МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра кібербезпеки

Миколишин Павло Петрович

Система запобігання проникненню в мережі Інтернет – речей / The intrusion prevention system for the Internet of Things

спеціальність: 125 – Кібербезпека

освітньо-професійна програма –Кібербезпека

Кваліфікаційна робота

Виконав студент групи КБм -21
П.П. Миколишин

Науковий керівник
к.т.н., доцент Сегін А.І.

Кваліфікаційну роботу допущено
до захисту:

«____» _____ ___2022 р.

Завідувач кафедри
_____ В.В.Яцків

ТЕРНОПІЛЬ – 2022

Факультет комп'ютерних інформаційних технологій
Кафедра кібербезпеки
Освітній ступінь «магістр»
спеціальність: 125 - Кібербезпека
освітньо-професійна програма –Кібербезпека

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ В.В. Яцків
____" _____ 2022 року

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
МИКОЛИШИН Павло Петрович
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи:
Система запобігання проникненню в мережі Інтернет – речей / The intrusion prevention system for the Internet of Things
керівник роботи к.т.н., доцент Сегін А.І.
затверджені наказом по університету від 31 грудня 2021 року № 606
2. Строк подання студентом закінченої випускної кваліфікаційної роботи 16 листопада 2022 року.
3. Вихідні дані до кваліфікаційної роботи: завдання на випускну кваліфікаційну роботу студента, наукові статті, технічна література.

4. Основні питання, які потрібно розробити:
- провести аналіз існуючих систем захисту мережі;
- охарактеризувати методи виявлення проникнень;
- розробка статистичних функцій і функцій безпеки на основі сигнатур;
- розробити хост-систему IPS для захисту мережі інтернет-речей;
- провести експериментальні дослідження ефективності розробленої системи захисту мережі;

5. Перелік графічного матеріалу у роботі.
Механізм виявлення Snort. Блок-схема.
Таблиця вартості компонентів. Таблиця.
Порівняння вартості зберігання. Таблиця.

6. Консультанти розділів кваліфікаційної роботи

	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 11 жовтня 2021 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строки виконання етапів кваліфікаційної роботи	Примітка
1	Аналіз предметної області	12.2021 р. – 03.2022 р.	
2	Система запобігання проникненню в мережі інтернет-речей	03.2022 р. – 05.2022 р.	
3	Розробка системи запобігання проникненню в мережі Інтернет – речей	05.2022 р. – 11.2022 р.	

Студент _____ Миколишин П.П.
(підпис)

Керівник роботи _____ к.т.н., доцент А.І. Сегін
(підпис)

АНОТАЦІЯ

Кваліфікаційна робота на тему «Система запобігання проникненню в мережі Інтернет – речей» на здобуття освітнього ступеня «Магістр» зі спеціальності 125 «Кібербезпека» освітньо-професійної програми «Кібербезпека» написана обсягом 73 сторінки і містить 44 ілюстрації, 3 таблиці, 32 джерело за переліком посилань.

Метою кваліфікаційної роботи є підвищення ефективності системи запобігання проникненню в мережі Інтернет – речей.

Методи досліджень. Для розв'язання поставлених задач у даній кваліфікаційній роботі використано: методи запобігання проникненню, методи захисту мережі, методи проектування.

Результати дослідження: удосконалено систему запобігання проникненню в мережі Інтернет – речей за рахунок використання набору правил.

Розроблено структуру системи запобігання з реалізацією на основі хоста на базі ОС Ubuntu Linux та використанням набору правил.

Результати роботи можуть успішно застосовуватися для різних корпоративніх та домашніх мереж.

Ключові слова: СИСТЕМА ЗАПОБІГАННЯ, МЕРЕЖЕВА СИСТЕМА ЗАПОБІГАННЯ ВТОРГНЕННЯМ, NIPS.

ABSTRACT

The qualification work on the topic "The intrusion prevention system for the Internet of Things" for obtaining the Master's degree in the specialty 125 "Cybersecurity" of the educational and professional program "Cybersecurity" is written in the volume of 73 pages and contains 44 illustrations, 3 tables, 32 sources according to the list links.

The purpose of the qualification work is to improve the effectiveness of the system for preventing intrusion into the Internet of Things network.

Research methods. To solve the tasks in this qualification work, the following methods were used: intrusion prevention methods, network protection methods, and design methods.

The results of the study: the system of preventing intrusion into the Internet of Things has been improved by using a set of rules.

The structure of the prevention system was developed with a host-based implementation based on the Ubuntu Linux OS and using a set of rules.

The results of the work can be successfully applied to various corporate and home networks.

Keywords: PREVENTION SYSTEM, NETWORK INTRUSION PREVENTION SYSTEM, NIPS.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ І ПОЗНАЧЕНЬ

IPS	система запобігання вторгненням
IDS	система виявлення вторгнення
IoT	безпека інтернет-речей
IP	інтернет протокол
HIDS	система виявлення вторгнень на базі хоста
NIDS	система виявлення вторгнень на базі мережі
HIPS	система запобігання вторгнень на базі хоста
NIPS	система запобігання вторгнень на базі мережі
FPGA	програмовані матриці
PR	права половина відкритого тексту.
PL	ліва половина відкритого тексту.
C	зишифрований текст.
CR	права половина зашифрованого тексту.
CL	ліва половина зашифрованого тексту.

ВСТУП

Актуальність роботи. Хакерство є основною проблемою в сучасному світі, пов’язаному через Інтернет. Хакери продовжують знаходити нові способи зламати мережі та викрасти цінну інформацію про жертву або встановити шкідливе програмне забезпечення для моніторингу фінансової діяльності жертви. Порушення безпеки та даних спричиняють величезні фінансові втрати в мільярди доларів для промисловості. Відповідно до нещодавнього звіту журналу Inc., компанії щороку втрачають від хакерів 400 мільярдів доларів, а до 2023 року ці втрати, за оцінками, зростуть до 1 трильйона доларів. Для злому та встановлення шкідливого програмного забезпечення хакери використовують мережу Інтернет речей. Інтернет речей (IoT) — це мережа інтелектуальних різнорідних об'єктів, що швидко росте. Це стосується окремих фізичних пристроїв, які взаємодіють з іншими такими пристроями. На відміну від бездротових сенсорних мереж (WSN), IoT підключається до всесвітньої мережі Інтернету, що піддає його глобальному вторгненню, а також до бездротових атак усередині мережі IoT. Він захищений криптографічними та мережевими методами безпеки, але вони вразливі до внутрішніх і зовнішніх атак. Пристрої IoT мають обмежені ресурси з точки зору обмеженого зберігання, а також обмеженого діапазону передачі та обробки. Щоб захистити ці пристрої від внутрішніх та зовнішніх кібератак, нам потрібна легка система безпеки, яка може вільно працювати на таких пристроях IoT. Нам також потрібна система, яка може визначити тип вторгнення та викликати тривогу у сценарії вторгнення, щоб вжити відповідних превентивних заходів. Отже, система виявлення вторгнень (IDS) відіграє важливу роль у запобіганні таких кібератак IoT.

Мета роботи полягає в створенні системи запобігання проникненню в мережі інтернет-речей, який забезпечує високу швидкодію та детермінований час виконання.

Для досягнення даної мети ставились наступні завдання:

1) дослідити існуючі системи запобігання проникненню, вказати їх переваги та недоліки;

2) дослідити різні типи систем та виконати постановку завдання;

3) проаналізувати систему IPS на основі мережі та хоста;

4) проаналізувати систему IDS на основі мережі та хоста;

5) розробити вдосконалений алгоритм захисту мережі інтернет-речей.

6) провести тестування розробленого програмного засобу на основі запропонованого алгоритму.

Об’єкт дослідження - процес виявлення аномальної активності в роботі комп’ютерної мережі.

Предмет дослідження – методи побудови систем виявлення проникнень в комп’ютерній мережі**.**

Наукова новизна одержаних результатів визначається наступним чином: розроблено алгоритм, який є більш продуктивний і швидший, ніж інші обчислення, і витрачає менше ресурсів на обробку, що дозволяє швидше реагувати на вторгнення мережі інтернет речей.

Практична цінність одержаних результатів полягає в тому, що:

1) обґрунтуванно доцільность розробки системи виявлення вторгнень;

2) проведено аналіз трафіку для виявлення вторгнень в комп’ютерну мережу;

3) описано принципи аналізу КМ на вторгнення;

4) розроблено системи виявлення вторгнень у комп'ютерній мережі.

Публікації та апробація до кваліфікаційної роботи. За результатами наукових досліджень, проведених у кваліфікаційній роботі, підготовлено тези доповіді на проблемно-наукових міжгалузевих конференціях «Автоматизація та комп’ютерно-інтегровані технології» (АКІТ-2022) та «Кібербезпека та комп’ютерно-інтегровані технології» (КБКІТ – 2022).

1. Миколишин П.П. Система запобігання проникненню в мережі інтернет-речей. Збірник матеріалів проблемної наукової міжгалузевої

конференції «Автоматизація та комп’ютерно-інтегровані технології» (АКІТ-2022). Тернопіль, 2022. С.91-93.

2. Миколишин П.П. Система запобігання проникненню в мережі Інтернет-речей на основі аномалій. Збірник матеріалів проблемної наукової міжгалузевої конференції «Кібербезпека та комп’ютерно-інтегровані технології» (КБКІТ-2022). Тернопіль, 2022. С.63-64.

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Основні поняття та доцільність використання систем виявлення вторгнень

Система запобігання вторгненням (IPS) та система виявлення вторгнення IDS (анг. Intrusion Detection System, Intrusion Prevention System) - це мережеві пристрої, які підвищують безпеку комп'ютерних мереж шляхом виявлення (IDS) або виявлення та блокування атак у реальному часі (IPS) [1]. Основними функціями IDS є моніторинг загроз, запис даних про їхню дію та повідомлення про них. У свою чергу, дії IPS спрямовані на припинення нападу, мінімізувати його вплив або активно реагувати на порушення безпеки. Ці рішення дозволяють значно підвищити рівень безпеки комп'ютера шляхом посилення його контролю у зв'язку з мережами. Ефективна система безпеки на основі IDS / IPS (рисунок 1.1) повинна врахувати специфіку діяльності підприємства, передбачувати різні джерела загроз у комп'ютерній мережі та на цій основі дати правильне рішення аналізу ризиків.

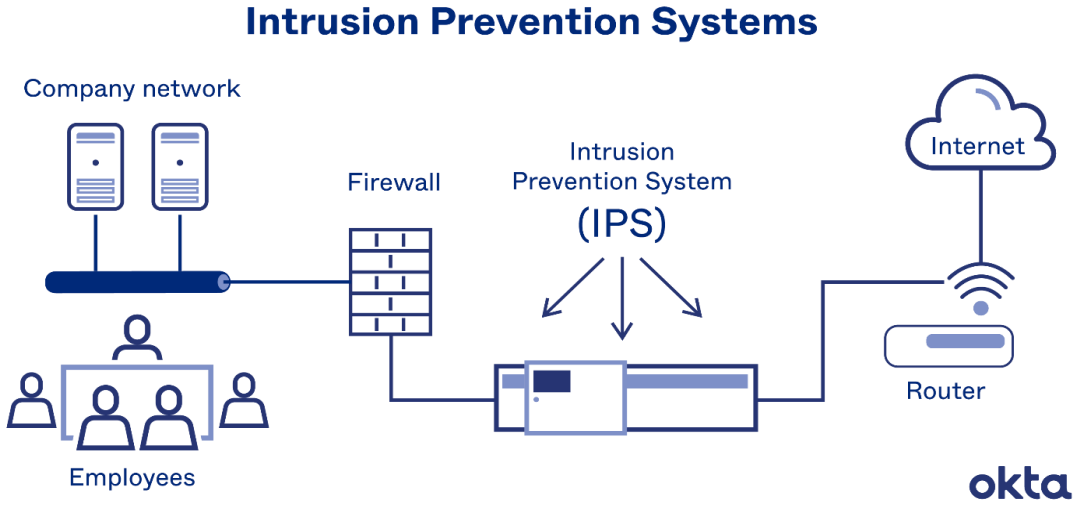


Рисунок 1.1 – Intrusion Prevention System [1]

Система IPS використовує багаторівневі механізми аналізу та безпеки, наприклад протоколи аналітики, виявлення аномалій у мережевому трафіку або

кореляція подій. Це також дозволяє створювати власні правила, засновані на порівнянні шаблонів атак.

Система IDS зазвичай працює як сніффер (комп'ютерна програма, завданням якої є захоплення та аналіз даних, що надходять у мережу) виявляє спробу порушення безпеки та інформує брандмауер про місцезнаходження (IP-адресу) зловмисника. Отже, брандмауер блокує пакети що надходять із зазначеної адреси, які піддаються атаці (рисунок 1.2).

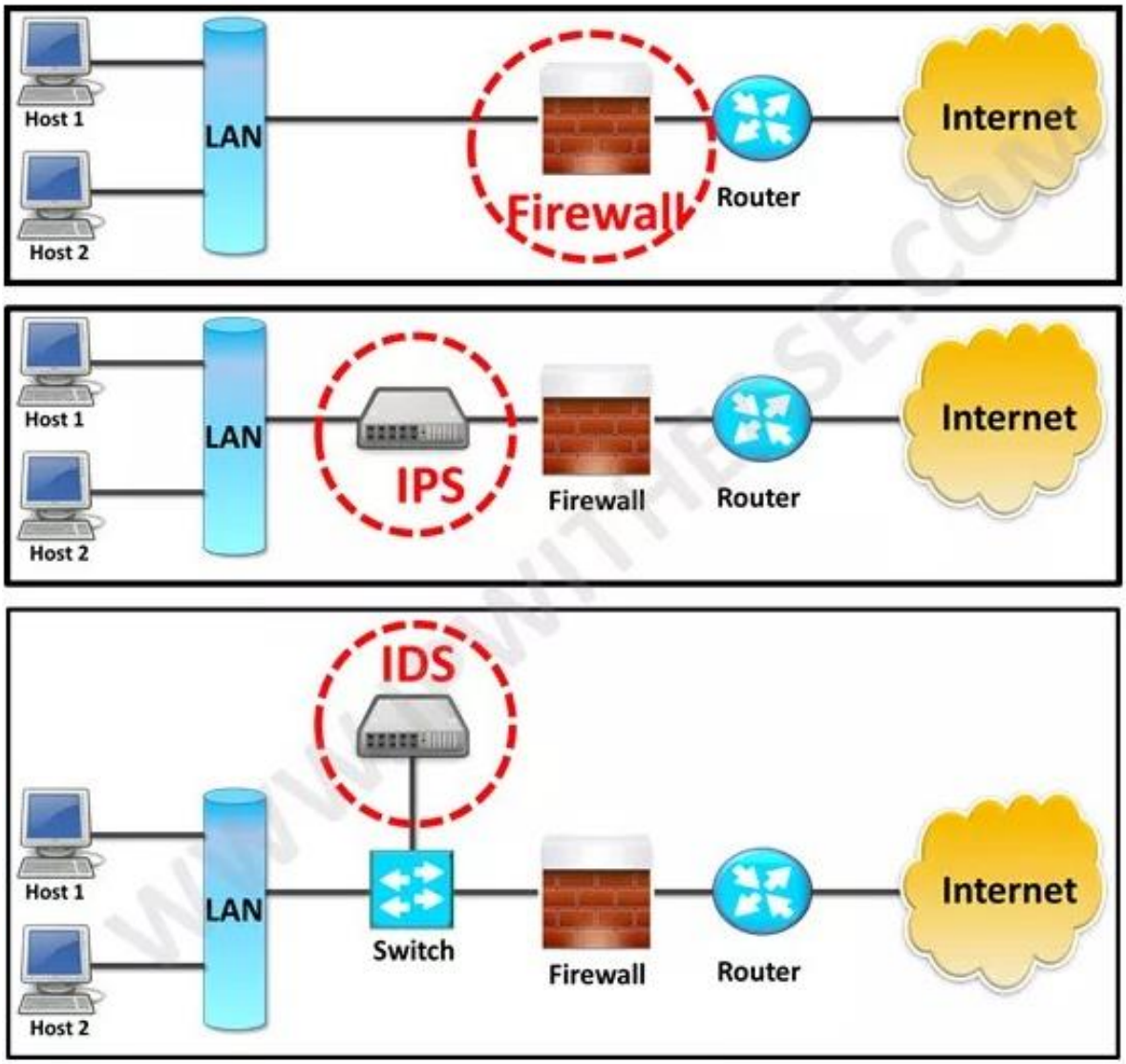


Рисунок 1.2 – Брандмауер у системі захисту мережі [2]

Пристрої Інтернету речей (IoT) швидко розвиваються, але ці пристрої мають обмежену пам'ять, обчислювальну потужність і обчислювальну

потужність, оскільки вони засновані на мікроконтролері нижчого класу. Основна проблема сьогодні полягає в тому, як захистити ці пристрої низького класу. Крім того, важливо полегшити впровадження для OEM-виробників. Близько 360 платформ IoT використовують понад 100 різних протоколів . Ці різновиди представляють кілька загроз, таких як загрози, пов'язані з аномаліями та вторгненнями в мережу. Трафік у мережі відстежується, щоб повідомляти про незвичайні дії, такі як аномальна поведінка, яка спричинила зловмисні атаки, наприклад, віруси, атаки на відмову в обслуговуванні (DoS) і розподілену атаку на відмову в обслуговуванні (DDoS), інші атаки можуть викликати випадкові збої та збої в роботі обладнання. Для забезпечення безпеки мережевої інфраструктури та комунікацій через Інтернет було розроблено кілька підходів і методів.

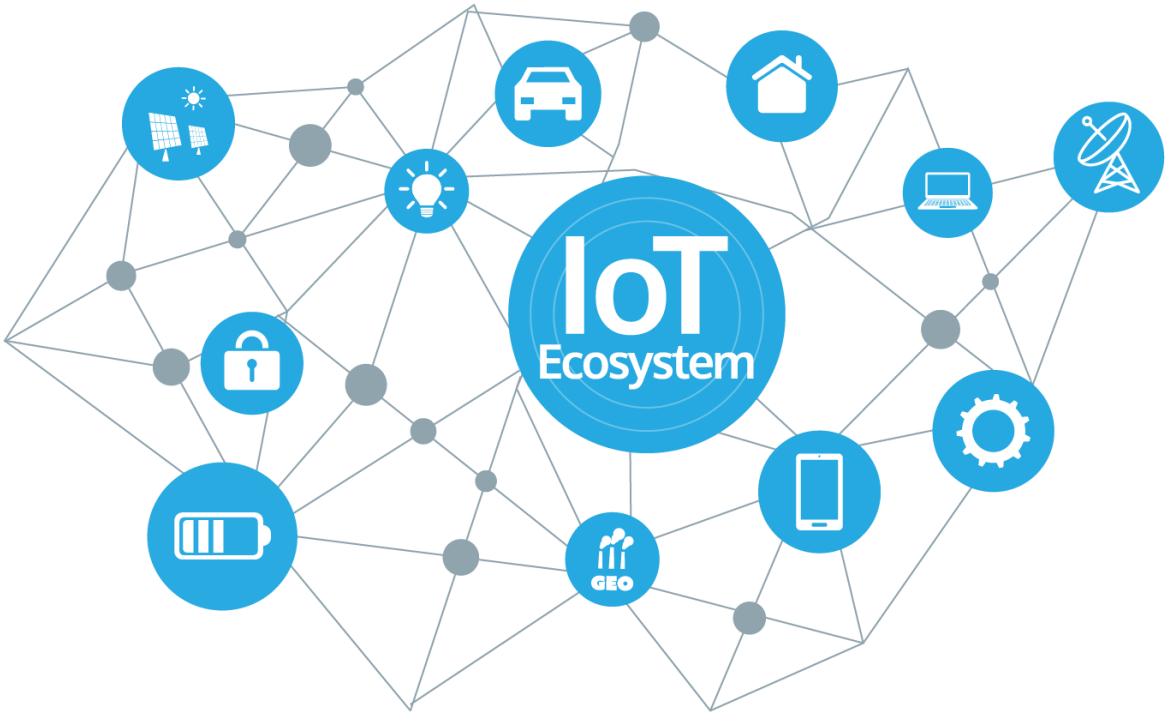


Рисунок 1.3 - Пристрої Інтернету речей [3]

Системи виявлення та запобігання вторгненням, пакети антивірусного програмного забезпечення та брандмауери є прикладами цього методу, і методи широко використовуються для досягнення вимог безпеки. Однак брандмауери

самі по собі не можуть захистити від усіх типів вторгнень і атак, коли вторгнення намагаються порушити безпеку мережі, використовуючи вразливі місця в мережі. Виявлення ненормальної поведінки в мережах, наприклад проникнень, зломів або будь-якої іншої форми підозрілої діяльності, називається виявленням вторгнень.

Система виявлення вторгнень (IDS) несе відповідальність за моніторинг усіх дій у мережі та поведінки користувачів, щоб перевірити наявність будь-яких підозрілих дій або будь-яких порушень зазначеної політики. Крім того, IDS може надати звіт станції керування. Також, IDS розглядається як додаткова стіна, яка забезпечує додатковий захист мережі. Він доступний цілодобово для отримання інформації про стан системи, моніторингу дій користувачів і надання звітів на станцію керування.

Класифікації IDS є мережевими, хостовими та гібридними. Класифікація залежить від джерела та типу інформації для виявлення порушень безпеки. Урядові сектори, приватні сектори, компанії, установи малого бізнесу, сектори охорони здоров'я та навіть окремі користувачі повинні запровадити IDS для ідентифікації атак і запобігання як у системах на основі хоста, так і в системах на основі мережі.

Операція містить набір правил і політик для виявлення будь-якого типу загроз, атак або вторгнень для отримання несанкціонованого доступу до будь-якого джерела даних або перехоплення пакета на шляху до місця призначення. Пристрої IoT, які безпосередньо підключаються до Інтернету, можуть бути піддані ряду загроз і можуть бути легко атаковані. Незважаючи на те, що для захисту такого середовища було застосовано кілька методів, наприклад, безпечна конфігурація, оновлені виправлення та брандмауери, усі вони непрості в обслуговуванні та не можуть гарантувати, що система може бути безпечною для різних типів атак. IDS забезпечує захист, у якому він відстежує мережу чи системи на наявність порушень політики чи зловмисної діяльності. IDS працює як «охоронець», який контролює мережу та забезпечує кращий захист, ніж інші засоби.

Безпека систем інтернет-речей (IoT) є серйозною проблемою через збільшення кількості послуг і користувачів у мережах IoT. Інтеграція систем IoT і розумних середовищ робить розумні об’єкти більш ефективними. Однак наслідки вразливостей безпеки IoT дуже небезпечні в критично важливих розумних середовищах, які використовуються в таких сферах, як медицина та промисловість. У розумних середовищах на основі IoT без надійних систем безпеки програми та служби будуть під загрозою. Конфіденційність, цілісність і доступність є трьома важливими концепціями безпеки додатків і служб у розумних середовищах на основі IoT [2]. Проблеми безпеки у мережі IoT на різних її рівнях показані на рисунку 1.4.

