Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(повне найменування вищого навчального закладу)
Факультет комп'ютерно-інформаційних систем і програмної інженерії
(назва факультету)
Кафедра кібербезпеки
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр
(освітній рівень)

на тему: " Порівняльний аналіз сигнатурного та поведінкового підходів в системах IDS "

Виконав: студент VI курсу, групи СБм-61

Спеціальності:
125 «Кібербезпека та захист інформації»
(шифр і назва напряму підготовки, спеціальності)
Ванца Віталій Іванович
підпис (прізвище та ініціали)

Керівник Стадник М. А.
підпис (прізвище та ініціали)

Нормоконтроль Тимощук Д. І.
підпис (прізвище та ініціали)

Завідувач кафедри Загородна Н.В.
підпис (прізвище та ініціали)

Рецензент
підпис (прізвище та ініціали)

м. Тернопіль – 2024

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра кібербезпеки
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Загородна Н.В.
(підпис) (прізвище та ініціали)
«__» _____ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Магістр
(назва освітнього ступеня)

за спеціальністю 125 Кібербезпека та захист інформації
(шифр і назва спеціальності)

Студенту Ванці Віталія Івановичу
(прізвище, ім'я, по батькові)

1. Тема роботи Порівняльний аналіз сигнатурного та поведінкового підходів в системах IDS

Керівник роботи Стадник Марія Андріївна,
кандидат технічних наук, доцент кафедри КБ
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «20» 11 2024 року № 4/7-1112

2. Термін подання студентом завершеної роботи 12.12.2024

3. Вихідні дані до роботи Методи виявлення вторгнень. Сигнатурні та поведінкові підходи. Інструменти pfSense та Suricata. Сценарії атак. Операційні системи Ubuntu Linux та Kali Linux.

4. Зміст роботи (перелік питань, які потрібно розробити)
Вступ. Аналіз типів атак та їх наслідків. Аналіз методів захисту від атак.
Дослідження сигнатурного та поведінкового підходів в системах IDS.
Оцінка можливостей pfsense та Suricata. Практична реалізація системи IDS.
Тестування ефективності IDS на основі реальних сценаріїв атак.
Охорона праці та безпека в надзвичайних ситуаціях. Висновки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)
Тема. Актуальність дослідження. Мета, об'єкт, предмет дослідження.
Наукова новизна та практичні результати. Завдання дослідження.
Системи виявлення вторгнень (IDS).
Схема лабораторного тестового середовища.
Налаштування сигнатурного правила для системи IDS Suricata.
Налаштування поведінкового правила для системи IDS Suricata
Моделювання кіберзагроз.
Тестування систем IDS.
Висновки.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Осухівська Г.М., к.т.н., доцент		
Безпека в надзвичайних ситуаціях	Клепчик В.М., проректор з адміністративно-господарської роботи та будівництва		

7. Дата видачі завдання 20.09.2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	20.09 – 22.09	Виконано
2.	Підбір джерел для аналізу правил IDS	25.09 – 10.10	Виконано
3.	Опрацювання джерел в галузі дослідження	11.10 – 15.10	Виконано
4.	Налаштування тестового лабораторного середовища	16.10 – 25.10	Виконано
5.	Розроблення правил IDS основі сигнатурного та поведінкового підходів	25.10 - 30.10	Виконано
6.	Оформлення розділу «Огляд предметної області»	30.10 – 05.11	Виконано
7.	Оформлення розділу «Розгортання тестового лабораторного середовища»	06.11 – 10.11	Виконано
8.	Оформлення розділу «Сигнатурний та поведінковий підходи в системах IDS»	11.11 – 25.11	Виконано
9.	Виконання завдання до підрозділу «Охорона праці та безпека в надзвичайних ситуаціях»	26.11-01.12	
10.	Оформлення кваліфікаційної роботи	02.12 – 10.12	Виконано
11.	Нормоконтроль	08.12 – 10.12	Виконано
12.	Перевірка на плагіат	12.12 – 14.12	Виконано
13.	Попередній захист кваліфікаційної роботи	20.12 – 15.12	Виконано
14.	Захист кваліфікаційної роботи	26.12.2024	

Студент _____ Ванца В. І.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Стадник М. А.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Порівняльний аналіз сигнатурного та поведінкового підходів в системах IDS // ОР «Магістр» // Ванца Віталій Іванович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра кібербезпеки, група СБм-61 // Тернопіль, 2024 // С. 81, рис. – 29, табл. – - , кресл. – 18, додат. – 2.

Ключові слова: Кіберзагрози, IDS, Suricata, pfSense, Kali, python.

У кваліфікаційній роботі магістра проведено дослідження систем виявлення вторгнень з акцентом на сигнатурний та поведінковий підходи до виявлення мережевих атак. Метою роботи було оцінити ефективність цих підходів у сучасних інформаційних системах та розробити рекомендації щодо їх оптимального використання.

У першому розділі було здійснено аналіз класифікації мережевих атак, їх впливу на конфіденційність, цілісність та доступність даних. Досліджено сучасні методи захисту, зокрема використання брандмауерів та систем IDS/IPS. Другий розділ було присвячено розгортанню тестового лабораторного середовища на базі гіпервізора Hyper-V та операційної системи Windows Server 2022 Core. Створене середовище дозволило моделювати реальні сценарії атак. У третьому розділі проведено практичне тестування систем IDS з використанням спеціально розробленого python-скрипта для моделювання атак. Результати показали, що сигнатурний підхід ефективно виявляє відомі загрози, але має обмеження щодо нових або модифікованих атак. Поведінковий підхід продемонстрував здатність виявляти аномальну активність, що дозволяє ідентифікувати нові типи загроз.

У підсумку, робота підтвердила доцільність інтеграції сигнатурного та поведінкового підходів у системах IDS для забезпечення комплексного захисту інформаційних систем. Отримані результати можуть бути використані для вдосконалення існуючих засобів кібербезпеки та розробки нових стратегій захисту від мережевих атак.

ABSTRACT

Comparative analysis of signature-based and behavioral approaches in ids systems // Thesis of educational level "Master"// Vitalii Vantsa // Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Cybersecurity, group СБм-61 // Ternopil, 2024 // P. 81, figs. 29, tables -, drawings 18, added. 2.

Keywords: Cyber threats, IDS, Suricata, pfSense, Kali, python.

In the master's thesis, the research was conducted on intrusion detection systems with a focus on signature and anomaly-based (behavioural) approaches to detecting network attacks. The aim of the work was to assess the effectiveness of these approaches in modern information systems and develop recommendations for their optimal use.

The first section analyses the classification of network attacks and their impact on the confidentiality, integrity and availability of data. Modern methods of protection, including the use of firewalls and IDS/IPS systems, were investigated. The second section was devoted to the deployment of a test lab environment based on the Hyper-V hypervisor and the Windows Server 2022 Core operating system. The created environment made it possible to model real attack scenarios. In the third section, practical testing of IDS systems was carried out using a specially developed python script for attack simulation. The results showed that the signature approach effectively detects known threats, but has limitations with respect to new or modified attacks. The behavioural approach has demonstrated the ability to detect anomalous activity, which allows identifying new types of threats.

In summary, the study confirmed the feasibility of integrating signature and behavioural approaches in IDS systems to ensure comprehensive protection of information systems. The results obtained can be used to improve existing cybersecurity tools and develop new strategies to protect against network attacks.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

DDoS	—	Distributed Denial of Service
DoS	—	Denial of Service
IPS	—	Intrusion Prevention System
OSI	—	Open Systems Interconnection
UDP	—	User Datagram Protocol
TCP	—	Transmission Control Protocol
ICMP	—	Internet Control Message Protocol
HTTP	—	Hypertext Transfer Protocol
DNS	—	Domain Name System
PF	—	Packet Filter
NGFW	—	Next-Generation Firewall
QoS	—	Quality of Service
IDS	—	Intrusion Detection System
SMB	—	Server Message Block
DHCP	—	Dynamic Host Configuration Protocol
SSH	—	Secure Shell
NAT	—	Network Address Translation

ВСТУП

Актуальність теми. У умовах стрімкого розвитку інформаційних технологій та зростання залежності суспільства від мережі Інтернет питання кібербезпеки набуває особливої важливості. Системи виявлення вторгнень є одним з ключових інструментів захисту мереж від атак, але її ефективність значною мірою залежить від використовуваних підходів. Сигнатурний метод дозволяє оперативно виявляти відомі загрози, тоді як поведінковий підхід спрямований на розпізнавання нових, невідомих атак. Аналіз переваг, недоліків та ефективності цих методів у реальних умовах є важливим кроком для підвищення надійності систем кіберзахисту.

Мета і задачі дослідження. Метою роботи є порівняльний аналіз сигнатурного та поведінкового підходів в системах IDS з метою оцінки їх ефективності у виявленні мережевих атак.

Для досягнення поставленої мети сформульовані наступні задачі:

- дослідити типи мережевих атак та методи їх запобігання;
- провести огляд існуючих підходів до виявлення вторгнень у системах IDS;
- здійснити аналіз можливостей платформи pfSense, зокрема IDS Suricata;
- провести налаштування та тестування системи IDS на основі сигнатурного та поведінкового методів;
- здійснити порівняльний аналіз ефективності цих підходів у реальних умовах.

Об’єкт дослідження. Системи виявлення вторгнень у комп’ютерних мережах.

Предмет дослідження. Сигнатурний та поведінковий підходи виявлення вторгнень у системах IDS.

Наукова новизна одержаних результатів кваліфікаційної роботи. Удосконалено підходи до вибору оптимальної стратегії виявлення вторгнень залежно від специфіки мережі, типу атак та ресурсів системи. Запропоновано

рекомендації для інтеграції сигнатурних і поведінкових методів у комбінованих системах IDS для підвищення їх ефективності.

Практичне значення одержаних результатів. Результати роботи можуть бути використані для вдосконалення існуючих систем IDS у комп'ютерних мережах. Практичні рекомендації щодо вибору підходу виявлення вторгнень є корисними для фахівців з кібербезпеки.

Апробація результатів магістерської роботи. Основні результати дослідження були представлені на VIII Міжнародній науковій конференції "Науковий простір: актуальні питання, досягнення та інновації", яка відбулася 29 листопада 2024 року у місті Житомир.

Публікації. Основні результати кваліфікаційної роботи опубліковано у працях конференції (див. додаток А).

РОЗДІЛ 1 ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Класифікація мережевих атак

Мережеві атаки є загрозою для функціонування інформаційних систем та безпеки даних. Вони спрямовані на порушення конфіденційності, цілісності, доступності мережевих ресурсів [1].

Атаки на конфіденційність спрямовані на отримання несанкціонованого доступу до конфіденційної інформації, що передається чи зберігається в мережевих системах. Їх метою є порушення принципу конфіденційності, який гарантує доступ до інформації лише уповноваженим суб'єктам. Одним із найпоширеніших видів атак є перехоплення трафіку (sniffing), яке полягає у моніторингу мережевого трафіку для вилучення незашифрованих даних, таких як паролі чи логіни. Використання шифрування, наприклад HTTPS або VPN, дозволяє значно зменшити вразливість до такого роду атак. Атаки типу «людина посередині» (MITM) реалізуються через перехоплення і зміну трафіку між двома сторонами, що дозволяє атакуючим викрадати конфіденційні дані. Захистом від них є впровадження протоколів TLS/SSL та політик автентифікації сертифікатів. Ще одним небезпечним методом є використання шкідливого програмного забезпечення, такого як кейлогери, які записують натискання клавіш і викрадають паролі чи інші чутливі дані. Запобігти цьому можна за допомогою антивірусних програм і регулярного оновлення систем.

Атаки на цілісність спрямовані на зміну, знищення або пошкодження даних, що передаються чи зберігаються в інформаційних системах. Основною метою таких атак є порушення цілісності даних, тобто їхньої достовірності та незмінності. У результаті цих атак дані можуть бути спотворені, підроблені або видалені, що призводить до неправильної роботи систем, фінансових втрат або втрати довіри до організації. Одним із найпоширеніших методів є атака типу spoofing, яка полягає у підробці ідентичності. Наприклад, атакуючий може підробити IP-адресу або електронну пошту, щоб видавати себе за легітимного користувача чи систему. Це дозволяє маніпулювати даними, пересилати їх до

злочинця чи створювати неправильні записи в базах даних. Захист від цього типу атак включає використання криптографічних методів автентифікації, таких як цифрові підписи чи сертифікати. Інший поширений вид атак - SQL-ін’єкція. Атакуючий використовує вразливості в програмах, що працюють із базами даних, вводячи в поля для введення шкідливий код. Унаслідок цього зловмисник може змінювати, видаляти чи викрадати дані. SQL-ін’єкції є небезпечними для веб-додатків, які не мають належної перевірки введених даних. Захистом є використання параметризованих запитів, валідація введення та регулярний аудит коду.

Атаки на доступність спрямовані на порушення можливості легітимного користувача отримати доступ до системи, сервісів чи ресурсів мережі. Основною метою таких атак є тимчасове або постійне блокування роботи системи, зниження її продуктивності або повна недоступність для користувачів. Такі атаки особливо небезпечні для критично важливих інфраструктур, які потребують безперервного функціонування, таких як медичні установи, фінансові системи чи державні органи. Одним із найпоширеніших типів атак на доступність є атаки типу DoS. Вони здійснюються шляхом надсилання великої кількості запитів до сервера, що перевищує його можливості обробки. Унаслідок цього сервер стає недоступним для звичайних користувачів. Відповідно, DDoS є розширенням цього типу атак, коли атака здійснюється з багатьох пристроїв одночасно [2]. Зазвичай, для DDoS атак використовуються ботнети - мережі заражених пристроїв, які виконують команди зловмисника [3]. Захист від DoS та DDoS атак включає використання систем виявлення аномалій у трафіку, фільтрування трафіку за допомогою брандмауерів, а також спеціалізованих рішень для протидії DDoS (наприклад, Cloudflare, Akamai). SYN Flood атака є однією з різновидів DDoS-атак [4]. Вона базується на використанні протоколу TCP. Зловмисник надсилає велику кількість запитів на встановлення з’єднання (SYN), але не завершує процес триетапного рукостискання, залишаючи ресурси сервера зайнятими. Це блокує можливість обслуговування легітимних запитів. Захистом є налаштування тайм-ауту для очікування завершення з’єднань, використання SYN Cookies або обмеження кількості напіввідкритих з’єднань.

Іншим видом атак є UDP Flood, яка базується на надсиланні великої кількості UDP-пакетів на випадкові порти сервера [5]. Це змушує сервер постійно перевіряти, чи відкриті порти, і надсилати відповідь ICMP Destination Unreachable, що витрачає його ресурси. Захистом є блокування невикористовуваних UDP-портів і обмеження частоти ICMP-відповідей. HTTP Flood атака спрямована на виснаження ресурсів веб-сервера шляхом надсилання великої кількості HTTP-запитів [6]. Ця атака є складнішою для виявлення, оскільки запити виглядають легітимними, і класичні засоби захисту можуть не розпізнати її як атаку. Захистом є використання механізмів CAPTCHA для перевірки користувачів, а також аналіз поведінки трафіку для виявлення підозрілої активності. DNS Amplification атака є різновидом DDoS-атак, де зловмисник використовує відкриті DNS-сервери для надсилання величезних обсягів трафіку до цільової системи [7]. Атакуючий надсилає запит із підробленою IP-адресою жертви до DNS-сервера, отримуючи значно більший за обсягом відповідь, який надсилається до цільової системи. Для захисту необхідно закрити відкриті рекурсивні DNS-сервери і використовувати протокол DNSSEC.

1.2 Наслідки атак для інформаційних систем

Атаки на інформаційні системи можуть мати серйозні наслідки, які впливають на різні аспекти їх функціонування. Вони стосуються не лише технічної частини, але й фінансових, репутаційних та юридичних аспектів діяльності організацій. Залежно від типу атаки, її інтенсивності та цілей, наслідки можуть включати порушення конфіденційності, цілісності, доступності, а також значні втрати для бізнесу.

Одним із найбільш серйозних наслідків є порушення конфіденційності даних, коли зловмисники отримують доступ до конфіденційної або чутливої інформації. Це може стосуватися персональних даних клієнтів, комерційних таємниць, фінансових звітів чи даних про транзакції. Такі витоки можуть призвести до юридичних санкцій, зокрема через порушення законів про захист

персональних даних. Крім того, компанії втрачають довіру клієнтів, що негативно впливає на їхню репутацію.

Порушення цілісності даних може призводити до значних операційних проблем. Якщо дані будуть змінені, підроблені або знищені, це може вплинути на прийняття рішень, які базуються на цих даних. Наприклад, зміна банківських реквізитів у системах управління фінансами може призвести до втрат великих сум коштів. У випадку критично важливих інфраструктур, таких як енергетичні чи медичні системи, це може створити загрозу життю та здоров’ю людей.

Порушення доступності інформаційних систем внаслідок атак, таких як DDoS, може спричинити значні простої в роботі. Втрата доступу до систем або даних у критично важливих секторах, таких як фінанси, охорона здоров'я чи транспорт, може завдати значних фінансових збитків. Наприклад, недоступність онлайн-банкінгу або платіжних систем впливає на довіру клієнтів і може призвести до значних втрат доходів.

Фінансові втрати є одним із найочевидніших наслідків атак. Вони включають прямі витрати, такі як компенсації клієнтам чи витрати на відновлення систем, а також непрямі, такі як втрати через простій, втрата клієнтів чи зниження ринкової вартості компанії. Великі корпорації можуть втратити мільйони доларів внаслідок тривалих атак або витоків інформації.

Репутаційні втрати також є серйозним наслідком атак. Коли клієнти або партнери дізнаються про вразливості компанії до кіберзагроз, це може підірвати їхню довіру до організації. Репутаційний збиток часто має довгостроковий характер, впливаючи на конкурентоспроможність компанії на ринку.

Юридичні та регуляторні наслідки виникають у випадках, коли атаки виявляють недотримання стандартів кібербезпеки або законодавчих вимог. Організації можуть стикнутися зі штрафами, судовими позовами або іншими санкціями.

Витрати на відновлення після атак також є значними. Це включає в себе виявлення та усунення вразливостей, відновлення втрачених даних, оновлення програмного забезпечення та інфраструктури, а також навчання персоналу. Такі

витрати можуть тривати місяці після атаки, особливо якщо вона була масштабною.

Соціальні та психологічні наслідки атак можуть бути особливо помітними у випадках, коли вони впливають на критично важливі послуги, такі як медицина, комунальні послуги чи транспорт. Збої в роботі таких систем викликають стрес, паніку або навіть ставлять під загрозу життя і здоров’я людей.

1.3 Сучасні методи захисту від різних типів атак

Ефективний захист інформаційних систем від різноманітних кібератак потребує впровадження сучасних методів, які забезпечують багаторівневу безпеку та адаптацію до нових загроз. Ці методи спрямовані на запобігання атакам, виявлення їх у режимі реального часу та мінімізацію наслідків. Кожен метод орієнтований на певний тип атак, проте їх комбінація забезпечує комплексний захист.

1.3.1 Використання брандмауерів

Брандмауери є одним із найважливіших компонентів системи інформаційної безпеки, які забезпечують захист мереж від несанкціонованого доступу та атак [8]. Вони функціонують як бар’єр між внутрішньою мережею організації та зовнішнім середовищем, фільтруючи вхідний і вихідний трафік на основі заданих правил. Основною метою використання брандмауерів є мінімізація ризиків, пов’язаних із вторгненням зловмисників, і захист ресурсів організації.

Брандмауери аналізують мережевий трафік, перевіряючи його відповідність визначеним політикам безпеки. Це дозволяє блокувати небажані пакети, які можуть містити загрози, наприклад шкідливе програмне забезпечення, або бути частиною атак, таких як DDoS чи спуфінг.

Сучасні брандмауери можуть працювати на різних рівнях моделі OSI, забезпечуючи гнучкість і потужність у захисті. Наприклад, вони можуть

фільтрувати пакети на рівні IP-адрес або портів (канальний та транспортний рівні) чи аналізувати конкретний вміст на рівні додатків.

Брандмауери класифікуються за методом роботи на п’ять основних типів, кожен з яких має свої унікальні особливості та підходи до забезпечення безпеки мережі [9].

Брандмауери з фільтрацією пакетів працюють на мережевому та транспортному рівнях моделі OSI, аналізуючи заголовки пакетів даних. Вони дозволяють або блокують трафік на основі параметрів, таких як IP-адреса, номер порту чи протокол (див. рисунок 1.1).

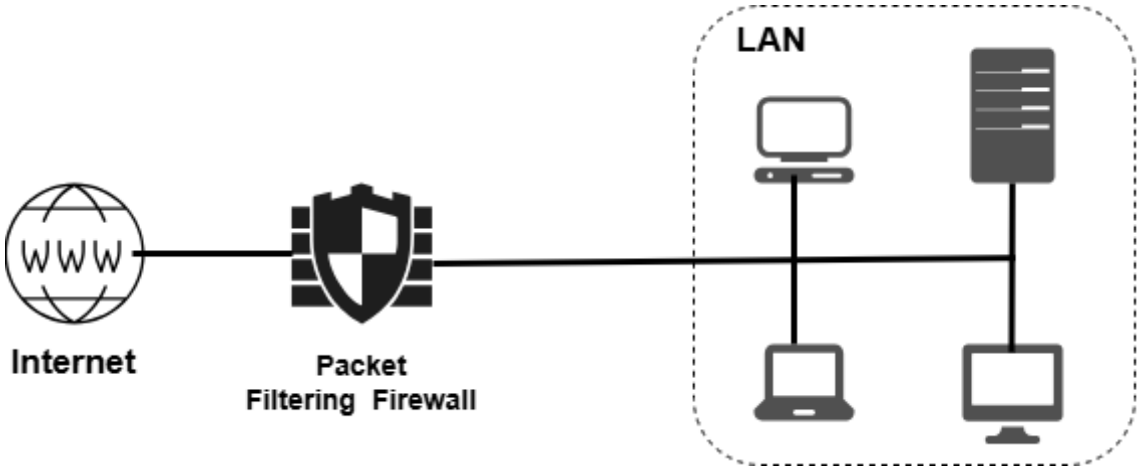

Рисунок 1.1 – Схема розташування брандмауера з фільтрацією пакетів

Цей тип є простим у впровадженні та швидким, але обмежений у виявленні складних атак, оскільки не аналізує вміст пакетів.

Брандмауери з перевіркою стану зберігають інформацію про активні з’єднання у таблиці станів, що дозволяє враховувати контекст сесії (див. рисунок 1.2).

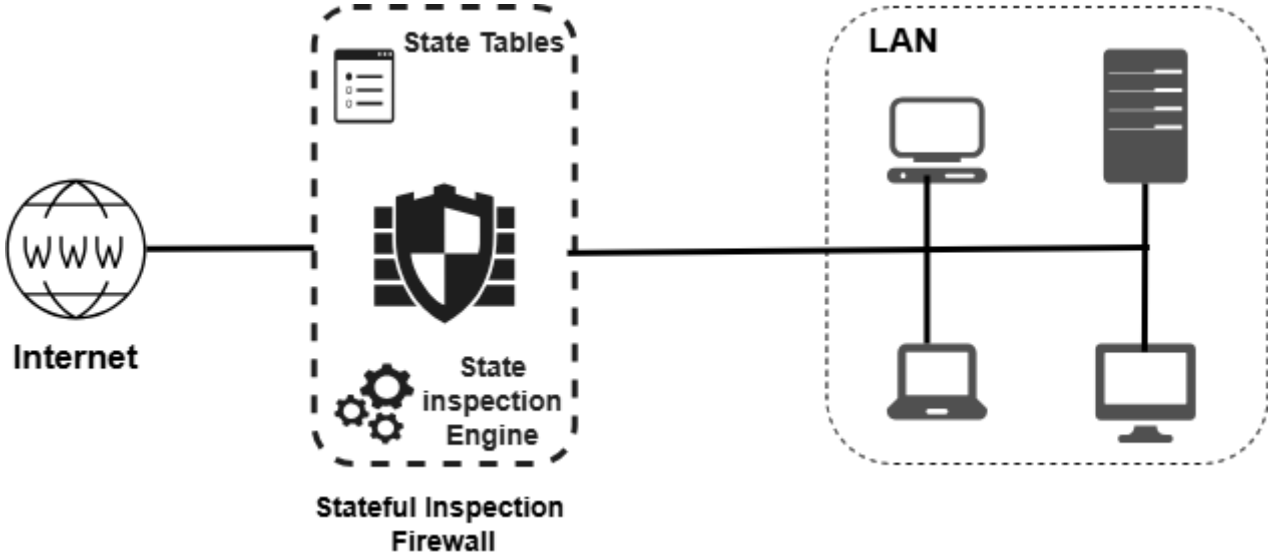

Рисунок 1.2 – Схема розташування брандмауера з перевіркою стану

Вони більш ефективні у порівнянні з фільтрацією пакетів, оскільки можуть захистити від атак, які спотворюють сеанси, наприклад, спуфінг. Проте їхній недолік полягає у вразливості до атак на ресурси, таких як SYN Flood.

