Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ДОПУСТИТИ ДО ЗАХИСТУ:
завідувачка кафедри кібербезпеки
та захисту інформації
_____ Наталія ЛУКОВА-ЧУЙКО
«14» червня 2022р.

ПОЯСНЮВАЛЬНА ЗАПИСКА

дипломної роботи
бакалавра
(назва освітнього рівня)

галузь знань 12 Інформаційні технології
(шифр і назва галузі знань)

спеціальність 125 Кібербезпека
(код і назва спеціальності)

освітня програма Кібербезпека
(назва освітньої програми)

на тему: «Виявлення вторгнення у корпоративні мережі»

Виконавець: студент IV курсу, групи КБ-41

_____ **Микита МЕРКУЛОВ**
(підпис) (ім'я прізвище)

	Прізвище, ініціали	**Підпис**
Керівник	Микола БРАІЛОВСЬКИЙ	

Нормоконтроль	Юрій ЩЕБЛАНІН	

Київ 2022

Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ЗАТВЕРДЖЕНО:
завідувачка кафедри кібербезпеки
та захисту інформації
_____ Наталія ЛУКОВА-ЧУЙКО
«01» листопада 2021 р.

ЗАВДАННЯ
на виконання дипломної роботи

спеціальності 125 Кібербезпека
(код і назва спеціальності)

освітньої програми Кібербезпека
(назва освітньої програми)

Студентові **КБ-41** (група) **Меркулову Микиті Денисовичу** (прізвище ім'я по-батькові)

Тема дипломної роботи Виявлення вторгнення у корпоративні мережі

1. ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Тема дипломної роботи затверджена на засіданні кафедри кібербезпеки та захисту інформації протокол №5 від 29.10.2021 р.

2. ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Технології виявлення вторгнення у корпоративні мережі

3. ЗМІСТ РОЗРАХУНКОВО-ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

Необхідно ознайомитись з основними технологіями виявлення вторгнення у Корпоративних мережах, обрати технології для практичної реалізації, розглянути Алгоритми створення правил кореляції та платформу категоризації загроз, Розробити програму для збору подій та правила кореляції.

4. ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Практична цінність Розроблено програму для збору подій з операційної

Системи і створено правила кореляції для виявлення вторгнення.

5. ДАТА ВИДАЧІ ЗАВДАННЯ

Дата видачі завдання: 29.10.2021 року

Завдання видав _____ Микола БРАІЛОВСЬКИЙ
(підпис) (ім'я, прізвище)

Завдання прийняв до виконання _____ Микита МЕРКУЛОВ
(підпис) (ім'я, прізвище)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Найменування етапів робіт	Строки виконання робіт (початок-кінець)	Відмітка про виконання
1	Уточнення постановки задачі	29.10.2021 – 22.01.2022	*виконано*
2	Аналіз літератури	29.01.2022 – 13.02.2022	*виконано*
3	Аналіз основних рішень для виявлення вторгнень	14.02.2022 – 15.02.2022	*виконано*
4	Вибір рішень для виконання практичної частини	16.02.2022 – 04.03.2022	*виконано*
5	Створення алгоритму написання правил кореляції та огляд синтаксису обраної системи	05.03.2022 – 21.03.2022	*виконано*
6	Огляд платформи Mitre Att&ck	22.03.2022 – 08.04.2022	*виконано*
7	Розробка програмної частини та написання правил кореляції	09.04.2022 – 10.05.2022	*виконано*
8	Оформлення пояснювальної записки	11.05.2022 – 27.05.2022	*виконано*
9	Підготовка до захисту дипломної роботи	28.05.2022 – 13.06.2022	*виконано*

Завдання видав _____ Микола БРАІЛОВСЬКИЙ
(підпис) (ініціали, прізвище)

Завдання прийняв до виконання _____ Микита МЕРКУЛОВ
(підпис) (ініціали, прізвище)

Термін подання дипломної роботи до ЕК 06 червня 2022 року

РЕФЕРАТ

Пояснювальна записка дипломної роботи складається зі вступу, трьох розділів, загальних висновків, списку використаних джерел та додатків. Основний текст займає 43 сторінок, включаючи вступ, три розділи дипломної роботи, висновки та список джерел. У пояснювальній записці міститься 29 картинки та 2 додатки.

Об'єкт дослідження: процес виявлення вторгнення у корпоративні мережі.

Метою даної роботи є розробка програми для збору інформації, а також створення правил кореляції для виявлення вторгнення у корпоративні мережі.

Для досягнення поставленої мети необхідно зробити наступне:

- Дослідити критерії порівняння SIEM систем.
- Обрати технічний інструментарій для реалізації практичного завдання.
- Розробити програму для збору логів з обраної операційної системи.
- Написати правила кореляції для обраної SIEM системи.

Предмет дослідження: технології виявлення вторгнення у корпоративні мережі.

Методи дослідження: спостереження, порівняння та аналіз відкритих джерел.

Практична значимість роботи полягає у розробці програмного забезпечення для збору подій операційної системи і правил для виявлення вторгнення на основі зібраних подій.

Актуальність роботи полягає в тому, що кількість загроз інформаційної безпеки збільшується з кожним роком, а технічні можливості нападників стають все більш різноманітними, отже своєчасне виявлення вторгнення є одним із найбільш важливих аспектів забезпечення безпеки корпоративних мереж.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

ПЗ	–	Програмне забезпечення
ШПЗ	–	Шкідливе програмне забезпечення
NIDS	–	Network Intrusion Detection System
HIDS	–	Host-based Intrusion Detection System
SIEM	–	Security information and event management
APT	–	Advanced Persistent Threat
IDS	–	Intrusion Detection System
IPS	–	Intrusion Prevention System
INIDS	–	Intelligent Network Intrusion Detection System

ЗМІСТ

ВСТУП

Актуальність роботи полягає в тому, що кількість загроз інформаційної безпеки збільшується з кожним роком, а технічні можливості нападників стають все більш різноманітними, отже своєчасне виявлення вторгнення є одним із найбільш важливих аспектів забезпечення безпеки корпоративних мереж.

На сьогоднішній день, захист корпоративних мереж є одним із основних завдань кібербезпеки. Із розвитком технологій, фахівці на підприємствах мають можливість використовувати цілий ряд продуктів та технологій з інформаційної безпеки – починаючи зі звичних антивірусів, закінчуючи платформами для пошуку вразивостей зі штучним інтелектом. Втім, як показує практика, навіть враховуючи немалий технічний потенціал для боротьби з кіберзагрозами, згідно дослідженню Дюкського університету [1], близько 80 відсотків компаній у США були зламані. Більш того, як можна побачити зі звіту компанії ESET за 2 квартал 2021 року [2], кіберзлочинцям вдалося використати широкий арсенал різного зловмисного програмного забезпечення (ПЗ) та здійснити атаки різного типу, що часто призводило до компрометації мережі компанії.

Усе це говорить про те, що проблема забезпечення безпеки корпоративних мереж досі є доволі гострою. Одним із методів захисту є виявлення вторгнення, адже це може не тільки запобігти подальшим діям злочинців у мережі, а й допомогти уникнути такого роду атак у майбутньому завдяки збору індикаторів компрометації та навчанню персоналу правил забезпечення кібербезпеки на прикладі минулого вторгнення.

Тому **метою роботи** є розробка програми для збору інформації, а також створення правил кореляції для виявлення вторгнення у корпоративні мережі.

Для досягнення поставленої мети необхідно вирішити такі *завдання*:

- Дослідити критерії порівняння SIEM систем.
- Обрати технічний інструментарій для реалізації практичного завдання.

- Розробити програму для збору логів з обраної операційної системи.
- Написати правила кореляції для обраної SIEM системи.

Об’єктом дослідження є процес виявлення вторгнення у корпоративні мережі.

Предметом дослідження є технології виявлення вторгнення у корпоративні мережі.

Методи дослідження: спостереження, порівняння та аналіз відкритих джерел.

РОЗДІЛ 1
ОГЛЯД ОСНОВНИХ СИСТЕМ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ КОРПОРАТИВНИХ МЕРЕЖ

1.1 Технології виявлення вторгнення у корпоративних мережах

До основних сучасних технологій, які використовуються для виявлення вторгнень у корпоративних мережах [3] відносяться:

- NIDS системи;
- HIDS системи;
- Honeypots;
- SIEM [4].

Розглянемо по черзі кожну з них.

1) NIDS (англ. Network Intrusion Detection System) – система виявлення мережевих вторгнень, яка відстежує такі види шкідливої діяльності, як атаки DoS, сканування портів або навіть спроби проникнення в мережу. Зазвичай, NIDS пасивно перевіряють трафік, що проходить через пристрої, на яких вони сидять. NIDS можуть бути апаратними або програмними системами і, залежно від виробника системи, можуть підключатися до різних мережевих середовищ, таких як Ethernet, Fast Ethernet та інші. Часто NIDS мають два мережеві інтерфейси. Один використовується для прослуховування мережевого трафіку у безладному режимі, а інший використовується для контролю та звітування [5].

2) HIDS (англ. Host-based Intrusion Detection System) - програма, яка перевіряє комп’ютер або мережу на предмет підозрілої діяльності, яка може включати вторгнення зовнішніх суб’єктів, а також неправомірне використання ресурсів або даних внутрішніми користувачами. Інструменти HIDS відстежують журнали, створені програмами, створюючи історичний запис діяльності та функцій, що дозволяє швидко шукати в них аномалії та ознаки вторгнення. Вони також

компілюють файли журналів і дозволяють упорядковувати їх у відповідності зі структурою каталогів журналів, що полегшує пошук або сортування файлів за програмою, датою чи іншими показниками [6].

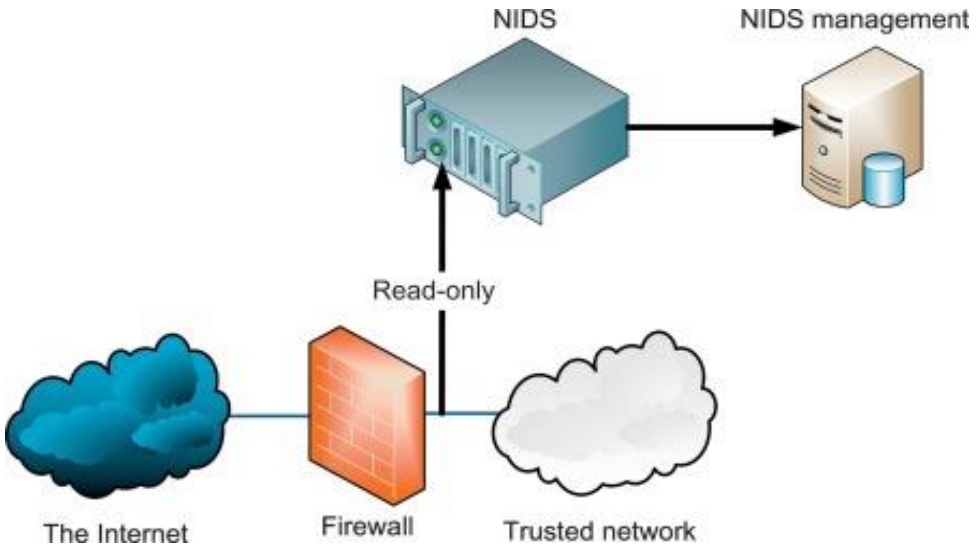


Рисунок 1.1 – Схема розташування NIDS у корпоративній мережі

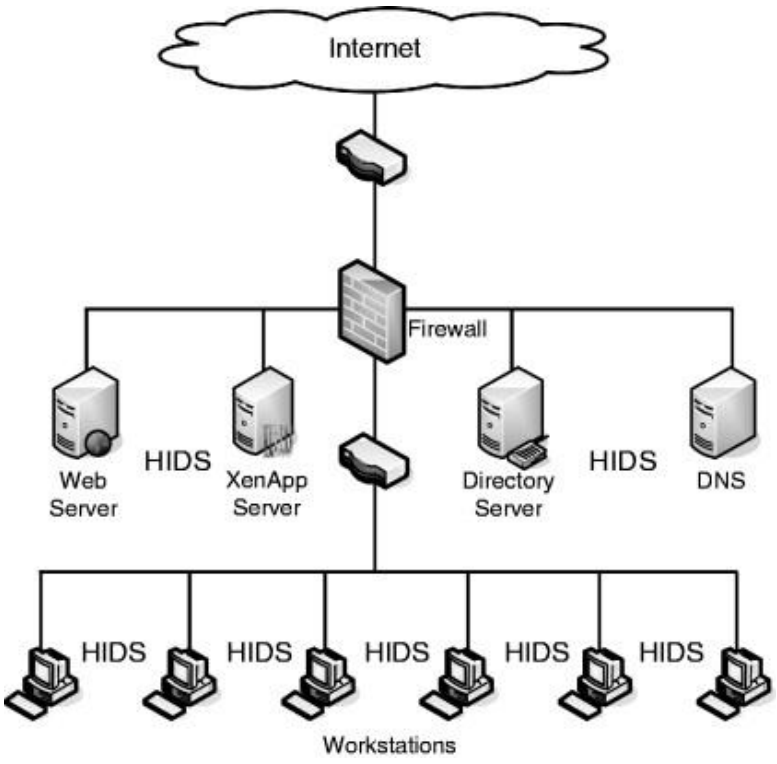


Рисунок 1.2 – Розташування HIDS у корпоративній мережі

1.2 Огляд Honeypot систем

Окремої уваги заслуговують так звані Honeypot (англ. Горщик з медом) – це система, основна мета якої бути атакованою. Завдяки ній, фахівці можуть дослідити хід атаки, техніки, які були використані, а також потенційно виявити проникнення на інші вузли корпоративної мережі. Залежно від характеристик, Honeypots поділяються на:

- Дослідницькі – використовуються дослідниками, військовими та іншими зацікавленими особами з метою вивчення потенційних загроз. Вони також можуть бути використанні фахівцями на підприємстві. Зазвичай, пастки такого типу надають багато детальної інформації про атаку, а також часто відтворюють повноцінну середу для більш детального вивчення зловмисних дій.

