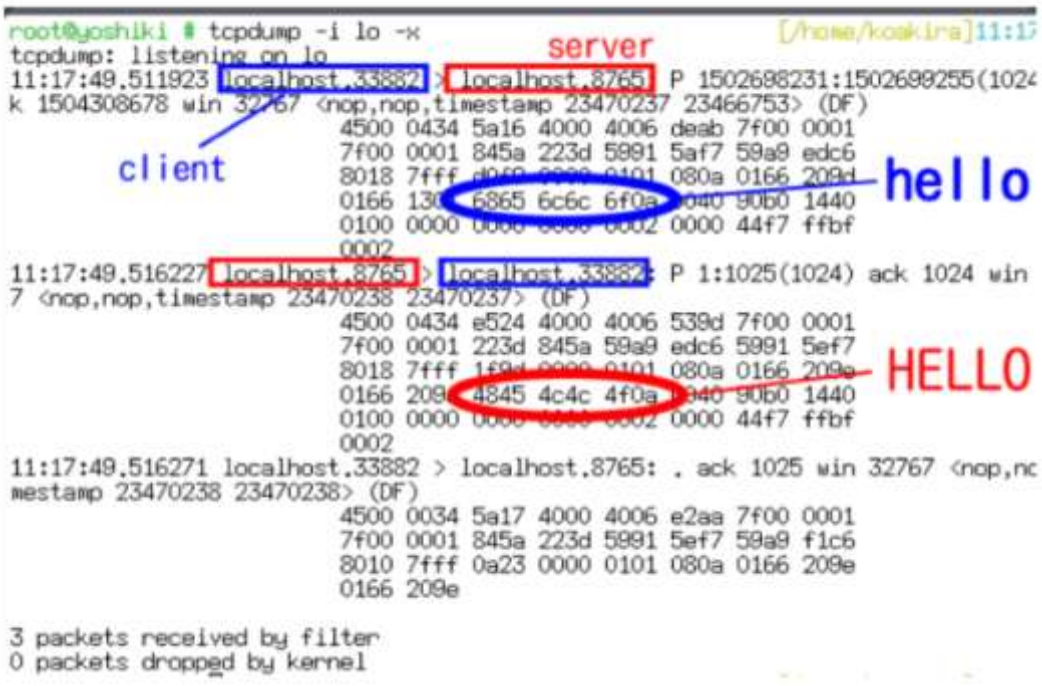

Рисунок 2.1 - Огляд TCPDump, показує потік характеристик TCP/IP

Основне обмеження TCPDump, він не надає адміністратору мережі візуального GUI захоплених даних для більшого аналізу, є тільки CLI. Оскільки, це текстовий формат і користувачеві легко користуватися ним дистанційно через з'єднання Telnet. Є ще кілька недоліків у TCPDump. До них належать:

- обмеження в аналізі трафіку, можуть застосовуватися лише протоколи на основі TCP;

- він повідомляє лише те, що знаходить у пакетах, якщо IP-адреса підроблена в трафіку, вона не має можливості повідомляти нічого іншого;

- пакети, заблоковані брандмауером, не відображаються.

2.2.2 Wireshark

Винайдений вченим Джеральдом Комбсом наприкінці 1997 року для перевірки та розпізнавання проблем мережі та моніторингу трафіку даних. Він назвав його Ethereal до травня 2006 року, після чого його назва змінилася на Wireshark. Це програмне забезпечення з відкритим кодом, безкоштовний інструмент і аналізатор пакетів для графічного інтерфейсу, який написаний мовою програмування на C та випущений під ліцензією GNU General Public License (GPL). Він працює на різних Unix-подібних операційних системах, включаючи Mac OS X, Linux, платформу Solaris, а також операційну систему Microsoft Windows. Інтерфейс командного рядка (CLI) Wireshark називається TShark, що дозволяє користувачеві працювати з ним за допомогою команд. Це як TCPDump з додатковим графічним інтерфейсом, підтримуючи різноманітні протоколи та можливість фільтрування та сортування. Він використовується в мережевому інструменті судового аналізу (NFAT) в організаціях.

Wireshark призначений для захоплення пакетів з працюючих мереж, а також для перегляду раніше збереженого файлу даних. Підтримуваний формат захоплення пакетів — це формат файлу "PCAP". Він відображає захоплені дані у байтовому та шістнадцятковому форматах, де відображаються різні типи використаних пакетів та протоколів. Це також дозволяє користувачеві збирати дані пакетів у потік TCP.

Він має інтерфейс з трьома панелями; панель підсумків або панель списку пакетів, яка показує різні аналізовані пакети, такі як номер кадру, дата, час, адреса призначення та джерела IP, протоколи верхнього рівня, довжина пакету та інформація вмісту трафіку з кольором для кожного захопленого типу пакету.

Друга панель - це захоплені деталі пакета. Коли пакет визначається на панелі списку пакетів, деталі з'являються на наступних двох панелях; деталі та байтові або шістнадцяткові панелі. Панель деталей відображається як деревоподібна структура протоколів, захоплених для різних застосунків, таких як протокол управління передачею (TCP), протокол дейтаграм користувача (UDP), протокол повідомлення Internet Control (ICMP), протокол передачі гіпертекстового тексту (HTTP) тощо. Третя панель називається панеллю даних або байтом, яка показує необроблені захоплені дані, що відображають байт пакета у шістнадцятковому форматі, кодування ASCII та текстові формати.

Тут важлива примітка: для запуску інструменту Wireshark він встановлює NIC в розрядний режим, що дозволяє sniffer бачити весь трафік на цьому інтерфейсі, а не лише трафік, адресований одному з налаштованих інтерфейсів (рисунок 2.2.). Окрім розрядного режиму, порт може показувати дзеркальне відображення до будь-яких точок мережі, коли розрядний режим охоплює її не всю.

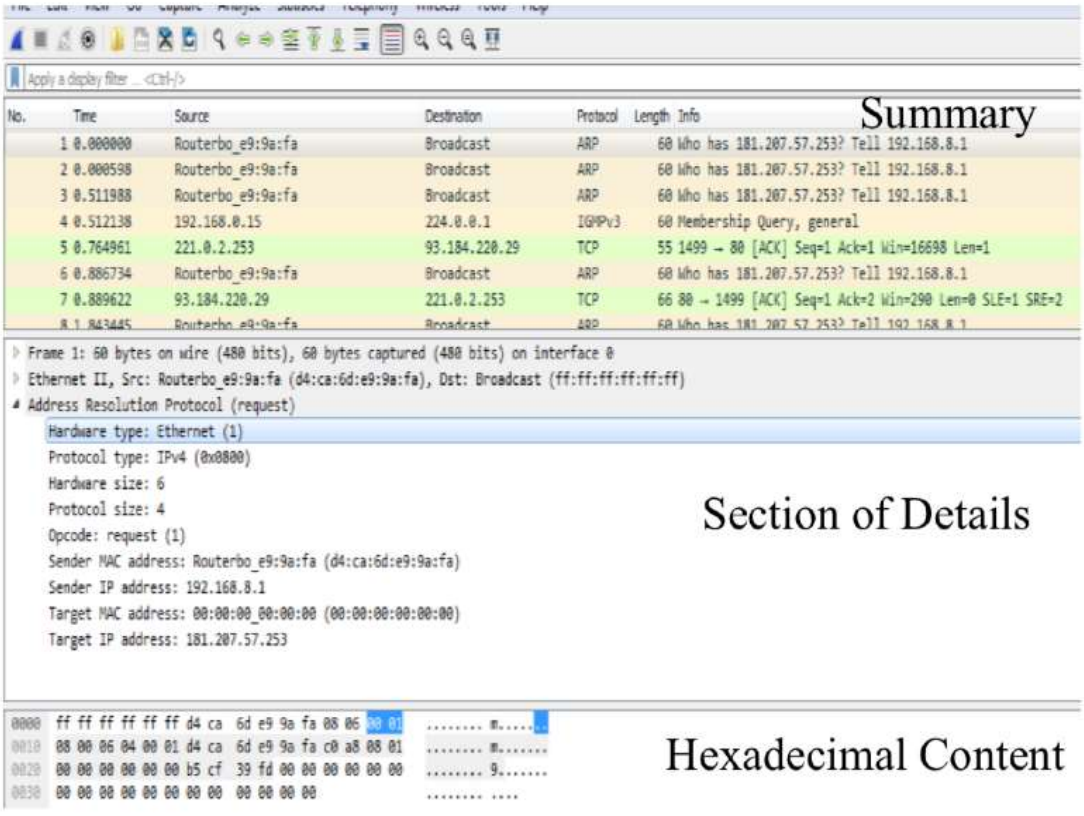

Рисунок 2.2 - Інтерфейс інструменту для обробка пакету Wireshark

Тут показаний інтерфейс Wireshark, що відображає три вікна; панель підсумків, деталі та байт з різними характеристиками трафіку мережі в читаному вигляді. У розділі деталей видно, що номер кадру становить 60 байт, тип мережі - це Ethernet II, тип протоколу - IPv4, а вміст корисної навантаження - ARP. Розмір кадру вимірюється за допомогою модуля максимальної передачі (MTU), і цей блок визначається відповідно до використовуваного типу мережі. Наприклад, MTU для асинхронного режиму передачі (ATM) становить 53 байти, MTU в мережах Ethernet і IPv4 - 1500 байт, і в інших типах мереж використовується перемикання кадрів, які досягають 9000 байт. Отже, розмір кадру змінюється відповідно до MTU у заданому типі мережі.

Крім того, розмір кадру визначають також відповідно до використовуваної програми в мережі. У цьому випадку, це 60 байтів, в яких ARP використовується для пошуку фізичної адреси інтерфейсу маршрутизатора або хоста, коли вказана його логічна IP-адреса. В іншому прикладі дані кадру розміром 500 байт за допомогою застосованої якості обслуговування (QoS) в деяких додатках охорони здоров’я на базі UDP і зроблено висновок, що він підходить як для затримки, так і для тремтіння, а також використано деякі інші QoS що призвело до отримання кадру розміром 1500 байт. Потім кадр з'являється в інструменті дослідження пакетів відповідно до програми або протоколу, який використовується в мережі.

Обмеження Wireshark, воно потребує найкращого розуміння форматів протоколів, HTTP та Cascade Style Sheet CSS, знання мови у форматі байтів. Він використовує формат файлу PCAP для фіксації трафіку, тому він може захоплювати пакети лише тих типів мережі, які підтримують формат файлу PCAP. Ще одним недоліком є те, що Wireshark не є автоматизованим інструментом, він не є підтримкою для моніторингу тривалого часу.

2.2.3 Colasoft

Це інструмент аналізатора протоколів мережевого протоколу із закритим джерелом, призначений для роботи на платформі операційної системи Windows, що використовується для особистого використання адміністратором мережі для усунення несправностей, моніторингу та діагностики трафіку в комп'ютерній мережі. Capsa випускає безкоштовні інструменти Colasoft, що забезпечує простоту використання, аналіз пакетів у режимі реального часу та надійний криміналістичний, поглиблений аналіз протоколів, і в нього 24-годинний моніторинг мережі. Він має особливість відкриття декількох інтерфейсів в одному екземплярі, надання користувачеві графічних інтерфейсів та матричних представлень.

Він має глибокий аналіз пакетів, що показують різні характеристики з можливістю генерування звітів, журналів та попередження лише голосовими та електронними повідомленнями для ліцензованих версій.

Він має різноманітні функції графічного інтерфейсу, відображаючи захоплену інформацію у графіках, матриці та відповідно до кожної характеристики трафіку мережі, що показує кожен протокол, що використовується в мережі. На рисунку 2.3 показаний інструмент Colasoft та його розширені функції GUI.

Є деякі обмеження Colasoft; це дорогий додаток, тоді як безкоштовна версія доступна, але з обмеженими можливостями, наприклад, безкоштовна версія не повідомляє користувача через електронну пошту та голосові канали. Ще два недоліки інструменту Colasoft - це те, що він працює лише на платформі операційної системи Microsoft Windows, і він підтримує лише 300 протоколів, що вважається меншим порівняно з іншими інструментами для пакетів, такими як Wireshark.

