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

КАФЕДРА ІНФОРМАЦІЙНОЇ ТА КІБЕРНЕТИЧНОЇ БЕЗПЕКИ

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Дослідження шляхів та розробка рекомендацій щодо технік виявлення атак на мережеві протоколи»

зі спеціальності *125 Кібербезпека*
(код, найменування спеціальності)

освітньо-професійної програми *Інформаційна та кібернетична безпека*
(назва програми)

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

Виконав: здобувач вищої освіти групи БСД-42
СИДОРЕНКО Максим
(прізвище, ім'я)

Керівник БОЙКО А.О.
(науковий ступінь, вчене звання, прізвище, ім'я)

Рецензент _____
(науковий ступінь, вчене звання, прізвище, ім'я)

Київ 2024

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

Кафедра Інформаційної та кібернетичної безпеки
Ступінь вищої освіти Бакалавр
Спеціальність 125 Кібербезпека
Освітньо-професійна програма Інформаційна та кібернетична безпека

ЗАТВЕРДЖУЮ
Завідувач кафедри ІКБ
_____ Галина ГАЙДУР
"___" _____ 2024 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ

_____
(прізвище, ім'я)

1. Тема кваліфікаційної роботи: «Дослідження шляхів та розробка рекомендацій щодо технік виявлення атак на мережеві протоколи»

керівник кваліфікаційної роботи Бойко А.О.
(прізвище, ім'я, науковий ступінь, вчене звання)
затверджені наказом Державного університету інформаційно-комунікаційних технологій від «___» _____ 2024 року № ____.

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи 31.06.2024 р.

3. Вихідні дані до кваліфікаційної роботи
Загальні знання про мережеві протоколи;
база знань про системи запобігання вторгнень ;
наукова та технічна література, експлуатаційна документація, нормативні документи.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Дослідження проблеми захисту мережевих протоколів.
2. Аналіз методів та засобів захисту від атак на мережеві протоколи на базі Snort.
3. Розроблення рекомендацій щодо застосування методів та засобів захисту мережевих протоколів.

5. Перелік графічного матеріалу

6. Дата видачі завдання 15.04.2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ зп	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Визначення актуальності проблеми виявлення атак на мережеві протоколи.	15.04.2024 р.	
2.	Аналіз наукової та технічної літератури з питань теми кваліфікаційної роботи.		
3.	Аналіз методів та засобів захисту від атак на мережеві протоколи на базі Snort.		
4.	Аналіз ефективності застосування Snort.		
5.	Розроблення рекомендацій щодо застосування методів та засобів захисту мережевих протоколів.		
6.	Оформлення результатів дослідження.		
7.	Підготовка доповіді до захисту.	31.05.2024 р.	

Здобувач вищої освіти _____ Максим СИДОРЕНКО
(підпис) *(ім'я, прізвище)*

Керівник кваліфікаційної роботи _____ Анна БОЙКО
(підпис) *(ім'я, прізвище)*

РЕФЕРАТ

Текстова частина роботи складається з 57 сторінок, 3 рисунків, 2 таблиць, 10 джерел.

Об'єктом дослідження є процес виявлення атак на мережеві протоколи.

Предметом дослідження є методи та засоби виявлення атак на мережеві протоколи

Мета цієї роботи полягає у проведенні дослідження та аналізі існуючих технік виявлення атак на мережеві протоколи з метою їх подальшого вдосконалення та розробки.

Методи дослідження – робота з існуючою базою інформації, її аналіз та порівняння, застосування практичних знань.

У сучасному цифровому світі безпека мереж стає найгострішою проблемою. Атаки на мережеві протоколи постійно зростають у складності та вдосконаленості, що вимагає неперервного вдосконалення методів їх виявлення та захисту. Це дослідження присвячене аналізу шляхів виявлення таких атак та розробці рекомендацій для їх ефективного контролю.

У роботі розглянуто сучасні загрози і властивості мережевих протоколів; визначено ризики та наслідки атак на мережеві протоколи, проаналізовано дані атаки; оглянуто можливості системи Snort; проаналізовано правила та підписи, які використовує Snort; досліджено ефективність застосування Snort; розроблено рекомендації щодо застосування методів та засобів захисту мережевих протоколів.

Галузь використання - інформаційна безпека, телекомунікації, інформаційні технології, кібербезпека організацій

Ключові слова: МЕРЕЖЕВІ ПРОТОКОЛИ, СИСТЕМИ ЗАХИСТУ, СИСТЕМА SNORT, КІБЕРАТАКИ, ВИЯВЛЕННЯ ТА ЗАПОБІГАННЯ ВТОРГНЕНЬ.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ASCII – American standard code for information interchange

BASE – Basic Analysis and Security Engine

CVE – Common Vulnerabilities and Exposures

DAQ – Data AcQuisition

DCE - Dead Code Elimination

DDoS - (distributed) denial-of-service attack

DNS – Domain Name System

DoS – Denial of Service

FTP – File Transfer Protocol

HTTP - HyperText Transfer Protocol

ICMP - Internet Control Message Protocol

IDS – Intrusion Detection System

IMAP – Internet Massage Protocol

IP – Internet Protocol

IPS – Intrusion Prevention System

IPsec – Internet Protocol Security

JSON – JavaScript Object Notation

MitM – Man in the Middle

NetBIOS – Network Basic Input/Output System

NGFW – Next-Generation Firewalls

NIDPS – Network-based Intrusion Detection and Prevention System

NIDS – Network Intrusion Detection System

NOP – No Operation

NVD – National Vulnerabilities Database

OSI – Open System Interconnection

PCAP – Packet Capture

POP – Post Office Protocol

PPP – Point to Point Protocol

RDP – Remote Desktop Protocol

RPC - Remote Procedure Call

SIP - Session Initiation Protocol

SLIP – Serial Line Internet Protocol

SMB – Server Message Block

SMTP - Simple Mail Transfer Protocol

SNMP – Simple Network Management Protocol

SQL – Structured Query Language

SSH – Secure Shell

SSL – Secure Sockets Layer

SSL – Secure Sockets Layer

TCP – Transmission Control Protocol

TLS – Transport Layer Security

UDP – User Datagram Protocol

VLAN – Virtual Local Area Network

VPN – Virtual Private Network

ВСТУП

Актуальність. Тема «Дослідження шляхів та розробка рекомендацій щодо технік виявлення атак на мережеві протоколи» є надзвичайно актуальною в сучасному світі інформаційної безпеки. Інтернет став невід'ємною частиною нашого життя, проте зростання кількості підключених пристроїв також призводить до збільшення загроз для мережевої безпеки.

Атаки на мережеві протоколи можуть мати серйозні наслідки, включаючи виток конфіденційної інформації, порушення функціонування систем та навіть збитки для бізнесу та особистих користувачів. Тому розробка ефективних технік виявлення таких атак є вельми важливою.

Здійснення досліджень у цій області дозволить виявити нові загрози та розвивати методи їх виявлення та запобігання. Це може включати аналіз вразливостей мережевих протоколів, створення інтелектуальних систем виявлення загроз, вдосконалення методів аудиту мережевої безпеки тощо

Звільнення та розвиток нових технологій, таких як Інтернет речей (IoT), обчислювальне хмарне середовище та штучний інтелект, додає складності до ландшафту мережевої безпеки. З кожним новим підключеним пристроєм збільшується потенційна площина атак для зловмисників. Такі атаки можуть бути спрямовані на кіберфізичні системи, енергетичні мережі, медичні пристрої та інші критичні інфраструктурні об'єкти.

Розробка ефективних технік виявлення атак на мережеві протоколи включає в себе не лише технічні аспекти, але й аспекти політики, правозахисту та етику. Наприклад, визначення нормальної поведінки мережі та виявлення аномальних патернів може вимагати використання алгоритмів машинного навчання та аналізу великих обсягів даних.

Завдяки постійному еволюційному характеру загроз мережевої безпеки, дослідники та фахівці з інформаційної безпеки повинні зосередитися на постійному вдосконаленні і розширенні своїх знань та навичок. Крім того, співпраця між галузями, урядовими організаціями, приватним сектором та академічною спільнотою є ключовою для успішного вирішення цих складних проблем.

Таким чином, дослідження шляхів і розробка рекомендацій щодо технік виявлення атак на мережеві протоколи залишається важливою та актуальною темою в сучасному інформаційному суспільстві.

Метою даної роботи є дослідження та аналіз існуючих технік виявлення атак на мережеві протоколи з метою їх подальшої розробки та удосконалення.

Об'єктом дослідження є виявлення атак на мережеві протоколи.

Предметом дослідження є методи дослідження та аналіз існуючих технік виявлення атак на мережеві протоколи з метою їх подальшої розробки та удосконалення.

Основні *завдання* включають в себе:

1. Аналіз існуючих методів виявлення атак на мережеві протоколи: огляд технік, алгоритмів та інструментів, які використовуються для виявлення атак на різноманітні мережеві протоколи.

2. Експериментальне дослідження ефективності методів: перевірка ефективності існуючих методів на реальних атаках або симульованих середовищах для виявлення їхніх переваг та недоліків.

3. Розробка рекомендацій щодо використання та удосконалення методів виявлення атак: на основі аналізу і експериментів розробка практичних рекомендацій з вдосконалення існуючих технік виявлення атак та їхнього застосування у реальних умовах.

Методи дослідження – робота з існуючою базою інформації, її аналіз та порівняння, застосування практичних знань.

Практичне значення: підвищення рівня навчання працівників в компаніях за допомогою стратегії розвитку персоналу, налаштування порогів та алармів, запровадження розвідувальних інструментів

Ця робота спрямована на підвищення рівня безпеки мереж та забезпечення надійного захисту від атак на мережеві протоколи, що є критично важливим для забезпечення безпеки інформації у сучасному цифровому середовищі.

1 ДОСЛІДЖЕННЯ ПРОБЛЕМ ЗАХИСТУ МЕРЕЖЕВИХ ПРОТОКОЛІВ

1.1. Огляд сучасних загроз та вразливостей, що стосуються мережевих протоколів

Уразливості мережевих протоколів можуть виникати з різних причин, таких як недостатня аутентифікація, атаки на переповнення буфера, використання слабких шифрувальних алгоритмів тощо. Для отримання більш детальної інформації про конкретні вразливості рекомендується дослідити базу даних уразливостей, таку як Common Vulnerabilities and Exposures (CVE) та National Vulnerability Database (NVD)

Уразливості з переповнення буфера є поширеними у мережевих протоколах і можуть бути використані для виконання віддаленого коду або відмови в обслуговуванні, якщо зловмисники здатні викликати переповнення буфера, введені даними, що веде до виходу за межі призначеної області пам'яті.

Використання слабких алгоритмів шифрування може зробити мережеві протоколи вразливими до криптографічних атак, які дозволяють зловмисникам розкрити чутливі дані або змінити їх.

Атаки на протоколи маршрутизації можуть призвести до перенаправлення трафіку через зловмисні маршрутизатори, що дозволяє зловмисникам перехоплювати або модифікувати дані, які передаються по мережі.

Протоколи аутентифікації також можуть бути вразливими до атак, які дозволяють атакувачам обійти механізми аутентифікації або викрасти ідентифікаційні дані користувачів.

Використання застарілих версій протоколів може призвести до виникнення вразливостей, які вже виправлені у більш нових версіях, тому важливо регулярно оновлювати програмне забезпечення та використовувати найновіші версії протоколів.

Ці уразливості демонструють важливість постійного вдосконалення безпеки мережевих протоколів для забезпечення захищеності та надійності мережевих інфраструктур.

Існує 2 найпоширеніші сучасні загрози мережевим протоколам:

Атаки типу "відмова в обслуговуванні" (DoS): Ці атаки спрямовані на перевантаження сервера або мережі трафіком, що робить їх недоступними для легальних користувачів.

Атаки типу "людина посередині" (MitM): Ці атаки дозволяють зловмиснику перехоплювати та модифікувати трафік між двома сторонами, які спілкуються.

Атаки на мережеві протоколи є серйозною загрозою для безпеки мереж та даних. Однією з найбільш поширених атак є атаки на переповнення буфера. Вони виникають, коли зловмисники намагаються передати до програми більше даних, ніж може вмістити буфер, що може призвести до перезапису важливої інформації у пам'яті і виконання коду, небажаного для програми. Ця атака може бути використана для відмови в обслуговуванні або виконання віддаленого коду.

