МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**

**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЗАХИСТУ ІНФОРМАЦІЇ
КАФЕДРА УПРАВЛІННЯ ІНФОРМАЦІЙНОЮ ТА КІБЕРНЕТИЧНОЮ БЕЗПЕКОЮ**

КВАЛІФІКАЦІЙНА РОБОТА

на тему: “РОЗРОБКА ТА РЕАЛІЗАЦІЯ СИСТЕМИ ВИЯВЛЕННЯ ТА РЕАГУВАННЯ НА КІБЕРАТАКИ У ВЕЛИКИХ КОРПОРАТИВНИХ МЕРЕЖАХ ”

на здобуття освітнього ступеня бакалавра
зі спеціальності 125 Кібербезпека та захист інформації
освітньої програми Управління інформаційною та кібернетичною безпекою

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

_____ Євгеній МИЩЕНКО
(підпис) *Ім’я, ПРІЗВИЩЕ здобувача*

Виконав: здобувач вищої освіти гр. УБД-42

Євгеній МИЩЕНКО
Ім’я, ПРІЗВИЩЕ

Керівник:
к.т.н., доцент Юрій ЩАВІНСЬКИИ
Ім’я, ПРІЗВИЩЕ

Рецензент:

_____
Ім’я, ПРІЗВИЩЕ

Київ 2024

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут захисту інформації

Кафедра управління інформаційною та кібернетичною безпекою

Ступінь вищої освіти бакалавр

Спеціальність 125 Кібербезпека та захист інформації

Освітня програма Управління інформаційною та кібернетичною безпекою

ЗАТВЕРДЖУЮ
Завідувач кафедри УІКБ
_____ Світлана ЛЕГОМІНОВА
“____”_____2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Мищенку Євгенію Ігоровичу
(прізвище, ім’я, по батькові здобувача)

1. Тема кваліфікаційної роботи “Розробка та реалізація системи виявлення та реагування на кібератаки у великих корпоративних мережах”,
керівник кваліфікаційної роботи ЩАВІНСЬКИЙ Юрій, к.т.н., доцент,
(ІПРІЗВИЩЕ, Ім’я,, науковий ступінь, вчене звання)

затверджені наказом Державного університету інформаційно-комунікаційних технологійвід “27“ *лютого 2024* р. № *36*.

2. Строк подання кваліфікаційної роботи “20” травня 2024р.

3. Вихідні дані до кваліфікаційної роботи: *наукова та технічна література. методи розробки систем, міжнародні стандарти кібербезпеки, платформа VirtualBox, стандартні бібліотеки Python для розробки моделей.*

4. Перелік питань, які мають бути розроблені:
   4.1. Аналіз сучасного стану кібербезпеки великих корпоративних мереж та систем виявлення і реагування на кібератаки.
   4.2. Аналіз вимог до систем виявлення та реагування на кібератаки.
   4.3. Проектування системи виявлення та реагування на кібератаки великих корпоративних мереж.
   4.4. Оцінка ефективності розробленої системи виявлення і реагування на кібератаки.

5. Перелік ілюстративного матеріалу: *презентація PowerPoint*

6. Дата видачі завдання “11“ березня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Етапи кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1.	Визначення об’єкту, предмету, мети та завдань дослідження.	18.03.2024	
2.	Збір та аналіз літератури.	29.03.2024	
3.	Аналіз сучасного стану кібербезпеки великих корпоративних мереж та систем виявлення і реагування на кібератаки.	08.04.2024	
4.	Аналіз вимог до систем виявлення та реагування на кібератаки.	22.04.2024	
5.	Проектування системи виявлення та реагування на кібератаки великих корпоративних мереж.	08.05.2024	
6.	Оцінка ефективності розробленої системи виявлення і реагування на кібератаки	12..05.2024	
7.	Формулювання висновків за результатами проведеного дослідження.	20.05.2024	
8.	Оформлення роботи.	22.05.2024	
9.	Оформлення презентації.	03.06.2024	
10.	Отримання рецензії на роботу.	03.06.2024	
11.	Захист в ДЕК.	12.06.2024	

Здобувач вищої освіти _____ Євгеній МИЩЕНКО
(підпис) *(Ім’я, ПРІЗВИЩЕ)*

Керівник
кваліфікаційної роботи _____ Юрій ЩАВІНСЬКИЙ
(підпис) *(Ім’я, ПРІЗВИЩЕ)*

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЗАХИСТУ ІНФОРМАЦІЇ

ПОДАННЯ ГОЛОВІ ЕКЗАМЕНАЦІЙНОЇ КОМІСІЇ ЩОДО ЗАХИСТУ КВАЛІФІКАЦІЙНОЇ РОБОТИ
на здобуття освітнього ступеня бакалавра

Направляється здобувач Мищенко Є.І. до захисту кваліфікаційної роботи
(прізвище та ініціали)
за спеціальністю 125 Кібербезпека та захист інформації
(код, найменування спеціальності)
освітньої програми Управління інформаційною та кібернетичною безпекою
(назва)
на тему: “Розробка та реалізація системи виявлення та реагування на кібератаки у великих корпоративних мережах ”

Кваліфікаційна робота і рецензія додаються.

Директор ННІЗІ _____ Віталій САВЧЕНКО
(підпис) *(Ім’я, ПРІЗВИЩЕ)*

Висновок керівника кваліфікаційної роботи

Здобувач МИЩЕНКО Євгеній у кваліфікаційній роботі проаналізував особливості кібербезпеки великих корпоративних мереж, існуючі системи виявлення і реагування на кібератаки та вимоги до них, дослідив методи створення систем та спроектував систему виявлення і реагування на кібератаки, оцінив її ефективність, розробив практичні рекомендації за темою дослідження.

МИЩЕНКО Євгеній показав розуміння проблеми дослідження та бачення основних теоретичних і практичних напрямів її вирішення, довів спроможність самостійного застосування методів наукового дослідження, проявив себе як організований, відповідальний виконавець. Результати дослідження апробовані на всеукраїнській науково-практичній конференції.

Все це дозволяє оцінити кваліфікаційну роботу здобувача МИЩЕНКО Євгенія на оцінку “відмінно” та присвоїти їй кваліфікацію бакалавра з кібербезпеки за освітньою програмою Управління інформаційною та кібернетичною безпекою.

Керівник кваліфікаційної роботи _____ Юрій ЩАВІНСЬКИЙ
(підпис) *(Ім’я, ПРІЗВИЩЕ)*

“____“_____2024 року

Висновок кафедри про кваліфікаційну роботу

Кваліфікаційна робота розглянута. Здобувач Мищенко Є.І. допускається до захисту даної роботи в Екзаменаційній комісії.

Завідувач кафедри
управління інформаційною
та кібернетичною безпекою _____ Світлана ЛЕГОМІНОВА
(підпис) *(Ім’я, ПРІЗВИЩЕ)*

ВІДГУК РЕЦЕНЗЕНТА
на кваліфікаційну бакалаврську роботу

здобувача вищої освіти МИЩЕНКО Євгенія
на тему “ Розробка та реалізація системи виявлення та реагування на кібератаки у великих корпоративних мережах ”

Актуальність. Останніми роками кількість кібератак на великі корпоративні мережі постійно зростає. Традиційні засоби захисту, такі як фаєрволи та антивірусні програми, часто не здатні ефективно протидіяти новим типам загроз. Це вимагає розробки нових методів і підходів до забезпечення кібербезпеки. За даними досліджень, кількість інцидентів, пов'язаних із витоком даних, зросла на десятки відсотків щорічно. Це свідчить про те, що кіберзлочинці стають все більш активними і винахідливими у своїх методах. Тема розробки та реалізації системи виявлення та реагування на кібератаки у великих корпоративних мережах є надзвичайно актуальною в сучасних умовах. Вона відповідає викликам, з якими стикаються організації по всьому світу, і має значний потенціал для наукових досліджень та практичного застосування.

Позитивні сторони.

1. У роботі досліджено особливості кібербезпеки великих корпоративних мереж, спроектовано систему виявлення та реагування на кібератаки мереж, оцінена ефективність створеної системи..

2. Кваліфікаційна робота оформлена відповідно до вимог. Виклад матеріалу здійснено відповідно до плану, зроблено логічні висновки. Ключові положення роботи представлено у вигляді рисунків.

3. Автор опрацював значну джерельну базу, яка складається з 40 науково-технічних та методичних публікацій, в тому числі англомовних.

4. За результатами дослідження запропоновані рекомендації з практичного застосування.

Недоліки.

Доцільно було приділити більше уваги альтернативних методів розробки систем реагування на кібератаки .

Однак, вищезгадані зауваження не впливають на загальну позитивну оцінку кваліфікаційної роботи.

Висновок: Кваліфікаційна робота виконана на належному науково-методичному рівні і заслуговує оцінки “відмінно”, а здобувач МИЩЕНКО Євгеній заслуговує присвоєння кваліфікації бакалавра з кібербезпеки за освітньою програмою Управління інформаційною та кібернетичною безпекою.

Рецензент: _____ _____ _____
(науковий ступінь вчене звання) *підпис* Ім’я, ПРІЗВИЩЕ

РЕФЕРАТ

Кваліфікаційна робота присвячена розробці та реалізації системи виявлення та реагування на кібератаки. Робота складається зі вступу, чотирьох розділів, що містять 59 рисунків, 5 таблиць, висновків і списку використаних джерел із 28 найменувань та 4 додатки. Загальний обсяг роботи становить аркушів, з яких 10 аркушів займають перелік умовних скорочень, список використаних джерел та додатки.

Метою роботи є розробка та реалізації системи виявлення та реагування на кібератаки у великих корпоративних мережах.

Об'єктом дослідження є великі корпоративні мережі, які піддаються кібератакам та потребують ефективних заходів захисту.

Предмет дослідження – система виявлення та реагування на кібератаки у великих корпоративних мережах.

Методи дослідження: контент-аналіз, порівняльний аналіз при дослідженні наукових джерел; експериментальний метод при проектуванні системи виявлення та реагування на кібератаки; Penetration Testing при тестуванні та оцінці здатності системи до виявлення та реагування; методи навчання штучних нейронних мереж при розробленні моделі.

Як результат у роботі проаналізовано особливість кіберзахисту корпоративних мережі та атак, що на них здійснюються. Розроблено та впроваджено систему виявлення та реагування на кібератаки, яка була протестована з використанням реальних даних, що підтвердило її ефективність та надійність.

Галузь застосування. Розроблена система виявлення та реагування на кібератаки може бути використана для захисту великої корпоративної мережі.

Ключові слова: КІБЕРБЕЗПЕКА КОРПОРАТИВНИХ МЕРЕЖ, СИСТЕМА ВИЯВЛЕННЯ ТА РЕАГУВАННЯ НА КІБЕРАТАКИ, МОДЕЛЮВАННЯ СИСТЕМ .

ABSTRACT

The qualification work is dedicated to the development and implementation of a system for detecting and responding to cyberattacks. The work consists of an introduction, four chapters containing 60 figures, 5 tables, conclusions, and a list of 28 references, along with 4 appendices. The total volume of the work is pages, of which 10 pages are occupied by the list of references, and the appendices.

The purpose of the work is to develop and implement a system for detecting and responding to cyberattacks in large corporate networks.

The object of the research is large corporate networks that are subject to cyberattacks and require effective protection measures.

The subject of the research is the system for detecting and responding to cyberattacks in large corporate networks.

Research methods include content analysis, comparative analysis in the study of scientific sources; the experimental method in designing the detection and response system; Penetration Testing in testing and evaluating the system's ability to detect and respond; and methods for training artificial neural networks when developing the model.

As a result, the work analyzes the features of cyber protection for corporate networks and the attacks directed at them. A system for detecting and responding to cyberattacks was developed and implemented, which was tested using real data, confirming its effectiveness and reliability.

Field of application. The developed system for detecting and responding to cyberattacks can be used to protect large corporate networks.

Keywords: CORPORATE NETWORK CYBERSECURITY, CYBERATTACK DETECTION AND RESPONSE SYSTEM, SYSTEM MODELING

ЗМІСТ

ВСТУП

Актуальність теми. На сьогоднішній день технології відіграють важливу роль у діяльності будь-якої компанії. Корпоративні мережі стають все більш складними та вразливими перед кібератаками.

Кібератаки стають все більш складними і шкідливими, завдаючи серйозної шкоди як фінансовому, так і репутаційному стану компанії. Ефективне виявлення і реагування на кіберзагрози стає критично важливим завданням для забезпечення безпеки компанії.

Саме тому розробка та реалізація системи виявлення та реагування на кібератаки у великих корпоративних мережах є вкрай актуальною темою дослідження.

Мета роботи полягає у розробці та реалізації системи виявлення та реагування на кібератаки у великих корпоративних мережах.

Для досягнення цієї мети в роботі необхідно виконати наступні завдання:

1. Проаналізувати сучасний стан кібербезпеки великих корпоративних мереж та систем виявлення і реагування на кібератаки.
2. Здійснити аналіз вимог до систем виявлення та реагування на кібератаки.
3. Спроектувати систему виявлення та реагування на кібератаки великих корпоративних мереж.
4. Оцінити ефективність розробленої системи виявлення і реагування на кібератаки.

Об'єкт дослідження **–** великі корпоративні мережі, які піддаються кібератакам та потребують ефективних заходів захисту.

Предмет дослідження – система виявлення та реагування на кібератаки у великих корпоративних мережах.

Методи дослідження: контент-аналіз, порівняльний аналіз при дослідженні наукових джерел; експериментальний метод при проектуванні системи виявлення та реагування на кібератаки; Penetration Testing при

тестуванні та оцінці здатності системи до виявлення та реагування; методи навчання штучних нейронних мереж при розробленні моделі.

Наукова новизна одержаних результатів. В результаті проведеного дослідження було удосконалено систему виявлення та реагування на кібератаки у великих корпоративних мережах, яка відрізняється від відомих тим, що використовує штучний інтелект на основі машинного навчання для виявлення невідомих раніше атак, розроблена удосконалена архітектура системи виявлення та реагування на основі системи запобігання вторгнень з відкритим вихідним кодом Snort, що дозволяє за рахунок динамічного розподілу ресурсів забезпечити високу швидкість виявлення та реагування в реальному часі.

Практичне значення одержаних результатів. Отримані результати можуть бути практично використані в великих корпоративних мережах з метою підвищення рівня кібербезпеки компанії та зменшення можливих збитків.

Апробація результатів кваліфікаційної роботи відбулася на Всеукраїнській науково-практичній конференції “Стратегії кіберстійкості: управління ризиками та безперервність бізнесу” 28 лютого 2024 року.

Розділ 1 АНАЛІЗ СУЧАСНОГО СТАНУ КІБЕРБЕЗПЕКИ ВЕЛИКИХ КОРПОРАТИВНИХ МЕРЕЖ

1.1 Види кібератак на великі корпоративні мережі та їх характеристики

Корпоративна мережа – це мережа, головне завдання якої, полягає в підтримці конкретного підприємства (рис.1.1.), що володіє даною мережею. Користувачами корпоративної мережі є тільки співробітники даного підприємства [1].

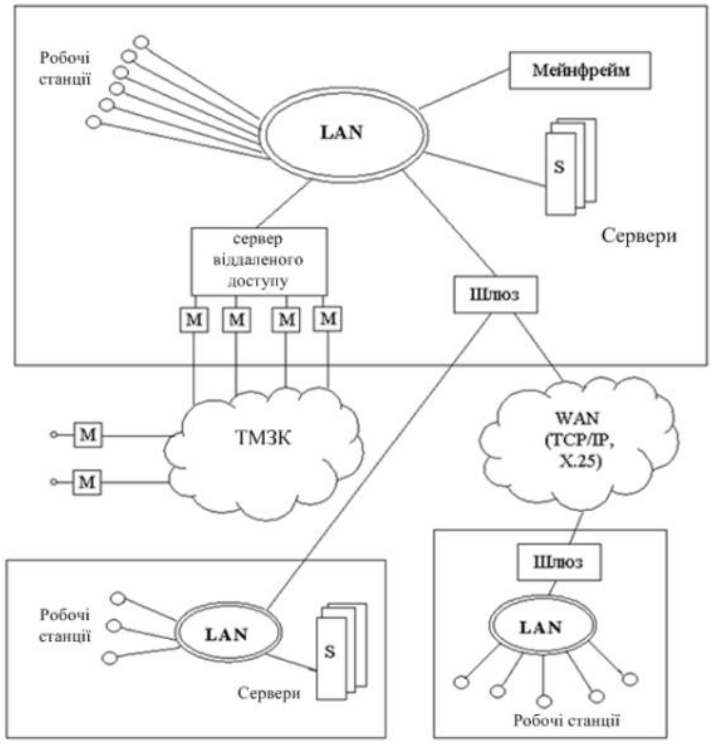

Рис. 1.1. Узагальнена структура корпоративної мережі

Така мережа зазвичай має ієрархічну структуру, де існують різні рівні доступу та прав доступу до ресурсів. Головною метою корпоративної мережі є забезпечення ефективної роботи всіх підрозділів та співробітників організації шляхом швидкого доступу до необхідної інформації та ресурсів.

Такі мережі можуть бути побудовані на основі різних технологій (рис. 1.2 - 1.3), включаючи традиційні провідні та бездротові з’єднання. Крім того,

корпоративна мережа може використовувати різні види архітектур, таких як клієнт-серверна або однорангова, залежно від потреб організації та обсягу роботи, яку вона виконує.

Рис. 1.2. Клієнт-серверна архітектура

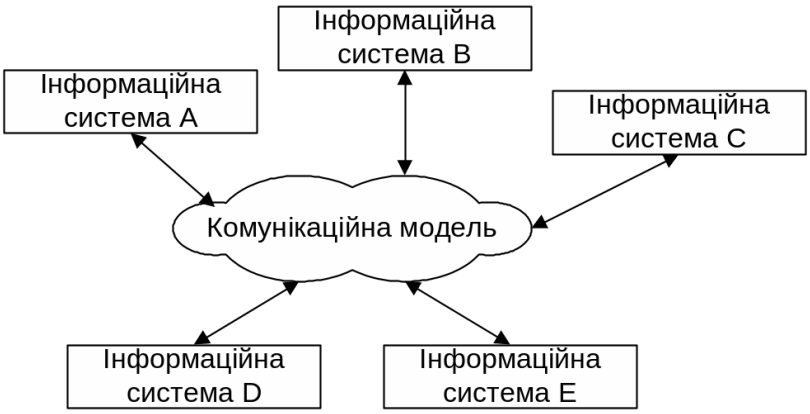

Рис. 1.3. Однорангова архітектура

Класифікація мережі за областю дії враховує географічний район, охоплений мережею та, в меншому ступені, розмір мережі.

Виділяються такі типи [1]:

- персональні мережі (Personal Area Networks - PAN);
- локальні мережі (Local Area Networks - LAN);

- кампусні мережі (Campus Area Networks);
- глобальні мережі (Wide Area Networks - WAN);
- віртуальні приватні мережі (Virtual Private Networks - VPN);
- однорангові мережі (Peer-to-Peer networks).

Топологія мережі, з свого боку визначає структуру зв'язків між її складовими частинами. Це ключовий аспект, який впливає на ефективність передачі і обробки інформації, а також на стійкість мережі до атак [1] (рис. 1.4, табл. 1.1).

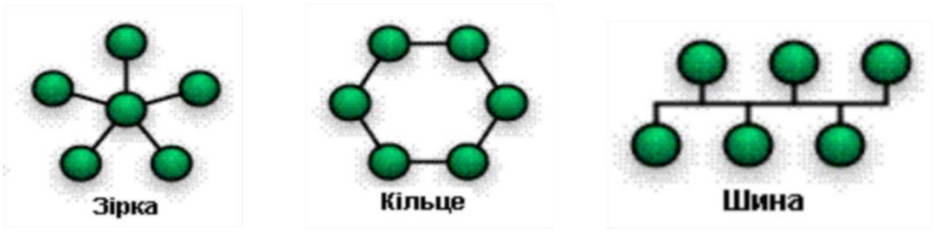

Рис. 1.4. Основні типи топології мережі

Таблиця 1.1

Порівняльна характеристика різних мережних топологій

Характеристика	**Топологія**		
	Зірка	**Кільце**	**Шина**
Вартість розширення	Незначна	Середня	Середня
Приєднання абонентів	Пасивне	Активне	Пасивне
Захист від відмов	Незначний	Незначний	Високий
Розмір мережі	Будь-який	Будь-який	Обмежений
Захист від прослуховування	Добрий	Добрий	Незначний
Вартість підключення	Незначна	Незначна	Висока
Поводження мережі при високих навантаженнях	Добре	Задовільне	Незадовільне
Можливість роботи в реальному режимі часу	Дуже добра	Добра	Незадовільне
Розведення кабелю	Добре	Задовільне	Добре
Обслуговування	Дуже Добре	Середнє	Середнє

В епоху інформаційних технологій великі корпоративні мережі стають головними цілями для кібератак. Кібератаки на корпоративні мережі являють собою поширену загрозу, яка розвивається, і здатна спричинити велику шкоду для організації. Розуміння різних типів кібератак має першочергове значення для організації, щоб посилити захист кібербезпеки та ефективно зменшити потенційні ризики.

