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

МОТРОНЮК Назар Богданович

Алгоритми отримання індикаторів компрометації з відкритих джерел за допомогою телеграм бота/ Algorithms for Obtaining Indicators of Compromise from Open Sources Using a Telegram Bot

Спеціальність 125 – Кібербезпека

Освітньо-професійна програма – Кібербезпека

Кваліфікаційна робота

Виконав студент групи КБм-22
Н.Б. Мотронюк
_____

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

кваліфікаційну роботу допущено
до захисту
"____" _____ 2023 р.
Завідувач кафедри
_____ **В.В. Яцків**

Тернопіль 2023

Факультет комп'ютерних інформаційних технологій
Кафедра кібербезпеки
Освітній ступінь «магістр»
спеціальність: 125 – Кібербезпека
освітньо-професійна програма – Кібербезпека

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ В.В.Яцків
«____» _____ 2022 року

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

МОТРОНЮК НАЗАР БОГДАНОВИЧ

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи:
Алгоритми отримання індикаторів компрометації з відкритих джерел за допомогою телеграм бота/
Algorithms for Obtaining Indicators of Compromise from Open Sources Using a Telegram Bot
керівник роботи к.т.н., доцент Н.Г. Яцків

затверджені наказом по університету від 1 грудня 2022 року № 491

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

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

4. Основні питання, які потрібно розробити:
- визначити сучасні методи виявлення кіберзагроз;
- провести аналіз аналіз сучасних загроз і вразливостей;
- розробка та оптимізація архітектури телеграм-бота.;
- інтеграція з передовими сервісами сканування;
- аналіз ефективності та практичні приклади використання.

5. Перелік графічного матеріалу у роботі:
- види кіберзагроз;
- реальні випадки використання бота для виявлення загроз;
- блок-схеми архітектури бота;
- вигляд та можливості користувача у взаємодії з ботом.

6. Консультанти розділів кваліфікаційної роботи

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

7. Дата видачі завдання 8 грудня 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строки виконання етапів кваліфікаційної роботи	Примітка
1	Аналіз предметної області	12.2022 р. – 03.2023 р.	
2	Виявлення та аналіз індикаторів компрометації	03.2023 р. – 05.2023 р.	
3	архітектура та алгоритм роботи телеграм-бота	05.2023 р. – 11.2023 р.	

Студент _____ Мотронюк Н.Б.
(підпис)

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

АНОТАЦІЯ

Кваліфікаційна робота на тему "Алгоритми отримання індикаторів компрометації з відкритих джерел за допомогою телеграм бота" обсягом 70 сторінок включає 21 ілюстрацію, 1 додаток та 24 джерела за переліком посилань.

Метою даної роботи є розробка та впровадження ефективного телеграм-бота, який автоматизує процес перевірки файлів на можливість інфікованості шкідливим кодом. Досліджено та використано сучасні сервіси перевірки вірусів, такі як VirusTotal, Hybrid Analysis та MetaDefender, для отримання об'єктивної інформації про безпеку файлів.

У роботі розглянуті та аналізовані приклади відповідей від зазначених сервісів, а також розроблений механізм створення звітів для користувачів на основі отриманих результатів. Звіти мають чітку структуру, включаючи інформацію про файл, виявлені загрози, рекомендації та історію аналізів.

Особливу увагу приділено взаємодії з користувачами через телеграм-бота. Розглянуто можливості надсилання звітів, отримання рекомендацій та інші важливі аспекти спілкування.

Розроблений телеграм-бот може ефективно використовуватися для перевірки файлів на віруси, забезпечуючи користувачів надійною та швидкою інформацією про безпеку їхніх файлів. Результати роботи можуть бути корисні для спеціалістів у галузі кібербезпеки та всіх, хто цікавиться захистом від шкідливих програм.

Ключові слова: ТЕЛЕГРАМ-БОТ, ПЕРЕВІРКА ВІРУСІВ, VIRUSTOTAL, HYBRID ANALYSIS, METADEFENDER, ЗВІТИ, БЕЗПЕКА ФАЙЛІВ.

ABSTRACT

The master's thesis on "Algorithms for Obtaining Indicators of Compromise from Open Sources Using a Telegram Bot" is 70 pages long and includes 21 illustrations, 1 appendices, and 24 references.

The purpose of this work is to develop and implement an effective Telegram bot that automates the process of checking files for the possibility of malware infection. Modern virus scanning services, such as VirusTotal, Hybrid Analysis, and MetaDefender, have been researched and used to obtain objective information about file security.

The paper reviews and analyzes examples of responses from these services, and develops a mechanism for generating reports for users based on the results obtained. The reports have a clear structure, including information about the file, detected threats, recommendations, and analysis history.

Special attention is paid to user interaction via a telegram bot. We have considered the possibilities of sending reports, receiving recommendations, and other important aspects of communication.

The developed Telegram bot can be effectively used to scan files for viruses, providing users with reliable and fast information about the security of their files. The results of the work can be useful for cybersecurity professionals and anyone interested in protecting against malware.

Keywords: TELEGRAM BOT, VIRUS SCAN, VIRUSTOTAL, HYBRID ANALYSIS, METADEFENDER, REPORTS, FILE SECURITY.

ВСТУП

В епоху швидкого технологічного розвитку, де кіберзагрози набувають все більшої сутності, проблема кібербезпеки стає важливішою, ніж коли-небудь. Магістерська робота, яку ви маєте нагоду розглядати, зосереджена на глибокому вивченні та практичному впровадженні передових технологічних підходів до виявлення та протидії кіберзагрозам. Цей дослідницький проект ставить своєю метою розкриття новаторських стратегій виявлення шкідливого програмного забезпечення та розробки телеграм-бота, який забезпечує автоматизоване проведення аналізу файлів індикаторів компрометації.

Мета і завдання дослідження. Розробка та апробація телеграм-бота для автоматизованого виявлення індикаторів компрометації в інформаційних системах.

Досягнення визначеної мети передбачає вирішення таких завдань:

– аналіз методів виявлення загроз;

– визначення оптимальних технічних інструментів;

– розробка модуля збору індикаторів;

– автоматизована генерація звітів на основі результатів аналізу;

– Реалізація надсилання звітів та надання рекомендацій через телеграм-бота;

– перевірка ефективності та надійності телеграм-бота, визначення можливостей удосконалення.

Об'єкт дослідження – процеси виявлення індикаторів компрометації в інформаційних системах за допомогою телеграм-бота.

Предмет дослідження – методи та технічні засоби виявлення та аналізу потенційно шкідливих файлів у реальному часі за допомогою телеграм-бота, включаючи інтеграцію з сервісами перевірки та створення звітів для користувачів.

Методи досліджень. У дослідженні використано аналіз літератури, програмування для створення телеграм-бота, емпіричні дослідження та аналіз отриманих результатів.

Наукова новизна одержаних результатів. Дослідження вносить новизну шляхом розробки та імплементації телеграм-бота для автоматизованого аналізу файлів, інтеграції з сервісами перевірки безпеки та створення інформативних звітів для користувачів, що сприяє підвищенню ефективності виявлення та реагування на потенційні загрози.

Практичне значення отриманих результатів. Розроблений телеграм-бот дозволяє автоматизовано виявляти та аналізувати потенційно небезпечні файли, спрощуючи процес моніторингу безпеки. Інтеграція з сервісами перевірки та створення інформативних звітів сприяє оперативному реагуванню на інциденти та надає користувачам зрозумілу інформацію для прийняття рішень. Це допомагає зменшити ризики витоку конфіденційної інформації та підвищити загальний рівень кібербезпеки.

Публікації та апробація КР.

Мотронюк Н.Б. Виявлення та аналіз індикаторів компрометації. Матеріали науково-практична конференція молодих вчених, аспірантів та студентів «Кібербезпека та комп’ютерно-інтегровані технології» (КБКІТ - 2023), Тернопіль, 2023. – С.93-96.

Мотронюк Н.Б. Архітектура та алгоритм роботи телеграм-бота. Матеріали науково-практичного симпозіуму «Захист інформації», Тернопіль, 2023. – С. 130-132.

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Основні поняття кібербезпеки

Інформаційна безпека є одним з найважливіших елементів сучасного світу та глобальної економіки. Вона має велике значення в епоху інформаційних технологій, коли обмін, зберігання та обробка даних стали важливою частиною майже всіх сфер життя. Забезпечення конфіденційності, цілісності та доступності даних стало викликом, який вимагає постійної уваги та інновацій. Однак разом з цими можливостями з'являються і нові загрози інформаційній безпеці.

Інформаційна безпека - це комплекс заходів і стратегій, спрямованих на захист інформації від несанкціонованого доступу, знищення та втрати. Ця сфера охоплює багато елементів, включаючи технічні рішення, законодавство, процедури і навіть людські ресурси. Забезпечення інформаційної безпеки стало викликом, який регулюється нормативно-правовими актами і вимагає співпраці всіх зацікавлених сторін.

Інформаційні системи та мережі знаходяться в центрі економіки та суспільства. Цифрова економіка, включаючи Інтернет, хмарні ресурси та електронну комерцію, стимулює інновації, продуктивність і зручність. Однак, збільшення обміну та обробки інформації також призводить до зростання загроз інформаційній безпеці.

Зловмисники, хакери, кіберзлочинці та державні суб'єкти можуть здійснювати різноманітні кібератаки, включаючи доступ до конфіденційної інформації, крадіжку даних, вимоги викупу, атаки на об'єкти критичної інфраструктури та інші цілі. Ці загрози можуть мати серйозні наслідки для приватних осіб, бізнесу та громадських організацій.

Індикатори компрометації (ІК) відіграють важливу роль у виявленні потенційних порушень безпеки та компрометації. Це сигнали або події, які вказують на можливість несанкціонованого доступу сторонніх осіб до систем або даних; виявлення та аналіз цих індикаторів дозволяє своєчасно реагувати на

інциденти, запобігати подальшим загрозам і знижувати ризики інформаційної безпеки.

Історичний досвід інформаційної безпеки показує, що характер загроз постійно розвивається і змінюється. Загроза несанкціонованого доступу та зловживання даними існувала з перших днів появи комп'ютерів та мереж. Навіть у 1960-70-х роках, коли були створені перші комп'ютерні системи, були зафіксовані перші випадки несанкціонованого доступу та комп'ютерних вірусів. З поширенням персональних комп'ютерів у 1980-х і 1990-х роках з'явилися перші антивірусні програми та засоби захисту.

З розвитком Інтернету та збільшенням кількості підключених до нього пристроїв і систем інформаційна безпека набула ще більшого значення. З появою Інтернету загрози стали більш поширеними і більш витонченими завдяки новим технологіям і методам (рисунок 1.1). Організаціям та урядам доводиться виділяти значні ресурси та приділяти значну увагу заходам інформаційної безпеки.

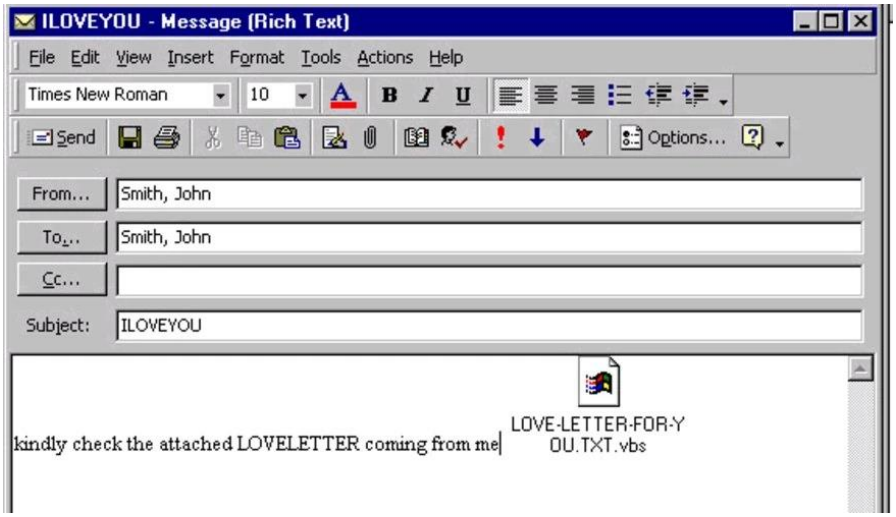


Рисунок 1.1 - Вірус “iloveyou”

У сучасному світі інформаційна безпека має вирішальне значення для економіки та суспільства в цілому. Зростання обсягів обміну інформацією та її обробки значно підвищило важливість інформаційної безпеки. Компанії та уряди виділяють значні ресурси на захист конфіденційної інформації від небажаних

атак. Збитки від кібератак можуть обчислюватися мільйонами доларів і завдати серйозної шкоди репутації організації.

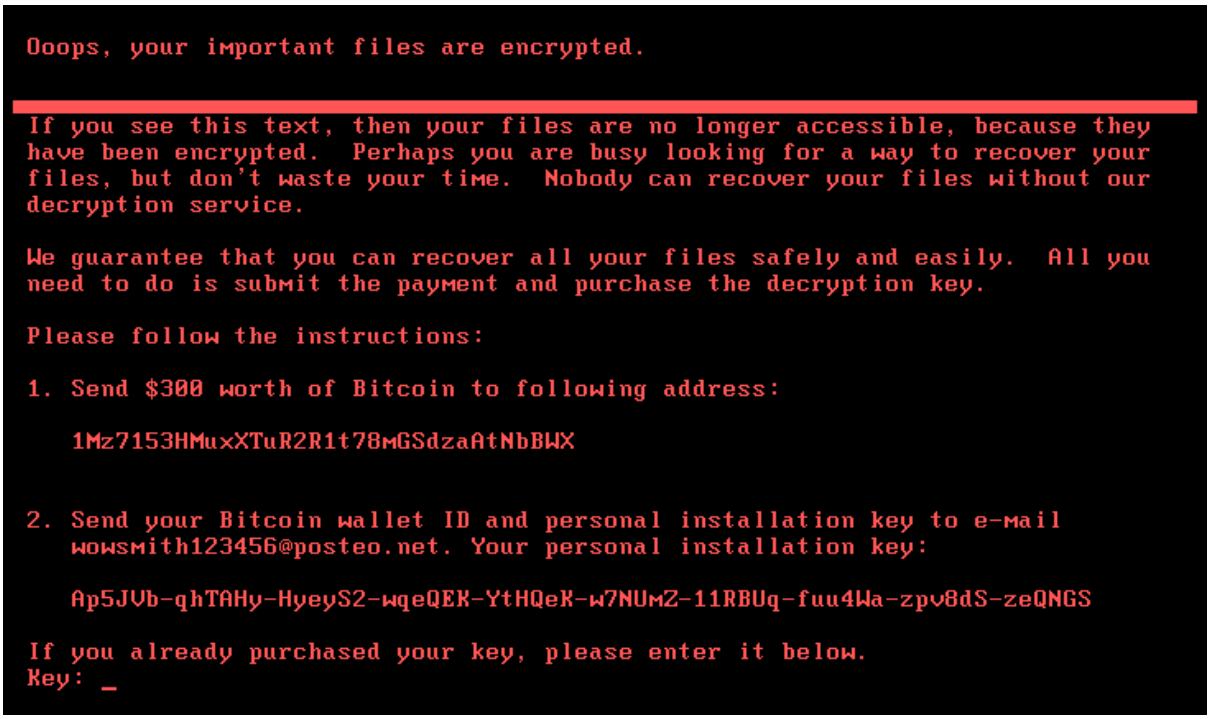


Рисунок 1.2 - Вірус Petya/NotPetya

Споживачі не застраховані від цих проблем. Від крадіжки персональних даних до онлайн-шахрайства - люди стикаються з різноманітними загрозами в Інтернеті (рисунок 1.2).

Нещодавнє збільшення кількості успішних кібератак свідчить про те, що звичайні заходи кіберзахисту недостатньо ефективні для сучасної діяльності кіберзлочинців. Наприклад, у травні 2021 року атака вірусу-шифрувальника серйозно загрожувала об'єктам критичної інфраструктури в США , аж до тимчасової зупинки важливого нафтопроводу.

