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

Гавриляк Михайло Васильович

Система виявлення мережевих вторгнень на основі Snort / Snort-based network intrusion detection system

спеціальність: 125 – Кібербезпека

освітньо-професійна програма –Кібербезпека

Кваліфікаційна робота

Виконав студент групи КБзм -21
М.В. Гавриляк

Науковий керівник
к.т.н., доцент Н.Г. Яцків

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
____" _____ 2021 року

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

ГАВРИЛЯК Михайло Васильович
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи:
Система виявлення мережевих вторгнень на основі Snort / Snort -based network intrusion detection system
керівник роботи к.т.н., доцент Н.Г. Яцків

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 відомих систем виявлення вторгнень;
– дослідити переваги та недоліки інструменту Snort;
– дослідити написання правил Snort;
– дослідити структуру інструменту Snort;
– встановити та налаштувати інструмент Snort на операційній системі Windows;
– провести тестування інструменту Snort.
5. Перелік графічного матеріалу у роботі.
Процес системи виявлення вторгнень.
Типи системи виявлення вторгнень
Архітектура Snort.
Архітектура запропонованої системи.

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 систем виявлення мережевих вторгнень	12.2021 р. – 03.2022 р.	
2	Структура системи та правила виявлення вторгнень на базі Snort	03.2022 р. – 05.2022 р.	
3	Дослідження та реалізація системи виявлення вторгнень	05.2022 р. – 11.2022 р.	

Студент _____ Гавриляк М.В.
(підпис)

Керівник роботи _____ к.т.н., доцент Н.Г. Яцків
(підпис)

АНОТАЦІЯ

Кваліфікаційна робота на тему «Система виявлення мережевих вторгнень на основі Snort» на здобуття освітнього ступеня «Магістр» зі спеціальності 125 «Кібербезпека» освітньо-професійної програми «Кібербезпека» написана обсягом 65 сторінок і містить 45 ілюстрацій, 5 таблиць, 2 додатки та 21 джерел за переліком посилань.

Метою кваліфікаційної роботи є дослідження системи виявлення вторгнень Snort, її тестування.

Методи досліджень. Для розв’язання поставлених задач у даній кваліфікаційній роботі використано: методи виявлення вторгнень, правила Snort.

Результати дослідження: встановлено та налаштовано інструмент Snort на операційній системі Windows.

Протестовано систему виявлення вторгнень Snort за допомогою командної стрічки Windows.

Результати тестування можуть успішно застосовуватися для виявлення потенційних загроз.

Ключові слова: СИСТЕМИ ВИЯВЛЕННЯ ВТОРГНЕНЬ, ІНСТРУМЕНТ SNORT, СТРУКТУРА СИСТЕМИ SNORT, ТЕСТУВАННЯ SNORT.

ABSTRACT

Qualification work on "Snort-based network intrusion detection system" for the degree of "Master" in the specialty 125 "Cybersecurity" educational and professional program "Cybersecurity" is written in 65 pages and contains 45 illustrations, 5 tables, 2 appendices and 21 source according to the list of links.

The purpose of the qualification work is to study the Snort intrusion detection system and test it.

Research methods. To solve the tasks in this qualification work, the following tools were used: intrusion detection methods, Snort rules.

Research results: Installed and configured Snort tool on Windows operating system.

Tested the Snort intrusion detection system twice using the Windows command line.

Test results can be successfully used to identify potential threats.

Keywords: INTRUSION DETECTION SYSTEMS, SNORT TOOL, SNORT SYSTEM STRUCTURE, SNORT TESTING.

ЗМІСТ

ВСТУП

Системи виявлення вторгнень відіграють важливу роль у захисті безпеки комп'ютерних систем та Інтернету, адже це активний засіб захисту. Система виявлення вторгнень в режимі реального часу може виявляти стан мережі, контролювати потік мережі та діяльності, а також видавати попередження, записувати інформацію до бази даних, на основі якої аналізує вторгнення та створює журнали вторгнень.

Систему виявлення вторгнень у режимі реального часу можна встановити на будь-якому вузлі мережі, вибравши різні місця. Їх можна адаптувати до різних мереж споруди, які можуть формувати тривимірну глибину системи захисту.

Зараз багато компаній встановлюють системи виявлення вторгнень у точках доступу внутрішньої або загальнодоступної мережі, наприклад точки доступу мобільного шлюзу, корпоративного комутатора.

В даній магістерській роботі розглянемо систему виявлення вторгнень Snort та порівняємо, чим snort відрізняється від інших систем виявлення вторгнень.

Мета і завдання дослідження. Метою кваліфікаційної роботи є дослідження системи виявлення вторгнень Snort, її тестування. Досягнення визначеної мети передбачає вирішення таких завдань:

– провести аналіз відомих систем виявлення вторгнень;

– дослідити переваги та недоліки інструменту Snort;

– дослідити написання правил Snort;

– дослідити структуру інструменту Snort;

– встановити та налаштувати інструмент Snort на операційній системі Windows;

– провести тестування інструменту Snort.

Об'єкт дослідження – процес виявлення вторгнень в комп'ютерні системи на базі Snort та операційної системи Windows.

Предмет дослідження – методи та алгоритми виявлення вторгнень на базі системи Snort.

Методи досліджень. Для розв'язання поставлених задач у даній кваліфікаційній роботі використано: методи виявлення вторгнень, правила Snort.

Наукова новизна одержаних результатів. Розроблено структуру системи виявлення вторгнень на базі інструменту Snort з використанням правил Snort.

Практичне значення отриманих результатів. Проведено тестування системи виявлення вторгнень на операційній системі Windows ОС.

Публікації та апробація КР.

1. Гавриляк М.В., Цаволик Т.Г., Інгнатьєв І.В. Функції та переваги системи виявлення вторгнень на базі SNORT Матеріали наукової конференції «Автоматизація та комп'ютерно-інтегровані технології» (АКІТ – 2022), Тернопіль, 2022. – С.97– 99.

2. Гавриляк М.В. Виявлення вторгнень на основі правил SNORT. Матеріали наукової конференції «Кібербезпека та комп'ютерно-інтегровані технології» (КБКІТ – 2022), Тернопіль, 2022. – С. 70–72.

1 АНАЛІЗ СИСТЕМ ВИЯВЛЕННЯ МЕРЕЖЕВИХ ВТОРГНЕНЬ

1.1 Класифікація системи виявлення вторгнень

Система виявлення вторгнень (IDS) – це система, яка намагається виконати виявлення вторгнень шляхом порівняння поведінки проти підозрілих моделей. Метою виявлення вторгнень є моніторинг мережевих ресурсів, а також виявлення ненормальної та нерегулярної поведінки та зловживань. Ця концепція існує протягом останніх кількох років, але лише нещодавно вона викликала різке зростання інтересу дослідників і розробників систем до включення в загальну систему інфраструктури інформаційної безпеки.

Система виявлення вторгнень перевіряє мережевий трафік для будь-якої підозрілої та нерегулярної діяльності і сповіщає системного або мережевого адміністратора , а в деяких випадках IDS також може протидіяти аномальним або зловмисного трафіку шляхом виконання таких дій, як блокування або ізоляції IP-адреси користувача або джерела від доступу мережі (рис.1.1).

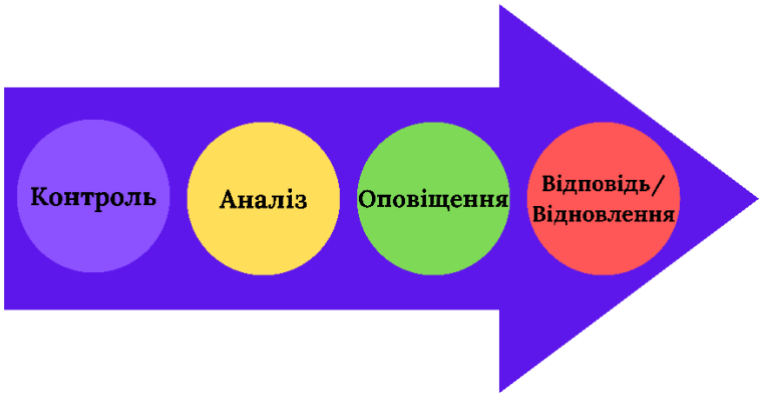


Рисунок 1.1 – Процес системи виявлення вторгнень

На рисунку 1.1 показано основний процес системи виявлення вторгнень, який здійснює моніторинг, аналіз, оповіщення та реагування на виявлений дефект. IDS створюються за допомогою програмного забезпечення, яке оцінює безпеку мережі за допомогою моніторингу мережевої діяльності. Програмне забезпечення дозволяє мережевому контролеру перевіряти мережу

вразливостей, таким чином, захищаючи потенційні лазівки перш, ніж зловмисники скористаються ними. IDS має різні типи, такі як IDS на основі мережі (NIDS), IDS на основі хоста (HIDS), IDS на основі аномалії (AIDS) і системи виявлення вторгнень мережевого вузла (NNIDS). Загалом розглядають шість типів класифікації системи виявлення вторгнень, їх можна переглянути на рисунку 1.2 [1].

HIDS	Host Based (IDS) IDS на основі хоста
NIDS	Network based IDS IDS на основі мережі
HDS	Hybryd IDS Гібридний IDS
NNIDS	Network node intrusion detection Виявлення вторгнень мережевого вузла
AIDS	Anomaly based IDS Ідентифікатори на основі аномалій
MDS	Misuse detection system Система виявлення зловживань

Рисунок 1.2 – Типи системи виявлення вторгнень

Система виявлення вторгнень на основі хоста (HIDS). Система виявлення вторгнень на основі хоста призначена для моніторингу, виявлення та реагування на систему користувача, активність і атаки на даний хост. Деякі надійні засоби пропонують централізоване управління політикою аудиту, постачання аналізу даних, статистичний аналіз і підтримку, а також надати певну міру доступу контролю. Виявлення вторгнень на основі хосту найкраще підходить для боротьби з внутрішніми загрозами та ненормальною поведінкою в локальних мережах, завдяки своїй здатності контролювати та реагувати на певні дії користувача та доступ до файлів на хост. Більша частина

комп'ютерних загроз походить завдяки побоюванням. IDS на основі хосту покладається на єдину систему і деталей журналу аудиту, які зберігаються на кожній машині. Якщо зловмисник захоплює систему, тоді він може підробляти двійкові файли IDS і змінювати журнали аудиту [2].

1.2 Мережеві система виявлення вторгнень

Система виявлення вторгнень на основі мережі (NIDS). Виявлення мережевого вторгнення має справу з інформацією, що передається по дроту між хостами, які зазвичай називають «аналізаторами пакетів». Мережа пристроїв IDS перехоплює пакети, що переміщуються різними комунікаційними середовищами та протоколами. Зазвичай використовується протокол TCP/IP. В результаті, він захоплює пакети та аналізує в декількох підходах. Кілька мереж пристроїв виявлення вторгнень просто порівнюють пакет до бази даних підписів. Це допомагає перевіряти, чи містяться будь-які відомі атаки та шкідливі пакети чи ні. Він також перевіряє пакет і його активність, оскільки це може свідчити про зловмисну поведінку вказаної угоди. У будь-якому випадку слід враховувати NIDS як обмежену. NIDS історично був нездатний працювати в наступних середовищах, таких як:

1. Комутовані мережі.
2. Зашифровані мережі.
3. Високошвидкісні мережі (все, що перевищує 100 Мбіт/с).

Різниця між хостом і виявленням вторгнень на основі мережі – це мережа Виявлення вторгнення (NIDS) працює з переданими даними від хоста до хоста, але стосується ідентифікатора на основі хоста, який є на цих хостах.

Гібридна система виявлення вторгнень (HIDS). Гібридні системи виявлення вторгнень полегшують управління та оповіщення з обох мереж і пристроїв виявлення вторгнень на основі хоста. Гібридні рішення забезпечують логічне доповнення до NIDS і HIDS - централізоване керування

виявленням вторгнень. Нещодавно Cisco випустила модуль для свого комутатора Catalyst 6000, який включає безпосередньо виявлення вторгнень у мережу перемикача, подолавши перший із цих недоліків. Крім того, в мережі ISS (Internet Security System) вказано, що тепер вони здатні "перехоплювати пакети" на гігабітних швидкостях, що є значним досягненням [3].

Виявлення вторгнень мережевого вузла (NNIDS). Виявлення вторгнення в мережевий вузол (NNIDS) розроблено для усунення внутрішніх дефектів традиційних ідентифікаторів мережі. Мережевий вузол витягує пакет, перехоплює технологію з дроту та вмикає її хост. За допомогою NNIDS "аналізатор пакетів" позиціонується таким, що захоплює пакети після досягнення ними цільової або кінцевої системи. Отриманий пакет у пункті призначення потім аналізується так, ніби він подорожував по мережі через звичайний «пакетний сніфер». Ця схема прийшла від HID (стандарт підключення через універсальну послідовну шину комп'ютерних пристроїв вводу-виводу, які призначені для прямої взаємодії з людиною) припущення, що кожен хост буде вже користується перевагами технології на основі хоста. У цій схемі мережевий вузол є просто іншим компонентом, який може підключатися до HID. Головним недоліком є те, що він оцінює лише адресовані пакети хосту, які на ньому існують. Традиційне мережеве виявлення вторгнень, з іншого боку, відстежує пакети на всю підмережу. У цьому випадку «Пакет-сніфери» нездатні переглянути всю підмережу, коли мережа використовує високошвидкісний зв'язок, комутатори або шифрування. Перевагою NNIDS є здатність захищати певні хости на основі пакетів безпеки та вирішувати проблеми в цих складних середовищах. Це дуже ефективно працює в тих середовищах, де звичайний NIDS є неефективним.

Ідентифікатори на основі аномалій (AIDS). Системи виявлення аномалій спостерігають за діяльністю, яка характеризує деякі зміни в звичайних користувацьких профілів як можливі вторгнення. Наприклад, звичайний профіль користувача може містити усереднені частоти деяких використовуваних системних команд під час сеансів входу. У випадку, коли

частоти відрізняються, системи фіксують це як загрозу. Це відбувається завдяки процесу постійного моніторингу. Ключова перевага виявлення аномалії полягає в тому, що для цього немає необхідності знати попередню інформацію або дані про вторгнення, таким чином ця система може виявляти нові загрози без цих даних [4].

Система виявлення зловживань (MDS). Системи виявлення зловживань використовують шаблони добре відомих атак або слабких місць для виявлення вторгнень. Ця система знаходить і ідентифікує відомі вторгнення за допомогою набору візерунків. Наприклад, якщо користувач не зміг увійти більше чотирьох спроб протягом певного часу, тоді цей процес буде оголошено, як атаку підбору пароля. Це можна виявити за допомогою підпису «якщо». Головний недоліком є відсутність здатності виявляти невідомі атаки. Концепція виявлення неправильного використання схем полягає в тому, що існують способи представлення атак у форму візерунка або підпису, для того щоб навіть при варіації однієї і тієї ж атаки можна виявити її. Це означає що ці системи схожі на системи виявлення вірусів. Вони можуть виявити багато відомих моделей атак, а інколи навіть всі, проте є слабкими проти невідомих атак. Цікавим моментом є те, що в той час як системи виявлення аномалій намагаються виявити погану поведінку, системи виявлення зловживань намагаються розпізнати погану поведінку за допомогою наведених шаблонів. Головна дилема в системах виявлення зловживань полягає в тому, як написати шаблон, який охоплює всі можливі варіанти відповідної атаки та як писати шаблони, які не відповідають ненав'язливій діяльності [5].

1.3Аналіз виявлення вторгнень з відкритим кодом

Snort. Snort – це програмне забезпечення з відкритим кодом і невеликим об’ємом програмне забезпечення, розроблене Мартіном Решем у 1998 році для виявлення та запобігання вторгнень у мережу з відкритим кодом системи.

Програмне забезпечення Snort діє як аналізатор пакетів, реєстратор пакетів або як засіб виявлення вторгнень у мережу та система профілактики [NIDS]. Snort читає мережеві пакети та відображає їх на консолі. Якщо консоль є у режимі аналізатора, він одночасно реєструватиме пакети на диск вибору реєстратора пакетів. Також, він буде стежити за мережею трафіку і аналізом трафіку відповідно до визначеного набору правил користувачем, під час виявлення вторгнення в мережу і системи профілактики. Це відноситься до мережі IDS. Також, його було розроблено для Linux і Windows, щоб виявляти нові загрози. Snort має можливість робити аналіз одночасного трафіку та реєстрації пакетів мережі Інтернет-протоколу (IP). Це також може стосуватись аналізу протоколів, пошуку вмісту та зіставлення. Snort використовує виявлення вторгнення на основі сигнатур, а також методи, засновані на аномаліях, на які можна покластися при налаштуванні правил користувача або підписів, які отримані з баз даних, таких як Emerging Threats [6].

Правила Snort. Заголовок складається з певних інструкцій, до яких відносяться: журнал або сповіщення; тип мережі пакет: tcp/udp/icmp/i т.д., IP джерела, призначення адреси та порти; сповіщення та, можливо, деякі кваліфікатори, сигнатури та тип класифікації (таблиця 1.1).

Таблиця 1.1 – Зразок правил Snort та його частин

Дія	Сповіщення
Протокол	TCP
Джерело і порт	Google/ any
Пункт призначення і порт	$Home_Net/ any
Попередження	“Msg Info: Psybnc Anomaly Access”
Кваліфікатор	Flow:From_Server, Established
Підпис	Content:”Welcome!Psybnc!”
Тип класифікації	Bad: Unknown

Коли Snort виявляє пакет, який відповідає неправильному значенню підпису, він збирає правила від користувача та виконує дію, указану в списку налаштованих користувачем правил. Загалом є дві можливі та типові дії наприклад оповіщення та пропуск (ігнорування).

В Таблиця 1.1 показано зразок правил Snort. Дія «попередження» створює дію, яка може зберігати журнал або надсилати його електронною поштою, а також може створити сигнал для машини Windows. Дія «пропуск» припиняє обробку пакетів, якщо підпис збігається в пакеті [7].

Snort інструменти дуже прості в установці і запуску, оскільки користувач може налаштувати правила. Він створює спливаючі вікна та оповіщення.

Ці спливаючі вікна контролюються Служба Windows Messenger. Snort має кілька вищенаведених переваг. Однак він має менше графічного інтерфейсу та пакетів, тому процес захоплення дуже повільний, і має лише середній рівень підтримки високошвидкісної мережі. Враховуючи вищезазначене, управління Snort є легким і підходить для всіх тип додатків.

SURICATA. Suricata – це швидке та надійне мережеве виявлення механізмів вторгнення. Вона здатна виявляти вторгнення у режимі реального часу. Як і Snort, Suricate також створена на основі методології підписів, керованими правилами/політиками безпеки та підходу до виявлення аномалій. Suricata перевіряє мережевий трафік використовуючи потужні та широкі правила, мовний підпис та має потужну підтримку сценаріїв мови програмування Lua для виявлення комплексних загроз. Suricata працює за правилом фільтрування сповіщень і порогового значення, глобального фільтрування сповіщень і порогового значення, а також порогового значення та швидкості обмеження параметрів на хост/підмережу.

Suricata має такі деталі у створенні правил процесів: параметри дії, заголовка та правила. Частина дії у Suricata містить пропуск, скидання, відхилення, і попередження. Якщо підпис збігається і містить «Пропуск», Suricata припиняє сканувати пакет і переходить до кінця всіх правил. Це

виконується лише для поточного пакета. Якщо дія вказана як «Drop», то лише це стосується режиму IPS/inline. Якщо програма знайде відповідний підпис, містить «Drop», він негайно зупиняється. Пакет більше не надсилатиметься. Якщо міститься ключове слово, як-от “Reject”, тоді одержувач не отримує повідомлення про те, що відбувається увімкнення, що призводить до тайм-ауту (з TCP). Цей інструмент генерує сповіщення для пакета. Якщо «Відхилити» вибрано ключове слово, воно відхиляє пакет. Обидва приймача і відправник отримують відхилений пакет. Є два типи відхилених пакетів, які будуть автоматично вибрані в цей випадок. Якщо неправильний пакет пов’язаний із TCP протоколом, це буде пакет Reset. Для всіх інших протоколів це буде пакет помилок ICMP [8].

Також генеруються сповіщення, коли ввімкнено режим Inline/IPS; неправильний пакет також буде відкинуто, як і з дія «скидання» в цьому процесі. Якщо підпис збігається і містить «Alert», пакет розглядатиметься, як будь-який інший пакет без загрози. Крім цього попередження буде створено інструментом Suricata. Тільки системний адміністратор може помітити це сповіщення.

Ще одна особливість правила SURICATA напрямок. Цей напрямок вказує відповідність підпису. Майже кожен підпис має стрілку вправо. Це означає, що лише пакети з однаковим напрямком можуть збігатися.

Таблиця 1.2 – Зразок правил SURICATA та його частин.

Правила параметрів	Зразок
Дія	Pass, Drop, alert and Reject
Напрямок	tcp $Home_PC any $external_PC any
Параметри правила	Meta-information, headers, payloads and flows

Таблиця 1.2 показує приклад правила Suricata з різними параметрами. Це включає три основні параметри: дія, напрямок і параметри правила. Як і snort, правила Suricata вказуються чотирма різними діями. Головною перевагою Suricata є можливість користувача налаштувати та використовувати ті самі набори правил Snort. Це має розширені функції, такі як багатопотокові можливості і прискорення GPU, але й створює більше помилкових тривог на момент виявлення. Система і мережа використання ресурсів є вичерпними у Suricata [9].

Bro IDS. Bro IDS – це виявлення вторгнень на основі аномалій, зазвичай використовується в поєднанні з Snort, так як вони добре доповнюють один одного. Цікаво, що Bro фактично є доменно-спеціальною мовою для мережевих програм, у яких Bro IDS написано. Технологія особливо ефективна при аналізі трафіку, і часто використовується в криміналістиці та пов'язаних випадків використання. Цей механізм політики має власну мову. Bro IDS складається з таких основних компонентів, як libpcap, Event Engine і Policy (інтерпретатор сценаріїв). Bro захоплення пакетів з мережевих інтерфейсів за допомогою бібліотеки libpcap приймає весь трафік, який надходить з мережевого рівня і відфільтровує неважливі елементи. Відфільтрований потік пакетів пересилається до механізму подій. Отримані пакети об'єднуються для отримання необхідної дії механізму подій. Сценарій політики інтерпретатор, зіставляє пакети з правилами. Це виявляє підозрілі та небезпечні дії та відкидає невідповідні пакети. Bro має зовсім інший підхід порівняно з іншими інструментами. Правила в Bro працюють зі скриптами. Це керований сценарій IDS. Він може підтримувати високу пропускну здатність середовищ. Час обробки менший у порівнянні з SNORT і SURICATA. Bro IDS розподіляє навантаження на кілька серверів. Цю платформу можна налаштувати для різноманітності безпеки мережі на додаток до NIDS, який може виконувати дуже важкі та різноманітні завдання.

Також цей інструмент може виявити шаблони діяльності, які інші системи IDS не можуть виявити, але це важко інтерпретувати та налаштувати. Потреба користувача - більше навичок програмування для виконання правил. Програмне забезпечення Bro більш адаптивне, що надає можливості для різних сценаріїв політики моніторингу конкретного сайту. Це забезпечує високу продуктивність і гнучкість. Також Bro мова справді пропонує набагато ширший спектр, дуже різні підходи до виявлення шкідливих дій, включаючи виявлення неправильного використання, аномалію виявлення та аналіз поведінки [10].