- Виробничі – використовуються на підприємствах з метою виявлення вторгнення, а також задля зниження ризиків та покращення загального захисту. Порівняно з дослідницькими, мають менше можливостей та менш реалістичну середу, зазвичай надають інформацію про скомпрометовані вузли та використані зловмисні програми [7].

Також, за рівнем взаємодії, ханіпоти можливо розподілити наступним чином:

- З високим рівнем взаємодії – збирають багато інформації завдяки реалістичній симуляції операційної системи, втім вважаються більш ризикованими адже дозволяють атакуючими здійсняти більше дій.

- З середнім рівнем взаємодії – збирають менше інформації, адже не симулюють операційну систему.

- З низьким рівнем взаємодії – збирають мінімальний обсяг даних, втім вважаються найбільш безпечними.

Honeypot може бути розташований:

- У демілітаризованій зоні. Там, зазвичай розташовуються сервери, які є загальнодоступними, тому наявність там пастки може значно покращити безпеку.

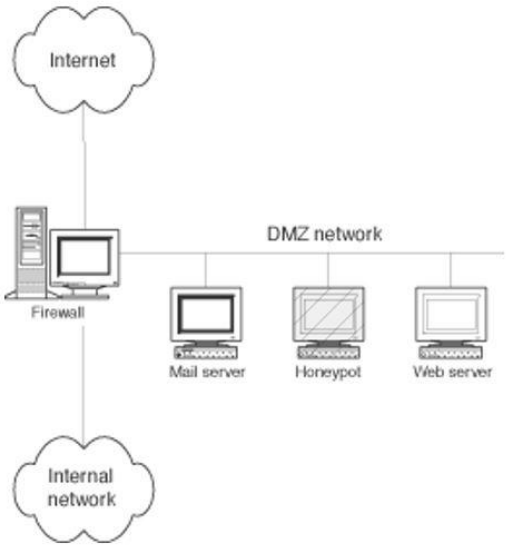


Рисунок 1.3 – Розташування пастки у демілітаризованій зоні

- У локальній мережі. У такому разі, honeypot може виявити вторгнення у корпоративну мережу завдяки логуванню.

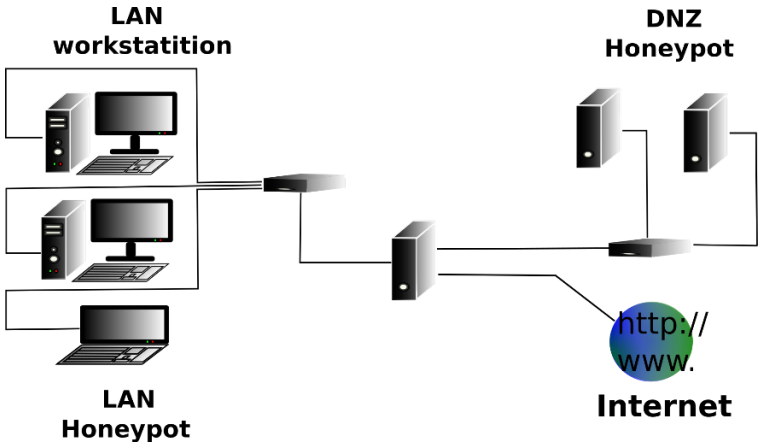


Рисунок 1.4 – Розташування пастки у локальній мережі

1.3 Огляд SIEM систем

Втім, найбільш використовуваним для виявлення вторгнень є SIEM (Security information and event management) – система, що забезпечує отримання багатьох видів подій, які походять від різних пристроїв та програм. Також, такого роду система здатна аналізувати отриманні журнали з метою виявлення підозрілої активності, яка може свідчити про вторгнення. Загальні здатності SIEM наступні:

- Агрегація даних: керування журналами об’єднує дані з багатьох джерел, включаючи мережу, безпеку, сервери, бази даних, додатки, надаючи можливість консолідувати дані, що відстежуються, щоб уникнути пропуску важливих подій.

- Кореляція: шукає загальні атрибути та об’єднує події у значущі групи. Ця технологія надає можливість виконувати різноманітні методи для інтеграції різних джерел, щоб перетворити дані в корисну інформацію.

- Сповіщення: Автоматичний аналіз корельованих подій

- Комплаєнс: програми можна використовувати для автоматизації збору даних про відповідність, створення звітів, які адаптуються до існуючих процесів безпеки, управління та аудиту.

- Зберігання: використання тривалого зберігання історичних даних для полегшення кореляції даних у часі та забезпечення збереження, необхідного для вимог до відповідності. Довгострокове збереження даних журналу має вирішальне значення для судово-медичних розслідувань, оскільки малоймовірно, що порушення мережі буде виявлено під час його виникнення.

- Криміналістичний аналіз: можливість пошуку в журналах на різних вузлах і періодах часу на основі певних критеріїв. Це пом’якшує необхідність збирати інформацію журналів.

Рисунок 1.5 – Функціонал сучасних SIEM систем [13]

Однією з основних переваг SIEM систем є те, що майже будь-яка технологія, або система, що генерує журнали подій може бути підключена до SIEM систем, такі як:

- міжмережевий екран;
- операційні системи вузлів;
- контролери доменів;
- сервера антивірусного захисту;
- системи виявлення вторгнення;
- файлові сервери;
- сервери баз даних;
- портали;
- мережеве обладнання.

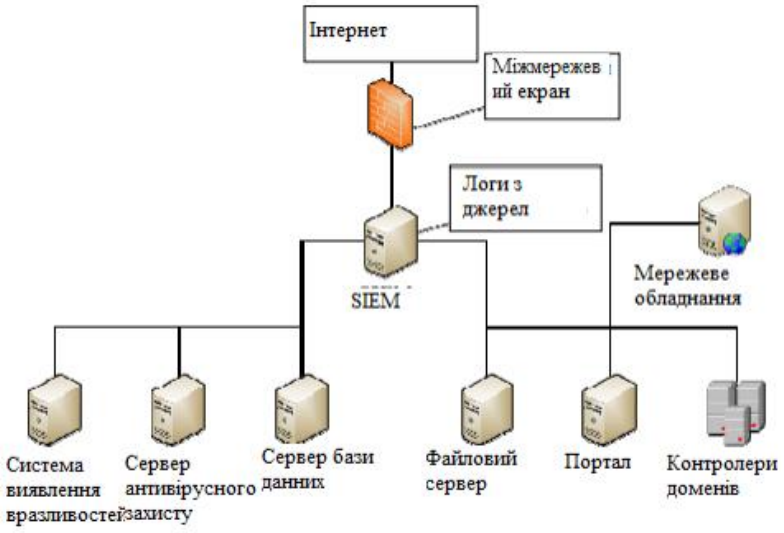


Рисунок 1.6 – Демонстрація підключення різних джерел подій

Таким чином, можливо дійти висновку, що саме SIEM системи мають найбільший потенціал виявлення вторгнення завдяки можливості збору файлів подій з різних джерел і можливості обробки завдяки правил кореляції. Враховуючи це, методика виявлення вторгнення на базі таких систем потребують подальшого дослідження.

1.4 Постановка завдання

Таким чином, для реалізації технології виявлення вторгнення на основі SIEM системи, необхідно зробити наступні кроки:

- Дослідити критерії порівняння SIEM систем.
- Обрати технічний інструментарій для реалізації практичного завдання.
- Розробити програму для збору логів з обраної операційної системи.
- Написати правила кореляції для обраної SIEM системи.

Висновки за розділом 1

Таким чином, у 1 розділі мною було розглянуто основні технології виявлення вторгнення, що використовуються на підприємствах: NIDS, HIDS, Honeypot та SIEM системи. Після розгляду дійшов висновку, що саме SIEM системи мають найбільший потенціал для виявлення вторгнення, після чого було поставлено задачі для виконання у наступних розділах дипломної роботи.

РОЗДІЛ 2
ВИБІР ТЕХНОЛОГІЙ ДЛЯ РЕАЛІЗАЦІЇ ВИЯВЛЕННЯ ВТОРГНЕННЯ

2.1 Критерії порівняння SIEM систем

Для виконання практичної роботи, необхідно підібрати SIEM систему. На даний момент існує широкий вибір таких систем, які можливо використовувати на підприємстві. Тим не менш, вибір окремої системи залежить багатьох чинників, у тому числі від необхідного функціонала, розташування(наприклад, дехто може надати перевагу розташуванню у хмарі), вартості і т.п.

Наведемо основні критерії порівнянь SIEM систем, на прикладі IBM Qradar та Arcsight [14]:

- Передобробка лог-файлів. Де-які системи спроможні ідентифікувати та аналізувати наданий файл подій “на льоту”, тобто, з мінімальним втручанням фахівця. Такий підхід значно покращує ефективність роботи, адже підключення, таким чином, проводиться значно швидше. Втім, якщо Qradar має такий функціонал, то Arcsight спочатку зберігає наданий файл у сховище, і тільки після цього обробляє його.

- Додавання конекторів. Легкість розробки і кількість перед встановлених конекторів є важливим, адже від легкості напряму залежить ефективність роботи, а від кількості передвстановленних – можливості підключати джерела подій без зайвих зусиль зі сторони відповідальних співробітників.

- Кількість стандартних правил кореляції. Зазвичай, фахівці з інформаційної безпеки, що займаються виявленням вторгнення та моніторингу пишуть подібні правила враховуючи потреби окремої організації, включаючи найпоширенішу операційну систему, додатки що працюють на тому чи іншому вузлі корпоративної мережі, і т.п. Втім, слід помітити, що стандартні правила також здатні виявляти зловмисну активність, більш того, зазвичай ці правила є перевіреними і універсальними. За бажанням, їх можливо як увімкнути, так і вимкнути. IBM Qradar

має вбудовані правила, так як є “рішенням з коробки”, ArcSight теж має певний набір таких правил, який є меншим.

- Автовизначення джерел подій. Де-які системи спроможні автоматично виявляти тип подій, що надсилається, інші потребують ручного налаштування. У той час, як Qradar має таку можливість, Arcsight - ні.

- Візуалізація зкорельованих подій. У той час, як IBM Qradar може надати таку важливу інформацію, як топологія мережі, інформацію про порушника чи вразливість, ArcSight лише констатує сам факт наявності події.

2.2 Вибір системи для реалізації виявлення вторгнення

Для проведення практичної частини роботи було обрано SUMO Logic Siem. Користуючись зазначеною методикою, виявимо основні його недоліки і переваги:

1) Має можливість автоматичної передобробки лог-файлів.

2) Має змогу додавання конекторів.

3) Має великий перелік вбудованих правил, який втім є недостатнім для гарантованого виявлення вторгнення.

4) Має змогу автовизначення джерела подій.

5) Має можливість візуалізації подій.

Також слід зазначити, що SUMO Logic є хмарним рішенням, що забезпечує швидку і безперебійну поставку та обробку подій різних джерел.