Здається, на ринку спостерігається тенденція до поширення програм-вимагачів, що збільшило частоту цих типів атак . Потенційні наслідки недостатнього захисту від кібератак варіюються від витоку даних і несанкціонованого використання обчислювальних потужностей до порушення роботи критично важливої інфраструктури. Це може вивести з ладу ключові об'єкти, такі як дамби, лінії електропередач або атомні електростанції.

У цьому контексті той факт, що кіберзлочинці здатні постійно обходити захисні заходи, такі як брандмауери та системи виявлення вторгнень, і

поширювати шкідливе програмне забезпечення, свідчить про те, що такі заходи є неефективними або, принаймні, можуть бути значно вдосконалені.

Одним з можливих покращень є підтримка цих інструментів в актуальному стані, не лише з точки зору програмного забезпечення, але й, особливо, з точки зору інформації. Це означає, що було б корисно мати інформацію про те, як поводяться поточні та нещодавні атаки, як їх виявляти і як їх можна зупинити. Таку інформацію часто розвідкою кіберзагроз. Збір та обмін розвіданими про кіберзагрози є потужним інструментом, оскільки він надає актуальну інформацію про поточні активні кібератаки, що дозволяє ефективно блокувати їх за допомогою традиційних засобів, таких як брандмауери. Наприклад, уявімо собі шкідливе програмне забезпечення, яке використовує протокол SMB для латерального переміщення всередині організації. Дослідник може помітити це і створити частину CTI, яка вказує, що при виявленні цього шкідливого програмного забезпечення в організації, порт, що відповідає протоколу SMB, повинен бути закритий. Однак, чим більше часу проходить між виявленням цієї особливості шкідливого програмного забезпечення та публікацією частини CTI, тим більшому ризику піддається організація. З цієї причини можливість ділитися та отримувати оновлену інформацію є критично важливою для того, щоб ця стратегія працювала.

Ця інформація може надходити у вигляді індикаторів компрометації (IoC). У попередньому прикладі індикаторами компрометації були хеш-значення виконуваного файлу шкідливого програмного забезпечення або порт SMB. В ідеалі, IoC мають бути написані формальною мовою, щоб можна було легко запрограмувати парсер і витягти інформацію автоматично. Натомість у реальності дослідники, які знаходять такі IoC, найчастіше публікують їх у блогах та соціальних мережах. Такі платформи не призначені для аналізу програмним забезпеченням, а скоріше для читання людиною. З

цієї причини вони написані природною мовою, наприклад, англійською, яку людям набагато легше і приємніше читати, але вона не є формальною, що ускладнює програмному забезпеченню ефективне вилучення інформації з неї [1].

Враховуючи важливість IoC для сьогодення і майбутнього кібербезпеки, має сенс докласти зусиль для вилучення такої інформації з оновлених джерел. Однак, як зазначалося раніше, ці джерела часто написані природною мовою, яка не піддається тривіальному програмному аналізу.

Дослідники кібербезпеки часто обговорюють деталі поточних або нещодавніх атак в Інтернеті. Ці деталі надзвичайно цінні для кібербезпеки, але є дві основні проблеми, пов'язані з отриманням розвідданих про загрози з такого роду джерел.

Перш за все, блоги з безпеки і сайти соціальних мереж призначені для читання іншими дослідниками або користувачами, тому вони не структуровані, що ускладнює вилучення цінних знань з них програмним забезпеченням, таким як платформа для розвідки загроз. Наприклад, повідомлення про шкідливу електронну пошту "облікового запису criminal@example.com" може містити такі речення, як "жертва отримала електронний лист від criminal@example.com, що містить корисне навантаження", "criminal@example.com надіслав підозрілий електронний лист жертві" або "шкідливе програмне забезпечення поширилося зі шкідливого DOC-файлу, який був прикріплений до електронного листа від criminal@example.com". Хоча всі ці речення означають, що обліковий запис електронної пошти є зловмисним, витягти цю інформацію автоматично, без втручання людини, не є тривіальним завданням. Таким чином, необхідно провести більш складну обробку, щоб автоматично витягти інформацію в корисному вигляді. Ця проблема не нова. Дійсно, обробка природної мови - це область штучного інтелекту, яка розробляє методи, що дозволяють програмному забезпеченню витягувати знання з текстів, написаних природною мовою.

Другий виклик полягає в тому, що існує величезна кількість онлайн-ресурсів, які надають таку інформацію, і людині неможливо виокремити корисні частини з усього цього в режимі реального часу. Хоча метою процесу оцінювання не було надання повної, актуальної CTI, це дає уявлення про обсяг роботи, яка була б присвячена виключно читанню дописів у блогах у пошуках цінної розвідувальної інформації про кіберзагрози. Це основна причина, чому потрібна система автоматичного вилучення: час експертів з безпеки занадто

цінний, щоб просити їх витрачати більшу частину свого часу на читання публікацій

Ці виклики, як правило, врівноважують один одного. Очевидним рішенням другої проблеми є швидка, автоматична обробка нової інформації з онлайн-ресурсів, що дозволить експертам з безпеки використовувати свій час для вирішення інших питань. Однак перший виклик діє як противага: чим менше залучена людина, тим точнішими мають бути результати від програмних систем. Втім, існують обчислювальні методи, які виглядають перспективними для того, щоб дозволити людині бути менш залученою до збору та оновлення розвідданих про кіберзагрози.

Крім того, сучасні інструменти, такі як iocextract , покладаються на регулярні вирази, які не можуть враховувати контекст, тому схильні до помилкових спрацьовувань. Розглянемо, наприклад, повідомлення в блозі про нову уразливість, виявлену в смартфонах Xiaomi, яка зачіпає, зокрема, версію 11.0.6.1 прошивки MIUI. Будь-який інструмент, заснований на регулярних виразах, сприйме номер версії за IPv4-адресу, оскільки вона має саме такий формат. Такі хибні спрацьовування повинні бути мінімізовані, оскільки інструменти для вилучення IoC призначені для неконтрольованого використання, що означає, що людина не повинна бути залучена до процесу.

З точки зору програмної системи, існує щонайменше дві проблеми, які необхідно вирішити для ефективного вилучення правильних КІЗ. Перша полягає в тому, щоб розпізнати, чи обговорюється в даній статті цінна тема для розвідки кіберзагроз, чи ні. Це пов'язано з тим, що часто навіть технічні блоги можуть публікувати статті, що не стосуються теми, наприклад, навчальні посібники. Другий етап передбачає оцінку достовірності видобутої CTI. Зокрема, індикатори компрометації мають певний формат, але цей формат є спільним для інших об'єктів, які не є IoC. Таким чином, знайдені ІВК необхідно опрацювати, щоб з'ясувати, чи є вони коректними чи ні.

Розглядаючи інформаційну безпеку, не можна не звернути увагу на проблему малих підприємств і підприємців, особливо на тих, які розвиваються у сферах, де конкуренція велика, а ресурсів недостатньо. Забезпечення ефективної

інформаційної безпеки може виявитися вельми витратною справою, особливо коли розглядається використання дорогих інструментів та послуг.

Малі підприємства часто обмежені фінансовими ресурсами та людськими кадрами, які можуть бути відведені на захист інформації. Вони можуть не мати можливості придбати високоякісні антивіруси, фаєрволи чи системи моніторингу мережі. Однак це не робить їх менш схильними до кіберзагроз. У зв'язку з цим, поява доступних та ефективних інструментів для моніторингу та виявлення індикаторів компрометації з відкритих джерел, таких як телеграм-боти, може стати справжньою інновацією [6-7].

Телеграм-боти можуть виконувати безліч завдань в інформаційній безпеці, включаючи виявлення загроз і збір інформації. Вони можуть бути розроблені для автоматичного моніторингу відкритих джерел, де інформація, яка може свідчити про можливі компрометації, стає доступною (рисунок 1.3). Наприклад, такі боти можуть відстежувати появу обговорень та витоків даних, які можуть стати індикаторами загрози для підприємства.

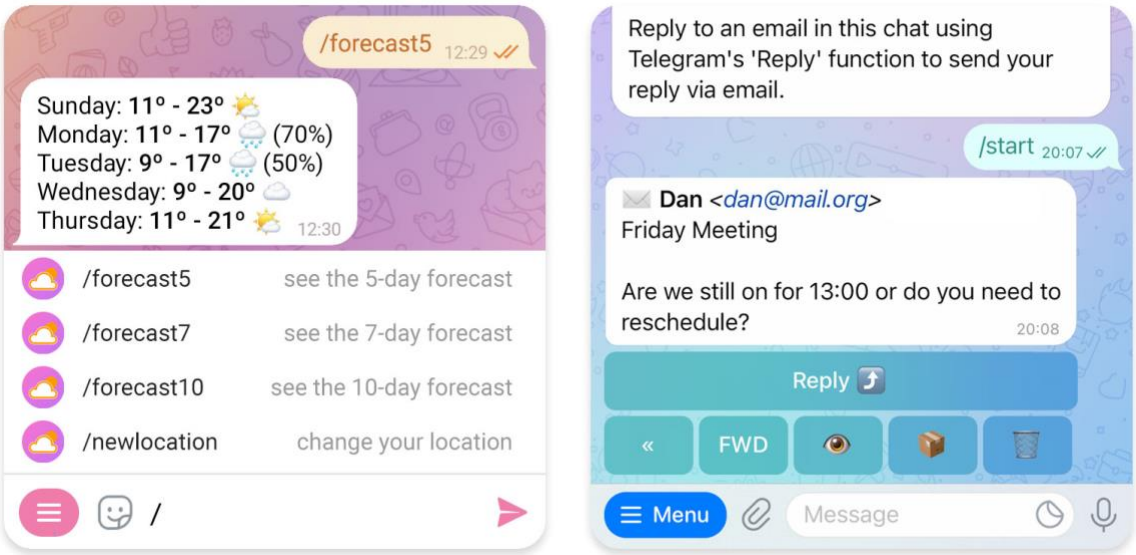


Рисунок 1.3 - Демонстрація можливостей телеграм-бота

Додатково, вони можуть реагувати на певні ключові слова чи фрази, які зазвичай вказують на можливу проблему. Це дає підприємствам можливість оперативно реагувати на загрози та компрометації, навіть при обмежених ресурсах.

Повний інтеграційний проект з використанням телеграм-бота для збору індикаторів компрометації може стати додатковою перевагою для малих підприємств. Цей проект може допомогти підприємствам виявляти загрози на ранніх стадіях, що зменшує ризики та можливі збитки. Також він дає можливість підприємствам заощадити кошти, які інакше були б витрачені на дорогі інструменти та послуги.

Ця ідея є цікавою не лише з погляду захисту інформації, але й з точки зору стимулювання розвитку та конкурентоспроможності малих підприємств. Можливість забезпечити ефективний захист інформації за доступну ціну може стати ключовим фактором у їх успішному функціонуванні.

У цьому контексті ця робота досліджує потенціал використання Telegram-ботів для збору IP з відкритих джерел. Telegram-боти стають поширеним і доступним інструментом для автоматизації різних завдань у сфері інформаційної безпеки, в тому числі збору та аналізу IP. Це так.

1.2 Сучасні методи виявлення кіберзагроз

Аналіз літератури є важливим кроком у дослідженні будь-якої теми, в тому числі визначення та аналізу індикаторів консенсусу (ІК) та використання телеграм-ботів в інформаційній безпеці. За результатами такого аналізу можна визначити поточний стан досліджень, сучасні тенденції та можливі прогалини в існуючих дослідженнях.

Протягом останніх кількох десятиліть спостерігається зростання інтересу до інформаційної безпеки та захисту даних. Дослідники та експерти в галузі кібербезпеки активно займаються розробкою методів виявлення, аналізу та запобігання інцидентам безпеки. З огляду на різні аспекти такої діяльності, темі виявлення та аналізу ІК присвячено низку публікацій.

Поняття індикатора порушення включає в себе широкий спектр даних, що вказують на можливість порушення інформаційної безпеки. Це і виявлення аномальної мережевої активності, аномальних вхідних або вихідних потоків даних, зміни в конфігурації системи або інші індикатори можливої загрози.

Важливо зазначити, що існує низка інструментів і технологій для виявлення ІК, багато з яких орієнтовані на великі підприємства та уряди зі значними бюджетами на кіберзахист. Однак ресурси, доступні для МСП, особливо стартапів і малого бізнесу, обмежені, тому важливо знайти альтернативні засоби для забезпечення інформаційної безпеки.

Огляд літератури також виявив низку досліджень, присвячених використанню Telegram-ботів в інформаційній безпеці. Ці дослідження вказують на потенціал ботів як засобу автоматизації різних процесів, таких як збір інформації та виявлення ІЧ-витоків. Такий підхід є особливо корисним для МСП, які не завжди можуть дозволити собі дорогі інструменти та послуги з кібербезпеки.

Загалом, аналіз літературних джерел дозволяє визначити актуальність теми, поточний стан досліджень та можливості для подальших розвідок у контексті виявлення та аналізу IP за допомогою телеграм-ботів.

Індикаторами порушення є активність та/або шкідливі роботи
виявлені в мережі або на хост-комп'ютері. На практиці це не просто перелік індикаторів

Це не перелік індикаторів, а первинна інформація про інцидент для аналізу, розслідування та реагування. Індикатори можна спостерігати як на рівні мережі, так і на рівні сервера.Вони допомагають виявити різні ситуації.

Наприклад це допомагає виявити випадки несанкціонованого входу або дії, пов'язані з несанкціонованим входом. Індикатор порушень експертами з кібербезпеки для виявлення спроб вторгнення на ранній стадії та часто використовується експертами з кібербезпеки для раннього виявлення спроб вторгнення та початкової оцінки потенційних загроз.

Приклади включають проактивне блокування шкідливого трафіку і виявлення скомпрометованих серверів. Сюди входить виявлення скомпрометованих серверів. IoC призначені для широкого розповсюдження і можуть містити такі шаблони. IoC може містити шаблони для IP-адрес, пов'язаних з фішингом і атаками на відмову в обслуговуванні (DoS).

Індикатори порушення включають імена файлів та поведінку, що спостерігається під час активної роботи шкідливого програмного забезпечення, в тому числі і те, і інше, обидва показники включені.

Найпоширенішими індикаторами є аномальний мережевий трафік. Припускаючи, що трафік у мережах, які ми захищаємо, є безпечним. Однак вихідний трафік може бути використаний для наступного
надсилання інформації на сервер зловмисника.

Аномалії в роботі привілейованих облікових записів.
Нерідко зловмисники використовують підвищення привілеїв у привілейованих облікових записах.

Географічно нерегулярні входи є вагомим доказом віддаленого доступу. Наприклад, трафік до та з країн, де компанія не веде бізнесу, є вагомим доказом віддаленого доступу.

Ще одна з причин для перевірки розмір HTML-сторінки. Наприклад.
Зловмисник використовує SQL-ін'єкцію і отримує HTML-відповідь.
Він отримує HTML-відповідь, яка набагато більша за звичайний запит.
Кілька запитів до одного і того ж файлу. Під час атаки URL-адреса, швидше за все, буде змінюватися з кожним запитом, але частина імені файлу - ні. Це відбувається тому, що зловмиснику потрібно виконати ряд різних перестановок. Індикатори порушень розрізняються за кількістю корисної інформації, яку вони містять. Неповні індикатори, тобто індикатори, які не містять додаткових даних, таких як контекст або час першого виявлення, не можуть надати корисну інформацію зловмиснику. У цьому випадку обґрунтувати блокування трафіку або процесів на основі отриманої інформації майже неможливо.

Причини використання індикаторів наступні:

Простота використання. IoC простий у використанні, якщо існує інфраструктура для його підтримки. Це так. STIX реалізується через YARA та вбудований метод розгортання MISP.
MISP.

Розповсюдження в таких проектах, як MISP, дозволяє великій кількості компаній з невеликою кількістю індикаторів захистити свої системи.

Одна з найбільших переваг. розрахункові індикатори роблять можливим аналіз ризиків,вплив або загрозу можна проаналізувати, а конкретні ризики можна розставити за пріоритетами, прийняти або скомпрометувати. Таким чином, компанії можуть отримати технічну свободу.

Можливість поєднувати нові механізми з давно існуючими даними, такими як DNS-запити або хеші вкладень електронної пошти, шукати ознаки попередніх порушень. За допомогою цієї технології можна надати повну картину попередніх атак і пом'якшити наслідки вторгнень, які вже були здійснені.

