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

КОСТЮК Олександр Васильович

Система автоматичного реагування на мережеві атаки / Automated Network Attack Response System

спеціальність: 125 – Кібербезпека

освітньо-професійна програма – Кібербезпека

Кваліфікаційна робота

Виконав студент групи
КБм -21
О.В. Костюк

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

Кваліфікаційну роботу
допущено до захисту:

«____» _____ ___2023 р.

Завідувач кафедри
_____ **В.В.Яцків**

ТЕРНОПІЛЬ - 2023

Факультет комп'ютерних інформаційних технологій

Кафедра кібербезпеки
Освітній ступінь «магістр»
спеціальність: 125 – Кібербезпека
освітньо-професійна програма – Кібербезпека

ЗАТВЕРДЖУЮ
Завідувач кафедри

_____ В.В.Яцків
«____» _____ 2022 року

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

КОСТЮК ОЛЕКСАНДР ВАСИЛЬОВИЧ
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи:
Система автоматичного реагування на мережеві атаки / Automated Network Attack Response System
керівник роботи к.т.н., доцент Н.Г. Яцків

затверджені наказом по університету від 1 грудня 2022 року № 491

2. Строк подання студентом закінченої кваліфікаційної роботи 1 грудня 2023 р.

3. Вихідні дані до кваліфікаційної роботи: завдання на кваліфікаційну роботу студента, наукові статті, технічна література.

4. Основні питання, які потрібно розробити:
– визначити методи ідентифікації мережевих атак;
– аналізувати типи та поведінку зловмисних IP-адрес;
– дослідити методи моніторингу та аналізу мережевого трафіку;
– розробити алгоритми автоматичного реагування на мережеві атаки;
– розглянути впровадження системи у мережеву інфраструктуру;
– реалізація системи на основі Python та інших програмних інструментів.

5. Перелік графічного матеріалу у роботі:
– виявлення та аналіз зловмисних IP-адрес;
– архітектура системи автоматичного реагування;
– взаємодія компонентів системи;
– процеси моніторингу та аналізу мережевого трафіку;

– візуалізація реакції системи на мережеві атаки.

6. Консультанти розділів кваліфікаційної роботи

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

7. Дата видачі завдання 8 грудня 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строки виконання етапів кваліфікаційної роботи	Примітка
1	Аналіз предметної області	12.2022 р. – 03.2023 р.	
2	Теоретичні основи системи автоматичного реагування на мережеві атаки	03.2023 р. – 05.2023 р.	
3	Розробка системи автоматичного реагування на мережеві атаки	05.2023 р. – 11.2023 р.	

Студент _____ Костюк О.В.
(підпис)

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

АНОТАЦІЯ

Магістерська робота на тему "Система автоматичного реагування на мережеві атаки" зі спеціальності 125 - кібербезпека представляє комплексне дослідження в області кібербезпеки, розробки та впровадження інноваційних рішень для захисту мережевих систем. Робота включає детальний аналіз сучасних загроз кібербезпеці, методологію розробки системи автоматичного реагування, а також практичне застосування розробленої системи.

Робота складається з 80 сторінок, містить 41 ілюстрацій, 1 додаток і базується на 26 наукових джерелах. Основний акцент у роботі зроблено на розробку та імплементацію системи, яка здатна ідентифікувати та реагувати на мережеві атаки в реальному часі, забезпечуючи надійний рівень захисту кіберпростору.

Основна увага в роботі приділена розробці алгоритмів для ідентифікації та блокування потенційно шкідливих IP-адрес, а також створення комплексної інфраструктури для моніторингу та аналізу мережевого трафіку. Робота включає детальний опис методології розробки, використаних технологій, а також практичну реалізацію системи.

Ця магістерська робота може бути корисною для спеціалістів у галузі кібербезпеки, розробників безпекових систем, а також наукових дослідників, зацікавлених у розвитку методів захисту інформаційних систем від мережевих атак. Пропоновані підходи та рішення можуть бути використані як основа для подальшого розвитку та вдосконалення систем автоматичного реагування на кіберзагрози.

Ключові слова: КІБЕРБЕЗПЕКА, МЕРЕЖЕВІ АТАКИ, СИСТЕМА АВТОМАТИЧНОГО РЕАГУВАННЯ, ІДЕНТИФІКАЦІЯ ЗЛОВМИСНИХ IP, МОНІТОРИНГ МЕРЕЖІ.

ABSTRACT

The master's thesis on "Automated Network Attack Response System" in the specialty 125 - Cybersecurity is a comprehensive study in the field of cybersecurity, development and implementation of innovative solutions for the protection of network systems. The work includes a detailed analysis of modern cybersecurity threats, a methodology for developing an automatic response system, and the practical application of the developed system.

The work consists of 80 pages, contains 41 illustrations, 1 appendix and is based on 26 scientific sources. The main emphasis of the work is on the development and implementation of a system that is able to identify and respond to network attacks in real time, providing a reliable level of cyberspace protection.

The paper focuses on the development of algorithms for identifying and blocking potentially malicious IP addresses, as well as creating a comprehensive infrastructure for monitoring and analyzing network traffic. The paper includes a detailed description of the development methodology, technologies used, and the practical implementation of the system.

This master's thesis can be useful for cybersecurity professionals, security system developers, and researchers interested in developing methods to protect information systems from network attacks. The proposed approaches and solutions can be used as a basis for further development and improvement of automatic response systems to cyber threats.

Keywords: CYBERSECURITY, NETWORK ATTACKS, AUTOMATIC RESPONSE SYSTEM, IDENTIFICATION OF MALICIOUS INTRUDERS, NETWORK MONITORING.

ЗМІСТ

ВСТУП

Актуальність роботи. У епоху стрімкого розвитку цифрових технологій та постійного зростання кіберзагроз, питання ефективного захисту інформаційних систем стає все більш актуальним. Постійна еволюція кібератак вимагає не лише підвищеної уваги до захисту даних, але й створення більш витончених методів боротьби з кіберзлочинністю.

Сучасний світ кібербезпеки вимагає надзвичайно гнучких та адаптивних підходів до захисту цифрової інфраструктури. В цьому контексті, особлива увага приділяється не тільки захисту від уже відомих загроз, але й прогнозуванню та нейтралізації майбутніх атак. З цієї причини, розробка нових, ефективних систем автоматичного реагування на мережеві атаки є ключовим завданням для забезпечення безпеки в кіберпросторі [18].

Метою даної роботи є розробка новітнього алгоритму, здатного ефективно ідентифікувати та нейтралізувати потенційні кіберзагрози. Важливою частиною дослідження є використання інструментів з відкритим кодом, які забезпечують гнучкість, доступність та можливість спільного розвитку рішень в науковій спільноті. Впровадження таких інструментів не лише розширює можливості дослідження, але й сприяє глобалізації зусиль у сфері кібербезпеки.

Щоб досягнення поставленої мети передбачається виконання ряду завдань:

– детальний аналіз існуючих методів автоматичного реагування;

– розробка і тестування нових алгоритмів;

– оцінка їх ефективності у реальних умовах.

Об'єктом дослідження є процеси програмного та апаратного опрацювання даних у сфері кібербезпеки.

Предметом дослідження – методи та алгоритми, які використовуються для автоматичного реагування на мережеві атаки. Значна увага приділяється також аналізу потенційних слабких місць в існуючих системах безпеки, щоб забезпечити їх вдосконалення.

Наукова новизна роботи полягає у розробці інноваційного алгоритму, який відповідає сучасним вимогам кібербезпеки та вирізняється підвищеною ефективністю порівняно з існуючими рішеннями. Практична цінність одержаних результатів полягає у впровадженні розробленого алгоритму у сфері кібербезпеки, що сприятиме підвищенню загального рівня захищеності інформаційних систем.

Публікації та апробація результатів дослідження підтверджують їх актуальність та значимість у сучасному науковому дискурсі кібербезпеки. Використання інструментів з відкритим кодом не тільки відкриває широкі можливості для співпраці та обміну досвідом, але й сприяє залученню ширшого кола дослідників для спільної роботи над покращенням і оптимізацією кібербезпеки. Це, в свою чергу, не тільки підвищує ефективність розроблених рішень, але й сприяє швидкому поширенню та адаптації нових технологій у галузі.

Таким чином, вступ визначає широкий контекст та зазначає значення даного дослідження в сучасному світі кібербезпеки, акцентуючи на важливості розробки ефективних та інноваційних рішень з використанням відкритого програмного забезпечення. Дослідження також відкриває перспективи для подальших напрацювань у цій сфері, закладаючи фундамент для розвитку більш стійких та надійних систем кіберзахисту.

Публікації та апробація КР.

1. Костюк О.В. Побудова систем виявлення мережевих кіберзагроз. Матеріали науково-практична конференція молодих вчених, аспірантів та студентів «Кібербезпека та комп’ютерно-інтегровані технології» (КБКІТ - 2023), Тернопіль, 2023. – С.80-83.
2. Костюк О.В. Ключові аспекти та переваги централізованого збору та збереження журналів роботи інфраструктури. Матеріали науково-практичного симпозіуму «Захист інформації», Тернопіль, 2023. – С. 102-105.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Визначення ключових понять: мережева безпека, мережеві атаки

Мережева безпека є одним з найважливіших аспектів в інформаційних технологіях, що забезпечує захист даних та інфраструктури від несанкціонованого доступу, зловживань, змін або знищення. Це включає в себе використання різноманітних технологій та процесів для забезпечення цілісності, конфіденційності та доступності даних. Ефективна мережева безпека також вимагає розробки та впровадження комплексної політики безпеки, яка охоплює правила, процедури та контрольні заходи для запобігання та виявлення загроз [1].

Мережеві атаки можуть приймати різноманітні форми та використовувати різні методи для досягнення своїх цілей. Ці атаки можуть включати в себе розподілені атаки типу «відмова в обслуговуванні» (DDoS), фішинг, встановлення шкідливого ПЗ, викрадення даних та інші. Кожна з цих атак має свою специфіку та вимагає різних методів захисту. Наприклад, ДДоС-атаки вимагають систем захисту, які можуть фільтрувати великі об'єми трафіку, в той час як захист від фішингу більше зосереджується на освіті користувачів та використанні розширених технологій фільтрації електронної пошти.

З розвитком технологій мережеві атаки стають все більш складними та витонченими, що змушує фахівців з мережевої безпеки постійно оновлювати свої знання та інструменти для ефективного захисту. Наприклад, використання штучного інтелекту та машинного навчання у сфері мережевої безпеки стає все більш поширеним, оскільки ці технології можуть допомогти в автоматизації виявлення та реагування на загрози в реальному часі [2].

З урахуванням зростаючого числа загроз та викликів у цифровому світі, мережева безпека є ключовою складовою стратегії захисту будь-якої організації. Вона вимагає комплексного підходу, який поєднує технічні, організаційні та освітні заходи для забезпечення ефективного захисту від мережевих атак.

1.2 Розуміння індикаторів занепокоєння та компрометації

Індикатори компрометації (Indicators of Compromise, IoC) - це своєрідні "відбитки пальців" кібератак. ІОС можуть включати в себе наступне:

– IP-адреси, які відомі як шкідливі або пов'язані з атаками.

– Домени, які відомі як шкідливі або пов'язані з атаками.

– Порти, які відомі як шкідливі або пов'язані з атаками.

– Набори байтів, які відомі як шкідливі або пов'язані з атаками.

– Вразливості, які відомі як небезпечні або вже використовувані в атаках.

IoC є ключовими у виявленні та аналізі безпекових інцидентів, оскільки вони надають чіткі свідчення про порушення безпеки.

Індикатори компрометації зазвичай збираються під час аналізу безпекових інцидентів та використовуються для створення сигнатур або правил, що застосовуються в системах безпеки, таких як системи виявлення вторгнень (IDS) та системи запобігання вторгненням (IPS). Наприклад, якщо певна шкідлива програма використовує відомий IP-адрес сервера команд та управління (C&C), цей IP-адрес може бути використаний як IoC для блокування спроб зв'язку з цим сервером.

Ефективне використання IoC вимагає розуміння того, як зловмисники можуть змінювати свої методи, а також здатності швидко оновлювати та адаптувати системи безпеки до нових загроз. Це включає аналіз та інтерпретацію IoC в контексті специфіки організації та її мережевого середовища [3].

Крім того, існують спільноти та платформи обміну інформацією про загрози, такі як MISP (Malware Information Sharing Platform) та AlienVault OTX, де організації можуть обмінюватися IoC. Це дозволяє швидше реагувати на нові загрози, використовуючи зібрані дані від різних джерел.

Варто зазначити, що індикатори компрометації це уже про реактивний підхід. Тобто атака уже відбувається і потрібно діяти, щоб унеможливити або хоча б сповільнити атаку. Цей індикатор базується на внутрішніх даних, таких як журнали роботи систем чи інформація з SIEM. Проте, використовуючи

розвідувальні дані з відкритих джерел (OSINT), можна збирати інформацію з загальнодоступних ресурсів, щоб ідентифікувати потенційні кібератаки та слідкувати за можливими загрозами. Зібрані таким чином елементи називаються - індикатори занепокоєння.

Враховуючи, що IoC можуть швидко застарівати, важливо регулярно оновлювати їх та поєднувати з іншими стратегіями безпеки, такими як поведінковий аналіз та глибоке навчання систем безпеки, щоб створити більш комплексну та ефективну оборонну стратегію [4].

1.3 Аналіз інструментів та технологій відслідковування мережевих атак

Інструменти та технології, які використовуються для відстеження мережевих атак, представляють собою різноманітні системи та методи, які можна класифікувати за різними критеріями. Один з основних критеріїв – це тип виявлення атак, який використовується інструментами. Розуміння цих типів є важливим для вибору найбільш ефективних інструментів в залежності від конкретної ситуації або потреби в безпеці.

За типом виявлення інструменти для відстеження мережевих атак можна поділити на такі групи:

Сигнатурний аналіз – це найпоширеніший тип виявлення атак. Сигнатурний аналіз базується на використанні бази даних сигнатур, яка містить відомі шаблони шкідливого коду, IP-адреси, URL та інші характеристики, які пов'язані з відомими атаками. Коли система виявляє вхідні дані, які відповідають одній з цих сигнатур, вона генерує сповіщення про можливу атаку. Цей метод є ефективним для захисту від широко відомих і документованих загроз, але він може бути менш ефективним проти нових або складно модифікованих атак (рисунок 1.1).

Рисунок 1.1 – Архітектура методології на основі сигнатур

Поведінковий аналіз – це більш складний тип виявлення атак, ніж сигнатурний аналіз. Поведінковий аналіз є суттєвою частиною сучасних систем безпеки, зосереджуючись на ідентифікації аномалій у мережевому трафіку або поведінці систем, які можуть сигналізувати про несанкціоновану або шкідливу діяльність. Цей підхід значно відрізняється від традиційних методів, таких як сигнатурний аналіз, оскільки він не залежить від попередньо відомих шаблонів атаки, а замість цього фокусується на виявленні непередбачуваних або аномальних поведінкових моделей. Одним із ключових елементів поведінкового аналізу є використання алгоритмів машинного навчання, які можуть виявляти складні взаємозв'язки та аномалії, які можуть не бути помітні для людського ока або традиційних моніторингових інструментів. Машинне навчання може використовувати різні техніки, такі як спостереження, безспостереження або навчання з підкріпленням, для аналізу даних і визначення потенційно шкідливих або підозрілих поведінкових патернів (рисунок 1.2).

Рисунок 1.2 – Архітектура методології на основі поведінкового аналізу

Іншим поширеним критерієм класифікації інструментів для відстеження мережевих атак є рівень інтелектуальної обробки, який вони використовують.

За рівнем інтелектуальної обробки інструменти для відстеження мережевих атак можна поділити на такі групи: IDS та SIEM.

Системи виявлення вторгнень (IDS) є одними з найважливіших інструментів в арсеналі мережевої безпеки. Їх основна функція - моніторинг мережевого трафіку та системних активностей на предмет виявлення підозрілих дій, що можуть вказувати на спробу вторгнення або інші порушення безпеки.

IDS поділяються на два основних типи:

– Мережеві IDS (NIDS**)**: Вони аналізують вхідний та вихідний мережевий трафік на рівні всієї мережі. Вони ефективні для виявлення загроз, що виникають ззовні мережі. Наприклад, Snort є одним з найбільш популярних NIDS, що дозволяє виявляти різноманітні атаки, такі як порушення політик доступу, сканування портів, атаки типу "denial-of-service" та інші.

– Хостові IDS (HIDS): Вони встановлюються на конкретному комп'ютері або сервері та моніторять вхідний та вихідний трафік цього хоста, а також

системні журнали та активності. Прикладом HIDS може бути OSSEC, який здатний виявляти зміни в системних файлах, потенційні rootkits, а також підозрілі вхідні записи в системних журналах.

IDS використовують різні методи для виявлення атак:

– Сигнатурний аналіз: IDS порівнює поточний мережевий трафік з базою даних відомих підписів атак. Це ефективно проти відомих загроз, але може не виявляти нові або модифіковані атаки.

– Аналіз аномалій: IDS використовує моделі "нормальної" мережевої активності та сповіщає, коли виявляє аномалії. Це дозволяє виявляти невідомі або нові атаки, але може призводити до вищої кількості помилкових спрацьовувань.

– Методи засновані на стані (Stateful methods): IDS аналізує контекст мережевого трафіку, враховуючи поточний стан сесій та з'єднань. Це допомагає зрозуміти, чи є певна активність частиною законного мережевого обміну даними. Ці системи здатні відслідковувати тривалі діалоги між хостами, що дозволяє виявляти комплексні атаки, які відбуваються протягом тривалого часу. Деякі атаки, які не можуть бути виявлені шляхом аналізу окремих пакетів, можуть бути ідентифіковані за допомогою аналізу поведінки у межах сесії. Наприклад, атаки, які починаються зі сканування портів, за яким слідує експлуатація вразливостей і завершується встановленням бекдору, може бути виявлена саме за допомогою даного методу.

– Методи незалежні від стану (Stateless methods): використовуються в IDS для виявлення атак на основі аналізу окремих мережевих пакетів без врахування контексту або історії з'єднань. Ці системи зазвичай швидше обробляють трафік, оскільки не потребують збереження та аналізу додаткової інформації про стан з'єднання. Вони можуть ефективно виявляти відомі типи атак, але менш ефективні проти комплексних або багатоетапних загроз.

В ідеалі, сучасні IDS поєднують обидва підходи, використовуючи stateful аналіз для глибокого розуміння мережевих сесій та stateless аналіз для швидкого виявлення та реагування на атаки.

IDS використовуються для виявлення різноманітних загроз, включаючи:

– Спроби несанкціонованого доступу або вторгнення.

– Атаки типу "Denial-of-Service" (DoS) та "Distributed Denial-of-Service" (DDoS).

– Сканування мережі та портів.

