НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки

До захисту допущено
Завідувач кафедри
_____ **Дмитро ЛАНДЕ**
(підпис)
«_____»_____2022 р.

Дипломна робота

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Системи, технології та математичні методи кібербезпеки»
спеціальності 125 «Кібербезпека»

на тему: Структурні методи вивчення поверхні атак
Виконав (-ла): здобувач вищої освіти **IV** курсу, групи ФБ-81_____
(шифр групи)

Кудін Іван Антонович _____ _____
(прізвище, ім'я, по батькові) (підпис)

Керівник к.е.н., доцент кафедри ІБ Ткач Володимир Миколайович _____
_____
(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові) (підпис)

Рецензент _____ _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище, ім'я, по батькові)
(підпис)

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.
Здобувач вищої освіти _____
(підпис)

Київ – 2022 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки

Рівень вищої освіти – перший (бакалаврський)
Спеціальність– 125 «Кібербезпека»
Освітньо-професійна програма «Системи, технології та математичні методи кібербезпеки»

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ ДмитроЛАНДЕ
(підпис)
«___»_____2022 р.

ЗАВДАННЯ
на дипломну роботу здобувачу вищої освіти

Кудін Іван Антонович _____
(прізвище, ім'я, по батькові)

1. Тема роботи: Структурні методи вивчення поверхні атак,

керівник роботи к.е.н., доцент кафедри ІБ Ткач Володимир Миколайович
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від « »_____2022 р. №

2. Термін подання здобувачем вищої освіти роботи 10 червня 2022 р.

3. Вихідні дані до роботи: наукова література та відкриті джерела за темою; попередні дослідження за темою; база даних MitreATT&CK від MITRE.

4. Зміст роботи: аналіз існуючих джерел даних про сучасні інтелектуальні атаки (APT-атаки) на інформаційні системи; теорія сучасних APT-атак; аналіз методів їх струтурізації та класифікації; статистичні дослідження існуючих даних про інтелектуальні атаки, що використовуються сучсними хакерскими угруповуваннями; розробка структурних методів аналізу інтелектуальних атак; практична демонстрація працездатності запропонованих методів виявлення інтелектуальних атак.

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо): презентація на тему «Структурні методи вивчення поверхні атак».

6. Дата видачі завдання18 жовтня 2021 року.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів дипломної роботи	Примітка
1	Вибір теми роботи	20.09.2021 – 15.10.2021	
2	Отримання завдання	18.10.2021	
3	Вивчення літератури за темою, вибір літературних джерел	19.10.2021 – 12.01.2022	
4	Вивчення технік ухилення, написання першого розділу	13.01.2022 – 20.02.2022	
5	Вивчення протидії технікам, написання другого розділу	21.02.2022 – 15.04.2022	
6	Аналіз отриманих знань та результатів	16.04.2022 – 01.05.2022	
7	Проходження переддипломної практики, написання третього розділу	02.05.2022 – 30.05.2022	
8	Отримання допуску до захисту	01.06.2022	
9	Створення презентації для захисту дипломної роботи	03.06.2022 – 13.06.2022	
10	Передзахист дипломної роботи	14.06.2022	
11	Доопрацювання дипломної роботи та презентації	15.06.2022 – 19.06.2022	
12	Захист дипломної роботи	21.06.2022	

Здобувач вищої освіти _____ Іван КУДІН
(підпис) (**Власне ім'я, ПРІЗВИЩЕ**)

Керівник роботи _____ Володимир ТКАЧ
(підпис) (**Власне ім'я, ПРІЗВИЩЕ**)

РЕФЕРАТ

Робота складається з 3-х розділів, містить 19 іллюстрацій, 5 таблиць, 22 літературні посилання, обсяг роботи – 60 сторінок.

Завданням роботи є: аналіз існуючих джерел даних про сучасні інтелектуальні атаки (APT-атаки) на інформаційні системи; теорія сучасних APT-атак; аналіз методів їх струтурізації та класифікації; статистичні дослідження існуючих даних про інтелектуальні атаки, що використовуються сучсними хакерскими угруповуваннями; розробка структурних методів аналізу інтелектуальних атак; практична демонстрація працездатності запропонованих методів виявлення інтелектуальних атак.

Метою роботи є вивчення особливостей побудови поверхні атак сучасних інформаційних систем, що дозволить підвищити ефективність виявлення атак сучасними системами захисту інформації.

Об’єктом дослідження є: бази даних та сховища даних атак на сучасні інформаційні системи. Предметом дослідження є: структура поверхні атак на інформаційні системи.

Наукова новизна полягає у тому, що до теперішнього часу не проводилось вивчення існуючої множини APT-атак, які використовуються хакерськими угрупуваннями, зведені в існуючі інформаційні сховища та бази даних атак (таких як база даних з ATT&CK від MITRE) з метою виявлення їх алгебраїчної структури.

Ключові слова: поверхня атак, MitreATT&CK, CVE, NVD, CWE, CAPEC, задача про покриття множини, статистичний аналіз, базові атаки.

ABSTACT

The work consists of 3 sections, contains 19 illustrations, 5 tables and 22 literature links, volume of work – 60 pages.

The task of the work is: analysis of existing data sources on modern intellectual attacks (APT-attacks) on information systems; theory of modern APT-attacks; analysis of methods of their structuring and classification; statistical researches of existing data on intellectual attacks, used by learned hacker groups; development of structural methods of analysis of intellectual attacks; practical demonstration of the working methods of the proposed methods of detection.

The aim of the work is to create methods of analysis of the attacks surface of information systems, which will allow to increase the efficiency of their detection by modern security systems.

The object of the research is: databases and storehouses of attack data on modern information systems. The subject of the research is: structure of the attacks surface of information systems.

The scientific novelty is that until now there has been no study of the existing set of APT-attacks used by hacker groups, brought into existing information stores and attack databases (such as the database with ATT&CK from MITRE) for the purpose of detecting their algebraic structure.

Key words: attack surface, MitreATT&CK, CVE, NVD, CWE, CAPEC, problem of coverage of sets, statistical analysis, elemetary attacks.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ОС – операційна система

ПЗ – програмне забезпечення

CVE – Common Vulnerabilities and Exposures

CWE – Common Weakness Enumeration

CAPEC – Common Attack Pattern Enumeration and Classification

NVD – National Vulnerability Database

IDS – Intrusion detection system

IoC – Indicator of Compartmentation,

IPS – Intrusion prevention system

SIEM – System information and event management

СВВ – системи виявлення вторгнень

Базова атака – кібератака, яка при данному рівні деталізації розглядається як «чорна скринька»

ВСТУП

Вивчення поверхні атак (за перекладом за змістом з NIST 800-53 [1] «поверхня атак – множина вразливих елементів системи, до яких можуть бути застосовані атаки») – один з найбільш важливих задач як при проектування систем захисту, так і застосувань систем захисту для вже існуючих інформаційних систем. Фактично за національною підходом до побудови захищених систем [2-4], поверхня атак є результатом вивчення вразливості інформаційної системи, її моделі загроз та, частково, аналізу ризику (в частині, що стосується визначення потенціальної можливості реалізації загроз за обраною моделлю супротивника). Від особливості поверхні атак суттєво залежить ефективність систем захисту, а від її потужності – витрати на системи захисту. При цьому поверхню атак можна визначати як від вразливості компонентів системи (через CWE-CVE до загроз і атак) так і з «іншого боку» - від атак. Перший підхід часто застосовується при проектуванні захищених систем, але при створенні систем захисту інформації (наприклад, таких як IPS) його застосування неможливе за рахунок пракиичної складності побудови множини всіх можливих атак, тому застосовується другий підхід – визначаються всі **існуючі** потенційні атаки на інформаційну систему. При другому підході визначення поверхні атак можливе тільки за рахунок вивчення особливостей структури побудови відомих баз даних існуючих кібератак та атак на систему безпеки інформації, які здійснювались хакерськими угрупуваннями чи створені спільнотою кібербезпеки, чому і присвячені дослідження, результати яких представлені в даній роботі.

Актуальність роботи визначається необхідністю створення практичних методів оцінки поверхні атак, які можуть бути практично застосовані для широкого класу інформаційних систем на основі дослідження існуючих баз атак.

Мета роботи: вивчення особливостей побудови поверхні атак сучасних інформаційних систем, що дозволить підвищити ефективність виявлення атак сучасними системами захисту інформації.

Мета роботи реалізується за допомогою наступних **задач дослідження**:

1. Провести аналіз існуючих джерел даних про сучасні інтелектуальні атаки (APT-атаки) на інформаційні системи.

2. Провести аналіз недоліків існуючих методів вивчення поверхні атак та виявити галузь їх застосування.

3. Провести аналіз методів струтурізації та класифікації атак; статистичні дослідження існуючих даних про інтелектуальні атаки, що використовуються сучсними хакерскими угруповуваннями та іншими базами даних, які використовуються спільнотою фахівців з кібербезпеки.

4. Розробити структурні методи аналізу інтелектуальних атак.

5. Навести практичне демонстрацію працездатності запропонованих методів виявлення інтелектуальних атак.

Предметом дослідження є: структурні властивості множини кібератак та атак ні безпеку інформації, доступних з баз даних, які використовуються спільнотою фахівців з кібербезпеки. **Об'єктом дослідження** є: множини кібератак та атак ні безпеку інформації, доступних з баз даних, які використовуються спільнотою фахівців з кібербезпеки.

Методом дослідження є статистичні методи аналізу, методи дискретної оптимізації та методи аналізу великих даних.

Наукова новизна полягає у тому, що методи аналізу поверхні атак при наявності універсальних баз даних атак та без прив'язки до ресурсів конкретних інформаційних систем практично не досліджувались, а статистичні та структурні дослідження баз даних з ATT&CK від MITRE, CAPEC, розрізнених даних про техніки атак різноманітних хакерських угрупувань не проводились.

Практичне значення одержаних результатів полягає в тому, розроблені методи вирішення задачі мінімального покриття поверхні атак можуть бути використані для підвищення ефективності рішення таких практичних задач захисту інформації:

розробки ефективних правил для «налаштування за замовченням» систем виявлення вторгнень;

верифікації результатів стандартного програмного забезпечення оцінки поверхні атак (наприклад такого як Microsoft Attack Surface Analyzer);

створення нових ефективних методів виявлення вторгнень.

Окремі результати роботи були отримані автором в рамках досліджень за грантовою угодою CRDF № G -202102-67499 від 19.07.2021року «Розподілена система раннього виявлення вторгнень та оцінки кібербезпеки».

1 ІСНУЧІ МЕТОДИ ВИЗНАЧЕННЯ ПОВЕРХНІ АТАК ДЛЯ РІЗНИХ ЗАВДАНЬ ІНФОРМАЦІЙНОЇ ТА КІБЕРБЕЗПЕКИ

1.1 Існуючи відминності термінологій та підходів національної та анго-американської шкіл з кібербезпеки

Аналіз чинних нормативних документів України та стандартів США з проектування та оцінки систем захисту інформації дозволяє зробити висновок про дещо відмінні підходи до тлумачення термінів «поверхня атак». Так згідно документу NIST 800-53 Security and Privacy Controls for Information Systems and Organizations [1] термін «поверхня атаки» або «attack surface» визначається як

Означення 1.

The set of points on the boundary of a system, a system component, or an environment where an attacker can try to enter, cause an effect on, or extract data from, that system, component, or environment.

Якщо застосувати переклад за змістом, то отримуємо наступне

Означення 1.1.

Поверхня атак – множина вразливих елементів системи, до яких можуть бути застосовані атаки.

За національною підходом до побудови захищених систем [2-4], цей термін є близчим до моделі вразлівості системи і моделі загроз інформаційної системи, аніж саме до атаки, **як реалізованої загрози.** До того, оскільки в теоретичних моделях, які будуть стисло розглянуті нижче, атаки також пов’язуються із метою атак, ресурсами, цінністю цих ресурсів, а отже – з аналізом ризику (в частині, що стосується визначення потенціальної можливості реалізації загроз за обраною моделлю супротивника), практичне застосування оцінок поверні атаки при національноу підході можливе як правило при проетуванні систем захисту або порівнні деяких систем за захищеностю.

Це не є дуже застосованим, коли йдеться про задачу виявлення атак і оцінки ефективності систем виявлення атак, коли саме про множину атак наявні

певні відомості та потрібно оцінити структурні властивості цієї множини для мінімізації кількості атак та вразливих місць цілого класу систем, які підлягають контролю.

1.2 Існуючи теоретичні моделі оцінки поверхні атак та їх характеристики

Розглянемо існуючі теоретичні моделі дослідження поверхні атак відповідно до робіт [2-4].

Як правило інформаційна система представлена у вигляді моделі кіцевого автомату, як набір чотирьох множин $Q =< S, A, I, T >$, де S – множина станів системи, A – множина системних дій, I – множина початкових станів, T – множина зв'язків між діями та станами (відношення дій та станів системи). При такому підході

Означення 2 [4].

Атака - це скінченна послідовність виконуваних дій $a_1, \dots, a_i, \dots, a_n$ така, що: $a_i \in A$, для $\forall i = \{1,2, \dots, n\}$, $a_1 \in I$, $\exists j = \{1,2, \dots, n\}, a_j \in A_S$ та мета дій досягненута у стані, досягнутому скінченим автоматом Q після виконання a_n. При цьому A_S – множина «небезпечних дій системи», тобто послідовність проходить дію, яка призводить до атаки або завершується в небезпечному стані системи.

Тоді визначення поверхні атак є наступним [4]:

Означення 3 [4].

Поверхня атаки системи- це пара $Q =< A_S, \bigcup_{a \in A_S} Res(a) >$, де перший компонент – це множина системних дій, а другий – це об'єднана множина ресурсів, визначена для кожної системної дії $a \in A_S$.

При застосуванні такого підходу можна провести дослідження та навіть вимірювання поверхні атак наступним чином [4]:

1. Провести ідентифікацію всіх цінних ресурсів системи, які є потенційними цілями атаки як$\bigcup_{a\in A_S} Res(a)$. Нехай $Type$ – це множина типів всіх цих ресурсів.

2. Задаючи множину $Prop$ властивостей, цікавих для зловмисника, згенерувати ієрархію типів на множині $Type$. Кожен вузол в цій ієрархії типів є класом атаки системи. Нехай $Attack\ Class$ – множина класів атак.