Антивіруси - це каталоги та бібліотеки на хостах інфраструктури, які підтримують Індикатор узгодження. Однак, у мережі з'являється шкідливе програмне забезпечення. Зловмисники використовують різні тактики, методи і процедури (TTP) та спеціалізовані інфраструктури. Якщо шкідливе програмне забезпечення опиняється у відкритому доступі в мережі і залишається в силі, виконуваний файл і хеш лишаються незмінними. У цьому випадку використання індикаторів залишається важливим.

Проте, навіть у такому випадку, коли використовується інший виконуваний файл, який змінює хеш, можна застосовувати заходи безпеки для запобігання атакам. Наприклад, можна блокувати відомі зловмисні DNS-запити. Таким чином, індикатори порушень не вирішують всі аспекти захисту інфраструктури, проте їх наявність є необхідною для створення багаторівневого захисту.

Найпоширенішими методами для класифікації індикаторів врегулювання є "Піраміда болю", яку представив Девід Б'янко у 2013 році. У цій піраміді оцінюється потенційна корисність даних про загрози, враховуючи складність їхнього збору та використання. Тобто, визначається, наскільки дані про загрози можуть бути корисними для конкретних завдань [2].

Діаграма на рисунку 1.4 представляє "піраміду болю" і показує взаємозв'язок між типами індикаторів, які можуть бути використані для виявлення активності зловмисника, і тим, скільки болю це може завдати зловмиснику.

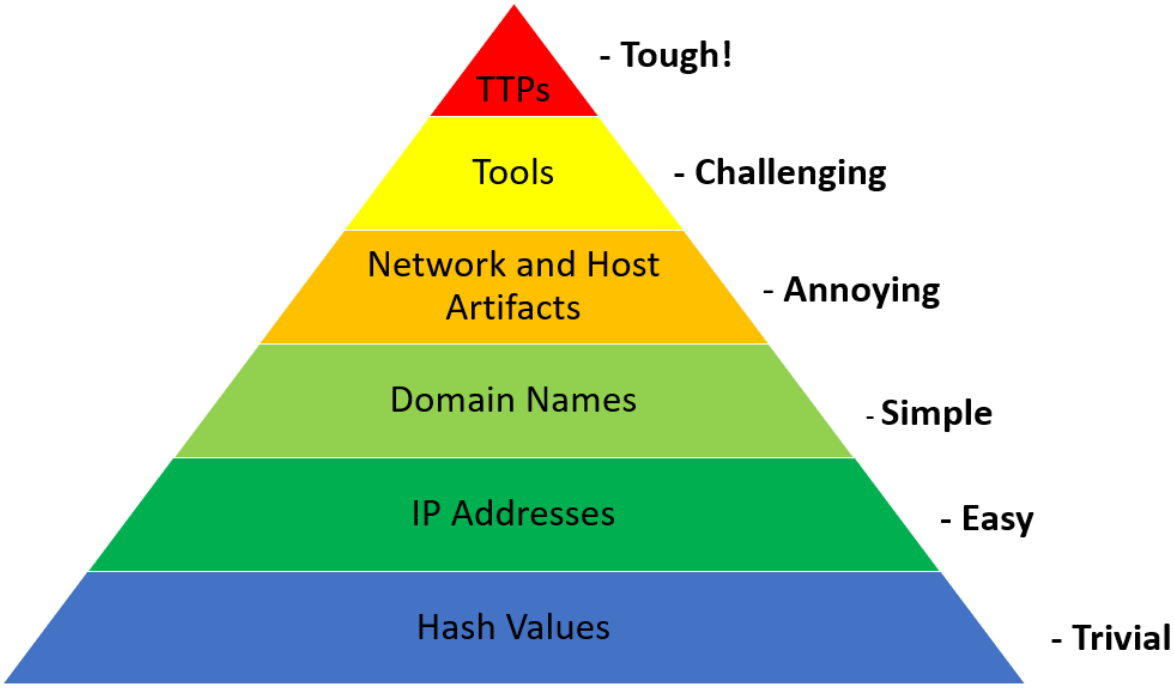


Рисунок 1.4 - Візуалізація піраміди болю

Чим вищий рівень піраміди, тим більше часу і зусиль потрібно для зміни підходу до атаки на інфраструктуру.

Піраміда болю включає різні рівні, які характеризуються відповідними аспектами "болю" і "продуктивності". Чим вище рівень в піраміді, тим складніше виявити індикатори цього рівня, але вони є більш корисними для визначення складних загроз. Рівні піраміди можуть бути описані наступним чином:

Рівень файлів та об'єктів: На цьому рівні зазвичай використовуються індикатори, які включають хеш-суми файлів, імена файлів або їхні розміри. Ці індикатори є легкими для виявлення, але менш корисними, оскільки їх легко змінити або приховати.

Рівень діяльності: На цьому рівні включаються індикатори, які відстежують IP-адреси, URL-адреси та діапазони портів. Їх виявлення може бути трохи складнішим, але вони надають більше інформації про дії загрози.

Рівень образу: Цей рівень використовує індикатори, які включають фрази, ключові слова та регулярні вирази. Вони дозволяють виявляти загрози на основі текстового аналізу інформації.

Рівень коду: На цьому рівні використовуються індикатори, які пов'язані з програмним кодом, такі як скрипти та вразливості програмного забезпечення. Це вже більш технічний рівень, і виявлення може бути складнішим.

Рівень дій: Тут включаються індикатори, які стосуються методів та структур атаки, тактики кіберзлочинця і цілі атаки. Це рівень, на якому можна розуміти, як саме працює загроза.

Рівень цілей: На найвищому рівні піраміди розглядаються індикатори, що вказують на мету атаки та стратегію кіберзлочинця. Ці індикатори найскладніше виявити, але вони надають найбільше інформації для розуміння загрози.

Ця модель допомагає організаціям визначати індикатори, які найкраще відповідають їхнім потребам та загрозам, що існують у їхніх системах. Вибір правильних індикаторів допомагає вчасно виявляти та відповідати на кіберзагрози, підвищуючи загальний рівень кібербезпеки.

Формулювання завдань дослідження та обґрунтування можливих напрямів їх вирішення є важливим етапом підготовки дипломної роботи. На основі аналізу предметної області та літератури можна сформулювати завдання дослідження та визначити напрямки, які можуть бути корисними для досягнення мети дослідження.

Мета дослідження. Розробити та налаштувати телеграм-бота для виявлення індикаторів порушень (ІП) з відкритих джерел: першим завданням є розробка самого бота, який може моніторити відкриті джерела та виявляти ІП. Це передбачає підбір необхідних інструментів і програмних бібліотек та розробку алгоритмів виявлення ІП.

Аналіз методів виявлення ІК та інтеграція з телеграм-ботами: Друге завдання полягає у вивченні існуючих методів виявлення ІК, таких як аналіз мережевої активності та аналіз подій. Потім буде показано, як ці методи можуть бути інтегровані з телеграм-ботами для забезпечення високоякісного ІЧ-виявлення.

Розробка системи оповіщення та реагування на виявлені ІР: у цьому завданні буде створено систему оповіщення для надсилання повідомлень користувачам та адміністраторам при виявленні ІР. Також будуть розроблені

процедури реагування на виявлені загрози та рекомендації щодо подальших контрзаходів.

1.3 Огляд технічних аспектів використання телеграм-ботів

Використання машинного навчання та текстової аналітики Для виявлення ІД з відкритих джерел можна використовувати методи машинного навчання та текстової аналітики. Наприклад, навчити моделі розпізнавати ключові слова та фрази, які можуть вказувати на загрозу.

Інтеграція з сучасними інструментами кібербезпеки: Для підвищення ефективності та точності виявлення КІ важливо розглянути можливість інтеграції з існуючими інструментами кібербезпеки, такими як системи моніторингу та аналізу мережевої активності.

Аналіз та використання інформації з відкритих джерел: Для ефективного виявлення ІК важливо аналізувати та використовувати інформацію з відкритих джерел, де міститься інформація про потенційні загрози, таких як новинні портали, форуми та соціальні мережі.

Оцінка ефективності та оптимізація ботів: важливим напрямком є оцінка ефективності розроблених ботів та пошук шляхів їх оптимізації для швидкого та надійного виявлення КІ.

Аналіз відповідності стандартам кібер гігієни та безпеки: важливо забезпечити відповідність розроблених ботів стандартам кібер гігієни та безпеки, що може бути досягнуто шляхом використання відповідних методів та підходів.

Ці виклики та сфери, які потребують вирішення, створюють основу для подальших досліджень та розробки телеграм-ботів для виявлення індикаторів порушень з відкритих джерел. Подальший процес дослідження включатиме розробку та налаштування бота, проведення експериментів та оцінку їх ефективності, а також пошук шляхів покращення функціональності та точності виявлення ІР.

2. ВИЯВЛЕННЯ ТА АНАЛІЗ ІНДИКАТОРІВ КОМПРОМЕТАЦІЇ

2.1 Теоретичне підґрунтя та концепції

У цьому розділі розглядаються теоретичні засади та ключові поняття, важливі для розуміння теми нашого дослідження, а саме визначення індикаторів консенсусу (ІК) та використання телеграм-ботів у цьому контексті.

Індикатори компрометації - це ознаки або показники того, що система або мережа інформаційної безпеки могла бути скомпрометована. Ці індикатори включають аномальну мережеву активність, аномальні журнали подій, несанкціонований доступ до ресурсів, атаки на вразливості тощо. Виявлення індикаторів порушення дозволяє швидко реагувати на потенційні загрози та вчасно вживати заходів для захисту ваших інформаційних активів.

Кібербезпека - це сфера, яка займається захистом інформаційних систем, мереж і даних від різних кіберзагроз. Сюди входять заходи щодо забезпечення конфіденційності, цілісності та доступності даних, виявлення та реагування на інциденти безпеки, а також запобігання потенційним загрозам.

Telegram-боти - це додатки, розроблені для платформи мессенджера Telegram, які автоматизують взаємодію з користувачами та надають низку послуг. Вони можуть використовуватися для різних функцій, таких як надсилання повідомлень, збір інформації або виконання завдань. Telegram-боти набули популярності завдяки простоті розробки та інтеграції з платформою Telegram.

Машинне навчання - це галузь штучного інтелекту, яка досліджує розробку алгоритмів і моделей, здатних навчатися на основі даних і виконувати завдання без явного програмування. Машинне навчання можна використовувати для аналізу та виявлення аномалій у мережевій активності, а також для виявлення IP.

Текстовий аналіз - це галузь обробки природної мови, яка досліджує методи та алгоритми для розуміння та аналізу текстової інформації. У контексті кібербезпеки текстовий аналіз може бути використаний для пошуку ключових слів і фраз у повідомленнях і журналах, які можуть вказувати на IP.

Моніторинг мережі: процес спостереження за діяльністю в комп'ютерних мережах з метою виявлення аномалій і потенційних загроз кібербезпеці. Моніторинг мережі включає аналіз пакетів даних, реєстрацію та виявлення аномальної активності.

Система виявлення індикаторів (IDS): спеціалізована система, яка виявляє аномальну або підозрілу активність у мережі та повідомляє про потенційні загрози; IDS виявляють ІС за допомогою різних методів, включаючи аналіз сигнатур та аналіз аномалій.

Системи управління інцидентами та подіями (SIEM): Системи SIEM поєднують моніторинг мережі та аналіз подій для виявлення ІК і реагування на інциденти. Дані з різних джерел можуть бути об'єднані та проаналізовані для виявлення загроз.

Виявлення аномалій: Цей метод виявлення ІС виявляє аномалії або незвичайні дії в системі. Аналіз аномалій може використовувати статистичні методи, машинне навчання та інші підходи для виявлення аномальних патернів.

Захист від кіберзагроз: це поняття включає в себе всі заходи, спрямовані на захист інформаційних активів від потенційних загроз, включаючи виявлення IP, реагування на інциденти та запобігання потенційним атакам.

Системи інформаційної безпеки: комплексні системи та методи захисту інформації та інформаційних систем. Включає різні аспекти кібербезпеки, такі як контроль доступу, шифрування та антивірусні заходи.

Загрози кібербезпеці включають низку атак та інцидентів, які призводять до компрометації інформаційних систем та мереж. Ці загрози можуть бути спрямовані на конфіденційність, цілісність та доступність даних і ресурсів. Загрози включають атаки внутрішніх користувачів, зовнішні атаки, атаки через вразливість програмного забезпечення, соціальну інженерію та інші.

Вразливості - це слабкі місця в інформаційних системах, які можуть бути використані зловмисниками для отримання несанкціонованого доступу або спричинення негативних наслідків. Вразливості можуть бути спричинені неефективним захистом, помилками в програмному забезпеченні, недостатньою

конфігурацією або іншими причинами. Виявлення та усунення вразливостей є важливою частиною кібербезпеки.

Ідентифікація - це визначення користувача або організації, яка намагається отримати доступ до системи або ресурсу. Автентифікація - це підтвердження та перевірка ідентичності цього суб'єкта. Використовуючи поєднання ідентифікації та автентифікації, система може визначити, чи має користувач дозвіл на доступ до певного ресурсу.

Авторизація визначає, які дії користувач може виконувати після проходження ідентифікації та автентифікації. Сюди входить визначення прав доступу, ресурсів, якими користувач може користуватися, та обмежень для запобігання несанкціонованому використанню ресурсів.

Кібергігієна включає заходи та практики для забезпечення безпеки інформаційних систем та мереж. Сюди входить захист від загроз, які можуть виникнути в результаті кібератак, включаючи запобігання інцидентам, їх виявлення та реагування на них.

Кібербезпека тісно пов'язана з бізнес-процесами, оскільки безпека інформаційних систем може впливати на функціонування компанії. Важливо враховувати бізнес-процеси при розробці стратегій кібербезпеки та заходів для забезпечення безперервності бізнесу.

Інцидент кібербезпеки - це подія, яка вказує на потенційне порушення інформаційної системи або мережі. Інциденти включають атаки, витік даних і втрату доступу до ресурсів. Виявлення та реагування на інциденти є важливою частиною кібербезпеки.

Захист інформації включає технічні та організаційні заходи для забезпечення конфіденційності, цілісності та доступності інформації. Сюди входять заходи для захисту даних і ресурсів, такі як шифрування, контроль доступу та резервне копіювання.

2.2 Аналіз сучасних загроз і вразливостей

Розповсюдження вірусів-викривачів: Вірус Петя, також відомий як "NotPetya," був однією з найпоширеніших та найагресивніших атак, яка вразила світ кіберзлочинності. Він розповсюдився через вразливості у віконних операційних системах та зашифрував дані користувачів, вимагаючи великий викуп для їх розблокування. Ця атака почалася в Україні та поширилася світом, вплинувши на компанії та організації у багатьох країнах. Внаслідок цього багато компаній стали свідками серйозних фінансових втрат і втрати довіри клієнтів.

Фішингові атаки та втрата особистої інформації: Один із найвідоміших прикладів фішингової атаки - це атака на компанію Equifax. У 2017 році зловмисники використали фішингову кампанію для отримання доступу до систем компанії та зламали базу даних, в якій містилася особиста інформація мільйонів осіб. Ця атака призвела до втрати соціальних номерів, адрес та інших конфіденційних даних понад 145 мільйонів осіб. Ця інформація стала легкодоступною для зловмисників, що могло призвести до шахрайства, крадіжки ідентичності та інших злочинів.

Атаки на інфраструктуру організацій: Однією з гучних атак на інфраструктуру є атака на компанію Maersk, проведена за допомогою вірусу-викривача "NotPetya". Зловмисники вимагали викуп у біткоїнах, загрожуючи розголошенням конфіденційної інформації компанії. Атака призвела до серйозних технічних проблем та перебоїв у роботі компанії, що відбулося великими фінансовими збитками та втратою репутації.

Кібершпигунство та втрата комерційних та військових секретів: Кібершпигунська атака групи APT29, також відомої як "Cozy Bear," на DNC в 2016 році вивільнила конфіденційну інформацію про політичні процеси та вибори. Ця атака призвела до серйозних внутрішніх конфліктів та розколу в політичному середовищі. Внаслідок цього атака була широко висвітлена в ЗМІ та призвела до міжнародних реакцій.