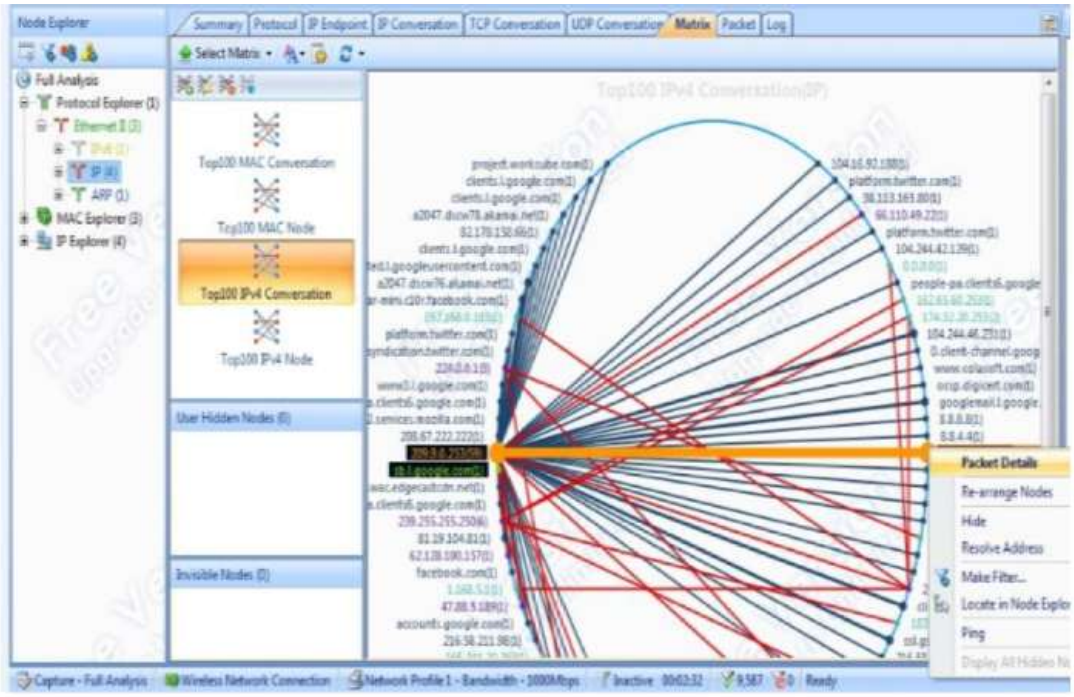

Рисунок 2.3 - Інтерфейс інструменту Colasoft

2.3 Порівняльний аналіз інструментів обробки пакетів

Для порівняння між перерахованими вище трьома методами мережевого аналізу ми залежимо від таких параметрів, як код з відкритим вихідним кодом, кількість підтримуваних протоколів, підтримувана операційна система, підтримує PCAP, користувальницький інтерфейс, вартість, форми декодування, визначені аномальні пакети, реконструюють потік TCP, тощо.

В таблиці 2.1 показано порівняння між інструментами Wireshark, TCPDump та Colasoft. З таблиці порівняння, жоден інструмент аналізу пакетів не веде його за всіма параметрами. Але порівняння з перевагами та недоліками допоможе розробникам вдосконалити інструмент дослідження пакетів, щоб подолати ці обмеження. Тут ми порівняємо інструменти, використовуючи якісні та кількісні параметри .

Colasoft має функції аналізу більш візуального пояснення зі статистикою захоплених пакетів, відображенням більшої кількості інформації про протоколи та користувацькі програми з графіками та матричним поданням для всіх

підключених кінцевих точок. Додаткові функції, які Colasoft включає в себе звіти, журнали аудиту та діагностику. Усі ці функції допомагають адміністраторові мережі діагностувати проблеми мережі. Також Colasoft має потужний аналіз та інтерпретацію потоку TCP; він має універсальну пропускну здатність, мережевий трафік та аналіз використання. Він має матричне подання та функції поглибленого декодування трафіку з багаторазовою поведінкою мережевого моніторингу. Крім того, вона має візуалізацію приховування комп'ютерних мереж.

Таблиця 2.1 - Порівняльні характеристики між Wireshark, TCPDump та Colasoft Packet Sniffing Tools

Параметри	**Сніферні інструменти**		
	TCPDump	**WireShark**	**Colasoft**
Відкритий код	+	+	-
Якими операційними системами підтримується	Linux(WINDum для Windows)	Linux, Windows	Windows
Число підтримки протоколів	TCP/IP	БІльш ніж 300	300
Користувацький інтерфейс	CLI	CLI і GUI	GUI
Вартість	Безкоштовно	Безкоштовно	999 $
Libpcap основа	+	+	-
Визначення прихований даних	-	+	+
Використання місцяна диску	484КБ	449МБ для Unix 89МБ для Windows	32МБ
Відображення в додатку шару протоколу	-	+	+
Декодування протоколу	Лише Hex, ASCII	Лише Hex, ASCII	EBDIC, Hex, ASCII
Відновлення TCP потоку	-	+(але лише форматоване)	+

Виявлення ненормальних даних	-	-(лише створює попередження)	+
Кілька інтерфейсів	-	-	+
Сповіщення знаходження	-	-	+
Відновлення HTTP веб сторінки	-	-(показує актуальнийконтент трафіку індивідуально)	-(показує лінки контенту трафіка індивідуально)
Мережева комунікаційна матрична мапа	-	-	+
Оцінка критичного танекритичного для бізнесу трафіку	-	+(за допомогою створення нових фільтрів та пошуку)	+(вбудована)
Можливість розробляти та налаштовувати и розробникам и	+(можливе налаштування користувачем під себе)	+	-(лише командою розробки capsa.co)
UDP трафік	-	+	+

Крім того, Colasoft має більш потужну видимість та стиль Windows 7 з простими графічними екранами, інформаційними панелями та мережевим аналізатором. За допомогою цього інструменту для користувача легко і просто зробити аналіз ,просто завдання, яке ви хочете, виконується натисканням миші. Отже, Colasoft стає більш зручним інструментом для дослідження пакетів, він забезпечує легкий для читання спосіб і має декілька інтерфейсів в одному екземплярі. Графіки показують більшу візуалізацію для різних статистичних даних і властивостей мережі. Wireshark обмежений цими можливостями GUI, і він не відкриває декілька інтерфейсів в одному екземплярі.

У порівнянні з Wireshark, Colasoft забезпечує більшу мережеву безпеку за допомогою сповіщень про попередження через аудіо та електронні листи. Недолік

Colasoft, він охоплює лише 300 протоколів, що дуже менше порівняно з Wireshark, який підтримує 1100 протоколів .

TCPDump - це дуже портативний і економічний пакет нюхає інструмент з точки зору використання пам’яті, оскільки він займає лише 484 КБ місця для встановлення. Хоча розмір інсталяційного файлу Wireshark на початку інсталяції становить 18 МБ, але після завершення інсталяції він споживає 81 Мбайт у Windows та 449 МБ дискового простору в операційній системі Linux. Місце для встановлення Colasoft становить 32 Мб. Отже, що стосується використання пам'яті, то Wireshark дуже дорогий.

Оскільки Wireshark - це відкритий код, кожен може завантажити його код і вдосконалити його. У світі існує багато універсальних розробників, які мають можливість налаштування та вдосконалення цього інструменту, в той час як Colasoft обмежений лише командою розробників компанії Capsa. Отже, Wireshark вважається хорошим інструментом дослідження пакетів для розуміння функцій програмування, і він відповідає вимогам користувачів мережі, здійснюючи налаштування без витрат. У інструменті Colasoft, якщо користувачеві потрібна певна налаштування для певної проблеми моніторингу мережі, він вимагає від компанії, яка надає оплату, за ці налаштування. За допомогою інструменту аналізу пакетів Wireshark ви можете отримати більше досвіду конфігурації TCP / IP, розуміючи структуру мережі, а також вона працює на різних платформах, включаючи Linux, Solaris, OS X і Windows.

Крім того, деякі автори проводять дослідження в цій галузі для вдосконалення інструменту Wireshark, тут вдосконалено інструмент дослідження пакетів Wireshark для виявлення вторгнення типів атак Denial of Service (DoS), особливо в тому випадку, коли можна подолати атаку затоплення ping, яка надсилає велику кількість команд ping на пристрій жертви.

2.3.1 Результати порівняльного аналізу

Усі ці інструменти мають загальні характеристики мережевих властивостей, але кожен інструмент має свою конкурентну особливість. Існують різноманітні якісні та кількісні параметри, які обговорюються та порівнюються на інструментах : Wireshark, TCPDump і Colasoft. З цих параметрів - кількість підтримуваних протоколів, відкритий вихідний код, платформа, що підтримується, бібліотека libpcap, підтримка PCAP, користувальницький інтерфейс, вартість, форми декодування, визначені аномальні пакети, мережевий зв’язок у матричній карті, реконструкція потоку TCP тощо.

Отже, кожен інструмент мережевого аналізатора не призводить до всіх мережевих параметрів. Оскільки інструмент Colasoft кращий за Wireshark у матричних та графічних звітах, Wireshark - це відкритий вихідний код, який легко розробляти та налаштовувати код всім відповідно до їх потреб, і він сумісний з різними платформами, такими як Linux та Операційні системи MS Windows, а Colasoft працює лише на операційних системах MS. З іншого боку, інструмент TCPDump - це легкий інструмент, який займає невеликий розмір простору і це конкурентна особливість, тому найкращим варіантом його дистанційного використання для моніторингу комп'ютерних мереж за допомогою інтерфейсу командного рядка. Іншим важливим фактором параметра є кількість протоколів, що підтримуються засобом сніферів пакетів. Wireshark підтримує величезну кількість протоколів понад 1000 протоколів, що є чудовим інструментом для моніторингу та контролю мереж, що використовується в різнорідних мережах, які мають різноманітні протоколи, включаючи відео та аудіо програми, в той час як інші інструменти мережевого моніторингу підтримують лише декілька таких протоколів оскільки інструмент Colasoft підтримує близько 300 протоколів, а TCPDump підтримує протокол TCP / IP і не підтримує протокол транспортного рівня протоколу користувача (UDP).

Крім того, враховуючи інші параметри, такі як вартість, Wireshark і TCPDump є безкоштовними інструментами, в той час як Colasoft є дорогим і дорожчим, ніж інші інструменти. Але Colasoft є більш сильним інструментом для виявлення ненормальних протоколів, що є конкурентною особливістю порівняно з іншими інструментами, такими як Wireshark, які лише попереджають. Інструмент Colasoft розроблений командою Capsa, який став хорошим графічним інтерфейсом і має більше функцій безпеки. Інтерфейс фільтрації також розглядається як конкурентна особливість в інструменті Colasoft, який є з графічним інтерфейсом та зручним для користувача, що дозволяє користувачеві легко фільтрувати та аналізувати протоколи та трафік даних. В таблиці 2.2 показано найкраще використання інструменту дослідження пакетів для кожного ресурсу мережі.

Найкраще використовувати Colasoft для сигналів про ненормальні та підроблені пакети, це забезпечує більшу безпеку та інтерфейси GUI. Хоча Wireshark підходить для навчання програмістами та розробниками, завантажуючи вихідний код і налаштовуйте його відповідно до потреб мережевого моніторингу. Інструмент TCPDump більше підходить для віддаленого логічного контролю доступу для моніторингу мережі за допомогою CLI.

Таблиця 2.2 - Найкраще використання інструментів обробки пакетів; Інструменти Colasoft, Wireshark та TCPDump.

Мережеві параметри	**Сніферні інстументи**
Захист комп’ютерних мереж	Colasoft
GUI	
Виявлення нетипових пакетів	
Мережеві сповіщення	
Розмір пакетів	
Мережеве спілкування	
Кілька або один інтерфейс	

Декодування протоколів Hex, ASCII, EBDIC	
Виявлення ненормальних пакетів	
Визначення пакетів з підробленими даними	
Відображення шарів протоколів додатку у OSI 7 моделі	
ПІдтримка OS	Wireshark
Налаштування і розробка всіма розробниками	
Час відповіді	
Пропускна здатність	
Швидкість обробки	
Число протоколів, що підтримуються	
Портативний і простий контроль віддаленого доступу	TCPDump

Крім того, деякі інші автори перевірили деякі мережеві параметри для порівняння між засобами дослідження пакетів; результат показаний у наступних пунктах.