Другою типовою атакою є атаки на протоколи маршрутизації. У цих атаках зловмисники намагаються вплинути на спосіб, яким маршрутизатори приймають рішення про передачу даних, щоб перенаправити трафік через свої вузли. Це може призвести до перехоплення або модифікації даних, що проходять через мережу.

Атаки на протоколи автентифікації є ще однією загрозою. У цих атаках зловмисники намагаються обійти механізми аутентифікації, щоб отримати несанкціонований доступ до системи або данних. Це може бути досягнуто шляхом використання слабких паролів, викрадання ідентифікаційних даних або вразливостей у програмному забезпеченні.

Крім того, атаки на мережеві протоколи можуть включати використання застарілих або вразливих версій протоколів, використання слабких алгоритмів шифрування або вразливостей в реалізації протоколів. Ці атаки можуть призвести до розкриття конфіденційної інформації, відмови в обслуговуванні або навіть до виконання віддаленого коду з боку зловмисника.

Усі ці види атак підкреслюють важливість постійного моніторингу та захисту мережевих протоколів для запобігання можливим атакам та забезпечення безпеки мережевої інфраструктури.

Деякі протоколи можуть використовувати слабкі алгоритми шифрування, які можуть бути зламані зловмисниками.

Програмні помилки: програмні помилки в програмному забезпеченні, яке використовує мережеві протоколи, можуть бути використані зловмисниками для отримання несанкціонованого доступу до систем або даних.

Приклади вразливостей мережевих протоколів:

CVE-2021-44228 (Log4Shell): ця вразливість, виявлена в бібліотеці log4j Java, дозволяла зловмисникам віддалено виконувати код на вразливих системах.

CVE-2020-1472 (BlueKeep): ця вразливість, виявлена в протоколі RDP, дозволяла зловмисникам віддалено виконувати код на вразливих системах.

CVE-2017-5753 (WannaCry): ця вразливість, виявлена в протоколі SMB, дозволяла зловмисникам шифрувати дані на вразливих системах і вимагати викупу за їх розшифровку.

Тренди у розвитку загроз і вразливостей в сфері інформаційної безпеки постійно змінюються та еволюціонують, відображаючи технологічний прогрес і нові виклики. Одним з таких трендів є збільшення складності та розміру мережевих інфраструктур, що призводить до збільшення поверхні атак і ризику вразливостей. Зловмисники постійно шукають нові методи інфільтрації великих мереж, використовуючи різноманітні техніки та програмні дірки.

Ще одним трендом є зростання кількості та складності атак з використанням шифрування. Зловмисники все частіше шифрують свій трафік, щоб уникнути виявлення і перехоплення, використовуючи різноманітні методи шифрування, включаючи сучасні алгоритми та протоколи. Це створює додаткові виклики для виявлення та аналізу загроз, оскільки зловмисники стають більш витонченими у своїх підходах.

Також спостерігається зростання використання соціальної інженерії в атаках. Замість традиційних технічних методів вторгнення, зловмисники використовують психологічні та маніпулятивні методи для отримання доступу

до цінної інформації або ресурсів. Це може включати в себе шахрайство по телефону, розсилку фішингових електронних листів або використання соціальних мереж для здійснення атак.

Загалом, тренди у розвитку загроз і вразливостей показують зростаючу складність та різноманітність атак, а також постійну потребу у вдосконаленні захисту та виявленні потенційних ризиків. Важливо постійно вдосконалювати методи захисту інформації, моніторити нові технологічні тенденції та реагувати на них, щоб забезпечити безпеку та захищеність мережевих інфраструктур.

Головною проблематикою є розкриття методів захисту від описаних загроз і вразливостей. Методи захисту від загроз і вразливостей в інформаційній безпеці охоплюють широкий спектр заходів, спрямованих на запобігання атакам та зменшення ризиків. Одним із основних методів є встановлення і використання сучасних систем захисту, таких як брандмауери, системи виявлення вторгнень (IDS) і захисту від вторгнень (IPS). Ці системи допомагають виявляти та блокувати потенційно небезпечний трафік, що намагається проникнути в мережу.

Додатковим методом захисту є використання шифрування, яке допомагає захистити конфіденційні дані під час їх передачі через мережу. Застосування сильних алгоритмів шифрування для захисту даних на рівні транспортного та додаткового рівнів допомагає уникнути перехоплення та розкриття чутливої інформації.

Іншим методом захисту є встановлення правильних політик безпеки, включаючи строгий контроль доступу, регулярне оновлення програмного забезпечення та встановлення сильних паролів. Важливо також надавати регулярну освіту користувачам щодо безпеки інформації, щоб запобігти соціально-інженерним атакам.

Для захисту від вразливостей протоколів, важливо регулярно оновлювати програмне забезпечення та використовувати найновіші версії протоколів, які мають виправлені вразливості. Крім того, проведення аудиту безпеки та пенетраційних тестів може допомогти виявити та виправити потенційні проблеми безпеки до того, як вони будуть використані зловмисниками.

Узагальнюючи, ефективні методи захисту від загроз і вразливостей включають в себе використання сучасних систем захисту, шифрування даних, встановлення правильних політик безпеки та регулярне оновлення програмного забезпечення. Крім того, важливо проводити освіту користувачів та регулярні аудити безпеки для підтримки найвищого рівня захисту мережевих інфраструктур.

Стандарти безпеки мережевих протоколів є важливою складовою захисту інформаційних систем та даних. Ці стандарти визначають набір правил, протоколів і механізмів, які регулюють безпеку передачі, обробки та зберігання даних у мережевому середовищі.

Один з найбільш широко використовуваних стандартів безпеки мережевих протоколів - це стандарт Secure Sockets Layer (SSL) або його наступник Transport Layer Security (TLS). Ці протоколи забезпечують захищену передачу даних через мережу шляхом шифрування ідентифікаційної інформації, що запобігає перехопленню або зміні даних під час їх передачі.

Ще одним важливим стандартом є протокол IPsec, який надає механізми для захищеної передачі даних на рівні мережевого рівня. IPsec дозволяє шифрувати та автентифікувати дані, що передаються між вузлами мережі, забезпечуючи конфіденційність, цілісність та аутентичність.

Для захисту мережевих протоколів також використовуються стандарти аутентифікації, такі як протокол Kerberos або механізми аутентифікації на основі токенів. Ці механізми дозволяють впевнитися в тому, що користувачі мають легальний доступ до мережевих ресурсів і що їх ідентичність підтверджена перед наданням доступу до даних чи сервісів.

Крім того, стандарти безпеки мережевих протоколів також включають в себе встановлення правильних політик безпеки, використання сильних паролів та механізмів контролю доступу, а також регулярне оновлення програмного забезпечення для усунення вразливостей.

Стандарти безпеки мережевих протоколів включають в себе широкий набір технологій, протоколів і механізмів, які спрямовані на захист конфіденційності, цілісності та доступності даних у мережевому середовищі.

1.2. Визначення потенційних ризиків та наслідків атак на мережеві протоколи для організацій

У ході аналізу було виділено декілька потенційних ризиків та проведено їх короткий опис.

Порушення конфіденційності даних:

Атаки, спрямовані на перехоплення чутливої інформації під час передачі через мережеві протоколи, є серйозною загрозою для конфіденційності даних. Під час таких атак зловмисники можуть використовувати різноманітні методи, такі як перехоплення пакетів даних, аналіз трафіку або використання програмного забезпечення для підміни або декодування інформації.

Ця проблема актуальна зокрема через розповсюдження бездротових технологій, які можуть стати джерелом збитку в разі недостатнього захисту мережі. У разі недостатнього шифрування або захисту мережевих протоколів, зловмисники можуть отримати доступ до конфіденційної інформації, такої як паролі, особисті дані, фінансова інформація тощо.

Пошкодження цілісності даних:

Пошкодження цілісності даних у результаті атак на мережеві протоколи є серйозною загрозою, оскільки може призвести до неправильності інформації та вплинути на діяльність організації. Це може включати в себе:

Зміна даних: зловмисники можуть змінювати дані, що передаються через мережу, наприклад, вносячи зміни до фінансових транзакцій або даних користувачів. Це може призвести до серйозних помилок у прийнятті рішень та фінансових втрат для організації.

Вилучення даних: атаки можуть бути спрямовані на вилучення конфіденційної інформації або важливих даних з системи, що може завдати значної шкоди роботі організації та порушити її функціонування.

Псування даних: зловмисники можуть намагатися зламати цілісність даних шляхом внесення помилкових чи випадкових змін, що призведе до неправильного інтерпретації інформації та негативно вплине на прийняття рішень в організації.

Переривання доступності сервісів:

Переривання доступності сервісів в результаті атак на мережеві протоколи є серйозною загрозою для організацій, оскільки може призвести до неможливості користувачів отримати доступ до необхідних послуг та сервісів. Це може мати наступні наслідки:

Втрати фінансових ресурсів: недоступність важливих сервісів може призвести до втрати доходу для організації, особливо у випадку, коли ці сервіси є ключовими для її функціонування.

Втрата клієнтів і клієнтської довіри: переривання доступності сервісів може призвести до розчарування користувачів та втрати їх довіри до організації. Це може призвести до втрати клієнтів на користь конкурентів і погіршення репутації бренду.

Шкоди для бізнесу та іміджу: недоступність сервісів може вразити репутацію організації і вплинути на її співробітників, партнерів та інвесторів.

Експлуатація вразливостей:

Зловмисники можуть використовувати вразливості в мережевих протоколах для здійснення атак на інфраструктуру організації, отримання несанкціонованого доступу та викрадення даних. Атаки на інфраструктуру: З використанням вразливостей у мережевих протоколах зловмисники можуть здійснювати атаки на інфраструктуру організації, такі як внедрення шкідливого програмного забезпечення, зміна налаштувань мережі або перехоплення даних.

Несанкціонований доступ: вразливості в мережевих протоколах можуть дозволити зловмисникам отримати несанкціонований доступ до систем та ресурсів організації, що може призвести до викрадення конфіденційної інформації або порушення її цілісності.

Викрадення даних: з використанням вразливостей у мережевих протоколах зловмисники можуть отримати доступ до конфіденційної інформації організації, включаючи особисті дані користувачів, фінансові дані та комерційну інформацію.

Вплив на ділову діяльність:

Атаки на мережеві протоколи можуть призвести до припинення або обмеження нормальної діяльності організації, що може мати серйозний вплив на її фінансовий стан та взаємовідносини з клієнтами та партнерами. Припинення діяльності: Серйозні атаки на мережеві протоколи можуть призвести до припинення роботи систем і послуг організації. Це може мати серйозні наслідки для бізнесу, зокрема, втрату прибутку та репутаційний збиток.

Обмеження доступу: іноді атаки можуть обмежити або заблокувати доступ до важливих ресурсів для співробітників та клієнтів. Це може призвести до перерв у роботі, затримок у виконанні завдань та загального незадоволення користувачів.

Фінансові втрати: припинення роботи і обмеження доступу можуть призвести до прямих фінансових втрат через втрату прибутку та накладення витрат на відновлення систем та послуг.

Порушення взаємовідносин: негативний вплив на функціонування організації може також мати наслідки для її взаємовідносин з клієнтами, партнерами та іншими зацікавленими сторонами. Це може призвести до втрати довіри та погіршення репутації.

Ці ризики та наслідки підкреслюють необхідність ефективного захисту мережевих протоколів для забезпечення безпеки та стабільності діяльності організацій.

1.3. Аналіз типових атак на мережеві протоколи (перехоплення пакетів, витік конфіденційної інформації, атаки типу «міжмережевий перехід» тощо)

Нами було взято 5 типових атак, та проведено їх опис. Опис представлено далі в тексті.

Перехоплення пакетів (Packet Sniffing):

Зловмисники використовують програмне забезпечення для перехоплення і аналізу мережевого трафіку. Це дозволяє їм переглядати надіслані через мережу

дані, включаючи конфіденційну інформацію, таку як паролі, особисті дані та фінансова інформація. За допомогою спеціалізованого програмного забезпечення, зловмисники можуть отримати доступ до мережевого трафіку і аналізувати його з метою перегляду та викрадення конфіденційної інформації. Це може стати вихідним пунктом для різноманітних інших атак, таких як використання знайдених паролів або особистих даних для несанкціонованого доступу до систем або послуг.