Атаки на критичну інфраструктуру: Україна стала свідком атаки на свою електроенергетичну систему у грудні 2015 року. Зловмисники вимкнули системи

керування та призвели до перебоїв у подачі електроенергії на півдні країни. Ця атака стала однією з перших в світі, коли кіберзагрози призвели до фізичних наслідків для критичної інфраструктури.

Ці приклади демонструють різноманітність кіберзагроз і їхній потенційно серйозний вплив на інформаційну безпеку та суспільство в цілому. Виявлення індикаторів компрометації стає надзвичайно важливим для запобігання та реагування на такі загрози та мінімізації їхніх наслідків.

2.3 Методи виявлення індикаторів компрометації

У цьому підрозділі детально розглядаються різні методи і технології виявлення індикаторів компрометації, які використовуються в сучасних системах кібербезпеки. Основними методами є такі

Сигнатурний аналіз: сигнатурний аналіз: цей метод заснований на використанні відомих сигнатур або шаблонів для конкретних типів атак. У разі виявлення сигнатури атаки система зіставляє її з виявленими даними і виявляє індикатори порушення.

Аналіз аномалій: цей підхід дає змогу виявити аномальну поведінку або поведінку системи. Для цього використовують алгоритми машинного навчання, які були навчені на нормальній поведінці і можуть розпізнавати аномалії.

Евристичні методи: у цих методах для виявлення ознак порушення використовують правила та евристику. Для виявлення потенційних загроз можуть використовуватися знання про загрози і сценарії атак.

Інтелектуальні системи виявлення: інтелектуальні системи виявлення використовують штучний інтелект і машинне навчання для виявлення ознак порушення. Вони здатні аналізувати великі обсяги даних і робити розширені висновки про потенційні загрози.

2.4 Роль ботів Telegram у виявленні індикаторів компрометації

У цьому підрозділі детально розглядається роль ботів Telegram у виявленні індикаторів компрометації та їхня функціональність. Пояснюється, яким чином ці боти можуть бути використані для автоматизації процесу збору та аналізу інформації з відкритих джерел з метою виявлення потенційних загроз.

Збір інформації: бот Telegram може бути налаштований на автоматичний збір інформації з різних джерел, включно з веб-сайтами, соціальними мережами та новинними порталами. Наприклад, бот може регулярно перевіряти наявність нових повідомлень на форумах, де обговорюють потенційні кіберзагрози, і надсилати сповіщення в разі виявлення підозрілих даних або посилань.

Аналіз інформації: бот Telegram може аналізувати зібрану ним інформацію і виявляти ознаки порушення. Наприклад, боти можуть аналізувати текстову інформацію на предмет ключових слів і фраз, які можуть вказувати на потенційну загрозу. Це може включати виявлення шкідливих URL-адрес або витоків даних.

Застосування: практичне застосування ботів Telegram може включати моніторинг соціальних мереж для виявлення обговорень кіберзагроз, аналіз новинних джерел для виявлення повідомлень про кібератаки або перевірку веб-сайтів на наявність шкідливих файлів.

Можливості вдосконалення: Telegram-бот може бути доопрацьований з метою підвищення його ефективності у виявленні ознак компрометації. Наприклад, можна вдосконалити алгоритм аналізу, щоб зменшити кількість хибних попереджень і розширити джерела інформації, що збирається.

2.5 Конкретні приклади та дослідження

У цьому розділі представлено конкретні практичні приклади використання методів виявлення індикаторів витоку та ролі ботів Telegram. Буде детально описано результати досліджень та особистий досвід використання цих методів для виявлення КІ.

Приклад 1: Використання бота Telegram для моніторингу форумів і обговорень потенційних кіберзагроз. Бот автоматично аналізує текстові повідомлення і виявляє ключові слова і фрази, що вказують на наявність загрози. У разі виявлення підозрілого повідомлення бот надсилає сповіщення адміністратору для подальшого аналізу.

Приклад 2: Використання методів аналізу аномалій для виявлення аномальної активності в комп'ютерних мережах Бот Telegram автоматично відстежує мережевий трафік і виявляє аномалії, як-от незвичні підключення до серверів або підвищене передавання даних. Це може свідчити про можливу кібератаку.

2.6 Порівняльний аналіз методів і підходів

У цьому розділі проводиться порівняльний аналіз різних методів виявлення індикаторів компрометації, їхніх переваг і недоліків.

Аналіз сигнатур.

Переваги:

Висока точність виявлення відомих загроз завдяки використанню точних сигнатур.

Можливість швидкого реагування на відомі атаки.

Недоліки.

Неефективний при виявленні нових невідомих загроз (атак "нульового дня").

Численні помилкові спрацьовування при зміні сигнатур атак.

Аналіз аномалій:

Переваги:

Ефективний під час виявлення незвичайних або рідкісних атак, оскільки не вимагає використання відомих сигнатур.

Може виявляти нові, раніше невідомі загрози.

Недоліки:

Багато помилкових спрацьовувань через звичайну аномальну активність.

Вимагає багато часу, оскільки моделі необхідно навчати на зразках нормальної активності.

Евристичний метод:

Переваги:

Можливість реагування на нові загрози на основі експертних знань і правил.

Менше помилкових спрацьовувань завдяки використанню чітких правил.

Недоліки.

Неефективність при виявленні невідомих атак без оновлення правил.

Залежність від якості експертних знань і валідності правил.

Інтелектуальні системи виявлення:

Переваги:

Здатність виявляти складні загрози і закономірності, які не можуть бути зрозумілі людиною.

Здатність адаптуватися до мінливих умов і загроз.

Недоліки.

Потрібен великий обсяг даних для навчання моделей.

Складність розуміння рішень, що приймаються інтелектуальними системами.

Порівняння цих методів показує, що жоден із них не є універсальним ідеальним рішенням для всіх ситуацій. Вибір методу виявлення індикаторів компрометації має ґрунтуватися на конкретних потребах і обставинах організації. Сигнатурний аналіз ефективний для виявлення відомих загроз, аналіз аномалій може допомогти виявити нові атаки, евристичні методи ефективні для виявлення закономірностей, заснованих на правилах, а інтелектуальні системи виявлення можуть використовуватися для адаптації до мінливих умов.

Крім того, роль Telegrambot у виявленні ознак компрометації полягає в автоматизації процесу збору та аналізу інформації з відкритих джерел, що дає змогу збільшити час реакції та знизити ризик помилкових спрацьовувань.

2.7 Сучасні проблеми та прогалини

Приклад 1: Нові типи атак Останніми роками збільшилася кількість нових типів кібератак, що становлять загрозу інформаційній безпеці. Одним із прикладів є атаки, що використовують соціальні мережі як засіб поширення шкідливих програм. Наприклад, фішингові атаки, спрямовані на користувачів соціальних мереж, можуть становити серйозну небезпеку для організацій, оскільки обманом змушують користувачів розкривати конфіденційну інформацію.

Приклад 2: Методи обходу системи безпеки: зловмисники постійно вдосконалюють способи обходу системи безпеки. Вони використовують шифрований трафік для приховування своїх дій, застосовують методи стеганографії для приховування шкідливого коду у звичайному потоці файлів або використовують вразливості "нульового дня", які ще не було виявлено та виправлено. Це створює постійну проблему для цілей виявлення індикаторів порушень, які повинні забезпечувати захист від нових методів атак.

Приклад 3: Галузеві проблеми: у різних галузях існують свої проблеми кібербезпеки. Наприклад, у фінансовій галузі, де обробка фінансових операцій є критично важливою, загрози фінансового шахрайства та атаки на банківські системи становлять особливу небезпеку. У таких ситуаціях методи виявлення мають бути спрямовані на раннє виявлення подібних загроз і своєчасне реагування на них.

Приклад 4: Необхідність превентивних заходів З огляду на постійні загрози інформаційній безпеці, важливо здійснювати превентивні заходи. До них належать регулярне оновлення програмного забезпечення та систем безпеки з метою усунення вразливостей, навчання співробітників методам забезпечення кібербезпеки, а також моніторинг внутрішньої діяльності з метою виявлення незвичайних закономірностей і можливих загроз.

Приклад 5: Необхідність реалізації заходів реагування: поряд із визначенням ІС важливо мати чіткий і добре опрацьований план реагування на інциденти. Кібератаки можуть поширюватися дуже швидко, і ефективні заходи

реагування можуть запобігти подальшим загрозам і збиткам. Дуже важливо розробити план реагування, що включає ізоляцію загрози, відновлення системи та повідомлення всіх зацікавлених сторін.

2.8 Потенційні переваги використання ботів Telegram

У цьому розділі описуються потенційні переваги використання ботів Telegram у процесі виявлення ознак порушень порівняно з іншими методами та інструментами; боти Telegram можуть принести значні переваги в цій галузі, які детально аналізуються тут:

Автоматизований збір інформації: боти Telegram можуть бути налаштовані на автоматичний збір інформації з різних відкритих джерел, таких як форуми, соціальні мережі та блоги. Це дає змогу швидко отримувати актуальні дані про потенційні загрози та індикатори порушень. Наприклад, боти можуть відстежувати форуми, де обговорюються нові методи атак, і завчасно повідомляти про них.

Аналіз тексту і мови: бот Telegram може бути налаштований на аналіз тексту і мови для виявлення підозрілих або загрозливих повідомлень. Наприклад, аналіз настроїв може використовуватися для виявлення загрозливих виразів або слів у тексті, щоб допомогти визначити ознаки порушення в онлайн-обговореннях.

Миттєве сповіщення та реагування: бот Telegram може миттєво надсилати попередження та повідомлення про потенційні загрози, що дає змогу швидко реагувати на інциденти. Наприклад, якщо бот виявить підозрілу активність або зміни в мережі, адміністратора або відповідальну особу можна повідомити для подальшого аналізу та реагування.

Спрощення аналізу великих обсягів даних: великі обсяги даних складно обробляти вручну, а боти Telegram можуть використовувати аналітичні алгоритми та штучний інтелект для опрацювання великих обсягів інформації та виділення потенційних ознак порушення. Наприклад, боти можуть автоматично агрегувати дані з різних джерел і виділяти ключові закономірності та аномалії.

Забезпечення безперервного моніторингу: оскільки боти Telegram працюють без перерв, вони можуть безперервно відстежувати інформаційний простір і виявляти ознаки порушення в режимі реального часу. Це особливо корисно для організацій, що мають велику кількість активів і схильні до ризику стати об'єктом кібератак.

Інтеграція з іншими інструментами: бот Telegram може бути легко інтегрований з іншими інструментами і системами моніторингу для створення комплексної системи виявлення ознак злому. Наприклад, боти можуть взаємодіяти з системами реєстрації подій та управління інцидентами для автоматичного введення інформації про інциденти та вжиття подальших заходів.

Приклад 1: спрощення аналізу даних Наприклад, бот Telegram може бути налаштований на аналіз потоків даних із різних джерел і виділення аномалій, що свідчать про можливе порушення. Бот може виявити підозрілу активність, незвично підвищену активність користувачів або зміни в мережевому трафіку, які можуть свідчити про атаку.

Приклад 2: Забезпечення безперервного моніторингу Великі підприємства та державні організації з великою кількістю комп'ютерів і мережевих активів можуть використовувати ботів Telegram для безперервного моніторингу своєї інфраструктури та реагування на події в режимі реального часу. Боти можуть автоматично виявляти потенційні загрози і повідомляти адміністраторів.

Приклад 3: Інтеграція з іншими інструментами Боти Telegram можуть бути інтегровані з іншими інструментами для створення повноцінної системи моніторингу та реагування на інциденти. Наприклад, бот може взаємодіяти з системою реєстрації подій, яка фіксує всю мережеву активність і автоматично створює інциденти на основі виявлених індикаторів порушення.

2.9 Висновки та обґрунтування обраного напряму

У цьому розділі було проаналізовано теоретичні аспекти та методи виявлення ознак порушень, а також розглянуто переваги використання ботів Telegram у цьому контексті. Висновки цього розділу є важливими для

подальшого розвитку дослідження та обраного напряму. Нижче наведено основні висновки та обґрунтування обраного напряму:

Аналіз теоретичних аспектів: у результаті вивчення теоретичних засад і концепцій, пов'язаних із виявленням індикаторів порушення, було виявлено ключові аспекти, що впливають на інформаційну безпеку. До таких аспектів належать аналіз тексту та мови, автоматизація збору інформації та спрощення аналізу великих обсягів даних. Ці теоретичні знання є основою для розробки практичних методів виявлення.

Аналіз методів і підходів У цьому розділі представлено огляд різних методів і підходів, що використовуються в галузі виявлення індикаторів компрометації. Цей аналіз допоможе зрозуміти сильні та слабкі сторони наявних методів і виявити можливості для їхнього вдосконалення або розробки нових підходів.

Порівняльний аналіз методів і підходів Важливо зазначити, що не існує універсального методу виявлення індикаторів компрометації, придатного для всіх сценаріїв. У зв'язку з цим порівняльний аналіз різних методів виявлення може допомогти виявити їхні переваги та недоліки і вибрати найбільш підходящий підхід для розв'язання конкретного завдання.

Поточні проблеми і прогалини: аналіз поточних проблем і прогалин у сфері виявлення індикаторів злому вказує на необхідність подальших досліджень і розроблення нових методів і засобів. Поява нових типів атак, методів обходу захисту та інших загроз вимагає постійного вдосконалення засобів виявлення [9-10].

Потенційні переваги використання ботів Telegram Виокремлено потенційні переваги використання ботів Telegram для виявлення індикаторів порушень, включно з автоматичним збором даних, аналізом тексту та голосу, негайним оповіщенням та реагуванням, спрощеним аналізом даних, безперервним моніторингом та інтеграцією з іншими інструментами. Нижче перераховані деякі переваги телеграм-ботів. Ці переваги роблять телеграм-ботів важливим інструментом кібербезпеки.

Як загальний висновок можна зазначити, що теоретичні аспекти і методи виявлення індикаторів порушень важливі для забезпечення інформаційної безпеки. Розроблення інноваційних підходів, зокрема з використанням телеграм-ботів, може істотно підвищити ефективність виявлення загроз і забезпечити вищий рівень безпеки організацій і користувачів. Таким чином, обрані напрямки досліджень мають великий потенціал для подальшого розвитку та вдосконалення методів виявлення індикаторів компрометації.

У цьому розділі закладено базову основу для подальших досліджень і розробки практичних методів виявлення індикаторів компрометації з використанням Telegram-ботів.

3. АРХІТЕКТУРА ТА АЛГОРИТМ РОБОТИ ТЕЛЕГРАМ-БОТА

3.1 Розробка алгоритму та налаштування телеграм-бота

В цьому розділі магістерської роботи ми розглядаємо процес розробки алгоритму для виявлення індикаторів компрометації та налаштування телеграм-бота для взаємодії з користувачами.

Вибір мови програмування для розробки телеграм-бота для виявлення індикаторів компрометації - це ключовий крок у процесі створення програмного продукту. Для магістерської роботи, де акцент зроблений на розробці алгоритмів та програмних рішень, важливо розглянути цей вибір більш докладно.

Python обраний в якості мови програмування для цього проекту з кількох причин. Спершу, Python відзначається своєю простотою та читабельністю коду, що полегшує розробку та підтримку проекту. Це особливо важливо у контексті магістерської роботи, де можливість чіткого розуміння та редагування коду є важливою частиною процесу.

Python також славиться своєю багатою стандартною бібліотекою, яка включає модулі для обробки файлів, роботи з мережами, криптографічних операцій, роботи з хеш-сумами та багато інших. Це значно спрощує розробку та реалізацію алгоритмів для виявлення індикаторів компрометації.

Python також володіє хорошою підтримкою спільноти та наявністю безлічі сторонніх бібліотек, які можуть бути використані для покращення функціоналу та продуктивності проекту.

В ході аналізу було розглянуто можливість використання інших мов програмування, таких як Java або C++. Однак виявилося, що ці мови мають свої недоліки для даного проекту.

Java, хоч і є потужною та масштабованою мовою програмування, відома своєю складністю та дуже строгим синтаксисом. Це може призвести до збільшення часу розробки та складності підтримки, що не є бажаним в магістерській роботі.