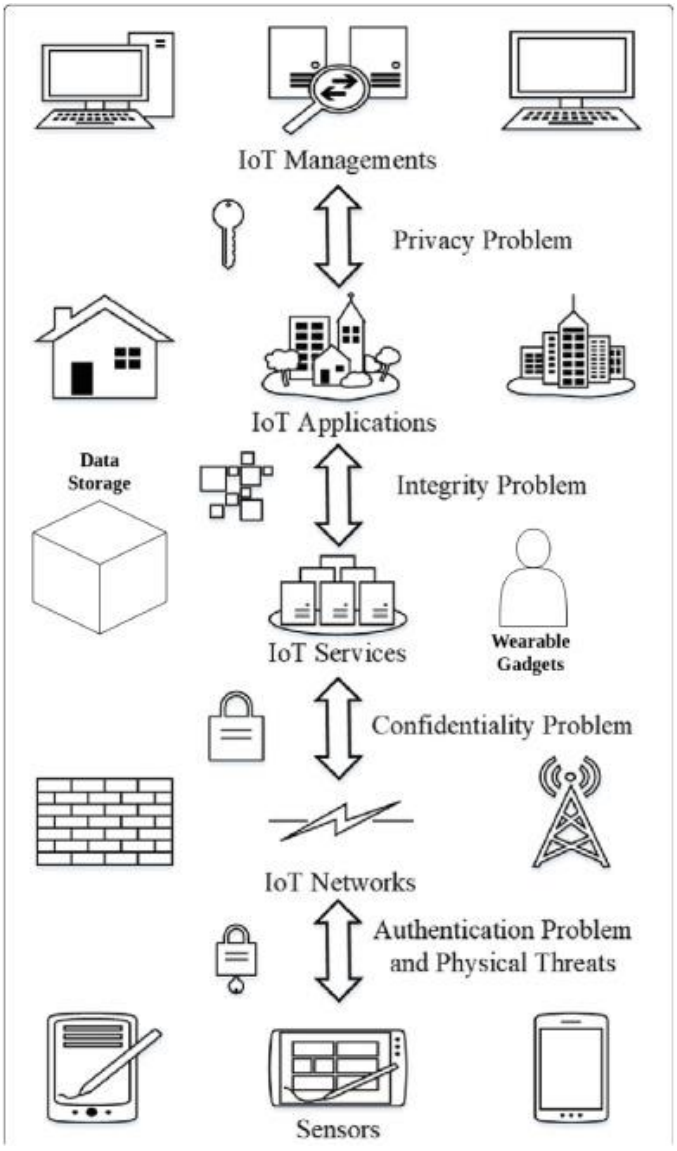


Рисунок 1.4 - Проблеми безпеки на рівнях IoT [4]

Отже, серйозною проблемою безпеки для систем виявлення вторгнень є зіткнення зі шкідливими варіантами програмного забезпечення, які призводять до порушень безпеки мережі та серйозних збоїв. Кібератаки є складнішими та складнішими в ідентифікації атак невідомого шкідливого програмного забезпечення через еволюцію передових методів ухилення від викрадення важливої інформації та ухилення від виявлення IDS. Крім того, існують загрози кібербезпеці під час спілкування в мережі. Таким чином, новітні методи та рішення необхідні для запобігання атакам і своєчасного виявлення вторгнень.

Тому розглянемо детальніше різні види, методи для систем виявлення та запобігання вторгнень в мережу інтернет-речей.

1.2 Аналіз системи виявлення загроз IDS у мережі IoT

IDS — це системи, які автоматично виявляють і аналізують аномальну та лякаючу поведінку на хості чи в мережі для моніторингу безпеки та захисту [3]. Просто виявлення вторгнення виявляє вторгнення. Іноді він визначає інструкції для доказів у деяких ситуаціях. Вторгнення є ексцентричністю мережі або комп'ютера від нормальної поведінки та означає загрозу, яка використовується для атаки з метою викрадення або пошкодження мережевих даних. Зараз люди використовують Інтернет та інші мережі для обміну та зберігання конфіденційних даних. Також IDS — це програма кібербезпеки, яка використовується брандмауером і антивірусним програмним забезпеченням.

Крім того, брандмауер обмежено аналізує онлайн-трафік. Хоча IDS можуть контролювати, контролювати та підтримувати всі мережеві потоки, навіть якщо в мережах відбуваються нестандартні дії або атаки загроз, це викликає попередження для мережевих адміністраторів. На рисунку 1.5 показано мережевий комунікаційний потік разом із системами виявлення вторгнень у мережах.

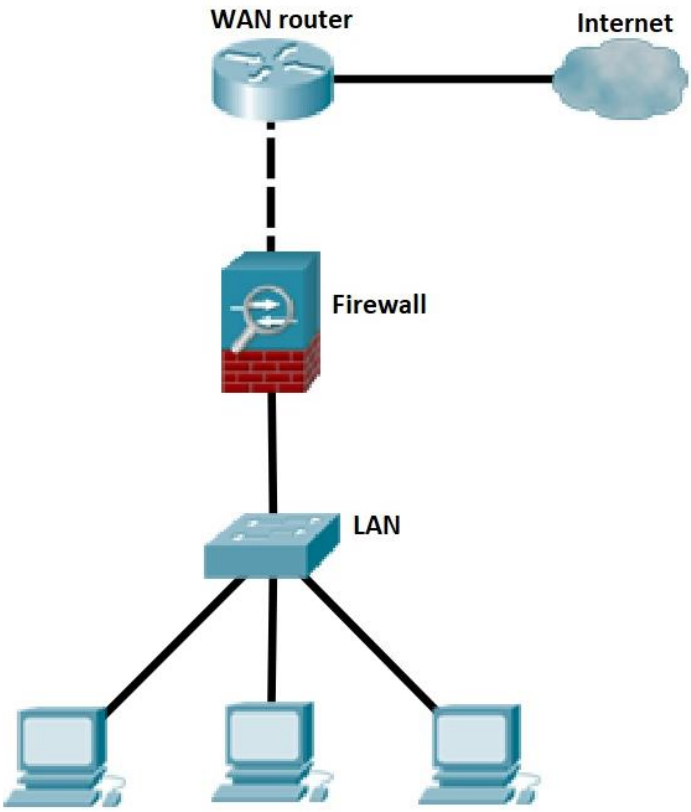


Рисунок 1.5 – Аналіз трафіку брандмауером [5]

Крім того, брандмауер обмежено аналізує онлайн-трафік. Хоча IDS можуть контролювати, контролювати та підтримувати всі мережеві потоки, навіть якщо в мережах відбуваються нестандартні дії або атаки загроз, це викликає попередження для мережевих адміністраторів. На рисунку 1.5 показано мережевий комунікаційний потік разом із системами виявлення вторгнень у мережах.

IDS в основному складаються з трьох сегментів. Спочатку дані про кібератаки збираються з вхідних даних, а потім обробляються для аналізу та виявлення кібератак другого сегмента (рисунок 1.6). Нарешті, у третьому сегменті повідомляється про напади. Методи машинного та глибокого навчання нещодавно використовувалися для прогнозування нормальної та ненормальної поведінки та нових невідомих атак у мережах за допомогою аналізу вхідних даних. Методи IDS можна класифікувати за різними типами, наприклад, системи виявлення вторгнень на основі сигнатур (SIDS), системи виявлення

вторгнень на основі аномалій (AIDS), виявлення на основі специфікацій, гібридне виявлення, IDS на основі хоста (HIDS) , мережевий IDS (NIDS) і розподілений IDS (DIDS) [8].

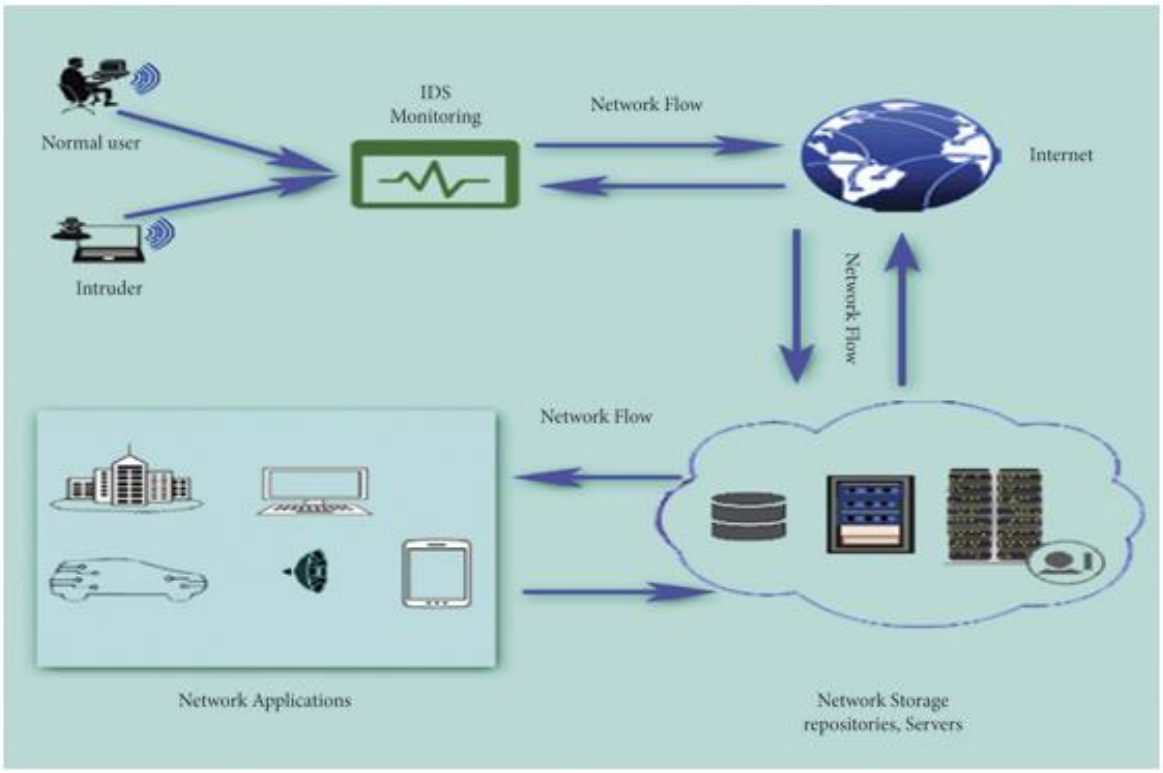


Рисунок 1.6 - Мережевий потік через мережі разом із IDS [6]

Методи виявлення, що використовуються в IDS:

1) виявлення на основі сигнатур (SIDS);

2) виявлення на основі аномалій (AIDS).

Системи виявлення вторгнень на основі сигнатур (SIDS). SIDS також називають виявленням на основі знань. Він аналізує та оцінює мережі на основі відомих шаблонів або відповідних сигнатур для пошуку сигнатур атак у базах даних сигнатур шляхом порівняння мережевого зв'язку та дій. Він зберігає поведінку та сигнатуру кожної атаки в мережі. Попередження створюється, коли сигнатура атаки знайдена або збігається зі збереженою базою даних сигнатур. Це означає, що SIDS виявляє лише атаки, сигнатура яких зберігається в базі даних. Нові атаки виявляються за допомогою SIDS, хоча він не такий точний у протиріччі варіацій атак. Система сповіщень зводить до мінімуму

помилкові сповіщення завдяки ефективній і точній ідентифікації та класифікації неправильної поведінки для оцінки захисних дій адміністраторів мережі. Проте дії, які не відповідають базі даних, вважаються невідомими порушеннями, нормальними або варіантами атаки. Тому SIDS потребує постійного оновлення бази даних для нових варіантів атак. Звичайні методи SIDS аналізують пакети лише шляхом порівняння з шаблонами в базі даних. Він не розпізнає нові варіанти атак.

Метод AIS (штучна імунна система) використовується для боротьби з обмеженнями SIDS [25]. У цій техніці використовується модель імунних клітин, яка працює на основі моделі нападів або сигнатур і оцінює їх за класифікацією на нормальні чи аномальні. Він також виявляє нові сигнатури шляхом постійного моніторингу системи. Крім того, IDS підпису Suricata на основі Linux використовується для вирішення проблеми обмеження ресурсів.

Система виявлення вторгнень на основі аномалій (AIDS). AIDS, також відомий як виявлення аномалій поведінки на основі профілю або динамічного виявлення, є найбільш широко використовуваною моделлю порівняно з SIDS завдяки своїй ефективності проти нових атак. AIDS зазвичай використовується для подолання обмежень SIDS. Виявлення неідентифікованих атак на різних етапах викликає попередження, щоб розпізнавати викриття та запобігати їм за допомогою можливих методів. AIDS постійно стежить за системою, щоб зібрати дані для виявлення та розпізнавання нормального чи ненормального. Розпізнавання атак нульового дня є основною метою AIDS, оскільки нові аномальні дії стосуються баз даних шаблонів. Він може навчитися аномальній поведінці в мережах. Наприклад, якщо відбувається будь-яка несанкціонована діяльність або викрадення облікового запису, генерується тривога. Аномальна поведінка — це нові звичайні дії, а не втручання, що не зазнали впливу, що призводить до високого рівня хибно-позитивних результатів.

Кілька методів нещодавно розроблено та представлено в літературі для виявлення та класифікації аномальної поведінки. Його вивчали протягом останніх двох десятиліть, але проблеми все ще не могли бути вирішені.

На рисунку 1.7 наведено механізм виявлення вторгнень у мережі IoT, який складається з таких етапів:

1) зловмисник надсилає зловмисний трафік через Інтернет на цільовий хост;

2) пакети даних надходять потрапляють як до мережі, так і до IDS;

3) у IDS пакет перевірятиметься датчиком;

4) зберігання звіту у журнал на консолі керування.

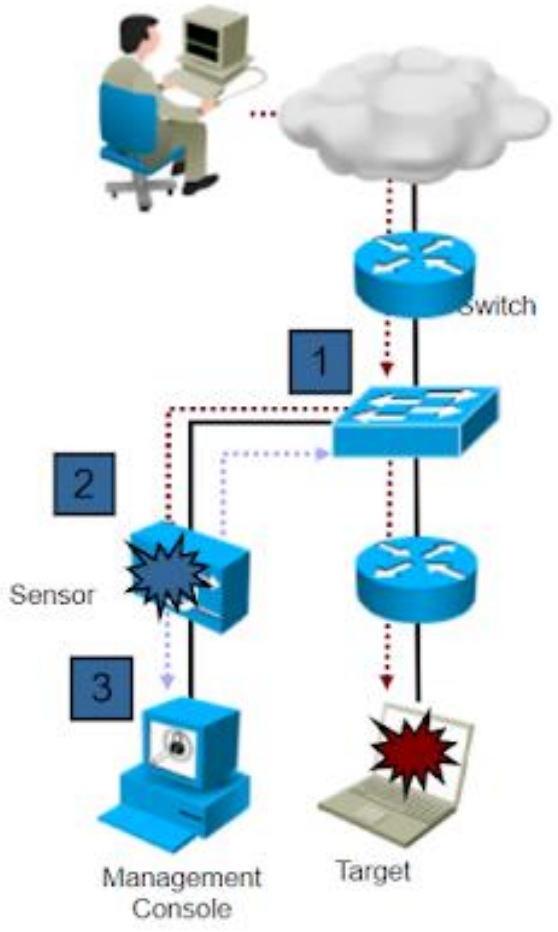


Рисунок 1.7 - Система виявлення вторгнень [7]

Типи тривог, що генеруються системою IDS:

1) помилкові тривоги:

а) помилкове спрацьовування: нормальний, звичайний мережевий трафік запускає дію, пов'язану з підписом;

б) помилкова false: незаконний мережевий трафік немає запускати дію, пов'язану з підписом, атака не виявлена;

2) справжні тривоги:

а) істинне try: нелегальний мережевий трафік викликає дію, пов'язану з підписом, проведену атаку виявлено;

б) справжній false: нормальний, нормальний мережевий трафік не запускає дію підпису, звичайний рух не викликає тривоги.

Система IDS може бути представлена в двох варіантах:

1) IDS на основі хоста (HIDS);

2) на основі мережі IDS (NIDS).

IDS на основі хоста (HIDS) — це програмне забезпечення, встановлене на головному комп’ютері мережі, яке досліджує, аналізує, збирає та контролює дії даних у мережі та хост-мережі, перевіряючи журнали брандмауера, сервера або бази даних [26]. HIDS обмежується виявленням аномальних атак одного хоста, одночасно виявляючи незадіяні атаки в мережі [7].

Операційна структура HIDS та її розташування в мережі показані на рисунку 1.8.

Мережевий IDS (NIDS) відстежує мережеві зв’язки, збираючи перехоплені пакети та інші дані через NetFlow [27]. Його основна мета — захистити мережі від зовнішніх атак, викликаючи попередження/тривогу, коли відбувається зловмисна атака. Цей IDS працює з кількома хостами в мережах і зовнішніми брандмауерами, відстежуючи та аналізуючи мережеві комунікації за допомогою програмного або апаратного забезпечення. Програмне забезпечення встановлюється на серверах для моніторингу, тоді як датчики підключаються до серверів для аналізу зв’язку в мережі. Як результат, NIDS є дуже ефективним і безпечним у виявленні зловмисних атак.

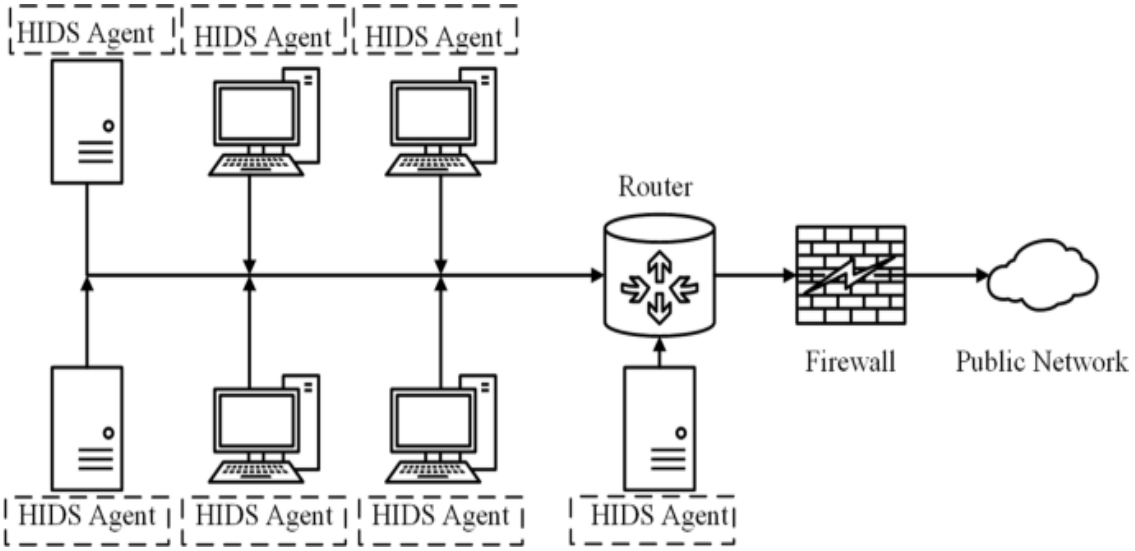


Рисунок 1.8 - Загальна архітектура IDS на основі хоста (HIDS). Операційна структура HIDS та її розташування в мережі [8]

NIDS має кілька обмежень; він не може обробляти та аналізувати величезні дані мережі через високу пропускну здатність і швидкість потоку трафіку. NIDS також не підтримує зашифровані мережеві пакети [28].

NIDS може бути програмною або апаратною системою. Наприклад, Snort NIDS є програмним NIDS. Операційна структура NIDS та її розташування в мережі показані на рисунку 1.9.

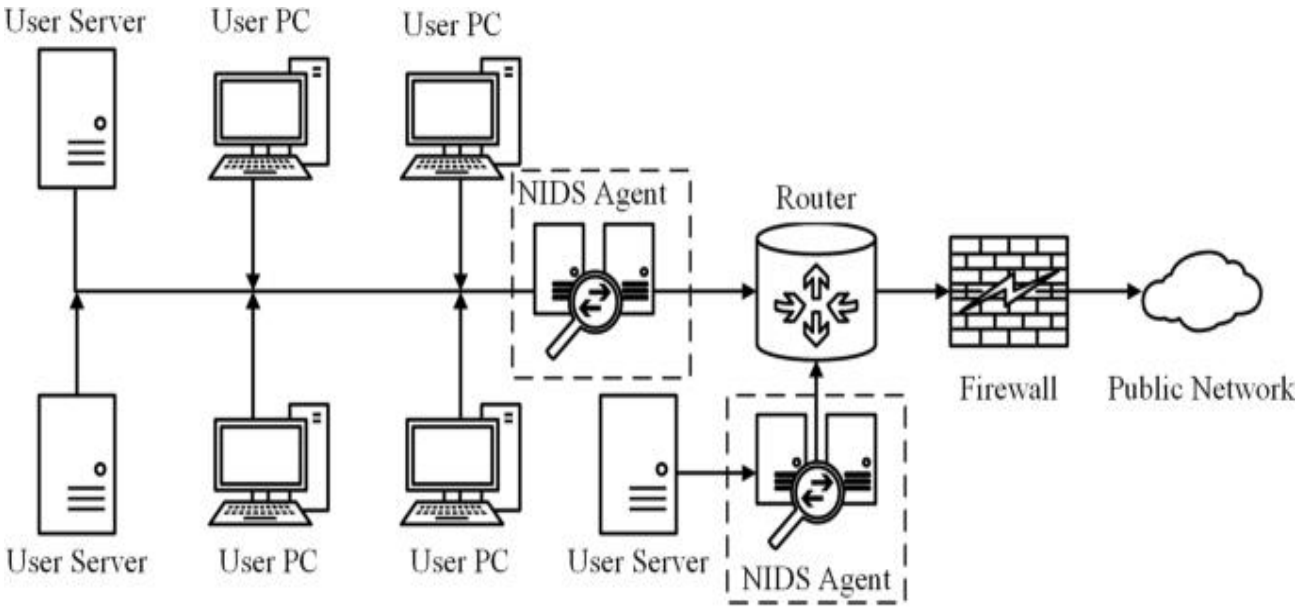


Рисунок 1.9 - Загальна архітектура мережевої IDS (NIDS). Операційна структура NIDS та її розташування в мережі [9]

Розширення мережі та збільшення обсягів трафіку вимагають впровадження IDS як апаратних систем, таких як архітектура інтелектуального датчика. Наприклад, програмовані матриці (FPGA) можна використовувати як основу апаратної NIDS. Особливі характеристики FPGA, такі як їхня здатність підтримувати високошвидкісні інтерфейси, динамічне перепрограмування та обробка дуже великого обсягу даних, роблять FPGA дуже придатними для використання в NIDS.

