Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)
Кафедра кібербезпеки
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр
(назва освітнього ступеня)

на тему: Дослідження вразливостей комп'ютерних систем з використанням веб-скрапінгу.

Виконав: студент VI курсу, групи СБм-61
спеціальності 125 Кібербезпека
(шифр і назва спеціальності)

(підпис) *Урбан Д.А.* (прізвище та ініціали)

Керівник (підпис) Александер М. Б-А. (прізвище та ініціали)

Нормоконтроль (підпис) *Лобур Т.Б.* (прізвище та ініціали)

Завідувач кафедри (підпис) *Загородна Н.В.* (прізвище та ініціали)

Рецензент (підпис) (прізвище та ініціали)

Тернопіль
2022

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра Кібербезпеки
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Загородна Н.В.
(підпис) (прізвище та ініціали)
«____» _____ 2022 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Магістр
(назва освітнього ступеня)

за спеціальністю 125 Кібербезпека
(шифр і назва спеціальності)

Студенту Урбан Дмитро Андрійович
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження вразливостей комп'ютерних систем з використанням веб- скрапінгу.

Керівник роботи Александер Марек Богуслав Антонович, д.т.н., професор кафедри КБ
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «_25_» _листопада_ 2022 року № _4/7-966_

2. Термін подання студентом завершеної роботи 14 грудня 2022р.

3. Вихідні дані до роботи Наукові публікації про загрози для комп'ютерних систем

4. Зміст роботи (перелік питань, які потрібно розробити): Вступ, Розділ 1. розгляд основних систем вразливості та їх різновидів, 1.1 Розгляд аналізу влазливостей, 1.2 Опис вразливостей, 1.3 Висновок до першого розділу, Розділ 2. Джерело даних та його аналіз,2.1 Збирання даних із соціальних мереж, 2.2 Методика отримання даних, 2.3 Висновок до другого розділу, Розділ 3 Практична реалізація, 3.1 Отримання даних необхідних для аналізу загроз, 3.2 Виявлення вразливостей та оповіщення користувачів, Розділ 4 Охорона праці та безпека в надзвичайних ситуаціях, 4.1 Охорона праці, 4.2 Різновид рятувальних робіт та надзвичайних ситуацій, 4.3 Сили і засоби для проведення рятувальних робіт, 4.4 Висновки до четвертого розділу, Висновки, Перелік використаних джерел.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)
1 Титульна сторінка. 2 Тема. Мета. Об'єкт. Предмет дослідження. 3 Завдання дослідження. 4. Синтаксис дефекту демонструє установку дефекту цілісності, 5. Джерела вразливості в архітектурі багаторівневої веб-програми із зовнішніми посиланнями та налаштуваннями для користувача, 6. Відображення даних отриманих в ході виконання програми в базі даних, 7. Скрипт, який необхідно запустити користувачу, для відправлення інформації про процеси, запущені на комп'ютері, 8. Отримання ID користувача у телеграмі. 9. Ключові слова для користувача. 10. Повідомлення користувачу, 11 Висновки. 12 Апробація результатів валіфіаційної роботи, 13 Завершальний слайд.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Осухівська Г.М., к.т.н., доцент		
Безпека в надзвичайних ситуаціях	Клепчик В.М., старший викладач		

7. Дата видачі завдання 14 листопада 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	14.11.2022-15.11.2022	*Виконано*
2.	Підбір необхідного джерела для отримання даних	16.11.2022-20.11.2022	*Виконано*
3.	Переклад та опрацювання наукових джерел про		
	веб-скрапінг та аналіз вразливостей різних систем та		
	платформ	21.11.2022-23.11.2022	Виконано
4.	Виконання дослідження щодо аналіз інструментів		
	для організації інфраструктури та аналізу		
	даних	24.11.2022-27.11.2022	Виконано
5.	Оформлення розділу «Розгляд основних систем		
	вразливості та їх різновидів»	28.11.2022-30.11.2022	*Виконано*
6.	Оформлення розділу «ДЖЕРЕЛО ДАНИХ ТА		
	ЙОГО АНАЛІЗ»	01.12.2022-04.12.2022	*Виконано*
7.	Оформлення розділу «Практична реалізація»	05.12.2022-07.12.2022	*Виконано*
8.	Виконання завдання до підрозділу «Охорона праці»	08.12.2022-09.12.2022	*Виконано*
9.	Виконання завдання до підрозділу «Безпека в надзви-		
	чай них ситуаціях»	10.12.2022-11.12.2022	*Виконано*
10.	Оформлення кваліфікаційної роботи	12.12.2022-13.12.2022	*Виконано*
11.	Нормоконтроль	14.12.2022-15.12.2022	*Виконано*
12.	Перевірка на плагіат	9.12.2022	*Виконано*
13.	Попередній захист кваліфікаційної роботи	16.12.2022	*Виконано*
14.	Захист кваліфікаційної роботи	.12.2022	

Студент _____ Урбан Д.А.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Александер М. Б-А.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Дослідження вразливостей комп’ютерних систем з використанням веб-скрапінгу // Кваліфікаційна робота освітнього рівня «Магістр» // Урбан Дмитро Андрійовчи // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп’ютерно-інформаційних систем і програмної інженерії, кафедра кібербезпеки, група СБм-61 // Тернопіль, 2022 // С. 56, рис. – 9, додат. – 2, бібліогр. – 15.

Ключові слова: ВРАЗЛИВОСТІ, АНАЛІЗ ВРАЗЛИВОСТЕЙ, ВЕБ-СКРАПІНГ, ANS, БЕЗПЕКА, CVE, МОДЕЛЬ, TWITTER.

Кваліфікаційна робота присвячена аналізу вразливостей та створення системи оповіщення користувача.

У першому розділі продемонтровано опис загроз різного типу та розгянуто думки різних авторів, таких як Cova, Abbot, Landwehr та його колеги, Bishop та інших на рахунок методики аналізу їх.

У другому розділі розглядаються варіанти різних соціальних мереж, як Twitter, Reddit та Facebook, які можна використати для отримання інформації та проводиться в розглядається методика за допомогою якої можна отримати дані з даних ресурсів.

У третьому розділі проводяться наглядна демонстрація роботи розробленого сервісу, який отримує дані, після чого аналізує їх та сповіщає кінцевого користувача про наявність вразливості у системі чи програмному забезпеченні, яке він використовує.

ANNOTATION

Vulnerability Assessment for Computer Systems using Web-Scraping // Qualification work of the educational level “Master” // Dmytro Urban // Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Cyber Security, SBm-61 group // Ternopil, 2022 // P. 56, fig. - 9, annexes - 2, references - 15.

Key words: VULNERABILITIES, VULNERABILITY ANALYSIS, WEB SCRAPING, ANS, SECURITY, CVE, MODEL, TWITTER.

The qualification work is devoted to the analysis of vulnerabilities and the creation of a user warning system.

In the first section the description of different types of threats is presented and the opinions of different authors such as Cova, Abbot, Landwehr and his colleagues, Bishop and others on the methodology of analyzing them are considered.

The second section discusses the options of different social networks such as Twitter, Reddit and Facebook that can be used to obtain information and discusses the methodology by which data can be obtained from these resources.

The third section provides a visual demonstration of the developed service, which receives data, then analyzes it and notifies the end user about the presence of a vulnerability in the system or software he uses.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

APT - Advanced Persistent Threat

CSRF - Cross-site request forgery

CVE - Common Vulnerabilities and Exposures

DNS - Domain Name System

FFRDC - Federally Funded Research and Development Center

MPA - Multi Page Application

RASP - Run-time Application Security Protection

WAF - Web-application firewall

XSS - Cross-site scripting

Tweet - Microblog post on the platform Twitter

БД - База даних

ПЗ - Програмне забезпечення

ЗМІСТ

ВСТУП

Актуальність теми. Зі стрімким розвитком науки і техніки, відповідно підвищується ступінь інформатизації світу, а також ускладнюються комп'ютерні системи. Разом з тим, швидкість оновлення комп'ютерних систем настільки висока, що розробники не встигають помічати недоліки у своїх розробках. Деякі зловмисники користуються цими недоліками, уникають стратегій безпеки і руйнують комп'ютерні системи. стратегії безпеки і руйнують комп'ютерні системи. Ступінь дефекту називається вразливістю комп'ютерної системи. Вразливість комп'ютерної системи є внутрішньою властивістю кожної комп'ютерної системи. Іншими словами, не існує комп'ютерної системи без вразливостей. З широким розповсюдженням комп'ютерної науки і техніки розвиваються і хакерські технології. Багато хакерів використовують недоліки комп'ютерних систем, атакують ці недоліки і паралізують роботу всієї мережі. Поява і розвиток комп'ютерних мереж робить незалежні хост-комп'ютери пов'язаними один з одним, що підвищує ефективність комп'ютерних систем. Коли люди користуються перевагами зручних Інтернет-сервісів, їм також доводиться стикатися з різними загрозами з боку Інтернету. Все більше хакерів розробляють комп'ютерні віруси та атакують комп'ютерні системи. комп'ютерні віруси та атакують комп'ютерні системи. Події у сфері безпеки виникли ще в період комп'ютерних терміналів. У той час хакер використовував комп'ютерний вірус для атаки на незалежний хост-комп'ютер. Після появи Інтернету комп'ютерні віруси можуть передаватися кількома шляхами. Спираючись на Інтернет, хакери винайшли багато способів атак на комп’ютерні системи. Більше цілей атаки та методів атаки змушують людей приділяти більше уваги безпеці комп’ютерної системи.

Мета і задачі дослідження. Метою даної кваліфікаційної роботи освітнього рівня «Магістр» є виявлення нових вразливостей в системах та повідомлення користувачів про їх, для своєчасної реакції на загрозу безпеці.

Для досягнення поставленої мети було потрібно виконати наступні завдання:

- розглянути різні джерела отримання даних;
- провести дослідження проблеми;
- провести характеристику аналізу загроз;
- розробити програмний код для збору та аналізу вразливостей;
- навести опис можливих вразливостей, які можуть виникнути;

Об'єкт дослідження. Процеси захисту інформації комп'ютерних систем.

Предмет дослідження. Аналіз та сповіщення користувачів про загрози, які виникають для системи.

Наукова новизна одержаних результатів кваліфікаційної роботи полягає у тому, що за допомогою даної розробки можливо скоротити час реакції користувачів на потенційні вразливості для комп'ютерної системи та дозволяє своєчасно захиститися від них.

Практичне значення одержаних результатів. Продемонстровано роботу сервісу, який аналізуючи вміст сторінок визначає появу загроз, які виникають для будь-яких систем та програм.

Апробація результатів магістерської роботи. Основні результати проведених досліджень обговорювались на: Міжнародній науковій конференції X науково-технічної конференції «інформаційні моделі, системи та технології» (м.Тернопіль).

Публікації. Основні результати кваліфікаційної роботи опубліковано у двох працях конференції (див. Додаток А, Б).

РОЗДІЛ 1. РОЗГЛЯД ОСНОВНИХ СИСТЕМ ВРАЗЛИВОСТІ ТА ЇХ РІЗНОВИДІВ

1.1 Розгляд аналізу вразливостей

Для аналізу вразливості веб-додатків розроблено декілька інструментів та методик додатків для аналізу вразливості веб-додатків. Cova та ін. [1] класифікують аналіз вразливостей веб-додатків за моделями виявлення та методами аналізу.

На більш високому рівні вразливості можуть бути згруповані та класифіковані за певними ознаками їх атрибутів та операцій. Існуючі підходи дозволяють користувачам зрозуміти основи вразливостей та їх функціонування в цілому. Abbot та ін. [2] використовували фіксований описовий синтаксис для зображення порушень цілісності характеристик вразливостей та можливих способів їх використання в операціях над ними.

У твердженнях, що описують дефект цілісності, кожен елемент, пов'язаний з дефектом, є заздалегідь визначений і віднесений до певного класу. Значення класу, яке відповідає сценарію дефекту буде обрано і використано в фактичному описі дефекту (наприклад, дефект цілісності установки, зображений на рис. дефект цілісності установки, проілюстрований на рисунок 1.1). Ця структура описує багато варіантів вразливості цілісності через різні комбінації елементів в певних контекстах взаємодії операційних систем. Модель дозволяє формально класифікувати вразливості через схожість операційних систем, користувацьких додатків

Syntax

A [*Class of User*] user acquires the potential to compromise the integrity of an installation via a [*Class of Integrity Flaw*] integrity flaw which, when used, will result in unauthorized access to a [*Class of Resource*] resource, which the user exploits through the method of [*Category of Method*] to [*Category of Exploitation*].

Syntax Elements

[*Class of User*]

- Applications
- Service
- Intruder

[*Class of Integrity Flaw*]

- Physical Protection
- Personnel
- Procedural
- Hardware
- Applications Software
- Operating System

[*Class of Resource*]

- Information
- Service
- Equipment

[*Category of Method*]

- Interception
- Scavenging
- Pre-emption
- Possession

[*Category of Exploitation*]

- Denial of Possession/Use
  - Steal equipment
  - Destroy equipment
  - Degrade service
  - Interrupt service
  - Destroy data
- Denial of Exclusive Possession/Use
  - Read/Transcribe data
  - Steal service
- Modification
  - Alter data
  - Alter equipment

Рисунок 1.1 - Синтаксис дефекту демонструє установку дефекту цілісності

Обмеженням цього методу є те, що результати аналізу є лише текстовими описами, які вимагають експертних знань для того, щоб зробити висновок про взаємозв'язок між синтаксичними елементами взаємозв'язку між елементами синтаксису з метою опису реальної вразливості.