Мережеві атаки різноманітні, як і цільові системи. Деякі із них складні, інші прості. Для розпізнавання атак важливо розуміти протокол TCP/IP. Поширення стека протоколів TCP/IP показало його слабкі сторони. Виявлення мережевих атак ускладняються тим, що системи мають надмірну функціональність ІС. Через це, цей тип є одним із лідерів небезпеки.

Спеціалісти розділяють атаки на мережу за характером впливу на два типи (рис. 1.5):

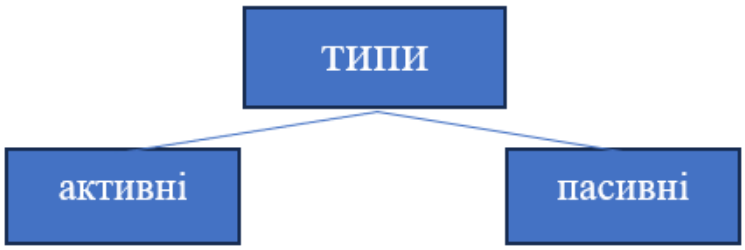

Рис. 1.5. Блок-схема «типи атак»

«Активна атака» - це спроби зловмисника змінити вміст повідомлень, загрожуючи цілісності та доступності системи. Внаслідок цієї атаки система завжди пошкоджена, а системні ресурси можуть бути змінені.

«Пасивна атака» - це тип атак, коли зловмисник спостерігає або копіює вміст повідомлень, загрожуючи конфіденційності без нанесення системі шкоди [2]. До цієї категорії підпадає таке шкідливе програмне забезпечення як прослуховувачі (sniffers).

Також важливо зазначити, що існують три різновиди мережевих атак:

- атаки розвідки;
- атаки відмови в обслуговуванні;

- атаки отримання доступу;

Частіше за все дані типи атак використовують комбіновано, щоб досягти цілей. Види кібератак на великі корпоративні мережі різняться від методів, якими зловмисник буде користуватися [3].

В таблиці 1.2 наведено основні механізми реалізації різних видів атак.

Таблиця 1.2

Основні механізми реалізації атак

№	Тип атаки	Механізм реалізації атаки
1	Віддалене проникнення	Віддалений виклик командного рядка шляхом переповнення буфера
2	Аналіз топології мережі	Передача мережних пакетів
3	Пошук уразливості	Сканування хосту
4	Відмова в обслуговуванні	Передача великої кількості мережних пакетів
5	Злам паролів	Багаторазові спроби аутентифікації в системі
6	Аналіз трафіка	Перехоплення трафіка
7	Несанкціонована аутентифікація	Порушення прав доступу
8	Шкідливе ПЗ	Приховане встановлення програмних модулів, прихований запуск процесів

Атаки розвідки є першим етапом у підготовці атаки на мережу. Зловмисник використовує їх для збору інформації, яка дозволить отримати інформацію про слабкі місця.

Атаки типу «Відмова в обслуговуванні» (DoS, Denial of Service) та (DDos, Distributed Denial of Sendee). DDoS-атаки є еволюцією DoS-атак. Різниця між ними полягає в тому що DoS – це атака з одного пристрою, популярність DoS-атак виправдовується її простотою реалізації. Якщо використовується велика кількість пристроїв, це можна називати розподіленою атакою «відмови в обслуговуванні», або DDoS [4]. DDoS може відбуватися на рівні мережі, наприклад, за допомогою надсилання великих обсягів пакетів SYN/ACK, що

можуть перезавантажити сервер, або на рівні програми шляхом виконання складних SQL-запитів [5].

До атак отримання доступа можна віднести:

1. Мережеві атаки типу «людина посередені» (MITM, Man in the middle). Відбувається, коли зловмисники перехоплюють трафік, що передається між мережою та зовнішніми джерелами даних. У багатьох випадках хакери здійсюють атаку типу «людина посередені» за допомогою слабких протоколів безпеки [6].
2. Неавторизований доступ. Відноситься до мережевих атак, під час якої зловмисник отримує доступ без отримання дозволу. Даний тип атаки може виникати через слабкі паролі, незашифровані мережі, внутрішні загрози [7].

SQL-ін’єкції – отримання несанкціонованого доступу до бази даних, шляхом введення шкідливого коду в поля звичайної форми [8].

Узагальнюючи всі типи атак, можна провести їх характеристику згідно рисунку 1.6.

Рис. 1.6. Типовий перелік типів атак на корпоративну мережу

Віддалене проникнення - атаки, що виконуються з віддалених мереж, зазвичай через Інтернет, з метою отримання несанкціонованого доступу до мережевих ресурсів.

Локальне проникнення - атаки, які вимагають фізичного доступу до мережевого обладнання або інфраструктури для виконання злому або отримання несанкціонованого доступу.

Віддалена відмова в обслуговуванні - атаки, які спрямовані на перевантаження мережевої інфраструктури або серверів, щоб призвести до відмови в обслуговуванні для законних користувачів.

Локальна відмова в обслуговуванні - атаки, які спрямовані на перешкоджання роботі локальних мережевих ресурсів або сервісів, зазвичай шляхом перевантаження або використання вразливостей.

Мережеві сканери - програми або інструменти, що використовуються для виявлення активних хостів та портів у мережі для подальшого аналізу або атаки.

Сканери вразливостей - інструменти, які сканують мережу або систему на наявність вразливостей, які можуть бути використані для несанкціонованого доступу або атак.

Зламники паролів - програми або методи, що використовуються для незаконного отримання доступу до системи або облікових записів, шляхом перебору або злому паролів.

Аналізатори протоколів (сніфери) - інструменти, які перехоплюють і аналізують мережевий трафік для виявлення конфіденційної інформації, такої як паролі або дані сесії.

Несанкціонований доступ до інформаційних ресурсів системи - атаки, що мають на меті отримання доступу до конфіденційної інформації або ресурсів системи без відповідних дозволів

Підозріла активність - виявлення або моніторинг діяльності, яка може свідчити про можливу атаку або порушення безпеки мережі.

Системні атаки - атаки, які спрямовані на злам або використання вразливостей в операційних системах або програмному забезпеченні для отримання несанкціонованого доступу або контролю над системою

З переліку типів атак на корпоративну мережу видно, що загрози для інформаційної безпеки охоплюють широкий спектр технік та методів.

1.2 Огляд існуючих систем виявлення та реагування на кібератаки

Виявлення кібератак і реагування на них є критично важливим для організації. Існує безліч систем виявлення та реагування на кібератак, які спрямовані на ідентифікацію потенційних загроз в мережі та інформаційних системах (Далі ІС). Незалежно від моделі та методу виявлення атак, виявлення атак і реагування на них мають відповідати потребам ІС. Деякі з них виявляють загрози на основі аналізу мережевого трафіку, інші використовують методи машинного навчання для виявлення інцидентів.

У відповідності із постановою Кабінету Міністрів України від 23 грудня 2020 р. № 1295 система виявлення вразливостей і реагування на кіберінциденти та кібератаки (рис. 1.7) – це сукупність програмних та програмно-апаратних засобів, які забезпечують проведення цілодобового моніторингу, аналізу та передачі телеметричної інформації про кіберінциденти та кібератаки, які відбулися або відбуваються на об’єктах кіберзахисту і можуть мати негативний вплив на їх стале функціонування [9].

Система виявлення та реагування на кібератаки (рис.1.8) – це комплексний підхід, який включає в себе три основні компоненти:

1. Виявлення та загроз (рис.1.9): виявлення вразливостей у мережі, ІС. Передбачає постійний моніторинг підозрілих дій.
2. Розвідка загроз: збір і аналіз інформації про загрози.
3. Реагування на загрозу: апаратні та програмні засоби які виявляють загрози та реагують на них [10].

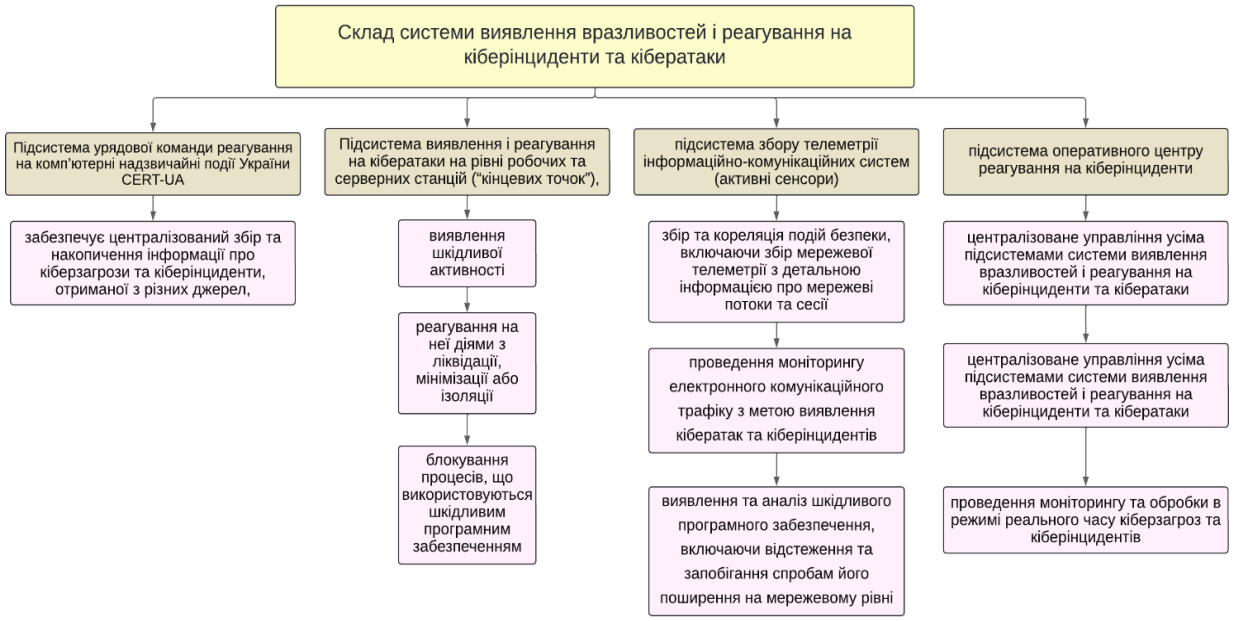

Рис. 1.7. Склад системи виявлення вразливостей і реагування на кіберінциденти та кібератаки

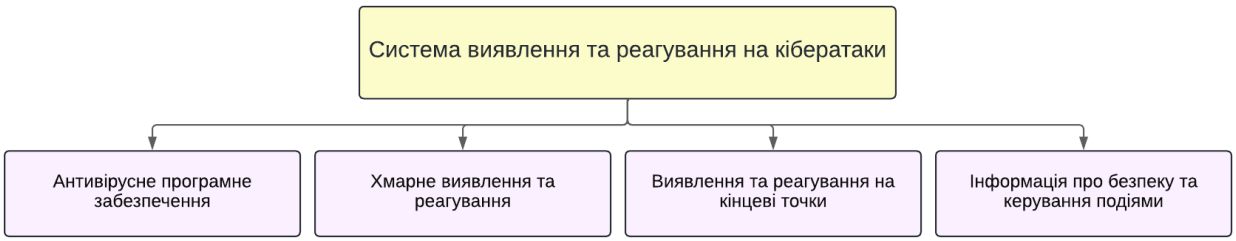

Рис. 1.8. Система виявлення та реагування на кібератаки

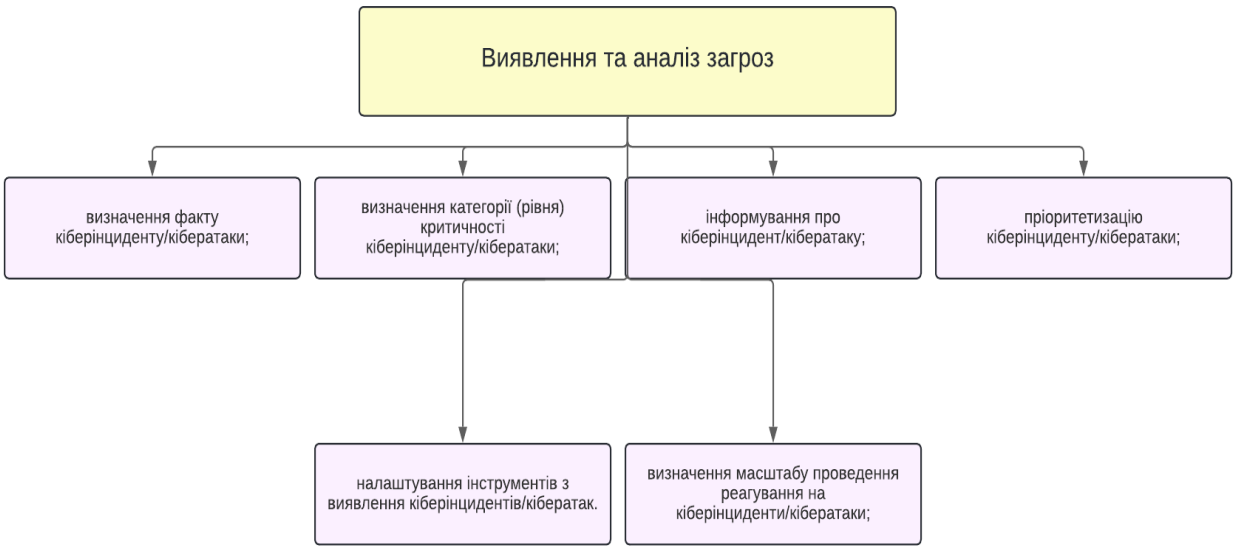

Рис. 1.9. Схема виявлення та аналіз загроз

Добре реалізовані системи виявлення та реагування на кібератаки використовують передові інструменти та методи, та можуть включати в себе:

хмарне виявлення та реагування (CDR, Cloud Detection and Response). Рішення CDR створене для захисту даних, прогам ті інфраструктури на хмарних платформах [11];

виявлення та реагування на кінцеві точки (EDR, Endpoint Detection and Response) – Захищає кінцеві пристрої такі як ноутбуки, ПК, мобільні пристрої, пристрої Інтернету речей, сервери та робочі станції. Основні функції які він виконує це – розслідування інцидентів, ізоляція та локалізація, криміналістичний аналіз, автоматичне реагування та інтеграція з іншими інструментами безпеки [11];

інформація про безпеку та керування подіями (SIEM, Security Information and Event Management) – програмний або програмно-апаратний засіб, який збирає дані про події, пов'язані з безпекою інформації. Далі ці дані аналізують та класифікують [11].

Основним компонентом для захисту корпоративної мережі, яке найчастіше використовують, є:

- розширене виявлення та відповідь (XDR, Extended Detection and Response) розроблена для забезпечення інтелектуальної, автоматизованої та інтегрованої безпеки. На відмінно від захисту кінцевих точок (EDR, Endpoint Detection and Response) XDR (Extended Detection and Response) проводить аналіз такої мережі, веб-трафіку та електроної пошти, робить глибокий аналіз даних на всіх рівнях мережі [12].

Особливостями XDR є:

- забезпечує швидке реагування на загрози;
- інтеграція з існуючими інструментами безпеки SIEM, SOAR;
- високі можливості аналізу.

- «Система виявлення вторгнень» (IDS, Intrusion Detection System) / «Система запобігання вторгненням» (IPS, Image Packaging System), щоб ідентифікувати відомі загрози. Ці системи ефективні проти відомих зловмисних

програм і атак, але їм важко виявляти складні кібератаки, які уникають виявлення на основі сигнатур. Система виявлення вторгнень (IDS) відстежує мережевий трафік/системну діяльність на наявність порушень. Класифікується за такими типами [13]:

- Мережеві СВВ (Network-based IDS, NIDS) – перехоплює весь трафік та аналізує його [14].
- Протокольні СВВ (Protocol-based IDS, PIDS) – відстежує і аналізує комунікаційні протоколи [15].
- Заснована на прикладних протоколах СВВ (Application Protocol-based IDS, APIDS) – веде спостереження і займається аналізом даних, які передаються через незвичні протоколи [16].
- Вузлова СВВ (Host-based IDS, HIDS) – розташована на хості, що відслідковує вторгнення, аналізує логи, файли та інше.

- «Система запобігання вторгненням» (IPS) базується на можливостях IDS, активно запобігаючи атакам. IPS може проактивно захищатись в реальному часі [17].

  Типи IPS:

  - Мережева система запобігання вторгенням (NIPS, Network-based Intrusion Prevention) – розгортається на рівні мережі. Виявляє та блокує угрози в ІС.
  - Система запобігання вторгнень на основі хоста (HIPS, Host-based Intrusion Prevention System) – встановлюється на кінцевій точці. Відстежує трафік до та від кінцевої точки.
  - Аналіз поведінки мережі (NBA, Network Behavior Analysis) – призначений для пошуку аномалій в мережі.
  - Система запобігання бездротовому вторгненню (WIPS, Wireless Intrusion Prevention System) – механізм контролю в бездротовій мережі.

Прикладом IPS/IDS системи є Snort. Snort – є класичною системою IPS та IDS з відкритим вихідним кодом. Snort вміє вести протоколювання, аналізувати

та шукати за вмістом. Застосовується для активного блокування або пасивного виявлення широкого спектра атак та зондувань. На базі Snort буде побудована система виявлення та реагування на кібератаки [18].

- «Мережеве виявлення та реагування» (NDR, Network Detection and response) – це рішення, яке відстежує та аналізує мережевий трафік для виявлення загроз в безпеці. NDR використовує автоматизоване виявлення потенційних загроз, що дозволяє швидко реагувати на них.

Основна відмінність між IDS/IPS і NDR полягає в рівні функціональності та здатності виявлення загроз. NDR пропонує розширені функції, включаючи фільтрацію подій для отримання контекстуальної інформації, застосування методів машинного навчання та штучного інтелекту для виявлення нових та деталізованих сигналів атак, інтерфейсів для пошуку загроз тощо (рис. 1.10).

Рис. 1.10. Різниця між встановленням IDS та IPS систем

Деякі системи NDR значною мірою покладаються на виявлення загроз на основі сигнатур IDS/IPS, але важливо зазначити, що немає IDS/IPS, здатних відповідати комплексним функціям NDR [19].

Основні переваги NDR:

- раннє виявлення загроз;

- швидке реагування на інциденти;
- комплексний аналіз загроз.

Заходи з реагування на кібератаки здійснюються суб'єктами кібербезпеки щоб забезпечити: швидке виявлення кібератаки; інформування відповідних органів та зацікавлених сторін про їх винекнення; запобігання, мінімізація та усунення негативних наслідків; виявлення вразливостей; забезпечення стійкості та надійності кіберзахисту; попередження повторного використання виявленого кіберінциденту та збереження електроних доказів у разі кібератаки [20].

Існуючі рішення для виявлення та реагування на кібератаки – це важливі інструменти для організації, що здійснюють заходи проти кіберзагроз, що ховаються в їхній мережевій інфраструктурі. Ці рішення працюють, безперервно скануючи та аналізуючи мережеву активність, швидко виявляючи можливі порушення безпеки або зловмисну діяльність.

Вони використовують різноманітні алгоритми та методи розпізнавання образів для виявлення аномалій, які можуть свідчити про потенційну загрозу безпеці. Після виявлення загрози ці рішення негайно оцінюють серйозність і потенційний вплив, що дозволяє організаціям вжити рішучих заходів.

Craig MacAlpine разом із «Незалежною платформою для досліджень і оглядів кібербезпеки» Expert Insights у статті написав про «вісім кращих рішень для виявлення та реагування на загрози» [21].

Найпопулярніші рішення про які було зазначено:

1. ESET PROTECT Enterprise. ESET пропонує комплексне рішення для виявлення та реагування – ESET Inspect – як частину пакету ESET PROTECT Enterprise. Компонент Inspect XDR забезпечує оцінку ризиків; дослідження та усунення загроз, включаючи хмарну веб-консоль; захист кінцевої точки; повне шифрування диска та розширений захист від загроз.

Коментар від Expert Insights: ESET PROTECT - надійне рішення з XDR для захисту кінцевих точок, забезпечує швидкий доступ до інформації про загрози і керовану підтримку. Рекомендується для організацій будь-якого розміру, особливо тим, хто цінує комплексний захист і XDR [21].

2. Check Point Infinity SOC – призначений лише для виявлення та реагування на загрози в мережі, хмарі, кінцевих точках, мобільних пристроях та IoT.