Програми для перехоплення пакетів можуть працювати на різних рівнях мережевого стека, включаючи фізичний рівень (наприклад, захоплення данних з мережевого кабелю), рівень каналу (наприклад, перехоплення Wi-Fi трафіку), рівень мережі (наприклад, захоплення IP-пакетів) та рівень транспортного з'єднання (наприклад, перехоплення TCP/IP пакетів).

Зловмисники можуть використовувати перехоплення пакетів для виявлення вразливостей в мережевій інфраструктурі, викрадення обмінюваних між системами даних, в тому числі інформації про автентифікацію, а також для аналізу загальної структури мережі та виявлення слабких місць в захисті.

Витік конфіденційної інформації (Information Leakage):

Помилки в програмному забезпеченні або налаштуваннях мережі можуть призвести до некоректного оброблення конфіденційної інформації, що призводить до її витоку. Це може стати результатом недостатньої аутентифікації, авторизації або захисту даних. Це може стати результатом різноманітних помилок у програмному забезпеченні або налаштуваннях мережі, які призводять до некоректного оброблення конфіденційної інформації та її витоку. Основні причини витоку інформації включають:

Недостатня аутентифікація та авторизація: помилки в системі аутентифікації та авторизації можуть дозволити несанкціонованим користувачам отримати доступ до конфіденційної інформації. Наприклад, слабкі паролі, використання стандартних або залишених значень аутентифікаційних параметрів можуть стати вихідними пунктами для атак на систему.

Некоректна обробка даних: помилки у програмному забезпеченні, такі як недоліки в коді або недостатня перевірка введених даних, можуть призвести до витоку конфіденційної інформації через несподівані вразливості системи.

Неправильні налаштування мережі: невірні налаштування мережевого обладнання, файерволів, маршрутизаторів та інших пристроїв можуть призвести до ненавмисного розкриття конфіденційної інформації. Наприклад, невірно налаштований файервол може дозволити несанкціонований доступ до внутрішніх мережевих ресурсів.

Недостатній захист даних: недоліки в системах шифрування, недостатній захист даних в пам'яті або на дисках, а також недостатнє використання методів захисту можуть сприяти витоку конфіденційної інформації.

Атаки типу "міжмережевий перехід" (Man-in-the-Middle Attacks):

Зловмисники вставляються між двома комунікуючими сторонами і перехоплюють та модифікують передавану між ними інформацію. Це може дозволити їм отримати доступ до конфіденційних даних або навіть виконувати дії в ім'я обох сторін, несприятливо впливаючи на їх взаємодію. В цьому виді атак зловмисники встановлюють контроль над комунікацією між двома легітимними сторонами і перехоплюють або модифікують передавані дані. Це дає їм можливість отримати доступ до конфіденційних даних, таких як паролі, особисті дані або фінансова інформація, а також виконувати дії в ім'я обох сторін, несприятливо впливаючи на їх взаємодію.

Процес атаки "міжмережевий перехід" може виглядати наступним чином:

Перехоплення комунікації: зловмисники встановлюють з'єднання з обома сторонами і починають перехоплювати передавану між ними інформацію. Це може бути зроблено за допомогою різноманітних технік, таких як перехоплення бездротового сигналу або використання шпигунського програмного забезпечення на комп'ютері жертви.

Модифікація даних: після того, як зловмисники перехопили комунікацію, вони можуть модифікувати дані, що передаються між сторонами. Наприклад, вони можуть змінити адресу доставки фінансового переказу або вставити шкідливий код у веб-сторінку.

Подальша передача: після модифікації даних зловмисники передають їх до легітимного отримувача, що робить їхню дію більш прихованою.

Атаки з використанням вразливостей протоколів (Protocol Vulnerability Exploitation):

Зловмисники можуть використовувати вразливості в реалізації мережевих протоколів для здійснення різноманітних атак, включаючи переповнення буфера, виконання коду та інші. Атаки, які використовують вразливості протоколів, є серйозною загрозою для мережевої безпеки. Зловмисники можуть ефективно використовувати недоліки в реалізації мережевих протоколів для впровадження різноманітних видів атак, включаючи переповнення буфера, виконання коду та інші методи, що дозволяють їм отримувати несанкціонований доступ до системи або виконувати шкідливі дії.

Атаки з відмовою в обслуговуванні (Denial-of-Service, DoS):

Зловмисники можуть спробувати перевантажити мережеві протоколи або ресурси організації, що призведе до перерв у роботі сервісів або навіть їх припинення. Це спроби зміни доступності ресурсів для легітимних користувачів шляхом перевантаження чи недоступності сервісів, системи або мережі. Ці атаки можуть включати в себе надмірне використання ресурсів, переповнення каналу зв'язку або використання вразливостей програмного забезпечення для зупинки, відмови в обробці запитів або відмови в обробці сервісів. Під час атак DoS зловмисники можуть використовувати широкий спектр методів, таких як відправлення великої кількості запитів до цільового сервера, злам системи захисту або використання.

1.4. Аналіз існуючих рішень із захисту мережевих протоколів

Захист мережевих протоколів - це важлива складова безпеки інформаційних систем. Існують різні підходи та рішення для захисту мережевих протоколів. Проаналізовано деякі з них.

Firewalls. Вони фільтрують трафік між мережею та Інтернетом на основі заданих правил. Firewalls можуть блокувати або дозволяти трафік на основі IP-

адрес, портів, протоколів тощо. Додатково можуть мати функціонал для виявлення вторгнень (IDS/IPS).

Packet Filtering Firewalls: Цей тип firewall аналізує заголовки пакетів даних і вирішує, чи слід пропускати пакет. Вони можуть блокувати або дозволяти трафік на основі IP-адрес, портів та протоколів.

Stateful Inspection Firewalls: Вони використовують стан підключення для аналізу трафіку, що проходить через них. Такі firewalls можуть дозволити трафіку, який відповідає вже існуючим з'єднанням, або який ініційований внутрішніми користувачами.

Proxy Firewalls: Вони дозволяють більш детальний контроль трафіку, оскільки вони перенаправляють всі запити через проксі-сервер, який вирішує, чи слід дозволити чи заборонити трафік.

Next-Generation Firewalls (NGFW): Ці firewalls використовують комбінацію методів фільтрації, виявлення вторгнень та інших заходів безпеки для захисту мережі. Вони можуть виявляти й блокувати небезпечні додатки, контролювати доступ на основі ідентифікації користувачів і багато іншого.

*Плюси:з*абезпечують базовий рівень захисту мережі; доступність і широке застосування; можливість налаштування правил для контролю трафіку.

Мінуси: можуть бути обхідні в деяких випадках, наприклад, атаками, які використовують вразливості протоколів; можливість ложнопозитивних спрацювань, особливо у випадку складних правил; не ефективні проти атак внутрішньої загрози.

Інтригування трафіку (Traffic Intrusion Detection). Використовуючи інтригування трафіку, системи можуть аналізувати пакети даних, що проходять через мережу, для виявлення підозрілого чи шкідливого змісту. Інтригування трафіку може виявляти атаки, такі як SQL-ін'єкції, кросс-сайтові скрипти, атаки DoS і т. д.

Воно поділяється на сигнатурне виявлення вторгнень (Signature-based IDS/IPS) - цей метод порівнює трафік із відомими сигнатурами відомих атак; поведінкове виявлення вторгнень (Behavior-based IDS/IPS) - аналізується поведінка системи для виявлення аномальних дій, що можуть вказувати на атаку.

Плюси: дозволяє виявляти навіть ранні стадії атак або нові види загроз; можливість виявлення навіть обхідних методів атак.

Мінуси: можливість великої кількості ложнопозитивів; вимагає значних обчислювальних ресурсів для аналізу трафіку; може не виявити атаки, які не залишають слідів в трафіку.

Шифрування трафіку (Traffic Encryption). Шифрування трафіку за допомогою протоколів, таких як SSL/TLS (використовується для зашифрування комунікації між клієнтами та серверами, забезпечуючи конфіденційність та цілісність даних), може захистити дані від перехоплення та зміни. Важливо застосовувати шифрування, особливо при передачі конфіденційної інформації через неприватні мережі, такі як Інтернет.

Плюси: забезпечує конфіденційність та цілісність даних; ефективний спосіб захисту даних, що передаються по ненадійній мережі.

Мінуси: може вплинути на швидкість передачі даних; додаткове навантаження на обладнання для шифрування та розшифрування даних; можливість використання шифрованого трафіку для хования шкідливих дій

Аутентифікація та авторизація (Authentication and Authorization). Використання механізмів аутентифікації (перевірка ідентичності користувача) і авторизації (контроль доступу до ресурсів) дозволяє обмежити доступ до мережевих ресурсів лише авторизованим користувачам.

Методи аутентифікації включають паролі, біометричну ідентифікацію, двофакторну аутентифікацію тощо. Методи авторизації визначають, які ресурси або функціональні можливості має дозволено користувач.

Плюси: забезпечують захист від несанкціонованого доступу до мережевих ресурсів; дозволяють гнучко налаштовувати права доступу для користувачів.

Мінуси: паролі або інші методи аутентифікації можуть бути скомпрометовані; вимагають управління ідентифікацією та авторизацією, що може бути складним у великих мережах.

Використання VPN (Virtual Private Networks). VPN створюють зашифрований тунель між двома точками у мережі, що дозволяє безпечно

передавати дані через ненадійну мережу, таку як Інтернет. VPN дозволяють забезпечити конфіденційність, цілісність та аутентифікацію даних.

Шифрування трафіку - всі дані, що передаються через VPN, зашифровані, що забезпечує конфіденційність. Тунельний протокол - VPN використовують протоколи, такі як IPSec, для створення безпечного тунелю між точками.

Плюси: забезпечують безпеку та конфіденційність даних, переданих через ненадійні мережі; дозволяють розширити мережу без необхідності фізичного розміщення нових мережевих елементів.

Мінуси: потребують додаткових ресурсів для налаштування та управління; можуть впливати на швидкість та продуктивність мережі через додаткове шифрування трафіку.

Блокування небезпечних протоколів. Деякі організації використовують методи блокування конкретних мережевих протоколів, які вони вважають небезпечними або непотрібними для внутрішньої мережі. Відомі протоколи, які можуть бути використані для атак (наприклад, NetBIOS), можуть бути блоковані на рівні мережевого обладнання.

Плюси: зменшує ризик атак через використання небезпечних протоколів.

Мінуси: може вплинути на функціональність деяких додатків або сервісів; не всі атаки базуються на використанні небезпечних протоколів.

Обмеження доступу до мережевих ресурсів. Це включає в себе використання технологій, таких як VLANs, щоб фізично або логічно розділити мережу і обмежити доступ до чутливих ресурсів.

Плюси: зменшує ризик несанкціонованого доступу до конфіденційної інформації; дозволяє гнучко налаштовувати права доступу для різних груп користувачів.

Мінуси: потребує вивчення і налаштування відповідних мережевих налаштувань, що може бути складним; може збільшити складність управління мережею.

Понадшарове виявлення загроз (Next-Generation Threat Detection). Використання алгоритмів машинного навчання і поведінкового аналізу для

виявлення нових та складних загроз, які можуть ухилятися від традиційних методів захисту.

Плюси: здатні виявляти нові та складні загрози, які можуть ухилятися від традиційних методів захисту; забезпечують більш високий рівень безпеки шляхом аналізу не тільки трафіку, але і поведінки системи.

Мінуси: вимагають значних ресурсів для реалізації, в тому числі ресурсів обчислювальної та мережевої інфраструктури; можуть мати велику кількість ложнопозитивних або ложно-негативних результатів.

Кожен з цих підходів може використовуватися окремо або в поєднанні з іншими для максимального захисту мережевих протоколів.