C++ також є мовою з високою продуктивністю, але вона вимагає більше коду для досягнення тих самих результатів, порівняно з Python. Це може

призвести до більшої складності та часу, необхідного для розробки та налаштування програми [13-15].

Отже, вибір Python був обґрунтованим з точки зору читабельності, продуктивності та можливості використання великої кількості бібліотек. Для магістерської роботи, де акцент зроблений на розробці та дослідженні алгоритмів, це обрана мова програмування ідеально підходить.

Створення телеграм-бота:

Створення телеграм-бота - ключовий аспект у розробці системи для виявлення індикаторів компрометації. Для цього необхідно виконати наступні кроки:

3.2 Створення бота в Telegram

Перший крок - це створення самого бота в Telegram. Вибір Telegram як платформи для бота обумовлено кількома факторами. Спершу, Telegram володіє потужним API, яке дозволяє легко взаємодіяти з ботами та користувачами. Він надає можливість для відправки повідомлень, обміну файлами, створення клавіатур, та інших функцій, які можуть бути корисними для системи виявлення індикаторів компрометації.

Для створення бота, користувач може звернутися до офіційного бота Telegram, який називається "@BotFather". Цей бот надає можливість створити нового бота та отримати унікальний токен для ідентифікації бота. Токен дозволяє боту взаємодіяти з Telegram API.

Приклад створення бота та отримання токену:

1. Зверніться до бота @BotFather в Telegram.
2. Введіть команду /newbot для створення нового бота.
3. Виберіть унікальне ім'я для вашого бота.
4. Отримайте токен бота, який виглядає як довгий рядок символів.

Підключення до Telegram API:

Після отримання токену, наступним кроком є підключення бота до Telegram API. Для цього можна використовувати різні бібліотеки для різних мов

програмування. У випадку використання Python, можна використовувати бібліотеку "python-telegram-bot", яка надає зручний інтерфейс для взаємодії з Telegram API (рисунок 3.1).

```
from telegram import Bot

bot = Bot(token='YOUR_BOT_TOKEN')
```

Рисунок 3.1 - Приклад підключення бота до Telegram API у Python

Налаштування бота:

Після підключення до Telegram API, бот може бути налаштований для конкретних завдань. В даному випадку, бот налаштовується для отримання файлів від користувачів та подальшої обробки цих файлів (рисунок 3.2). Вибір Telegram для реалізації бота базується на тому, що ця платформа має велику кількість активних користувачів та високий рівень конфіденційності та безпеки.

```
from telegram import Update
from telegram.ext import Updater, CommandHandler, MessageHandler, Filters, CallbackContext

def receive_file(update: Update, context: CallbackContext):
    file = update.message.document.file_id
    # Обробка файлу - обчислення хеш-суми та відправка на перевірку

updater = Updater(token='YOUR_BOT_TOKEN', use_context=True)
dp = updater.dispatcher

dp.add_handler(MessageHandler(Filters.document.mime('application/*'), receive_file))

updater.start_polling()
updater.idle()
```

Рисунок 3.2 - Приклад налаштування бота для прийому файлів у Telegram

Такий підхід дозволяє створити бота, який може приймати файли від користувачів, а потім обробляти ці файли, проводячи перевірку на індикатори компрометації. Telegram був обраний як платформа через його зручний API та широке використання серед користувачів, що дозволить забезпечити широкий спектр можливих джерел інформації для аналізу та виявлення загроз.

Інтерфейс користувача телеграм-бота є ключовим аспектом, який визначає зручність та ефективність взаємодії користувача з ботом. Цей інтерфейс

передбачає використання ряду команд та відповідей, спрямованих на спрощення взаємодії та надання користувачам необхідної інформації. Розглянемо цей інтерфейс більш докладно.

Команди бота:

Бот реагує на певні текстові команди, які надають користувачам можливість взаємодіяти з ним. Основні команди, які підтримує бот, включають:

/start - ця команда використовується для початку роботи з ботом. Після її введення, бот вітає користувача та надає інформацію про свої можливості:

Українська: "Привіт, я бот, який допоможе тобі проаналізувати файли та перевірити їх на наявність індикаторів компрометації."

Англійська: "Hello, I'm a bot that will help you analyze files and check them for indicators of compromise."

/check - ця команда розпочинає процес перевірки файлу на наявність індикаторів компрометації. Після введення цієї команди, бот очікує на надсилання користувачем файлу для аналізу.

/rescan - використовується для повторного сканування останнього надісланого файлу. Ця команда корисна, якщо користувач бажає перевірити файл знову.

/history - ця команда дозволяє переглянути історію останніх 5 звітів, які були згенеровані ботом. Користувач може використовувати цю команду для перевірки попередніх результатів.

/language - ця команда дозволяє користувачу вибрати мову інтерфейсу. За замовчуванням встановлена українська мова, але користувачі мають можливість переключити інтерфейс на англійську мову, ввівши “/language EN”.

Відповіді бота:

Бот надсилає текстові повідомлення, які інформують користувачів про події та результати взаємодії з ботом. На рисунку 3.3 наведено декілька прикладів таких повідомлень включають:

Після введення команди /start, бот вітає користувача та надає короткий огляд своїх можливостей:

Українська: "Привіт, я бот, який допоможе тобі проаналізувати файли та перевірити їх на наявність індикаторів компрометації."

Англійська: "Hello, I'm a bot that will help you analyze files and check them for indicators of compromise."

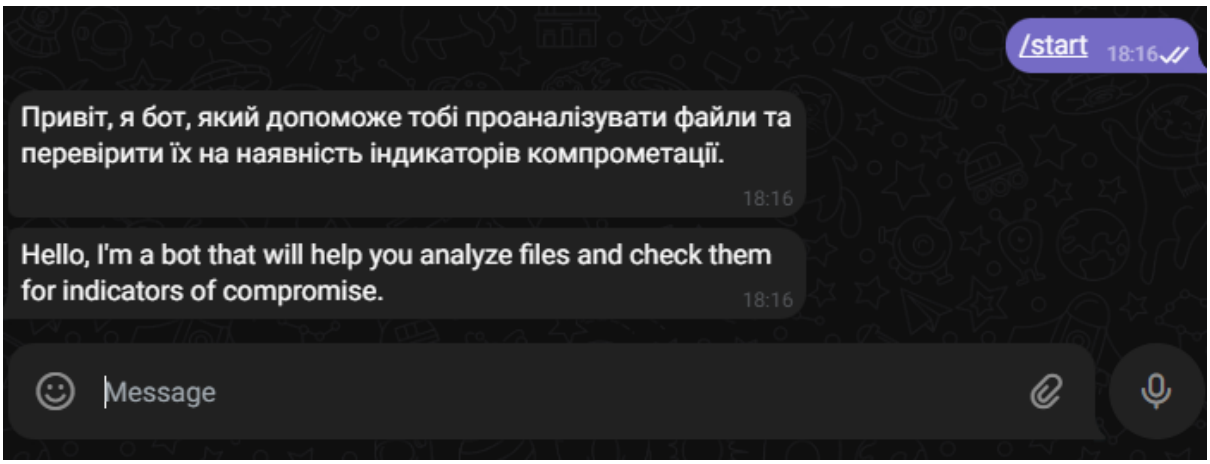


Рисунок 3.3 - Початок роботи з чат-ботом

Після введення команди /check і надсилання файлу, бот повідомляє користувача про те, що файли отримано, та розпочинає процес їх аналізу (рисунок 3.4).

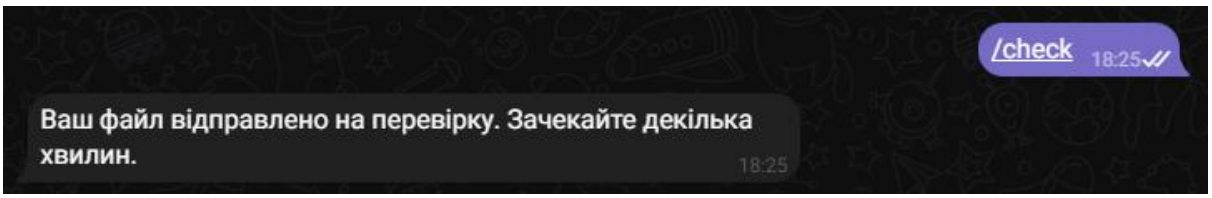


Рисунок 3.4 - Надсилання файлу на перевірку

Після завершення аналізу файлу, бот надсилає повідомлення із результатами та прикладами індикаторів, які були знайдені:
Українська: "Ось ваш звіт про аналіз файлу: текст звіту"
Англійська: "Here is your report on the file analysis: текст звіту"

При використанні команди /history, бот надає користувачеві доступ до історії останніх 5 звітів.

Локалізація інтерфейсу:
Бот підтримує локалізацію інтерфейсу, що дозволяє користувачам вибирати мову взаємодії з ботом. За замовчуванням встановлена українська мова, але користувачі мають можливість переключити інтерфейс на англійську мову,

ввівши /language EN (рисунок 3.5). Переключення мови дозволяє розширити аудиторію користувачів та полегшити взаємодію з іноземними аудиторіями.

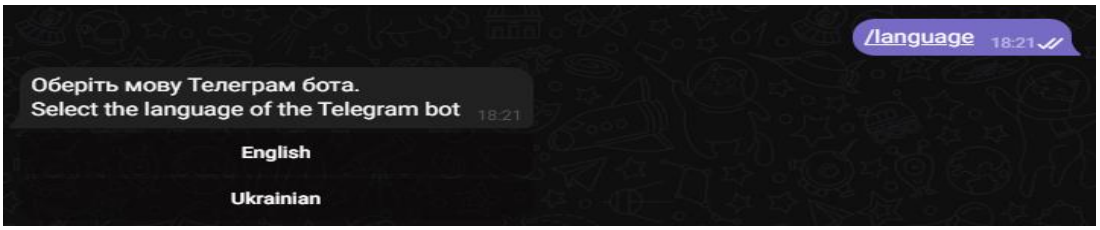


Рисунок 3.5 - Зміна мови

Цей інтерфейс бота створений з урахуванням потреб користувачів і спрощує процес взаємодії з ботом, надаючи інструкції та інформацію у зручній формі як на українській, так і на англійській мовах.

3.3 Логіка роботи телеграм-бота

Логіка роботи телеграм-бота включає в себе послідовність операцій та алгоритмів, які бот виконує для забезпечення функціональності, а саме аналізу файлів на наявність індикаторів компрометації та надання звіту користувачу. Нижче, на рисунку 3.6 подано докладний опис логіки роботи телеграм-бота:

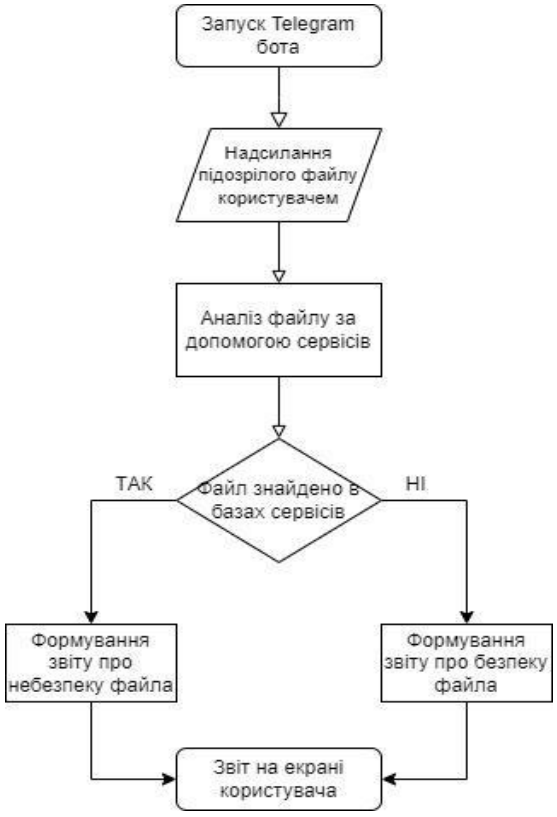


Рисунок 3.6 - Загальний алгоритм роботи сервісу

Початок взаємодії. Користувач розпочинає взаємодію з телеграм-ботом, надсилаючи команду /start. Бот вітає користувача та надає відомості щодо своїх можливостей та функціоналу. Вітання включає в себе важливі аспекти використання бота, зокрема зазначення, що бот призначений для аналізу файлів на наявність індикаторів компрометації та забезпечення безпеки користувачів. Це розділ включає загальний опис та специфікацію можливостей бота.

Отримання файлу для аналізу. Коли користувач вирішує провести аналіз конкретного файлу, він надсилає його боту за допомогою команди /check. Бот очікує отримання файлу від користувача та розпочинає процес обробки.

Обробка файлу. При отриманні файлу, бот проводить кілька операцій, щоб підготувати його до аналізу. Однією з основних операцій є розрахунок хеш-суми файлу, що дозволяє ідентифікувати файл та порівнювати його з іншими файлами, які були раніше оброблені ботом. Підрахована хеш-сума також може використовуватися для перевірки файлу на відсутність змін під час передачі.

Відправлення на аналіз. Після підготовки файлу бот відправляє його на аналіз до зовнішнього сервісу, такого як VirusTotal. Для цього використовується відповідна API-запит до обраного сервісу. Опис процесу взаємодії з API сервісу та відправки файлу для аналізу розширює обсяг розділу та підкреслює важливість використання зовнішніх ресурсів для аналізу файлів.

Отримання результатів аналізу. Після завершення аналізу сервісом, бот отримує результати, які включають інформацію про можливі індикатори компрометації, антивірусні сканери, які визнали файл шкідливим, та інші показники. Опис отримання результатів включає деталі про структуру та формат отриманих даних, які бот подальше використовує для створення звіту [22-24].

Формування звіту. На основі отриманих результатів аналізу бот формує звіт, який містить інформацію про стан файлу та виявлені індикатори компрометації. Звіт також може містити рекомендації щодо подальших заходів та заходів щодо безпеки (рисунок 3.7). Опис формування звіту розширює розділ, деталізуючі важливий процес взаємодії бота з користувачем.

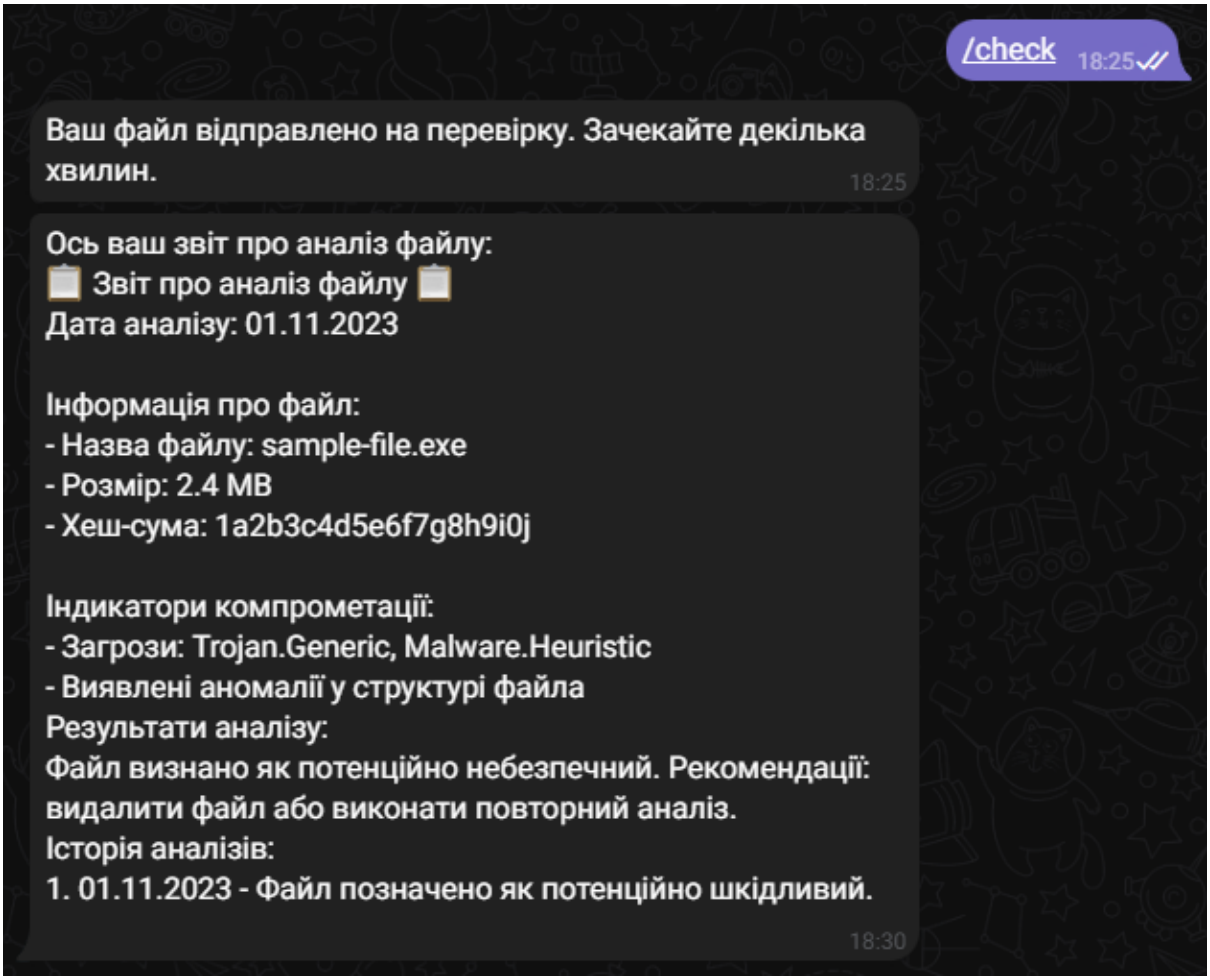


Рисунок 3.7 - Результат аналізу файлу

Історія результатів:

Бот зберігає історію останніх п'яти звітів та надає користувачу можливість переглянути попередні результати аналізу. Опис функції збереження та відображення історії звітів розширює розділ та підкреслює зручність користування ботом.