Open WIPS-ng. OpenWIPS-ng є модульним і відкритим кодом бездротової системи запобігання вторгнення. Вона захоплює бездротовий трафік, виявляє та ідентифікує стандартні і приховані мережі, щоб спробувати виявити вторгнення. IDS на основі шаблонів провела останні дослідження, які зосереджені на конфігурації рішення IDS, це дозволяє використовувати метод виявлення для того, щоб базуватися на істотній частині мережі, наприклад захисту конкретних протоколів як основи методу виявлення. Він складається з трьох частин:

Датчик: пристрій, який фіксує бездротовий зв'язок трафік і надсилає його на сервер для аналізу. Також повідомляє про атаки.

Сервер: збирає дані з усіх датчиків, аналізує їх і реагує на атаки. Він також реєструє та сповіщає на випадок атаки.

Інтерфейс: GUI(графічний інтерфейс користувача) керує сервером і дисплеями, повідомляє інформацію про загрозу на вашій бездротовій мережі. Це про виявлення вторгнень на основі сигнатур. Він може працювати в стандартному обладнанні й виконувати процеси такі як: сканування, виявлення та запобігання вторгненню [11].

Відкрите бездротове запобігання вторгненню техніки є модульною та базується на плагінах. Для цього потрібно деяке обладнання та компоненти. Основна перевага використання цього інструменту може запобігти підключенню користувачів до інших мереж. Тут є кілька датчиків підтримки,

тому точність виявлення значно вища. Хоча у нього більше переваг, є декілька негативних сторін в цьому інструменті, які лише підтримують бездротове рішення безпеки. Якщо законних користувачів атакують, це від'єднає обох з мережі, доки проблему не буде вирішено, або на деякий час. Трафік між датчиком і сервером не шифрується, цей інструмент дозволяє адміністратору завантажувати плагіни для додаткових функцій, і він покладається на бездротове з'єднання мережі. Більшість його функцій виявлення засновані на плагінах, спільні бібліотеках. Необхідні деякі основні перевірки, які мають завжди запускатися перед плагінами. Це потрібно зробити швидко, оскільки інструмент він запускає датчики. Якщо зловмисника знайдено, він скасує автентифікацію та сповістить адміністратора [12].

OSSEC. OSSEC – це IDS на основі хоста. Він масштабований, має багатоплатформне вторгнення на основі хосту з відкритим кодом з системою виявлення (HIDS). Він лише зберігає сповіщення. Тому накладні витрати на зберігання зменшуються. Інструмент має потужний механізм кореляції та аналізу, інтегрований засіб аналізу журналів; перевірку цілісності файлу; процес моніторингу реєстру Windows; централізоване забезпечення виконання політики; виявлення root-kit; реальний час оповіщення та активне реагування. Цей інструмент здатний виявляти DOS-атаки. Переваги OSSEC полягають у тому, що його легко встановити та налаштувати. І він може підтримувати мультиплатформу. Це виконує перевірку цілісності файлу для Платформи UNIX і Windows. І головна перевага цього інструменту: він може виконувати цілісність реєстру перевірки Windows. В OSSEC, перехід до новішої версії платформи може бути складним, адже процес оновлення замінює існуючі правила стандартними правилами. Таким чином, правила користувача можна видалити. Цей засіб використовує механізми попереднього обміну. Ключі Pre sharing можуть бути проблематичними, коли Windows увімкнено режим сервер-агент.

OSSEC складається з кількох частин. Це має можливість моніторингу розподілених серверів. Він отримує інформацію від агентів і з пристрою без

агента. Тут розподілений сервер зберігає журнали доступу та журнали аудиту системи. Агент - маленький програмний блок, який передає дані на сервер для аналізу і кореляції. Безагентна схема дозволяє користувачу моніторинг цілісності файлів на них без сервера чи агента. Ця схема раніше контролювала брандмауери, маршрутизатори та навіть Unix-подібні системи. [13]

Fragroute. Fragrouter – це засіб виявлення вторгнень у мережу (NIDS), набір інструментів ухилення. Він реалізує більшість атак, описаних в безпечних мережах, наприклад вставка, ухилення та відмова в обслуговуванні (DOS) тощо, уникає вторгнення в мережу. Fragroute може використовувати Протоколи TCP/IP і односторонній фрагментований маршрутизатор. IP-пакети надсилаються від зловмисника до Fragrouter, який перетворює їх у сегментований потік даних відправлення до жертви. Fragrouter допомагає зловмиснику запускати IP-атаки, уникаючи виявлення. Він має просту мову набору правил для затримання, копіювання, скидання, розриву, друку, зміни порядку, сегментуванню або вихідного маршруту з усіма вихідними пакетами, призначеними для системи з мінімальною підтримкою. Fragrouter використовує наступний зразок формату правила:

fragroute -f <Iconfigfile>dst<destination>

Основна перевага fragroute полягає в тому, що він не потребує додаткових бібліотек.

Security Onion. Насправді це Linux на базі Ubuntu, дистрибутив для IDS і моніторингу безпеки мережі (NSM), що складається з кількох відкритих вище вихідних технології, які працюють узгоджено одна з одною . Платформа Security Onion надає всебічну інформацію виявлення вторгнень, моніторингу безпеки мережі та керування журналами, поєднуючи найкращі можливості Snort, Suricata, Bro та інших інструментів, таких як Sguil, Squert, Snorby, ELSA, Xplico серед інших та для тих хто бажає отримати найкращі з вищезгаданих інструментів один пакет. В цьому випадку Security Onion найкращий для

використання. Security Onion містить три основні функції, такі як: повне захоплення пакетів процесу, на основі мережі (NIDS) і на основі хоста системи виявлення вторгнень (HIDS) та потужні інструменти аналізу, а також забезпечує журнал і сповіщення даних про виявлені події та дії. Security Onion надає кілька варіантів IDS [14].

Sguil. Sguil – це графічний інтерфейс, що забезпечує роботу в реальному часі, доступ до записаних подій, даних сеансу та пакетних даних за допомогою систем Snort або Suricata IDS. Sguil полегшує практику моніторингу безпеки мережі та подій керованого аналізу.

У таблицях 1.3 і 1.4 показано порівняння різними інструментами. Це показує базове порівняння з підтримкою платформи та основою опису. Тут наведено наведено порівняння на основі продуктивність кожного інструменту з точки зору різних особливостей.

Таблиця 1.3 – Порівняння різних інструментів системи виявлення вторгнень

Ім’я інструменту	Провайдер	Тип	Опис	Платформа
SNORT	Cisco system	NIDS, NIPS	Може виявити Dos, CGI, вторгнення, сканування портів, SMB і атаки рівня. SNORT має здатність створювати одночасний аналіз трафіку та пакет Вхід у мережі Інтернет-протоколу (IP)	Linux, windows, Крос-платформа

Продовження таблиці 1.3

SURICATA	Open information security foundation	NIDS, NIPS	Автоматичне визначення протоколу. Процес узгодження та сумісності з SNORT.	Linux, unix, MAC, windows та інші
Bro IDS	Vern Paxson	NIDS, AIDS	Сумісний з SNORT	Linux, MAC OS , FreeBSD
OpenWIPS-ng	Aircrack-NG	NIPS	Openwips-Ng є відкритим кодом і модульною бездротовою IPS	Linux
Security Onion	-	NIDS	Містить Snort, Suricata, Sguil, Squert, Snorby, Bro, Networkminer, Xplico, та багато інших інструментів безпеки	Linux
OSSEC	Daniel B. Cid	HIDS	має потужну кореляцію і механізм аналізу, інтегрований аналіз журналів.	Крос-платформа

Продовження таблиці 1.3

FRAGROUTE	Dug Song	NIDS	Цей інструмент має хорошу репутацію. Допомога в тестуванні мережевих вторгнень системи виявлення, брандмауери та базова поведінка з TCP/IP.	Linux

Таблиця 1.4 – Порівняння на основі функцій між різними інструментами системи виявлення вторгнень

Параметр	Snort	Suricata	Bro IDS	Openwips-NG	OSSEC	FRAGROUTE
Контекстні підписи	-	+	+	-	+	-
Функція IPS	+	+	_	+	-	-
Dos атака	+	+	+	+	+	+

Кожен інструмент має свою особливість і перевагу. За різними параметрами проводяться порівняння. Порівнюючи IDS, на основі мережі та хоста IDS - мається на увазі порівняння із загальними наборами функцій.

У цьому розділі ми розглянули різні інструменти системи виявлення вторгнень. Даний аналіз дає можливість порівняти переваги та недоліки інструментів, які використовуються для виявлення та запобігання вторгненням. Було проаналізувано шість типів IDS і сім інструментів системи вторгнень, типи IDS є мережевими система виявлення вторгнень на основі хоста, гібридна IDS, мережевий хост, IDS на основі аномалій та неправильне використання системи виявлення вторгнень. З проведеного порівняння між

цими інструментами та методами, ми підсумовуємо деякі моменти. Кілька інструментів підтримують лише невеликі типи загрози та проблеми безпеки. Правильне визначення правил допоможе привести найвищий рівень виявлення, тому правила повинні бути налаштовано належним чином. Кілька інструментів все ще мають проблеми з точністю виявлення зловмисників, з мінімальною підтримкою обладнання та датчиків. Кожен інструмент особливий по своєму [15].

Оскільки Snort у порівнянні з іншими інструментами має дуже прості в установці і запуску параметри – на мою думку, він є фаворитом у цьому списку.

2 СТРУКТУРА СИСТЕМИ ТА ПРАВИЛА ВИЯВЛЕННЯ ВТОРГНЕНЬ НА БАЗІ SNORT

2.1 Структура системи виявлення вторгнень SNORT

Останні технічні досягнення призвели до використання технологій у дуже важливих сферах, таких як: електронна комерція, банківська справа, страхування, системи охорони здоров’я тощо. Необмежені можливості Інтернету та простота спілкування також становлять значний ризик різноманітних атак на користувачів та їхні дані. Однією з основних вимог користувачів мережі стало проектування захищеної мережі, яка забезпечить безпечну передачу та зберігання даних. Незважаючи на наявність різних систем виявлення та запобігання атак, проблеми присутні і сьогодні. Зловмисні атаки стають все більш витонченими, що робить їх виявлення ще більш складним.

У великих компаніях і організаціях небезпека виходить не тільки з зовнішньої мережі, але також і зсередини, іноді працівниками, які можуть використовувати свій доступ у зловмисних цілях, але ще частіше в результаті соцінженерні атаки. Виявлення вторгнень – це здатність виявляти будь-який вид несанкціонованого доступу до комп’ютерної система або мережі. Вторгнення в систему можна визначити як спробу зловмисника отримати доступ до комп’ютера або мережі в обхід механізму безпеки. Крім того, вторгнення часто має на меті скомпрометувати CIA (конфіденційність, цілісність і доступність). Система виявлення вторгнень використовується для вирішення проблеми вторгнень шляхом моніторингу мережевого трафік і події, детально аналізуючи їх і виявляючи несанкціоновані вторгнення.

У більшості середовищ Snort IDS налаштовано на журнал сповіщення після виявлення потенційно небезпечних мережевих пакетів. Для захисту мережі потрібен детальний аналіз цих зареєстрованих сповіщень. Однією з найбільших проблем у моніторингу є великий обсяг генерованого оповіщення. Системи IDS постійно вдосконалюються, але також стають ще більш

чутливими до різних типів атак. Одним із наслідків є високий показник помилкових спрацьовувань сповіщень, що робить виявлення реальної небезпеки мережі складним завданням. Крім того, неможливість знайти зв'язок між великою кількістю сповіщень робить процес захисту та передбачення наступного кроку зловмисника ще більш складним. Тому успішне вилучення важливої інформації з великого обсягу даних IDS є надзвичайно важливим завданням [16].

Візуальний аналіз мережевих даних допомагає мережевим адміністраторам визначити моделі та тенденції трафіку, а також помітити можливі відхилення в безпеці. Також, аналізуючи згенеровані журнали трафіку, мережа візуалізуючи події, дозволяє планувати необхідні дії та кроки безпеки швидше і зрозуміліше. Системи візуалізації можуть мати різні вхідні дані, в тому числі необроблені мережеві дані, мережеві події від мережевих пристроїв (маршрутизаторів, комутаторів тощо), події безпеки від систем IDS/IPS/брандмауера та журналів програм. Оскільки кількість трас мережевих подій постійно зростає, обов’язково потрібно звернути увагу на надійність системи моніторингу та можливість обробки значної кількості даних. Багато популярних мереж з відкритим кодом з рішеннями візуалізації є дуже надійними. Також, значною перевагою є те, що візуальна реакція є негайною, без жодних проміжних компоненти для попередньої обробки сповіщень.

IDS можна класифікувати на основі збігу сигнатур та на основі аномалій, а також, на основі техніки класифікації, яка використовується для поділу мережевих пакетів на дві частини - звичайні та шкідливі. Системи на основі підписів використовують властивості попередніх атак для виявлення загроз.

Підпис – це шаблон відомої атаки або загрози, яка попередньо ідентифікована та збережена в базі даних. Відповідні IDS порівнюють мережевий трафік із шаблонами зловмисних спроб, щоб розпізнати можливі вторгнення. Окрім того, що він ефективний, він базується на підписах техніки виявлення та може мати проблеми з виявленням нових і попередніх невідомих загроз.

Виявлення аномалій – це процес порівняння мережевого трафіку з визначенням нормальної мережевої активності, щоб розпізнати значущі відхилення. Вважається, що аномалія мережевого трафіку відхиляється від відомого трафіку поведінкою. Ця поведінка настільки нетипова, що викликає підозру в зловмисних діях. Ці системи можуть виявляти атаки, але також можуть мати хибні спрацьовування оповіщень. Крім того, в останніх системах IDS часто використовується штучний інтелект з метою подальшого покращення виявлення системи вторгнень.

З точки зору компонентів NIDS, типовий NIDS збирає дані з мережі, розподіляє їх на мережеві датчики і далі на мережеві аналізатори, які класифікують ці дані як безпечні або шкідливі, а також визначають рівень загрози. NIDS також включає оповіщення сповіщень, які генеруються як сповіщення на екрані, звукові сповіщення, повідомлення електронною поштою, тощо. Більшість NIDS використовують кілька датчиків, які повинні бути розміщені на ключових точках мережі. Вони можуть бути розгорнуті в одному з двох режимів. Вбудований датчик зазвичай розміщується на межі мережі, щоб контролювати весь трафік, який проходить через неї. Пасивний датчик відстежує дзеркальний мережевий трафік замість фактичного трафіку.

Snort – це широко використовувана програма з відкритим вихідним кодом, яка легко конфігурується та портативна система виявлення мережевих вторгнень на основі шаблону. Snort легко розгорнутий на різноманітних вузлах мережі. Крім того, його робота ефективна і не займає багато пам'яті та процесорного часу. Snort використовує набір підписів, які визначають, що саме є атакою, і таким чином дозволяє виявити атаки та шкідливі дії. Набори підписів Snort називаються правилами Snort. Правило є формально визначається так:

<rule action><protocol><source ip><source port>
<direction><dest ip><dest port><rule options>

Поле дії правила визначає тип правила Snort (сповіщення, журнал). Найбільший загальними є правила сповіщень, які зберігають дані сповіщень для подальшого аналізу та пізніше пошуку. Решта полів описують основні атрибути мережевих пакетів. Поле параметрів правила визначає одну або кілька пар ключ-значень, які додатково описують правило (тип класу, повідомлення, тощо). Кожне правило також призначає пріоритет сповіщенню, відповідно до класу оповіщення. Пріоритет 1 вказує на найбільш серйозну загрозу, а пріоритет 4 означає найменшу загрозу. Приклад правила Snort:

Alert tcp $EXTERNAL NET any ->$HOME NET any (msg:
’SCAN SYN FIN’ flags: SF, 12; reference: arachnids, 198;
classtype: attempted-recon;)

Основні компоненти архітектури Snort наведені на рисунку 2.1. Пакет сніффер збирає мережевий трафік і направляє його на декодер, який обробляє захоплені пакети, щоб ізолювати заголовки протоколу на кожному з рівнів OSI (модель взаємодії відкритих систем). Фактичне виявлення вторгнення здійснюється в блоці механізму виявлення. Цей модуль аналізує кожен пакет і перевіряє його на відповідність усім правилам. Дія (сповіщення), визначене правилом, запускається кожного разу, коли виявляється пакет, який відповідає умові, визначеній правилом.

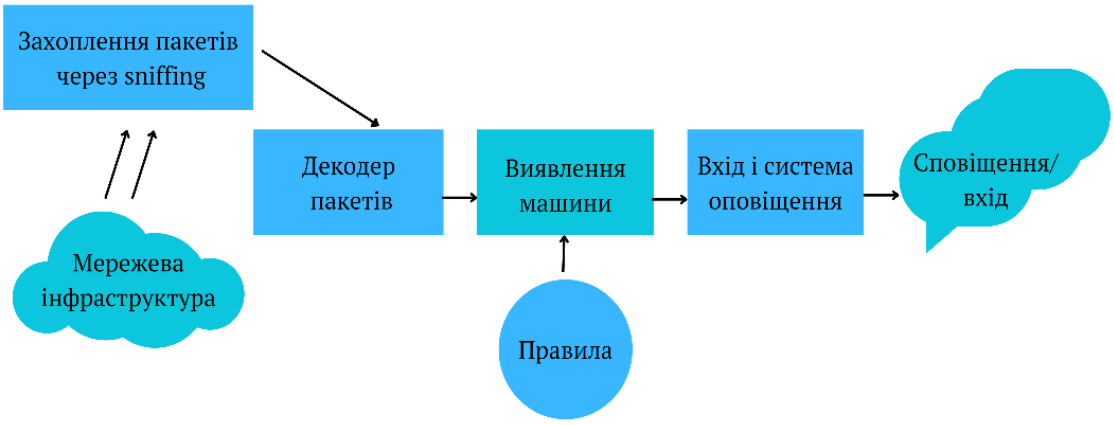


Рисунок 2.1– Архітектура Snort

Пропонований інтерфейс візуалізації системи Snort IDS реалізований як клієнт-серверна програма, яка структурує та графічно представляє сповіщення про дорожній рух Snort. Пропонований інтерфейс дозволяє користувачам візуально аналізувати журнали трафіку в режимі реального часу та легко виявляють відхилення від нормального трафіку, які можуть вказують на можливі вторгнення. Щоб продемонструвати та оцінити запропоноване рішення, Snort інтерфейс візуалізації інтегровано в систему, компоненти якої є наведено на рисунку 2.2. З вибраних ключових точок мережі трафік надсилається на машину, на якій виконується Snort IDS. Для відповідного підпису Snort використовує зареєстровану базу правил із відкритим кодом. Для того, щоб виявлення було максимально точним важливо регулярно оновлювати базу даних. Snort IDS реєструє сповіщення у файловій системі машини у форматі JSON. Реалізований графічний інтерфейс Snort зчитує дані з лог-файлу, обробляє і відображає його користувачу.

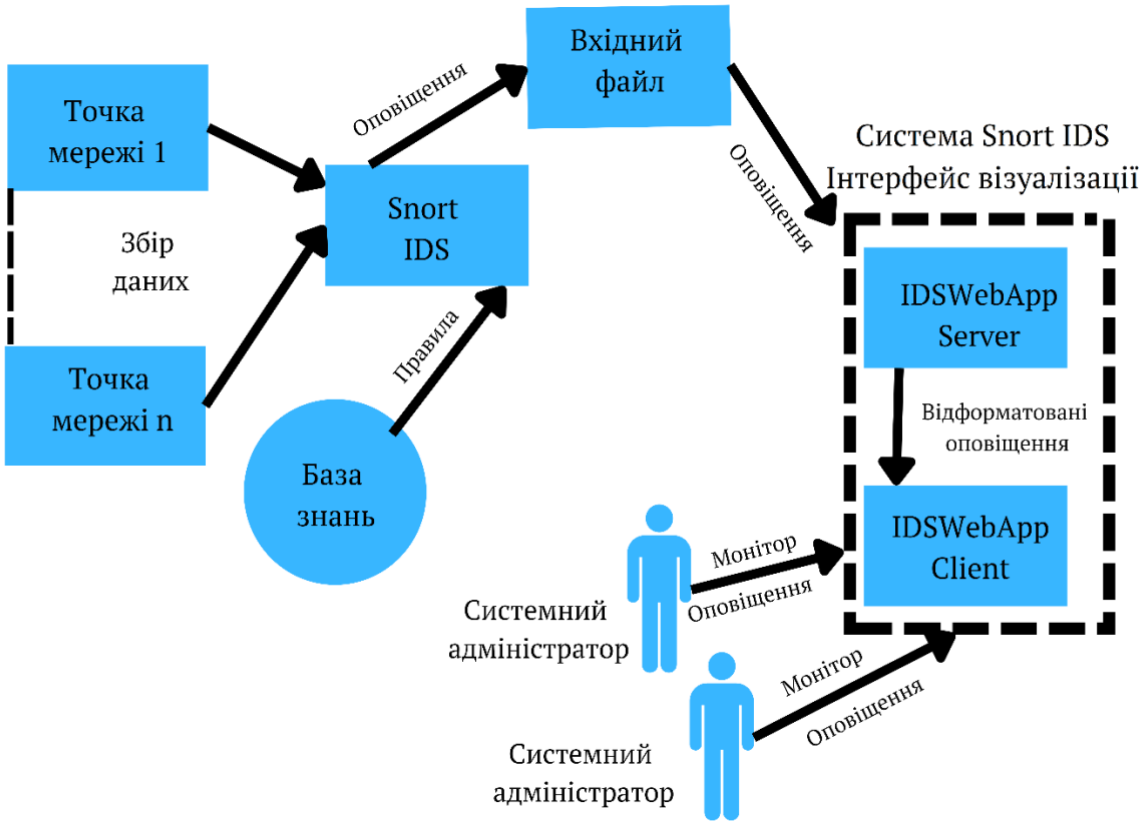


Рисунок 2.2 – Архітектура запропонованої системи

Використовуючи клієнт-серверну модель, запропонований інтерфейс візуалізації дозволяє ефективно графічно подавати сповіщення, створені Snort IDS. Серверна сторона програми (сервер IDSWebApp) читає файл журналу Snort IDS під час запуску програми (первинне читання), а також кожного разу, коли цей файл змінюється, тобто коли Snort створює нове сповіщення. Сервер IDSWebApp також форматує ці сповіщення, щоб їх можна було правильно надіслати клієнту. Після цього, сервер IDSWebApp надсилає зібрані сповіщення клієнту за допомогою веб-сокета. Сторона клієнта (клієнт IDSWebApp) отримує дані через веб-сокет і читає сповіщення, надіслані сервером. Його основна функція полягає в організації та відображенні сповіщень Snort для системного адміністратора в режимі реального часу, оновлюючи інтерфейс із кожним новим сповіщенням. Це призводить до миттєвого відображення нових сповіщень для користувача. Крім того, клієнт сортує сповіщення, отримані від сервера, за чотирма критеріями та відображає найпоширеніші адреси джерел атаки, найпоширеніші пріоритети попереджень, найпоширеніші класи атак і найпоширеніші дати атак. Статистика сповіщень регулярно оновлюється, показуючи найпоширеніші атрибути атак. Крім того, якщо користувач вибирає певне сповіщення, клієнт IDSWebApp відобразить детальну інформацію про нього. Таким чином виявлення можливе порушення дорожнього руху стає швидким і простим. Робота графічного інтерфейсу та зв’язку клієнт-серверу для відображення попереджень для користувача наведена на послідовній схемі (рисунок 2.3). [17].