2 АНАЛІЗ МЕТОДІВ ТА ЗАСОБІВ ЗАХИСТУ ВІД АТАК НА МЕРЕЖЕВІ ПРОТОКОЛИ НА БАЗІ SNORT

2.1. Огляд функціональності та можливостей системи виявлення вторгнень (IDS) та системи запобігання вторгнень (IPS) SNORT

Snort — це класичний NIDS (Network Intrusion Detection System), який вирізняється своєю довголітньою історією та активним розвитком. Почавши свій шлях у 1998 році як проєкт з відкритим кодом, ця система стала справжнім символом незалежності та ефективності в області мережевої безпеки.

Придбана компанією Cisco в 2008 році, Snort зберіг свою унікальну ідентичність, ставши частиною великої екосистеми безпеки, але залишаючись доступним та досить привабливим для маленьких та середніх компаній.

Його унікальність полягає в тому, що Snort поєднує в собі сніфер пакетів і можливість налаштування правил, а також надає користувачам широкі можливості для контролю та аналізу мережевого трафіку. Це інструмент, який пропонує зрозуміле та функціональне рішення для тих, хто займається запобіганням вторгнень у мережі. [1].

Snort може реєструвати події, аналізувати їх та проводити пошук за вмістом. Використовується для активного блокування або пасивного виявлення різноманітних атак та зондувань. Система може розпізнавати атаки, які приховуються в шумі, коли зловмисник надсилає велику кількість підозрілих пакетів на цільову систему, переповнюючи її та приховуючи свою атаку.

Хоча IDS Snort не може гарантувати 100% виявлення всіх підозрілих подій, на сьогоднішній день жодна система не має таких можливостей. Теоретично Snort можна обійти декількома способами. Наприклад, застосування експлоітів. Хоча Snort може реагувати на підозрілий код, аналізуючи NOP, цей код можна легко модифікувати, щоб шкідливі пакети залишались непоміченими для IDS. [1].

Snort функціонує за наступною схемою. При отриманні пакету, він послідовно проходить через декодери та препроцесори, а тільки потім потрапляє до детектора, де застосовуються правила виявлення загроз [1].

Декодери відповідають за те, щоб ізвлекти дані мережного і транспортного рівнів (IP, TCP, UDP) з протоколів канального рівня (Ethernet, 802.11, Token Ring…). Препроцесори, з свого боку, готують дані з протоколів транспортного та мережевого рівнів до застосування правил. Наприклад, препроцесор TCP виконує такі завдання:

- Контроль стану (перевірка дотримання протоколу);
- Складання сеансу (об'єднання даних з декількох пакетів сеансу);
- Нормалізація протоколу. [1].

Правильна настройка препроцесорів може значно підвищити продуктивність системи та зменшити кількість надлишкових даних, що потрапляють у детектор. Крім того, завдяки особливостям архітектури Snort, досить легко підключити власний препроцесор. Це призводить до формування "надпакетів", до яких застосовуються правила перед відправкою в детектор.

Процес застосування правил полягає у пошуку в "надпакеті" сигнатур, визначених у правилах. Самі правила містять опис трафіку, сигнатури, опис загрози та опис реакції на виявлення [1].

Популярність та широке поширення Snort зумовлені його відкритим вихідним кодом. Це одна з найдавніших систем IDS/IPS у світі, розроблена Мартіном Рошем у 1998 році. Рош є видатним фахівцем з інформаційної безпеки та автором багатьох книг з цієї галузі. Створення Snort було відповіддю на відсутність на той час ефективних та безкоштовних систем, що могли б попереджати мережеві атаки [2].

Дана IPS виявляє наступне:

- Підозрілий трафік
- Використання експлойтів (виявлення Shellcode)
- Сканування системи (порти, ОС, користувачі і т.д.)
- Атаки на різні служби (Telnet, FTP, DNS, і т.д.)

● DoS / DdoS атаки

● Web-атаки на сервера (cgi, php, frontpage, iss і т.д.)

● Ін’єкції SQL, Oracle і т.д.

● Атаки по протоколам SNMP, NetBios, ICMP

● Атаки за протоколами SMTP, IMAP, pop2, pop3

● Різні Backdoors

● Web-фільтри

● віруси

Крім усього іншого Snort має:

● Можливість написання власних правил

● Розширення функціональності, використовуючи можливість підключення модулів

● Гнучку систему оповіщення про атаки (Log файли, пристрої виведення, БД І.Д.)

Snort підтримує такі інтерфейси для прослуховування:

● Ethernet

● SLIP

● PPP

Причиною такого поширення Snort є його кросплатформеність, яка дозволяє встановлювати програму як на популярних, так і на менш відомих операційних системах. Функціонал Snort можна значно розширити завдяки розширенню - inline, яке дозволяє прив’язати firewall до правил. Часто таке з'єднання дозволяє виявляти спроби DDoS атак ще на їх початковій стадії та приймати відповідні заходи. Проте, цей метод захисту має свої недоліки: якщо хакер знає, що трафік буде блокуватися, він може підробити пакети так, ніби вони надходять з важливих серверів [3].

Розглянемо архітектуру Snort.

В основі системи Snort лежить двигун, який в свою чергу складається з декількох модулів:

● Сніфер пакетів відіграє важливу роль у перехопленні даних, які передаються через мережу, та їх подальшій передачі на декодер. Ці функції

реалізовані за допомогою бібліотеки DAQ (Data AcQuisition). Сніфер пакетів має два режими роботи: пасивний, коли він перехоплює дані мережі в реальному часі, та режим читання, коли дані читаються з попередньо підготовленого файлу [4].

● Декодер пакетів виконує кілька важливих функцій, серед яких сортування заголовків перехоплених пакетів, пошук відхилень і аномалій, аналіз TCP-прапорів та перевірка певних протоколів і їх роботи. Цей модуль працює з протоколами стеку TCP/IP.

У відміну від декодера, який працює з пакетами, переданими на 2-3 рівнях моделі OSI, модуль препроцесора здійснює більш детальний аналіз і формалізацію пакетів, збираючи дані з 3, 4 і 7 рівнів. Серед найпоширеніших препроцесорів є: frag3 - для роботи з фрагментованим трафіком, http_inspect - для нормалізації HTTP-трафіку, sfPortscan - для виявлення підозрілих дій з портами, DCE/RPC2. Також існує велика кількість декодерів для інших протоколів, таких як IMAP, Telnet, SMTP, SIP, SSL, FTP, SSH і т.д. У деяких випадках системні адміністратори пишуть власні препроцесори, які зазвичай спрямовані на конкретні потреби [4].

● Двигун, що відповідає за виявлення атак, має двокомпонентну структуру. Оскільки Snort працює за правилами, то є конструктор правил, який відповідає за збір усіх доступних правил і сигнатур атак та їх компіляцію в цільний набір даних. Цей набір максимально оптимізований для використання системою під час пошуку порушень чи аномалій.

● Движок виявлення атак має дві складові. Конструктор правил збирає безліч різних важливих правил у єдиний набір. Цей набір оптимізований для подальшого використання підсистемою інспекції захопленого та обробленого трафіку у пошуках порушень.

● Модуль виведення в системі Snort відповідає за формування результату перевірки. Після проведення атаки система може згенерувати відповідне повідомлення або записати інформацію про атаку у спеціальний файл. Snort працює з різними форматами файлів, такими як syslog, ASCII, PCAP, Unified2. [4].

2.2. Вивчення правил та підписів, які використовуються Snort для виявлення атак на мережеві протоколи

Заголовок складається з певних інструкцій, до яких відносяться:

- Журнал або сповіщення
- Тип мережевого пакету: TCP/UDP/ICMP/тощо
- IP-адреса та порти джерела і призначення
- Сповіщення та, можливо, деякі кваліфікатори
- Сигнатури та тип класифікації (таблиця 2.1) [5].

Таблиця 2.1

Зразок правил Snort та його частин

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

Коли Snort виявляє пакет, що відповідає неправильному значенню сигнатури, він застосовує правила, надані користувачем, та виконує вказану дію зі списку налаштованих правил. Зазвичай ці дії включають оповіщення або пропуск (ігнорування). [5].

У Таблиці 2.1 наведено приклад правил Snort. Дія "попередження" створює подію, яка може зберігати журнал або надсилати його електронною поштою, а також може викликати сигнал для машини Windows. Дія "пропуск" припиняє обробку пакетів, якщо підпис збігається в пакеті. [5].

Установка та налаштування інструментів Snort дуже прості, оскільки користувач може налаштувати правила. Він генерує спливаючі вікна та оповіщення, які контролюються Службою Windows Messenger. Snort має декілька переваг, описаних вище. Однак він має менше графічного інтерфейсу та менше пакетів, тому процес захоплення може бути досить повільним, і підтримка високошвидкісних мереж середня. З урахуванням цього, управління Snort легке і підходить для різних типів додатків. [6].

Snort - це популярна програма з відкритим вихідним кодом, яка проста в налаштуванні та переносна система виявлення мережевих вторгнень на основі шаблонів. Вона легко розгортатися на різних вузлах мережі і працює ефективно, не вимагаючи значних ресурсів пам'яті та процесорного часу. Snort використовує набір підписів, які визначають атаку, що дозволяє виявляти шкідливі дії. Ці набори підписів називаються правилами Snort, які формально визначають атаки та інші загрози.. Правило є формально визначається так:

<rule action><protocol><source ip><source port>
<direction><dest ip><dest port><rule options> [6]

Поле дії в правилах визначає тип правила Snort, такий як сповіщення або журналізація. Найбільш загальними є правила сповіщень, які зберігають дані сповіщень для подальшого аналізу та пошуку. Решта полів описують основні атрибути мережевих пакетів. Поле параметрів визначає одну або кілька пар ключ-значень, які додатково характеризують правило (наприклад, тип класу, повідомлення і т. д.). Кожне правило також має пріоритет оповіщення, який визначається за класом оповіщення. Пріоритет 1 вказує на найбільш серйозну загрозу, а пріоритет 4 - на найменшу. Приклад правила Snort:

Alert tcp $EXTERNAL NET any ->$HOME NET any (msg:
’SCAN SYN FIN’ flags: SF, 12; reference: arachnids, 198;
classtype: attempted-recon;) [6]

Основні компоненти архітектури Snort зображені на рисунку 2.1. Пакет сніффера збирає мережевий трафік і передає його на декодер, який обробляє захоплені пакети, щоб визначити заголовки протоколу на кожному рівні OSI (модель взаємодії відкритих систем). Фактичне виявлення вторгнень відбувається в блоку механізму виявлення. Цей модуль аналізує кожен пакет і перевіряє його відповідність усім правилам. Дія (сповіщення), визначена правилом, виконується кожного разу, коли виявляється пакет, який відповідає умові, визначеній в правилі [6].

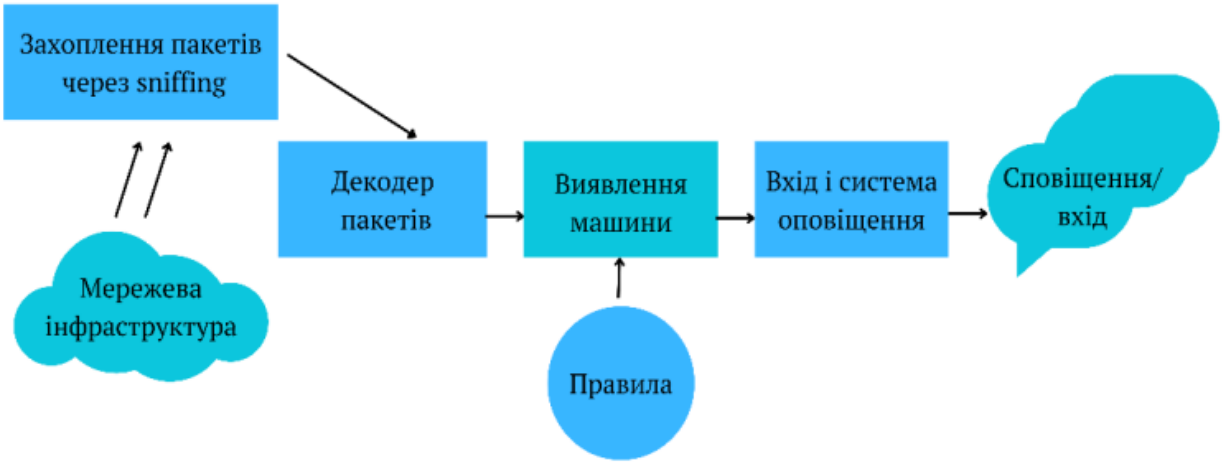

Рисунок 2.1. Архітектура Snort [7]

Запропонований інтерфейс візуалізації системи Snort IDS є клієнт-серверною програмою, яка організовує та графічно відображає сповіщення про діяльність Snort IDS. Цей інтерфейс дозволяє користувачам в реальному часі візуально аналізувати журнали трафіку та легко виявляти відхилення від нормального трафіку, які можуть свідчити про можливі вторгнення. Для демонстрації та оцінки запропонованого рішення інтерфейс візуалізації Snort інтегровано в систему, компоненти якої зображені на рисунку 2.2. З ключових точок мережі трафік надсилається на машину, на якій працює Snort IDS. Для відповідного виявлення атак Snort використовує базу правил з відкритим кодом, яку необхідно регулярно оновлювати для максимально точного виявлення. Сповіщення від Snort IDS зберігаються у файловій системі машини у форматі JSON. Графічний інтерфейс Snort зчитує дані з лог-файлу, обробляє їх та відображає користувачу [6].