Snort описується як легкий міжплатформний інструмент виявлення вторгнень у мережу з відкритим кодом. Вважається одним з найпопулярніших IDS. Snort — це модель виявлення на основі сигнатур, яка розроблена для спостереження та моніторингу пакетів мережевого трафіку та виявлення будь-якої підозрілої активності, вторгнення чи загроз у пакетах за допомогою попередньо визначених правил виявлення [29]. Він зберігає базу даних попередньо визначених правил і політик, які використовуються для опису різних типів атак, сигнатур і шаблонів цих атак. Крім того, базу даних можна оновлювати, додаючи нові правила для виявлення нових виявлень будь-якої аномальної поведінки або шаблону атак. Snort має можливість аналізувати вміст мережевого пакета, щоб виявити будь-які можливі загрози або атаки.

1.3 Аналіз системи запобігання вторгненням IPS у мережі IoT

Система запобігання вторгненням (IPS) (рисунок 1.10) — іноді її називають системою запобігання виявлення вторгнень (IDPS) — це технологія безпеки мережі та ключова частина будь-якої корпоративної системи безпеки, яка постійно відстежує мережевий трафік на наявність підозрілої активності та вживає заходів для її запобігання [30]. Багато в чому автоматизовані рішення IPS допомагають відфільтрувати цю зловмисну активність до того, як вона досягне інших пристроїв безпеки або елементів керування, ефективно

зменшуючи ручні зусилля груп безпеки та дозволяючи іншим продуктам безпеки працювати ефективніше. Вона є більш досконалою, ніж система виявлення вторгнень (IDS), яка просто виявляє зловмисну активність, але не може вжити заходів проти неї, окрім сповіщення адміністратора.

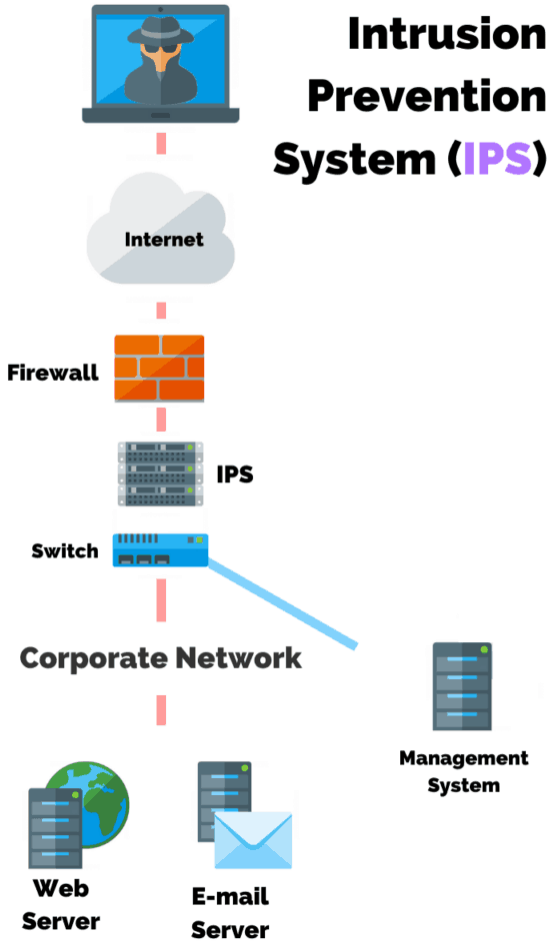


Рисунок 1.10 - Система запобігання вторгненням (IPS) [10]

IPS включають брандмауери, антивірусне програмне забезпечення та програмне забезпечення для захисту від спуфінгу. Крім того, організації використовуватимуть IPS для інших цілей, таких як виявлення проблем із політикою безпеки, документування існуючих загроз і стримування окремих осіб від порушення політики безпеки. IPS стали важливим компонентом усіх основних інфраструктур безпеки в сучасних організаціях.

Система запобігання вторгненням працює шляхом активного сканування пересиланого мережевого трафіку на наявність зловмисних дій і відомих

моделей атак [31]. Механізм IPS аналізує мережевий трафік і постійно порівнює бітовий потік зі своєю внутрішньою базою даних сигнатур на предмет відомих шаблонів атак. IPS може скинути пакет, визначений як зловмисний, і в подальшому заблокувати весь майбутній трафік з IP-адреси або порту зловмисника. Законний трафік може тривати без видимих перебоїв у роботі.

Системи запобігання вторгненням також можуть виконувати більш складне спостереження та аналіз, наприклад спостерігати та реагувати на підозрілі шаблони трафіку або пакети. Механізми виявлення можуть включати:

1) збіг адрес;

2) зіставлення рядків і підрядків HTTP;

зіставлення загального шаблону;

3) аналіз TCP з'єднання;

4) виявлення аномалії пакетів;

5) виявлення аномалій дорожнього руху;

6) відповідність порту TCP/UDP.

IPS зазвичай записує інформацію, пов’язану з спостережуваними подіями, повідомляє адміністраторів безпеки та створює звіти. Щоб допомогти захистити мережу, IPS може автоматично отримувати оновлення для запобігання та безпеки, щоб постійно відстежувати та блокувати нові загрози в Інтернеті.

Заходи протидії вторгненню. Багато IPS також можуть реагувати на виявлену загрозу, активно запобігаючи її успіху. Вони використовують кілька методів реагування, які включають:

1) зміна середовища безпеки – наприклад, шляхом налаштування брандмауера для посилення захисту від раніше невідомих уразливостей;

2) зміна вмісту атаки – наприклад, шляхом заміни шкідливих частин електронного листа, як-от помилкових посилань, попередженнями про видалений вміст;

3) надсилання системним адміністраторам автоматичних сигналів сповіщення про можливі порушення безпеки;

4) скидання виявлених шкідливих пакетів;

5) скидання підключення;

6) блокування трафіку з неправильної IP-адреси.

Системи запобігання вторгненням IPS згідно класифікації можна розділити на чотири основні типи [32]:

1) мережева система запобігання вторгненням (NIPS): аналізує активність протоколу в усій мережі, шукаючи будь-який ненадійний трафік (рисунок 1.11);

2) система запобігання бездротовому вторгненню (WIPS): аналізує активність мережевого протоколу в усій бездротовій мережі, шукаючи ненадійний трафік;

3) система запобігання вторгнень на основі хоста (HIPS): вторинний пакет програмного забезпечення, який слідкує за одним хостом для виявлення зловмисної активності та аналізує події, що відбуваються на цьому хості;

4) аналіз поведінки мережі (NBA): аналізує мережевий трафік для виявлення загроз, які створюють дивні потоки трафіку. Найпоширенішими загрозами є атаки на відмову в обслуговуванні, різні форми зловмисного програмного забезпечення та порушення політики. зіставлення шаблонів для виявлення атак. Виявлення можна уникнути, зробивши незначне коригування архітектури атаки.

Класифікації IPS показана на рисунку 1.12.

Більшість систем запобігання вторгненням використовують один із трьох методів виявлення: на основі сигнатур, на основі статистичних аномалій і аналізу протоколу зі збереженням стану.

Виявлення на основі сигнатур: IDS на основі сигнатур відстежує пакети в мережі та порівнює їх із заздалегідь визначеними шаблонами атак, відомими як «сигнатури».

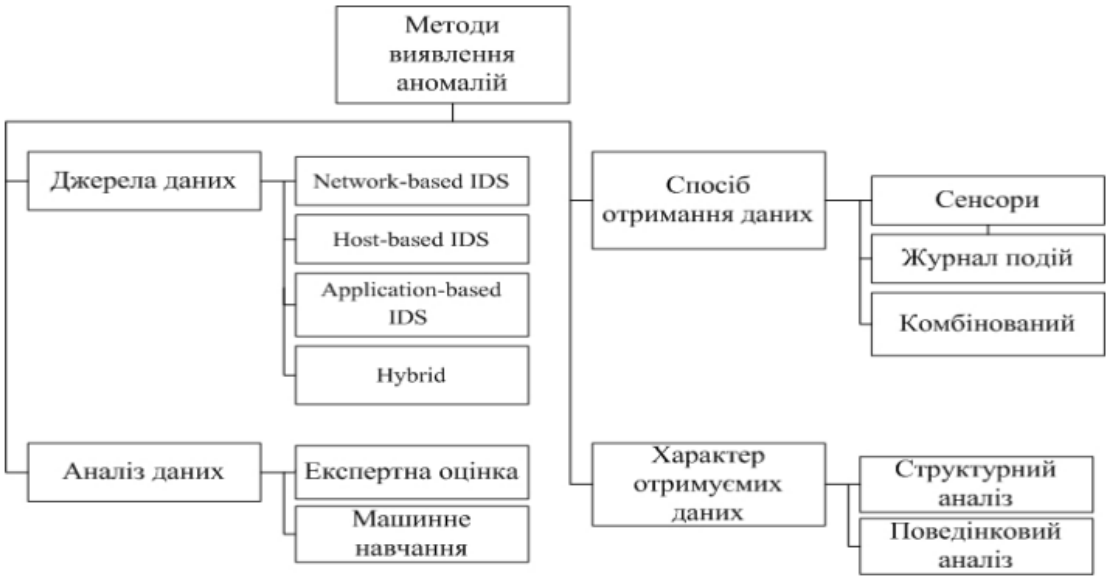


Рисунок 1.11 - Класифікації IPS [11]

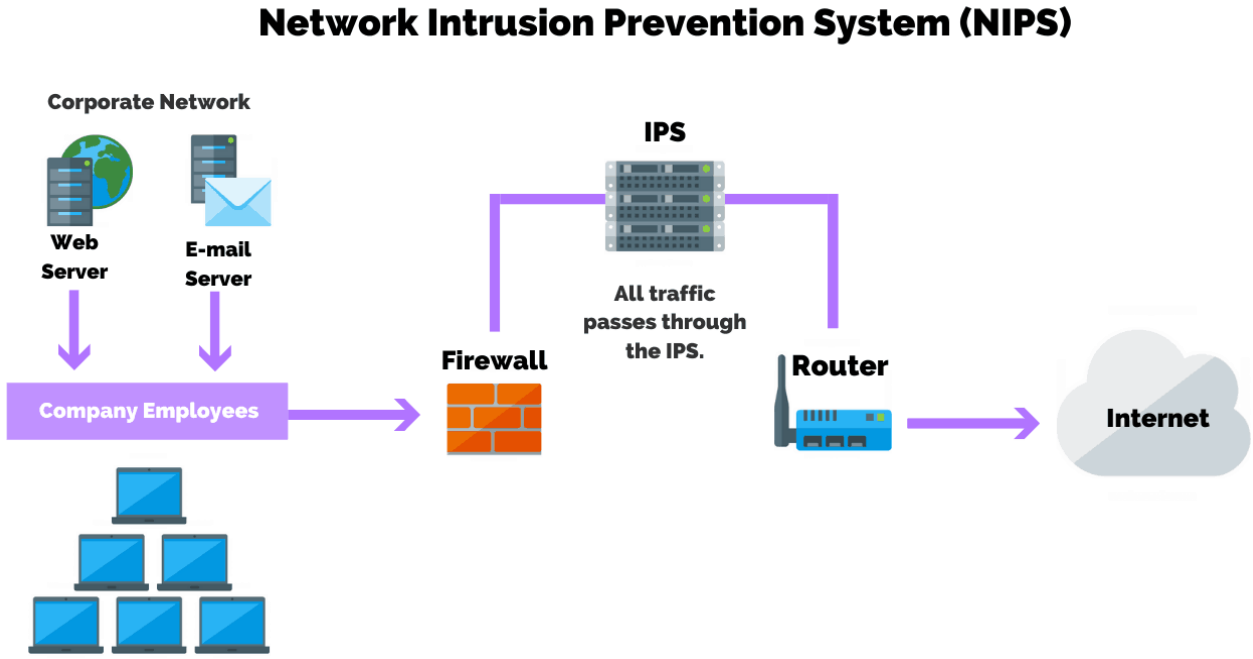


Рисунок 1.12 - Мережева система запобігання вторгненням (NIPS) [12]

Статистичне виявлення аномалій: IDS на основі аномалій відстежуватиме мережевий трафік і порівнюватиме його з очікуваними моделями трафіку. Базовий рівень визначає, що є «нормальним» для цієї мережі – який тип пакетів зазвичай передається через мережу та які протоколи використовуються. Однак це може викликати хибний позитивний сигнал

тривоги для законного використання пропускної здатності, якщо базові лінії не налаштовані інтелектуально.

Виявлення аналізу протоколу за станом: цей метод визначає відхилення протоколу шляхом порівняння спостережуваних подій із попередньо визначеними профілями нормальної активності.

На рисунку 1.13 показано топологію системи IPS у мережі IoT.

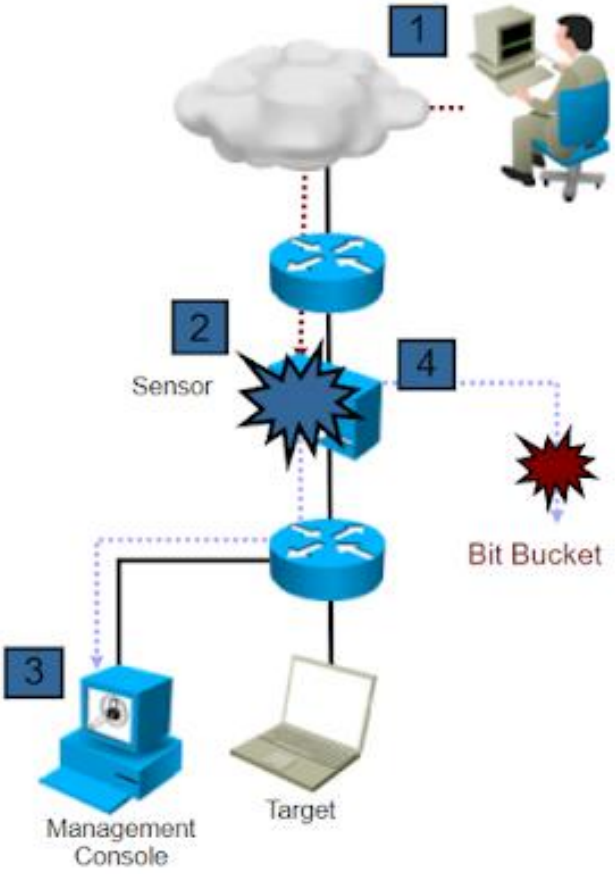


Рисунок 1.13 - Топологія IPS [13]

Послідовність дій зламу IPS:

1) зловмисник надсилає зловмисний трафік через Інтернет на цільовий хост;

2) пакет даних досягає IPS який далі перевіряється датчиком;

3) збереження звіт журналу на консолі керування та запис про дію;

4) надсилання шкідливого пакету даних у бітове відро та видалення його.

Система запобігання вторгненням пропонує багато переваг:

1) додаткова безпека: IPS працює в парі з іншими рішеннями безпеки та може ідентифікувати загрози, які не можуть ці інші рішення. Особливо це стосується систем, які використовують виявлення аномалій. Він також забезпечує чудову безпеку програм завдяки високому рівню обізнаності про програми;

2) підвищена ефективність інших елементів керування безпекою: оскільки IPS відфільтровує зловмисний трафік до того, як він досягне інших пристроїв безпеки та елементів керування, це зменшує навантаження на ці засоби керування та дозволяє їм працювати ефективніше;

3) економія часу: оскільки IPS значною мірою автоматизована, вона потребує менше часу від IT-команд;

4) відповідність: IPS відповідає багатьом вимогам відповідності, встановленим PCI DSS, HIPAA та іншими. Він також надає цінні дані аудиту;

5) налаштування: IPS можна налаштувати з налаштованими політиками безпеки, щоб забезпечити контроль безпеки, специфічний для підприємства, яке його використовує.

1.4 Перспективи побудови системи захисту в мережі IoT. Постановка завдання

Системи виявлення та запобігання мережевим вторгненням NIDS / NIIPS мають перевагу перед іншими системами у тому, що вони не створюють навантаження на контрольовані хости. Вибір систем NIDS / NIPS підтверджується низькою кількістю помилкових спрацьовувань.

Новітні системи NIDS також дозволяють розпізнавати керовану операційну систему на основі характеристик певного стеку TCP/IP. Ця функція дозволяє вводити ієрархічні дані про важливості окремих тривог.

Впроваджена система NIDS / NIPS вимагає постійної адаптації до мінливих загроз і характеру мережевого трафіку. NIDS / NIPS може бути безпечним рішенням для захисту інформації для робочих зон і серверів.

Цілями завдання є розробити IPS, який може працювати на хост-машині та допомагати запобігати атакам мережевого вторгнення на хост-машину. Тобто, розробка полегшеного програмного забезпечення для запобігання вторгненням для системи на базі ОС Ubuntu Linux із консоллю керування, запровадження можливостей моніторингу мережі в програмне забезпечення, розробка статистичних функцій і функцій безпеки на основі сигнатур, які дозволяють раннє виявлення мережевих атак , реалізовувати механізми реагування на мережеві атаки та надавати користувачам можливість розробляти власні правила мережевої безпеки та впроваджувати їх за допомогою програмного забезпечення IPS.

Зважаючи на це, у кваліфікаційній роботі слід виконати такі завдання:

1) провести аналіз існуючих систем захисту мережі;

2) охарактеризувати методи виявлення проникнень;

3) розробка статистичних функцій і функцій безпеки на основі сигнатур;

4) розробити хост-систему IPS для захисту мережі інтернет-речей;

5) провести експериментальні дослідження ефективності розробленої системи захисту мережі.

Отже, у даному розділі:

1) наведено аналіз систем виявлення та запобігання вторгнення в мережі IoT;

2) розглянуто можливі проблеми у безпеці IoT;

3) показана загальна архітектура мережевої IDS;

4) описано алгоритм виявлення системою вторгнень;

5) наведено огляд основних методів виявлення, що використовуються в IDS;

6) цілі та завдання кваліфікаційної роботи.

2. СИСТЕМА ЗАПОБІГАННЯ ПРОНИКНЕННЮ В МЕРЕЖІ ІНТЕРНЕТ-РЕЧЕЙ

2.1 Аналіз побудови системи

З огляду на попередній аналіз систем виявлення і запобіганню вторгнень у мережі IoT необхідно побудувати систему на основі IPS. При багаторівневому підході на екранах IPS буде надаватися один хост. Цей підхід включає три рівні: аналізатор записів, аналізатор активності інфраструктури та аналізатор асоціацій. Перевага цієї стратегії полягає в тому, що вона дозволяє розпізнавати та уникати як на основі міток, так і на основі аномалій. Однак для захисту інформації про структуру та системну активність потрібен великий обсяг пам'яті.

На даний час найкращими системами протидії перериванню є Snort та Sourcefire. Створення повнофункціонального середовища Snort, яке відображає реальну виробничу реалізацію IDS, передбачає встановлення та налаштування кількох окремих інструментів, як показано на логічній схемі нижче (рисунок 2.1). У Snort існує велика кількість доступних препроцесорів і правил різних типів, які можуть бути корисними в різних середовищах залежно від того, що в цих середовищах працює, які інформаційні активи потребують захисту, а також від поведінки користувачів або бізнес-процесів, які відбуваються. Отримання та аналіз мережевого трафіку в Snort часто займає центральне місце. Другою основною функцією є обробка сповіщень та інших типів вихідних даних, створених IDS.syslog , запис виводу журналу на екран чи консоль моніторингу або генерування виводу в спеціальному форматі unified2 Snort і обробка його за допомогою Barnyard . Для наших цілей, головне завдання полягає в тому, щоб отримати дані попередження в базу даних для подальшої перевірки та аналізу, тому потрібно змусити Snort виробляти вихідні дані unified2 і використовувати Barnyard2 для завантаження цього виводу в базу даних MySQL.

Основною перевагою використання Barnyard замість прямого ведення журналу бази даних є його швидкість – буферизація виводу unified2 у файл для

обробки Barnyard потребує менше ресурсів процесора, ніж підтримка активного з'єднання з базою даних і вставлення записів подій у базу даних – тому у середовищах, де обробка IDS ємність є пріоритетом, гарною ідеєю буде перекласти обробку виводу на такий інструмент, як Barnyard. Після того, як вихідні дані Snort з'являться в базі даних, наступним кроком потрібно зробити дані попереджень доступними для візуалізації та аналізу, у цьому випадку потрібно використати BASE (програма, в основному написана на PHP), щоб надати зведену та детальну інформацію про сповіщення, створені Snort або будь-яким іншим. інший інструмент, який може мати можливість і дозволи для запису в таблиці в базі даних MySQL.

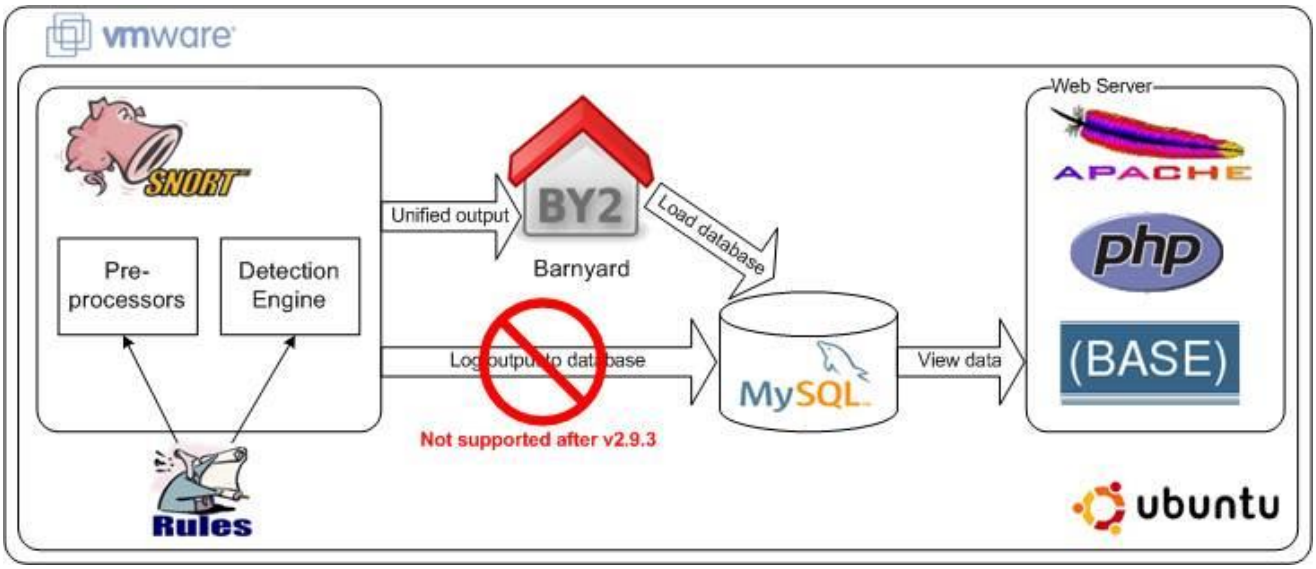


Рисунок 2.1 – Середовище Snort [14]

Snort — це потужна система виявлення вторгнень (IDS) і система запобігання вторгненням (IPS) з відкритим вихідним кодом , яка забезпечує аналіз мережевого трафіку в реальному часі та реєстрацію пакетів даних. SNORT використовує мову на основі правил, яка поєднує методи перевірки аномалій, протоколів і сигнатур для виявлення потенційно зловмисної діяльності.