- час відповіді

Він визначається як тривалість періодів часу (вимірюється у одиницях часу) для певної конкретної події. Автори роблять висновок, що час реакції Wireshark менше часу реакції Colasoft.

- пакети за секунди (PPS)

Це кількість переданих пакетів за одну секунду. Добре видно, що Wireshark має менші втрати пакетів, ніж Colasoft. Отже, Wireshark вважається кращим порівняно з Colasoft для повторної передачі пакетів.

- розподіл розміру пакету

Менший розмір пакетів може призвести до меншої напруги в мережі, тоді як великий розмір пакетів збільшує навантаження на мережу. Після експерименту вони дійшли висновку, що розмір пакету довжини у Wireshark становить 558,76 байт, тоді як у Colasoft - 434 . Отже, Colasoft надсилає довжину пакету середнього

розміру; це краще, ніж інструмент Wireshark для завантаження комп'ютерної мережі. Colasoft Capsa не підкреслює систему та мережу.

- пропускна здатність (біт на секунду)

Це обсяг даних, оброблений системою, виміряний у секунду. Після експерименту показано, що пропускна здатність у Colasoft Capsa є великим діапазоном і він швидко змінюється. Ці випадкові зміни погано впливають на систему мережі, оскільки вони заважають роботі системи та комп'ютерної мережі. Тоді як Wireshark має системність та хороший діапазон моделей мережі комп'ютера. Середній біт на секунду (bps) у Wireshark становить 115,398 кбіт / с, а в Colasoft - 6,34 кбіт / с. Отже, Wireshark має більшу пропускну здатність більше, ніж Colasoft Capsa, тоді Wireshark має більшу продуктивність з постійними коливаннями, а також, не бачачи високих відсічок Bps.

2.4 FATT (Fingerprint all the things)

2.4.1 Опис FATT

На основі попереднього аналізу існуючих рішень з точки зору функціоналу пакетного сніфера було створено власний — FATT(Fingerprint all the things). Розроблено скрипт для вилучення метаданих та відбитків. Загальний опис створеного продукту:

- розроблене рішення є кросплатформним (Linux, macOS та Windows).

- з можливістю вилучення метаданих та цифрових відбитків з різних джерел трафіку з поточного та файлу пакетних даних мережі (.pcap).

- реалізована підтримка протоколів SSL/TLS, SSH, RDP, HTTP. Для вилучення цифрових відбитків пристроїв було обрано такі методи реалізації для відповідних протоколів:

1. JA3 для протоколу TLS.

JA3 — це метод створення відбитків SSL/TLS, який повинен бути легким для інтеграції на будь-якій платформі і може бути легко застосований для дослідження загроз. Це набагато ефективніший спосіб виявлення зловмисної активності через SSL, ніж індикатори компрометації на основі IP або домену. Оскільки JA3 виявляє клієнтську програму, не має значення, чи зловмисне програмне забезпечення використовує DGA (алгоритми генерації домену) або різні IP-адреси для кожного хоста, JA3 може виявити саме шкідливе програмне забезпечення на основі того, як проходить взаємодія, а не на тому, які компоненти взаємодіють. JA3 також є чудовим механізмом виявлення в замкнених середовищах, де дозволено встановлювати лише кілька конкретних програм. У таких типах середовищ можна скласти білий список очікуваних програм, а потім попередити про будь-які інші звернення JA3.

2. HASSH для протоколу SSH

"HASSH" - це мережевий стандарт відбитків, який може бути використаний для ідентифікації конкретних реалізацій клієнта та сервера SSH. Відбитки можна легко зберігати, шукати та ділитися ними у вигляді відбитка MD5. "hassh" і "hasshServer" - хеші MD5, побудовані з певного набору алгоритмів, які підтримуються різними SSH-клієнтськими та серверними програмами. Ці алгоритми обмінюються після початкового тристороннього handshake-у TCP як пакети з чітким текстом, відомі як повідомлення "SSH_MSG_KEXINIT", і є невід'ємною частиною налаштування остаточного зашифрованого каналу SSH. Існування та впорядкування цих алгоритмів є досить унікальним, щоб його можна було використовувати як відбиток, щоб допомогти визначити базовий додаток для клієнтів і серверів або унікальну реалізацію, незалежно від нібито ідентифікаторів вищого рівня, таких як рядки "Клієнт" або «Сервер» (рисунок 2.4).

3. RDFP для протоколу RDP.

RDFP - експерементально розроблений RDP відбиток для стандартного RDP протоколу(для інших варіаій безпеки RDP-протоколу використовується TLS, таким чином відбтиток може бути захаплений за допомогою JA3).

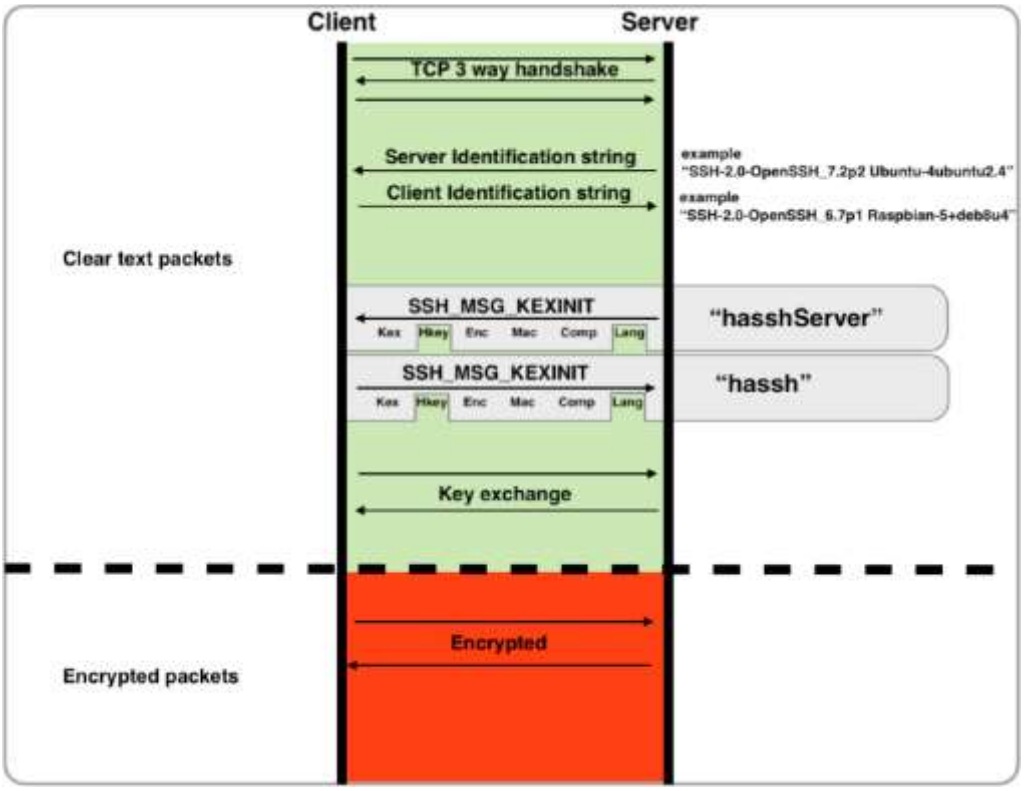

Рисунок 2.4 - Принцип роботи HASSH

2.4.2 Використання FATT

Основним варіантом використання даного ПЗ це моніторинг приманок (honeypots), також продукт можна використовувати для таких цілей, які перед нами ставить мережевий криміналістичний аналіз.

Як користуватися fatt: usage: fatt.py [-h] [-r READ_FILE]

[-d READ_DIRECTORY] [-i INTERFACE]

[-fp [{tls,ssh,rdp,http,gquic}[{tls,ssh,rdp,http,gquic} ...]]] [-da DECODE_AS]

[-f BPF_FILTER]

[-j][-o OUTPUT_FILE]

[-w WRITE_PCAP] [-p]

Необов’язкові аргументи:

- -r, --read_file — ім’я файлу пакетних даних мережі(.pcap)
- -d, --read_directory — директорія файлу пакетних даних мережі(.pcap);
- -i, --interface — інтерфейс, який досліджується;

- -fp, --fingerprint — види протоколів, які включені(за замовчуванням - всі);
- -da, --decode_as — словник типу {decode_criterion_string: decode_as_protocol}, який використовується для того, щоб повідомити tshark як розшифрувати протоколи в незвичайних ситуаціях;
- -f, --bpf_filter — BPF фильтр(лише для режиму поточного трафіку);
- -j, --json_logging — формат виведення JSON;
- -o, --output_file — файл журналу(за замовчуванням fatt.log);
- -w --write_pcap — запис захоплених пакетів з поточного трафіку до файлу пакетних даних мережі;
- -p, --print_output — виведення вихідних данних.

Результати роботи FATT співпали з відповідними у Wireshark - з цього можна зробити висновок про правильність реалізації підтримки відповідних протоколів.

Розглянемо приклади роботи для файлів пакетних даних мережі(.pcap):

- для вразливості CVE-2020-0708 RDP (BlueKeep).
- перевірка за допомогою іншої вразливості CVE-2020-0708 PoC. Цього разу ми не бачимо повідомлення RDP ClientInfo, оскільки PoC використовує TLS (не стандартний протокол безпеки RDP). Таким чином, ми можемо просто побачити повідомлення із запитом на підключення, але якщо ви декодуєте пакет як TLS, ви можете бачити відбитки клієнта TLSHello та JA3. Можна декодувати певний порт як інший протокол.

2.4.3 Вектор розвитку FATT

Вектор розвитку створеного программного продукту:

- Підтримка нових протоколів (ETF QUIC, MySQL, MSSQL, SMTP, SMB).
- Реалізація модуля візуалізації(UI частини програмного продукту).

Розглянувши та порівнявши поведінку вже існуючого програмного забезпечення для аналізаторів мережевого трафіку, таких як Wireshark(раніше відомий як ethereal), TCPDUMP та Colasoft. Кожна з цих програм пропонує різні функції та обмеження для використання відповідного програмного продукта в залежності від потреб.

Аналізуючи існуючі системи, розробили відповідний аналог з базовим функціоналом, але з унікальними методами створення відбитків для протоколів, ми зробили наступні висновки:

- сніфери видають лише журнал даних, який повинен аналізувати мережевий адміністратор, щоб знайти помилку або атаку на мережевий адаптер;

- поточні системи здатні показувати лише журнали пакетів;

- обмеження аналізу на основі протоколу включають той факт, що це надзвичайно трудомістко захопити кожен пакет, вивчити їх, розібрати кожен та вручну здійснити дію на основі інтерпретацій аналізу.

3 ВІЗУАЛІЗАЦІЯ ДАНИХ NETFLOW НА ОСНОВІ ГРАФІКІВ

В даному розділі представлено інноваційний підхід до обробки та візуалізації даних NetFlow. Даний метод візуалізації на основі графів заповнює прогалину між візуалізацією високоагрегованої інформації, представленої у вигляді діаграм, і детальною інформацією, представленою у вигляді лог-файлів. У представленому методі візуалізації вузли графа означають мережеві пристрої, а орієнтовані ребра представляють зв'язок між цими пристроями. В роботі також представлено утилізацію зовнішніх джерел даних (DNS, імена портів тощо), що допомагає представити дані NetFlow більш інтуїтивно зрозумілим чином. Таким чином, цей підхід є дуже природним як для мережевих адміністраторів, так і для неспеціалістів. На основі цих методів було розроблено інструмент підтвердження концепції під назвою NetFlow Visualizer, який зараз пропонується як плагін для датчиків NetFlow.

Використання даних NetFlow у сфері моніторингу мереж зростає. Доступні інструменти для обробки даних NetFlow, такі як NFSen, оснащені великомасштабною візуалізацією, представленою у вигляді графіків (рис. 3.1), з одного боку, і дуже детальною візуалізацією, представленою у вигляді списків лог-файлів даних NetFlow, з іншого боку (рис. 3.2).

Цих методів недостатньо для аналізу мережевого трафіку і що між цими двома методами існує прогалина. Графіки можуть дати аналітику повну картину ситуації в мережі. Перегляд лог-файлів дає аналітику всі деталі. Однак, маючи тисячі зареєстрованих з'єднань, обробляти ці дані надзвичайно складно, особливо коли аналітик не знає, що саме він/вона шукає.