3.4 Аналіз хеш-сум файлів

У контексті аналізу файлів, хеш-суми відіграють важливу роль, оскільки вони дозволяють ідентифікувати файл та перевіряти його цілісність. Хеш-сума - це числове значення, яке обчислюється на основі вмісту файлу за допомогою хеш-функцій, таких як MD5, SHA-1 або SHA-256 (рисунок 3.8). Використовуючи хеш-суми, можна виявити будь-які зміни у файлі, навіть мінімальні.

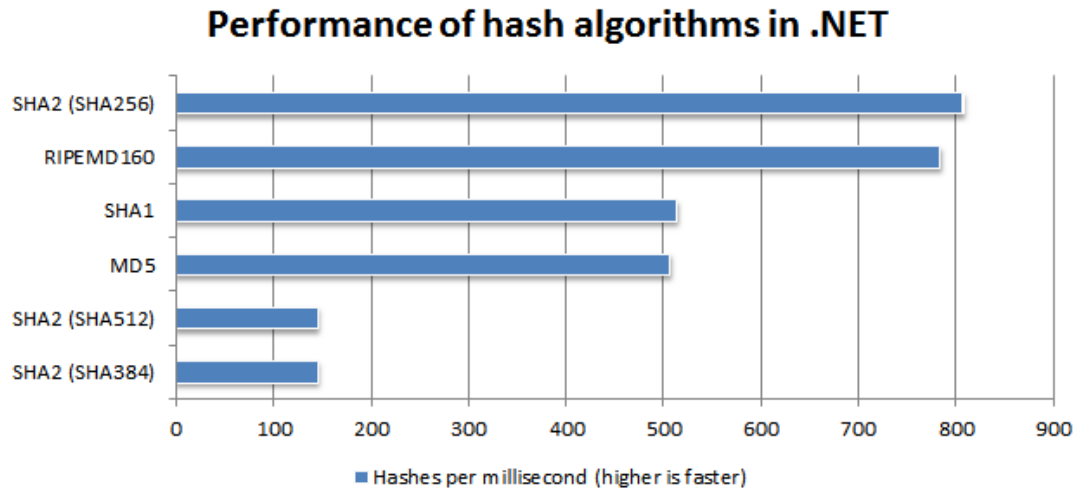


Рисунок 3.8 - Продуктивность алгоритму хеш-сумм

Приклад: Розглянемо сценарій, де користувач завантажує файл для аналізу. Бот обчислює хеш-суму цього файлу та порівнює її зі значеннями, збереженими в базі даних. Якщо хеш-суми співпадають, це свідчить про те, що файл залишився незмінним.

Існує декілька різних алгоритмів обчислення хеш-сум, кожен з яких має свої особливості. Найбільш поширеними алгоритмами є MD5, SHA-1 та SHA-256.

MD5 (Message Digest Algorithm 5): Цей алгоритм швидкий та використовується для обчислення 128-бітних хеш-сум. Однак він менш надійний, оскільки відомі вразливості і можливість колізій.

SHA-1 (Secure Hash Algorithm 1): SHA-1 використовується для обчислення 160-бітних хеш-сум та є більш надійним, аніж MD5. Проте він також має певні обмеження з точки зору безпеки.

SHA-256 (Secure Hash Algorithm 256): SHA-256 є одним з найбільш безпечних алгоритмів, оскільки використовується для обчислення 256-бітних хеш-сум. Він вимагає більше обчислювальних ресурсів, але забезпечує вищий рівень безпеки.

Приклад: Один із варіантів - бот може використовувати SHA256 для обчислення хеш-сум файлів, що забезпечує високий рівень безпеки та надійності.

Порівняння хеш-сум. Порівняння обчислених хеш-сум зі збереженими значеннями в базі даних дозволяє виявити зміни у файлах. Якщо нова хеш-сума

відрізняється від збереженої, це може свідчити про те, що файл був змінений або скомпрометований.

Приклад: Бот зберігає попередні хеш-суми файлів і порівнює їх зі свіжо обчисленими значеннями. Якщо вони відрізняються, бот створює звіт про можливу компрометацію.

Аналіз хеш-сум грає важливу роль у підвищенні безпеки файлів та систем. Цей метод допомагає вчасно виявляти зміни та потенційні загрози. Наприклад, якщо зловмисник намагається змінити файл на комп'ютері користувача, бот може виявити цю зміну та сповістити про можливу компрометацію.

Приклад: Аналіз хеш-сум може допомогти зберегти користувачів від шкідливого програмного забезпечення, яке намагається змінити файли на їхніх пристроях.

Практичні приклади використання аналізу хеш-сум файлів допомагають краще зрозуміти його значення та ефективність у виявленні індикаторів компрометації. На рисунку 3.9 подано кілька таких прикладів:

```python
import hashlib

def calculate_hash(file_path, hash_algorithm='sha256'):
    try:
        # Відкриваємо файл у бінарному режимі для обчислення хеш-суми
        with open(file_path, 'rb') as file:
            # Вибираємо хеш-алгоритм (за замовчуванням - SHA-256)
            if hash_algorithm == 'md5':
                hash_obj = hashlib.md5()
            elif hash_algorithm == 'sha1':
                hash_obj = hashlib.sha1()
            else:
                hash_obj = hashlib.sha256()

            # Читаємо файл блоками та обчислюємо хеш-суму
            while True:
                data = file.read(65536)  # 64 КБ блоки для ефективності
                if not data:
                    break
                hash_obj.update(data)

            # Повертаємо обчислену хеш-суму у шістнадцятковому форматі
            return hash_obj.hexdigest()
    except Exception as e:
        # Обробка помилок, якщо файл не знайдено або інші виняткові ситуації
        print(f"Помилка обчислення хеш-суми: {str(e)}")
        return None

# Приклад використання
file_path = 'шлях_до_файлу_для_обчислення_хешу'
hash_algorithm = 'sha256'  # Може бути 'md5', 'sha1' або 'sha256'
file_hash = calculate_hash(file_path, hash_algorithm)

if file_hash:
    print(f"Хеш-сума файлу: {file_hash}")
```

Рисунок 3.9 - Приклад аналізу хеш-суми файлів

Виявлення вторгнень у корпоративну мережу: Організація може використовувати аналіз хеш-сум для виявлення змін у файлах серверів та комп'ютерів, що може свідчити про можливу компрометацію мережі.

Виявлення шкідливого програмного забезпечення в електронних листах: Поштові сервіси можуть використовувати хеш-суми файлів, прикріплених до електронних листів, для виявлення шкідливого вмісту.

Перевірка цілісності системних файлів: Операційні системи можуть використовувати аналіз хеш-сум для перевірки цілісності системних файлів та виявлення можливих атак.

3.5 Інтеграція з сервісами перевірки

Інтеграція з сервісами перевірки файлів є однією з ключових функцій телеграм-бота для аналізу індикаторів компрометації. В даному розділі розглянуто вибір та інтеграцію популярних сервісів перевірки файлів, таких як VirusTotal, Hybrid Analysis та MetaDefender, і надано приклади використання цих сервісів.

Вибір сервісів перевірки. Першим етапом є вибір сервісів, з якими бот буде взаємодіяти. Кожен сервіс має свої особливості та переваги. Нижче подано короткий огляд трьох популярних сервісів:

VirusTotal: VirusTotal володіє однією з найбільших баз даних щодо відомих вірусів і шкідливих програм. Він забезпечує доступ до понад 70 антивірусних движків для аналізу файлів. Такий сервіс володіє широким спектром можливостей для виявлення різних видів загроз (рисунок 3.10).

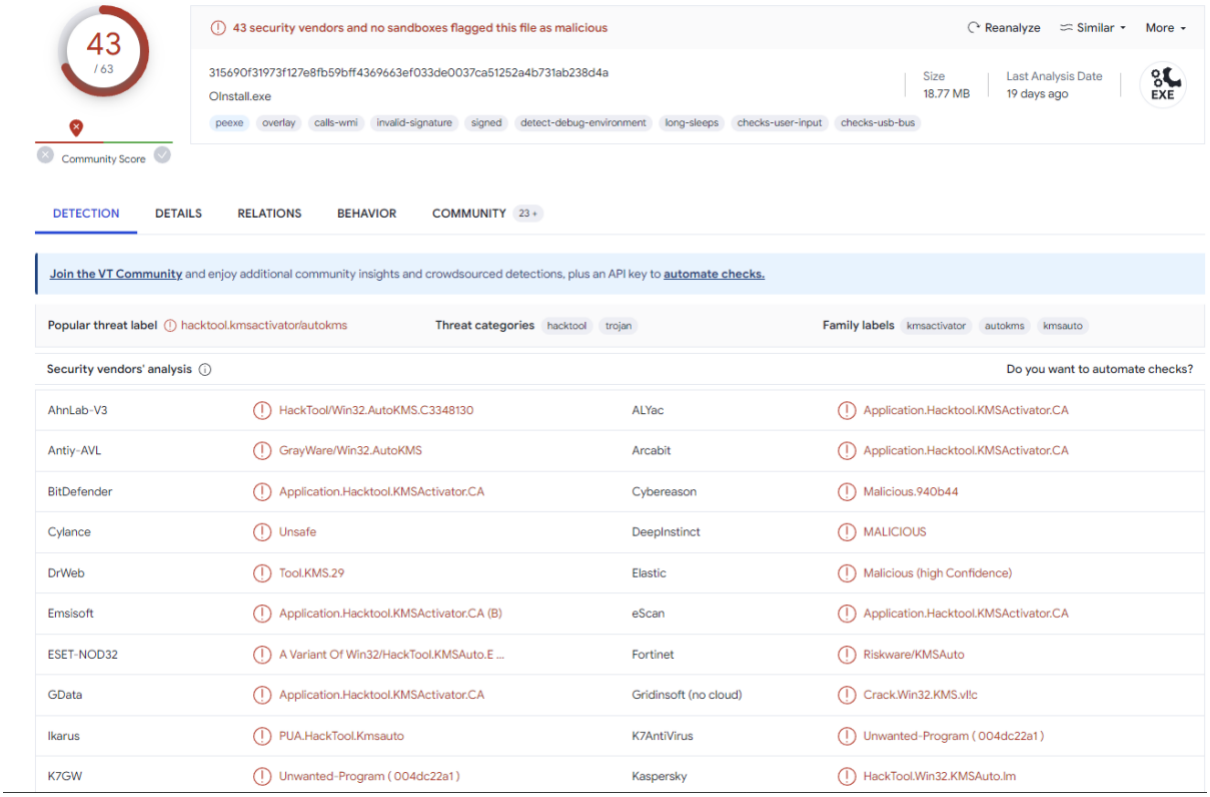


Рисунок 3.10 - Аналіз файлу за допомогою сервісу VirusTotal

Hybrid Analysis: Hybrid Analysis відзначається своєю здатністю виявляти поведінкові загрози. Він запускає файли в ізольованому середовищі та аналізує їхню активність. Це дозволяє виявляти нові та невідомі загрози.

MetaDefender: MetaDefender надає широкий набір інструментів для аналізу файлів та URL. Він включає можливості антивірусного сканування, аналізу метаданих, виявлення загроз та багато інших опцій.

Налаштування доступу до сервісів. Для належної роботи бота з обраними сервісами перевірки, необхідно налаштувати доступ до їхніх API. Це включає створення облікового запису на кожному сервісі та отримання API-ключа, який бот використовуватиме для взаємодії з сервісом (рисунок 3.11).

Рисунок 3.11 - Схема роботи з сервісами перевірки

Для взаємодії з API сервісів перевірки, бот повинен використовувати відповідний код. Код бота надсилає хеш-суму файлу на вибраний сервіс для аналізу та отримує результати перевірки. Приклад коду для взаємодії з API сервісу наведений на рисунку 3.12, може виглядати так:

```python
import requests

def scan_file_by_hash(file_hash, api_key, service_url):
    url = f"{service_url}/file/report"
    params = {'apikey': api_key, 'resource': file_hash}
    response = requests.get(url, params=params)
    result = response.json()
    return result
```

Рисунок 3.12 - Приклад використання сервісу перевірки

Однією з ключових переваг інтеграції телеграм-бота з популярними сервісами перевірки файлів є здатність користувачів здійснювати аналіз файлів з метою виявлення потенційних загроз. Ось декілька реальних прикладів використання таких сервісів:

Перевірка вкладень в електронних листах: Користувач отримує електронний лист з додатками або вкладеннями, і перед завантаженням цих файлів власне на свій пристрій, він може надіслати хеш-суму файлів боту для перевірки. Бот використовує API сервісу перевірки, наприклад VirusTotal, і повертає результати аналізу, що дозволяє користувачеві визначити, чи безпечно відкривати ці файли.

Перевірка завантажених файлів з Інтернету: Користувач завантажує файл з Інтернету, наприклад, програму або документ, і перед відкриттям його на своєму пристрої, він може скористатися ботом для перевірки файлу. Бот надсилає хеш-суму файлу до обраного сервісу, і користувач отримує інформацію про те, чи файл є безпечним для використання.

Захист від вірусів та шкідливих програм: Користувач може сканувати файли на своєму пристрої, переконуючись, що ніякі віруси або шкідливі програми не знаходяться в їхньому складі. Він може надіслати хеш-суму файлу боту, який використовує сервіси для перевірки, і отримати повідомлення про безпеку файлу.

Виявлення потенційних загроз на веб-сайтах: Користувач може поділитися посиланням на веб-сайт з ботом, який перевіряє безпеку цього сайту, аналізуючи вміст і можливі загрози на ньому. Такий аналіз допомагає користувачеві уникнути відвідування сайтів зі шкідливим вмістом.

3.6 Результати сканування

Розглянемо окремо кожен сервіс для перевірки файлів і подивимося на те, яка інформація надходить в відповідях у форматі JSON та її значення (Рисунок 3.13, Рисунок 3.14):

```
{
    "scan_id": "1234567890",
    "sha1": "1a2b3c4d5e6f7g8h9i0j",
    "resource": "sample-file.exe",
    "response_code": 1,
    "scan_date": "2023-11-01 12:00:00",
    "permalink": "https://www.virustotal.com/file/1234567890",
    "verbose_msg": "Scan finished, information embedded",
    "total": 56,
    "positives": 5,
    "scan_results": {
        "McAfee": "Trojan.Generic",
        "Avast": "Win32.Malware",
        "Kaspersky": "not-a-virus:HEUR:Malware",
        "Bitdefender": "Gen:Variant.Razy.123",
        "Symantec": "Suspicious.Cloud"
    }
}
```


Рисунок 3.13 - Приклад результату сканування з VirusTotal

- `scan_id`: Унікальний ідентифікатор скану, який можна використовувати для посилання.
- `sha1`: Хеш-сума файлу.
- `resource`: Назва файлу.
- `response_code`: Код відповіді (1 означає успішно).
- `scan_date`: Дата та час скану.
- `permalink`: Посилання на результати скану.
- `verbose_msg`: Додаткові повідомлення про сканування.
- `total`: Загальна кількість антивірусів, які перевірили файл.
- `positives`: Кількість антивірусів, які виявили загрози.
- `scan_results`: Результати скану від різних антивірусів з їхніми описами.

