Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(повне найменування вищого навчального закладу)
Факультет комп'ютерно-інформаційних систем і програмної інженерії
(назва факультету)
Кафедра кібербезпеки
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр
(освітній рівень)

на тему: "Аналіз ефективності програмного забезпечення для виявлення шкідливого коду"

Виконав: студент

Спеціальності:
125 «Кібербезпека»
(шифр і назва напряму підготовки, спеціальності)
Заїка Андрій Олександрович
підпис (прізвище та ініціали)

Керівник Муж В.В.
підпис (прізвище та ініціали)

Нормоконтроль Тимощук Д.І.
підпис (прізвище та ініціали)

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

Рецензент
підпис (прізвище та ініціали)

м. Тернопіль – 2024

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)
«__» _____ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 125 Кібербезпека
(шифр і назва спеціальності)

Студенту Заїці Андрію Олександровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Аналіз ефективності програмного забезпечення для виявлення шкідливого коду

Керівник роботи Муж Валерій Вікторович, доцент кафедри кібербезпеки
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «03» 04 2024 року № 4/7-349

2. Термін подання студентом завершеної роботи

3. Вихідні дані до роботи Методи аналізу для виявлення шкідливого коду

4. Зміст роботи (перелік питань, які потрібно розробити)
Методи аналізу програмного забезпечення
Класифікація методів аналізу
Методи тестування програмного забезпечення
Практичне застовування методів
Безпека життєдіяльності, основи охорони праці
Висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)
Тема, мета, задачі. Класифікація шкідливого програмного забезпечення. Методи виявлення шкідливого коду. Програмні рішення для виявлення шкідливого коду. Критерії оцінки ефективності програмного забезпечення для виявлення шкідливого коду. Дослідження та тестування програмного забезпечення для виявлення шкідливого коду. Практичне застосування методів. Висновки

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи хорони праці			

7. Дата видачі завдання 29.01.2024

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	16.02 – 19.02	Виконано
2.	Визначення теми та мети дипломної роботи	20.02 – 27.02	Виконано
3.	Ознайомлення з літературою та джерелами	28.02 – 16.03	Виконано
4.	Вивчення класифікації шкідливого програмного забезпечення	17.03 – 20.03	Виконано
5.	Вивчення методів виявлення шкідливого коду	20.03 – 05.04	Виконано
6.	Аналіз програмних рішень для виявлення шкідливого коду	06.03 – 17.04	Виконано
7.	Розробка критеріїв оцінки ефективності програмного забезпечення	18.04 – 29.04	Виконано
8.	Дослідження та тестування програмного забезпечення	30.04 – 13.05	Виконано
9.	Аналіз результатів тестування та розробка рекомендацій	14.05 – 21.05	Виконано
10.	Оформлення кваліфікаційної роботи	22.05 – 05.06	Виконано
11.	Перевірка на плагіат кваліфікаційної роботи	06.06 – 12.06	Виконано
12.	Попередній захист кваліфікаційної роботи	18.06 – 21.06	Виконано
13.	Захист кваліфікаційної роботи	27.06.2024	

Студент _____ Заїка А.О.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Муж В.В.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Аналіз ефективності програмного забезпечення для виявлення шкідливого коду// Кваліфікаційна робота ОР «Бакалавр» //Заїка Андрій Олександрович// Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра кібербезпеки, група СБ-41 // Тернопіль, 2024 // С. 60 , рис. – 16 , табл. – , кресл. –__ , додат. – ___.

КЛЮЧОВІ СЛОВА: аналіз, вірус, програмне забезпечення, код.

Кваліфікаційна робота присвячена аналізу ефективності програмного забезпечення для виявлення шкідливого коду. Метою дослідження є провести аналіз та порівняти ефективність різних програмних продуктів для виявлення шкідливого коду. Предметом дослідження кваліфікаційної роботи є ефективність програмних продуктів для виявлення шкідливого коду.

Актуальність теми дослідження виявлення шкідливого коду важко переоцінити, враховуючи стрімкий розвиток інформаційних технологій та зростання кіберзагроз. Шкідливе програмне забезпечення (шкідливий код) є серйозною загрозою для безпеки інформаційних систем, зокрема для персональних комп'ютерів, корпоративних мереж і державних інфраструктур.

У першому розділі кваліфікаційної роботи описані класифікації шкідливого ПЗ, описані методи як виявляти шкідливий код, і якими методами можна боротись з ним.

У другому розділі були розглянуті методи проведення дослідження та тестування програмного забезпечення для виявлення шкідливого коду.