Тому в цій роботі представлено метод візуалізації даних NetFlow на основі графів (рисунок 3.3), який може забезпечити візуалізацію на основі графів. Представлений метод фокусується на мережевих пристроях (вузлах графа) і зв'язку між цими пристроями (орієнтовані ребра), агрегованих на різних рівнях.

Цей масштабований рівень деталізації (рівень агрегації) підходить для аналізу мережевого трафіку, де аналітик бачить повну картину ситуації в мережі і може сфокусуватися на кожній окремій передачі (потоці) даних одразу.

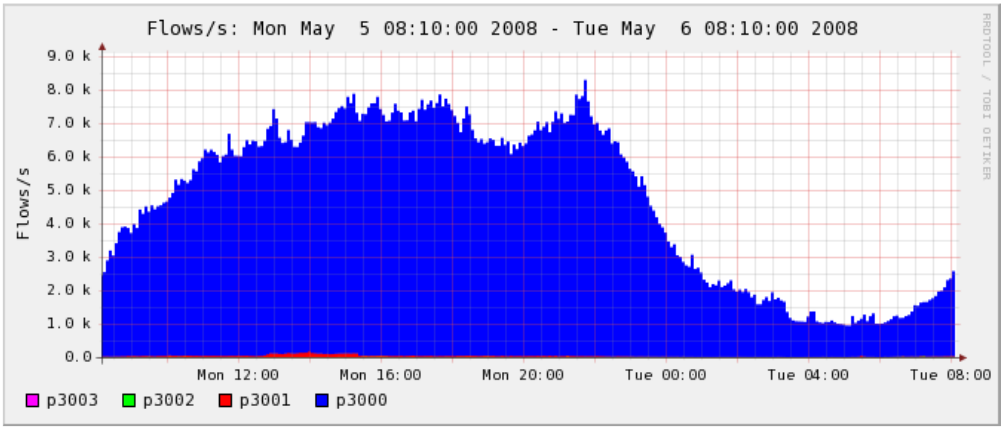

Рисунок 3.1 - Приклад візуалізації діаграми (на основі даних NetFlow)

```
** nfdump -M /data/nfsen/profiles-data/live/p3001  -T  -r 2008/05/05/nfcapd.200805052010 -a  -A srcip,dstip,dstport -o long -c 50
nfdump filter:
any
Date flow start          Duration Proto      Src IP Addr:Port           Dst IP Addr:Port   Flags Tos  Packets    Bytes Flows
2008-05-05 20:09:58.181    1.965     0       [redacted]17.26:0   ->      [redacted]5.37:80    .A....   0       18      720     9
2008-05-05 20:09:58.094    0.138     0       [redacted]35.30:0   ->      [redacted]5.37:80    .AP...   0        3      956     2
2008-05-05 20:09:58.131    1.923     0       [redacted]209.170.0   ->    [redacted]5.37:80    .AP.S.   0       18     1705     3
2008-05-05 20:09:58.282    0.700     0       [redacted]149.16:0   ->     [redacted]5.37:80    .AP.SF   0       29     2262     2
2008-05-05 20:09:58.338    0.015     0       [redacted]5.37:0   ->       [redacted]7.12:59338 .AP..F   0       26    33590     1
2008-05-05 20:09:57.987    1.098     0       [redacted]5.37:0   ->       [redacted]209.170:7897 .AP.SF   0       26    30396     3
2008-05-05 20:09:58.594    0.000     0       [redacted]5.37:0   ->       [redacted]153.203.28418 .A...F   0        4      160     2
2008-05-05 20:09:58.653    0.000     0       [redacted]5.37:0   ->       [redacted]153.203:28638 .A..S.   0        2       96     1
2008-05-05 20:09:58.676    0.000     0       [redacted]5.37:0   ->       [redacted]153.203:28640 .A..S.   0        2       96     1
2008-05-05 20:09:58.677    0.037     0       [redacted]5.37:0   ->       [redacted]153.203:28641 .AP.S.   0        4      918     1
2008-05-05 20:09:57.603    0.145     0       [redacted]5.37:0   ->       [redacted]209.170:7754 .AP..F   0       14    18528     1
2008-05-05 20:09:58.649    0.000     0       [redacted]5.37:0   ->       [redacted]153.203:28637 .A..S.   0        2       96     1
2008-05-05 20:09:58.594    1.631     0       [redacted]153.203:0   ->      [redacted]5.37:80    .AP.SF   0       21     4097     7
2008-05-05 20:09:58.331    1.902     0       [redacted]72.210:0   ->       [redacted]5.37:80    .AP...   0       24     2173     4
2008-05-05 20:09:57.586    2.290     0       [redacted]5.37:0   ->       [redacted]17.26:4172 .A....   0       88   124960     2
2008-05-05 20:09:59.002    0.000     0       [redacted]5.37:0   ->       [redacted]5.101:2514 .A..S.   0        2      120     1
2008-05-05 20:09:59.032    1.165     0       [redacted]5.37:0   ->       [redacted]209.170:10506 .AP.SF   0       50    66434     3
2008-05-05 20:09:58.643    0.090     0       [redacted]5.37:0   ->       [redacted]153.203:28636 .AP.S.   0        6     1104     1
2008-05-05 20:09:55.649    1.330     0       [redacted]20.19:0   ->      [redacted]5.37:80    .A.R..   0       10      400     1
2008-05-05 20:09:52.923    0.000     0       [redacted]5.37:0   ->       [redacted]52.91:41944 .A....   0        2      104     1
2008-05-05 20:09:59.002    1.278     0       [redacted]5.101:0   ->      [redacted]5.37:80    .A..S.   0        3      172     2
2008-05-05 20:09:58.513    0.699     0       [redacted]5.37:0   ->       [redacted]149.16:52981 .AP.SF   0       34    35224     1
Summary: total flows: 50, total bytes: 324311, total packets: 387, avg bps: 352655, avg pps: 52, avg bpp: 838
Time window: 2008-05-05 20:09:51 - 2008-05-05 20:14:57
Total flows processed: 8037, Records skipped: 0, Bytes read: 417936
Sys: 0.002s flows/second: 2680787.2   Wall: 0.037s flows/second: 211834.5
```

Рисунок 3.2 - Приклад лістингу лог-файлу NetFlow

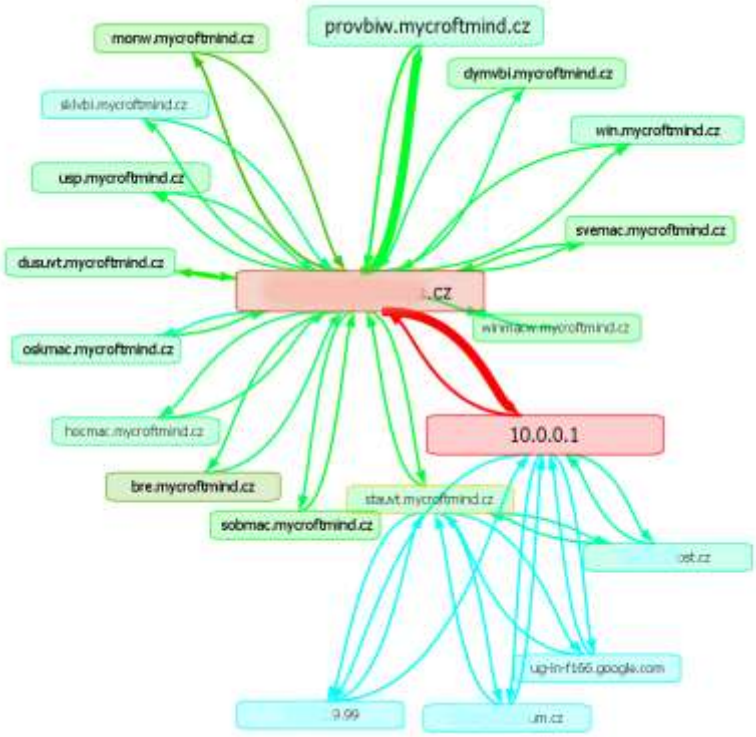

Рисунок 3.3 - Візуалізація даних NetFlow на основі графів

Також дуже важливо зробити якомога менше механічної роботи для аналітика. Інша концепція стосується використання зовнішніх джерел даних. Ключова ідея полягає в тому, щоб надати додаткову інформацію (доменні імена, інформацію про порти тощо), яка допоможе аналітику в процесі аналізу даних. Зауважимо, що це та інформація, яку аналітик шукає вручну під час роботи з файлами журналів NetFlow.

Існує кілька підходів, які використовують графічну візуалізацію в області моніторингу мережі.

Інтерактивна мережева візуалізація активного трафіку (INAV) - це рішення для моніторингу, призначене для використання в мережевих середовищах в режимі реального часу. Воно відстежує поточний активний трафік між вузлами.

Netview - це графічний інструмент візуалізації мережі, який відображає пов'язане зображення мережі в реальному часі. Трафік, помічений netview-

сервером, класифікується, агрегується і, в свою чергу, анімується netview-клієнтом.

jpcap - бібліотека захоплення мережевих пакетів, що забезпечує декомпозицію та графічну візуалізацію мережевого трафіку в реальному часі.

Всі ці рішення використовують схожий метод візуалізації, але вони мають іншу мету, ніж наш підхід. Вони використовуються для моніторингу мережевого трафіку в режимі реального часу і не підтримують порівняння та аналіз різних часових інтервалів. Тому обробляється лише обмежена кількість атрибутів трафіку. Можливості фільтрації також не є достатніми. Вони не надають інформацію WHOIS або додаткову інформацію про порт.

3.1 Властивості методу візуалізації

Мотивацією для даної роботи є програма досліджень і розвитку візуальної аналітики для сприяння передовому аналітичному розумінню.

Основні властивості даного методу візуалізації можна охарактеризувати наступними пунктами (представлено лише фундаментальну функціональність, пов'язану з візуалізацією):

– графічна візуалізація (так звана динамічна візуалізація інтелект-карт) мережевих пристроїв, що взаємодіють між собою. Метод візуалізації, що підходить для представлення з'єднань. Ребра орієнтовані відповідно до напрямку потоку;

– візуалізація деталей комунікації та статистики на основі електронних таблиць;

– багаторівнева деталізація пропонує агрегацію зв'язку між мережевими пристроями за протоколами, деталізацію зв'язку за протоколами та портами призначення або візуалізацію даних NetFlow в чистому вигляді;

– динамічне налаштування візуалізації відповідно до фактичних даних. Забарвлення та розмір вузлів, що представляють мережеві пристрої, та ребер, що

представляють зв'язок між цими пристроями. Забарвлення та розмір динамічно змінюється відповідно до обраних атрибутів записів NetFlow та їх поточних значень (піків та мінімумів), наявних у поточних даних. Наприклад, розмір вузла відповідає кількості пакетів, переданих вузлом, його колір - кількості переданих даних, розмір ребра - кількості переданих потоків, а його колір - кількості переданих пакетів;

– користувацьке налаштування візуалізації для вибраних вузлів, що представляють мережеві пристрої. Ці визначені користувачем пристрої можуть бути візуалізовані з використанням різних форм або розмірів вузлів. Налаштування візуалізації прив'язані до IP-адрес.

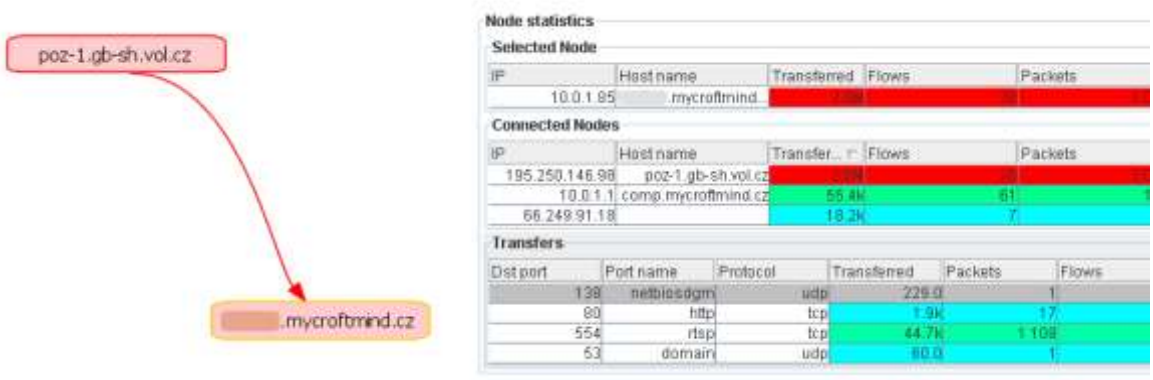

Рисунок 3.4 - Приклад графічної візуалізації, доповненої табличною візуалізацією статистики для обраного вузла

Представлений метод візуалізації побудований на інструментарії візуалізації Prefuse, доповненому стандартними JAVA-компонентами для візуалізації електронних таблиць (рис. 3.4).