Варіативність елементів описового синтаксису в описі вразливостей також вимагає залучення різних експертно-залежних описів. У нашій моделі властивості додатку розглядаються як описовий "елемент" вразливості, а взаємозв'язок між властивостями вразливості, а зв'язок між властивостями визначається структурою додатку та його функціонуванням, що дозволяє уникнути залежності від текстового опису. Варіанти будуть представлені у вигляді комбінації існуючих властивостей. Крім того, наша модель надає масштабовану інформацію що описує, як вразливість стає доступною і коли вона експлуатується, що може статися. Більше того, нам не потрібно описувати загальні класи, як це зроблено в Class of Integrity Flaw та Category of Exploitation [2], оскільки вони можуть бути можуть бути виведені з основних атрибутів вразливості та її операцій.

Landwehr та його колеги [3] проаналізували програмні вразливості, використовуючи даних про дефекти програмного забезпечення та класифікували їх на основі генезису дефекту, часу виникнення на стадіях розробки та місцезнаходження. виникнення на стадіях розробки та місця, де були виявлені недоліки.

Оскільки підхід не обмежується певними класами систем, він може бути використаний для аналізу навіть апаратних вразливостей для аналізу навіть апаратних вразливостей. Однак, основним обмеженням цього підходу є те, що його описовий метод підходу є те, що його описова методологія орієнтована на операції. Таким чином, він не не описує в явному вигляді уразливості додатків, викликані властивостями об'єктів (наприклад переповнення буфера в уразливості типу даних або синтаксису команди в уразливості ін'єкції). Він також не розглядає комбінацію пов'язаних систем в одній уразливості, що часто зустрічається у веб-додатках.

Bishop [4] класифікує вразливості відповідно до інфраструктури та мережевих систем. Бішоп зазначає, що визначення та класифікація недоліків безпеки (або вразливостей) не є послідовними ні з точки зору виду недоліків ні з точки зору рівня концептуальної абстракції вразливості. Таким чином, багато описів недоліків можуть походити від однієї вади. Наприклад, переповнення буфера може бути описано як неповну перевірку параметра з точки зору дефектного процесу. З точки зору операційної системи ця вада описується як порушувана заборона/обмеження, оскільки параметр може посилатися на недозволену адресу. Непослідовна перевірка параметрів, описана на низькому рівні абстракції, може бути абстракції, може бути віднесена до стану гонки/асинхронної перевірки/проблеми неадекватної серіалізації або логічної помилки, що може бути використана на більш високих рівнях абстракції. Для уникнення неузгодженості та дублювання була розроблена інша модель аналізу вразливостей, яка інша модель аналізу

вразливостей, в якій вразливість декомпозується на примітивні умови, що формують набір незалежних характеристик вразливості [5].

Кожен набір характеристик вразливості є вибіркою з елементів повного набору характеристик вразливості по відношенню до системи. Оскільки характеристика вразливості є необхідною умовою для вразливості, то заперечення виявленої характеристики призведе до вимкнення пов'язаних з нею вразливостей. відповідні вразливості. Існуючі проблеми полягають у наступному: (а) як визначити (а) як визначити базовий набір характеристик вразливості та (б) характеристики вразливості не є повними та унікальними. вразливості не є повними та унікальними і не є "атомарними" описами. Таким чином, важко розробити ефективні інструменти, які здатні виявляти невідомі вразливості на основі знання характеристик існуючих вразливостей та систем, таких як програмні додатки, операційні системи та інфраструктура апаратні системи, які можуть бути причетні до існування, виникнення або до існування, виникнення або наслідків вразливостей. Крім того, визначення та класифікація нових характеристик невідомих вразливостей також є бажаними вимогами до таких інструментів. Neuhaus та ін. [6] аналізують вразливість на прикладі програмних компонентів та встановлюють шаблон ознак вразливого компонента на основі особливостей компонента. На основі синтаксису мови програмування (C та C++), що підтримує імпорт та виклик функцій, а також структуру додатку, визначається місцезнаходження вразливого компоненту та його зв'язки з іншими компонентами. Вразливого компонента та його взаємозв'язки з іншими компонентами під час компіляції та виконання. Патерн ознак вразливих компонентів у програмному забезпеченні видобувається видобувається як еталон для визначення того, чи можуть бути вразливими подібні імпорт або виклики функцій можуть бути вразливими. Результати застосовні до структурованого, заснованого на компіляції програмного забезпечення де структура компонентів та взаємозв'язки зазвичай фіксовані, а виконання самодостатнім. Таким чином, вони не можуть бути безпосередньо

застосовані для опису вразливостей в веб додатках, які базуються на інтерпретації з використанням декларативних мов та скриптів, а також мають динамічні, високопродуктивні скриптами, мають динамічний, високоінтерактивний та контекстно-залежний вміст, а також підтримуються декількома різними динамічний, високоінтерактивний та контекстно-залежний вміст, а також підтримуються декількома різними зовнішніми "компонентами", такими як Веб-браузер, сервер баз даних, вбудований вміст та гіперпосилання.

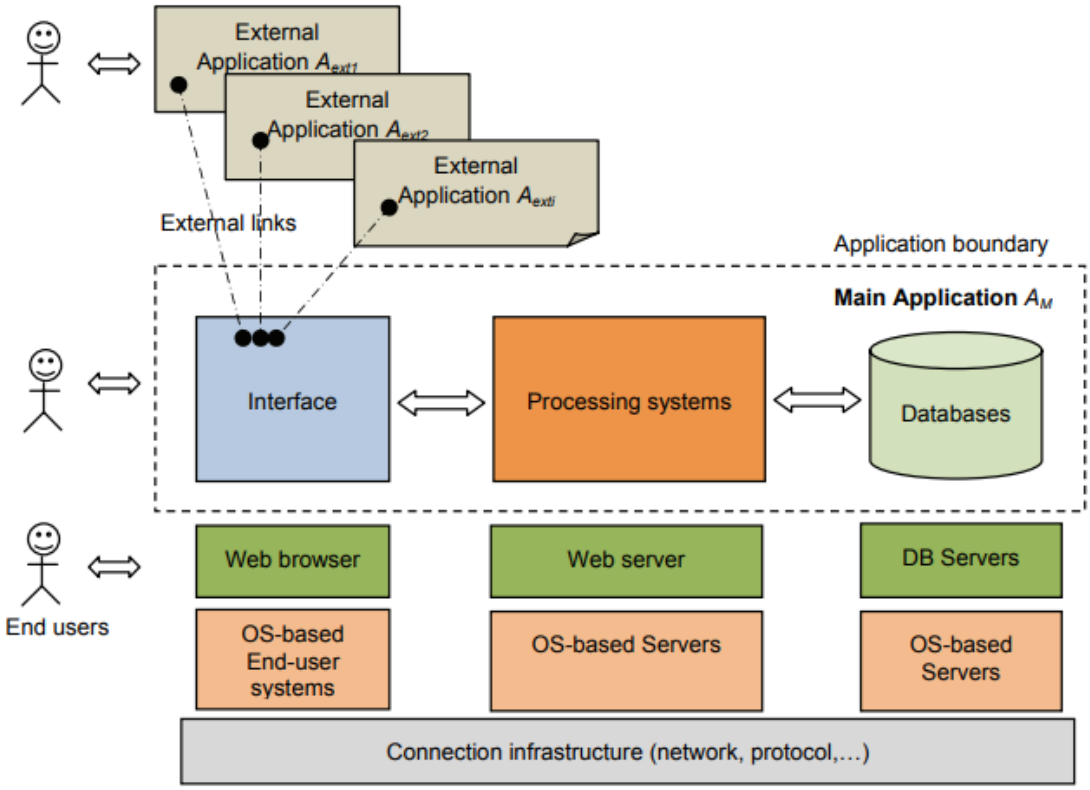

Рисунок 1.2 - Джерела вразливості в архітектурі багаторівневої веб-програми із зовнішніми посиланнями та налаштуваннями для користувача.

Вразливість може бути визначена як результат слабких місць у конкретному програмному забезпеченні продукті або протоколі [7]. Перелік типових вразливостей (Common Weaknesses Enumeration, CWE1) описує

взаємозв'язок вразливостей за рівнями абстракції. Крім того, вразливість веб-додатків веб додатків вимагає особливих міркувань. По-перше, вразливість веб-додатків вразливість може походити з джерел, відмінних від самого додатку. Веб додаток має багаторівневу архітектуру, що складається з окремо розроблених компонентів: систем інфраструктури, баз даних, систем обробки (back-end рівень), рівень інтерфейсу користувача (інтерфейсний рівень) та веб-браузери (рисунок 1.2). Як правило, Web 2.0 має більш гнучкі можливості кастомізації, що дозволяє кінцевим користувачам налаштовувати вміст додатків за допомогою декількох технологій, таких як Асинхронний JavaScript та XML (AJAX), Flash, JavaScript Object Notation (JSON), Simple Object Access Protocol (SOAP), Representational State Transfer (REST) для полегшення комунікації, обміну інформацією, інтероперабельності та співпраці у всесвітній мережі Інтернет. Таким чином, додаток може бути більш вразливим, оскільки він піддається впливу більш незахищеного вмісту [6].

По-друге, методи розробки наступного покоління, що застосовуються до веб-додатків, створюють нові проблеми безпеки. додатків, створюють нові проблеми безпеки, в той час як поточні проблеми все ще не вирішені [8]. Ще не до кінця вирішені [8]. Наприклад, в той час як уразливості, викликані застарілими методами програмування, не вирішені повністю, асинхронна веб розробка JavaScript+XML (AJAX) XML (AJAX) відкриває новий вектор атаки для шкідливого коду, який веб-розробник може не повністю перевірити. Сканери такі як Acunetix розробили функцію сканування для сканування AJAX / Web 2.0 веб-додатків та пошуку вразливостей. додатків AJAX / Web 2.0 та пошуку вразливостей.

Очевидно, що окрім самої вразливості, аналіз вразливостей повинен враховувати варіанти вразливостей для того, щоб встановити більш повне покриття для усунення вразливостей. Однак, варіанти вразливостей розглядаються варіанти вразливостей розглядаються по-різному з декількох

точок зору. У графовій моделі мережі атак [9], варіанти вразливостей можуть бути визначені шляхом задання умов та переходів атаки (експлуатації) вразливостей, виходячи з конкретного контексту середовища та конкретних програмних пакетів або версій. Bishop розглядає різні вразливості з точки зору аналізу на основі поглядів на існуючі інфраструктурних систем [5]. Підхід Берге та ін. [10] передбачає, що варіант може бути отриманий за рахунок коригування ключових властивостей вразливості. Варіант вразливості також розглядається як порушення обмежень, що накладаються ресурсами комп'ютерної системи або припущеннями, зробленими ресурсами комп'ютерної системи або припущеннями, зробленими щодо використання цих ресурсів комп'ютерної системи або припущень щодо використання цих ресурсів. На практиці поняття варіанту вразливості в сканерах визначається по-різному в сканерах визначається по-різному. Варіанти вразливості можуть визначатися як варіація значень параметрів атаки, які можуть бути використані в одному вразливому місці успішної симуляції атаки. Такий підхід, який був застосований в сканерах Acunetix та IBM застосований в Acunetix та IBM AppScan, дозволяє користувачеві побачити будь-який окремий варіант уразливості як зразок атаки на уразливість, а отже стає дуже наочною ілюстрацією експлуатації уразливості. Наочною ілюстрацією використання уразливостей. За допомогою потенційних атак на основі цих зразків, користувач може застосувати відповідні засоби запобігання. Однак, цей метод обмежує себе в межах існуючого набору даних атак на вразливості, в той час як реальна атака, як правило, є в той час як фактична атака зазвичай відбувається з непередбачуваних значень.

Хоча сканер дає певні можливості для виправлення ситуації, він не забезпечує достатнього покриття для кінцевих користувачів: вони повинні покладатися на свій досвід для врегулювання та розширення можливості запобігання, такі як фільтрація даних з різними значеннями. Інші сканери (наприклад, HP WebInspect, Cenzic HailStorm) виявляють кілька вразливих

місця у веб-додатку, але пропонують лише одне можливе значення атаки на виявлені вразливі місця. Головна проблема полягає у тому, що станеться, якщо типовий випадок, згенерований сканером вже був належним чином відфільтрований додатком, але інші випадки експлуатації все ще залишаються під прикриттям. Обмеженість охоплення можливості сканера вразливостей не можуть змоделювати всі можливі використання уразливостей. Іншими словами, сканеру може здаватися, що вразливості не існує, але насправді вона існує. Сканеру може здаватися, що вразливість не існує, але насправді вона існує.

В роботі Berghe та ін. [10] основна увага приділяється вразливостям, що виникають в окремих компонентах у веб-сервісах. Вони припустили, що вразливість безпеки присутня у властивостях системи. На варіанти вразливостей можна впливати, змінюючи вибраних властивостей існуючої вразливості. Запропоновано метод вимірювання вразливостей на основі властивостей та аналізу архітектурної досконалості функціональних компонентів додатку для побудови прогнозної таксономії вразливостей. Лінійна кореляція між вразливостями та властивостями системи використовується для оцінки та зважування вразливостей, які з великою ймовірністю можуть з'явитися в тому чи іншому компоненті системи. У цьому підході нова вразливість розглядається як розглядається як варіація існуючої вразливості, присутньої в системі.