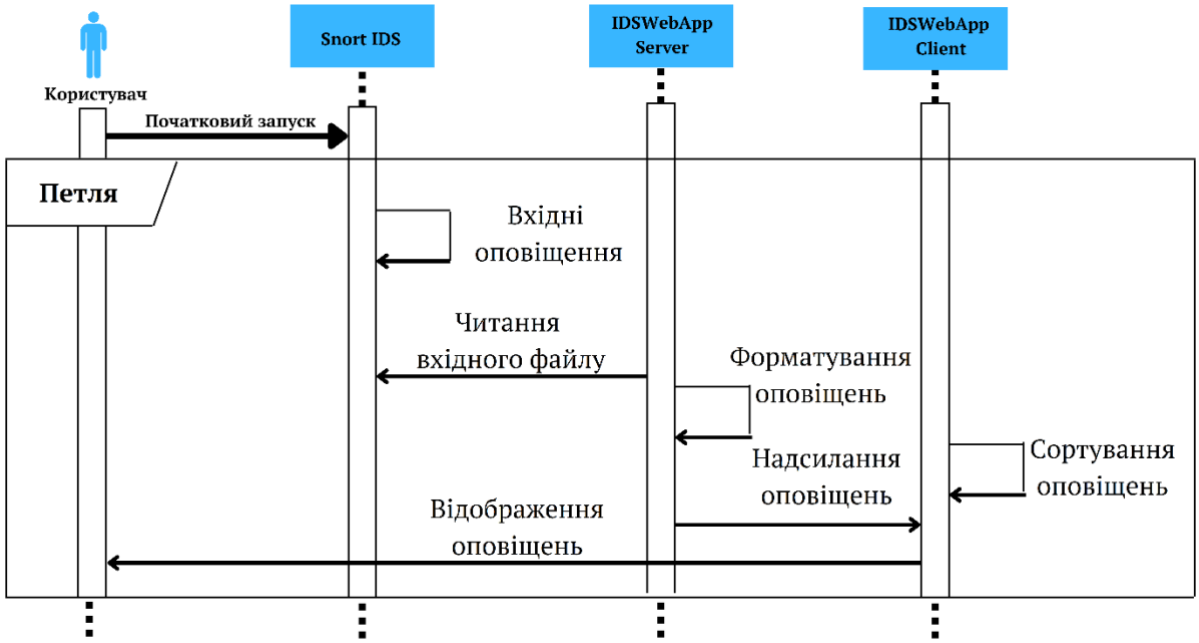


Рисунок 2.3 – Послідовна схема роботи інтерфейсу візуалізації Snort IDS

2.2 Написання правил Snort

Snort є одним з найпопулярніших NIDS. Snort є відкритим вихідним кодом, що означає, що оригінальний вихідний код програми доступний будь-кому безкоштовно, і це дозволило багатьом людям зробити свій внесок у створення програм та проаналізувати їх. Snort використовує найпоширенішу ліцензію з відкритим кодом, відому як GNU (General Public License). Snort, як і більшість NIDS, використовує набір сигнатур для визначення того, що є атакою. Підписи Snort регулярно оновлюються на веб-сайті SNORT, як правило, кілька разів на день, що можна підтвердити, періодично перевіряючи мітки часу біля доступних завантажень Snort. Як показано на рисунку 2.4, Snort є гнучким у тому, як його можна використовувати. Файл, що містить раніше зареєстрований трафік, можна використовувати як вхідні дані для SNORT точно так само, як і живий трафік. Snort також підтримує низку вихідних даних, таких як збереження сповіщень у файлах або базах даних або

створення журналу мережевого трафіку всього отриманого трафіку для подальшої обробки у випадку живого захоплення трафіку. Існує гнучкість SNORT для підтримки практично будь-якого методу виведення завдяки можливості підтримки як внутрішніх, так і сторонніх плагінів виводу [20, 21].

Оскільки чітке розуміння правил SNORT має вирішальне значення для нашого дослідження, наведено детальне пояснення. Короткий виклад цієї інформації відображено на рисунку 2.4. Набори підписів SNORT, які використовуються для виявлення порушень безпеки, називаються правилами SNORT. Групи правил SNORT називають файлом rules, кожне з яких можна вибірково включити до файлу конфігурації SNORT snort.conf. Файл .rules — це простий текстовий файл, у якому кожен рядок містить окреме правило. Наступні примітки щодо формату правил SNORT були зібрані за допомогою посібника користувача SNORT, яку можна отримати на веб-сайті SNORT.

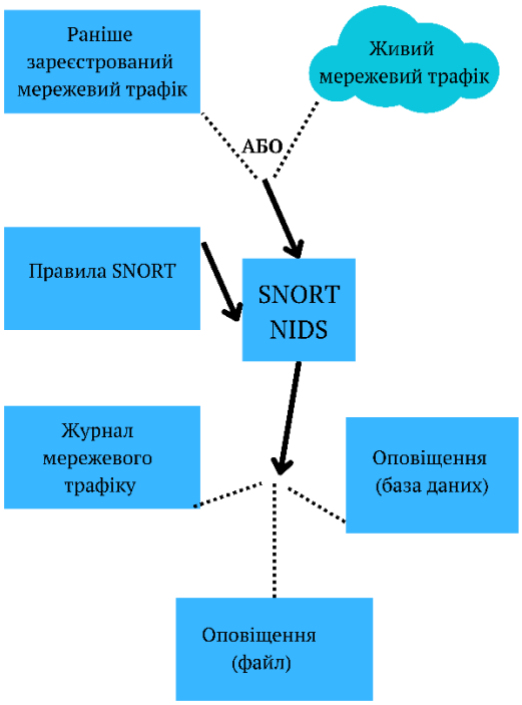


Рисунок 2.4 – Діаграма гнучкості SNORT

Правило формально визначено, як показано нижче. Текст у <кутових дужках> замінено відповідною обов’язковою змінною без кутових дужок.

Текст у [квадратних дужках] необов’язковий або нічого не представляє, або представляє сам текст, у будь-якому випадку без квадратних дужок.

<rule action> <protocol> [!]<source ip> [!]<source port>
<direction> <dest ip> <dest port> <rule options>

example: alert tcp any any -> any 25
[create an alert for any incoming traffic send to port 25]

Термін «Дія правила» описує, яка відповідь виконується у випадках, коли умови правила збігаються при порівнянні з інтернет-пакетом. Зазвичай дією правила є попередження, що зазвичай означає збереження даних попередження у файлі чи базі даних для подальшого використання, отримання або для обробки іншою програмою. Пакети, що генерують сповіщення, також реєструються. Інша дія включає журнал, коли генерувати сповіщення недоцільно. Інші дії включають передачу (пропуск пакета) та активацію (запуск інших правил або дій).

Для операторів IP SNORT використовує номер блоку IP/CIDR (безкласова міждоменна маршрутизація) після IP-адреси. Пакетні дані повинні ідентифікувати себе як такі, що надходять з указаного діапазону IP-адрес або йдуть до нього. Перед IP-адресою можна поставити необов’язковий знак оклику, щоб змінити значення правила. Значення можна вказати як діапазони операторів IP/CIDR, напр. 192.168.2.0/12. Усі можливі IP-адреси можна представити за допомогою ключового слова any.

Пакетні дані мають ідентифікувати себе як такі, що надходять із наданого Інтернет-порту або діапазону портів (Source/Destination Port). Перед оператором порту можна поставити необов’язковий знак оклику, щоб інвертувати значення правила. А певний номер порту або діапазон можна вказати за допомогою двокрапки (:), щоб відокремити найнижчий і найвищий номери портів, напр. 1:1027. Крім того, усі можливі номери портів можна

представити за допомогою ключового слова any. Оператор напряму визначає, чи йде пакет від джерела до пункту призначення чи навпаки.

Існують також додаткові параметри правила, включаючи додаткові умови для відповідності правила, повідомлення, яке буде використовуватися в сповіщеннях, і параметри для активації правил. Зацікавленого читача направляємо на веб-сайт SNORT для отримання додаткової інформації. Параметри правила відокремлюються один від одного, крапкою з комою. З деякими опціями пов’язано значення параметра, у цьому випадку двокрапка розділяє ім’я опції та її значення.

Створення нових правил шляхом узагальнення правил SNORT. Враховуючи Інтернет-пакет, який містить варіант відомої атаки, повинен існувати певний автоматичний спосіб ідентифікації пакета як такого, що майже відповідає сигнатурі атаки NIDS. Якщо певний оператор має набір умов проти нього, елемент може відповідати деяким умовам. У той час як логіка надає значення false для запиту «чи відповідає цей елемент умовам», наша логіка може дозволити елементу відповідати меншою мірою, а не повністю. Цей принцип можна застосувати під час порівняння інтернет-пакетів із набором умов у правилі SNORT. Гіпотеза полягає в тому, що якщо всі умови, крім однієї, виконуються, попередження з нижчим пріоритетом може бути видано проти Інтернет-пакету, оскільки пакет може містити варіант відомої атаки. У реалізації узагальнення, у випадку зіставлення мережевих пакетів із правилами передбачає дозвіл пакету генерувати сповіщення, якщо:

- умови правила не збігаються, але більшість з них збігаються;
- єдині умови, які не збігаються, майже збігаються.

Як приклад, припустімо, що певне правило стверджує, що сповіщення повинно бути згенеровано, у разі, якщо пакет має певну довжину, на певному порту та містить певний бітовий шаблон. Використовуючи узагальнення, пакет, що відповідає цим критеріям, за винятком, можливо, іншого порту або з дещо іншим бітовим шаблоном, все одно вважатиметься таким, що відповідає, і буде створено (змінено) сповіщення [18].

Реалізація складається з трьох компонентів (рис. 2.5).

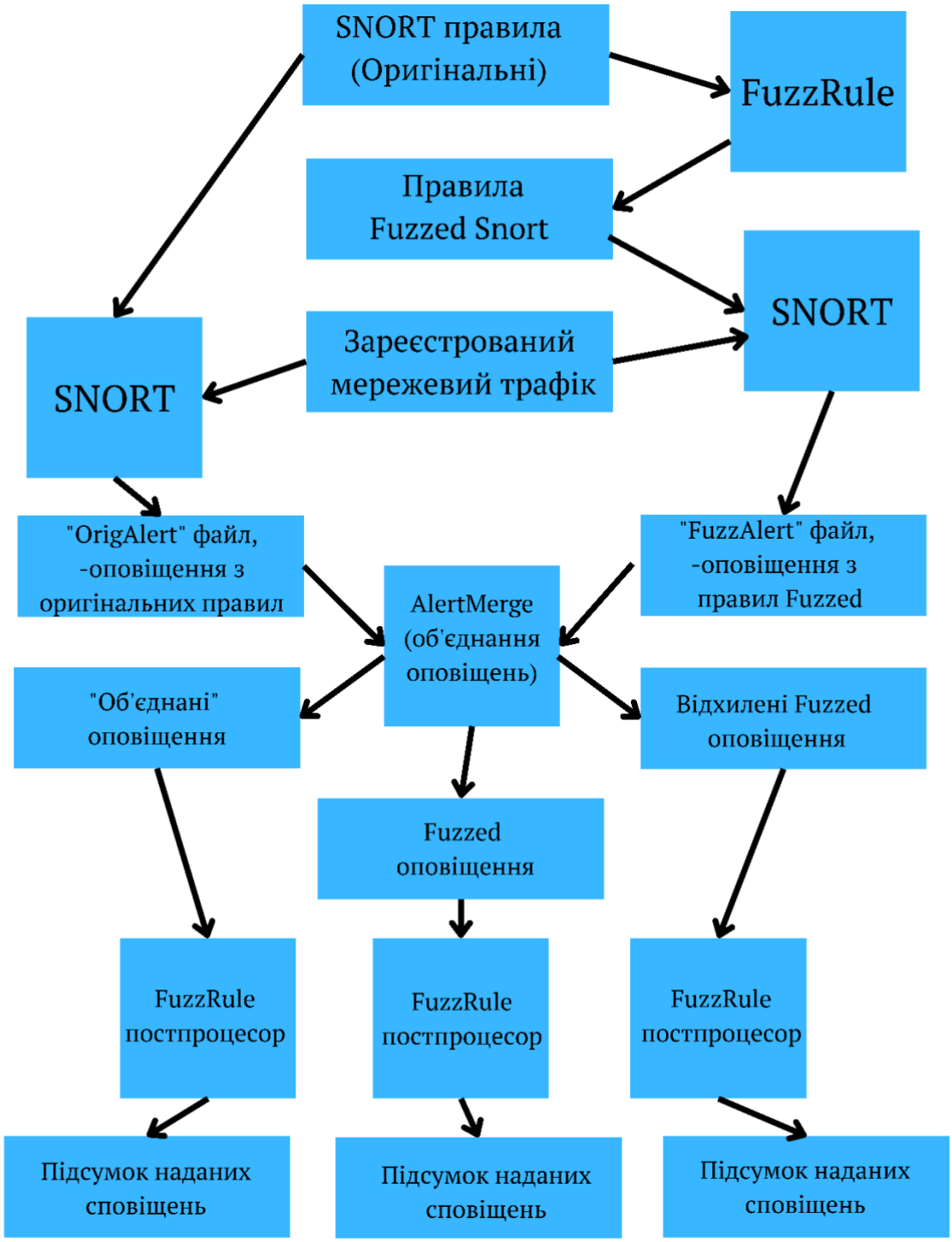


Рисунок 2.5 – Діаграма потоку даних для загальної системи.

1. Перший компонент під назвою FuzzRule, обробляє правила Snort (файли .rules) і створює два нових набори правил, використовуючи два принципи узагальнення (Invert і Content);

2. Другий компонент, AlertMerge, об’єднує файли сповіщень, створені на основі вихідних правил, із файлами сповіщень, створених на основі узагальнених правил;

3. Третій компонент, постпроцесор FuzzRule, підсумовує надані сповіщення. Перевіряючи цей підсумок, ми можемо визначити, де генерується велика кількість помилкових спрацьовувань. Таким чином, ми можемо налаштувати FuzzRule, щоб зменшити кількість помилкових сповіщень. [19]

FuzzRule. Компонент FuzzRule, схематичний огляд якої наведено на рисунку 2.5, відповідає таким специфікаціям:

За наявності файлу .rules програма зберігає резервну копію вихідного файлу перед його заміною. Для кожного правила SNORT у вихідному файлі .rules програма включає оригінальне правило в новий файл .rules і дотримується цього з кожною варіацією правила, створеного за допомогою наше узагальнення. Кожна узагальнена варіація оригінального правила генерується шляхом інвертування або видалення значення одного з параметрів правила. Таким чином, при порівнянні властивостей пакета з узагальненим правилом, пакет повинен відповідати всім випадкам, подібним до вихідного правила. Між видаленням і інвертуванням є різниця, і правильної поведінки можна досягти лише шляхом параметри правила інвертування. На основі початкового набору експериментів визначили наступні параметри правила як хороші кандидати для узагальнення. Будь-яке з них, присутніх у правилі, узагальнено за допомогою зазначеного методу (докладніше в цьому розділі):

– Інверсія: порт, IP-адреса, напрямок, протокол, вміст, вміст URI;
– Спеціальна інверсія: глибина, зсув.
– Узагальнення вмісту: вміст, вміст URI.
– Як оригінальні правила, так і узагальнені варіації правил мають повідомлення-попередження з тегами, щоб можна було ідентифікувати, яким чином відповідне правило було узагальнено.

Надається опція програми, яка надає кожному узагальненому варіанту правила нижчий пріоритет. Пріоритет – це числове значення від одиниці і

більше, де одиниця є найвищим пріоритетом і представляє найсильнішу атаку, а будь-яке більше значення меншу. Налаштування пріоритету не використовується у Snort, але служить індикатором для оператора, який переглядає файл попередження.

2.3 Узагальнення правила за допомогою інверсії

Під час розробки програми спочатку застосовано правила для формування узагальненого правила, шляхом видалення єдиного параметра. Наприклад, враховуючи наведене нижче початкове правило, один узагальнений варіант надається згодом. Якщо видалити параметр offset, більша кількість пакетів відповідатиме узагальненому правилу, а не оригінальному:

```
alert udp any any -> any 69 (msg:TFTP GET Admin.dll;
 content:
 |0001|; offset:0; depth:2; content:admin.dll; offset:2;
 nocase; classtype:successful-admin; reference:url,
www.cert.org/advisories/CA-2001-26.html; sid:1289; rev:2;)

 alert udp any any -> any 69 (msg:TFTP GET Admin.dll;
 content:
|0001|; offset:0; content:admin.dll; offset:2;
nocase; classtype:successful-admin; reference:url,
www.cert.org/advisories/CA-2001-26.html; sid:1289; rev:2;)
```

Використання наведеного вище принципу узагальнення видалення означає, що якщо пакет відповідає вихідному правилу, він, як правило, також відповідає всім узагальненим варіантам того самого правила. Згідно з

дизайном, SNORT створює не більше одного сповіщення на пакет. Коли ми вперше спробували підхід до видалення, ми очікували, що SNORT за замовчуванням відповідатиме оригінальному правилу, оскільки воно з’являється перед узагальненими варіаціями у файлі .rules. Однак під час виконання тестів сповіщення генерувалися лише на основі загальних правил. Ретельніше дослідження показало, що SNORT розміщує правила в ефективній системі у стилі бінарного дерева для швидшої обробки та проходить дерево, спочатку зіставляючи недорогі правила відповідності. Таким чином, правила з меншою кількістю параметрів, як-от узагальнені правила видалення, що збігаються перед їхніми оригінальними аналогами. Зміна пріоритету та/або ідентифікатора SNORT сповіщень не може жодним чином змінити цю поведінку. [20]

Тому довелося застосувати новий принцип. Замість того, щоб видалити параметри правил, ми інвертували їх. Принцип інвертування параметра правила визначається як відповідність пакету лише в тих випадках, які схожі, але де оригінальний параметр правила не мав би збігаються. Різниця між принципом узагальнення видалення та принципом узагальнення інверсії пояснюється в таблиці 2.1 за допомогою правила з чотирма умовами A, B, C і D. Принцип видалення означає, що тільки, якщо виконуються не всі початкові умови правила, збігається максимум одне узагальнене правило. Те саме стосується принципу інверсії. Однак, якщо всі початкові умови правила відповідають, усі узагальнені правила за принципом видалення також відповідатимуть. Навпаки, згідно з принципом інверсії, якщо всі вихідні умови правила збігаються, жодне з узагальнених правил не збігається. Останнє є бажаним результатом і, отже, нашим вибором для впровадження.

Використовуючи принцип інверсії, застосування узагальнення є простим для більшості правил, напр. інвертування портів, IP-адрес, протоколів, напрямків трафіку або заперечення всього вмісту чи рядки вмісту URI. На жаль, для деяких варіантів правила пошук узагальненого відповідника є більш складним.

Таблиця 2.1– Різниця між принципами узагальнення та інвертування.

Правило	Видалення				Інвертування			
Оригінал	A	B	C	D	A	B	C	D
Узагальнення 1	A	B	C	-	A	B	C	не D
Узагальнення 2	A	B	-	D	A	B	не C	D
Узагальнення 3	A	-	C	D	A	не B	C	D
Узагальнення 4	-	B	C	D	не A	B	C	D

Як приклад, узагальнені версії параметрів правил глибини та зміщення. Параметри глибини та зміщення впливають на те, з якою частиною пакетних даних буде відповідати параметр вмісту. Значення зміщення означає, що рядок вмісту не порівнюється, поки не буде зміщена кількість байтів у пакетних даних. Значення глибини визначає, на скільки байтів від початку зсуву (або початку пакета, якщо зміщення не вказано) має проводитися порівняння між даними пакета та рядком вмісту.

Також, варто зазначити ефективний метод узагальнення вмісту, оскільки це часто є ключем до зіставлення правила з шаблонами трафіку. Параметр вмісту визначає рядок для пошуку в пакетах. Застосування узагальнення до опції вмісту означає, що окремі символи у вмісті замінюються знаком питання (?), щоб позначити будь-якого персонажа під час події. Крім того, значення опції вмісту можна трохи скоротити, що може дозволити збіг, якщо початкові або кінцеві символи в послідовності атаки відрізняються. Цей тип узагальнення застосовується до всіх правил зі змістом (фактичним вмістом) і uricontent (наприклад, веб-адреси). У всіх випадках ряд узагальнених правил створюється шляхом заміни одного символу по черзі на символ (?).

AlertMerge. Компонент AlertMerge схематично показаний на рисунку 2.6. Компонент приймає два файли попереджень, обидва згенеровані SNORT для одного трафіку. Один із файлів попереджень генерується з використанням

оригінальних, а інший — узагальнених правил. Відтепер ці файли називатимуться файлами вихідного та узагальненого сповіщень відповідно. Створюється три файли виводу, кожен з тим самим іменем файлу, що й файл попередження, але з конкретним розширення, що додається в кінці імені. Один файл із розширенням .merged, який містить деякі сповіщення з узагальненого файлу сповіщень і всі сповіщення з вихідного файлу попереджень.