Шлюзи на рівні схеми працюють на транспортному рівні, створюючи віртуальний канал для передачі даних між джерелом і приймачем Цей тип забезпечує базовий контроль з’єднань і маскує внутрішню мережу, але не аналізує вміст трафіку, що робить його менш ефективним для складних сценаріїв.

Шлюзи прикладного рівня, також відомі як проксі-сервери, функціонують на прикладному рівні моделі OSI. Вони аналізують специфічні протоколи, такі як HTTP чи SMTP, що дозволяє ефективно блокувати загрози, характерні для веб-додатків і електронної пошти. Такий детальний аналіз є ресурсозатратним, тому продуктивність цих брандмауерів може бути нижчою.

Брандмауери нового покоління (NGFW) поєднують функції брандмауерів з перевіркою стану із додатковими можливостями, такими як аналіз вмісту трафіку, інтеграція з системами виявлення атак (IDS/IPS), захист від загроз нульового дня та аналіз зашифрованих даних. NGFW є найсучаснішим і найефективнішим рішенням для складних корпоративних середовищ, хоча їх висока вартість і вимоги до ресурсів можуть бути перешкодою для впровадження в малих організаціях (див. рисунок 1.3).

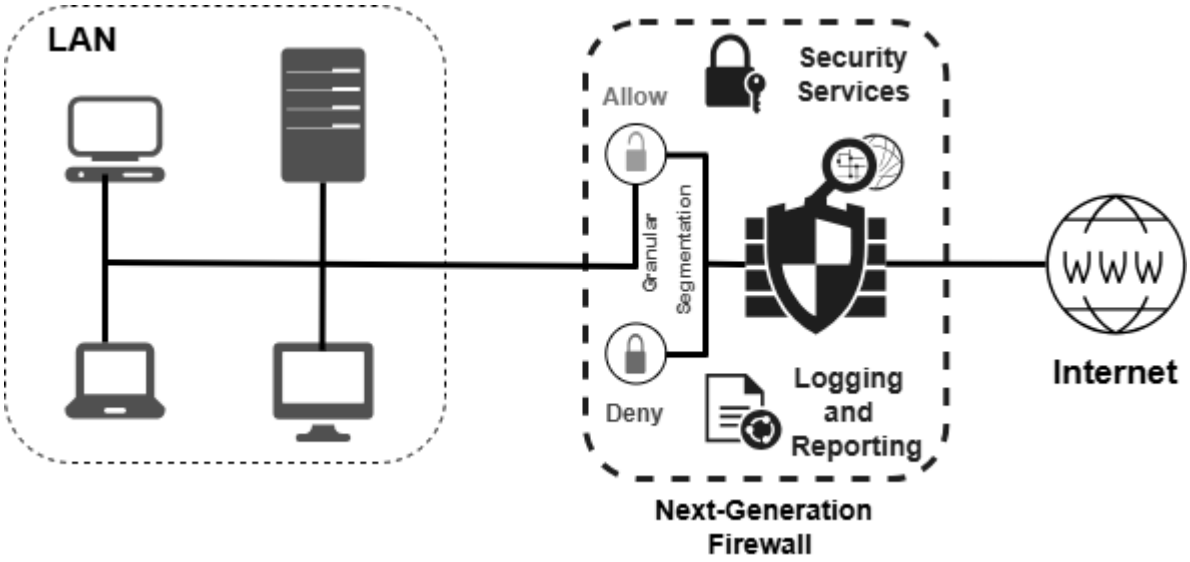

Рисунок 1.3 – Схема розташування брандмауера з перевіркою стану

Кожен із цих типів брандмауерів має свої переваги та недоліки, тому вибір конкретного рішення залежить від потреб організації, рівня загроз та обсягу мережевого трафіку. У багатьох випадках найбільш ефективним підходом є комбінування кількох типів брандмауерів для забезпечення багаторівневого захисту мережі. Брандмауери є першою лінією захисту, що значно знижує ризики проникнення зловмисників до мережі. Вони забезпечують надійний контроль трафіку, адаптуються до нових загроз і інтегруються з іншими інструментами кібербезпеки.

1.3.2 Виявлення та протидія за допомогою IDS/IPS

Системи виявлення та протидії вторгненням є важливими елементами сучасної кібербезпеки. Вони забезпечують моніторинг мережевого трафіку та системної активності для виявлення потенційних атак або аномальних дій, а також можуть автоматично реагувати на загрози. IDS/IPS працюють на різних рівнях мережевої моделі OSI, аналізуючи дані для запобігання несанкціонованому доступу, поширенню шкідливого програмного забезпечення або порушенню цілісності даних.

Системи виявлення вторгнень зосереджуються на моніторингу мережевого трафіку та системних подій для виявлення підозрілої активності [10] (див. рисунок 1.4).

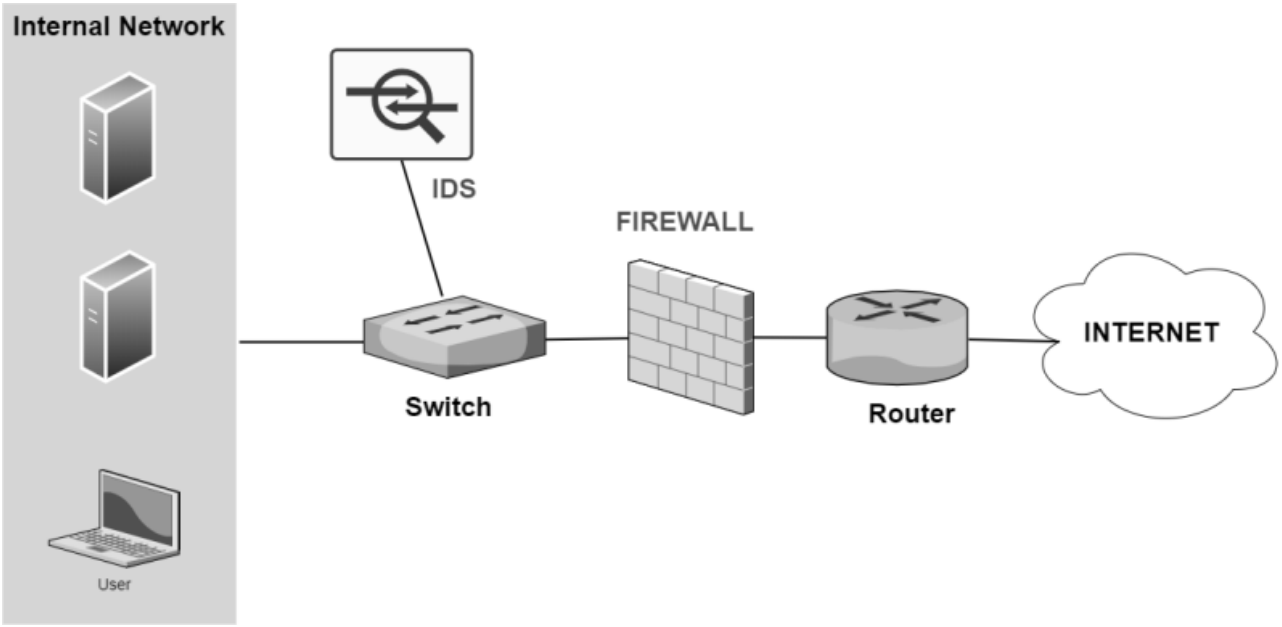

Рисунок 1.4 – Схема розташування IDS

Вони класифікуються на два основних типи: сигнатурні та поведінкові. Сигнатурні IDS працюють на основі бази відомих шаблонів атак (сигнатур) і виявляють загрози, що відповідають цим шаблонам. Цей підхід є ефективним для ідентифікації відомих атак, але може не спрацювати проти нових загроз або атак нульового дня. Поведінкові IDS, навпаки, аналізують аномалії в трафіку або поведінці користувачів, порівнюючи їх із базовими профілями. Це дозволяє виявляти нові типи атак, але може генерувати більше хибних спрацювань.

Системи протидії вторгненням не лише виявляють загрози, але й автоматично блокують їх [11] (див. рисунок 1.5).

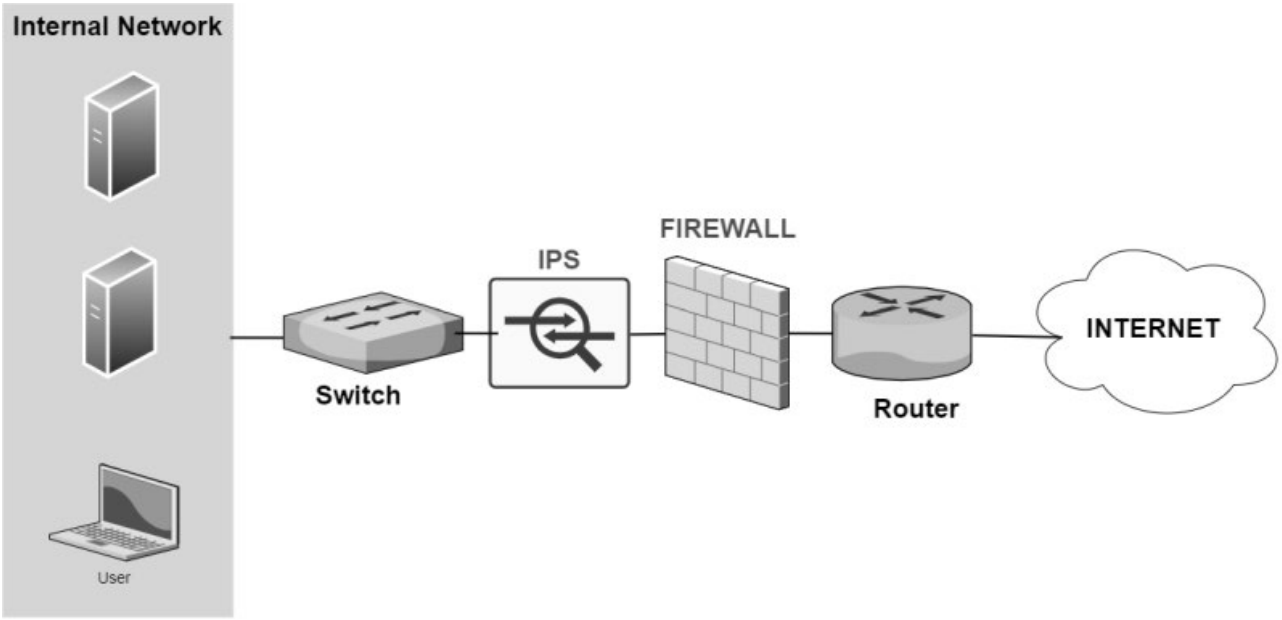

Рисунок 1.5 – Схема розташування IPS

IPS працюють у реальному часі, зупиняючи атаки до того, як вони зможуть завдати шкоди. Як і IDS, IPS також поділяються на сигнатурні та поведінкові, але їхня інтеграція в мережевий трафік дозволяє їм активніше втручатися, наприклад, блокуючи підозрілі IP-адреси або скидаючи небезпечні з’єднання.

Основні функції IDS/IPS:

- моніторинг мережі для виявлення невідповідностей у поведінці трафіку чи користувачів;
- ідентифікація загроз на основі відомих шаблонів атак;
- автоматичне блокування зловмисного трафіку, переривання з’єднань або ізоляція заражених вузлів (для систем IPS);
- запис дій для подальшого аналізу, що допомагає розслідувати інциденти та вдосконалювати політики безпеки.

Методи реагування систем IDS/IPS:

- IDS/IPS генерує сповіщення для адміністратора у випадку виявлення загрози, надаючи інформацію про її характер і джерело;
- IPS перериває з’єднання або блокує певні пакети, якщо вони відповідають сигнатурі або виявляються аномальними.

Інтеграція IPS із брандмауером дозволяє автоматично змінювати правила для запобігання повторенню атак. Заражені або скомпрометовані вузли можуть бути ізольовані від мережі до усунення загрози.

Сучасні IDS/IPS вторгненням представляють широкий спектр рішень, які відрізняються функціональністю, продуктивністю та способами інтеграції в інформаційні системи. Вони використовуються як у малих мережах, так і в масштабних корпоративних інфраструктурах, забезпечуючи ефективний захист від широкого спектра загроз.

Snort - одна з найпопулярніших IDS/IPS із відкритим вихідним кодом. Вона працює за сигнатурним та поведінковим методами і здатна виявляти відомі атаки за допомогою бази правил. Snort широко застосовується завдяки своїй гнучкості, розширюваності та великій спільноті користувачів, які постійно оновлюють базу сигнатур. Це рішення ідеально підходить для невеликих мереж або як базовий інструмент у великих інфраструктурах.

Suricata - багатопотокова IDS/IPS, яка підтримує аналіз високошвидкісного трафіку та інтеграцію з іншими інструментами безпеки [12]. Suricata може працювати як у сигнатурному, так і в поведінковому режимах, а також аналізувати протоколи прикладного рівня, наприклад HTTP, TLS чи DNS. Вона є особливо ефективною в умовах високонавантажених мереж.

Zeek - поведінкова IDS, яка орієнтована на аналіз мережевого трафіку та виявлення аномалій [13]. Zeek фокусується на аналізі прикладного рівня і забезпечує глибоку інспекцію трафіку. Вона використовується для виявлення складних атак, наприклад тих, що включають декілька етапів вторгнення.

Cisco Firepower - комерційне рішення, яке поєднує функції брандмауера, IPS і системи глибокого аналізу трафіку [14]. Firepower інтегрується з іншими рішеннями Cisco, надаючи широкі можливості для управління загрозами, автоматизації захисту та моніторингу в режимі реального часу.

Palo Alto Networks - системи нового покоління, які об'єднують функціонал брандмауера та IPS із глибокою інтеграцією штучного інтелекту [15]. Ці системи аналізують трафік у реальному часі, включаючи зашифрований, і можуть виявляти та блокувати складні атаки нульового дня.

Fortinet FortiGate - NGFW, що включає інтегровану IPS [16]. FortiGate підтримує захист від складних загроз, із використанням штучного інтелекту та хмарних технологій. Це рішення добре підходить для організацій будь-якого розміру завдяки масштабованості та простоті управління.

1.4 Системи виявлення вторгнень

Системи виявлення вторгнень - це програмно-апаратні рішення, призначені для моніторингу мережевого трафіку або активності систем з метою виявлення несанкціонованих дій, спроб атак чи аномальних подій. IDS аналізують дані в реальному часі або зберігають їх для подальшого аналізу, допомагаючи виявляти потенційні загрози або мінімізувати наслідки інцидентів. Вони доповнюють інші засоби безпеки, такі як брандмауери чи антивірусні програми, забезпечуючи більш детальну перевірку активності в мережі.

Системи IDS класифікуються за кількома основними критеріями. За джерелом даних вони поділяються на мережеві (NIDS), які аналізують трафік у мережі, та хостові (HIDS), які працюють на окремих пристроях чи серверах, відстежуючи системні журнали, зміни файлів та активність додатків [17]. За методом виявлення IDS бувають сигнатурними, що працюють на основі бази даних відомих шаблонів атак, поведінковими, які виявляють аномалії в трафіку або поведінці користувачів, та гібридними, що комбінують обидва підходи для підвищення точності. IDS дозволяють своєчасно виявляти різні типи атак, такі як сканування портів, спроби доступу до заборонених ресурсів чи виконання шкідливого коду. Їх переваги полягають у ранньому виявленні загроз, зниженні ризиків поширення атак і наданні даних для аналізу інцидентів. Водночас IDS мають певні обмеження: вони не блокують атаки, а лише сповіщають про них (це функція IPS.

Незважаючи на це, IDS є важливим елементом сучасної кібербезпеки, який у поєднанні з іншими засобами, такими як брандмауери чи SIEM-системи, створює комплексний підхід до протидії загрозам. Сучасні системи IDS/IPS постійно вдосконалюються, інтегруючи нові методи аналізу, такі як машинне

навчання, для виявлення складних і невідомих загроз. Вибір системи залежить від розміру мережі, типу загроз, з якими стикається організація, та доступного бюджету. Інтеграція IDS/IPS із SIEM-системами та іншими інструментами кіберзахисту забезпечує ще більш потужний і комплексний підхід до безпеки.

Основними викликами є зменшення кількості хибних спрацювань, забезпечення продуктивності при обробці великого обсягу трафіку та адаптація до нових загроз. Поведінкові системи часто потребують тривалого налаштування профілів нормальної активності, а сигнатурні є вразливими до атак нульового дня.

Сигнатурний метод виявлення вторгнень є одним із найпоширеніших підходів у IDS. Цей метод базується на порівнянні отриманих даних з мережевого трафіку з базою даних відомих шаблонів атак, які називаються сигнатурами. Сигнатура представляє собою унікальний набір характеристик, що ідентифікує конкретну атаку чи шкідливе програмне забезпечення. Наприклад, це може бути специфічна послідовність байтів у пакеті даних, характерні параметри протоколу.

Під час роботи сигнатурна IDS аналізує вхідний і вихідний трафік, перевіряючи його на відповідність наявним сигнатурам у базі. Якщо виявляється збіг, система генерує сповіщення. Цей підхід дозволяє швидко і точно виявляти відомі атаки, забезпечуючи високий рівень захисту від загроз, які вже документовані.

Перевагою сигнатурного методу є його висока точність у виявленні відомих загроз та низький рівень хибних спрацювань. Оскільки система шукає конкретні шаблони, ймовірність помилкового сповіщення мінімальна. Проте цей метод має суттєвий недолік: він не здатний виявляти нові, невідомі або модифіковані атаки, які не включені до бази сигнатур. Це робить систему вразливою до атак нульового дня та складних загроз, що постійно еволюціонують.

Для ефективної роботи сигнатурних IDS необхідне регулярне оновлення бази сигнатур. Постачальники безпеки та спільноти кібербезпеки постійно додають нові сигнатури, але затримка між появою нової загрози та її включенням до бази може створити вікно вразливості. Крім того, великий обсяг сигнатур

може впливати на продуктивність системи, оскільки аналіз кожного пакета вимагає більше ресурсів.

Сигнатурний метод часто використовується в комбінації з іншими підходами, такими як поведінковий аналіз, для створення багаторівневої системи захисту. Це дозволяє компенсувати недоліки кожного методу та підвищити загальну ефективність виявлення загроз. У сучасних умовах швидкої еволюції кіберзагроз комбінований підхід є найбільш оптимальним для забезпечення надійної кібербезпеки.

Поведінковий метод виявлення вторгнень є підходом, що базується на аналізі аномальної поведінки в мережі чи системі. На відміну від сигнатурного методу, який порівнює трафік із відомими шаблонами атак, поведінковий метод використовує моделі "нормальної" активності (базові лігнії), щоб виявляти відхилення, які можуть свідчити про можливу атаку. Цей підхід є ефективним для виявлення нових або невідомих атак, які не можуть бути ідентифіковані за допомогою сигнатур.

Основою поведінкового методу є формування базової лінії нормальної поведінки мережі, системи або користувачів. Це досягається шляхом моніторингу та аналізу трафіку чи активності протягом певного періоду. Наприклад, нормальна поведінка може включати середній обсяг трафіку, типові часові проміжки активності, стандартні протоколи та порти, що використовуються. Якщо виявляється відхилення від цієї базової лінії, система розцінює це як потенційну загрозу та генерує сповіщення.

Поведінковий метод виявлення особливо корисний для ідентифікації складних і багаторівневих атак, які можуть не мати чіткої сигнатури. Наприклад, метод дозволяє виявляти спроби несанкціонованого доступу, використання незвичних протоколів, атаки типу MITM або DDoS, які характеризуються раптовим збільшенням обсягу трафіку.

Головною перевагою поведінкового методу є здатність виявляти нові та модифіковані атаки, які не включені до бази відомих шаблонів. Це робить його ефективним засобом протидії атакам нульового дня. Крім того, поведінковий підхід дозволяє виявляти складні сценарії атак, які розгортаються протягом

тривалого часу. Однак цей метод має свої недоліки. Основна проблема - висока кількість хибних спрацювань, оскільки будь-яка нетипова активність, навіть легітимна, може бути розцінена як загроза. Це вимагає значних ресурсів для аналізу сповіщень та коригування базової лінії.

Ефективність поведінкового методу значно залежить від технологій, що використовуються для аналізу. Сучасні системи інтегрують машинне навчання та штучний інтелект, які дозволяють більш точно формувати базові моделі поведінки, адаптуватися до змін у мережі та знижувати кількість хибних спрацювань. Наприклад, алгоритми кластеризації можуть групувати схожі типи поведінки, а методи виявлення аномалій дозволяють фокусуватися лише на значущих відхиленнях.

Поведінковий метод є важливим інструментом у сучасних IDS і часто використовується в комбінації із сигнатурним методом. Такий гібридний підхід забезпечує більш високий рівень безпеки, оскільки компенсує недоліки кожного окремого методу. У сучасному середовищі, де загрози швидко змінюються, поведінковий аналіз є ключовим компонентом багаторівневої системи кіберзахисту.

1.5 Висновки до розділу 1

В першому розділі було проведено комплексний аналіз класифікації мережевих атак, їх наслідків для інформаційних систем, сучасних методів захисту, а також засобів виявлення та протидії за допомогою систем IDS/IPS. Було розглянуто основні типи атак на конфіденційність, цілісність і доступність даних, а також виявлено їхні ключові характеристики, методи реалізації та шляхи запобігання.

Також було досліджено наслідки атак для інформаційних систем, які включають порушення конфіденційності даних, цілісності інформації, недоступність ресурсів, фінансові втрати, репутаційні збитки, а також юридичні наслідки. Зокрема, підкреслено значення безперервного моніторингу та впровадження заходів безпеки для мінімізації ризиків і відновлення

функціональності після атак. Було детально описано використання брандмауерів як одного з базових засобів кіберзахисту, їхні типи, особливості роботи та можливості інтеграції з іншими компонентами захисту. Окрему увагу приділено системам виявлення та протидії вторгненням IDS/IPS, які забезпечують виявлення атак у режимі реального часу та автоматичне реагування на загрози.

Було розглянуто два основних методи виявлення атак в системах IDS: сигнатурний і поведінковий. Сигнатурний метод ефективний для виявлення відомих атак, але має обмеження у протидії загрозам нульового дня. Натомість поведінковий метод дозволяє ідентифікувати нові та модифіковані атаки через аналіз аномалій у поведінці трафіку чи користувачів, хоча може генерувати більше хибних спрацювань. Підкреслено важливість комбінованого використання обох методів для досягнення найвищого рівня ефективності систем виявлення вторгнень.

Розділ сформував науково-практичну базу для подальшого аналізу та практичного впровадження систем захисту інформаційних систем, забезпечивши розуміння основних принципів роботи сучасних інструментів кібербезпеки.