Як і методологія Бішопа, цей підхід також ґрунтується на поєднанні властивостей для визначення варіанту вразливості. За допомогою цих двох підходів можна визначити схожість або ймовірність двох вразливостей на основі набору властивостей (або характеристик). Однак, оскільки взаємозв'язки між властивостями не визначені, ці два підходи не визначені, ці два підходи обмежені в описі випадків, в яких вразливості можуть бути пов'язані між собою. Вразливості можуть бути пов'язані між собою.

Наприклад, Cross-Site Scripting може дозволити зловмиснику генерувати контрольовані зловмисником скрипти для обійти захист від Cross-Site Request Forgery уразливості. Bazaz та ін. розглядають взаємозв'язки між вразливостями програмного забезпечення, виконуваним процесом та наявними ресурсами системи. Вони виводять специфічну концепцію вразливості, в якій програмний додаток є вразливим до експлойтів, коли він порушує ресурсні обмеження та припущення щодо використання ресурсів. Аналіз вразливостей аналіз вразливостей базується на ресурсах системи, якими керує користувачем. Однак такий підхід має основну проблему: оновлення вразливостей вимагає більше зусиль для додавання нових обмежень та припущень від виявлених та класифікованих вразливостей та класифікованих вразливостей. Крім того, багато уразливостей, таких як SQL Injection, які використовують слабкість у валідації SQL-команд, не пов'язані з порушенням ресурсних обмежень. Нещодавнє дослідження Aime та ін. [11] включає в себе оцінку вразливостей та оцінку серйозності. Дерево несправностей та дерево подій у структурі графа ризиків використовуються для представлення незахищеної ситуації додатку, і вони використовуються для аналізу того, як окремі несправності можуть або поширюватися, або використовуватися для спричинення небажаних наслідків для систем. Граф ризиків включає кореневий вузол, який представляє кінцевий негативний наслідок. Кожна можлива подія, яка може призвести до наслідку, стає нащадком кореня. Кожен нащадок, у свою чергу, може бути коренем повного піддерева (списку) всіх подій, які можуть безпосередньо призвести до нього. Цей тип графа є актуальним і графа є актуальним і був адаптований до нашого вразливого майнового відношення граф. Ми використовуємо граф в більш широкому сенсі: визначення зв'язків вразливої властивості та інших пов'язаних властивостей всередині додатку. Властивостей та інших пов'язаних всередині додатку або від/до зовнішніх сутностями, що виникають в результаті роботи додатку. В роботах використовується підхід графа атак. Можливий стан атаки

стан визначається як вузол, а ребро представляє зміну стану, викликану одиничною дією зловмисника (включаючи звичайні переходи користувача, якщо зловмисник отримав доступ до зловмисник отримав доступ до облікового запису звичайного користувача) або діями, виконаними помічником (наприклад, виконання помічником (наприклад, запуск троянського коня), як основу для визначення та та представлення атаки та її наслідків за ступенем тяжкості. Справа в тому, що кількість комбінацій вузлів комбінацій вузлів і ребер в графі атаки може вибухнути, було зазначено в дослідженнях Swiler та ін. Для вирішення проблеми комбінаторного вибуху, Свілер та ін. запропонували автоматизований генератор графів, який отримує шаблони атак, конфігураційний файл та шаблони атак, конфігураційний файл та профіль зловмисника на вході та генерує граф атаки. Christodorescu та ін. [13] запропонували метод, заснований на поведінці, для аналізу зловмисної поведінки та виявлення зловмисних зловмисної поведінки та виявлення шкідливих програм та їх варіантів. Зловмисна поведінка відрізняється від нормальної поведінка відрізняється від нормальної поведінки при взаємодії з операційної системи і представляється у вигляді графа залежностей.

Граф залежностей однозначно описує операції, що виконуються даним шкідливим програмним забезпеченням. Поняття підходу "шкідлива безпечна поведінка" адаптована в нашому підході з використанням відношеннями на основі властивостей, які застосовуються до всіх властивостей, що описують додаток відносно його вразливості. На практиці, деякі підходи до опису вразливостей пропонують розглядати вразливість з точки зору зловмисника розглядати вразливість з точки зору зловмисника. Моделі атак зазвичай використовуються в інструментах захисту від атак, в яких одна або декілька симуляцій атак будуть запускаються на цільову систему для перевірки можливостей безпеки, а також наявності дірок у безпеці, якщо такі існують. наявність дірок у безпеці, якщо такі існують. Інструмент ATiki, розроблений Штефаном і Шумахером [9], представляє контекст атаки за допомогою

графових моделей, побудованих з інформації про атаку, витягнутої зі стандартних документів з розробки процесу (керівництва з програмування, контрольні списки та вимоги до забезпечення якості) та неформальних баз даних опису. вимог до якості) та неформальних баз даних описів, таких як Bugtraq. Конкретний контекст описується на основі множинного успадкування в об'єктно-орієнтованій парадигмі. Вимагається, щоб кожна умова і перехід були поміщені в контекст для класифікації та взаємозв'язку між вразливостями та подіями в конкретній системі повинні бути виражені. Цей підхід встановлює описову ієрархію атак від контекстно-специфічних до загального класу і забезпечує гнучку веб-платформу для спільної роботи з перехресними посиланнями. Перевагою запропонованої моделі є те, що вона дозволяє розширювати опис атаки з найменшою зміною вихідних взаємозв'язків між існуючими атаками та їх властивостями.

Зокрема, для веб-додатків існує класифікація вразливостей в залежності від моделі виявлення помилок та методики аналізу. Для веб-додатків, які, як правило, розробляються з використанням додатків, які зазвичай розробляються з використанням високорівневих декларативних декларативними мовами високого рівня, уразливості найчастіше виникають в місцях, де скриптові алгоритми взаємодіють з іншими системами та компонентами, такими як алгоритми взаємодіють з іншими системами або компонентами, такими як бази даних, файлові системи, операційні системи або мережа. Іншими словами, системи можуть бути скомпрометовані за допомогою веб-технологій, наприклад, експлуатація через веб-скрипт може призвести до порушення безпеки. скрипту може призвести до порушення безпеки. В основному, ці методи аналізу виявляють сигнатури специфічних особливостей для визначення відомих вразливостей, наявних у додатку. Методи припускають, що вразливість буде існувати в певних місцях в додатку, а характерні ознаки вразливості будуть присутні у відомих

пов'язаних виконавчих компонентах. Наприклад, такі методи, як як tainting technique (Halfond та ін. [14], Huang та ін., Nguyen-Tuong та ін.), техніка псування (Halfond та ін. [14], Huang та ін., Nguyen-Tuong та ін.), введення помилок (Ghosh та ін.) або тестування на проникнення зосереджені на виявленні аномальних на виявленні аномальних станів програми та ненормальної поведінки додатків у у відповідь на заздалегідь визначені дії, що надсилаються додатку навмисно. Наступні кроки можуть включати висновок про вразливість, відстеження походження вразливості та ресурси вразливості та припущення про можливість її використання на основі відомих шаблонів вразливості вразливості та структури вразливих компонентів і додатку. Метод ін'єкції помилок (Ghosh et al) сканує вихідний код, вводить помилки для оцінити поведінку додатку. Цей метод визначає вразливість як порушення в компонентах, пов'язаних з додатком, таких як вихідний код, конфігурація та взаємодія між компонентами; отже, він припускає що будь-яка аномальна поведінка розглядається як існування потенційної вразливості. Відстеження та аналіз пошкоджень (наприклад, Halfond та ін. [14], WebSARRI Huang та ін., Nguyen-Tuong та ін. або Pixy) відстежують пошкоджену інформацію в значеннях даних або в інформаційному потоці, аналізувати потокочутливий, міжпроцедурний та контекстно-чутливий потік даних для автоматичного виявлення вразливих місць програми. виявляти вразливі місця в програмі. Зосереджуючись на забрудненості та відстежуючи розповсюдження зіпсованої інформації, процес безпеки може відфільтрувати потенційну експлуатацію та вказати точну початкову позицію експлуатації і забезпечити відповідну санітарну обробку. Як негативний статичний, так і негативний динамічний підходи використовують аналіз забрудненої інформації. Інші позитивні та динамічні методології, такі як тестування на проникнення або моделювання атаки, використовують підхід тестування "чорного ящика". Метод змушує аномальні стани програми під час виконання програми та спостерігає за поведінку програми. Виявлення аномальних станів

оцінює поведінку програми додатку в порівнянні з моделлю поведінки без атак, яка створюється на основі моніторингу програми під час нормальної роботи і зробить висновок про існування потенційних вразливостей. У той час як методи сканування виявляють вразливості в додатках, Woo et al. запропонували підхід, в якому статистична методологія працює з даними про виявлені вразливості у веб-браузерах. Woo та ін. розширюють модель виявлення вразливостей (Vulnerability Discovery Model, VDM), запропоновану Alhazmi та Малайя , назвавши її логістичною моделлю Alhazmi-Malaiya Logistic (AML). Набори даних про виявлення вразливостей, які досліджуються та підходять до моделі виявлення вразливостей веб-браузера, будуть використовуватися для прогнозування як поточних, так і майбутніх так і майбутніх вразливостей.

1.2 Опис вразливостей

Хоча існує багато інструментів аналізу та виявлення вразливостей для веб-додатків, жоден з них не надає повного рішення або покриття методологію пошуку вразливостей, як це потрібно [15]. Різні методи сканування в сканерах веб-додатків реалізовані різні методи сканування, а результати сканування може допомогти оцінити безпеку веб-додатку. Однак, жоден сканер не забезпечує технологічно незалежного покриття можливих вразливостей. Експерименти, показують, що результати сканування вразливостей є залежними від сканера, мови, типу та вимагають трудомісткого, дорогого та схильного до помилок аналізу. Як наслідок, користувачі повинні покладатися на комбінацію інструментів виявлення вразливостей і повинні розуміти різні формати опису вразливостей. Однак, як показано, сканери зазвичай перетинаються у своїх висновках щодо вразливостей, що призводить до збільшення витрат, зниження продуктивності, надлишку даних та накладних витрат на аналізу та виявлення вразливостей. Таким чином, необхідно

розробити загальний метод опису вразливостей необхідно розробити загальний метод опису вразливостей.

OASIS рекомендує використовувати стандартну методологію веб-додатків Vulnerability Description Language (AVDL), "стандартний формат XML, який дозволяє суб'єктам (таким як додатки, організації або інститути) обмінюватися інформацією про вразливості веб-додатків, а саме обмінюватися інформацією щодо вразливостей веб-додатків". Іншим є стандарт безпеки веб-додатків - WA. Як AVDL, так і WAS розвиватимуться як загальний формат опису для кожного класу вразливостей і допоможуть мінімізувати вищезазначені ризики. класу вразливостей і допоможуть мінімізувати вищезазначені недоліки. Бази даних вразливостей представляють інформацію про вразливості в різних форматах даних. На абстрактному рівні CVE надає всебічний загальний.

На абстрактному рівні CVE надає всебічний загальний опис вразливостей та взаємозв'язків між вразливостями. Інформація призначена для загального обміну знаннями та визначення типів вразливостей. інформація призначена для загального обміну знаннями та визначення типів вразливостей, а не а не є специфічною для конкретної вразливості або сканера вразливостей. Таким чином, користувачі, як правило, потребують навичок експертного рівня для аналізу вразливостей. Інші джерела бази даних вразливостей, такі як Інші джерела бази даних вразливостей, такі як Common Vulnerabilities and Exposures (CVE2), US-CERT Vulnerability Notes Database3 надають інформацію про конкретні уразливості зі сценарним описом та пропонують посилання на додаткові дані, які більшість користувачів не бажають шукати шукати. Інші ресурси опису вразливостей походять з різних джерел та підтримують різні типи інформації. Наприклад, виробники засобів захисту (такі як McAfee, Acunetix, Cenzic) надають супровідні документи вразливостей, які можуть бути відскановані сканерами. Постачальники програмного забезпечення (такі як Microsoft Security Bulletin4 , Adobe Security

Bulletins and Advisories5) надають документи про конкретні вразливості, виявлені в їхніх продуктах. Особисті звіти, такі як в , детально описують вразливості з деталізацією виявлення вразливостей в певних сценаріях. Крім того, веб-сайти, форуми або списки розсилки організацій, що займаються питаннями безпеки також надають інформацію про вразливості з різних точки зору. Хоча описи з цих джерел можуть описувати одну і ту ж вразливість одну й ту ж саму вразливість, вони не мають єдиного стандартного формату, а отже, існує багато збігів. Така інформація, як правило, є результатом роботи окремих осіб і тому для її аналізу необхідна експертиза в певній галузі.

1.3 Висновок до першого розділу

В ході написання першого розділу було розглянуто різні методики, які використовують для виявлення та аналізу вразливостей, такі як CVE, яка дає загальний опис вразливості, OASIS, яка використовується для поширення загроз веб додатків, та було описано рішення запропоновані рядом авторів у використаних джерелах, як Cova, Abbot, Landwehr та його колеги, Bishop та інші.