3. Визначимо вагову функцію $F: Attack\ Class \rightarrow [0,1]$, щоб призначити ваги для кожної атаки, ідентифікованої на кроці 2. Ваги представляють собою імовірності атаки. Клас атаки з більшою вагою показує, що ресурси цього класу є більш привабливими для атакуючого, чим ресурси атаки із нижчою вагою. Інший шлях призначення вагових коефіцієнтів: вищі ваги відповідають атакам тих класів, які частіше трапляються.

4. Порівняти дві версії системи, A та B по класам атак. Є багато способів такого порівняння. Один спосіб – порахувати кількість проявів кожної атаки в обох версіях та порявняти числа. Чим вищий показник для даного класу атаки, тим більше виявляє себе атака цього класу. Інший шлях – поєднати ваги, визначені в п.3, як ваги цих показників, тобто ми можемо порахувати зважену суму проявів кожної атаки, тобто поверхня атаки для ресурса типу $T \in Type$ з класами атак $S_1, S_2, \dots, S_k$ та їх підтипами дається як $P_k = \sum_{i=1}^{k} n(S_i) \cdot w_i$, де $n(S_i)$ – кількість випадків атаки класу S_i та w_i – вага, надана S_i на кроці 3.

Застосування наведеного підходу до визначення поверхні атак має певні обмеження.

По-перше, ваги атак прив’язані або до цінності ресурсів системи, що відразу звужує галузь застосування методу, або до частоти атак, яку невідомо звідки отримувати.

По-друге, виділення типів атак пов’язане також до цінності ресурсів системи, а звідси – не є універсальним.

По-третє, виділення множини атак системи як правило або робиться за ланцюжком (вразливість (спочатку CWE, потім CVE) – загроза – ймовірність реалізації – атака), а тому також не є універсальним.

Загальний висновок – розглянутий підхід орієнтований на проектування або оцінку захищеності конкретних інформаційних систем, а не на виявлення атак.

Тому для цілей виявлення атак будемо застосовувати «обернену» постановку задачі – будемо вивчати структуру існуючої множини відомих кібератак, з метою вивчення структурних закономірностей таких атак, виділення з них елементарних (базових) атак. Після виділення базових атак та дослідження чи утворює множина атак певну алгебраїчну структуру відносно операцій над цими базовими атаками, ми вирішуємо задачу про покриття множини атак мінімальною кількістю підмножин атак, складених із цих базових атак. Це і є рішенням задачі мінімізації кількості правил виявлення атак для системи виявлення вторгнень.

1.2 ІСНУЮЧИ РЕЗУЛЬТАТИ У СФЕРІ ВИВЧЕННЯ СТРУКТУРИ КІБЕРАТАК

Для аналізу існуючих методів класифікації кібератак знадобляться власне офіційні ресурси на яких викладені ці класифікації. У даній роботі це:

- Mitre ATT&CK [5];
- CVE (Common Vulnerabilities and Exposures) [6];
- NVD (National Vulnerability Database) [7];
- CWE (Common Weakness Enumeration) [8];
- CAPEC (Common Attack Pattern Enumeration and Classification) [9].

Окремо від класифікацій вище, варто згадати Lockheedkillchain [10], як одну з перших спроб формально описати дії злочинця під час кібератаки, що отримала розвиток у матриці MITRE ATT&CK, та не так давно у 2017 році, коли Paul Pols описав Unified Cyber Kill Chain[7]. Гарне порівняння цих трьох концепцій можна знайти у [8].

Крім цих ресурсів, для виявлення недоліків сучасних класифікацій стане у нагоді робота [9]. У роботі [10] детально розглянуто однин з основних недоліків– відсутність достатньо чіткого зв'язку між вищезгаданими ресурсами та запроновано шляхи для її вирішення.

Для практичного застосування класифікацій сучасних кібератак обрано атаки групи Armagedon, які детально описані у [11].

Мабуть найбільш відомою серед фахівців є класифікація CVE [6] (Common Vulnerabilities and Exposures), проект був запущенний американською некомерційною організцією Mitre у 1999. Це список вразливостей з коритким описом до кожної з них, що включає вразливі версії программного забезпечення чи операційних систем з цією вразливісю, вендора, тип вразливості. Кожній вразливості присвоєно ідентефікаційний номер та рік у якому її було знайдено. Тобто, строго кажучи CVE це класифікація вразливостей, а не кібератак, але найточнішим словом мабуть буде "словник". Проте, її можна розглядати і як класифікацію кібератак, яка відповідає на питання "Як можна атакувати?"(те чи

інше програмне забезпечення). Головними перевагами CVE є її розповсюдженість та велика швидкість оновлення.

У 2005 році NISTзапустив похідний від CVE проект NVD [7], який значно спростив роботу із цією класифікацією. У NVD додали зручний пошук за вендором, програмним продуктом, роком та типом вразливості. Важливою інновацією стало введення рейтингу для кожної вразливості, що призначається експертним методом.

У 2005 році також з'явилася класифікація CWE [8] (Common Weackness Enumeretion). На відміну від CVE це класифікація недоліків програмного забезпечення, деякі з яких можуть призвести до критичних вразливостей. Вся класифікація має вигляд дерева, коренями у якому є так званні представлення. Це найбільш широкі класси, основні з них – концепції розробки, концепції архітектури, концепції досліджень. До концепцій розробки відносять недоліки, що виникають юезпосередньо при розробці програмного забезпечення, відповідно концепції архітектури – при його проектуванні. Концепції досліджень спрямовані на абстрактні поняття, формальні визначення.

Кожна CWE має ідентифікатор id(CWE-125) та багато супутньої інформації, з якої у контексті поставленої задачі цікавими є її приклади, пов'язані CWE та CVE які утворюються внаслідок наявності цього CWE. Проте недоліком цієї класифікації на відміну від CVE є відносно низька швидкість оновлення, головною перевагою – логічний зв'язок з CVE та CAPEC.

Класифікація шаблонів атак CAPEC [9] (Common Attack Pattern Enumeration and Classification), започаткована у 2007 році має аналогічну до CWE структуру дерева. Найбільші його класи називаються механізми атак та об'єкти атак. У кожному з цих классів шаблони атак розділени на категорії. Після категорій йдуть беспосередньо шаблони атак, що також мають кілька рівнів деталізації: від найбільш загільних Meta Attack Pattern до конкретних атак – Detailed Attack Pattern. Наприклад у представленні механізми атак є категорія – маніпуляція структурами данних, у цій категорії є Meta Pattern маніпуляція з буфером, у ньому у свою чергу Standart Attack Pattern Buffer Overflow, у якому

містяться детальні описи найбільш конкрктних Detailed Attack Patterns цієї відомої атаки. Шаблон атак, містить багато супутньої інформації: опис, передумови для здійснення такої атаки, небохідні навички та ресурси у зловмисника. Головним недоліком CAPEC є відсутність прямого зв’язку з CVE.

Три розглянуті вище класифікації є достатньо повними та загальноприйнятими у кібербезпеці, проте жодна з них не відповідає на важливе питання "З якою ціллю зловмисник виконує ту чи іншу дію, експлуатує вразливість?".

Тож тепер, слід розглянути розглянути матрицю MitreATT&CK, найбільш повну класифікацію кібератак насьогодні.

MitreATT&CK неможливо розлядати откремо від Killchain. Як вже згадувалося вище, це загальний опис дій під час здійснення сучасної кібератаки. Вперше термін Killchain у контексті кібербезпеки був запропонований корпорацією LockheedMartin, тому іноді сьогодні поряд з більш розповсюдженою назвою CyberKillChain можна побачити LoockheedKillChain. LockheedMartin запропонувала концепцію опису послідовності дій кіберзлочинця з 7 простих етапів: розвідки, озброєння, доставки, експлуатації, інсталяції, контролю та кінцевої мети зловмисника. Візуальне представлення CyberKillchain можна побачити на рисунку 1.1:

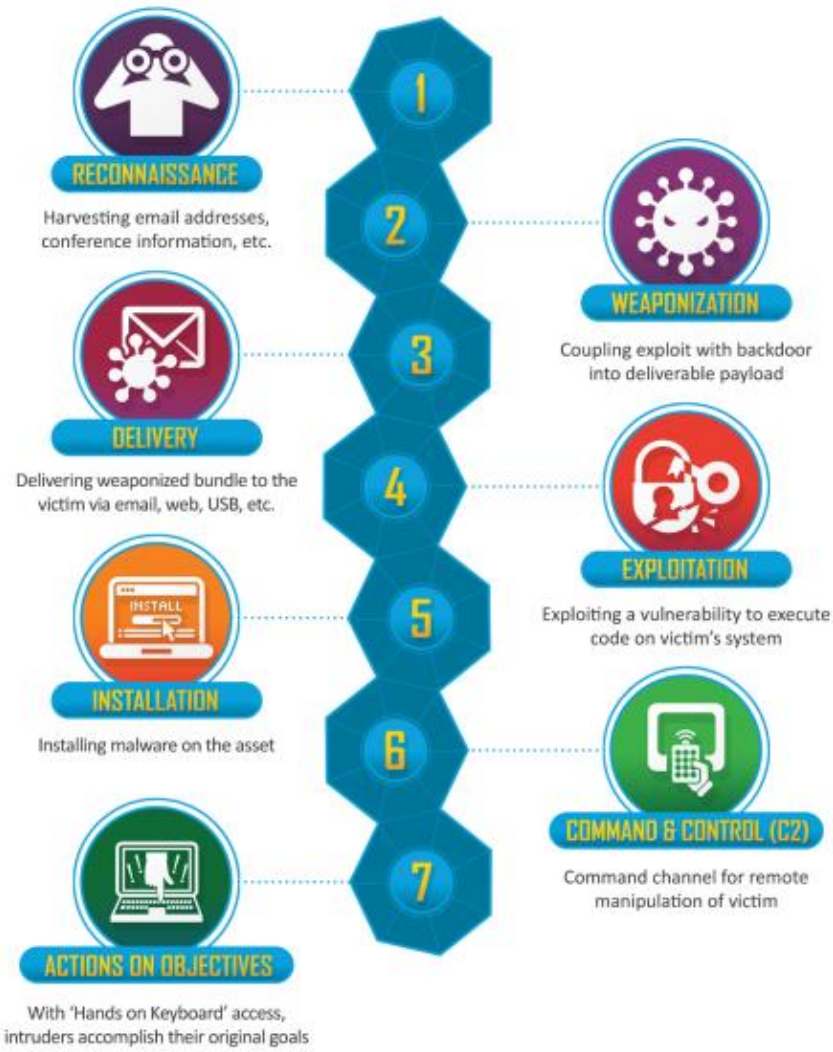

Рисунок 1.1 – Візуальне представлення Loockheed KillChain

Звичайно, він був занадто загальним, тож killchain, що використовується у Mitreє більш деталізованим, він має 14 етапів. Ще більше ідею деталізації Killchain розвинув Paul Pols описавши UnifiedCyberKillChain, що складається із 18 етапів.

У Mitre етпапи Killchain називаються тактиками, і сприймаются як локальні цілі (напр. «провести розвідку», «уникнути виявлення»). Під кожною тактикою розташовані способи її реалізації, техніки. Таким чином ця класифікація набуває вигляду матриці. Таких матриць виділяють кілька. Основна – Enterprise, об'єднує у собі всю наявну інформацію. Ця матриця зображена на рисунку 1.2:

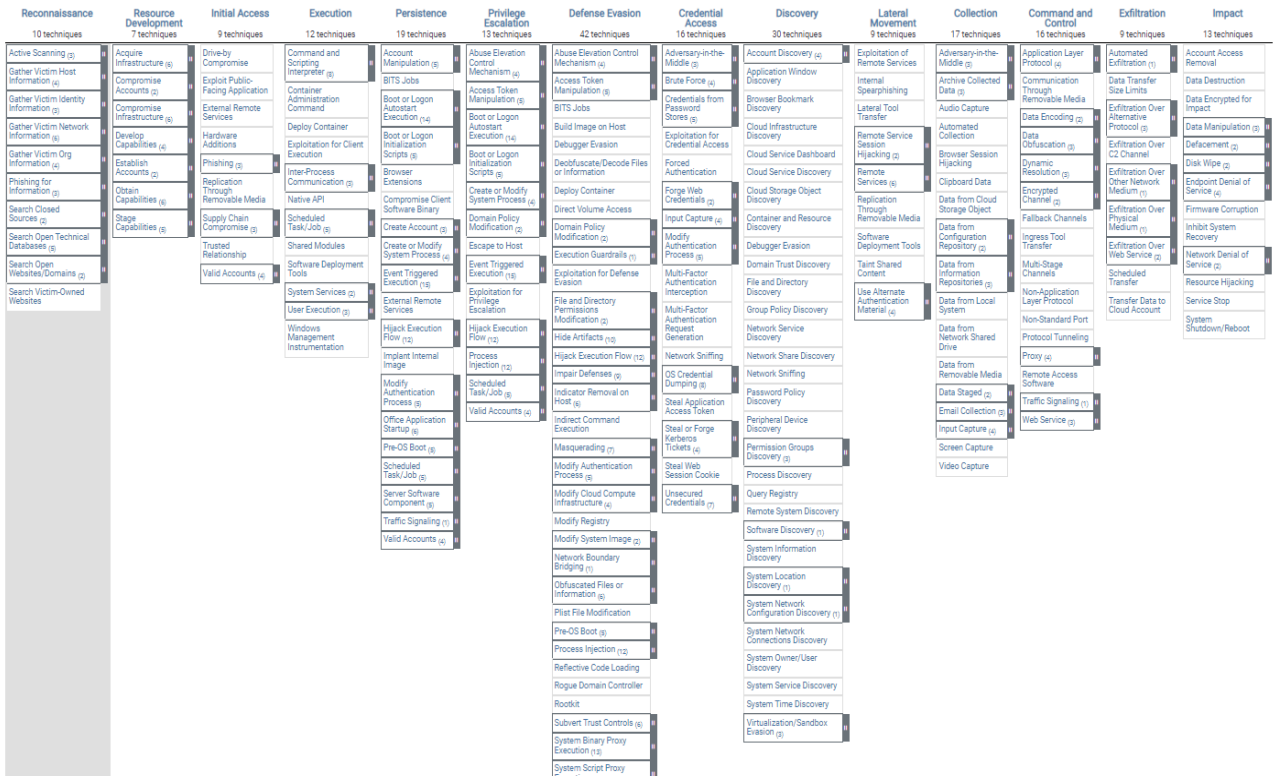

Рисунок 1.2 – матриця Mitre ATT&CK Enterprise

Окрему назву має її частина PRE-ATT&CK, що фокусується на перших двох етапах Killchain, підготовці до атаки. Є матриці для різних операційних систем: Linux, Windows, macOS, для мобільних систем IOS та Android, для атак на хмарні сервіси. Проте, всі вони містяться у Enterprise, тож є просто зручними фільтрами. Щодо технік, у матриці наявний детальний опис кожної з них та поряд з ним рекомендації по уникненню можливості таких дій та рекомендації для їх виявлення у системі. Також там містяться реальні приклади виконання технік з хакерськими угруповуваннями, що їх здійснювали. Інформація про угруповання, включаючи техніки, що вони використовують окремим розділом міститься на офіційному сайті Mitre. Існує популярний інструмент, Mitre AKK&CK Navigator [6], що значно полегшує роботу з матрицею. Це веб-додаток, що пропонує зручну візуалізацію. Наприклад, для порівняння дій кількох угруповань, можна різними кольорами виділити техніки, що використовує лише одна група та іншим ті техніки, що використовують обидві. Таким чином можна візуально знаходити найбільш популярні техніки. Navigator також дозволяє виділяти техніки за платформою призначення (Windows, Linux).