Згідно дослідженню компанії Gartner [8], дана система отримала оцінку 4.4 з 5, що свідчить про ефективність та загальну задоволеність клієнтами від досвіду роботи з SUMO.

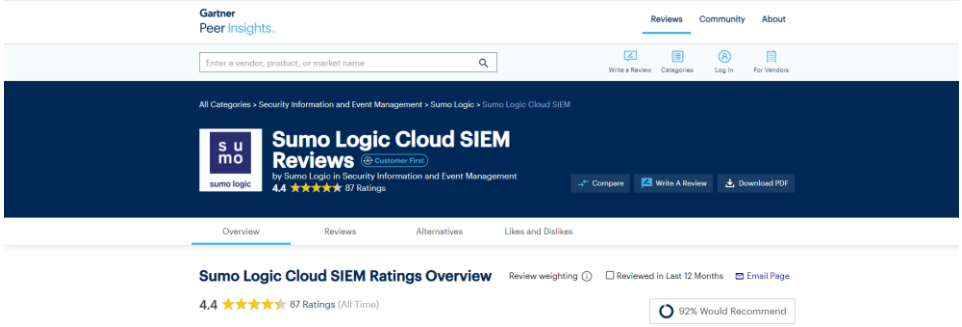


Рисунок 2.1 – Оцінка Gartner

Окрім того, згідно з маркетингового дослідженню Magic Quadrant цієї ж компанії, SUMO Logic має впевненні позиції на міжнародному ринку:

Source: Gartner (June 2021)

Рисунок 2.2 – Результати маркетингового дослідження Gartner

Sumo Logic робить аналіз даних простішим та ефективнішим, надаючи своїм користувачам глибоку видимість повного налаштування додатків та інфраструктури. Адміністратори можуть збирати дані, прискорювати доставку додатків, дотримуватись вимог і покращувати безпеку. Основною метою цього рішення є надання часових рядів і показників керування журналами, що дозволяє користувачам створювати та запускати програми AWS, Azure, Google Cloud Platform або Hybrid. Sumo Logic працює в хмарі, з легким розгортанням та запуском, тому користувачі отримують повну функціональність за дуже короткий проміжок часу [9].

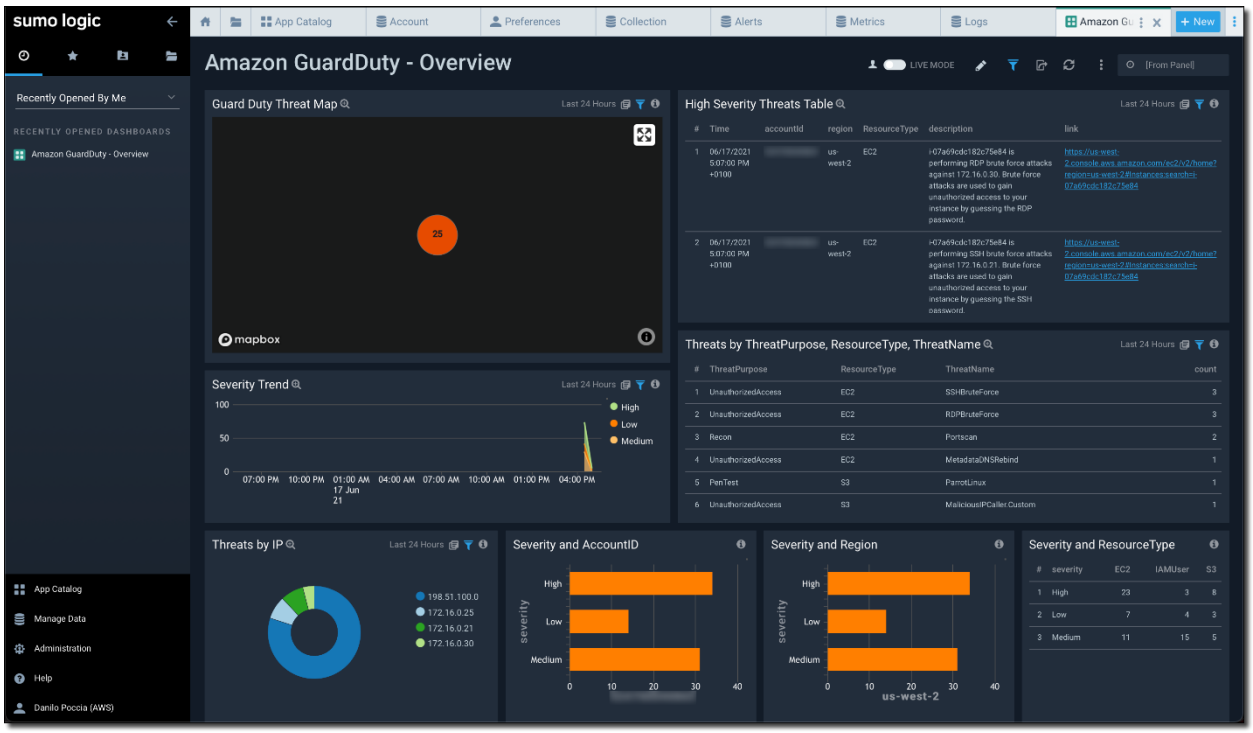


Рисунок 2.3 – Приклад інтерфейсу Sumo Logic SIEM

2.3 Вибір мови програмування

У якості мови програмування було обрано Python. Це – інтерпретована мова програмування, що має строгу динамічну типизацію та була розроблена Гвідо ван Россумом у 1990 році.

Згідно інформації Північно-Східного університету [10], Python даний момент є найбільш популярною мовою програмування. До його переваг відносять [11]:

- Python має модулі та пакети, що полегшує повторне використання коду.
- Python є мовою з відкритим вихідним кодом
- Немає компіляції коду – цикл Edit-test-debug швидкий.
- Підтримка обробки винятків.
- Автоматичне управління пам'яттю.

Python використовується у багатьох сферах розробки, таких як веб-розробка, розробка застосунків для ПК, машинне навчання, тощо

Порівняємо Python з іншою мовою програмування, яка також підходить для виконання практичної частини роботи – з c# [16].

У той час, як обидві мови є з відкритим доступом, Python має набагато більше бібліотек, є динамічною мовою програмування у той час як c# є статичною. Окрім того, зазначається, що Python має більш простий синтаксис, а також має менше символів у той час як c# є набагато складнішим і має більше символів. Втім, слід зазначити, що не дивлячись на усі свої переваги, Python вважається більш повільним.

C#	Python
Open-source language developed by Microsoft	Open-source language with free distribution
Has .NET framework's base category library	Rich with a customary vast library
Statistically typed	Dynamically Typed
Organized and consistent syntax	Simple, easy to read and write, and doesn't have many symbols
Static language	Dynamic language
Development is fast and offers better performance	Fast development but performance is a bit lacking to C#
A wide variety of application can be build with C# programming language	Enormous number of apps can be built with Python as well

Рисунок 2.4 – Порівняння c# і Python

2.4 Вибір операційної системи

У якості операційної системи було обрано Windows. Цей вибір мотивується тим, що за багатьма джерелами вона є найпоширенішою операційною системою серед усіх, у тому числі і в корпоративних мереж. Так, згідно ресурсу statista.com [12], близько 70 відсотків використовуваних операційних систем є саме Windows:

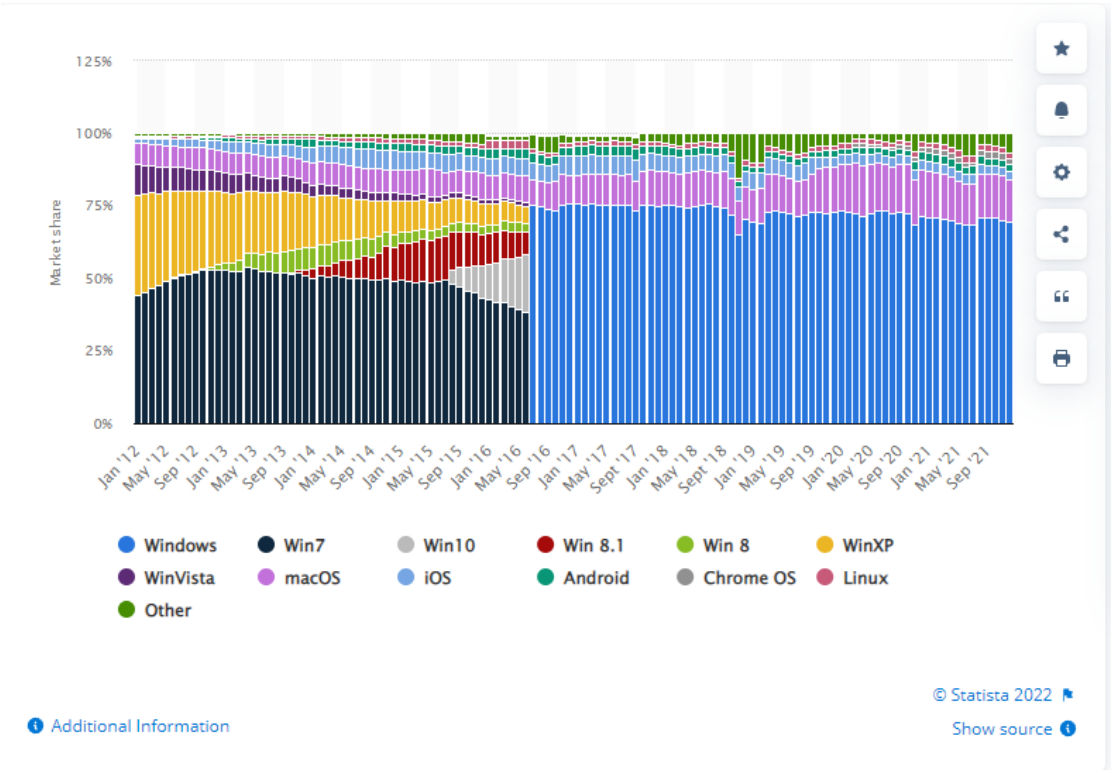


Рисунок 2.5 – Порівняльна таблиця розповсюдженості операційних систем

ОС Windows має широкий перелік переваг, які роблять її більш популярною ніж інші операційні системи:

1) Windows має великий перелік ПЗ, який може бути встановлений.

2) Windows є комерційним продуктом, а отже користувачі мають технічну підтримку.

3) Windows легко настроїти, і не надто складно адмініструвати.

4) Ця операційна система також має повну сумісність з майже усім устаткуванням.

Втім, слід зазначити, що вона має і певні недоліки:

1) Високі системні вимоги.

2) Проблеми з кібербезпекою.

Так, згідно виданню pcmag [17], у 1 кварталі 2020 року, близько 83.45 відсотків, від усіх зловмисних програм, були написані під Windows:

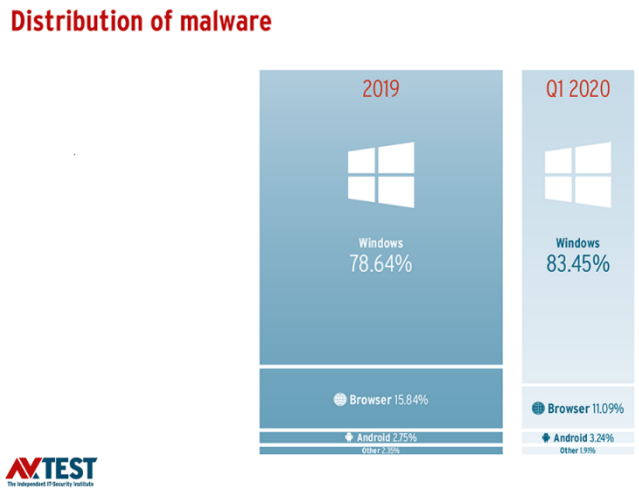


Рисунок 2.6 – Кількість зловмисного ПЗ у 1 кварталі 2020

Таким чином, можливо дійти висновку, що виявлення вторгнень в операційну систему Windows, враховуючи її розповсюдженість та проблеми з забезпеченням кібербезпеки, є одним з передових задач фахівців на підприємствах.