РОЗДІЛ 2 РОЗГОРТАННЯ ТЕСТОВОГО ЛАБОРАТОРНОГО СЕРЕДОВИЩА

2.1 Схема лабораторного середовища

Тестове лабораторне середовище є важливим компонентом досліджень у галузі кібербезпеки. Середовище забезпечує можливість моделювання реальних загроз та оцінку ефективності захисних заходів у контрольованих умовах. Воно дозволяє дослідникам і практикам проводити експерименти з виявлення вразливостей, тестування систем виявлення вторгнень, протидії вторгненням, налаштування брандмауерів, а також оцінювання новітніх технологій і методів захисту. Створення лабораторного середовища включає розгортання віртуальної мережевої інфраструктури, яка відображає основні елементи корпоративних мереж, зокрема сервери, користувацькі пристрої, системи моніторингу та безпеки.

У рамках такого середовища реалізується кілька зон, кожна з яких виконує певну функцію. Зона атакуючого, наприклад, включає інструменти для імітації атак, такі як Kali Linux, що дозволяє відтворювати сценарії реальних загроз, зокрема DDoS, MITM або SQL-ін’єкції. Зона захисту містить засоби безпеки, такі як брандмауери (pfSense, OPNsense) та IDS/IPS (Snort, Suricata), які оцінюються на предмет здатності виявляти та блокувати загрози. Інша частина лабораторії, така як серверна зона чи користувацькі сегменти, надають можливості для аналізу ефективності систем захисту в умовах, наближених до реальних.

Однією з основних переваг лабораторного середовища є його гнучкість і безпечність. Віртуалізація дозволяє швидко розгортати й масштабувати середовище, додаючи нові вузли, мережі чи системи захисту. Завдяки ізоляції мережевих сегментів можна безпечно виконувати тестування навіть найбільш руйнівних атак. Це дає змогу дослідникам отримувати точні дані про роботу систем і створювати оптимальні стратегії захисту.

Лабораторне середовище також є важливим інструментом навчання. Фахівці з кібербезпеки можуть вивчати принципи роботи захисних систем, розуміти поведінку загроз та вдосконалювати навички реагування на інциденти.

Такі середовища забезпечують доступ до реалістичних сценаріїв, що робить навчання більш ефективним і наближеним до реальних умов.

На рисунку 2.1 показано тестове лабораторне середовище на основі гіпервізора Hyper-V.

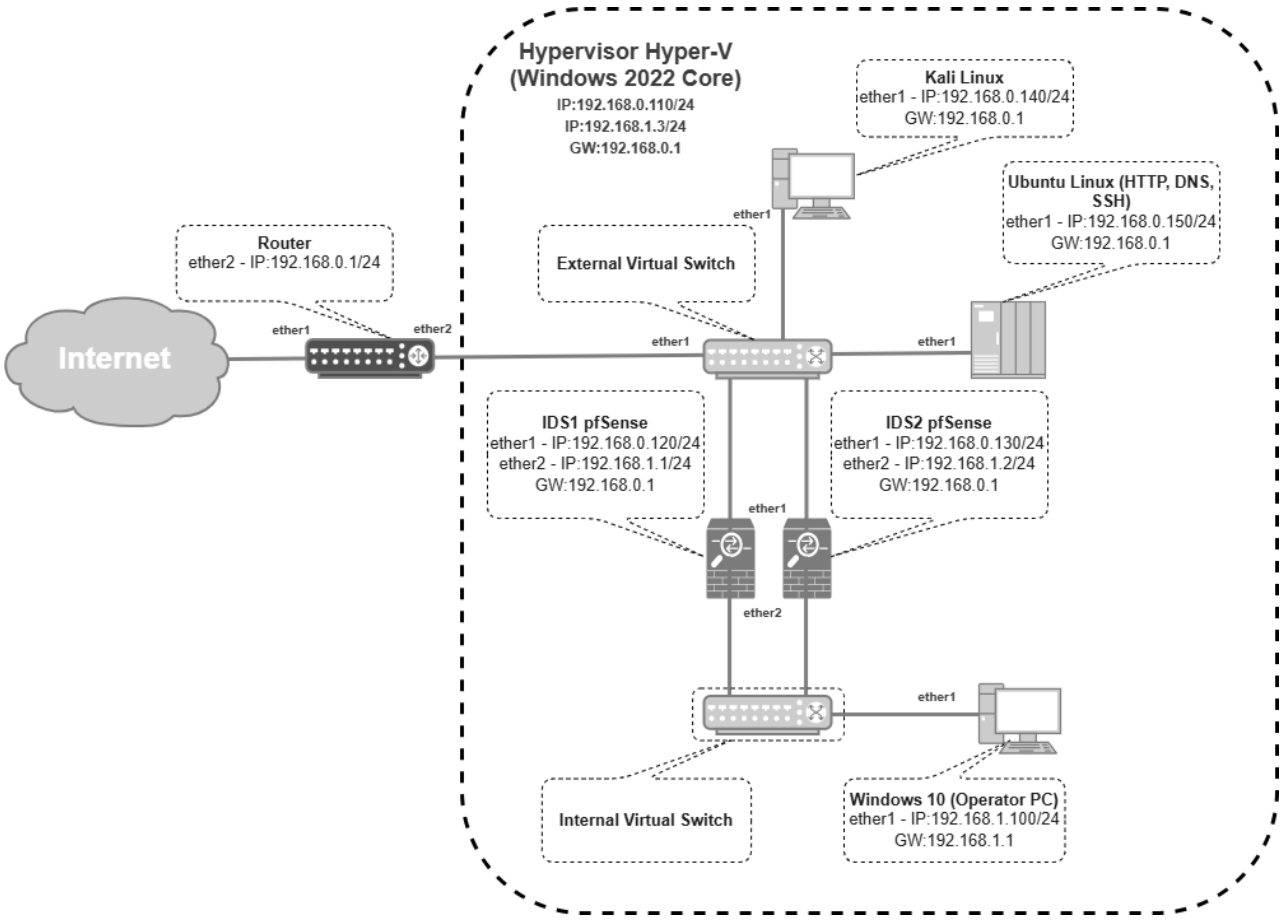

Рисунок 2.1 – Схема лабораторного тестового середовища на баз Hyper-V

Схема лабораторного тестового середовища на базі гіпервізора Hyper-V призначена для дослідження ефективності сигнатурного та поведінкового підходів IDS. Це середовище моделює реальні умови, дозволяючи аналізувати захисні механізми та тестувати реакцію систем на різні типи атак. Гіпервізор Hyper-V, розгорнутий на Windows Server 2022 Core, виступає основною платформою для управління віртуальними машинами. Він має два віртуальні комутатори: зовнішній (External Virtual Switch) для підключення до інтернету через маршрутизатор та внутрішній (Internal Virtual Switch), який створює ізольовану внутрішню мережу для управління. У зовнішній мережі розташовані ключові вузли, зокрема Kali Linux для моделювання атак, Ubuntu Linux зі

службами HTTP, DNS і SSH, а також дві системи IDS на базі pfSense, які реалізують сигнатурний та поведінковий підходи для виявлення загроз.

IDS1, що використовує сигнатурний метод, аналізує трафік на основі шаблонів атак, перевіряючи його відповідність базі сигнатур. IDS2, із впровадженням поведінкового підходу, відслідковує аномалії у трафіку, порівнюючи його з умовною базовою лінією нормальної активності. Обидві системи мають дві мережеві карти: одну для зовнішнього трафіку та іншу для підключення до внутрішньої мережі. У внутрішній мережі працює віртуальна машина з Windows 10, яка використовуються для управління. Така інфраструктура дозволяє тестувати як сценарії атак, так і звичайну поведінку користувачів.

Середовище налаштоване так, щоб трафік між інтернетом і внутрішньою мережею проходив через віртуальний комутатор, який працює в режимі моста. До даного моста підключені обидві системи IDS, що дає можливість оцінювати їхню ефективність у реальному часі. Kali Linux створює різні типи загроз, що дозволяє тестувати здатність систем виявляти як відомі, так і нові загрози. Ubuntu Linux працює як серверний вузол та є ціллю атаки на доступні сервіси. Windows 10 використовується для управління IDS на базі pfSense та гіпервізором Hyper-V на базі Windows Server 2022 Core.

Ця лабораторія створена для проведення порівняльного аналізу точності, швидкості виявлення та рівня хибних спрацювань кожної із систем IDS. Вона дозволяє оцінити, який підхід краще справляється з виявленням типів атак у різних сценаріях. Лабораторне середовище також підтримує масштабованість, забезпечуючи можливість додавання нових вузлів або систем для розширення функціональності та адаптації до змінних умов дослідження.

2.2 Компоненти лабораторного середовища

2.2.1 Windows Server 2022 з функцію Hyper-V

Windows Server 2022 - це сучасна серверна операційна система, розроблена Microsoft, яка забезпечує потужні можливості для побудови надійних і масштабованих серверних інфраструктур [18]. Ця ОС є частиною сімейства Windows Server і спрямована на вирішення потреб як малих організацій, так і великих підприємств, що використовують локальні, гібридні чи хмарні середовища. Вона пропонує новітні функції у сфері безпеки, віртуалізації, управління мережею та підтримки сучасних робочих навантажень.

Windows Server 2022 побудовано на базі Windows Server 2019 із додатковими покращеннями. Основною особливістю є розширені засоби безпеки. В ОС інтегровано вдосконалені засоби шифрування, зокрема шифрування AES-256 для SMB, що значно підвищує захист даних під час передачі. Windows Server 2022 підтримує покращену інтеграцію з хмарними сервісами Azure через Windows Admin Center, що дозволяє організаціям використовувати гібридні середовища. Це включає можливості резервного копіювання даних у хмарі, інтеграцію з Azure Security Center та управління віртуальними машинами, розгорнутими в Azure.

Windows Server 2022 Core - це серверна операційна система у вигляді базової редакції, яка забезпечує мінімалістичне середовище для роботи серверів. Цей режим роботи відрізняється від повної (Desktop Experience) версії відсутністю графічного інтерфейсу, що значно зменшує використання ресурсів і забезпечує кращу безпеку. Windows Server Core ідеально підходить для серверів, які працюють із критичними робочими навантаженнями, не потребуючи взаємодії з графічним інтерфейсом користувача.

Цей варіант операційної системи розроблений для розгортання інфраструктури серверів із максимальною продуктивністю та безпекою. Завдяки відсутності зайвих компонентів графічного інтерфейсу зменшуються ресурси, необхідні для роботи, що дозволяє зосередити всі потужності на обслуговуванні серверних служб і додатків. Windows Server 2022 Core також має менший "простір атаки" через обмежену кількість компонентів, які потенційно можуть містити вразливості. Основна взаємодія з Windows Server 2022 Core здійснюється через командний рядок (Command Prompt), PowerShell або

Windows Admin Center. Адміністратори можуть виконувати всі необхідні завдання, такі як налаштування мережі, керування ролями та функціями сервера, або діагностика проблем за допомогою текстових інструментів. Додатково, Windows Admin Center забезпечує зручний веб-інтерфейс для дистанційного управління та моніторингу серверів Core.

Це рисунку 2.2 показано консольне середовище Windows Server 2022 Standard Core, де використовується інструмент SConfig для базового управління сервером через текстовий інтерфейс.

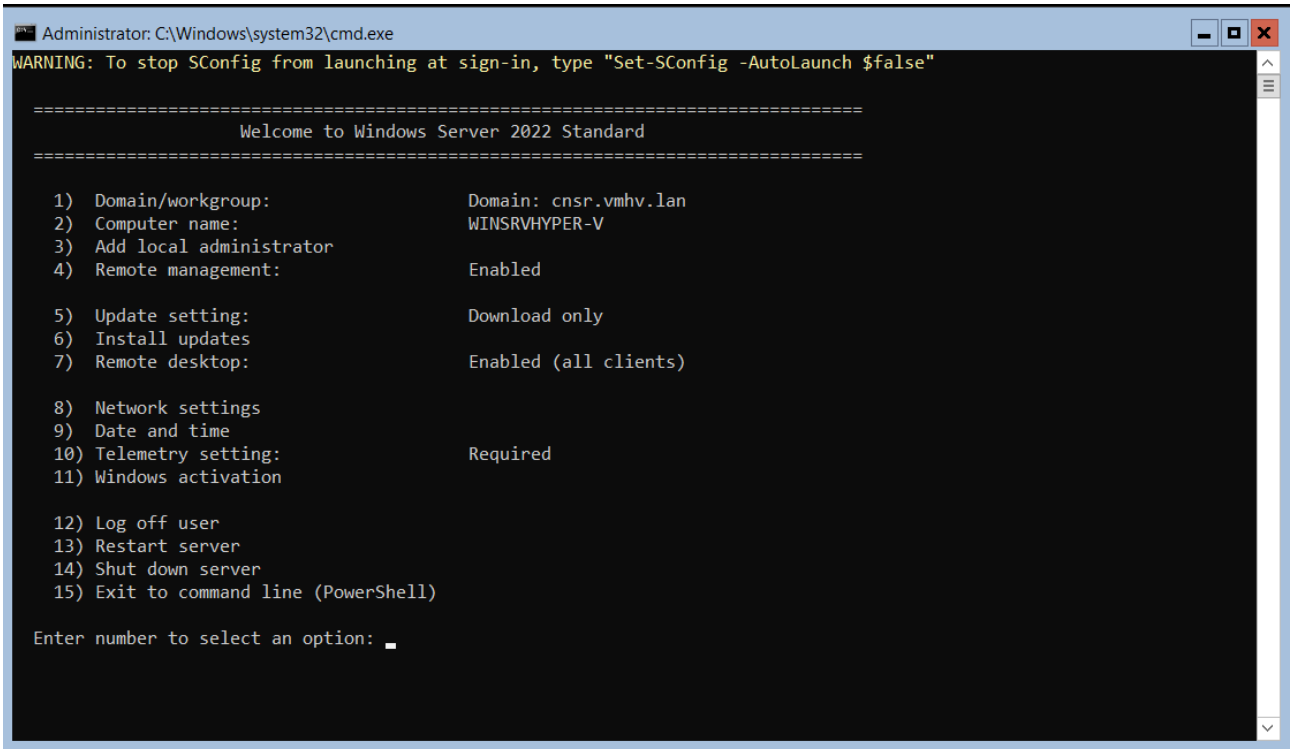

Рисунок 2.2 – Інструмент SConfig для управління Windows Server 2022 Standard Core

Цей інструмент дозволяє адміністраторам швидко виконувати основні задачі конфігурації без графічного інтерфейсу, що зменшує використання ресурсів і підвищує безпеку. Він є ефективним для серверів, розгорнутих у середовищах із високими вимогами до продуктивності, або для дистанційного управління через текстовий інтерфейс. Інтеграція з PowerShell забезпечує гнучкість і автоматизацію адміністрування.

Цей інтерфейс підходить для використання в лабораторному середовищі та виробничих сценаріях, де графічний інтерфейс не потрібен, а основні налаштування можуть бути виконані через консоль.

Windows Server 2022 Core підтримує всі основні серверні ролі, такі як Active Directory, DNS-сервер, DHCP-сервер, файлові сервери, Hyper-V та інші.

Windows Server 2022 Core підтримує технології віртуалізації Hyper-V, що дозволяє розгортати віртуальні машини з мінімальними витратами ресурсів.

Гіпервізор Hyper-V у Windows Server 2022 Core є потужним інструментом для створення та управління віртуальними машинами, що забезпечує високий рівень продуктивності та безпеки [19]. Це вбудована функція серверної операційної системи, яка дозволяє працювати з кількома віртуальними машинами на одному фізичному сервері, використовуючи мінімальну кількість ресурсів завдяки відсутності графічного інтерфейсу. Windows Server 2022 Core є ідеальною платформою для розгортання сучасних інфраструктур віртуалізації, включаючи тестові лабораторії, серверні ферми та хмарні середовища.

Hyper-V підтримує роботу з сучасним апаратним забезпеченням, використовуючи такі функції, як апаратна віртуалізація Intel VT-x та AMD-V, динамічний розподіл пам’яті і технології безпеки, наприклад Shielded VMs для захисту віртуальних машин від несанкціонованого доступу. Гнучкість гіпервізора забезпечується підтримкою віртуальних комутаторів. Використання Hyper-V у середовищі Core забезпечує ізоляцію віртуальних машин для підвищення безпеки та оптимізує використання системних ресурсів. Технології знімків (Checkpoints) дозволяють створювати резервні копії віртуальних машин і спрощують процес тестування змін або відновлення після збоїв.

Управління Hyper-V здійснюється через PowerShell, який надає потужний набір команд для створення, налаштування та моніторингу віртуальних машин, або через Windows Admin Center, який забезпечує веб-інтерфейс для дистанційного адміністрування. У нашому лабораторному середовищі було використано Hyper-V Manager на клієнтських віртуальній машині управляння з Windows 10 для віддаленого доступу до гіпервізора Hyper-V (див. рисунок 2.3).

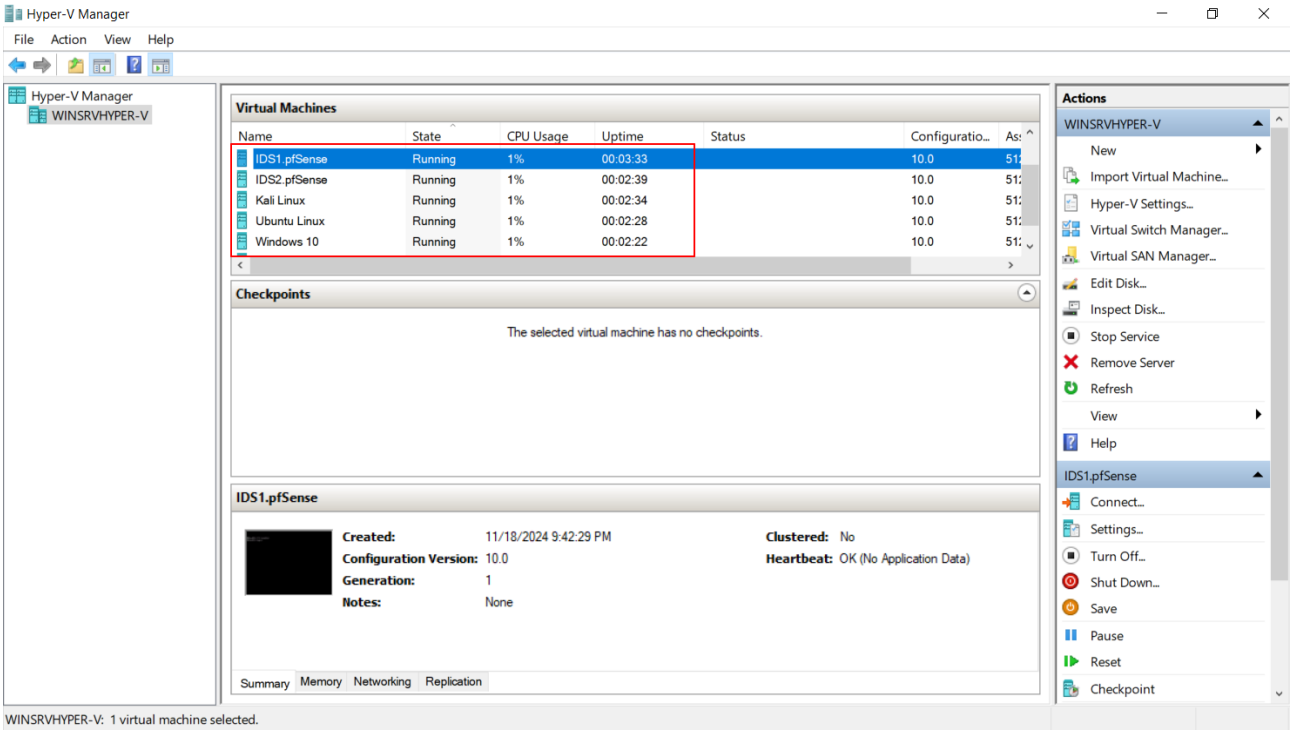

Рисунок 2.3 – Використання Hyper-V Manager

Hyper-V у Windows Server 2022 Core є оптимальним вибором для сценаріїв, де важлива висока продуктивність, безпека та гнучкість. Його функціональність дозволяє використовувати платформу як для створення лабораторного середовища для тестування IDS/IPS-систем і моделювання атак, так і для розгортання масштабованих серверних інфраструктур.

2.2.2 Маршрутизатор pfSense та IDS Suricata

Маршрутизатор pfSense - це програмне рішення, що базується на операційній системі FreeBSD, яке забезпечує широкий спектр мережевих функцій, включаючи маршрутизацію, фільтрацію трафіку, балансування навантаження, VPN та функції безпеки [20]. PfSense є одним із найпопулярніших інструментів для побудови мережевої інфраструктури завдяки своїй надійності, гнучкості, високій продуктивності та можливості безкоштовного використання в межах проєктів із відкритим кодом.

PfSense працює як програмний маршрутизатор, що може бути встановлений на спеціалізованих апаратних пристроях або віртуальних машинах, що робить його ідеальним вибором для як невеликих організацій, так і великих

корпоративних мереж. Висока функціональність досягається за рахунок підтримки статичної та динамічної маршрутизації з використанням протоколів, таких як OSPF, BGP та RIP. PfSense також забезпечує фільтрацію трафіку за допомогою брандмауера, який працює на основі системи PF, що дозволяє створювати гнучкі правила контролю доступу до мережі.

Серед основних можливостей pfSense є налаштування міжмережевих з'єднань із підтримкою NAT, що дозволяє з'єднувати кілька мереж або надавати доступ до зовнішніх ресурсів через одну IP-адресу. Система також підтримує VPN-з'єднання, включаючи OpenVPN, IPsec та WireGuard, що забезпечує безпечний віддалений доступ до корпоративних мереж або створення міжмережевих тунелів. Це особливо важливо для організацій, які працюють у географічно розподілених середовищах або мають потребу у віддаленому доступі. Також pfSense надає засоби для балансування навантаження, що дозволяє рівномірно розподіляти мережевий трафік між кількома інтернет-з'єднаннями або серверами, забезпечуючи високу доступність і продуктивність мережі. Додатково, вбудована підтримка QoS дозволяє пріоритизувати трафік для критично важливих застосунків, таких як VoIP чи відеоконференції.

Завдяки інтеграції з інструментами моніторингу та журналювання, такими як Syslog або SNMP, pfSense забезпечує детальний контроль за станом мережі та виявлення потенційних проблем у реальному часі. Інтерфейс управління pfSense представлений у вигляді веб-консолі, яка надає доступ до всіх функцій системи через зручний і зрозумілий інтерфейс (див. рисунок 2.4).