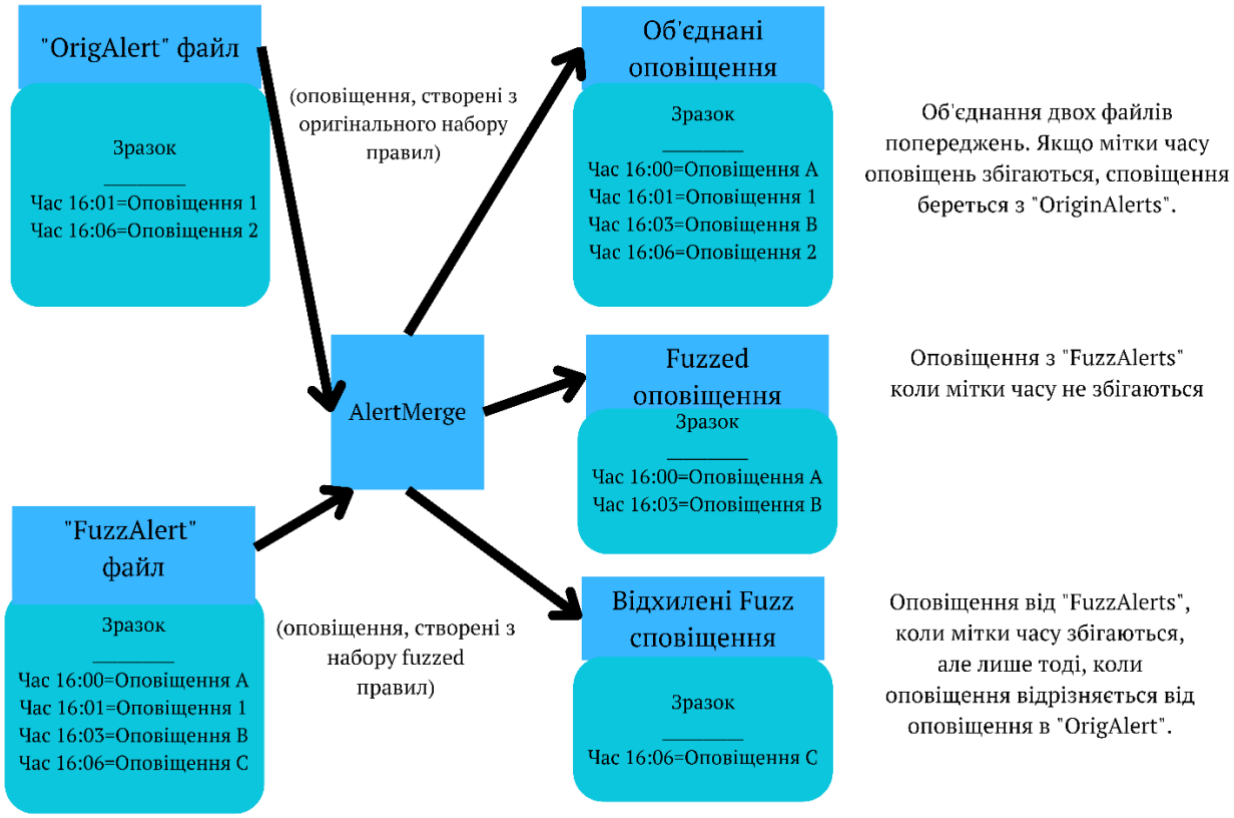


Рисунок 2.6 – Огляд діаграми потоку даних AlertMerge

Як обговорювалося раніше, SNORT може попереджати про менш важливе узагальнене правило замість оригінального правила, якщо пакет відповідає обом. Таким чином, файл попередження лише з узагальнених правил може означати, що трафік є менш серйозним, ніж насправді. Процес злиття гарантує, що в об’єднаному файлі сповіщень записується лише одне сповіщення на пакет даних. Якщо два сповіщення, по одному з кожного з двох заданих файлів сповіщень, генеруються з одного пакету, тоді лише сповіщення

з оригінального файлу сповіщень зберігається у файлі .merged. Сповіщення зберігаються в хронологічному порядку. Один файл із розширенням .fuzz, який містить усі сповіщення, створені за узагальненими правилами, які прийняті до файлу .merged. А також, один файл із розширенням .rejected_fuzz, яке містить усі сповіщення з узагальненого файлу сповіщень, що не потрапили до файлу .merged. Цей файл корисний для визначення того, які узагальнені правила мають пріоритет над початковими правилами.

Постпроцесор FuzzRul. Компонент пост-обробки FuzzRule підсумовує файл попередження, створений Snort. Файл підсумовується незалежно від того, чи було його взято безпосередньо з Snort, чи файл сповіщення був одним із чотирьох можливих результатів AlertMerge. Після цього, підсумок попередження надається як загальна кількість випадків кожного попередження та загальна кількість випадків кожного реалізованого методу узагальнення, включаючи кількість випадків вихідних попереджень.

3 ДОСЛІДЖЕННЯ ТА РЕАЛІЗАЦІЯ СИСТЕМИ ВИЯВЛЕННЯ ВТОРГНЕНЬ

3.1 Встановлення програми Snort

Для встановлення програми Snort варто перейти за посиланням на офіційний сайт: https://www.snort.org/downloads та вибрати версію програми. На рисунку 3.1 відображено сайт програми Snort із варіантами вибору версій.

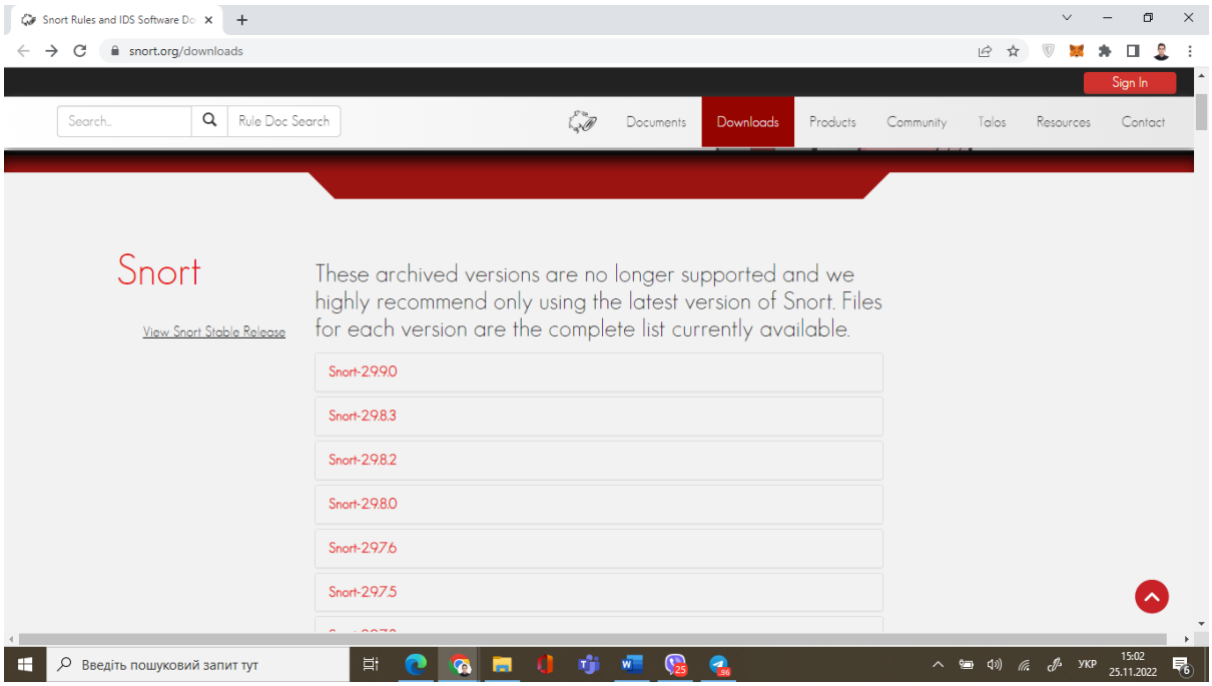


Рисунок 3.1 – Сайт програми Snort

Для того, щоб користувач зміг встановити більший асортимент версій – необхідно зареєструватись на сайті Snort. На рисунку 3.2 зображено вигляд кабінета користувача.

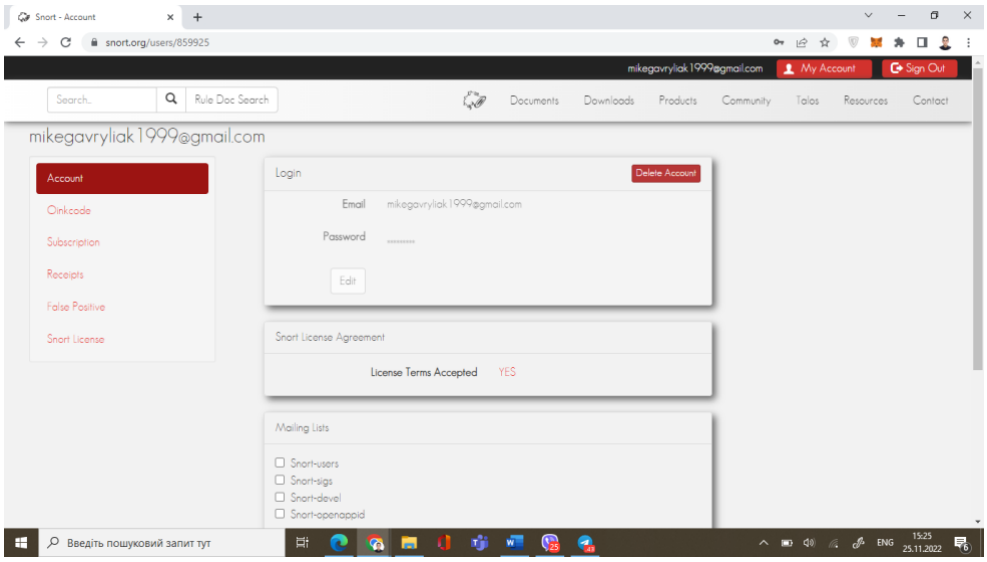


Рисунок 3.2 – Вигляд кабінету користувача

Знаходимо версію для Windows ОС та готуємось до її встановлення. В нашому випадку – це версія 2.9.20 (рисунок 3.3).

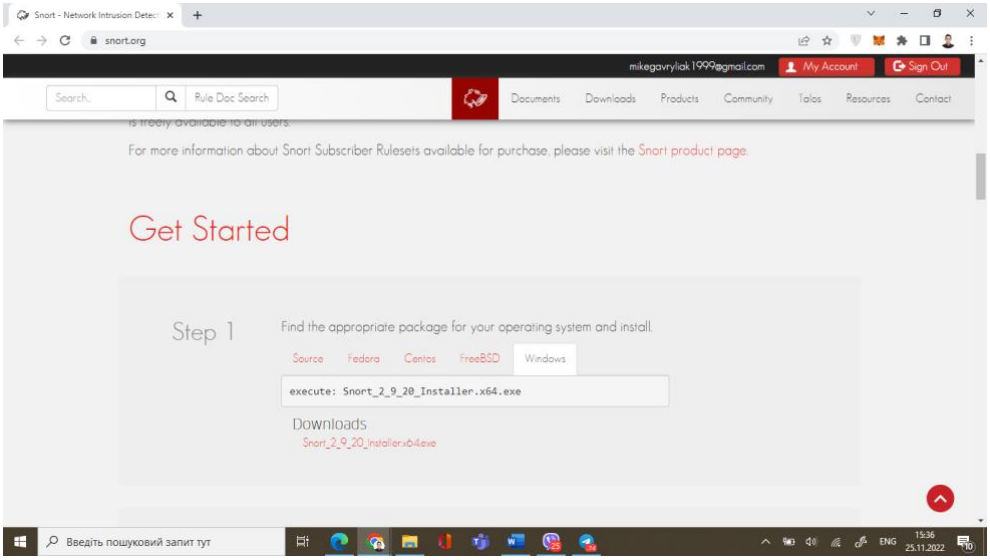


Рисунок 3.3 – Встановлення версії Snort 2.9.20 для Windows

Перший крок, погоджуємось із умовами використання (рисунок 3.4)

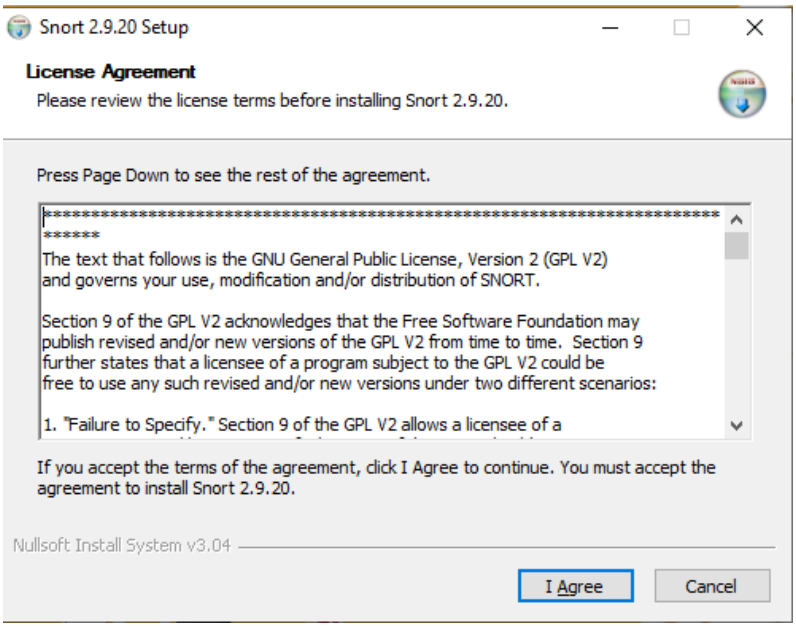


Рисунок 3.4 – Перший крок встановлення Snort

Обираємо компоненти для Snort. Варто обрати всі три компоненти “Snort”, “Dynamic Modules”, “Documentation”. Вибір компонентів відображено на рисунку 3.5.

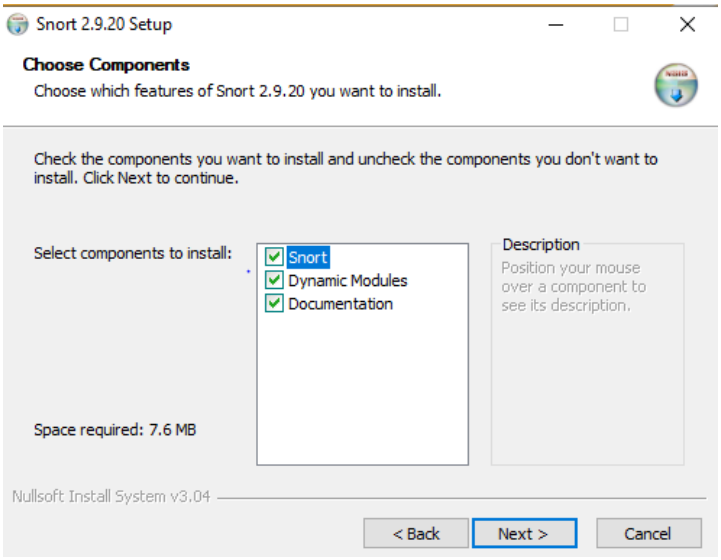


Рисунок 3.5 – Вибір компонентів Snort

Після цього обираємо локацію, куди встановити Snort. В нашому випадку – це локальний диск C. Програму Snort встановлено (рисунок 3.6).

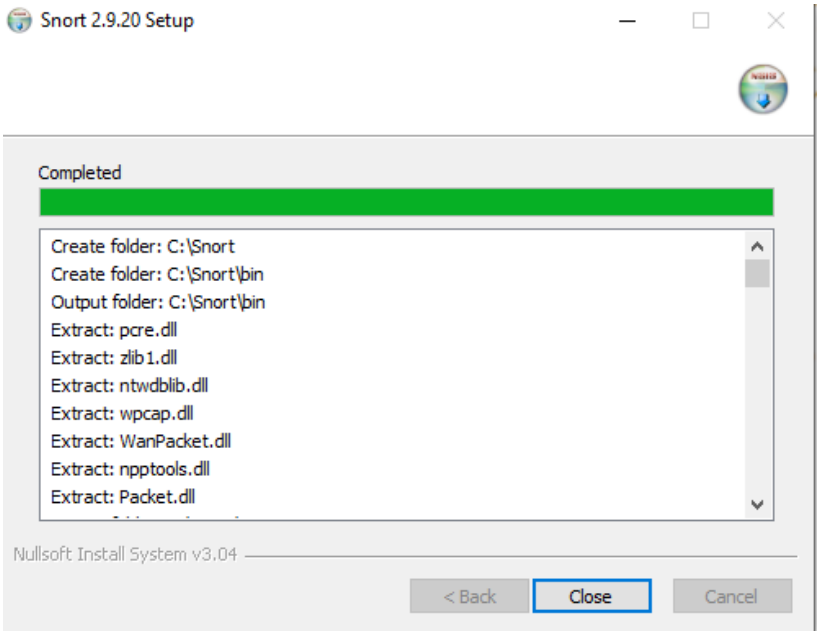


Рисунок 3.6 – Успішне встановлення Snort

Після закриття програми – отримуємо сповіщення (рисунок 3.7) про те, що необхідно встановити npcap із сайту www.nmap.org.

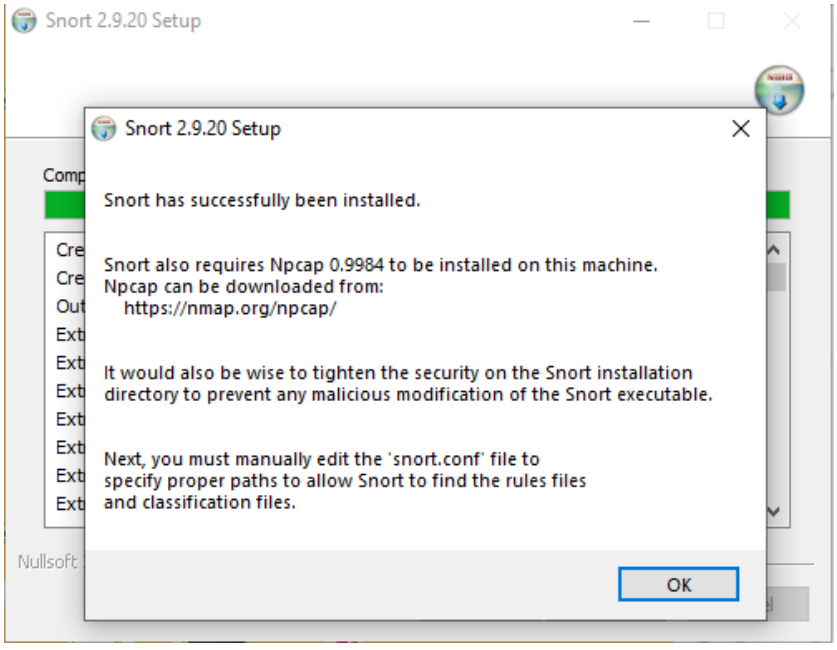


Рисунок 3.7 – Сповіщення про потребу встановлення Npcap.

Відвідуємо сайт www.nmap.org/npcap/ та завантажуємо додаток до програми Snort. Після чого переходимо до встановлення. Обираємо всі три опції встановлення (рисунок 3.8).

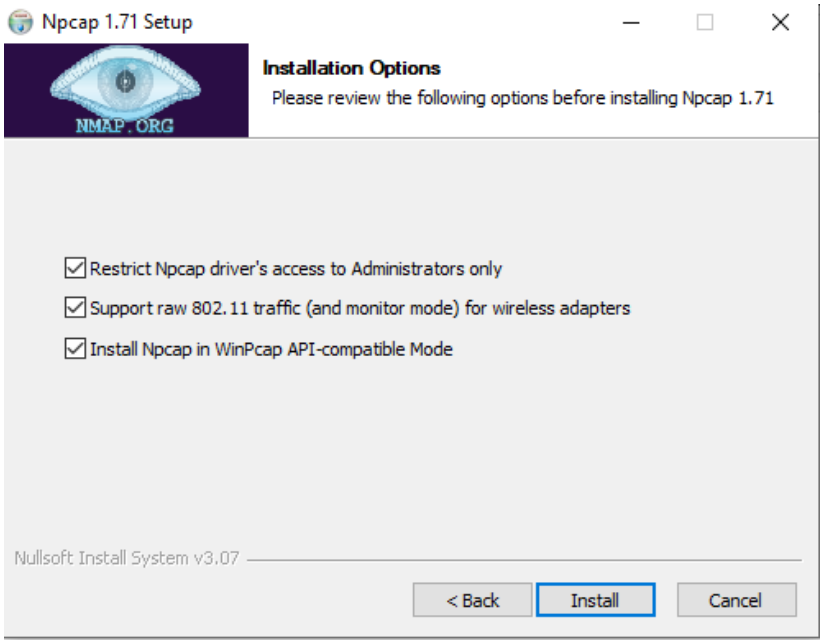


Рисунок 3.8 – Вибір опцій встановлення Npcap

Натискаємо “install” та бачимо, що Npcap успішно встановлено (рисунок 3.9).

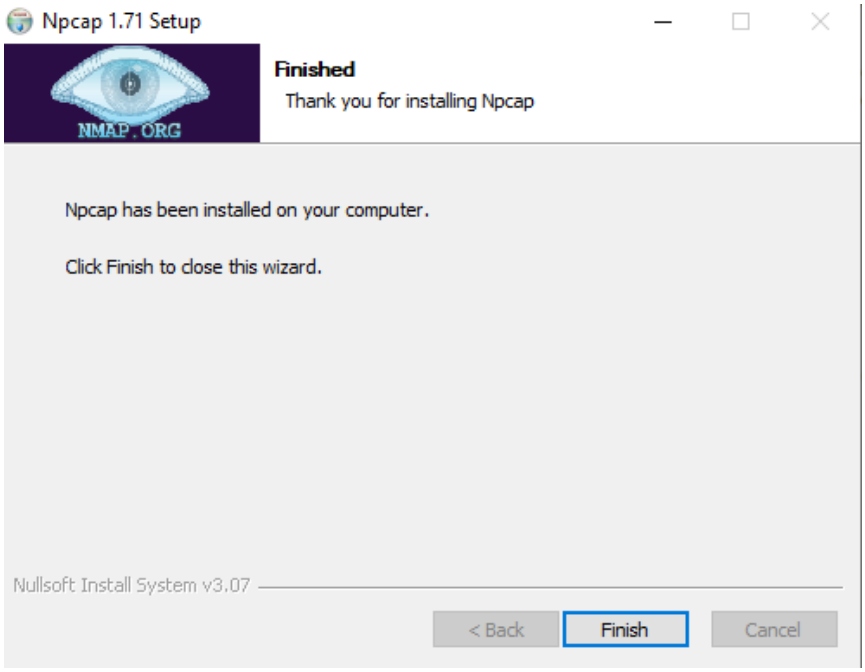


Рисунок 3.9 – Успішне встановлення Npcap

Після успішного встановлення версії Snort та Npcap встановлюємо правила.

Їх також знаходимо на офіційному сайті Snort. Правила необхідно підібрати відповідно до версії Snort. Завантажуємо архів та розпаковуємо його. В результаті з архіву мають розпакуватись наступні папки: “rules”, “so_rules”, “etc”, “preproc_rules” (рисунок 3.10).

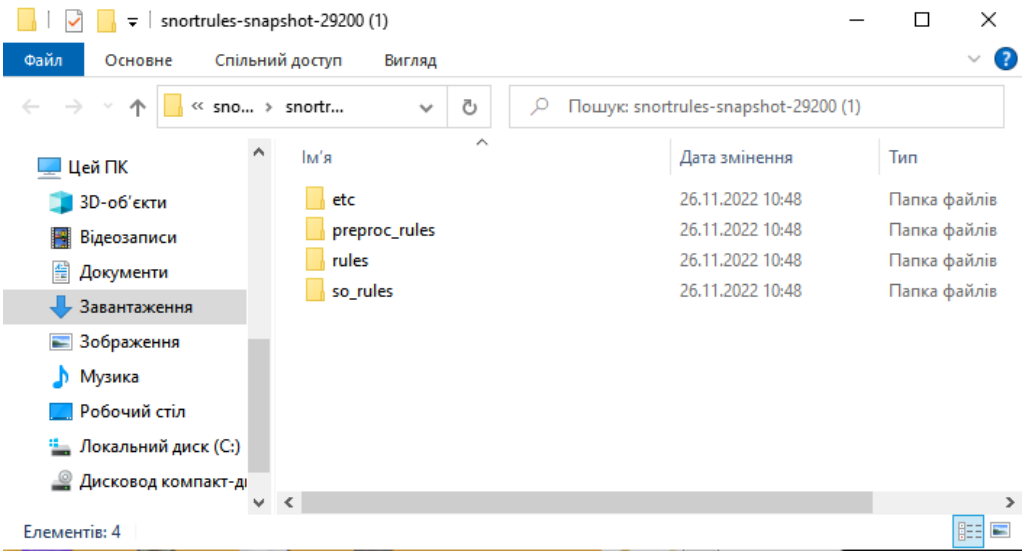


Рисунок 3.10 – Розпакування архіву правил Snort

3.2 Налаштування програми Snort

Для того, щоб перевірити, чи правильно встановлено Snort – відкриваємо командну стрічку (за допомогою адміністратора) і вказуємо спочатку локальний диск, де завантажено Snort за допомогою команди “cd C: \”. Після цього вводимо команду “cd Snort”. Переходимо до папки bin за допомогою команди “cd bin” та вводимо команду “Snort -V”, щоб перевірити, яка саме версія Snort встановлена.