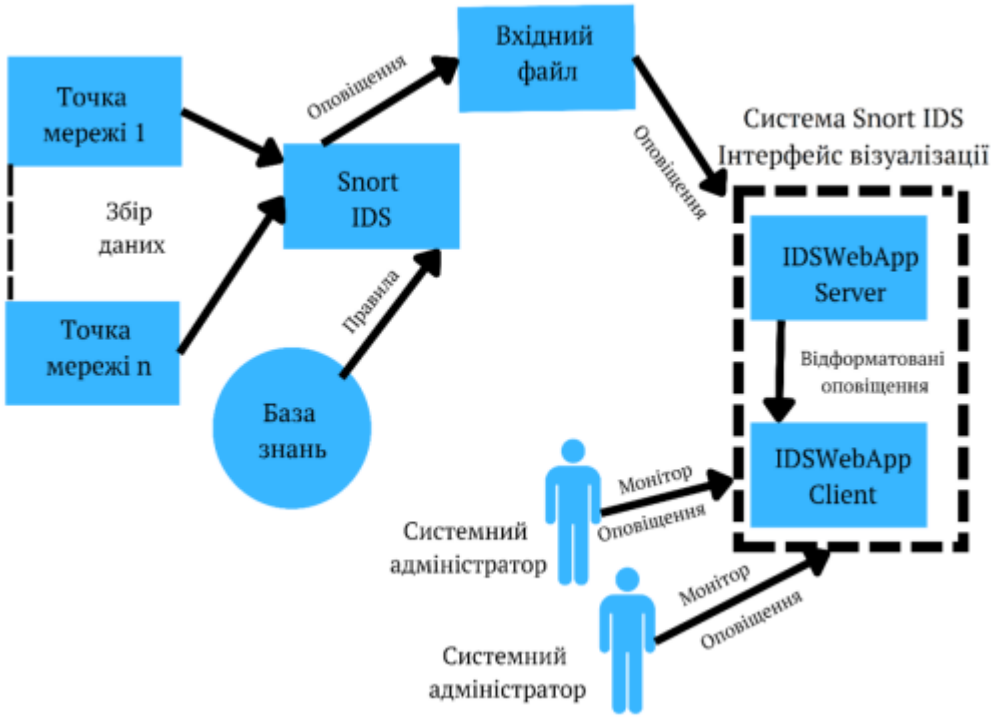

Рисунок 2.2. Архітектура запропонованої системи [7]

Використовуючи клієнт-серверну модель, запропонований інтерфейс візуалізації дозволяє ефективно графічно подавати сповіщення, створені Snort IDS. Серверна частина програми (сервер IDSWebApp) читає файл журналу Snort IDS під час запуску програми (первинне читання), а також кожного разу, коли цей файл змінюється, тобто коли Snort створює нове сповіщення. Сервер IDSWebApp також форматує ці сповіщення для коректної відправки клієнту. Після цього, сервер IDSWebApp надсилає зібрані сповіщення клієнту через веб-сокет. Сторона клієнта (клієнт IDSWebApp) отримує дані через веб-сокет і читає сповіщення, надіслані сервером. Основна функція клієнта полягає в організації та відображенні сповіщень Snort для системного адміністратора в режимі реального часу, оновлюючи інтерфейс з кожним новим сповіщенням. Це призводить до миттєвого відображення нових сповіщень для користувача. Крім того, клієнт сортує сповіщення, отримані від сервера, за чотирма критеріями та відображає найпоширеніші адреси джерел атаки, найпоширеніші пріоритети попереджень, найпоширеніші класи атак і найпоширеніші дати атак. Статистика сповіщень регулярно оновлюється, показуючи найпоширеніші атрибути атак. Крім того, якщо користувач вибирає певне сповіщення, клієнт IDSWebApp відображує детальну

інформацію про нього. Таким чином, виявлення можливого порушення дорожнього руху стає швидким і простим. Робота графічного інтерфейсу та зв'язку клієнт-серверу для відображення попереджень для користувача представлена на послідовній схемі (рисунок 2.3). [6].

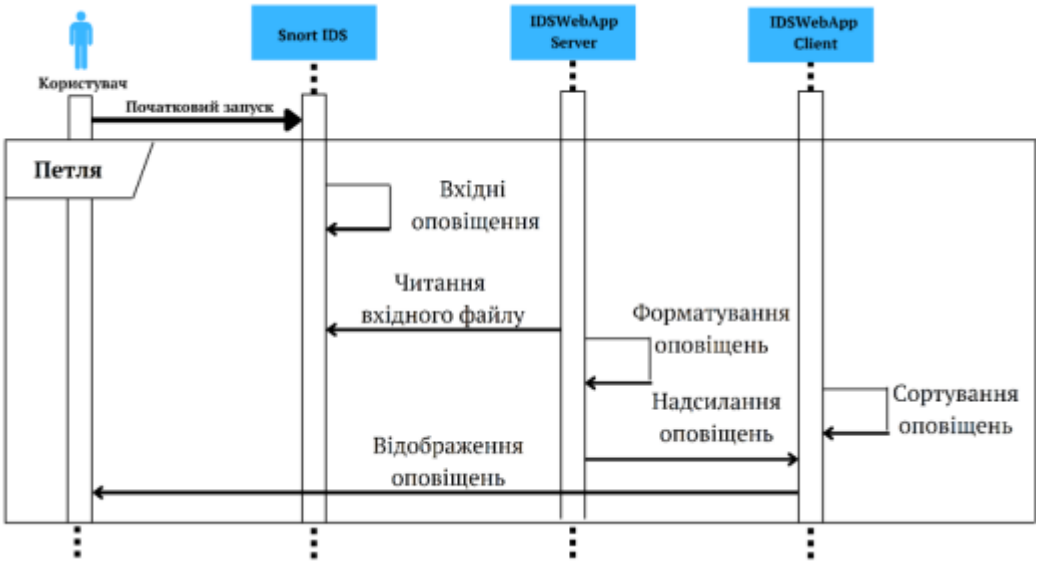

Рисунок 2.3. Послідовна схема роботи інтерефейсу візуалізації Snort IDS [7]

2.3 Порівняння з іншиим системами виявлення вторгнень та системами запобігання вторгнень для захисту від атак на мережеві протоколи

Suricata. Suricata - це програмне забезпечення з відкритим кодом, призначене для аналізу мережі та виявлення загроз. Воно використовується багатьма приватними і державними організаціями і впроваджується великими постачальниками для захисту їх активів. Suricata може виступати як система виявлення вторгнень (IDS) і система запобігання вторгненням (IPS). Розроблене Фондом відкритої інформаційної безпеки (OSIF), воно є безкоштовним інструментом, що використовується як малими, так і великими підприємствами. Система використовує набір правил і мову підписів для виявлення та запобігання загрозам. Suricata може працювати на операційних системах Windows, Mac, Unix і Linux [8].

Suricata має менше застарілого коду і використовує новіші технології порівняно з конкурентами, що сприяє його швидкості. Розробники також

забезпечили сумісність зі стандартними утилітами аналізу результатів, тому Suricata підтримує такі ж модулі, як Snort. У Suricata є два режими IPS: NFQ і AF_PACKET. NFQ IPS режим працює так:

1) Пакет потрапляє в iptables;

2) Правило iptables направляє його в чергу NFQUEUE, наприклад iptables -I INPUT -p tcp -j NFQUEUE;

3) З черги NFQUEUE пакети можуть оброблятися на рівні користувача, що і робить Suricata;

4) Suricata проганяє пакети за налаштованими правилами (rules) і в залежності від них може винести один з трьох вердиктів: NF_ACCEPT, NF_DROP або NF_REPEAT;

5) Пакети, що потрапляють у NF_REPEAT, можуть бути промарковані в системі, і спрямовані назад на початок поточної таблиці iptables, що дає величезний потенціал впливу на подальшу долю пакетів за допомогою правил iptables [8].

Починаючи з версії 1.4, Suricata може працювати як IPS за допомогою zero copy режиму системи AF_PACKET, але з деякими обмеженнями. Щоб це працювало, система повинна діяти як шлюз з двома мережевими інтерфейсами. Якщо пакет відповідає DROP правилу, він просто не пересилається на другий інтерфейс. Плюс zero copy забезпечує швидкість обробки пакетів [8].

McAfee Network Security Platform. McAfee Network Security Platform є оптимальним вибором для великих компаній, які можуть вкласти значні кошти в захист своєї мережі, оскільки вартість цього продукту починається від $10 000. Така ціна виправдана, оскільки ця IDS блокує велику кількість загроз, забезпечує захист від доступу до шкідливих сайтів, запобігає DDoS-атакам і т.д. Це рішення дозволяє блокувати нові та невідомі атаки за допомогою перевірки трафіку як із використанням, так і без використання сигнатур. Технологія виявлення вторгнень без використання сигнатур дозволяє ідентифікувати шкідливий мережевий трафік та блокувати раніше невідомі атаки, для яких не існують сигнатур [9].

Підтримка McAfee Threat Intelligence Exchange дозволяє в реальному часі отримувати інформацію про загрози як у фізичних, так і віртуальних мережах. Інтеграція з McAfee Advanced Threat Defense та McAfee MOVE AntiVirus (який є одним із компонентів рішення McAfee Cloud Workload Security) дозволяє автоматизувати складні процеси забезпечення безпеки в програмно-визначеному центрі обробки даних [9].

Ключові можливості:

– Розшифровка SSL для перевірки вхідного та вихідного мережного трафіку;

– Централізоване управління з метою оптимізації збору інформації та контролю;

– Інтеграція з набором рішень McAfee для захисту на всьому шляху від пристроїв до хмари [9].

Zeek. Zeek (раніше відомий як Bro) - це відкрите програмне забезпечення для аналізу мережевого трафіку. Інструмент спостерігає за мережевим трафіком і може працювати як у режимі стандартного виявлення вторгнень, так і у режимі виявлення шкідливих сигнатур. Zeek також може виявляти події та дозволяє встановлювати власні політики за допомогою скриптів. Це безкоштовне програмне забезпечення призначене для отримання сотень параметрів мережевих даних у режимі реального часу. Інструмент має попередньо налаштовані аналізатори для багатьох протоколів, таких як HTTP, SSL, DNS, FTP тощо, і дозволяє створювати спеціалізовані аналізатори для протоколів, які ще не підтримуються. Zeek може виявляти аномалії, але це робиться не так ефективно, як у традиційних IDS, наприклад, Suricata [9].

Цей інструмент відтворює роботу маршрутизатора в мережі, щоб отримати копію трафіку. Потім він обробляє, аналізує та структурує цей трафік на основі протоколів. Після цього оброблені дані зберігаються у різних файлах журналу, таких як dns.log, http.log, con.log і т. д. Зазвичай ці файли використовуються для завантаження у платформу SIEM.

Zeek є відмінним джерелом мережевих даних для полювання, моніторингу та аналізу. Якщо його правильно налаштувати, він не сповільнює мережу та не перевантажує групи безпеки непотрібними даними. Він витягує поля з мережевих даних, щоб надати проаналізовану та корисну інформацію, яку потім можна використовувати для створення значущих виявлень для кращого захисту мережі [9].