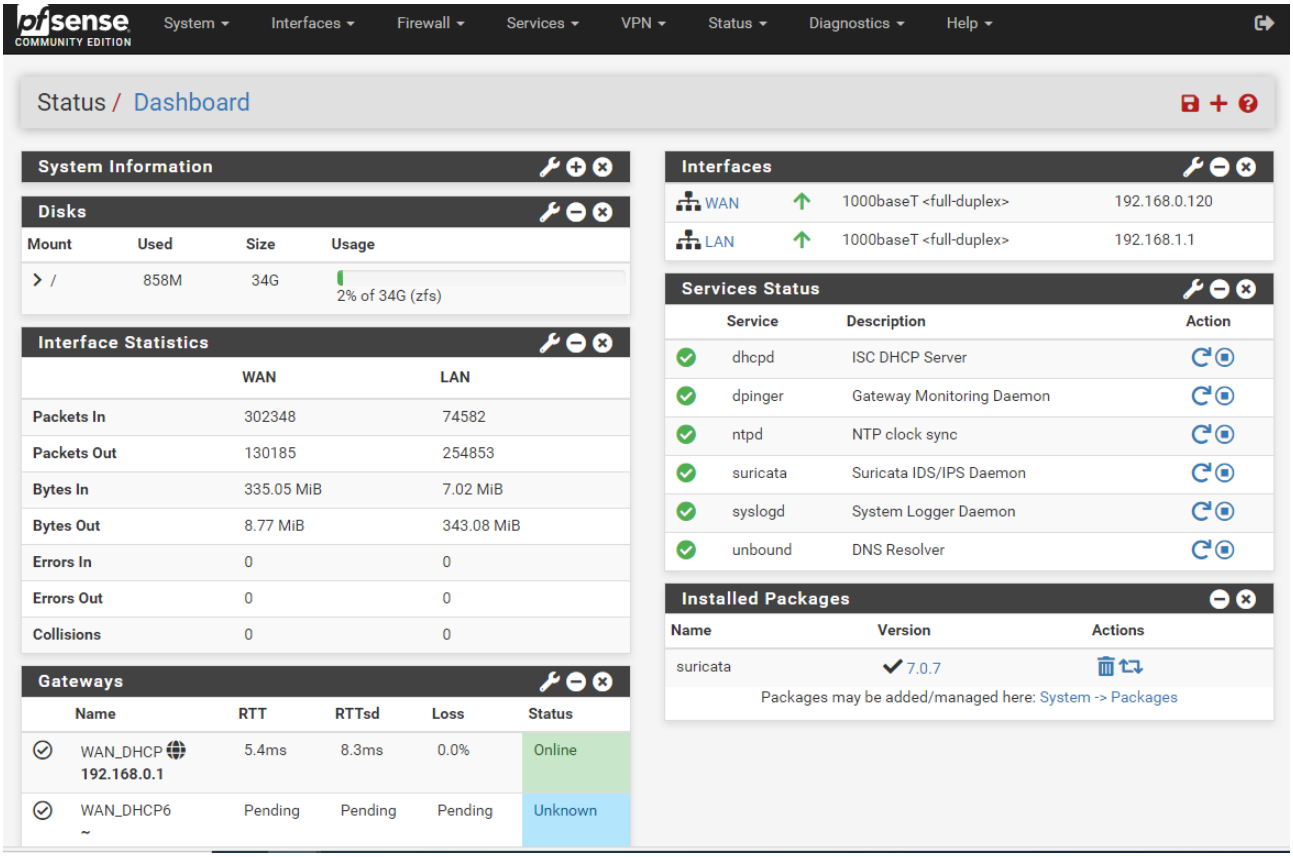

Рисунок 2.4 – Інтерфейс управління pfSense

Адміністратори також можуть виконувати конфігурацію через командний рядок або автоматизувати процеси за допомогою API. Маршрутизатор pfSense підтримує модульну архітектуру, яка дозволяє розширювати функціональність за допомогою додаткових модулів і пакетів. Наприклад, можна встановити IDS/IPS-системи на основі Snort чи Suricata, що забезпечує виявлення та блокування мережевих загроз. Інші популярні модулі включають Captive Portal для створення точок доступу з автентифікацією користувачів, Squid для роботи як проксі-сервер і підтримку фільтрації контенту, а також RADIUS-сервер для централізованого управління автентифікацією.

Suricata - це сучасна високопродуктивна IDS/IPS, яка інтегрується в pfSense для забезпечення безпеки мережі. Вона дозволяє здійснювати глибокий аналіз трафіку, виявляти загрози у реальному часі та автоматично реагувати на них. Suricata є відкритим програмним забезпеченням із багатопотоковою архітектурою, що забезпечує високу швидкість обробки мережевого трафіку, навіть у середовищах із великим навантаженням.

В інтеграції з pfSense, Suricata працює як модуль, який встановлюється через менеджер пакетів pfSense. Після встановлення Suricata може функціонувати як система IDS або IPS, залежно від потреб користувача. Suricata використовує бази сигнатур для виявлення загроз. Вона підтримує різні формати сигнатур, зокрема правила, сумісні з Snort, що дозволяє використовувати готові бази даних, такі як Emerging Threats, або створювати власні. Це забезпечує виявлення широкого спектра атак, зокрема DDoS, SQL-ін’єкцій, XSS, сканування портів і шкідливого програмного забезпечення. Для підвищення ефективності захисту Suricata також включає можливості аналізу протоколів, таких як HTTP, DNS, TLS, та аналіз трафіку, що проходить через зашифровані канали.

Інтеграція Suricata з pfSense дозволяє призначити систему для моніторингу або захисту окремих мережевих інтерфейсів. Це дозволяє адміністратору гнучко налаштовувати захист, наприклад, обмежуючи моніторинг тільки до зовнішніх з'єднань або до певних сегментів мережі. Suricata надає детальні журнали активності, які дозволяють аналізувати виявлені загрози, і легко інтегрується з іншими системами моніторингу, такими як SIEM або Syslog.

Suricata також підтримує поведінковий аналіз через виявлення аномалій у трафіку. Це робить її особливо корисною для виявлення нових типів атак, які не мають відомих сигнатур. Крім того, її можливість обробляти великий обсяг трафіку у багатопотоковому режимі робить її придатною для великих корпоративних мереж із високим навантаженням.

Використання Suricata в pfSense дозволяє налаштувати автоматичні дії у разі виявлення загроз, наприклад, блокування підозрілих IP-адрес чи трафіку. Це забезпечує ефективний захист у реальному часі без необхідності постійного втручання адміністратора. Адміністратори можуть створювати й налаштовувати власні списки дозволених або заблокованих адрес для підвищення точності захисту.

Панель керування Suricata в pfSense забезпечує інтуїтивно зрозумілий веб-інтерфейс для налаштування та управління всіма аспектами роботи системи (див. рисунок 2.5).

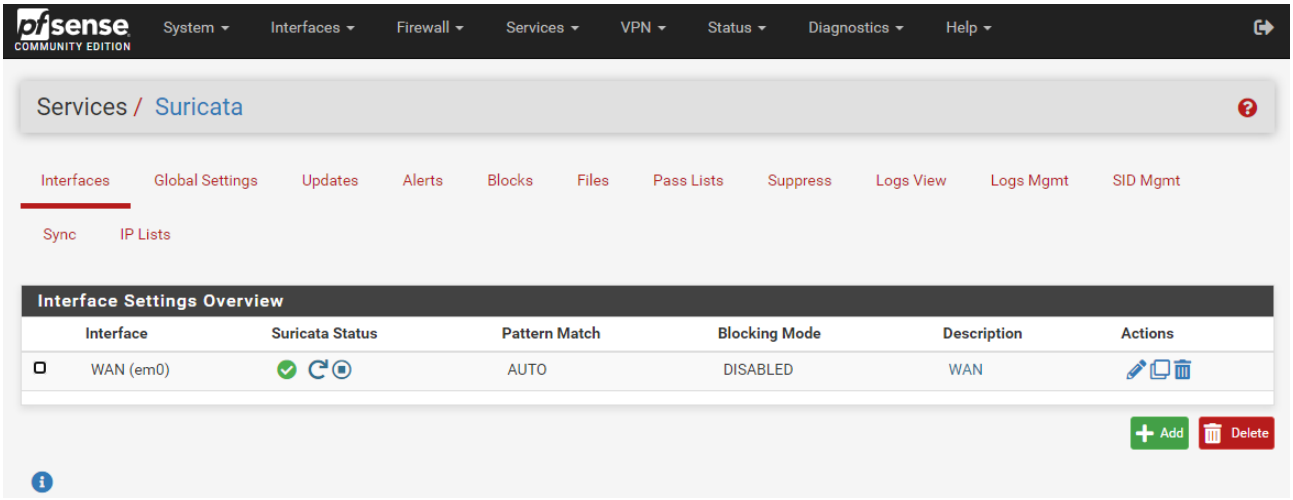

Рисунок 2.5 – Інтерфейс управління Suricata в pfSense

Ця панель є частиною адміністративного веб-інтерфейсу pfSense, і через неї адміністратор може налаштувати моніторинг, сигнатури, політики блокування, журнали та інші параметри, необхідні для забезпечення ефективного захисту мережі. На головній сторінці панелі Suricata відображається список всіх мережевих інтерфейсів, доступних для моніторингу. Адміністратор може вибрати, які з них будуть використовуватись Suricata, наприклад, зовнішній інтерфейс для моніторингу вхідного трафіку чи внутрішній для аналізу локальної мережі. Для кожного інтерфейсу можна окремо налаштувати режими роботи Suricata: IDS або IPS.

Під час налаштування Suricata адміністратор може вказати, які набори сигнатур використовувати. Є можливість інтегрувати популярні бази правил, такі як Emerging Threats чи Snort VRT. Suricata дозволяє автоматично завантажувати й оновлювати ці бази, що забезпечує актуальність захисту проти найновіших загроз. Крім цього, можна додавати користувацькі правила для вирішення специфічних задач безпеки. Панель також дозволяє гнучко налаштувати політики блокування. Адміністратор може визначити, чи будуть певні типи виявлених загроз лише журналюватися або ж блокуватися в режимі реального часу при роботі в режимі IPS. Доступні опції для створення списків виключень (whitelist) або блокування (blacklist), що дозволяє уникати хибних спрацювань і підвищувати точність роботи системи.

Suricata має інструмент журналювання та моніторингу. Адміністратор може переглядати всі виявлені події, включаючи деталі щодо джерела та призначення трафіку, типу загрози та відповідного правила. Журнали можна фільтрувати за різними параметрами, такими як дата, IP-адреса чи специфічна сигнатура, що полегшує аналіз інцидентів.

2.2.3 Сервер Ubuntu Linux

Ubuntu Server - це операційна система з відкритим вихідним кодом, яка оптимізована для серверних середовищ і широко використовується у розгортанні різних серверних рішень [21]. Вона базується на ядрі Linux і забезпечує стабільність, безпеку та гнучкість, що робить її популярним вибором для як малих, так і великих корпоративних інфраструктур.

Ubuntu Server надає широкий спектр можливостей для побудови серверів, включаючи веб-сервери, бази даних, сервери файлів, поштові сервери, сервери віртуалізації, а також сервери для обслуговування додатків. Вона підтримує хмарні середовища, такі як AWS, Azure та Google Cloud, що дозволяє розгортати масштабовані й гнучкі рішення. Також є повна підтримка можливості встановлення в віртуалізованих середовищах на основі гіпервізорів Huper-V, KVM, XEN та VMware ESXi. Операційна система Ubuntu Server забезпечує мінімалістичний інтерфейс без графічного оточення, що дозволяє оптимізувати використання системних ресурсів. Керування системою здійснюється через командний рядок (CLI), що надає адміністраторам повний контроль над сервером. Завдяки цьому Ubuntu Server є ефективним вибором для серверів з високими вимогами до продуктивності.

Ubuntu Server підтримує широкий спектр серверних застосунків. Для створення веб-серверів доступні Apache, Nginx або LiteSpeed, які забезпечують підтримку сучасних веб-технологій. Для баз даних можна використовувати MySQL, PostgreSQL, MariaDB чи MongoDB, що дозволяє налаштувати сервер для обробки великих обсягів даних. Якщо потрібен сервер додатків, можна розгорнути програмне забезпечення на базі Java, Node.js або Python.

Ubuntu Server також має відмінну підтримку мережевих служб. Це включає DNS-сервери (Bind9), DHCP-сервери (ISC DHCP), файлові сервери (Samba для SMB або NFS для Linux), а також сервери пошти (Postfix або Exim). Для підвищення безпеки система підтримує брандмауери (ufw, iptables, nftables) та засоби шифрування (OpenSSL, GnuPG), а також інтеграцію з VPN-рішеннями, такими як OpenVPN, WireGuard чи IPSec.

На рисунку 2.6 показано мережеві налаштування Ubuntu Linux Server.

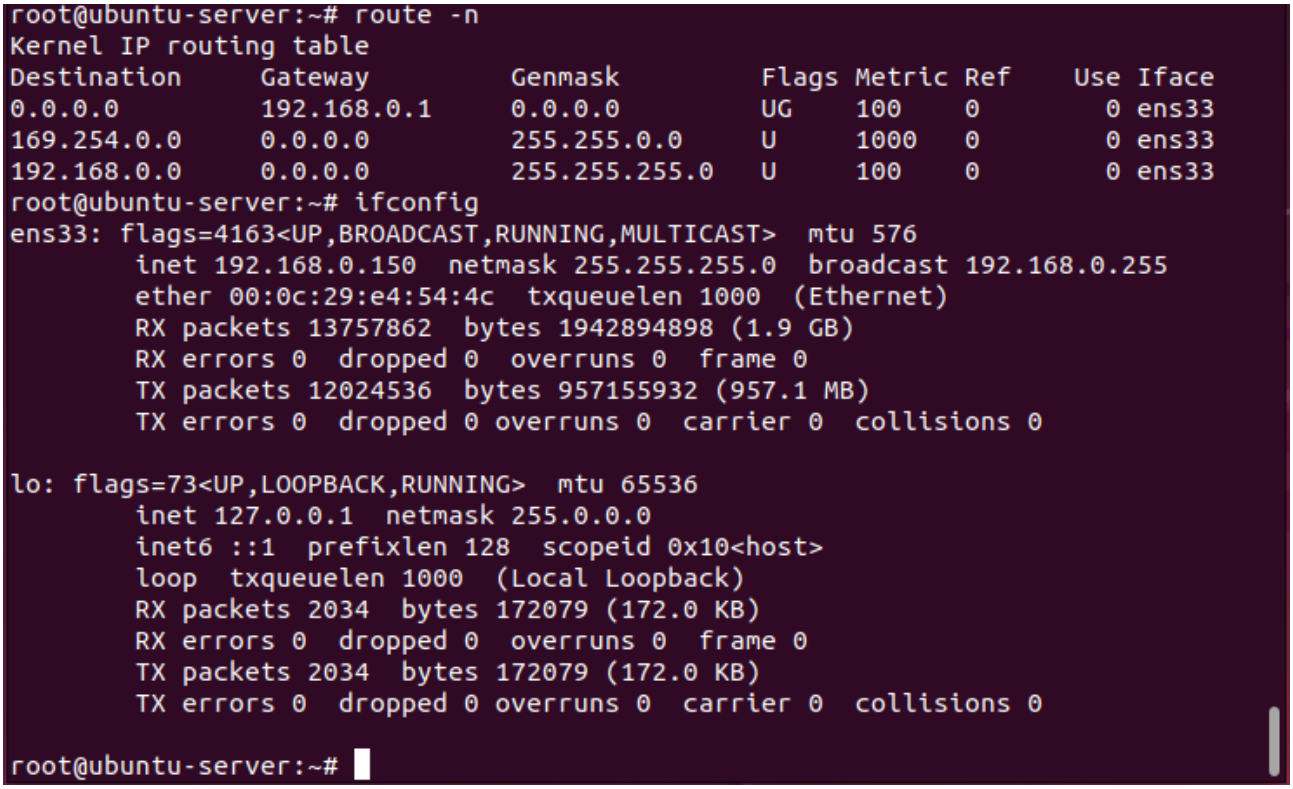

Рисунок 2.6 – Мережеві налаштування Ubuntu Linux Server

Команда `ifconfig` показує інформацію про мережеві інтерфейси. Команда `route -n` показує таблицю маршрутизації.

На рисунку 2.7 показано, що на хості працюють сервіси DNS, SSH, HTTP кожен з яких надає специфічні мережеві функції.

```
root@ubuntu-server:~# netstat -an | more
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:953           0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:953           0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:53            0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:53            0.0.0.0:*               LISTEN
tcp        0      0 192.168.0.150:53        0.0.0.0:*               LISTEN
tcp        0      0 192.168.0.150:53        0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN
tcp        0      0 192.168.0.150:39000     77.120.62.8:80          TIME_WAIT
tcp6       0      0 ::1:631                 :::*                    LISTEN
tcp6       0      0 ::1:953                 :::*                    LISTEN
tcp6       0      0 ::1:953                 :::*                    LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
tcp6       0      0 ::1:53                  :::*                    LISTEN
tcp6       0      0 ::1:53                  :::*                    LISTEN
```

Рисунок 2.7 – Статус сервісів в Ubuntu Linux Server

Сервер Ubuntu Linux має відкриті порти для різних сервісів, таких як SSH, HTTP, DNS. Встановлене DHCP-з'єднання підтверджує, що сервер отримує мережеві налаштування через DHCP.

Використання сервісів SSH, HTTP та DNS у лабораторному середовищі створює основу для моделювання реальних мережевих сценаріїв, що є важливим елементом у дослідженні кібербезпеки. Ці сервіси широко використовуються в реальних інфраструктурах, тому їх інтеграція в тестове середовище дозволяє імітувати типові умови, з якими стикаються системи IDS/IPS.

2.2.4 Операційна система Kali Linux

Kali Linux - це спеціалізована операційна система з відкритим кодом, розроблена для тестування на проникнення (penetration testing), аналізу вразливостей, цифрової криміналістики, зворотного інжинірингу та дослідження кібербезпеки [22]. Вона базується на дистрибутиві Debian Linux і містить понад 600 попередньо встановлених інструментів, які використовуються професіоналами з кібербезпеки для перевірки безпеки інформаційних систем.

Основною перевагою Kali Linux є її орієнтація на завдання в галузі кібербезпеки. Вона забезпечує доступ до потужного набору інструментів,

включаючи такі популярні утиліти, як Metasploit, Wireshark, Nmap, Burp Suite, John the Ripper, Aircrack-ng та багато інших. Ці інструменти дозволяють виконувати завдання з пошуку вразливостей, проведення атак, збору інформації, аналізу мережевого трафіку, зламу паролів, перевірки безпеки веб-додатків та моделювання реальних атак.

Kali Linux підтримує роботу в різних середовищах, включаючи локальні інсталяції, віртуалізацію, контейнеризацію за допомогою Docker та розгортання в хмарних сервісах, таких як AWS чи Azure. Крім того, вона підтримує роботу на мобільних пристроях і вбудованих системах завдяки ARM-версії для Raspberry Pi та інших платформ.

У контексті лабораторного середовища Kali Linux є ключовим інструментом для моделювання атак, таких як тестування на проникнення, імітація DDoS-атак, виконання атак типу Man-in-the-Middle, перевірка стійкості до SQL-ін'єкцій, атак на бездротові мережі чи спроб зламу паролів. Завдяки цьому Kali Linux дозволяє виявляти слабкі місця в захисних механізмах, налаштовувати та перевіряти ефективність систем IDS/IPS.

На рисунку 2.8 показано мережеві налаштування Kali Linux.

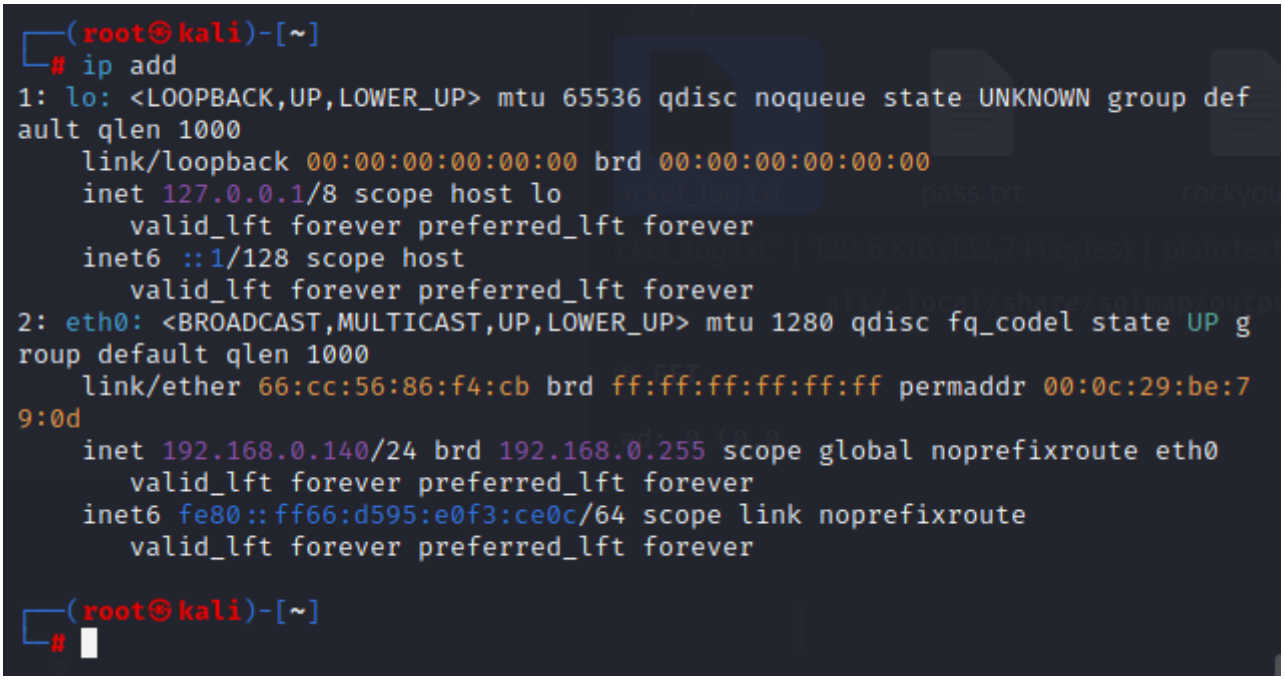

Рисунок 2.8 – Мережеві налаштування Kali Linux

Kali Linux є ідеальною платформою для створення та запуску Python-скриптів, призначених для симуляції різних типів атак. Python як універсальна мова програмування має численні бібліотеки, які дозволяють автоматизувати сценарії атак, тестувати мережі та аналізувати вразливості. Kali Linux постачається з попередньо встановленим Python, що робить її готовою для використання скриптів без додаткових налаштувань. Залежно від завдань, Python-скрипти можуть моделювати різні атаки, включаючи сканування портів, DDoS-атаки, брутфорс, SQL-ін’єкції, атаки на протоколи та багато іншого.

2.3 Тест працездатності лабораторного середовища

Тест працездатності лабораторного середовища є критично важливим етапом перед початком будь-яких досліджень або експериментів у сфері кібербезпеки. Цей процес дозволяє переконатися, що всі елементи інфраструктури функціонують належним чином і готові до моделювання реальних сценаріїв атак, перевірки захисних механізмів або виконання навчальних завдань. Відсутність цього тестування може призвести до непередбачуваних помилок, неточних результатів та втрати часу. Основна важливість цього етапу полягає в тому, що він забезпечує перевірку мережевої взаємодії між усіма вузлами середовища, включаючи Kali Linux, сервер Ubuntu Linux та інші компоненти. Це включає перевірку доступності IP-адрес, роботи мережевих інтерфейсів, шлюзів та маршрутизації. Без функціональної мережі виконання тестів або моделювання атак стає неможливим.

Крім того, тестування дозволяє впевнитися, що сервіси, які використовуються для моделювання атак, такі як SSH, HTTP та DNS, правильно налаштовані та працюють. Це включає перевірку їх доступності через відповідні порти та відповідність їхнього функціоналу цілям експерименту. На цьому етапі також виявляються потенційні технічні проблеми. Тест працездатності забезпечує точність результатів, оскільки лабораторне середовище має бути налаштованим максимально наближено до реальних умов, щоб результати тестів

були репрезентативними. Це також гарантує, що середовище готове до виконання всіх запланованих сценаріїв.

На рисунку 2.9 показано результат виконання команди ping на Kali Linux для перевірки доступності сервера Ubuntu Linux з IP-адресою 192.168.0.150.