РОЗДІЛ 2. ДЖЕРЕЛО ДАНИХ ТА ЙОГО АНАЛІЗ

2.1 Збирання даних із соціальних мереж

Соціальні мережі стали величезним майданчиком для обміну інформацією, який лише продовжує збільшуватися в розмірах. Соціальні медіа визначаються як "форми електронної комунікації (такі як як веб-сайти для соціальних мереж і мікроблогів), за допомогою яких користувачі створюють онлайн-спільноти для обміну інформацією, ідеями, особистими повідомленнями та іншим контентом (наприклад, відео)". Станом на січень 2022 року, за оцінками, налічується 4,3 мільярда активних користувачів соціальних медіа, що на 13,2% більше, ніж у 2020 році. Більше того, компанія Cisco прогнозує 16 що глобальний IP-трафік на місяць досягне 396,0 ЕБ (ексабайт) до 2022 року, що означатиме потрійне збільшення обсягу даних у порівнянні з 2017 роком. Тому можна припустити, що зростання використання соціальних мереж також сприяло зростанню обсягів трафіку.

Дані соціальних мереж є прибутковими і описуються дослідниками як "очевидно, найбільша, найбагатша і найбільш динамічну доказову базу людської поведінки". У соціальних мережах можна знайти незліченну кількість спільнот, представлених у соціальних мережах, включаючи величезну спільноту кібербезпеки, які відомі своєю співпрацею та обміном інформацією. Це вказує на важливість наявності засобів ефективного використання соціальних медіа як інструменту в сфері КІБ. Одним з методів вилучення даних з соціальних мереж є веб-скрепінг. Цей процес означає копіювання даних з веб-сайту та їх локальне зберігання у структурованому форматі. Перевага скрейпінгу полягає в тому, що він перевага вилучення даних, на відміну від потокової передачі даних безпосередньо з веб-сайту без їх збереження, полягає в тому, що дослідник може на відміну від потокового передавання даних безпосередньо з веб-сайту без їх збереження, полягає в

тому, що дослідник може пізніше повернутися до даних для проведення подальшого аналізу.

Існують деякі етичні проблеми, пов'язані з вилученням даних з Інтернету, які в основному стосуються подальше нечесне застосування збережених даних, питання авторського права та питання, пов'язані з приватності. Однак це дослідження не стосується цих питань, оскільки не встановлено жодного зв'язку між автором та розміщеним ним контентом. між автором та розміщеним ним контентом, що робить набір даних анонімним.

Twitter — це платформа соціальних медіа та веб-сайт мікроблогів, ідеальний для OSINT у багатьох галузях дослідження, включаючи кібербезпеку. Були також інші платформи соціальних мереж, які розглядалися для цього дослідження, однак після початкової підготовки стало очевидним, що Twitter не має собі рівних за своїм потенціалом у CTI. Станом на січень 2021 року існує оцінка 353 мільйони активних користувачів Twitter і Twitter входять до числа найбільш використовуваних соціальних мереж. Крім того, щодня публікується в середньому 500 мільйонів твітів за статистикою 2020 року. Тому обсяг даних у Twitter значний, що збільшує можливість пошуку корисної інформації.

Reddit - одна з найрізноманітніших соціальних мереж в Інтернеті. Згідно з власною статистикою компанії, він має 430 мільйонів активних користувачів щомісяця та 52 мільйони користувачів щодня. Хоча це не так багато, як у інших соціальних медіа-гігантів, він більш ніж компенсує це своєю гнучкістю та різноманітною користувацькою базою.

На відміну від інших сайтів соціальних мереж, Reddit дозволяє людям створювати підредіти, невеликі сторінки спільноти, присвячені певним темам. Кожен субреддіт містить теми, окремі повідомлення, надіслані користувачами, які можуть включати фотографії, відео та GIF-файли. У межах

теми інші користувачі можуть відповідати на допис і вести бесіди. На головній сторінці субредактиву теми можна сортувати за популярністю, кількістю голосів, кількістю відповідей або свіжістю. Також можна шукати старі повідомлення в субредітах, і результати пошуку можуть бути відсортовані аналогічно.

Це робить Reddit одним з найбільш гнучких і зручних для користувача сайтів соціальних мереж в Інтернеті. Будь-хто може створити сабреддіт на будь-яку тему. Немає необхідності використовувати хештеги або позначати людей, щоб приєднатися до розмови, і користувачі залишаються анонімними. Це робить людей більш охочими та здатними формувати спільноти за своїми інтересами та вподобаннями. В результаті на Reddit існують десятки тисяч процвітаючих спільнот, присвячених різним темам - від схуднення до відеоігор, від політики до улюблених брендів.

Ось чому Reddit є чудовою мішенню для веб-скрепінгу. Сайт переповнений інформацією на нішеві теми. Дослідники можуть зіскребти дані Reddit, щоб дізнатися, що люди думають про різні теми, зібрати поради та підказки з різних питань або виявити тенденції в громадській думці. Reddit також значно легше сканувати, ніж інші сайти соціальних мереж. Не потрібно вгадувати хештеги або створювати акаунти, прив'язані до імені реальної людини. Легко зібрати всю доступну інформацію по темі, не пропускаючи цінні розмови через налаштування приватності або заборони IP-адрес.

Facebook є невичерпним джерелом інформації, яку можна використовувати для кількох цілей. Часто ці дані є помічником для досягнення бізнес-цілей, проведення політичних та соціологічних досліджень тощо. Використання скрейпу сторінок Facebook часто називають незаконною діяльністю. Ми не погоджуємося з таким твердженням у випадку з видобутком

інформації, яка знаходиться у відкритому доступі, адже будь-яка особа може зібрати цікаві дані і без скрайбінгу. Хоча останній може зайняти значно більше часу в залежності від обсягу завдання.

Що стосується даних, які можна отримати за допомогою скрейперів Facebook, то вони безпосередньо залежать від ваших цілей. При скрапінгу профілів користувачів Facebook можна отримати таку інформацію, як ID, ім'я, фотографії, поточне місто, кількість підписників тощо. Коротше кажучи, всю інформацію, яка вам може знадобитися про користувачів. При скануванні груп Facebook ви також можете отримати корисну інформацію про кількість постів у групі, кількість лайків, коментарів, основні теми постів або згадки певних ключових слів, які вам потрібні для підрахунку.

Аналогічно можна збирати дані про згадки продуктів, популярність подій на основі постів і коментарів. Можливості софту для скрапінгу обмежуються лише вашими потребами і точністю ключових пошукових запитів.

2.2 Методика отримання даних

Веб-скрапінг, також відомий як веб-вилучення або збирання— це техніка вилучення даних із даних глобальної павутини (WWW) і зберігання його у файл системи або базі даних для подальшого пошуку чи аналізу. Веб-дані зазвичай видаляються за допомогою протоколу передачі гіпертексту (HTTP) або інтернет браузеру. Це можуть зробити вручну користувачі або автоматично ботами чи веб-сканерами. Так як сайт постійно генерує багато різних даних, веб-скрейпінг широко відома, ефективна та потужна технологія збирання даних для Big Data(набори інформації настільки великих розмірів, що традиційні способи та підходи не можуть бути застосовані до них.). Для адаптації до різних сценаріїв, поточний метод веб-скрейпінгу був налаштований на спеціальні процедури за допомогою людини,

використовуйючи повністю автоматичні системи можливо перетворити весь веб-сайт на добре організований набір даних.

Найсучасніший інструмент для веб-скрейпінгу не тільки може аналізувати мову розмітки або JSON(формат даних) файли а також займатися візуальною аналітикою і обробка природною мовою для імітації продуктивності користувача такої як перегляд веб-контенту. Процес вилучення даних з Інтернету можна розділити на два послідовні кроки; отримання мережевих ресурсів, а потім витягувати з отриманих даних необхідну інформацію. Зокрема, мережа скраперу починається зі створення HTTP запиту на отримання ресурсів веб-сайту. Цей запит можна відформатувати в будь-якому форматі. URL-адреса, що містить запит GET або фрагмент HTTP, що містить повідомлення запиту POST. Отримавши один раз успішно запим та обробивши цільовий веб-сайт, можна буде його замінити та отримати ще раз з інакшими параметрами не змінюючи при цьому програму веб скрапера.

Ресурси можуть існувати у кількох форматах, наприклад веб-сторінки каналів даних на основі HTML, XML або JSON формат або мультимедійні дані, такі як зображення, аудіо, або відеофайли. Після завантаження веб-даних, процес продовжує аналізувати, переформатовувати та впорядковувати дані в структуровані. Інтернет має два основних модулі отримувач(грабер) - модуль для складання HTTP-запитів, такі як Urllib2 або selenium, інший для аналізу та вилучення інформації з необробленого HTML-коду, наприклад BeautifulSoup або Pyquery. Модуль Urllib2 визначає набір функцій для роботи з HTTP запитами, такі як аутентифікація, перенаправлення, cookie, тощо, тоді як Selenium є мережею, яка створить оболонку браузера наприклад Google Chrome або Internet Explorer, і дозволяє користувачам автоматизувати процес перегляду веб-сайт за допомогою програмування. Для отримання даних із HTML та інших документів XML використовують Beautiful Soup.

За допомогою цього ми отримуємо зручні функції Pythonic для навігації, пошуку та зміни дерев розбору і зручні інструмент для розкладання файлів HTML та вилучення необхідної інформації через lxml або html5lib. BeautifulSoup може автоматично визначити кодування і перетворювати його в код, який може прочитати клієнт. Те ж саме робить і Pyquery який надає набір функцій, подібних до Jquery для розбору документів xml. Проне на відміну від BeautifulSoup, Pyquery підтримує лише lxml і використовуєть для швидкої обробки XML фалів. Існують різні типів веб-скребків, деякі призначені для автоматичного розпізнавання структури даних сторінок, такі як Nutch або Scrapy, інші надають графічний веб-інтерфейс . Якщо немає необхідності писати скрапер вручну можна використати код, наприклад з Import.io. На сайті присутній сильний і розширюваний веб-сканер, написаний на Java, що забезпечує точну конфігурацію, паралельну компіляцію, підтримку robots.txt та машининного навчання.

Scrapy, написаний на Python, може використовувати повторну структура мережевого сканування. Це прискорює процес створення та розширення великих проектів сканування. Крім того, він надає веб-оболонку для імітація поведінки веб-перегляду людини. Щоб дозволити і звичайним людям, а не лише програмістам збирати веб-контент, за допомогою графічного інтерфейсу, розробленого для спрощення труднощів повязаних із використанням веб-скребків. Серед них Import.io — типовий сканер виділивши дані з веб-сайту без створення коду дозволяє користувачам ідентифікувати та конвертувати неструктуровані веб-сторінки в структурованому форматі. Графічний інтерфейс Import.io для ідентифікації даних дозволяє користувачам вивчати та розуміти, що стягувати із сторінки. Потім отримані дані можна зберегти в спеціальному місці на хмарних серверах, а також можна експортувати в CSV, JSON, і формат XML.

Веб-сканер з графічним інтерфейсом для легкого складання та візуалізації, передачі даних у режимі реального часу на основі SVG або WebGL двигуні, але не може працювати з великою кількістю даних.

Веб скрапінг можна використовувати для таких сценарії, як захоплення фотографій, зміна ціни тощо. Порівнювати зміни, які відбулися у відображенні продуктів, колекції списків нерухомості, погода, моніторинг даних, виявлення змін веб-сайту та інтеграція веб-даних. Наприклад, у мікромасштабі ціни на акції можна регулярно порівнювати враховуючи, що ціна змінюється з часом, також можна відслідковувати канали соціальних мереж і шукати колективний заклик до голосування визначаючи лідерів думок. На макрорівні майже кожен має метадані, нейронні мережі постійно навчаються для створення пошукових запитів, наприклад Google Search або Bing Пошук.

Для ефективної кібербезпеки потрібен інтелект, безпосередньо застосовний до механізмів захисту, які виявляють і запобігають вторгненням. Один із основних артефактів проактивної кібербезпеки є індикаторами компромісів, які можна помістити в автоматичне зловмисне програмне забезпечення та загрозу системи виявлення, а також допомагають працівникам інформаційної безпеки аналізувати шкідливі події. МОК є «одним із найпоширеніших видів технічної розвідки вторгнення» , які поширюються в межах спільноти кібербезпеки на різних визначені платформи, а також соціальні мережі. МОК можуть висловлюватися в широкому діапазоні форм, однак найбільш часто торгуються фрагментами даних - це хеші шкідливих файлів, IP-адреси, доменів та подібної інформації, яку можна легко застосувати для виявлення вторгнень системи.

Традиційно IOC розподіляються між професіоналами на таких платформах, як MISP – Threat Intelligence Sharing Platform, це програмне забезпечення з відкритим вихідним кодом для «збирання, зберігання, розповсюдження і обмін індикаторами та загрозами кібербезпеки» , де організації можуть обмінюватися виявленою інформацією з довіреними

сторонами. Є також більше загальнодоступні платформи зі схожою інформацією, наприклад VirusTotal, де він знаходиться можливий пошук через IOC, агрегований з «антивірусних систем, сканерів веб-сайтів, інструменти аналізу файлів і URL-адрес, а також внески користувачів». Однак така інформація може також можна знайти на різних платформах соціальних мереж, особливо в Twitter. Крім того організацій, існує також спільнота незалежних мисливців за загрозами в Twitter їхні відкриття. У наступній главі буде спроба витягти IOC із зіскребаного Дані Twitter за допомогою регулярних виразів.