3.2 Візуалізатор NetFlow

Представлений метод візуалізації був реалізований компанією Mycroft Mind Inc. Отриманий продукт називається NetFlow Visualizer і надається разом з

датчиками INVEA-TECH Inc. зондами FlowMon у вигляді безкоштовного плагіну. Паралельно розробляється інноваційна версія NetFlow Visualizer.

NetFlow Visualizer - це клієнтська частина клієнт-серверного рішення. Сервер називається NFSel і його призначенням є надання даних NetFlow. NFSel є частиною зонду FlowMon і невидимий для користувачів. NFSel - це XML-RPC сервер, який надає дані NetFlow від збирача даних NetFlow за допомогою стандартного інструменту під назвою NFDump. NFSel перетворює ці дані у формат GraphML. На рисунку 3.5 наведено вичерпну ілюстрацію архітектури.

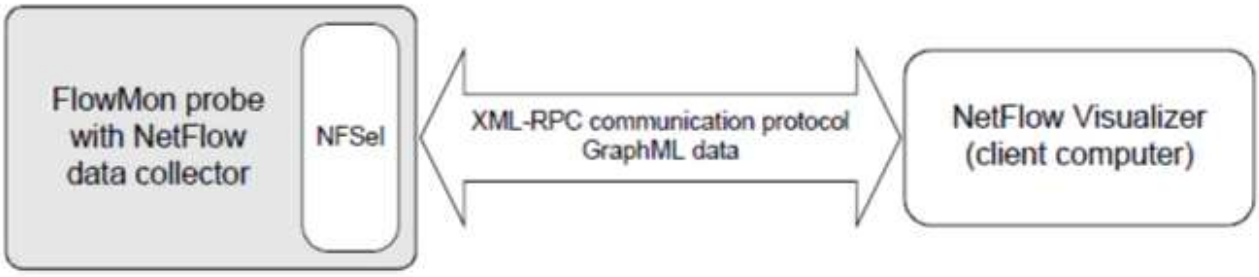

Рисунок 3.5 - Оглядова діаграма архітектури системи

Сам по собі NetFlow Visualizer - це клієнтський Java-додаток, який візуалізує дані NetFlow, надані NFSel за запитом у форматі GraphML. NetFlow Visualizer використовує графічний метод візуалізації і надає користувачеві інтерфейс (рис. 3.6) з додатковими можливостями фільтрації та пошуку, а також використання зовнішніх джерел даних. NetFlow Visualizer надає базові можливості фільтрації, використовуючи параметри чистих даних NetFlow (протоколи, кількість переданих байт, пакетів тощо).

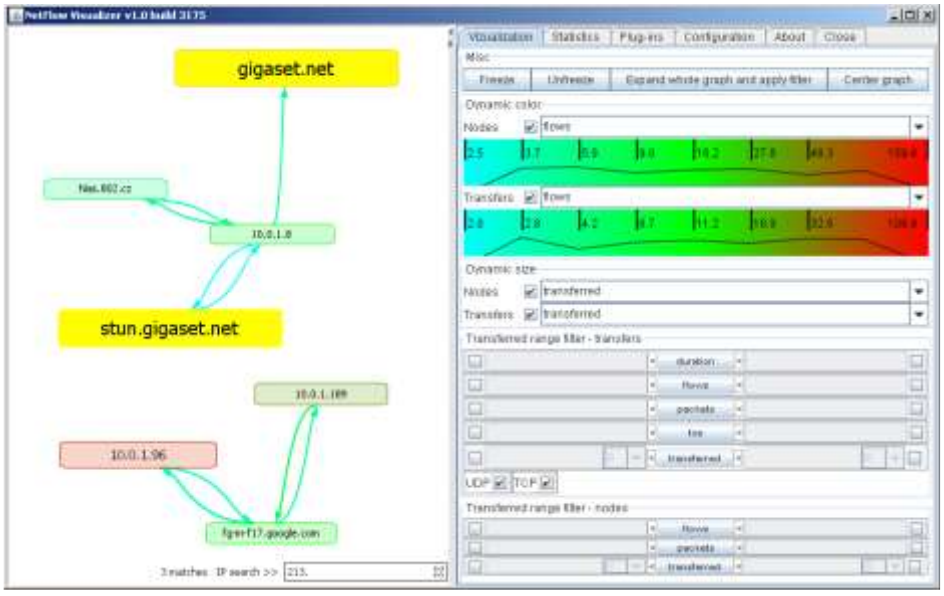

Рисунок 3.6 - Інструмент NetFlow Visualizer. Ілюстрація вкладки властивостей візуалізації та функції пошуку за IP-адресою

Ще одне розширення полягає у використанні зовнішніх джерел даних. Метою використання джерел даних є візуалізація зв'язку в мережі більш природно і зрозуміло для операторів. NetFlow Visualizer намагається зробити якомога більше механічної роботи за аналітика, щоб він міг зосередитися на своїй роботі, а не на пошуку імен портів або перекладі IP-адрес в доменні імена. NetFlow Visualizer використовує наступні джерела даних :

– DNS (Служба доменних імен) - люди звикли працювати з іменами; комп'ютери, однак, використовують числові ідентифікатори. Переклад IP-адрес у відповідні доменні імена має вирішальне значення, особливо у великих мережах. Інформація про доменне ім'я відсутня в даних NetFlow. Тому її слід отримати з відповідного DNS-сервера в Інтернеті.

– Служба WHOIS - іноді виникає необхідність пошуку додаткової інформації про мережевий пристрій, наприклад, його місцезнаходження або адміністративний контакт. Пряма інтеграція служби WHOIS економить час аналітика.

– Назви портів - навіть досвідчені мережеві адміністратори можуть бути не знайомі з незвичними номерами портів. Мотивація подібна до DNS. Ім'я та опис - це набагато більше, ніж номер. Це джерело даних забезпечує переклад пар протокол, порт в ім'я порту та його опис.

4.3 Варіант використання

У цьому розділі ми хотіли б представити простий приклад використання і порівняти NetFlow Visualizer з класичним підходом за допомогою інструментів NFDump і NFSen. Нашим прикладом використання буде дослідження трафіку між найбільшими виробниками/споживачами даних у досліджуваній мереж

Процедура використання з використанням NFDump

1. Складіть запит, щоб отримати топ N виробників або споживачів трафіку. Відмітьте результати. Приклад відповідного запиту до NFDump:

nfdump -M /live/p3000 -T -r nfcapd.200805130405 -n 10 -s ip/bytes

2. Скласти запит для отримання зв'язку пристроїв з попереднього кроку (агрегація за IP-адресою джерела, IP-адресою призначення). Приклад відповідного запиту до NFDump:

nfdump -M /live/p3000 -T -r nfcapd.200805130405 -a -A srcip,dstip,proto IP XXX.YYY.ZZZ.UUU або IP...

3. Складіть запит для отримання зв'язку між вибраними IP-адресами (чисті дані або агреговані за портом призначення). Приклад відповідного запиту NFDump для отримання чистих даних NetFlow:

nfdump -M /live/p3000 -T -r nfcapd.200805130405 IP XXX.YYY.ZZZ.UUU або IP...

Процедура використання з використанням NFSen

1. Отримайте топ N виробників або споживачів трафіку, використовуючи користувацький інтерфейс для топ N статистичних даних (див. рисунок 7).

2. Встановіть фільтр в інтерфейсі користувача для відображення потоків. Скопіюйте IP-адреси, отримані на попередньому кроці, до текстового поля "Фільтр". В інтерфейсі користувача встановіть агрегацію за IP-адресою джерела, IP-адресою призначення та протоколом.

3. Аналогічно до попереднього кроку. Додайте агрегацію з використанням порту призначення або повністю перемістіть агрегацію, щоб отримати чисті дані NetFlow.

Процедура використання з використанням NetFlow Visualizer

1. Встановіть інтервал часу і натисніть кнопку "Завантажити дані", NFSel отримає дані з колектора і доставить їх в NetFlow Visualizer.

2. За допомогою інтерфейсу користувача встановіть фільтр "вузли, що передали більше ніж".

3. Розгорніть одним клацанням миші один або всі вузли, щоб отримати зв'язок між пристроями, що задовольняють фільтру.

Рисунок 3.7 - Інтерфейс користувача NFSen для статистики top N

4. Відкрийте одним клацанням миші вкладку статистики для вибраного пристрою, щоб отримати дані, згруповані за портами призначення (див. рисунок 4), або відкрийте межу між двома пристроями, щоб отримати чисті дані NetFlow у вигляді таблиці (рисунок 3.8), використовуючи її контекстне меню.

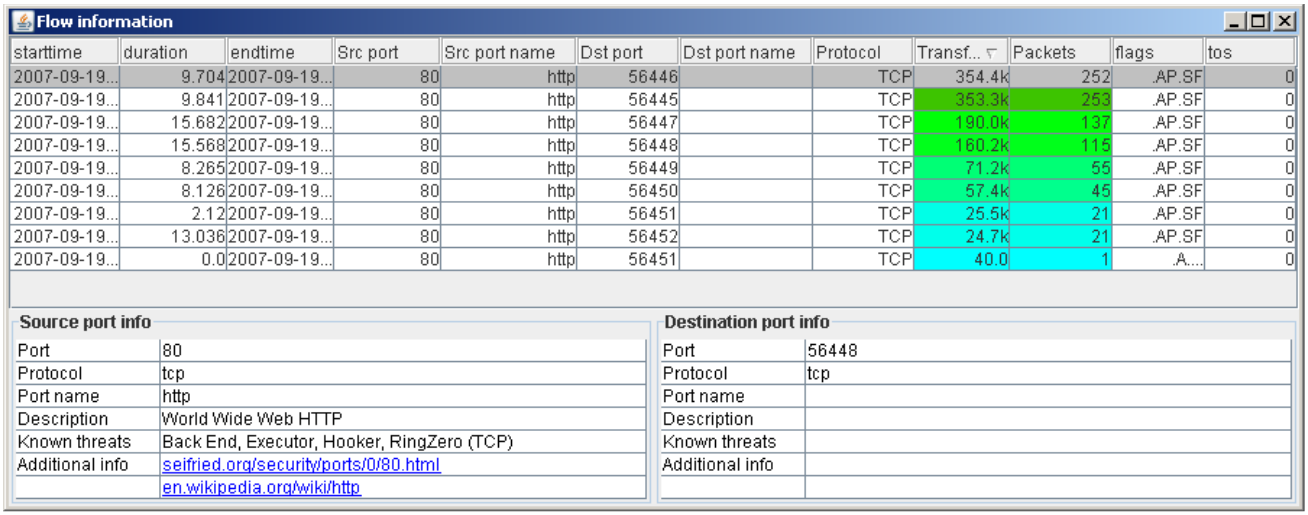

Рисунок 3.8 - Таблиця чистих даних NetFlow (відповіді веб-сервера клієнту

Підіб'ємо підсумки наведеного прикладу використання. Використовуючи NFDump або NFSen, користувачеві доводиться придумувати і писати власні команди або налаштовувати складний фільтр. Очевидно, що використання такого інструменту, як NetFlow Visualizer, може значно підвищити продуктивність праці мережевих аналітиків і зробити аналіз даних NetFlow доступним навіть для неспеціалістів.