2.5 Створення правил кореляції та аналіз синтаксису Sumo Logic

Написання правил кореляції здійснюється за наступним алгоритмом:

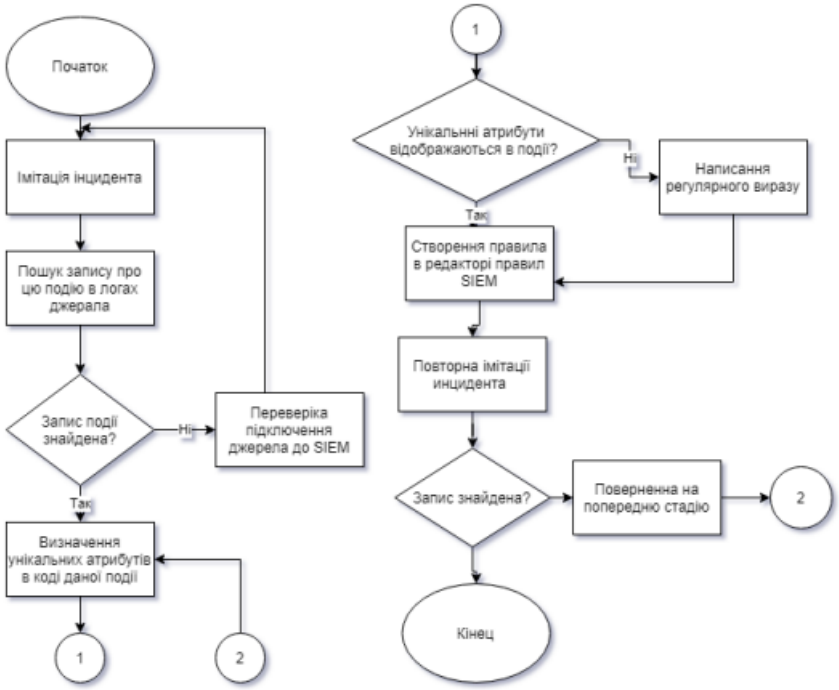


Рисунок 2.7 – Алгоритм написання правил кореляції

Втім, окремим важливим завданням під час написання правила може стати відсортовування хибно позитивних подій – тобто ті, що підійшли під логіку правила, але не є індикатором зловмисної активності. У такому разі, зазвичай проводиться аналіз такого роду події, визначення причини її появи та внесення змін у саме правило з метою уникнення такої ситуації у майбутньому.

Написання правил кореляції може здійснюватися двома методами:

1) Використовуючи вбудований конструктор правил.
2) Використовуючи синтаксис SIEM системи.

Перший метод вважається зручнішим, втім, як показує практика, багато SIEM систем не мають такої функції, як конструктор правил. Окрім того, використовуючи його можуть виникнути проблеми з написанням складних правил, або правил, що направленні на виявлення зловмисної активності базуючись на кастомних журналах подій.

Таким чином, використання синтаксису SIEM систем відбувається набагато частіше. Використовуючи його, фахівці з кібербезпеки пишуть вирази, що мають мету, використовуючи лог файли, знаходити зловмисні події, і таким чином виявляти вторгнення.

Розглянемо основні властивості такого синтаксису на прикладі Sumo Logic SIEM [15].

До операторів належать:

&& - логічне “і”;

|| - логічне “або”;

! – логічне “ні”;

/ - оператор поділу.

Окрім цього, він має усі математичні оператори, такі як додавання, порівнювання, тощо. Окремою перевагою CSE синтаксису є функції, такі як:

- Array_contains – перевіряє належність певного значення в обраному масиві;

- base64Decode – автоматично декодує строку з формату Base64;

- isNumeric – перевіряє, чи є обраний елемент числовим значенням.

Більш того, підтримуються умови у виразах, такі як “if”, “else”, тощо.

2.6 Категоризація загроз для написання правил, використовуючи MITRE Att&ck

Доволі часто, для більш ефективного написання правил, а також для кращої категоризації вже написаних правил необхідно об’єднати їх за якимось принципом. Одним із основних таких способів є категоризація за використовуваною нападниками техніки. Також, завдяки цьому методу, можливо зробити список технік та прийомів, що використовуються кожним окремим кіберзлочинним угрупуванням, і поступово писати правила виявлення під кожну з них. Тут на допомогу приходить Mitre Att&ck.

Mitre Att&ck (англ. Adversarial Tactics, Techniques & Common Knowledge) – спосіб класифікації дій нападників базуючись на реальних атаках. Платформа Att&ck має вигляд структурованого списку різних технік, що об’єднані між собою у тактики. Вона також має 3 різних напрями: Enterprise (техніки, які використовувалися проти корпоративних мереж і підприємств), Mobile (використовувалися під час атаки на мобільні пристрої) та Pre-Att&ck (описує дії зловмисника перед початком атаки).

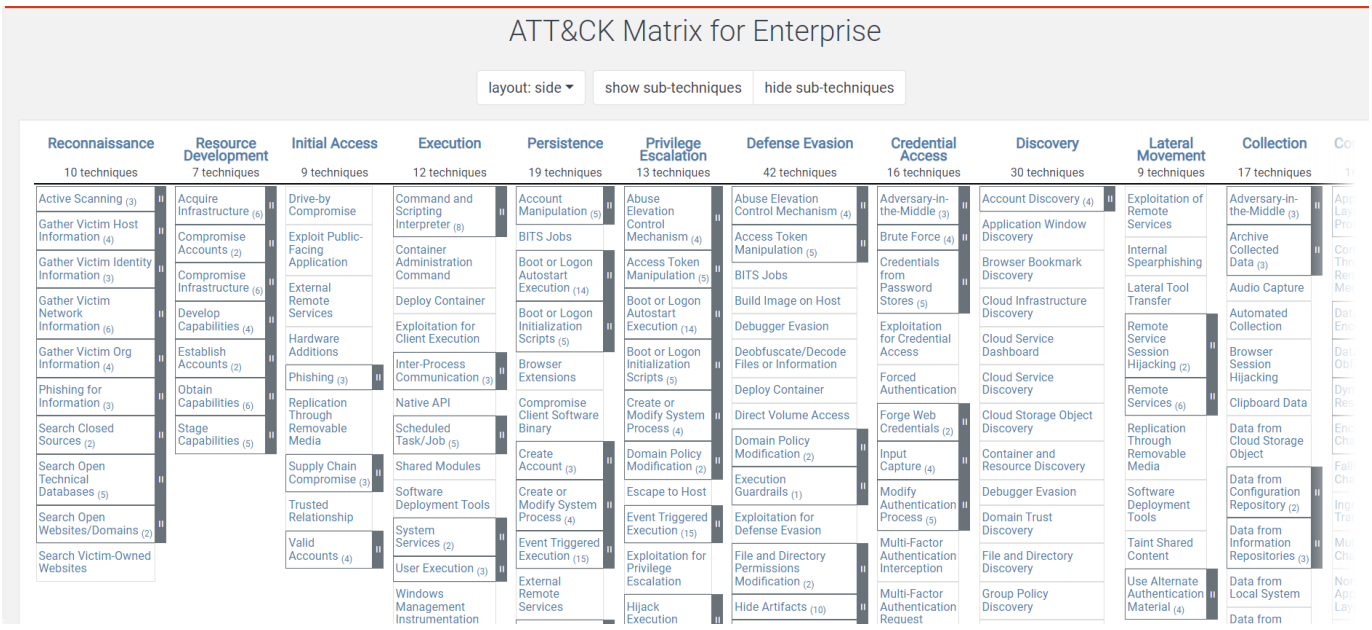


Рисунок 2.8 – Вигляд фреймворку Mitre Att&ck

Кожна техніка має свій ідентифікатор. Наприклад, візьмемо фішинг. У платформі Att&ck він має ідентифікатор T1566, відноситься до тактики “Первинний доступ” і має 3 підтехніки.

Окрім того, для кожної техніки доступна наступна інформація:

- Короткий опис.
- Приклади використання нападниками.
- Назви кримінальних угрупувань, що використовували її.
- Можливі шляхи попередження використання цієї техніки.
- Можливі шляхи виявлення експлуатації.
- Операційні системи та платформи, на яких ця техніка може використовуватися.

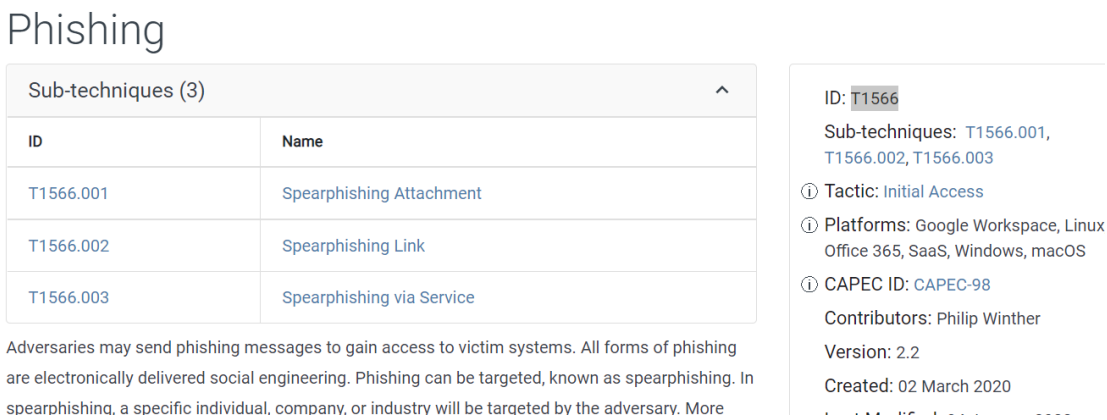


Рисунок 2.9 – Техніка “Phishing” у Mitre Att&ck

Окремої уваги заслуговує можливість відстежування списку технік, що використовує кожне окреме кіберзлочинне угрупування, адже це надає можливість надати перевагу написанню правил під зловмисні прийоми, що використовуються здебільшого проти сектора промисловості у якому працює компанія. Так, наприклад, якщо корпоративна мережа підприємства працює у банківській сфері, інтерес може представляти угрупування “APT-38”, що активне вже великий

проміжок часу (від 2014 року). Серед технік, які використовують ці злочинці, можливо побачити наступні:

- Перебір паролей.
- Відключення систем забезпечення інформаційної безпеки.
- Збір інформації про скомпрометовану систему.
- Зміни в реєстрі тощо.

Висновки за розділом 2

У розділі 2 було проаналізовано критерії вибору SIEM систем, а також порівняно різні доступні SIEM системи, після чого було обрано потрібну для реалізації виявлення вторгнення. Також, було обрано мову програмування Python для реалізації програмної частини та проведено порівняння з іншою мовою програмування з метою демонстрації її переваг. Після цього було обрано операційну систему Windows для виконання практичної частини. Також, було розглянуто алгоритм для написання правил кореляції для виявлення вторгнення у корпоративні мережі, а також синтаксис у Sumo Logic та фреймворк для категоризації загроз Mitre Att&ck.

РОЗДІЛ 3
РЕАЛІЗАЦІЯ ВИЯВЛЕННЯ ВТОРГНЕННЯ

3.1 Створення програми збору подій

Задля реалізації системи логування було використану мову python. Принцип роботи програми наведений на наступному рисунку:

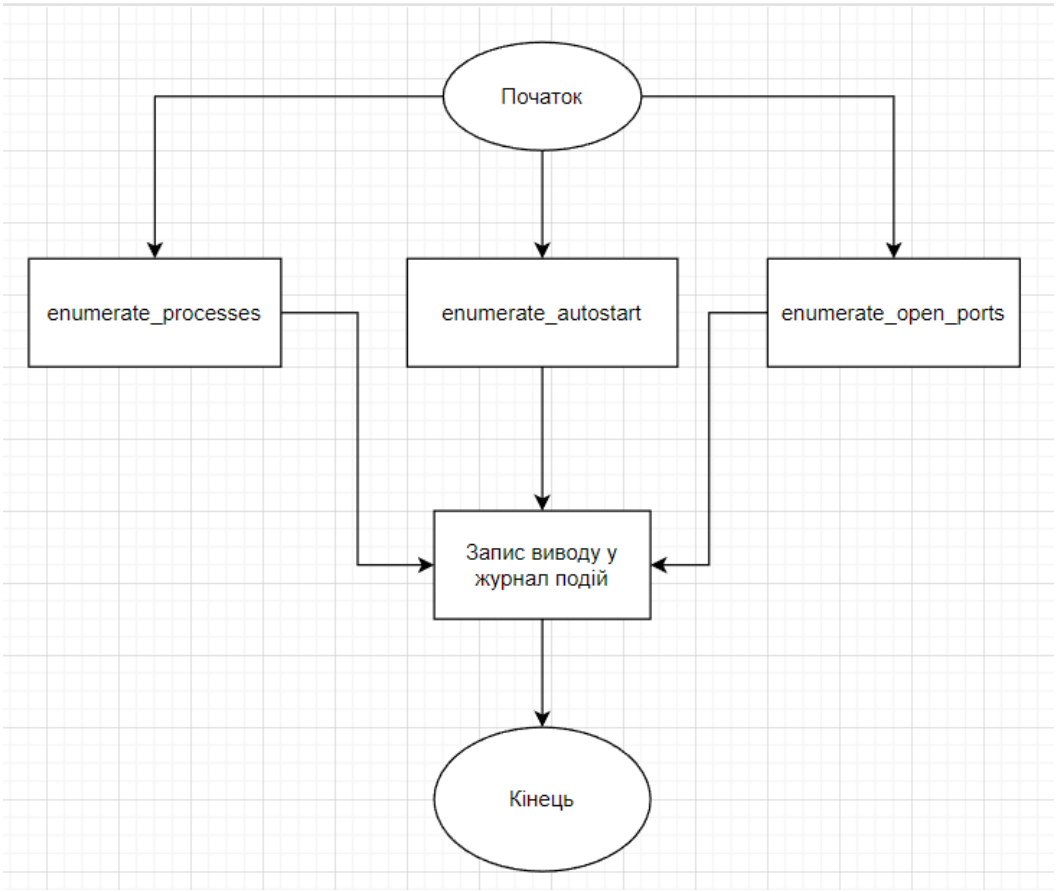


Рисунок 3.1 – Алгоритм роботи програми

Програма має 3 функції:

1) enumerate_processes – функція, що має мету збирати інформацію про запущені процеси. Для збору інформації використовується бібліоткека python psutil. Збирається наступна інформація:

- назва процесу: ProcName;

- PID (ідентифікатор процесу): PID;

- назва машини: Hostname;

- ім’я користувача: User;

- IP адреса: IpAddr;

- поточну дату: CurrentDate;

- шлях процесу в системі і аргументи командної строки: ProcCMD;

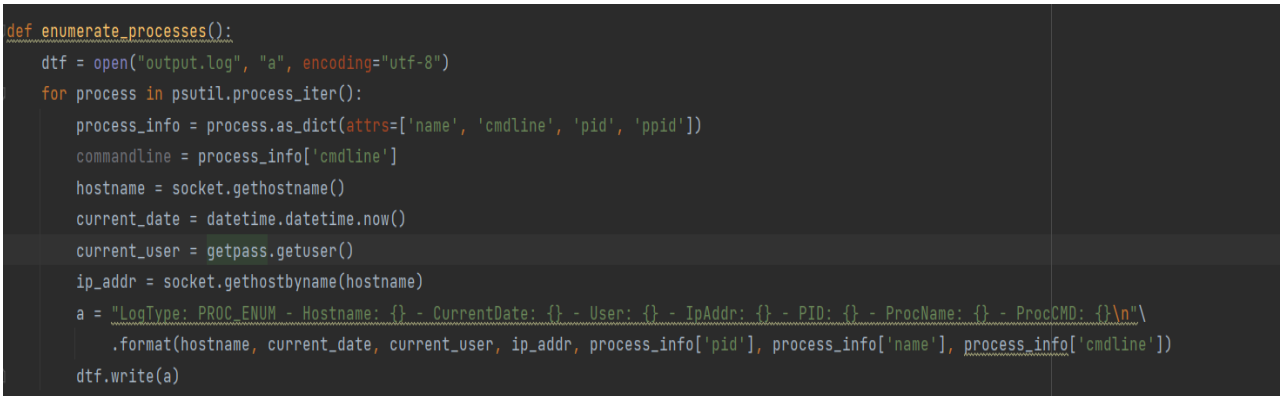
- тип логу: LogType.

Таким чином, приклад події виглядає наступним чином:

“LogType: PROC_ENUM – Hostname: DESKTOP-SNL26NF – CurrentDate: 2022-05-29 06:24:40.045347 – User: Nik – IpAddr: 192.168.157.1 – PID: 356 – ProcName: WmiPrvSE.exe – ProcCMD: None”

З цієї події можливо зрозуміти, що процес з назвою WmiPrvSE.exe був запущений з ідентифікатором 356, на вузлі “DESKTOP-SNL26NF” з поточним користувачем “Nik”, який має IP адресу 192.168.157.1 без аргументів командної строки.

Код функції виглядає наступним чином:



```
def enumerate_processes():
    dtf = open("output.log", "a", encoding="utf-8")
    for process in psutil.process_iter():
        process_info = process.as_dict(attrs=['name', 'cmdline', 'pid', 'ppid'])
        commandline = process_info['cmdline']
        hostname = socket.gethostname()
        current_date = datetime.datetime.now()
        current_user = getpass.getuser()
        ip_addr = socket.gethostbyname(hostname)
        a = "LogType: PROC_ENUM - Hostname: {} - CurrentDate: {} - User: {} - IpAddr: {} - PID: {} - ProcName: {} - ProcCMD: {}\n"\
            .format(hostname, current_date, current_user, ip_addr, process_info['pid'], process_info['name'], process_info['cmdline'])
        dtf.write(a)
```

Рисунок 3.2 – Код функції enumerate_processes

2) enumerate_autostart – це функція, що перевіряє деякі ключи реєстру(а саме SOFTWARE\Microsoft\Windows\CurrentVersion\Run у просторі HKEY-_CURRENT_USER та HKEY_LOCAL_MACHINE) з метою генерування списку програм, що були додані на автозапуск під час кожного запуску роботи вузла.

Під час роботи програми збирається наступна інформація:

- назва машини: Hostname;

- ім’я користувача: User;

- IP адреса: IpAddr;

- поточну дату: CurrentDate;

- назва процесу: ProcName;

- шлях процесу і аргументи командної строки: ProcPath.

Приклад події виглядає наступним чином:

“LogType: AUTOSTART_ENUM – Hostname: DESKTOP-SNL26NF – CurrentDate: 2022-05-30 15:07:53.689591 – User: Nik – IpAddr: 192.168.157.1 – ProcName: Discord – ProcPath: C:\Users\Nik\AppData\Local\Discord\Update.exe –processStart Discord.exe”

З цієї події ми можемо дізнатися, що станом на 15:07:37, 30 травня 2022, користувач Nik на робочій машині DESKTOP-SNL26NF, з IP адресою 192.168.157.1 мав в автозапуску програму “Discord” з наступним шляхом і командними аргументами: C:\Users\Nik\AppData\Local\Discord\Update.exe –processStart Discord.exe

Код функції виглядає наступним чином:

```
def enumerate_autostart():
    dtf = open("output.log", "a", encoding="utf-8")
    access_registry = winreg.ConnectRegistry(None,winreg.HKEY_LOCAL_MACHINE)
    access_key = winreg.OpenKey(access_registry, r"SOFTWARE\Microsoft\Windows\CurrentVersion\Run")
    hostname = socket.gethostname()
    current_date = datetime.datetime.now()
    current_user = getpass.getuser()
    ip_addr = socket.gethostbyname(hostname)
    for n in range(100):
        try:
            machine_value = winreg.EnumValue(access_key, n)
        except:
            break
        a = "LogType: AUTOSTART_ENUM - Hostname: {} - CurrentDate: {} - User: {} - IpAddr: {} - ProcName: {} - ProcPath: {}\n"\
            .format(hostname, current_date, current_user, ip_addr, machine_value[0], machine_value[1])
        dtf.write(a)
    access_registry = winreg.ConnectRegistry(None, winreg.HKEY_CURRENT_USER)
    access_key = winreg.OpenKey(access_registry, r"SOFTWARE\Microsoft\Windows\CurrentVersion\Run")
    for n in range(100):
        try:
            machine_value = winreg.EnumValue(access_key, n)
        except:
            break
        a = "LogType: AUTOSTART_ENUM - Hostname: {} - CurrentDate: {} - User: {} - IpAddr: {} - ProcName: {} - ProcPath: {}\n"\
            .format(hostname, current_date, current_user, ip_addr, machine_value[0], machine_value[1])
        dtf.write(a)
```

Рисунок 3.3 – Код функції enumerate_autostart

3) enumerate_open_ports – перевіряє відкриті порти на машині, на який запущена. Збирає наступну інформацію:

- назва машини: Hostname;
- ім’я користувача: User;
- IP адреса: IpAddr;
- поточну дату: CurrentDate;
- назва процесу: ProcName;
- номер відкритого порту: OpenedPort.

Слід зауважити, що функція може потребувати деякий час для завершення роботи.

```
def enumerate_open_ports():
    dtf = open("output.log", "a", encoding="utf-8")
    hostname = socket.gethostname()
    current_date = datetime.datetime.now()
    current_user = getpass.getuser()
    ip_addr = socket.gethostbyname(hostname)
    for port in range(1, 65535):
        try:
            sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
            socket.setdefaulttimeout(1)
            result = sock.connect_ex(('127.0.0.1', port))
            if (result == 0):
                a = "LogType: PORT_ENUM - Hostname: {} - CurrentDate: {} - User: {} - IpAddr: {} - OpenedPort: {}"\
                    .format(hostname, current_date, current_user, ip_addr, port)
                dtf.write(a)
                sock.close()
        except socket.error:
            pass

enumerate_autostart()
```

Рисунок 3.4 – Код функції enumerate_open_ports

Повний лістинг коду наведено у додатку А.

Результат роботи програми виводиться у файл “output.log”. Шлях до цього файлу задається у самій програмі, і може бути змінений у будь-який час. В залежності від кількості процесів, програм в автозапуску а також відкритих портів, кількість записів за один повний цикл роботи програми може налічувати від 100 унікальних записів у файл.

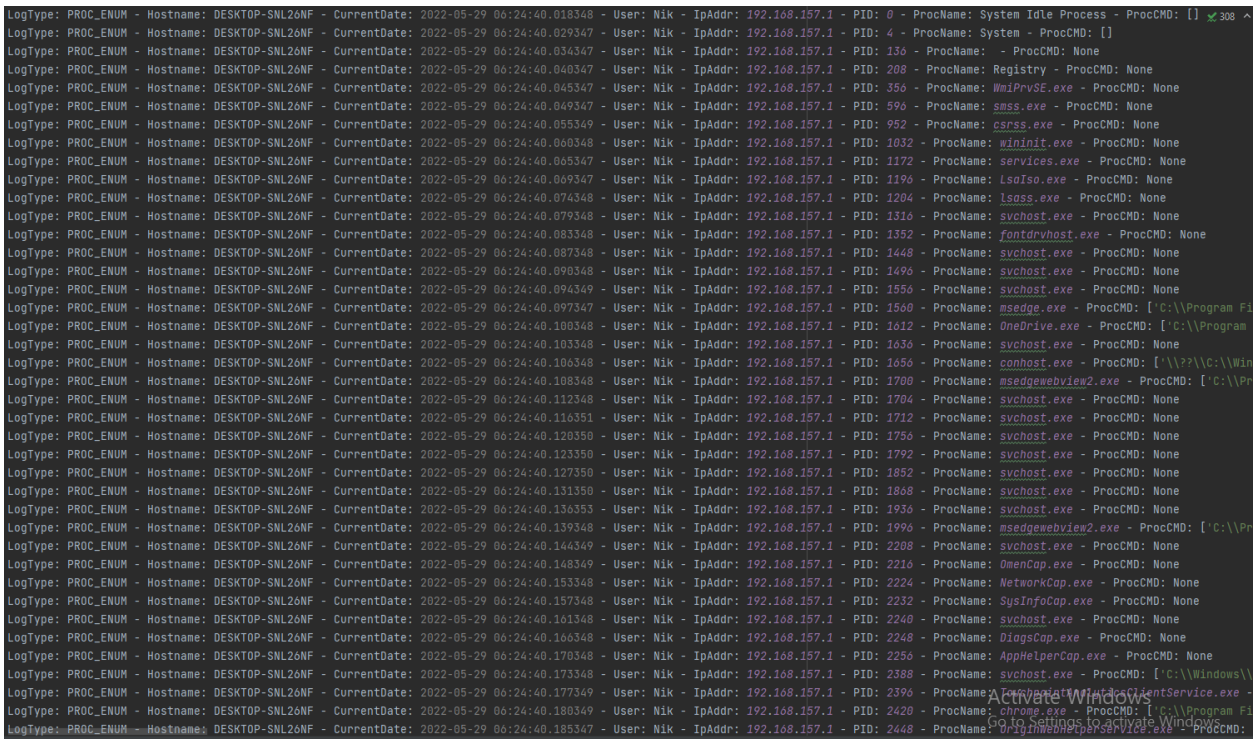


Рисунок 3.5 – Приклад створеного журналу подій

Слід зазначити, що використовуваний формат словника, тобто, використовується формат “Ключ: значення – Ключ: значення”. Це було зроблено для того, щоб уникнути можливих проблем з розпізнаванням подій на самій SIEM-системі. Окрім того, згідно офіційної рекомендації SUMO Logic, такий формат є кращим для реалізації власного журналу подій. Повний лістинг програмного коду наведено у додатку А.