Хоча веб-скрейпінгу є потужною технікою проте це досить суперечливо при зборі великих наборів даних, оскільки може піднімати юридичні питання, пов'язані з авторським правом, умовами обслуговування та «привласнення власності». Сканер aweb копіює частину даних безкоштовно у вигляді графіків або таблиць на веб-сторінках, без будь-яких порушень авторських прав. Авторські права на такі дані, як конкретні місця або спеціальні параметри захищені законом щодо умов використання, хоча більшість веб-додатків містять договора у формі про надання послуг їх виконання часто знаходиться в сірій зоні. Приклад, власник веб-сканера, який порушує умови використання може стверджувати, що він або вона ніколи не зустрічалися або офіційно не Погодилися з умовами обслуговування. Крім того, якщо веб-сканер надто часто надсилаються запити на дані власник сайту може відмові в обслуговуванні. Зазвичай веб-скребки уникають цієї проблеми, дотримуючись обмеженої частота запитів.

Видалення IOC з даних Twitter є ефективною процедурою кібербезпеки. Аномалії в даних можна враховувати в правилах фільтрації. З результатів відокремлюючи МОК від даних, скопійованих з Twitter, зрозуміло, що деякі типи інформації доступніші за інші. Щоб цей метод працював ефективно

необхідно встановити ретельний і точний набір правил видалення. В ідеалі – вилучені цінності повинні бути правильно класифіковані, щоб їх можна було точно застосувати компоненти мережевої інфраструктури. Крім того, безпечні та надійні точки даних не слід додавати до IOC, оскільки це може призвести до втрати даних. Далі – аналізуючи кожен тип МОК, який виконується в цій главі, виходить, що шкідливий пошук доменного імені успішно досяг очікуваних результатів можна знайти в отриманих даних. Наклепне доменне ім’я є чітким показником IOC, оскільки це звичайний метод створення для суб’єктів загрози Інформація, яку вони надають, безпечна для користувача.

Такі домени можна додавати безпосередньо в механізм веб-фільтрації організації для запобігання випадковому зараженню користувачів комп'ютера в мережі. Однак не існує єдиного правила видалення для всіх шкідливих URL-адрес, тому важко уникнути фільтрації необхідних даних або їх видалення. Крім того, не всі мисливці за загрозами здатні захиститися інформацію, яку вони публікують, тому вам також потрібно буде витягти популярні URL-адреси, щоб визначити, що є шкідливим, потрібен контекст, а не лише окремий домен Ім'я. Видалення шкідливих IP-адрес має подібні проблеми з фільтрацією URL-адрес, але вони пов’язані Завдяки простій структурі IP це дослідження виявилося легшим у реалізації. Однак таке відділення МОК від набору даних Twitter має свої особливості ускладнення. існує багато IP-адрес, які потрібно додати до білого списку. Це багато служб вирішення DNS, чиї захищені адреси часто з’являються.

Вміст Twitter, який обговорює питання кібербезпеки. Ще одна хвиля, яка стає у цьому дослідженні очевидно, що такі дані, як дата і час, можуть мати подібну модель IP-адреси. Тому регулярний вираз потрібно побудувати більш точно, щоб цей метод був ефективним. Якби система зберігала помилкові спрацьовування в цій формі процес веб-фільтрації, швидше за все, не мав би руйнівних результатів, однак було б неправильним використанням пам'яті та сховища.

Веб-програми можуть виконувати одну з наведених нижче дій, щоб заблокувати інтернет або перешкоджати його роботі інструменти реєстрації даних для збору даних веб-сайт. Ці заходи можуть визначити чи хірургічне втручання проводиться особою або працівниками. Деякі ключові види діяльності включають: дослідження «відбитків пальців» HTML-заголовків, які використовуються для ідентифікації відвідувачів, шкідливий або безпечний; репутація інтелектуальної власності, необхідно визначити, де знаходиться IP-адреса, її історія використання, зафіксована під час атак на веб-сайти, буде розглядатися з підозрою і більш імовірно після ретельного розгляду; Аналіз поведінки виявляє аномалії моделі поведінки, наприклад підозріле розміщення, високий рівень запитів і дотримання винятків, перегляд шаблонів і прогресивні завдання. Наприклад, він фільтрує роботів з набором завдань.

Соціальні мережі з кожним роком ростуть і є одним із факторів, що сприяють цій тенденції. Так само, як помітне збільшення кіберпростору. Ці зони кіберпростору, що розширюються, відкриваються двері до безпрецедентних дій усіх масштабів для забезпечення кібербезпеки. Чиновники повинні боротися з ними ефективно і продуктивно займатися цими, необхідно використовувуючи стійкі кіберзагрози і деякі методи пом’якшення. Одиним із методів можна збирати CTI з загальнодоступних джерел. Соціальна мережа фактично одна з найбільших публічних баз знань, що об'єднує широкий спектр інформації, включаючи дані про інциденти кібербезпеки, загрози, дійових осіб та допомогу програм. Тому для покращення можливостей кібербезпеки вони повинні існувати. Рішення для фільтрації та вилучення такої інформації з соціальних мереж. У цій статті пропонується підхід до використання соціальних медіа для підтримки кібербезпеки. Отримайте необхідну інформацію для пом’якшення загроз. Вирішіть пропоноване дослідження вперше використовує сканери в соціальних мережах платформи Twitter і використання ключових слів

кібербезпеки для отримання відповідних даних. Пошук, фільтрація та кількісна оцінка необхідних даних у послідовності на реплікованих даних, оцініка загальних потенціалів Twitter для відстеження інцидентів кібербезпеки. Шукаючи застарілі дані, щоб отримати інформацію, придатну для активного використання кібербезпека, наприклад МОК або вразливості.

2.3 Висновок до другого розділу

В ході написання другого розділу було проведено аналіз різних соціальних мереж та можливостей для отримання даних з них та подальше їх використання. Проаналізувавши всі переваги та недоліки було вирішено, що найбільш підходящими соціальними мережами можна вважати Twitter та Reddit, оскільки вони дозволяють просто та зручно отримати необхідну інформацію для аналізу загроз.

Також було охарактеризовано та описано схеми організації перевірки безпеки, аналізу та отримання інформації про вразливості.

РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ

3.1 Отримання даних необхідних для аналізу загроз

Вибір Twitter акаунта пав на офіційни акаунт CVE оскільки - Common Vulnerabilities and Exposures (CVE) - це база даних публічно оприлюднених проблем інформаційної безпеки. Номер CVE однозначно ідентифікує одну вразливість зі списку. CVE забезпечує зручний, надійний спосіб для постачальників, підприємств, науковців та всіх інших зацікавлених сторін обмінюватися інформацією про проблеми кібербезпеки. Підприємства зазвичай використовують CVE та відповідні оцінки CVSS для планування та визначення пріоритетів у своїх програмах управління вразливостями.

Вперше запущена в 1999 році, CVE управляється і підтримується Національним центром досліджень і розробок з кібербезпеки FFRDC (Federally Funded Research and Development Center), що управляється корпорацією MITRE. CVE фінансується федеральним урядом США, при цьому Міністерство внутрішньої безпеки США (DHS) та Агентство кібербезпеки і безпеки інфраструктури США (CISA) надають операційні кошти. CVE є загальнодоступною і безкоштовною для всіх бажаючих.

До того, як у 1999 році було започатковано CVE, було дуже важко обмінюватися даними про вразливості між різними базами даних та інструментами. Кожен постачальник підтримував власну базу даних з власною системою ідентифікації та різними наборами атрибутів для кожної вразливості. CVE гарантує, що кожен інструмент може обмінюватися даними з іншими інструментами, а також забезпечує механізм, за допомогою якого можна порівнювати різні інструменти, такі як сканери вразливостей.

Хоча дехто може поставити під сумнів, що публічне розкриття вразливостей полегшує хакерам використання цих вразливостей, загальновизнано, що переваги переважають над ризиками. CVE включає лише публічно відомі вразливості та загрози безпеці. Це означає, що хакери можуть отримати доступ до даних, пов'язаних з CVE, незалежно від того, чи є вони в переліку CVE чи ні. Крім того, деталі CVE часто не включаються до списку вразливостей до тих пір, поки відповідний постачальник не випустить патч або інше виправлення, що гарантує, що підприємства зможуть захистити себе після оприлюднення інформації. Крім того, обмін інформацією в індустрії кібербезпеки може допомогти прискорити пом'якшення наслідків, а також гарантувати, що всі організації будуть захищені швидше, ніж якщо їм доведеться самостійно виявляти та знаходити способи усунення вразливостей CVE.

Мною була побудована програма на основі програмної мови Пайтон, та за допомогою бібліотек requests, використовується для отримання з’єднання із веб-сторінкою, re та json, використовуються для пошуку інфіормації та зручного запису даних, та telebot, використовується для роботи із телеграм ботом. Програма складається із файлів constants.py(змінні, необхідні для роботи програми), main.py(основна програма), parser_data.json(дані отриманні при роботі програми), user_data.json(файл із інформацією про користувачів).

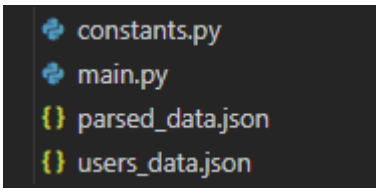

Рисунок 3.1 – Демонстрація структури проекту.

Ми отримуємо дані зі сторінки твітер акаунта «CVE» із юзер айді «CVEnew» після чого записуємо у зручний для читання формат(літинг 3.1)

Лістинг 3.1 – Отримання даних із твіттер акаунта

```
      url                                                    =
"https://twitter.com/i/api/graphql/zEAmYdYUvRuhrr9zjc5Vvg/UserTw
eets?" \

"variables=%7B%22userId%22%3A%2282180628746174054 4%22%2C%22count
%22%3A40%2C%22includePromotedContent%22%3Atrue%2C%22" \

"withQuickPromoteEligibilityTweetFields%22%3Atrue%2C%22withSuper
FollowsUserFields%22%3Atrue%2C%22" \

"withDownvotePerspective%22%3Afalse%2C%22withReactionsMetadata%2
2%3Afalse%2C%22" \

"withReactionsPerspective%22%3Afalse%2C%22withSuperFollowsTweetF
ields%22%3Atrue%2C%22withVoice%22%3Atrue%2C%22" \

"withV2Timeline%22%3Atrue%7D&features=%7B%22responsive_web_twitt
er_blue_verified_badge_is_enabled%22%3Atrue%2C%22" \

"verified_phone_label_enabled%22%3Afalse%2C%22responsive_web_gra
phql_timeline_navigation_enabled%22%3Atrue%2C%22" \

"view_counts_public_visibility_enabled%22%3Afalse%2C%22view_coun
ts_everywhere_api_enabled%22%3Afalse%2C%22" \

"tweetypie_unmention_optimization_enabled%22%3Atrue%2C%22respons
ive_web_uc_gql_enabled%22%3Atrue%2C%22vibe_api_enabled%22%3Atrue
%2C%22"\

"responsive_web_edit_tweet_api_enabled%22%3Atrue%2C%22graphql_is
_translatable_rweb_tweet_is_translatable_enabled%22%3Atrue%2C%22
" \

"standardized_nudges_misinfo%22%3Atrue%2C%22tweet_with_visibilit
y_results_prefer_gql_limited_actions_policy_enabled%22%3Afalse%2
C%22" \

"interactive_text_enabled%22%3Atrue%2C%22responsive_web_text_con
versations_enabled%22%3Afalse%2C%22responsive_web_enhance_cards_
enabled%22%3Atrue%7D"

          headers = {
              'authorization':                              'Bearer
AAAAAAAAAAAAAAAAAAAAAANRILgAAAAAAnNwIzUejRCOuH5E6I8xnZz4puTs%3D1Z
v7ttfk8LF81IUq16cHjhLTvJu4FA33AGWWjCpTnA',
```

```
              'cookie':
'_twitter_sess=BAh7CSIKZmxhc2hJQzonQWN0aW9uQ29udHJvbGxlcjo6Rmxhc
2g6OkZsYXNo%250ASGFzaHsABjoKQHVzZWR7ADoPY3JlYXRlZF9hdGwrCMe6LVRz
AToMY3NyZl9p%250AZCIlYjI3YjAwNTZkZDg1ZDA2MDI2NzA3ZGY0YTA3ZGVkNDM
6B2lkIiU0N2Fl%250AZGM5MzIzMmM1NDJmMzZhZTk0OTk1NTRlM2MwYw%253D%25
3D--bdf78d74648ba7ee09ffd6ce2b73d23bcae38145;
guest_id_marketing=v1%3A159484514908990613;
guest_id_ads=v1%3A159484514908990613;
_ga=GA1.2.881263206.1644676930;
personalization_id="v1_Df16969CUlQSEtvFFr5qdw==";
guest_id=v1%3A165843030332732125;              g_state={"i_l":0};
kdt=gPHU7qNgaYeTTTIey00GQkochPWL61DF8xynjn6x;
auth_token=8e85b3b99cd32a271b7d629e9722ba48f33d4f04;
ct0=a2311fc59827ef8e4efd92e34477cebba914fb1a57595eb5011472e0921c
77bc6436831ae9e906776d53552f538aac99dc0f0a4bee5e24beb907428fc979
9bd77b345d97c59a61483761b8065a7aaee5;
twid=u%3D992713660756832257;    _gid=GA1.2.1856984179.1671378609;
external_referer=padhuUp37zj9327dHdO0nT0Mc%2Bdf1I7x|0|8e8t2xd8A2
w%3D',
              'x-csrf-token':
'a2311fc59827ef8e4efd92e34477cebba914fb1a57595eb5011472e0921c77b
c6436831ae9e906776d53552f538aac99dc0f0a4bee5e24beb907428fc9799bd
77b345d97c59a61483761b8065a7aaee5'
          }

          response = requests.request("GET", url, headers=headers)

          result_json = json.loads(response.text)
```

В результаті виконання коду ми отримуємо на виході словник із даних, які відображаються на сторінці з яким будемо працювати далі в програмі.