На рисунку 3.11 відображено успішне встановлення Snort за допомогою командної стрічки.

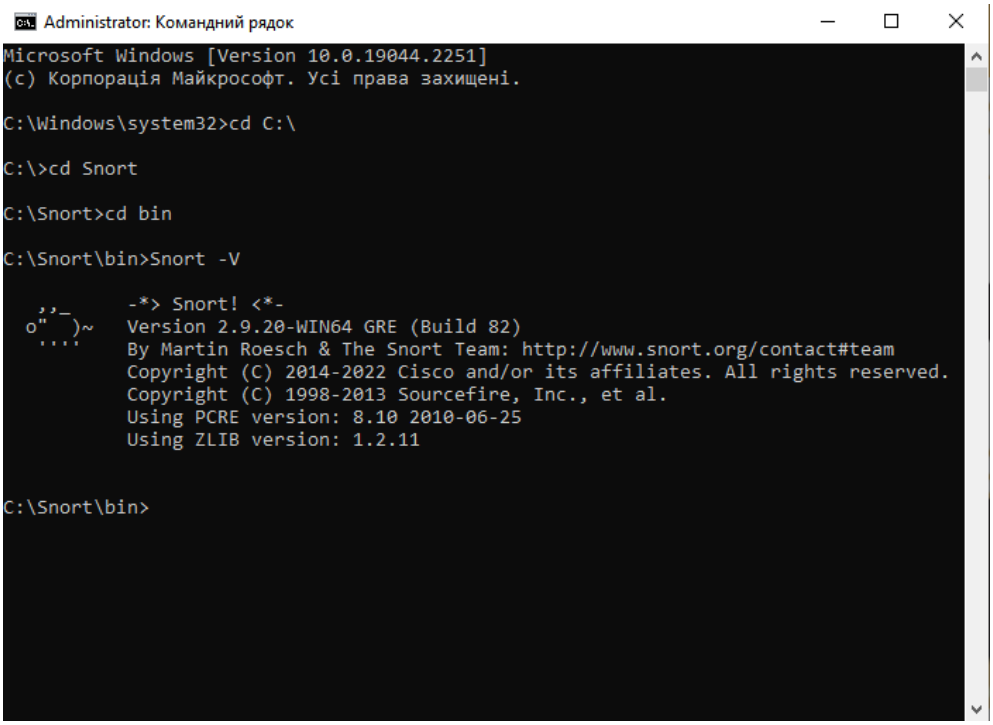


Рисунок 3.11 – Інформація про Snort у командній стрічці.

Копіюємо весь вміст папки rules із завантаженими на розпакованими правилами до c:\Snort\rules. Також копіюємо весь вміст preproc_rules папки до c:\Snort\preproc_rules, а з so_rules до папки c:\Snort\so_rules (рисунок 3.12).

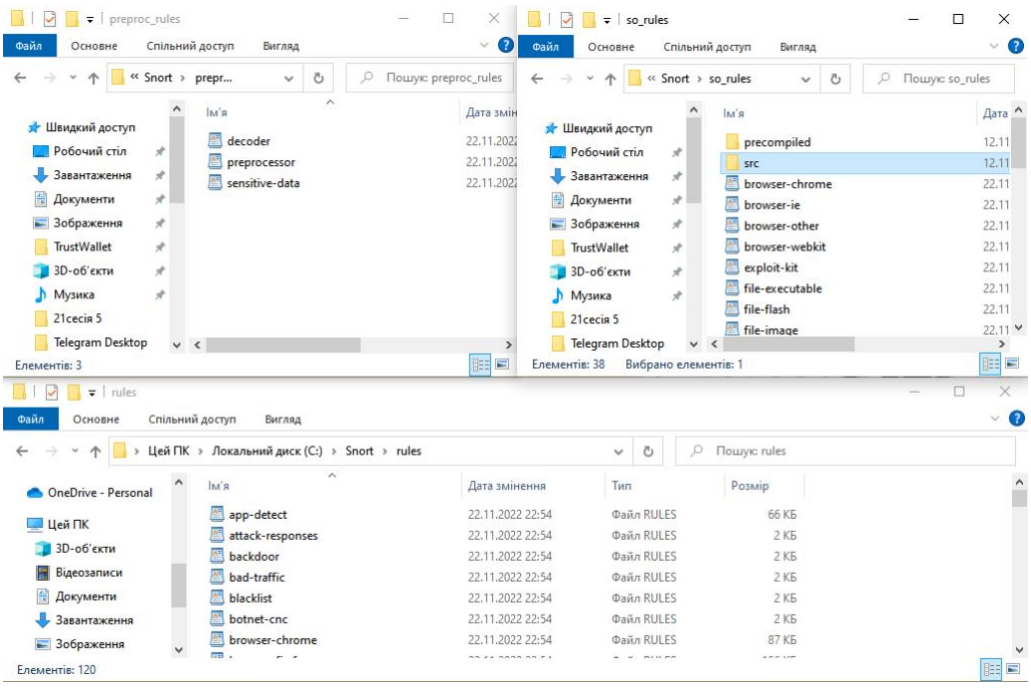


Рисунок 3.12 – Переміщення файлів із розпакованого архіву Snort

Надалі переходимо до c:/Snort/etc і відкриваємо файл Snort.conf за допомогою Notepad++. Відкривши файл прокручуємо вниз, в рядку ipvar HOME_NET any змінюємо “any” на IP-адресу пристрою. Те саме потрібно зробити у 48-ому рядку (рисунок 3.13).

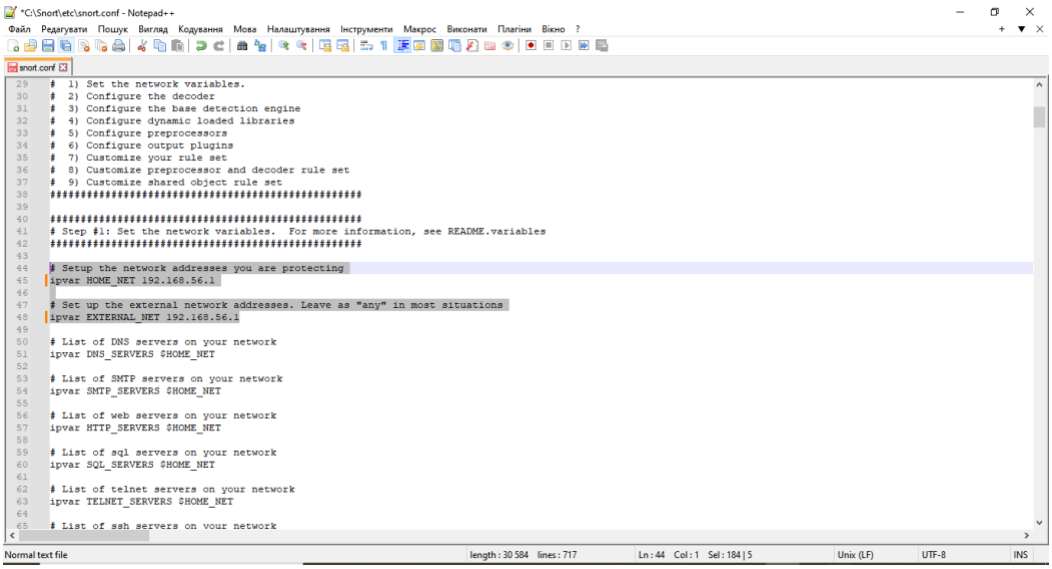


Рисунок 3.13 – Редагування IP-адреси у файлі Snort.conf

Далі, прокручуємо вниз до рядка RULE_PATH, замінюємо ../rules на c:\Snort\rules, замінюємо ../so_rules на c:\Snort\so_rules та замінюємо ../preproc_rules на c:\Snort\preproc_rules. Крім того, замінюємо WHITE_LIST_PATH та BLACK_LIST_PATH з ../rules на c:\Snort\rules (рисунок 3.14).

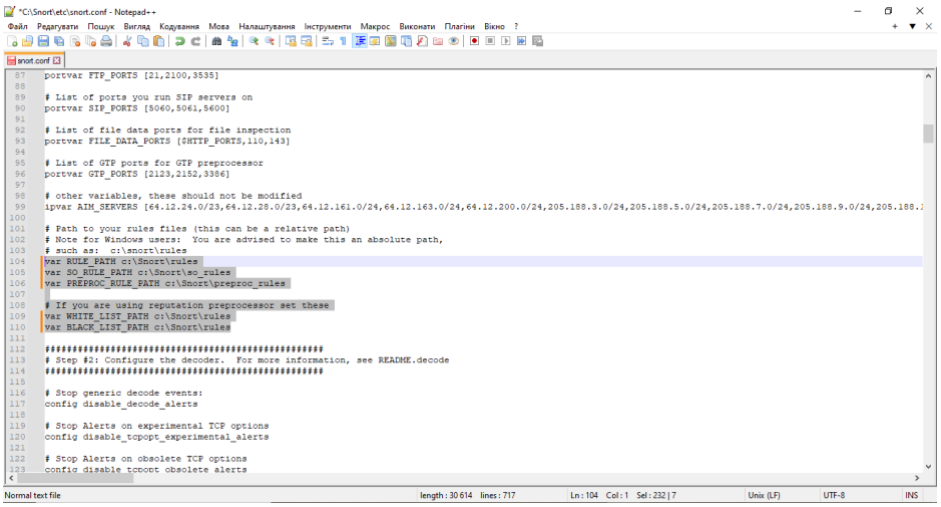


Рисунок 3.14 – Редагування шляху правил у файлі Snort.conf.

Створюємо два файли “whitelist”.conf та “blacklist”.conf перейшовши за адресою C:\Snort\rules (рисунок 3.15).

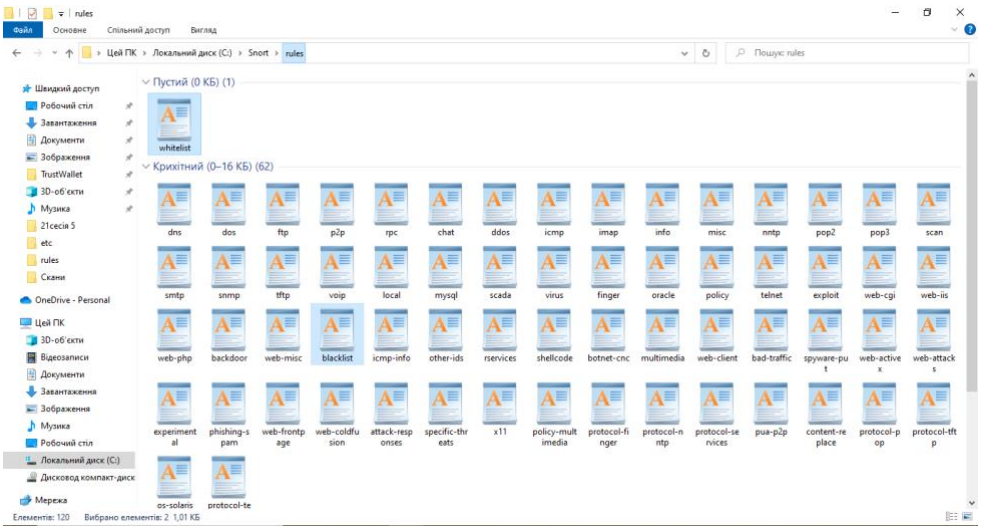


Рисунок 2.15 – Створення файлів “whitelist”.conf та “blacklist”.conf

У файлі Snort.conf замінюємо команду #config logdir: на config logdir:c:\Snort\log (рисунок 3.16).

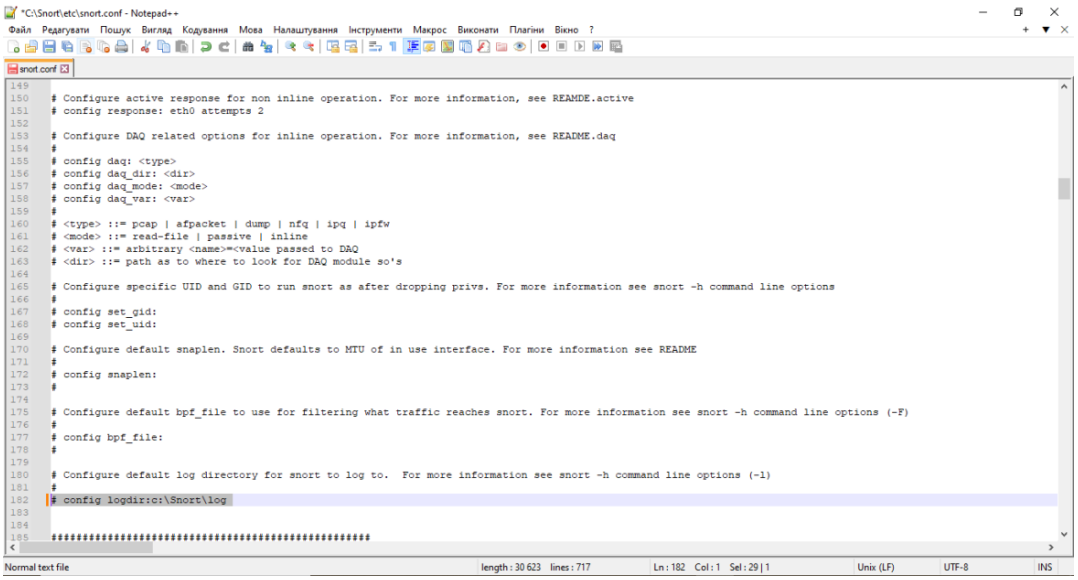


Рисунок 3.16 – Заміна команди #config logdir

Окрім цього у файлі змінюємо наступні бібліотеки: usr/local/lib/snort_dynamicpreprocessor на C:\Snort\lib\snort_dynamicpreprocessor, usr/local/lib/snort_dynamicengine/libsf_engine.so на C:\Snort\lib\snort_dynamicengine\sf_engine.dll.

Коли бібліотеки замінено – закоментовуємо їх з допомогою знаку (#). Додаємо коментар **(#)** перед усіма перерахованими препроцесорами під вбудованою нормалізацією пакетів. Вони генерують помилки під час виконання. Заміну бібліотек відображено на рисунку 3.17.

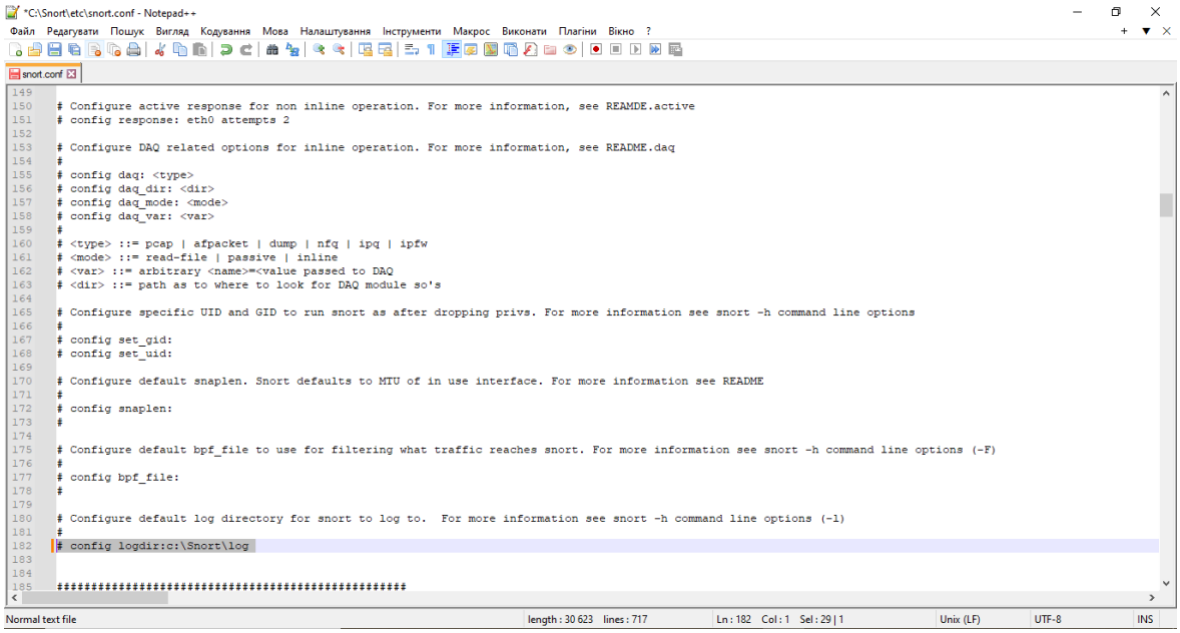


Рисунок 3.17 – Заміна бібліотек

Додаємо коментар **(#)** перед усіма перерахованими препроцесорами під вбудованою нормалізацією пакетів. Вони генерують помилки під час виконання.

Перелік закоментованих препроцесорів відображено на рисунку 3.18.

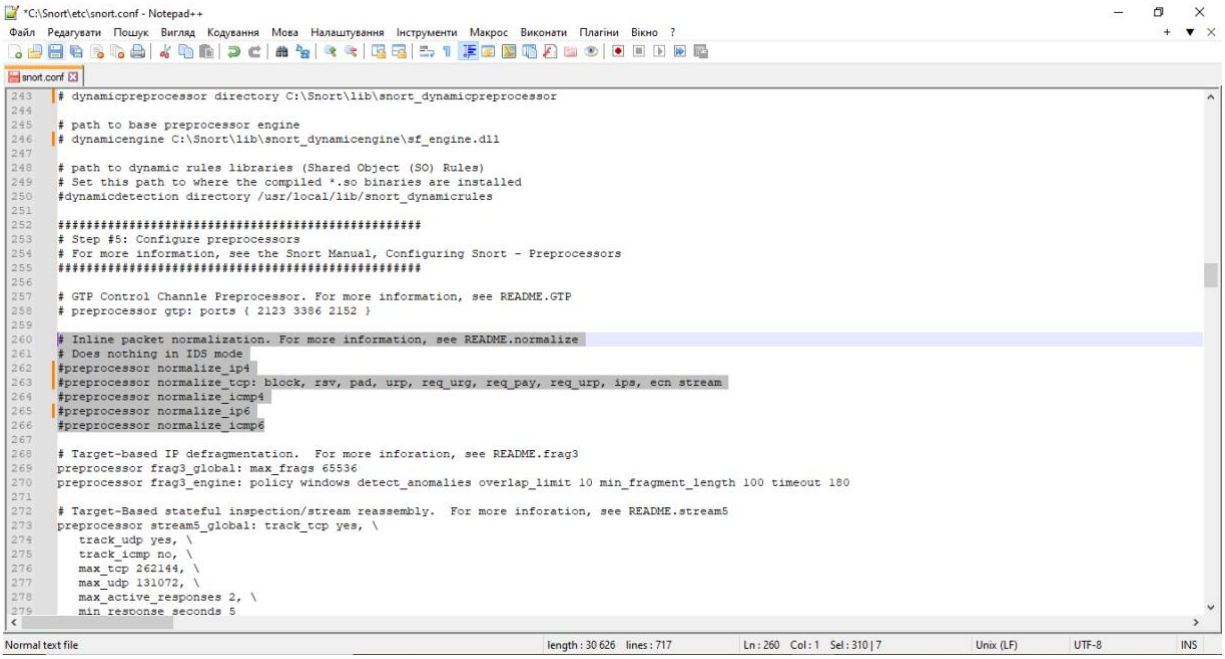


Рисунок 3.18 – Закоментовані препроцесори.

Тепер, налаштовуємо плагіни, замінивши classification.config на C:\Snort\etc\classification.config, після чого змінюємо розташування reference.config на C:\Snort\etc\reference.config. (рисунок 3.19).

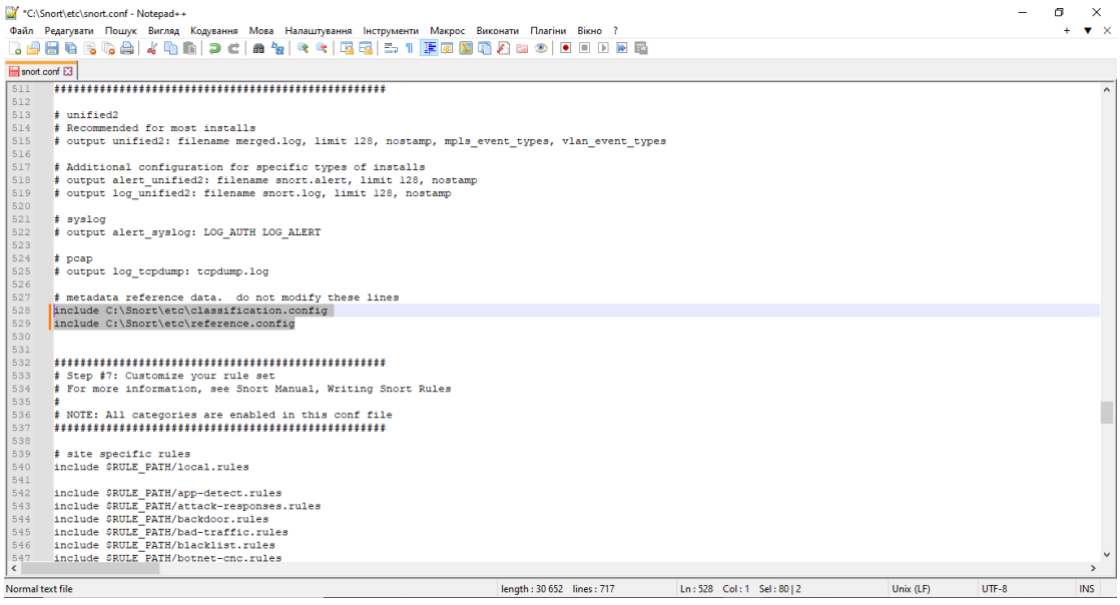


Рисунок 3.19 – Налаштування плагінів

У наступному рядку додаємо output alert_fast: alert.ids для snort, щоб скинути всі логи alert.ids (рисунок 3.20).