Для того, щоб налаштувати виконання програми на певний проміжок часу, необхідно скористуватися утілітою Windows schtasks.exe. Прикладом використання цієї програми для налаштування програми на виконання кожну годину може стати наступна команда:

schtasks /Create /SC HOURLY /TN PythonTask /TR “Шлях до програми”, де:

/Create – створює нове завдання для виконання

/SC HOURLY – налаштовує виконання на 1 раз на годину

/TN -назва створеного завдання

/TR – шлях до програми

```
C:\Users\Nik\Pictures>schtasks /Create /SC HOURLY /TN PythonTask /TR logcollector.exe
SUCCESS: The scheduled task "PythonTask" has successfully been created.
```

Рисунок 3.6 – Успішне виконання schtasks.exe

3.2 Передання журналу подій у SUMO Logic SIEM

Для того, щоб додати файл з подіями до SUMO Logic Siem, можливо скористуватися двома методами:

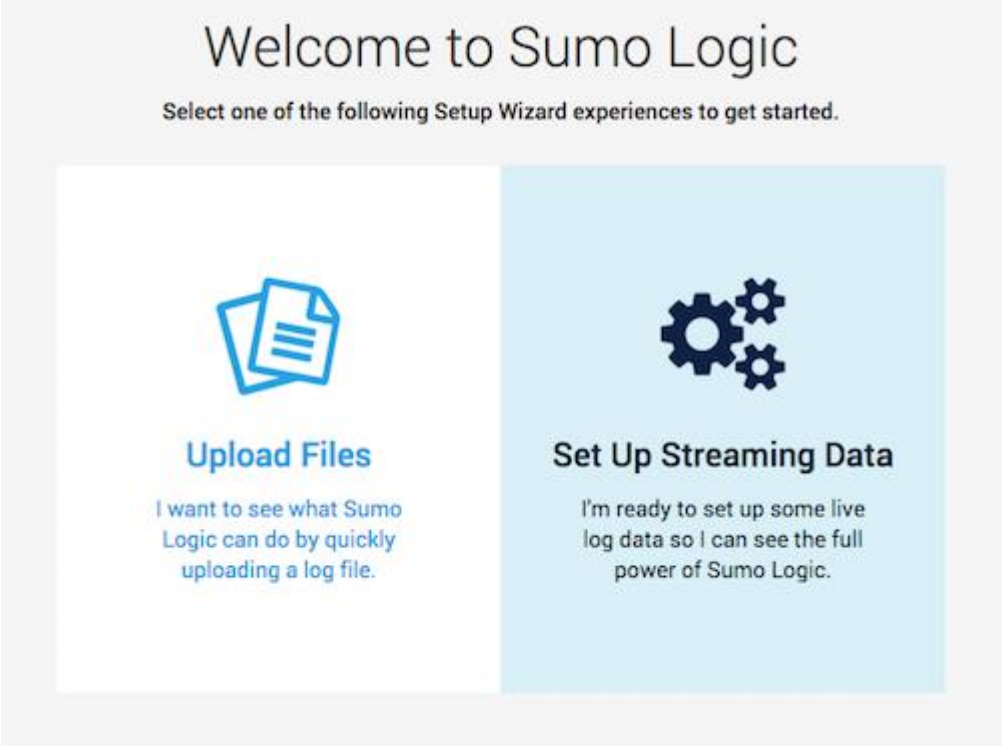


Рисунок 3.7 – Методи завантаження журналів подій SUMO Logic

1) Завантажити лог файл використовуючи інтерфейс SUMO. Такий підхід часто використовується фахівцями з кібербезпеки для простого перегляду подій, втім, він має важливий недолік – кожний раз завантажувати файл подій у сумо є не тільки не зручно і вимагає багату часу, а й не є ефективним. Більш того, це може призвести до плутанини у багатьох назвах журналів подій.

2) Налаштувати потокових даних – не такий простий метод у виконанні, втім, має певну кількість переваг, у тому числі можливість автоматично збирати вказані

журнали подій прямо з вузлів корпоративної мережі, можливість додаткового до налаштування обробки самих подій, тощо.

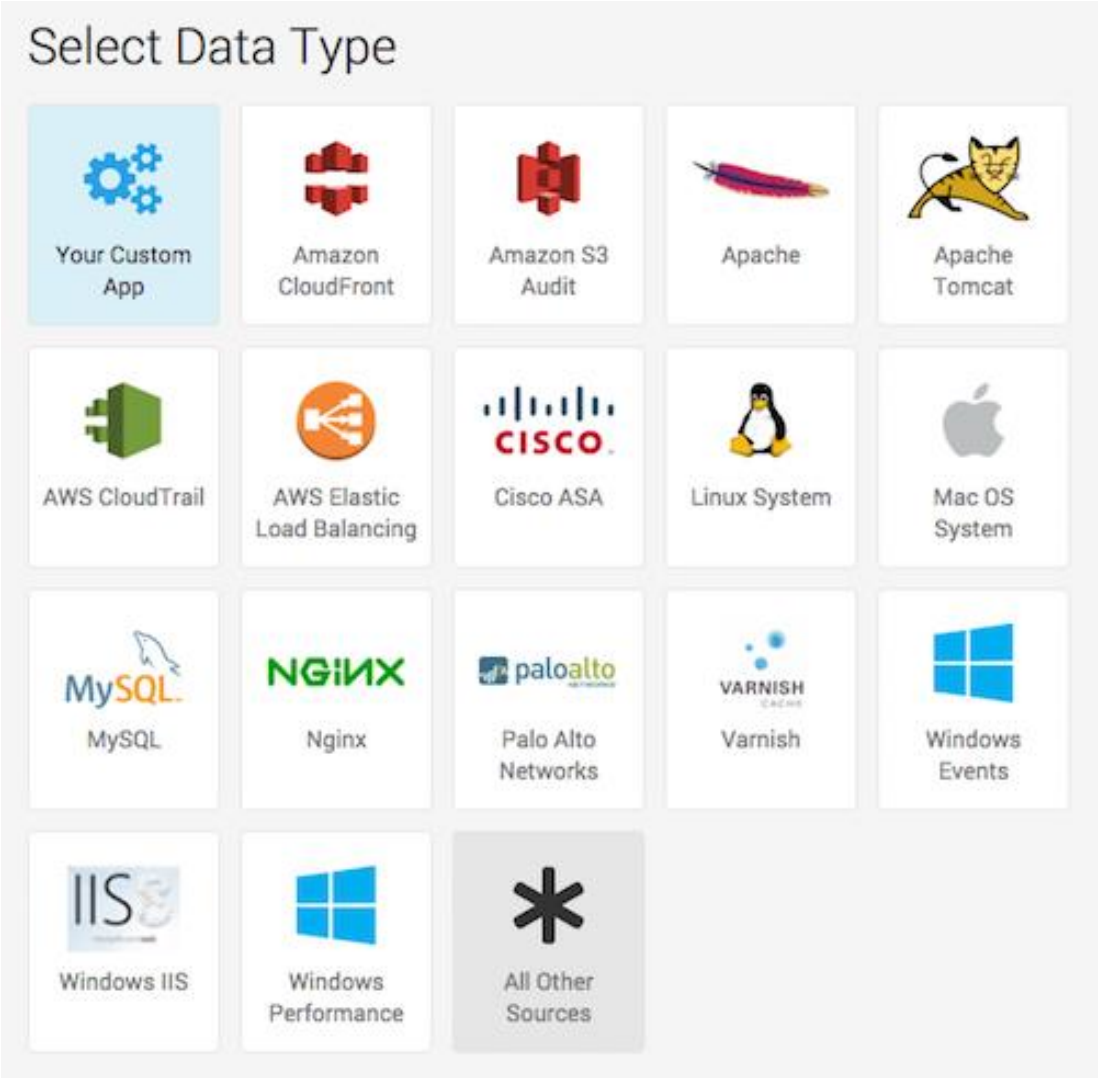


Рисунок 3.8 – Вибір типу логу у SUMO Logic

Для проведення роботи мною було обрано перший варіант.

Таким чином, після завантаження журналу подій, слід запам’ятати назву файлу, який був вказаний під час налаштування. Після цього, необхідно перейти до вкладки “Search”, де вказати назву у форматі “_source=*НАЗВА ЛОГ ФАЙЛУ*”. Таким чином, маючи назву файлу “testcustomwinact.log”, було отримано наступний результат:

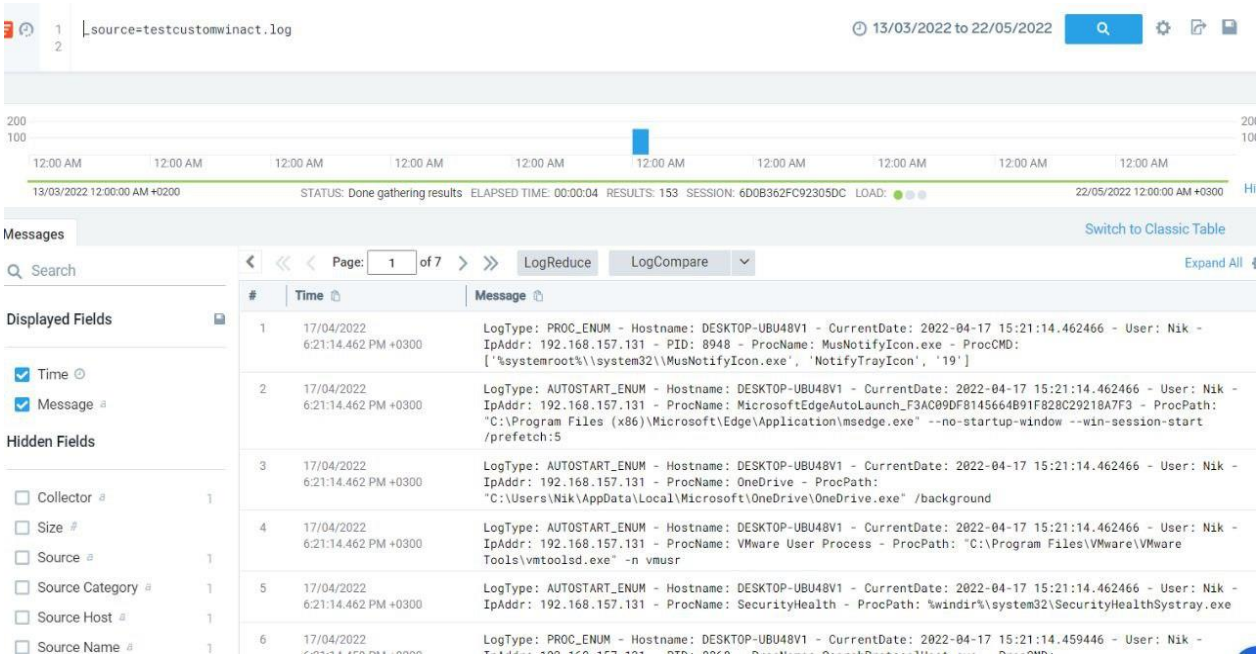


Рисунок 3.9 – Загальний вигляд інтерфейсу пошуку

На рисунку відображені події за певний вказаний проміжок часу. Тепер, в залежності від обраної категорії події, ми можемо відобразити лише вибрані події використовуючи наступний вираз:

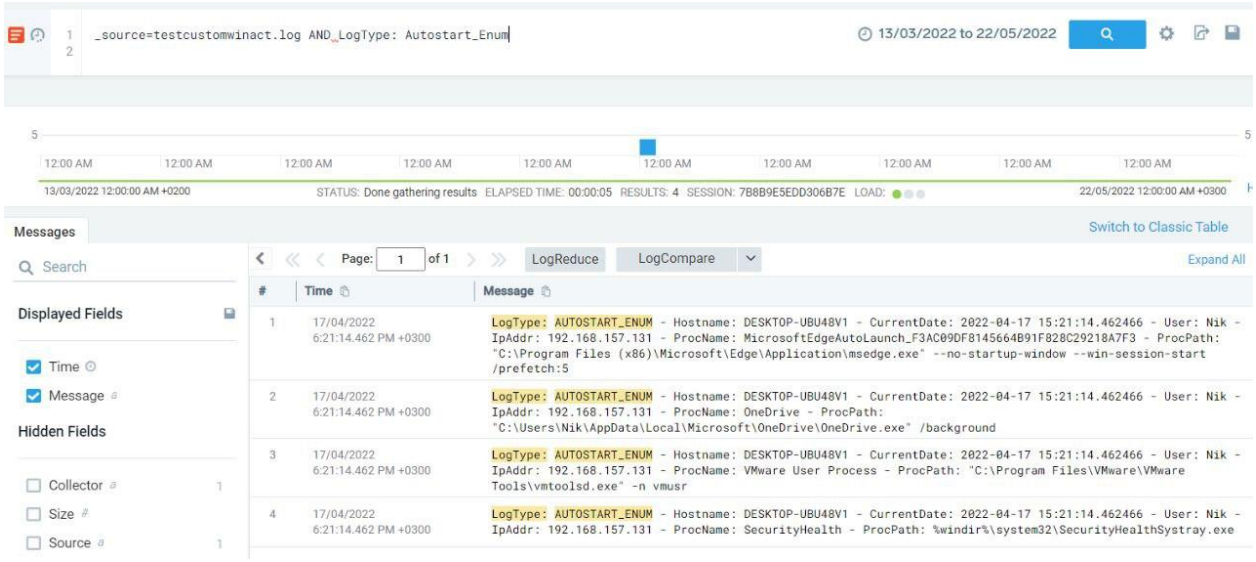


Рисунок 3.10 – Пошук за типом подій “Autostart_Enum”

Таким чином, використовуючи отримані події, ми можемо написати правила, використовуючи інтерфейс та синтаксис Sumo Logic SIEM.