Революційність Mitre AKK&CK полягає у тому, що вона відповідає відразу на два питання: “Що хоче зробити зловмисник?” та “Як він цього досягає?”. Перевагою перед більш традиційними класифікаціями також є зручна візуалізація, що значно полегшує роботу. Проте, головним недоліком Mitre є її неоднозначність. Одна й та сама атака через матрицю Mitre може бути описана кількома різними способами.

1.3 Використання CAPEC для опису атак угрупування Armageddon

У відкритому звіті СБУ по діяльності угруппуваннння Armagedon [15], міститься звіт, що описує усі кібератаки, які використоввують ці зловмисники у термінах технік матриці Mitre. На цьому прикладі продемонструємо недоліки та переваги таких класифікацій як MitreATT&CK та CAPEC.

Таблиця 1.1 – Описання атак технік Mitre, що використовує Armageddon за допомогою шаблонів атак CAPEC та відповідних їм технік Mitre.

Тактика	Техніка MITRE	Шаблон CAPEC	Назва у MITRE	Назва у CAPEC
Initial Access	T1566.001	Capec-163	Spearphishing Attachment	Spear Phishing
	T1566.002		Spearphishing Link	
Execution	T1059.001	Capec-251	PowerShell	Local Code Inclusion
	T1059.005		Visual Basic	
	T1053.005	CAPEC-557	Scheduled Task	Schedule Software To Run
	T1047	CAPEC-478	WMI	Modification of Windows Service

Тактика	Техніка MITRE	Шаблон CAPEC	Назва у MITRE	Назва у CAPEC
				Configuration
	T1059	Capec-248	Command and Scripting Interpreter	Command Injection
	T1559.001	CAPEC-586	Inter-Process Communication: Component Object Model	Object Injection
	T1106	Capec-113	Native API	Interface Manipulation
	T1204.001	CAPEC-416	User Execution: Malicious Link	Manipulate Human Behavior
	T1204.002	CAPEC-17	User Execution: Malicious Image	Using Malicious Files
Persistence	T1547.001	Capec-564	Boot or Logon Autostart Execution	Run Software at Logon
	T1137.001	CAPEC-160	Office Application Startup: Office Template Macros	Exploit Script-Based APIs
Defense Evasion	T1027	CAPEC-267	Obfuscated Files or Information	Leverage Alternate Encoding
	T1140	CAPEC-188	Deobfuscate/Decode Files or Information	Reverse Engineering
	T1070	Capec-93	Indicator Removal on Host	Log Injection-Tampering-

Тактика	Техніка MITRE	Шаблон CAPEC	Назва у MITRE	Назва у CAPEC
				Forging
	T1112	Capec-203	Modify Registry	Manipulate Registry Information
	T1036	CAPEC-177	Masquerading	Create files with the same name as files protected with a higher classification
	T1221	CAPEC-17	Template Injection	Using Malicious Files
	T1497.002	CAPEC-480	Virtualization/Sandbox Evasion: User Activity Based Checks	Escaping Virtualization
	T1218.011	CAPEC-640	System Binary Proxy Execution: Rundll32	Inclusion of Code in Existing Process
Credential Access	T1003	CAPEC-566	OS Credential Dumping	Dump Password Hashes
Discovery	T1082	CAPEC-54	System Information Discovery	Query System for Information
	T1120	CAPEC-646	Peripheral Device	Peripheral

Тактика	Техніка MITRE	Шаблон CAPEC	Назва у MITRE	Назва у CAPEC
			Discovery	Footprinting
	T1033	CAPEC-577	System Owner/User Discovery	Owner Footprinting
Lateral Movement	T1091	CAPEC-457	Replication Through Removable Media	USB Memory Attacks
	T1534	Capec-163	Internal Spearphishing	Spear Phishing
	T1025	CAPEC-456	Data from Removable Media	Infected Memory
	T1113	CAPEC-648	Screen Capture	Collect Data from Screen Capture
	T1119	CAPEC-150	Automated Collection	Collect Data from Common Resource Locations
Command and Control	T1105	CAPEC-185	Ingress Tool Transfer	Malicious Software Download
	T1219	CAPEC-253	Remote Access Software	Remote Code Inclusion
Exfiltration	T1041	CAPEC-161	Exfiltration Over C2 Channel	Infrastructure Manipulation

За результатами практичного застосування класифікації за техніками MITRE до атак, які здійснювались хакерським угрупуванням Армагеддон, отримано висновок про існування неоднозначності класифікації окремих видів атак.

Причин такої неоднозначності дві. По-перше, це некоректна побудова структурної ієрархії технік. Наприклад, Masquerading (T1036) – підвид Obfuscated Files or Information (T1027), але виділяються як окрема техніка; а обидві техніки є підвидом стеганографічних технік, а така гілка класифікації технік взагалі відсутня. По-друге, це ситуація, коли один об’єкт (техніка атаки) може бути віднесений до різних гілок ієрархії. Наприклад, техніка крадіжки облікових записів користувача Credential Access (TA0006) є окремим підвидом техніки System Information Discovery (T1082), але вони знаходяться навіть в різних підкласах, хоча для атаки використовується одна і та сама утиліта mimikatz for Windows.

Ще один цікавий висновок – використання звичайних утиліт-інструментів адміністратора, яке НЕ Є ТЕХНІКАМИ атак, в якості техніки атаки. Прикладом може бути використання mimikatz for Windows (розповсюджений інструмент для перехоплення відкритих сесій в ОС Windows для отримання даних автентифікації користувачів, які входять в систему).

1.4 Використання принципів killchain для запобігання кібертакам на прикладі протидії техніці T113

Часто, для обходу блокування скомпрометованих IP адрес використовується метод підміни IP адреси за допомогою proxy-сервера або VPN. Був обраний саме цей метод, через те, що він став особливо актуальним після початку агресії рф, адже хакерські угруповування країни-агрессора постійно використовують його для атаки на державні ресурси, до того ж він дуже яскраво демонструє переваги та недоліки killchain. При описі за допомогою Killchain знову бачимо основну проблему сучасних класифікацій – неоднозначність. Підміна ip-адреси може бути віднесена одразу до кількох етапів killchain у MitreATT&CK–Initialaccess та DefenceEvasion.

Проте, основною перевагою Killchain є те, що заблкувавши зловмисника на одному з етапів ми робимо неможливою(або значно ускладнюємо) усі його подальші дії. Дійсно, якщо ефективно захиститися від T1190 можна уникнути подальшої атаки. Найефективнішим захистом в данному випадку буде створення сервісу, що визначатиме, належить ipproxy-сервер. Визначити це можна кількома способами – або знайти спільні сигнатури для серверів того чи іншого vpn-сервісу (відкриті порти або сертифікати) або створивши базу данних з ip-адрес найпопулярніших сервісів. Для реалізації сервісу було обрано другий варіант.

Спочатку було складено список із найпопулярніших VPN-сервісів, до нього увійшли зокрема NORD, ExpressVPN, TurboVPN. Для створення бази ip-адрес цих сервісів на мові Python 3.9 розроблено скрипт, що автоматично збирає ip-адреси,поміщаючи їх до бази данних SQLLite, та інструмент для роботи з базою.

Скріншоти з кодом вищезгаданих інструментів зображено на рисунку 1.2.1 та рисунку 1.2.2. API визначає, чи є ip справжнім, або ж належить до VPN-сервіса.

Формат запиту: https://31.131.25.20/checkip/v1/?ip=193.118.53.94. Таким чином, блокуючи доступ до ресурсу для адресvpn-сервісів ми фактично розриваємо killchain на этапі Initial Accesss.

```
import os
import time
from requests import get
with open("Ghost_VPNcOuntries.txt.txt", 'r') as countries:
    for line in countries:
        os.system("sudo cyberghostvpn --traffic --country-code" + line.replace("\n", "") + "--connect ")
        ip = get('https://api.ipify.org').content.decode('utf8')
        Nordip = open("GhostVPN_IP.txt.txt", 'a+')
        if ip in Nordip:
            time.sleep(120)
            os.system("sudo nordvpn disconnect")
            continue
        else: # если нет - записали в файл
            print('My public IP address is: {}'.format(ip))
            time.sleep(120)
            os.system("sudo nordvpn disconnect")
```

Рисунок 1.2.1 – Утиліта для збирання IP адрес VPN. Варіант для NORD

```
def add_data(service, file):
        data = open("{}".format(file), 'r')
        for line in data:
            line = line.replace("\n", "")
            response = requests.get(f'https://ipapi.co/{line}/json/').json()
            country = response.get("country_name")
            click.execute("INSERT INTO ip_data VALUES (?,?,?)", (line, country , service))
            base.commit()
def add_data_manually(service):
    while True:
        ip = input("Enter ip you want to add:")
        if ip == "STOP":
            break
        else:
            response = requests.get(f'https://ipapi.co/{ip}/json/').json()
            country = response.get("country_name")
            click.execute("INSERT INTO ip_data VALUES (?,?,?)", (ip, country, service))
            base.commit()
def show_table():
    table = click.execute("SELECT * FROM ip_data")
    for value in table:
        print(value)
def dell_table():
    click.execute(f"DELETE FROM ip_data WHERE service = '{args.service}'")
    base.commit()
```

Рисунок 1.2.2 – Фрагмент коду утиліти для роботою з базою даних

На наступупному рисунку 1.2.3 приклад спрацюванння APIна підключення з VPN:

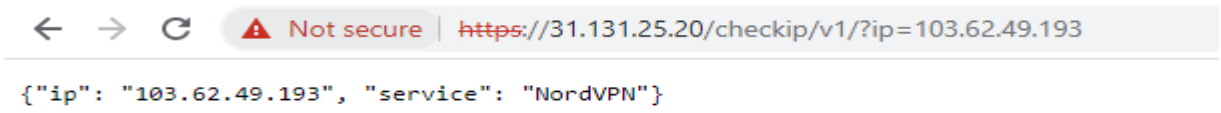

Рисунок 1.2.3 – реакція API на підключеняння з ip, щоналежить VPN-сервісу

Висновки до розділу 1

У першому розділі були розглянуті результати аналізу існуючих методів вивчення поверні атак для різних задач інформаційної та кібербезпеки, проаналізовані недоліки існуючих баз даних кібератак та класифікацій атак.

Розглянутий підхід орієнтований на проектування або оцінку захищеності конкретних інформаційних систем, а не на виявлення атак.

Тому для цілей виявлення атак будемо застосовувати «обернену» постановку задачі – будемо вивчати структуру існуючої множини відомих кібератак, з метою вивчення структурних закономірностей таких атак, виділення з них елементарних (базових) атак. Після виділення базових атак та дослідження чи утворює множина атак певну алгебраїчну структуру відносно операцій над цими базовими атаками, ми вирішуємо задачу про покриття множини атак мінімальною кількистю підмножин атак, складених із цих базових атак. Це і є рішенням задачі мінімізації кількості правил виявлення атак для системи виявлення вторгнень.

За результатами практичного застосування класифікації за техніками MITRE до атак, які здійснювались хакерским угрупуванням Армагеддон, отримано висновок про існування неоднозначності класифікації окремих видів атак.

Причин таких неоднозначності дві. По-перше, це некоректна побудова структурної ієрархії технік. Наприклад, Masquerading (T1036) – підвид Obfuscated Files or Information (T1027), але виділяються як окрема техніка; а обидві техніки є підвидом стеганографічних технік, а така гілка класифікації

технік взагалі відсутня. По-друге, це ситуація, коли один об’єкт (техніка атаки) може бути віднесений до різних гілок ієрархії. Наприклад, техніка крадіжки облікових записів користувача Credential Access (TA0006) є окремим підвидом техніки System Information Discovery (T1082), але вони знаходяться навіть в різних підкласах, хоча для атаки використовується одна і та сама утиліта mimikatz for Windows.

Ще один цікавий висновок – використання звичайних утиліт-інструментів адміністратора, яке НЕ Є ТЕХНІКАМИ атак, в якості техніки атаки. Прикладом може бути використання mimikatz for Windows (розповсюджений інструмент для перехоплення відкритих сесій в ОС Windows для отримання даних автентифікації користувачів, які входять в систему).

2 СТРУКТУРНІ МЕТОДИ ВИВЧЕННЯ ПОВЕРХНІ АТАК

2.1 Задача розробки формальної мови кібератак та виявлення атак «нульового дня»

Після аналізу та практичного застосування сучасних класифкацій кібератак, вивчення існуючих баз даних кібератак, очевидним стає те, що всі вони мають ті чи інші недоліки, деякі з них занадто загальні, деякі недостаньо чіткі, неоднозначні. Кожна має свої переваги, але жодну з них не можна назвати «формальною мовою для опису кібератак».

Вирішення задачі розробки «формальної мови кібератак» тісно пов’язано з іншою важливою задачею – захистом від 0-day атак, адже чіткий математичний апарат для опису кібератак, дозволив би описувати не тільки існуючи, а й гіпотетично можливі атаки.

Визначимося із деякими термінами, які іноді вживаються у дещо різних «модифікаціях». Традиційне визначення, приведене вище ми будемо називати «послабленим».