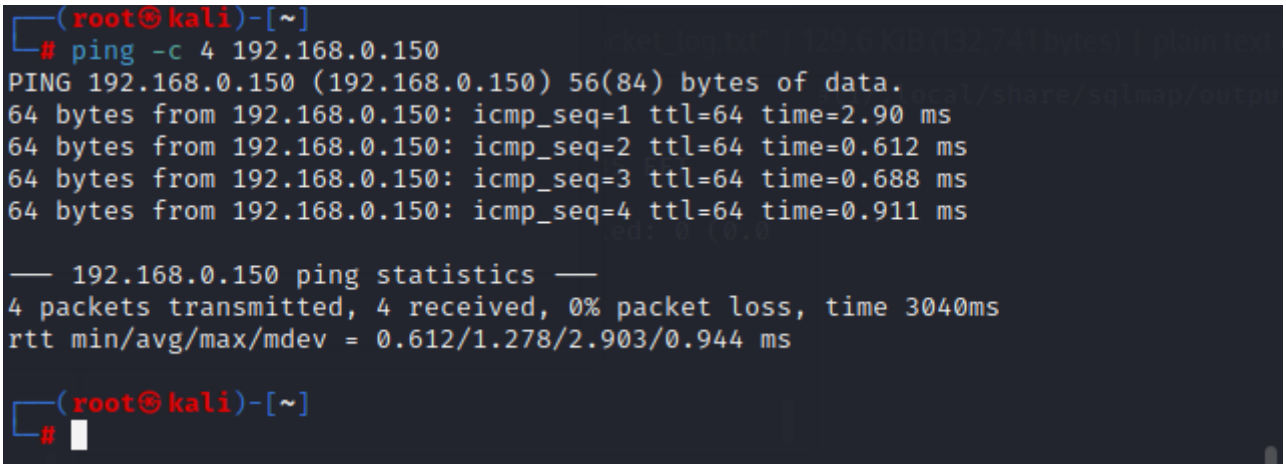

Рисунок 2.9 – Вивід команди `ping` в операційній системи Kali Linux

Цей результат підтверджує, що сервер Ubuntu Linux з IP-адресою 192.168.0.150 доступний для з'єднань, а мережеве з'єднання є стабільним. Відсутність втрат пакетів і низький час відгуку свідчать про хорошу якість з'єднання.

На рисунку 2.10 показано результат результат виконання команди `nmap -sS -p 1-1024 -sV 192.168.0.150` для сканування TCP-портів у діапазоні 1-1024 на сервері з IP-адресою 192.168.0.150. Результати показують стан портів, доступність сервісів і їхні версії.

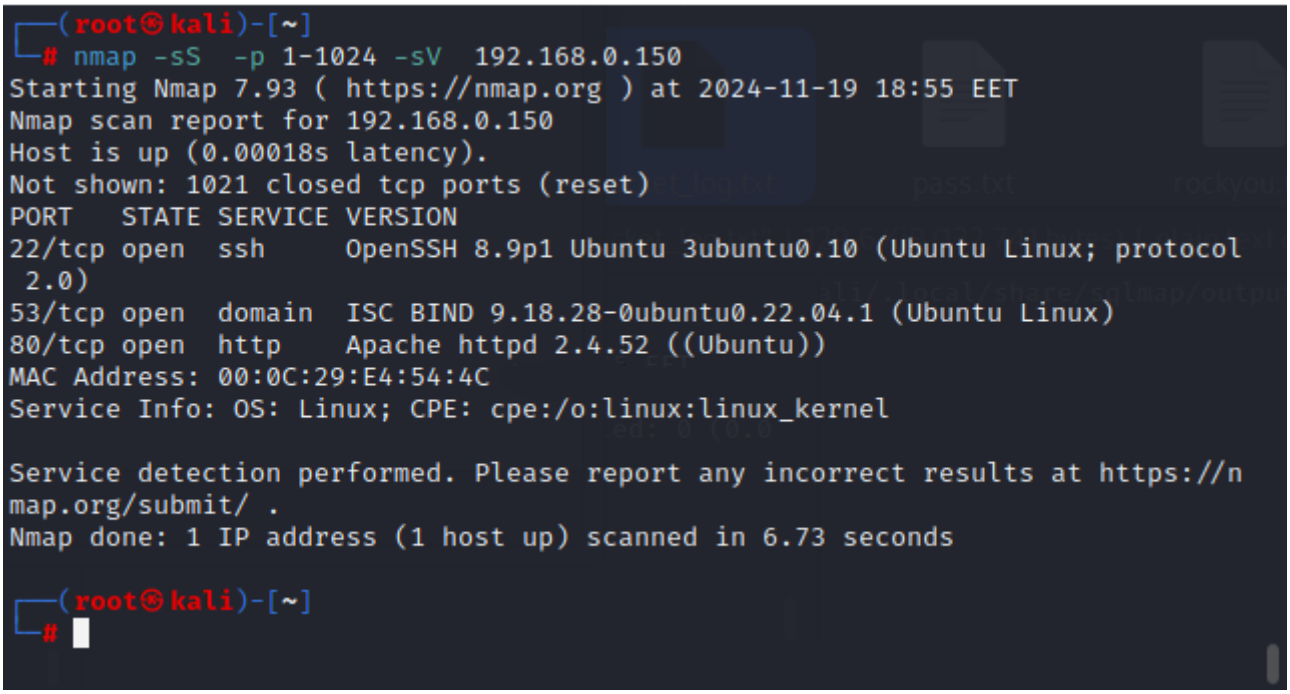

Рисунок 2.10 – Результати сканування відкритих портів в Ubuntu Linux Server

Результати підтверджують, що ці сервіси працюють та доступні в межах лабораторного середовища. Це означає, що лабораторне середовище готове до подальшого використання, зокрема для моделювання атак, тестування систем IDS або інших досліджень у сфері кібербезпеки.

2.4 Висновки до розділу 2

В другому розділі було розглянуто процес розгортання тестового лабораторного середовища для досліджень у сфері кібербезпеки. Детально описано схему лабораторного середовища, яке базується на гіпервізорі Hyper-V, що працює на платформі Windows Server 2022 Core. Це середовище створено для оцінки ефективності сигнатурного та поведінкового підходів IDS. Було представлено ключові компоненти середовища, такі як маршрутизатор pfSense з інтегрованою системою Suricata для моніторингу трафіку, сервер Ubuntu Linux із налаштованими сервісами DNS, SSH та HTTP, а також Kali Linux для моделювання атак.

Розділ акцентує увагу на важливості правильної організації та налаштування лабораторного середовища, оскільки це є основою для коректного проведення досліджень. Було розглянуто особливості налаштування віртуальної інфраструктури, зокрема використання віртуальних комутаторів для забезпечення ізоляції та безпеки, а також організацію зон із різними функціональними ролями. Окрему увагу приділено тестуванню працездатності середовища. Результати тестування довели, що всі елементи інфраструктури функціонують належним чином, сервіси доступні, а середовище готове для моделювання атак і аналізу ефективності захисних механізмів.

Таким чином, у другому розділі було створено фундамент для подальшого дослідження IDS, що дозволить провести аналіз її ефективності, точності виявлення та адаптивності до різних типів загроз.

РОЗДІЛ 3 СИГНАТУРНИЙ ТА ПОВЕДІНКОВИЙ ПІДХОДИ В СИСТЕМАХ IDS

3.1 Сигнатурний підхід виявлення загроз

Сигнатурний підхід виявлення загроз в IDS є одним із найпоширеніших і найефективніших методів для ідентифікації загроз у мережевому трафіку [23]. Цей підхід базується на використанні бази даних сигнатур, які являють собою заздалегідь визначені шаблони або унікальні характеристики відомих атак, що дозволяє швидко та точно виявляти загрози. Сигнатурний метод активно застосовується для виявлення різних типів атак, таких як SQL-ін’єкції, експлуатація відомих вразливостей у програмному забезпеченні тощо.

Основним принципом роботи сигнатурного підходу є порівняння вхідного трафіку з базою сигнатур. Кожна сигнатура являє собою опис характерних ознак атаки, які можуть включати певну послідовність байтів у даних, використання специфічних параметрів протоколів, аномалії в заголовках мережевих пакетів або послідовність команд. Якщо система IDS знаходить збіг між трафіком і однією з сигнатур у базі, вона генерує попередження про можливу загрозу. Таким чином, сигнатурний метод дозволяє виявляти загрози з високою точністю, якщо вони вже відомі та описані у вигляді сигнатури.

Однією з головних переваг сигнатурного підходу є його низький рівень хибних спрацювань. Завдяки чіткому визначенню шаблонів атак система IDS рідко сприймає легітимний трафік як загрозу, що зменшує кількість помилкових попереджень і полегшує роботу адміністраторів. Крім того, цей підхід дозволяє оперативно виявляти відомі загрози, забезпечуючи швидку реакцію на інциденти.

Однак сигнатурний підхід має і певні обмеження, які знижують його ефективність у сучасних умовах кіберзагроз. Одним із головних недоліків є залежність від бази сигнатур. Якщо атака є новою або модифікованою, система IDS не зможе її розпізнати, оскільки відповідна сигнатура ще не внесена до бази. Це робить сигнатурні системи вразливими до атак нульового дня та інших

інноваційних загроз. Крім того, для забезпечення актуальності захисту база сигнатур потребує регулярного оновлення. Відсутність своєчасних оновлень може суттєво знизити ефективність системи. Ще одним викликом є обробка великої кількості сигнатур у високонавантажених мережах, що може негативно вплинути на продуктивність системи та швидкість аналізу трафіку.

Сигнатурний підхід часто використовується в поєднанні з іншими методами, такими як поведінковий аналіз або машинне навчання. Такий комбінований підхід дозволяє компенсувати недоліки кожного методу та підвищити загальну ефективність системи IDS. Наприклад, поведінковий аналіз може виявляти нові та невідомі загрози, тоді як сигнатурний підхід забезпечує швидку і точну ідентифікацію вже відомих атак.

На рисунку 3.1 показано приклад налаштування сигнатури для системи IDS Suricata.

Рисунок 3.1 – Налаштування сигнатурного правила для системи IDS Suricata

Це правило створене для виявлення специфічного шаблону у TCP-трафіку, який відповідає зазначеному вмісту.

де:

- `alert` – означає тип дії, яку виконує система IDS при виявленні збігу з правилом. У цьому випадку Suricata створює сповіщення (alert);

- `tcp` – вказує, що правило застосовується до TCP-трафіку;

- `any any -> any any` – визначає напрямок та джерело/призначення трафіку:
  a) `any any` (зліва) означає, що правило застосовується до будь-якої IP-адреси та порту джерела;
  b) `->` вказує напрямок трафіку;
  c) `any any` (справа) означає, що правило застосовується до будь-якої IP-адреси та порту призначення.
- `msg:"Custom Pattern Detected"` – текстове повідомлення, яке генерується у випадку спрацювання правила. Це пояснює, чому саме було створено сповіщення;
- `content:"ABCD1234XYZ"` – основна умова спрацювання правила. Система IDS шукатиме в трафіку вміст, що відповідає тексту "ABCD1234XYZ";
- `sid:1000001` – унікальний ідентифікатор правила (Signature ID), який використовується для ідентифікації правила в базі даних;
- `rev:1` – номер ревізії правила. Він вказує версію правила, що корисно для оновлення або внесення змін;
- `classtype` – категорія загрози. У цьому випадку вказано, що загроза стосується порушення політики безпеки;
- `metadata` – метадані, що вказують дату створення та останнього оновлення правила. Це допомагає у відстеженні змін і оновлень.

В лабораторному середовищі дане правило буде використано для виявлення специфічного шаблону в TCP-трафіку, який містить вміст `"ABCD1234XYZ"`. Це дозволяє імітувати реальний сценарій, у якому система IDS реагує на заздалегідь відомі загрози або порушення політики безпеки. Правило буде застосовуватись для тестування ефективності сигнатурного підходу в системах виявлення вторгнень Suricata, зокрема здатності виявляти загрози, створювати сповіщення та правильно класифікувати загрозу.

На рисунку 3.1 показано інтерфейс, що дозволяє адміністратору керувати правилами Suricata в pfSense. Правило з SID 1000001 є активним і налаштованим для виявлення визначеного шаблону у TCP-трафіку, що відповідає сигнатурі.

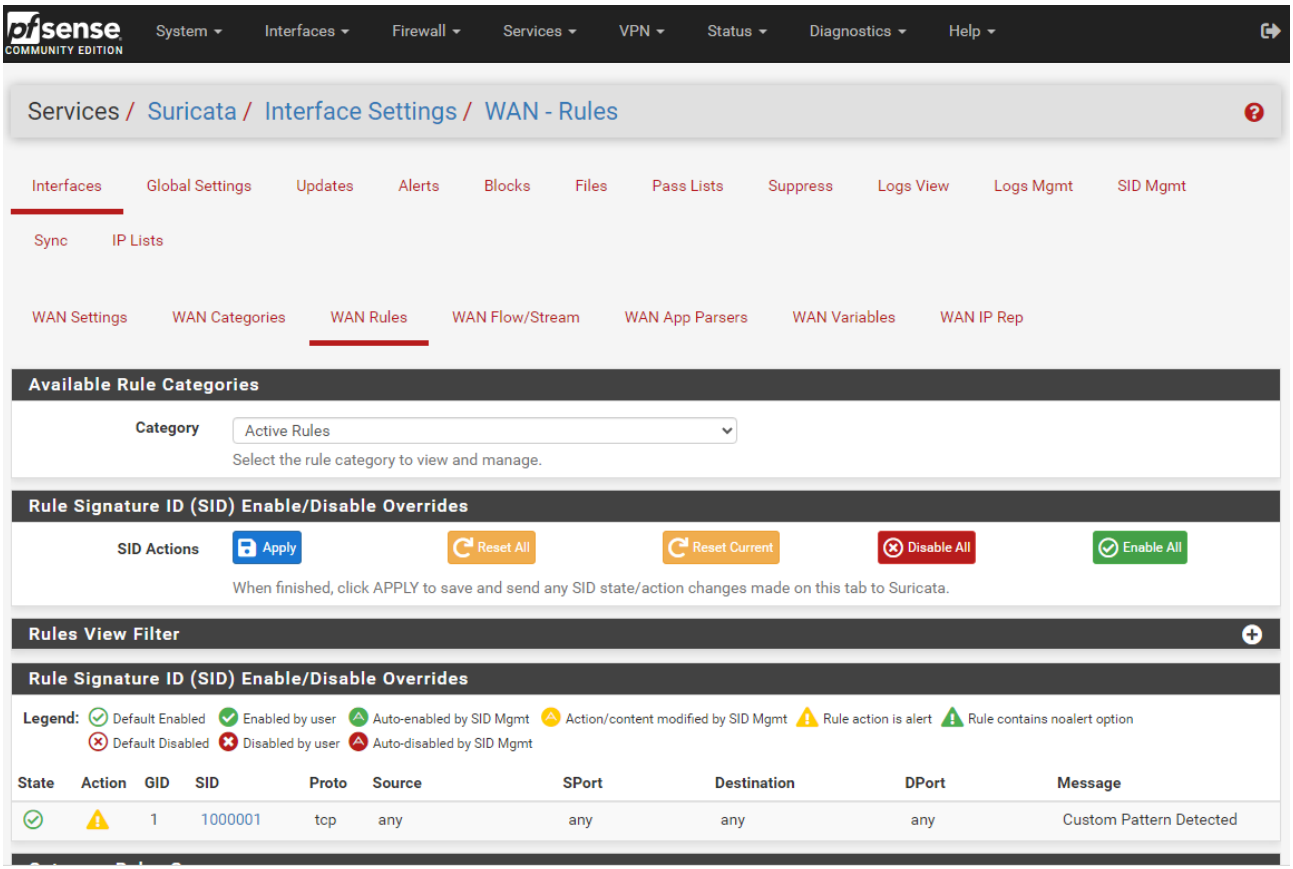

Рисунок 3.2 – Інтерфейс управління правилами Suricata в IDS1 pfSense

Основною метою використання цього правила є перевірка працездатності системи IDS у лабораторних умовах. Воно дозволяє оцінити, як система реагує на трафік, що відповідає певному шаблону, та як точно вона генерує сповіщення.

3.2 Поведінковий підхід виявлення загроз

Поведінковий підхід виявлення загроз у системах IDS є сучасним і гнучким методом, що дозволяє ідентифікувати загрози на основі аналізу відхилень у мережевій активності від встановлених норм [24]. На відміну від сигнатурного підходу, який працює з наперед визначеними шаблонами відомих атак, поведінковий метод базується на створенні базової лінії нормальної активності

та порівнянні поточного трафіку або дій користувачів з цією базовою моделлю. Виявлення аномалій є ключовим інструментом цього підходу.

Основою поведінкового підходу є аналіз великої кількості даних для формування профілю нормальної поведінки системи, користувачів або мережі. Такий профіль може враховувати обсяги трафіку, типові протоколи та порти, часові закономірності доступу до ресурсів, характерні шаблони використання додатків. Наприклад, якщо зазвичай сервер отримує певний обсяг запитів у визначені години, а раптово обсяг трафіку суттєво зростає або змінюється його структура, система може розцінити це як потенційну загрозу.

Поведінковий метод дозволяє виявляти складні або нові типи атак, які не мають чіткої сигнатури. Це особливо важливо для протидії атакам нульового дня, багатоетапним вторгненням або аномальній активності, пов'язаній із компрометацією облікових записів. Наприклад, поведінковий аналіз може допомогти виявити спроби крадіжки даних, що супроводжуються аномально високою кількістю запитів до бази даних, або виявити шкідливу активність, пов’язану зі змінами в схемі використання протоколів.

Проте поведінковий підхід має і свої обмеження. Одним із головних викликів є велика кількість хибних спрацювань. Через те, що будь-яке відхилення від базової лінії розцінюється як потенційна загроза, легітимна активність, яка виходить за межі нормальної, також може генерувати сповіщення. Наприклад, разове підвищення трафіку під час пікових навантажень або проведення технічних робіт може бути помилково інтерпретоване як атака. Для зменшення кількості хибних спрацювань необхідно точно налаштовувати параметри системи, що вимагає часу і ресурсів.

На рисунку 3.3 показано приклад правила для системи IDS Suricata, яке використовується для виявлення аномального обсягу трафіку.

Рисунок 3.3 – Налаштування поведінкового правила для системи IDS Suricata

Це правило демонструє, як поведінковий підхід може бути реалізований у системах виявлення загроз.

де:

- `alert` – означає тип дії, яку виконує система IDS при виявленні збігу з правилом. У цьому випадку Suricata створює сповіщення (alert);
- `tcp` – вказує, що правило застосовується до TCP-трафіку;
- `any any -> any 80` – визначає напрямок трафіку:
  a) `any any` означає, що правило застосовується до будь-якої IP-адреси та порту джерела;
  b) `any 80` означає, що трафік спрямований на порт 80 (HTTP).
- `msg:"Abnormal Traffic Volume Detected"` – текст повідомлення, яке генерується системою IDS у разі спрацювання правила. Повідомлення вказує на виявлення аномального обсягу трафіку;
- `flags` – фільтр, який визначає, що правило застосовується до пакетів із прапором SYN, тобто на стадії встановлення TCP-з'єднання;
- `threshold both, track by_src, count 100, seconds 60` – ключова умова правила:
  a) `type both` – визначає, що аномальний трафік враховується як для сповіщення, так і для блокування;
  b) `track by_src` – відстежується джерело трафіку;

с) `count 100, seconds 60` – якщо за 60 секунд надходить 100 і більше SYN-запитів від одного джерела, система IDS сприймає це як загрозу.

- `sid:1000002` – унікальний ідентифікатор правила (Signature ID), що використовується для його ідентифікації;
- `rev:1` – номер ревізії правила, який вказує на його версію;
- `classtype` – класифікація загрози. У цьому випадку це спроба здійснення DoS-атаки;
- `metadata` – метадані, які містять інформацію про дату створення та останнього оновлення правила.

У лабораторному середовищі поведінковий підхід надає можливість тестувати здатність системи IDS виявляти невідомі загрози або нові типи атак. Наприклад, імітація DDoS-атак дозволяє перевірити, наскільки ефективно система розпізнає відхилення від базового рівня трафіку. Крім того, поведінковий аналіз може бути застосований для моніторингу активності користувачів, що дозволяє виявляти компрометацію облікових записів або зловмисне використання привілеїв.

На рисунку 3.4 показано інтерфейс, що дозволяє адміністратору керувати правилами Suricata в pfSense. Правило з SID 1000002 є активним і налаштованим для моніторингу та аналізу аномального мережевого трафіку.

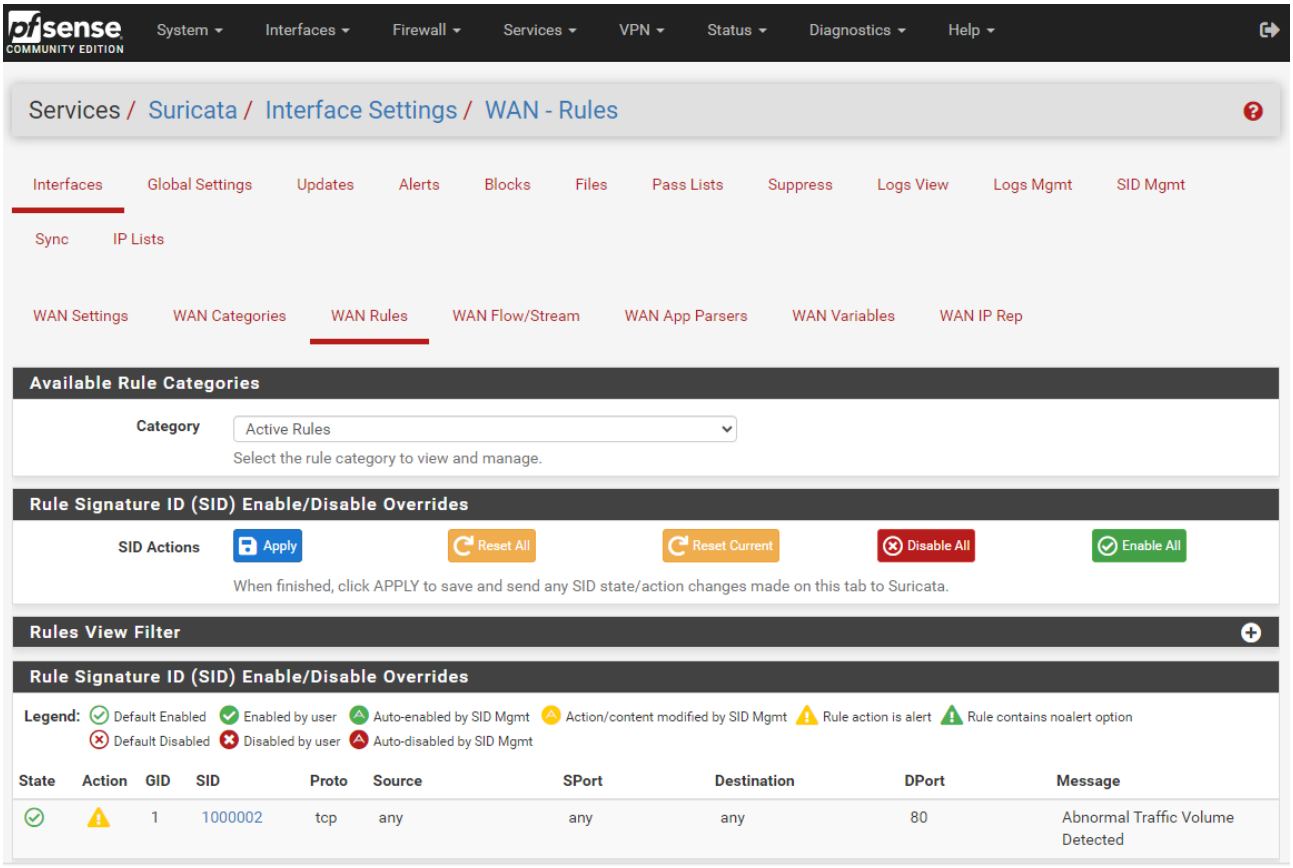

Рисунок 3.4 – Інтерфейс управління правилами Suricata в IDS2 pfSense

Це правило налаштовується для виявлення аномально великого обсягу трафіку, що надходить до порту HTTP (80). Воно може сигналізувати про потенційну DoS-атаку або іншу аномалію, яка порушує нормальну поведінку мережі. Повідомлення "Abnormal Traffic Volume Detected" дозволяє адміністратору легко ідентифікувати проблему.