– Використання відомих вразливостей.

– Аномалії в мережевому трафіку, що можуть вказувати на неправомірні дії.

IDS забезпечують важливий рівень захисту в мережевій безпеці, дозволяючи своєчасно реагувати на потенційні загрози. Важливо, щоб IDS належним чином налаштовувалися та регулярно оновлювалися для ефективного захисту від сучасних кіберзагроз.

Важливими прикладами є Snort, Suricata, та Bro (тепер званий Zeek). Коротко розглянемо їх розміщення, застосування та налаштування.

Snort є одним із найвідоміших мережевих IDS (NIDS) та може бути встановлений на будь-якому мережевому вузлі для моніторингу вхідного та вихідного трафіку. Його часто використовують у корпоративних мережах для виявлення атак, таких як порушення політик доступу, сканування портів, атаки типу "denial-of-service" та інші. Налаштування Snort вимагає визначення правил, що вказують, на які типи трафіку або діяльності слід звертати увагу. Конфігураційний файл Snort дозволяє адміністраторам налаштувати параметри мережі, правила обробки трафіку та відповіді на виявлені загрози. Він також підтримує велику кількість плагінів та доповнень для розширення своєї функціональності.

Suricata є високопродуктивним NIDS, IPS та мережевим моніторингом безпеки. Його можна встановити на стратегічних точках мережі для аналізу

трафіку та виявлення шкідливих дій. Suricata здатна обробляти великі об'єми трафіку, використовуючи багатопоточність, і є ефективною у багатоядерних системах. Адміністратори можуть налаштувати Suricata за допомогою YAML-конфігураційних файлів, де визначаються правила обробки трафіку, сигнатури атак та інші параметри. Підтримує правила від Snort, що полегшує міграцію та інтеграцію з існуючими системами.

Bro, нині відомий як Zeek, є потужною платформою для моніторингу мережевого трафіку, яка зосереджена на безпеці та продуктивності. Він може бути використаний для детального логування мережевих з'єднань, скриптів для аналізу трафіку, а також як NIDS. Zeek використовує скриптову мову для опису логіки моніторингу мережі та реагування на події, що надає велику гнучкість. Він дозволяє адміністраторам створювати детальні скрипти для виявлення складних атак, аналізувати протоколи на високому рівні та виконувати глибокий аналіз мережевого трафіку [19].

Кожен з цих інструментів має свої унікальні особливості та краще підходить для певних сценаріїв використання. Вибір між ними часто залежить від конкретних потреб організації, розміру мережі, доступних ресурсів та рівня технічної експертизи.

Системи управління інцидентами безпеки (SIEM). Зважаючи на обширність систем та їхню різноманітність, з певного моменту стає неможливий ручний перегляд журналів роботи та сповіщень безпеки. Щоб це покращити та мати більший огляд безпеки використовуються Security Information and Event Management (SIEM). SIEM-системи представляють собою комплексне рішення для управління інформацією та подіями в області безпеки, що забезпечує централізований огляд безпеки в реальному часі шляхом збору, аналізу та представлення даних з різних джерел. Ці системи збирають логи та інші дані безпеки з різноманітних джерел, включаючи мережеві пристрої, сервери, IDS/IPS, фаєрволи, антивірусні системи тощо (рисунок 1.3).

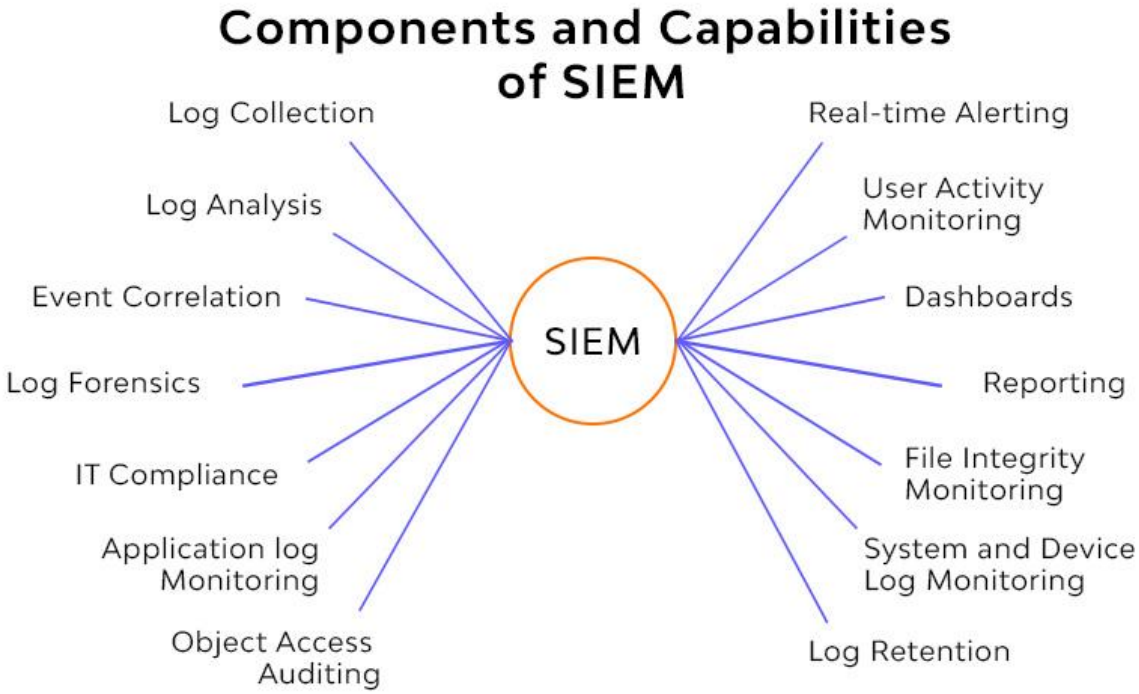


Рисунок 1.3 – Компоненти та функції SIEM

Використовуючи потужні алгоритми кореляції, SIEM-системи аналізують зібрані дані на предмет виявлення патернів, що можуть вказувати на шкідливу або підозрілу діяльність. При виявленні потенційних загроз або інцидентів система генерує сповіщення та може автоматизувати певні дії реагування. Також ця система надає зручні інструменти для генерації звітів та інформаційних панелей, що дозволяє керівникам безпеки та IT-командам швидко оцінювати стан безпеки [23].

Приклади SIEM-систем:

– Splunk - відомий своєю гнучкістю та потужною системою збору та аналізу даних. Splunk дозволяє користувачам створювати складні запити для аналізу логів та генерувати детальні звіти.

– IBM QRadar - ця система забезпечує широкий спектр можливостей з аналізу даних безпеки, включаючи кореляцію подій, аналіз аномалій та засоби виявлення загроз на основі штучного інтелекту.

– LogRhythm - поєднує традиційні функції SIEM з розширеними можливостями для аналізу мережевого трафіку та виявлення аномалій, а також має вбудовані інструменти для управління інцидентами.

Впровадження SIEM-системи є ключовим елементом комплексної стратегії безпеки будь-якої організації, оскільки вона забезпечує широкий огляд безпекового стану та дозволяє оперативно реагувати на інциденти. Це не тільки покращує захист інформаційних активів, а й допомагає організаціям дотримуватися нормативних вимог та стандартів у сфері безпеки інформації.

Варто відмітити, також, інструменти мережевого моніторингу, такі як Wireshark та tcpdump. Вони разом із системи виявлення/запобігання вторгненням (IDS/IPS) виконують різні, але доповнюючі функції у сфері мережевої безпеки. Ці інструменти використовуються для детального аналізу та діагностики мережевого трафіку. Вони забезпечують глибокий аналіз пакетів для розслідування конкретних інцидентів або збору доказів. Використання інструментів мережевого моніторингу частіше використовується для аналізу після події, де вони допомагають у розслідуванні та аналізі інцидентів. За допомогою детального аналізу трафіку, включаючи інспекцію окремих пакетів, можна глибше зрозуміти природу мережевої активності. Тому проаналізуємо ці інструменти більш детально.

Wireshark є одним з найпопулярніших інструментів для аналізу мережевого трафіку. Це безкоштовний та відкритий інструмент, який дозволяє захоплювати та детально аналізувати трафік у реальному часі або збережений у файлі. Він використовується для виявлення аномалій у мережі, діагностики проблем, аналізу протоколів мережевого рівня та виявлення спроб несанкціонованого доступу. За допомогою наявної графічної оболонки дозволяє легко налаштувати фільтри для захоплення певного трафіку (рисунок 1.4).

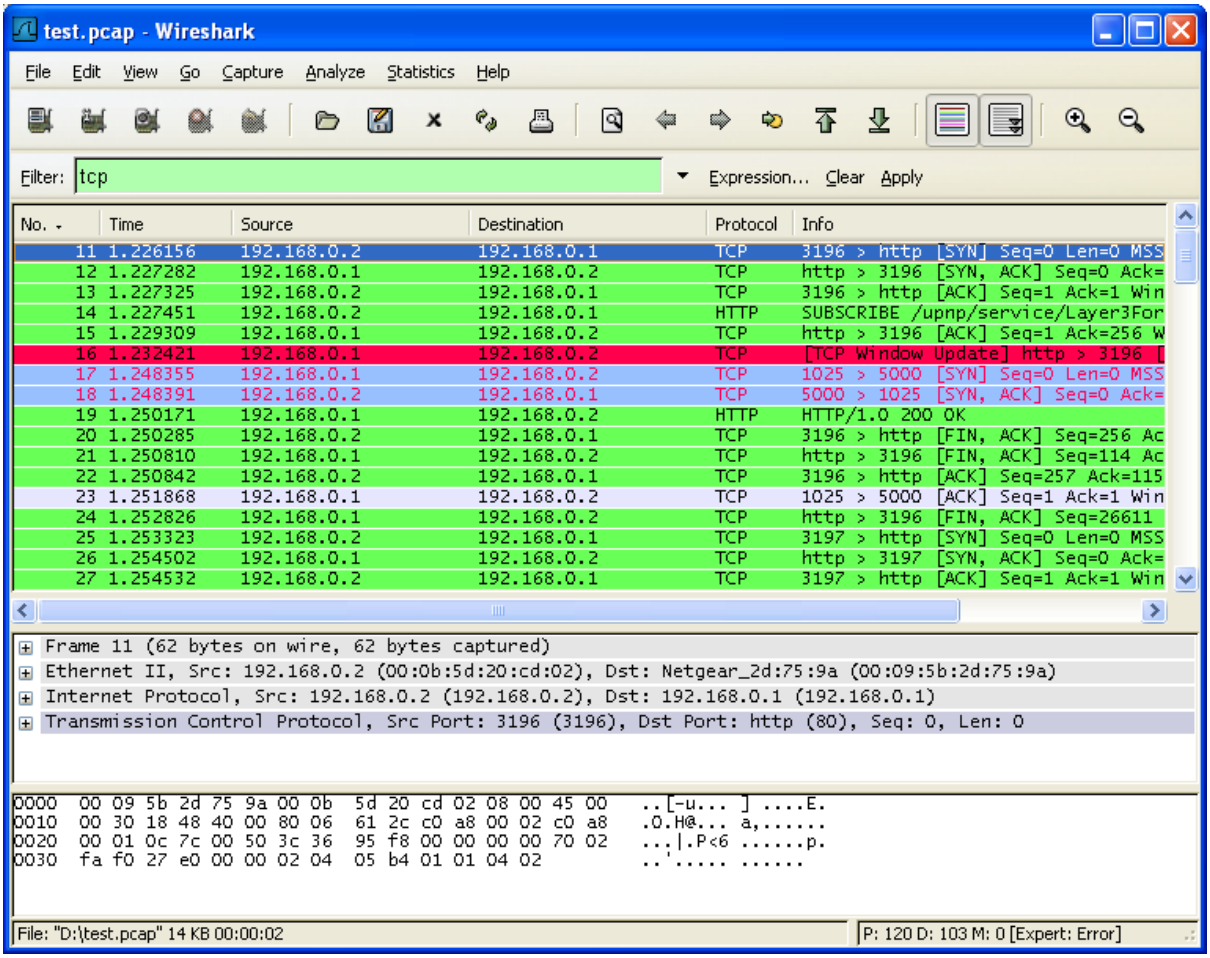


Рисунок 1.4 – Приклад використання фільтрів. Фільтрація за протоколом TCP

Користувачі можуть використовувати функцію "Capture Filters" для збору пакетів за певними критеріями та "Display Filters" для детального аналізу захоплених даних. Wireshark дозволяє аналізувати різні протоколи та їх взаємодію, що є корисним для виявлення незвичайних патернів або неправильної поведінки протоколів.

tcpdump – це потужний інструмент командного рядка для аналізу мережевого трафіку, який доступний в більшості UNIX-подібних операційних системах. Він дозволяє захоплювати та аналізувати трафік, що проходить через мережевий інтерфейс, та є корисним для виявлення аномалій, дослідження мережевих проблем та моніторингу безпеки. tcpdump працює з командного рядка і дозволяє використовувати різноманітні опції для фільтрації та аналізу трафіку (рисунок 1.5).

```
:~$ sudo tcpdump -s 0 -v -n -l | egrep -i "POST /|GET /|Host:"

tcpdump: listening on enp7s0, link-type EN10MB (Ethernet), capture size 262144 bytes
        POST /wp-login.php HTTP/1.1
        Host: dev.example.com
        GET /wp-login.php HTTP/1.1
        Host: dev.example.com
        GET /favicon.ico HTTP/1.1
        Host: dev.example.com
        GET / HTTP/1.1
        Host: dev.example.com
```

Рисунок 1.5 – Приклад використання фільтрів. Отримання URL-адреси HTTP-запитів

Команди можна налаштувати для захоплення пакетів за IP-адресами, портами, типами протоколів та іншими параметрами. Хоча tcpdump менш інтуїтивно зрозумілий, ніж Wireshark, він є дуже потужним інструментом, особливо для автоматизації мережевого моніторингу та аналізу на серверах без графічного інтерфейсу [6].

Обидва інструменти, Wireshark та tcpdump, є важливими для фахівців з мережевої безпеки. Вони дозволяють глибоко зануритися у деталі мережевого трафіку, виявляти підозрілі або аномальні дії, що можуть вказувати на наявність безпекових інцидентів чи потенційних вразливостей. Використання цих інструментів вимагає певного рівня технічної експертизи, але вони надають безцінний внесок у діагностику та відстеження стану мережевої безпеки.

1.4 Платформи для обміну інформацією про загрози

Перед розглядом платформ для обміну інформацією про загрози, важливо зрозуміти концепцію "піраміди болі" у кібербезпеці. Ця концепція відіграє ключову роль у визначенні, який тип інформації про загрози є найціннішим та найбільш ефективним для захисту від кібератак. "Піраміда болі" ілюструє,

наскільки складно для зловмисників змінювати різні елементи їхніх атак, починаючи від простих хешів до складних тактик та процедур (рисунок 1.6) [5].

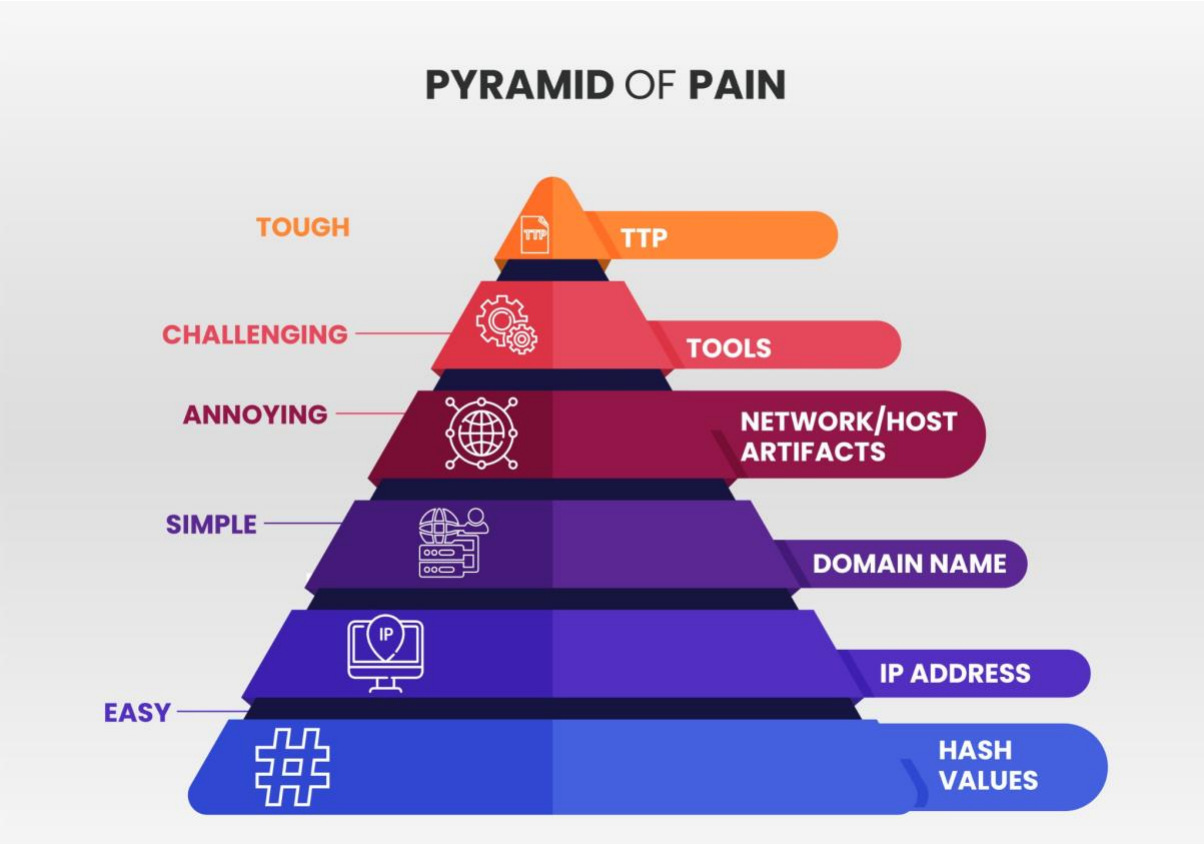


Рисунок 1.6 – Піраміда болю

"Піраміда болі", розроблена Девідом Бьянко, представляє ієрархію індикаторів компрометації (IoC) та інших атрибутів кіберзагроз. Вона ранжує типи IoC залежно від того, наскільки важко зловмисникам змінити їх, та наскільки це "болісно" для них.

Рівні піраміди:

1. Хеші. Легко змінити, найменш болісно для зловмисників.
2. IP-адреси. Трохи важче змінити, але все ще не дуже болісно.
3. Доменні імена. Вимагають більше зусиль для зміни.
4. Мережеві артефакти. Наприклад, формати відправлення даних, які важче змінювати.

5. Характеристики використаних інструментів. Ще складніше для зловмисників змінити.