Визначення 1. Вразливість нульового дня (в «послабленому варіанті») – програмна вразливість, виявлена зловмисниками перед тим, як про неї дізналися виробники програми, а для вразливостей нульового дня ще не випущені патчі, відповідно експлоїт нульового дня – це метод, який зловмисники використовують для атаки на системи з не виявленими раніше вразливістю, а атака нульового дня – це використання експлойту нульового дня для затаки на інформаційну систему, в якій є вразливість. Зауважимо, що походження терміну пов'язане з тією обставиною, що вразливість або атака стає публічно відома до моменту випуску виробником ПЗ виправлень помилки (тобто потенційно вразливість може експлуатуватися на копіях, що працюють, програми без можливості захиститися від неї).

Визначення 2. Вразливість нульового дня (в «суворому варіанті») –це взагалі на даний час **невідома** нікому окрім зловмисника вразливість, яка експлуатується зловмисниками у атаках.

Різниця між визначеннями очевидна: в першому визначені деяка інформація про вразливість вже відома (як наприклад, на теперішній час з Apache log4j), в другому ми не маємо **жодної інформації про вразливість до початку атаки.**

При аналізі можливості застосування цих методів до атак «нульового дня» потрібно мати на увазі, при застосуванні сигнатурних методів аналізу часто замість розширених сигнатур атак використовуються так звані «індикатори компрометації» (англ. – «Indicator of Compartmentation, IoC). IOC — це об'єкт, що спостерігається в мережі або операційній системі, який з великою ймовірністю вказує на компрометацію пристрою. До таких об'єктів можуть бути віднесені виявлені сигнатури вірусів, хеш-суми шкідливих файлів, адреси командних серверів ботнетів. Індикатор компрометації використовується для раннього виявлення спроб проникнення в комп'ютерні системи та первинної оцінки можливої загрози. Часто саме IoC дозволяє виявляти атаки нульового дня, оскільки навіть при відсутності оновлень та детальних правил виявлення, можна виявляти характерні фрагменти текстових повідомлень, коду, тощо. Саме так відбулось із «найсвіжішою» 0-day вразливістю Apache log4j, які виявляються за рахунок фільтрування усіх запитів, що містять “${jndi:ldap://…..“ та їх подальшому ручному аналізі. Зрозуміло, що при цьому спостерігається надвеликий відсоток хибних спрацьовувань (false/positive) і для великих інформаційних систем це є практично неможливим. Більш того якщо розглядати «суворе» визначення 0-day атаки, то сигнатурні методи та більшість методів, заснованих на правилах взагалі не виявляють такі атаки.

Отже однією з найбільших проблем кіберзахисту є відсутність реальних дієвих механізмів попередження атак, якщо не визначено певні сигнатури таких атак або правила виявлення таких атак. А отже, практично єдиним методом виявлення таких атак є аналіз аномальної поведінки процесів обробки інформації

в системи. Тут слід розглядати аномалію, як незвичну поведінку інформаційної системи, що виходить за межі очікуваної, а не «нормальної», оскільки нормальність поведінки може бути визначена в межах суб’єктивного підходу для різних систем. А відхилення від очікуваної поведінки базуватиметься в першу чергу на прогнозуванні. Для розробки таких механізмів необхідно забезпечити безперервне надходження вхідних даних про поведінку користувача, що і є основою для визначення аномальності.

Приведемо результати аналізу відомих методи виявлення 0-day атак на основі аномалій [16].

Суть методів виявлення атак, заснованих на аномаліях, полягає в тому, щоб записати шаблон нормальної активності мережі та реагувати на відхилення від цього шаблону. Можливий варіант реалізації у вигляді бази сигнатур "нормальних" пакетів у мережі і статистичної системи виявлення відхилень, коли аналізатор шукає якісь рідкісні дії, або активності. Події в системі досліджуються статистично, щоб знайти ті з них, прояви яких виглядають аномальними. Чому методи виявлення аномалій як вихідні дані використовують саме часові ряди – тому що це дозволяє максимально бути вільним від природи, яка викликає аномалію. Дійсно про аномальну поведінку можна говорити тоді, коли є зміни певного показника або показників з плином часу. Тоді значення показника в дискретні моменти часу і можуть розглядатися без прив’язки до його природи і є універсальними.

Властивістю часового ряду є:

прив'язка кожного виміру (шаблону, дискретного значення) до часу його виникнення,

рівна відстань за часом між вимірами,

можливість з даних попереднього періоду відновити поведінку процесу у поточному та наступних періодах.

Аномальна поведінка може бути визначена як перевищення порогу відхилення від характеристик часового ряду. Такими характеристиками є:

період – часовий відрізок постійної для ряду довжини, після завершення якого значення ряду повторюються, тобто якщо a_i – значення елементу ряду в i момент часу, l – період ряду, то $a_i = a_{i+l}$;

цикл змін - характерні зміни ряду, пов'язані з зовнішніми причинами, які порушують періодичну поведінку ряду (тобто на певних відрізках ряду для кількох значень виконується нерівність $a_i \neq a_{i+l}$)

тренд - тенденція до довгострокового збільшення чи зменшення значень ряду.

Зміни цих характеристик породжують різні види аномалій, серед яких виділяють наступні типи аномалій [16]:

точкові аномалії, коли спостерігається відхилення у поведінці в окремих точках;

групові аномалії, у яких аномально поводяться група точок, кожна з яких окремо аномальною не є;

аномалії контексту, коли аномалія пов'язана із зовнішніми даними, які впливають на значення ряду (наприклад, негативна температура на вулиці влітку).

Точкові аномалії розпізнаються найпростіше – це окремі точки, у яких поведінка процесу різко відрізняється з інших точок. Наприклад, можна спостерігати різке відхилення значень параметра окремій точці. Такі значення називаються "викиди", вони сильно впливають на статистичні показники процесу і їх легко виявити, встановивши порогові значення для величини, що спостерігається. Складніше виявити аномалію у ситуації, як у кожній точці процес поводиться «нормально», але у сукупності значення кількох точках поводяться «аномально». До такої аномальної поведінки можна віднести, наприклад, зміну форми сигналу, зміну статистичних показників (середнє значення, мода, медіана, дисперсія), поява взаємної кореляції між двома параметрами, невеликі або короткострокові аномальні зміни амплітуди тощо. Проблемою є розпізнавання аномалій, які не можна виявити звичайними статистичними методами за рахунок нечіткого визначення аномалій, відсутність

розмітки, неочевидна кореляція. До цих пір алгоритми Self-Organizing Tree Algorithm (SOTA) з пошуку аномалій в часових рядах мають високий рівень False/Positive. Вручну можна виявити лише невелику кількість аномалій, переважно точкових, за наявності хорошої візуалізації даних. Групові аномалії складніше виявити вручну, особливо якщо йдеться про велику кількість даних та аналіз інформації про кілька пристроїв. Також складним виявлення є випадок «аномалії в часі», коли нормальний за параметрами сигнал з'являється в «неправильний» час. Тому при пошуку аномалій застосовуються різні статистичні методи [16], при цьому перевагу надається статистикам, які «працюють» на «коротких» послідовностях вхідних даних.

Великою проблемою в пошуку аномалій на реальних даних є те, що дані, як правило, не розмічені, тому апріорі точно не визначено, що таке аномалія, немає правил для пошуку. У таких ситуаціях необхідно застосовувати методи навчання без вчителя (unsupervised learning), при цьому моделі самостійно визначають взаємозв'язки та характерні закони даних.

Всі існуючі методи виявлення аномалій можна поділити на такі типи [16]:

proximity-based: виявлення аномалії на основі інформації про близькість параметрів або послідовність параметрів фіксованої довжини, підходить для виявлення точкових аномалій та викидів, але не дозволить виявити зміни у формі гістограм (графіків змін даних), характерна візуалізація точкової аномалії наведена на рисунку 2.1, а групової аномалії (наприклад, зміни частоти) – на рисунку 2.2:

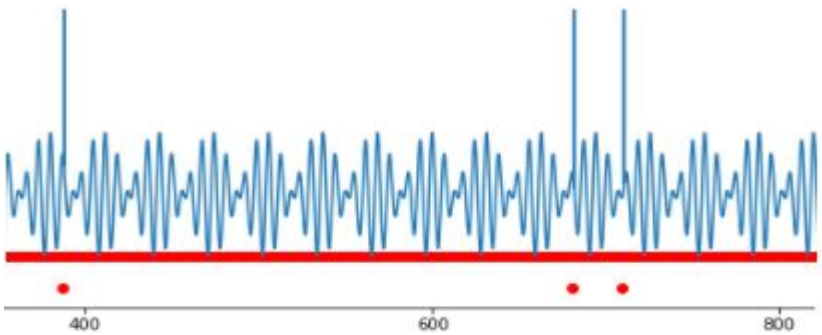

Рисунок 2.1 - Характерний вид точкових аномалій

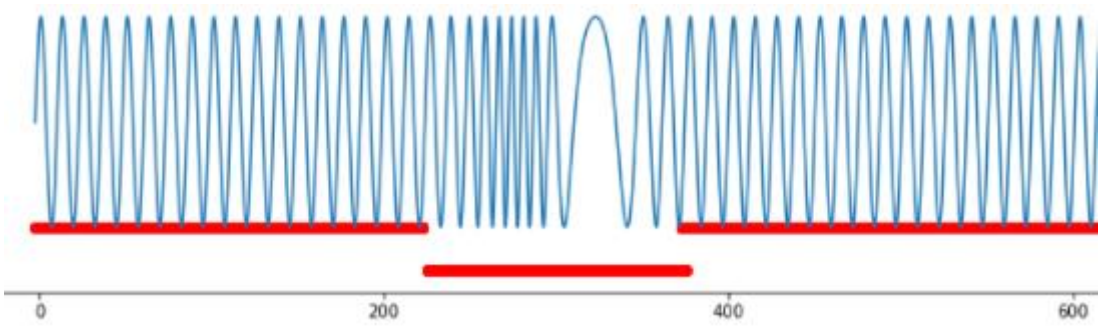

Рисунок 2.2 - Характерний вид групової аномалії

prediction-based: побудова прогнозної моделі та порівняння прогнозу та фактичної величини, найкраще застосовується до часових рядів з вираженими періодами та циклами змін (рисунок 2.3);

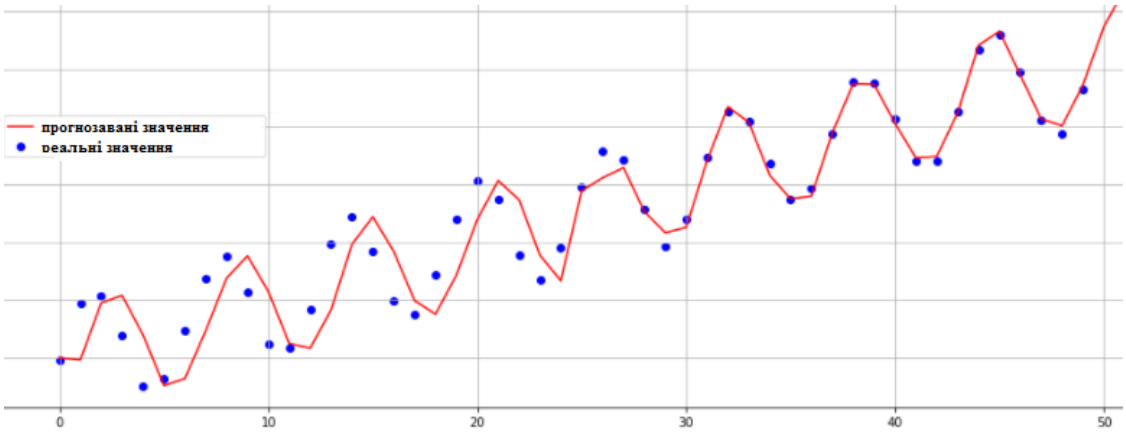

Рисунок 2.3 - Ідея методів виявлення аномалій на основі прогнозування

reconstruction-based: методи, засновані на реконструкції фрагментів даних, використовують відновлення (реконструкцію) фрагмента даних, тому може виявляти як точкові аномалії, і групові аномалії, зокрема зміни у вигляді графіку, який відображає зміну часового ряду.

Тривіальний приклад реалізації методів першого типу – контроль перевищення заданого порога значень. У prediction-based методах основне завдання – побудувати якісну модель процесу, щоб змоделювати сигнал та порівняти отримані змодельовані значення з вихідними (істинними) (рис. 2.3). Якщо передбачений і істинний сигнал близькі, то поведінка вважається «нормальною», а якщо значення моделі сильно відрізняються від істинних, то

поведінка системи на цій ділянці оголошується аномальною. Найбільш поширені методи для моделювання часових рядів – SARIMA та рекурентні нейронні мережі. Оригінальний підхід використовується в reconstruction-based моделях – спочатку модель навчають кодувати та декодувати сигнали з наявної вибірки, при цьому закодований сигнал має набагато меншу розмірність, ніж вихідний, тому моделі доводиться вчитися «стискати» інформацію. Після навчання моделі дають на вхід сигнали, що є відрізками часового ряду, який досліджується, і якщо кодування-декодування проходить успішно, то поведінка процесу вважається «нормальним», інакше поведінка оголошується аномальною. Одним з нещодавно розроблених reconstruction-based методів, що показують хороші результати у виявленні аномалій, є TadGAN [16].

В практичних системах всі методи виявлення аномалій часто будують за різною функціональністю аномалій, а саме:

Альфа-аномалія. Незвичайно високий рівень трафіку типу точка-точка Викид у поданні трафіку байти/с, пакети/с по одному домінуючому потоку джерело-призначення. Невелика тривалість (до 10 хвилин)

DoS-, DDoS-атака. Розподілена атака типу відмова у обслуговуванні на одну жертву Викид у поданні трафіку пакети/с, потоки/с, від безлічі джерел до одного адресою призначення. Перевантаження Незвичайно високий попит на один мережевий ресурс чи сервіс Стрибок у трафіку по потоках/с до одного домінуючій IP-адресі та домінуючій порту. Зазвичай короткочасна аномалія.

Сканування мережі/портів. Сканування мережі за певними відкритим портам або сканування одного хоста по всіх портах з метою пошуку вразливостей. Стрибок у трафіку по потоках/с, з декількома пакетами в потоках від однієї домінуючої IPадреси.