Дане правило буде використано у лабораторному середовищі для аналізу здатності системи Suricata виявляти відхилення від нормального трафіку. Це дозволяє оцінити ефективність поведінкового підходу у виявленні DoS-атак та інших аномалій, які не завжди можна передбачити за допомогою статичних сигнатур.

3.3 Методи моделювання кіберзагроз

Моделювання кіберзагроз є невід’ємною частиною сучасних досліджень у сфері кібербезпеки. Воно дозволяє відтворити реальні сценарії атак у контрольованому середовищі з метою тестування та оцінки ефективності

захисних механізмів. Цей процес включає створення та симуляцію різноманітних загроз, що відображають сучасні методи зловмисників, і дозволяє ідентифікувати слабкі місця в системах безпеки. Ключовим аспектом моделювання є створення умов, які максимально наближені до реальних. Одним із важливих етапів моделювання є визначення мети атаки. Це може бути порушення конфіденційності, доступності або цілісності даних. Наприклад, атаки на конфіденційність можуть включати перехоплення трафіку або злом паролів, тоді як атаки на доступність часто реалізуються через перевантаження серверів. Для цілей моделювання використовується підхід, що включає як відомі техніки, так і створення нових сценаріїв, які можуть не мати прецедентів у базах даних сигнатур.

Значну роль у моделюванні кіберзагроз відіграють автоматизовані інструменти, які дозволяють ефективно імітувати складні атаки. Такі інструменти, як nmap, можуть бути використані для сканування мереж, визначення відкритих портів і служб, що працюють на них, а Metasploit Framework дозволяє виконувати експлуатацію вразливостей.

Використання спеціалізованих python-скриптів у Kali Linux є одним із найефективніших способів моделювання кіберзагроз у лабораторному середовищі. Python, як універсальна мова програмування, пропонує багатий набір бібліотек і модулів для роботи з мережею, що дозволяє створювати скрипти для різних типів атак [25]. У середовищі Kali Linux python інтегровано за замовчуванням, що дає змогу безпосередньо створювати та виконувати сценарії без додаткової інсталяції. Python-скрипти в контексті кіберзагроз використовуються для симуляції різних типів атак, таких як сканування портів, DDoS-атаки, SQL-ін’єкції, brute force, атаки типу Man-in-the-Middle. Наприклад, використовуючи бібліотеку scapy, можна створювати спеціалізовані пакети для модифікації заголовків IP, TCP або UDP, що дозволяє генерувати трафік, характерний для певного типу атак. Для симуляції DoS-атак python надає прості рішення для створення великого обсягу трафіку за допомогою багатопотоковості. З використанням бібліотек socket і threading можна генерувати численні запити до цільового сервера, імітуючи навантаження, яке

характерне для атаки на доступність (DoS або DDoS). У поєднанні з бібліотекою random можна створити варіативність у вхідних запитах, що робить атаку більш реалістичною.

Для моделювання атак типу brute force python може використовувати бібліотеку paramiko для підбору паролів до SSH-сервісів. Це дозволяє перевірити стійкість захисту від підбору паролів у реальному середовищі. SQL-ін'єкції також можуть бути змодельовані за допомогою python-скриптів. Використовуючи бібліотеки requests або urllib, можна створити автоматизовані запити до веб-додатків із включенням шкідливого коду в поля введення. Це дозволяє перевірити, чи веб-додаток має достатні механізми для захисту від введення небезпечного SQL-коду. Окремим напрямом є створення аномального трафіку для перевірки поведінкового підходу виявлення загроз. Python дозволяє генерувати раптові піки активності, змінювати структуру трафіку або вводити нетипові дані в запити.

Використання python у Kali Linux дає можливість автоматизувати процес тестування, що особливо важливо в умовах, коли потрібна перевірка великої кількості сценаріїв. Наприклад, можна створити скрипт, який автоматично змінює IP-адреси джерела, порти, вміст пакетів і навіть час генерації трафіку. Це дозволяє моделювати складні загрози, які можуть бути складно виявити за допомогою стандартних методів.

У лабораторному середовищі використання python-скриптів у Kali Linux дозволяє тестувати ефективність захисних систем, таких як IDS, оцінювати точність виявлення загроз і перевіряти стійкість мережевих інфраструктур до реальних атак. Це не лише сприяє вдосконаленню методів захисту, але й забезпечує навчання фахівців у максимально наближених до реальних умов сценаріях.

В додатку Б наведено python-скрипт, який використовує бібліотеку scapy для створення та надсилання спеціалізованих мережевих пакетів з метою моделювання кіберзагроз та тестування сигнатурних та поведінкових правил в IDS.

На рисунку 3.5 показано фрагмент коду, який містить імпорт необхідних бібліотек і функції для перевірки прав доступу (root-привілеїв) і для логування подій.

```
import sys
import time
import os
from scapy.all import *

# Function to check for root privileges
def check_root():
    if not os.geteuid() == 0:
        sys.exit("This script must be run with administrator (root) privileges.")

# Function for logging
def log_message(message):
    with open("packet_log.txt", "a") as log_file:
        log_file.write(message + "\n")
```

Рисунок 3.5 – Фрагмент коду імпорту бібліотек і функцій перевірки прав доступу та для логування подій

Цей фрагмент коду починається з імпорту бібліотек, необхідних для виконання функцій скрипта. Імпортуються модулі `sys`, `time` та `os`. Модуль `sys` використовується для взаємодії з інтерпретатором python, наприклад, для завершення виконання скрипта за допомогою `sys.exit()`. Модуль `time` дозволяє працювати з часовими затримками, а модуль `os` використовується для взаємодії з операційною системою, наприклад, для перевірки ідентифікатора користувача або роботи з файлами. Додатково імпортується бібліотека `scapy`, яка потрібна для створення, модифікації та надсилання мережевих пакетів.

Функція `check_root()` виконує перевірку наявності прав адміністратора. Вона використовує метод `os.geteuid()`, який повертає ідентифікатор поточного користувача. У системах Linux ідентифікатор root дорівнює нулю. Якщо ідентифікатор не дорівнює нулю, функція завершує виконання програми, використовуючи `sys.exit()`, і виводить повідомлення, що скрипт повинен запускатися з правами адміністратора. Це необхідно для виконання операцій із сирими мережевими пакетами, які потребують підвищених привілеїв.

Функція `log_message(message)` забезпечує запис повідомлень у файл `packet_log.txt`. Вона відкриває файл у режимі запису, використовуючи

конструкцію `with open`, яка гарантує автоматичне закриття файлу після завершення роботи з ним. Повідомлення, передане у функцію, записується в кінець файлу разом із символом нового рядка, що дозволяє фіксувати кожне повідомлення на окремому рядку. Ця функція важлива для ведення журналу подій, таких як запис інформації про відправлені пакети або будь-які інші дії, виконані скриптом.

Код забезпечує два критично важливі процеси: перевірку привілеїв користувача для безпечного виконання завдань та збереження інформації про виконані дії для подальшого аналізу або діагностики.

На рисунку 3.6 показано фрагмент коду, що визначає чотири параметри за замовчуванням, які використовуються для надсилання мережевих пакетів у скрипті.

```
# Default parameters
default_dst_ip = "192.168.0.150"
default_port = 80
default_count = 10
default_delay = 0.01  # Delay between packets in seconds
```

Рисунок 3.6 – Фрагмент коду встановлення параметрів за замовчуванням

Змінна `default_dst_ip` встановлює IP-адресу за замовчуванням, на яку будуть надсилатися пакети. У цьому випадку це адреса Ubuntu Linux Server. Вона вказує на цільову машину в локальній мережі. Змінна `default_port` задає порт за замовчуванням, до якого спрямовуватимуться пакети. У цьому прикладі це порт 80. Змінна `default_count` встановлює кількість пакетів, які мають бути відправлені. Значення за замовчуванням - 10, що означає. Змінна `default_delay` визначає затримку між відправленням кожного пакета в секундах. Значення 0.01 задає затримку 10 мілісекунд, що забезпечує високу частоту відправлення пакетів, але з можливістю контролю швидкості.

Ці параметри за замовчуванням можуть бути змінені користувачем через аргументи командного рядка. Вони забезпечують початкові значення, які дозволяють виконати тест без додаткових налаштувань.

На рисунку 3.7 показано фрагмент коду, що визначає функцію `parse_arguments()`, яка відповідає за обробку аргументів командного рядка, що передаються при запуску скрипта. Вона дозволяє змінювати параметри, використовуючи введені значення, або застосовувати параметри за замовчуванням, якщо аргументи не вказані.

```
# Parse command-line arguments
def parse_arguments():
    dst_ip = default_dst_ip
    port = default_port
    count = default_count
    delay = default_delay

    if len(sys.argv) > 1:
        dst_ip = sys.argv[1]  # Destination IP address
    if len(sys.argv) > 2:
        port = int(sys.argv[2])  # Destination port
    if len(sys.argv) > 3:
        count = int(sys.argv[3])  # Number of packets
    if len(sys.argv) > 4:
        delay = float(sys.argv[4])  # Delay between packets

    return dst_ip, port, count, delay
```

Рисунок 3.7 – Функція обробки аргументів командного рядка

На початку функції змінні `dst_ip`, `port`, `count` та `delay` ініціалізуються значеннями за замовчуванням, які були визначені раніше: `default_dst_ip`, `default_port`, `default_count` та `default_delay`. Функція перевіряє, чи було передано аргументи командного рядка. Для цього використовується метод `len(sys.argv)`, який визначає кількість аргументів, включаючи ім'я самого скрипта. Перевірка `if len(sys.argv) > 1` означає, що якщо користувач передав більше одного аргументу (адресу призначення), значення змінної `dst_ip` буде замінено на перший аргумент командного рядка `sys.argv[1]`. Умова `if len(sys.argv) > 2` перевіряє наявність другого аргументу. Якщо він вказаний, змінна `port` змінюється на значення другого аргументу командного рядка `sys.argv[2]`. Значення аргументу конвертується в ціле число за допомогою `int()`. Далі, умова `if len(sys.argv) > 3` перевіряє, чи передано третій аргумент. У разі його наявності змінна `count` приймає значення третього аргументу, конвертованого в ціле число. Остання перевірка `if len(sys.argv)`

`> 4` визначає, чи є четвертий аргумент. Якщо так, значення змінної `delay` змінюється на четвертий аргумент, конвертований у число з плаваючою крапкою за допомогою `float()`.

Після обробки всіх аргументів функція повертає чотири змінні: `dst_ip, port, count, delay`. Ці значення можуть бути використані в інших частинах скрипта для налаштування параметрів його роботи. Функція дозволяє гнучко конфігурувати поведінку скрипта залежно від вимог користувача, що значно підвищує його зручність у використанні.

На рисунку 3.8 показано фрагмент коду, що визначає змінну `custom_string`, яка містить рядок "ABCD1234XYZ". Цей рядок виступає унікальним шаблоном (патерном), який буде включено в навантаження (payload) мережевих пакетів, створених і відправлених скриптом.

```
# Custom pattern string
custom_string = "ABCD1234XYZ"
```

Рисунок 3.8 – Встановлення значення сигнатури

Цей рядок використовується для ідентифікації пакетів, згенерованих під час виконання. Наприклад, системи IDS, налаштовані на виявлення цього шаблона, можуть ідентифікувати пакети як загрозу. Наявність такого спеціального рядка дало можливість налаштовувати правила в системах IDS для цільового тестування їхньої ефективності.

Цей рядок може бути змінений відповідно до потреб користувача або експерименту, що дозволяє легко модифікувати параметри тестування та адаптувати скрипт під конкретні вимоги.

На рисунку 3.9 показано фрагмент коду, що визначає функцію `create_and_send_packet`, яка відповідає за створення та відправку мережевих пакетів на задану IP-адресу та порт.

```python
# Function to create and send packets
def create_and_send_packet(dst_ip, port, count, delay):
    for i in range(count):
        pkt = IP(dst=dst_ip) / TCP(dport=port, flags="S") / Raw(load=custom_string)

        # Send packet without verbose output
        send(pkt, verbose=False)

        # Log the packet (commented out)
        log_message(f"Sent packet {i+1} to {dst_ip}:{port}")

        # Delay between packet sends
        time.sleep(delay)

        # Print status every 100 packets
        if (i + 1) % 100 == 0:  # Every 100 packets
            print(f"Sent {i + 1} packets.")
```

Рисунок 3.9 – Функція надсилання пакетів

На початку функції запускається цикл `for i in range(count`. На кожній ітерації циклу створюється новий мережевий пакет. Пакет `pkt` формується з трьох основних частин. `IP(dst=dst_ip)` визначає IP-заголовок із вказанням IP-адреси отримувача, заданої параметром `dst_ip`. `TCP(dport=port, flags="S")` формує TCP-заголовок із зазначенням порту отримувача `port` та прапора SYN `(flags="S")`, який використовується для ініціації TCP-з'єднання. `Raw(load=custom_string)` додає до пакета корисне навантаження (payload) у вигляді рядка, збереженого в змінній `custom_string`. Після створення пакет відправляється за допомогою функції `send(pkt, verbose=False)`. Аргумент `verbose=False` вимикає вивід додаткової інформації про процес відправлення, що знижує шум у консолі. Рядок `log_message(f"Sent packet {i+1} to {dst_ip}:{port}")` використовується для журналювання кожного відправленого пакета. Функція `time.sleep(delay)` додає затримку між відправленням кожного пакета. Значення затримки задається параметром `delay`.

Ця функція забезпечує контрольований процес створення та відправки мережевих пакетів, дозволяючи налаштовувати їхню кількість, параметри і затримку, що робить її корисною для тестування мереж або симуляції загроз.

На рисунку 3.10 показано фрагмент коду, що є секцією скрипта, яка забезпечує послідовне виконання функцій для перевірки прав доступу, отримання параметрів і відправки мережевих пакетів.

```
# Check for root privileges
check_root()

# Get parameters
dst_ip, port, count, delay = parse_arguments()

# Send packets
create_and_send_packet(dst_ip, port, count, delay)

print(f"Sent {count} packets to {dst_ip}:{port}. Done.")

#sudo python3 content_custom3.py <IP> <port> <count> <delay>
#sudo python3 content_custom3.py 192.168.0.150 80 10 0.01
```

Рисунок 3.10 – Фрагмент коду послідовного виконання функцій скрипта

Перша команда `check_root()` викликає функцію, яка перевіряє, чи має скрипт права адміністратора. Якщо скрипт запущений без цих прав, виконання буде припинено із повідомленням про помилку. Це забезпечує, що операції на низькому рівні, такі як створення і відправка мережевих пакетів, виконуються коректно. Далі йде виклик функції `parse_arguments()`, яка обробляє параметри, передані в командному рядку, і повертає значення змінних `dst_ip`, `port`, `count`, і `delay`. Ці параметри визначають IP-адресу отримувача, порт, кількість пакетів і затримку між їх відправленнями. Функція `create_and_send_packet(dst_ip, port, count, delay)` виконує основну задачу скрипта - створює і відправляє пакети на вказану адресу і порт, використовуючи задані параметри. Ця функція формує пакети, додає до них корисне навантаження і забезпечує потрібну кількість і затримку. Останній рядок `print(f"Sent {count} packets to {dst_ip}:{port}. Done.")` виводить підсумкове повідомлення в консоль, яке інформує користувача про завершення роботи скрипта. Воно включає інформацію про кількість відправлених пакетів, цільову IP-адресу і порт. Закоментовані рядки надають приклади запуску скрипта.

3.4 Тестування систем IDS

Тестування систем IDS з використанням python-скрипта та сигнатурного й поведінкового правил дозволяє оцінити ефективність цих систем у реальних сценаріях кіберзагроз. Основна мета цього тестування полягає у перевірці

здатності систем IDS ідентифікувати, фіксувати та реагувати на різні типи атак у лабораторному середовищі, зокрема ті, що моделюються за допомогою спеціалізованого скрипта.

Під час тестування python-скрипт генерує трафік, який відповідає певним характеристикам загроз. Скрипт створює спеціалізовані пакети, які містять у своєму корисному навантаженні визначений патерн "ABCD1234XYZ".

На першому етапі тестування за допомогою python-скрипта `content_custom3.py` в операційній системі Kali Linux виконаємо перевірку коректності спрацювання сигнатурного правила в IDS1.

На рисунку 3.11 демонструється результат виконання python-скрипта `content_custom3.py` в операційній системі Kali Linux. Скрипт був запущений із правами адміністратора за допомогою команди sudo, що є необхідним для виконання операцій на низькому рівні мережі, таких як створення та відправка спеціалізованих пакетів.

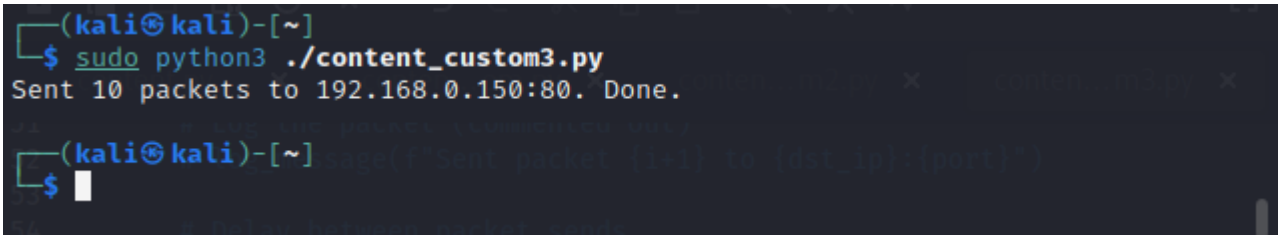

Рисунок 3.11 – Результат виконання python-скрипта `content_custom3.py` в операційній системі Kali Linux на першому етапі тестування

Сигнатурне правило в IDS1 Suricata виявило ці пакети через наявність визначеного патерну у корисному навантаженні.

На з рисунку 3.12 представлено інтерфейс розділу "Alerts" системи IDS1 Suricata, що працює в середовищі pfSense. У даному розділі відображаються журнали сповіщень про загрози, виявлені на основі активних правил системи IDS.

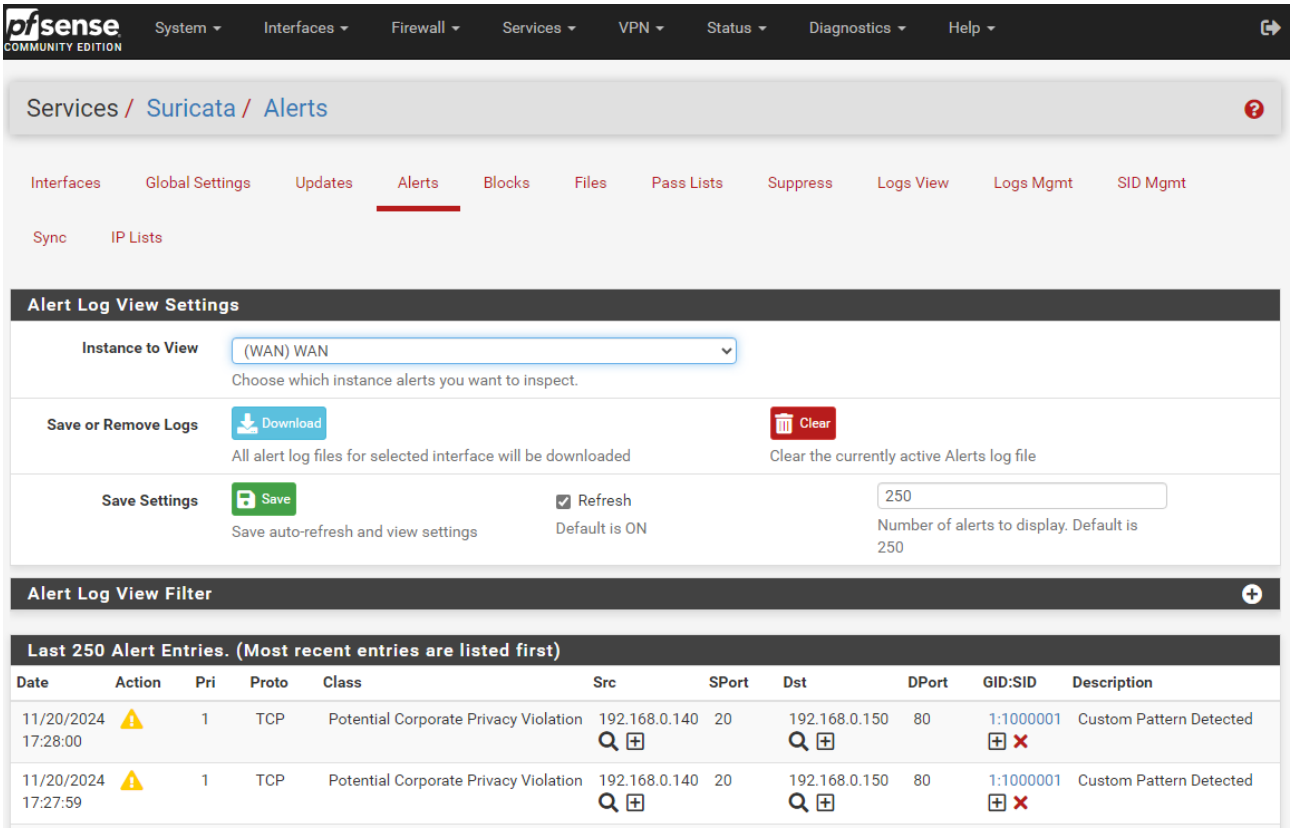

Рисунок 3.12 – Журнали сповіщень про загрози на першому етапі тестування IDS1

На рисунку відображені деталі сповіщень. Поле "Description" надає короткий опис виявленої загрози, у даному випадку - "Custom Pattern Detected". Також відображається ідентифікатор правила (GID:SID), що використовується для виявлення загрози "1:1000001". Цей запис сповіщень свідчить про те, що правило з ідентифікатором "1:1000001" спрацювало на трафіку, надісланому з IP-адреси 192.168.0.140 до IP-адреси 192.168.0.150 через порт 80. Це відповідає тестовим пакетам, створеним python-скриптом, які містять визначений патерн "ABCD1234XYZ". Система Suricata успішно ідентифікувала цей патерн, відповідно до налаштованого сигнатурного правила.

В свою чергу IDS2, налаштований за допомогою поведінкового правила, не виявив атаку. Це пояснюється тим, що поведінковий підхід базується на аналізі аномалій у трафіку відносно визначеної базової лінії нормальної активності. У даному тестуванні відправка лише 10 пакетів із використанням Python-скрипта не перевищила порогові значення, визначені у поведінковому правилі. Таким чином, система IDS2 не розцінила цей трафік як аномальний або потенційно

шкідливий. Це демонструє, що поведінковий підхід має чіткі обмеження у виявленні низькоінтенсивних атак або подій, які залишаються в межах нормальної активності, визначеної для мережі.

На другому етапі тестування було внесено зміни до python-скрипта `content_custom3.py`. Зокрема, було змінено визначений патерн на "ABCD-1234XYZ", а змінну `default_count` встановлено у значення 1000, що суттєво збільшує кількість надісланих пакетів. Після цього скрипт було повторно виконано для генерації трафіку.

Як видно з рисунку 3.13, система IDS1 Suricata, налаштована для використання сигнатурного підходу, атаку не виявила.