Коментар від Expert Insights: передова система безпеки, що використовує штучний інтелект для пошуку та усунення загроз у реальному часі. Надає швидке та точне запобігання атакам на всіх рівнях мережі та кінцевих точках, з детальними звітами і видимістю пристроїв. Рекомендується для організацій від малого та середнього бізнесу до корпоративного рівня [21].

3. Darktrace DETECT & Response – це хмарне рішення, кероване штучним інтелектом, яке легко розгортати. Не потребує технічного обслуговування після запуску. Штучний інтелект дає змогу виявляти атаки нульового дня та внутрішні загрози.

Коментар від Expert Insights: гнучке та ефективне рішення, яке забезпечує високий рівень захисту, виявлення і ізоляцію загроз, включаючи облікові записи, внутрішні загрози та нові види шкідливих програм. Він демонструє свою ефективність у всіх сферах роботи, надаючи операторам SOC детальні звіти та інтуїтивно зрозумілий інтерфейс користувача. Рекомендується для організацій будь-якого розміру, що прагнуть підвищити безпеку мережі [21].

4. Heimdal Extended Detection & Response (XDR) – універсальне, багатошарове та багатофункціональне рішення XDR , яке пропонує комплексний підхід. Особливостями є: потужна та багатофункціональна інформаційна панель, механізм виявлення розширеного захисту від загроз, методики MITRE ATT&CK.

Коментар від Expert Insights: потужне та точне рішення з централізованою керованою консоллю, ідеальний вибір для організацій, що шукають кращий контроль, інтелектуальні функції та автоматичне реагування. Надійність виявлення та підтримки під час адаптації роблять його рекомендованим вибором для організацій будь-якого розміру та галузей [21].

5. Trellix Extended Detection and Response XDR – хмарний продукт, який забезпечує цілодобовий моніторинг і запобігання, класифікуючи сповіщення за пріоритетом. Особливостями є: блокує електрону пошту, мережу

та атаки на кінцеві точки; потужні канали розвідки про загрози; застосування та категоризація загроз; поглиблене відстеження, звітування та аналіз.

Коментар від Expert Insights: інтелектуальний інструмент, який визначає та позначає загрози, пріоритизуючи їх для забезпечення ефективності реагування. Надає розширений контекст і класифікує атаки, включаючи багатоетапні, атаки нульового дня та інші. Забезпечує не лише профілактику, а й глибокий аналіз та відстеження, спрощуючи процес для персоналу [21].

6. Rapid7 Thread Command – потужна програма виявлення загроз і розвідки, яка швидко та ефективно реагує на загрози. Покладається на джерела з відкритої, глубокої та темної мережі.

Коментар від Expert Insights: простий та потужний інструмент, який надає широке розуміння загроз і контексту, спрощуючи усунення несправностей і допомагаючи приймати обґрунтовані рішення. Він пропонує швидке виявлення та надсилає автоматичні відповіді попередження по мережі, щоб забезпечити швидку реакцію на загрози. Рекомендується для організацій різного рівня та розміру, забезпечуючи спрощений підхід до безпеки [21].

7. Vectra Thread Detection and Response Platform – реалізує полювання за загрозами за допомогою штучного інтелекту. Особливостями є: збір даних, журналів і подій у реальному часі по всій мережі, включаючи всіх користувачів і кінцеві точки; Attack Signal Intelligence автоматично виявляє, сортує та визначає пріоритетність невідомих загроз.

Коментар від Expert Insights: гнучкий і всеосяжний інтсрумент виявлення загроз і реагування, має широкий спектр варіантів розгортання. Його функція навчання штучного інтелекту визначає пріоритетність загроз у реальному часі, забезпечуючи ефективне реагування команди [21].

8. Watchguard Thread Detection And Response – це рішення добре інтегрується з антивірусним програмним забезпеченням. Особливостями є: функція Host Sensor забезпечує покращену видимість загроз на всіх кінцевих точках і надсилає адміністраторам евристичні та поведінкові дані; обмеження

хосту та автоматична відповідь допомагають негайно контролювати, ізолювати та реагувати на загрози, щойно їх виявлено за допомогою ThreatSync;

Коментар від Expert Insights: це комбінація комплексних і потужних інструментів безпеки, що надають розширений захист від загроз і зловмисного програмного забезпечення. Відзначається високою ефективністю у боротьбі з різними формами атак, включаючи програми-вимагачі та атаки ботнетів. Рекомендований для малих і середніх підприємств, які шукають доступне та просте використання, а також хочуть підвищити безпеку своєї мережі в хмарному середовищі [21].

За допомогою таблиці 1.3 можемо порівняти системи виявлення та реагування.

Таблиця 1.3

Порівняння різних систем виявлення та реагування на кібератак

Ім’я інструменту	**Тип**	**Опис**
SNORT	IDS, IPS	Програмне забезпечення з відкритим початковим кодом для виявлення та запобігання вторгнень у мережу.
ESET PROTECT Enterprise	EDR	Рішення для ефективного керування безпекою на базі хмари, що забезпечує захист на рівні кінцевих точок.
Check Point Infinity SOC	SOC	Комплексне рішення з централізованим керуванням та аналітикою для захисту мережі, кінцевих точок та хмарних інфраструктур.
Darktrace DETECT & Response	EDR/XDR	Система штучного інтелекту, яка використовує аналіз здатностей та машинного навчання для виявлення та відповіді на загрози.
Heimdal Extended Detection & Response	EDR/XDR	Рішення для виявлення та відповіді на загрози на рівні кінцевих точок та мережі, засноване на машинному навчанні та аналізі.
Trellix Extended Detection and Response XDR	XDR	Рішення, що об'єднує в собі засоби виявлення та відповіді на загрози для кінцевих точок, мереж та хмарних інфраструктур.

Продовження таблиці 1.3

Ім’я інструменту	Тип	Опис
Rapid7 Thread Command	EDR/XDR	Платформа для виявлення та відповіді на загрози, що поєднує в собі функції EDR та XDR.
Vectra Thread Detection and Response Platform	XDR	Рішення для виявлення та відповіді на загрози з використанням аналізу мережевого трафіку та машинного навчання.
Watchguard Thread Detection And Response	XDR	Платформа для виявлення та відповіді на загрози, що об'єднує в собі засоби захисту мережі та кінцевих точок.

У сукупності всі ці інструменти дозволяють виявляти загрози, завчасно зупиняти загрози та зменшувати ризики. Для досягнення ефективної «Системи виявлення та реагуванні на кібератаки» все це повинно працювати в тандемі.

1.3 Аналіз вимог до системи виявлення та реагування на кібератаки

Проаналізувавши дані в попередніх розділах, можна зробити наступні висновки щодо існуючих на сьогодні систем виявлення та реагування на кібератаки.

По-перше. Багато з існуючих систем виявлення кіберзагроз є досить ефективними у виявленні аномалій та потенційно шкідливих дій у мережах. Вони використовують різноманітні методи, включаючи сигнатурне виявлення, машине навчання, штучний інтелект для виявлення загроз.

По-друге. Деякі системи реагування на кібератаки можуть автоматично вживати заходів для стримування або нейтралізації загроз, що виникають. Це може включати блокування шкідливого трафіку, ізоляцію заражених систем, видалення зловмисного програмного забезпечення та інші дії.

Однак, існуючі системи виявлення та реагування на кібератаки мають певні обмеження (рис. 1.11):

- складність та постійне зростання кіберзагроз: Зловмисники постійно розробляють нові методи атак, в результаті це ускладнює їх виявлення та усунення загрози;
- висока вартість впровадження та експлуатації: Впровадження та підтримка систем може бути дуже витратним, що робить їх недоступними для багатьох організацій;
- складність розгортання та налаштування: Деякі системи вимагають значних ресурсів та експертних знань для розгортання та налаштування, що робить їх недоступними для багатьох організацій.

Рис. 1.11. Спільні недоліки сучасних систем виявлення та реагування на кібератаки

По третє, ефективність систем виявлення та реагування залежить від якості
та актуальності використовуваних баз даних загроз, правил та алгоритмів.

Отже, можна сформувати такі вимоги до наближеної до наближеної до ідеальної системи виявлення та реагування на кібератаки:

1. Актуальність бази відомих сигнатур для виявлення всіх відомих загроз та вразливостей;
2. Здатність виявляти невідомі раніше атаки та вразливості;
3. Надійність та неперервність роботи системи, постійний збір необхідних даних;
4. Низький рівень хибних спрацювань та помилок;
5. Забезпечення високої швидкості роботи мережі;

6. Легкість роботи з системою як для користувача, так і для сторонніх програм;
7. Виявлення в режимі реального часу для запобігання потенційній шкоді, що вимагає постійного моніторингу та аналізу трафіку;
8. Автоматичне реагування, яке включає блокування трафіку та ізоляцію заражених пристроїв;
9. Інтеграція з існуючими інструментами безпеки, такими як SIEM та брандмауери;
10. Маштабованість для обробки збільшеного трафіку та складної мережевої архітектури без втрати продуктивності;

Впровадження системи, яка відповідає цим вимогам, значно підвищить рівень кібербезпеки та захистить організацію від кібератак

Висновки до розділу 1.

У цьому розділі розглянуто різні види атак, які можуть виникати у корпоративних мережах, а також їх особливості.

Крім того, було оглянуто різні системи виявлення та реагування на кібератаки, які допомагають підприємствам захищати свою інфраструктуру. Ці системи включають в себе інструменти IDS/IPS, системи виявлення та реагування (EDR), XDR.

Було проаналізовано 9 таких систем, порівнявши їх основні характеристики та можливості. Серед них виділяється Snort своєю простотою встановлення та запуску, що робить його привабливим вибором для багатьох організацій. Тому в даній кваліфікаційній роботі система виявлення та реагування на кібератаки буде побудована на основі Snort.

Розділ 2 ПРОЕКТУВАННЯ СИСТЕМИ ВИЯВЛЕННЯ ТА РЕАГУВАННЯ НА КІБЕРАТАКИ

2.1 Визначення функціональних та нефункціональних вимог

Функціональні вимоги – це перелік вимог до системи, які описують що система повинна робити, які завдання повинна виконувати і які можливості повинна підтримувати [22].

Основними функціональними вимогами є:

- система повинна вміти виявляти різні типи кібератак, такі як вторгнення, ШПЗ, DDoS-атаки та інші;
- крім виявлення має вміти чітко аналізувати виявлені загрози, визначати їх джерело та потенційний вплив на систему;
- система повинна надавати своєчасні сповіщення про виявленні загрози;
- автоматичне реагування на кібератаки.

Нефункціональні вимоги – це вимоги до програмного комплексу, що не стосуються його прямої функціональності. Вони описують якісні характеристики системи, такі як надійність, продуктивність, безпека, конфіденційність, зручність користування, сумісність, підтримуваність, масштабованість тощо [23].

Основними нефункціональними вимогами є:

- продуктивність: система повинна мати можливість обробляти великі обсяги даних та швидко реагувати на кібератаки;
- система повинна бути масштабованною, щоб забезпечувати потрібну продуктивність;
- система повинна бути доступною 24/7;
- інтерфейс повинен бути зручним.

Ці вимоги до системи виявлення та реагування на кібератаки допоможуть забезпечити розробку ефективної та надійної системи.

2.2 Вибір архітектури системи

Архітектура системи – це сукупність зв'язків між частинами системи. Вона описує, як різні компоненти системи взаємодіють один з одним, щоб досягти загальної мети. Це високорівнева структура, яка визначає основні елементи системи та їх взаємозв'язки [24].

Архітектура системи також включає в себе визначення ключових принципів, паттернів та стратегій, які використовуються для побудови системи. Вона визначає загальну організацію, структуру та поведінку системи, а також забезпечує основу для розвитку, модифікацій та розширень у майбутньому. Правильно спроектована архітектура дозволяє забезпечити ефективність, надійність, масштабованість та інші важливі якості системи протягом її життєвого циклу.

Обираючи IDS/IPS для розробки системи виявлення та реагування на кібератаки, було керовано рядом ключових факторів.. Перш за все, IDS та IPS є важливим компонентом мережі, оскільки дозволяють виявляти та реагувати на потенційну шкідливу діяльність в реальному часі. Це дозволяє оперативно реагувати на загрози та мінімізувати шкоду, завдану корпоративній інфраструктурі.

Крім того, вибір IDS/IPS ґрунтується на їхній здатності аналізувати мережевий трафік та виявляти нестандартні, підозрілі або відомі типи атак, використовуючи різні методи та підходи.

Загалом, вибір IDS/IPS для системи виявлення та реагування на кібератаки в корпоративній мережі аргументований їхньою ефективністю, здатністю до автоматизації та широким спектром функціональності.

Щодо основи для системи, було взято Snort через його відкритий код, широкий спектр підтримуваних протоколів та плагінів, а також активну спільноту користувачів та розробників. Використання Snort дозволить побудувати потужну та надійну систему виявлення та реагування на кібератаки, забезпечуючи високий рівень безпеки.

Рисунок 2.1. показує як в загальному повинна виглядати системи виявлення та реагування на кібератаки.

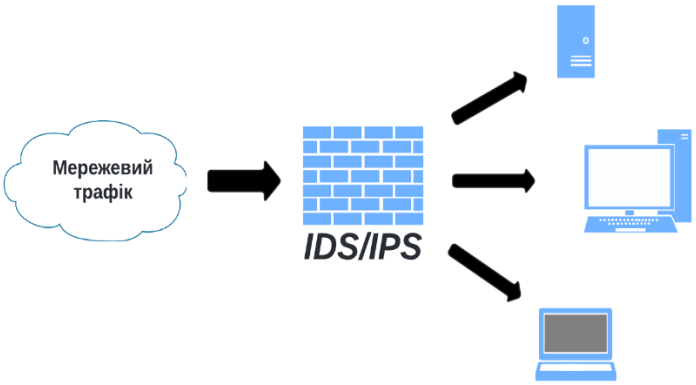

Рис. 2.1. Система виявлення та реагування на кібератаки

Було зазначено раніше, що система виявлення та реагування на кібератаки буде побудована на базі Snort. Тому доцільно буде оглянути основні компоненти архітектури Snort, які показав на рисунку 2.2.

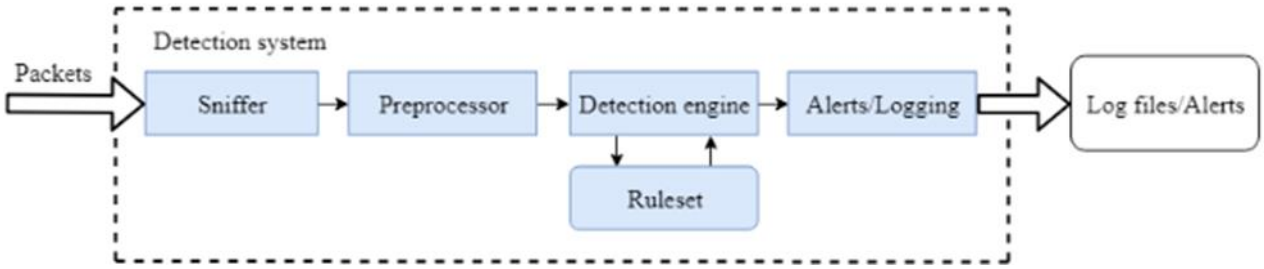

Рис. 2.2. Архітектура Snort

Аналізатор пакетів (sniffer) пакетів захоплює мережевий трафік і передає його на дешифратор пакетів (preprocessor), який обробляє отримані пакети, щоб ізолювати заголовки протоколу на кожному з рівнів моделі OSI (це концептуальна модель, що використовується для розуміння та опису мережевих протоколів (рис.2.3). Вона складається з семи рівніх, які представляють різні функції комунікаційних систем, починаючи від фізичного з'єднання до рівня додатків. Кожен рівень виконує певні завдання, забезпечуючи стабільну та ефективну передачу даних через мережу [25]). Виявлення відбувається в блоці механізм

виявлення (detection engine). Цей модуль аналізує кожен пакет і перевіряє його на відповідність правилам. Наступним етапом є система реєстрації/оповіщення (logger/alerts), запускається кожного разу, коли виявляється пакет, на цьому етапі відбувається реагування на загрозу [26].

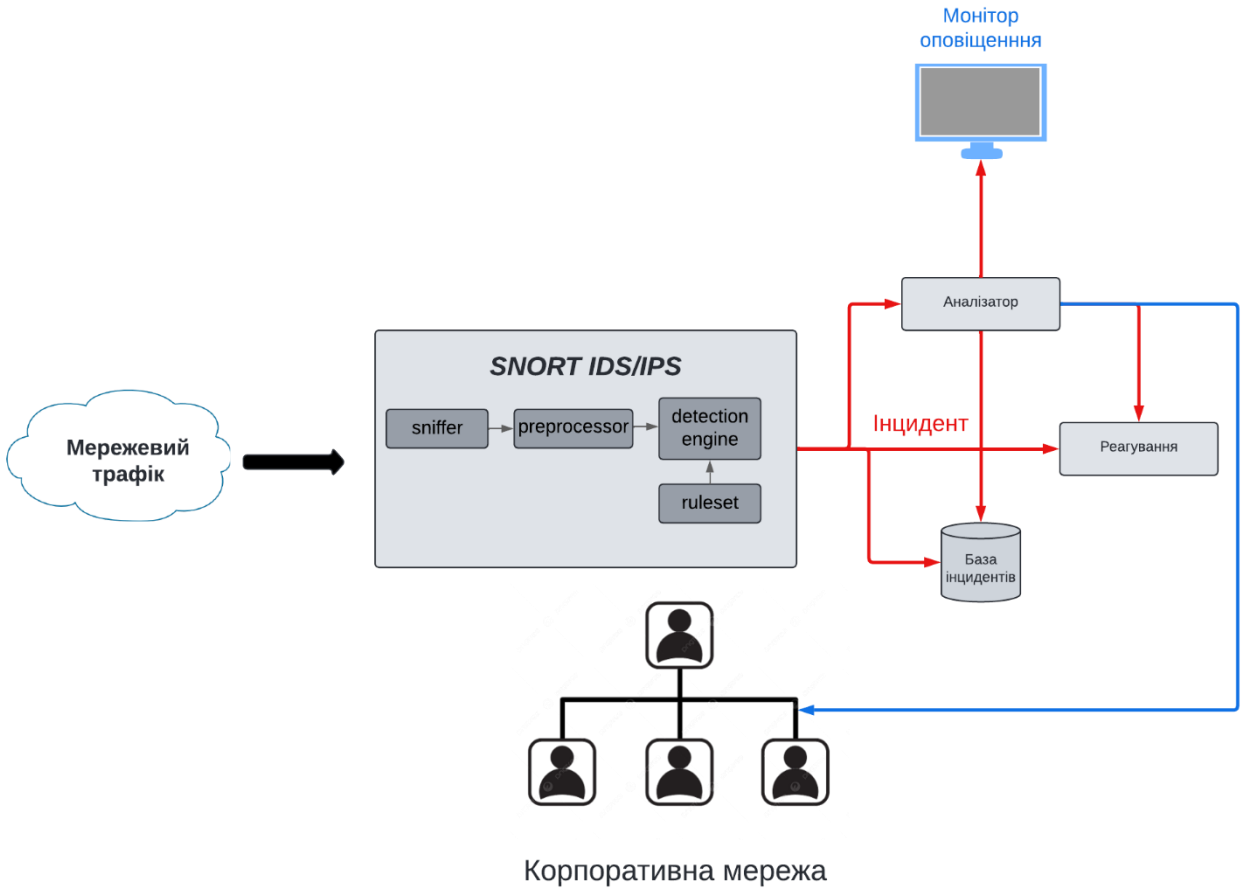

Рис. 2.3. Архітектура запропонованої системи

Система, яка запропонована зображена на рисунку 2.3, працює наступним чином: спочатку умовно заражений пакет потрапляє в поле зору IDS/IPS. Snort проводить аналіз за допомогою алгоритму, який буде заданий. Після цього пакет може бути пропущений або заблокований, і застосовуються відповідні заходи безпеки.

Система виявлення та реагування на кібератаки надає низку переваг для забезпечення кібербезпеки корпоративної мережі. Вона виконує функції моніторингу, аналізу, виявлення загроз та реагування на них, забезпечуючи таким

чином цілісність та захист ІС. Розробка та впровадження вдосконалених алгоритмів аналізу даних і виявлення загроз є актуальним напрямом досліджень.

2.3 Розробка алгоритмів виявлення та реагування на кібератаки

Щодня виявляють нові вразливості, критичні для функціонування ІС. Аналітики безпеки розбирають ці нові вразливості, виділяють те, що необхідно для їх запуску, і пишуть сигнатури (ознаки атаки або вірусу, що використовуються для їх виявлення [27]) для блокування будь-яких експлойтів, націлених на них.