Використовуючи Snort, мережеві адміністратори можуть виявляти атаки відмови в обслуговуванні (DoS) і розподілені атаки DoS (DDoS), атаки

на загальний інтерфейс шлюзу (CGI), переповнення буфера та приховане сканування портів . Snort створює низку правил, які визначають зловмисну мережеву активність, ідентифікують шкідливі пакети та надсилають сповіщення користувачам.

Механізм виявлення Snort зображений на рисунку 2.2.

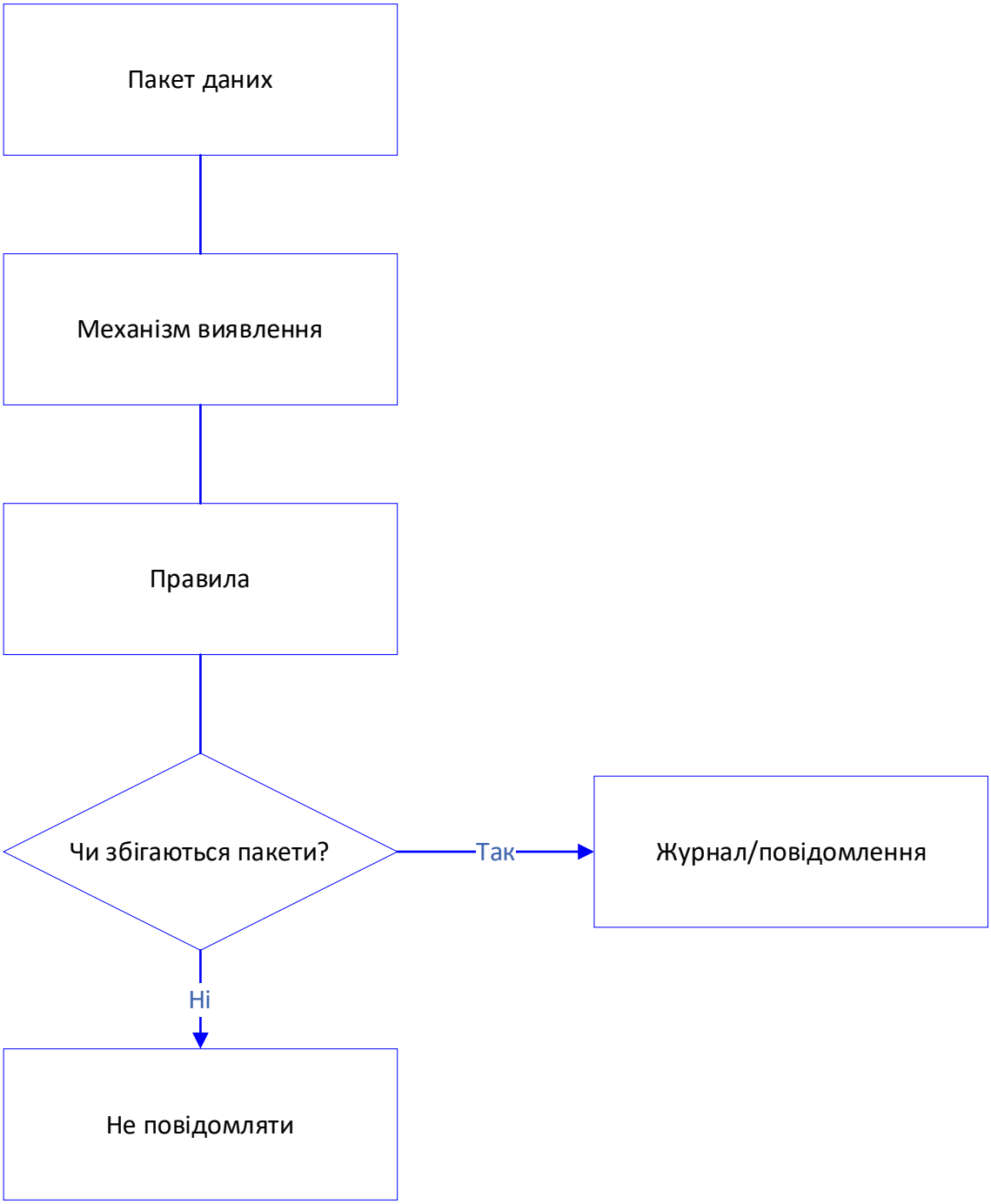


Рисунок 2.2 - Механізм виявлення Snort

Особливості Snort. Існують різні функції, які роблять Snort корисним для мережевих адміністраторів для моніторингу їхніх систем і виявлення шкідливих дій. До них належать:

1) монітор руху в реальному часі. Snort можна використовувати для моніторингу трафіку, який надходить і виходить з мережі. Він відстежуватиме трафік у режимі реального часу та видаватиме сповіщення користувачам, коли виявлятиме потенційно шкідливі пакети чи загрози в мережах Інтернет-протоколу (IP);

2) протоколювання пакетів. Snort дозволяє реєструвати пакети в режимі реєстратора пакетів, що означає, що він записує пакети на диск. У цьому режимі Snort збирає кожен пакет і записує його в ієрархічний каталог на основі IP-адреси хост-мережі;

3) аналіз протоколу. Snort може виконувати аналіз протоколу, який є процесом мережевого аналізу, який фіксує дані на рівнях протоколу для додаткового аналізу. Це дає змогу адміністратору мережі додатково досліджувати потенційно зловмисні пакети даних, що має вирішальне значення, наприклад, у специфікації протоколу стека протоколу керування передачею/IP (TCP/IP);

4) відповідність вмісту. Snort порівнює правила за протоколом, таким як IP і TCP, потім за портами, а потім за тими з вмістом і без. Правила, у яких є вміст, використовують збіг кількох шаблонів, що підвищує продуктивність, особливо коли мова йде про протоколи, як-от протокол передачі гіпертексту (HTTP). Правила, які не мають змісту, завжди оцінюються, що негативно впливає на продуктивність;

5) відбитки ОС. Відбитки операційної системи (ОС) використовують концепцію, згідно з якою всі платформи мають унікальний стек TCP/IP. За допомогою цього процесу Snort можна використовувати для визначення платформи ОС, яка використовується системою, яка звертається до мережі;

6) можна встановити в будь-якому мережевому середовищі. Snort можна розгорнути на всіх операційних системах, включаючи Linux і Windows, і як частину всіх мережевих середовищ;

7) відкрите джерело. Як частина програмного забезпечення з відкритим вихідним кодом, Snort є безкоштовним і доступним для всіх, хто хоче використовувати IDS або IPS для моніторингу та захисту своєї мережі;

8) правила легко запровадити. Правила Snort легко впроваджувати та забезпечувати роботу моніторингу та захисту мережі. Його мова правил також є дуже гнучкою, а створювати нові правила досить просто, що дозволяє мережевим адміністраторам відрізняти звичайну активність в Інтернеті від аномальної чи зловмисної активності.

Використання правил в Snort. Правила, визначені в Snort, дозволяють програмному забезпеченню виконувати ряд дій, які включають:

1) виконувати перевірку пакетів. Snort можна використовувати для аналізу пакетів, який збирає всі дані, що передаються в мережу та з неї. Збір окремих пакетів, які надходять до та від пристроїв у мережі, дозволяє детально перевірити, як передається трафік;

2) налагодження мережевого трафіку. Після реєстрації трафіку Snort можна використовувати для налагодження шкідливих пакетів і будь-яких проблем конфігурації;

3) створення сповіщень. Snort генерує сповіщення для користувачів, як визначено в діях правил, створених у файлі конфігурації. Щоб отримувати сповіщення, правила Snort мають містити умови, які визначають, коли пакет слід вважати незвичайним або зловмисним, ризики використання вразливостей і можуть порушувати політику безпеки організації або становити загрозу для мережі;

4) створення нових правил. Snort дозволяє користувачам легко створювати нові правила в програмному забезпеченні. Це дозволяє мережевим адміністраторам змінювати те, як вони хочуть, щоб перетворення Snort працювало для них, і процеси, які воно має виконувати. Наприклад, вони

можуть створювати нові правила, які вказують Snort запобігати бекдор-атакам, шукати певний вміст у пакетах, показувати дані мережі, вказувати, яку мережу контролювати, і друкувати сповіщення на консолі;

5) розрізняти звичайну та зловмисну діяльність в Інтернеті.

Використання правил Snort дозволяє мережевим адміністраторам легко відрізнити звичайну, очікувану активність в Інтернеті від усього, що виходить за межі норми. Snort аналізує мережеву активність у реальному часі, щоб виявити шкідливу активність, а потім генерує сповіщення для користувачів.

Основним недоліком Snort є те, що він використовує сигнатурну систему для розпізнавання переривання. Якщо відбуваються якісь дивні зміни, Snort не зможе розпізнати цей напад. Мобільний агент залежить від пропонованого IDPS. Захищений портативний оператор є надійним в спостереженні за системою, обробці журналів, виявленні атак та забезпечення безпеки хоста за допомогою постійної комп'ютерної реакції. Справжнім недоліком цього є те, що якщо метою зловмисників є мобільний агент, то для будь-якого IDPS захист зламаного фреймворку стає монотонним. Отже, для захисту портативного спеціаліста потрібна інша структура безпеки. Девід і Паола запропонували метод, який демонструє зв'язок використання з робочою структурою і розглядає як можна зламати IDS нерозпізнано, за допомогою методу зіставлення послідовностей. Однак в цій стратегії немає інформації, необхідної для проведення такого нападу. Також нічого не відомо про те, чого очікують нападники в разі спрацювання IDS.

IPS – це інновація в галузі системної безпеки/запобігання ризикам, яка перевіряє потоки мережної активності для виявлення та протидії неправомірному використанню. У більшості випадків зловживання безсиллям відбувається як згубний внесок у цільову програму або адміністрацію, яку зловмисники використовують, щоб перешкодити та отримати контроль над програмою або машиною. Після успішної операції зловмисник може заблокувати цільову програму (викликавши відмову в обслуговуванні) або, можливо, отримати доступ до всіх прав та дозволів, доступних для

скомпрометованої програми. На рисунку 2.3 показано внутрішній дизайн каркасу NIPS.

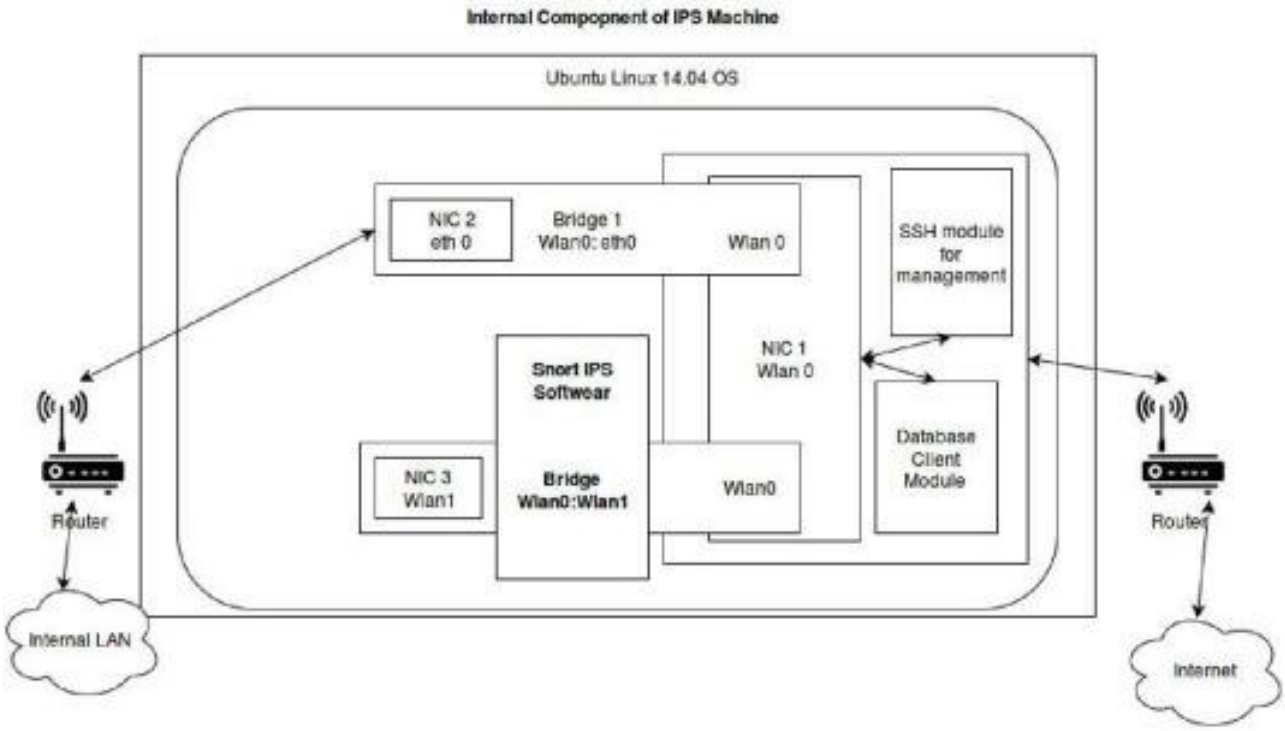


Рисунок 2.3 - Внутрішня архітектура системи NIPS. [14]

IPS зазвичай знаходиться за брандмауером і дає додатковий рівень захисту для системи. На відміну від свого попередника, IDS, яка є прихованою структурою, яка виводить інформацію про активність і повідомляє про небезпеки, налаштована на вбудований режим (для безпосереднього зв'язку між джерелом і метою), в даний час досліджуючи та вживаючи механізовані дії для всіх потоків руху , що входять до системи. Зокрема, ці дії включають: відправлення попередження головному пристрою (як в IDS), фільтрація шкідливих пакетів, блокування активності з вихідної адреси і скидання з'єднання. IPS має працювати ефективно, щоб не заважати роботі мережі для частини вбудованої безпеки. Він також повинен працювати швидко, тому що атака може статися близько і послідовно.

На рисунку 2.4 показано зовнішню архітектуру системи NIPS.

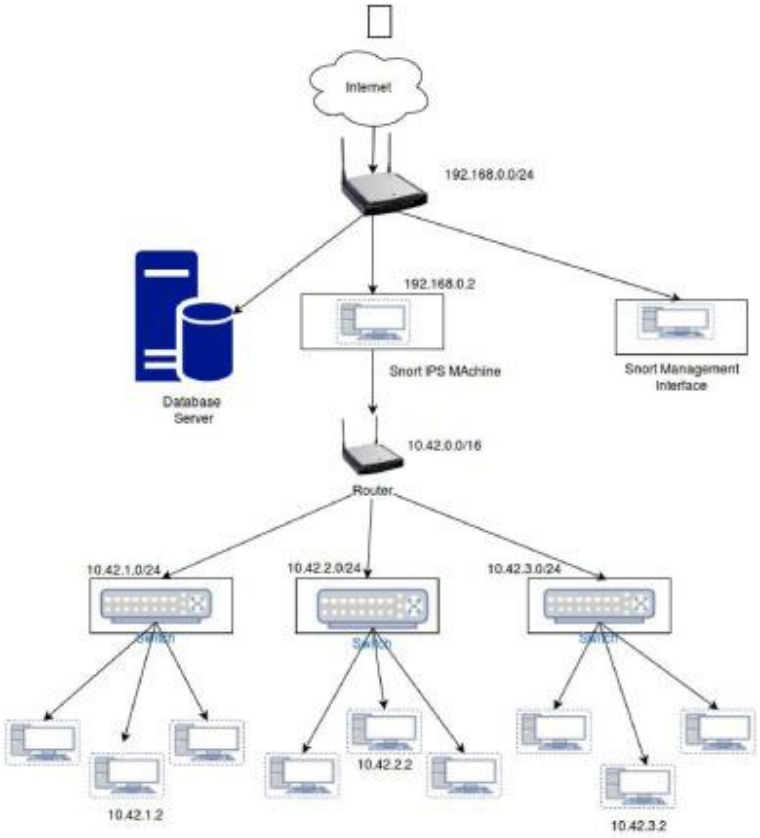


Рисунок 2.4 - Зовнішня архітектура системи NIPS. [15]

2.2. Аналіз компонентів IPS

До складу системи IPS входять такі компоненти як вбудований режим, мережевий кран, порт комутатора, базова мережа, екстранети, точки віддаленого доступу, платформи віртуалізації та первинні мережеві сегменти.

Вбудований режим: IPS налаштований на вбудований режим за брандмауером або комутатором, щоб усі операції системи проходили через нього. Це компонування підтримує два режими IPS (блокування) та IDS (тривожна сигналізація).

Мережевий кран: кран – це гаджет обладнання, що дозволяє інформації проходити через систему. Відведення убік зазвичай використовується для вбудованих IPS пристроїв для IPS, які не мають функції швидкого відкриття, або для асоціацій, де може бути бажано зберегти їх вбудований IP окремо від

системи для обслуговування або реконфігурації. Відведення відновлення часто використовується для неактивних налаштувань IDS, коли порти обходу на спостережуваних пристроях комутатора витрачені (рисунок 2.5).

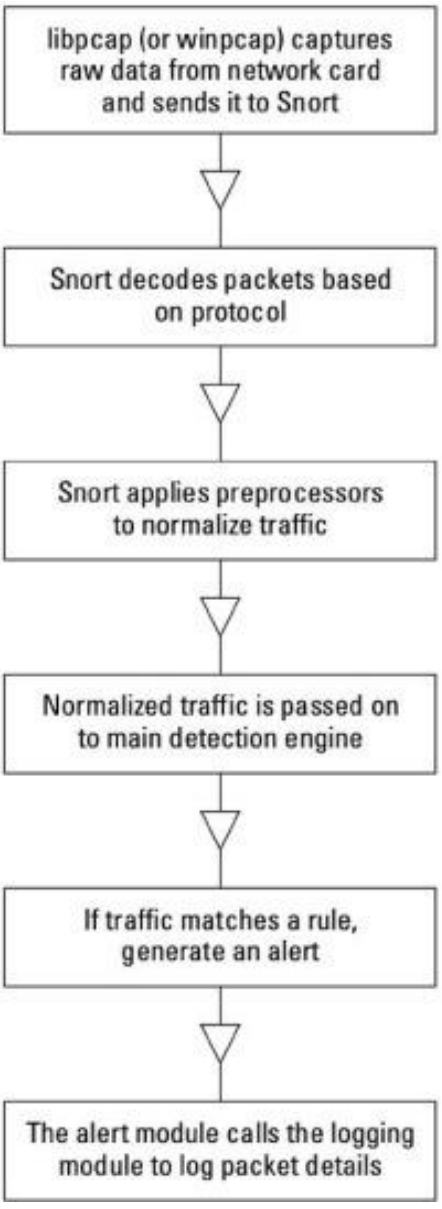


Рисунок 2.5 - Схема обробки внутрішніх пакетів Snort. [16]

Порт комутатора: це порт на системному комутаторі, де можна спостерігати дублювання всіх дій, що протікають через комутатор, що забезпечує відокремлене налаштування IDS.

Базова мережа чи мережа центру обробки даних (рисунок 2.6): дедалі більше компаній розширюють гарантії своїх периферійних IPS, впроваджуючи

датчики IPS (зазвичай встановлені у прихованому режимі IDS) у центрі чи інформаційному центрі. Це створює додатковий рівень бар'єру та ідентифікує атаки, доставлені на робоче місце вручну за допомогою портативних пристроїв обробки даних.

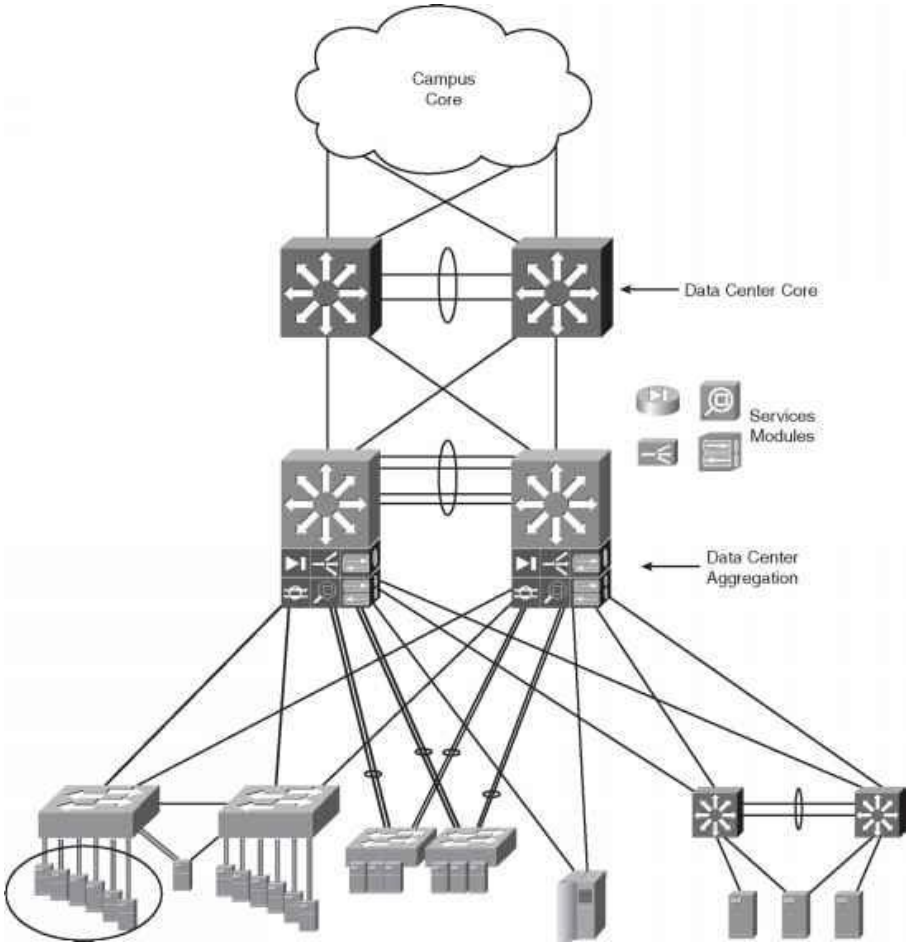


Рисунок 2.6 - Мережа центру обробки даних [17]

Платформи віртуалізації: хоча віртуалізація має великі переваги з погляду зниження витрат, вона також становить нові небезпеки та вразливості. Фізична IPS, що встановлюється перед віртуалізацією, або віртуальна IPS, що впроваджується при кожній віртуалізації, допоможе захистити від прихованих атак, що починаються зсередини або фокусуються на віртуальних машинах.