У третьому розділі кваліфікаційної роботи описано роботу програм які працюють за допомогою методів які описані в першому розділі. Також був проведений порівняльний аналіз сервісів які були описані, в третьому розділі.

ANNOTATION

Analysis of the effectiveness of software for the detection of malicious code// Qualification work of OR "Bachelor" // Zaika Andrii Oleksandrovich // Ternopil National Technical University named after Ivan Pulyu, Faculty of Computer Information Systems and Software Engineering, Department of Cyber Security, Group SB-41 / / Ternopil, 2024 // P. 60, fig.– 16, tab. - , chair. –__ , add. - ___.

Keywords: analysis, virus, software, code.

The qualification work is devoted to the analysis of the effectiveness of software for detecting malicious code. The purpose of the study is to analyze and compare the effectiveness of various software products for detecting malicious code. The subject of the qualification work research is the effectiveness of software products for detecting malicious code.

The relevance of the research topic of malicious code detection is difficult to overestimate, given the rapid development of information technologies and the growth of cyber threats. Malicious software (malicious code) is a serious threat to the security of information systems, in particular to personal computers, corporate networks and government infrastructures.

In the first section of the qualification work, the classifications of malicious software are described, the methods for detecting malicious code are described, and what methods can be used to combat it

In the second chapter, the methods of research and testing of software to detect malicious code were considered

The third section of the qualification work describes the operation of programs that work using the methods described in the first section. A comparative analysis of the services described in the third section was also conducted.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ПЗ — Програмне забезпечення

СЗВ — Система запобіганню вторгнень

СВВ — Система виявлення вторгнення

СВА — Система виявлення аномалій

ПЗТ — Програмне забезпечення для тестування

ШПЗ — Шкідливе програмне забезпечення

АПЗ — Антивірусне програмне забезпечення

ВСТУП

У сучасному світі, де комп'ютерні технології пронизали всі аспекти життя, кіберзагрози стають все більш небезпечними та витонченими. Шкідливе програмне забезпечення, таке як віруси, хробаки, трояни, шпигунські програми та програмне забезпечення для викупу, може завдати значної шкоди комп'ютерним системам, даним та особистій інформації. Тому використання ефективного програмного забезпечення для виявлення шкідливого коду є критично важливим для захисту комп'ютерних систем та мереж від кібератак.

Зростання кіберзагроз робить цю тему актуальною для всіх верств населення. Приватні користувачі, які використовують комп'ютери для роботи, особистих цілей чи розваг, ризикують втратити дані, зіткнутися з порушенням конфіденційності та зазнати фінансових збитків внаслідок кібератак. Підприємства можуть зазнати значних фінансових втрат через простої, пошкодження даних та втрату репутації. Витік даних може призвести до штрафів та санкцій з боку регулюючих органів. Кібератаки на державні установи можуть поставити під загрозу національну безпеку, порушити роботу критичної інфраструктури та підірвати довіру до влади.

Метою цієї кваліфікаційної роботи є аналіз ефективності різних програмних рішень для виявлення шкідливого коду. Для досягнення цієї мети буде проведено комплексне дослідження, яке включатиме виконання завдань:

- Проаналізувати ефективність різних програмних рішень для виявлення шкідливого коду.
- Визначити найбільш ефективні програмні рішення для різних типів користувачів та організацій.
- Провести порівняльний аналіз щодо вибору та використання програмного забезпечення для виявлення шкідливого коду.

1 ТЕОРЕТИЧНІ ОСНОВИ АНАЛІЗУ ЕФЕКТИВНОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ВИЯВЛЕННЯ ШКІДЛИВОГО КОДУ

1.1 Класифікація шкідливого програмного забезпечення

Шкідливе програмне забезпечення, також відоме як "шкідливий код", може бути класифіковано за різними критеріями, такими як його мета, функціональність, метод поширення та вплив на систему[4].

Як їх класифікують за метою:

- Шкідливі програми для крадіжки даних. Ці програми призначені для крадіжки особистої інформації, фінансових даних або іншої конфіденційної інформації з заражених систем. До цієї категорії належать шпигунські програми, кейлоггери, програмне забезпечення для викрадення паролів та програми для зняття даних з банківських карт[4].
- Шкідливі програми для руйнування. Ці програми призначені для пошкодження або знищення даних, систем або мереж. До цієї категорії належать віруси, хробаки, трояни, програмне забезпечення для витирання даних та програмне забезпечення для DDoS-атак[4].
- Шкідливі програми для порушення роботи. Ці програми призначені для порушення нормальної роботи комп'ютерних систем або мереж. До цієї категорії належать рекламні програми, браузерні викрадачі, програмне забезпечення для блокування файлів та програмне забезпечення для виведення з ладу[4].
- Шкідливі програми для шантажу. Ці програми призначені для вимагання грошей або інших цінностей у власників заражених систем. До цієї категорії належать програмне забезпечення для викупу, криптоджекери та здирницькі програми[4].