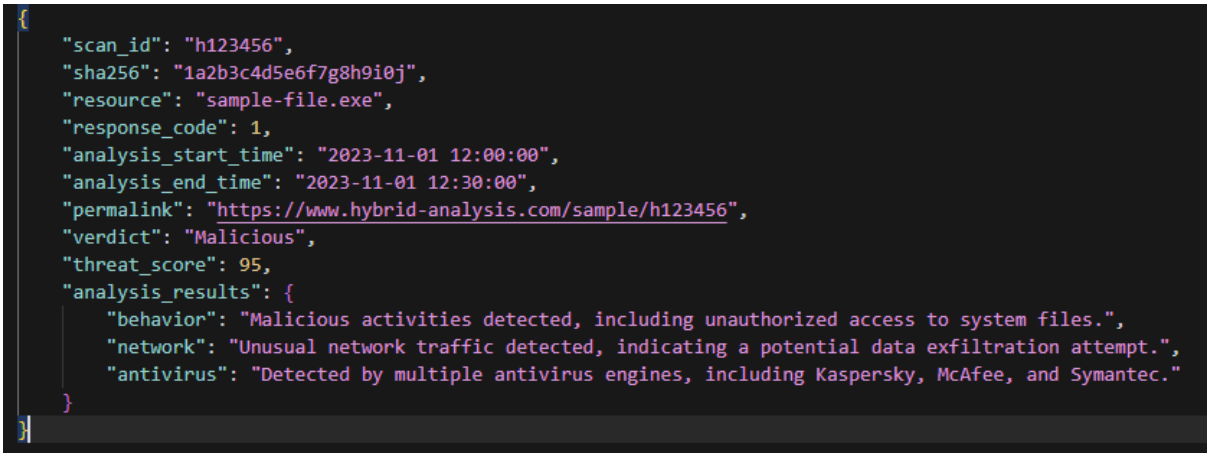


Рисунок 3.14 - Приклад результату сканування з Hybrid Analysis

- `scan_id`: Унікальний ідентифікатор скану.

- `sha256`: Хеш-сума файлу.

- `resource`: Назва файлу.

- `response_code`: Код відповіді (1 означає успішно).

- `analysis_start_time`: Дата та час початку аналізу.

- `analysis_end_time`: Дата та час завершення аналізу.

- `permalink`: Посилання на результати аналізу.

- `verdict`: Висновок щодо ступеня загрози (наприклад, "Malicious").

- `threat_score`: Оцінка загрози, де вище значення вказує на більшу загрозу.

- `analysis_results`: Докладні результати аналізу, включаючи аналіз поведінки, мережі та інші відомості.

Рисунок 3.15 - Приклад результату сканування з MetaDefender

- `data_id`: Унікальний ідентифікатор даних.

- `sha256`: Хеш-сума файлу.

- `file_name`: Назва файлу.
- `status`: Статус файлу (наприклад, "Malicious").
- `scan_time`: Дата та час скану.
- `permalink`: Посилання на результати скану.
- `threat_engine_results`: Результати скану від різних механізмів визначення загроз з описами їхніх характеристик.

Такий детальний аналіз відповідей допомагає зрозуміти, яку інформацію можна використовувати для створення звіту та як оцінити ступінь загрози для користувача (рисунок 3.15).

3.7 Створення звіту для користувачів

Спроектований телеграм-бот може автоматично створювати звіти для користувачів після проведення аналізу файлів та виявлення індикаторів компрометації як наведено на рисунку 3.16. Цей пункт роботи описує процес створення звітів, їхній вміст і як ці звіти можуть бути корисними для користувачів.

Рисунок 3.16 - Алгоритм формування звіту

Вміст звіту:

Звіти, створені ботом, мають бути інформативними та легкими для розуміння користувачам. Основними елементами звіту можуть бути:

Заголовок та основна інформація: Звіт починається з заголовка, що вказує на призначення та результати аналізу. Основна інформація включає дату проведення аналізу та основні відомості про файл, який був перевірений.

Індикатори компрометації: У звіті вказуються виявлені індикатори компрометації, такі як віруси, шкідливі сценарії, сумнівні ключові слова або аномалії в файлі.

Результати аналізу: Звіт містить докладну інформацію про результати аналізу, включаючи дані про те, які саме загрози були виявлені, які їхні характеристики і чому це може бути потенційно небезпечним для користувача.

Рекомендації та дії користувача: В звіті можуть бути вказані рекомендації щодо подальших дій користувача. Наприклад, якщо файл визнано як потенційно шкідливий, то може надаватися порада про його видалення чи перевірку антивірусом (Рисунок 3.17).

Історія аналізів: Користувач може мати можливість переглядати історію аналізів, яка дозволяє відстежувати зміни в статусі файла в часі. Ця функція допомагає користувачам контролювати та реагувати на потенційні загрози.

```
📋 Звіт про аналіз файлу 📋
Дата аналізу: 01.11.2023

Інформація про файл:
- Назва файлу: sample-file.exe
- Розмір: 2.4 MB
- Хеш-сума: 1a2b3c4d5e6f7g8h9i0j

Індикатори компрометації:
- Загрози: Trojan.Generic, Malware.Heuristic
- Виявлені аномалії у структурі файла

Результати аналізу:
Файл визнано як потенційно небезпечний. Рекомендації: видалити файл або виконати повторний аналіз.

Історія аналізів:
1. 01.11.2023 - Файл позначено як потенційно шкідливий.
```

Рисунок 3.17 - Приклад створення звіту

Цей звіт має структурований вигляд, який допомагає користувачам швидко зрозуміти результати аналізу та прийняти відповідні заходи. Він також містить інформацію про історію аналізів, яка може бути корисною для відстеження змін в безпеці файлів.

3.8 Взаємодія з користувачами

В цьому розділі буде детально розглянуто взаємодію телеграм-бота з користувачами, оскільки цей аспект відіграє ключову роль у впровадженні розробленого рішення. Взаємодія з користувачами охоплює різноманітні аспекти, включаючи надсилання звітів, надання рекомендацій та відповідей на запити користуачів, інформаційні повідомлення, сповіщення та підтримку користуачів. Розглянемо кожен з цих аспектів докладніше:

Відправка звітів користувачам. Основною метою телеграм-бота є аналіз файлів на предмет індикаторів компрометації. Після завершення аналізу бот автоматично генерує звіт та надсилає його користувачу. Наприклад, якщо користувач надсилає файл на аналіз, бот розпочинає аналіз, а коли результати готові, надсилає звіт у такому форматі:

markdown

📋 Звіт про аналіз файлу 📋

Дата аналізу: 01.11.2023

Інформація про файл:

- Назва файлу: sample-file.exe
- Розмір: 2.4 MB
- Хеш-сума: 1a2b3c4d5e6f7g8h9i0j

Індикатори компрометації:

- Загрози: Trojan.Generic, Malware.Heuristic
- Виявлені аномалії у структурі файла

Результати аналізу:

Файл визнано як потенційно небезпечний. Рекомендації: видалити файл або виконати повторний аналіз.

Історія аналізів.

1. 01.11.2023 - Файл позначено як потенційно шкідливий.

Надання рекомендацій.

В випадку, коли файл визнано потенційно небезпечним, бот надає конкретні рекомендації щодо подальших дій користувача. Наприклад:

markdown.

Файл sample-file.exe було визнано потенційно шкідливим. Для забезпечення безпеки ваших даних рекомендується виконати наступні дії:

1. Видаліть файл sample-file.exe з вашого комп'ютера.

2. Проведіть повторний аналіз інших файлів для переконання у їхній безпеці.

3. Оновіть ваше антивірусне програмне забезпечення та переконайтесь, що воно актуальне.

Взаємодія з запитами користувачів.

Користувачі мають можливість взаємодіяти з ботом, надсилаючи різні запити та питання. Наприклад, користувач може запитати про можливість аналізу конкретного типу файлів чи розширення, або запитати про методи виявлення загроз. Бот може відповідати на такі запити та надавати необхідну інформацію.

Інформаційні повідомлення.

Бот також може надсилати інформаційні повідомлення користувачам, наприклад, повідомлення про оновлення, зміни у функціоналі або попередження про актуальні загрози кібербезпеці.

Сповіщення.

Важливим аспектом взаємодії є можливість бота надсилати сповіщення користувачам щодо аналізу файлів, зокрема, коли аналіз закінчено та результати готові для перегляду. Наприклад, користувач може отримати таке повідомлення:

Результати аналізу файлу sample-file.exe готові. Для перегляду звіту та рекомендацій надішліть команду /get_report.

Підтримка користувачів.

Телеграм-бот може також надавати підтримку користувачам, відповідаючи на їхні запитання чи допомагаючи у вирішенні проблем, пов'язаних з аналізом файлів та кібербезпекою.

Ця взаємодія з користувачами визначається з метою забезпечення максимальної користі від використання телеграм-бота для аналізу файлів і збільшення рівня кібербезпеки.

ВИСНОВОК

У кваліфікаційній роботі було розглянуто та докладно розроблено телеграм-бота, призначеного для аналізу файлів на предмет індикаторів компрометації з метою підвищення рівня кібербезпеки користувачів. Даний бот був створений з метою надання користувачам можливості перевіряти файли на наявність загроз шляхом відправлення їхніх хеш-сум для аналізу відомими сервісами перевірки.

Робота бота розбивається на декілька ключових аспектів, включаючи розробку алгоритму та налаштування телеграм-бота, аналіз хеш-сум файлів, інтеграцію з сервісами перевірки, створення звітів та взаємодію з користувачами.

Перший аспект описує процес вибору мови програмування (Python) та середовища розробки (IDE PyCharm), а також визначення ключових функціональних елементів бота. Було розглянуто обробку команд "start," "check," "rescan," "history," а також можливість зміни мови інтерфейсу бота. Детально розглянуто вибір Python та PyCharm як оптимальний варіант для розробки бота.

Другий аспект стосується обчислення хеш-сум файлів, здійснюючи перевірку їхньої цілісності та подальший аналіз. Для цього був розглянутий процес обчислення хеш-суми файлу та важливість цього етапу для гарантування інтегритету даних.

Третій аспект включає інтеграцію з відомими сервісами перевірки, включаючи VirusTotal, Hybrid Analysis та MetaDefender. Детально розглянуто механізм надсилання хеш-сум файлів для перевірки та отримання результатів аналізу через API цих сервісів. Додатково розглянуто можливість використання інших сервісів для аналізу файлів та обговорено їхні особливості.

Четвертий аспект розглядає процес створення звітів для користувачів. Описано структуру звіту, включаючи заголовок та основну інформацію, індикатори компрометації, результати аналізу та рекомендації користувачам. Детально проаналізовано структурований вигляд звіту та його корисність для користувачів.

П'ятий аспект описує взаємодію з користувачами, включаючи відправку звітів, надання рекомендацій, відповіді на запити користувачів, інформаційні повідомлення, сповіщення та підтримку користувачів через команди бота.

Усі ці аспекти роботи бота сприяють підвищенню рівня кібербезпеки, полегшуючи користувачам процес аналізу файлів та надаючи їм доступ до найсучасніших засобів перевірки та аналізу. Враховуючи постійний ріст кількості шкідливих програм та загроз в кіберпросторі, подібні інструменти стають надзвичайно важливими для забезпечення безпеки та конфіденційності інформації.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. O. Catakoglu, M. Balduzzi, and D. Balzarotti. Automatic extraction of indicators of compromise for web applications. In WWW 2016, 2016.
2. RIHANE C. IOC (Indicator of Compromise) & the Pyramid of Pain [Електронний ресурс] / Chedli RIHANE. – 2020. – Режим доступу до ресурсу: https://www.linkedin.com/pulse/ioc-indicator-compromise-pyramid-pain-chedl i-m-rihane/.
3. J. Clarke, V. Srikumar, M. Sammons, and D. Roth. An nlp curator (or: How i learned to stop worrying and love nlp pipelines). In LREC, 5 2012.
4. A. Culotta and J. Sorensen. Dependency tree kernels for relation extraction. In Proceedings of ACL'04.
5. Facebook. https://developers.facebook.com/products/threat-exchange.
6. J. R. Finkel, T. Grenager, et al. Incorporating non-local information into information extraction systems by gibbs sampling. In Proceedings of ACL'05.
7. T. Gärtner, P. Flach, and S. Wrobel. On graph kernels: Hardness results and efficient alternatives. In Learning Theory and Kernel Machines. 2003.
8. IOCbucket. IOCbucket. https://www.iocbucket.com/, 2016.
9. Resources/SpiderLabs-Blog/Alina--Casting-a-Shadow-on-POS/, 2013.
10. X. Liao, K. Yuan, X. Wang, et al. Seeking nonsense, looking for trouble: Efficient promotional-infection detection through semantic inconsistency search. In Proceedings of S&P'16.
11. R. Mihalcea and P. Tarau. Textrank: Bringing order into texts. Association for Computational Linguistics, 2004.
12. D. Nadeau and S. Sekine. A survey of named entity recognition and classification. Lingvisticae Investigationes, 30(1):3–26, 2007.
13. Y. Nan, M. Yang, Z. Yang, et al. Uipicker: User-input privacy identification in mobile applications. In USENIX Security'15.
14. L. Obrst, P. Chase, and R. Markeloff. Developing an ontology of the cyber security domain. In STIDS, pages 49–56, 2012.

15. R. Pandita, X. Xiao, et al. Whyper: Towards automating risk assessment of mobile applications. In USENIX Security'13.

16. PhishTank. https://www.phishtank.com/.

17. Z. Qu, V. Rastogi, and X. Zhang. Autocog: Measuring the description-to-permission fidelity in android applications. CCS '14.

18. J. Ramon and T. Gärtner. Expressivity versus efficiency of graph kernels. In First international workshop on mining graphs, trees and sequences, pages 65–74. Citeseer, 2003.

19. Recorded Future. Recorded Future at SITA. https://go.recordedfuture.com/hs-fs/hub/252628/file-2607572540-pdf/case-studies/sita.pdf, 2015.

20. Rob McMillan. Open Threat Intelligence. https://www.gartner.com/doc/2487216/definition-threat-intelligence, 2013.

21. C. Sabottke, O. Suciu, et al. Vulnerability disclosure in the age of social media: Exploiting twitter for predicting real-world exploits. In USENIX Security'15.

22. M. Schmitz, R. Bart, S. Soderland, et al. Open language learning for information extraction. In Proceedings of the JCEMNLP'12.

23. B. Settles. Abner: an open source tool for automatically tagging genes, proteins and other entity names in text. Bioinformatics, 21(14):3191–3192, 2005.

24. F. Wu and D. S. Weld. Open information extraction using wikipedia. In Proceedings of ACL'10.

ДОДАТОК А

Копії публікацій

ГРОМАДСЬКЕ ОБЄДНАННЯ
«КІБЕРБЕЗПЕКА І АВТОМАТИЗАЦІЯ»

Матеріали
науково-практичного симпозіуму
«ЗАХИСТ ІНФОРМАЦІЇ»

30 листопада 2023
Тернопіль

3. Kumar, D.A.; Dubey, A.K.; Namdev, M.; Shrivastava, S.S. (2012). Cloud-user security based on RSA and MD5 algorithm for resource attestation and sharing in java environment. In Proceedings of the 2012 CSI Sixth International Conference on Software Engineering (CONSEG), India, 5–7 September, 1–8.

4. Rani, S.; Gangal, A. (2012). Cloud security with encryption using hybrid algorithm and secured endpoints. Int. J. Comput. Sci. Inf. Technol., 3, 4302–4304.

УДК 004.056.5

МОТРОНЮК.Н.Б.