Після того як ми отримали дані ми робимо перевірку їх наявності в базі даних, яку ми заповнюємо інформацією про вразливості, яку отримали раніше та створюємо список із даних, якщо вразливість не була раніше зафіксована в базі даних.

Лістинг 3.2 – Перевірка наявності

```
      parsed_data = open("parsed_data.json", "r")
          parsed_data = json.loads(parsed_data.read())

          new_vulnerability_list = []
                                for            twitt            in
result_json['data']['user']['result']['timeline_v2']['timeline']
['instructions'][1]['entries']:
```

```
            if "tweet" in twitt['entryId']:
                                                                                if twitt['content']['itemContent']['tweet_results']['result']['legacy']['full_text'] not in parsed_data:
                        new_vulnerability_list.append(twitt)
```

Приклад вигляду інформації в базі даних відображений на рисунку 3.2

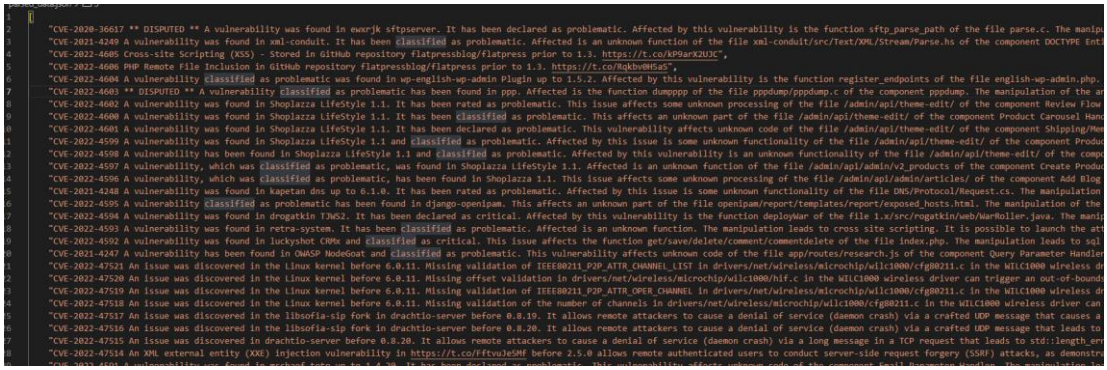

Рисунок 3.2 – Відображення даних отриманих в ході виконання програми в базі даних

Оскільки використовували лише один ресурс даних є не надійним і може давати збої в роботі та також може не відображати всіх можливі вразливості, які появляються щодня, було додано друге джерело даних.

Отримуємо дані про нові пости, які появилися на сторінці та перетворюємо їх у читабельний формат для програми(лістинг 3.3).

Лістинг 3.3 – Отримання інформації із reddit

```
url="https://www.reddit.com/r/cybersecurity/?f=flair_name%3A%22New%20Vulnerability%20Disclosure%22"

response = requests.request("GET", url)

search_data=re.search("window.___r = ({.+?(?=};)})", response.text).group(1)

json_search = json.loads(search_data)
```

Після того як ми отримали дані ми робимо перевірку їх наявності в базі даних, яку ми заповнюємо інформацією про вразливості, яку отримали раніше та створюємо список із даних, якщо вразливість не була раніше зафіксована в базі даних.

Лістинг 3.4 – Перевірка наявності нових постів

```
parsed_data = open("parsed_data1.json", "r")
parsed_data = json.loads(parsed_data.read())

new_posts = []
for post in json_search['posts']['models']:
    if json_search['posts']['models'][post]['title'] not in parsed_data:
        new_posts.append(json_search['posts']['models'][post])
```

3.2 Виявлення вразливостей та оповіщення користувачів

Далі в коді ми робимо перевірку на наявність ключових слів у описі вразливості, зазвичай в описі є одне або два ключових слова, які ми отримуємо від користувача, який запускає скріп(рисунок 3.3) який стягує інформацію про процеси, які є запущені в даний момент в системі(лістинг 3.5), також користувач може сам відправити список ключових слів, які необхідні для коректної роботи системи, при створенні свого запису в базі даних, на основі яких програма визначає чи необхідно оповістити користувача(лістинг 3.6) і в разі необхідності відправляє користувачу інформацію, яка є мінімально необіхною для визначення загрози і швидкого сповіщення в разі виникнення загрози(лістинг 3.7).

Лістинг 3.5 – Код скрипта, який запукає користувач

```
import wmi

from soccet import send_user_data

# Initializing the wmi constructor
f = wmi.WMI()
```

```
send_user_data(f.Win32_Process())
```

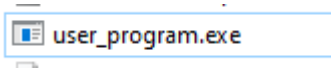

Рисунок 3.3 – Скрипт, який необхідно запустити користувачу “user_probram.exe” для відправлення інформації про процеси, запущені на комп’ютері

Лістинг 3.6 – Аналіз повідомлення на предмет наявності ключових слів.

```
parsed_data = open("parsed_data.json", "r")
    parsed_data = json.loads(parsed_data.read())

    user_data = open("users_data.json", "r")
    user_data = json.loads(user_data.read())

    for vulnerability in new_vulnerability_list:
        if "tweet" in vulnerability['entryId']:
                                                text_value      =
vulnerability['content']['itemContent']['tweet_results']['result
']['legacy']['full_text']
                                                        url       =
vulnerability['content']['itemContent']['tweet_results']['result
']['legacy']['entities']['urls'][0]['expanded_url']
                  for user in user_data:
                      for keyword in user['keywords']:
                         if re.search(f"[,.\s]*{keyword}[,.\s]*",
text_value):
                           call_user(text_value, url, user['id'],
keyword)
                             break
                  parsed_data.append(text_value)
```

Оскільки нам необхідна початкова точка для аналізу від користувача вимагається надати його унікальний ID, необхідний для відправлення повідомлення через телеграм бот, який можна отримати через телеграм бота https://t.me/my_id_bot (рисунок 3.4), також необхідно надати список ключових слів по який користувач хоче отримувати інформацію. Частіше за все це програми та опереційні системи, які користувач використовує(рисунок 3.5)

Рисунок 3.4 - Отримання ID користувача у телеграмі

```
"keywords": ["windows", "linux", "sftpserver"]
```

Рисунок 3.5 – Ключові слова для користувача

Лістинг 3.7 – Відправлення повідомлення корисувачу

```
message = f"There is a new vulnerability for {keyword}.
\n=====\n {text_value} \n=====\n {url}"

bot.send_message(chat_id=user_id, text=message)
```

В результаті виконання програми користувач отримав сповіщення про вразливість, короткий опис та посилання на сторінку із більш детальною інформацією про вразливість(рисунок 3.5).

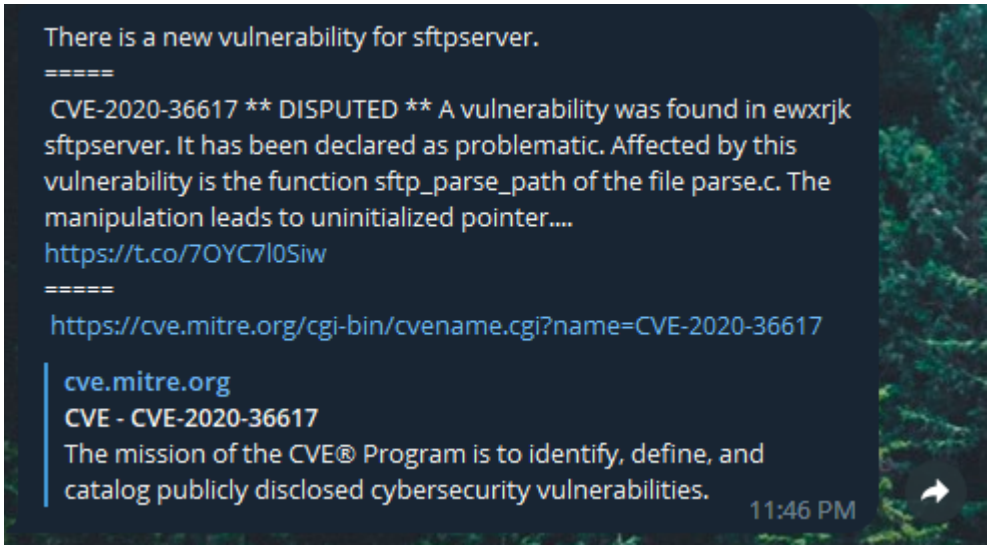

Рисунок 3.5 – Повідомлення користувачу про вразливість з Twitter

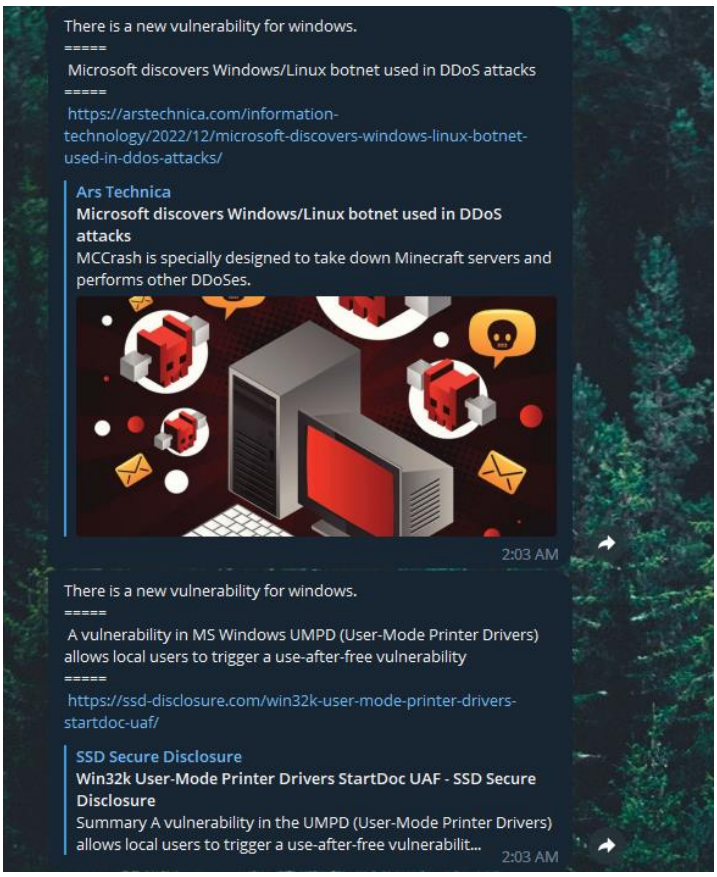

Рисунок 3.6 – Інформаці про вразливості із Reddit

Програма запускає скан раз на п’ять хвилин, що дає користувачу близько 14 днів на фікс, оскільки за цей час більшість вразливостей стають загально відомими і велика кількість хакерів пробують використати його у своїх цілях.

3.3 Висновок до третього розділу

В даному розділі наведено практичну реалізацію сервісу, який отримує дані із соціальних мереж Twitter та Reddit, та після цього аналізуючи вміст отриманих даних, визначає чи необхідно сповістити користувача про наявність нової вразливості у телеграм бот, що дозволяє скоротити час ознайомлення кінцевого користувача про наявність загроз для його системи.

РОЗДІЛ 4. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Охорона праці

Будь-яке підприємство, установа чи організація, що використовують у своїй діяльності найману працю робітників, повинні передбачити порядок та основні правила взаємодії між підприємством і залученими робітниками. Для цього в компаніях розробляють та затверджують локальні нормативні акти з охорони праці. При підготовці будь-якого документа, а особливо документа з охорони праці, головне — дотриматися юридичних норм та чинного законодавства.

Оскільки, проведення робіт з розробки та використання системи передбачає використання комп'ютерної техніки, зокрема ПК та периферійних пристроїв, то обов'язковим є дотримання вимог з охорони праці і техніки безпеки.

Для всіх споруд і приміщень, в яких експлуатуються відеотермінали та ЕОМ, повинна бути визначена категорія з вибухопожежної і пожежної безпеки відповідно до НАПБ Б.03.002-2007 "Норми визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою" (замість ОНТП 24-86), та клас зони згідно з ПУЕ. Відповідні позначення повинні бути нанесені на вхідні двері приміщення.

Будівлі і ті їх частини, в яких розташовуються ЕОМ, повинні мати не нижче II ступеня вогнестійкості. Приміщення для обслуговування, ремонту та налагодження ЕОМ повинні належати за пожежовибухобезпекою до категорії В відповідно до НАПБ Б.03.002-2007, а за класом приміщення – до П-IIа за ПБЕ. Якщо відповідно до ДБН В.1.1-7-2002 ці приміщення повинні бути відокремленими від приміщень іншого призначення протипожежними

стінами, то межа їх вогнестійкості визначається відповідно до ДБН В.1.1-7-2002.

Неприпустимим є розташування приміщень категорій А і Б (НАПБ Б.03.002-2007), а також виробництв з мокрими технологічними процесами поряд з приміщеннями, де розташовуються ЕОМ, виконується їх обслуговування, налагодження і ремонт, а також над такими приміщеннями або під ними.