Метод візуалізації даних NetFlow на основі графіків був створений, представлений, оцінений і обговорений з експертами в області мереж і безпеки. Основний висновок полягає в тому, що такий метод має великий потенціал для доповнення існуючих методів і є дуже корисним для конкретних випадків використання. Той факт, що цей метод був реалізований комерційною компанією і надається клієнтам як інструмент візуалізації під назвою NetFlow Visualizer з датчиками даних NetFlow, підтверджує його корисність і потенціал.

Представлений метод візуалізації був адаптований в проекті CAMNEP для візуалізації аномального трафіку відповідно до результатів рівня виявлення.

Розробка представленого інструменту візуалізації та самого методу візуалізації ще не завершена. Одна з найбільших проблем полягає в тому, щоб надати аналітику точну порцію необхідної інформації. Дані мережевого трафіку величезні, і швидке розуміння буде неможливим, якщо аналітик буде

перевантажений інформацією. Необхідно надати аналітику потужний та інтуїтивно зрозумілий спосіб визначення і вираження його/її фактичного фокусу. Інший виклик полягає в тому, щоб дозволити будь-яку кількість центрів фокусування, щоб користувачі могли переглядати деталі для більш ніж одного вузла за один раз. Запропоноване рішення полягає у поєднанні візуалізації на основі графіків та електронних таблиць в одному робочому просторі. Таким чином, вузли графіка будуть містити електронні таблиці з деталями.

Під час проекту CAMNEP було отримано експертні знання з аналізу мережевого трафіку. З точки зору майбутніх досліджень і розробок кілька результатів були вирішальними:

Виявлення аномалій на основі передових статистичних методів і математичних моделей добре працює для масових аномалій і загальних відхилень у поведінці мережі, але не має достатньої точності для виявлення цілеспрямованих атак, складних постійних загроз, незначних аномалій або для виявлення конкретних можливих небажаних додатків.

Дані потоку є односпрямованими. TCP-з'єднання представлені двома записами потоку, які безпосередньо не пов'язані між собою. Немає прямої інформації про те, чи є конкретний потік запитом, відповіддю або просто одним потоком без відповіді. Ця неточність може призвести до неправильної інтерпретації статистики трафіку та хибних спрацьовувань при виявленні аномалій.

Візуалізація мережевого трафіку та виявлених аномалій відіграє важливу роль в оцінці аномалій та аналізі трафіку. Простіше кажучи, одна картинка краще, ніж тисяча слів.

4 CAMNEP: СИСТЕМА ВИЯВЛЕННЯ МЕРЕЖЕВИХ ВТОРГНЕНЬ НА ОСНОВІ АГЕНТІВ

Представлено прототип агентної системи виявлення вторгнень, призначеної для розгортання у високошвидкісних магістральних мережах. Основним внеском системи є інтеграція декількох методів виявлення аномалій за допомогою колективного моделювання довіри в групі агентів, що співпрацюють між собою, кожен з яких використовує певний алгоритм виявлення. Аномалії використовуються як вхідні дані для моделювання довіри. На цьому етапі кожен агент визначає довіру до потоку на основі агрегованих аномалій. Агрегація виконується за допомогою розширених моделей довіри, які моделюють довіру до узагальнених локалізованих ідентифікаторів, представлених набором спостережуваних ознак. Система базується на статистиці трафіку у форматі NetFlow, що збирається спеціальними мережевими картами з апаратним прискоренням, і здатна здійснювати спостереження за гігабітними мережами в режимі реального часу.

Сучасне покоління мережевих пристроїв дозволяє в режимі реального часу робити структуровані знімки трафіку в мережі. Ця інформація надається у форматі NetFlow, запровадженому CISCO, і дозволяє спостерігати за окремими потоками в мережі. Потік - це односпрямований компонент TCP-з'єднання (або еквівалент UDP/ICMP), що визначається як набір пакетів з ідентичними IP-адресами джерела та призначення, портами та протоколом. Наявність такої інформації дозволяє розгортати системи аналізу мережевої поведінки (Network Behavior Analysis, NBA), які обробляють цю інформацію і роблять висновки про зловмисність певних груп потоків. Системи NBA не призначені для виявлення прихованих атак на окремі хости, але забезпечують можливість виявлення атак, які є важливими з точки зору мережі, таких як горизонтальне сканування (використовується для створення карти мережі на предмет онлайн-хостів, типових

для розповсюдження шкідливого програмного забезпечення), вертикальне сканування (використовується для визначення послуг, що пропонуються хостом), атаки на відмову в обслуговуванні та інші відповідні події. Крім того, методи, описані в даній системі, також спрямовані на виявлення активності хостів, які були захоплені зловмисником (як правило, за допомогою зомбі-мереж) і використовуються для подальшого вербування або експлуатації зомбі.

Система CAMNEP, представлена в цій роботі, використовує набір методів виявлення аномалій. Ці алгоритми підтримують модель очікуваного трафіку в мережі та порівнюють її з реальним трафіком для виявлення розбіжностей, які ідентифікуються як можливі атаки. Ці методи ефективні проти атак "нульового дня" і раніше невідомих загроз, але страждають від порівняно високого рівня помилок, часто класифікуючи легітимний трафік як зловмисний (помилкові спрацьовування) або не виявляючи зловмисні потоки (помилкові негативні спрацьовування). CAMNEP вирішує цю проблему шляхом використання класичних агентних методів: довіри та репутації для покращення якості класифікацій окремих агентів. Він об'єднує кілька методів, використовуючи колективний процес виявлення, заснований на довірі. Таке поєднання дозволяє співвідносити результати використаних методів і комбінувати їх для підвищення ефективності.

4.1 Архітектура системи

Архітектура системи розділена на кілька рівнів, що відрізняються функціональністю, фізичним розподілом і вимогами до швидкості обробки. Рівень збору трафіку використовує апаратно-прискорені зонди NetFlow для збору трафіку з гігабітних мереж і вилучення значущих ознак для виявлення атак. Потім рівень виявлення класифікує попередньо оброблений трафік і виявляє атаки, які представляються операторам агентами візуалізації з інтерфейсного рівня оператора.

Нижні рівні засновані на апаратних мережевих датчиках і модифікованому програмному забезпеченні з відкритим вихідним кодом, інтелектуальне ядро системи розроблено в рамках мультиагентної платформи AGLOBE, яка полегшує співпрацю агентів і інтеграцію системи під час виконання.

4.1.1 Збір даних про трафік

Рівень збору та попередньої обробки трафіку відповідає за збір мережевого трафіку, попередню обробку даних та їх розподіл на верхні рівні системи. Він використовує тільки характеристики потоку, засновані на інформації з заголовків пакетів, і тому може аналізувати навіть зашифрований трафік.

Тому використовуються апаратні прискорені зонди NetFlow під назвою FlowMon. Зонд FlowMon - це пасивний пристрій моніторингу мережі на базі апаратного забезпечення COMBO, що забезпечує високу продуктивність і точність. Зонд обробляє трафік зі швидкістю 1 Гбіт/с в обох напрямках і експортує необхідні дані NetFlow до різних колекторів. Висока продуктивність карти гарантує надійну роботу навіть в умовах атаки, коли характеристики трафіку роблять обробку більш інтенсивною з точки зору обчислень.

Сервери-колектори зберігають вхідні пакети з даними NetFlow від датчиків FlowMon у своїй внутрішній базі даних. Кожен сервер колектора надає інтерфейс для графічного та текстового представлення необробленого мережевого трафіку, простої фільтрації потоку, агрегації та оцінки статистики, використовуючи IP-адреси, порти та протоколи джерела та призначення. Агенти виявлення підключаються до колектора для отримання даних і використовують їх для виявлення.

Навіть після розгортання зондів у мережі, що моніториться, вони можуть бути перепрограмовані для отримання нових характеристик трафіку. Система повністю реконфігурується, і зонди можуть адаптувати свої функції та поведінку відповідно до змін на рівні агентів.

4.1.2 Виявлення атак довірчими агентами

Мета кооперативного рівня виявлення загроз полягає в тому, щоб забезпечити оцінку шкідливості окремих потоків у кожному наборі потоків, що спостерігаються системою. Для досягнення цієї мети було використано методи моделювання довіри та розширено їх, щоб покрити специфічні потреби домену.

Кожен агент виявлення містить один з методів виявлення аномалій в поєднанні з розширеною моделлю довіри: агент MINDS, який реагує на кількість потоків від та до хостів у мережі та виявляє розбіжності між минулим та поточним трафіком, агент Xu, який міркує про трафік від окремих хостів, використовуючи нормалізовані ентропії та правила, агент Lakhina Entropy, який будує модель, що прогнозує ентропію характеристик трафіку від окремих хостів та ідентифікує аномалії як різницю між прогнозованим та реальним значенням, і, нарешті, агент Lakhina Volume, який застосовує той самий метод до трафіку.

Всі агенти, незалежно від їх типу, обробляють дані, отримані з рівня збору, в три окремі етапи (рисунок 4.1): виявлення аномалії, оновлення довіри та колективний висновок про довіру.

У сфері мережевої безпеки низька довіра до потоку означає, що потік розглядається як частина атаки. Надійність визначається в інтервалі [0, 1], де 0 відповідає повній недовірі, а 1 - повній довірі. Ідентичність кожного потоку визначається ознаками, які можна спостерігати безпосередньо на потоці: srcIP, dstIP, srcPrt, dstPrt, протокол, кількість байт і пакетів. Якщо два потоки в наборі даних мають однакові значення цих параметрів, вони вважаються ідентичними. Контекст кожного потоку визначається особливостями, які спостерігаються на інших потоках в тому ж наборі даних, наприклад, кількість схожих потоків з того ж srcIP, або ентропія dstPrt всіх запитів з того ж хоста, що і оцінюваний потік. Хоча агенти в представленій системі використовують однакове представлення ідентичності, контекст визначається ознаками, які використовуються відповідними методами виявлення аномалій для того, щоб зробити висновки

щодо аномальності потоку. Ідентичність і контекст використовуються для визначення простору ознак, метричного простору, на якому працює модель довіри кожного агента. Метрика простору описує схожість між ідентичностями та контекстами потоків і є специфічною для кожного агента.

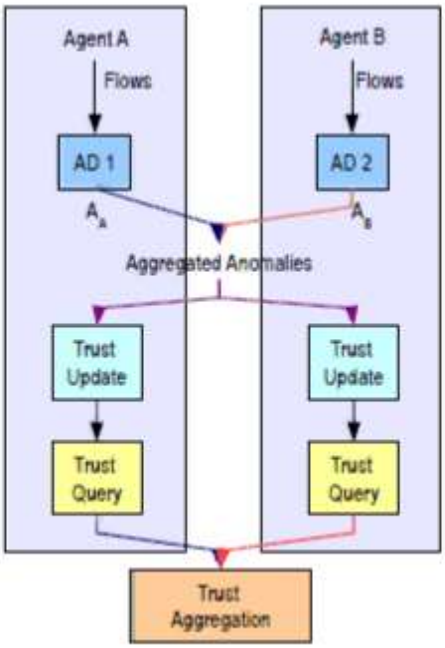

Рисунок 4.1 - Робота рівня агента

Виявлення аномалій. На етапі виявлення аномалій агенти використовують вбудований метод виявлення аномалій для визначення аномалії кожного потоку як значення в інтервалі [0, 1], де 1 означає максимальну аномалію, а нуль - відсутність аномалії взагалі. Значення аномалії передаються іншим агентам виявлення і використовуються як вхідні дані на другому етапі обробки.

Оновлення довіри. Під час оновлення довіри агенти інтегрують значення аномалій, визначені для окремих потоків на першому етапі, у свої моделі довіри. Оскільки міркування про довіру до кожного окремого потоку є теоретично неможливими і непрактичними (потоки є однократними подіями за визначенням), модель зберігає довіру до значущих зразків потоків (наприклад, центроїдів (нечітких) кластерів) у просторі ідентичності-контексту, а аномалія кожного потоку використовується для оновлення довіри до центроїдів у його околицях.

Вага, яка використовується для оновлення довіри до центроїда за допомогою значень аномалії, наданих для потоку, зменшується з віддаленням від центроїда. Таким чином, оскільки кожен агент використовує окрему функцію відстані, кожен агент має свій власний погляд на ситуацію в проблему. Потоки кластеризуються за різними критеріями, а перехресна кореляція, реалізована шляхом спільного використання значень аномалій, що використовуються для оновлення довіри, допомагає усунути випадкові аномалії.