Діяльність хробаків Шкідлива програма, яка самостійно поширюється по мережі та використовує вразливість ОС Викид у трафіку без домінуючої адреси

призначення, але завжди з одним або декількома домінуючими портами призначення

Точка-мультиточка Розповсюдження контенту від одного сервера багатьом користувачам. Викид у пакетах, байтах від домінуючого джерела до кількох призначень, все до одному добре відомому порту

Відключення Мережні неполадки, що викликають падіння у трафіку між однією парою джерело-призначення Падіння трафіку по пакетах, потоках та байтах зазвичай до нуля. Може бути довготривалим і включати всі потоки джерело-призначення від або до одного маршрутизатора

Перемикання потоку Незвичайне перемикання потоків трафіку з одного вхідного маршрутизатора на інший Падіння в байтах чи пакетах в одному потоці трафіку та викид в іншому, може застосовуватися до кількох потоків трафіку.

Повернемось до задачі створення формальної мови кібератак.

Ключовим для створення такої мови є те, що кожна кібератака являє собою певний набір дій. Найменшими одиницями для описання цих дій сьогодні є техніки матриці MitreATT&CK та шаблони атак Capec. Проте, вивчаючи ці класифікації, можна помітити, що кожна складна техніка фактично є надбудовою над більш простими діями, або комбінацією кількох технік. Це означає, що теоретично кожну техніку можна розкласти на складові і виділити у кожній з них “ядро”. Надалі будемо називати його “Базовою аткакою”. В основі подальших ідей щодо базових атак лежить припущення:

“Захист від “базових атак” забезпечує захист від великої кількості складних кібератак, що є їхніми надбудовами або комбінаціями.”

У випадку, якщо твердження вірне, це дозволить значно зменшити кількість сигнатур у системах захисту кінцевих точок (Endpoint Detection & Responce(EDR)) та системах виявлення\попередження вторгнень(Intrusion Detection System\Intrusion Prevention System(IDS/IPS)), адже замість захисту від кожної атаки, необхідно буде зробити сигнатури лише для базових атак. Навіть якщо твердження виявиться невірним, виявлення “базових атак” дозволить створити більш детальну класифікацію технік та визначити якою алгебраїчною структурою є множина кібератак, як максимум у подальшому це дозволить створити алгебру кібератак.

Отже, спочатку треба визначити за якими критеріями вважати атаку “базовою”. Інтуїтивно зрозуміло, що це має бути найбільш проста дія, що виконує зловмисник, але формальні критерії для виділення базових атак – це найбільш складне та приципове питання

Найочевидніший спосіб, вважати базовими атаками найбільш росповсюджені за стистикою. Проблемою у цьому випадку є власне збір статистики, який не є тривіальною задачею. Ідеальною була б статистика за конкретними CVE, скільки разів вони були використані, такої статистики не знайдено. Проте, за матрицею Mitre ATT&CK можливо статиску найчастіше використовуваних технік зібратати опосередковано, за допомогою MitreATT&CKNavigator, що дозволяє виділяє порівнювати набір технік, що використовується для різних операційних систем, або різними хакерськими угруповуваннями. Таким чином можна виділити техніки, що використовуються більшістю акторів для всіх операційних систем та вважати їх базовими атаками.

Ключова ідея другого методу виділення цих атак: „Базові атаки – це звичайні дії користовуча в системі”. Наприклад, звичайне підключення до сайту, не має на меті нічого протиправного, це елементарна дія, проте багато підключень одночасно – DDOS атака. Інший приклад, відправлення TCP-пакету, саме по собі абсолютно регулярна дія, проте якщо ми відправляємо пакет з певним флагом, ця дія перетворюється на сканування. Очевидною перевагаю цього методу, є можливість визначити над множиною атак ряд операцій, як у першому прикладі отримано DDOS атаку – конкатенацією елементарних дій.

Очевидною проблемою виділення базових атак є «рівень деталізації», при якому атаку будемо вважати базовою. При цьому в різних практичних ситуаціях виділення базових атак може бути різним. Наприклад, для задачі нашого дослідження спочатку доцільно зупинитись на самостійних техніках матриці MITRE – наприклад завантаження файлів певного типу (наприклад, docx, exe, pdf) тощо.

Після виділення множини базових атак та введення операції – “ієрархічної суперпозиції” – об’єднання їх в killchaine, можна вирішувати задачі оптимізації множини всіх можливих атак за допомогою вирішення задачі про покриття.

2.2 Задача про покриття множини атак

Задача про покриття множини за складністю відноситься до класу NP.

Наведемо її класичне формулювання [17]:

Задані множина $X, |X| = m$ та родина його підмножин $\{S_1, \dots, S_n\}, S_j \subseteq X$, таких що $X = \bigcup_{j=1}^{n} S_j$. Знайти мінімальну за потужністю множину підмножин, яке покриває X, тобто $|K| \to min, K \subseteq \{1,2, \dots, n\}, X = \bigcup_{k \in K} S_k$.

Задача в такому (потрібному для вирішення задачі про мінімізацію поверхні атак) формулюванні належить навіть до NP-повних задач, тому на існування поліноміального алгоритму її вирішення є мало надії. Для її вирішення, як правило, застосовуються жадібні або генетичні алгоритми. В свою чергу для роботи цих алгоритмів ми повинні виділити найбільш елементарні атаки, які частіше всього зустрічаються в матриці MITRE.

2.3 Аналіз ефективності рішення задачі про покриття

Одним з ефективних практичних підходів до вирішення таких NP-повних задач є використання наближених алгоритмів жадібного або генетичного типів. Наприклад жадібний алгоритм вирішення задачі покриття передбачає використання простої жадібної евристики: «Обираємо на кожній ітерації чергову підмножину, яка містить максимальну кількість елементів, не покритих на попередніх ітераціях». Такий Set Cover Resolve алгоритм гарантує точність рішення $1 + lnm$, де m – потужність множини, для якої визначається покриття. Тут точність такого наближеного рішення означає у скільки разів таке рішення буде відхилятись від мінімального.

Доведено [17-19], що мультиплікативна похибка такого жадібного алгоритму не перевищує $1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{m} = lnm$ та навіть $lnm - lnlnm + O(1)$.

Стандартний жадібний алгоритм рішення задачі про покриття виглядає достатньо просто:

Алгоритм SCR.

На кожній ітерації обрати максимально непокриту множину.

Для нашого випадку це означає обрати максимальну кількість атак, яка містить обрану базову атаку, а це означає, що гістограма частот базових атак, розташована «по зменшенню абсолютних значень» і є рішенням задачі покриття множини атак. Зауважимо, що при «розфарбуванні атак» залишаються певні пробіли, як при зборі статистики за хакерськими угрупуваннями, так і за зловмисним кодом. Перше пояснюється, що не всі атаки можна віднести до конкретного хакерського угрупування, друге – не всі елементарні атаки реалізовані у зловмисному коді.

Зауважимо, що часто на практиці постановка задачі про покриття для поверхні атак є не зовсім класичною, оскільки, як правило, нам достеменно не відома множина X. Дійсно, в теорії проектування захищених інформаційних систем панує декілька різних підходів [2]: проектування системи захисту інформації «з нуля», паралельно до проектуванню самої інформаційної системи (або конструктивний підхід) та добудови системи захисту до вже існуючої інформаційної системи (або підхід «доданого захисту»). В першому підході потужність множини X визначають як максимальну кількість всіх потенційних загроз системі, які тільки можуть існувати завдяки вразливості всіх її програмних та апаратних підсистем. Такий підхід застосовується на практиці не часто та вимагає значних ресурсів. При другому підході ми якраз намагаємось з певних міркувань про супротивника та систему визначити максимальну поверхню атак та обмежити систему захисту протидією нею. Тому «генерація» максимально можливої поверхні атак за допомогою мінімальної множини підмножин елементарних атак є дуже актуальною задачею.

Зважаючи на наведене вище, пропонується застосувати «обернену» постановку задачі – розглянути атаки, які утворюються на найбільш часто вживаних техніках з матриці MITRE та порахувати який відсоток від загальних

атак вони складають – тобто застосувати конструктивний підхід, утворюючи множину X та оцінювати її потужність. Цей підхід дозволить також більш точно вивчити структуру складних APT-атак, що утворюють поверхню атак, зокрема визначити чи утворюють атаки деяку алгебраїчну структуру.

Фактично, коли ми отримуємо гістограму частот елементарних технік (за введеною нами термінологією – базових атак) матриці MITRE ми з певною точністю вирішуємо задачу про покриття. Тому в нашому випадку задачу про покриття доцільно розглядати в «максимізаціонному варіанті» або «задачі K - покриття» [17], а саме:

Задані множина $X, |X| = m$ та родина його підмножин $\{S_1, \dots, S_n\}, S_j \subseteq X$. Обрати такі k підмножин, що потужність їх об'єднання була б максимальною. тобто $|J| = k, J \subseteq \{1,2, \dots, n\}, \bigcup_{j \in J} S_j \to max$.

Така постановка задачі набагато «зручніша» для нас, оскільки потужність множини X як правило невідома. Відомо [19], що жадібний алгоритм SCR, розглянутий нами вище, є поліноміальним $(1 - e^{-1})$ наближеним алгоритмом рішення «задачі K - покриття». Зауважимо, що зазначена точність є прийнятною для практичного застосування.

Отже існує практично ефективний алгоритм рішення задачі побудови близької до оптимальної поверхні атак, для роботи якого нам потрібно визначити множину базових атак (і підмножини атак, які відбудовуються операцією ієрархічної суперпозиції в підмножини атак, які покривають всю (або практично всю) поверхню атак. Для виділення базових атак застосуємо статистичні методи для вивчення потужності підмножин, що містять певні техніки атак.

2.4 Застосування жадібного алгоритму рішення задачі про покриття до множини атак матриці MITRE

Як було зазначно вище, для вирішення задачі про покриття множини всіх атак достатньо виділити підмножини, які містять певну базову атаку, в порядку

зменшення потужності цих підмножин. У випадку застосування алгоритму до бази атак, яка міститься в матриці MITRE ми повинні визначити частоти атак, які містять певні техніки атак – це і буде потужність відповідних підмножин, які складають покриття усієї бази атак. Якщо починати з технік, які зустрічаються найчастіше, ми отримуємо розташування підмножин в порядку зменшення їх потужності, що і буде рішенням задачі. Відповідні потужності (частоти) і будуть похибкою алгоритму – тож нам достатньо зупинитись, як тільки наша похибка буде задовільняти нашим потребам.

У більшості відкритих звітів з діяльності кіберзлочинців або кібератаках за певний період(рік), наведено деякі загальні факти, однак немає чітих детальних статистичних данних, щодо типів кібератак, що здійснювалися. Так, у звіті [20], автори пишуть, що 23% всіх кібератак належать до типу ransomeware, програм-вимагачів, на другому місці Server access attacs (11%), на третьому – buisness email compromise(). Також наводиться статистика за типами ransomeware, відповідно до неї на REvil, Ryuk, LockBit 2.0, припадає більшіть(в відсотковому відношенні %) кібератак типу Ransomeware. У звітах [21] містится інформація про те, що 98% malware доставляется до цілей через email, та більшість цих файлів – doc або doc, другий за розповсюдженістю тип - .exe . В цілому це корисні факти, на які слід звернути увагу при виділення базових атак, проте вони не дуже системні та фрагментарні, а для виділення базових атак за статичними данними необхідно створити чітку ієрархію кібератак за частотою використання. Такі бази данних є у компаній, що надають рішення у сфері кібербезпеки, проте всі вони є закритими. Також вивчивши цю літературу стає зрозумілим, що майже всі сучасні атаки так чи інакше використовують соціальну інженерію. Проте цей тип атак є найскладнішим з точки зору формалізації та безпеки, його вивчення та структуризація є предметом для окремих досліджень.

Виходячи з міркувань, що наведені вище прийнято рішення про збір непрямої статистики частоти використання технік матриці Mitre ATT&CK, за допомогою інструменту Mitre Navigator [22]. Це веб-додаток з відкритим кодом створений Mitre для роботи з цією матрицею. Найголовнішою його перевагою

для збору статистики є те, що він дозволяє виділяти техніки Mitre, що використовуються різними хакерськими угруповуваннями або програмним забезпеченням та надавати їм рейтинг, за допомогою якого можна порахувати частоту. Опишемо трохи детальніше, як проводився збір статистики.

Mitre Navigator при всій своїй зручності має один суттевий недолік у контексті задачі, що вирішується, він дозволяє створити максимум 10 аркушів за раз. Через це алгоритм виглядав наступним чином.