6. Тактика, техніка та процедури (TTP). Найболісніше для зловмисників змінювати, оскільки вимагає значних змін у їхній поведінці та методах.

Розуміючи цю ієрархію, можна ефективніше використовувати платформи обміну інформацією про загрози. Такі платформи дозволяють організаціям обмінюватися інформацією, що покриває різні рівні піраміди, від базових даних до більш складних аналітичних знахідок. Цей обмін допомагає організаціям швидше адаптуватися до нових загроз та посилювати свої оборонні стратегії, виходячи з актуальних даних та інтелектуального аналізу.

Тепер, коли ми розглянули значення "піраміди болі", можемо перейти до розгляду платформ обміну інформацією про загрози, які є невід'ємною частиною стратегії кібербезпеки будь-якої організації.

Платформи для обміну інформацією про загрози є ключовими інструментами у сфері кібербезпеки, що дозволяють організаціям ділитися даними про виявлені загрози, включаючи індикатори компрометації (IoC), тактику, техніки та процедури (TTP), що використовуються зловмисниками. Це сприяє колективній обороні, дозволяючи організаціям більш швидко реагувати на нові та еволюціонуючі загрози [25].

Основні характеристики:

– Обмін інформацією про загрози: платформи забезпечують механізм для швидкого та ефективного обміну інформацією про загрози між різними організаціями та спільнотами.

– Інтеграція з іншими інструментами безпеки: багато з цих платформ можуть інтегруватися з існуючими інструментами безпеки, такими як IDS/IPS, SIEM та іншими, для автоматичного оновлення баз даних загроз та поліпшення виявлення та реагування.

– Спільнота та співпраця: платформи надають можливість для співпраці між різними секторами та спільнотами, що сприяє обміну кращими практиками та спільному вирішенню проблем безпеки.

Приклади платформ:

– MISP (Malware Information Sharing Platform & Threat Sharing) – це відкрите програмне забезпечення, що дозволяє організаціям ефективно ділитися, зберігати та кореляційно аналізувати Інформацію про Загрози. MISP особливо корисний для співпраці на міжорганізаційному рівні (рисунок 1.7).

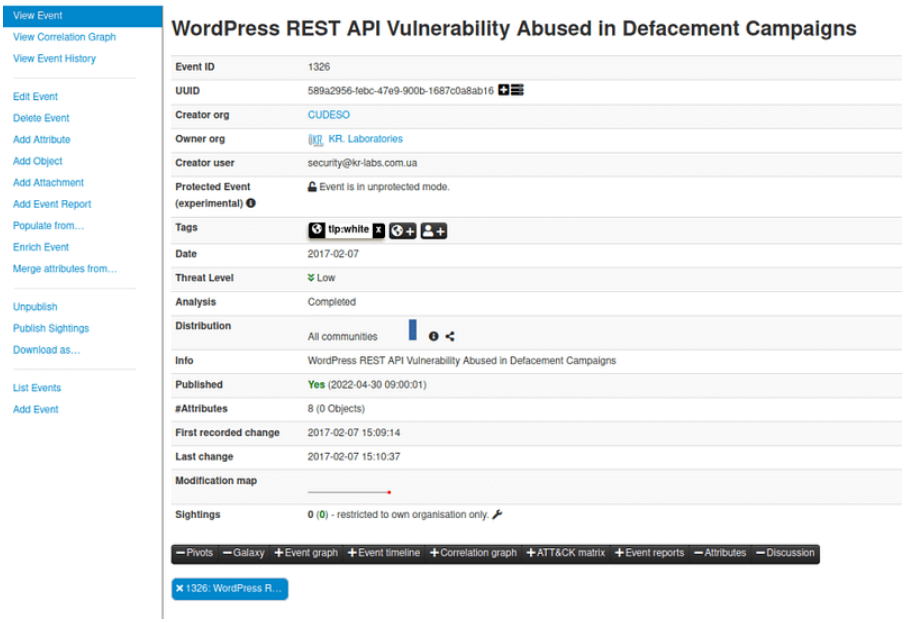


Рисунок 1.7 – Приклад відображення ідекаторів компрометацій на сайті MISP

– AlienVault OTX (Open Threat Exchange) - це глобальна спільнота, сприяє співпраці та обміну інформацією про кіберзагрози серед аналітиків безпеки, дослідників та IT-професіоналів. Спільнота надає унікальну можливість для своїх учасників публікувати, обговорювати та аналізувати останні дані про кіберзагрози в режимі реального часу, що дозволяє швидко реагувати на нові виклики у сфері кібербезпеки. Вона дозволяє користувачам публікувати та обговорювати дані про загрози в режимі реального часу (рисунок 1.8).

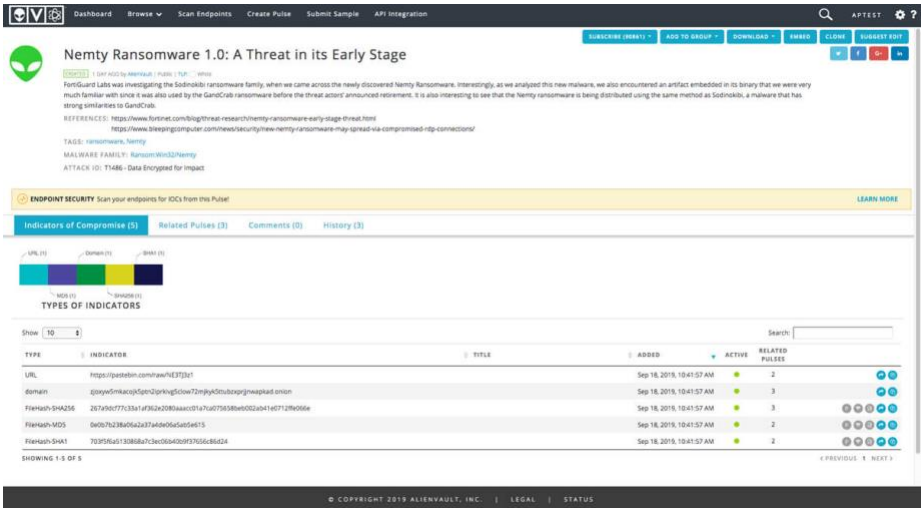


Рисунок 1.8 - Приклад відображення ідекаторів компрометацій на сайті AlienVault

Обмін інформацією про загрози дозволяє організаціям оперативно адаптуватися до нових та розвиваючихся загроз. Це забезпечує доступ до свіжих даних про загрози та IoC, що може бути негайно використано для підвищення захисту. Наприклад, якщо одна організація виявляє новий вид шкідливого ПЗ або метод атаки, вона може швидко поділитися цією інформацією через платформу, дозволяючи іншим організаціям вжити заходів для захисту від цієї загрози [7].

Крім того, платформи для обміну інформацією про загрози сприяють розвитку спільноти кібербезпеки, сприяючи співпраці, розподілу ресурсів та обміну знаннями між фахівцями з усього світу. Таким чином було розглянуто різноманітні інструменти та технології, які відіграють важливу роль у виявленні та аналізі мережевих атак. У цілому, ці інструменти та платформи займають важливу роль у стратегії кібербезпеки будь-якої організації. Вони не тільки допомагають у виявленні та реагуванні на мережеві атаки, але й сприяють розумінню трендів та змін у ландшафті кіберзагроз [13].

2 ТЕОРЕТИЧНІ ОСНОВИ СИСТЕМИ АВТОМАТИЧНОГО РЕАГУВАННЯ НА МЕРЕЖЕВІ АТАКИ

2.1 Концептуальні основи та архітектура системи

В даному розділі розглядаються механізми побудови та загальна архітектура систем автоматичного реагування на мережеві атаки. Система автоматичного реагування на мережеві атаки розробляється з метою не тільки виявлення загроз, але й оперативного втручання для мінімізації їх впливу на інформаційну інфраструктуру. Основні принципи таких систем включають:

– Проактивний моніторинг. Безперервне спостереження за мережевим трафіком та системними журналами для виявлення потенційних загроз.

– Аналіз даних та виявлення загроз. Використання різноманітних методів, від сигнатурного аналізу до поведінкового аналізу, для ідентифікації шкідливих або підозрілих активностей.

– Автоматизація реагування. Здатність системи автоматично реагувати на виявлені загрози, наприклад, блокуванням IP-адрес або ізоляцією скомпрометованих систем.

– Адаптивність та навчання. Важливість адаптації до змін у ландшафті загроз та використання результатів попередніх атак для підвищення ефективності виявлення та реагування.

Розробка архітектури системи автоматичного реагування на мережеві атаки потребує глибокого розуміння потенційних загроз та вимог до безпеки відповідної мережевої інфраструктури. Цей процес включає ряд ключових етапів та компонентів:

– Сенсори та збірники даних. Критично важливі елементи для забезпечення точного збору інформації про мережеву активність. Вони можуть бути розміщені на стратегічно важливих точках в мережі, наприклад, на кордонах мережі або в центральних вузлах, де вони можуть ефективно захоплювати трафік та журнали активності.

– Центральний аналітичний модуль – ядро системи, яке використовує складні алгоритми та механізми аналізу для ідентифікації підозрілих або шкідливих патернів. Цей модуль може включати машинне навчання та штучний інтелект для підвищення точності виявлення загроз.

– Модуль рішень та реагування. Після виявлення потенційної загрози, цей модуль активує відповідні механізми реагування. Це може бути автоматичне блокування виявлених загроз або генерація сповіщень для подальшого ручного втручання.

– Інтерфейс користувача та звітність забезпечують зручний доступ до інформації про стан мережевої безпеки, історії інцидентів та поточних загроз. Це дозволяє адміністраторам безпеки та мережевим інженерам швидко оцінювати ситуацію та вживати необхідних заходів.

– Інтеграція з іншими системами безпеки є важливою характеристикою, що забезпечує гнучкість та ефективність системи. Інтеграція з SIEM-системами, IDS/IPS та іншими захисними механізмами дозволяє створити єдину об'єднану систему захисту.

Під час розробки архітектури системи важливо враховувати не тільки поточні, але й майбутні потреби організації. Це включає можливість масштабування системи для відповідності зростанню обсягу мережевого трафіку та збільшенню кількості кінцевих точок. Крім того, важливо забезпечити високий рівень гнучкості та налаштування системи, щоб вона могла адаптуватися до змін у ландшафті загроз та нових вимог до безпеки.

Ефективна система автоматичного реагування на мережеві атаки - це більше, ніж просто набір технічних інструментів; це стратегічний компонент у всебічному підході до управління мережевою безпекою. Її успішність залежить від правильного вибору технологій, ефективного планування архітектури та постійного оновлення відповідно до розвитку кіберзагроз [7].

2.2 Принципи роботи та роль сенсорів у системі

Сенсори в системі автоматичного реагування на мережеві атаки є критичними компонентами, які відповідають за збір даних про активність мережі та систем. Вони діють як первинні точки даних для подальшого аналізу та ідентифікації потенційних загроз. Сенсори можуть бути встановлені на різних рівнях мережевої інфраструктури, включаючи край мережі (периферію), кордони мережі та всередині мережі. Вони функціонують за наступними принципами:

1. Безперервний Моніторинг. Сенсори безперервно моніторять мережевий трафік та системні журнали, фіксуючи всі важливі події та активності.

2. Збір Даних. Вони збирають різноманітні типи даних, включаючи мережеві пакети, метадані трафіку, системні журнали, хеши файлів тощо.

3. Попередня Обробка. Важливо не тільки збирати дані, а й ефективно їх передавати до центральної системи для аналізу. В деяких випадках сенсори можуть виконувати первинну обробку даних, щоб зменшити обсяг переданої інформації та підвищити ефективність аналізу.

Сенсори відіграють ключову роль у виявленні та реагуванні на мережеві атаки. Це перша лінія оборони в системі автоматичного реагування. Вони допомагають виявляти аномалії в мережевому трафіку, такі як незвичайні обсяги трафіку, підозрілі патерни з'єднань або спроби використання відомих вразливостей. З їх допомогою можна виявляти конкретні шкідливі активності, використовуючи сигнатурний або поведінковий аналіз, в інших випадках вони надають цінні дані для глибокого аналізу й кореляції, які виконуються в рамках центрального аналітичного модуля системи [22].

Ефективність системи автоматичного реагування на мережеві атаки значною мірою залежить від якості та точності даних, що надходять від сенсорів. Відповідний вибір, розміщення та налаштування цих сенсорів є ключовими для успішного виявлення та реагування на мережеві загрози.

Зазвичай розрізняють різни типу сенсорів від цього залежить, які дані буде отримано для обробки.

Операційна система Windows/Linux/Unix/MacOS може слугувати джерелом необхідних даних. Для прикладу, із Windows зазвичай збирають дані подій Windows, системних журналів та інших моніторингових інструментів, що вбудовані в операційну систему. Ці інструменти можуть включати спеціалізоване ПЗ для моніторингу системних викликів, реєстру Windows, а також активності файлової системи, прикладом такого ПЗ є утиліти Sysinternals. Значну частину інформації про роботу мережі можна отримати з Windows, якщо вона використовується на підприємстві, як сервер інфраструктури. Для прикладу RADIUS або DNS сервер. Журнали роботи цих сервісів надзвичайно важливі для аналізу наявності зловмисних дій в корпоративній мережі [21].

У Linux/Unix середовищах сенсори часто зосереджені на зборі системних журналів через Syslog, моніторингу активності ядра та мережевих інтерфейсів. Додатково можуть використовуватися інструменти, такі як Auditd, для детального моніторингу системних подій.

Наступним важливим джерелом даних є мережеве обладнання. У ролі сенсорів, мережеві маршрутизатори та комутатори виконують ключову функцію у зборі даних про стан та активність мережі. Вони безпосередньо відстежують трафік, що проходить через них, дозволяючи збирати цінні дані про пропускну спроможність мережі та потоки даних. Протоколи, такі як SNMP (Simple Network Management Protocol) та NetFlow, використовуються для збору детальної статистики та інформації про стан мережевого обладнання, можуть виявляти незвичайні або підозрілі зміни в мережевому трафіку, які вказують на потенційні загрози. Брандмауери та системи виявлення/запобігання вторгненням (IDS/IPS) функціонують як вбудовані сенсори в мережевій інфраструктурі. Ці системи безперервно моніторять мережу на предмет спроб вторгнення або шкідливих активностей, забезпечуючи надійний збір даних про можливі безпекові інциденти, та документують конкретні деталі зафіксованих спроб вторгнення,

включаючи тип атаки, застосовані методи та потенційні IP-адреси нападників. Зібрана інформація стає ключовою для подальшого аналізу стану безпеки та розробки відповідних стратегій реагування на загрози.

Мережеве обладнання, як маршрутизатори, комутатори, брандмауери та IDS/IPS, ефективно виконують роль сенсорів у системах автоматичного реагування на мережеві атаки. Вони забезпечують цінну інформацію про мережевий трафік та безпекові події, що є невід'ємною частиною комплексного моніторингу та захисту мережі.

Виконання збору даних із описаних вище сенсорів, відбувається через налаштування необхідної інфраструктури. Зокрема, потрібно інсталювати та налаштувати агенти збору журналів та механізм їх перенаправлення до систем збору й обробки цих даних. Для цієї цілі варто використати рішення ELK, а саме елемент Beats.

Beats є частиною Elastic Stack (раніше відомий як ELK Stack) і представляє собою набір легких, однофункціональних агентів для збору різних типів даних. Кожен "beat" спеціалізується на конкретному типі даних. Наприклад, Filebeat збирає журнали файлів, Metricbeat збирає метрики машин та систем, Packetbeat аналізує мережевий трафік і так далі.

Beats використовуються для збору даних у великих і розподілених системах, де потрібно ефективно збирати та відправляти дані до центрального репозиторію, як Elasticsearch або Opensearch. Вони особливо корисні в розподілених системах, де потрібен легкий механізм збору даних з різних вузлів. Beats є ідеальним вибором у ситуаціях, де потрібно зібрати специфічні типи даних з різних джерел із мінімальним впливом на продуктивність системи.

Зважаючи на те, що beats це агенти, які інсталюються в операційну систему, то використати їх на мережевому обладнанні неможливо. Для цього типу сенсорів ліпше використати Syslog. Syslog є стандартним протоколом для системного логування в Unix-подібних системах. Він дозволяє збирати журнальні повідомлення від різних джерел, таких як сервери, мережеві пристрої,

та системні програми. Syslog пропонує стандартизований формат для журналів, що полегшує централізований збір та аналіз. Він часто використовується для збору системних журналів у великих мережах і інфраструктурах. Це включає сервери, мережеві пристрої та інші компоненти інфраструктури та є оптимальним вибором, коли потрібно стандартизоване рішення для збору системних журналів у великомасштабних і різноманітних мережевих середовищах.

Beats простіші у встановленні та налаштуванні, особливо в середовищах, де вже використовується Elastic Stack. Syslog є більш традиційним і гнучким рішенням, що підходить для широкого спектру середовищ та обладнання. Якщо потрібно зібрати дуже специфічні типи даних, то Beats може бути кращим вибором завдяки своїй спеціалізованій природі. Натомість Syslog ідеально підходить для використання на мережевому обладнанні [10].

2.3 Система збору та обробки журналів

В сучасному світі кібербезпеки, де організації щодня стикаються з величезними обсягами даних, системи збору та обробки журналів відіграють ключову роль. Вони не просто збирають і зберігають інформацію, але й перетворюють великі масиви даних на зрозумілу, структуровану і проаналізовану інформацію. Це включає в себе все: від базових системних журналів до складних метрик процесів та мережевого трафіку. Завдяки цим системам, організації здатні виявляти, аналізувати та реагувати на безпекові загрози набагато ефективніше. Системи, як Logstash та OpenSearch, є критично важливими в контексті збору, обробки та аналізу великих обсягів даних журналів. Ці системи дають змогу ефективно обробляти, фільтрувати та аналізувати дані, які надходять з різних джерел, що є ключовим для забезпечення безпеки мережі.

Системи, як Logstash та OpenSearch, є критично важливими в контексті збору, обробки та аналізу великих обсягів даних журналів. Ці системи дають

змогу ефективно обробляти, фільтрувати та аналізувати дані, які надходять з різних джерел, що є ключовим для забезпечення безпеки мережі.

Logstash, як частина Elastic Stack, служить як потужний інструмент для збору та обробки даних. Його головна задача - збирати дані з різноманітних джерел та перетворювати їх у структурований формат для подальшого аналізу (рисунок 2.1).