Як їх класифікують за функціональністю:

- Віруси - це самовідтворювані програми, які можуть заражати інші файли або програми. Вони можуть поширюватися через електронну пошту, мережеві підключення або знімні носії[4].
- Хробаки - це самовідтворювані програми, які можуть поширюватися через мережі без втручання користувача. Вони можуть розсилати спам, викрадати дані або руйнувати системи[4].
- Трояни - це програми, які маскуються під легітимні програми, щоб обдурити користувачів на їх встановлення. Після встановлення вони можуть виконувати шкідливі дії, такі як крадіжка даних, пошкодження систем або завантаження іншого шкідливого програмного забезпечення[4].
- Програмне забезпечення для викупу шифрує дані користувачів і вимагає викупу за їх розшифровку[4].
- Криптоджекери - це програми, які використовують ресурси комп'ютера користувача для видобутку криптовалюти без його згоди[4].
- Здирницькі програми погрожують публікацією або знищенням даних користувача, якщо він не заплатить викуп[4].

Класифікація за впливом на систему**:**

- Шкідливі програми низького рівня. Ці програми не завдають значної шкоди системі та можуть бути легко видалені[4].
- Шкідливі програми середнього рівня. Ці програми можуть пошкодити файли або системні налаштування, але зазвичай не призводять до втрати даних або непрацездатності системи[4].
- Шкідливі програми високого рівня. Ці програми можуть завдати значної шкоди системі, призвести до втрати даних або зробити систему непрацездатною[4].

Класифікація шкідливого програмного забезпечення за методом поширення.

Метод поширення є одним з ключових факторів, які визначають, як шкідливе програмне забезпечення потрапляє на комп'ютери та заражає їх. Знання про різні

методи поширення може допомогти користувачам та адміністраторам систем вжити відповідних заходів для запобігання зараженню[4].

Ось деякі з найпоширеніших методів поширення шкідливого програмного забезпечення.

Електронна пошта.

За допомогою електронної пошти можна відправляти зараженні вкладення та фішингові листи. Заражені вкладення можуть поширювати ШПЗ, такі як файли Microsoft Office, PDF-файли або архіви. Коли користувач відкриває заражене вкладення, шкідливий код запускається та заражає комп'ютер. Фішингові листи - це електронні листи, які намагаються обдурити користувачів, щоб вони розкрили особисту інформацію або завантажили заражені файли. Ці листи часто маскуються під повідомлення від законних компаній або організацій.

Мережеві підключення:

- Заражені веб-сайти. Шкідливе програмне забезпечення може поширюватися через заражені веб-сайти. Коли користувач відвідує заражений веб-сайт, шкідливий код може автоматично завантажитися на його комп'ютер[4].
- Завантаження файлів. Шкідливе програмне забезпечення може поширюватися через завантаження файлів з ненадійних джерел, таких як файлообмінні мережі або піратські веб-сайти[4].
- Мережеві атаки. Зловмисники можуть використовувати мережеві атаки, щоб проникнути в комп'ютерні системи та встановити шкідливе програмне забезпечення[4].

Знімні носії.

Шкідливе програмне забезпечення може поширюватися через заражені USB-накопичувачі. Коли користувач підключає заражений USB-накопичувач до свого комп'ютера, шкідливий код може автоматично запуститися. Шкідливе програмне забезпечення може поширюватися через заражені DVD-диски та CD-диски. Коли користувач відтворює заражений диск, шкідливий код може запуститися та заразити комп'ютер[4].

Соціальні мережі.

Шкідливі посилання можуть поширюватися через соціальні мережі. Коли користувач клікає на таке посилання, він може бути перенаправлений на заражений веб-сайт або завантажити шкідливий файл. Фішингові повідомлення можуть поширюватися через соціальні мережі. Ці повідомлення часто маскуються під повідомлення від друзів або знайомих і можуть спонукати користувачів розкрити особисту інформацію або завантажити заражені файли[4].

Слабкі паролі.