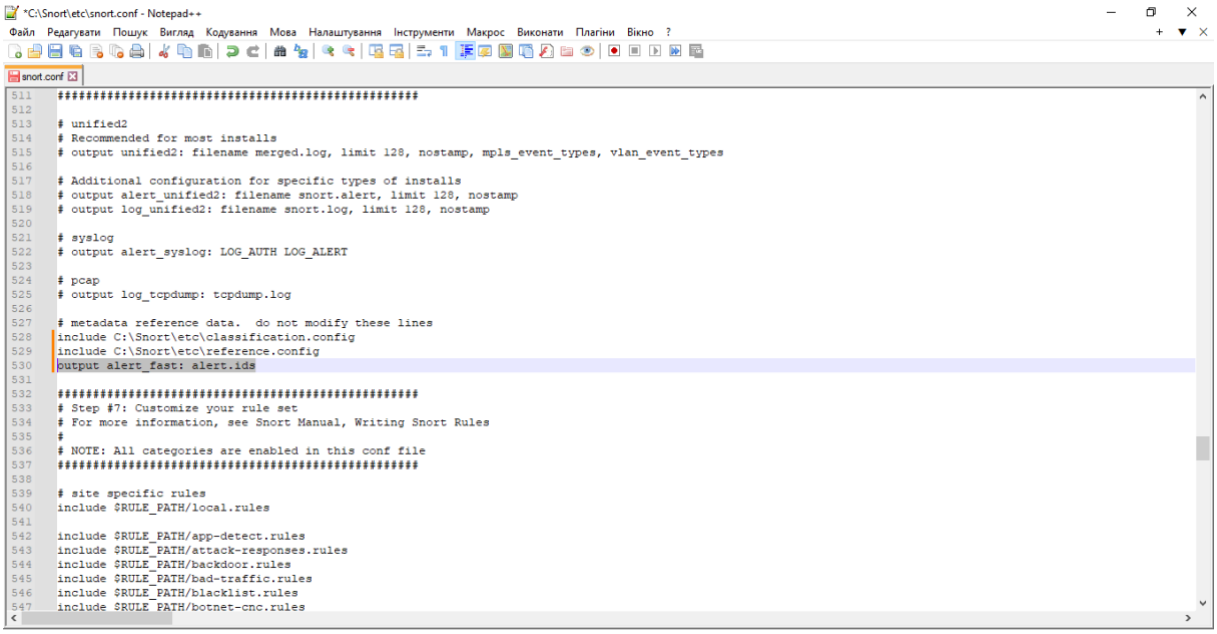


Рисунок 3.20 – Додавання команди output alert_fast: alert.ids

Оскільки, за замовчуванням рядок ipvar не розпізнається snort - замінюємо його на var. Для цього використаємо команду Ctrl+H. (рисунок 3.21).

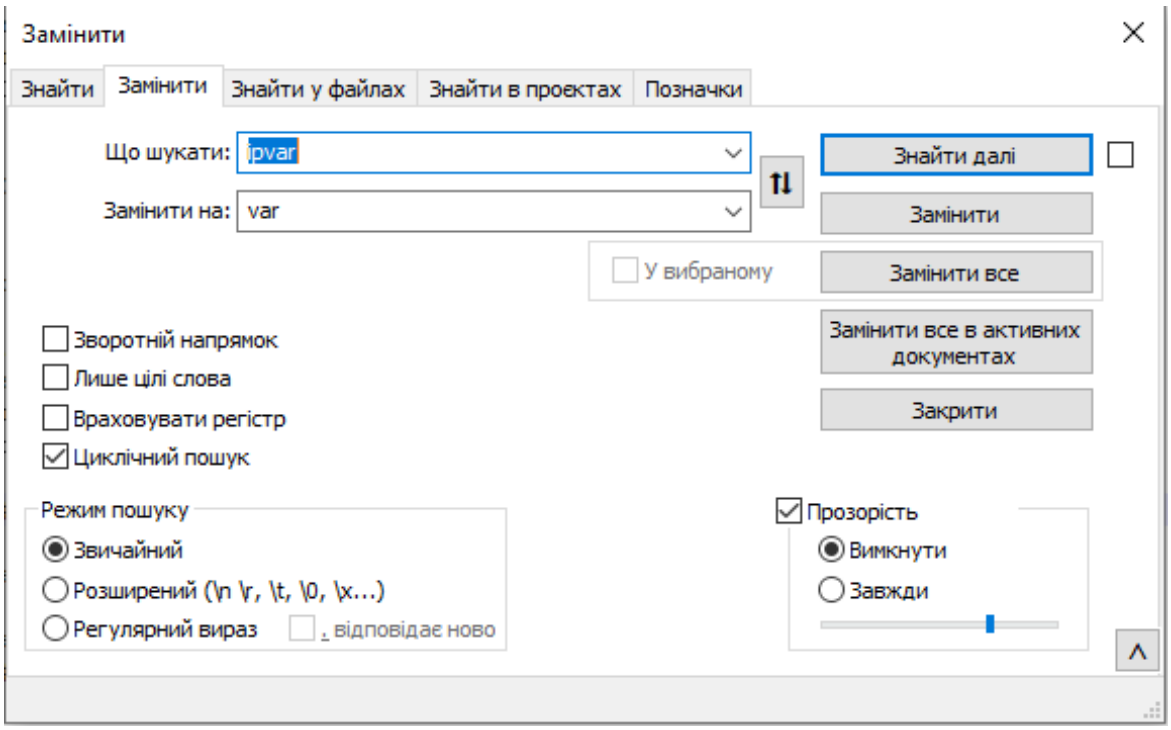


Рисунок 3.21 – Заміна рядку ipvar на var

Останнім кроком є видалення зворотної косої риски та додавання символів коментарів (#) у рядках, які можна знайти за тегом «reputation preprocessor». Коли завдання виконане – зберігаємо файл та закриваємо вікно (рисунок 3.22).

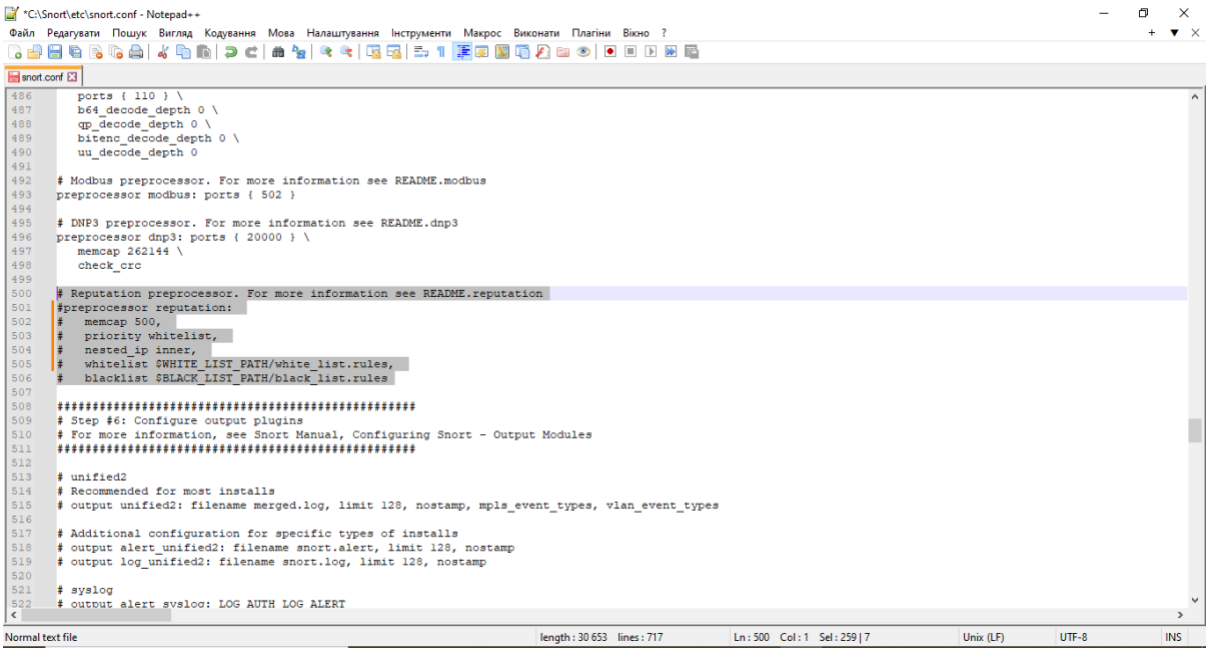


Рисунок 3.22 – Останній крок перед запуском Snort у файлі Snort.conf

Залишилось встановити правило Snort. Для цього переходимо за адресою c:\Snort\rules і відкриваємо icmp-info.rules за допомогою Notepad++ (рисунок 3.23). Наприкінці додаємо правило (обов’язково):

alert tcp any any -> any any(msg: "Testing Alert" ; sid:1000001).

У цьому випадку немає критеріїв, тому він завантажуватиме будь-який пакет ICMP, який отримує. У наведеному вище правилі ми також надаємо ідентифікатор підпису (sid), який вкрай необхідний. За домовленістю, коли пишуться власні правила Snort, які повинні починатись з 999999.

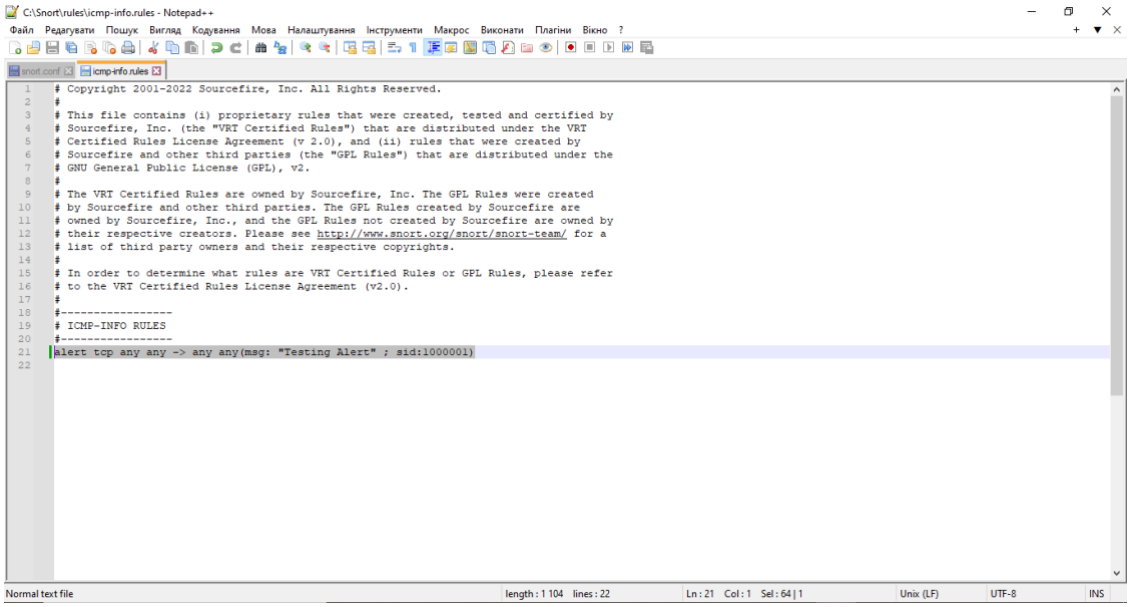


Рисунок 3.23 – Додавання правила у файлі icmp-info.rules

Щоб переконатися, що snort справді генерує сповіщення, відкриваємо командний рядок, переходимо за адресою c:\Snort\bin та пишемо команду:

snort -iX -A console -c C:\snort\etc\snort.conf -l C:\Snort\log -K ascii

де X – індексний номер вашого пристрою. Натискаємо Enter, і все готово.

На рисунку 3.24 відображено успішний запуск Snort на Windows ОС через командну стрічку [21].

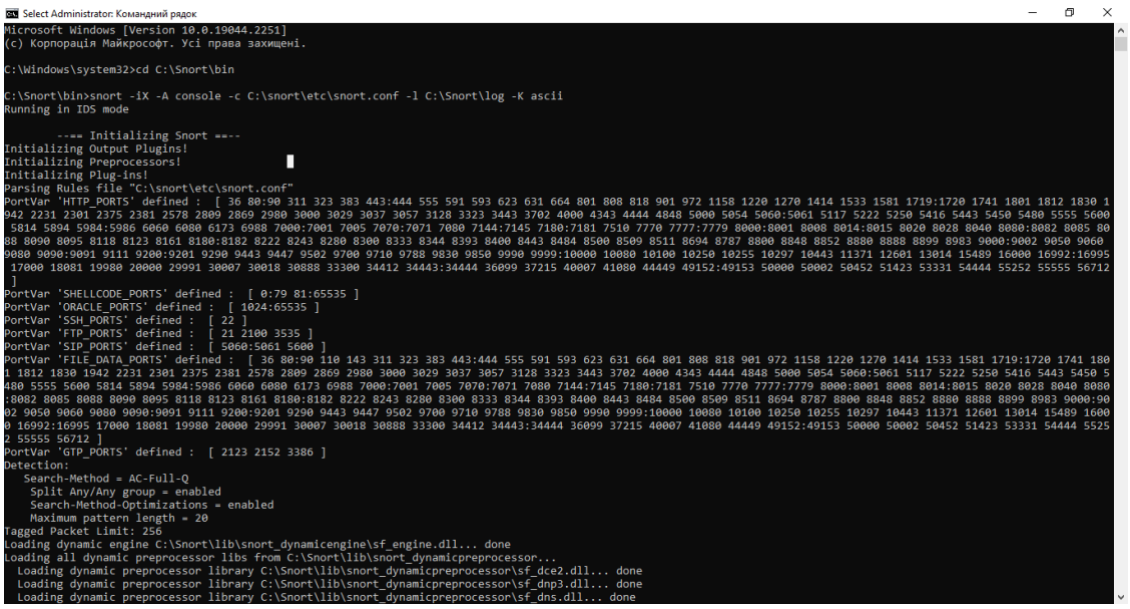


Рисунок 3.24 – Успішний запуск Snort через командну стрічку

3.3 Тестування роботи системи виявлення вторгнень

Для того, щоб перевірити, чи все працює необхідно протестувати Snort. Перший тест виконуємо через командну стрічку за допомогою команди

snort -i 1 -c c:\Snort\etc\snort.conf -T . На рисунку 3.25 відображено запуск першого тестування.

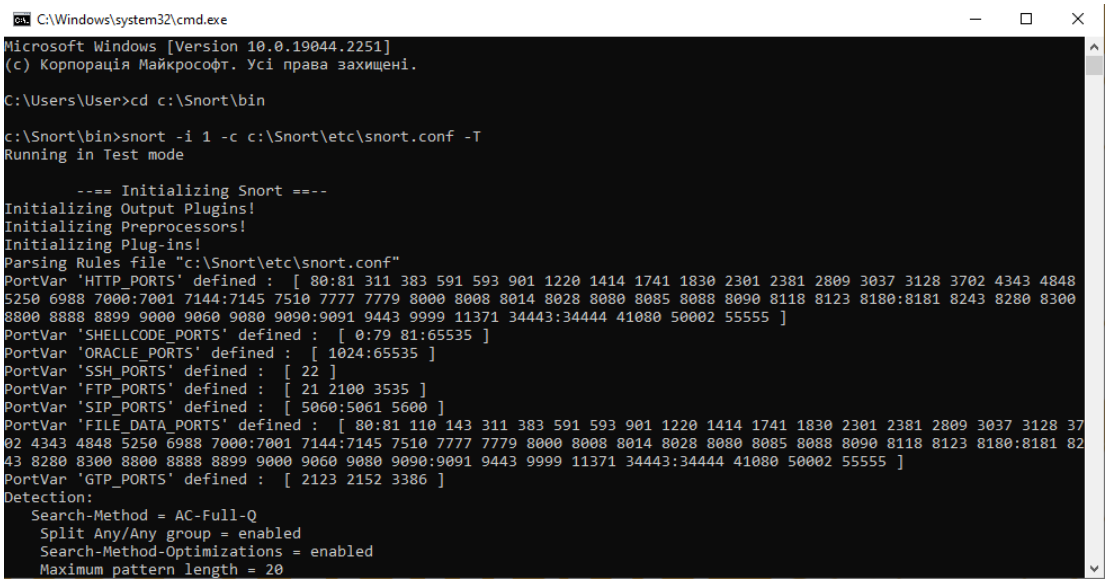


Рисунок 3.25 – Запуск першого тестування

Бачимо, що виконались всі препроцесори (рисунок 3.26)

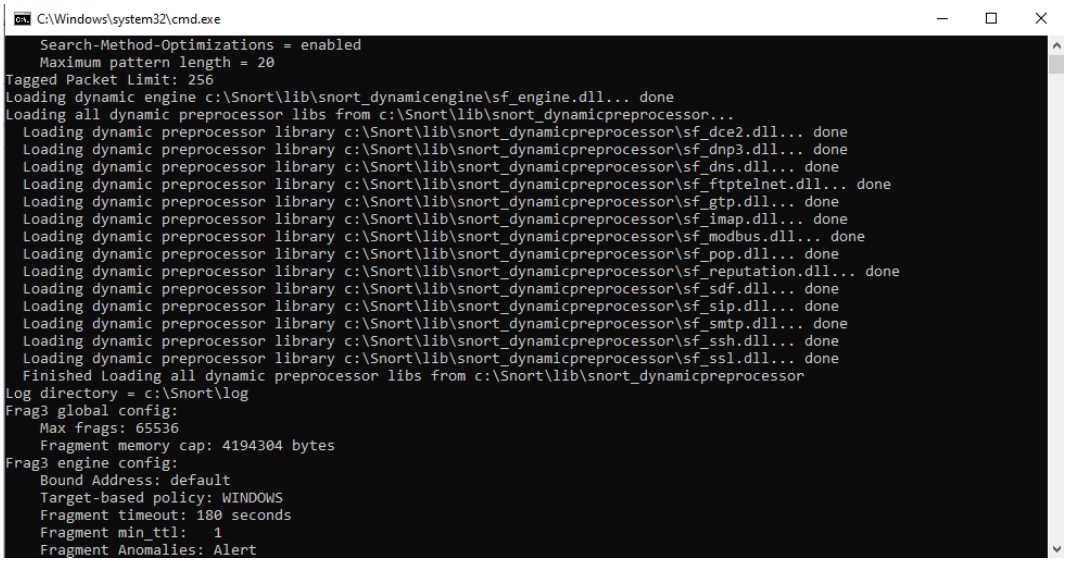


Рисунок 3.26 – Успішне виконання препроцесорів

Також, аналізуємо і бачимо, що правила успішно зчитались. Загалом прочитано 10614 правил різних видів (рисунок 3.27). Верифікація препроцесорів успішна.

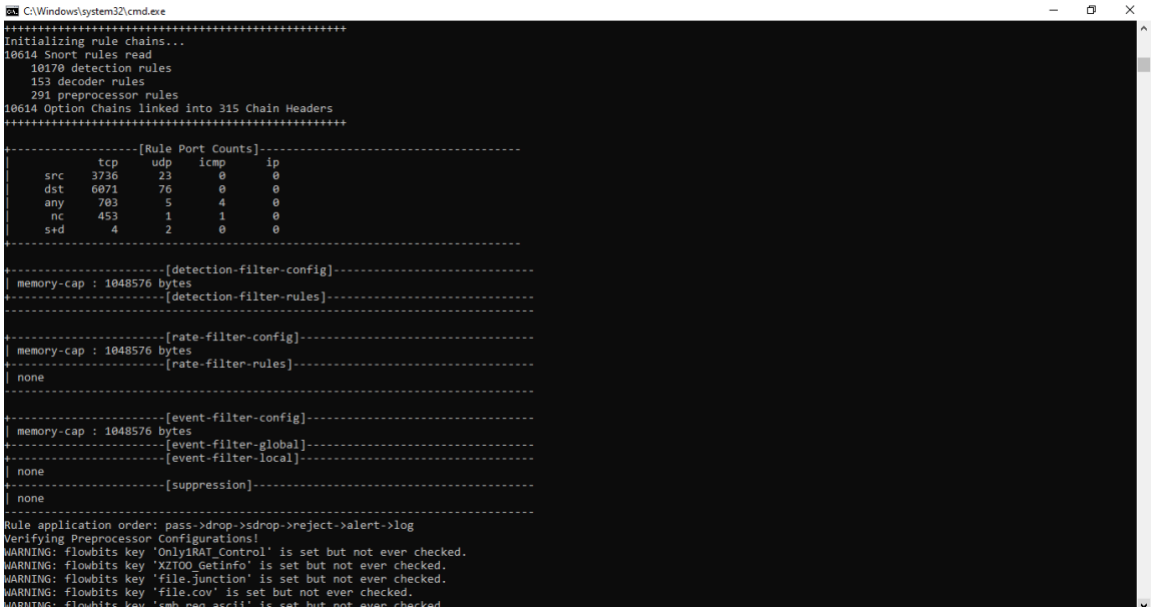


Рисунок 3.27 – Успішне виконання правил

В кінцевому результаті бачимо, що тестування відбулось успішно. (рисунок 3.28).

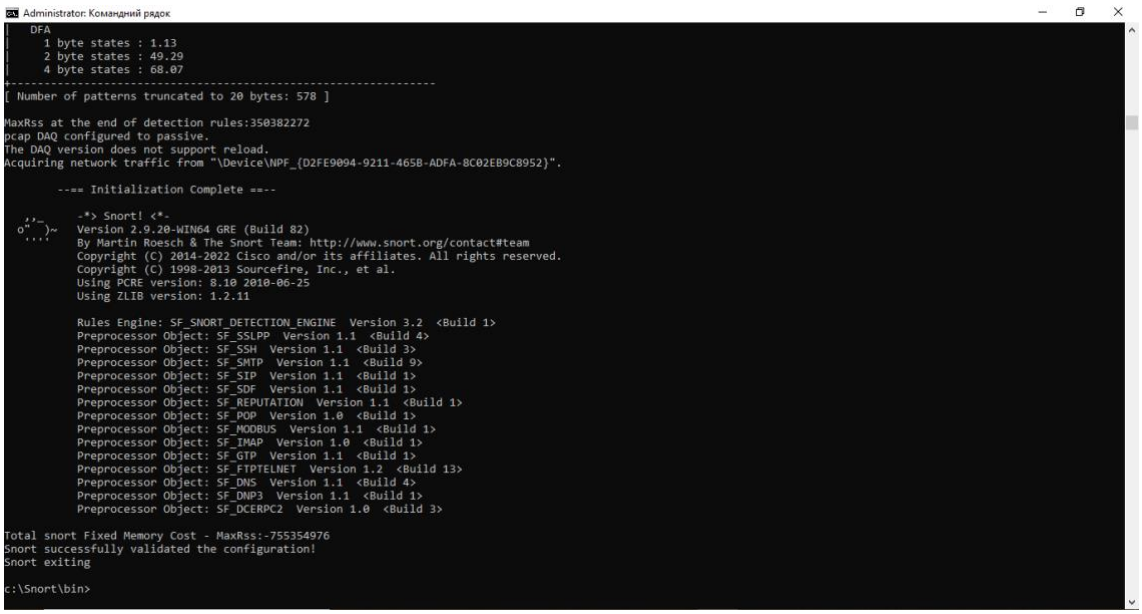


Рисунок 3.28 – Успішне перше тестування Snort

Як бачимо перше тестування пройшло успішно. Тепер, запустимо більш складне тестування у командній стрічці. Для початку – переходимо за адресою c:\Snort\rules та відкриваємо файл local.rules. Додаємо до файлу такі оповіщення:

alert icmp any any -> any any (msg:”Testing ICMP”;sid:1000001;)

alert tcp any any -> any any (msg:”Testing TCP”;sid:1000002;)

alert udp any any -> any any (msg:”Testing udp”;sid:1000003;)

На рисунку 3.29 відображено додавання оповіщень у файл local.rules.