3.3 Написання правил для типу подій “PROC_ENUM”

Для написання правил, використовуючи тип логу PROC_ENUM було обрано одну з технік mitre att&ck – маскування (id – T1036). Суть маскування полягає в тому, що нападники можуть зробити їх зловмисний файл схожим на легітимний, для того щоб ввести в оману фахівців з кібербезпеки або не визвати підозри у продуктів інформаційної безпеки. Було реалізовано набір правил, що перевіряють на правильність системні програми, адже вони доволі часто використовуються преступниками.

Зазвичай, основною папкою для запуску системних програм є або System32, або SysWOW64. Запуск з будь-якої іншої папки є дуже підозрілим і вимагає подальшого дослідження.

Таким чином, правило під svchost.exe (використовується для служб, які завантажуються з динамічних бібліотек) виглядає наступним чином:
“_source=testcustomwinact.log AND LogType: PROC_ENUM AND ProcName: svchost.exe AND !(ProcCMD: *system32* OR ProcCMD: None)”.

Перевіримо на хибно позитивні події, використовуючи SUMO Logic та наш журнал подій:

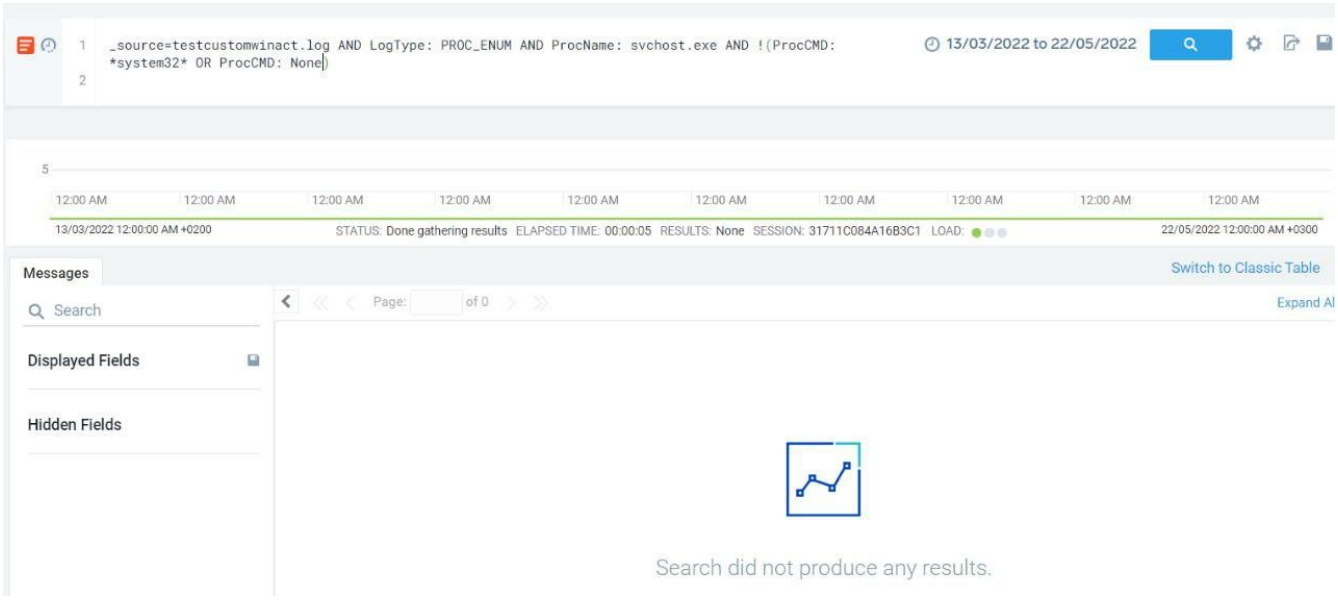


Рисунок 3.11 – Результат пошуку за виразом

Як можна побачити, вираз не повернув ніяких результатів, таким чином ми можемо бути впевнені, що правило працює, а зловмисної активності такого роду на системі на відбувається. Втім, результат роботи виразу на імітованих зловмисних подіях наведено у додатку Б.

Напишемо подібне правило для деяких інших системних програм:

“_source=testcustomwinact.log AND LogType: PROC_ENUM AND ProcName: conhost.exe AND !(ProcCMD: *system32* OR ProcCMD: None)”.

“_source=testcustomwinact.log AND LogType: PROC_ENUM AND ProcName: explorer.exe AND !(ProcCMD: *system32* OR ProcCMD: None)”.

“_source=testcustomwinact.log AND LogType: PROC_ENUM AND ProcName: ctfmon.exe AND !(ProcCMD: *system32* OR ProcCMD: None)”.

“_source=testcustomwinact.log AND LogType: PROC_ENUM AND ProcName: cmd.exe AND !(ProcCMD: *system32* OR ProcCMD: None)”.

“_source=testcustomwinact.log AND LogType: PROC_ENUM AND ProcName: powershell.exe AND !(ProcCMD: *system32* OR ProcCMD: None)”.

“_source=testcustomwinact.log AND LogType: PROC_ENUM AND ProcName: regsvr.exe AND !(ProcCMD: *system32* OR ProcCMD: None)”.

“_source=testcustomwinact.log AND LogType: PROC_ENUM AND ProcName: reg.exe AND !(ProcCMD: *system32* OR ProcCMD: None)”.

3.4 Написання правил для типу подій “AUTOSTART_ENUM”

Для написання правил, використовуючи тип логу AUTOSTART_ENUM було виявлено, що додавання деяких програм в автозапуск, або передання підозрілих аргументів в командну строку таких програм може бути ознакою зловмисної активності. Так, наприклад, запуск файлу з розширенням .dll використовуючи rundll32.exe є дуже підозрілим, і це може означати те, що кіберзлочинці таким чином закріплюються в системі.

Вираз для виявлення такої активності виглядає наступним чином:

“_source=testcustomwinact.log AND LogType: AUTOSTART_ENUM AND ProcPath: "rundll32.exe" AND ProcPath: ".dll"”

Після запуску виразу на SUMO Logic, можемо побачити, що ніяких результатів повернено не було:

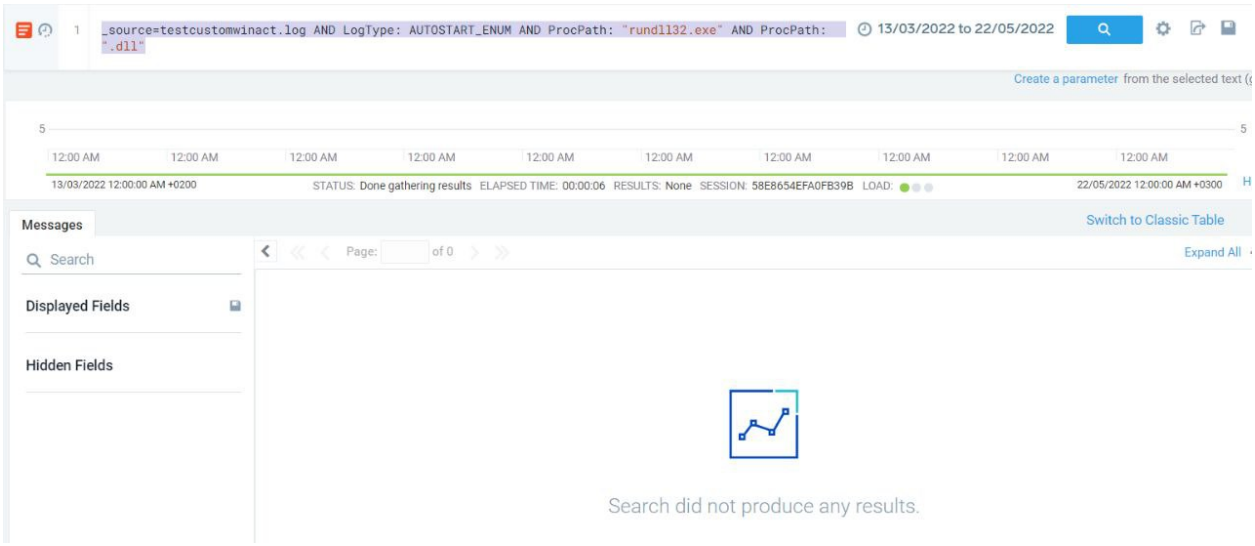


Рисунок 3.12 – Результат пошуку за виразом

Ще одним подібним підозрілим признаком є запуск скрипта, написаного на мові програмування python, використовуючи стандартну програму python.exe. Таке правило буде виглядати наступним чином:

“_source=testcustomwinact.log AND LogType: AUTOSTART_ENUM AND ProcPath: *python* AND ProcPath: ".py"”

Під час перевірки на SUMO Logic SIEM результату повернено не було:

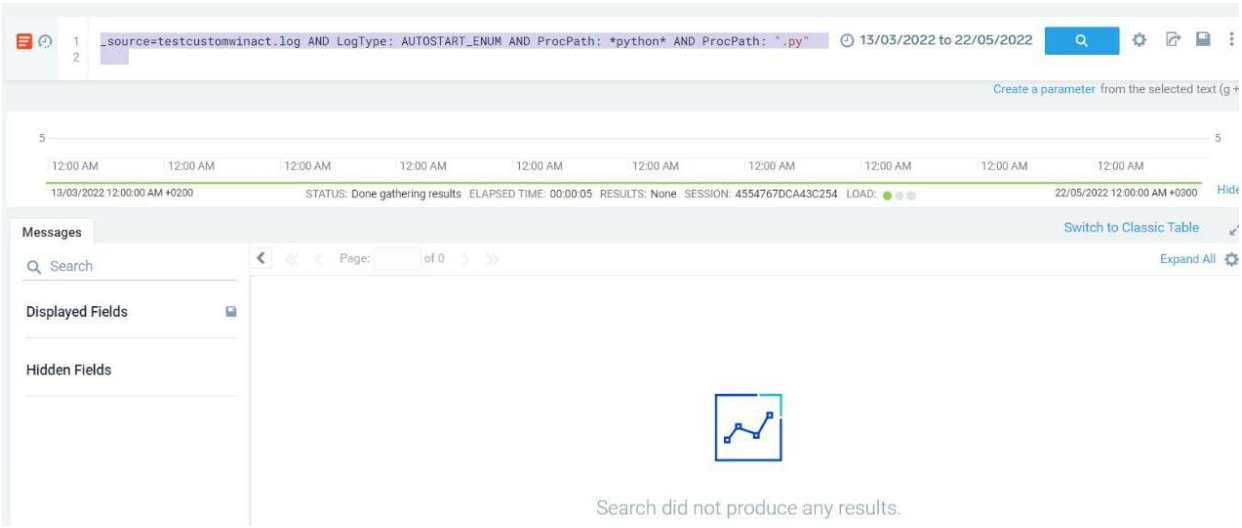


Рисунок 3.13 – Результат пошуку за виразом

3.5 Написання правил для типу подій “PORT_ENUM”

Для того, щоб написати правила, використовуючи тип логу PORT_ENUM, було розглянуто два сценарії:

1) Був відкритий специфічний порт, що не використовується на підприємстві. Наприклад, деякі організації використовують протокол RDP (Remote Desktop Protocol) задля віддаленого підключення до робочих станцій або серверів. Стандартним портом для нього є 3389, але теоретично можливий сценарій, де адміністратори підприємства домовляться про використання іншого, наприклад 3390. Отже, використання стандартного порту таким чином є підозрілою активністю, адже це може свідчити або про те, що співробітники віддалено підключаються до робочої станції в обхід стандартної процедури, або про компрометацію робочого місця і подальшого використовування її через RDP.