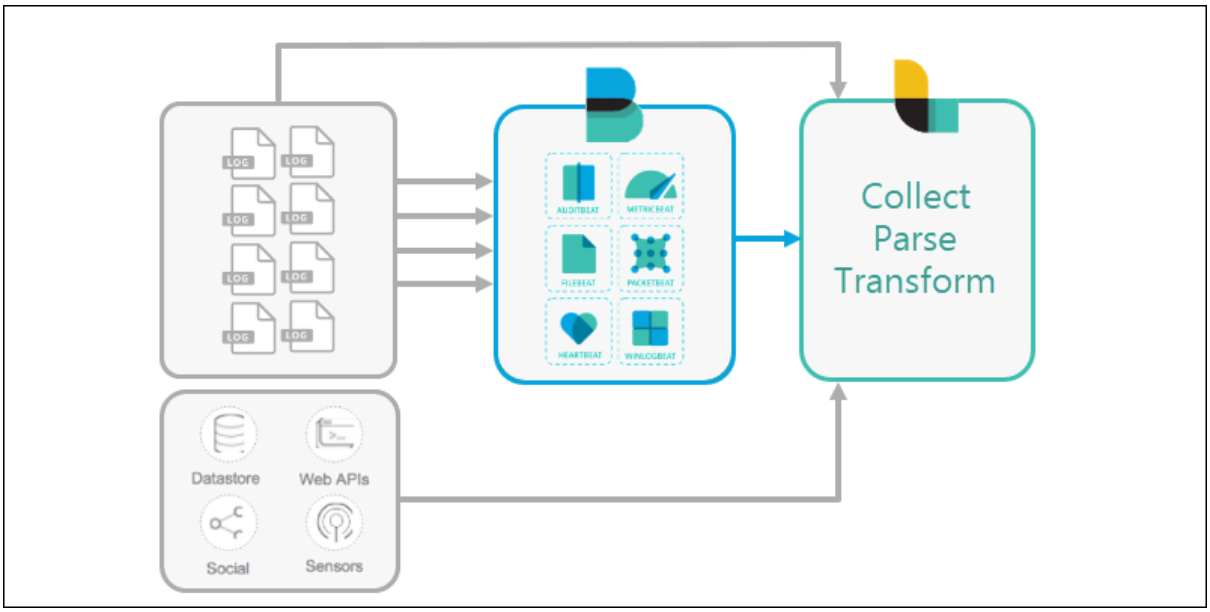


Рисунок 2.1 – Logstash. Отримання даних з різних джерел.

Він здатний обробляти дані з широкого спектру джерел, включаючи системні журнали, потоки даних від мережевих пристроїв, застосунків, баз даних тощо. Архітектура Logstash зазвичай включає в себе розподілені агенти, які збирають дані та передають їх до централізованого вузла обробки. Це дозволяє гнучко масштабувати систему відповідно до потреб організації. Система підтримує розширену конфігурацію фільтрів та плагінів для індивідуалізації обробки даних, що робить її надзвичайно гнучкою для різних сценаріїв використання. Як приклад можна підключити AbuseDB, для перевірки IP-адреси на зловмисність.

OpenSearch, як альтернатива Elasticsearch, є високопродуктивною системою для зберігання, пошуку та аналізу великих обсягів даних. Він

забезпечує швидкий доступ та аналіз зібраних даних. Використовується для зберігання та індексації даних, що прибувають з Logstash, дозволяючи користувачам здійснювати складні запити та отримувати швидкі відповіді. З особливостей, в порівнянні з Elasticsearch, надає безкоштовний функціонал машинного навчання для обробки журналів та виявлення аномалій або загроз. Система підтримує масштабованість та розподілене зберігання, що дозволяє ефективно обробляти великі обсяги даних.

Ефективне застосування Logstash та OpenSearch у сфері кібербезпеки та аналізу даних вимагає докладного налаштування та оптимізації для досягнення оптимальної продуктивності. Ретельно налаштовані, ці інструменти забезпечують високу продуктивність системи, що критично важливо для обробки великих обсягів даних. Масштабованість дозволяє системам адаптуватися до змінних потреб і обсягів даних, а точність і своєчасність обробки даних є фундаментальними для забезпечення ефективної аналітики та виявлення загроз, що дозволяє реагувати на кіберзагрози вчасно та ефективно.

Розробка механізму отримання та обробки журналів, які надсилаються з сенсорів, потребує вірного налаштування Logstash. Потрібно налаштувати конфігурацію вводу (input), фільтрацію та обробку, а також виведення (output) даних у сховище даних. Для прикладу, буде розглянуто налаштування отримання та обробки журналів мережевого екрану маршрутизатора Mikrotik. Щоб досягти бажаного результату, потрібно виконати наступні дії:

1. Увійти на маршрутизатор Mikrotik RouterOS.
2. Налаштувати систему журналювання, перейшовши до System > Logging. Далі додати нове правило журналювання, вказавши remote як дію та вказавши IP-адресу та порт сервера Logstash.
3. На сервері Logstash створити новий файл /etc/logstash/conf.d/mikrotik.conf та внести у нього усю необхідну конфігурацію (рисунок 2.2).

```
input {
  syslog {
    port => 5140
  }
}

filter {
  if [message] =~ /firewall/ {
    grok {
      match => { "message" => "%{SYSLOGTIMESTAMP:timestamp} %{SYSLOGHOST:host} %{DATA:firewall_action}
                              src=%{IP:src_ip} dst=%{IP:dst_ip} %{GREEDYDATA:message_details}" }
    }

    date {
      match => [ "timestamp", "MMM  d HH:mm:ss", "MMM dd HH:mm:ss" ]
    }

    geoip {
      source => "src_ip"
    }
  }
}

output {
  opensearch {
    hosts => ["http://localhost:9200"]
    index => "mikrotik-logs-%{+YYYY.MM.dd}"
    user => "opensearch_username"
    password => "opensearch_password"
  }
}
```

Рисунок 2.2 – Конфігурація logstash для розбору журналу MikroTik Firewall

4. Після цього необхідно перезапустити службу Logstash.

5. Переглянути отримані дані можна за допомогою графічної оболонки Opensearch Dashboard або через Opensearch API (рисунок 2.3), в індексі mikrotik-logs-<поточна дата>.

```
curl -X GET "http://localhost:9200/mikrotik-logs-*/_search?pretty" \
     -H 'Content-Type: application/json' -d'
{
  "query": {
    "match_all": {}
  },
  "size": 10
}'
```

Рисунок 2.3 – Приклад отримання даних з Opensearch через REST API

Разом Logstash та OpenSearch формують міцну основу для комплексного моніторингу та аналізу даних, що є життєво необхідним для ефективної роботи

систем автоматичного реагування на мережеві атаки. Їхня здатність адаптуватися до різноманітних вимог та інтеграція з іншими інструментами і технологіями робить їх незамінними в складному світі сучасної кібербезпеки.

2.4. Методи аналізу IP адрес та інтеграція з індикаторами безпеки

Аналіз IP адрес є ключовим елементом в системах кібербезпеки. Він включає в себе використання різноманітних методик для виявлення потенційних загроз, в тому числі використання даних з відкритих джерел індикаторів компрометації (IoC). Ці методи допомагають спеціалістам з безпеки виявляти і протидіяти атакам більш ефективно. Перший крок в аналізі IP-адрес - це визначення, які з них можуть бути потенційно небезпечними. Існують різні аналітичні методи аналізу IP адрес. Розглянемо кілька найбільш популярних.

Сигнатурний аналіз, заснований на порівнянні виявлених IP-адрес із відомими базами даних шкідливих адрес. Цей традиційний метод ефективний проти відомих загроз, але менш ефективний для виявлення нових або модифікованих атак [16].

Поведінковий аналіз, зосереджений на виявленні аномальної активності, що відрізняється від типових шаблонів мережевої поведінки. Включає аналіз мережевих патернів і трафіку для виявлення підозрілих дій.

Географічний аналіз, полягає у визначенні фізичного розташування IP-адреси, що може надати важливу інформацію щодо потенційних ризиків, особливо якщо активність відбувається з географічно підозрілих регіонів.

Репутаційний аналіз включає оцінку довіри до IP-адреси на основі її попередньої активності і взаємодій. Репутаційний аналіз може виявити потенційно шкідливі адреси на основі їх історії поведінки [12].

Одним з потужних інструментів в аналізі кіберзагроз є кореляція IP-адрес з Індикаторами компрометацій. Такі індикатори - це дані, які зазвичай отримуються з публічних підписок, таких як MISP і AlienVault. Вони містять інформацію про відомі загрози та підозрілі IP-адреси. Таким чином, щоб використовувати IoC для проактивного аналізу потрібно підключитися до

кількох публічних або пропрієтарних підписок та отримувати оновленя інформації про ідентифікатори компрометацій. Далі ці дані потрібно порівнювати з активністю корпоративної мережі. Використання інструментів для автоматизованої кореляції IoC з IP-адресами в корпоративній мережі, допоможе виявити відповідність між підозрілими адресами та відомими загрозами. Таким чином налаштувавши сповіщення, а в більш продвинутому варіанті і реакцію із автоматичного переналаштування програмно-апаратних засобів, на таке співпадіння допоможе швидко заблокувати зловмисні дії в корпоративній мережі [17].

Методи аналізу IP-адрес та кореляція з Індикаторами компрометацій є важливою частиною захисту корпоративної мережі від кіберзагроз. Вони дозволяють виявляти та реагувати на потенційні небезпеки вчасно. Завжди слід слідкувати за новими методами та технологіями, оскільки кіберзагрози постійно еволюціонують.

2.5 Стратегії і алгоритми блокування атак

У світі кібербезпеки, де загрози постійно розвиваються, важливо мати ефективні стратегії і алгоритми для блокування атак в реальному часі. У цьому розділі ми дослідимо різні рішення, які можуть бути використані для динамічного блокування атак, включаючи як відкриті, так і пропрієтарні рішення, а також алгоритми їх роботи та процес аналізу та блокування.

Реалізація стратегій і алгоритмів блокування атак є важливою складовою кібербезпеки в сучасному цифровому світі. Ці стратегії і алгоритми мають на меті захистити мережі та системи від потенційних загроз і небажаних дій. Вони знаходять широке застосування у різних сферах, від корпоративних мереж до державних інфраструктур, і вимагають ретельної реалізації та конфігурації.

Реалізація включає в себе встановлення і налаштування різноманітних інструментів та систем, таких як фаєрволи, IDS/IPS системи, системи моніторингу подій і журналювання, а також інші заходи, спрямовані на виявлення і блокування потенційно шкідливої активності. Реалізація також

передбачає регулярне оновлення баз даних загроз і алгоритмів аналізу для забезпечення актуального захисту.

Застосування цих стратегій і алгоритмів є критичним для забезпечення безпеки і надійності мереж і систем. Вони допомагають виявляти та блокувати атаки на різних рівнях, від перехоплення пакетів на мережевому рівні до виявлення вторгнень на рівні застосунків. Застосування цих стратегій і алгоритмів дозволяє попередити втрату даних, фінансові втрати і репутаційні ризики, пов'язані з кібератаками.

Загальний принцип полягає у тому, що системи блокування атак працюють на основі правил і сигнатур, які визначають, яка активність вважається підозрілою чи шкідливою. Якщо така активність виявляється, система може автоматично реагувати, блокуючи або ізолювати підозрілі ресурси або IP-адреси. Виконання дій блокування відбувається через зміну налаштувань периметру мережі.

У відповідь на сталі та постійно зростаючі загрози кібербезпеки, реалізація і застосування стратегій і алгоритмів блокування атак стають більш важливими, ніж будь-коли раніше. Це вимагає постійного вдосконалення та оновлення захисних заходів, а також розуміння сучасних методів атак та їхніх векторів. Лише таким чином можна забезпечити ефективний захист від кіберзагроз і зберегти безпеку в цифровому світі.

Аналіз подібних реалізацій допоможе більш детально зрозуміти, як працює цей механізм. Для цього буде розглянута дві популярних реалізації автоматичного блокування мережевих загроз [8].

Fail2ban – є високоефективним програмним забезпеченням, що використовується для підвищення безпеки серверів. Цей інструмент автоматично моніторить лог-файли серверів, виявляє підозрілу поведінку, таку як повторні спроби неавторизованого доступу, та активно блокує IP-адреси порушників, запобігаючи можливим кібератакам. Fail2ban значно зменшує ризик

проникнення шкідливих користувачів та ботів, захищаючи системи від різноманітних мережевих загроз. Його механізм роботи полягає в наступному:

1. Моніторинг логів. Fail2ban постійно моніторить лог-файли різних служб, таких як SSH, Apache, Postfix і багато інших. Ці логи містять інформацію про спроби авторизації та інші події в системі.

2. Виявлення аномалій. Fail2ban аналізує логи на предмет аномальної активності, такої як повторні невдачі авторизації, неправильні паролі, агресивні запити тощо. Це дозволяє визначити потенційно небезпечні дії.

3. Заходи превентивного блокування. Якщо Fail2ban виявляє небажану активність, він вживає заходів для блокування цієї активності. Зазвичай це включає в себе додавання IP-адреси, яка вказується в логах, до файрвола або іншого механізму блокування.

4. Часовий обмежувач. Fail2ban також використовує часові обмеження для блокування. Це означає, що IP-адреса блокується на певний період часу, після чого блокування автоматично знімається. Це допомагає запобігти постійному блокуванню IP-адрес, що може бути результатом помилкових спроб.

5. Конфігурація і настроювання. Fail2ban дозволяє адміністраторам налаштовувати правила для виявлення аномалій і заходів для блокування. Це дозволяє адаптувати програму до конкретних потреб і вимог безпеки.

Snort – це система виявлення вторгнень (IDS) та система захисту від вторгнень (IPS), яка використовується для моніторингу та захисту мережі від потенційних атак. Механізм роботи Snort також має кілька ключових аспектів:

1. Сигнатурний аналіз. Snort використовує сигнатурний аналіз для виявлення відомих патернів атак. Він має базу даних сигнатур, які описують характерні підписи атак і вірусів. Якщо трафік відповідає якій-небудь сигнатурі, Snort сповіщає про це.

2. Аналіз аномалій. Крім сигнатурного аналізу, Snort також використовує аналіз аномалій для виявлення незвичної активності, яка може бути індикатором атаки. Він аналізує різні характеристики мережевого трафіку,

такі як розмір пакетів, частота запитів, високе використання ресурсів і багато інших.

3. Динамічні блокування атак. Snort може бути налаштований для автоматичного блокування трафіку, що відповідає виявленим атакам. Це може включати в себе відключення з'єднання з атакуючими IP-адресами або навіть блокування всієї підмережі.

4. Підтримка протоколів і додаткових модулів. Snort підтримує багато різних протоколів і може бути розширений за допомогою додаткових модулів і правил. Це дозволяє адаптувати Snort до конкретних потреб мережі.

Як системи виявлення вторгнень і захисту від вторгнень, Fail2ban і Snort спільно виконують важливу роль в забезпеченні безпеки мереж і систем. Fail2ban спеціалізується на захисті від небажаної активності, такої як спроби вторгнення по SSH, в той час як Snort виявляє і відстежує широкий спектр атак і вторгнень, що використовують різні протоколи і вектори нападу. За допомогою цих систем адміністратори можуть активно моніторити та захищати мережу від потенційних загроз і атак.

2.6 Висновки щодо теоретичних досліджень

Завершуючи розділ, присвячений теоретичним основам системи автоматичного реагування на мережеві атаки, підходимо до ключових висновків цієї глибокої та многогранної теми. Протягом цього розділу підняли завісу над складним світом кібербезпеки, розглядаючи різноманітні аспекти, які формують сучасні стратегії захисту мережевих систем.

Детально розглянули архітектуру системи, зосереджуючись на ключових компонентах, таких як сенсори, системи збору та аналізу даних, та навіть глибше - на методах аналізу IP адрес та їх кореляції з індикаторами компрометації. Цей аналіз відкрив двері у світ, де кожен біт даних може бути ключем до виявлення та блокування потенційних загроз.

Значну увагу було приділено стратегіям та алгоритмам блокування атак, виявляючи, як сучасні технології та інноваційні підходи допомагають у протидії

кіберзагрозам. Від авангардних алгоритмів машинного навчання до класичних методів сигнатурного аналізу - кожен з них відіграє свою роль у цілісній стратегії захисту.

Цей розділ не лише висвітлює складність та глибину теми кібербезпеки, але й підкреслює важливість безперервного навчання та адаптації у світі, де технології та загрози розвиваються надзвичайно швидко. Він показує, що захист мережевих систем - це не просто встановлення програмного забезпечення чи обладнання, а це постійний процес аналізу, оцінки та вдосконалення.

У цілому, другий розділ піднімає нас на новий рівень розуміння в області кібербезпеки, демонструючи важливість кожного аспекту в комплексній системі захисту. Він залишає нас з підвищеним відчуттям обережності, але й з впевненістю в тому, що належні знання та інструменти можуть значно зміцнити наш захист у цифровому світі.

3 РОЗРОБКА СИСТЕМИ АВТОМАТИЧНОГО РЕАГУВАННЯ НА МЕРЕЖЕВІ АТАКИ

3.1 Проектування архітектури додатку

У сучасному світі кібербезпеки, розробка ефективної системи для автоматичного реагування на мережеві атаки вимагає глибокого розуміння потреб безпеки та вибору оптимальних технологічних рішень. В цьому контексті, вибір мови програмування для розробки додатку є критично важливим. Вибір падає на Python3, і ось чому.

Python3 виділяється своєю гнучкістю та широкими можливостями. Як високорівнева, інтерпретована мова, Python забезпечує значну економію часу та ресурсів при розробці, завдяки своїй читабельності та простоті синтаксису. Ці характеристики роблять Python ідеальним для швидкого розроблення та тестування, що є важливим у динамічному полі кібербезпеки.

Крім того, Python може похвалитися однією з найбільших спільнот розробників, що забезпечує велику кількість готових рішень та бібліотек. Ця підтримка спільноти є важливою для вирішення складних задач і забезпечення безпеки додатку. Велика кількість бібліотек, таких як requests для мережевих запитів, pandas для аналізу даних, та scikit-learn для машинного навчання, робить Python незамінним інструментом для розробки систем кібербезпеки.

Провівши порівняльний аналіз з іншими мовами програмування дійшов висновку, що для реалізації цієї задачі найкраще підходить саме мова програмування Python. Ця мова програмування має наступні переваги. Вона відома своєю читабельністю та простотою синтаксису. В порівнянні з мовами, такими як C++ або Java, Python дозволяє швидше реалізувати складні ідеї. Чистий та простий синтаксис Python сприяє швидкому розробленню та легшому утриманню коду. Має величезну кількість бібліотек та фреймворків, які можуть використовуватися для різних задач від обробки даних (numpy, pandas) до розробки веб-додатків (Django, Flask). Це ставить його в вигідне становище порівняно з мовами, які можуть вимагати більше зусиль для інтеграції з

зовнішніми бібліотеками або не мають такої ж багатої екосистеми. Це міжплатформенна мова, що дозволяє запускати розроблені додатки на різних операційних системах без значних змін у коді. Це контрастує з деякими мовами, які можуть бути більш залежними від платформи. Легко інтегрується з іншими мовами програмування та системами. Це дозволяє використовувати Python як "клей" для з'єднання різних систем і компонентів. Наприклад, можна легко інтегрувати Python-скрипти з системами на базі C або Java.