Колективна оцінка довіри. На останньому етапі обробки кожен агент визначає ступінь довіри до кожного потоку (з необов'язковим кроком нормалізації), всі агенти надають свою оцінку довіри (концептуально - думку про репутацію) агентам агрегації та агентам візуалізації, а потім агреговані значення можуть бути використані для фільтрації трафіку.

Для того, щоб бути успішним, довіра, агрегована системою, повинна бути якомога ближчою до зловмисності потоку. Коли ми розглядаємо зловмисні та ненадійні потоки як множини (насправді це нечіткі множини), ми хочемо, щоб вони якомога більше перекривали один одного. Ми можемо визначити типові помилки класифікації, використовуючи довіру та зловмисність потоку. Зловмисні потоки, яким довіряють, позначаються як хибнонегативні, а потоки, яким не довіряють, але які є легітимними, позначаються як хибнопозитивні. Зазвичай, коли ми налаштовуємо систему на зменшення одного з цих наборів, розмір іншого збільшується. Інтуїтивно може здатися, що ми готові ігнорувати більшу кількість хибних спрацьовувань, ніж хибних відмов. Однак це рідко трапляється в системах IDS, розгорнутих для оперативного використання, оскільки легальний трафік значно перевищує кількість атак, і навіть низький відсоток хибних спрацьовувань робить систему непридатною для використання.

Важливість моделі довіри полягає в перехресній перевірці висновків про аномалії в моделях довіри агентів виявлення, кожна з яких базується на різних умовах трафіку. Для того, щоб класифікувати набір потоків як атаку, потоки з цього набору повинні знаходитися в околицях центроїдів з низькою довірою в

моделях більшості агентів. На практиці більшість потоків атак потрапляють в околиці одного центроїда, як ми бачимо на рис. 4.2. З іншого боку, коли один з агентів створює хибнопозитивний результат, потоки, ймовірно, розсіюються в моделях довіри інших агентів, і при остаточному об'єднанні переважає в середньому вища достовірність пов'язаних центроїдів.

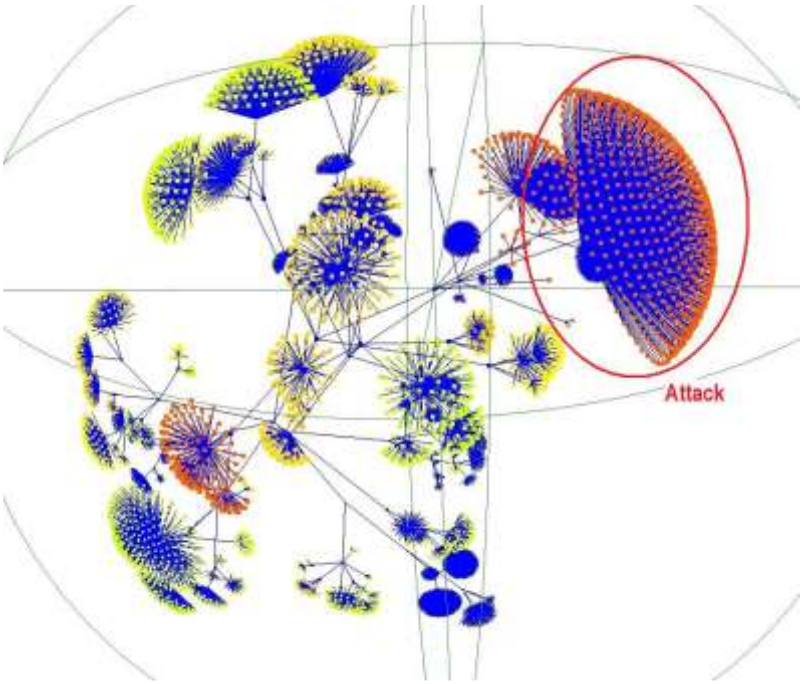

Рисунок 4.2 - Погляд на модель довіри агента виявлення. Потоки відображаються у вигляді гілок дерева, прикріплених до найближчого центроїда моделі довіри. Слід зазначити, що потоки атак сконцентровані біля одного центроїда

4.2 Рівень інтерфейсу оператора та аналітика

Рівень інтерфейсу оператора та аналітика в системі CAMNEP представлений агентом Visio Agent. Цей агент забезпечує візуалізацію мережевого трафіку на основі графів, де вузли представляють конкретні хости, а орієнтовані ребра відображають мережеві потоки. Візуалізація на основі графів доповнюється високорівневою візуалізацією за допомогою гістограми довіри та швидким поглядом на характеристики трафіку, що надається інструментом

статистичного аналізу. Крім того, що візуалізатор можна використовувати для представлення всіх даних, він дозволяє вибірково візуалізувати вибрані користувачем групи потоків з гістограми та/або компонентів аналізу.

Visio Agent забезпечує візуальну підтримку аналітичних міркувань з використанням результатів рівня виявлення. Візуалізація мережі на основі графіків апроксимує структуру топології мережі та зв'язку, тому є природною як для мережевих адміністраторів, так і для неспеціалістів. Вона також автономно збирає допоміжні дані від імені своїх користувачів.

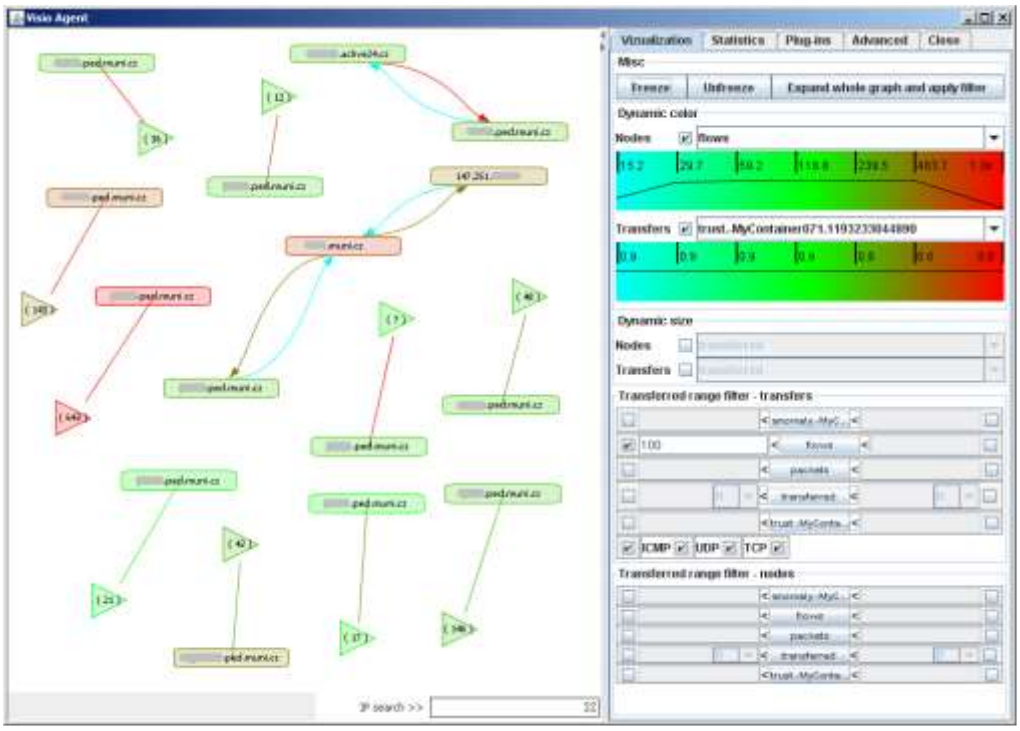

Рисунок 4.3 - Приклад загальної ситуації в мережі із застосованим фільтром.

Графічне представлення трафіку на основі графів розширено кількома важливими функціями. Користувач може перерахувати потоки і статистику трафіку, пов'язану з кожною межею/хостом, і відобразити пов'язану з ними довіру. Трафік може бути відфільтрований і агрегований за багатьма відповідними характеристиками, включаючи довіру і значення аномалій. Візуальні атрибути відображення (такі як розмір і колір вузла/ребра) також можуть адаптуватися до цих характеристик, що полегшує орієнтацію

користувача. Інформація, надана третіми сторонами (DNS, whois), легко інтегрується у візуалізацію.

Оскільки сучасний мережевий трафік є безмасштабною мережею, особливо важливою є візуалізація супервузлів, тобто вузлів з великою кількістю з'єднань. Ці вузли є типовими для багатьох сценаріїв атак, а також для цінних цілей. Тому візуалізатор замінює одномоментні з'єднання до/від цих вузлів спеціальним представленням "хмари" трафіку і виділяє лише ті вузли, які також з'єднуються з іншими вузлами мережі, що спостерігається.

4.3 Оцінка системи

Будь-яка система виявлення вторгнень оцінюється з точки зору хибних спрацьовувань/хибних відмов. Було проведено кілька серій експериментів, як з відомими атаками з добре визначеними характеристиками, так і з реальними атаками, що спостерігалися на захищених мереж. У цьому розділі представлено добірку результатів.

У першій серії експериментів було виміряно здатність системи виявляти мікс атак, включаючи вертикальне та горизонтальне сканування (TCP SYN, TCP CONNECT та UDP), атаки грубої сили на SSH-паролі, відбитки пальців ОС та інші. Надійність, присвоєна атакам, показана на рис. 4.4, де було виділено атаки, які не були виявлені системою. Атака вважається виявленою, якщо (її надійність нижче 0.2, або якщо надійність більш ніж на один σ нижче середнього значення розподілу надійності. Можна бачити, що система стабільно виявляє атаки з більш ніж 400 потоками протягом 5 хвилин - це відповідає приблизно 1% від обсягу трафіку, виміряного в кількості потоків.

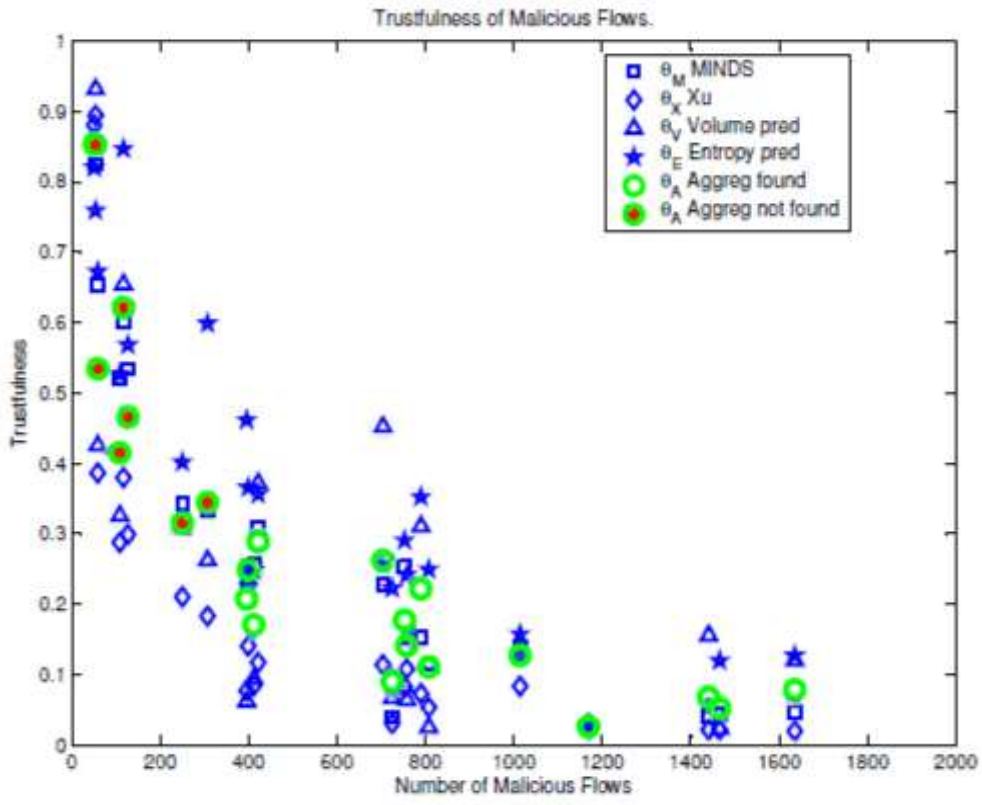

Рисунок 4.4 - Середня довіра до потоків атак в залежності від розміру атаки

Було проведено оцінку алгоритму на 30-хвилинній вибірці реального мережевого трафіку. Продуктивність системи дещо вища, ніж при ручному аналізі в автономному режимі, який виконувався досвідченим адміністратором і зайняв більше одного дня. Звірка результатів показала, що система має дещо нижчий відсоток хибних спрацьовувань і дещо вищий відсоток істинних спрацьовувань, але основні атаки (вузли бот-мережі та переповнення буфера) були виявлені стабільно. В обох випадках хибні спрацьовування становили приблизно половину зареєстрованих інцидентів (із загальної кількості 17).