Алгоритм Sieve

1. Створюємо новий аркуш
2. За допомогою функцій Mitre Navigator виділяємо техніки, що використовує перше хакерське угруповування
3. Присвоюємо цим технікам рейтинг 1
4. Повторюємо етапи 1, 2 3, ще для наступних 8 технік та на десятому аркуші знаходимо сумму попередньо присвоєних рейтингів
5. Знаходимо сумму 9 таких сумм
6. Повторюємо етапи 1-5 9 разів
7. Знаходимо сумму 9 сумм етапу 6

Таким чином на останьому аркуші кожній техніці буде присвоєно рейтинг, що відповідає кількості програм/хакерських угруповань, що її використовують. Для кращого сприйнятття, наведена візуалізація:

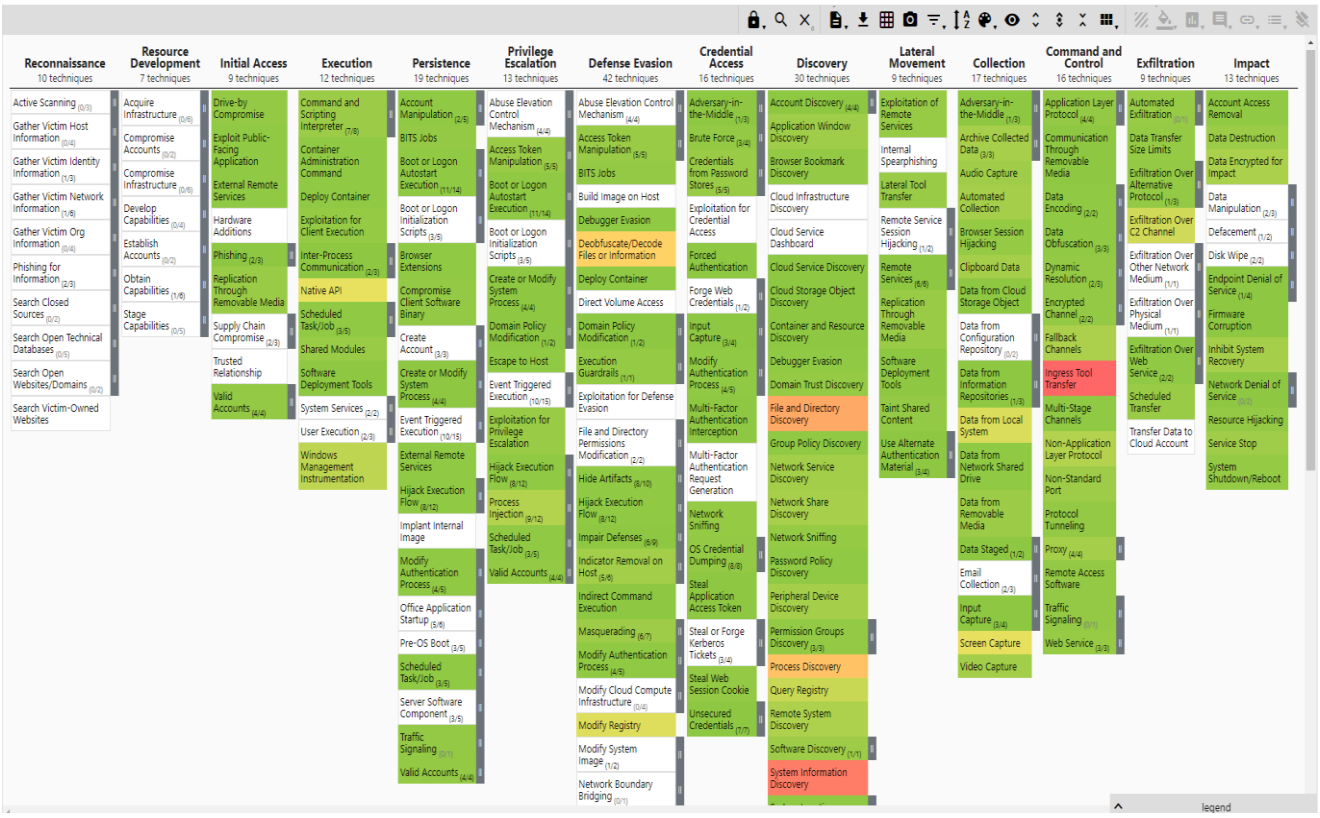

Рисунок 2.4. - Візуалізація частоти використання технік у програмногму забезпеченні у Mitre ATT&CK

Збір ститистики проводився окремо за хакерськими угруповуваннями, що наявні в Mitre та за програмним забезпеченням. Слід звернути увагу, що збір проводився саме за програмним забезпеченням, а не malware оскільки у вибірці присутні як віруси (Emotet, NotPetya, Black Energy) так і звичайне программне забезпечення що може бути використано зловмисниками у своїх цілях, наприклад cmd та mimikatz. Розмір вибиркі склав 130, для хакерських угруповань та 583 для програмного забезпечення.

Для розуміння отриманих даних, приведених в таблицях 2.1 – 2.4, потрібно зауважити, що одна техніка може використовуватись **більше ніж в одній атаці,** а відсоток рахується відносно **всій кількості атак.**

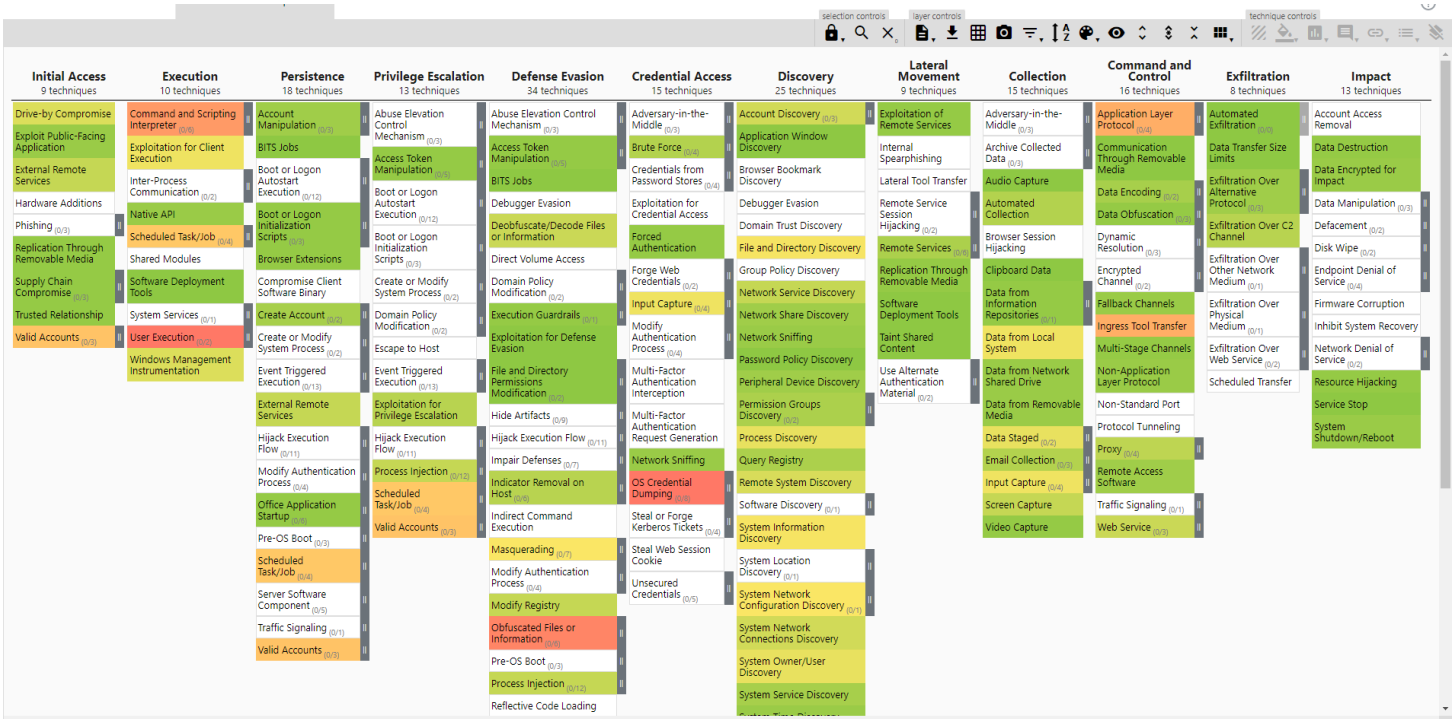

Рисунок 2.5. - Візуалізація частоти використання технік у Mitre ATT&CK хакерськими угрупуваннями

Таблиця 2.1. Частота використання технік у програмному забезпеченні, техніки, що використовувалися > 200 разів

Тактика Mitre	Техніка матриці Mitre ATT&CK	Частота використання техніки(%)
Command and Control	Ingress Tool Transfer	50,9 %
Discovery	System information discovery	45,8%
Defense Evasion	Obfuscated Files or Information	40,7%
Command and Control	Web Protocols	39,9%
Discovery	File and Directory Discovery	37%
Execution	Command and Scripting Interpreter: Windows Command Shell	36,8%

Таблиця 2.2 – Частота використання технік у програмному забезпеченні, техніки, що використовувалися >= 100 разів

Тактика Mitre	Техніка матриці Mitre ATT&CK	Частота використання техніки(%)
Discovery	Process Discovery	32,2%
Defence Evasion	Indicator removal on host:	29,8%

	File Deletion	
Discovery	System Network Configuration Discovery	29,7%
Defence Evasion	Deobfuscate/Decode Files or Information	29%
Persistence, Privilage Escalation	Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder	27,1%
Discovery	System Owner/User Discovery	22,6%
Execution	Native API	20%
Collection	Screen capture	19,6%
Command and Control	Encrypted Channel: Symmetric Cryptography	19,2%
Defence Evasion	Modify Registry	18%
Collection	Data from Local System	17,2%

Таблиця 2.3 – Частота використання технік у програмному забезпеченні, техніки, що використовувалися > 50 разів

Тактика Mitre	Техніка матриці Mitre ATT&CK	Частота використання техніки(%)
Collection, Credential access	Input Capture: Keylogging	17%
Defence Evasion	Masquerading: Match Legitimate Name or Location	14,9%
Exfiltration	Exfiltration Over C2 Channel	14,8%
Persistence, Privilage Escalation	Create or Modify System Process: Windows Service	14,4%
Execution	Command and Scripting Interpreter: PowerShell	14,2%
Execution, Persistence, Privilage Escalation	Scheduled Task/Job: Scheduled Task	13,6%
Discovery	Software Discovery: Security Software Discovery	13%
Discovery	Query Registry	12,7%
Command and Control	Data Encoding: Standard Encoding	12%
Execution	Windows Management Instrumentation	10,8%

Defense Evasion	Obfuscated Files or Information: Software Packing	9,8%
Execution	User Execution: Malicious File	9,4%
Defense Evasion	Impair Defenses: Disable or Modify Tools	9,4%
Collection	Data Staged: Local Data Staging	9,4%
Discovery	System Time Discovery	9,3%
Discovery	System Network Connections Discovery	8,9%
Execution	Command and Scripting Interpreter: Visual Basic	8,7%
Defense Evasion	System Binary Proxy Execution: Rundll32	8,7%

Таблиця 2.4 – Частота використання технік хакерськими угрупованнями, техніки, що використовувалися > 20 разів

Тактика Mitre	Техніка матриці Mitre ATT&CK	Частота використання техніки(%)
Execution, Defence Evasion	Scripting	45,3%
Execution	User Execution	41,8%
Credential Access	OS Credential Dumping	41,8%
Defense Evasion	Obfuscated files or information	39,5%
Execution	Command and Scripting Interpreter: PowerShell	38,4%
Command and Control	Ingress Tool Transfer	32,3%
Command and Control	Application Layer Protocol	32,3%
Initial Access, Persistence, Privilege Escalation	Valid Accounts	29,1%
Execution, Persistence, Privilege Escalation	Scheduled Task/Job	28%

В цілому техніки, що використовує программне забезпечення співпадають з найчастішими техніками хакерських угруповань.

Висновки до розділу 2

В розділі 2 запропоновані структурні методи дослідження поверхні атак шляхом вивчення особливостей існуючих даних про атаки (звіти про діяльність хакерских угрупувань та баз, які використовуються фахівцями кібербезпеки, таких як MITRE ATT&CK). Проведений статистичний аналіз технік атак бази MITRE ATT&CK для виділення базових атак. Визначені найбільш часто зустрічаємі техніки, це Command and Control та Discovery. Запропоновано застосувати «обернену» постановку задачі – розглянути атаки, які утворюються на найбільш часто вживаних техніках з матриці MITRE та порахувати який відсоток від загальних атак вони складають – тобто застосувати конструктивний підхід, утворюючи множину всіх можливих атак та оцінювати її потужність. Цей підхід дозволить також більш точно вивчити структуру складних APT-атак, що утворюють поверхню атак, зокрема визначити чи утворюють атаки деяку алгебраїчну структуру.

Для вирішення задачі про покриття множини всіх атак достатньо виділити підмножини, які містять певну базову атаку, в порядку зменшення потужності цих підмножин. У випадку застосування алгоритму до бази атак, яка міститься в матриці MITRE ми повинні визначити частоти атак, які містять певні техніки атак – це і буде потужність відповідних підмножин, які складають покриття усієї бази атак. Якщо починати з технік, які зустрічаються найчастіше, ми отримуємо розташування підмножин в порядку зменшення їх потужності, що і буде рішенням задачі. Відповідні потужності (частоти) і будуть похибкою алгоритму – тож нам достатньо зупинитись, як тільки наша похибка буде задовільняти нашим потребам.

3 АНАЛІЗ ПРАКТИЧНОЇ ЕФЕКТИВНОСТ СТРУКТУРНИХ МЕТОДІВ ВИВЧЕННЯ ПОВЕРХНІ АТАК

Третій розділ присвячений оцінки ефективності запропонованих методів на практиці. Зауважимо, що теоретичну ефектиність алгоритму рішення задачі оптимізації було оцінено в другому розділі як ефективність та точність жадібного алгоритму вирішення задачі про покриття, до якого, як було показано вище, і зводиться задача оцінки поверхні атак для підвищення ефективності систем виявлення вторгнень.

Практична ефективність пов’язана із вирішенням задачі налаштування правил «за замовченням» для системи виявлення вторгнень. При цьому кількість правил повинна бути мінімальною, а кількість атак, яки виявляються за їх допомогою – максимальною.

Для демонстрації практичної ефективности запропонованих методів ми обираємо декілька найчастіших технік з бази атак MITRE та демонструємо як за допомогою захисту від однієї такої техніки ми захищаємося від як мінімум від K атак (тут K – параметр задачі про максимальне покриття (розділ 2). Також буде продемонстровано (додатково до демонстрації, наведеної в розділі 1) як руйнуванням одного ланцюжка (шляхом захисту від елементарної атаки в цьому ланцюжку) ми руйнуємо весь killchain, а значить – усю APT-атаку.