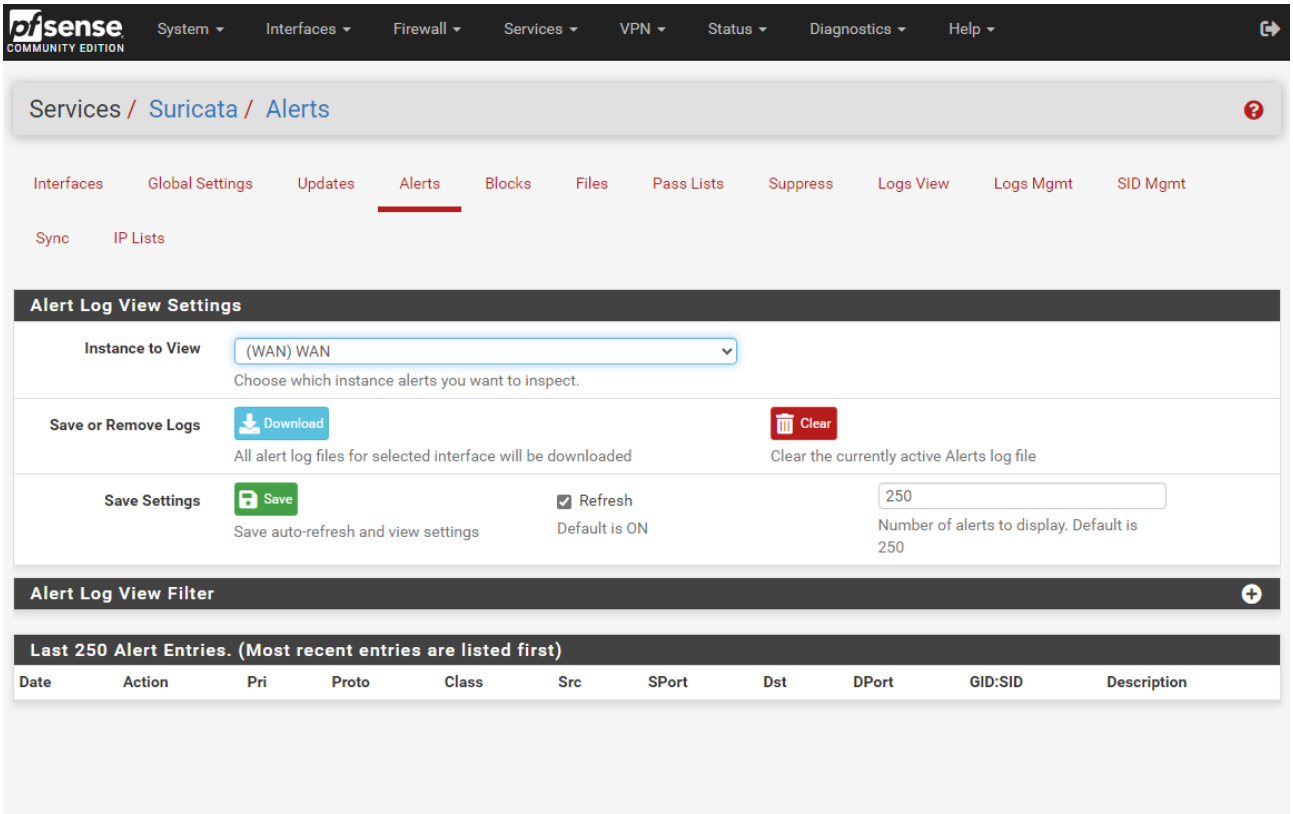

Рисунок 3.13 – Журнали сповіщень про загрози на другому етапі тестування IDS1

Це пояснюється тим, що змінений патерн "ABCD-1234XYZ" більше не відповідає сигнатурі, яка була визначена у правилі для IDS1. Сигнатурний підхід залежить від точного збігу зі встановленими шаблонами, і будь-яка зміна у патерні чи даних може призвести до того, що атака залишиться непоміченою. Це

підтверджує чутливість сигнатурного підходу до зміни характеристик трафіку та необхідність регулярного оновлення бази сигнатур для забезпечення ефективності системи.

На рисунку 3.14 показано результати роботи системи IDS2 Suricata під час другого етапу тестування. IDS2, який налаштований із використанням поведінкового правила, успішно виявив атаку.

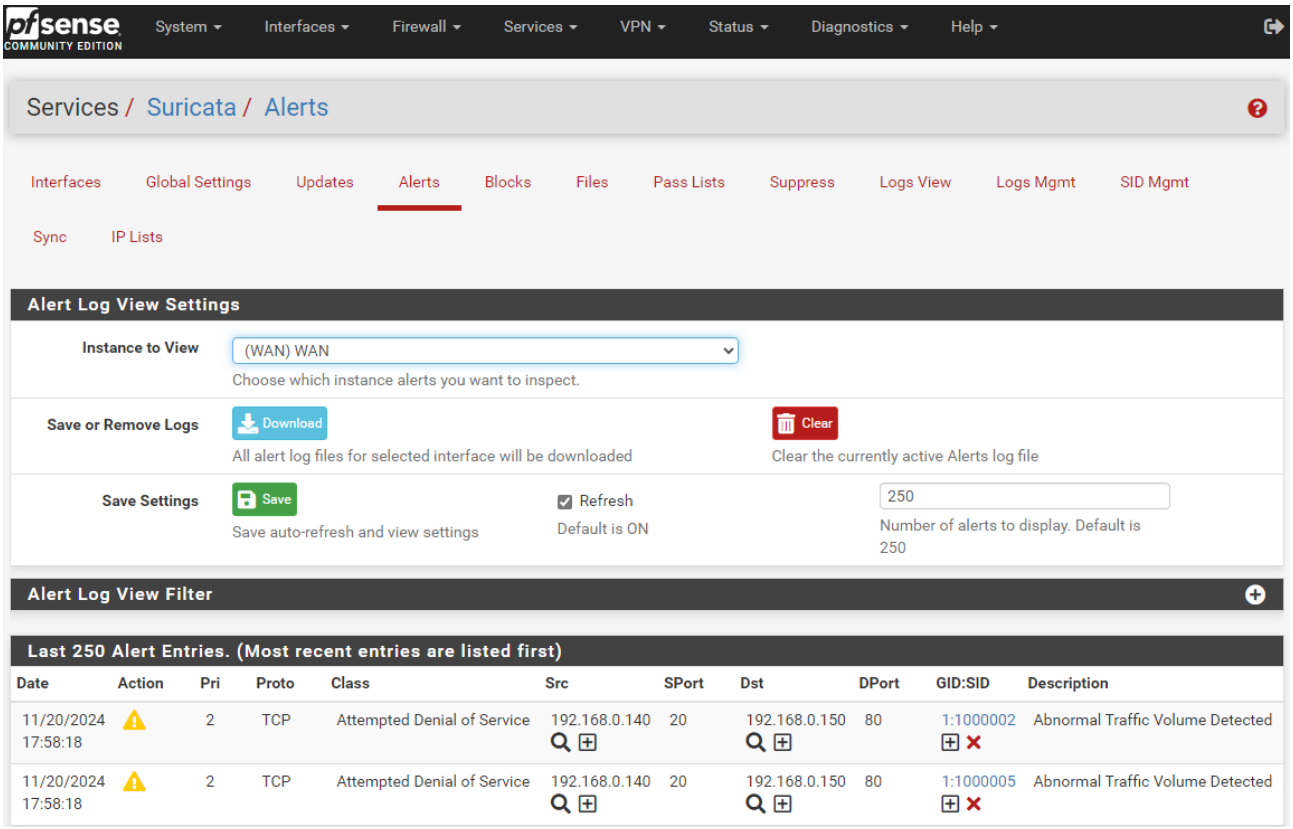

Рисунок 3.14 – Журнали сповіщень про загрози на другому етапі тестування IDS2

Це підтверджується спрацюванням правила з описом "Abnormal Traffic Volume Detected" для TCP-трафіку, що генерувався скриптом із параметрами зміненої кількості пакетів (1000) та модифікованого патерну ("ABCD-1234XYZ").

Поведінковий підхід IDS2 зреагував на аномально великий обсяг трафіку, який перевищив допустиму базову лінію. Це демонструє, що поведінковий метод є ефективним у випадках, коли характеристика трафіку відхиляється від нормальної активності, незалежно від специфічного вмісту даних. На відміну від

сигнатурного підходу IDS1, який не спрацював через відсутність точного збігу патерну, IDS2 орієнтується на аналіз обсягу та характеру трафіку, що дозволило йому виявити атаку навіть після зміни ключових параметрів.

Таким чином, результати тестування підтверджують перевагу поведінкового підходу в умовах, коли атака має нові або змінені параметри, які ще не внесені до бази сигнатур.

У результаті проведеного тестування було оцінено ефективність сигнатурного та поведінкового підходів у системах IDS в умовах моделювання реальних атак.

Сигнатурний підхід (IDS1 Suricata) продемонстрував високу точність у виявленні відомих загроз за умови наявності чітко визначених сигнатур. Під час першого етапу тестування система успішно зафіксувала атаку, оскільки патерн у трафіку повністю відповідав сигнатурі, зазначеній у правилі. Однак, у другому етапі, коли вхідні дані були змінені (модифіковано патерн та збільшено обсяг трафіку), сигнатурний підхід виявився неефективним. Це свідчить про основний недолік цього методу: залежність від оновлення бази сигнатур і нездатність реагувати на нові або змінені загрози.

Поведінковий підхід (IDS2 Suricata), натомість, продемонстрував гнучкість і здатність адаптуватися до нових умов. Під час першого етапу тестування система не зафіксувала аномалій, оскільки трафік не перевищив базову лінію поведінки. Однак у другому етапі, із значним збільшенням обсягу трафіку, поведінкове правило успішно спрацювало, виявивши аномальну активність. Це підтверджує перевагу цього підходу в умовах, коли характеристики загрози невідомі або змінюються.

Таким чином, сигнатурний підхід є ефективним для виявлення відомих загроз, але його обмеження виявляються при роботі з новими або зміненими атаками. Поведінковий підхід демонструє кращу адаптивність і здатність виявляти аномальні сценарії, але може бути менш точним у деяких випадках, що потребує додаткової оптимізації правил і налаштувань для зниження рівня хибних спрацювань. Загалом, поведінковий підхід є важливим компонентом сучасних систем IDS. Він доповнює сигнатурний метод, забезпечуючи

можливість виявлення складних загроз, які не можуть бути ідентифіковані на основі статичних шаблонів. Обидва методи мають свої сильні сторони, і їх інтеграція в одну систему дозволить досягти максимальної ефективності у виявленні та запобіганні кіберзагрозам.

3.5 Висновки до розділу 3

В третьому розділі було проведено детальний аналіз сигнатурного та поведінкового підходів в IDS з практичним тестуванням у лабораторному середовищі. Сигнатурний підхід продемонстрував високу ефективність у виявленні відомих загроз завдяки чітко визначеним шаблонам. Під час тестування сигнатурне правило успішно виявило трафік, що містив визначений патерн, довівши свою корисність у сценаріях, де загрози добре задокументовані. Проте тестування також виявило обмеження цього підходу: зміна шаблону атаки призвела до того, що система не змогла ідентифікувати нову загрозу. Це підкреслює залежність сигнатурного методу від своєчасного оновлення бази сигнатур і його вразливість до атак нульового дня.

Поведінковий підхід, у свою чергу, показав високу гнучкість і здатність виявляти нові та невідомі загрози на основі аналізу аномалій у мережевій активності. Хоча під час першого етапу тестування поведінкове правило не зафіксувало атаку через відсутність значних відхилень від базової лінії, на другому етапі, із збільшенням обсягу трафіку, система успішно виявила аномальну активність. Це демонструє перевагу поведінкового підходу у виявленні атак, які не мають чітко визначених сигнатур. Проте варто зазначити, що цей метод потребує ретельного налаштування параметрів, щоб уникнути хибних спрацювань.

У результаті було зроблено висновок, що ефективність систем IDS значно підвищується при поєднанні сигнатурного та поведінкового підходів. Така інтеграція дозволяє компенсувати недоліки кожного методу: сигнатурний підхід забезпечує точність у виявленні відомих загроз, а поведінковий дозволяє реагувати на нові типи атак. Практичні результати тестування підтверджують

доцільність використання обох підходів у сучасних системах кіберзахисту для забезпечення надійного виявлення та протидії кіберзагрозам.

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Охорона праці

Метою кваліфікаційної роботи магістра є порівняльний аналіз сигнатурного та поведінкового підходів в системах IDS з метою оцінки їх ефективності у виявленні мережевих атак. Оскільки, такі роботи передбачають використання комп'ютерної техніки, зокрема ПК та серверного обладнання, то обов'язком виконавця такого процесу є дотримання вимог охорони праці і техніки безпеки.

Роботодавець зобов'язаний згідно Закону України «Про охорону праці» стаття 13 «Управління охороною праці та обов'язки роботодавця» створити на робочому місці в кожному структурному підрозділі умови праці відповідно до нормативно-правових актів, а також забезпечити додержання вимог законодавства щодо прав працівників у галузі охорони праці [26,27,28].

Із цією метою роботодавець забезпечує функціонування системи управління охороною праці, а саме:

- створює відповідні служби і призначає посадових осіб, які забезпечують вирішення конкретних питань охорони праці, затверджує інструкції про їхні обов'язки, права та відповідальність за виконання покладених на них функцій, а також контролює їх додержання;

- розробляє за участю сторін колективного договору і реалізує комплексні заходи для досягнення встановлених нормативів та підвищення існуючого рівня охорони праці;

- забезпечує виконання необхідних профілактичних заходів відповідно до обставин, що змінюються;

- впроваджує прогресивні технології, досягнення науки і техніки, засоби механізації та автоматизації виробництва, вимоги ергономіки, позитивний досвід з охорони праці тощо;

- забезпечує належне утримання будівель та споруд, виробничого обладнання та устаткування, моніторинг за їх технічним станом;

- забезпечує усунення причин, що призводять до нещасних випадків, професійних захворювань, та здійснення профілактичних заходів, визначених комісіями за підсумками розслідування цих причин;

- організовує проведення аудиту охорони праці, лабораторних досліджень умов праці, оцінку технічного стану виробничого обладнання та устаткування, атестацій робочих місць на відповідність нормативно-правовим актам з охорони праці в порядку і строки, що визначаються законодавством, та за їх підсумками вживає заходів з усунення небезпечних і шкідливих для здоров'я виробничих факторів;

- розробляє і затверджує положення, інструкції, інші акти з охорони праці, що діють у межах підприємства та встановлюють правила виконання робіт і поведінки працівників на території підприємства, у виробничих приміщеннях, на будівельних майданчиках, робочих місцях відповідно до нормативно-правових актів з охорони праці, забезпечує безоплатно працівників нормативно-правовими актами підприємства з охорони праці;

- здійснює контроль за дотриманням працівником технологічних процесів, правил поведінки з машинами, механізмами, устаткуванням та іншими засобами виробництва, використанням засобів колективного та індивідуального захисту, виконанням робіт відповідно до вимог з охорони праці;

- організовує пропаганду безпечних методів праці та співробітництво з працівниками у галузі охорони праці.

Роботодавець несе безпосередню відповідальність за порушення нормативно-правових актів з охорони праці.

Для забезпечення оптимальних умов праці працівників при розробці, встановлення та налаштуванні систем високої доступності на основі кластерів необхідно передбачити відповідність мікроклімату у приміщеннях згідно вимог ДСН 3.3.6.042-99. Категорія робіт при виконанні даного виду завдань належить до легкої – Іб. Для того щоб визначити, чи відповідає повітря приміщення встановленим нормам, необхідно кількісно оцінити кожний з його параметрів. Оптимальні показники мікроклімату, які необхідно забезпечити у приміщеннях,

де експлуатуються ПК у теплу пору року повинні становити: температура – +22 оС - +24 оС, відносна вологість – 40-60%, швидкість руху повітря 0,1 м/с.

Окрім, забезпечення оптимальних показників мікроклімату, необхідно передбачити ще й оптимальні показники шуму та вібрації на робочих місцях.

Граничні величини шуму на робочих місцях регламентуються ДСН 3.3.6.037 – 99 „Державні санітарні норми виробничого шуму, ультразвуку та інфразвуку”. В ньому закладено принцип встановлення певних параметрів шуму, виходячи з класифікації приміщень та їх використання для трудової діяльності. Окрім цього, на робочих місцях працівників необхідно забезпечити дотримання вимог НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями».

Основними вимогами, визначеними у цьому нормативному документі є:

- площу та об'єм для одного робочого місця оператора визначають згідно з вимогами ДСанПіН 3.3.2-007-98. Площа має бути не менше 6,0 кв.м, об'єм – не менше 20,0 куб.м.;

- заземлені конструкції, що знаходяться в приміщеннях, де розміщені робочі місця операторів (батареї опалення, водопровідні труби, кабелі із заземленим відкритим екраном), мають бути надійно захищені діелектричними щитками або сітками з метою недопущення потрапляння працівника під напругу;

- приміщення, де розміщені робочі місця операторів, крім приміщень, у яких розміщені робочі місця операторів великих ЕОМ загального призначення (серверні), повинні бути оснащені системою автоматичної пожежної сигналізації.

Дотримання даних вимог є необхідними при створенні системи високої доступності на основі кластерів з використанням серверного обладнання та ПК.

4.2 Підвищення стійкості роботи об’єктів господарської діяльності у воєнний час

Для покращення стійкості роботи об’єктів вивчають фактори, які впливають на стійкість та оцінюють стійкість елементів і галузей виробництва проти

уражаючих факторів ядерної, хімічної і біологічної зброї, стихійних лих і виробничих аварій. Щоб підвищити стійкість необхідно завчасно організувати і провести організаційні, інженерно-технічні й технологічні заходи [29].

Здійснення організаційних заходів передбачає завчасну підготовку всіх структур цивільного захисту, служб і формувань до надзвичайних ситуацій, в тому числі і військових дій. Вжиттям технологічних заходів підвищується стійкість роботи об'єктів шляхом змінювання технологічних процесів, режимів, можливих в умовах різних надзвичайних ситуацій. Інженерно-технічні заходи мають забезпечити підвищену стійкість виробничих споруд, технологічних ліній, устаткування, комунікацій об'єкта до впливу уражаючих факторів під час військових дій. При проведенні цих заходів необхідно враховувати конкретні умови об'єкта народного господарства. Проте є загальні організаційні інженерно-технічні заходи, які мають проводитись на всіх об'єктах.

Одним з найбільш важливих завдань в умовах воєнного часу і надзвичайних ситуацій є забезпечення захисту людей та їх життєдіяльності.

Для підвищення стійкості об’єктів господарювання та захисту людей необхідно:

– створити на об'єкті надійну систему оповіщення про загрози нападу противника, радіоактивне забруднення, хімічне і біологічне зараження, загрозу стихійного лиха і виробничої аварії;

– організувати розвідку і спостереження за радіоактивним забрудненням, хімічним і біологічним зараженням;

– організувати гідрометеорологічне спостереження за рівнем води, напрямком і швидкістю вітру, рухом і поширенням хмари радіоактивного забруднення, сильнодіючих отруйних речовин і отруйних речовин;

– створити фонд захисних споруд ЦО, запасів засобів індивідуального захисту і забезпечення своєчасної видачі їх населенню;

– завчасно підготуватись до масової санітарної обробки населення і знезаражування одягу;

– організувати взаємодію з установами охорони здоров'я для медичного обслуговування населення в умовах воєнного часу.

Також в умовах воєнного часу необхідно провести підготовку до евакуації населення, розміщеного в зонах можливих руйнувань і катастрофічного затоплення. Це передбачає завчасну підготовку місць евакуації, організацію прийому евакуйованого населення на територію населених пунктів. Окрім цього, необхідно забезпечити постачання продуктів харчування, питної води, предметів першої необхідності та провести заходи щодо морально-психологічної підготовки населення до виживання в умовах воєнного часу, забезпечити процес чіткого інформування про обстановку та правила дій і поведінки населення в надзвичайних ситуаціях воєнного часу [29].

Для забезпечення стійкості роботи об'єктів повинні проводитись інженерно-технічні заходи на мережах комунального господарства з метою захисту джерел тепла із заглибленням у ґрунт комунікацій. Котельні слід розміщувати в спеціальному окремо розміщеному приміщенні.

Якщо об'єкт одержує тепло з міської теплоцентралі, необхідно провести заходи для забезпечення стійкості трубопроводів і розподільних пристроїв, підведених до об'єкта. Теплова мережа має будуватися за кільцевою системою з прокладанням труб у спеціальних каналах зі з'єднанням паралельних ділянок. Для відключення пошкоджених ділянок мають бути встановлені запірно-регулюючі засувки, вентилі та ін. Ці пристосування необхідно розміщувати в оглядових колодязях, на території, що не завалюється при руйнуванні будівель.

Система каналізації має будуватись окремо: одна для дощових, друга для промислових і господарських вод. На об'єкті має бути не менше двох виводів з підключенням до міських каналізаційних колекторів, а також виводи і колодязі з аварійними засувками на об'єктових колекторах з інтервалом 50 м на території, що не завалюється, для аварійного скидання неочищеної води в найближчі штучні та природні заглиблення.

На деяких промислових об'єктах є системи для забезпечення технології виробництва: для подання кисню, аміаку, стиснутого повітря та інших рідких і газових реактивів. Для цих систем розробляють заходи для попередження виникнення вторинних факторів зброї, стихійних лих та виробничих аварій і катастроф.

Створення резерву енергетичних потужностей за рахунок автономних пересувних електростанцій, а також місцевих джерел електроенергії. Підготовка автономних електростанцій до роботи за спеціальним режимом (графіком) для забезпечення технологічних процесів виробництва, для яких неможливі тривалі перерви в електропостачанні. З метою попередження аварій на електричних мережах необхідно установити автоматичну систему відключення при виникненні перенапруги. Повітряні лінії електропостачання замінити на підземно-кабельні. Створення необхідних запасів (резервів) паливно-мастильних матеріалів та інших видів палива й організація їх безпечного зберігання.

Щоб не допустити зупинки підприємства через дефіцит палива, необхідно підготуватись для роботи на різних видах палива: нафта, вугілля, газ. Для підвищення стійкості забезпечення водою слід провести такі заходи.

Необхідно створити основні і резервні джерела водопостачання. Як резервне джерело краще мати артезіанську свердловину, яку необхідно підключити до системи водопостачання. Крім того, воду можна брати з близько розміщеної природної водойми або спорудити штучну водойму чи резервуари з обладнанням пристроїв для збору і перекачування води. Всі ділянки водопостачання повинні бути заглиблені в ґрунт з обладнанням пожежних гідрантів і пристроїв для відключення пошкоджених ділянок. Локальні мережі водопостачання окремих великих підприємств варто з'єднати із 80 загальноміською системою водопостачання в єдине кільце.

Підвищенню стійкості забезпечення водою сприяє подавання води безпосередньо в мережу поза водонапірними баштами, спорудження обвідних ліній для подання води поза пошкодженими спорудами.

Завчасне вжиття заходів захисту джерел водопостачання, водопровідних споруд, свердловин і шахтних колодязів від забруднення радіоактивними речовинами, зараження хімічними і біологічними засобами. Підготовка меліоративних, гідротехнічних та іригаційних споруд і систем до експлуатації в надзвичайних умовах.

Для забезпечення виробництва продукції необхідні електроенергія, паливо, мастила, засоби захисту рослин, міндобрива, профілактичні й лікувальні препарати ветеринарної медицини, запасні частини, сировина та інші матеріально-технічні засоби. Забезпечення об'єктів цими ресурсами дасть можливість випускати необхідну продукцію в надзвичайних умовах мирного і воєнного часу. Тому повинні проводитись такі заходи, які б забезпечили стійкість постачання і сприяли підвищенню захисту мережі електро-, водо-, газопостачання, транспортних комунікацій і джерел постачання всім необхідним для забезпечення функціонування галузей сільського господарства в надзвичайних умовах.

З метою попередження аварій на електричних мережах необхідно встановити автоматичну систему відключення перенапруги. Повітряні лінії електропостачання слід замінити на підземно-кабельні. Газ використовується як паливо і на хімічних підприємствах у технологічному процесі. Для безперебійного забезпечення газом, газові мережі необхідно підводити до об'єкта з двох напрямків, які мають бути з'єднані в єдине кільце з обладнанням для можливого дистанційного автоматичного управління й у разі необхідності відключення пошкоджених ділянок. На великих підприємствах необхідно мати підземні ємності із закачаним резервним газом.

На підприємствах, де використовується пара, необхідно захистити джерела його постачання, заглибити в ґрунт комунікації паропостачання і встановити запірні пристосування.

Запас резервних матеріалів необхідно розраховувати на такі строки роботи підприємства, за які можливе відновлення регулярного постачання.