Такий підхід до захисту дуже добре захищає мережі від відомих загроз, але що, якщо загроза невідома? Що робити, якщо вразливість виявлена, експлойт для неї написаний, а спільнота безпеки про неї нічого не знає? Потрібен інший підхід до захисту, який не потребує попереднього знання про атаку, щоб функціонувати.

Одним з підходів до захисту, який може допомогти у виявленні невідомих загроз, є використання систем машинного навчання для виявлення аномалій. Цей підхід полягає в тому, щоб навчити комп'ютерну систему розпізнавати звичну поведінку мережі чи програмного забезпечення, а потім виявляти будя-які відхилення від цієї звичної моделі.

Машинне навчання може використовувати алгоритми класифікації або кластеризації для виявлення аномалій у трафіку мережі, змінах у використанні ресурсів системи чи незвичайних діях програм. Наприклад, модель може автоматично сповіщати про можливу атаку, якщо спостерігається надмірна активність, яка не відповідає звичному шаблону.

Звичайно, цей підхід також має свої обмеження, такі як необхідність постійного оновлення моделей навчання з урахуванням нових варіантів загроз. Але він може бути корисним додатковим інструментом у боротьбі з невідомими загрозами, особливо коли традиційні методи виявлення не ефективні.

Для розробки алгоритмів виявлення та реагування на кібератаки будуть використані такі інструменти [28]: бібліотека TensorFlow, фреймворк Keras на

основі архітектури SnortML. Програмний код буде написаний на мові програмування Python.

Бібліотека в програмувані – це колекція попередньо написаного коду, який розробники можуть використовувати повторного у своїх програмах. Це економить час і зусилля, адже їм не потрібно писати той самий код з нуля. Бібліотеки пропонують безліч функціональних можливостей [29].

Бібліотека TensorFlow – це відкрита бібліотека для машинного навчання. Вона використовується для створення та навчання моделей машинного навчання. TensorFlow вважається кращою в своєму класі.[30].

Keras – це інструмент для створення нейронних мереж, який надає зручний інтерфейс для розробки моделей. Він базується на Python і може використовувати в якості надбудови над TensorFlow. Основна перевага полягає в тому, що Keras дозволяє швидко прототипувати моделі завдяки своїй простоті, модульності та можливості розширення.

Цей фреймворк підтримує як згорткові, так і рекурентні мережі, а також їх комбінації. Крім того, для обчислень можна використовувати як процесор, так і графічну відеокарту, що робить його досить універсальним у плані апаратних можливостей [31].

Для реалізації алгоритму на основі машинного навчання в якості середовища розробки була використана безкоштовна версія PyCharm 2023.3.5 від компанії JetBrains.

Спочатку з офіційного сайту потрібно встановити Python. Далі потрібно підготувати середовище. Заходимо в консоль (cmd), переходимо до папки де буде зберігатися програмний код та прописуємо такі команди (рис. 2.4):

```
PS D:\model1> python3 -m venv venv
Python
PS D:\model1>
```

Рис. 2.4. Створення віртуального середовища

Ця команда створює віртуальне середовище Python з назвою venv у поточній директорії. У віртуальному середовищі будуть зберігатися всі установлені пакети, що дозволяє уникнути конфліктів з іншими версіями пакетів у системі (рис. 2.4).

Наступним кроком буде активація віртуального середовища (рис. 2.5):

```
PS D:\model1> source venv/bin/activate
```

Рис. 2.5. Активація віртуального середовища

Ця команда активує віртуальне середовище Python, щоб командна оболонка використовувала його версію Python та пакети (рис. 2.5).

Далі потрібно встановити TensorFlow. Для цього в консолі прописується «pip install tensorflow». Дана команда встановлює бібліотеку TensorFlow в активованому середовищі. Після цього буде доступ до всіх функцій та класів TensorFlow (рис. 2.6).

```
Successfully installed MarkupSafe-2.1.5 absl-py-2.1.0 astunparse-1.6.3 certifi-2024
.2.2 charset-normalizer-3.3.2 flatbuffers-24.3.25 gast-0.5.4 google-pasta-0.2.0 grp
cio-1.63.0 h5py-3.11.0 idna-3.7 keras-3.3.3 libclang-18.1.1 markdown-3.6 markdown-i
t-py-3.0.0 mdurl-0.1.2 ml-dtypes-0.3.2 namex-0.0.8 numpy-1.26.4 opt-einsum-3.3.0 op
tree-0.11.0 packaging-24.0 protobuf-4.25.3 pygments-2.18.0 requests-2.31.0 rich-13.
7.1 setuptools-69.5.1 six-1.16.0 tensorboard-2.16.2 tensorboard-data-server-0.7.2 t
ensorflow-2.16.1 tensorflow-intel-2.16.1 termcolor-2.4.0 typing-extensions-4.11.0 u
rllib3-2.2.1 werkzeug-3.0.3 wheel-0.43.0 wrapt-1.16.0
PS D:\model1\pythonProject>
```

Рис. 2.6. Успішне встановлення TensorFlow

Наступним кроком буде написання коду, У цьому кроці імпортується необхідні бібліотеки для проекту. Зокрема, налаштовується змінна середовища TF_CPP_MIN_LOG_LEVEL на значення '3', щоб приховати повідомлення з низьким рівнем журналювання від TensorFlow (рис. 2.7).

```
import os
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3'

import numpy as np
import tensorflow as tf
from tensorflow.keras import layers
from urllib.parse import unquote_to_bytes
```

Рис. 2.7. Імпорт необхідних бібліотек та налаштування середовища TensorFlow

Далі потрібно написати код, який буде обробляти дані найпопулярніших атак для подальшого використання у моделі (рис.2.8).

```
data = [
    {'str':'foo=1%27%20OR%201=1%2D%2D', 'attack':1 },
    {'str':'<script>alert("XSS Attack");</script>', 'attack':1 },
    {'str':'ping -c 1 example.com', 'attack':1 },
    {'str':'../../../../../etc/passwd', 'attack':1 },
    {'str':'<img src="http://malicious-site.com/transfer?amount=10000&recipient=attacker" onload="this.submit()">', 'attack':1 },
    {'str':'username=admin&password=\' or \'1\'=\'1', 'attack':1 },
    {'str':'<img src=x onerror=alert(document.cookie)>', 'attack':1 },
    {'str':'ls | cat', 'attack':1 },  # Інша Command Injection атака
    {'str':'../../../../../../../../../../../etc/passwd', 'attack':1 },
    {'str':'<a href="http://malicious-site.com/attack" onclick="document.forms[0].submit(); return false;">Click me</a>', 'attack':1 }
]
```

Рис. 2.8. Прикладні дані

Далі потрібно обробити рядки запитів, перетворивши їх у числові послідовності та зберегти мітки атак (рис. 2.9).

```
maxlen = 1024

X = []
Y = []

1 usage
def decode_query(str):
    return unquote_to_bytes(str.replace('+', ' '))


for item in data:
    arr = decode_query(item['str'])[:maxlen]
    arrlen = len(arr)
    seq = [0] * maxlen
    for i in range(arrlen):
        seq[maxlen - arrlen + i] = arr[i]
    X.append(seq)
    Y.append(item['attack'])
```

Рис. 2.9. Код обробки даних

Після цих кроків можна приступати до побудови та навчання моделі машинного навчання. Наступним етапом буде написання коду для створення, компіляції та навчання моделі (Додаток А).

На цьому кроці можна визначити архітектуру моделі. Використовується «Sequential» API бібліотеки TensorFlow для послідовного визначення шарів моделі. В нашому випадку (рис. 2.10) вона включає в себе вхідний шар (Input Layer), вбудований шар (Embedding Layer), LSTM шар (LTSM Layer) та щільний шар (Dense Layer) з активацією «sigmoid».

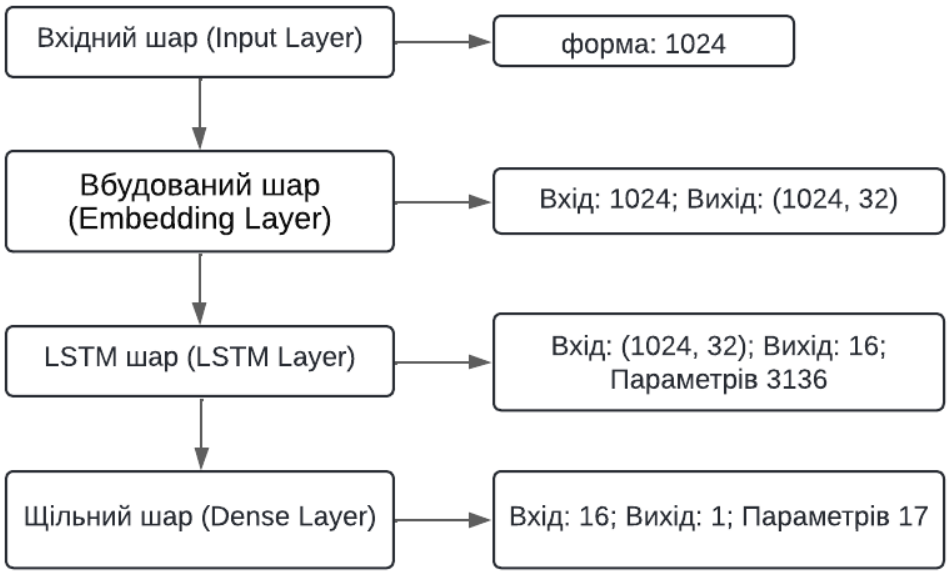

Рис. 2.10. Схема послідовності шарів

Input Layer має форму 1024. Це вхідний шар, який приймає послідовності даних фіксованої довжини 1024. Він не має параметрів, його головна роль - передача вхідних даних до наступного шару.

Embedding Layer. Вхідний розмір - 1024. Вихідний розмір - (1024, 32), це означає, що є дві осі які мають розмірність 1024, а друга -32. Це може інтерпретуватися як масив з 1024 рядків, кожен з яких має розмірність по 32 елементи. Всього в цьому шарі 8192 параметри, які представляють собою ваги для векторного представлення кожного токену.

LSTM Layer. Вхідний розмір - (1024, 32), що вказує на послідовність векторів з Embedding Layer. Вихідний розмір - (16), що вказує на 16-мірний вектор, що представляє відомості, витягнуті з послідовності векторів вхідних даних. Цей шар має 3136 параметрів, які використовуються для ваг LSTM і пов'язаних з ними входів та вихідів.

Dense Layer. Вхідний розмір – 16. Вихідний розмір - 1, оскільки цей шар має один нейрон, який видає одне значення. Загалом цей шар має 17 параметрів: 16 для ваг, які зв'язують кожен вхід з нейроном, та 1 зсув.

Структура моделі на рисунку 2.11.

```
model = tf.keras.Sequential([
    layers.Input(shape=(maxlen,), batch_size=1),
    layers.Embedding( input_dim: 256,  output_dim: 32),
    layers.LSTM(16),
    layers.Dense( units: 1, activation='sigmoid')])
```

Рис. 2.11. Побудова моделі

Наступним кроком є компіляція моделі, це крок включає функцію втрати, оптимізатор і метрики для оцінки моделі (рис.2.12).

```
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
```

Рис. 2.12. Команда «Компіляція моделі»

Для відображення структури моделі використаємо команду «model.summary()».. Вона використовується щоб перевірити правильність налаштувань та кількість параметрів (рис. 2.13).

```
C:\Users\zenine\AppData\Local\Programs\Python\Python312\python.exe D:\model1\pythonProject\main.py
Model: "sequential"
┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━┓
┃ Layer (type)                    ┃ Output Shape           ┃       Param # ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━┩
│ embedding (Embedding)           │ (1, 1024, 32)          │         8,192 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ lstm (LSTM)                     │ (1, 16)                │         3,136 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ dense (Dense)                   │ (1, 1)                 │            17 │
└─────────────────────────────────┴────────────────────────┴───────────────┘
 Total params: 11,345 (44.32 KB)
 Trainable params: 11,345 (44.32 KB)
 Non-trainable params: 0 (0.00 B)
```

Рис. 2.13. Структура навчання моделі

Наступним етапом є навчання моделі, цей етап включає навчання на наших даних. Тут потрібно вказати кількість епох (epoch - це один цикл усього навчального набору даних [32]) та розмір пакета для навчання (рис.2.14).

```
model.fit(np.asarray(X).astype(np.float32),
          np.asarray(Y).astype(np.float32),
          epochs=100, batch_size=1)
```

Рис. 2.14. Код «Навчання моделі»

Далі модель зберігається за допомогою функції «model.save(“model”)», а потім конвертується в формат TFLite для оптимізації та використання на пристроях з обмеженими ресурсами (рис.2.15).

```
model.export("model")

converter = tf.lite.TFLiteConverter.from_saved_model("model")

classifier_model = converter.convert()

with open('classifier.model', 'wb') as f:
    f.write(classifier_model)
```

Рис. 2.15. Код «Збереження моделі та конвертація в TFLite»

Завершальним етапом розробки коду для моделі є його випробування. Запускаємо код за допомогою інтерпретатора Python щоб побачити результат (рис.2.16).

```
Epoch 98/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 2s 169ms/step - accuracy: 1.0000 - loss: 5.7385e-04
Epoch 99/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 2s 162ms/step - accuracy: 1.0000 - loss: 5.6218e-04
Epoch 100/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 2s 164ms/step - accuracy: 1.0000 - loss: 5.5642e-04
Saved artifact at 'model'. The following endpoints are available:
```

Рис. 2.16. Успішне навчання моделі

На рисунках 2.17, 2.18 показано, що після досягнення 100-ї епохи можна визнати, що модель вичерпала можливості для подальшого вдосконалення та досягла оптимального рівня навчання. Це значить про успішне навчання моделі.

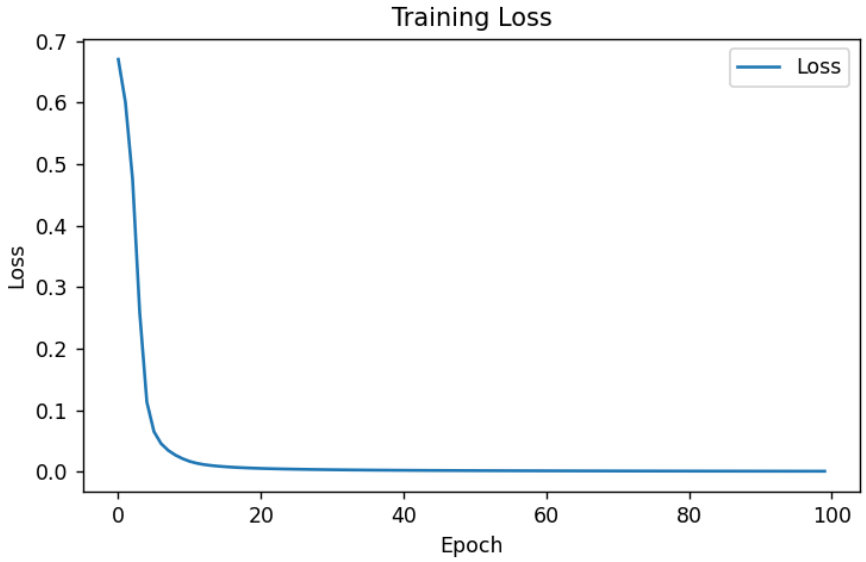

Рис. 2.17. Графік втрат

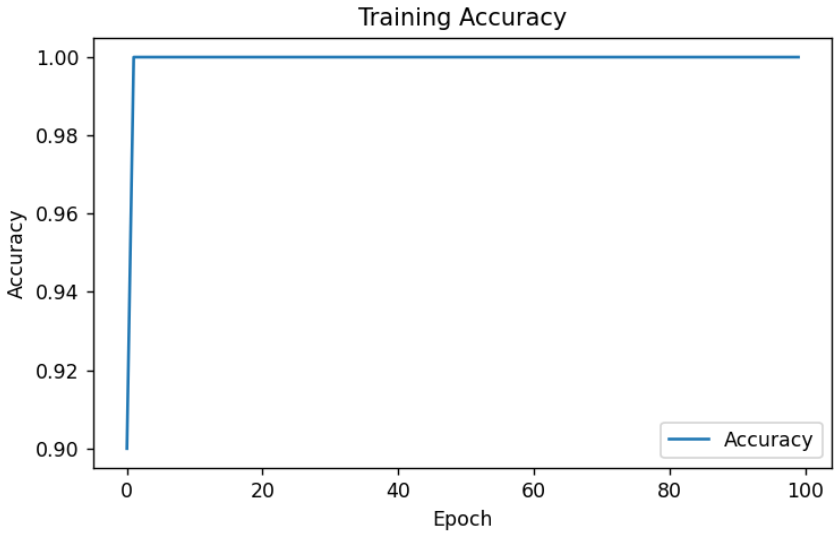

Рис. 2.18. Графік точності

За допомогою бібліотеки «matplotlib.pyplot» можна відобразити графіки втрат та точності. Графіки показують зміну значень функції втрати та метрики точності на тренувальному наборі даних протягом кожної епохи. На графіках показано, що з кожною епохою значення функції втрати зменшується, а точність збільшується, що свідчить про те, що модель успішно навчається на вхідних даних і набуває здатності до класифікації атак та безпечних запитів.

Висновки до розділу 2.

У цьому розділі розглянуто функціональні та нефункціональні вимоги до системи виявлення та реагування на кібератаки. Також обрана архітектура, на якій буде побудована система виявлення та реагування на кібератаки. Вона відрізняється від відомих тим, що може легко розгортатися без великих втрат на налаштування. Крім того, було розроблено алгоритм для виявлення та реагування на загрози за допомогою машинного навчання, що дає змогу автоматизувати процес виявлення аномальної активності та швидко реагувати на потенційній загрози

Розділ 3 РЕАЛІЗАЦІЯ СИСТЕМИ ВИЯВЛЕННЯ ТА РЕАГУВАННЯ НА КІБЕРАТАКИ

3.1 Вибір технологій та інструментів реалізації

Перш за все, щоб втілити систему виявлення та реагування на кібератаки, необхідно створити лабораторію, де буде демонструватися робота цієї системи. Лабораторія буде побудована на платформі VirtualBox, щоб ефективно відображати функціонал даної системи. VirtualBox – це інструмент із графічним інтерфейсом, який дозволяє розгортати сервери, операційні системи як віртуальні машини [33]. Розглянемо спосіб роботи лабораторії за схемою, яка показана на рисунку 3.1.

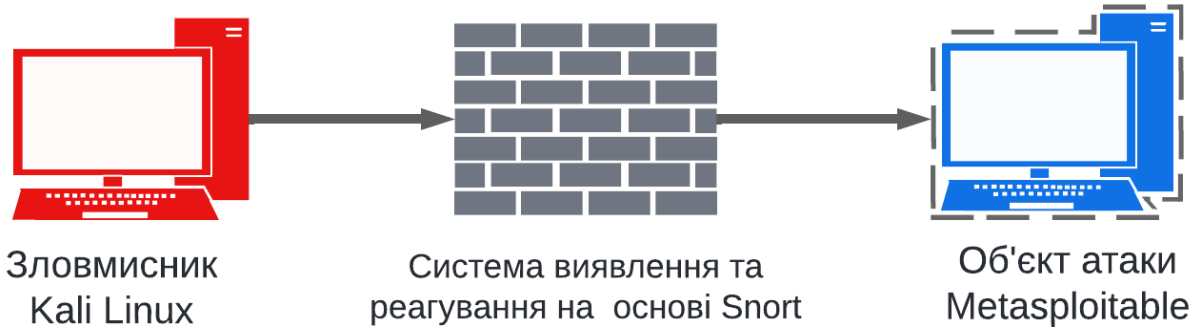

Рис. 3.1. Схема лабораторії

В цій конфігурації лабораторії спостерігаємо комплексну систему виявлення та реагування на потенційні атаки в мережі. Процес роботи розділяється на три ключові складові, кожна з яких відіграє свою роль:

1. Зловмисник (Kali Linux).

Цей етап відображає роль атакуючої сторони, яка використовує систему Kali Linux. Kali Linux – це операційна система, створена для спеціалістів з кібербезпеки. Вона базується на Linux і містить багато інструментів для тестування на проникнення, цифрової криміналістики та аудиту системи. Можна

сказати, що це арсенал для хакерів або етичних хакерів. Інструменти Kali Linux допомагають знайти слабкі місця в комп'ютерних системах [34].

2. Система виявлення та реагування на основі Snort.

На цьому етапі використовується система Snort, яка буде базуватися на операційній системі Ubuntu. Ubuntu – це безкоштовна операційна система з відкритим кодом, створена на базі ядра Linux. Вона відома своєю зручністю для користувачів, стабільністю та великою спільнотою користувачів [35]. Для того, щоб Snort міг працювати на основі недавно створеного алгоритму, необхідно встановити та налаштувати інструмент під назвою SnortML.