3.1 Опис тестового полігону для оцінки практичної ефективності розроблених методів

Для того, щоб продемонструвати практичну користь від виділення базових атак, а також перевірки тези, що була сформульована вище, виконано завдання розгортання тестової мережі з трьох віртуальних машин. Дві з цих машин грають роль жертви, це LinuxUbuntu 22.04 та Windows 10 Home. Роль хакера буде грати машина з LinuxKali. Хараектеристики машин наведено на рисунках 3.1, 3.2, 3.3:

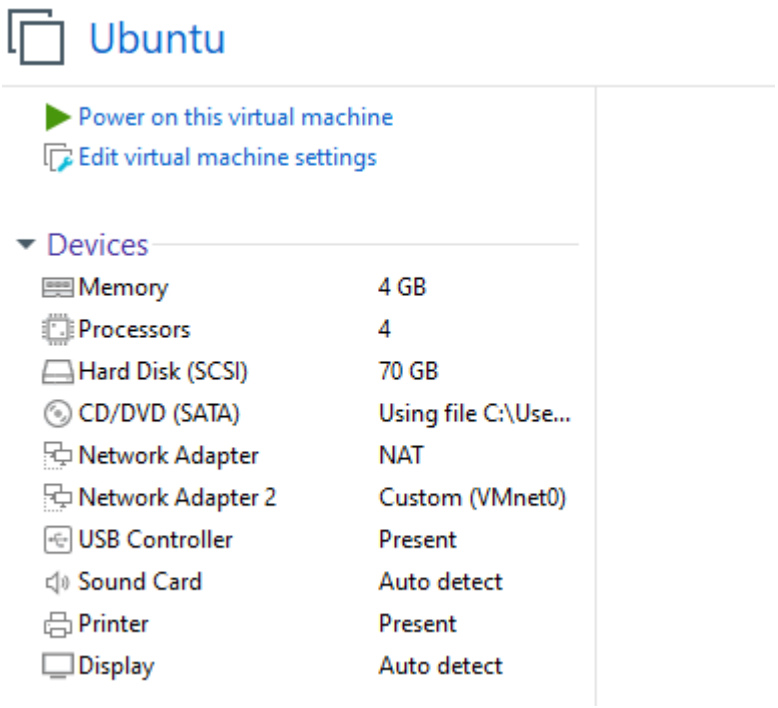

Рисунок 3.1 – Характеристики віратуальної машини Ubuntu

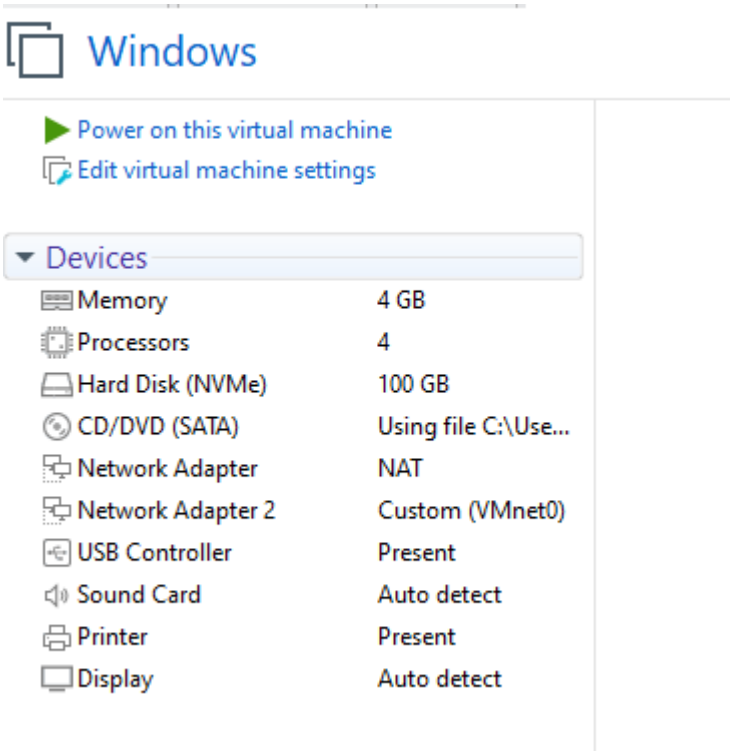

Рисунок 3.2 – Характеристики віратуальної машини Windows

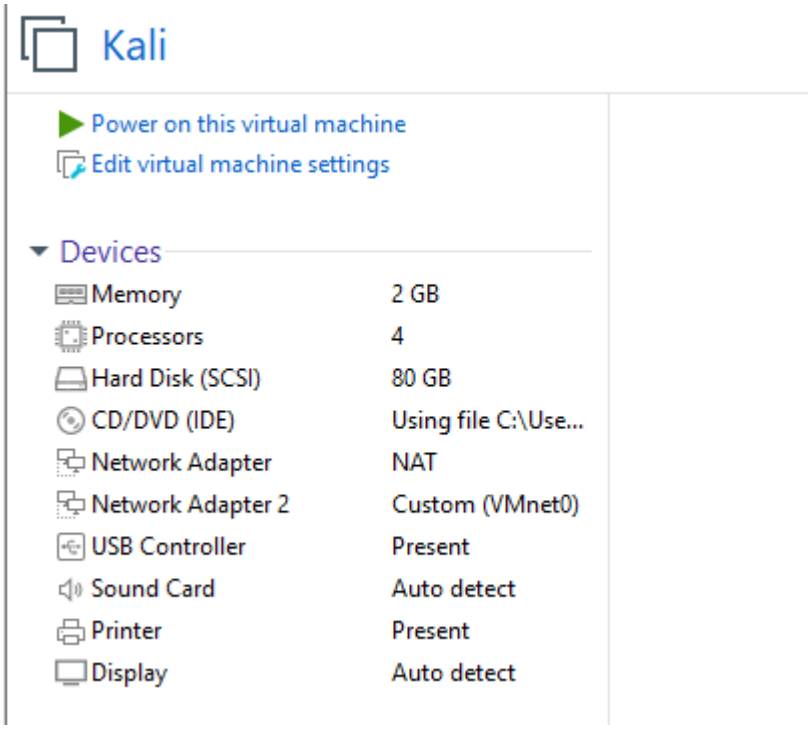

Рисунок 3.3 – Характеристики віратуальної машини Kali

На віртуальної машині Ubuntuрозгорнуто IPSSnort, на Windows 10 – SIEM Spunk. Як відомо, 0-day атаки краще визначаються SIEM системами, оскільки вони спрямовані на моніторинг аномальної активності. Прописавши у правилах SNORT сигнатури для базових атак, можна буде порівняти ефективність цього методу з фективністю стандартного набору правил SNORT та ефективністю виявлення 0-day атак у порівнянні з SIEM.

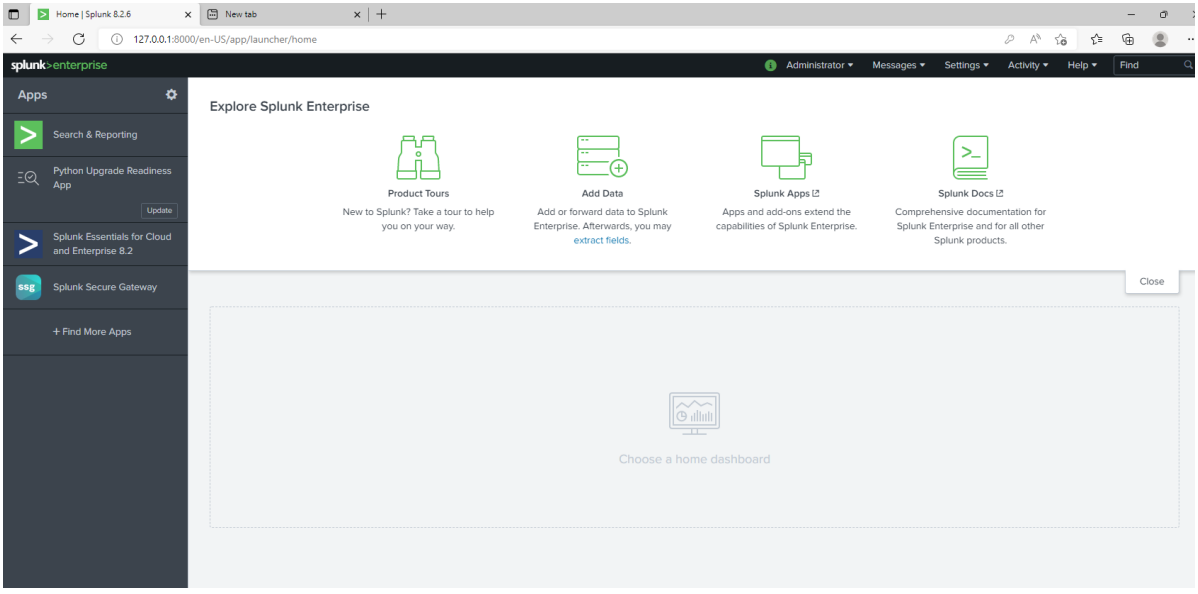

Рисунок 3.4 – Скріншот розгорнутого Splunky Windows 10

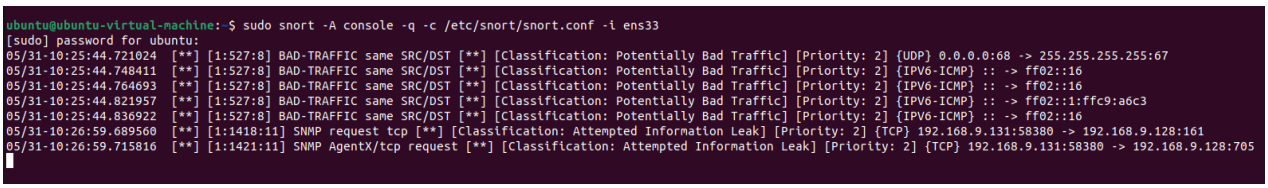

Рисунок 3.5 – Скріншот роботи SNORT, що встановлено на Ubuntu

3.2 Результати практичної ефективності виявлення атак за допомогою використання сигнатур базових атак

Оскільки запроновані саме методи, а не їх готові до використання реалізації, для демонстрації принципів їх роботи вистачить одного прикладу для кожного з методів.

Для демонстрації ефективності першого методу взято найбільш популярну згідно зібраної статистики техніку Mitre - Ingress Tool Transfer. Це техніка завантаження допоміжних файлів за допомогою системних утиліт, таких як wget, scp, curl тощо. Захистившись від цієї техніки ми захищаємось відразу від 51% всіх можливих атак(за матрицею Mitre), у тому числі, від деяких атак нульового дня, оскільки якщо у них присутня ця техніка(що є доволі вірогідним враховуючи її популярність) це унеможливить або ускладнить подільшу атаку, тобто тут застосовується принцип розриву killchain. Найпопулярнішими для зловмисників типами файлів, як відомо із [20] є файли пакету Microsoft office та exe, що використовуються в 98%випадків. Тому правила для Snort, яке наведено нижче просто забороняють завантажувати ці типи файлів:

alert tcp any any -> $HOME_NET any (msg:”Potentially malicious .exe file!”; content: „|4d 5a|”; sid: 1000004; rev:2;)

alert tcp any any -> $HOME_NET any (msg:”Potentially malicious offiice file!”; content: “| 2e 78 6d 6c|”; sid: 1000005; rev:2;):

Приклад спрацювання одного з цих правил на скріншоті нижче:

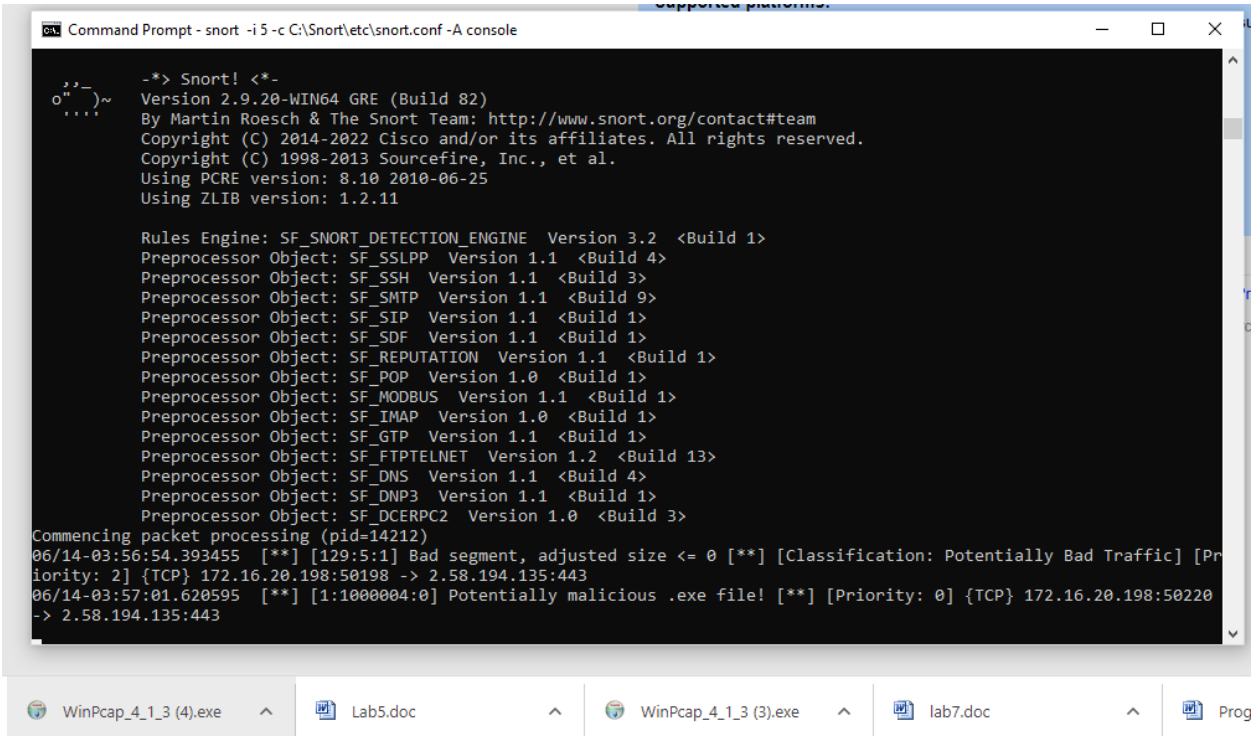

Рисунок 3.2.1- IPS Snort виявила заватаження .exe файлу

Щоб продемонструвати другий метод, необхідно детальніше розглянути техніки Mitre, що були найпопулярнішими у нашій статистиці, візьмемо перші 7 за частотою використання. Уважно розібравшись з їх суттю можна знайти у 4 з 7 цих правил одну спільну рису – всі вони використовують командну оболонку – термінал або cmd.