Протягом тривалого часу система Snort була лідером серед систем попередження та виявлення вторгнень. Однак з розвитком нових технологій, таких як багатоядерні процесори, впровадження IPv6, зростання кількості користувацьких додатків та обсягу трафіку, ця система не змогла повністю адаптуватися до нових умов.

Хоча в Snort з’явилася підтримка IPv6, можливість інспектування рівня додатків та інші покращення, вона все ще залишалася однопотоковою, що значно уповільнювало її роботу. [9].

Альтернативою Snort є Suricata. Основна відмінність між цими двома інструментами полягає в тому, що Suricata багатопотокова. Це означає, що інструмент може використовувати кілька ядер одночасно, що дозволяє краще балансувати навантаження. Це дозволяє обробляти більше даних, не повертаючись до кількості запроваджених правил, що дає Suricata невелику перевагу над Snort [9].

Однією з ключових особливостей Suricata є те, що вона, крім власних розробок, використовує практично всі функціональні можливості, які вже наявні у Snort. Suricata формує виведення подій у форматі JSON, що значно спрощує її інтеграцію зі сторонніми програмами, такими як системи моніторингу та візуалізації логів (наприклад, Kibana).

Ще однією перевагою Suricata є можливість обробки на рівні 7 OSI, що підвищує його здатність виявляти шкідливі програми для додатків. У правилах можна не прив'язуватися строго до номера порту, як це робиться у Snort; достатньо вказати протокол та дію. Далі модулі Suricata самостійно розбираються з трафіком і виявляють протокол, навіть якщо використовується нестандартний порт [9].

Мінус Suricata – велика кількість налаштувань і недостатньо виразна в деяких питаннях документація.

Zeek - чудовий інструмент для виявлення загроз. Багато IDS, такі як Suricata, зосереджені на виявленні на основі сигнатур і правил. Zeek можна використовувати як традиційний IDS. Однак Zeek можна використовувати для глибшого аналізу конкретних мережевих протоколів. Чим більше даних вам доведеться обробляти під час пошуку загроз, тим краще. Недолік Zeek полягає в складності спілкування з інструментом, оскільки технологія зосереджена на функціональності, а не на графічному інтерфейсі [9].

McAfee Network Security Platform є дорогою системою, що може виявитися недоліком у порівнянні з іншими системами, які мають відкритий вихідний код. Проте вона є значно ефективнішою, коли потрібно забезпечити максимальний рівень захисту мережі. Перевагою McAfee є можливість працювати як з використанням сигнатур, так і без них, що дозволяє виявляти раніше невідомі атаки. Ще одним недоліком є уповільнення мережі через великий функціонал. Після аналізу відомих систем виявлення та запобігання вторгнень можна виділити такі їхні основні риси (Таблиця 2.2) [9].

Таблиця 2.2

Порівняльний аналіз сучасних систем виявлення та запобігання вторгнень

Назва системи	Переваги	Недоліки	Спосіб вирішення
Snort	Можливість працювати під прикриттям; здатність виявляти і блокувати різноманітні атаки.	Сповільнення роботи через однопотоковість; обмежене сканування лише за вказаним номером порту.	Модернізація системи; оптимізація використання ресурсів; реалізація багатопотоковості.
Suricata	Багатопотоковість та висока продуктивність; легка інтеграція зі сторонніми програмами; обробка на рівні OSI 7.	Велика кількість налаштувань; недостатня чіткість документації в деяких питаннях.	Використання сторонніх програм (наприклад, Kibana) для полегшення спілкування з системою.
McAfee Network Security Platform	Можливість виявлення невідомих раніше	Висока вартість; сповільнення роботи мережі через	Рекомендація використовувати систему лише в разі

	атак завдяки роботі як з використанням сигнатур, так і без них; забезпечення максимального захисту мережі.	великий функціонал.	необхідності максимального захисту, який виправдовує затрати на її встановлення та незручності пов'язані з її використанням.
Zeek	Підтримка різних режимів роботи; гнучкий функціонал; можливість створювання власних скриптів політик та спеціальних аналізаторів протоколів.	Складність спілкування з інтерфейсом та налаштування.	Систему доцільно використовувати лише фахівцям з великим досвідом роботи з подібними системами; вона призначена для використання в мережах, де необхідна гнучкість та велика кількість налаштувань.

Основні недоліки сучасних систем виявлення та запобігання вторгнень включають:

1) Замедлення роботи системи або мережі.

2) Велика кількість налаштувань.

3) Складність роботи з інтерфейсом.

4) Складність налаштування системи для недосвідчених користувачів [9].

2.4. Аналіз ефективності застосування Snort у виявленні та запобіганні атак на мережеві протоколи, включаючи його можливості та обмеження

Snort має три режими роботи:

- Режим сніфера, який просто перехоплює дані і виводить їх на екран.

- Режим реєстратора пакетів, який перехоплює дані і зберігає їх у відповідних файлах.

- Режим системи виявлення вторгнень, який перехоплює дані, аналізує їх і автоматично реєструє пакети, якщо не вказано інше [10].

Barnyard2 є інтерпретатором двійкових файлів, створених Snort, з відкритим вихідним кодом.

Стандартний спосіб запису подій в Snort через консоль або у файл є досить ресурсоємким. Найкращим варіантом є збереження подій Snort у базі даних MySQL [10].

PulledPork - це скрипт, який завантажує, комбінує, встановлює та оновлює правила для Snort з різних джерел. З його допомогою можна завантажити безкоштовний набір правил спільноти Snort без створення облікового запису на Snort.org.

BASE (Basic Analysis and Security Engine) - це базовий двигун аналізу та безпеки, який використовується для візуалізації виявлених атак.

Індикаторами ефективності систем виявлення та запобігання вторгнень повинні бути: швидкість виявлення атак і рівень помилкових спрацьовувань. Обмеження потужності вказує на несправність системи: як тільки вона досягає граничної потужності, пакети відкидаються, що може призвести до невиявлення шкідливого вмісту [10].

Для кількісної оцінки метрик ефективності систем виявлення та запобігання вторгнень можна використати такі показники: охоплення (кількість атак, які можна виявити), ймовірність помилкових спрацьовувань, ймовірність виявлення резистивних атак, здатність обслуговувати канал з високою пропускною здатністю і ємністю. Продуктивність системи складається з різних компонентів і тому не є окремою метрикою.

Потрібно вести облік таких показників: швидкість в байтах на секунду, кількість пакетів на секунду та загальна кількість мережевих атак. Крім того, для кожної системи виявлення та запобігання вторгнень в мережі необхідно відслідковувати зменшення кількості втрачених пакетів, записувати фактичні тригери, помилкові спрацьовування, негативні тригери та загальну кількість спрацьованих тривог. Нарешті, слід відстежувати використання центрального процесора та пам'яті на хості, постійне зберігання, пропускну здатність інтерфейсу та статистику файлів підкачки [10].

VMware Workstation 15 використовувалася як платформа для віртуалізації, завдяки своїй високій продуктивності вводу-виводу та жорсткого диска порівняно

з іншими засобами віртуалізації. Операційною системою було обрано 32-розрядну Ubuntu 18.04 LTS, яка регулярно оновлюється і має широку базу спільноти. Ubuntu є найпопулярнішою операційною системою Linux [10].

Стандартна апаратна конфігурація для системи виявлення та запобігання вторгнень в мережу включала 2,8 ГГц чотирьохядерний процесор Intel Xeon (E5462) з 3 ГБ DDR2 800 МГц повністю буферизованою пам'яттю. Кожна система також мала максимальний об’єм жорсткого диска 20 ГБ. Мережевий трафік передавався окремо для кожної системи. Система, що відтворювала мережевий трафік, використовувала одне ядро та 1 ГБ оперативної пам'яті [10].

Хост-система VMware мала 2 ГБ оперативної пам'яті і 1 ядро, що обмежувало його продуктивність у використанні на випробувальному стенді.

Ідентичні методи реєстрації, які називаються Barnyard, MySQL та AcidBase, використовувались як для систем виявлення вторгнень в мережі, так і для систем запобігання. Версія Snort v 2.9.8.3, після того, як нові правила були завантажені, визначення правил були в Snort 11065 [10].

При виборі мережевого трафіку для тестування систем виявлення та запобігання вторгненню в мережу потрібно враховувати кілька особливостей. По-перше, атакуючий трафік можна використовувати окремо або разом з контекстним фоновим трафіком. Фоновий трафік може бути реальним або імітованим. Якщо це реальна поведінка, її можна залишити без змін або дезінфікувати, тобто видалити дані користувача та IP-адреси [10].

Для тестування було корисно і бажано використовувати реальний мережевий трафік у фоновому режимі. Однак повторення експериментів з реальним трафіком було б непередбачуваним через його динаміку. Тому було обрано використання трафіку, захопленого з файлу pcap. Це дозволило системі виявлення та запобігання вторгненням відтворювати мережевий трафік в автономному режимі, що дозволяло його аналізувати з різною швидкістю за допомогою TCPReplay. Крім того, це усунуло ризики для критично важливих мереж [10].

Існує багато джерел тестового трафіку для завантаження, але, на жаль, багато з них дезінфікуються, що робить їх непридатними для оцінки NIDPS, яка проводить

глибокий аналіз пакетів. Існують інструменти, такі як TCPdump Randomiser, які додають корисне навантаження до очищення даних. Однак реалістичність таких змінених даних може бути сумнівною. Контекстний злом також забезпечує джерела для захоплення трафіку, але зміст трафіку не завжди задокументовано, тому його потрібно визначати перед використанням. Наприклад, деякі атаки можуть бути успішними, а інші - ні. На основі цих питань було вирішено зафіксувати фоновий трафік із задіяних веб-серверів та серверів додатків, а потім поєднати його з керованим трафіком, створеним за допомогою Metasploit Framework [10].

Використаний трафік було зафіксовано для запуску атак Metasploit на комп'ютері з операційною системою Microsoft Windows 2000. Обрано Windows 2000 як найбільш підходящу для Metasploit порівняно з іншими операційними системами. Для цього було налаштовано велику кількість служб і додатків, які перестали працювати, щоб дозволити якомога більше атак. На жаль, не всі з них було можливо отримати. Атаки, перелічені в таблиці 3.4, були записані за допомогою Wireshark, який захоплював як фоновий, так і атакуючий типи трафіку. Частина програми Wireshark, Edicap, використовувалася для зміни часової позначки використаного трафіку та співставлення її з фоновим трафіком. Після цього їх було об’єднано в хронологічному порядку, щоб атакуючий трафік відобразився на другому плані.

Продуктивність NIDPS тісно пов'язана з продуктивністю центрального процесора. Тому Snort повинні завантажувати центральний процесор, щоб оцінити їх роботу в стресових умовах [10].

Для зменшення кількості логічних та фізичних ядер використовувався VMware. Це дозволило створювати регульовані та вимірювані навантаження. З цією метою був використаний інструмент cpulimit, який обмежує відсоток потужності центрального процесора кожного потоку.

Snort дозволяють внутрішнє відтворення файлів pcap, що дозволяє оцінити продуктивність системи. Однак цей метод не враховує максимальну швидкість без

втрат (MLFR). Тому для оцінки швидкості трафіку використовувався TCPReplay [10].

Відстежувались такі ресурси: використання центрального процесора, використання пам'яті, опір пропускної здатності пам'яті та пропускна здатність мережі. Це було зроблено за допомогою інструмента командного рядка Linux dstat [10].

Атакуючий трафік даних маршрутизувався через обидві NIDPS з різними конфігураціями процесора. Сюди входять: 2-ядерна конфігурація ядра процесора, 1 ядро, навантаження 50% та 75%. Здатність NIDPS читати пакети вимірювалась, як і точність попереджень, з особливою увагою приділялись помилково негативні результати. Тестовий трафік був перенаправлений в середовище за допомогою TCPReplay, помноженого на 40, щоб відтворити його 40 разів швидше, ніж при захопленні. Це забезпечило можливість завершення експериментів вчасно, без втрати пакетів [10].