Стіни кабін виготовляються з негорючих матеріалів. Дозволяється виготовляти їх зі скла та металевих конструкцій. У кабіні мусить бути оглядове вікно (вікна). Висота оглядового вікна має бути не менше 1,5 м, а відстань від підлоги не більше 0,8 м.

Приміщення з ЕОМ, крім приміщень, в яких розміщуються ЕОМ типу ЕС, СМ та інші великі ЕОМ загального призначення, повинні бути оснащені системою автоматичної пожежної сигналізації відповідно до вимог НАПБ Б.06.004-97 “Перелік однотипних за призначенням об'єктів, які підлягають обладнанню автоматичними установками пожежогасіння та пожежної сигналізації”, та ДБН В.2.5-13–98 “Інженерне обладнання будинків і споруд. Пожежна автоматика будинків і споруд” (назміну СНиП 2.04.09-84 "Пожарная автоматика зданий и сооружений") з димовими пожежними сповіщувачами та переносними вуглекислотними вогнегасниками з розрахунку відповідно до “Типових норм належності вогнегасників” (наказ МНС України 02.04.2004 № 151), з урахуванням граничнодопустимих концентрацій вогнегасної рідини відповідно до вимог Правил пожежної безпеки в Україні. В інших приміщеннях допускається встановлювати теплові пожежні сповіщувачі.

4.2 Різновиди рятувальних робіт та надзвичайних ситуацій

Унаслідок надзвичайних ситуацій у населених пунктах країни і на підприємствах можуть виникнути руйнування, зараження місцевості

радіоактивними та хімічними речовинами. Люди можуть опинитися у завалах, пошкоджених та палаючих будинках, інших непередбачуваних ситуаціях. У зв'язку з цими обставинами буде потрібне проведення заходів із рятування людей, надання їм допомоги, локалізації аварій та усунення пошкоджень. При вирішенні цих проблем виходять з того, що в осередках ураження і районах лиха будуть проводитися не тільки суто рятувальні роботи, а й деякі невідкладні, що не пов'язані з рятуванням людей. Рятувальні та інші невідкладні роботи (РіНР) проводяться з метою порятунку людей та надання допомоги ураженим, локалізації аварій та усунення пошкоджень, створення умов для наступного проведення відновлювальних робіт. При проведенні РіНР великого значення має дотримання певних умов. Такими умовами є: своєчасне створення угруповань, сил, що залучаються для проведення РіНР, своєчасне ведення розвідки, швидкий рух і введення сил у осередок ураження, безперервне проведення РіНР до їх повного завершення, тверде й оперативне управління силами, що залучаються до проведення РіНР, і всебічне забезпечення їх діяльності.

Заходи, що відносяться до рятувальних робіт:

- розвідка маршрутів, за якими вводяться або виводяться формування ЦО;
- локалізація і гасіння пожеж;
- пошук і рятування людей з-під завалів;
- відкриття зруйнованих захисних споруд і рятування людей;
- подача повітря у завалені захисні споруди;

- надання ураженим першої медичної допомоги та їх евакуація;
- саноброблка людей та знезараження їх одягу;
- знезараження місцевості, споруд, техніки.

Крім рятувальних робіт, в осередках ураження проводяться невідкладні роботи, до яких відносяться:

- прокладання маршрутних шляхів на заражених територіях і будування проїздів у завалах;
- локалізація аварій на комунально-енергетичних мережах, лініях зв'язку та їх відновлення;
- закріплення або ліквідація конструкцій споруд, які загрожують падінням та перешкоджають проведенню рятувальних робіт;
- ліквідація боєприпасів та інших вибухонебезпечних предметів (балони з газом, бочки з бензином тощо).

Керівництво проведенням усіх цих робіт у надзвичайних ситуаціях проводяться надзвичайними комісіями держави, області, міста тощо.

При аваріях на об'єктах народного господарства, установах, якщо їх наслідки не виходять за межі об'єктів захисних зон, керівництво роботами проводиться адміністрацією підприємств (див. Додаток 3).

Виникнення стихійних лих, а також аварій та катастроф можна в деяких випадках прогнозувати. Ці прогнози, як правило, закладаються в плани ЦО підприємств, установ, що передбачають попереджувальні заходи, які повинні зменшити наслідки аварій і катастроф.

Характер та обсяг таких заходів залежать від виду та рівня аварії або стихійного лиха, масштабів і часу їх виникнення.

Загалом до таких заходів відносяться:

- приведення в готовність засобів захисту;
- перевірка готовності систем оповіщення;

- підготовка і видача населенню засобів індивідуального захисту та особистої профілактики;
- проведення санітарно-епідеміологічних заходів;
- підготовка до евакуації або відселення та їх проведення;
- вивезення матеріальних цінностей;
- захист продуктів харчування, джерел води тощо;
- герметизація приміщень і т.п.

Способи і послідовність проведення цих робіт залежать від обставин, що склались у районі аварії чи катастрофи, та наявності сил і засобів для проведення таких рабіт.

Ліквідація наслідків надзвичайної ситуації проводиться для відновлення роботи підприємств, організацій, навчальних закладів тощо.

При ліквідації наслідків надзвичайної ситуації здійснюються такі заходи:

- розвідка осередків надзвичайних ситуацій;
- локалізація і гасіння пожеж;
- відбудівля споруд і шляхів сполучення;
- проведення ізоляційне обмежених заходів в осередках інфекційного зараження;
- проведення спецобробки населення;
- дезактивація, дегазація техніки, майна, доріг, місцевості тощо.

Розвідку осередків надзвичайних ситуацій проводять сили Збройних сил, ЦО і невоєнізовані формування підприємств, організацій тощо.

Воєнізовані сили розвідки ЗС і ЦО включають підрозділи радіаційної, хімічної, біологічної та інженерної розвідок. У завдання цих підрозділів входить виявлення загального стану в осередках та визначення меж ураження,

руйнування, повені І пожеж, а також виставлення спосте-реження на особливо важливих напрямках (станціях, переправах, перехресті доріг тощо).

У місцях розташування евакуйованого населення, на маршрутах виходу з осередків надзвичайних ситуацій розвідка ведеться силами невоєнізованих формувань підприємств та організацій.

Аварійно-рятувальні й лікувально-евакуаційні заходи проводяться додатково до заходів, вжитих підрозділами ЗС, ЦО, медичних установ в осередках надзвичайних ситуацій. Ці роботи виконує населення, яке потрапило в осередок або розміщене на шляху розширення ураженого повітря, пожежі, повені тощо. Для допомоги у проведенні цих робіт в осередки надзвичайних ситуацій направляють сили і засоби спеціальних формувань ЗС, ЦО, Мінохорони здоров'я, комунальних служб, Міністерства охорони навколишнього середовища та інші.

Локалізація і гасіння пожеж проводяться з метою збереження матеріальних цінностей держави й окремих громадян. Здійснюється це протипожежними формуваннями ЗС, ЦО, МВС, Мінохорони навколишнього середовища із залученням до цих робіт робітників, службовців і населення, що близько проживає до осередку надзвичайної ситуації.

Відбудівля споруд і шляхів сполучення здійснюється з метою поновлення роботи життєво важливих органів міста, району тощо. До них належать: телеграф, телефон, лікарні, мости, залізниця, шляхи евакуації і підвезення матеріальних засобів та інші.

Щоб запобігти поширенню епідемічних хвороб, проводять протиепідемічні заходи. До цих робіт залучають медичні заклади, санітарні дружини підприємств, навчальних закладів.

Усі протиепідемічні заходи в осередку організовує санепідемстанція або пересувний протиепідемічний загін. Проводять цю роботу медичні служби поліклінік, амбулаторій та інших лікувально-профілактичних заходів.

4.3 Сили і засоби для проведення рятувальних робіт

Для проведення рятувальних робіт залучаються невоєнізовані формування ЦО, військові частини і підрозділи, медорганізації тощо Невоєнізовані формування мають у першу чергу проводити рятувальні роботи на об'єктах народного господарства.

До них входять формування загального призначення, які мають у своєму складі аварійно-технічні формування, формування механізації робіт тощо. Вони можуть бути посилені протипожежними, дорожніми, автомобільними та іншими підрозділами.

Від швидкості та рішучості дій формувань залежить життя багатьох людей та збереження матеріальних цінностей.

При рятувальних роботах потрібно дотримуватись таких заходів безпеки:

- пересування людей і автомобілів дозволяється тільки позначеними та розвіданими шляхами;
- забороняється вести роботи біля конструкцій, які загрожують падінням;
- зберігати режим радіаційного та хімічного захисту;
- проведення робіт у задимлених та загазованих приміщеннях групами по 2—3 чоловіки в особистих засобах захисту;
- проведення робіт на електромережах, електроустановках тільки після їх відключення і заземлення;
- освітлення ділянок роботи вночі та за несприятливої погоди.

Під стійкістю роботи об'єкта народного господарства розуміють здатність підприємства, установи попереджувати виникнення виробничих аварій, катастроф, протистояти впливу уражаючих факторів, аби запобігти або зменшити загрозу життю і здоров'ю робітників і службовців, матеріальних втрат, а також забезпечити відновлення порушеного виробництва в мінімально короткий термін.

Стійка робота промислового підприємства складається:

- зі стійкості інженерно-технічного комплексу (будівель, споруд, систем енерго-, газо-, водозабезпечення тощо) до дії зовнішніх факторів при аваріях, катастрофах, а також при застосуванні щодо них сучасної зброї;
- зі стійкості виробничої діяльності (захист виробничого персоналу, надійність систем управління, поновлення роботи у стислі терміни).

Фактори, від яких залежить стійка робота об'єктів у НС мирного і воєнного часу:

- Надійність захисту робітників і службовців.
- Безпечність розташування об'єкта щодо зон масштабних зруйнувань.
- Можливість інженерно-технічного комплексу протистояти Уражаючим діям сучасної військової зброї.
- Безперервність постачання електроенергією, паливом, газом І всім необхідним для випуску продукції.
- Надійність керування виробництвом, силами і засобами ЦО.
- Підготовленість підприємства до поновлення виробництва І проведення рятувальних робіт.

4.4 Висновки до четвертого розділу

Таким чином, у результаті аналізу вимог щодо проведення рятувальних та інших невідкладних робіт на об'єкті господарської діяльності в осередку ураження.

ВИСНОВКИ

В ході виконанні кваліфікаційної робота освітнього рівня «Магістр» було охарактеризовано методику аналізу та виявлення вразливостей у комп’ютерних системах.

Було розроблена система, яка може автономно отримувати дані про нові вразливості із соціальних мереж Twitter та Reddit, які виникають та опрацювавши їх сповіщати користувача в момент знаходження вразливості у одному із компонентів його системи.

В рамках проекту створено модель опису на основі властивостей, яка підтримує аналізу та опису вразливостей веб-додатків. Поєднання інформації з декількох джерел та вихідних даних сканерів поєднання інформації з декількох джерел та результатів роботи сканерів може дати вичерпний опис вразливості як на абстрактному, так і на детальному рівні.

Оцінку та сканування вразливостей слід проводити на регулярній основі - IT-середовища постійно змінюються (наприклад, оновлення програмного забезпечення або зміна конфігурації системи може призвести до появи нової вразливості), а нові загрози продовжують з'являтися, тому важливо швидко виявляти та усувати вразливості, щоб обмежити ризики кібербезпеки.

Сканування вразливостей є лише частиною оцінки вразливостей - інші процеси, такі як тестування на проникнення, можуть виявити різні типи загроз для IT у вашій організації. Тестування на проникнення доповнює сканування вразливостей і корисно для визначення того, чи можна впливати на вразливість, і чи не призведе ця дія до пошкодження, втрати даних або інших проблем.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1 - M. Cova, V. Felmetsger, and G. Vigna, "Vulnerability Analysis of Web-based Applications," in Test and Analysis of Web Services, L. Baresi and E. D. Nitto, Eds., ed: Springer Berlin Heidelberg, 2007, с. 363-394.

2 - R. P. Abbott, J. S. Chin, J. E. Donnelley, W. L. Konigsford, S. Tokubo, and D. A. Webb, "Security Analysis and Enhancements of Computer Operating Systems," Lawrence Livermore Laboratory, Technical Report TR NBSIR-76- 1041, Квітень 1976

3 - C. E. Landwehr, A. R. Bull, J. P. McDermott, and W. S. Choi, "A Taxonomy of Computer Program Security Flaws, with Examples," Center for Computer High Assurance Systems Information Technology Division, Technical Report NRL/FR/5542--93-9591, 1994.

4 - M. Bishop, "A Taxonomy of UNIX System and Network Vulnerabilities," Department of Computer Science University of California at Davis, Technical Report SCE-95-10, Травень 1995.

5 - M. Bishop, "Vulnerabilities Analysis," in the 2nd International Workshop on Recent Advances in Intrusion Detection (RAID'99), West Lafayette, Indiana, USA, 1999, с. 125-136, doi: 10.1.1.39.6062.

6 - S. Neuhaus, T. Zimmermann, C. Holler, and A. Zeller, "Predicting vulnerable software components," presented at the 14th ACM Conference on Computer and Communications Security, Alexandria, Virginia, USA, 2007.

7 - S. Christey and C. Harris. Introduction to Vulnerability Theory (Version 1.0.1 ed.) [draft]. Available: http://cwe.mitre.org/documents/vulnerability_theory/ intro.html. Last accessed: Грудень 2010.

8 - SecureEnterprise2.0Forum, "Top Web 2.0 Security Threats," Security Enterprise 2.0 Forum, Industry Report 17 Лютий 2009.