3. Об'єкт атаки (Metasploitable).

Цільова система Metasploitable слугує об'єктом атак. Metasploitable - це навмисно вразлива віртуальна машина, яку можна використовувати для проведення атак [36]. Metasploitable буде імітацією корпоративної мережі.

3.2 Розробка програмного забезпечення

Для розробки програмного забезпечення, що реалізує систему виявлення та реагування на кібератаки, першим кроком є встановлення віртуального середовища (рис. 3.2). Для цього використовується VirtualBox. Переходимо на офіційний сайт та завантажуємо VirtualBox [37] (Додаток Б).

Рис. 3.2. VirtualBox

Після встановлення VirtualBox необхідно створити віртуальні машини для кожного компонента лабораторії. Для цього необхідно завантажити відповідні образи операційних систем та налаштувати їх параметри відповідно до вимог системи.

Першим кроком буде створення віртуальної машини для системи виявлення та реагування на кібератаки на основі Snort. Для цього переходимо на офіційний сайт Ubuntu та завантажуємо образ. Після чого створюємо машину в VirtualBox. Запускаємо та встановлюємо її (рис.3.3).

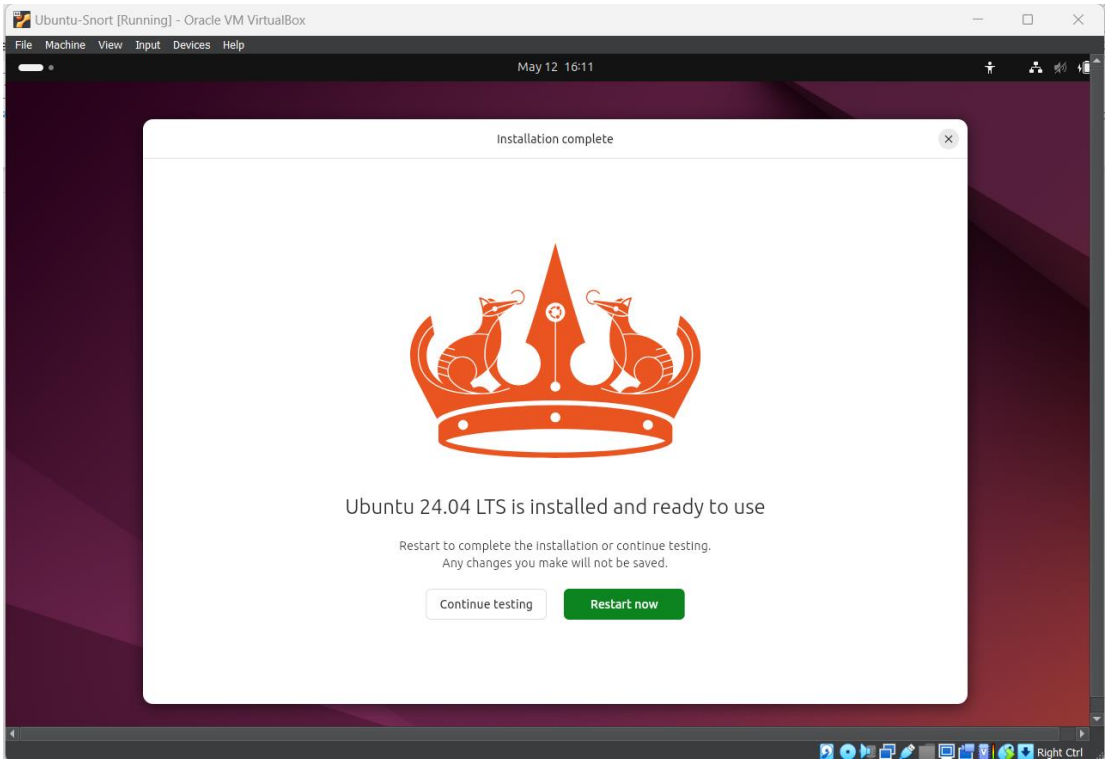

Рис. 3.3. Встановлений Ubuntu

Другим кроком буде встановлення Metasploitable, це в нас об’єкт атаки. Заходимо на офіційний сайт Metasploitable та завантажуємо образ та встановлюємо його на VirtualBox (рис.3.4).

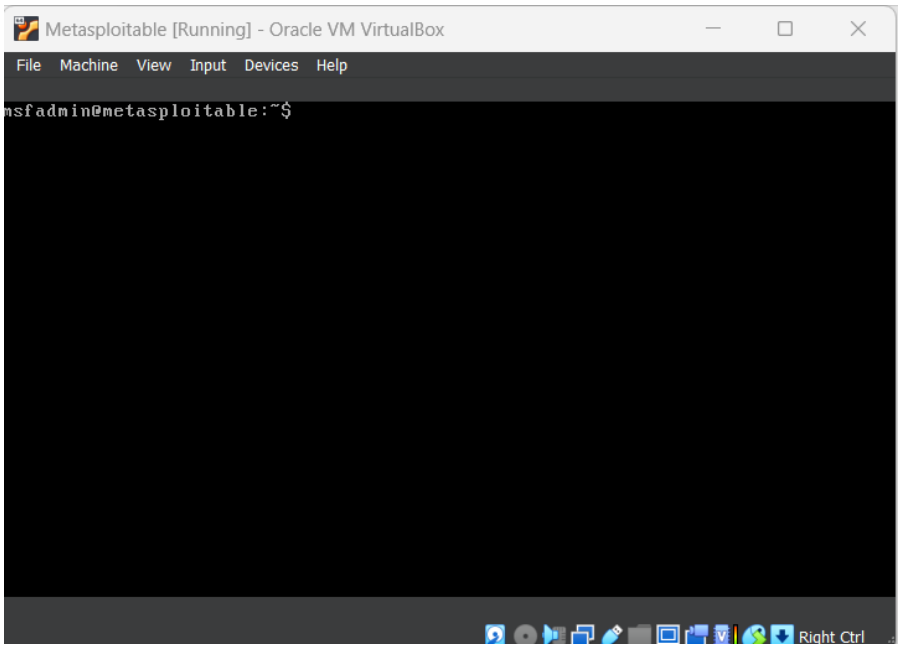

Рис. 3.4. Встановлений Metasploitable

Після створення віртуальних машин необхідно налаштувати мережеве з'єднання між ними, щоб забезпечити взаємодію компонентів системи. Це можна зробити шляхом налаштування віртуальних мереж VirtualBox та налаштування самих систем. Процес налаштування показаний в Додатку А. Дивлячись на рисунок 3.5, зловмисник на Kali Linux отримав IP «10.0.3.4», а об'єкт атаки Metasploitable отримав IP «10.0.1.1». Тоді як система виявлення та реагування отримає три IP адреси: Основна адреса Ubuntu IP 10.0.2.15; IP 10.0.3.5 буде слугувати для під'єднання до Kali Linux; IP 10.0.1.3 буде точкою підключення до об'єкту атаки (Додаток В).

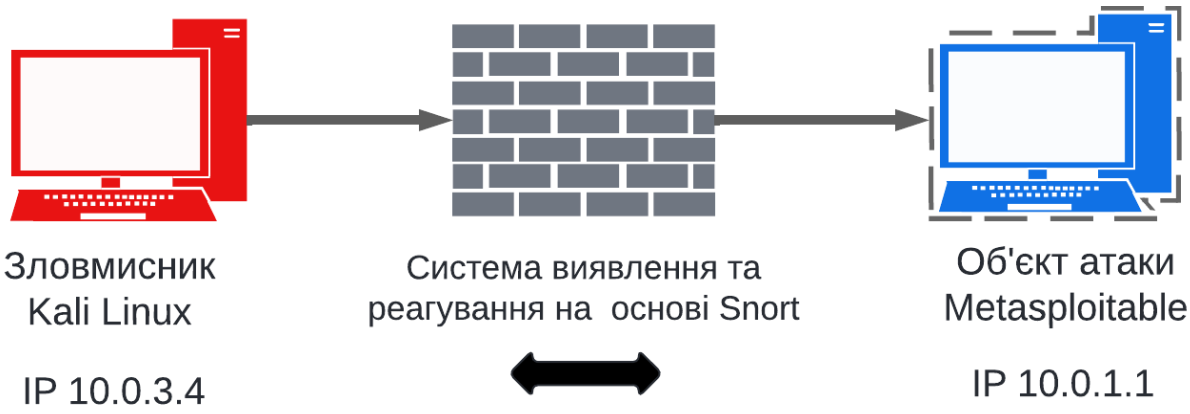

Рис. 3.5. Розподілення IP адрес для віртуальної лабораторії

Налаштовуємо лабораторію так, щоб зловмисник ніяк не зміг обійти систему виявлення та реагування. Потрібно використати команду traceroute, яка потрібна для визначення маршруту проходження даних. На рисунку 3.6 показано, що пакет спочатку потрапляє на віртуальну машину Ubuntu (Snort) а вже потім на об’єкт атаки.

Рис. 3.6. Використання команди traceroute

Після завершення створення віртуальної лабораторій потрібно встановити та налаштувати систему виявлення та реагування на кібератаки на основі Snort. Для цього запускаємо Ubuntu, переходимо в консоль та розпочинаємо налаштування. Для початку потрібно встановити бібліотеки, щоб Snort коректно працював. За допомогою команди яка показана на рисунку 3.7 потрібно встановити їх.

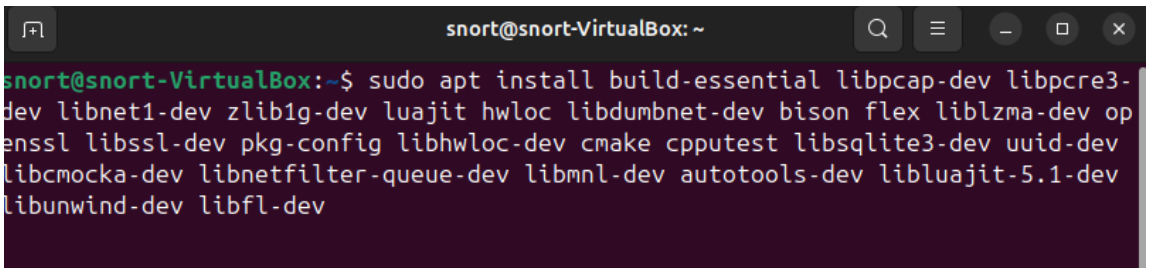

Рис. 3.7. Процесс встановлення бібліотек

Після встановлення всіх бібліотек настав час встановити Snort (рис.3.8). Для цього заходимо на офіційний сайт Snort, завантажуємо архів та розархівовуємо його для подальшого встановлення системи.

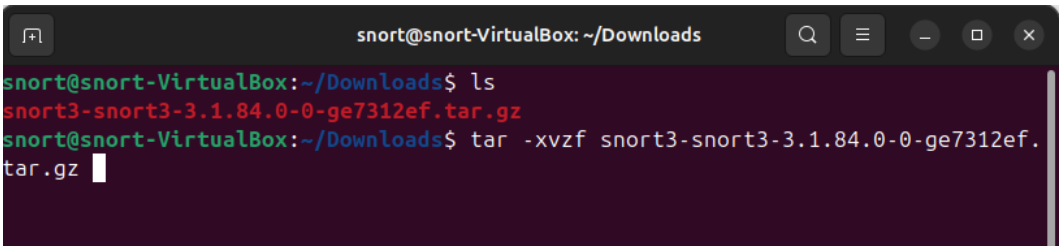

Рис. 3.8. Процес встановлення Snort

Далі потрібно скомпілювати збірку, щоб мати змогу встановити Snort. Консольні команди які компілюють та встановлюють бачимо на рисунку 3.9 та 3.10.

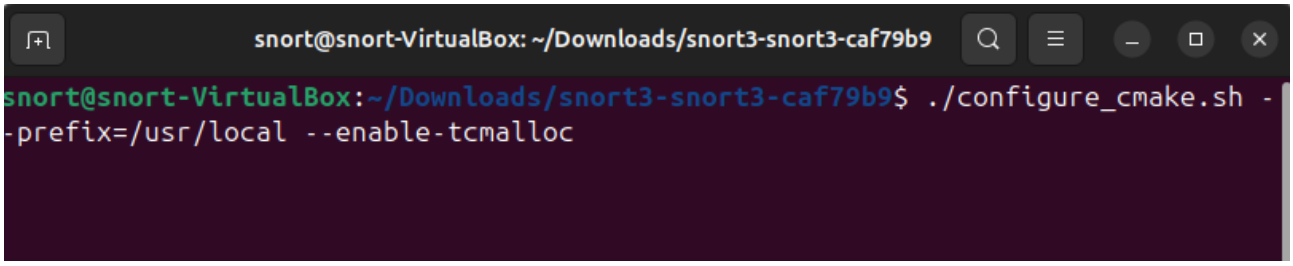

Рис. 3.9. Підготовчий етап до компіляції

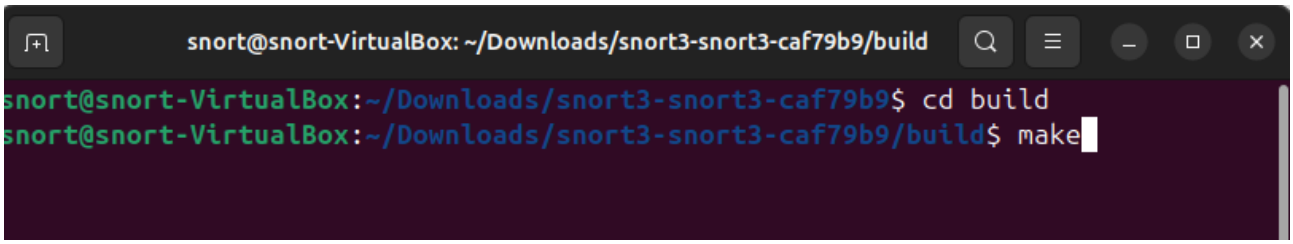

Рис. 3.10. Компіляція та встановлення Snort

Щоб переконатися, що установка пройшла успішно, вводимо у консолі «snort -v». Ця команда не лише перевірить коректність встановлення, але й покаже версію Snort. Згідно з рисунком 3.11 встановлено версію 3.1.84, яка є найсвіжішою та підтримує машине навчання.

Заключним етапом буде налаштування Snort. Спочатку потрібно налаштувати карту мережевого інтерфейсу. Це потрібно для того, щоб Snort бачив трафік який проходить через нього.

```
snort@snort-VirtualBox:~/Downloads/snort3-snort3-caf79b9/build$ snort -v
--------------------------------------------------
o")~   Snort++ 3.1.84.0
--------------------------------------------------
--------------------------------------------------
Network Policy : policy id 0 :
--------------------------------------------------
Inspection Policy : policy id 0 :
--------------------------------------------------
pcap DAQ configured to passive.
--------------------------------------------------
host_cache
    memcap: 33554432 bytes

Snort successfully validated the configuration (with 0 warnings).
o")~   Snort exiting
```

Рис. 3.11. Версія Snort

За допомогою консольної команди «ip link set dev enp0s9 promisc on» вмикаємо параметр «Promisc» для мережевої карти (рис.3.12).

```
snort@snort-VirtualBox:~/Downloads/snort3-snort3-caf79b9/build$ ip link set dev
enp0s9 promisc on
```

Рис. 3.12. Параметр «Promisc» для мережевої карти

Вимикаємо розвантаження інтерфейсу, щоб Snort не скорочував великі пакети розміром понад 1518 байт. Процес вимкнення та перевірки бачимо на рисунку 3.13.

```
snort@snort-VirtualBox:~/Downloads/snort3-snort3-caf79b9/build$ sudo ethtool -k
enp0s9 | grep receive-offload
generic-receive-offload: on
large-receive-offload: off [fixed]
snort@snort-VirtualBox:~/Downloads/snort3-snort3-caf79b9/build$ sudo ethtool -K
enp0s9 gro off lro off
snort@snort-VirtualBox:~/Downloads/snort3-snort3-caf79b9/build$ sudo systemctl d
aemon-reload
snort@snort-VirtualBox:~/Downloads/snort3-snort3-caf79b9/build$ systemctl enable
 --now snort3-nic.service
```

Рис. 3.13. Налаштування мережевої карти

Наступним кроком буде налаштування правил Snort для виявлення та реагування на загрози. Спочатку потрібно завантажити найпоширеніші правила,

які використовуються. Для цього переходимо на офіційний сайт Snort, завантажуємо правила, розархівовуємо їх і створюємо папку для правил, та переносимо їх туди (Додаток Г).

Тепер, коли у нас є правила для початку роботи, відкриваємо основний конфігураційний файл Snort, та встановлюємо IP-адресу мережі в параметрі «HOME_NET», яку будемо захищати. У даному випадку будемо захищати IP 10.0.1.1, це IP-адреса «Об'єкта атаки» (рис.3.14).

```
  GNU nano 7.2              /usr/local/etc/snort/snort.lua *
-- 8. configure tweaks

---------------------------------------------------------------------------
-- 1. configure defaults
---------------------------------------------------------------------------

-- HOME_NET and EXTERNAL_NET must be set now
-- setup the network addresses you are protecting
HOME_NET = '10.0.1.1/24S'

-- set up the external network addresses.
-- (leave as "any" in most situations)
EXTERNAL_NET = 'any'

include 'snort_defaults.lua'

---------------------------------------------------------------------------
```

Рис. 3.14. Встановлення значення в файлі «snort.lua»

Після виконання всіх описаних дій можемо реалізувати алгоритм, який був розроблений у розділі 2, відкривши консоль та ввівши консольну команду, яка показана на рисунку 3.15.

```
snort@snort-VirtualBox:~/Downloads$ snort -q --lua 'snort_ml_engine = { http_param_model = "snort.model" };'
snort@snort-VirtualBox:~/Downloads$ snort -q --lua 'snort_ml = {};'
```

Рис. 3.15 – Підключення алгоритму

Після підключення алгоритму (рис.3.15) система готова для виявлення та реагування на кібератаки. Однак для підтвердження коректної роботи необхідно провести тестування.

3.3 Тестування та налагодження

Щоб переконатись, чи система працює коректно, потрібно її протестувати. Для початку за допомогою влаштованої системи перевірки оцінимо синтаксис головних файлів Snort. Для цього використаємо команду "snort -c /usr/local/etc/snort/snort.lua". На рисунку 3.16 можна побачити, що перевірка пройшла успішно, і не виявлено жодної помилки.

```
Snort successfully validated the configuration (with 0 warnings).
o")~   Snort exiting
snort@snort-VirtualBox:~/Downloads$
```

Рис. 3.16. Перевірка синтаксиса

Наступним кроком є перевірка функціональності системи виявлення та реагування на загрози. Для цього буде створено правило, яке блокуватиме використання утиліти nmap на мережевому рівні. Nmap - це безкоштовна програма з відкритим кодом для дослідження мережі та перевірки безпеки [38].

Правило буде функціонувати наступним чином: зловмисник надсилає пакет даних за допомогою nmap, який блокується на етапі системи виявлення та реагування. Детальну схему можна переглянути на рисунку 3.17.

Рис. 3.17. Схема блокування пакету nmap

Спочатку потрібно створити правило для Snort. Відкриваємо віртуальну машину Ubuntu та створюємо файл з назвою «local.rules» в директорії /usr/local/etc/rules/.

У нашому випадку правило буде виглядати таким чином:

«block tcp any any -> $HOME_NET any (msg: "NMAP ping sweep Scan"; dsize:0;sid:10000004; rev: 1;)»

Розберемо кожен елемент правила:

1. «block tcp any any -> $HOME_NET any». Цей фрагмент вказує, що правило буде спрацьовувати на TCP пакунки (nmap відправляє TCP пакунки щоб дізнатися відкриті порти).
2. «msg: "NMAP ping sweep Scan"; dsize:0;sid:10000004; rev: 1;». Повідомлення, яке буде відображатися при спрацюванні правила.
3. «sid:10000004;». Унікальний ідентифікатор правила.
4. «rev:1». Номер ревізії правила, який вказує на версію цього конкретного правила (рис.3.18).

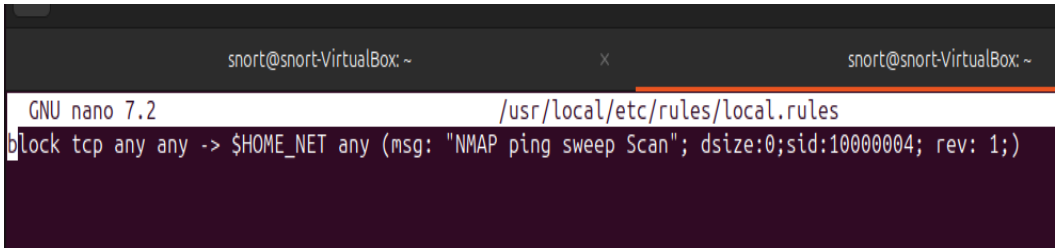

Рис. 3.18. Правило яке блокує nmap

Запускаємо Snort, та за допомогою nmap з віртуальної машини Kali Linux відправляємо,TCP пакети на віртуальну машину Metasploitable. На рисунку 3.19 показано, що пакет успішно заблокований.