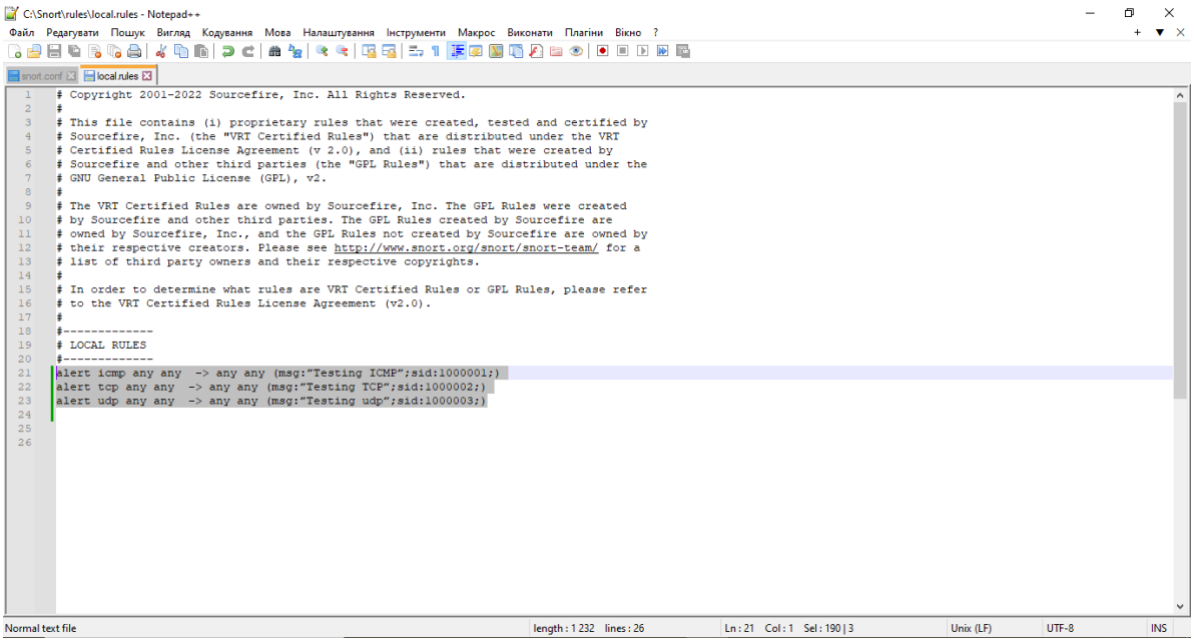


Рисунок 3.29 – Додавання оповіщень до файлу local.rules

Після цього, за допомогою командної стрічки запускаємо тест командою snort -i 1 -c c:\Snort\etc\snort.conf -A console.

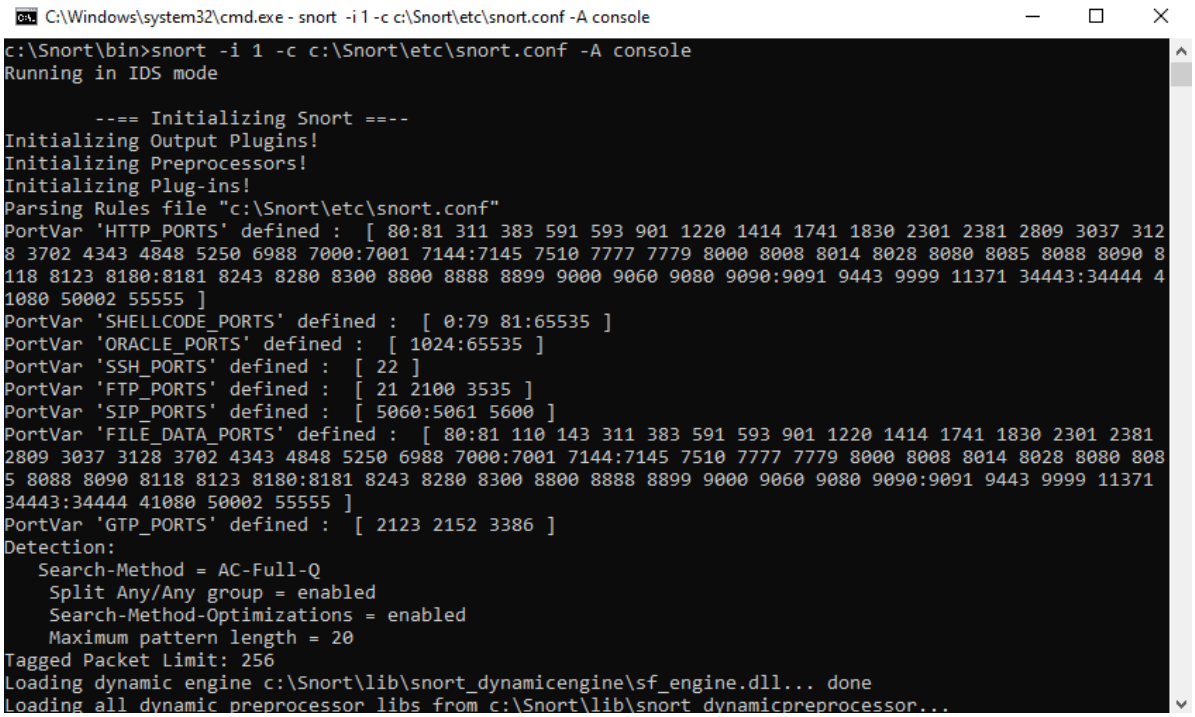


Рисунок 3.30 – Запуск другого тесту.

Після запуску другого тесту бачимо, що препроцесори також успішно запустились (рисунок 3.31).

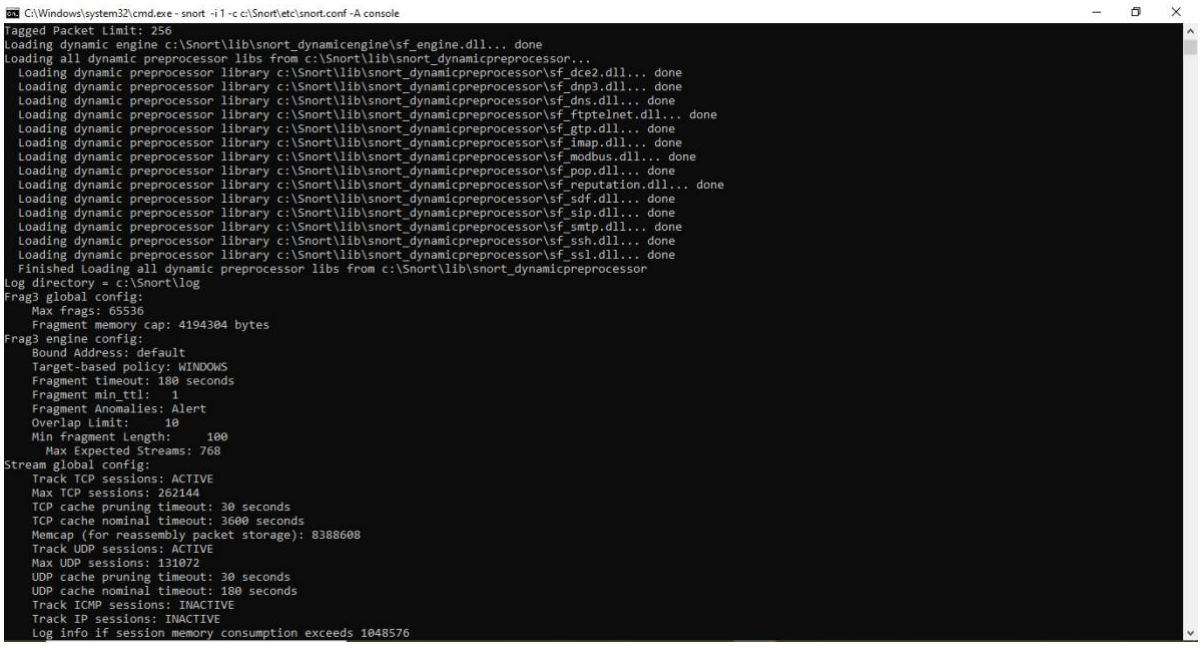


Рисунок 3.31 – Успішний запуск препроцесорів другого тесту.

На рисунку 3.32 бачимо, що правила успішно зчитались і препроцесори всі верифіковано.

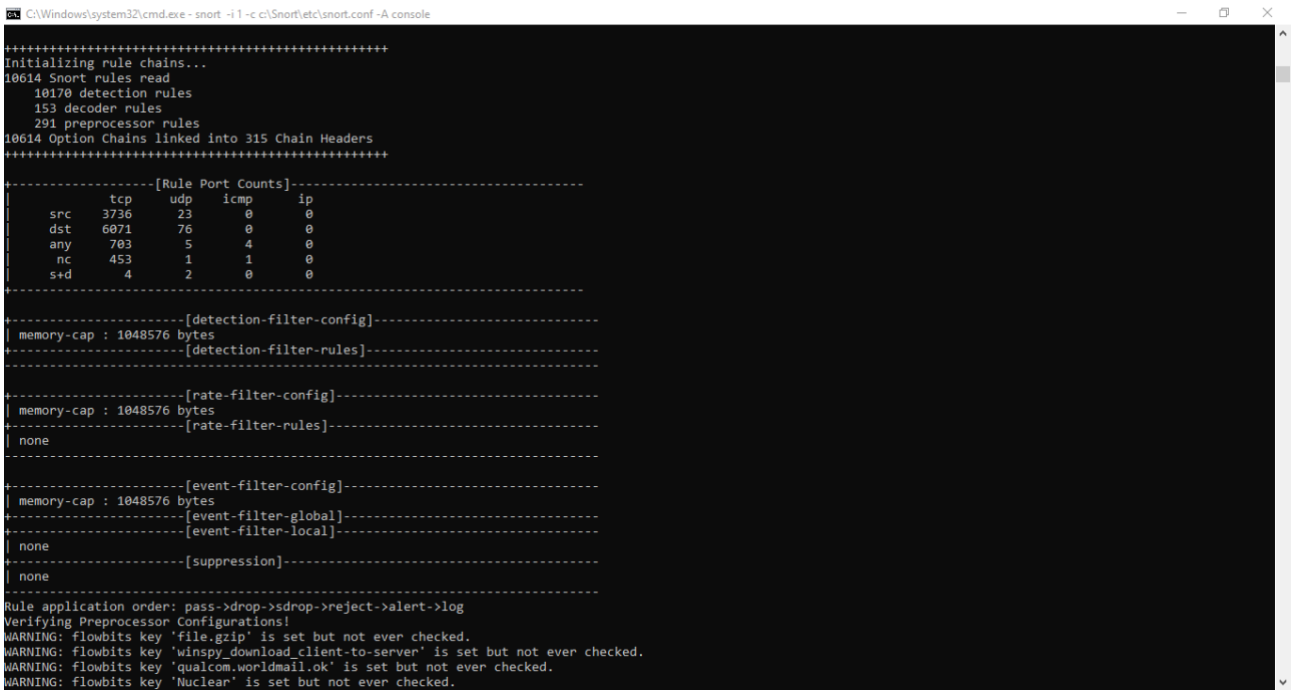


Рисунок 3.32 – Успішне зчитування правил та верифікація препроцесорів

Тепер, спостерігаємо, що ініціалізація Snort успішно відбулась (рисунок 3.33).

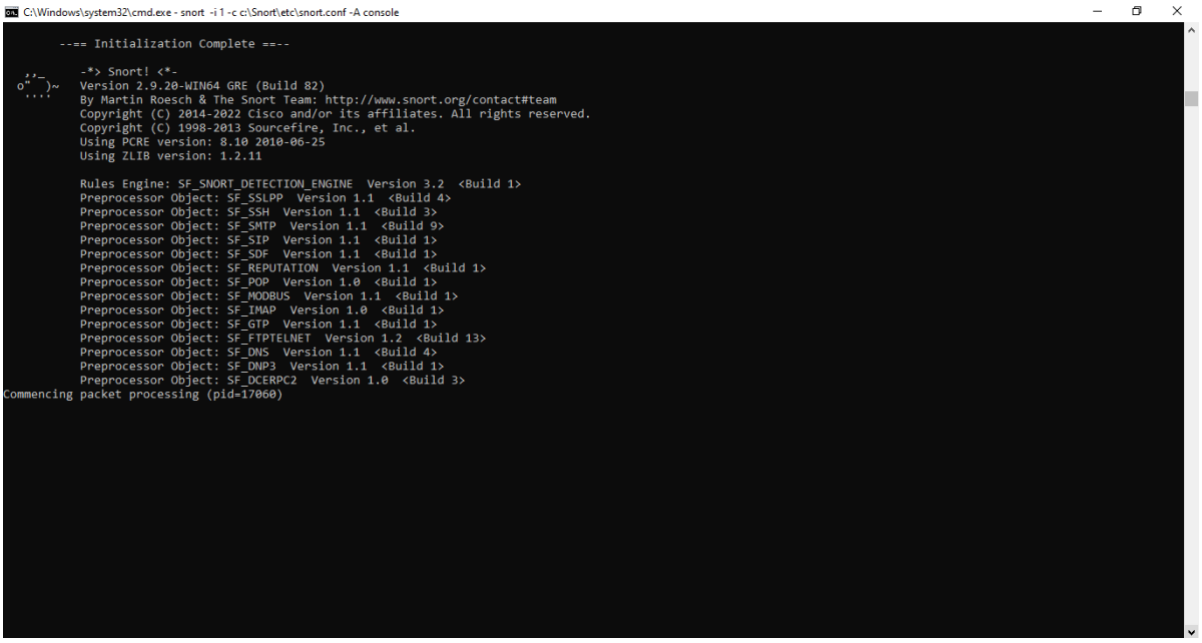


Рисунок 3.33 – Успішна ініціалізація Snort

Після цього заходимо у браузер, в нашому випадку це браузер Google Chrome та відкриваємо будь-який сайт. Наприклад: www.google.com та вводимо будь-що у пошуковій стрічці (рисунок 3.34). Як бачимо процес обробки пакетів не запустився, хоча жодних помилок не виникло.

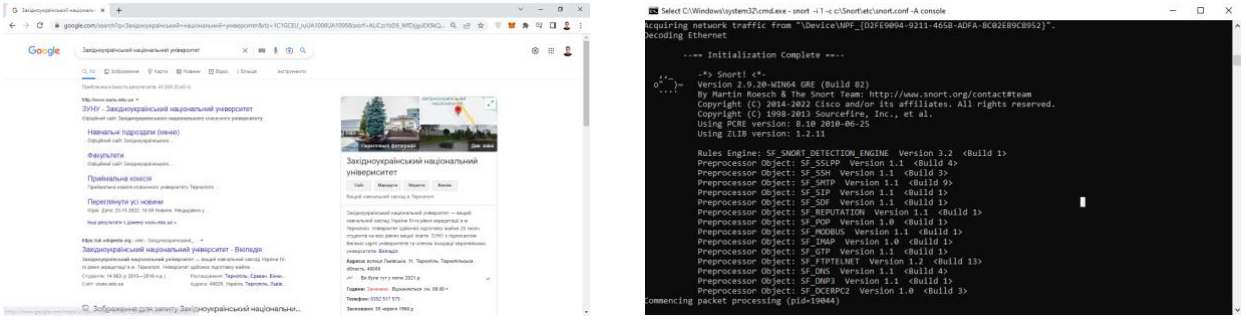
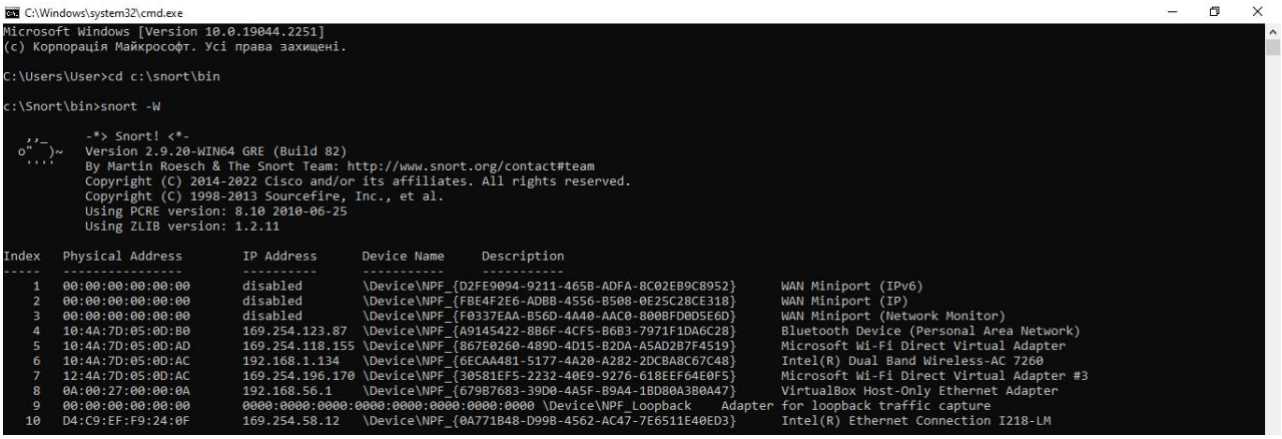


Рисунок 3.34 – Невдалий запуск процесу

Для того, щоб вирішити цю проблему – запускаємо заново термінал, заходимо за допомогою команди cd c:\snort\bin. Після цього вводимо команду Snort -W, де дізнаємось про перелік доступних інтерфейсів. Бачимо, що адреса інтернет з'єднання це 6-ий інтерфейс (рисунок 3.35).



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.19044.2251]
(c) Корпорація Майкрософт. Усі права захищені.

C:\Users\User>cd c:\snort\bin

c:\Snort\bin>snort -W

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.20-WIN64 GRE (Build 82)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014-2022 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using PCRE version: 8.10 2010-06-25
           Using ZLIB version: 1.2.11