9 - J. Steffan and M. Schumacher, "Collaborative Attack Modeling," in the 2002 ACM Symposium on Applied Computing (SAC 2002), Madrid, Spain, 2002, c. 253-259, doi: 10.1145/508791.508843.
10 - C. V. Berghe, J. Riordan, and F. Piessens, "A Vulnerability Taxonomy Methodology applied to Web Services," in the 10th Nordic Workshop on Secure IT-systems (NORDSEC 2005), Tartu, Estonia, 2005, doi: 10.1.1.61.286.
11 - M. D. Aime, A. Atzeni, and P. C. Pomi, "The Risks With Security Metrics," presented at the 4th ACM Workshop on Quality of Protection, Alexandria, Virginia, USA, 2008.
12 - L. P. Swiler, C. Phillips, and T. Gaylor, "A Graph-Based NetworkVulnerability Analysis System," Sandia National Labs., Albuquerque, NM (United States), Technical Report 01 Січень 1998
13 - M. Christodorescu, S. Jha, and C. Kruegel, "Mining specifications of malicious behavior," presented at the 1st Conference on India Software Engineering, Hyderabad, India, 2008.
14 - W. G. J. Halfond, A. Orso, and P. Manolios, "Using Positive Tainting and Syntax-aware Evaluation to Counter SQL Injection Attacks," in the 14th ACM SIGSOFT International Symposium on Foundations of Software Engineering (ACM SIGSOFT 2006/FSE-14), Portland, Oregon, USA, 2006, c. 175-185, doi: 10.1145/1181775.1181797.
15 M. Cova, V. Felmetsger, and G. Vigna, "Vulnerability Analysis of Web-based Applications," in Test and Analysis of Web Services, L. Baresi and E. D. Nitto, Eds., ed: Springer Berlin Heidelberg, 2007, c. 363-394.

Додаток А – апробація наукової роботи

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ІВАНА ПУЛЮЯ

М А Т Е Р І А Л И

X НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

«ІНФОРМАЦІЙНІ МОДЕЛІ, СИСТЕМИ ТА ТЕХНОЛОГІЇ»

7–8 грудня 2022 року

ТЕРНОПІЛЬ
2022

УДК 004.056
Д. Урбан
(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

АНАЛІЗ ЗАГРОЗ КОМП'ЮТЕРНИХ СИСТЕМ

UDC 004.056
D. Urban

ANALYSIS OF COMPUTER SYSTEM THREATS

Уразливість нульового дня – це слабке місце в комп'ютерній системі, яким може скористатися зловмисник і яке не помічають уражені сторони. Атака нульового дня – це спроба загрози проникнути, пошкодити або іншим чином скомпрометувати систему, уражену невідомою вразливістю. За характером нападу жертва не матиме засобів захисту, тому ймовірність успіху є високою.

Професіонали з IT-безпеки ще ніколи не стикалися з такими загрозами, чи то від величезного зростання віддаленої роботи, чи від агресивних хакерів, спонсорованих національною державою, таких як ті, хто причетний до зламу SolarWinds. Незважаючи на те, що завжди знайдуться нові діри, які потрібно закрити, уразливості системи безпеки зазвичай виникають через кілька тих самих причин: невиправлені вразливості, неправильні конфігурації чи помилки користувача, і навіть найбільш технічно підковані компанії вразливі до цих помилок.

Оскільки комп'ютери та інші цифрові пристрої стали важливими для бізнесу та торгівлі, вони також дедалі частіше стають об'єктами атак. Для того, щоб компанія чи окрема особа могли впевнено використовувати комп'ютерний пристрій, вони спочатку повинні бути впевнені, що пристрій жодним чином не скомпрометовано та що всі комунікації будуть безпечними.[1]

Кіберфізичні системи систем (SoSs) – це великомасштабні системи, створені з незалежних і автономних кіберфізичних складових систем (КС), які можуть взаємодіяти для досягнення цілей високого рівня також за втручання людей. Забезпечення безпеки в таких SoSs означає, серед інших функцій, прогнозування та передбачення розвитку функціональних можливостей SoSs, зрештою виявлення можливих шкідливих явищ, які можуть виникнути в результаті взаємодії КС та людей. Такі явища, які зазвичай називають емерджентними явищами, часто є складними і їх важко зафіксувати: перша поява емерджентного явища в кіберфізичному SoSs часто є несподіванкою для спостерігачів. Адекватна підтримка для розуміння явища, що виникає, допоможе зменшити як ймовірність проектних або експлуатаційних недоліків, так і час, необхідний для аналізу відносин між КС, що завжди має ключове економічне значення. Проте не варто вважати, що використання IDS та автоматизація аналізу log-файлів дозволить виявити всі загрози безпеки. Кожен засіб захисту адресовано конкретній загрозі безпеки в системі. Більше того, кожен засіб захисту має слабкі та сильні сторони. Тільки правильно підібравши та налаштувавши ці засоби, можна захиститися від максимально великого спектру атак. [2]

Література

1. Security of Cyber-Physical Systems. URL: https://www.powermag.com/security-of-cyber-physical-systems/.
2. What is a Zero-Day Exploit? URL: https://www.intel.com/content/www/us/en/business/enterprise-computers/resources/what-is-a-zero-day-exploit.html.

Додаток Б – лістинг програми main.py

```
import requests
import json
import re
import telebot
import time

from constants import BOT_TOCKEN

bot = telebot.TeleBot(BOT_TOCKEN)

def get_twitter_page():
    """
    Check for new CVE that program has received
    """
    url =
"https://twitter.com/i/api/graphql/zEAmYdYUvRuhrr9zjc5Vvg/UserTw
eets?" \

"variables=%7B%22userId%22%3A%2282180628746174054 4%22%2C%22count
%22%3A40%2C%22includePromotedContent%22%3Atrue%2C%22" \

"withQuickPromoteEligibilityTweetFields%22%3Atrue%2C%22withSuper
FollowsUserFields%22%3Atrue%2C%22" \

"withDownvotePerspective%22%3Afalse%2C%22withReactionsMetadata%2
2%3Afalse%2C%22" \

"withReactionsPerspective%22%3Afalse%2C%22withSuperFollowsTweetF
ields%22%3Atrue%2C%22withVoice%22%3Atrue%2C%22" \

"withV2Timeline%22%3Atrue%7D&features=%7B%22responsive_web_twitt
er_blue_verified_badge_is_enabled%22%3Atrue%2C%22" \
```

```
"verified_phone_label_enabled%22%3Afalse%2C%22responsive_web_gra
phql_timeline_navigation_enabled%22%3Atrue%2C%22" \

"view_counts_public_visibility_enabled%22%3Afalse%2C%22view_coun
ts_everywhere_api_enabled%22%3Afalse%2C%22" \

"tweetypie_unmention_optimization_enabled%22%3Atrue%2C%22respons
ive_web_uc_gql_enabled%22%3Atrue%2C%22vibe_api_enabled%22%3Atrue
%2C%22"\

"responsive_web_edit_tweet_api_enabled%22%3Atrue%2C%22graphql_is
_translatable_rweb_tweet_is_translatable_enabled%22%3Atrue%2C%22
" \

"standardized_nudges_misinfo%22%3Atrue%2C%22tweet_with_visibilit
y_results_prefer_gql_limited_actions_policy_enabled%22%3Afalse%2
C%22" \

"interactive_text_enabled%22%3Atrue%2C%22responsive_web_text_con
versations_enabled%22%3Afalse%2C%22responsive_web_enhance_cards_
enabled%22%3Atrue%7D"

    headers = {
        'authorization': 'Bearer
AAAAAAAAAAAAAAAAAAAAANRILgAAAAAAnNwIzUejRCOuH5E6I8xnZz4puTs%3D1Z
v7ttfk8LF81IUq16cHjhLTvJu4FA33AGWWjCpTnA',
        'cookie':
'_twitter_sess=BAh7CSIKZmxhc2hJQzonQWN0aW9uQ29udHJvbGxlcjo6Rmxhc
2g6OkZsYXNo%250ASGFzaHsABjoKQHVzZWR7ADoPY3JlYXRlZF9hdGwrCMe6LVRz
AToMY3NyZl9p%250AZCIlYjI3YjAwNTZkZDg1ZDA2MDI2NzA3ZGY0YTA3ZGVkNDM
6B2lkIiU0N2Fl%250AZGM5MzIzMmM1NDJmMzZhZTk0OTk1NTRlM2MwYw%253D%25
3D--bdf78d74648ba7ee09ffd6ce2b73d23bcae38145;
guest_id_marketing=v1%3A159484514908990613;
guest_id_ads=v1%3A159484514908990613;
```

```
_ga=GA1.2.881263206.1644676930;
personalization_id="v1_Df16969CUlQSEtvFFr5qdw==";
guest_id=v1%3A165843030332732125; g_state={"i_l":0};
kdt=gPHU7qNgaYeTTTIey00GQkochPWL61DF8xynjn6x;
auth_token=8e85b3b99cd32a271b7d629e9722ba48f33d4f04;
ct0=a2311fc59827ef8e4efd92e34477cebba914fb1a57595eb5011472e0921c
77bc6436831ae9e906776d53552f538aac99dc0f0a4bee5e24beb907428fc979
9bd77b345d97c59a61483761b8065a7aaee5;
twid=u%3D992713660756832257; _gid=GA1.2.1856984179.1671378609;
external_referer=padhuUp37zj9327dHdO0nT0Mc%2Bdf1I7x|0|8e8t2xd8A2
w%3D',
        'x-csrf-token':
'a2311fc59827ef8e4efd92e34477cebba914fb1a57595eb5011472e0921c77b
c6436831ae9e906776d53552f538aac99dc0f0a4bee5e24beb907428fc9799bd
77b345d97c59a61483761b8065a7aaee5'
    }

    response = requests.request("GET", url, headers=headers)

    result_json = json.loads(response.text)

    return result_json


def cve_tracker(result_json):
    """
    Check for new CVE that program has received
    """
    parsed_data = open("parsed_data.json", "r")
    parsed_data = json.loads(parsed_data.read())

    new_vulnerability_list = []
    for twitt in
result_json['data']['user']['result']['timeline_v2']['timeline']
['instructions'][1]['entries']:
```

```
        if "tweet" in twitt['entryId']:
            if
twitt['content']['itemContent']['tweet_results']['result']['lega
cy']['full_text'] not in parsed_data:
                new_vulnerability_list.append(twitt)

    return new_vulnerability_list


def reddit_get():
    """
    Get data from reddit and convert them into useful format
    """
    url =
"https://www.reddit.com/r/cybersecurity/?f=flair_name%3A%22New%2
0Vulnerability%20Disclosure%22"

    response = requests.request("GET", url)

    search_data = re.search("window.___r = ({.+?(?=};)})",
response.text).group(1)

    json_search = json.loads(search_data)

    return json_search


def post_check(json_search):
    """
    Check for new post that program has received
    """
    parsed_data = open("parsed_data1.json", "r")
    parsed_data = json.loads(parsed_data.read())

    new_posts = []
```

```
    for post in json_search['posts']['models']:
        if json_search['posts']['models'][post]['title'] not in
parsed_data:

new_posts.append(json_search['posts']['models'][post])

    return new_posts



def user_check_reddit(new_posts):
    """
    Check all new posts info if we need to notify users
    """
    parsed_data = open("parsed_data.json", "r")
    parsed_data = json.loads(parsed_data.read())

    user_data = open("users_data.json", "r")
    user_data = json.loads(user_data.read())

    for post in new_posts:
        text_value = post['title']
        if post.get("source"):
            url = post['source']['url']
        elif post.get("media"):
            url = post['media']['markdownContent']

        for user in user_data:
            for keyword in user['keywords']:
                if re.search(f"[,.\s/]*{keyword}[,.\s/]*",
text_value.lower()):
                    call_user(text_value, url, user['id'],
keyword)
                    break
        parsed_data.append(text_value)
```

```
    json.dump(parsed_data, open("parsed_data.json", "w"),
indent=4)



def user_check(new_vulnerability_list):
    """
    Check all new vulnerabilities info if we need to notify
users
    """
    parsed_data = open("parsed_data.json", "r")
    parsed_data = json.loads(parsed_data.read())

    user_data = open("users_data.json", "r")
    user_data = json.loads(user_data.read())

    for vulnerability in new_vulnerability_list:
        if "tweet" in vulnerability['entryId']:
            text_value =
vulnerability['content']['itemContent']['tweet_results']['result
']['legacy']['full_text']
            url =
vulnerability['content']['itemContent']['tweet_results']['result
']['legacy']['entities']['urls'][0]['expanded_url']
            for user in user_data:
                for keyword in user['keywords']:
                    if re.search(f"[,.\s]*{keyword}[,.\s]*",
text_value):
                        call_user(text_value, url, user['id'],
keyword)
                        break
            parsed_data.append(text_value)
```

```
    json.dump(parsed_data, open("parsed_data.json", "w"),
indent=4)


def call_user(text_value, url, user_id, keyword):
    """
    Send information to the user
    """
    message = f"There is a new vulnerability for {keyword}.
\n=====\n {text_value} \n=====\n {url}"
    bot.send_message(chat_id=user_id, text=message)


def main_function():
    result_json = get_twitter_page()

    new_vulnerability_list = cve_tracker(result_json)

    user_check(new_vulnerability_list)

    json_search = reddit_get()

    new_post = post_check(json_search)

    user_check_reddit(new_post)


while True:
    main_function()
    time.sleep(300)
```