Кожного разу, коли запускалося тестування, реєструвались початок і кінець трафіку запуску NIDPS. Це було відправною точкою для аналізу системи попередження та статистики. Інформація про попередження записувалась за допомогою acidbase, також відомого як вихідний файл unified2, який архівується для подальшого використання. Статистика продуктивності NIDPS закривалась, записувалась, кількість генерованих сповіщень, кількість оброблених пакетів та їх відношення до оброблюваних мережевих протоколів. Трафік проходив через хости 192.168.16.2 та 192.168.16.128, але був позначений як небажаний трафік [10].

Найважливішим показником оцінки IDPS є точність. Це розглядалося як покриття атаки системи, помилковий позитивний, помилковий негативний, потенційний, і здатність обробляти великий трафік, тобто з великою пропускною здатністю. Snort не спромігся попередити про експлойт ms01_033_idq, коли центральний процесор мав менше 50% навантаження. Частково це пов'язано з тим, що Snort менше контролює функціонування оповіщень під час атаки. Snort не зміг попередити ms01_033_idq двома правилами з набору правил VRT, ідентифікаторами 1245 та 1244. Snort має набагато нижчі системні вимоги, тому він

не може працювати з втратою пакетів при максимальному навантаженні системи [10].

3 РОЗРОБКА РЕКОМЕНДАЦІЙ ЩОДО ЗАСТОСУВАННЯ МЕТОДІВ ТА ЗАСОБІВ ЗАХИСТУ МЕРЕЖЕВИХ ПРОТОКОЛІВ

3.1. Налаштування та оптимізація Snort для максимального виявлення та запобігання атак на мережеві протоколи

Налаштування та оптимізація Snort можуть значно підвищити ефективність виявлення та запобігання атак на мережеві протоколи. Ось кілька кроків, які можна виконати:

Регулярне оновлення правил Snort є критично важливим для забезпечення максимальної ефективності виявлення та запобігання атак на мережеві протоколи. Ось деякі додаткові поради щодо цього:

1. Автоматичні оновлення: налаштуйте автоматичне оновлення правил Snort, щоб забезпечити їхню регулярність. Це може бути виконано через інтеграцію з репозиторіями сигнатур або використання спеціальних інструментів керування правилами, таких як PulledPork.

2.Використання комерційних підписок: якщо це можливо, розгляньте підписку на комерційні правила, які можуть містити додаткові сигнатури та отримувати оновлення частіше.

3. Спостереження за новими загрозами: слідкуйте за новими загрозами та вразливостями, оскільки оновлення правил можуть включати сигнатури для їх виявлення.

4. Тестування перед впровадженням: перед тим як застосувати нові правила на продуктивне середовище, перевірте їх на тестовому середовищі, щоб уникнути можливих проблем.

5. Моніторинг змін: слідкуйте за змінами у важливих правилах та сигнатурах, щоб бути в курсі того, як вони можуть вплинути на виявлення атак.

6. Оновлення додаткових компонентів: окрім самих правил Snort, також оновлюйте інші компоненти, які використовуються разом із Snort, такі як preprocessor і додаткові файли конфігурації.

7. Аналіз логів після оновлення: після застосування оновлень перевірте логи, щоб переконатися, що нові правила не спричинили фальшивих позитивів або інших проблем.

Використання правил з високою точністю допомагає уникнути фальшивих спрацювань та зменшити кількість ложнопозитивних результатів. Однак, це може також призвести до пропуску деяких атак, тому важливо збалансувати точність та обсяг виявлення. Ось деякі способи розширення цієї стратегії:

1. Перевірка правил на власному середовищі: перевіряйте правила з високою точністю на своєму середовищі, щоб переконатися, що вони не блокують легітимний трафік.

2. Додаткові перевірки: додайте додаткові правила або скрипти для перевірки важливих параметрів трафіку перед застосуванням правил з високою точністю, щоб зменшити ризик блокування дійсного трафіку.

3. Регулярні аудити та аналіз: проводьте регулярні аудити ефективності правил з високою точністю, аналізуйте логи та виправляйте проблемні ситуації.

4. Використання правил з адаптивними порогами: деякі правила можуть мати можливість налаштування порогів для кількості спрацювань перед активацією. Використання цієї функції може допомогти уникнути блокування легітимного трафіку.

5. Контекстне виявлення: розгляньте використання правил, які враховують контекст дійсності трафіку (наприклад, час, попередній зв'язок тощо), щоб зменшити ризик блокування легітимного трафіку.

6. Моніторинг та звітність: підтримуйте моніторинг і звітність щодо спрацювання правил, щоб оперативно виявляти проблеми та вносити корективи.

7. Оновлення і підтримка: постійно оновлюйте та підтримуйте правила, щоб вони відображали актуальні загрози та сучасні методи нападу.

Видалення зайвих правил - це важлива частина оптимізації Snort, яка допомагає підвищити ефективність системи та зменшити навантаження на неї. Ось детальніше про цей аспект:

1. Аналіз контексту: оцініть, які з правил дійсно відповідають вашому середовищу та мережевим потребам. Це може включати правила для конкретних протоколів, портів чи типів атак.

2. Використання специфічних правил: деякі правила можуть бути призначені для конкретних протоколів або програм. Якщо вони не використовуються в вашій мережі, вони можуть бути видалені.

3. Відсів потенційно небезпечних правил: перегляньте правила, які відносяться до відомих, але неактивних загроз. Наприклад, якщо певний тип атаки вже не актуальний або не відомий вам, можна видалити відповідні правила.

4. Оцінка впливу на продуктивність: деякі правила можуть бути дуже ресурсомісткими або призводити до багатьох фальшивих спрацювань. Видалення цих правил може поліпшити продуктивність системи та зменшити навантаження.

5. Постійний аналіз і оновлення: регулярно переглядайте та оновлюйте список правил, оскільки мережеві умови та загрози можуть змінюватися. Це допоможе тримати правила актуальними та ефективними.

6. Збереження резервних копій: перед видаленням будь-яких правил зробіть резервну копію, щоб у разі потреби відновити їх.

7. Тестування перед видаленням: перевірте видалені правила на тестовому середовищі, щоб переконатися, що вони не вплинуть на ефективність виявлення атак.

8. Логування змін: ведіть журнал процесу видалення правил, щоб мати можливість відстежувати зміни та повертатися до попереднього стану у разі потреби.

Налаштування порогів спрацьовування є важливим кроком у підтримці ефективності Snort та уникненні перевантаження системи фальшивими позитивами. Ось деякі способи розширення цієї стратегії:

1. Налаштування чутливості правил: зменшіть чутливість правил, які мають високу ймовірність спрацювання на нормальному трафіку. Це може бути зроблено за допомогою параметрів, таких як threshold, depth та distance в конфігураційних файлах Snort.

2. Виключення правил для конкретних IP-адрес або мереж: якщо певний трафік або мережеві сегменти вам відомі та надійні, можна виключити відповідні правила для цих адрес або мереж, щоб уникнути фальшивих спрацьовувань.

3. Установка порогів на часові інтервали: встановлення порогів на кількість спрацьовувань правила протягом певного часового інтервалу може допомогти уникнути перевантаження системи фальшивими позитивами.

4. Використання динамічних порогів: розгляньте використання алгоритмів, які адаптивно налаштовують пороги спрацьовування в залежності від обсягу та характеру трафіку. Наприклад, алгоритм Sliding Window може автоматично адаптувати пороги на основі останніх спрацьовувань.

5. Моніторинг та аналіз логів: підтримуйте моніторинг та аналіз логів Snort, щоб виявляти будь-які ознаки перевантаження або надмірної кількості фальшивих позитивів. На основі цього можна налаштувати пороги спрацьовування.

6. Регулярне оновлення порогів: періодично переглядайте та оновлюйте налаштування порогів спрацьовування на основі аналізу трафіку та інцидентів безпеки.

7. Тестування перед впровадженням: перед застосуванням змін у налаштуваннях порогів спрацьовування перевірте їх на тестовому середовищі, щоб переконатися в їхній ефективності та відсутності негативного впливу на продуктивність.

Використання портових фільтрів є важливим аспектом налаштування Snort для забезпечення більшої ефективності та уникнення непотрібних спрацьовувань правил. Тут деякі способи розширення цієї стратегії:

1. Зведення ризику: розгляньте порти, які найбільш імовірно використовуються для потенційно небезпечних протоколів або послуг. Наприклад, порти, що відповідають за веб-сервери (наприклад, 80 для HTTP, 443 для HTTPS) або для електронної пошти (25 для SMTP, 110 для POP3), можуть бути вибрані для обмеження.

2. Використання кастомних правил: створіть кастомні правила для конкретних портів, які виключають або обмежують деякі типи трафіку.

Наприклад, існує мати правило, яке відфільтровує трафік на порті 25 (SMTP), якщо не використовується поштовий сервер.

3. Врахування внутрішньої політики безпеки: встановлення фільтрів на портах може відображати внутрішню політику безпеки вашої організації. Наприклад, якщо у вас є сервіси, які повинні бути доступні тільки внутрішньо, можна обмежити правила на внутрішніх мережних портах.

4. Виключення відомих безпечних портів: відомі безпечні порти, які використовуються для системних служб або невразливих протоколів, можуть бути виключені з аналізу, щоб уникнути непотрібних перевірок.

5. Контроль за відомими загрозами: зверніть особливу увагу на порти, які часто використовуються для відомих атак. Застосування більш жорстких правил на цих портах може допомогти вчасно виявляти та запобігати атакам.

6. Моніторинг та аналіз результатів: після встановлення портових фільтрів регулярно переглядайте логи Snort, щоб переконатися, що правила працюють належним чином і не блокують легітимний трафік.

Настроювання перехоплення пакетів:

При виборі інтерфейсів та налаштуванні фільтрів для перехоплення пакетів важливо врахувати потреби вашої мережі та обсяг трафіку. Ось кілька способів, які можуть допомогти оптимізувати цей процес:

1. Вибір потрібних інтерфейсів: оберіть лише ті мережеві інтерфейси, які використовуються в вашій мережі і можуть бути джерелом або призначенням трафіку для аналізу.

2. Встановлення фільтрів: налаштуйте фільтри для перехоплення тільки тих пакетів, які вас цікавлять. Це може бути здійснено за допомогою фільтрів pcap або BPF (Berkley Packet Filter), щоб обмежити аналіз до певних IP-адрес, портів або протоколів.

3. Виключення непотрібного трафіку: виключіть перехоплення пакетів, які не є цікавими для вас, наприклад, внутрішніх адрес або безпечних портів, щоб зменшити навантаження на систему.

Використання режиму Inline:

Режим Inline дозволяє Snort негайно реагувати на потенційно шкідливий трафік, блокуючи його ще до того, як він досягне цільової системи. Ось як це може бути розширено:

1. Увімкнення режиму Inline: налаштуйте Snort для роботи в режимі Inline, щоб він міг надійно блокувати небезпечний трафік на мережевому рівні.

2. Вибір правил для блокування: використовуйте правила з високою точністю та достовірністю для блокування, а також правила, які відповідають конкретним загрозам вашої мережі.

3. Настройка дій при виявленні загрози: визначте, що робити, коли Snort виявить потенційну загрозу. Це може бути блокування, відхилення або логування подій.

Налаштування логування:

Логування подій є важливим аспектом моніторингу безпеки мережі. Тут кілька способів розширення цього аспекту:

1. Деталізоване логування: налаштуйте Snort для зберігання докладних логів про виявлені загрози, включаючи інформацію про IP-адреси, порти, типи атак та іншу корисну інформацію.

2. Централізоване логування: налаштуйте систему логування, яка централізує логи Snort з усіх ваших сенсорів для зручності моніторингу та аналізу.

Моніторинг та аналіз:

Моніторинг та аналіз логів Snort є важливими етапами для виявлення потенційних загроз та вразливостей в вашій мережі. Деякі способи розширення цього аспекту:

1. Використання інструментів аналізу: використовуйте спеціальні інструменти для аналізу логів Snort, які допоможуть вам ідентифікувати вразливі місця та потенційні атаки.

2. Регулярний аудит і звітність: проводьте регулярні аудити безпеки та створюйте звіти про стан безпеки мережі на основі аналізу логів Snort.

Регулярне тестування:

Періодичне тестування ефективності Snort допоможе переконатися, що він працює належним чином і виявляє потенційні загрози. Деякі можливості розширення цього аспекту:

1. Сценарії тестування: розробіть сценарії тестування, які відображають реальні сценарії атак і перевіряють реакцію Snort.