Екстранети (рисунок 2.7): більші асоціації з асоціаціями екстрамережі з системами партнера або провайдера можуть поміщати вбудований гаджет IPS перед відповідними комутаторами, щоб захистити від потенційних атак, що

наближаються, і гарантувати, що прилеглі шкідливі програми не поширяться на системи партнера.

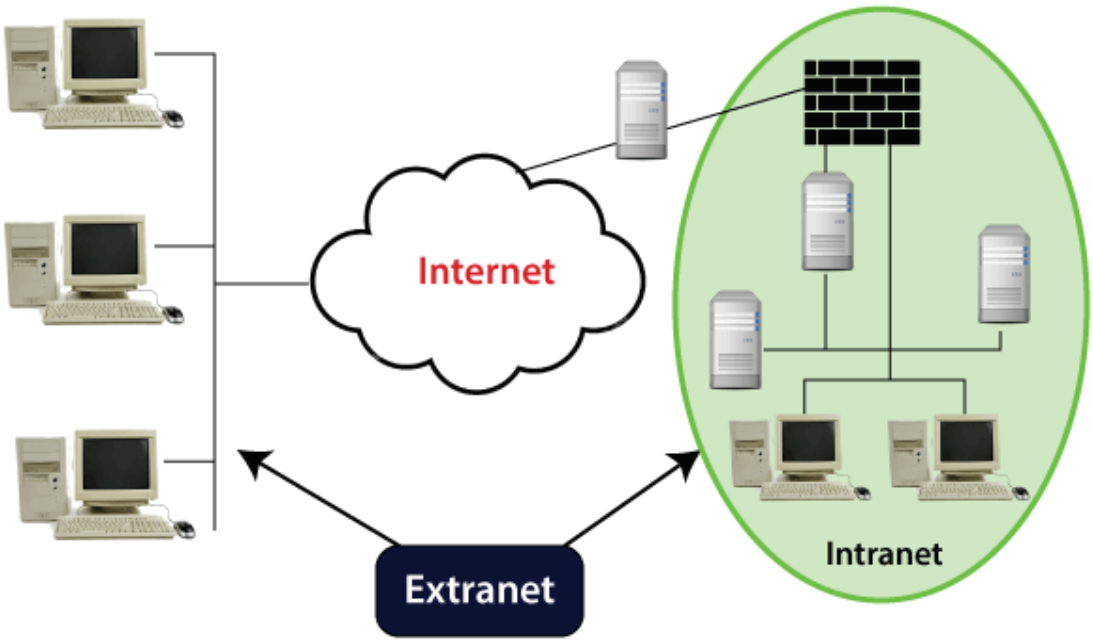


Рисунок 2.7 – Екстранет [18]

Точки віддаленого доступу: наймані працівники та відвідувачі зазвичай взаємодіють із системою через віддалений доступ. Оскільки ці гаджети регулярно не контролюються IT-фахівцями, датчики IPS за віддаленими проходами відстежують потенційний небажаний рух.

Первинні мережні сегменти: це можуть бути системи, що містять базові структури (наприклад, сервери, що містять інформацію, пов'язану з грошима або відновленням), де збої можуть бути справжніми. Процес сегментації мережі передбачає поділ фізичної мережі на різні логічні підмережі. Після того, як мережу було поділено на менші більш керовані одиниці, елементи керування застосовуються до окремих розділених сегментів.

Для вирішення поставлених цілей буде використовуватися таке програмне забезпечення як Barnyard2 v2.1-13, базовий механізм аналізу та безпеки (BASE) v1.4.5, Snort v2.9.8.2 та операційна система Linux Ubuntu 18.04

Server або Desktop Version LTS. На рисунку 2.8 показано обладнання, необхідне реалізації IPS.

Component/Item Name	Description	Quantity Needed
IPS Machine	HP ProLiant XL450 Gen9 Server	1
Ethernet Adapters	Intel® Ethernet Server Adapter I210	2
Router	Cisco WRVS4400N Wireless-N Gigabit Security Router	2
Switches	Cisco 24 Ports 10/100 Switch (SF90-24)	As required
Wireless Network Adapters	Netgear 802.11ac Dual Band Gigabit A6200	2
Database Server	Oracle SPARC Database Server	1

Рисунок 2.8 - Апаратні вимоги IPS, що впроваджується [19]

2.3 Режими роботи Snort

Snort (рисунок 2.9) можна налаштувати для роботи в чотирьох різних режимах, тобто як сніффер пакетів, реєстратор пакетів, система виявлення та вбудований режим (IPS):

1) режим Sniffer - у цьому режимі Snort використовує пакет інструмент захоплення для перехоплення пакетів із мережевого трафіку та відобразити його на консолі. У режимі Sniffer журнал не ведеться;

2) режим реєстратора - у цьому режимі реєструються пакети вторинне сховище для використання в майбутньому;

3) режим NIDS - у цьому режимі Snort аналізуватиме вміст пакета, порівняйте його з набором попередньо визначених правила та створювати сповіщення, якщо знайдено відповідність (тобто, якщо пакет визнано шкідливим);

4) вбудований режим - цей режим відомий як вторгнення режим профілактики. У цьому режимі Snort братиме необроблені дані з IPTABLES і перевірте його на відповідність набору правил. Якщо якісь сповіщення генерується, то потім правила IPTABLE оновлюються відповідно щоб запобігти зловмисній діяльності.

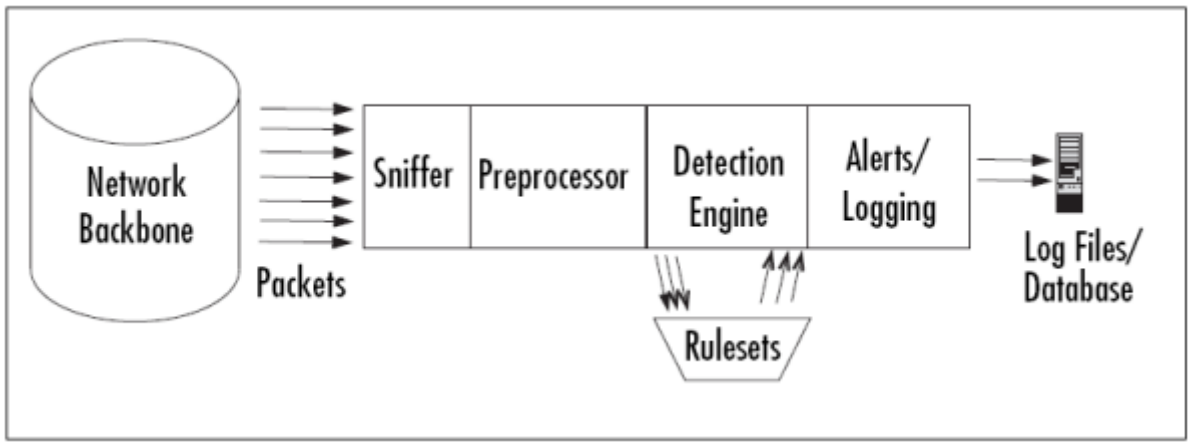


Рисунок 2.9 - Механізм виявлення Snort [20]

Модуль Sniffer використовується для аналізу пакетів з мережі. Це для цього використовує бібліотеку pcap (Packet Capture) [21]. Одного разу пакет перехоплюється, він передається декодеру для розміщення різні покажчики для подальшої обробки. Сніффер пакетів – це пристрій (апаратний або програмне забезпечення), що використовується для підключення до мереж [22]. Працює аналогічно мода на телефонне прослуховування, але воно використовується для даних мережі замість голосових мереж. Мережевий сніффер дозволяє програмі або апаратному пристрою прослуховувати мережевий трафік даних.

Сніфери пакетів мають різне використання:

1) аналіз мережі та усунення несправностей;

2) аналіз ефективності та порівняльний аналіз;

3) підслуховування відкритих паролів та інше цікаві шматочки даних.

Шифрування вашого мережевого трафіку може перешкодити людям ваші пакети та їх прочитати. Тому шифруванням займається мережевий інструмент . Також можна використовувати сніфери пакетів для різних типів повідомлень.

Декодер розмізщує різні покажчики в структурі пакета, щоб полегшити іншим модулям отримання даних певного рівня. Після встановлення покажчиків пакет передається препроцесору.

Оскільки snort є системою виявлення вторгнень на основі правил, він не може виявити аномалію, поширену на кілька пакетів. Препроцесор використовується саме для цього. Це окремий плагін-модуль, який можна завантажити, зробивши запис у Snort конфігураційний файл. Після запуску snort із препроцесором підтримка виявлятиме аномалії на основі типу препроцесора [23].

Після того, як пакети будуть оброблені всіма включеними препроцесорів, вони передаються механізму виявлення. Механізм виявлення є основним елементом IDS на основі сигнатур в Snort. Механізм виявлення приймає отримані дані з препроцесорів і його плагінів, і ці даніперевіряється за набором правил. Якщо правила збігаються з даними в пакет, вони надсилаються до процесора оповіщення.

Функція IDS на основі підпису досягається за допомогою різних наборів правил. Набори правил є згруповані за категоріями (троянські коні, переповнення буфера та доступ до різних програм) і регулярно оновлюються [24].

Самі правила складаються з двох частин:

- заголовок правила: заголовок правила в основному є дія, яку потрібно виконати (журнал або сповіщення), тип мережевого пакета (TCP, UDP, ICMP тощо), IP джерела та призначення адреси та порт;
- параметр правила: параметр – це вміст у пакет, який має зробити пакет відповідним правилу.

2.4 Висновки до розділу

Отже, системи виявлення та запобігання мережевим вторгненням NIDS/NIIPS мають перевагу перед іншими системами виявлення та запобігання вторгненням, оскільки вони не обтяжують контрольовані хости. Під час впровадження особливу увагу слід приділити конфігурації впровадженої системи NIDS/NIPS, оскільки якщо вона буде налаштована неправильно то генеруватиме сотні тисяч помилкових спрацьовувань, які істотно заважають аналізу подій. Надійність NIDS/NIPS підтверджується низькою кількістю помилкових спрацьовувань.

Найновіші системи NIDS також можуть розпізнавати керовану операційну систему на основі характеристик заданого стеку TCP/IP система. Ця функція дозволяє вводити дані ієрархію важливості для окремих тривог залежно від типу системи. Впровадженню системи NIDS/NIPS має передувати детальний аналіз контрольованої мережі IoT, як її структури, так і характеру даних надіслано в ньому. Це має спрацювати для правильного розташування елемента системи NIPS, що є визначальним про стабільність мережі. Впроваджена система NIDS/NIPS вимагає постійної адаптації до мінливих загроз і характеру мережевого трафіку. Дуже важливо в цьому процесі аспект документування введених підписів і команди, щоб підтримувати прозорість і читабельність і відтворюваність - у разі можливої перебудови система. Добре налаштований NIDS є найкращим джерелом інформації про безпеку мережі ІКТ.

До переваг та недоліків Snort можна віднести те, що це дуже гнучкий додаток. Завдяки його модульній конструкції і можливості додавати або вимикати спеціалізовані програмні компоненти Snort, він може бути потужним інструментом для поглибленої реалізації захисту мережі. Це дозволяє будь-кому, хто вміє програмувати, створювати та впроваджувати власні модулі препроцесора та налаштовувати роботу Snort відповідно до особливостей

середовища. Налаштування також можна здійснити за допомогою спеціалізованих конфігурацій існуючих препроцесорних модулів як операції виведення сповіщень.

Snort дійсно має деякі обмежені недоліки, коли справа доходить до аномалії виявлення. Система не була розроблена для такого типу операцій, але деякі препроцесорні модулі намагаються додати цю функціональність [18]. Наразі ці модулі не вважаються ефективними у виявленні. Там також занепокоєння щодо того, наскільки ефективним є механізм виявлення терміни виконання обробки. Базовий двигун вважається цілком ефективна, але є припущення щодо того, наскільки ефективною стає система при використанні з модулями препроцесора. Додатковий функціонал є добре, але яку ціну ви повинні заплатити за цю функціональність.

Використання правил SNORT дозволяє мережевим адміністраторам легко відрізнити звичайну, очікувану активність в Інтернеті від усього, що виходить за межі норми. SNORT аналізує мережеву активність у реальному часі, щоб виявити шкідливу активність, а потім генерує сповіщення для користувачів.

У даному розділі проведено:

1) аналіз побудови системи захисту мережі;

2) описано внутрішню архітектуру системи NIPS;

3) розроблено принципи налаштування інструментів безпеки;

4) наведено програмне та апаратне забезпечення, яке буде використовуватися під час виконання роботи;

5) проведене налаштування компонентів захисту мережі.

3. РЕАЛІЗАЦІЯ СИСТЕМИ ЗАПОБІГАННЯ ПРОНИКНЕННЮ В МЕРЕЖІ ІНТЕРНЕТ-РЕЧЕЙ

3.1. Налаштування інструментів безпеки

Встановлення та налаштування інструментів безпеки будуть відбуватися на попередньо встановленій Linux Ubuntu 18.04 у VirtualBox.

Для початку роботи необхідно встановити систему виявлення вторгнень та запобігання вторгненням Snort:

sudo apt-get install -y libnghttp2-dev

cd ~/snort_src wget https://snort.org/downloads/snort/snort-2.9.17.tar.gz

tar -xvzf snort-2.9.17.tar.gz cd snort-2.9.17 ./configure --enable-sourcefire

make

sudo make instal

Результат правильного встановлення та налаштування Snort показаний на рисунку 3.1.

Рисунок 3.1 – Встановлення Snort

Далі потрібно налаштувати Snort для запуску як NIDS.

Перш ніж для правильного налаштування Snort, давайте подивимося на файл довідки (рисунок 3.2). Для цього використаємо таку команду snort –help.

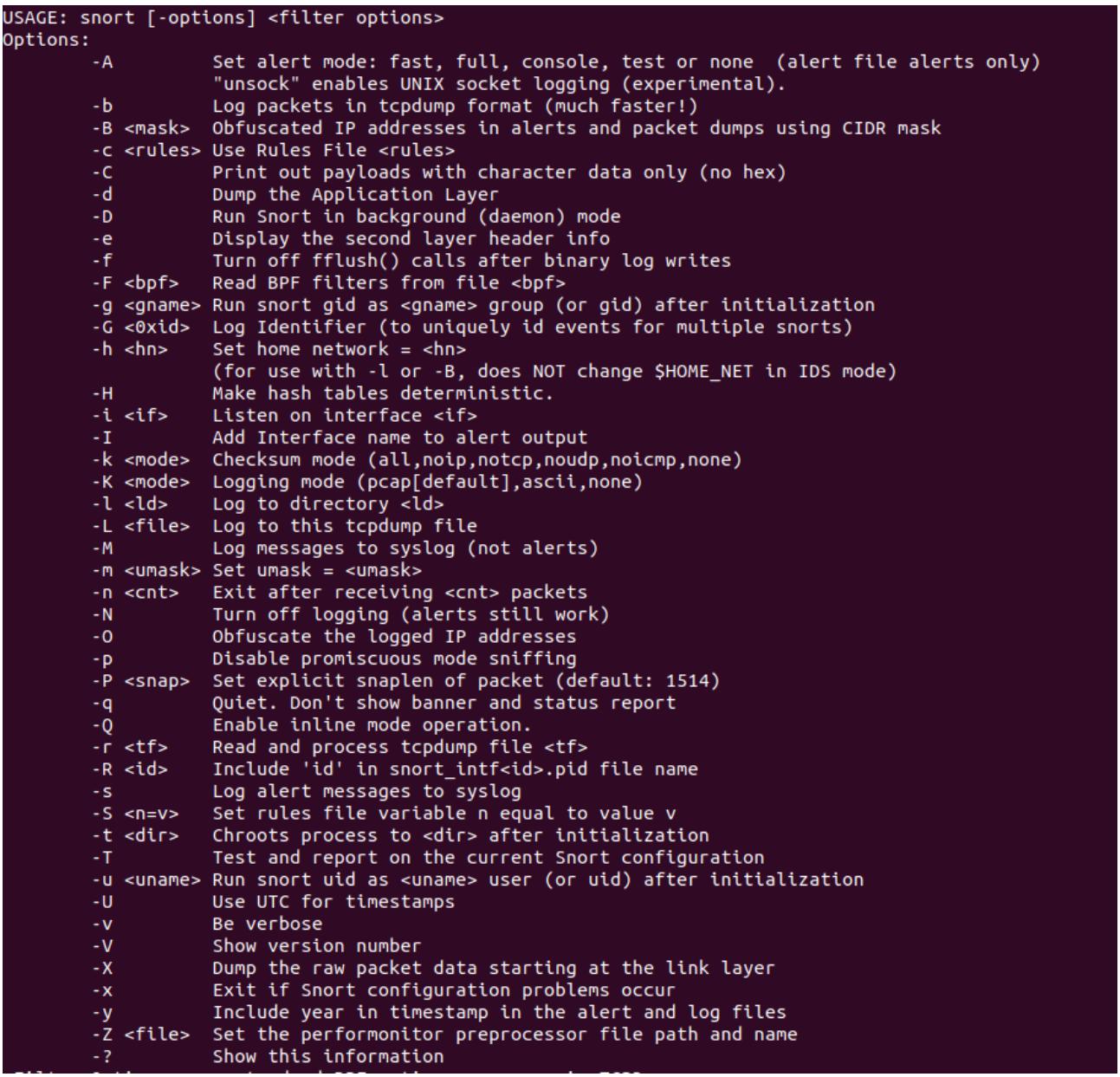


Рисунок 3.2 – Параметри довідки Snort

Параметри довідки Snort:

1) перший параметр -c разом із розташуванням файлу правил Snort вказує Snort використовувати свої правила. Правила в Snort схожі на сигнатури вірусів; вони призначені для виявлення шкідливих пакетів і програмного забезпечення;

2) другий — -d, який вказує Snort показати прикладний рівень даних;

3) третій — -e, який повідомляє Snort про відображення інформації другого рівня або рівня каналу даних, який містить MAC-адресу;

4) перемикач -i дозволяє нам вказати інтерфейс, який ми хочемо використовувати. За замовчуванням Snort використовує інтерфейс eth0, тому нам потрібно використовувати його, лише якщо ми хочемо використовувати інший, наприклад wlan0;

5) перемикач -l вкаже Snort, де реєструвати дані. Як ми побачимо, залежно від конфігурації Snort за замовчуванням записує в /var/log/snort, але ми можемо вказати тут інше розташування, розмістивши шлях після перемикача -l;

6) перемикач -v повідомляє Snort бути багатослівним, тобто багатослівним, щоб надати нам всю інформацію.

Щоб використовувати Snort як NIDS (рисунки 3.3-3.4), потрібно вказати у Snort включити файл конфігурації та правила. Як правило, ми можемо знайти файл conf за адресою /etc/snort/snort.conf , і цей файл вказуватиме на правила Snort. Нам потрібно вказати ключ -c , а потім розташування:

sudo snort -vde -c /etc/snort/snort.conf

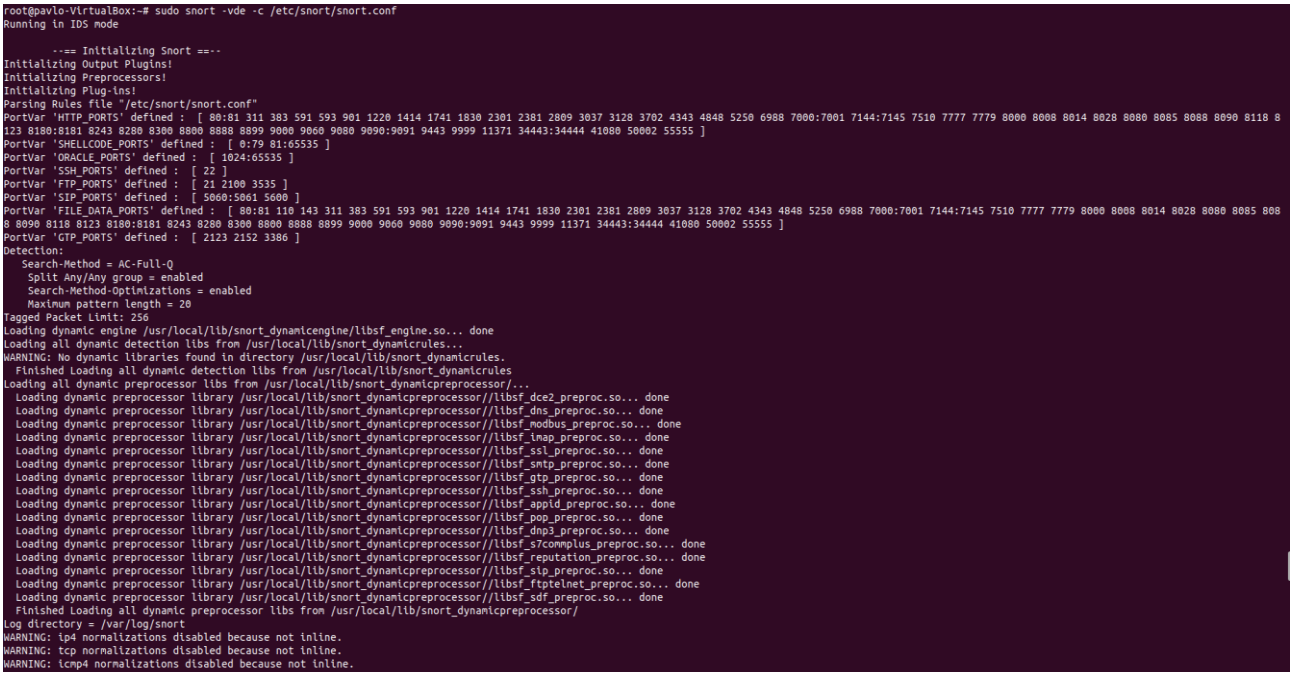


Рисунок 3.3 – Snort як NIDS

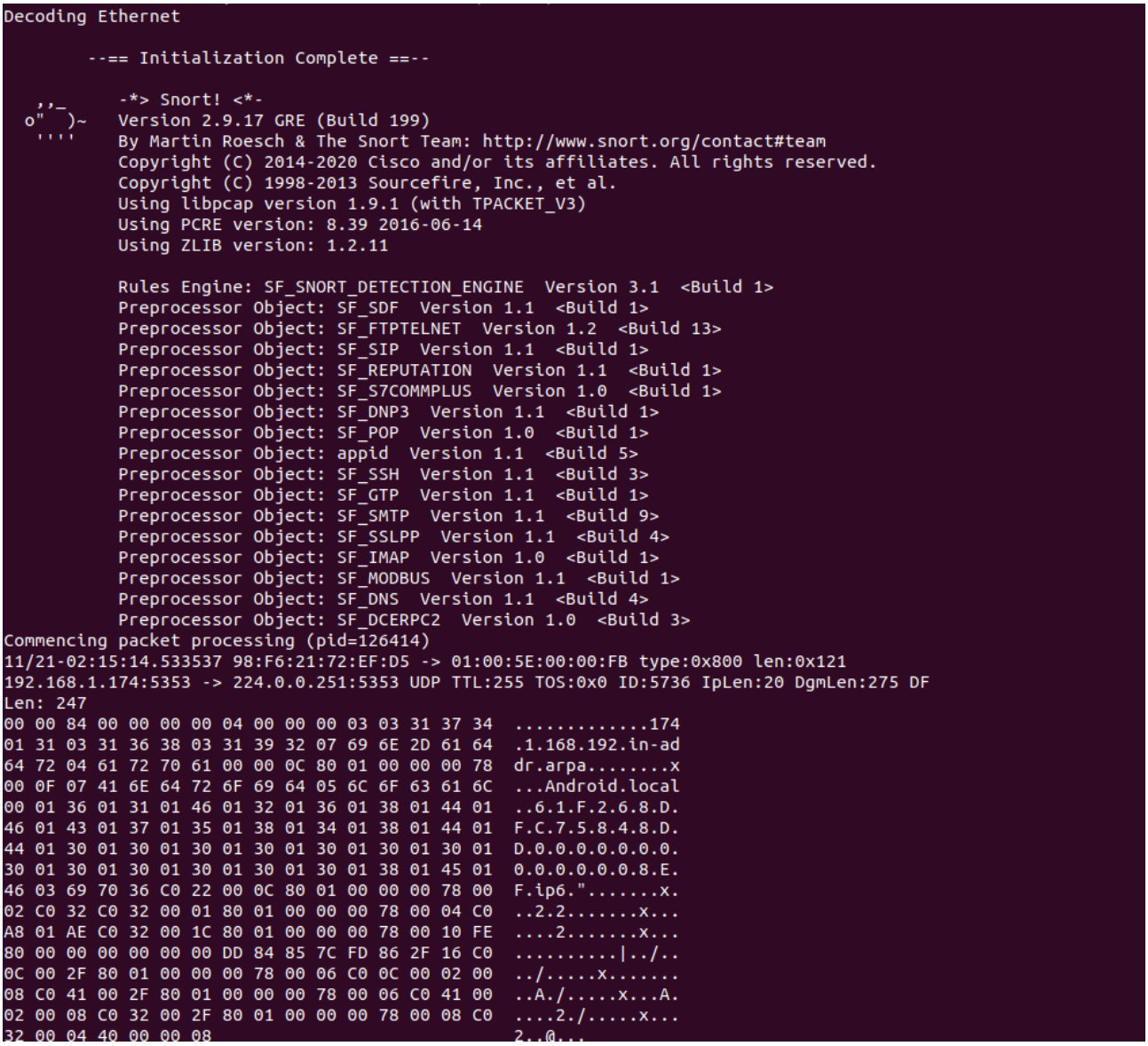
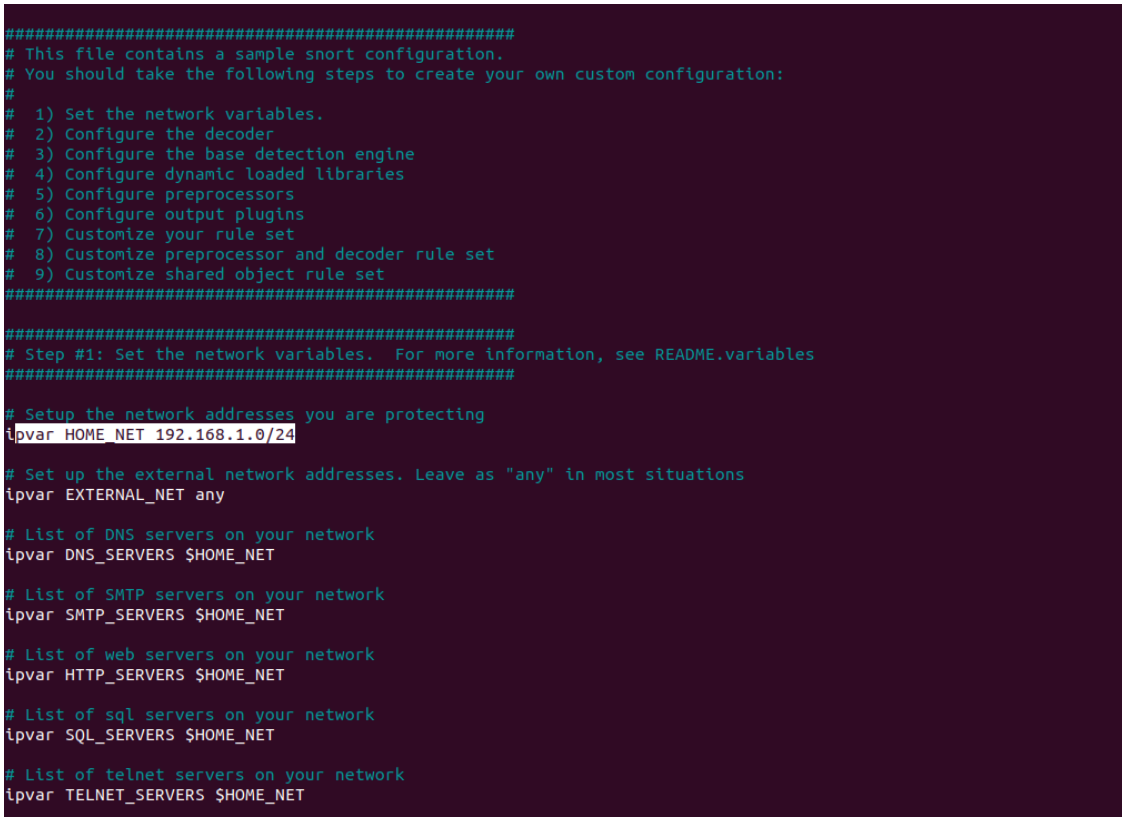


Рисунок 3.4 – Snort як NIDS

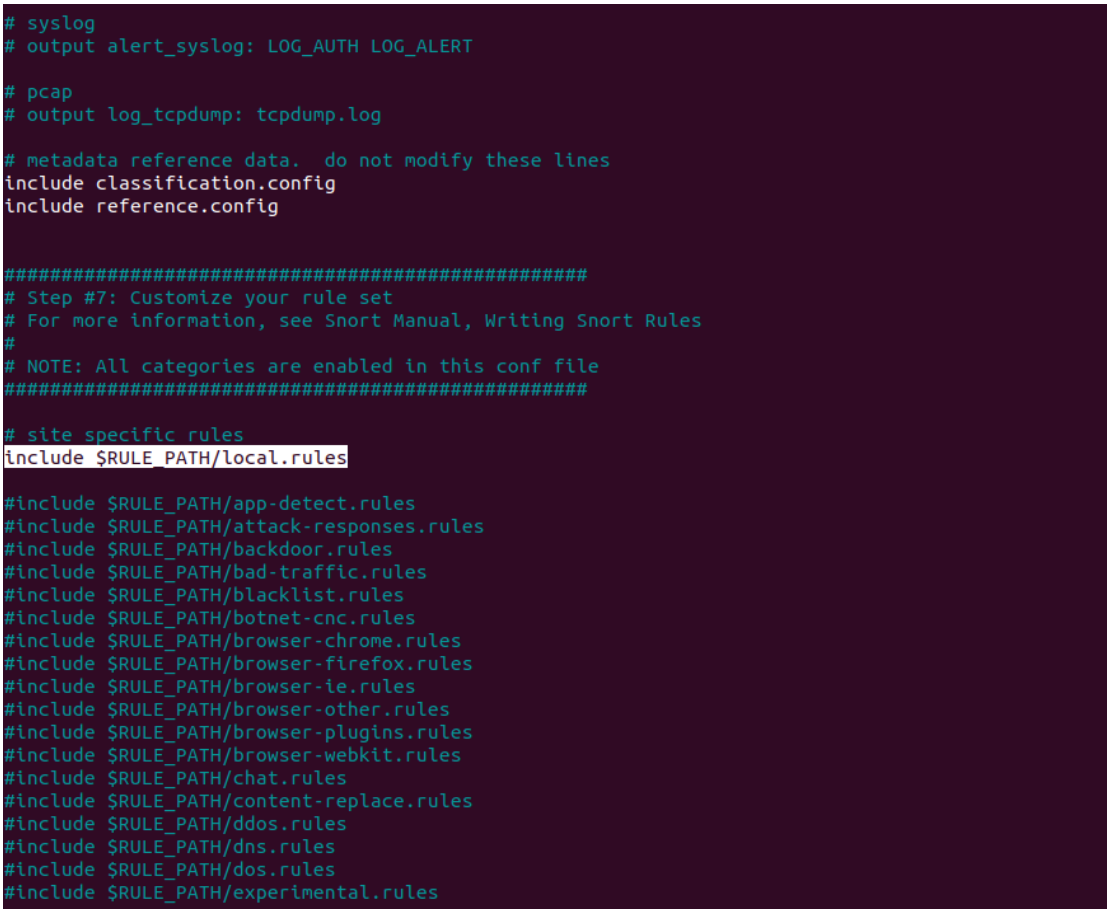
Щоб налаштувати правила відповідно до наших певних задач нам потрібно змінити певні правила і налаштування Snort. Для цього використаємо команду sudo nano /etc/snort/snort.conf

На наведеному рисунку 3.5 ми бачимо виділений рядок «ipvar HOME_NET». Ця змінна вказує на мережу, яку потрібно захистити. HOME_NET — це ім’я змінної, якій призначається IP-адреса. Це може бути одна IP-адреса, список IP-адрес або навіть підмережа в нотації CIDR. Тому вказуємо в полі змінної домашню адресу мережі, яку потрібно захистити (рисунок 3.6).



```
##################################################
# This file contains a sample snort configuration.
# You should take the following steps to create your own custom configuration:
#
#  1) Set the network variables.
#  2) Configure the decoder
#  3) Configure the base detection engine
#  4) Configure dynamic loaded libraries
#  5) Configure preprocessors
#  6) Configure output plugins
#  7) Customize your rule set
#  8) Customize preprocessor and decoder rule set
#  9) Customize shared object rule set
##################################################

##################################################
# Step #1: Set the network variables.  For more information, see README.variables
##################################################

# Setup the network addresses you are protecting
ipvar HOME_NET 192.168.1.0/24

# Set up the external network addresses. Leave as "any" in most situations
ipvar EXTERNAL_NET any

# List of DNS servers on your network
ipvar DNS_SERVERS $HOME_NET

# List of SMTP servers on your network
ipvar SMTP_SERVERS $HOME_NET

# List of web servers on your network
ipvar HTTP_SERVERS $HOME_NET

# List of sql servers on your network
ipvar SQL_SERVERS $HOME_NET

# List of telnet servers on your network
ipvar TELNET_SERVERS $HOME_NET
```

Рисунок 3.5 – Екран відкриття файлу конфігурації



```
# syslog
# output alert_syslog: LOG_AUTH LOG_ALERT

# pcap
# output log_tcpdump: tcpdump.log

# metadata reference data.  do not modify these lines
include classification.config
include reference.config


##################################################
# Step #7: Customize your rule set
# For more information, see Snort Manual, Writing Snort Rules
#
# NOTE: All categories are enabled in this conf file
##################################################

# site specific rules
include $RULE_PATH/local.rules

#include $RULE_PATH/app-detect.rules
#include $RULE_PATH/attack-responses.rules
#include $RULE_PATH/backdoor.rules
#include $RULE_PATH/bad-traffic.rules
#include $RULE_PATH/blacklist.rules
#include $RULE_PATH/botnet-cnc.rules
#include $RULE_PATH/browser-chrome.rules
#include $RULE_PATH/browser-firefox.rules
#include $RULE_PATH/browser-ie.rules
#include $RULE_PATH/browser-other.rules
#include $RULE_PATH/browser-plugins.rules
#include $RULE_PATH/browser-webkit.rules
#include $RULE_PATH/chat.rules
#include $RULE_PATH/content-replace.rules
#include $RULE_PATH/ddos.rules
#include $RULE_PATH/dns.rules
#include $RULE_PATH/dos.rules
#include $RULE_PATH/experimental.rules
```

Рисунок 3.6 – Параметри, доступні для конфігурації

Щоб не дозволити Snort використовувати певний набір, просто закоментовуємо частину «include».

Після внесення будь-яких змін зберегаємо файл і перевіряємо конфігурацію за допомогою перемикача -T (рисунок 3.7): sudo snort -T -c /etc/snort/snort.conf

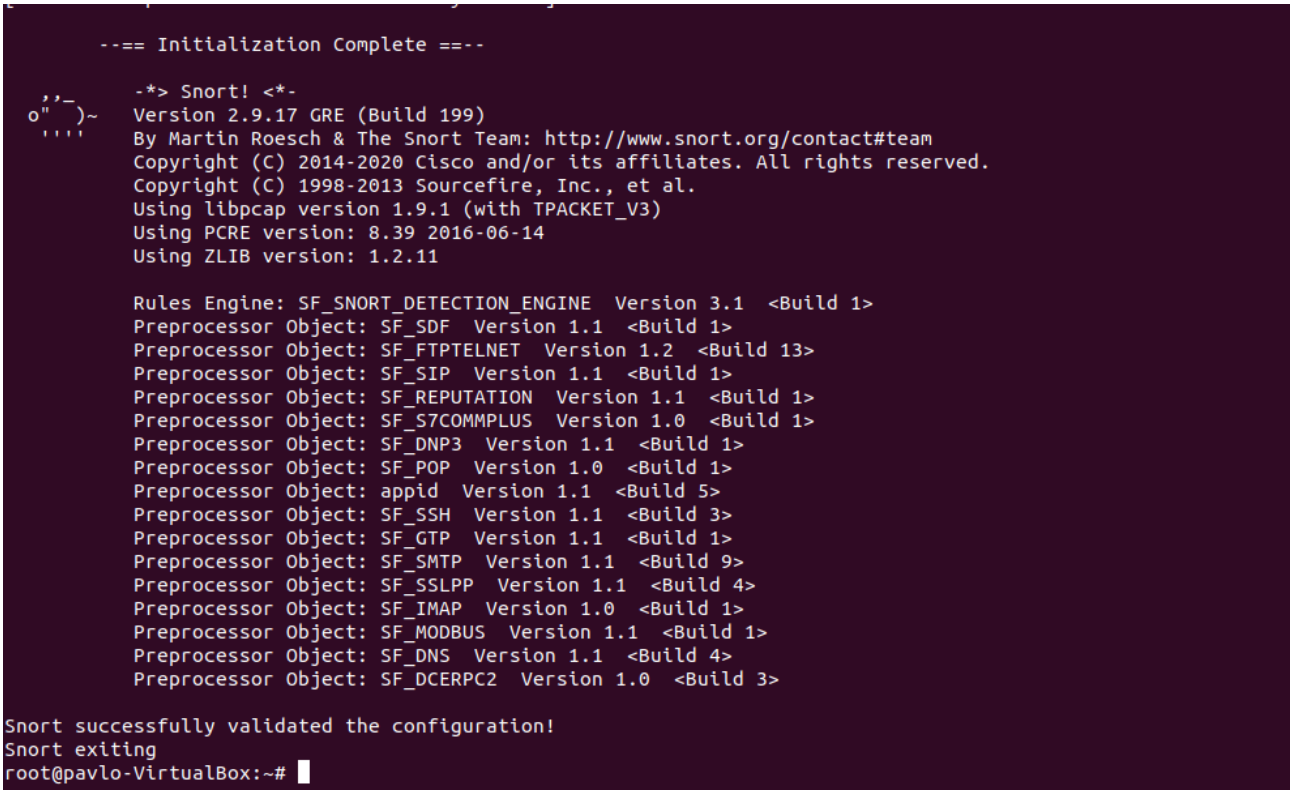


Рисунок 3.7 – Остаточна перевірка конфігурації

Оскільки шкідливе програмне забезпечення і методи злому мережі постійно розвиваються, Snort потрібні нові набори правил, щоб розрізняти останні види підозрілої активності (набори правил подібні до антивірусних міток). BASE надає веб-інтерфейс для опитування та вивчення аварійних сигналів, що виходять із середовища Snort. По-перше, ми повинні представити все необхідне з магазинів Ubuntu: sudo apt-get install -y build-essential libpcap-dev libpcre3-dev libdumbnet-dev bison flex zlib1g-dev.

Пакети які необхідно попередньо встановити:

1) bison і flex: вони мають необхідний синтаксичний аналізатор (DAQ представлений нижче);

2) libpcap-dev: бібліотека для захоплення мережевого трафіку, необхідного Snort;

3) libpcre3-dev: бібліотека можливостей для стандартних функцій потрібних Snort;

4) libdumbnet-dev: бібліотека libdnet надає зрозумілий та зручний інтерфейс для кількох низькорівневих розкладів системного адміністрування;

5) snort використовує бібліотеку захисту інформації (DAQ) для перехоплення дзвінків для зв'язування бібліотек захоплення. DAQ завантажується та вводиться із сайту SNORT.

Однією з проблем, які виникають із Snort, є те, як Snort може продовжувати обробку мережевого трафіку без відкидання пакетів і виконання великих операцій виводу, таких як надсилання сповіщень і реєстрація їх у системний журнал або базу даних. Для цього використовується додаткова програми, яка буде зосереджена на створенні сповіщень. Зв’язок між Snort і цією вторинною програмою має здійснюватися шляхом буферизації даних попереджень за допомогою певних типів файлів. Тому, для цих завдань використовують Barnyard.

Barnyard можна розглядати як асинхронний інструмент обробки подій і диспетчеризації, призначений для використання зі Snort. Крім того, використовуючи Barnyard разом із snort, ми матимемо додаткові переваги, оскільки ми можемо розділити рівень доступу до snort. Ще однією перевагою використання Barnyard є те, що ми завжди можемо повторно проаналізувати архівні дані, видані уніфікованим плагіном Barnyard.

Принцип роботи полягає в тому, що Snort зберігає події у двійковій формі, а потім Barnyard2 зчитує ці події асиметрично та зберігає їх у базі даних MYSQL. Barnyard потребує деяких запитів для встановлення та роботи, ймовірно, таких як компілятор C, сервер і клієнт MYSQL. Після чого Snort створює унікальний формат бінарних документів, що називається уніфікованим. Barnyard переглядає цей документ, а потім повторно надсилає інформацію до бази даних на сервері. На відміну від модуля прибутковості бази

даних, Barnyard знає про неможливість надіслати сигнал тривоги до бази даних і припиняє надсилання попереджень. Також важливо пам'ятати, коли база даних може знову підтвердити асоціацію та розпочати повторне відправлення сигналів тривоги.

Для встановлення Barnyard2 (рисунок 3.8) попередньо встановлюємо базу даних MYSQL вибираючи потрібну розрядність системи:

```
sudo apt-get install -y mysql-server libmysqlclient-dev mysql-client autoconf libtool-
./configure --with-mysql --with-mysql libraries=/usr/lib/x86_64-linux-gnu або ./configure --with-mysql --with-mysql-libraries=/usr/lib/i386-linux
```

Щоб завантажити, розпакувати та встановити Barnyard, потрібно використати наступні команди:

```
wget https://github.com/firnsy/barnyard2/archive/7254c24702392288fe6be948f88afb74040f6dc9.tar.gz \-O barnyard2-2-1.14-336.tar.gz
tar zxvf barnyard2-2-1.14-336.tar.gz
mv barnyard2-7254c24702392288fe6be948f88afb74040f6dc9 barnyard2-2-1.14-336
cd barnyard2-2-1.14-336
autoreconf -fvi -I ./m4
make
sudo make install
```

Усі конфігурації знаходяться в спеціальному конфігураційному файлі /etc/snort/barnyard2.conf. Цей файл буде містити всю інформацію, необхідну для підключення Banyard2 до бази даних MySQL. Наступний крок - тестування системи.

Для цього виконуємо такі команди:

```
sudo /usr/local/bin/snort -q -u snort -g snort -c /etc/snort/snort.conf -i eth0
sudo barnyard2 -c /etc/snort/barnyard2.conf -d /var/log/snort -f snort.u2 -w /var/log/snort/barnyard2.waldo -g snort -u snort
```

Перевірка правильності встановлення та налаштування програми представлена на рисунках 3.9 та 3.10.