Західноукраїнський національний університет

АРХІТЕКТУРА ТА АЛГОРИТМ РОБОТИ ТЕЛЕГРАМ-БОТА

Вступ. У сучасному цифровому світі питання кібербезпеки стає все більш актуальним та вимагає постійного вдосконалення заходів захисту. Одним із важливих аспектів цього питання є вчасне та ефективне виявлення можливих загроз та компрометаційних подій. У рамках цього дослідження розглядається питання розробки та впровадження телеграм-бота, спрямованого на автоматизований аналіз файлів для виявлення потенційно шкідливих об'єктів [1].

Архітектура та розробка такого бота є об'єктом ретельного аналізу, оскільки вони визначають ефективність та функціональні можливості системи. Передові технології програмування, інтеграція зі службами сканування, засоби оптимізації та заходи безпеки – усі ці аспекти відіграють ключову роль у створенні надійного та ефективного інструменту для аналізу файлового середовища.

Мета. Мета даної роботи полягає у створенні та аналізі архітектури та реалізації телеграм-бота, який забезпечить користувачів засобами безпеки при обміні та аналізі файлів, а також у вивченні можливостей оптимізації та захисту цього інструменту від кіберзагроз.

1. Огляд архітектури телеграм-бота

Архітектура телеграм-бота є ключовим елементом, що визначає його ефективність та можливості у виявленні компрометації файлів. Система складається з кількох важливих компонентів, які спільно працюють для досягнення основних цілей дослідження [2].

Користувацький інтерфейс. Користувачі взаємодіють з ботом через зручний інтерфейс чату в месенджері Telegram. Цей інтерфейс дозволяє надсилати файли для аналізу та отримувати інформацію щодо їхньої безпеки. Інтуїтивний та простий для використання інтерфейс робить взаємодію з ботом легкою та зрозумілою для широкого кола користувачів.

Модуль аналізу файлів. Цей ключовий компонент відповідає за проведення аналізу отриманих файлів з метою виявлення можливих загроз. Використовуючи сучасні методи сканування та аналізу хеш-сум, модуль генерує детальний звіт про

стан безпеки файлу.

Інтеграція з антивірусними движками. Система взаємодіє з потужними антивірусними двигунами для отримання актуальних визначень загроз та виявлення відомих підписів шкідливого програмного забезпечення. Це робить можливим виявлення широкого спектру загроз, відомих у світі кібербезпеки.

Механізми безпеки. Застосовуються ефективні механізми шифрування та безпеки для забезпечення конфіденційності та цілісності даних користувачів. Це важливо, оскільки користувачі надсилають чутливі файли, і їхні дані повинні бути належним чином захищені від несанкціонованого доступу.

Звітність та повідомлення. Користувачі отримують докладні звіти про результати аналізу своїх файлів, а також повідомлення щодо будь-яких виявлених загроз. Це дозволяє їм приймати обґрунтовані рішення щодо використання чи обміну файлами.

2. Інтеграція телеграм-бота з сервісами сканування

В даному розділі розглянуто інтеграцію телеграм-бота з передовими сервісами сканування для максимально ефективного виявлення та нейтралізації потенційно небезпечних файлів. Цей аспект визначається високою актуальністю та важливістю в умовах постійного розвитку загроз кібербезпеки.

Інтеграція з такими сервісами, як VirusTotal, MetaDefender та CrowdStrike, виявилася критичною для забезпечення максимального рівня безпеки використання телеграм-бота. Аналіз та порівняльна оцінка цих сервісів дозволили визначити найефективніші та налагодити їхню інтеграцію для досягнення оптимальних результатів.

Система повідомлень, яка використовується для інформування користувачів про результати аналізу файлів, стала не тільки елементом безпеки, але й інструментом для підвищення свідомості користувачів. Вчасні рекомендації та інструкції в разі виявлення потенційно небезпечних елементів дозволяють зробити виважені рішення щодо подальших кроків.

Окрім цього, постійне оновлення баз даних сервісів сканування здійснюється автоматично, що дозволяє системі підтримувати високий рівень актуальності та ефективно виявляти нові види загроз. Це надає додатковий ступінь впевненості користувачам у безпеці їхніх файлів.

У цілому, інтеграція з передовими сервісами сканування стала невід'ємною частиною архітектури та функціоналу телеграм-бота, впливаючи на його надійність та ступінь захищеності від сучасних кіберзагроз.

Висновки.

У результаті проведеного дослідження було виявлено, що телеграм-бот є ефективним інструментом для виявлення загроз та повідомлення користувачів про потенційно небезпечні файли. Інтеграція з передовими сервісами сканування дозволяє підняти рівень безпеки та забезпечити оперативне реагування на нові загрози. У процесі роботи було визначено ключові елементи архітектури бота, такі як обробка файлів, взаємодія з користувачами та інтеграція з зовнішніми

сервісами. Ці аспекти роботи бота ретельно проаналізовано та оптимізовано для досягнення найвищого рівня ефективності. Отже, розроблений та досліджений телеграм-бот відповідає вимогам сучасних стандартів кібербезпеки та може бути успішно використаним для захисту інформації та систем від потенційних загроз. Результати цієї роботи мають важливе значення для подальших досліджень у галузі кібербезпеки та розробки захисних систем.

Перелік використаних джерел.

1. Як створити чат-бот для Telegram-каналу - інструкція для адміністраторів. [Електронний ресурс]. - Режим доступу: https://netpeak.net/uk/blog/yak-stvoriti-chat-bot-dlya-telegram-kanalu-instruktsiya-dlya-administratoriv/

2. Indicators of Compromise (IoC): Examples, Lifecycle, and Security Impact. [Електронний ресурс]. - Режим доступу: https://www.aquasec.com/cloud-native-academy/vulnerability-management/indicators-of-compromise/

УДК 512.12

НЕМЕШ І.В.[1], ДОДЬ О.А.[1], ЛИСОБЕЙ Л.В.[2]

[1]Західноукраїнський національний університет

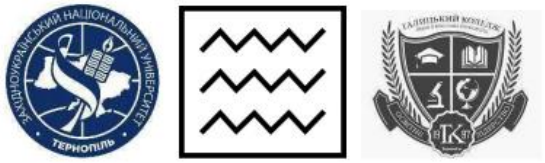
[2]Тернопільський академічний ліцей «Українська гімназія» імені Івана Франка

МЕТОД ВИЗНАЧЕННЯ ЧАСУ ТА СЕРЕДНЬОГО ЧИСЛА ІТЕРАЦІЙ АПРОКСИМУЮЧОГО *k*-АРНОГО АЛГОРИТМУ ЕВКЛІДА

Вступ. Побудова ефективного програмного забезпечення для реалізації алгоритмів обчислення найбільшого спільного дільника (НСД) натуральних чисел з точки зору їх додатків до обчислень з довгими числами має велике значення в сучасній криптографії та теорії чисел [1]. Процедура обчислення НСД присутня у багатьох криптосистемах, алгоритмах факторизації, інших додатках сучасної обчислювальної математики. Саме тому дослідження та реалізація алгоритмів обчислення НСД є важливою та актуальною тематикою.

Найбільш поширеним із таких алгоритмів є класичний алгоритм Евкліда. Однак відомі й інші алгоритми, загальною метою створення яких було зменшення складності обчислення НСД. За допомогою відповідних програм можна отримати чисельні оцінки збіжності аналізованих алгоритмів для різних вхідних наборів даних, вивести точні оцінки часу виконання процедури обчислення НСД та числа ітерацій, довести ефективність використання *k*-арного алгоритму Соренсона KARI та апроксимуючого алгоритму АКА для вирішення практичних завдань теорії чисел та криптографії [2]. Також визначено новий комбінований алгоритм КОМБІ, що поєднує простоту класичного алгоритму Евкліда та переваги *k*-арного і виконується швидше за обидва алгоритми.

Мета: Розробити методику та визначити середній час роботи і кількість ітерацій апроксимуючого *k*-арного алгоритму Евкліда.



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

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

науково-практична конференція
молодих вчених, аспірантів та студентів

29–31 серпня 2023
Тернопіль

УДК 004.056.5

Мотронюк.Н.Б.

Західноукраїнський національний університет

ВИЯВЛЕННЯ ТА АНАЛІЗ ІНДИКАТОРІВ КОМПРОМЕТАЦІЇ

Вступ. В сучасному світі кібербезпека стає надзвичайно важливою галуззю, оскільки кількість кіберзагроз та атак неперервно зростає. Однією із ключових складових ефективної кібербезпеки є виявлення та аналіз індикаторів компрометації. Робота присвячена вивченню методів та сервісів, які дозволяють виявляти потенційно шкідливі файли та аналізувати їхню активність.

Поєднання традиційних методів та сучасних технологічних рішень стає ключовим фактором в ефективній боротьбі із загрозами кіберпростору. У цьому контексті досліджуються сервіси виявлення загроз, такі як VirusTotal, Hybrid Analysis, та MetaDefender, що надають можливість автоматизованого аналізу файлів та виявлення потенційно небезпечної активності.

Мета: Аналіз та оцінка ефективності сервісів виявлення та аналізу індикаторів компрометації в онлайн-середовищі.

1. Методи виявлення індикаторів компрометації

У відомому кіберпросторі, яким стає сучасний світ, завдання виявлення індикаторів компрометації (IoC) набуває все більшої актуальності та важливості в контексті забезпечення надійності інформаційної безпеки та стабільності інфраструктури. Враховуючи цю необхідність, розглянемо різні методи, які застосовуються для виявлення IoC, кожен з яких вирізняється своєю унікальністю та значущістю в сфері кібербезпеки [1].

Хеш-суми файлів, які можуть включати в себе алгоритми MD5, SHA-1, або SHA-256, виступають як цифрові "відбитки" для різних файлів. Цей метод дозволяє швидко та ефективно виявляти відомі зразки шкідливого програмного забезпечення, базуючись на порівнянні хеш-сум з відомими в базах даних.

Аналіз поведінки системи –це подальший крок в розумінні IoC. Цей метод включає в себе вивчення активності програм та процесів, аналіз змін у системі та виявлення незвичайних дій, що можуть свідчити про можливу компрометацію. Реагуючи на аномалії в системній поведінці, ми маємо можливість оперативно виявляти потенційні загрози.

Сигнатурний аналіз використовує унікальні відбитки вірусів для розпізнавання шкідливих програм. Цей метод базується на порівнянні сигнатур відомих загроз з аналізованим файлом. Сучасні антивірусні системи широко використовують цей підхід.

Аналіз мережевого трафіку є важливим методом виявлення IoC, що дозволяє виявити підозрілі підключення, атаки "Command and Control" та інші аномалії у взаємодії комп'ютерів та серверів. Це важливо для оперативного реагування на загрози на рівні мережі [2].

Аналіз вразливостей використовує інформацію про вразливості систем та програмного забезпечення для виявлення можливих шляхів атак. Реагуючи на ці вразливості, ми можемо зменшити ризик компрометації та вдосконалити системи

захисту.

Ця комплексна система методів виявлення IoC вирізняється своєю динамічністю та адаптивністю, що є важливим для забезпечення ефективного та надійного захисту інформації та інфраструктури в умовах постійно зростаючих кіберзагроз.

2. Приклади використання сервісів для виявлення індикаторів компрометації

У сучасному цифровому ландшафті виявлення та аналіз індикаторів компрометації (IoC) є надзвичайно важливим етапом в управлінні кібербезпекою. Здатність оперативно реагувати на потенційні загрози та ефективно виявляти аномальну активність в мережі стає ключовою у боротьбі з сучасним кіберзлочинністю. Розглянемо деякі конкретні приклади використання сучасних сервісів для виявлення та аналізу IoC.

VirusTotal. Цей онлайн-сервіс виявлення вірусів і шкідливого програмного забезпечення надає можливість користувачам завантажувати файли для аналізу. З використанням декількох антивірусних движків, VirusTotal надає збалансований погляд на потенційні загрози та їхні відмінності.

Shodan. Для виявлення вразливостей в мережевому обладнанні та підключених пристроях широко використовується сервіс Shodan. Аналіз мережевих артефактів дозволяє не лише виявляти потенційно вразливі точки входу для атак, але й розробляти стратегії для їхнього ефективного усунення.

GreyNoise. Для відфільтрування нормальної мережевої активності використовується GreyNoise. Цей сервіс дозволяє виділяти незвичайні події та з'єднання, спрощуючи процес виявлення потенційно шкідливих дій в мережі.

ThreatConnect. Як платформа для об'єднання та аналізу інформації про загрози, ThreatConnect дозволяє сполучати дані з різноманітних джерел. Інтеграція таких даних надає можливість оперативно розпізнавати та реагувати на складні загрози.

Snort. Для реального виявлення загроз на рівні мережі використовується система Snort, яка аналізує пакети даних. Цей підхід забезпечує швидке виявлення потенційно шкідливих дій та активності.

Ці приклади демонструють, як використання сучасних сервісів може суттєво покращити здатність організацій виявляти та реагувати на потенційні кіберзагрози.

Сполучення різноманітних сервісів дозволяє забезпечити повноцінний підхід до кібербезпеки та ефективно контролювати цифрові ризики.

З урахуванням важливості кібербезпеки в сучасному цифровому середовищі, виявлення та аналіз індикаторів компрометації набуває стратегічного значення для організацій. У цьому контексті практичне використання цих процесів розглядається як критичний елемент сучасних стратегій кібербезпеки, спрямованих на забезпечення надійності та захищеності інформаційних ресурсів.

Однією з ключових сфер практичного застосування є оперативна реакція на інциденти. Швидке виявлення і аналіз індикаторів компрометації дозволяє організаціям ефективно реагувати на кібератаки, скорочуючи час виявлення та відновлення. Це має критичне значення для забезпечення безпеки та стійкості в

умовах постійно зростаючих загроз.

Подальший аспект використання полягає в систематичному моніторингу безпеки мережі. Аналіз індикаторів компрометації надає можливість постійно проводити моніторинг стану безпеки та вчасно реагувати на нові загрози. Це забезпечує підтримку високого рівня захищеності та виключає можливість виникнення серйозних інцидентів.

Захист від розповсюдження загроз – ще один аспект, де виявлення та аналіз індикаторів компрометації виявляється дуже ефективним. Можливість швидко реагувати на нові методи атак дозволяє попереджувати поширення шкідливого програмного забезпечення та мінімізувати ризики для організації.

Оптимізація ресурсів кібербезпеки є ще однією перевагою використання цих процесів. Зосередження уваги на найбільш критичних аспектах захисту та автоматизація аналізу дозволяє ефективно використовувати ресурси, підтримуючи ефективний захист інформаційних активів.

Завершуючи, важливість ретроспективного аналізу та накопичення досвіду неможливо переоцінити. Аналіз індикаторів компрометації минулих інцидентів допомагає вдосконалювати стратегії безпеки та уникати подібних ситуацій у майбутньому.

Висновок. У результаті дослідження необхідно зазначити, що виявлення та аналіз індикаторів компрометації грає ключову роль у стратегії кібербезпеки. Ефективна реалізація цих процесів дозволяє організаціям оперативно виявляти та вирішувати кіберзагрози, забезпечуючи надійний захист їхніх інформаційних ресурсів. Практичне використання виявлення та аналізу індикаторів компрометації охоплює широкий спектр заходів забезпечення кібербезпеки, включаючи оперативну реакцію на інциденти, систематичний моніторинг безпеки та захист від розповсюдження загроз. Ця стратегія сприяє створенню стійкої та ефективної системи захисту, яка адаптується до кіберсередовища, яке постійно змінюється. Впровадження стратегії виявлення та аналізу індикаторів компрометації стає важливим етапом для будь-якої сучасної організації, що прагне захистити свою інформацію та забезпечити стійкість в умовах постійних кіберзагроз.

Перелік використаних джерел.

1. O. Catakoglu, M. Balduzzi, and D. Balzarotti. Automatic extraction of indicators of compromise for web applications. In WWW 2016, 2016.

2. RIHANE C. IOC (Indicator of Compromise) & the Pyramid of Pain / Chedli RIHANE. – 2020. [Електронний ресурс]. – Режим доступу: https://www.linkedin.com/pulse/ioc-indicator-compromise-pyramid-pain-chedl i-m-rihane/