Python ефективно використовується для мережевих операцій та обробки даних, що є особливо важливим для систем автоматичного реагування на мережеві атаки. Бібліотеки, такі як Scapy для мережевих протоколів, або Pandas для обробки та аналізу даних, роблять Python міцним інструментом у цій області.

Щодо архітектури додатку, вона буде складатися з декількох модулів, кожен з яких відповідає за певну функціональність. Модулі збору даних будуть збирати інформацію з різних джерел, включаючи системні журнали та мережеві потоки. Модуль аналізу даних буде обробляти цю інформацію, виявляючи потенційні загрози та аномалії. Також буде реалізовано модуль відповіді на інциденти, який автоматизує процес реагування на виявлені загрози.

Враховуючи ці переваги, Python вибрано як основу для розробки системи автоматичного реагування на мережеві атаки, оскільки він забезпечує ідеальне поєднання гнучкості, міцної підтримки, і широких можливостей для рішення складних задач у сфері кібербезпеки.

Щодо розгортання та роботи, додаток буде розроблено таким чином, щоб максимально автоматизувати процес виявлення та реагування на загрози. Використання скриптових можливостей Python дозволяє легко інтегрувати додаток з різними системами та інструментами. Це забезпечить швидке та ефективне реагування на інциденти, мінімізуючи потенційну шкоду від кібератак.

Завершуючи, архітектура додатку, розроблена на Python3, є сучасним та гнучким рішенням для системи автоматичного реагування на мережеві атаки.

Вона поєднує в собі швидкість розробки, масштабованість та високу ефективність, що є необхідними у боротьбі з постійно змінюваними кіберзагрозами [9].

3.2 Налаштування компонентів системи: сенсори, збір, обробка та збереження даних

Створення та налаштування індексів у системі Opensearch для зберігання інформації про зловмисні індикатори та дозволених IP адрес є важливою складовою системи автоматичного виявлення та реагування на мережеві атаки. Розглянемо теоретичну основу та практичну реалізацію цих індексів. Таким чином потрібно створити два індекси, один для наповнення індикаторами компрометацій у форматі IP адрес, другий тимчасовий для внесення IP адрес, які не були ідентифіковані як зловмисні. Тимчасовий індекс буде зберігатися 1 день, а далі перестворюватися з очищенням усіх попередніх даних. Це необхідно для того, щоб уникнути ситуацію, коли безпечна IP адреса стає скомпрометованою та може використовуватися для зловмисних дій.

Припускаючи наявність вже існуючої інфраструктури, економія ресурсів може бути досягнута через її ефективне налаштування. Важливим аспектом такого налаштування є створення та оптимізація індексів, які відіграють ключову роль у забезпеченні швидкого та точного доступу до даних. Цей процес може бути виконаний через графічний інтерфейс або за допомогою запитів REST API, залежно від потреб та уподобань користувача. Наприклад, для створення індексу в Opensearch, що використовується для збереження зловмисних IP-адрес, можна використати REST API (рисунок 3.1). Такий підхід дозволяє автоматизувати та налаштовувати процеси за допомогою програмних скриптів, що є особливо корисним для великих та складних систем.

```
└─[$] curl -k -X PUT "https://opensearch.local.dev:9200/ioc_malicious_ip" \
     -u "admin:opensearchadmin" \
     -H "Content-Type:application/json" -d'
{
  "mappings": {
    "properties": {
      "ip_address": {
        "type": "ip"
      },
      "country": {
        "type": "keyword"
      },
      "malicious_type": {
        "type": "text"
      }
    }
  }
}'
```

Рисунок 3.1 – Створення індексу "ioc_malicious_ip" з відповідним мапінгом

Реалізація додаткового індексу, що використовуватиметься як тимчасовий (зберігатиметься 1 день), щоб знизити навантаження на системні ресурси і не перевіряти безліч разів одні й ті ж IP адреси. В цьому індексі будуть зберігатися IP адреси, які уже були перевірені системою і отримали позначку "чисто". Про необхідність використання такого підходу буде розібрано далі за текстом. Створення такого індексу має певну особливість. При його реалізації будуть використовуватися шаблони та компоненти. Шаблони індексів (index templates) та компоненти в Opensearch використовуються для визначення налаштувань, мапінгів та інших конфігурацій, які автоматично застосовуються до індексів, що відповідають певним шаблонам імен. Розглянемо основні елементи конфігурації.

Значення "index_patterns" визначає шаблони імен індексів, до яких буде застосовуватися цей шаблон. У цьому випадку, "logs-analyzerip.list-prod" означає, що шаблон буде застосовуватися до всіх індексів, які відповідають цьому шаблону імен (рисунок 3.2).

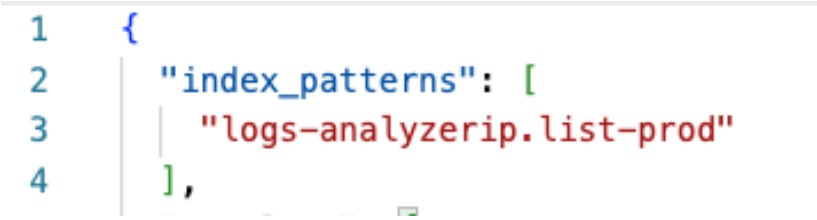


Рисунок 3.2 – Визначення шаблону імен індексів, до яких він застосовуватиметься

Значення template вказує на необхідні налаштування даного шаблону. У ньому вказуються мапінги та псевдоніми. Мапінг визначає, як поля даних будуть співставлені та індексовані. Наприклад, "country" як keyword означає, що значення країни будуть зберігатися як точні рядки, ідеальні для фільтрації та агрегації. "malicious_type" як text означає, що це поле буде проіндексоване для повнотекстового пошуку. ip_address як ip означає спеціальне зберігання та оптимізацію для IP-адрес (рисунок 3.3) [14].

Значення "aliases" створює псевдонім для індексів, які відповідають шаблону. У цьому випадку, "logs-analyzerip.list-latest" буде псевдонімом для індексів, що відповідають шаблону.

```
"template": {
  "mappings": {
    "properties": {
      "country": {
        "type": "keyword"
      },
      "malicious_type": {
        "type": "text"
      },
      "ip_address": {
        "type": "ip"
      }
    }
  },
  "aliases": {
    "logs-analyzerip.list-latest": {}
  }
},
```

Рисунок 3.3 – Налаштування мапінгу та псевдонімів

Ключ "composed_of" містить список компонентів шаблону, які містять знову ж таки налаштування, мапінги та інші конфігурації. Компоненти можуть бути використані для повторного використання спільних конфігурацій у різних шаблонах (рисунок 3.4).

```
23    "composed_of": [
24      "hot-delete_no-replica"
25    ],
```

Рисунок 3.4 – Список компонентів шаблонів

Значення пріоритет застосування шаблону, якщо індекс відповідає кільком шаблонам. Шаблони з вищим пріоритетом мають перевагу. Ключ "_meta" зберігає метадані про шаблон, які не впливають на його поведінку. Він корисний для зберігання інформації про шаблон для кінцевих користувачів або адміністраторів. Ключ "data_stream" використовується для того, аби визначити, що шаблон призначений для data streams, і вказує основне поле часової мітки (@timestamp), яке використовується в data streams. Такий тип індексу застосовується до даних, які постійно змінюються. Останнім параметром вказується назва шаблону в ключі "name" (рисунок 3.5)

```
23    "composed_of": [
24      "hot-delete_no-replica"
25    ],
26    "priority": "0",
27    "_meta": {
28      "flow": "components"
29    },
30    "data_stream": {
31      "timestamp_field": {
32        "name": "@timestamp"
33      }
34    },
35    "name": "logs-analyzerip.list-prod"
36  }
37
```

Рисунок 3.5 – Загальні параметри роботи шаблону

Ці параметри разом дозволяють детально контролювати, як індекси будуть створені, налаштовані та управляються в Opensearch, що забезпечує гнучкість та ефективність управління великими обсягами даних.

Для автоматизації управління життєвим циклом індексів в Opensearch використовується Index State Management (ISM) політики. Це дозволяє визначити, як індекси повинні бути оброблені та модифіковані протягом їх життєвого циклу, включаючи створення, зміну налаштувань, розміщення в різних теплових зонах (наприклад, гарячі, теплі, холодні), а також видалення. Теплові зони – це умовне позначення, яке дозволяє розміщувати дані на різних апаратних вузлах. Наприклад, індекси до яких йде постійне звернення на читання, розміщуються на швидких дисках та з великим об'ємом оперативної пам'яті, а індекси без активних пошуків розміщуються на повільніших дисках.

Політика ISM має наступні ключові компоненти, а саме: ім'я, опис, значення за замовчуванням, етапи зберігання та шаблон, який застосований до індексу. Беручи це до уваги далі буде розглянуто таку конфігурацію.

Ім'я політики – це унікальний ідентифікатор для політики, і зазначається він в ключі "policy_id". Наступним йде опис політики, де описується для чого використовується дана політика. Значення розміщення за замовчуванням визначає на яких серверах буде зберігатися цей індекс. Загалом ця частина має наступний вигляд (рисунок 3.6).

```
1  {
2    "policy": {
3        "policy_id": "logs-analyzerip.list-prod",
4        "description": "IP addresses getting from different system logs and marked as not malicious ",
5        "default_state": "hot",
```

Рисунок 3.6 – Загальна інформація, яка описує політику збереження індексу

Визначення станів та дій, які будуть виконуватися в кожному стані зазнача.ться в уключі "state". У цьому ключі вказуються усі необхідні умови та дії при збереженні індексу. Першим ключем вказується над яким станом

виконується дія. Далі проводиться перелік дій та умов, які мають бути для виконання цих дій (рисунок 3.7).

```
"states": [
    {
        "name": "hot",
        "actions": [
            {
                "retry": {
                    "count": 3,
                    "backoff": "exponential",
                    "delay": "1m"
                },
                "rollover": {
                    "min_size": "10gb",
                    "min_index_age": "1d",
                    "copy_alias": false
                }
            }
        ],
```

Рисунок 3.7 – Налаштування станів та дій над індексами у цьому стані

Після завершення налаштувань, індекси в системі переходять у різні стани для оптимізації подальшої обробки даних (рисунок 3.8).

```
        "transitions": [
            {
                "state_name": "delete",
                "conditions": {
                    "min_rollover_age": "1h"
                }
            }
        ]
    },
```

Рисунок 3.8 – Налаштування переміщення між станами індексів

Таким чином для видалення індексу після його переміщення до стану "delete" необхідно виконати дію із видалення, для цього описується додатковий стан та дії для нього (рисунок 3.9).

```
32          {
33              "name": "delete",
34              "actions": [
35                  {
36                      "retry": {
37                          "count": 3,
38                          "backoff": "exponential",
39                          "delay": "1m"
40                      },
41                      "delete": {}
42                  }
43              ],
44              "transitions": []
45          }
46      ],
```

Рисунок 3.9 – Дії при переході індексу в статус "delete"

Останньою дією лишається описати, до яких шаблонів індексів буде застосовується дана політика життєвого циклу індексів (рисунок 3.10).

```
47      "ism_template": [
48          {
49              "index_patterns": [
50                  "logs-analyzerip.list-prod"
51              ],
52              "priority": 1
53          }
54      ]
55  }
56  }
```

Рисунок 3.10 – Налаштування застосування політики життєвого циклу до індексів, що відповідають зазначеному шаблону.

Застосування шаблону (рисунок 3.11) та політики (рисунок 3.12) може бути реалізовано, як через веб інтерфейс Opensearch, так і через REST API запит.

Рисунок 3.11 – Створення шаблону індексів за допомогою REST API

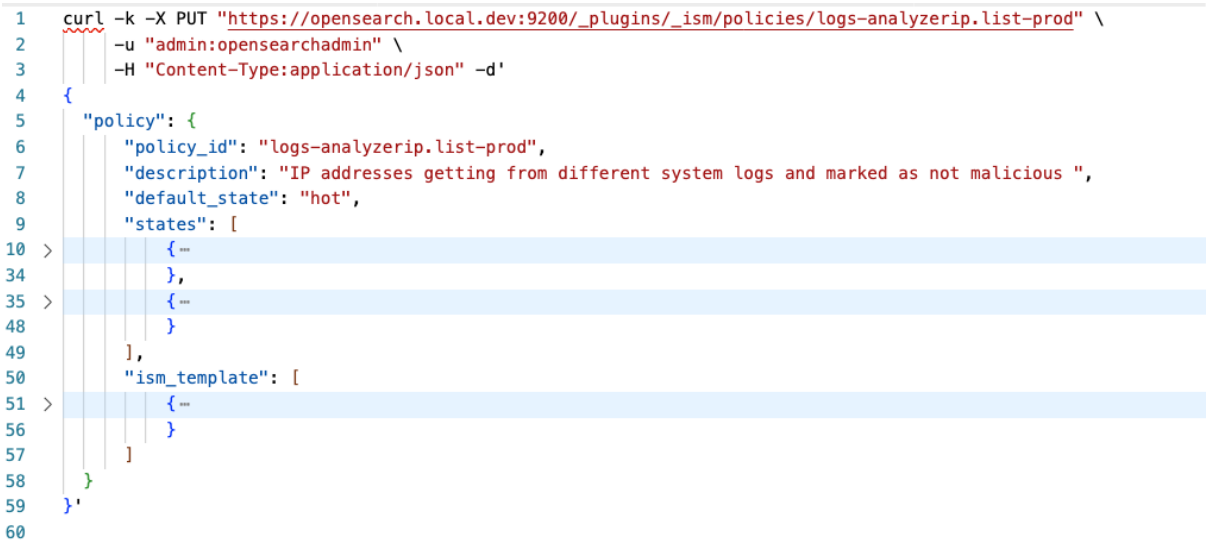


Рисунок 3.12 – Створення та застосування політики життєвого циклу індексів (ISM) за допомогою REST API.

Після ініціалізації та налаштування сховища журналів, наступним кроком є конфігурація сенсорів для збору даних. Для збирання журналів роботи DNS, спочатку виконується налаштування системи аудиту. Після цього здійснюється інсталяція та конфігурація спеціалізованої системи збору журналів, такої як winlogbeat (рисунок 3.13). Winlogbeat ефективно вилучає та передає журнали подій з Windows до сховища для подальшого аналізу та обробки.

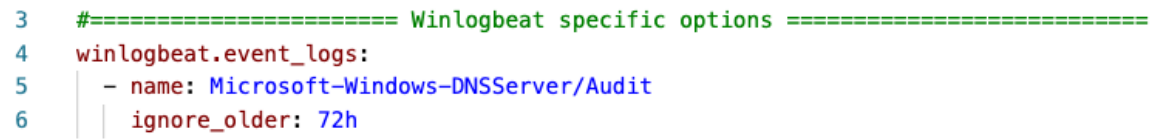


Рисунок 3.13 - Підключення збору журналів DNS в налаштуваннях winlogbeat.

Обробка журналів DNS відбувається на стороні Logstash за відповідними правилами, для розділення запису на окремі елементи такі як домен та IP адреса, це необхідно для того аби легко та швидко працювати з даними [24].

На веб серверах необхідно підключити модуль nginx в налаштуваннях filebeat та налаштувати збір журналів nginx (рисунок 3.14). Веб сервер має бути налаштований таким чином, щоб писати журнали у відповідні шляхи.

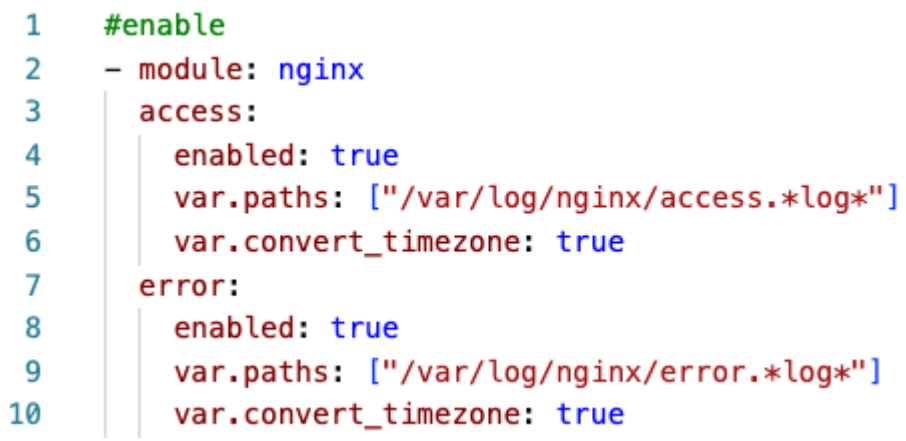


Рисунок 3.14 – Налаштування модуля NGINX на веб сервері.

Налаштування мережевого брандмауера на маршрутизаторі Mikrotik включає використання специфічних команд для додавання правил фільтрації трафіку, які визначають типи атак і відповідно маркують їх (рисунок 3.15). Це робиться з метою забезпечення більш ефективного моніторингу та виявлення потенційно шкідливих дій у мережі. Розширені налаштування брандмауера на Mikrotik забезпечують ефективний захист мережі та допомагають підтримувати стабільність і безпеку системи.

```
/ip firewall filter
add chain=detect-ddos dst-limit=32,32,src-and-dst-addresses/10s
    action=log log-prefix="DDoS Detected: " comment="DDoS Detection"
add chain=detect-ddos dst-limit=32,32,src-and-dst-addresses/10s action=return
/ip firewall filter
add action=add-dst-to-address-list address-list=ddos-targets
    address-list-timeout=10m chain=detect-ddos comment="Add target to DDoS targets list"
add action=add-src-to-address-list address-list=ddos-attackers
    address-list-timeout=10m chain=detect-ddos comment="Add attacker to DDoS attackers list"
/ip firewall raw
add action=drop chain=prerouting dst-address-list=ddos-targets
    src-address-list=ddos-attackers
    log-prefix="DDoS Attack Blocked: " comment="Block DDoS Attack"
/ip settings set tcp-syncookies=yes
/ip firewall filter
add chain=detect-ddos dst-limit=32,32,src-and-dst-addresses/10s
    protocol=tcp tcp-flags=syn,ack action=log
    log-prefix="SYN-ACK Flood Detected: " comment="SYN-ACK Flood Detection"
```

Рисунок 3.15 – Налаштування фаєрволу Mikrotik

Налаштувавши типи журналів та додавання міток до журналів про атаки, необхідно налаштувати вивантаження журналів на зовнішній syslog сервер (рисунок 3.16).

```
/system logging action
add name=remote target=remote remote=logstash.local.dev remote-port=514 src-address=0.0.0.0
/system logging
add action=remote topics=firewall,raw
```

Рисунок 3.16 – Налаштування вивантаження журналів на syslog сервер

Таким чином зібрані журнали роботи DNS серверу, фаєрволу маршрутизатора та інших систем надсилаються на сервер logstash для подальшої обробки і аналізу.