Передбачити, на випадок перебоїв в постачанні підприємствами-суміжниками, створення місцевих матеріалів, сировини для виготовлення комплектуючих виробів і інструментів силами свого підприємства [29].

Для підвищення стійкості та забезпечення збереження (відновлення) будівель і споруд в умовах воєнного часу необхідно:

– провести оцінку можливих ступенів руйнування будівель і споруд господарства населеного пункту, визначити обсяг невідкладних ремонтних робіт, потреби в будівельних матеріалах;

– створити і підготувати спеціальні формування для ремонтно-відновних, будівельних та інших робіт на об'єкті;

– розробити комплекс протипожежних заходів, які виключали б можливість виникнення масових пожеж.

Для забезпечення надійності системи управління і зв'язку потрібно організувати захищений пункт управління, забезпечити його засобами зв'язку, які б дали можливість швидко доводити сигнали ЦЗ до всіх виробничих підрозділів і населення у місцях проживання. При цьому необхідно здійснити планування збору даних про обстановку, передачу команд і розпоряджень в умовах впливу на об'єкт уражуючих факторів. Для підвищення стійкості системи управління і зв’язку в умовах воєнного часу необхідно організувати використання радіозасобів, засобів телефонного зв'язку а також забезпечити зв’язок із колонами евакуйованого населення, що перебувають у дорозі, і відповідальними особами, які супроводжують їх під час евакуації, забезпечити дублювання ліній і каналів зв'язку.

ВИСНОВКИ

Під час виконання кваліфікаційної роботи магістра було проведено всебічне дослідження систем IDS з акцентом на сигнатурний та поведінковий підходи виявлення вторгнень. Метою роботи було оцінити ефективність цих підходів у виявленні різних типів мережевих атак та розробити рекомендації щодо їх інтеграції в сучасні системи кібербезпеки.

У першому розділі здійснено детальний аналіз класифікації мережевих атак, їх наслідків для інформаційних систем та сучасних методів захисту. Розглянуто атаки на конфіденційність, цілісність та доступність, визначено їхні основні характеристики та методи реалізації. Також проаналізовано наслідки таких атак, включаючи фінансові втрати, репутаційні ризики та юридичні наслідки для організацій. Було підкреслено важливість впровадження сучасних засобів захисту, таких як брандмауери та системи IDS/IPS, для мінімізації ризиків та забезпечення стійкості інформаційних систем.

У другому розділі описано процес розгортання тестового лабораторного середовища, яке базується на гіпервізорі Hyper-V та операційній системі Windows Server 2022 Core. Це середовище дозволило моделювати реальні мережеві сценарії та оцінювати ефективність сигнатурного та поведінкового підходів IDS у контрольованих умовах. Детально розглянуто компоненти середовища, зокрема маршрутизатор pfSense з інтегрованою системою Suricata, сервер Ubuntu Linux з налаштованими сервісами та Kali Linux для моделювання атак. Проведено тестування працездатності середовища, яке підтвердило його готовність до проведення експериментів та досліджень.

Третій розділ був присвячений практичному дослідженню сигнатурного та поведінкового підходів в IDS. За допомогою спеціально розроблених python-скриптів було змодельовано різні типи атак та проведено тестування систем IDS. Результати показали, що сигнатурний підхід є ефективним для виявлення відомих атак з чітко визначеними сигнатурами, але виявляє обмеження при зустрічі з новими або модифікованими загрозами. Поведінковий підхід

продемонстрував здатність виявляти аномальну активність у мережі, включаючи нові типи атак, завдяки аналізу відхилень від базової лінії нормальної поведінки.

Проведене тестування підтвердило, що комбінування сигнатурного та поведінкового підходів у системах IDS забезпечує більш високий рівень захисту інформаційних систем. Такий інтегрований підхід дозволяє ефективно виявляти як відомі загрози, так і нові, що тільки з'являються, мінімізуючи ризики для організацій.

Таким чином, у ході роботи було досягнуто поставленої мети та підтверджено доцільність використання комбінованих методів виявлення загроз у сучасних системах кібербезпеки. Отримані результати можуть бути використані для подальшого вдосконалення систем IDS та розробки рекомендацій щодо їх впровадження в різних організаціях. Робота підкреслює важливість постійного оновлення та адаптації засобів захисту в умовах швидкого розвитку кіберзагроз, а також необхідність комплексного підходу до забезпечення безпеки інформаційних систем.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Types of network attacks against confidentiality, integrity and avilability. Free Networking tutorials, System Administration Tutorials and Security Tutorials - omnisecu.com. URL: https://www.omnisecu.com/ccna-security/types-of-network-attacks.php (date of access: 12.11.2024).
2. ZAGORODNA, N., STADNYK, M., LYPA, B., GAVRYLOV, M., & KOZAK, R. (2022). Network Attack Detection Using Machine Learning Methods. Challenges to national defence in contemporary geopolitical situation, 2022(1), 55-61.
3. What is a DDoS botnet? Cloudflare. URL: https://www.cloudflare.com/learning/ddos/what-is-a-ddos-botnet/ (date of access: 04.11.2024).
4. SYN flood attack. Cloudflare. URL: https://www.cloudflare.com/learning/ddos/syn-flood-ddos-attack/ (date of access: 04.11.2024).
5. UDP flood attack. Cloudflare. URL: https://www.cloudflare.com/learning/ddos/udp-flood-ddos-attack/ (date of access: 04.11.2024).
6. HTTP flood attack. Cloudflare. URL: https://www.cloudflare.com/learning/ddos/http-flood-ddos-attack/ (date of access: 04.11.2024).
7. DNS amplification attack. Cloudflare. URL: https://www.cloudflare.com/learning/ddos/dns-amplification-ddos-attack/ (date of access: 04.11.2024).
8. What is a firewall? How network firewalls work. Cloudflare. URL: https://www.cloudflare.com/learning/security/what-is-a-firewall/ (date of access: 04.11.2024).
9. Types of Firewalls Defined and Explained URL: https://www.paloaltonetworks.com/cyberpedia/types-of-firewalls (date of access: 04.11.2024).

10. Tymoshchuk, V., Mykhailovskyi, O., Dolinskyi, A., Orlovska, A., & Tymoshchuk, D. (2024). OPTIMISING IPS RULES FOR EFFECTIVE DETECTION OF MULTI-VECTOR DDOS ATTACKS. Матеріали конференцій МЦНД, (22.11. 2024; Біла Церква, Україна), 295-300.
11. What is an intrusion prevention system (IPS)? | fortinet. Fortinet. URL: https://www.fortinet.com/resources/cyberglossary/what-is-an-ips (date of access: 04.11.2024).
12. Suricata User Guide – Suricata 8.0.0-dev documentation. Suricata User Guide – Suricata 8.0.0-dev documentation. URL: https://docs.suricata.io/en/latest/ (date of access: 12.11.2024).
13. Zeek documentation – book of zeek (v7.0.4). Zeek Documentation – Book of Zeek (git/master). URL: https://docs.zeek.org/en/current/ (date of access: 20.11.2024).
14. Lypa, B., Horyn, I., Zagorodna, N., Tymoshchuk, D., Lechachenko T., (2024). Comparison of feature extraction tools for network traffic data. CEUR Workshop Proceedings, 3896, pp. 1-11.
15. Advanced threat prevention. Palo Alto Networks. URL: https://www.paloaltonetworks.ca/network-security/advanced-threat-prevention (date of access: 20.11.2024).
16. Intrusion prevention | administration guide. Fortinet Document Library | Home. URL: https://docs.fortinet.com/document/fortigate/7.6.0/administration-guide/565562/intrusion-prevention (date of access: 20.11.2024).
17. Tymoshchuk, D., Yasniy, O., Mytnyk, M., Zagorodna, N., Tymoshchuk, V., (2024). Detection and classification of DDoS flooding attacks by machine learning methods. CEUR Workshop Proceedings, 3842, pp. 184 - 195.
18. Windows Server documentation. Microsoft Learn: Build skills that open doors in your career. URL: https://learn.microsoft.com/en-us/windows-server/ (date of access: 20.11.2024).

19. ТИМОЩУК, Д., ЯЦКІВ, В., ТИМОЩУК, В., & ЯЦКІВ, Н. (2024). INTERACTIVE CYBERSECURITY TRAINING SYSTEM BASED ON SIMULATION ENVIRONMENTS. MEASURING AND COMPUTING DEVICES IN TECHNOLOGICAL PROCESSES, (4), 215-220.
20. PfSense documentation | pfsense documentation. Netgate Documentation | Netgate Documentation. URL: https://docs.netgate.com/pfsense/en/latest/ (date of access: 04.11.2024).
21. ТИМОЩУК, Д., & ЯЦКІВ, В. (2024). USING HYPERVISORS TO CREATE A CYBER POLYGON. MEASURING AND COMPUTING DEVICES IN TECHNOLOGICAL PROCESSES, (3), 52-56.
22. Kali docs | kali linux documentation. Kali Linux. URL: https://www.kali.org/docs/ (date of access: 12.11.2024).
23. Tymoshchuk, V., Vantsa, V., Karnaukhov, A., Orlovska, A., & Tymoshchuk, D. (2024). COMPARATIVE ANALYSIS OF INTRUSION DETECTION APPROACHES BASED ON SIGNATURES AND ANOMALIES. Матеріали конференцій МЦНД, (29.11. 2024; Житомир, Україна), 328-332.
24. Tymoshchuk, V., Vorona, M., Dolinskyi, A., Shymanska, V., & Tymoshchuk, D. (2024). SECURITY ONION PLATFORM AS A TOOL FOR DETECTING AND ANALYSING CYBER THREATS. Collection of scientific papers «ΛΌГОΣ», (December 13, 2024; Zurich, Switzerland), 232-237.
25. The python tutorial. Python documentation. URL: https://docs.python.org/3/tutorial/index.html (date of access: 20.11.2024).
26. Закон України Про охорону праці Відомості Верховної Ради України (ВВР), № 49, 1992. 668 с.
27. Наказ міністерство соціальної політики України № 207 Про затвердження вимог щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями від 14.02.2018.
28. ДСАНПІН 3.3.2.007-98: Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин №7 від 10.12.98.

29. Стручок В.С. Техноекологія та цивільна безпека. Частина «Цивільна безпека». Навчальний посібник. Тернопіль: ТНТУ. 2022. 150 с.

Додаток А Публікація

ЗБІРНИК НАУКОВИХ ПРАЦЬ

З МАТЕРІАЛАМИ VIII МІЖНАРОДНОЇ НАУКОВОЇ КОНФЕРЕНЦІЇ

29 ЛИСТОПАДА 2024 РІК

М. ЖИТОМИР, УКРАЇНА

«НАУКОВИЙ ПРОСТІР: АКТУАЛЬНІ ПИТАННЯ, ДОСЯГНЕННЯ ТА ІННОВАЦІЇ»

ЗБІРНИК НАУКОВИХ
ПРАЦЬ З МАТЕРІАЛАМИ
VIII МІЖНАРОДНОЇ
НАУКОВОЇ КОНФЕРЕНЦІЇ

НАУКОВИЙ ПРОСТІР: АКТУАЛЬНІ ПИТАННЯ, ДОСЯГНЕННЯ ТА ІННОВАЦІЇ

29 листопада 2024 рік
м. Житомир, Україна

Вінниця, Україна
«UKRLOGOS Group»
2024

СЕКЦІЯ XXI.
ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ТА СИСТЕМИ

DOI 10.62731/mcnd-29.11.2024.004

COMPARATIVE ANALYSIS OF INTRUSION DETECTION APPROACHES BASED ON SIGNATURES AND ANOMALIES

Tymoshchuk Vitaliy
ORCID ID: 0009-0007-2858-9434
First-level student of higher education
Faculty of Applied Information Technologies and Electrical Engineering
Ternopil Ivan Puluj National Technical University, Ukraine

Vantsa Vitalii
Second-level student of higher education
Faculty of Computer Information Systems and Software Engineering
Ternopil Ivan Puluj National Technical University, Ukraine

Karnaukhov Andrii
ORCID ID: 0000-0003-3042-3066
Third-level student of higher education
Faculty of Computer Information Systems and Software Engineering
Ternopil Ivan Puluj National Technical University, Ukraine

Orlovska Anastasiia
ORCID ID: 0009-0001-0286-0691
First-level student of higher education
Faculty of Computer Information Systems and Software Engineering
Ternopil Ivan Puluj National Technical University, Ukraine

Scientific supervisor: Dmytro Tymoshchuk
ORCID ID: 0000-0003-0246-2236
Senior lecturer at the Department of Cyber Security
Ternopil Ivan Puluj National Technical University, Ukraine

Abstract. *The article provides a comparative analysis of two main approaches to intrusion detection in IDS: signature and anomaly-based. The principles of operation of each approach, their advantages and disadvantages, as well as efficiency in detecting various types of cyber*

threats are considered. An experimental study in the laboratory environment is carried out, the results of which confirm the expediency of combined protection using both methods to ensure a high level of security.

Keywords: *IDS, signature, anomaly, Suricata, pfSense, attacks, cybersecurity.*

Introduction. Intrusion detection systems (IDSs) are an integral part of modern cybersecurity, providing real-time monitoring of network traffic to detect threats [1]. The variety of constantly evolving threats creates a need for effective detection methods, among which the most common are signature and behavioural (anomaly-based) approaches. Each of these methods has its own advantages, disadvantages and areas of application. The purpose of this article is to study the effectiveness of signature and behavioural approaches in intrusion detection, and to assess their ability to respond to different types of attacks in modern IDS systems. To achieve this goal, both approaches were experimentally tested in a laboratory environment with the simulation of real cyber threats.

Research methodology. To conduct this study and achieve its goal, we created a controlled test environment that reflected typical network conditions [2,3]. The infrastructure consisted of several nodes: an Ubuntu Linux server [4] with HTTP, DNS, and SSH services, Suricata [5] based IDS systems in pfSense [6] that used signature and behavioural approaches to detect intrusions [7], and Kali Linux for attack simulation [8]. The environment was deployed on the Hyper-V hypervisor [9] running on Windows Server 2022 Core (Fig. 1).

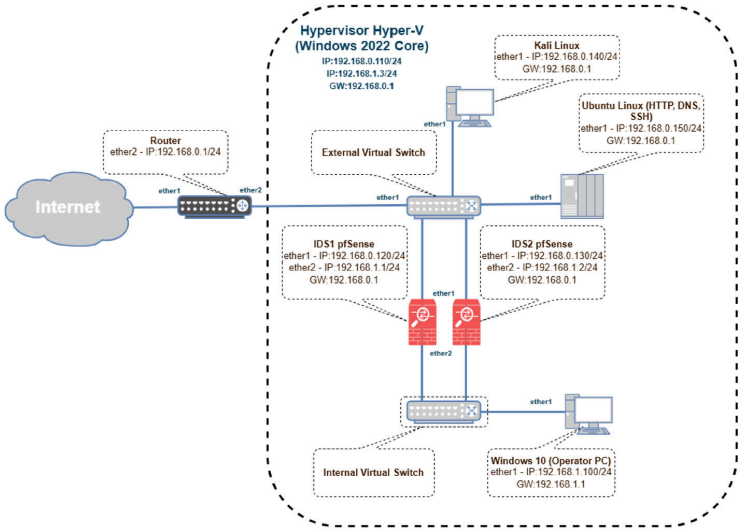

Fig. 1. **Structure of the laboratory environment**

To model the threats, a specially developed python script was used to generate network traffic with various parameters, including the use of certain patterns and abnormally high traffic volumes. The signature approach was based on rules that identified a predefined pattern in network packets. The behavioural approach used an analysis of traffic anomalies, including calculating the frequency of requests and deviations from the baseline of normal activity. The testing was conducted in two stages, each aimed at assessing the response of IDS systems to different types of threats. The first stage involved generating traffic with a standard pattern, while the second stage changed the pattern and significantly increased the traffic volume.

The signature approach is one of the most common methods of threat detection in IDS systems. It is based on comparing incoming traffic with a database of signatures, which are patterns of known attacks. Signatures can include a specific byte sequence, specific protocol parameters, or anomalies in packet headers. When the IDS finds a match with a signature, it generates a threat alert.

During the first stage of testing, the signature rule successfully detected traffic with the defined pattern (Fig. 2).

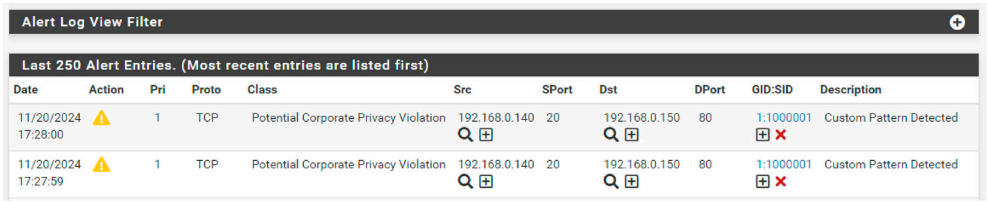

Fig. 2. **Alert logs in IDS1 at the first stage of testing**

This confirms the high accuracy of the method in identifying known threats. However, the second stage, where the pattern was changed and the traffic volume increased, showed the limitations of the approach. The system was unable to detect the attack due to the lack of a suitable signature. This highlights the vulnerability of the signature approach to new and modified threats and the need to regularly update the signature database to ensure that protection remains up-to-date. Nevertheless, the signature approach has a low false positive rate because it identifies threats only by clearly defined patterns.

The behavioural approach, also known as anomaly analysis, is based on creating a baseline of normal network activity and identifying deviations from this line. This method allows us to identify new and unknown threats that

cannot be detected by the signature approach. Behavioural analysis takes into account traffic volumes, time patterns, specific protocols and ports. At the first stage of testing, the behavioural approach did not detect any threats, as the traffic did not exceed the acceptable baseline. However, in the second stage, with a significant increase in traffic, the IDS configured for the behavioural approach successfully detected the attack (Fig. 3).

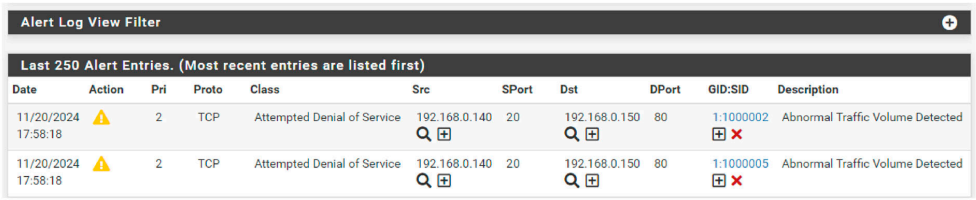

Fig. 3. **Alert logs in IDS2 at the second stage of testing**

This confirms the method's ability to detect threats characterised by deviations from normal activity, even in the absence of a specific pattern.

The main challenge of the behavioural approach is the high probability of false positives, as any deviation from the baseline can be interpreted as a threat. To reduce the false alarm rate, careful tuning of the system parameters is required.

Research results. Testing has confirmed that the signature approach is effective in detecting threats that have known patterns. Its accuracy and low false positive rate make it an ideal choice for protecting against well-documented attacks. However, this approach has proven ineffective against new or modified threats. The behavioural approach has shown high flexibility and the ability to adapt to new conditions, detecting anomalous scenarios that cannot be predicted by the signature method. However, its effective application requires setting up a baseline, which takes time and resources.

The optimal solution is a combined approach that integrates the advantages of both methods. The integration of these two approaches in modern IDS systems is a promising direction. Combined systems can provide both accurate detection of known threats using the signature method and adaptability to new threats through behavioural analysis. Such hybrid approaches minimise the disadvantages of each method separately and provide a high level of protection.

Conclusions. A comparative analysis of the signature and behavioural approaches to intrusion detection shows that each of them has its strengths

and weaknesses. The signature-based approach provides accuracy in detecting known threats, while the anomaly-based approach allows detecting new and unknown attacks. The results of the experimental study confirm the expediency of the combined use of both approaches to achieve maximum efficiency of IDS systems. The integration of methods allows compensating for the shortcomings of each of them and providing more reliable protection of information systems. Further research will be aimed at implementing machine learning algorithms to improve the anomaly-based approach and reduce the false alarm rate.

References:

1. What is intrusion detection systems (IDS)? How does it work? | fortinet. (n.d.). Fortinet. https://www.fortinet.com/resources/cyberglossary/intrusion-detection-system
2. Tymoshchuk, D., & Yatskiv, V. (2024). Using hypervisors to create a cyber polygon. Measuring and Computing Devices in Technological Processes, (3), 52–56. https://doi.org/10.31891/2219-9365-2024-79-7
3. Тимощук, В., & Тимощук, Д. (2022). Віртуалізація в центрах обробки даних-аспекти відмовостійкості. Матеріали X науково-технічної конференції „Інформаційні моделі, системи та технології “Тернопільського національного технічного університету імені Івана Пулюя, 95-95.
4. Ubuntu desktop guide. (n.d.). Official Ubuntu Documentation. https://help.ubuntu.com/lts/ubuntu-help/index.html
5. Suricata User Guide — Suricata 8.0.0-dev documentation. (n.d.). Suricata User Guide — Suricata 8.0.0-dev documentation. https://docs.suricata.io/en/latest/
6. PfSense documentation | pfsense documentation. (n.d.). Netgate Documentation | Netgate Documentation. https://docs.netgate.com/pfsense/en/latest/
7. What are the detection methods of IDS? (n.d.). Network Threat Detection and Response | Stamus Networks. https://www.stamus-networks.com/blog/what-are-the-detection-methods-of-ids
8. Kali docs | kali linux documentation. (n.d.). Kali Linux. https://www.kali.org/docs/
9. Hyper-V technology overview. (n.d.). Microsoft Learn: Build skills that open doors in your career. https://learn.microsoft.com/en-us/windows-server/virtualization/hyper-v/hyper-v-overview

Додаток Б Скрипт content_custom3.py

```
import sys
import time
import os
from scapy.all import *

# Function to check for root privileges
def check_root():
    if not os.geteuid() == 0:
        sys.exit("This script must be run with administrator (root) privileges.")

# Function for logging
def log_message(message):
    with open("packet_log.txt", "a") as log_file:
        log_file.write(message + "\n")

# Default parameters
default_dst_ip = "192.168.0.150"
default_port = 80
default_count = 10
default_delay = 0.01  # Delay between packets in seconds

# Parse command-line arguments
def parse_arguments():
    dst_ip = default_dst_ip
    port = default_port
    count = default_count
    delay = default_delay

    if len(sys.argv) > 1:
        dst_ip = sys.argv[1]  # Destination IP address
    if len(sys.argv) > 2:
        port = int(sys.argv[2])  # Destination port
    if len(sys.argv) > 3:
        count = int(sys.argv[3])  # Number of packets
    if len(sys.argv) > 4:
        delay = float(sys.argv[4])  # Delay between packets

    return dst_ip, port, count, delay

# Custom pattern string
custom_string = "ABCD1234XYZ"
```

```python
# Function to create and send packets
def create_and_send_packet(dst_ip, port, count, delay):
    for i in range(count):
        pkt = IP(dst=dst_ip) / TCP(dport=port, flags="S") / Raw(load=custom_string)

        # Send packet without verbose output
        send(pkt, verbose=False)

        # Log the packet (commented out)
        log_message(f"Sent packet {i+1} to {dst_ip}:{port}")

        # Delay between packet sends
        time.sleep(delay)

        # Print status every 100 packets
        if (i + 1) % 100 == 0:  # Every 100 packets
            print(f"Sent {i + 1} packets.")

# Check for root privileges
check_root()

# Get parameters
dst_ip, port, count, delay = parse_arguments()

# Send packets
create_and_send_packet(dst_ip, port, count, delay)

print(f"Sent {count} packets to {dst_ip}:{port}. Done.")

#sudo python3 content_custom3.py <IP> <port> <count> <delay>
#sudo python3 content_custom3.py 192.168.0.150 80 10 0.01
```