Рисунок 3.8 - Встановлення barnyard2

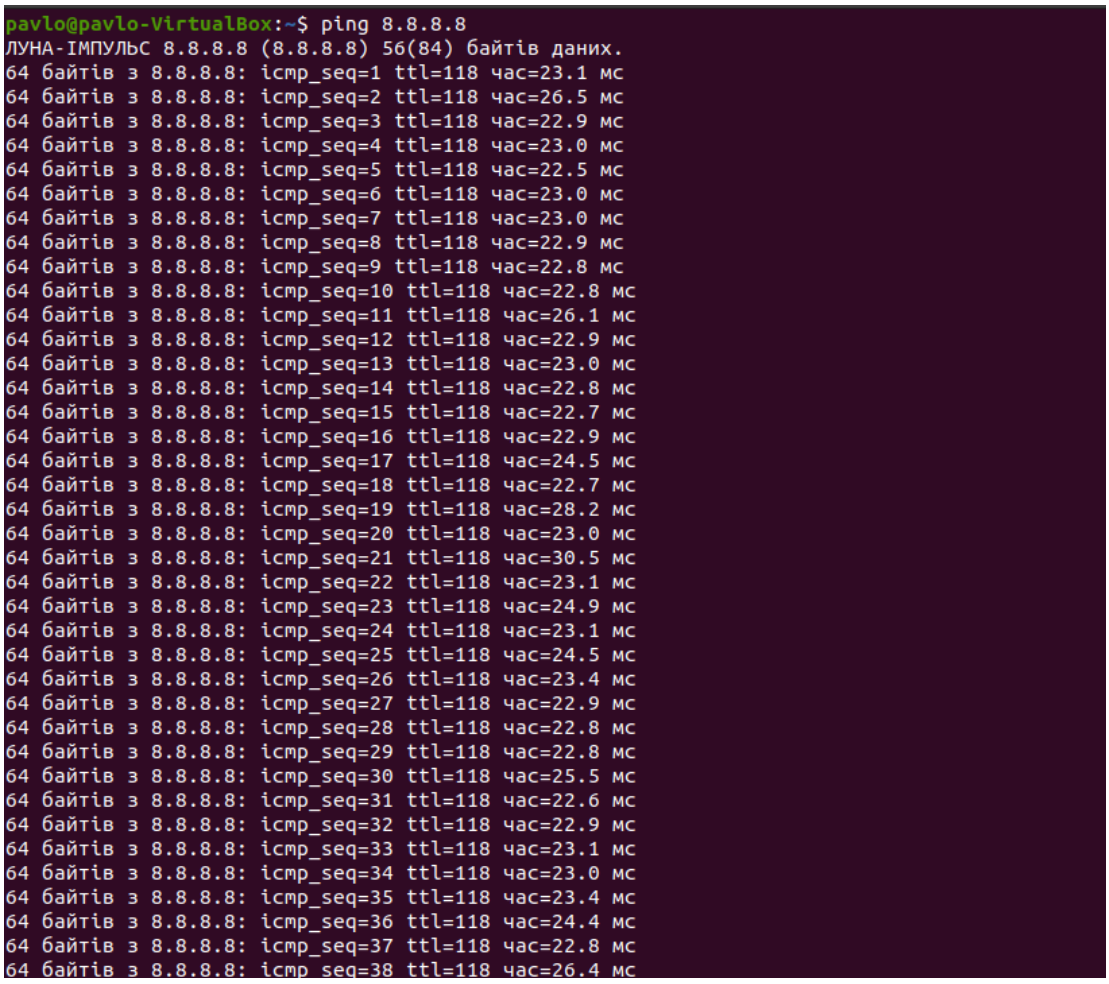


Рисунок 3.9 - Перевірка роботи barnyard2

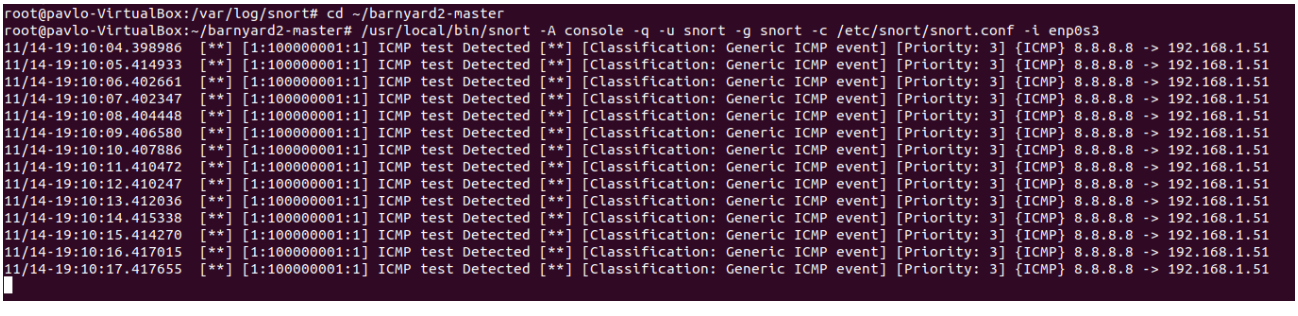


Рисунок 3.10 - Перевірка роботи barnyard2

Оскільки Snort має кілька режимів, Barnyard також надає два режими: пакетна обробка та безперервна обробка. Спочатку в режимі пакетної обробки Barnyard обробить усі попередньо визначені уніфіковані файли, а потім завершить роботу. Перевагами цього режиму є отримання матеріальних даних з уніфікованого файлу, перезавантаження старих даних у базу даних або тестування нових плагінів, які використовуються в snort. По-друге, у безперервному режимі події можуть бути оброблені миттєво, якщо вони викликали сповіщення про snort.

Barnyard має ще одну можливість дуже легко локалізувати повідомлення попереджень, оскільки дані завантажуються з файлів sid-msg.map і gen-msg.map. На відміну від snort, який має 48 файлів правил, препроцесорів і параметрів правил. У результаті, якщо нам потрібно локалізувати повідомлення за допомогою Barnyard, нам потрібно лише створити нові sid-msg.map і gen-msg.map. Коли необхідно додати нове правило, новий запис можна вставити до цих двох основних файлів Barnyard.

Одним із популярних інструментів із відкритим кодом для перегляду даних Snort є Basic Analysis and Security Engine (BASE), інструмент на базі PHP, який отримує дані з бази даних, наприклад MySQL, і форматує їх для представлення у веб-браузері. Цей набір інструкцій зосереджений на BASE.

Інструмент BASE – це важливий інструмент розслідування та забезпечення безпеки. Ця програма надає веб-інтерфейс для запитів та перевіряє сигнали тривоги, що виходять з інфраструктури Snort IDS. Окремо SNORT продовжує працювати у незалежному режимі як аналізатор пакетів та реєстратор. У поєднанні з різними додатками та деякими конфігураціями середовище Snort стає значно кориснішим, ніж NIDS. Головною перевагою використання BASE є те, що його легко налаштувати для роботи. Наприклад:

```
sudo add-apt-repository ppa:ondrej/php
sudo apt-get update
sudo apt-get install -y apache2 libapache2-mod-php5.6 php5.6-mysql php5.6-cli php5.6 php5.6-common php5.6-gd php5.6-cli php-pear php5.6-xml
sudo pear install -f --alldeps Image_Graph
```

Завантаження та встановлення ADODB:

```
cd ~/snort_src
wget https://sourceforge.net/projects/adodb/files/adodb-php5-only/adodb-520-for-php5/adodb-5.20.8.tar.gz
tar -xvzf adodb-5.20.8.tar.gz
sudo mv adodb5 /var/adodb
sudo chmod -R 755 /var/adodb
sudo chmod -R 755 /var/adodb
```

Заваантаження BASE та копіювання в корінь apache:

```
cd ~/snort_src
wget http://sourceforge.net/projects/secureideas/files/BASE/base-1.4.5/base-1.4.5.tar.
tar xzvf base-1.4.5.tar.gz
sudo mv base-1.4.5 /var/www/html/base/
```

Створення файлу конфігурації BASE:

```
cd /var/www/html/base
sudo cp base_conf.php.dist base_conf.php
```

Далі потрібно встановіти дозволи для папки BASE, оскільки пароль знаходиться в базовому файлі conf.php, тому ми повинні заборонити іншим користувачам читати його:

```
sudo chown -R www-data:www-data /var/www/html/base
sudo service apache2 restart
```

Останній крок для налаштування BASE виконується через http. Потрібно перейти за адресою у браузері http://ServerIP/base/index.php і клацніть посилання на сторінку налаштування (ServerIP це IP сервера Snort) після чого натиснувши Create BASE AG у верхньому правому куті сторінки переходимо на головну сторінку.

Результати роботи інструменту BASE наведені на рисунках 3.11-3.12.

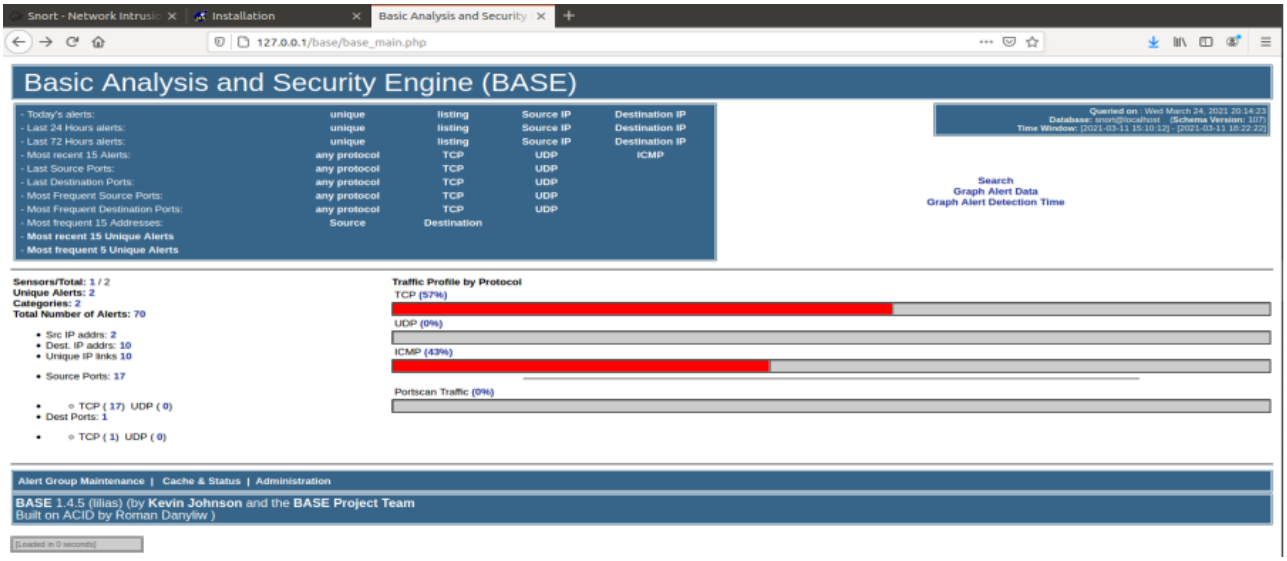


Рисунок 3.11 - Базовий аналіз та механізм безпеки 1.

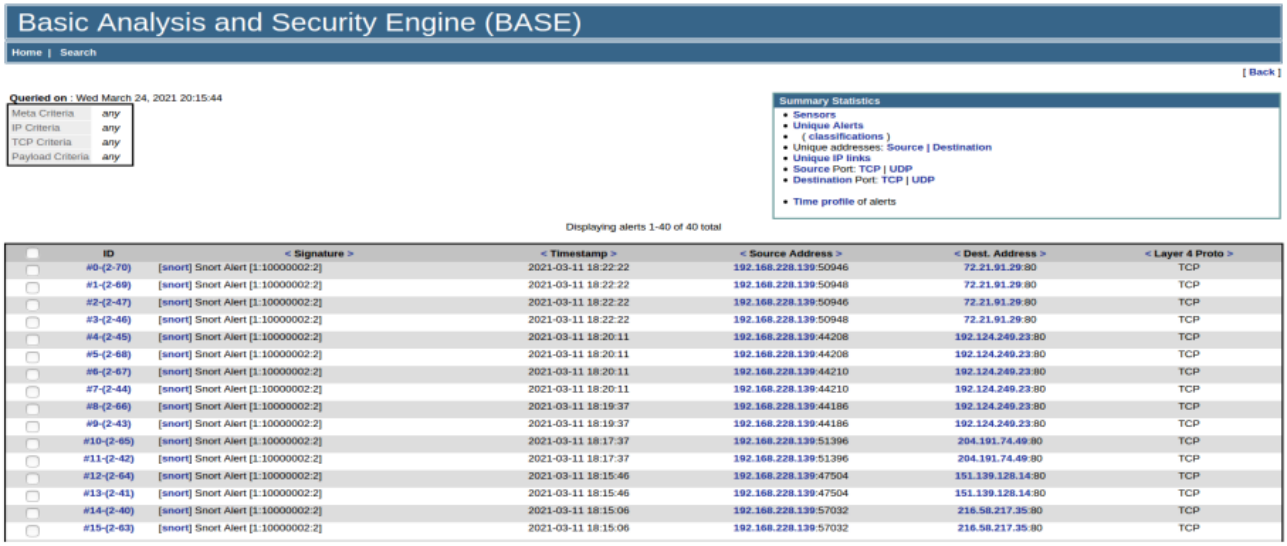


Рисунок 3.12 - Базовий аналіз та механізм безпеки 2.

Для керуваннями правилами Snort використовують PulledPork. PulledPork - це вміст Perl, який послідовно завантажує останні набори правил Snort. Особливості та можливості PulledPork:

1) автоматичне завантаження, аналіз, модифікація стану та модифікація правил для всіх ваших наборів правил snort;

2) перевірка контрольної суми для всіх основних завантажень правил;

3) автоматична генерація оновленого файлу sid-msg.map;

4) можливість включити ваші local.rules у файл sid-msg.map;

5) можливість отримувати архіви правил із власних URL-адрес;

6) повна підтримка Shared Object;

7) повна підтримка списку репутації IP;

8) можливість завантажувати кілька різних наборів правил одночасно;

9) веде точний журнал змін.

Для встановлення PulledPork потрібно спочатку встановити передумови:

```
sudo apt-get install -y libcrypt-ssleay-perl liblwp-useragent-determined-perl
```

Завантажте останню версію PulledPork і встановіть. Тут ми копіюємо фактичний файл perl до /usr/local/bin і необхідні файли конфігурації до /etc/snort :

```
cd ~/snort_src
wget https://github.com/shirkdog/pulledpork/archive/master.tar.gz -O pulledpork-master.tar.gz
tar xzvf pulledpork-master.tar.gz
cd pulledpork-master/
sudo cp pulledpork.pl /usr/local/bin
sudo chmod +x /usr/local/bin/pulledpork.pl
sudo cp etc/*.conf /etc/snort
```

Для перевірки чи працює PulledPork, потрібно виконати наведену нижче команду, де буде показано результат:

```
pavo@snortserver:~$ /usr/local/bin/pulledpork.pl -V
PulledPork v0.7.3 - Making signature updates great again!
user@snortserver:~$
```

Для змін у конфігураційному файлі PulledPork використовуємо файл pulledpork.conf, що міститься у /etc/snort/pulledpork.conf

Запускаємо PulledPork вручну, щоб переконатися, що він коректно працює:

```
https://github.com/shirkdog/pulledpork
_____ ____
`----,\    )
`--==\\  /    PulledPork v0.7.3 - Making signature updates great again!
`--==\\/
.-~~~~-.Y|\\_  Copyright (C) 2009-2016 JJ Cummings
@_/        /  66\_  cummingsj@gmail.com
|    \   \   _(")
\   /-| ||'--'  Rules give me wings!
\_\  \_\\
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

Після виконання цієї команди в каталозі /etc/snort/rules з'явиться файл snort.rules і .so- правила у /usr/local/lib/snort_dynamicrules.

PulledPork збирає всі набори правил, які він завантажує, у ці два файли. .so -правила, або Shared Object , або попередньо відкомпільовані правила скомпілювані постачальником правил для певних платформ і являють собою складніші правила, а також дозволяють обробити їх (наприклад, для виявлення атак, які ще не пропатчені, але повинні визначатися без розкриття самої суті вразливості).

Щоб Snort використовував завантажені правила, необхідно в його конфігурації /etc/snort/snort.conf вказати шлях до цих правил. Додаємо наступний рядок відразу за include $RULE_PATH/local.rules:

include $RULE_PATH/snort.rules

Після додавання шляху до нових правил конфігурації snort.conf ми повинні протестувати, що Snort коректно працює в режимі NIDS з включеним набором правил за допомогою PulledPork:

```
sudo snort -T -c /etc/snort/snort.conf -i eth0
```

Переконавшись у коректності конфігурації Snort, ми можемо перевірити, що Snort та Barnyard2 завантажуються коректно, запустивши їх у фоновому режимі:

```
sudo snort -u snort -g snort -c /etc/snort/snort.conf -i eth0 -D
sudo barnyard2 -c /etc/snort/barnyard2.conf -d /var/log/snort -f snort.u2 -w /var/log/snort/barnyard2.waldo -g snort -u snort -D
```

При перевірці коректної роботи Snort отримуємо результат, який зображений на рисунку 3.13.

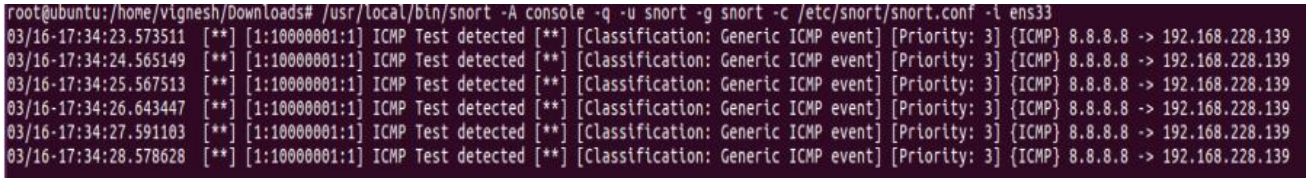


Рисунок 3.13. Перевірка роботи Snort

3.2. Реалізація системи

Для реалізації системи запобігання проникненню в мережі інтернет-речей для початку створимо правила для Snort. Список правил, налаштованих для запуску, показано на рисунках 3.14 та 3.15.

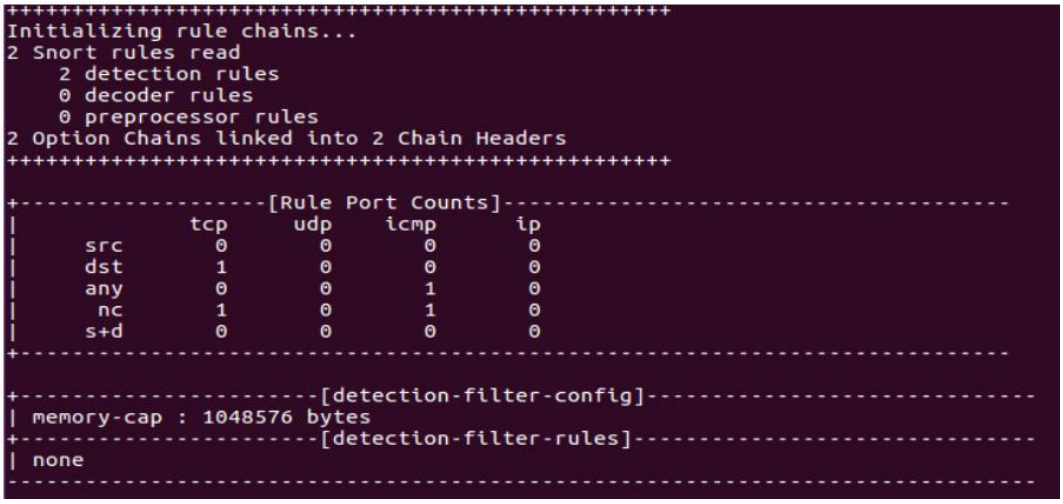


Рисунок 3.14 – Створення правил Snort

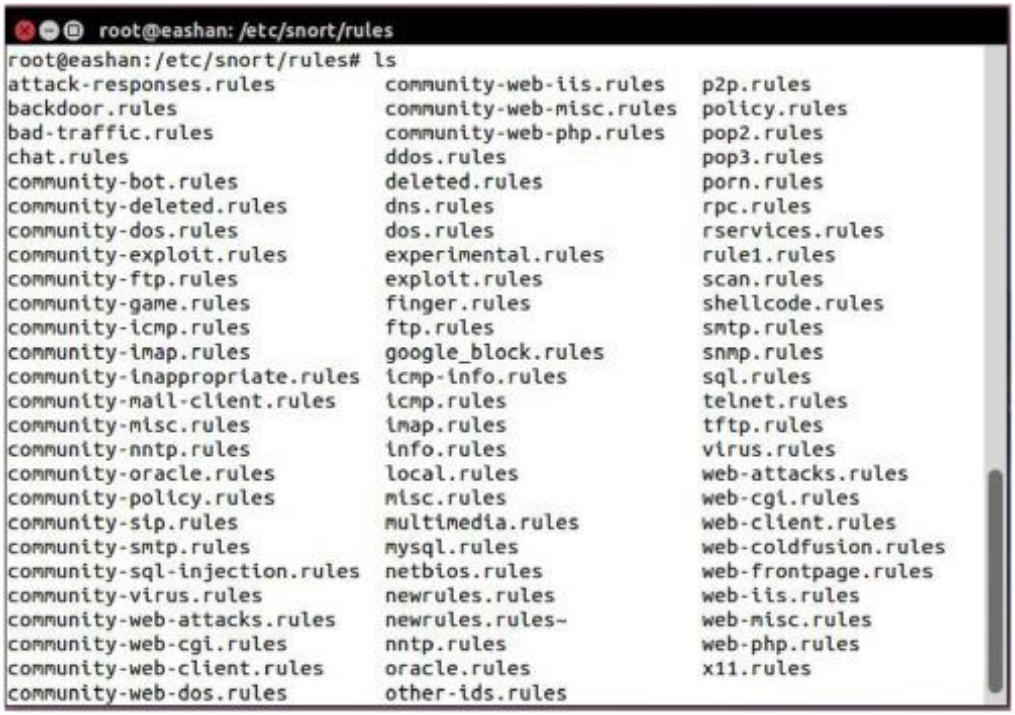


Рисунок 3.15 – Список правил

Команда SNORT для початку вбудованої обробки пакетів через мережевий міст, створений між мережевими інтерфейсами wlan0 і wlan1 (рисунок 3.16).

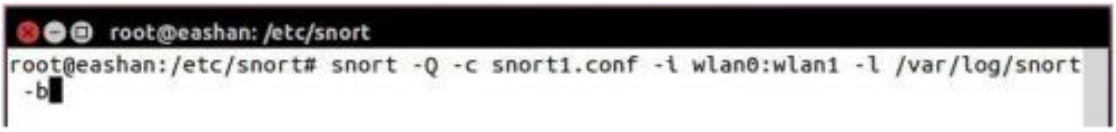


Рисунок 3.16 – Конфігурація Snort.

Виконання команди на кроці 2 дає результат, подібний до показаного на рисунку 3.17.

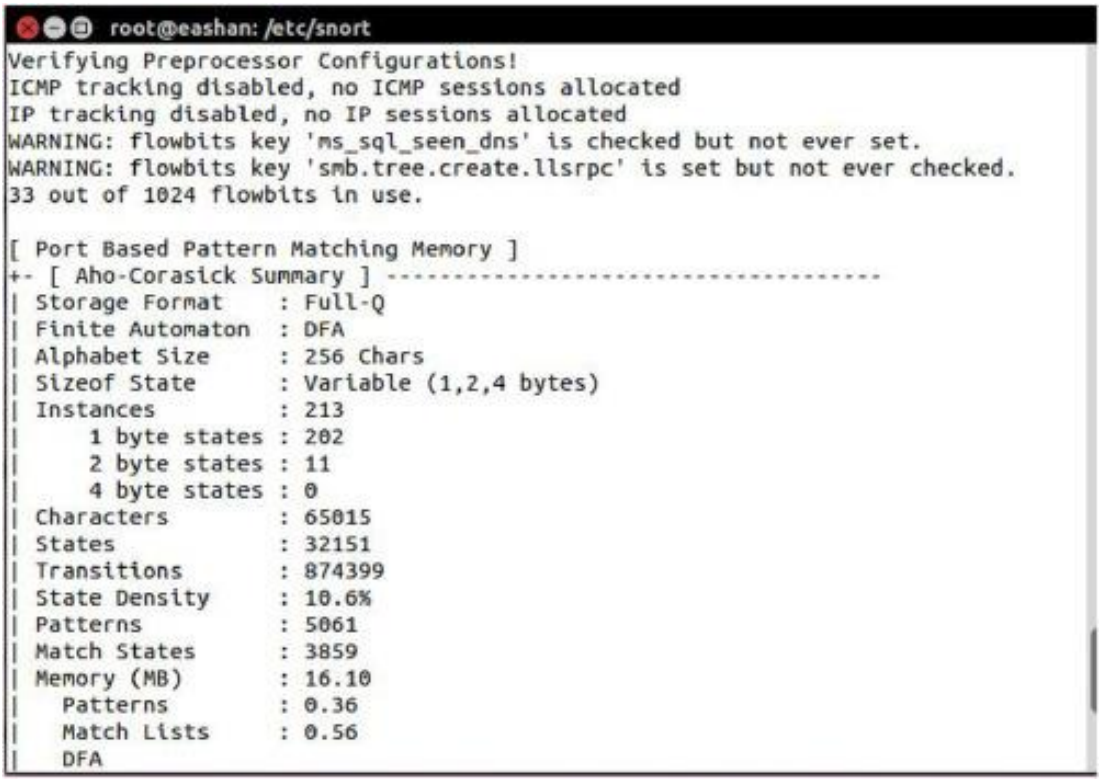


Рисунок 3.17 – Конфігурація Snort.

Обробку вбудованого пакета завершено. Підсумок аналізу сеансу надає SNORT (показано на рисунку 3.18).

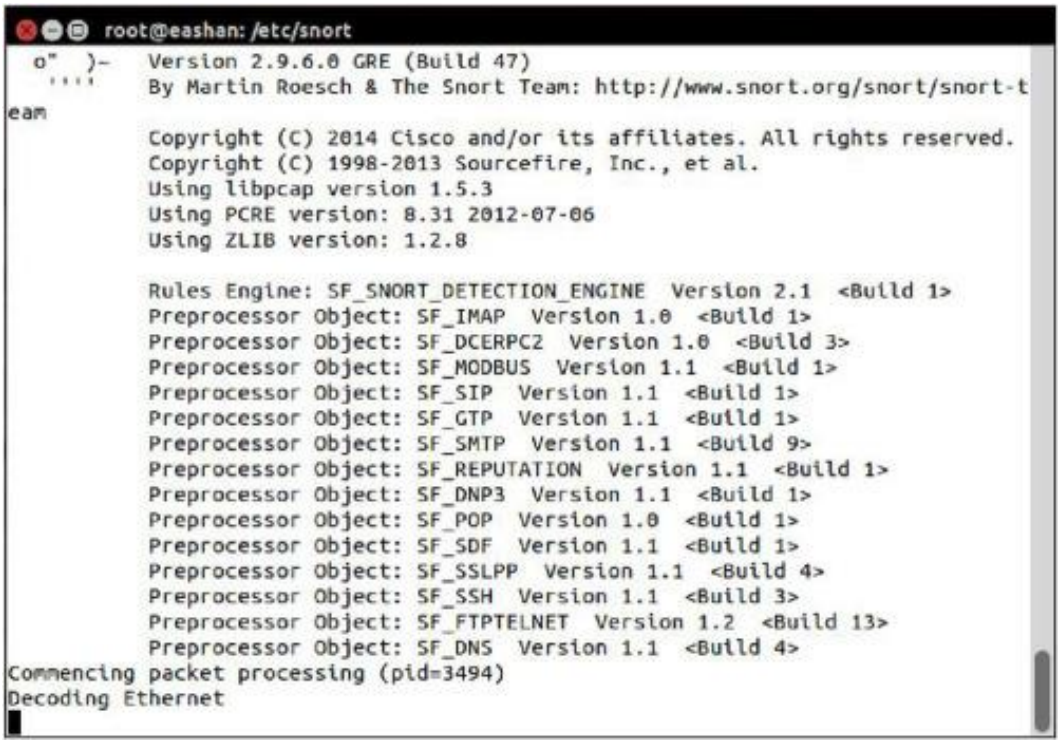


Рисунок 3.18 – Підсумок аналізу.

SNORT реєструє свої сповіщення та пакети в каталозі /var/log/snort, як зазначено на кроці 2 (рисунок 3.19).

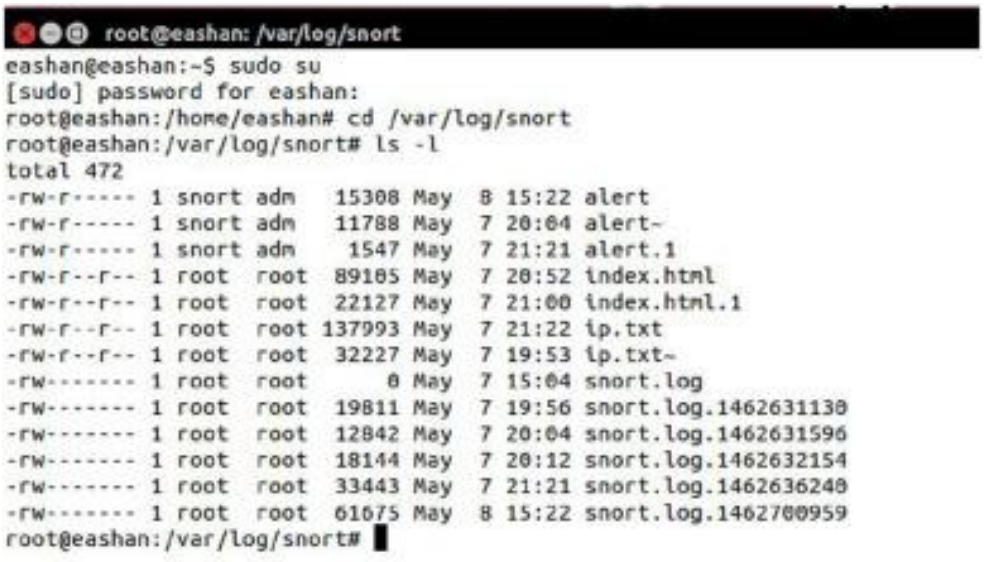


Рисунок 3.19 – Запис журналів.

Сповіщення, створені SNORT показані на рисунку 3.20.

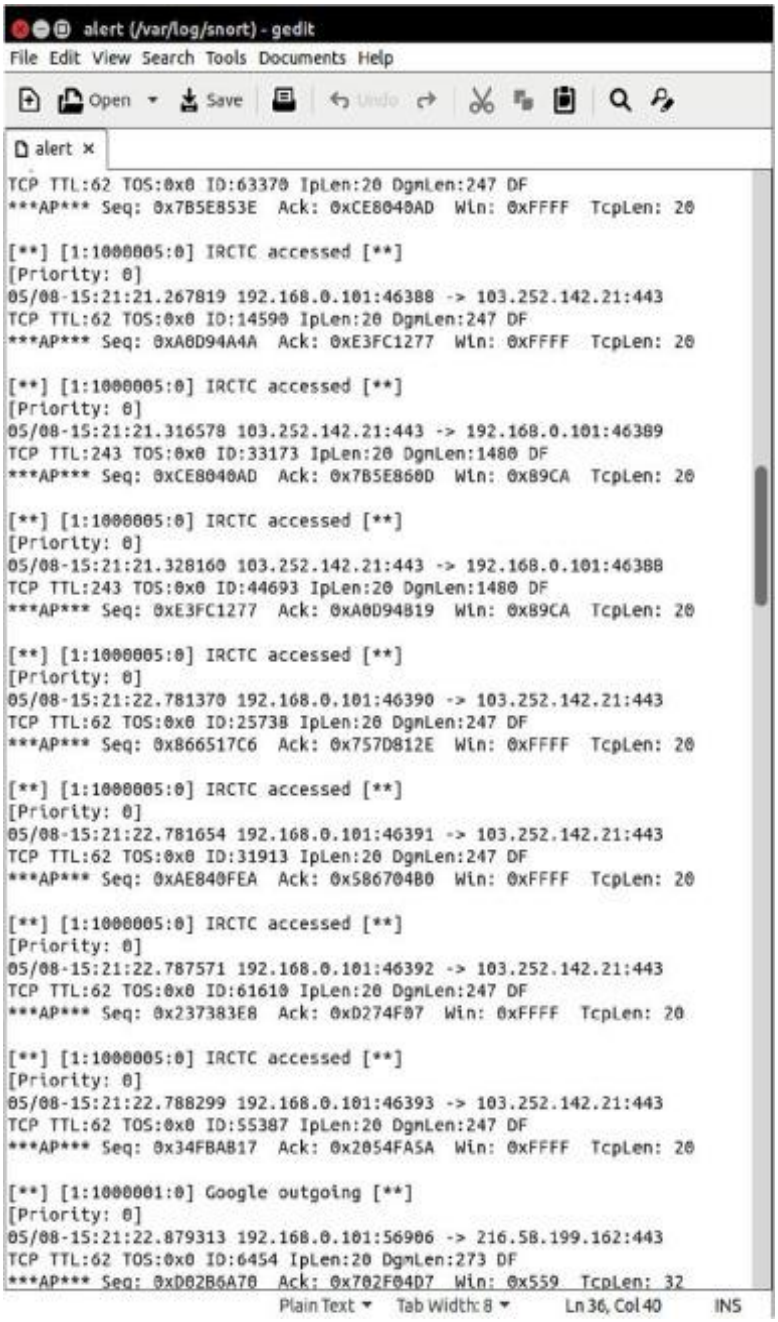


Рисунок 3.20 – Генерування сповіщень

Пакети, зареєстровані SNORT, показані на рисунку 3.21.

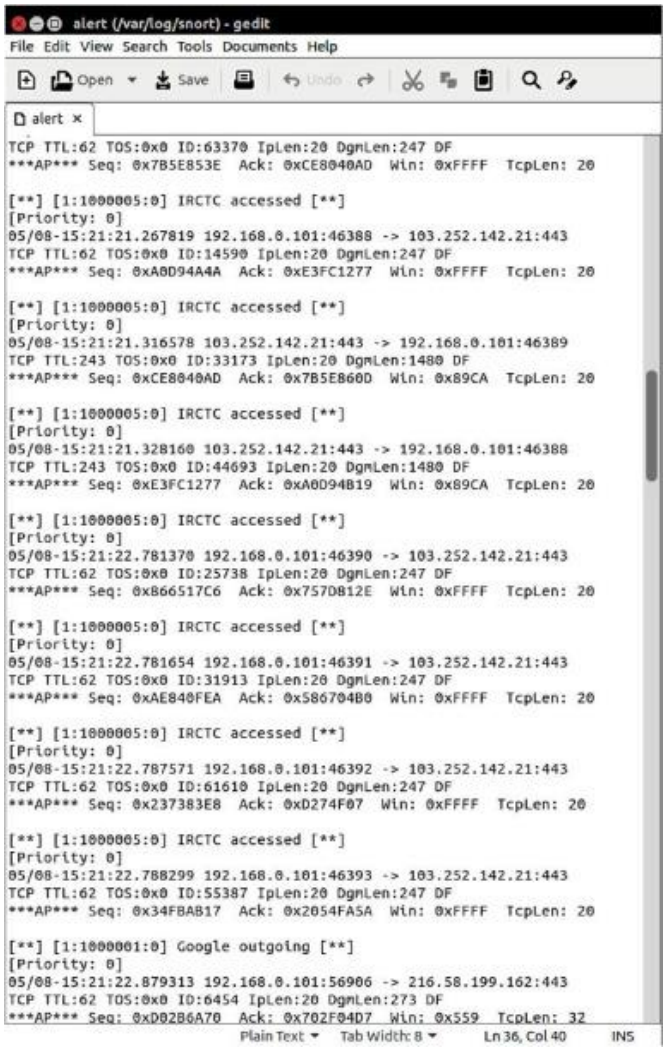


Рисунок 3.21 – Логування пакетів.

Додаткові правила визначено нами для запуску правил пасивного сповіщення (рисунок 3.22).

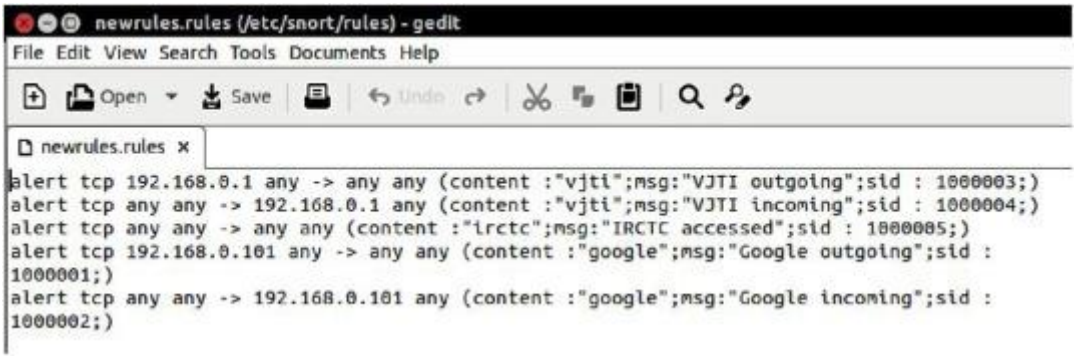


Рисунок 3.22 – Правила, які використовуються для запуску пасивного сповіщення.

3.3 Огляд результатів роботи системи

У цьому підрозділі ми робимо припущення для обчислення вартості передачі запропонованої системи. У таблиці 3.1 наведено передбачувану вартість компонентів у запропонованому протоколі. Для еліптичної кривої $E_p(a, b)$ усі параметри (p, a і b) вважаються 160-бітними. Отже, точка ECC $X, Y = (x_P, y_P) \in E_p(a, b)$ займає (80 + 80) = 160 біт. Для запропонованого протоколу параметрами зв'язку є X_i, Y_i та ID_i, X_j, Y_j та ID_j.

Таким чином, вартість зв'язку становить $(X_i, Y_i, ID_i) + (X_j, Y_j, ID_j)$ і (800 + 800) = 384 біти, що менше, ніж для існуючих системи, як зазначено в таблиці 3.2 .

Таблиця 3.1 - Таблиця вартості компонентів.

Компоненти	Розмір в бітах
ID_i, ID_j	32
(X_i, Y_i)	80
(Y_i, Y_i)	80
(P_i, P_i)	160
$2(X_i + Y_i + ID_i)$	(160+160+64) = 384
$(P_{x+} P_y)$	(160+160) = 320

Таблиця 3.2 - Порівняння вартості зв'язку.

Система	Кількість повідомлень	Кількість біт
M.M. Fouda, Z.M. Fadlullah	3	4448
T.W. Chim, S.-M. Yiu	3	3744
Запропонована система	2	384

Зробимо припущення щодо обчислення вартості зберігання запропонованої системи. У таблиці 3.3 представлено припущену вартість компонентів у запропонованому протоколі. У запропонованій системі інтелектуальний лічильник зберігає два виклики (тобто P_x, P_y) і споживає 20 + 20 = 40 байт у своєму згаданому сховищі. Це менше, ніж для існуючих систем, зазначених у таблиці 3.3.

Таблиця 3.3 - Порівняння вартості зберігання.

Система	Кількість повідомлень	Кількість біт
M.M. Fouda, Z.M. Fadlullah	3	3232
T.W. Chim, S.-M. Yiu	3	320
Запропонована система	2	320

Отже, дана система є швидшою і ефективною порівняно з її аналогами. Тому можемо сказати, що ця розробка є доцільною для використання і можливе її впровадження у різні корпоративні, домашні мережі.

ВИСНОВКИ

Системи виявлення та запобігання мережевим вторгненням NIDS / NIIPS мають перевагу перед іншими системами у тому, що вони не створюють навантаження на контрольовані хости. Вибір систем NIDS / NIPS підтверджується низькою кількістю помилкових спрацьовувань.

Новітні системи NIDS також дозволяють розпізнавати керовану операційну систему на основі характеристик певного стеку TCP/IP. Ця функція дозволяє вводити ієрархічні дані про важливості окремих тривог.

Впроваджена система NIDS / NIPS вимагає постійної адаптації до мінливих загроз і характеру мережевого трафіку. NIDS / NIPS може бути безпечним рішенням для захисту інформації для робочих зон і серверів.

У даній роботі:

1) розглянуто можливі проблеми у безпеці IoT.

2) показана загальна архітектура мережевої IDS.

3) описано алгоритм виявлення системою вторгнень.

4) наведено огляд основних методів виявлення, що використовуються в IDS.

5) цілі та завдання кваліфікаційної роботи.

6) аналіз побудови системи захисту мережі.

7) описано внутрішню архітектуру системи NIPS.

8) розроблено принципи налаштування інструментів безпеки.

9) наведено програмне та апаратне забезпечення, яке буде використовуватися під час виконання роботи;

10) проведене налаштування компонентів захисту мережі.

11) наведено аналіз систем виявлення та запобігання вторгнення в мережі IoT.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. M. Roesch. Snort: lightweight intrusion detection for networks. Lisa, 99 (1999), pp. 229-238

2. A. Saeed, A. Ahmadinia, A. Javed, H. Larijani. Intelligent intrusion detection in low-power iots. ACM Trans. Internet Technol., 16 (4) (2016), pp. 1-25

3. E. Benkhelifa, T. Welsh, W. Hamouda. A critical review of practices and challenges in intrusion detection systems for iot: toward universal and resilient systems. IEEE Commun. Surv. Tutorials, 20 (4) (2018), pp. 3496-3509

4. C. Kreibich, Network Intrusion Detection: Evasion, Traffic Normalization, and End-To-End Protocol Semantics.

5. Nourian A, Madnick S (2018) Системно-теоретичний підхід до загроз безпеці в кіберфізичних системах, застосований до Stuxnet. IEEE Transact Dependable Secure Comput 15(1):2–13.

6. Scarfone Karen. Guide to Intrusion Detection and Prevention Systems (IDPS) — 2007. — 127 p.

7. Mattord Verma. Principles of Information Security — 2008. — 300 p.

8. ДСТСЗІ СБ України. НД ТЗІ 2.5-010-03 «Вимоги до захисту інформації WEB-сторінки від несанкціонованого доступу» — 2003. — 16 с.

9. М. В. Грайворонський, О. М. Новіков. Безпека інформаційно-комунікаційних систем — 2009. — 608 с.

10. Syngress. Snort IDS and IPS Toolkit — 2007. — 197 p.

11. Корнієнко Б.Я. Дослідження імітаційного полігону захисту критичних інформаційних ресурсів методом IRISK. Моделювання та інформаційні технології. 2018. Вип. 83. С. 34-41.

12. Корнієнко Б.Я. Побудова та тестування імітаційного полігону захисту критичних інформаційних ресурсів. Наукоємні технології. 2017. № 4 (36). С. 316 - 322.

13. Korniyenko B., Yudin A., Galata L. Risk estimation of information system.Wschodnioeuropejskie Czasopismo Naukowe. 2016. № 5. P. 35 - 40.

14. Корнієнко Б.Я., Юдін О.К., Снігур О.С.Безпека аутентифікації у web-ресурсах. Захист інформації. 2012. № 1 (54). С. 20 -25. DOI: 10.18372/2410-7840.14.2056 (ukr).

15. Корнієнко Б.Я., Максімов Ю.О., Марутовська Н.М. Прикладні програми управління інформаційними ризиками. Захист інформації. 2012. № 4 (57). С. 60 – 64. DOI: 10.18372/2410-7840.14.3493 (ukr).

16. Galata, L., Korniyenko, B., Yudin, A.: Research of the simulation polygon for the protection of critical information resources. In: CEUR Workshop Proceedings, Information Technologies and Security, Selected Papers of the XVII International Scientific and Practical Conference on Information Technologies and Security (ITS 2017), 30 Nov 2017, Kyiv, Ukraine. vol. 2067. pp. 23–31., urn:nbn:de:0074-2067- 8. 101

17. Корнієнко Б.Я. Безпека інформаційно-комунікаційних систем та мереж. Навчальний посібник для студентів спеціальності 125 «Кібербезпека». – К.:НАУ, 2018. – 226 с.

18. Корнієнко Б.Я., Галата Л.П. Оптимізація системи захисту інформації корпоративної мережі. Математичне та комп'ютерне моделювання. Серія: Технічні науки, Випуск 19, 2019. - С. 56-62.

19. Korniyenko B., Galata L. Implementation of the information resources protection based on the CentOS operating system. Conference Proceedings of 2019 IEEE 2nd Ukraine Conference on Electrical and Computer Engineering (UKRCON - 2019) July 2 – 6, 2019, Lviv, Ukraine. - pp. 1007-1011.

20. Галата Л.П., Корнієнко Б.Я., Заболотний В.В. Математична модель протидії загрозам у системі захисту критичних інформаційних ресурсів. Наукоємні технології, Том 43, № 3, 2019. – С. 300 – 306.

21. Корнієнко Б.Я. Modeling of information security system in computer network. Безпека інформаційних систем і технологій, Том №1 (1), 2019. – С.36-41.

22. Korniyenko B., Galata L., Ladieva L. Research of Information Protection System of Corporate Network Based on GNS3. Conference Proceedings of 2019 IEEE Inernational Conference on Advanced Trends in Information Theory (IEEE ATIT - 2019) Dezember 18 – 20, 2019, Kyiv, Ukraine. - pp. 244-248.

23. Korniyenko B., Galata L., Ladieva L. Mathematical model of threats resistance in the critical information resources protection system. CEUR Workshop Proceedings, Selected Papers of the XIX International Scientific and Practical Conference "Information Technologies and Security" (ITS 2019) Kyiv, Ukraine, November 28, 2019. Vol-2577. P.281-291.

24. Корнієнко Б.Я. Кібернетична безпека – операційні системи і протоколи. ISBN 978-3-330-08397-4, LAMBERT Academic Publishing, Saarbrucken, Deutschland. − 2017. – 122 с.

25. Korniyenko B.Y., Galata L.P. Design and research of mathematical model for information security system in computer network. Науковий журнал «Наукові технології». – 2017, № 2 (34), С. 114 - 118. 102

26. Корнієнко Б.Я. Інформаційна безпека та технології комп'ютерних мереж: монографія. ISBN 978-3-330-02028-3, LAMBERT Academic Publishing, Saarbrucken, Deutschland. − 2016. – 102 с.

27. Korniyenko B., Galata L., Kozuberda O. Modeling of security and risk assessment in information and communication system. Sciences of Europe. – 2016. – V. 2. – No 2 (2). – P. 61 -63.

28. Korniyenko B. The classification of information technologies and control systems. International scientific journal. – 2016. –№ 2. – P. 78 - 81.

29. Корнієнко Б.Я. Інформаційні технології оптимального управління виробництвом мінеральних добрив :монографія. – К.: Вид-во Аграр Медіа Груп, 2014. – 288 с.

30. Korniyenko B., Galata L., Ladieva L. Security Estimation of the Simulation Polygon for the Protection of Critical Information Resources / B. Korniyenko, //CEUR Workshop Proceedings, Selected Papers of the XVIII International Scientific

and Practical Conference "Information Technologies and Security" (ITS 2018) Kyiv, Ukraine, November 27, 2018, Vol-2318, - P. 176-187, urn:nbn:de:0074-2318-4.

31. Миколишин П.П. Система запобігання проникненню в мережі інтернет-речей. Збірник матеріалів проблемної наукової міжгалузевої конференції «Автоматизація та комп’ютерно-інтегровані технології» (АКІТ-2022). Тернопіль, 2022. С.91-93.

32. Миколишин П.П. Система запобігання проникненню в мережі Інтернет-речей на основі аномалій. Збірник матеріалів проблемної наукової міжгалузевої конференції «Кібербезпека та комп’ютерно-інтегровані технології» (КБКІТ-2022). Тернопіль, 2022. С.63-64.