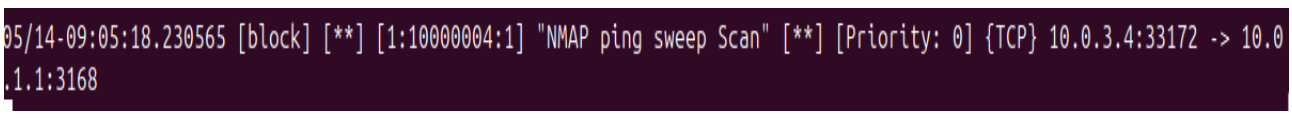

Рис. 3.19. Блокування пакету від nmap

На рисунку 3.20 видно, що 6 пакетів TCP були заблоковані відповідно до цього правило.

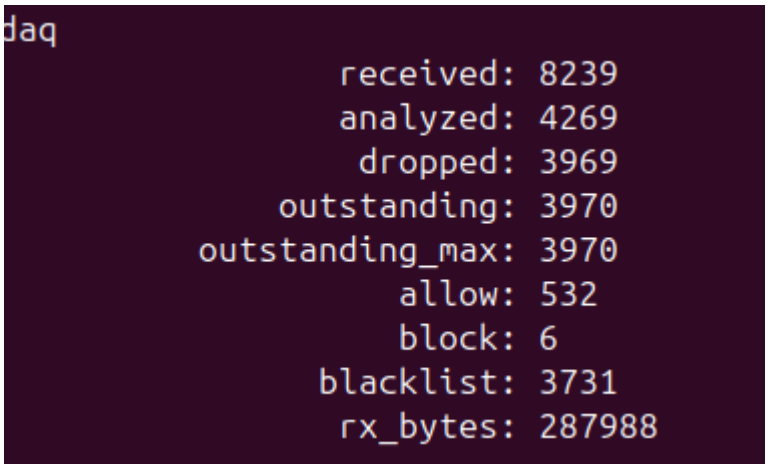

Рис. 3.20. Загальний результат роботи Snort

Незважаючи на те, що було підключено алгоритм на основі машинного навчання, Snort ще не вміє блокувати шкідливі програми. Для цього потрібно зробити декілька налаштувань в конфігураційному файлі "snort.conf", який знаходиться в /etc/snort/snort.conf.

На рисуну 3.21 показано, що були розкоментовані такі рядки (правила):

#include $RULE_PATH/malware-backdoor.rules

#include $RULE_PATH/malware-cnc.rules

#include $RULE_PATH/malware-other.rules

#include $RULE_PATH/malware-tools.rules

Рис. 3.20. Налаштування конфігураційного файла

Висновки до розділу 3.

У цьому розділі було вибрато технології та інструменти для реалізації системи виявлення та реагування. З їх допомогою створено лабораторію для тестування системи виявлення та реагування на основі Snort. Встановили та налаштували Snort для подальшого тестування на реальних даних.

Тестування дало можливість оцінити ефективність системи виявлення та реагування на основі Snort в реальних умовах. Це дозволило перевірити, наскільки ефективно система спрацьовує. Таке тестування є важливим етапом у розробці та впровадженні систем безпеки, оскільки воно дозволяє підвищити рівень захищеності та готовності до реагування на потенційні загрози.

Розділ 4 ОЦІНКА ЕФЕКТИВНОСТІ ТА БЕЗПЕКИ СИСТЕМИ

4.1 Проведення тестування на реальних даних

У цьому розділі акцентовано увагу на проведенні тестування на реальних даних за допомогою Metasploit Framework та інших інструментів. Metasploit Framework є одним із найпотужніших та найпопулярніших інструментів для проведення тестування на проникнення. Він пропонує широкий спектр модулів для експлуатації вразливостей, що дозволяє дослідникам імітувати атаки зловмисників та виявляти слабкі місця в системах.

Тестування на реальних даних (рис.4.1) надає компаніям низку важливих переваг, які допомагають підвищити їхню інформаційну безпеку та стійкість до кіберзагроз.

Рис. 4.1. Ключові аспекти тестування на реальних даних

Спочатку почнемо тестування за допомогою експлойту «distcc_exe». Експлойт «distcc_exe» націлений на вразливість у «distcc». «Distcc» - це служба, що прискорює процес компіляції, використовуючи вільні ресурси обробки інших комп'ютерів у мережі. Коли distcc налаштований на комп'ютері, він може передавати завдання компіляції на іншу систему. Ця система-одержувач повинна мати запущений демон «distccd» та встановлений сумісний компілятор для обробки переданого коду [39].

Основна проблема виникає, коли «distcc» налаштовано без обмежень доступу. Зокрема, він слухає TCP-порт 3632 і приймає завдання компіляції від будь-якого віддаленого хоста. Ця відсутність автентифікації означає, що зловмисник може надсилати спеціально створені завдання компіляції, що містять команди, які «disttc» виконуватиме.

Щоб скористатися цією вразливістю, зловмисники зазвичай виконують такі дії: виявлення, виконання експлойтів.

Розгортаємо лабораторію, яку створили в 3 розділі та починаємо виконання експлойту.

Вікриваємо термінал на Kali Linux та запускаємо Metasploit Framework. За допомогою консольної команди «msfconsole» запускаємо його).

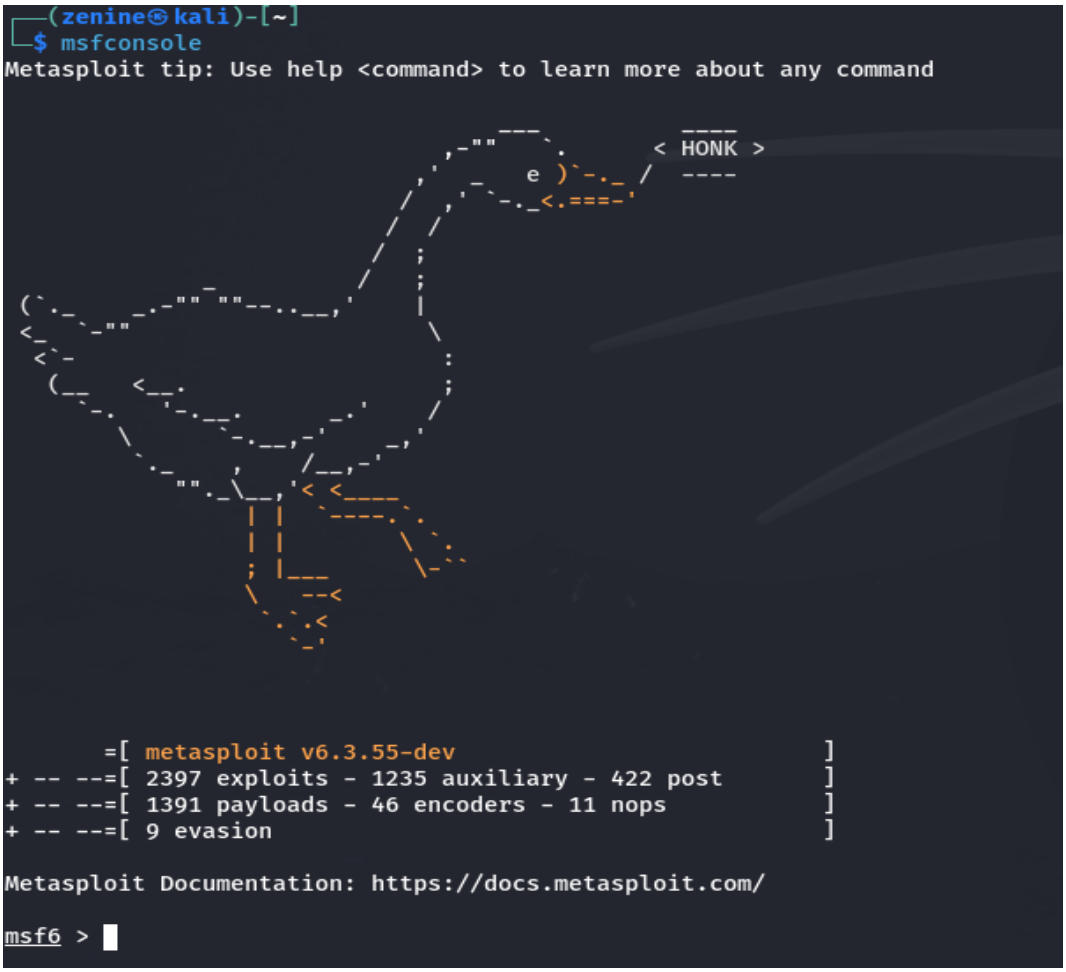

Рис. 4.2. Інтерфейс Metasploit Framework

У консолі Metasploit Framework потрібно виконати наступні команди для налаштування та запуску експлойту. Команди зображені на рисунку 4.3.

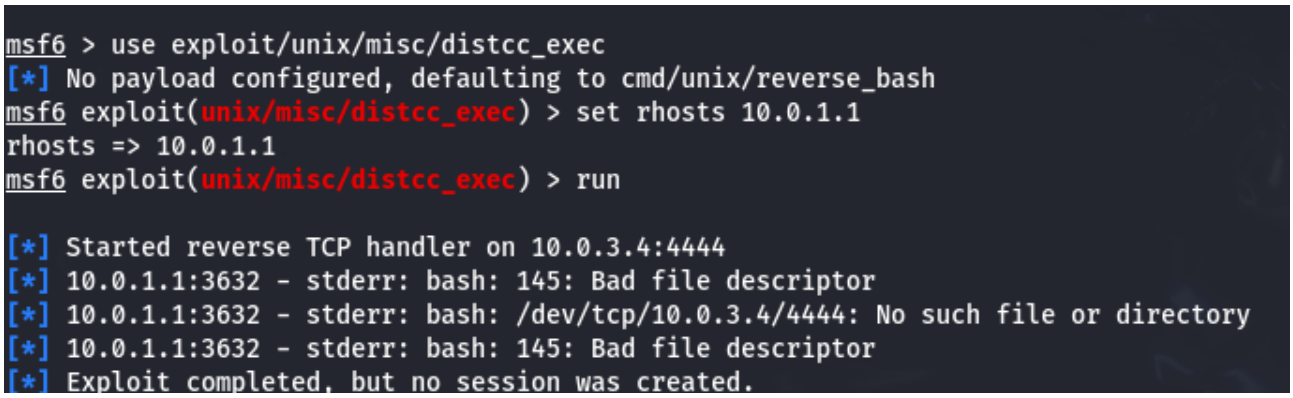

Рис. 4.3. Виконання експлойту

Спочатку за допомогою команди «use ….» находимо сам експлойт. Далі встановлюємо IP-адресу нашого об’єкта атаки, та запускаємо експлойт. Експлойт був заблоковний.

На рисунку 4.4 показано, що система виявлення та реагування успішно виявила та заблокувала експлойт. Це свідчить про ефективність захисних механізмів, впроваджених у цільовій системі. Оскільки експлойт «distcc_exe» був успішно заблокований, варто звернути увагу на інші потенційні вразливості.

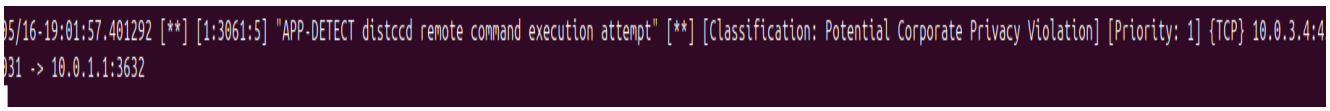

Рис. 4.4. Виявлення експлойта «distcc_exec»

Атака на відмову в обслуговуванні (DoS) спрямована на те, щоб зробити цільову систему або мережу недоступною для користувачів, перевантажуючи її запитами. У цьому розділі проведемо тестування атаки DoS за допомогою інструменту hping3.

Відкриваємо термінал на Kali Linux та запускаємо атаку за допомогою hping3 (рис.4.5).

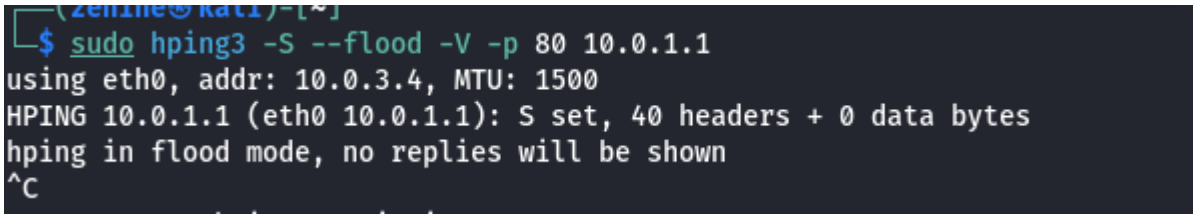

Рис. 4.5. Атака на «Об’єкт атаки»

У цій команді:

- «-S» встановлює SYN прапор TCP-пакета.
- «--flood» вказує hping3 надсилати пакети якнайшвидше.
- «-V» виводить детальну інформацію про атаку.
- «-p» встановлює порт на який буде відбуватися атака.
- «10.0.1.1» – IP-адреса цільової машини.

Результатом буде велика кількість SYN-запитів, які надсилатимуться до «Об’єкта атаки», намагаючись перезавантажити її мережевий стек.

Система виявлення та реагування на основі Snort успішно зафіксувала спробу атаки та заблокувала її. Можемо побачити як працює Snort проти DoS на рисунку 4.6.

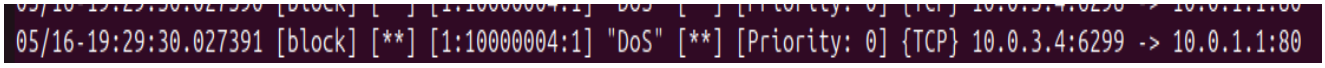

Рис. 4.6. Результат блокування DoS

На рисунку 4.7 можемо побачити статистику «hping3», на якій можемо побачити «100% packet loss». Це значить, що успіху в атаці було не досягнуто. Тестування атаки DoS за допомогою «hping3» продемонструвало ефективність системи виявлення та реагування на кібератаки.

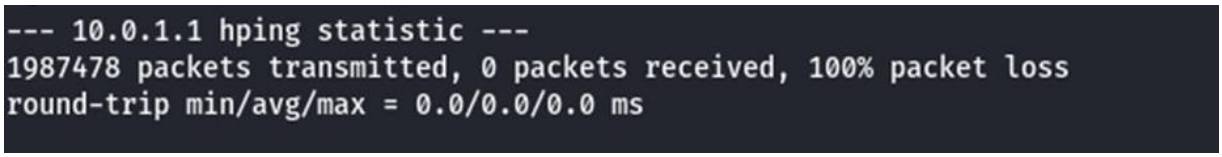

Рис. 4.7. Статистика «hping3»

Однією з критичних вразливостей, яку будемо розглядати, є вразливіть «PHP CGI Argument Injection». PHP CGI Argument Injection - це вразливість безпеки, яка виникає в середовищах CGI (Common Gateway Interface) через неправильну обробку параметрів командного рядка. Це може дозволити зловмисникам виконувати довільні команди на сервері, де запущений PHP через CGI, використовуючи спеціально сформовані HTTP-запити [40].

У консолі Metasploit Framework виконуємо наступні команди для налаштування та запуску експлойту. Команди зображені на рисунку 4.8.

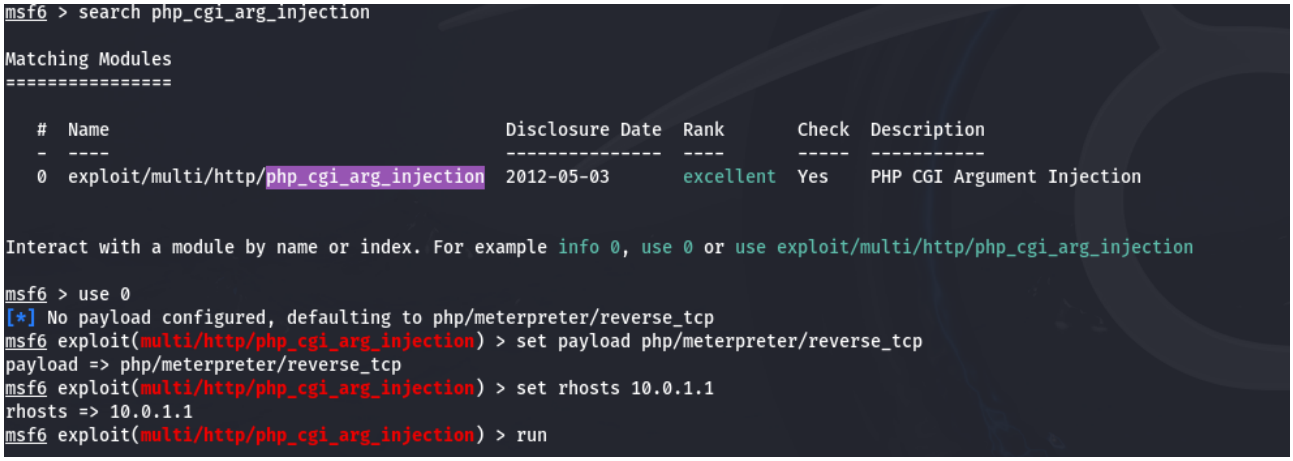

Рис. 4.8. Виконання експлойту «PHP CGI Argument Injection»

Спочатку здійснюємо пошук експлойту за допомогою команди «search …». Після знаходження потрібного експлойту, вибираємо його командою «use 0». Далі, налаштовуємо експлойт за допомогою команд «set payload …» для вибору корисного навантаження, «set rhosts» для встановлення цільової IP-адреси. Після завершення налаштувань, запускаємо експлойт командою «run».

На рисунку 4.9 показано результат блокування «CGI Argument Injection». Система виявлення та реагування на загрози успішно зафіксувала спробу атаки та негайно її заблокувала. Це підтверджує ефективність налаштувань безпеки та вживаних заходів захисту.

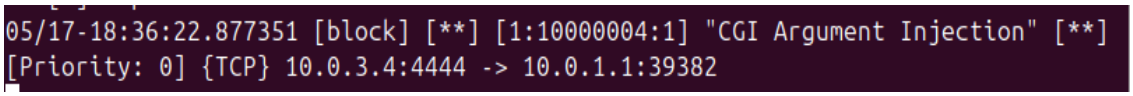

Рис. 4.9. Результат блокування «CGI Argument Injection»

Проведення тестування на проникнення за допомогою «Metasploit Framework» та «hping3» показало високу ефективність захисних механізмів цільової системи. Система виявлення та реагування на кібератаки успішно виявила та заблокувала кілька спроб атак, включаючи «distcc_exec» , «DoS», «PHP CGI Argument Injection». Це підкреслює важливість регулярного тестування на проникнення та оновлення систем безпеки для підтримання високого рівня захищеності інформаційних систем.

4.2 Аналіз результатів тестування

Система виявлення та реагування на кібератаки на основі Snort показала високу ефективність під час проведення тестування на проникнення. Тестування включало різні типи атак, такі як експлойт «distcc_exe», атака DoS за допомогою hping3 та експлойт «PHP CGI Argument Injection». Всі ці атаки були виявлені та заблоковані системою Snort, що свідчить про її надійність та ефективність.

Виявлення і блокування експлойтів та атак є ключовими функціями будь-якої системи безпеки. У даному випадку Snort виявився досить ефективним у виявленні атак, які використовуються для експлуатації вразливостей "distcc_exe" та "PHP CGI Argument Injection". Це свідчить про те, що система відповідає на сучасні загрози і може надійно захищати систему від потенційних атак.

Крім того, система Snort успішно впоралася із атакою DoS, яка була проведена за допомогою інструменту hping3. Заблокування такої атаки є дуже важливою функцією для забезпечення доступності мережевих ресурсів та захисту від перевантаження системи. Таблиця 4.1 відображає результати аналізу трьох типів атак, проведених під час тестування на проникнення.

Таблиця 4.1

Аналіз атак

Етап тестування	**Інструмент**	**Цільова вразливість**	**Результат**
Експлойт «disctcc_exe»	Metasploit Framework	distcc	Заблоковано
Атака «DoS»	Hping3	Відмова в обслуговуванні	Заблоковано
«PHP CGI Argument Injection»	Metasploit Framework	PHP через CGI	Заблоковано

Для кожного типу атаки вказується, чи було виявлено і заблоковано атаку за допомогою системи Snort.