На сервері logstash проводиться обробка цих журнальних даних, в результаті чого вони розбиваються на окремі складові, зокрема виокремлюються ідентифікатори (IP адреси) і розміщуються в окремих полях. Цей процес включає в себе фільтрацію, розбиття та інші маніпуляції з даними для підготовки їх для подальшого аналізу.

Після обробки і виокремлення IP адрес, дані надсилаються на зберігання в індекс Opensearch. У цьому індексі дані зберігаються у вигляді, який легко

розуміти і обробляти для подальших аналітичних завдань. Ця система допомагає в моніторингу та аналізі мережевої активності та безпеки.

3.3 Розробка алгоритму ідентифікації зловмисності

Розробка алгоритму ідентифікації зловмисності – це надзвичайно важливий етап в забезпеченні безпеки корпоративної мережі. Завдання розробити систему, яка буде спроможною вчасно виявляти потенційно небезпечну активність та атаки, і вживати відповідні заходи безпеки.

Для цього використовуються різні джерела отримання IP-адрес і визначається послідовність дій, яка допоможе відслідковувати потенційну зловмисну активність. Найперше, аналізуються DNS-запити і перевіряється, чи містяться ці IP-адреси в індексі з індикаторами компрометацій (IoC). Якщо так, то ця IP-адреса вже відома через попередні інциденти. Тому відразу запускається механізм блокування цієї IP адреси на периметрі мережі.

Якщо ж IP-адреса отримана з логів фаєрволу чи антивірусу, та містить помітки про атаку, тоді відразу додається її до індексу IoC та запускається механізм блокування на периметрі мережі, щоб зменшити ризик [24].

У випадку журналів роботи NGINX разом із IP адресами, що перевіряються в індексі IoC, проводиться більш глибокий аналіз поведінки. Отримуються дані за короткий проміжок часу, для прикладу 5 хв, та аналізується кількість помилкових запитів з кожної IP адреси, що робить такі запити. Якщо ця кількість перевищує певне порогове значення, відповідні IP-адреси додаються до індексу IoC. Іншим показником враховується UserAgent та URI в запитах. Наприклад, ведеться спостереження за URI, які завідомо відомі як потенційно шкідливі (наприклад, що містять git або php), якщо їх не використовують системи, що відслідковуються. Ця послідовність дій має наступне пояснення (рисунок 3.17) .

```
# ПІДОЗРІЛІ URL
nginx_url_malicious = ".*(env|git|php|base64|well|vscode|script\
    |ftp|phpstorm|admin|setup|wp|wordpress|geoserver|webui|hudson).*"
```

Рисунок 3.17 – Приклад зловмисних частин в URL запиті

Для аналізу кількості помилкових запитів потрібно спочатку отримати дані з логів Nginx за вибраний проміжок часу. Ці логи містять записи про всі HTTP-запити, які прийшли на веб-сервер. Дані запити аналізуються та визначається кількість запитів, які спрямовані на неіснуючі ресурси. Наприклад, якщо в логах є багато запитів на шляхи типу /git або /phpmyadmin, при цьому система не використовує такі ресурси, то це може свідчити про спробу несанкціонованого доступу. Якщо кількість таких хоча б раз зустрічається в поточній вибірці, то відповідні IP-адреси позначаються як потенційно шкідливі. Після визначення кількості помилкових запитів, перевіряється UserAgent, що супроводжує ці запити. Наприклад, якщо UserAgent вказує на те, що це запит від відомого бота чи агента, що часто використовується для сканування веб-ресурсів, то це може підвищити ризик і вказувати на потенційну атаку. Наприклад, якщо спостерігаються запити від певної IP-адреси з UserAgent, який вказує на бота, і ці запити спрямовані на /wp-login, то це може бути ознакою спроби злому веб-сайту (рисунок 3.18).

51.222.41.85	301	Mozilla/5.0 (X11; Fedora; Linux x86_64; rv:94.0) Gecko/20100101 Firefox/95.0	-	/wp-login.php
51.222.41.85	503	Mozilla/5.0 (X11; Fedora; Linux x86_64; rv:94.0) Gecko/20100101 Firefox/95.0	htt…	/wp-login.php
57.129.23.166	301	Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chro…	-	/.env
95.108.213.101	403	Mozilla/5.0 (compatible; YandexBot/3.0; +http://yandex.com/bots)	-	/
34.132.26.49	301	Mozilla/5.0 (X11; Fedora; Linux x86_64; rv:94.0) Gecko/20100101 Firefox/95.0	-	/wp-login.php

Рисунок 3.18 – Приклад зловмисних запитів

В результаті цього аналізу мається можливість виявити потенційно шкідливу активність на основі логів Nginx та внести відповідні IP-адреси до індексу з IoC. Це допомагає реагувати на потенційні загрози та забезпечити безпеку мережі.

Далі вивчаються DNS-логи. Отримані IP адреси, з DNS журналів, перевіряються в індексі IoC і відповідно, якщо адресу знайдено в індексі IoC, то запускається механізм блокування цієї IP адреси на периметрі мережі. Натомість, якщо IP адресу не знайдено, тоді жодних додаткових дій не застосовується [11].

Ключовим етапом є можливість змінювати налаштування мережевих пристроїв через SSH або API. За необхідності відбувається підключення до цих пристроїв, зчитуються їхні конфігурації та вносяться зміни, додаючи зловмисні IP-адреси до списків блокування через мережевий екран iptables, для linux, або мережевий екран, для маршрутизаторів.

Важливо також фіксувати події та результати роботи у журналах. Це допомагає вести облік і аналізувати здійснені дії. Сповіщення в месенджер про блокування також грає важливу роль у вчасному реагуванні на потенційні загрози.

Ця система працює як сервіс та періодично перевіряє джерела IP-адрес для виявлення зловмисної активності, забезпечуючи надійний рівень безпеки мережі.

Після проведення аналізу логів Nginx та визначення потенційно шкідливої активності, наступним кроком є розробка додатку, який автоматизує цей процес та забезпечує моніторинг та реагування на можливі загрози.

Розробка додатку на мові програмування Python 3 є стратегічним вибором, який дозволяє використовувати гнучкість та багатство функціоналу цієї мови. Python3 є високорівневою мовою, яка забезпечує чистоту коду та легкість у підтримці, що робить її ідеальною для створення складних застосунків, таких як аналітичні та моніторингові системи. Додаток буде регулярно зчитувати логи Nginx за обраний проміжок часу та виконувати аналіз на предмет невдалих запитів та потенційно шкідливих URI. Для отримання необхідних даних виконується запит в OpenSearch.

Спершу виконується визначення шаблонів для фільтрації URL, реферерів та User-Agents. Скрипт використовує регулярні вирази для ідентифікації URL-

адрес, реферерів та User-Agents, які слід ігнорувати під час перевірки. Наприклад, nginx_url_exclude_from_check містить шаблони для URL-адрес, які не потрібно перевіряти. Як от адреси, пов'язані з електронною поштою, пошуком тощо. Це ті елементи, які завідома легітимні та використовуються в продуктовому середовищі інфраструктури, що захищається. Аналогічно, nginx_referer_exclude_from_check та nginx_agent_exclude_from_check визначають шаблони для реферерів та User-Agents, які слід виключити з перевірки. Водночас, шаблони в nginx_agent_malicious визначають підозрілі User-Agents, які можуть вказувати на автоматизовані сканери або ботів. Ці регулярні вирази використовуються для відфільтрування певних типів трафіку або запитів на основі їх URL, реферера, чи User-Agent, щоб сконцентруватися на більш підозрілих випадках (рисунок 3.19).

```
# ДОЗВОЛЕНІ URL З ЯКИХ РОБИТЬСЯ ЗАПИТ
nginx_url_exclude_from_check = ".*(email|robots|apple|promo|claims|\
  documents|profile|product|search|catalog).*"
# ДОЗВОЛЕНІ ІДЕТИФІКАТОРИ РЕФЕРЕРА З ЯКИХ РОБИТЬСЯ ЗАПИТ
nginx_referer_exclude_from_check = "https*"
# ДОЗОВЛЕНІ ВЕБ АГЕНТИ
nginx_agent_exclude_from_check = ".*(fwdcdn|comscore).*"
# ПІДОЗРІЛІ ВЕБ АГЕНТИ
nginx_agent_malicious = ".*(scanner|bot|spider|abuse|proxy|grab|fuzz|crawler).*"
# ПІДОЗРІЛІ URL
nginx_url_malicious = ".*(env|git|php|base64|well|vscode|script\
    |ftp|phpstorm|admin|setup|wp|wordpress|geoserver|webui|hudson).*"
```

Рисунок 3.19 – налаштування виключень та підозрілих елементів

Основні дії включають використання регулярних виразів та часових фільтрів для вибіркового аналізу записів, зосереджуючись на значущих даних, таких як підозрілі запити, виявлені через шаблони URL та коди відповідей.

Процес починається зі частини коду, який виконує комплексний моніторинг, виявляючи та блокуючи підозрілі IP-адреси. Цей додаток визначає шаблони для виключення певних URL, реферерів та User-Agents, після чого формує запити до Opensearch для збору даних про підозрілі дії (рисунок 3.20). Використовуючи агрегації, скрипт виділяє IP-адреси (рисунок 3.21),

підготовлюючи їх до блокування та додавання до списку IoC (індикаторів компрометації).

```
"bool": {
  "must": [
    { "range": { "nginx.access.response_code": { "gt": 300 } } },
    { "range": { "@timestamp": { "gte": match_check_time, "lte": "now" } } },
    { "bool": { "minimum_should_match": 1, "should": [
      { "regexp": { "nginx.access.url": nginx_url_malicious } },
      { "regexp": { "nginx.access.agent": nginx_agent_malicious } } ] }
    }
  ],
  "must_not": [
    { "wildcard": { "nginx.access.referrer": nginx_referer_exclude_from_check } },
    { "regexp": { "nginx.access.agent": nginx_agent_exclude_from_check } },
    { "regexp": { "nginx.access.url": nginx_url_exclude_from_check } },
    { "term": { "nginx.access.response_code": 499 } },
    { "match_phrase": { "nginx.access.url": "/.well-known/traffic-advice" } }
  ]
}
},
```

Рисунок 3.20 – Визначення включень та виключень у запиті до OpenSearch

```
"aggs": {
  "unique_ips": {
    "terms": {
      "field": "nginx.access.remote_ip.keyword",
      "size": 10000
    }
  }
}
```

Рисунок 3.21 – Агрегація отриманих даних за необхідним полем

Процес також включає перевірку журналів NGINX і брандмауера. Запити до NGINX зосереджені на виявленні підозрілих URL та веб-агентів, а також записах із статусами відповіді понад 300. Одночасно брандмауер шукає дії, що викликають підозру, і збирає дані про джерела цих активностей (рисунок 3.22).

Далі використовуються функції, такі як append_to_ioc для перевірки та додавання нових IP-адрес до індексу ioc_malicious_ip.

```
query_firewall = {
  "query": {
    "bool": {
      "must": [
        { "match": { "reason": "detected" } },
        { "range": { "@timestamp": { "gte": "now-1m", "lte": "now" } } }
      ]
    }
  },
  "aggs": {
    "unique_source_ips": {
      "terms": { "field": "source_ip", "size": 10000 }
    }
  },
  "size": 0
}
```

Рисунок 3.22 – Запит даних із фаєрволу маршрутизатора

Для аналізу код витягує агреговані дані з відповідей Opensearch, що пов'язані з підозрілими запитами до nginx (response_nginx_malicious) та з даними брандмауера (response_fw). Для response_nginx_malicious, вибірка зосереджена на унікальних IP-адресах, які були виявлені в результаті підозрілих запитів. Аналогічно, для response_fw відбираються унікальні IP-адреси, виявлені брандмауером. Проводиться об'єднання списоків IP-адрес, виявлених як в журналах nginx, так і в даних брандмауера. Це створює єдиний список підозрілих IP-адрес для подальшої перевірки. Скрипт перебирає кожну IP-адресу в об'єднаному списку. Та для кожної адреси збільшує лічильник malicious_ip_found, що відслідковує загальну кількість виявлених зловмисних IP. Для виявлених зловмисних IP викликається функція append_to_ioc, яка додає IP-адресу до індексу індикаторів компрометації (IoC). Далі виконується блокування IP-адреси за допомогою двох функцій: block_malicious_ip_web для веб-сервера (за допомогою SSH та iptables) та block_malicious_ip_router для маршрутизатора MikroTik (через API MikroTik) (рисунок 3.23). Разом із цим в консолі запуску відображаються дії пов’язані із виявленням та блокуванням зловмисної IP-адреси (рисунок 3.24).

```
# Отримання агрегованих даних. З виділенням виключно IP адрес
response_nginx_malicious = response_nginx_malicious['aggregations']
response_nginx_malicious = [item['key'] for item in response_nginx_malicious['unique_ips']['buckets']]
response_fw = response_fw['aggregations']
response_fw = [item['key'] for item in response_fw['unique_fqdn']['buckets']]
# Об'єднання знайдених в журналах IP адерс для подальшої перевірки
combined_malicious_ips = response_nginx_malicious + response_fw
# Перевірка знайдених зловмисних IP в індесі з IoC,
# додавання та блокування зловмисної IP
for ip in combined_malicious_ips:
    malicious_ip_found+=1
    append_to_ioc(ip)
    block_malicious_ip_web(ip, linux_host, linux_port, linux_username, linux_key_path)
    block_malicious_ip_router(ip, host_mk, port_mk, username_mk, password_mk)
print("INFO: Found: " + str(malicious_ip_found))
```

Рисунок 3.23 – отримання відфільтрованих IP та запуск механізмів захисту

```
INFO: Checked IP addresses: 706
INFO: Malicious IP addresses found: 0

WARN: Malicious IP 20.40.217.3 find in index with IoC
WARN: Malicious IP 83.97.73.87 find in index with IoC
WARN: NEW Malicious IP found: 31.7.58.42
WARN: Malicious IP 57.129.23.166 find in index with IoC
WARN: NEW Malicious IP found: 78.153.140.219
WARN: NEW Malicious IP found: 66.249.66.199
WARN: NEW Malicious IP found: 66.249.66.36
WARN: NEW Malicious IP found: 66.249.66.38
WARN: NEW Malicious IP found: 66.249.66.39
WARN: NEW Malicious IP found: 95.214.235.169
INFO: Found: 10
```

Рисунок 3.24 – Приклад журналювання виявлення зловмисних IP

Після виявлення, додавання до індексу з IoC та блокування досліджених зловмисних IP адрес, відбувається подальший аналіз IP адрес, які не відповідають шаблонам зловмисності [15].

Відбувається виконання двох запитів до Opensearch. Перший (response_dns) витягує дані з індексу DNS (index_dns), а другий (response_nginx) з індексу nginx (index_nginx). Ці запити мають на меті зібрати відповідні дані про активність мережі. Код фільтрує отримані відповіді, зосереджуючись на агрегованих даних, зокрема на унікальних IP-адресах, які зустрічаються в журналах DNS та nginx. Проводиться об'єднання знайдених IP-адрес. Списки IP-

адрес з обох джерел (DNS та nginx) об'єднуються в один загальний список combined_all_ips для подальшого аналізу. Після чого відбувається пошук зловмисних IP-адрес. Виконується пошук у індексі ioc_malicious_ip Opensearch для кожної IP-адреси з об'єднаного списку. Якщо IP-адреса зустрічається в результатах пошуку, це вказує на її зловмисний характер. Перевірка та блокування зловмисних IP-адрес відбувається наступним чином. Скрипт перебирає всі IP-адреси з об'єднаного списку. Якщо адреса знаходиться серед виявлених зловмисних (found_ips), виводиться повідомлення про її зловмисний характер [20].

Для кожної зловмисної IP-адреси збільшується лічильник malicious_ip_found і виконується її блокування на веб-сервері та маршрутизаторі через відповідні функції (block_malicious_ip_web, block_malicious_ip_router). Після закінчення перебору всіх IP адрес виводяться інформаційні повідомлення про кількість перевірених IP-адрес (i) та кількість виявлених зловмисних IP-адрес (malicious_ip_found).

Отримання IP адрес з журналів роботи DNS сервера, додатково включає у себе виключення відомих публічних або внутрішніх доменів (рисунок 3.25).

```
"must_not": {
  "regexp": {
    "fqdn.keyword": ".*\\.google\\.com|.*\\.microsoft\\.com|one\\.one\\.one\\.one|.*eset\\.com|.*\\.
    ntp\\.org|.*\\.gstatic\\.com|.*\\.e5\\.sk|.*\\.live\\.com|.*\\.msedge\\.net|.*\\.googleapis\\.
    com|.*\\.trafficmanager\\.net|.*\\.slack\\.com|slack\\.com|.*\\.youtube\\.com|.*\\.skype.com|
    skype\\.com|.*\\.windowsupdate\\.com|1\\.1\\.1\\.1|.*\\.easy4ip\\.com|.*\\.mikrotik\\.com|.*\\.
    easy4ipcloud\\.com|.*\\.office365\\.com|.*\\.in-addr\\.arpa|.*\\.doubleclick\\.net|.*\\.
    apple-dns\\.net|.*\\.mixpanel\\.com|.*\\.akamai\\.net|.*\\.telegram\\.org|.*\\.app-measurement\\.
    com|app-measurement\\.com|.*\\.ytimg\\.com|.*\\.resolver\\.arpa|8\\.8\\.8\\.8|.*\\.privatbank\\.
    ua|.*\\.google\\.com\\.ua|.*\\.google-analytics\\.com|.*\\.microsoftonline\\.com|.*\\.facebook\\.
    com||.*\\.amazonaws\\.com|.*\\.akadns\\.net|.*\\.instagram\\.com|.*\\.viber\\.com|.*\\.pki\\.
    goog|.*\\.apple\\.com|.*\\.apache\\.org|.*\\.lencr\\.org|slackb\\.com|.*\\slackb\\.com|.*\\.
    office\\.com|.*\\.bing\\.com|dns\\.google|.*\\.googlesyndication\\.com|.*\\.tiktokcdn\\.com"
  }
}
```

Рисунок 3.25 – Регулярний вираз виключення з перевірки відомих публічних доменів

Цим самим отримується більш релевантна вибірка іще на початковій стадії роботи (рисунок 3.26). Далі відбувається перевірка отриманих IP адрес, за останню хвилину, на наявність у індексі IoC.

```
query = {
  "size": 0,
  "query": {
    "bool": {
      "must": {
        "range": {
          "@timestamp": {
            "gte": "now-1m",
            "lte": "now"
          }
        }
      },
      "must_not": {
        "regexp": {…
      }
    }
  },
  "aggs": {
    "unique_fqdn": {
      "terms": {
        "field": "Message.AnswerSection.IP.keyword",
        "size": 10000
      }
    }
  }
}
```

Рисунок 3.26 – Запит в OpenSearch. Отримання IP адрес для перевірки.