Шкідливі програми можуть використовувати слабкі паролі для отримання доступу до комп'ютерних систем. Зловмисники можуть використовувати автоматизовані інструменти для перебору паролів або отримати доступ до паролів через соціальну інженерію чи фішингові атаки[4].

Важливо знати про різні методи поширення шкідливого програмного забезпечення та вживати відповідних заходів для захисту своїх комп'ютерів. Користувачі повинні використовувати надійні паролі, уникати відкриття невідомих вкладення та завантаження файлів з ненадійних джерел, а також оновлювати програмне забезпечення та операційну систему.

Адміністратори систем повинні використовувати брандмауери, системи виявлення вторгнень та інші засоби безпеки для захисту своїх мереж. Знання про класифікацію шкідливого програмного забезпечення за методом поширення може допомогти вам вибрати правильні засоби захисту та мінімізувати ризик зараження.

1.2 Методи виявлення шкідливого коду

Існує багато різних методів виявлення шкідливого коду, кожен з яких має свої переваги та недоліки. Найпоширеніші методи включають.

Сигнатурний аналіз.

Сигнатурний аналіз є одним з найпростіших та найпоширеніших методів виявлення шкідливого коду. Він ґрунтується на порівнянні файлів або фрагментів коду з відомими зразками шкідливого програмного забезпечення, які зазвичай

називають "сигнатурами". Ці сигнатури генеруються на основі специфічних характеристик шкідливого програмного забезпечення, таких як фрагменти коду, хеш-суми або інші унікальні маркери[5].

Переваги:

- Простота та легкість реалізації. Сигнатурний аналіз є простим у розумінні та реалізації методом, що робить його доступним для широкого кола користувачів[5].
- Швидкість. Сканування за допомогою сигнатурного аналізу може бути дуже швидким, адже порівняння файлів з відомими зразками відбувається швидко[5].
- Ефективність проти відомих загроз. Сигнатурний аналіз є дуже ефективним проти відомих загроз, адже він може чітко ідентифікувати шкідливе програмне забезпечення, для якого вже існують сигнатури[5].

Недоліки:

- Неефективність проти нових загроз. Сигнатурний аналіз не може виявити нові або невідомі зразки шкідливого програмного забезпечення, адже для них ще не створено сигнатур[5].
- Залежність від оновлень. Ефективність сигнатурного аналізу залежить від частоти оновлення сигнатурних баз. Якщо сигнатурні бази не оновлюються регулярно, нові загрози можуть залишатися невизнаними[5].
- Можливість помилкових спрацьовувань. Сигнатурний аналіз може іноді помилково ідентифікувати легітимні файли як шкідливі, що може призвести до блокування доступу до них або їх видалення[5].

Евристичний аналіз.

Евристичний аналіз ґрунтується на загальних характеристиках та поведінці шкідливого програмного забезпечення, а не на порівнянні з відомими зразками. Цей метод використовує правила та алгоритми, які аналізують код, структуру та поведінку файлів, щоб визначити, чи мають вони ознаки шкідливості[2].

Переваги:

- Здатність виявляти нові загрози. Евристичний аналіз може виявляти нові та невідомі зразки шкідливого програмного забезпечення, адже він не залежить від наявності сигнатур[2].
- Зниження кількості помилкових спрацьовувань. Евристичний аналіз, як правило, має меншу кількість помилкових спрацьовувань, ніж сигнатурний аналіз, адже він не ідентифікує легітимні файли лише на основі їх схожості з відомими зразками шкідливого програмного забезпечення[2].

Недоліки:

- Складність та висока вартість. Евристичний аналіз може бути складним у реалізації та потребувати значних ресурсів, що робить його менш доступним для широкого кола користувачів.
- Можливість помилкових спрацьовувань. Евристичний аналіз іноді може помилково ідентифікувати легітимні файли як шкідливі, адже він ґрунтується на загальних характеристиках та поведінці, які можуть бути присутні і в легітимних програмах.
- Необхідність постійного вдосконалення. Евристичний аналіз потребує постійного вдосконалення правил та алгоритмів для того, щоб адаптуватися до нових загроз та зменшити кількість помилкових спрацьовувань[2].

Аналіз поведінки.

Аналіз поведінки ґрунтується на спостереженні за поведінкою програм під час їх виконання. Цей метод аналізує дії програм, такі як системні виклики, доступ до файлів та мережева активність, щоб визначити, чи мають вони ознаки шкідливості[3].