Під час проведення тестування на проникнення важливим аспектом є не тільки факт блокування атак, але й час, необхідний для виявлення та блокування спроб зловмисних дій. У цьому контексті варто розглянути детальніше, як швидко система виявлення та реагування змогла відреагувати на різні типи атак.

При тестуванні експлойту «distcc_exe» за допомогою Metasploit Framework, система виявлення та реагування змогла миттєво ідентифікувати спробу експлуатації вразливості та блокувати її. Час від моменту запуску експлойту до його блокування був практично миттєвим, що свідчить про високу ефективність налаштувань захисту.

Під час атаки на відмову в обслуговуванні (DoS), проведеної за допомогою hping3, система виявлення та реагування Snort також продемонструвала швидку реакцію. Велика кількість SYN-запитів, надісланих до цільової системи, була швидко зафіксована, і захисний механізм встиг заблокувати атаку ще до того, як вона змогла завдати значної шкоди. Втрата 100% пакетів під час атаки свідчить про те, що система реагування діяла в реальному часі, ефективно запобігаючи будь-якій загрозі.

Вразливість «PHP CGI Argument Injection» також була піддана тестуванню за допомогою Metasploit Framework. Як і в попередніх випадках, система

виявлення та реагування змогла швидко визначити спробу експлуатації вразливості та заблокувати її. Час від моменту запуску атаки до її блокування був настільки малим, що зловмисник не мав жодної можливості отримати доступ до цільової системи.

Також проаналізувавши можливості можна дійти до висновку, що система готова масштабуватися. Це може бути досягнуто розширенням інфраструктури виявлення та реагування. Встановлення додаткових сенсорів та вузлів моніторингу в різних сегментах мережі дозволить більш ефективно контролювати трафік.

Інтеграція з іншими системами захисту є важливим кроком у масштабуванні. Інтеграція з системами, що аналізують поведінку користувачів та мережевого трафіку, дозволить виявляти аномалії та потенційні загрози на ранніх етапах. Використання платформ управління подіями та інформаційною безпекою (SIEM) для централізованого збору, аналізу та кореляції даних з різних джерел підвищить ефективність виявлення складних атак.

Система виявлення та реагування на кібератаки на основі Snort успішно справилася з різними видами атак під час проведення тестування на проникнення. Її надійність та ефективність підтверджують її здатність надійно захищати систему від потенційних загроз.

4.3 Оцінка безпеки розробленої системи та пропозиції з практичного застосування

Оцінка безпеки розробленої системи є важливим етапом при впровадженні будь-якого безпекового рішення. Розглянемо плюси та мінуси розподіленої системи безпеки, які наведені в таблиці, і дамо детальну характеристику кожному аспекту.

Ефективне виявлення загроз є одним з основних переваг розподіленої системи безпеки. Такий підхід дозволяє швидко і точно ідентифікувати різноманітні загрози, що значно підвищує загальний рівень захисту мережі.

Завдяки цьому можна своєчасно реагувати на потенційні атаки та запобігати їм, що мінімізує ризики та можливі збитки.

Використання Snort з відкритим кодом забезпечує гнучкість та адаптивність системи. Snort є потужним інструментом, який дозволяє налаштовувати систему під конкретні потреби організації. Це надає можливість легко впроваджувати нові правила і політики безпеки, що робить систему більш адаптивною до змін у зовнішньому середовищі та нових типів загроз.

Інтеграція з іншими системами безпеки є ще однією важливою перевагою. Розподілена система може бути інтегрована з іншими інструментами та платформами безпеки, такими як системи управління інформацією про безпеку та подіями (SIEM). Це дозволяє створити комплексну систему безпеки, яка покриває всі аспекти захисту інформації, забезпечуючи більш ефективне та узгоджене реагування на інциденти.

Підтримка широкого спектру протоколів та плагінів робить систему універсальною та ефективною у різних сценаріях використання. Це дозволяє їй працювати з різними типами мережевого трафіку та додатків, що підвищує її функціональні можливості та забезпечує вищий рівень захисту.

Використання машинного навчання для виявлення аномалій значно підвищує рівень захисту, оскільки система може адаптуватися до нових типів атак та прогнозувати потенційні загрози. Машинне навчання дозволяє виявляти невідомі загрози та аномалії, що не можуть бути виявлені традиційними методами.

Можливість швидкого реагування на загрози та мінімізація потенційної шкоди є ще однією значущою перевагою розподіленої системи. Вона забезпечує швидке реагування на інциденти безпеки, що дозволяє мінімізувати шкоду від атак та зменшити час відновлення.

Автоматизація процесів виявлення та реагування знижує навантаження на IT-персонал, підвищує ефективність та швидкість реагування на інциденти, а також зменшує ймовірність людських помилок. Це дозволяє швидше виявляти та усувати загрози, забезпечуючи безперервний захист мережі.

Проте система безпеки має і свої мінуси. Однією з основних проблем є можливі помилкові спрацьовування, які потребують додаткового налаштування. Це може викликати додаткові труднощі та витрати часу на моніторинг і налаштування системи, що впливає на її ефективність.

Важливою проблемою є також необхідність постійного моніторингу та обслуговування для підтримки актуальності сигнатур та правил. Це вимагає значних ресурсів та зусиль з боку IT-фахівців, що може бути ресурсомістким процесом.

Аналіз великого обсягу мережевого трафіку може вимагати значних обчислювальних ресурсів, що впливає на продуктивність системи. Обробка та аналіз великих обсягів даних можуть спричиняти додаткові навантаження на інфраструктуру, що впливає на загальну ефективність системи.

Алгоритми машинного навчання потребують регулярного навчання на нових даних, що також вимагає значних обчислювальних ресурсів та часу. Це може бути додатковим викликом для організацій, які використовують систему безпеки, оскільки потрібно забезпечити постійне оновлення та підтримку актуальності моделей машинного навчання.

Відсутність захисту від усіх можливих загроз, зокрема тих, що не були враховані під час налаштування системи, є ще однією проблемою. Жодна система не може забезпечити 100% захист від усіх загроз, тому нові та невідомі типи атак можуть залишатися непоміченими, якщо вони не були передбачені у попередніх налаштуваннях.

Можливість потреби додаткового програмного забезпечення та інструментів для повноцінної роботи системи є ще одним мінусом. Для ефективної роботи розподіленої системи безпеки може знадобитися додаткове програмне забезпечення, що підвищує загальні витрати на впровадження та підтримку системи.

Відсутність можливості виявлення абсолютно нових типів атак без попереднього налаштування також є обмеженням системи. Вона може бути

обмежена у виявленні нових типів атак, якщо вони не були передбачені у попередніх налаштуваннях та правилах.

Таким чином, система виявлення та реагування на кібератаки є потужним інструментом для забезпечення кібербезпеки, проте її впровадження та підтримка вимагають значних ресурсів та налаштування. Вона надає високу гнучкість та можливість інтеграції з іншими системами, а також ефективно використовує машинне навчання для виявлення нових загроз. Водночас, необхідність постійного моніторингу та обслуговування, ризик помилкових спрацьовувань і вимоги до обчислювальних ресурсів є вагомими мінусами, які слід враховувати при виборі та налаштуванні такої системи.

В таблиці 4.2 показано плюси та мінуси системи виявлення та реагування на кібератаки.

Таблиця 4.2

Плюси та мінуси розробленої системи

Плюси	**Мінуси**
Ефективне виявлення загроз	Система може мати помилкові спрацьовування, які потребують додаткового налаштування.
Використання Snort з відкритим кодом забезпечує гнучкість та адаптивність	Вимагає постійного моніторингу та обслуговування для підтримки актуальності сигнатур та правил
Можливість інтеграції з іншими системами безпеки	Може потребувати значних ресурсів для аналізу великого обсягу мережевого трафіку
Підтримка широкого спектру протоколів та плагінів	Алгоритми машинного навчання потребують регулярного навчання на нових даних, що може бути ресурсомістким
Використання машиного навчання для виявлення аномалій, що дозволяє виявляти невідомі загрози	Відсутність захисту від усіх можливих загроз, зокрема тих, що не були враховані під час налаштування системи
Можливість швидкого реагування на загрози та мінімізація потенційної шкоди	Може потребувати додаткового програмного забезпечення та інструментів для повноцінної роботи
Автоматизація процесів виявлення та реагування	Відсутність можливості виявлення абсолютно нових типів атак без попереднього налаштування

Разом з тим, не зважаючи на окремі недоліки, практичне застосування розробленої на основі штучного інтелекту системи виявлення та реагування на кібератаки у великих корпоративних мережах дозволить:

- при інтеграції з існуючими системами захисту впровадити систему виявлення та реагування на кібератаки як доповнення до наявних систем захисту інформації (Firewall, IDS/IPS, SIEM), це значно підвищить рівень безпеки, забезпечуючи додатковий шар захисту, заснований на штучному інтелекті;

- для аналіз поведінкових патернів користувачів використовувати систему для постійного моніторингу та аналізу поведінки користувачів в корпоративних мережах, дозволить виявляти аномалії та потенційні загрози, пов'язані з компрометацією облікових записів або внутрішніми загрозами;

- для автоматизованого реагування на інциденти застосовувати систему для автоматичного реагування на виявлені кібератаки, наприклад, блокування підозрілих облікових записів, ізоляція скомпрометованих сегментів мережі або автоматичне оновлення правил доступу для запобігання поширенню атак;

- використовувати можливості штучного інтелекту для прогнозування можливих кібератак на основі історичних даних та поточних трендів. Це дозволить здійснювати проактивні заходи для підвищення рівня захисту корпоративних мереж.

- впроваджувати систему як інструмент для навчання IT-персоналу та підвищення обізнаності про сучасні кіберзагрози, можливості симуляції кібератак і аналізу інцидентів допоможуть персоналу краще розуміти механізми атак і ефективніше реагувати на них.

- використовувати систему для забезпечення відповідності стандартам і регламентам з кібербезпеки, таким як GDPR, ISO/IEC 27001, постійний моніторинг і звітність дозволять ефективно управляти ризиками та забезпечити високий рівень захисту даних.

- застосовувати систему для оптимізації використання мережевих ресурсів, аналіз трафіку та виявлення аномальної активності дозволить краще управляти мережевими ресурсами та вчасно реагувати на потенційні загрози.

- інтеграція з розвідувальними системами для збору даних про нові загрози та вразливості, дозволить системі виявлення та реагування на кібератаки швидко адаптуватися до нових викликів і забезпечити захист від нових типів атак.

- впроваджувати систему як модульне рішення, яке можна легко масштабувати відповідно до зростання мережі, це дозволить забезпечити ефективний захист незалежно від розміру корпоративної мережі.

- постійно вдосконалювати алгоритми та моделі штучного інтелекту на основі нових даних про кіберзагрози, підтримувати високу ефективність системи виявлення та реагування на кібератаки в умовах швидко змінюваного кіберландшафту.

Висновки до розділу 4.

У цьому розділі здійснено оцінку ефективності та безпеки системи за допомогою тестування на реальних даних з використанням Metasploit Framework та інших інструментів. Тестування продемонструвало високу ефективність захисних механізмів системи виявлення та реагування на кібератаки. Система успішно виявила та заблокувала кілька типів атак, включаючи експлойт "distcc_exe", атаку на відмову в обслуговуванні (DoS) за допомогою hping3 та вразливість "PHP CGI Argument Injection".

Таким чином, проведені тести підтвердили, що система виявлення та реагування на кібератаки на основі Snort є надійною та ефективною у захисті від різноманітних загроз. Регулярне тестування на проникнення та оновлення систем безпеки є критично важливими для підтримання високого рівня захищеності інформаційних систем.

Розроблені пропозиції значно покращать ефективність захисту великих корпоративних мереж від кібератак.

ВИСНОВКИ

У кваліфікаційній роботі проведено аналіз та практичне дослідження систем виявлення та реагування на кібератаки в корпоративних мережах.

У першому розділі було детально розглянуто різні види атак та системи виявлення і реагування, зокрема IDS/IPS, EDR та XDR. Порівняння дев'яти систем показало, що Snort відзначається своєю простотою встановлення та ефективністю, що робить його привабливим вибором для побудови системи захисту.

Другий розділ присвячено функціональним та нефункціональним вимогам до системи, а також виборі архітектури, яка дозволяє легко розгортати систему без великих витрат на налаштування. Також розроблено алгоритм для автоматизації виявлення та реагування на загрози за допомогою машинного навчання.

У третьому розділі описано вибір технологій та інструментів для реалізації системи. Створено лабораторію для тестування, в якій Snort було налаштовано та протестовано. Тестування підтвердило ефективність системи в умовах реальних загроз.

Четвертий розділ присвячено оцінці ефективності та безпеки системи через тестування з використанням реальних даних. Результати тестування показали високу ефективність Snort у виявленні та блокуванні різних типів атак, підтверджуючи її надійність та ефективність у захисті корпоративних мереж.

Розроблені пропозиції з практичного застосування розробленої системи значно покращать ефективність захисту великих корпоративних мереж від кібератак

Загалом, дослідження підтвердило, що система виявлення та реагування на основі Snort є дієвим рішенням для захисту від кібератак, а регулярне тестування та оновлення систем безпеки є необхідними для підтримання високого рівня захищеності.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Жураковський Б., Зенів І. Комп'ютерні мережі: навч. Посіб. Київ : КПІ ім. Ігоря Сікорського, 2020. 329 с.

2. Cybersecurity Unveiled: Understanding External and Internal Threats. *With Secure.* URL: https://www.withsecure.com/en/expertise/blog-posts/external-and-internal-threats (date of access: 01.05.2024).

3. Active and Passive attacks in Information Security. *Geeksforgeeks.* URL: https://www.geeksforgeeks.org/active-and-passive-attacks-in-information-security/ (date of access: 01.05.2024).

4. Rajeev S., Mangey R. Distributed Denial of Service Attacks: Concepts, Mathematical and Cryptographic Solutions. Germany: De Gruyter, 2021. 218 p.

5. Network Attacks and Network Security Threats. *Cynet.* URL: https://www.cynet.com/network-attacks/network-attacks-and-network-security-threats/ (date of access: 01.05.2024).

6. Djibril C.D. Man-in-the-Middle Attacks: Defensive Measures. USA: Kindle Edition, 2023. 416 p.

7. What is a Network Attack? Network attacks and network securi-ty threats explained. *Forcepoint.* URL: https://www.forcepoint.com/cyber-edu/network-attack (date of access: 01.05.2024).

8. Vance C. Ethical Hacking Volume 15: SQL Injection : Types, Methods, Tools, Evasion, Countermeasures. USA: Independently published, 2023. 146 p.

9. Деякі питання забезпечення функціонування системи виявлення вразливостей і реагування на кіберінциденти та кібератаки : Поста-нова Каб. Міністрів України від 23.12.2020 р. № 1295 : станом на 7 верес. 2022 р. URL: https://zakon.rada.gov.ua/laws/show/1295-2020-п#Text (дата звернення: 02.05.2024).

10. Threat Detection and Response as a Service: A Comprehensive Primer for Cybersecurity Architects. *CyVent*. URL: https://www.cyvent.com/blog/threat-detection-and-response-as-a-

service#:~:text=TDR%20is%20a%20comprehensive%20approach,Incident%20Response%20(I.R.). (date of access: 02.05.2024).

11. Your Guide to Threat Detection and Response. *Powerdmarc*. URL: https://powerdmarc.com/threat-detection-and-response/ (date of access: 02.05.2024).

12. James S. Extended Detection and Response (XDR). California: Cisco, 2023. 49 p.

13. Praveen S. IDS/IPS : An In-Depth Guide to IDS/IPS (Intrusion Detection System/Intrusion Prevention System): Defending the Digital Fortress. USA: Kindle, 2023. 199 p.

14. What is a Network Intrusion Detection system (NIDS)? *Bunny.net*. URL: https://bunny.net/academy/security/what-is-network-intrusion-detection-nids/ (date of access: 02.05.2024).

15. What is an Intrusion Detection System? *Paloalto*. URL: https://www.paloaltonetworks.com/cyberpedia/what-is-an-intrusion-detection-system-ids (date of access: 02.05.2024).

16. What is an intrusion detection system (IDS)? *IBM*. URL: https://www.ibm.com/topics/intrusion-detection-system (date of access: 02.05.2024).

17. How Can Companies Detect a Potential Cyber Attack. *EC-COUNCIL UNIVERSITY*. URL: https://www.eccu.edu/blog/methods-technologies-detect-cyber-attacks/ (date of access: 02.05.2024).

18. Ashley T. Snort Primer: A FAQ Based Introduction To The Most Popular Open-Source IDS/IPS Program. California: CreateSpace Independent Publish-ing Platform, 2015. 114 p.

19. What is the Difference Between IDS/IPS and NDR? *Stamus Networks*. URL: https://www.stamus-networks.com/blog/what-is-the-difference-between-ids/ips-and-ndr (date of access: 02.05.2024).

20. Про затвердження Методичних рекомендацій щодо реагування суб’єктами забезпечення кібербезпеки на різні види подій у кіберпросторі : Наказ Адміністрації Держспецзв'язку від 03.07.2023 № 570: станом на 3 липня 2023

року URL: https://zakon.rada.gov.ua/rada/show/v0570519-23#Text (дата звернення: 02.05.2024).

21. The Top 8 Threat Detection And Response Solutions. *Expert Insights*. URL: https://expertinsights.com/insights/top-threat-detection-and-response-solutions/ (date of access: 03.05.2024).

22. Що входить до функціональних вимог: ключові складові. *Умови та критерії.* URL: https://trava.zapisi.cx.ua/ukraincyam/shho-vkhodit-do-funkcionalnikh-vimog-klyuchovi-skladovi.html (дата звернення: 04.05.2024).

23. Нефункціональні вимоги: приклади, типи, підходи. *QATestLab*. URL: https://training.qatestlab.com/blog/technical-articles/non-functional-requirements-examples-types-approaches/ (дата звернення: 04.05.2024).

24. Архітектура програмного забезпечення: все що треба знати. *Wezom*. URL:https://wezom.com.ua/ua/blog/arhitektura-programmnogo-obespecheniya (дата звернення: 04.05.2024).

25. OSI Model A Complete Guide - 2021 Edition / ed. By The Art of Service. California: The Art of Service - OSI Model Publishing, 2020. 304 p.

26. Snort-based Smart and Swift Intrusion Detection System. Researchgate. URL: https://expertinsights.com/insights/top-threat-detection-and-response-solutions/ (date of access: 06.05.2024).

27. Сигнатура атаки. *Wkipedia*. URL: https://uk.wikipdia.org/wiki/ (дата звернення: 06.05.2024).

28. Talos launching new machine learning-based exploit detection engine. *Snort Blog.* URL: https://blog.snort.org/2024/03/talos-launching-new-machine-learning.html (date of access: 06.05.2024).

29. Що таке бібліотека в програмуванні? *FoxmindED*. URL: https://foxminded.ua/shcho-take-biblioteka-v-prohramuvanni/ (дата звернення: 07.05.2024).

30. TensorFlow. *Wikipedia*. URL : https://en.wikipedia.org/wiki/TensorFlow (дата звернення: 07.05.2024).

31. About Keras 3. *Keras*. URL: https://keras.io/about/ (date of access: 07.05.2024).

32. Епоха. *Coinmarketcap*. URL: https://coinmarketcap.com/academy/uk/glossary/epoch (date of access: 07.05.2024).

33. Arun K. Oracle VirtualBox Administration: A beginners guide to virtualization! Canada: Independently published, 2019. 188 p.

34. Glen D S. The Ultimate Kali Linux Book - Third Edition: Harness Nmap, Metasploit, Aircrack-ng, and Empire for cutting-edge pentesting. USA: Packt Publishing, 2024. 828 p.

35. Ubuntu. *Wikipedia*. URL: https://uk.wikipedia.org/wiki/Ubuntu (date of access: 08.05.2024).

36. Metasploit Lab. *Offsec*. URL: https://www.offsec.com/metasploit-unleashed/requirements/ (date of access: 08.05.2024).

37. Harry C. VirtualBox: An Ultimate Guide Book on Virtualization with VirtualBox. Scotts Valley: CreateSpace, 2015. 70 p.

38. Gordon F.L. Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning. USA: Nmap Project, 2009. 464 p.

39. Pentesting distcc. *Hacktricks*. URL: https://book.hacktricks.xyz/v/ua/network-services-pentesting/3632-pentesting-distcc (date of access: 09.05.2024).

40. PHP CGI Argument Injection. *Rapid7*. URL: https://www.rapid7.com/db/modules/exploit/multi/http/php_cgi_arg_injection/ (date of access: 09.05.2024).