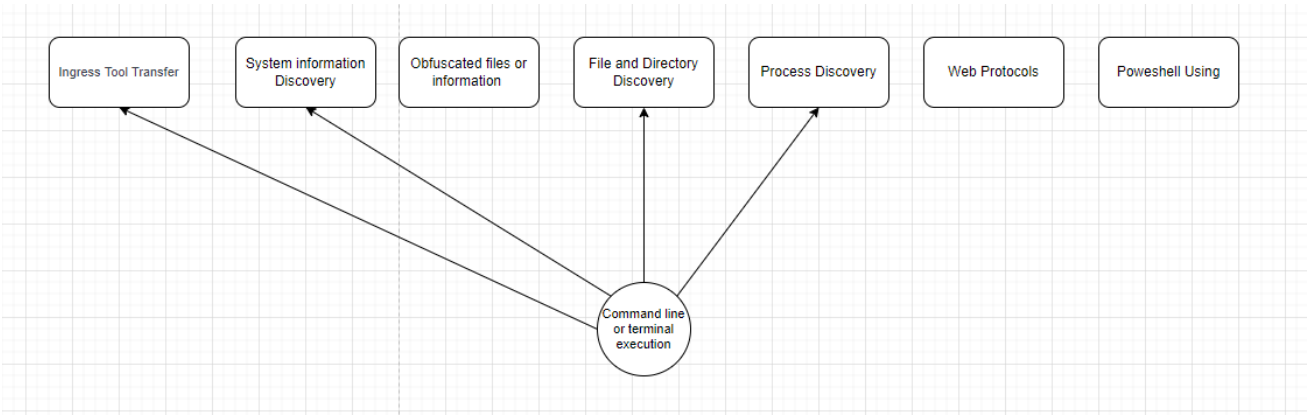

Рисунок 3.2.2 – Командну оболонку використовують 4 з 7 найпопулярніших технік

Тож у цьому випадку просто заблоковуваши командну оболонку у системі, можна захиститися від ще ширшого классу атак. У Windows та Linux це можна зробити за кілька хвилин, у Windows, редагуючи реестр, у linux – зміною прав доступу до терміналу.

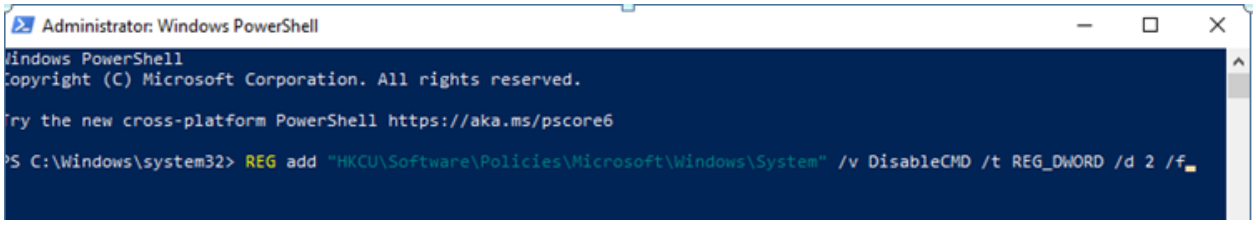

Рисунок 3.2.2 – Редагування реестру Windows

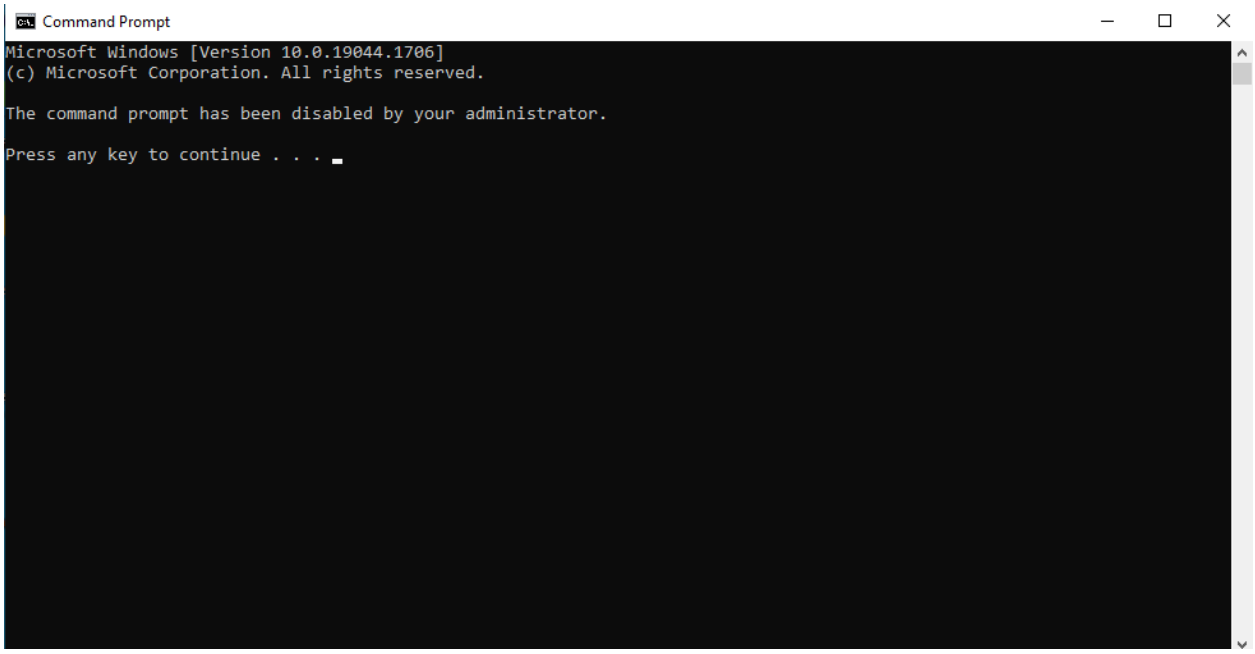

Рисунок 3.2.3 – Відмова у використанні cmd звичайному користувачеві

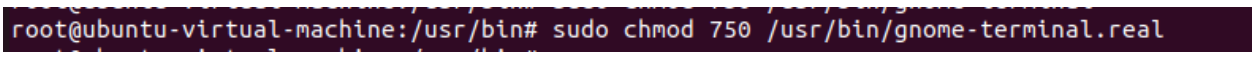

Рисунок 3.2.4 – Зміна прав доступу для термінала ubuntu

Використання командної оболонки – це звичайна дія користовача, проте її запуск дуже часто є елементом атаки. Яскравим прикладом є і DOS атака, яка складається по суті з багатьох підключень до ресурсу – абсолютно звичайних дій у системі. Виходячи з цього прикладу можна сказати, що множина базових атак за алгебраїчною структурою як мінімум є групоїдом, оскільки має операцію

конкатенації. Перевірка цієї множини на інші алгебраїчні властивості є темою для подальших досліджень.

Спільною перевагою обох підходів є можливість значно зменшити кількість правил для виялення атак, що означає значно простіше адміністрування систем IPS/IDS, особливо у великих організаціях.

„Вузьким місцем" обох підходів є компроміс між зручністю використання системи та безпекою. Цей компроміс залежить від конкретних обставин у яких застасовуються ці методи. Іноді, достатньо додати ір адреси із яких заборонено завантажувати данні типи файлів(для першого випадку) або ж заборонити коритуватися командною оболонкую усім користувачам окрім адміністратора(у другому випадку), що незначно вплине на відсоток атак, від якого захищена система(свій у кожному конкретному випадку) проте значно покращить зручність роботи з системою. Виходячи з цього, однією с тем для подальшх досліджень може бути розробка программного забезпечення, що автоматично шукає компромісс між жорсткістю правил для визначення базових атак, так щоб вони максимально покривали поверхню атак та зручністю роботи користувача це робити.

Висновки до розділу 3

В третьому розділі приведено результати оцінки практичної ефективності отриманих методів.

Для перевірки цього методу розгорнуто тестову мережу із трьох віртуальних машин. Дві з цих машин грають роль жертви, це LinuxUbuntu 22.04 та Windows 10 Home. Роль хакера «грає» машина з LinuxKali.

На базі найчастіших технік з бази атак MITRE продемонстровано як за допомогою захисту від однієї такої техніки ми захищаємося від як мінімум від K атак (тут K – параметр задачі про максимальне покриття (розділ 2). Також було продемонстровано (додатково до демонстрації, наведеної в розділі 1) як

руйнуванням одного ланцюжка (шляхом захисту від елементарної атаки в цьому ланцюжку) ми руйнуємо весь killchain, а значить – усю APT-атаку

ВИСНОВКИ

В даній роботі представлені результати досліджень актуальної на даний час теоретичної та практичної задачі – створення методів аналізу поверхні атак інформаційних систем, що дозволять підвищити ефективність їх виявлення сучасними системами захисту інформації. Актуальність задачі пояснюється необхідністю створення практичних методів оцінки поверхні атак, які можуть бути практично застосовані для широкого класу інформаційних систем на основі досліідження існуючих баз атак. В ході вирішення задачі було проведено аналіз існуючих джерел даних про сучасні інтелектуальні атаки на інформаційні системи; аналіз недоліків існуючих методів вивчення поверхні атак та виявити галузь їх застосування; аналіз методів струтурізації та класифікації атак; статистичні дослідження існуючих даних про інтелектуальні атаки, що використовуються сучсними хакерскими угруповуваннями та іншими базами даних, які використовуються спільнотою фахівців з кібербезпеки; розроблені структурні методи аналізу інтелектуальних атак на основі жадібного алгоритму вирішення задачі про покриття множини атак; наведено практичну демонстрацію працездатності запропонованих методів виявлення інтелектуальних атак.

В результаті проведених досліджень з’ясовано:

головним недоліком існуючих класифікацій кібератак є неоднозначність та відсутність чіткого зв’язку між різними класифікаціями. Описано атаки, що використовує хакерське угрупування Armageddon в термінології Mitre та Capec. За результатами цього опису практично піддтвержено вказані вище недоліки, а також виявлено недосконаліть у структурі технік матриці Mitre;

використання звичайних утиліт-інструментів адміністратора, яке НЕ Є ТЕХНІКАМИ атак, в якості техніки атаки;

найбільш часто зустрічаємі техніки атак в базі даних MITRE - це Command and Control та Discovery.

В роботі запропоновано:

- для визначення поверхні атак сучасних інформаційних систем з метою виявлення її особливостей, які корисні для виявлення атак, застосувати «обернену» постановку задачі побудови поверхні атак – розглянути атаки, які утворюються на найбільш часто вживаних техніках з матриці MITRE та порахувати який відсток від загальних атак вони складають;

- жадібний алгоритм вирішення задачі про покриття множини атак, ідея якого полягає у виділенні підмножини, які містять певну базову атаку, в порядку зменшення потужності цих підмножин. У випадку застосування алгоритму до бази атак, яка міститься в матриці MITRE ми повинні визначити частоти атак, які містять певні техніки атак – це і буде потужність відповідних підмножин, які складають покриття усієї бази атак. Якщо починати з технік, які зустрічаються найчастіше, ми отримуємо розташування підмножин в порядку зменшення їх потужності, що і буде рішенням задачі. Відповідні потужності (частоти) і будуть похибкою алгоритму – тож нам достатньо зупинитись, як тільки похибка буде задовільняти нашим потребам;

- на базі виділення підмножин, які покривають множину атак, запропоновано метод зменшення кількості правил системи виявлення атак та практично продемонстровано можливість його роботи на прикладі системи SNORT.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Security and Privacy Controls for Information Systems and Organizations. – NIST Special Publication 800-53 Revision 5. - September 2020. – 465 p.

2. Грайворонський М.В., Новіков О.М. «Безпека інформаційно-комунікаційних систем». – К.: Видавнича група BHV.- 2009.-608 с.

3. Качинський А.Б. Безпека складних систем // А.Б. Качинський. – К.: ТОВ «Юстіон», 2017. – 494 с.

4. Коломицев М.В., Стьопочкіна І.В. Комплексні системи захисту інформації: проектування, впровадження, супровід/ Методичні вказівки до комп'ютерного практикуму. – НТУУ «КПІ імені Ігоря Сікорського». – Електронне видання, 2017 р.

5. Матриця MitreATT&CK – 2022 – Режим доступу: https://attack.mitre.org/

6. Common Vulnerabilities and Exposures – 2022 – Режимдоступудоресурсу: https://cve.mitre.org/

7. National Vulnerability Database – 2022 – https://nvd.nist.gov/

8. Common Weakness Enumeration– 2022 – Режимдоступудоресурсу: https://cwe.mitre.org/

9. Common Attack Pattern Enumeration and Classification – 2022 – Режимдоступудоресурсу: https://capec.mitre.org/

10. LockheedKillchain – 2022 – Режим доступу до ресурсу: https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html

11. Unified Cyber Kill Chain – 2022 –Режим доступу до ресурсу: https://www.unifiedkillchain.com/

12. Cyber Kill Chain, MITRE ATT&CK, and Purple Team, Jorge Orchilles – 2022 –Режимдоступудоресурсу: https://www.sans.org/blog/cyber-kill-chain-mitre-attack-purple-team/

13. Interactive Visualization and Text Mining For the CAPEC Cyber Attack Catalog, Steven Noel – 2015 – Режимдоступудоресурсу:https://csis.gmu.edu/noel/pubs/2015_CAPEC_viz.pdf

14. Tracing CAPEC Attack Patterns from CVE Vulnerability Informationusing Natural Language Processing Technique, Kenta Kanakogi, Hironori Washizaki, Yoshiaki Fukazawa, Shinpei Ogata, Takao Okubo, Takehisa Kato, Hideyuki Kanuka, AtsuoHazeyama, Nobukazu Yoshioka – 2015 – Режим доступу до ресурсу: https://scholarspace.manoa.hawaii.edu/server/api/core/bitstreams/164f1948-02b1-425b-8001-cc0ff8eb926c/content

15. Garmaredon/Armageddon Group FSB RF cyber attacks against Ukraine. – Kyiv: SSU,2021. – 35 p. - Режим доступу до ресурсу: https://ssu.gov.ua/uploads/files/DKIB/Technical%20report%20Armagedon.pdf

16. “TadGAN: Time Series Anomaly Detection Using Generative Adversarial Networks”, https://arxiv.org/abs/2009.07769

17. Sherry Liang, Khalid Alanazi, Kumail Al Hamoud Set covering problem / Режим доступу до ресурсу: https://optimization.cbe.cornell.edu/index.php?title=Set_covering_problem#References

18. D.S. Johnson. Approximation algorithms for combinatorial problems // STOC '73: Proceedings of the fifth annual ACM symposium on Theory of computing. - New York, NY, USA: ACM Press, 1973. - pp. 38-49.

19. Petr Slavík A Tight Analysis of the Greedy Algorithm for Set Cover / Journal of Algorithms, Volume 25, Issue 2. – 1997. – Pages 237-254.

20. X-Force Threat Intelligence Index, Camille Singleton, Charlotte Hammond, Vio Onut etc. - 2022 – Pages 5-12.

21. 2020-data-breach-investigations-report, Akamai Technologies etc. – 2020.

22. Mitre ATT&CK Navigator – 2022 – Режим доступу: https://mitre-attack.github.io/attack-navigator/