2) Був виявлений відкритий порт, що часто використовується певним видом зловмисного ПЗ. Було обрано саме такий сценарій, бо перший стосується окремих випадків корпоративних мереж.

Наприклад, зловмисне ПЗ російського походження “Trickbot” часто використовувало порт 8082 для комунікації з сервером нападників. Таким чином, вираз для виявлення подібної активності виглядатиме наступним чином:

_source=testcustomwinact.log AND LogType: PORT_ENUM AND OpenedPort: 8082

Після перевірки, знайдено результатів не було, таким чином, зловмисної активності виявлено не було:

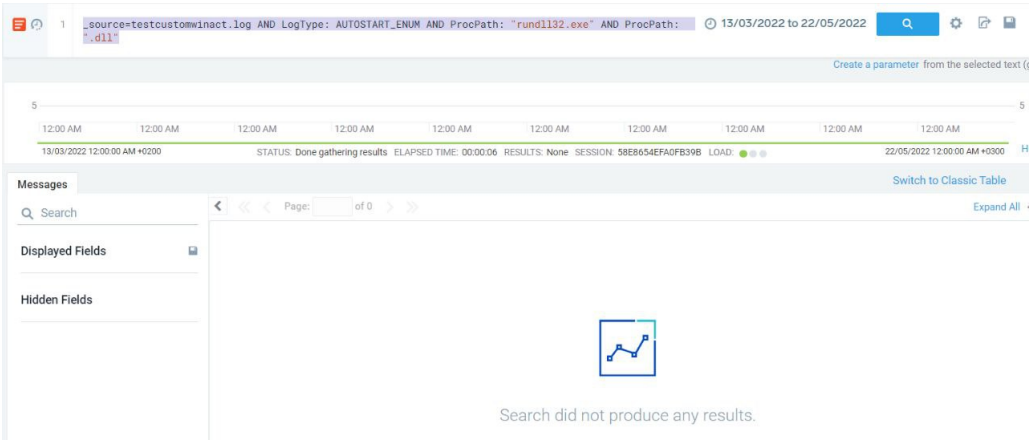


Рисунок 3.14 – Результат пошуку за виразом

Висновки за розділом 3

Під час виконання завдання, поставленого на дипломну роботу було створено програму, яка збирає певну інформацію з операційної системи і формує журнал подій. Було проведено детальний розбір кожної функції, що входить у склад програми, наведено особливості виводу кожної з них. Також, було проаналізовано можливості формування журналу подій, досліджено можливі шляхи передачі сформованих подій в SUMO Logic SIEM. Після цього, було написано ряд правил, що дозволять на практиці виявити вторгнення у корпоративні мережі.

ВИСНОВКИ

На сьогоднішній день, враховуючи потенціал нападників та розповсюдженість компрометації різних підприємств, своєчасне виявлення вторгнення є критичним для забезпечення безпеки корпоративних мереж. У першому розділі було розглянуто основні технології виявлення вторгнення на підприємствах, до їх числа належать NIDS, HIDS, Honeypot та SIEM системи. Після розгляду дійшов висновку, що саме SIEM системи мають найбільший потенціал для виявлення вторгнення, після чого було поставлено задачі для виконання.

У другому розділі було проаналізовано критерії вибору SIEM системи, на основі яких було обрано систему під назвою “Sumo Logic”. Окрім цього, було обрано мову програмування для реалізації програмної частини, а також операційну систему для збору подій. Після цього, було створено алгоритм для написання правил кореляції, розглянутий синтаксис написання правил у Sumo Logic, а також досліджено категоризацію загроз використовуючи фреймворк Mitre Att&ck.

У третьому розділі було створено програму для збору подій та створення лог файлу, розкрито її функціонал та методи передачі подій на Sumo Logic. Після цього, було написано ряд правил кореляції, спрямований на виявлення зловмисних дій нападників на вузлах корпоративної мережі. Було продемонстровано результат роботи таких виразів.

Таким чином, можна дійти висновку, що поставлені у першому розділі завдання були виконані, а мета дипломної роботи – досягнута.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Duke CFO Global Business Outlook. More than 80 percent of firms say they have been hacked – Режим доступу: https://cfosurvey.fuqua.duke.edu/press-release/more-than-80-percent-of-firms-say-they-have-been-hacked/

2. WeLiveSecurity ESET Threat Report T3 2021 – Режим доступу: https://www.welivesecurity.com/2022/02/09/eset-threat-report-t32021/

3. Корпоративна мережа [Електронний ресурс]. – Режим доступу: http://xn--r1a3b.xn--b1amgblet.xn--j1amh/index.php/%D0%9A%D0%BE%D1%80%D0%BF%D0%BE%D1%80%D0%B0%D1%82%D0%B8%D0%B2%D0%BD%D0%B0_%D0%BC%D0%B5%D1%80%D0%B5%D0%B6%D0%B0

4. Мєшков В. І., Віролайнен В. О. Аналіз сучасних систем виявлення та запобігання вторгнень в інформаційно-телекомунікаційних системах [Електронний ресурс] / В. І. Мєшков, В. О. Віролайнен // Поблеми безпеки інформації в інформаційно-комунікаційних системах, Д.: НТУУ КПІ РТФ, 2015. С. 4. Режим доступу: https://ela.kpi.ua/bitstream/123456789/17609/1/meshkov.pdf

5. Network Based Intrusion Detection System. ScienceDirect. [Електронний ресурс]. – Режим доступу: https://www.sciencedirect.com/topics/computer-science/network-based-intrusion-detection-system

6. 8 Best HIDS Tools—Host-Based Intrusion Detection Systems. DNSstuff. [Електронний ресурс]. – Режим доступу: https://www.dnsstuff.com/host-based-intrusion-detection-systems

7. Titarmare N., Hargule N., Gupta A. An Overview of Honeypot Systems. An Overview of Honeypot Systems. International Journal of Computer Sciences and Engineering. 2019, 7. 394-397. 10.26438/ijcse/v7i2.394397. [Електронний ресурс]. – Режим доступу: https://www.researchgate.net/publication/332113726_An_Overview_of_Honeypot_Systems

8. Sumo Logic Cloud SIEM Ratings Overview. Gartner Peer Insights. [Електронний ресурс]. – Режим доступу: https://www.gartner.com/reviews/market/security-information-event-management/vendor/sumo-logic/product/sumo-logic-cloud-siem?marketSeoName=security-information-event-management&vendorSeoName=sumo-logic&productSeoName=sumo-logic-cloud-siem

9. Sumo Logic. Datashield. [Електронний ресурс]. – Режим доступу: https://www.datashieldprotect.com/tools/sumo-logic

10. The 10 Most Popular Programming Languages to Learn in 2022. Northeastern University Graduate Programs. [Електронний ресурс]. – Режим доступу: https://www.northeastern.edu/graduate/blog/most-popular-programming-languages/

11. Python Programming Language: Step by Step Guide 2022. [Електронний ресурс]. – Режим доступу: https://hackr.io/blog/python-programming-language

12. Global market share held by computer operating systems 2012-2021. Statista. [Електронний ресурс]. –
Режим доступу: https://www.statista.com/statistics/268237/global-market-share-held-by-operating-systems-since-2009/

13. Крижановський В. Г., Сергієнко С. П. Апаратно-програмні засоби захисту інформації у корпораціях: навчально-методичний посібник. / В. Г. Крижановський, С. П. Сергієнко // Вінниця : ДонНУ імені Василя Стуса, 2019. – 36 с.

14. Макаров А. М. Комплексна система захисту інформації в локальній мережі. Моніторинг / А. М. Макаров; Сумський державний університет. – 2020. – 41 с.

15. CSE Rules Syntax. Sumo Logic. [Електронний ресурс]. – Режим доступу: https://help.sumologic.com/Cloud_SIEM_Enterprise/CSE_Rules/11_CSE_Rules_Syntax

16. C# vs Python: Choosing the Right Language For Your Project. LITSLINK. [Електронний ресурс]. – Режим доступу: https://litslink.com/blog/csharp-vs-python-choosing-right-language-for-your-project

ДОДАТОК А
ЛІСТІНГ ПРОГРАМИ ДЛЯ ЗБОРУ ПОДІЙ ОПЕРАЦІЙНОЇ СИСТЕМИ

```
import psutil
import socket
import datetime
import getpass
import winreg

def enumerate_processes():
    dtf = open("output.log", "a", encoding="utf-8")
    for process in psutil.process_iter():
        process_info = process.as_dict(attrs=['name', 'cmdline', 'pid', 'ppid'])
        commandline = process_info['cmdline']
        hostname = socket.gethostname()
        current_date = datetime.datetime.now()
        current_user = getpass.getuser()
        ip_addr = socket.gethostbyname(hostname)
        a = "LogType: PROC_ENUM - Hostname: {} - CurrentDate: {} - User: {} - IpAddr:
{} - PID: {} - ProcName: {} - ProcCMD: {}\n"\
            .format(hostname, current_date, current_user, ip_addr, process_info['pid'],
process_info['name'], process_info['cmdline'])
        dtf.write(a)

def enumerate_autostart():
    dtf = open("output.log", "a", encoding="utf-8")
    access_registry = winreg.ConnectRegistry(None,winreg.HKEY_LOCAL_MACHINE)
    access_key = winreg.OpenKey(access_registry,
r"SOFTWARE\Microsoft\Windows\CurrentVersion\Run")
```

```
    hostname = socket.gethostname()
    current_date = datetime.datetime.now()
    current_user = getpass.getuser()
    ip_addr = socket.gethostbyname(hostname)
    for n in range(100):
        try:
            machine_value = winreg.EnumValue(access_key, n)
        except:
            break
        a = "LogType: AUTOSTART_ENUM - Hostname: {} - CurrentDate: {} - User: {} -
IpAddr: {} - ProcName: {} - ProcPath: {}\n"\
            .format(hostname, current_date, current_user, ip_addr, machine_value[0],
machine_value[1])
        dtf.write(a)
    access_registry = winreg.ConnectRegistry(None, winreg.HKEY_CURRENT_USER)
    access_key = winreg.OpenKey(access_registry,
r"SOFTWARE\Microsoft\Windows\CurrentVersion\Run")
    for n in range(100):
        try:
            machine_value = winreg.EnumValue(access_key, n)
        except:
            break
        a = "LogType: AUTOSTART_ENUM - Hostname: {} - CurrentDate: {} - User: {} -
IpAddr: {} - ProcName: {} - ProcPath: {}\n"\
            .format(hostname, current_date, current_user, ip_addr, machine_value[0],
machine_value[1])
        dtf.write(a)

def enumerate_open_ports():
    dtf = open("output.log", "a", encoding="utf-8")
```

```
    hostname = socket.gethostname()
    current_date = datetime.datetime.now()
    current_user = getpass.getuser()
    ip_addr = socket.gethostbyname(hostname)
    for port in range(1, 65535):
        try:
            sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
            socket.setdefaulttimeout(1)
            result = sock.connect_ex(('127.0.0.1', port))
            if (result == 0):
                a = "LogType: PORT_ENUM - Hostname: {} - CurrentDate: {} - User: {} -
IpAddr: {} - OpenedPort: {}"\
                    .format(hostname, current_date, current_user, ip_addr, port)
                dtf.write(a)
                sock.close()
        except socket.error:
            pass

enumerate_autostart()
```

ДОДАТОК Б

ПРИКЛАД РЕЗУЛЬТАТУ РОБОТИ ПРАВИЛ КОРЕЛЯЦІЇ ПІСЛЯ ІМІТАЦІЇ ЗЛОВМИСНИХ ПОДІЙ