ДОДАТКИ

Додаток А

Лістинг програми на Python

```
import os
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3'

import numpy as np
import tensorflow as tf
from tensorflow.keras import layers
from urllib.parse import unquote_to_bytes
import matplotlib.pyplot as plt
import json

# Декодування запитів
data = [
    { 'str':'foo=1%27%20OR%201=1%2D%2D', 'attack':1 },
    { 'str':'<script>alert("XSS Attack");</script>', 'attack':1 },
    { 'str':'ping -c 1 example.com', 'attack':1 },
    { 'str':'../../../../../etc/passwd', 'attack':1 },
    { 'str':'<img src="http://malicious-
site.com/transfer?amount=10000&recipient=attacker" onload="this.submit()">',
'attack':1 },
    { 'str':'username=admin&password=\' or \'1\'=\'1', 'attack':1 },
    { 'str':'<img src=x onerror=alert(document.cookie)>', 'attack':1 },
    { 'str':'ls | cat', 'attack':1 },
    { 'str':'../../../../../../../../../../../etc/passwd', 'attack':1 },
    { 'str':'<a href="http://malicious-site.com/attack"
onclick="document.forms[0].submit(); return false;">Click me</a>', 'attack':1 }
]

maxlen = 1024

X = []
Y = []

def decode_query(str):
    return unquote_to_bytes(str.replace('+', ' '))

for item in data:
    arr = decode_query(item['str'])[:maxlen]
    arrlen = len(arr)
    seq = [0] * maxlen
    for i in range(arrlen):
        seq[maxlen - arrlen + i] = arr[i]
    X.append(seq)
    Y.append(item['attack'])

# Створення моделі
model = tf.keras.Sequential([
    layers.Input(shape=(maxlen,), batch_size=1),
    layers.Embedding(256, 32),
    layers.LSTM(16),
    layers.Dense(1, activation='sigmoid')
])
```

```
model.compile(loss='binary_crossentropy', optimizer='adam',
metrics=['accuracy'])

model.summary()

# Тренування моделі та збереження історії тренування
history = model.fit(np.asarray(X).astype(np.float32),
                    np.asarray(Y).astype(np.float32),
                    epochs=100, batch_size=1)

# Збереження моделі
model.save("model.keras")

# Конвертація моделі у формат TFLite
converter = tf.lite.TFLiteConverter.from_saved_model("model.keras")
classifier_model = converter.convert()

# Збереження історії тренування

with open('training_history.json', 'w') as f:
    json.dump(history.history, f)

# Побудова графіків втрат та точності
plt.figure(figsize=(12, 4))

# Графік втрат
plt.subplot(1, 2, 1)
plt.plot(history.history['loss'], label='Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.title('Training Loss')
plt.legend()

# Графік точності
plt.subplot(1, 2, 2)
plt.plot(history.history['accuracy'], label='Accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.title('Training Accuracy')
plt.legend()

plt.tight_layout()
plt.show()
```

Додаток Б

Встановлення віртуальної машини VirtualBox.

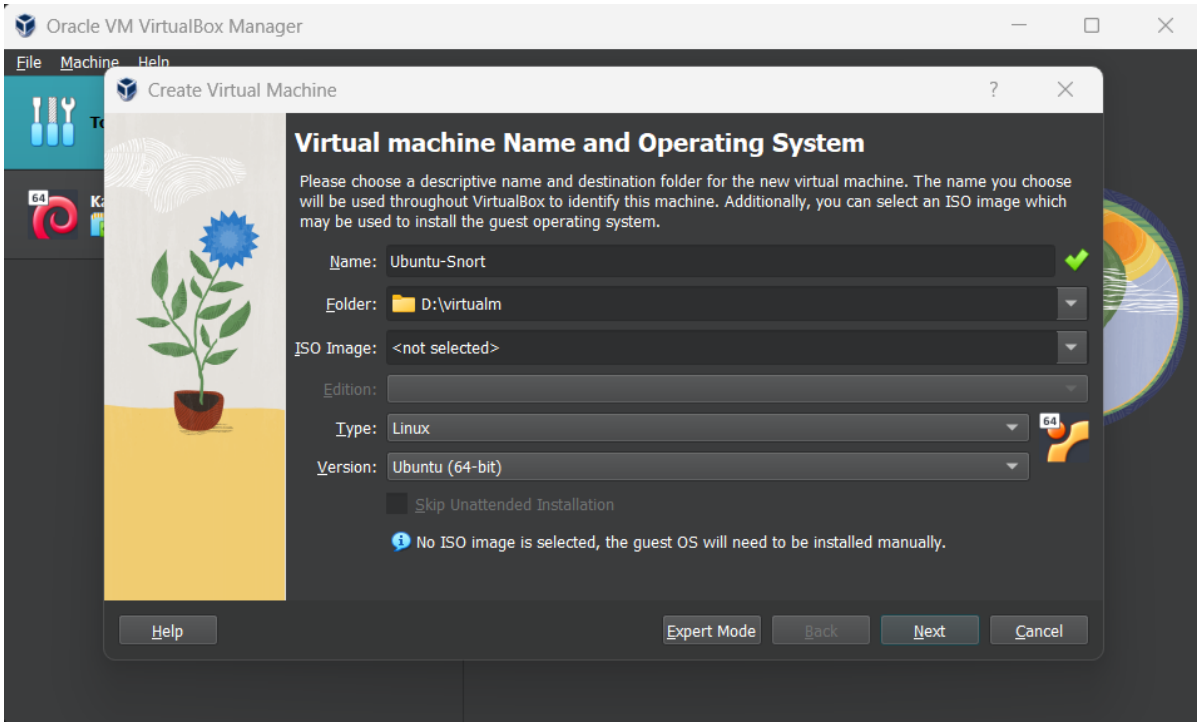

Рис. Б.1. Налаштування мережі в віртуальній машині.

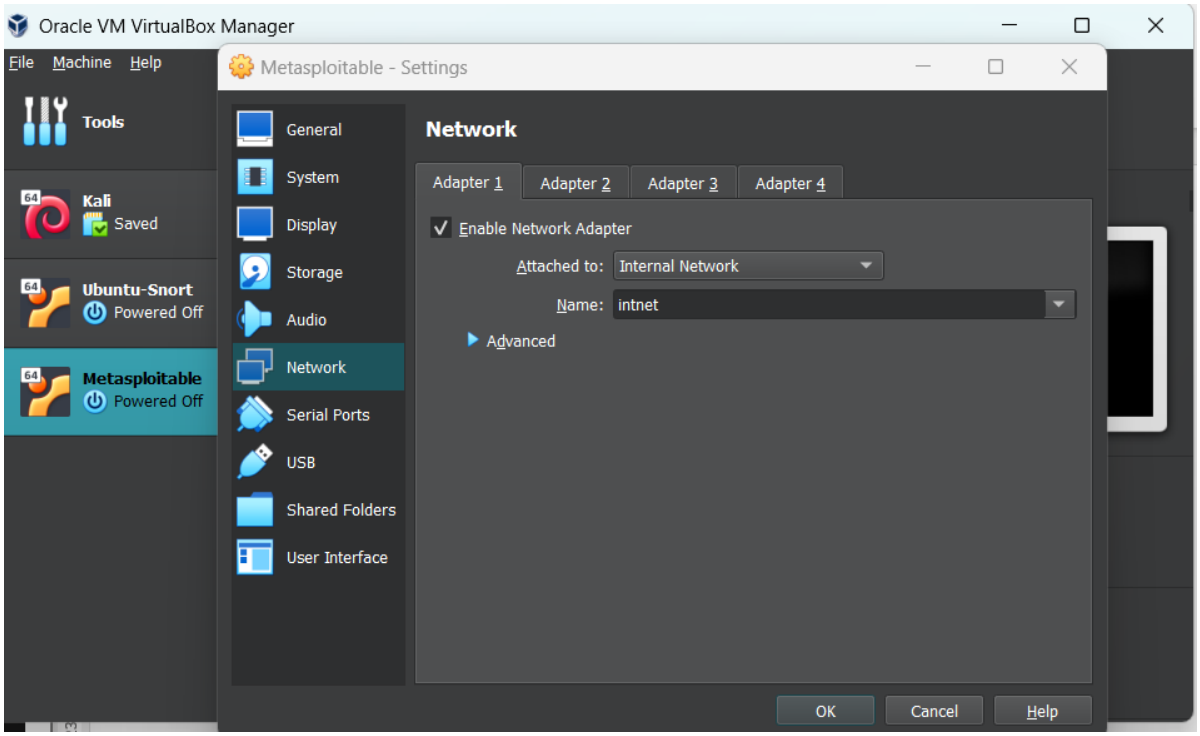

Рис. Б.2. Перевірка успішного налаштування мережі на об’єкті атаки.

Додаток В

Перевірка налаштувань мережі на віртуальних машинах

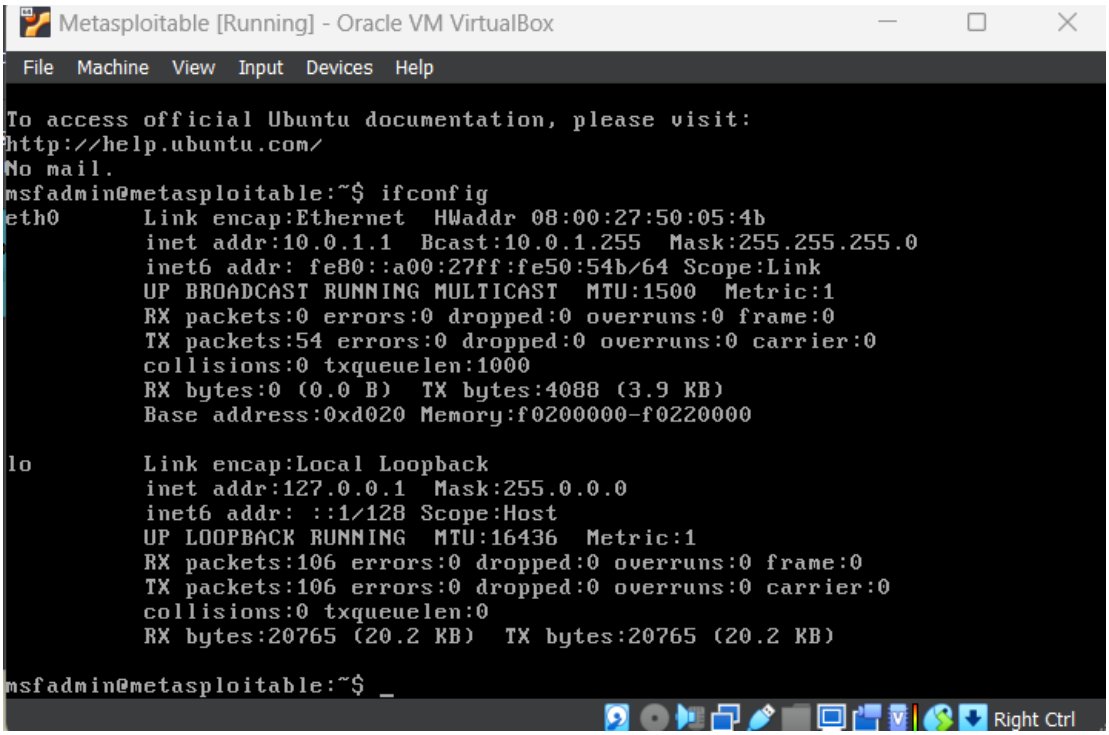

Рис. В.1. Перевірка успішного налаштування мережі на віртуальній машині об’єкта атаки.

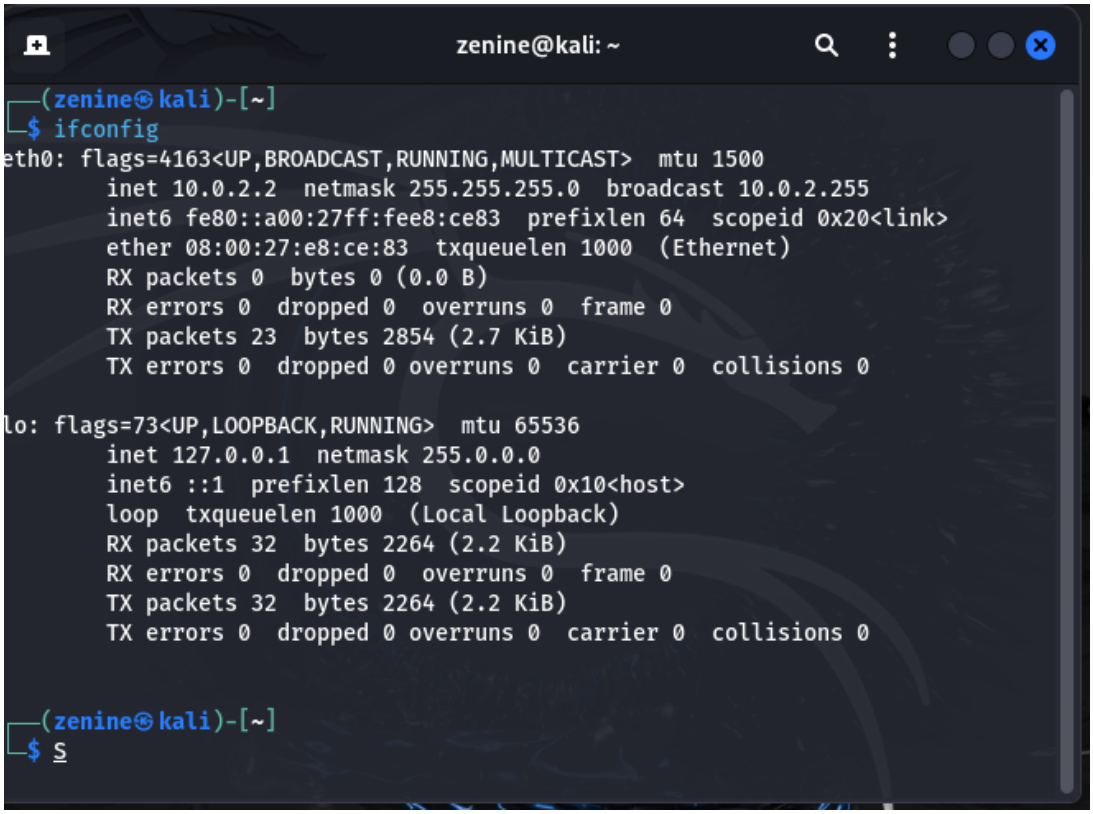

Рис. В.2. Перевірка успішного налаштування мережі на машині зловмисника.

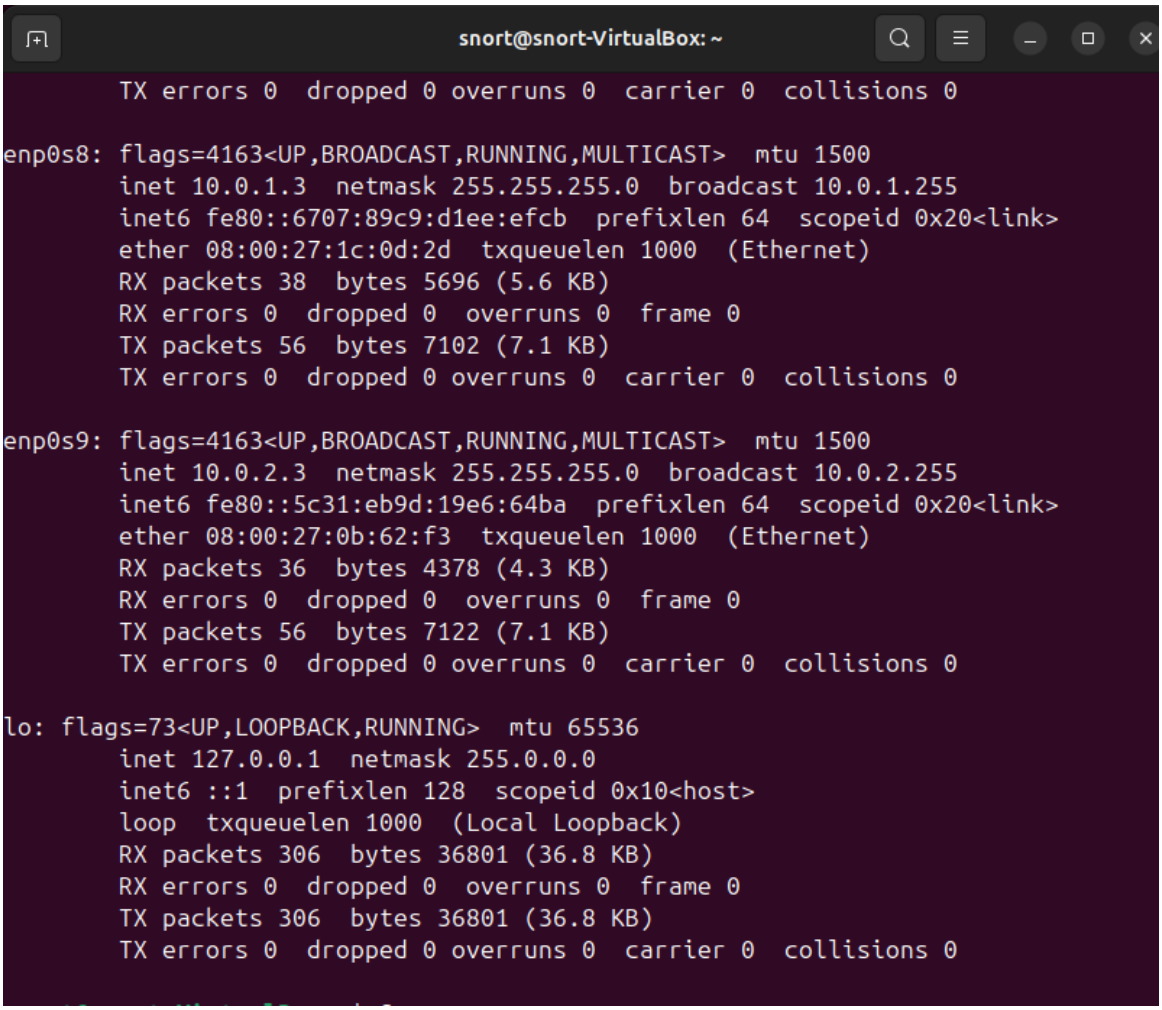

Рис. В.3. Перевірка успішного налаштування мережі на системі виявлення та реагування на кібератаки.

Додаток Г

Процес створення в терміналі папки, розархівування архіву, та перенесення в головну директорію.

```
snort@snort-VirtualBox:~/Downloads$ sudo mkdir /usr/local/etc/rules
snort@snort-VirtualBox:~/Downloads$ tar xz -C snort3-
snort3-3.1.28.0/       snort3-snort3-caf79b9/
snort@snort-VirtualBox:~/Downloads$ tar xz -C snort3-
snort3-3.1.28.0/       snort3-snort3-caf79b9/
snort@snort-VirtualBox:~/Downloads$ ls
3.1.28.0.tar.gz                 snort3-snort3-3.1.84.0-0-ge7312ef.tar.gz
snort3-3.1.28.0                 snort3-snort3-caf79b9
snort3-community-rules.tar.gz
snort@snort-VirtualBox:~/Downloads$ tar xzf snort3-community-rules.tar.gz
snort@snort-VirtualBox:~/Downloads$ ls
3.1.28.0.tar.gz          snort3-community-rules.tar.gz
snort3-3.1.28.0          snort3-snort3-3.1.84.0-0-ge7312ef.tar.gz
snort3-community-rules   snort3-snort3-caf79b9
snort@snort-VirtualBox:~/Downloads$ sudo move
```

Рис. Г.1. Створення папки для правил та розархівування архіву.

```
snort@snort-VirtualBox: ~/Downloads
snort@snort-VirtualBox: ~/Downloads   snort@snort-VirtualBox: ~/Downloads/snort3-...
snort@snort-VirtualBox:~/Downloads$ sudo mv
3.1.28.0.tar.gz
snort3-3.1.28.0/
snort3-community-rules/
snort3-community-rules.tar.gz
snort3-snort3-3.1.84.0-0-ge7312ef.tar.gz
snort3-snort3-caf79b9/
snort@snort-VirtualBox:~/Downloads$ sudo mv snort3-community-rules /usr/local/et
c/rules/
snort@snort-VirtualBox:~/Downloads$ ls -l /usr/local/etc/rules/snort3-community-
rules/
total 2292
-rw-r--r-- 1 snort snort    7834 May  8 17:32 AUTHORS
-rw-r--r-- 1 snort snort   15127 May  8 17:32 LICENSE
-rw-r--r-- 1 snort snort  494963 May  8 17:32 sid-msg.map
-rw-r--r-- 1 snort snort 1799043 May  8 17:32 snort3-community.rules
-rw-r--r-- 1 snort snort   21084 May  8 17:32 VRT-License.txt
snort@snort-VirtualBox:~/Downloads$ S
```

Рис. Г.2. Перенесення правил до головної директорії Snort.