Якщо порівняти результати системи з результатами інтегрованих методів виявлення аномалій за показниками хибних спрацьовувань/хибних відмов, то агреговані результати перевершують будь-який з методів за обома критеріями (у співвідношенні 10 при врахуванні джерел трафіку), а також перевершують класифікацію за усередненими аномаліями в 2 рази за FP, виявляючи при цьому більше справжніх атак.

У даній роботі представлено мультиагентний фреймворк, який дозволяє інтегрувати декілька існуючих методів аналізу поведінки в мережі. Агентні методи використовуються не лише на рівні коду та інтеграційних фреймворків, але й як логічне ядро підходу, що ґрунтується на традиційному моделюванні довіри та простому механізмі репутації. Експериментальні результати на реальному трафіку, а також оцінка, проведена мережевими адміністраторами, натякає на те, що ця комбінація є важливою не тільки з точки зору досліджень, але й з промислової точки зору.

Той факт, що система присвоює оцінку довіри, а не бінарну мітку (атака/легітимність), як це роблять класичні IDS-системи, насправді є перевагою. Надійність разом з кількістю потоків також є гарною оцінкою пріоритету, який вимагає атака. Основним результатом роботи системи є гістограма поточної довіри до трафіку. Ця форма дуже зручна для швидкого аналізу. Після того, як трафік класифіковано, система залишає подальші кроки аналізу на оператора. У поточній версії системи методи виявлення аномалій підібрані для боротьби з одними і тими ж типами атак, хоча і з різною ефективністю. Це дуже сильне обмеження, і ми прагнемо послабити його в майбутньому, запровадивши більш досконалий механізм репутації замість простого усереднення.

ВИСНОВКИ

Досліджено процес аналіза трафіку. На основі цього дослідження проведено порівняльний аналіз існуючих засобів аналізу трафіку. Реалізовано базовий функіонал пакетного аналізатора для деяких мережевих протоколів з унікальною реалізацією методів створення відбитків.

Аналізатори мережевого трафіку, як правило, мають модульну архітектуру: з часом з'являються нові протоколи, і їх необхідно підтримувати. Розширювати систему, в якій функції розбору даних всіх протоколів зосереджені в одному функціональному модулі, важко. У разі модульної архітектури для кожного протоколу створюється окремий модуль, в якому визначаються методи та структури даних для роботи з цим протоколом. Виникає додатковий питання про залежності: при додаванні нового модуля необхідно «повідомити» про його існування іншим. Вносити зміни в код існуючих модулів неефективно: може бути порушена логіка їх роботи або внесені помилки, налагодження яких важке. До того ж потрібна повторна збірка змінених модулів. Тому необхідно мінімізувати кількість внесених в існуючу розробку змін, необхідних для додавання підтримки нового протоколу.

Відзначимо, що деякі практичні завдання вирішуються за допомогою аналізу файлу зі збереженим трафіком (будемо називати такий файл мережевою трасою):

- відтворення помилок розбору;
- розробка (налагодження) розбирачів;
- розслідування інцидентів порушення інформаційної безпеки. Тому вкрай важливо забезпечити можливість використання «результатів» offline-аналізу для роботи в режимі online, тобто перенесення модулів розбору між інструментами offline- і online-аналізу.

Необхідність в проведенні розбору заголовків мережевих пакетів, як вже було зазначено, виникає при вирішенні багатьох практичних завдань. Важливо

розуміти, що фахівець в області мережевої безпеки в загальному випадку може не мати високі навичками в програмуванні. Тому необхідно надати високорівневий інтерфейс (API), що дозволяє користувачеві підтримувати в рамках системи нові (зокрема закриті) мережеві протоколи.

Розглянувши та порівнявши поведінку вже існуючого програмного забезпечення для аналізаторів мережевого трафіку, таких як Wireshark (раніше відомий як ethereal), TCPDUMP та Colasoft. Кожна з цих програм пропонує різні функції та обмеження для використання відповідного програмного продукта в залежності від потреб.

Аналізуючи існуючі системи, в роботі було розглянуто відповідний аналог з базовим функціоналом, але з унікальними методами створення відбитків для протоколів, можна зробити наступні висновки:

- сніфери видають лише журнал даних, який повинен аналізувати мережевий адміністратор, щоб знайти помилку або атаку на мережевий адаптер.

- поточні системи здатні показувати лише журнали пакетів.

- обмеження аналізу на основі протоколу включають той факт, що це надзвичайно трудомістко захопити кожен пакет, вивчити їх, розібрати кожен та вручну здійснити дію на основі інтерпретацій аналізу.

Метод візуалізації даних NetFlow на основі графів був створений, представлений, оцінений і обговорений з експертами в області мереж і безпеки. Основний висновок полягає в тому, що такий метод має великий потенціал для доповнення існуючих методів і є дуже корисним для конкретних випадків використання. Той факт, що цей метод був реалізований комерційною компанією і надається клієнтам як інструмент візуалізації під назвою NetFlow Visualizer з датчиками даних NetFlow, підтверджує його корисність і потенціал.

Представлений метод візуалізації був адаптований в проекті CAMNEP для візуалізації аномального трафіку відповідно до результатів рівня виявлення.

Розробка представленого інструменту візуалізації та самого методу візуалізації ще не завершена. Одна з найбільших проблем полягає в тому, щоб надати аналітику точну порцію необхідної інформації. Дані мережевого трафіку

величезні, і швидке розуміння буде неможливим, якщо аналітик буде перевантажений інформацією. Необхідно надати аналітику потужний та інтуїтивно зрозумілий спосіб визначення і вираження його/її фактичного фокусу. Інший виклик полягає в тому, щоб дозволити будь-яку кількість центрів фокусування, щоб користувачі могли переглядати деталі для більш ніж одного вузла за один раз. Запропоноване рішення полягає у поєднанні візуалізації на основі графів та електронних таблиць в одному робочому просторі. Таким чином, вузли графіка будуть містити електронні таблиці з деталями.

Під час проекту CAMNEP було отримано експертні знання з аналізу мережевого трафіку. З точки зору майбутніх досліджень і розробок кілька результатів були вирішальними:

Виявлення аномалій на основі передових статистичних методів і математичних моделей добре працює для масових аномалій і загальних відхилень у поведінці мережі, але не має достатньої точності для виявлення цілеспрямованих атак, складних постійних загроз, незначних аномалій або для виявлення конкретних можливих небажаних додатків.

Дані потоку є односпрямованими. TCP-з'єднання представлені двома записами потоку, які безпосередньо не пов'язані між собою. Немає прямої інформації про те, чи є конкретний потік запитом, відповіддю або просто одним потоком без відповіді. Ця неточність може призвести до неправильної інтерпретації статистики трафіку та хибних спрацьовувань при виявленні аномалій.

Візуалізація мережевого трафіку та виявлених аномалій відіграє важливу роль в оцінці аномалій та аналізі трафіку.

У даній роботі представлено мультиагентний фреймворк, який дозволяє інтегрувати декілька існуючих методів аналізу поведінки в мережі. Агентні методи використовуються не лише на рівні коду та інтеграційних фреймворків, але й як логічне ядро підходу, що ґрунтується на традиційному моделюванні довіри та простому механізмі репутації. Експериментальні результати на реальному трафіку, а також оцінка, проведена мережевими адміністраторами,

говорить про те, що ця комбінація є важливою не тільки з точки зору досліджень, але й з промислової точки зору.

Той факт, що система присвоює оцінку довіри, а не бінарну мітку (атака/легітимність), як це роблять класичні IDS-системи, насправді є перевагою. Надійність разом з кількістю потоків також є гарною оцінкою пріоритету, який вимагає атака. Основним результатом роботи системи є гістограма поточної довіри до трафіку. Ця форма дуже зручна для швидкого аналізу. Після того, як трафік класифіковано, система залишає подальші кроки аналізу на оператора. У поточній версії системи методи виявлення аномалій підібрані для боротьби з одними і тими ж типами атак, хоча і з різною ефективністю.

ПЕРЕЛІК ПОСИЛАНЬ

1. Arnold, J., and Kaashoek, M. F. Ksplice: Automatic Rebootless Kernel Updates. In Proceedings of the 4th ACM European Conference on Computer Systems (2019), EuroSys ’19, pp. 187–198.

2. Anthony Liguori. Powering Next-Gen EC2 Instances: Deep Dive into the Nitro System. https://www.youtube.com/watch?v=e8DVmwj3OEs, November 2018

3. Cooper, B. F., Silberstein, A., Tam, E., Ramakrishnan, R., and Sears, R. Benchmarking Cloud Serving Systems with YCSB. In Proceedings of the 1st ACM Symposium on Cloud Computing (2020), SoCC ’20, pp. 143–154.

4. Cortez, E., Bonde, A., Muzio, A., Russinovich, M., Fontoura, M., and Bianchini, R. Resource Central: Understanding and Predicting Workloads for Improved Resource Management in Large Cloud Platforms. In Proceedings of the 26th Symposium on Operating Systems Principles (2017), SOSP ’17, pp. 153–167.

5. Google Compute Engine, Nested virtualization overview.https://cloud.google.com/compute/docs/instances/nestedvirtualization/ overview.

6. Google Compute Engine, Preemptible VM Instances. https://cloud. google.com/compute/docs/instances/preemptible.

7. Intel Processor Microcode Package for Linux. https://github.com/intel/Intel-Linux-Processor-Microcode-Data-Files.

8. Intel Software Guard Extensions (Intel R SGX). https://www.intel.com/content/www/us/en/architecture-andtechnology/software-guard-extensions.html.

9. Kpatch: Dynamic Kernel Patching, GitHub Repository. https://github.com/dynup/kpatch.

10. KVM, CPU Hotplug. https://www.linux-kvm.org/page/CPUHotPlug.

11. Open vSwitch. http://openvswitch.org/

12. Open vSwitch, Bonding. https://docs.openvswitch.org/en/latest/topics/bonding/.

13. Openstack web page. https://www.openstack.org/.

14. Physical Memory Model. https://www.kernel.org/doc/html/latest/vm/memory-model.html.

15. QEMU Migration Documentation. https://github.com/qemu/qemu/blob/master/docs/devel/migration.rst.

16. Redis, Web Page. https://redis.io/.

17. SUSE Linux Enterprise Live Patching. https://www.suse.com/products/live-patching/.

18. The Linux x86 Boot Protocol. https://www.kernel.org/doc/Documentation/x86/boot.txt.

19. Virtuozzo Hybrid Server. https://www.virtuozzo.com/virtuozzohybrid-server/.

20. VMware vSphere, Enable CPU Hot Add . https://docs.vmware.com/en/VMware-vSphere/7.0/com.vmware.vsphere.vm_admin.doc/GUID-285BB774-CE69-4477-9011-598FEF1E9ACB.html.

21. Xen Project Software Overview. https://wiki.xenproject.org/wiki/Xen_Project_Software_Overview.

22. Hyper-V Architecture. https://docs.microsoft.com/enus/windows-server/administration/performance-tuning/role/hyper-v-server/architecture

23. Payment Card Industry Data Security Standard, Requirements and Security Assessment Procedures Version 3.2.1. https://www.pcisecuritystandards.org/

24. CRIU, Upstream Kernel Commits. https://criu.org/Upstream_kernel_commits

25. Microsoft Azure, Hypervisor Security on the Azure Fleet. https://docs.microsoft.com/en-us/azure/security/fundamentals/Hypervisor.

ДЕМОНСТРАЦІЙНІ МАТЕРІЛИ