2. Перевірка правильності конфігурації: перевірте правильність налаштувань Snort під час тестування, включаючи фільтри, правила та реакцію на атаки.

3. Оновлення і підтримка: постійно оновлюйте та підтримуйте Snort, щоб мати доступ до оновлених правил і функцій для ефективного виявлення атак.

Ці кроки допоможуть вам максимізувати ефективність Snort та забезпечити надійний рівень захисту вашої мережі.

3.2. Розробка стратегій моніторингу та реагування на атаки на мережеві протоколи з використанням Snort

Розробка стратегій моніторингу та реагування на атаки на мережеві протоколи з використанням Snort включає в себе кілька ключових кроків, щоб ефективно виявляти, відслідковувати та реагувати на потенційні загрози. Ось деякі стратегії:

1. Налаштування та встановлення Snort:

- Вибір сенсорів: виберіть місця у вашій мережі для розміщення сенсорів Snort, де вони можуть моніторити весь трафік.

- Встановлення Snort: встановіть та налаштуйте Snort на сенсорах для аналізу трафіку. Врахуйте використання режиму Inline для негайного блокування атак.

2. Налаштування правил та фільтрів:

- Вибір правил: встановіть правила Snort, які відповідають вашим потребам безпеки та відображають особливості вашої мережі.

- Фільтрація трафіку: налаштуйте фільтри для перехоплення тільки необхідних пакетів та виключення трафіку, який не викликає підозр.

3. Моніторинг та аналіз:

- Логування подій: налаштуйте Snort для створення докладних логів про виявлені загрози.

- Централізоване логування: настройте централізоване логування, щоб забезпечити централізований доступ до логів з усіх сенсорів.

- Моніторинг: постійно моніторте логи Snort для виявлення аномального трафіку та потенційних атак.

4. Реагування на виявлені загрози:

- Автоматичне блокування: використовуйте режим Inline для негайного блокування атак або підозрілого трафіку.

- Реагування вручну: аналізуйте виявлені загрози та вживайте відповідні заходи, якщо необхідно. Це може включати ізоляцію комп'ютерів, блокування IP-адрес, а також розслідування подій.

5. Оновлення та тестування:

- Оновлення правил: регулярно оновлюйте правила Snort, щоб вони відображали актуальні загрози.

- Тестування ефективності: періодично тестуйте ефективність Snort, щоб переконатися, що він працює належним чином і виявляє потенційні атаки.

6. Підвищення навичок персоналу:

- Навчання персоналу: забезпечте навчання вашому персоналу з моніторингу безпеки та використання Snort для ефективного реагування на загрози.

- Установка процедур реагування: розробіть процедури реагування на інциденти безпеки, щоб ваш персонал знаходився в курсі та знаходився готовим діяти у випадку виявлення загрози.

Розробка цих стратегій допоможе вашій організації підтримувати високий рівень захисту мережі та даних від потенційних атак.

Зважаючи на комплексність і постійну зміну загроз кібербезпеки, розробка стратегій моніторингу та реагування на атаки на мережеві протоколи з використанням Snort потребує ретельного планування та найкращих практик. Ось додаткові аспекти, які можна включити до стратегії:

7. Налаштування порогів та алармів:

- Встановлення критичних порогів: налаштуйте пороги спрацьовування правил та аларми, щоб вчасно виявляти підозрілий та аномальний трафік.

- Коригування порогів: постійно переглядайте та коригуйте пороги на основі аналізу трафіку та інцидентів безпеки для забезпечення оптимального спрацювання алармів.

8. Забезпечення неперервності роботи:

- Резервне копіювання та відновлення: розробіть процедури резервного копіювання та відновлення для Snort і його конфігурацій, щоб забезпечити неперервну роботу системи.

- Шкалювання і висока доступність: розгляньте можливості шкалювання та високої доступності для Snort, особливо в великих або критичних мережах.

9. Використання розвідувальних інструментів:

- Використання Threat Intelligence: інтегруйте дані з Threat Intelligence для покращення розпізнавання та виявлення загроз.

- Моніторинг зовнішньої активності: слідкуйте за зовнішньою активністю та новинами щодо відомих загроз для активного реагування.

10. Постійне навчання та оновлення:

- Навчання персоналу: забезпечте постійне навчання та підвищення кваліфікації вашого персоналу з моніторингу безпеки та використання Snort.

- Стеження за оновленнями та новими функціями: будьте в курсі оновлень Snort та нових функцій, щоб використовувати найновіші можливості для захисту вашої мережі.

11. Тестування та сценарії інцидентів:

- Створення сценаріїв інцидентів: розробіть сценарії та плани дій для реагування на різноманітні типи кібератак.

- Проведення імітаційних вправ: регулярно проводьте імітаційні вправи та тестування для перевірки ефективності вашої стратегії та підготовленості персоналу.

Ці додаткові кроки допоможуть удосконалити вашу стратегію моніторингу та реагування на атаки на мережеві протоколи з використанням Snort і підвищити рівень безпеки вашої мережі.

3.3. Визначення оптимального співвідношення між захистом та продуктивністю при застосуванні заходів захисту мережевих протоколів

Оптимальне співвідношення між захистом та продуктивністю при застосуванні заходів захисту мережевих протоколів може бути досягнуте шляхом збалансованого підходу, який враховує ризики безпеки та потреби у продуктивності мережі. Ось кілька кроків для досягнення такого співвідношення:

1. Аналіз ризиків безпеки: оцініть потенційні загрози та вразливості вашої мережі. Це допоможе визначити, які заходи захисту є необхідними для запобігання атакам.

2. Проведення аудиту безпеки мережі: перевірте існуючі заходи захисту та визначте, чи вони вистачають для захисту від відомих загроз.

3. Вибір правильних заходів захисту: оберіть заходи захисту, які найбільш ефективно відповідають ідентифікованим ризикам. Це може включати файерволи, ідентифікацію та управління доступом, моніторинг мережі тощо.

4. Тестування та оцінка продуктивності: перевірте, як заходи захисту впливають на продуктивність мережі. Це може включати швидкість передачі даних, час відклику та інші метрики продуктивності.

5. Оптимізація параметрів захисту: налаштуйте параметри захисту таким чином, щоб забезпечити ефективний захист без надмірного впливу на продуктивність мережі. Наприклад, встановлення правильних правил файерволу або використання проксі-серверів для фільтрації трафіку.

6. Стеження за новими загрозами та технологіями: залишайтесь в курсі останніх розвитків у сфері кібербезпеки та мережевих технологій, щоб забезпечити постійне оновлення заходів захисту та оптимізацію продуктивності.

7. Неперервний моніторинг та налагодження: постійно відслідковуйте стан безпеки мережі та її продуктивність і вносьте необхідні корективи для підтримки оптимального співвідношення між захистом та продуктивністю.

Важливо розуміти, що оптимальне співвідношення може змінюватися з часом, оскільки з'являються нові загрози та технології, а також змінюються потреби бізнесу.

Висновок полягає в тому, що досягнення оптимального співвідношення між захистом та продуктивністю при застосуванні заходів захисту мережевих протоколів є ключовим завданням для будь-якої організації з мережевою інфраструктурою. Забезпечення високого рівня захисту від кіберзагроз без збоїв у продуктивності мережі допоможе уникнути серйозних проблем із безпекою та забезпечить ефективну роботу бізнесу.

Важливо враховувати індивідуальні потреби та характеристики кожної мережі при визначенні оптимального співвідношення. Це може вимагати постійного моніторингу, оновлення заходів захисту та налагодження параметрів для досягнення балансу між безпекою та продуктивністю.

Нарешті, враховуючи постійне зростання кількості кіберзагроз і еволюцію мережевих технологій, необхідно продовжувати вдосконалювати стратегії захисту та реагувати на нові виклики, щоб забезпечити стійкість мережі і надійний захист у майбутньому.

ВИСНОВКИ

У ході даного дослідження проведено аналіз технік виявлення атак на мережеві протоколи. У наш час кіберзлочинність набрала нових обертів і потребує більш глибокого та доцільного аналізу. Повноцінна робота будь-якої сучасної організації залежить від роботи мережевих протоколів та їх захисту.

У першому розділі проведено дослідження проблем захисту мережевих протоколів. Було виділено найпоширеніші загрози мережевим протоколам – DoS та MitM. Розкрито методи захисту від загроз та вразливостей мережевих протоколів. До основних методів належить встановлення і використання систем захисту, використання шифрування, встановлення політики безпеки, оновлення програмного забезпечення. Також виділено потенційні ризики: порушення конфіденційності даних, пошкодження цілісності даних, переривання доступності сервісів, експлуатація вразливостей, вплив на ділову діяльність.

Останнім кроком є аналіз існуючих рішень із захисту мережевих протоколів: firewalls, інтригування трафіку, шифрування трафіку, аутентифікація та авторизація, використання VPN, блокування небезпечних протоколів, обмеження доступу до мережевих ресурсів, понадшарове виявлення загроз; та зазначено їх плюси і мінуси.

У другому розділі зроблено аналіз методів та засобів захисту від атак на базі Snort. Було проведено загальний опис роботи Snort, розглянута архітектура Snort. Вивчили правила та підписи, які використовуються, вказано декілька правил. Також було проілюстровано архітектуру Snort та архітектуру системи, котра була запропонована в ході роботи. Провели порівняння Snort з іншими системами виявлення вторгнень та встановили переваги і недоліки кожної з них. Перевагами Snort є те, що система має можливість працювати під прикриттям, здатна виявляти і блокувати різноманітні атаки.

У третьому розділі було розроблено рекомендації щодо застосування методів та засобів захисту мережевих протоколів. Налаштування Snort можна зробити за допомогою наступних кроків: регулярне оновлення правил, використання правил з високою точністю, видалення зайвих правил,

використання портових фільтрів, налаштування логування та інші. Також зазначили, що для встановлення оптимального співвідношення захист/продуктивність потрібно зробити наступні кроки: Аналіз ризиків, проведення аудиту безпеки мережі, вибір правильних заходів захисту, тестування та оцінка продуктивності, оптимізація параметрів захисту, стеження за новими загрозами та технологіями, неперервний моніториг та налагодження.

Отже, у сучасному світі технології розвиваються швидко, і як наслідок розвиваються способи створення загроз. Управління даними загрозами потребує швидкого реагування, постійного вдосконалення та створення нових способів уникнення кіберзагроз. Потрібно постійно забезпечувати стійкість мережі для повноцінної роботи всіх складових сучасного суспільства.

ПЕРЕЛІК ПОСИЛАНЬ

1. NORT Users Manual. URL: http://manual-snort-org.s3-website-us-east-1.amazonaws.com/

2. Mitnick K. The Art of Invisibility: The World's Most Famous Hacker Teaches You How to Be Safe in the Age of Big Brother and Big Data / Kevin Mitnick., 2019. – 320 с.

3. Snort instaling on Window. URL: https://www.snort.org/.

4. Програмний комплекс захисту мережевого периметру інформаційно-комунікаційної системи підприємства. URL: https://essuir.sumdu.edu.ua/bitstream-download/123456789/84406/1/Hura_bac_rob.pdf;jsessionid=C8E7F7B96FDA0B1ADADF99DA232A4BCE

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

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

7. Система виявлення мережевих вторгнень на основі Snort. URL: http://dspace.wunu.edu.ua/bitstream/316497/46657/1/%D0%93%D0%B0%D0%B2%D1%80%D0%B8%D0%BB%D1%8F%D0%BA.pdf

8. Suricata. URL: https://suricata.io/features/

9. Аналіз сучасних відкритих систем виявлення та запобігання вторгнень. URL: https://www.researchgate.net/publication/371149535_ANALIZ_SUCASNIH_VIDKRITIH_SISTEM_VIAVLENNA_TA_ZAPOBIGANNA_VTORGNEN

10. Виявлення вторгнення в комп’ютерну мережу на основі аналізу трафіку. URL: https://er.nau.edu.ua/bitstream/NAU/47214/1/%D0%A4%D0%9A%D0%9A%D0%9F%D0%86_123_2020_%D0%9A%D0%BE%D0%B2%D0%B0%D0%BB%D1%8C%20%D0%94.%D0%A1..pdf