Для реалізації поповнення індексу компрометацій окрім інфомраціх з відкритих джерел використовуються знайдені IP адреси, які приймали участь в атаках на корпоративну мережу.

Функція append_to_ioc(ip_address) використовується для додавання IP-адреси до списку індикаторів компрометації (IoC) у відповідному індексі. Цей

процес має такий вигляд. Виконується пошук у індексі ioc_malicious_ip Opensearch за заданою IP-адресою [26].

Якщо IP-адреса вже є в цьому індексі, виводиться попередження, що зловмисна IP-адреса вже знайдена.

Якщо IP-адреса нова (не знайдена у індексі), виводиться повідомлення про нову зловмисну IP-адресу, і вона додається до індексу ioc_malicious_ip (рисунок 3.27).

```
def append_to_ioc (ip_address):
    search_result = client.search(index="ioc_malicious_ip",
    body={"size": 10, "query": {"bool": {"must": {"terms": {"ip_address": [ip_address]}}}}})
    found_ips = {hit['_source']['ip_address'] for hit in search_result['hits']['hits']}
    if ip_address in found_ips:
        print("WARN: Malicious IP " + str(ip_address) + " find in index with IoC")
    else:
        print("WARN: NEW Malicious IP found: " + str(ip_address))
        doc = {
          "ip_address": ip_address,
          "country": "-"
        }
        response = client.index(
          index="ioc_malicious_ip",
          body=doc
        )
```

Рисунок 3.27 – Функція додавання IP адреси до індексу зловмисних.

Для ефективної роботи з OpenSearch, особливо важливо дотримуватися правильного формату запису даних при додаванні зловмисних IP-адрес до відповідного індексу. Це забезпечує точність і консистентність даних, що є критичним для аналітики та виявлення загроз. Неправильне форматування може призвести до помилок у зберіганні та обробці інформації, що може погіршити якість аналітичних висновків та зменшити ефективність системи моніторингу. Тому, важливо забезпечити, щоб всі дані, які імпортуються в OpenSearch, відповідали заданому стандарту форматування (рисунок 3.28).

```
{ "index" : { "_index" : "ioc_malicious_ip"}}
{"ip_address":"212.193.30.144","country":"RU"}
{ "index" : { "_index" : "ioc_malicious_ip"}}
{"ip_address":"167.71.249.184","country":"US"}
{ "index" : { "_index" : "ioc_malicious_ip"}}
{"ip_address":"101.0.57.158","country":"IN"}
{ "index" : { "_index" : "ioc_malicious_ip"}}
{"ip_address":"120.85.115.148","country":"CN"}
{ "index" : { "_index" : "ioc_malicious_ip"}}
{"ip_address":"125.127.132.112","country":"CN"}
```


Рисунок 3.28 – Формат даних для запису документу з IoC в OpenSearch

Блокування зловмисних IP адрес виконується наступними функціями: block_malicious_ip_web та block_malicious_ip_router.

Функція block_malicious_ip_web, яка приймає аргументами:

– ip_address – зловмисна IP адреса, з'єднання з якою потрібно обмежити

– host – IP адреса або доменне ім'я вузла на якому потрібно обмежити з'єднання з зловмисною IP адресою

– port – порт, що використовується для SSH підключення

– username – ім'я користувача для SSH підключення, та необхідними правами доступу

– key_path – шлях до SSH ключа, який використовується для підключення

Для блокування зловмисної IP адреси на linux сервері, який виконує роль веб сервера, до нього встановлюється SSH-з'єднання, використовуючи бібліотеку Paramiko. Виконується команда iptables для додавання правила, що блокує вхідний та вихідний трафік заданої зловмисної IP-адреси (рисунок 3.29).

```
def block_malicious_ip_web (ip_address, host, port, username, key_path):
    ssh = paramiko.SSHClient()
    ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())
    ssh.connect(host, port, username, key_filename=key_path)
    ssh.exec_command("iptables -I INPUT -s {0} -j DROP".format(ip_address))
    ssh.exec_command("iptables -I OUTPUT -d {0} -j DROP".format(ip_address))
```

Рисунок 3.29 – Функція блокування зловмисної IP адреси на linux сервері

Функція block_malicious_ip_router, яка приймає аргументами:

– ip_address – зловмисна IP адреса, з'єднання з якою потрібно обмежити

– host_mk – IP адреса маршрутизатора на якому потрібно обмежити з'єднання з зловмисною IP адресою

– port_mk – порт, що використовується для підключення

– username_mk – ім'я користувача для підключення, та необхідними правами доступу

– password_mk – пароль користувача для підключення

Щоб блокувати зловмисні IP-адреси на маршрутизаторі MikroTik встановлюється з'єднання з маршрутизатором MikroTik через API. Додається правило в брандмауэр MikroTik для блокування вказаної IP-адреси, використовуючи команду /ip/firewall/address-list/add (рисунок 3.30).

```
def block_malicious_ip_router (ip_address, host_mk, port_mk, username_mk, password_mk):
    logging.basicConfig(level=logging.INFO, \
                        format='%(asctime)s %(name)-24s %(levelname)-8s %(message)s', \
                        datefmt='%H:%M:%S %d-%m-%Y')
    try:
        api = connect(username=username_mk, password=password_mk, host=host_mk, port=port_mk)
        logging.info(f'Connected to {host_mk}')
        api('/ip/firewall/address-list/add', \
            **{'list': 'block_list', \
              'address': ip_address, \
              'comment': 'Blocked by script'})
        logging.info(f'IP address {ip_address} blocked')
    except Exception as e:
        logging.error('Error: %s', e)
```

Рисунок 3.30 – Функція блокування зловмисної IP адреси на маршрутизаторі

Кожна з цих функцій виконує важливу роль у процесі захисту мережі, забезпечуючи можливість швидкого реагування на зловмисні дії та підвищення загальної безпеки інфраструктури. Таким чином, оптимізувавши дані вірним запитом до індеків OpenSearch та поєднавши спільні елементи, отримується швидка й проактивна система реагування на зловмисні запити. Це забезпечує не лише ефективне виявлення та блокування потенційних загроз, але й сприяє

зміцненню загальної безпеки інфраструктури. Завершуючи цей розділ, слід підкреслити, що розвиток та вдосконалення системи автоматичного реагування на мережеві атаки є постійним процесом. Враховуючи швидкі зміни у сфері кіберзагроз та технологій, система повинна бути гнучкою, здатною адаптуватися до нових викликів та трендів у сфері кібербезпеки. Подальші дослідження та розробки повинні зосередитися на інтеграції передових технологій, таких як машинне навчання та штучний інтелект, для підвищення ефективності системи та її здатності прогнозувати та запобігати мережевим атакам ще до їх виникнення. Таким чином, ця робота закладає фундамент для майбутнього розвитку в області автоматичного реагування на мережеві атаки, що є важливим кроком до забезпечення стабільності та безпеки інформаційних систем у нашому постійно еволюціонуючому цифровому світі.

ВИСНОВКИ

Кваліфікаційна робота, присвячена розробці системи автоматичного реагування на мережеві атаки, охоплює комплексні аспекти кібербезпеки, включаючи теоретичні основи, проектування, розробку та реалізацію системи. Значну увагу приділено технологіям та методам виявлення та блокування атак, включаючи використання сучасних інструментів та підходів.

1. На теоретичному рівні розглянуті ключові компоненти системи кібербезпеки, включаючи сенсори, системи збору та аналізу даних. Акцент зроблено на методах аналізу IP-адрес та їх кореляції з індикаторами компрометації. Підкреслюється важливість безперервного навчання та адаптації в динамічному світі кібербезпеки.

2. Система розроблена на мові програмування Python 3, обраній за її гнучкість, широкі можливості та підтримку спільноти. Архітектура додатку включає модулі збору даних, аналізу даних та відповіді на інциденти. Особливу увагу приділено ефективному управлінню даними за допомогою системи Opensearch, налаштуванню індексів та використанню політик Index State Management (ISM).

3. Розроблено алгоритм ідентифікації зловмисності, що включає аналіз DNS-запитів, логів фаєрволу та сервера NGINX. Використовуються методи аналізу поведінки та розпізнавання шкідливих IP-адрес для блокування на периметрі мережі. Система допомагає виявляти потенційно шкідливу активність, забезпечуючи надійний рівень безпеки мережі.

4. Розроблені механізми блокування зловмисних IP-адрес, включаючи взаємодію з мережевими пристроями через SSH або API. Система здатна реагувати на зловмисні дії, збільшуючи загальну безпеку інфраструктури. Важливим аспектом є журналювання подій, що дозволяє аналізувати та вести облік дій системи.

Ця робота виявляє глибоке розуміння ключових аспектів кібербезпеки, а також ілюструє ефективне використання передових методів для створення

всебічної системи захисту. Вона успішно поєднує теоретичні засади кібербезпеки з їх практичним застосуванням, акцентуючи на критичній необхідності адаптації до динамічного, постійно еволюціонуючого цифрового середовища. Особливий фокус зроблено на інноваційні підходи та стратегії, які спрямовані на зміцнення безпеки мережевих систем. Виконана робота не лише демонструє значний вклад у розвиток інструментів та методик кібербезпеки, але й вносить цінний внесок у забезпечення стійкості та надійності мережевих інфраструктур в умовах сучасних кібервикликів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Stallings, William. Network Security Essentials: Applications and Standards // Pearson. – Upper Saddle River, 2016.
2. Sanders, Chris; Smith, Jason. Applied Network Security Monitoring // Syngress. – Waltham, 2014.
3. Trost, Ryan. Practical Intrusion Analysis: Prevention and Detection for the Twenty-First Century // Addison-Wesley Professional. – Boston, 2009.
4. Chismon, David; Ruks, Martyn. Threat Intelligence: Collecting, Analysing, Evaluating // InfoSecurity. – Hoboken, 2015.
5. What is the Pyramid of Pain? [Електронний ресурс]. - Режим доступу: https://www.attackiq.com/glossary/pyramid-of-pain/
6. Mohay, George. Computer and Intrusion Forensics // Artech House. – Norwood, 2003.
7. Aumasson, Jean-Philippe; Demetrio, Luca. Network Traffic Analysis: Methods and Techniques // Springer. – Cham, 2020.
8. Diogenes, Yuri; Ozkaya, Erdal. Cybersecurity – Attack and Defense Strategies // Wiley. – Hoboken, 2018.
9. Seitz, Justin. Black Hat Python: Python Programming for Hackers and Pentesters // No Starch Press. – San Francisco, 2014.
10. Pease, Andrew. Threat Hunting with Elastic Stack: Solve complex security challenges with integrated prevention, detection, and response // Packt Publishing. – 2021
11. Importance of IOC Detection Rules [Електронний ресурс]. - Режим доступу: https://www.talanoscybersecurity.com/blogs/news/importance-of-ioc-detection-rules
12. Kim, Peter. The Hacker Playbook 3: Practical Guide To Penetration Testing // Secure Planet. – Los Angeles, 2018.
13. AlienVault OTX (Open Threat Exchange) [Електронний ресурс]. - Режим доступу: https://otx.alienvault.com/faq

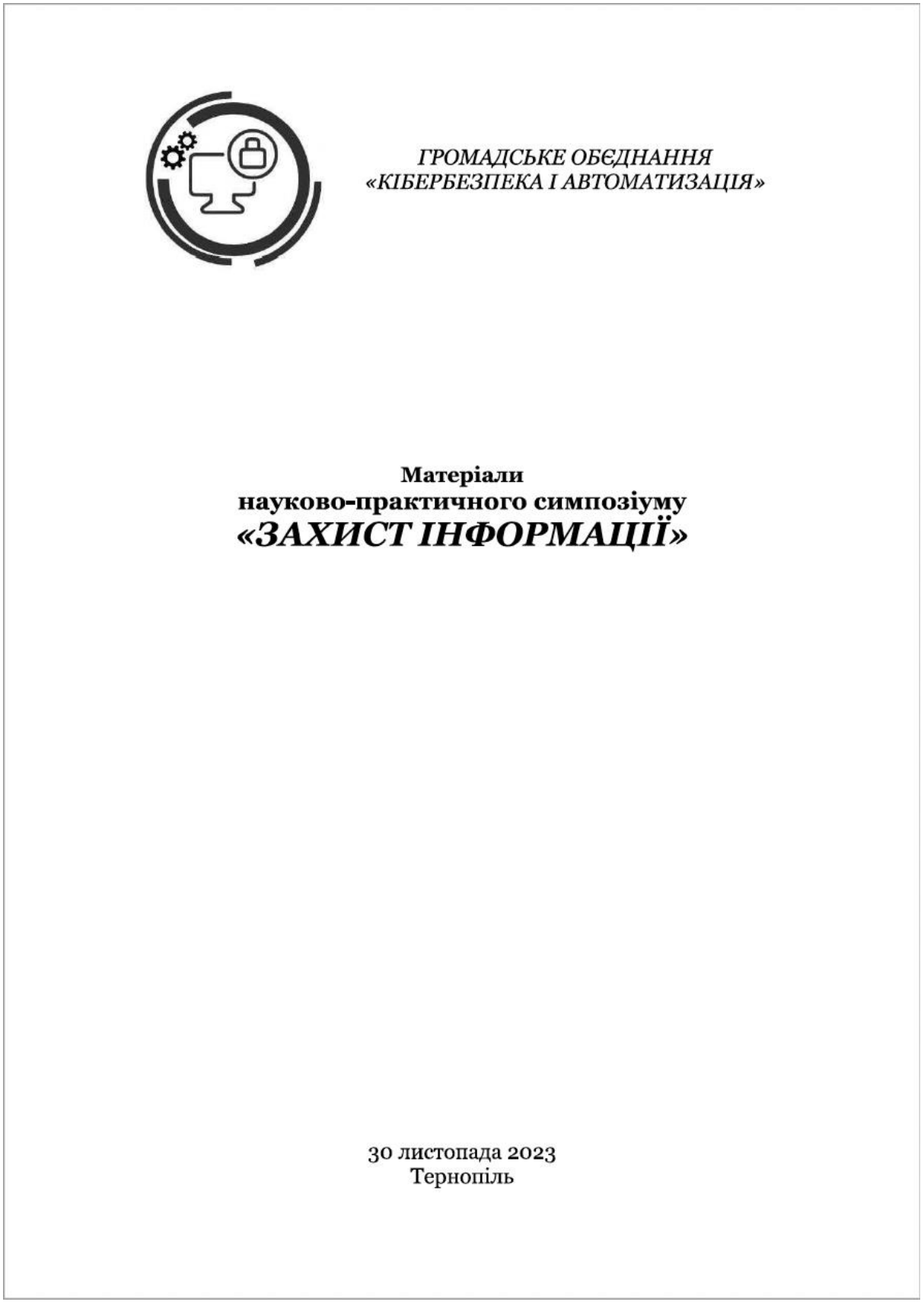
14. IP Geolocation As A Tool Against Cyberattacks [Електронний ресурс]. - Режим доступу: https://goabacus.com/ip-geolocation-as-a-tool-against-cyberattacks/
15. Indicators of compromise explained [Електронний ресурс]. - Режим доступу: https://www.microsoft.com/en-us/security/business/security-101/what-are-indicators-of-compromise-ioc
16. How Real-Time Cyber Threat Intelligence Sharing Enables Security Collaboration [Електронний ресурс]. - Режим доступу: https://cyware.com/security-guides/cyber-threat-intelligence/how-real-time-cyber-threat-intelligence-sharing-enables-security-collaboration-4151
17. Cyber Threat Information Sharing (CTIS) - Shared Cybersecurity Services (SCS) [Електронний ресурс]. - Режим доступу: https://www.cisa.gov/resources-tools/services/cyber-threat-information-sharing-ctis-shared-cybersecurity-services-scs
18. Кібербезпека: що це таке та чому це необхідно? [Електронний ресурс]. - Режим доступу: https://datalabsua.com/ua/cyber-security-what-is-this-and-why-it-is-important/
19. Advanced Threat Hunting [Електронний ресурс]. - Режим доступу: https://www.expertware.net/Solutions/Cyber-Security/Advanced-Threat-Hunting
20. Introducing query rules in Elasticsearch 8.10 [Електронний ресурс]. - Режим доступу: https://www.elastic.co/blog/introducing-query-rules-elasticsearch-8-10
21. Comparing Threat Intelligence Information with Windows Logs via Monitor [Електронний ресурс]. - Режим доступу: https://forum.opensearch.org/t/comparing-threat-intelligence-information-with-windows-logs-via-monitor/14021
22. Identify and remediate security threats to your business using security analytics with Amazon OpenSearch Service [Електронний ресурс]. - Режим доступу:

https://aws.amazon.com/ru/blogs/big-data/identify-and-remediate-security-threats-to-your-business-using-security-analytics-with-amazon-opensearch-service/

23. Система управління інформацією та подіями безпеки – SIEM [Електронний ресурс]. - Режим доступу: https://eska.global/blog/sistema-upravlinnya-informaciyeyu-ta-podiyami-bezpeki-siem
24. Detecting Malware/APT Through Automatic Log Analysis [Електронний ресурс]. - Режим доступу: https://www.radware.com/blog/security/2018/05/detecting-malware-apt-through-automatic-log-analysis/
25. Classification and Analysis of Malicious Code Detection Techniques Based on the APT Attack [Електронний ресурс]. - Режим доступу: https://www.mdpi.com/2076-3417/13/5/2894/html
26. Catch IP Address threats in your logs to analyze and mitigate them [Електронний ресурс]. - Режим доступу: https://blogs.oracle.com/observability/post/catch-ip-address-threats-in-logs

ДОДАТОК А

Копії публікацій



ГРОМАДСЬКЕ ОБЄДНАННЯ
«КІБЕРБЕЗПЕКА І АВТОМАТИЗАЦІЯ»

Матеріали
науково-практичного симпозіуму
«ЗАХИСТ ІНФОРМАЦІЇ»

30 листопада 2023
Тернопіль

Перелік використаних джерел.

1. Ransomware attack: What is it and how does it work? [Електронний ресурс]. - Режим доступу: https://nordvpn.com/uk/blog/what-is-ransomware/

2. StopRansomware Guide. [Електронний ресурс]. - Режим доступу: https://www.cisa.gov/stopransomware

УДК 004.056.5

КОСТЮК О. В.