Index   Physical Address        IP Address      Device Name     Description
-----   ----------------        ----------      -----------     -----------
    1   00:00:00:00:00:00       disabled        \Device\NPF_{D2FE9094-9211-465B-ADFA-8C02EB9C8952}      WAN Miniport (IPv6)
    2   00:00:00:00:00:00       disabled        \Device\NPF_{FBE4F2E6-ADBB-4556-B508-0E25C28CE318}      WAN Miniport (IP)
    3   00:00:00:00:00:00       disabled        \Device\NPF_{F0337EAA-B56D-4A40-AAC0-800BFD0D5E6D}      WAN Miniport (Network Monitor)
    4   10:4A:7D:05:0D:B0       169.254.123.87  \Device\NPF_{A9145422-8B6F-4CF5-B6B3-7971F1DA6C28}      Bluetooth Device (Personal Area Network)
    5   10:4A:7D:05:0D:AD       169.254.118.155 \Device\NPF_{867E0260-489D-4D15-B2DA-A5AD2B7F4519}      Microsoft Wi-Fi Direct Virtual Adapter
    6   10:4A:7D:05:0D:AC       192.168.1.134   \Device\NPF_{6ECAA481-5177-4A20-A282-2DCBA8C67C48}      Intel(R) Dual Band Wireless-AC 7260
    7   12:4A:7D:05:0D:AC       169.254.196.170 \Device\NPF_{30581EF5-2232-40E9-9276-618EEF64E0F5}      Microsoft Wi-Fi Direct Virtual Adapter #3
    8   0A:00:27:00:00:0A       192.168.56.1    \Device\NPF_{679B7683-39D0-4A5F-B9A4-1BD80A3B0A47}      VirtualBox Host-Only Ethernet Adapter
    9   00:00:00:00:00:00       0000:0000:0000:0000:0000:0000:0000:0000 \Device\NPF_Loopback    Adapter for loopback traffic capture
   10   D4:C9:EF:F9:24:0F       169.254.58.12   \Device\NPF_{0A771B48-D99B-4562-AC47-7E6511E40ED3}      Intel(R) Ethernet Connection I218-LM
```

Рисунок 3.35 – Перелік доступних інтерфейсів Snort.

Після цього – змінюємо команду у терміналі на 6-ий інтерфейс та запускаємо заново процес: snort -i 6 -c c:\Snort\etc\snort.conf -A console.

В результаті отримали сповіщення у командній стрічці про процеси, які відбуваються при запуску браузера та відкритті сторінки. Успішний процес тестування відображений на рисунку 3.36.

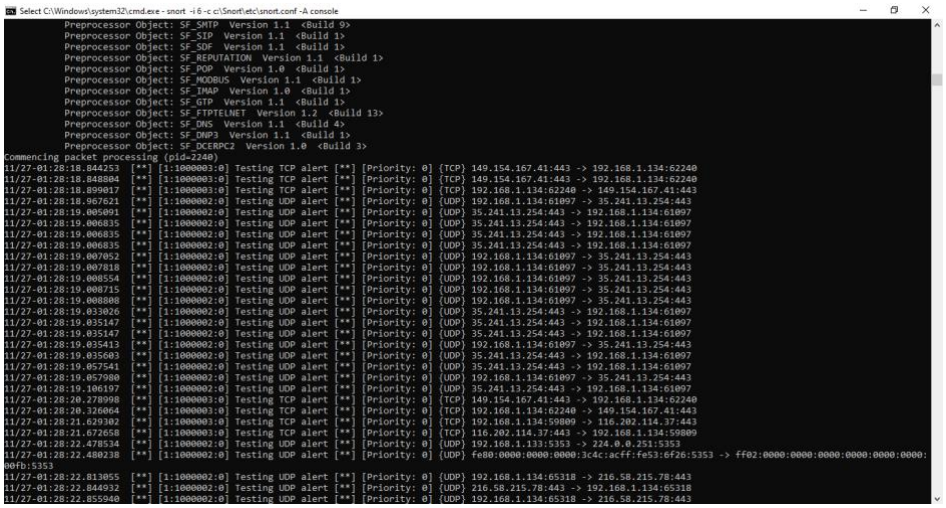


Рисунок 3.36 – Успішний процес другого тесту.

Для того, щоб подивитись статистику, відкриваємо термінал та натискаємо клавіші Ctrl+C. Тут відображається статистика по всіх параметрах Snort та процесах, які відбулись. (рисунок 3.37).

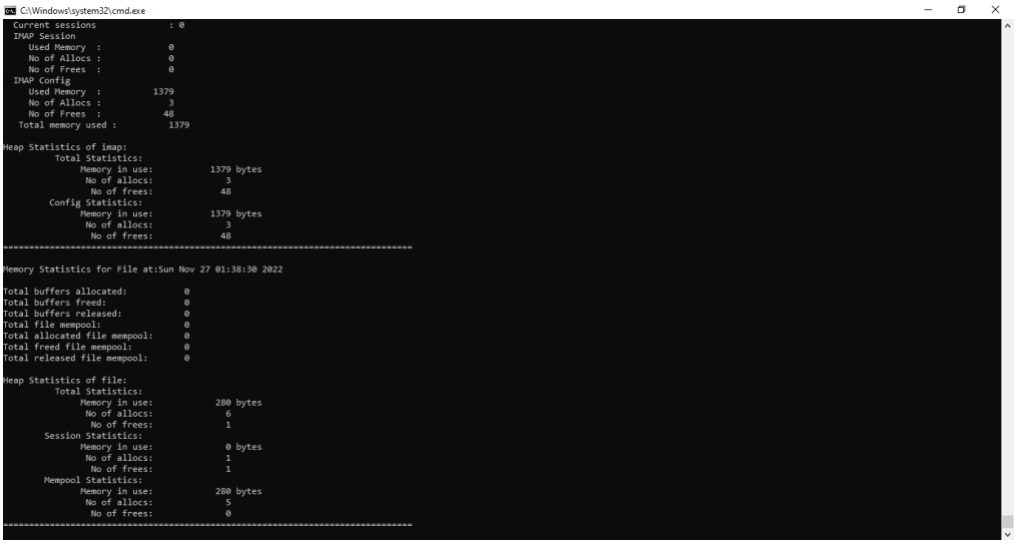


Рисунок 3.37 – Статистика Snort.

У цьому розділі ми розглянули та попрактикувались, як саме запускати на налаштувати Snort на Windows ОС. Провели успішні два тестування, в яких розглянули, яким чином користувач отримує оповіщення у текстовому вигляді в командній стрічці. Також, опрацювали помилку, яка виникла в ході виконання тестування.

ВИСНОВКИ

В кваліфікаційній роботі розв’язано актуальну задачу підвищення ефективності роботи системи виявлення вторгнень. При цьому отримано наступні результати.

1. Проведено класифікація систем виявлення вторгнень та аналіз платформ з відкритим кодом. Обґрунтовано вибір платформ Snort, в якості інструменту для реалізації системи виявлення вторгнень.

2. Проведено порівняння наявних систем виявлення вторгнень і системи виявлення вторгнень Snort.

3. Розроблено структуру системи виявлення вторгнень на базі інструменту Snort та правил виявлення вторгнень.

4. Досліджено способи написання правил інструменту Snort.

5. Встановлено та налаштовано систему виявлення вторгнень Snort на операційній системі Windows та успішно проведено тестування системи.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Gupta A., Sharma L. S. A categorical survey of state-of-the-art intrusion detection system-Snort. Int. J. Inf. Comput. Secur., 2020, 13.3/4: 337-356.

2. Порівняльний аналіз систем виявлення і запобігання вторгнень (Електронний ресурс). - Режим доступу: https://openarchive.nure.ua/bitstream/document/10555/1/RYKOV.pdf

3. Understanding and Configuring Snort Rules. (Електронний ресурс). – Режим доступу: https://www.rapid7.com/blog/post/2016/12/09/understanding-and-configuring-snort-rules/

4. What is Snort?. (Електронний ресурс). - Режим доступу: https://www.fortinet.com/resources/cyberglossary/snort

5. 10 найкращих систем виявлення вторгнень. (Електронний ресурс). - Режим доступу: https://uk.myservername.com/top-10-best-intrusion-detection-systems

6. Roesch, M. (1999). Snort: Lightweight intrusion detection for networks. vol. 229. Santa Clara, CA: Stanford Telecommunications Inc.

7. Norton, M., & Roelker, D. (2002). SNORT 2.0: Hi-performance multi-rule inspection engine. Columbia: Sourcefire Network Security Inc.

8. Caswell, B., Beale, J., & Baker, A. (2007). *Snort IDS and IPS toolkit*. New York: Syngress.

9. Tjhai, G. C., Papadaki, M., Furnell, S. M., & Clarke, N. L. Investigating the problem of IDS false alarms: An experimental study using Snort. In International Information Security Conference.

10. Zhou, Z., Zhongwen, C., & Tiecheng, Z. (2010). The study on network intrusion detection system of Snort. In 2010 2nd International Conference, IEEE.

11. Zhang, X., Li, C., Zheng, W., "Intrusion Prevention System Design", The Fourth International Conference on Computer and InformationTechnology (CIT'04), 2004

12. «Методи ідентифікації аномальних станів для систем виявлення вторгнень», 22.02.2019. (Електронний ресурс). - Режим доступу: https://dspace.nau.edu.ua/handle/NAU/38772/

13. Peng Lu. (2013). “A lightweight intrusion detection system snort. Silicon valley” (4), 57-57

14. Sagala, A. . (2016). “Automatic SNORT IDS rule generation based on honeypot log.”

15. Kuvaiskii, D., Chakrabarti, S., & Vij, M. (2018). “Snort intrusion detection system with intel software guard extension (intel sgx).”

16. Tjhai, G. C. , Papadaki, M. , Furnell, S. M. , & Clarke, N. L. . (2008). “Investigating the problem of IDS false alarms: An experimental study using Snort. Proceedings.”

17. Park, W., & Ahn, S. (2017). “Performance comparison and detection analysis in snort and suricata environment. Wireless Personal Communications”

18. Jianying Liang. (2015). “Based on the snort network intrusion defense explored.”

19. SNORT IDS. Електронний ресурс. – Режим доступу: http://www.snort.org/

20. Гавриляк М.В., Цаволик Т.Г., Інгнатьєв І.В. Функції та переваги системи виявлення вторгнень на базі SNORT. Матеріали наукової конференції «Автоматизація та комп’ютерно-інтегровані технології» (АКІТ – 2022), Тернопіль, 2022. – С.97– 99.

21. Гавриляк М.В. Виявлення вторгнень на основі правил SNORT. Матеріали наукової конференції «Кібербезпека та комп’ютерно-інтегровані технології» (КБКІТ – 2022), Тернопіль, 2022. – С. 70–72.

ДОДАТОК А

Копії публікацій

ЗАХІДНОУКРАЇНСЬКИЙНАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ПРИКАРПАТСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТЕФАНИКА
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВОДНОГО ГОСПОДАРСТВА ТА ПРИРОДОКОРИСТУВАННЯ
НАЦІОНАЛЬНИЙ ТРАНСПОРТНИЙ УНІВЕРСИТЕТ
НАДВІРНЯНСЬКИЙ КОЛЕДЖ НТУ
ГАЛИЦЬКИЙ КОЛЕДЖ ІМ. В. ЧОРНОВОЛА

Проблемно-наукова міжгалузева конференція

АВТОМАТИЗАЦІЯ ТА КОМП'ЮТЕРНО-ІНТЕГРОВАНІ ТЕХНОЛОГІЇ

(АКІТ – 2022)

21—23 лютого 2022 року

Тернопіль

АВТОМАТИЗАЦІЯ ТА КОМП'ЮТЕРНО-ІНТЕГРОВАНІ ТЕХНОЛОГІЇ

БЕЗПЕКА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

УДК 004.056

Гавриляк М.В.[1], Цаволик Т.Г.[1], Ігнатєв І.В.[1]

Західноукраїнський національний університет

ФУНКЦІЇ ТА ПЕРЕВАГИ СИСТЕМИ ВИЯВЛЕННЯ ВТОРГНЕНЬ НА БАЗІ SNORT

Вступ. Для кожного користувача в інтернеті головним пріоритетом є захист конфіденційної інформації. Система виявлення вторгнень в режимі реального часу є правильним рішенням для кожного користувача. Вона може моніторити стан мережі, контролювати потік мережі та діяльності, видавати попередження, записувати інформацію до бази даних, на основі якої аналізує вторгнення та створює журнали вторгнень. В даній статті розглянемо інструмент системи виявлення вторгнень Snort, його функції та переваги.

Мета: дослідження функцій та переваг інструменту Snort.

1. Функції SNORT

Snort – це програмне забезпечення з відкритим кодом і невеликим обсягом програмне забезпечення, розроблене Мартіном Решем у 1998 році для виявлення та запобігання вторгнень у мережу. Програмне забезпечення Snort діє як аналізатор пакетів, реєстратор пакетів або як засіб виявлення вторгнень у мережу та система профілактики. Snort читає мережеві пакети та відображає їх на консолі. Якщо консоль є у режимі аналізатора, він одночасно реєструватиме пакети на диск вибору реєстратора пакетів. Також, він буде стежити за мережею трафіку і аналізом трафіку відповідно до визначеного набору правил користувачем, під час виявлення вторгнення в мережу і системи профілактики. Його було розроблено для Linux і Windows, щоб виявляти нові загрози.

Набори правил перетворюють Snort на IDS – без них це просто аналізатор пакетів. Існує два типи наборів правил: набори правил спільноти та набори правил для підписників Snort. Набори правил спільноти безкоштовні.

Архітектуру Snort зображено на рисунку 1 [1].

Рисунок 1. Архітектура інструменту Snort

Snort має можливість робити аналіз одночасного трафіку та реєстрації пакетів мережі Інтернет-протоколу (IP). Це також може стосуватись аналізу протоколів, пошуку вмісту та зіставлення. Snort використовує виявлення вторгнення на основі сигнатур, а також методи, засновані на аномаліях, на які можна покластися при налаштуванні правил користувача або підписів, які отримані з баз даних, таких як Emerging Threats.

2. Переваги інструменту Snort

Якщо порівнювати Snort із іншими інструментами – можна чітко помітити значні переваги, які наведені в таблиці 1 [3].

Таблиця 1 - Порівняння різних інструментів системи виявлення вторгнень

Ім'я інструменту	Провайдер	Тип	Опис	Платформа
SNORT	Cisco system	NIDS, NIPS	Може виявити Dos, CGI, вторгнення, сканування портів, SMB і атаки рівня. SNORT має здатність створювати одночасний аналіз трафіку та пакет Вхід у мережі Інтернет-протоколу (IP)	Linux, windows, Крос-платформа
SURICATA	Open information security foundation	NIDS, NIPS	Автоматичне визначення протоколу. Процес узгодження та сумісності з SNORT.	Linux, unix, MAC, windows та інші
Bro IDS	Vern Paxson	NIDS, AIDS	Сумісний з SNORT	Linux, MAC OS , FreeBSD
OpenWIPS-ng	Aircrack-NG	NIPS	Openwips-Ng є відкритим кодом і модульною бездротовою IPS	Linux
Security Onion	-	NIDS	Містить Snort, Suricata, Sguil, Squert, Snorby, Bro, Networkminer, Xplico, та багато інших інструментів безпеки	Linux
OSSEC	Daniel B. Cid	HIDS	Має потужну кореляцію і механізм аналізу, інтегрований аналіз журналів.	Крос-платформа
FRAGROUTE	Dug Song	NIDS	Цей інструмент має хорошу репутацію. Допомога в тестуванні мережевих вторгнень системи виявлення, брандмауери та базова поведінка з TCP/IP.	Linux

Серед переваг інструменту Snort варто зазначити наступні [2, 3]:

- дозволяє зберігати пакети у файл для аналізу в інших інструментах;
- Snort інструменти дуже прості в установці і запуску, оскільки користувач може налаштувати правила;
- інструмент створює спливаючі вікна та оповіщення. Ці спливаючі вікна контролює Служба Windows Messenger;
- оскільки Snort є інструментом з відкритим вихідним кодом, його можна легко налаштувати відповідно до вимог будь-якої компанії, а ще він безкоштовний;
- доступний на двох основних платформах Windows та Linux, є кросплатформним інструментом, тож ним можна користуватись одночасно в цих двох операційних системах;
- однією з ключових особливостей Snort є створення нового підпису для відстеження вразливостей;
- він зберігає записи пакетів даних зі своєї IP-адреси, у зрозумілій для людини формі;
- його можна використовувати для контролю та моніторингу як домашнього DSL-з'єднання, так і корпоративного веб-сайту;
- Snort можна використовувати як пастку для запису небажаної присутності трафіку, який не повинен бути знайденим у мережі;
- Snort може ідентифікувати переповнення буфера, CGI-атаки, приховане сканування портів, запити NetBIOS та вразливостей системи, NMAP та інші сканери портів і DDOS-клієнтів. Він сповіщає користувачів про ці атаки та шкідливі функції.

Однак, окрім переваг, Snort має і декілька недоліків, зокрема:

- вразливий до DoS-атак зсередини мережі;
- процес захоплення пакетів дуже повільний і має лише середній рівень підтримки високошвидкісної мережі.
- якщо знадобляться будь-які корпоративні налаштування, то це потребуватиме певних витрат.

Висновки. Проведені дослідження дозволяють зробити висновок, що інструмент Snort є багато функціональним, легким у встановленні та зрозумілим для користувача. Саме цей інструмент має значну кількість переваг, що робить його унікальним і зручним у порівнянні з іншими інструментами системи виявлення вторгнень.

Перелік використаних джерел.

1. Gupta A., Sharma L. S. A categorical survey of state-of-the-art intrusion detection system-Snort. Int. J. Inf. Comput. Secur., 2020, 13.3/4: 337-356.

2. What Is SNORT? (Електронний ресурс). – Режим доступу: https://www.fortinet.com/resources/cyberglossary/snort

3. 10 найкращих систем виявлення вторгнень (IDS). (Електронний ресурс). – Режим доступу: https://uk.myservername.com/top-10-best-intrusion-detection-systems


КІБЕРБЕЗПЕКА ТА
КОМП'ЮТЕРНО-ІНТЕГРОВАНІ
ТЕХНОЛОГІЇ
КБКІТ-2022
науково-практична конференція
молодих вчених
аспірантів та студентів
м.Тернопіль

ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ ПОЛІТЕХНІЧНИЙ УНІВЕРСИТЕТ
ГАЛИЦЬКИЙ ФАХОВИЙ КОЛЕДЖ ІМ. В. ЧОРНОВОЛА

КІБЕРБЕЗПЕКА ТА КОМП'ЮТЕРНО-ІНТЕГРОВАНІ ТЕХНОЛОГІЇ (КБКІТ – 2022)

науково-практична конференція
молодих вчених, аспірантів та студентів

29–31 серпня 2022
Тернопіль

УДК 004.056

Гавриляк М.В.

Західноукраїнський національний університет

ВИЯВЛЕННЯ ВТОРГНЕНЬ НА ОСНОВІ ПРАВИЛ SNORT

Вступ. Для інструменту системи виявлення вторгнень Snort важливим елементом є набір правил, які використовує користувач. Коли Snort виявляє пакет, який відповідає неправильному значенню підпису, він збирає правила від користувача та виконує дію, указану в списку налаштованих користувачем правил.

В даній статті розглянемо, як працюють правила інструменту Snort в операційній системі Windows.

Мета: Дослідження системи виявлення вторгнень на основі правил Snort в операційній системі Windows.

1. Особливості правил SNORT

Правила Snort – це вказівки, які користувач надає своєму персоналу служби безпеки. Такі правила мають наступні особливості [1, 2]:

1. Тригерні сповіщення. Сповіщення про зловмисну діяльність, яке може становити потенційну загрозу для користувача, є особливістю правила Snort. Таку важливу інформацію, як IP-адреса, мітка часу, тип ICPM (міжмережевий протокол керуючих повідомлень), довжина IP-заголовка тощо, можна відстежити за допомогою правила Snort.

Існує кілька режимів сповіщень, які можна створити, одні з найпопулярніших: швидкий, повний, жодного, CMG, Unsock і Console. Кожен з них унікальний і відмінний один від одного.

Наприклад, якщо користувачу потрібен повний звіт із детальною інформацією, правило виглядатиме так: Output alert_full: alert.full. Якщо користувачу потрібен швидкий звіт, який не обов'язково повинен бути таким докладним, як повний звіт, його можна отримати за таким правилом: Output alert_fast: alert.fast.

2. Гнучке використання. Правило snort – це однорядковий код, який допомагає ідентифікувати трафік. Однак сучасні правила snort задовольняють більші та динамічніші вимоги, тому можуть бути більш детальними.

Крім того, користувач може завантажити Snort Rules та використовувати в будь-якій операційній системі. Немає ніяких обмежень, чи це Linux, Unix, Windows, Ubuntu чи будь-який інша операційна система, Snort однаково захищає мережу користувача. А ще, це все можна встановити безкоштовно.

3. Визначення шахрайства. Система виявлення вторгнень з відкритим кодом допомагає ідентифікувати та розрізняти регулярні й суперечливі дії у мережі. Це відбувається завдяки правилам Snort, вони посилаються на мову, яка допомагає уможливити таке спостереження. Ця мова є дуже простою, нею може користуватися будь-хто, хто має базові знання кодування. Snort поєднує в собі три методи виявлення потенційного кібершахрайства [3]:

1. Підпис: IDS на основі підписів відноситься до ідентифікації пакетів

даних, які раніше становили загрозу. Він визначає історичні закономірності або популярні і шкідливі послідовності.

2. Протокол: у цьому методі Snort виявляє підозрілу поведінку джерела IP-адреси – Інтернет-протоколу. Кожен комп'ютер має унікальну IP-адресу, а дані, отримані з недовірливої IP-адреси, виявляються та надсилаються в режимі реального часу.

3. Перевірка на основі аномалій: є велика різниця між IDS на основі підпису/протоколу та перевіркою на основі аномалій. У той час як інші два покладаються на попередню або історичну поведінку, IDS на основі аномалій виявляє та сповіщає про будь-який тип поведінки, який можна розглядати підозрілим.

2. Встановлення інструменту Snort на ОС Windows

Спочатку необхідно завантажити Snort та правила Snort для windows з офіційного сайту www.snort.org. При встановленні Snort у кореневий каталог необхідно підтвердити встановлення програмного інструменту Winpcap.

Відкриваємо командний рядок Windows та прописуємо команду: «cd Snort», завдяки їй перевіряємо, чи встановлений Snort. Перевіряємо, чи створено каталог bin у папці каталогу, після чого переходимо до каталогу Bin і перевіряємо версію Snort. В нашому випадку встановлена версія 2.9.20.

Наступник кроком копіюємо завантажені заздалегідь правила та додаємо їх до папки на диску C:\Snort\rules. Після цього переходимо на C:/Snort/etc та відкриваємо Snort.conf за допомогою Wordpad. У цьому файлі додаємо біля HOME_NET власну IP-адресу, прокручуємо вниз до RULE_PATH, замінюємо ../rules на c:\Snort\rules, замінюємо ../so_rules на c:\Snort\so_rules та замінюємо ../preproc_rules на c:\Snort\preproc_rules. Крім того, замінюємо WHITE_LIST_PATH та BLACK_LIST_PATH з ../rules на c:\Snort\rules

Далі переходимо за адресою C:\Snort\rules і створюємо два текстові файли з іменами **whitelist** і **blacklist** і змінюємо їх розширення з **.txt** на **.rules**

Повертаємось у Wordpad, у файлі Snort.conf замінюємо команду #config logdir: на config logdir:c:\Snort\log.

Окрім цього у wordpad змінюємо наступні бібліотеки: usr/local/lib/snort_dynamicpreprocessor

на C:\Snort\lib\snort_dynamicpreprocessor,

usr/local/lib/snort_dynamicengine/libsf_engine.so

на C:\Snort\lib\snort_dynamicengine\sf_engine.dll

Коли бібліотеки замінено – закоментовуємо їх з допомогою знаку (#).

Додаємо коментар **(#)** перед усіма перерахованими препроцесорами під вбудованою нормалізацією пакетів. Вони генерують помилки під час виконання.

Тепер, налаштовуємо плагіни, замінивши classification.config на C:\Snort\etc\classification.config, після чого змінюємо розташування reference.config на C:\Snort\etc\reference.config. У наступному рядку додаємо output alert_fast: alert.ids для snort, щоб скинути всі логи alert.ids Після чого у файлі Snort.conf замінюємо ipvar на var. За замовчуванням рядок ipvar не розпізнається snort, тому ми замінюємо його на var.

Останнім кроком є видалення зворотної косої риски та додавання символів

коментарів (#) у рядках, які можна знайти за тегом «reputation preprocessor». Коли завдання виконане – зберігаємо файл та закриваємо вікно.

Залишилось встановити правило Snort. Для цього переходимо за адресою c:\Snort\rules і відкриваємо snort3-community.rules у wordpad.

Наприкінці додаємо правило (обов'язково), наприклад:

alert tcp any any -> any any(msg: "Testing Alert" ; sid:1000001)

У цьому випадку немає критеріїв, тому він завантажуватиме будь-який пакет ICMP, який отримує. У наведеному вище правилі ми також надаємо **ідентифікатор підпису** (sid), який вкрай необхідний. За домовленістю, коли ви пишете власні правила Snort, які повинні починатись з 999999.

Щоб переконатися, що snort справді генерує сповіщення, відкриваємо командний рядок, переходимо за адресою c:\Snort\bini та пишемо команду.

snort -iX -A console -c C:\snort\etc\snort.conf -l C:\Snort\log -K ascii,

де **X** – індексний номер вашого пристрою. Натискаємо Enter, і все готово.

На рисунку 1 приведено результат запуску Snort на Windows ОС.

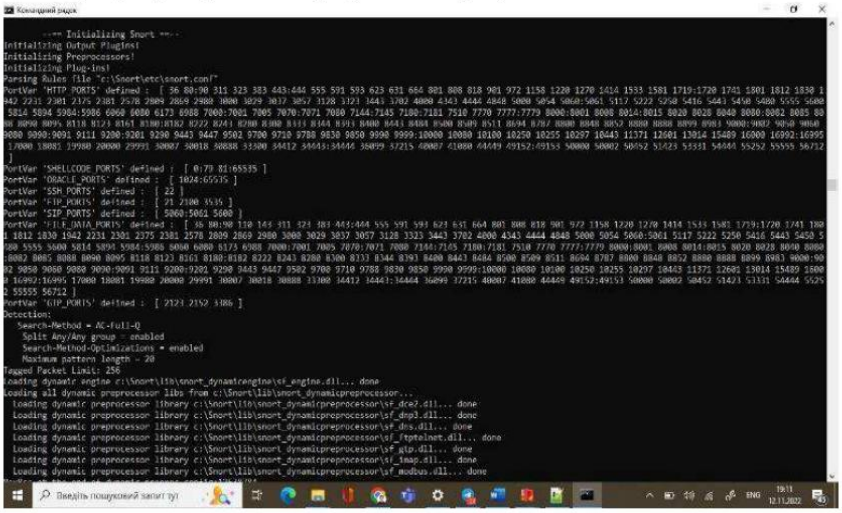


Рисунок 1 –Запуск Snort

Висновок. В роботі розглянуто приклад використання правил Snort для виявленні вторгнень. Правила Snort мають наступні особливості, а саме: тригерні сповіщення, гнучке використання та визначення шахрайства. Встановлення інструменту Snort на Windows ОС дозволяє користувачу розгорнути систему виявлення вторгнень на пристрої.

Перелік використаних джерел.

1. Gupta A., Sharma L. S. A categorical survey of state-of-the-art intrusion detection system-Snort. Int. J. Inf. Comput. Secur., 2020, 13.3/4: 337-356.

2. Порівняльний аналіз систем виявлення і запобігання вторгнень (Електронний ресурс). – Режим доступу: https://openarchive.nure.ua/bitstream/document/10555/1/RYKOV.pdf

3. Understanding and Configuring Snort Rules. (Електронний ресурс). – Режим доступу: https://www.rapid7.com/blog/post/2016/12/09/understanding-and-configuring-snort-rules/

Зав. кафедри кібербезпеки
д.т.н., проф. Василю ЯЦКІВУ

ДОВІДКА ПРО ВИКОРИСТАННЯ

Виконана студентом групи КБзм-21 факультету комп'ютерних інформаційних технологій Західноукраїнського національного університету Гавриляком М.В. кваліфікаційна робота на тему „Система виявлення мережевих вторгнень на основі SNORT" відповідає замовленню організації, має практичну значимість і планується до використання.

Міський голова Богдан КЕЛІЧАВИЙ