Західноукраїнський національний університет

КЛЮЧОВІ АСПЕКТИ ТА ПЕРЕВАГИ ЦЕНТРАЛІЗОВАНОГО ЗБОРУ ТА ЗБЕРЕЖЕННЯ ЖУРНАЛІВ РОБОТИ ІНФРАСТРУКТУРИ

Вступ. У сучасному світі мережева інфраструктура стає все більш складною і розгалуженою, обслуговуючи потреби підприємств і організацій будь-якого масштабу. Кожного дня зростає кількість пристроїв, програм та мережевих послуг. З огляду на цю динаміку, важливо забезпечити ефективний моніторинг та управління інфраструктурою, щоб вчасно виявляти проблеми, забезпечувати безпеку та оптимізувати роботу [1].

Один із важливих аспектів сучасного моніторингу мережі - це збір та аналіз журналів подій (логів) з різних джерел в одній централізованій системі. Цей підхід дозволяє отримати повну картину подій, що відбуваються у мережі та системі, а також ефективно реагувати на проблеми та загрози [2].

Необхідність централізованого збору журналів стає більш актуальною в умовах зростаючої складності інформаційних систем та загроз цифровій безпеці. Розглянемо цю тему детальніше і визначимо, як цей підхід може сприяти покращенню ефективності моніторингу та безпеки мережі.

Мета. Дослідження переваги централізованого збору журналів у мережевому середовищі.

1. Огляд централізованого збору логів елементів інфраструктури

Збір логів - це процес перехоплення, обробки та зберігання журналів подій з мереж, інфраструктур та додатків. Ці журнали містять дані про помилки, зміни конфігурацій та інші важливі події. Процес включає збір логів з різних джерел, таких як мережеві пристрої та сервери, а також їх агрегацію й обробку для вилучення корисної інформації. Логи зберігаються у централізованій базі даних, деякі з них архівуються для виконання регуляторних вимог.

Різні програмні засоби використовуються для збору логів. До них належать ELK Stack для зберігання, пошуку, обробки та візуалізації логів, OpenSearch - безкоштовний аналог ELK з функціоналом аналізу в реальному часі, Elastic Beats для легковагового збору логів, та Syslog - стандартний протокол для UNIX-подібних систем і мережевих пристроїв.

2. Типи журналів, що використовуються для дослідження

Централізований збір логів має важливу роль у моніторингу та аналізі систем, мереж і додатків, надаючи доступ до важливої інформації про їхній стан. Журнали фіксують різні дані, включаючи зміни конфігурації, помилки, відмови, спроби злому, а також інциденти безпеки. Вони допомагають ідентифікувати потенційні конфлікти та незвичні активності, що можуть виникнути через помилки конфігурації або несанкціонований доступ.

Додатково, логи надають інформацію про використання системних ресурсів, таких як процесор, пам'ять, та дисковий простір, що є ключовим для оптимізації продуктивності та виявлення аномальної активності. Дані про активність користувачів, такі як їх входи та виходи, допомагають виявити несанкціоновану діяльність. Моніторинг мережевого трафіку важливий для виявлення аномалій, які можуть бути ознаками атак або проблем з безпекою.

Зібрані дані використовуються не тільки для виявлення та реагування на поточні проблеми, але й для створення регулярних звітів, що допомагають аналізувати тенденції та покращувати загальну ефективність та безпеку мережі та інфраструктури.

Для детального налаштування журналювання варто використати рекомендації та інструкції CIS 20 Benchmark. Ці рекомендації дозволять досягти максимального огляду та видимості інфраструктури, а також налаштувати відповідні метрики та мітки для відслідковування зловмисних дій.

3. Огляд реалізації системи централізованого збору журналів роботи

Архітектура системи централізованого журналювання складається з наступних елементів: джерела даних, сервер збору та обробки даних, сховище даних, інструменти візуалізації та аналізу (рисунок 1).

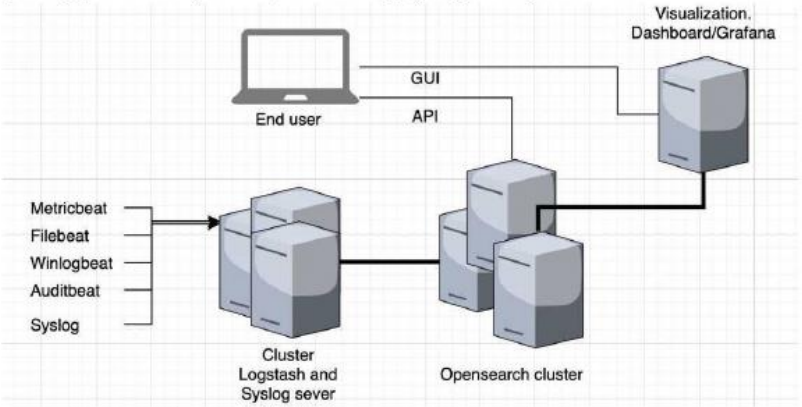


Рисунок 1 - Архітектура системи централізованого журналювання.

На початку цього процесу стоять джерела даних. Це можуть бути різні джерела, такі як сервери під управлінням Unix-подібних операційних систем,

Windows-сервери та інші мережеві пристрої. Щоб забезпечити збір всієї необхідної інформації, використовуються Filebeat, Winlogbeat - легкі агенти, які встановлюються на кожному джерелі даних, а також перенаправлення журналів роботи мережевих пристроїв та додатків на сервера syslog.

Основним центром обробки та агрегації журналів є сервер збору даних - Logstash. Цей сервер приймає дані від Filebeat, Winlogbeat, а також від мережевих джерел через протокол Syslog. Logstash відповідає за обробку, фільтрацію та перетворення даних перед їх відправленням у сховище даних.

Для зберігання журналів та легкого доступу до них використовується Opensearch. Це розподілене сховище даних, яке забезпечує масштабованість та надійність. Дані зберігаються у формі журналів і можуть бути легко розподілені та репліковані для забезпечення надійності.

Візуалізація даних виконується за допомогою таких інструментів як Opensearch Dashboard та Grafana. Вони дозволяють створювати графіки, звіти та дашборди, які допомагають моніторити журнали та виявляти потенційні проблеми. Окремою особливістю є доменна авторизація для доступу до інформаційних панелей Grafana, та можливість надавати гнучко налаштовані права виключно на необхідні поля при використанні Opensearch Dashboard.

Висновки.

Централізований збір журналів є ключовою компонентою сучасних систем моніторингу та безпеки мережі. Правильно спланована архітектура цього компоненту дозволяє ефективно зібрати, обробити та зберегти дані з різних джерел для подальшого аналізу та виявлення проблем.

Використання ELK Stack (Elasticsearch, Logstash, Kibana) та Beats дозволяє побудувати потужну систему збору та обробки журналів. Filebeat і Winlogbeat легко інтегруються з різними джерелами даних, а Logstash надає гнучкість у фільтрації та обробці цих даних. Opensearch забезпечує надійне зберігання та доступ до журналів, а інформаційні панелі Opensearch Dashboard та Grafana допомагають візуалізувати та аналізувати дані. Правильно спроектована архітектура з урахуванням заходів забезпечення безпеки гарантує надійність та конфіденційність оброблених даних. Ця система забезпечує можливість вчасно реагувати на події, виявляти проблеми та покращувати безпеку мережі, роблячи її більш стійкою до кіберзагроз та непередбачених ситуацій. Розроблення та налагодження такої архітектури варте зусиль, оскільки вона стає невід'ємною частиною сучасного інфраструктурного управління та кіберзахисту.

Перелік використаних джерел.

1. Дзен логування. Як полюбити свої логи та почати жити. [Електронний ресурс].- Режим доступу: https://dou.ua/lenta/columns/about-logging/

2. Centralized Logging with ELK Stack (Elasticsearch, Logstash, and Kibana) On Ubuntu 14.04. [Електронний ресурс].- Режим доступу: https://www.digitalocean.com/community/tutorial-series/centralized-logging-with-elk-stack-elasticsearch-logstash-and-kibana-on-ubuntu-14-04

3. Loki - збирання логів, використовуючи підхід Prometheus [Електронний

ресурс].- Режим доступу: https://prohoster.info/uk/blog/administrirovanie/loki-sbor-logov-ispolzuya-podhod-prometheus

4. The Elastic Stack (ELK): how to set up centralized logging with Logstash, Elasticsearch & Kibana. [Електронний ресурс].- Режим доступу: https://kruschecompany.com/logstash-elasticsearch-kibana/

УДК 004.056.55

КУРТЯК А.М., САВРІЙ С.В.

[1]Західноукраїнський національний університет

ФОРМУВАННЯ ВИМОГ ДО КРИПТОГРАФІЧНИХ АЛГОРИТМІВ НА ОСНОВІ ДИНАМІЧНОГО ХАОСУ

Вступ. Два останні десятиліття відомі надзвичайним інтересом до можливості використання динамічного хаосу для шифрування даних на концептуальному рівні [1]. Між хаотичними та криптографічними системами є своєрідний взаємозв'язок [2]. Тому і в нелінійній динаміці, і в криптографії матеріалізується нелінійне перетворення інформації. На практичному рівні між хаотичними та криптографічними системами є своя схожість. У класичних роботах К. Шеннона також можна знайти згадку про хаотичні сигнали.

Перетворення, які гарно перемішують, часто досягаються шляхом повторення двох простих некомутованих операцій. Якщо добре перемішати, то перетворення функції ускладнюються за рахунок підвищення чутливості всіх змінних. Маленьке збурення у будь-якій з них призводить до значної зміни кінцевого результату. Все вищесказане і визначає актуальність даної тематики.

Мета: формування вимог до криптографічних алгоритмів на основі динамічного хаосу.

1. Криптографія і хаос

Потокові системи шифрування дуже чутливі до пропусків знаків шифрованого тексту, тому в них доводиться стежити за узгодженням порядку застосування перетворень при зашифруванні та розшифруванні. Шляхом введення в повідомлення, що передається, спеціальних маркерів, що безсумнівно, призведе до ускладнення самої системи, можна забезпечити це узгодження. Інше рішення полягає у застосуванні систем із самосинхронізацією. Тут пропущений знак впливає тільки на кілька послідовних станів. Таким чином, використання заданої властивості динамічного хаосу в традиційних криптографічних схемах допоможе спростити криптографічну процедуру.

Використання хаотичної динаміки у системах шифрування може дати останнім нову якість. Така криптографія є цілочисельною. І повідомлення, і гама (якщо йдеться про шифри гамування), з якою воно додається за модулем або по операції XOR, є послідовностями, наприклад, нулів та одиниць. Хаотичні сигнали є безперервними. Їх реалізація на цифрових процесорах чи комп'ютерах



ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВОДНОГО ГОСПОДАРСТВА ТА ПРИРОДОКОРИСТУВАННЯ
ГАЛИЦЬКИЙ ФАХОВИЙ КОЛЕДЖ ІМ. В. ЧОРНОВОЛА

КІБЕРБЕЗПЕКА ТА КОМП'ЮТЕРНО-ІНТЕГРОВАНІ ТЕХНОЛОГІЇ (КБКІТ – 2023)

науково-практична конференція
молодих вчених, аспірантів та студентів

29–31 серпня 2023
Тернопіль

УДК 004.056.5

Костюк О. В.[1]

[1]*Західноукраїнський національний університет*

ПОБУДОВА СИСТЕМ ВИЯВЛЕННЯ МЕРЕЖЕВИХ КІБЕРЗАГРОЗ

Вступ. Мережева безпека є однією з найбільш важливих тем сучасного інформаційного суспільства. З кожним днем мережі стають все більш пов'язаними з нашим повсякденним життям і стають важливим інструментом для роботи багатьох підприємств і установ. Тим не менш, зростаюча залежність від мережевих технологій також призводить до збільшення небезпек і ризиків для мережевої безпеки.

Сучасні зловмисники використовують різноманітні методи та техніки для атак на мережеву інфраструктуру, спрямовану на крадіжку конфіденційної інформації, вимагання викупу або завдання збитків. Ці атаки можуть мати серйозні наслідки для бізнесу, громадської безпеки та особистої приватності. Тому вивчення та розуміння найпоширеніших атак на мережеву інфраструктуру стає вельми актуальним завданням для спеціалістів з кібербезпеки.

Метою даної статті є проведення аналізу найпоширеніших атак на мережеву інфраструктуру, виявлення їх характеристик, методів викриття та захисту. В дослідженні будуть розглянуті різні типи атак, включаючи DDoS-атаки, фішинг, злами систем та інші. Відомості, отримані в результаті цього аналізу, можуть бути корисні для підвищення ефективності заходів забезпечення безпеки мережевої інфраструктури та захисту від потенційних загроз [1].

Мета. Дослідження найпоширеніших атак та захисту мережевого периметру.

1. Огляд найпоширеніших атак та їх ідентифікація

В сучасному цифровому світі, де мережі стають все більш складними та розгалуженими, кібербезпека стає надзвичайно важливим аспектом для організацій та користувачів. Зловмисники постійно шукають нові способи атак і експлуатації вразливостей в мережевій інфраструктурі, щоб завдати шкоди або здійснити крадіжку цінних даних. Тому необхідно розглянути деякі з найбільш поширених мережевих атак, їх характеристики та методи ідентифікації [2, 3].

1) Man-in-the-Middle (MitM) – це хитрий метод, при якому атакувальник витісняє себе між легітимними комунікуючими сторонами в мережі. Це відкриває можливість зловмиснику перехоплювати та навіть змінювати передачу даних між цими сторонами, що призводить до витоку конфіденційної інформації та загрози безпеці.

2) ARP Spoofing (ARP Poisoning) – також відомий як ARP Poisoning, є прийомом, при якому атакувальник надсилає фальшиві ARP-запити для змусу системи помилково відповідати на свої запити. Це може призвести до перенаправлення мережевого трафіку через систему зловмисника для перехоплення або модифікації даних.

3) DNS Spoofing – це атака, під час якої зловмисник фальсифікує DNS-

відповіді для перенаправлення користувачів на фальшиві веб-сайти або сервери. Цей прийом може використовуватися для перехоплення обміну інформацією та отримання конфіденційних даних.

4) Phishing – це атака, при якій зловмисник відправляє фішингові повідомлення, які схожі на легітимні комунікації від відомих організацій або сервісів. Метою таких повідомлень є шахрайське отримання конфіденційних даних, таких як паролі або особисті дані.

5) SYN Flood – це атака на рівні транспортного протоколу, під час якої зловмисник надсилає велику кількість SYN-запитів на сервер, не завершуючи 3-вей рукостискання. Це може призвести до вичерпання ресурсів сервера та його недоступності.

Ці атаки є лише кількома прикладами загроз, які існують у сфері мережевої безпеки. Розглянемо способи ідентифікації та захисту від цих атак для забезпечення безпеки мережевого периметру.

2. Мережевий моніторинг для виявлення кіберзагроз

Мережевий моніторинг є важливою складовою системи кібербезпеки організації. Його головною метою є виявлення незвичної активності в мережі, яка може свідчити про кіберзагрози або інші аномалії. У цьому розділі розглянемо інструменти та методи мережевого моніторингу та виявлення потенційних кіберзагроз.

Першим кроком у реалізації мережевого моніторингу є встановлення спеціалізованої системи, яка буде відповідальною за перехоплення та аналіз мережевого трафіку. Одним з найпоширеніших інструментів для цього є Wireshark. Ця програма дозволяє перехоплювати пакети даних, що проходять через мережу, та проводити їх аналіз.

Після встановлення системи моніторингу необхідно правильно сконфігурувати її для відповідності потребам конкретної організації. Це включає в себе вибір типів трафіку для моніторингу, налаштування фільтрів для виділення релевантної інформації та визначення правил виявлення загроз.

Система моніторингу постійно перехоплює трафік, що проходить через мережу. Спеціалісти з кібербезпеки або мережеві адміністратор, що відповідають за безпеку мережі, аналізують цю інформацію з метою виявлення незвичної активності. Вони спостерігають за подіями та діями, які можуть свідчити про потенційну кіберзагрозу чи інші аномалії.

Важливою частиною мережевого моніторингу є виявлення патернів та загроз. Процес збору та аналізу патернів атак включає в себе не тільки моніторинг актуальних загроз, але й прогнозування потенційних атак на основі зібраної інформації. Використання даних із відкритих джерел, як от бази даних загроз, публікації від кібербезпекових фірм, та аналіз соціальних мереж, дозволяє формувати актуальні та ефективні патерни для виявлення загроз.

Також важливим є аналіз локальних логів та інформації. Дані про попередні атаки та інциденти безпеки власної мережі є цінним джерелом для створення специфічних правил виявлення, дозволяючи адаптувати систему до особливостей конкретної мережевої інфраструктури. Наприклад, підозрілий

збільшений обсяг запитів до конкретного сервера може вказувати на DDoS-атаку

На основі результатів мережевого моніторингу можуть бути прийняті необхідні заходи безпеки. Це може включати в себе ізоляцію атакованих пристроїв, блокування певних IP-адрес чи зміну прав доступу. Важливо оперативно реагувати на виявлені загрози для мінімізації можливих ризиків.

Мережевий моніторинг є ключовим елементом в забезпеченні безпеки мережі та виявленні кіберзагроз. Застосування відповідних інструментів та правильна організація процесу моніторингу допомагають захищати організацію від потенційних кібератак та забезпечувати безпеку мережевого середовища.

Висновок. Дослідження показало, що система мережевого моніторингу для виявлення кіберзагроз є фундаментальним інструментом у сучасній кібербезпеці. Її розгалужена архітектура, що охоплює сенсори мережевого трафіку, аналітичні сервери, системи виявлення інцидентів (SIEM), правила виявлення загроз, та інтегровані системи сповіщень та реагування, формує багаторівневий захист від різноманітних кіберзагроз. Збір та аналіз патернів атак, як з відкритих, так і з внутрішніх джерел, розширює можливості для раннього виявлення та ефективного реагування на загрози. Автоматизація процесів виявлення та реагування дозволяє швидко ідентифікувати та локалізувати потенційні атаки, знижуючи ризики та вплив на інфраструктуру організації.

У цілому, розвиток та впровадження комплексних систем мережевого моніторингу є ключовим для забезпечення цифрової безпеки в епоху постійно зростаючих кіберзагроз. Такий підхід не лише забезпечує захист від поточних загроз, але й адаптується до майбутніх викликів, гарантуючи стабільність та безпеку мережевих систем в довгостроковій перспективі.

Перелік використаних джерел.

1. Як використовувати Wireshark для захоплення, фільтрації та перевірки пакетів. [Електронний ресурс].- Режим доступу: https://ua.phhsnews.com/articles/howto/how-to-use-wireshark-to-capture-filter-and-inspect-packets.html

2. Optimal monitoring and attack detection of networks modeled by Bayesian attack graphs. [Електронний ресурс].- Режим доступу: https://cybersecurity.springeropen.com/articles/10.1186/s42400-023-00155-y

3. Snort Tutorial and Practical Examples. [Електронний ресурс].- Режим доступу: https://hackertarget.com/snort-tutorial-practical-examples/