Переваги:

- Здатність виявляти нові та невідомі загрози. Аналіз поведінки може виявляти нові та невідомі зразки шкідливого програмного забезпечення, адже він не залежить від наявності сигнатур або специфічних характеристик шкідливого коду.

– Виявлення шкідливих програм, які використовують нові методи приховування. Аналіз поведінки може виявляти шкідливі програми, які намагаються приховати свою діяльність, використовуючи нові методи.
– Зниження кількості помилкових спрацьовувань. Аналіз поведінки, як правило, має меншу кількість помилкових спрацьовувань, ніж сигнатурний аналіз, адже він не ідентифікує легітимні файли лише на основі їх схожості з відомими зразками шкідливого програмного забезпечення[3].

Недоліки:

– Може генерувати багато помилкових спрацьовувань. Аналіз поведінки може генерувати багато помилкових спрацьовувань, адже деякі легітимні програми можуть мати поведінку, схожу на шкідливі.
– Потребує значних ресурсів. Аналіз поведінки може потребувати значних ресурсів, таких як потужні процесори та багато пам'яті, що робить його менш доступним для деяких користувачів.
– Складність аналізу. Аналіз поведінки може бути складним для аналізу та інтерпретації, що потребує кваліфікованих фахівців[3].

Статичний аналіз.

Статичний аналіз аналізує код програм без їх виконання. Цей метод може виявити потенційно шкідливий код, досліджуючи структуру коду, його змінні, типи даних та інші компоненти[13].

Переваги:

– Швидкість. Статичний аналіз може бути дуже швидким, адже він не потребує виконання коду.
– Здатність аналізувати великі обсяги коду. Статичний аналіз може аналізувати великі обсяги коду без значних ресурсів.
– Виявлення потенційних вразливостей. Статичний аналіз може виявити потенційні вразливості в коді, які можуть бути використані зловмисниками для створення шкідливого програмного забезпечення[13].

Недоліки:

- Статичний аналіз не може гарантувати, що код, який він ідентифікує як потенційно шкідливий, буде дійсно шкідливим.
- Статичний аналіз може пропустити деякі типи шкідливого коду, які використовують складні методи приховування.
- Аналіз результатів статичного аналізу може бути складним для інтерпретації, що потребує кваліфікованих фахівців[13].

Динамічний аналіз.

Динамічний аналіз аналізує код програм під час їх виконання. Цей метод може спостерігати за поведінкою коду в режимі реального часу та виявляти шкідливі дії[12].

Переваги:

- Динамічний аналіз може виявляти шкідливий код, який активується лише в певних умовах, наприклад, при виконанні певних дій або при доступі до певних ресурсів.
- Динамічний аналіз може аналізувати складні програми, які можуть бути складними для статичного аналізу.
- Динамічний аналіз може виявляти шкідливі програми, які намагаються приховати свою діяльність, використовуючи нові методи[12].

Недоліки:

- Динамічний аналіз може генерувати багато помилкових спрацьовувань, адже деякі легітимні програми можуть мати поведінку, схожу на шкідливі.
- Динамічний аналіз може потребувати значних ресурсів, таких як потужні процесори та багато пам'яті, що робить його менш доступним для деяких користувачів[12].

Самонавчання.

Методи машинного навчання та штучного інтелекту (ШІ) використовуються для розробки нових методів виявлення шкідливого коду. Ці методи можуть навчатися на великих наборах даних шкідливого та легітимного програмного забезпечення, щоб автоматично ідентифікувати нові та невідомі загрози[6].

Переваги:

- Самонавчані методи можуть виявляти нові та невідомі загрози, адже вони не залежать від наявності сигнатур або специфічних характеристик шкідливого коду.
- Самонавчані методи можуть адаптуватися до нових методів приховування, які використовують зловмисники для створення шкідливого програмного забезпечення.
- Самонавчані методи можуть мати меншу кількість помилкових спрацьовувань, ніж традиційні методи, адже вони навчаються на великих наборах даних, які містять як шкідливе, так і легітимне програмне забезпечення[6].

Недоліки:

- Розробка та впровадження самонавчаних методів може бути складною та дорогою, адже потребує значних ресурсів та експертних знань.
- Самонавчані методи залежать від якості та розміру навчальних даних. Якщо дані не містять достатньої кількості інформації про нові та невідомі загрози, методи можуть бути неефективними.
- Самонавчані методи можуть бути упередженими, якщо навчальні дані містять упередження. Це може призвести до помилкового ідентифікації легітимного програмного забезпечення як шкідливого[6].

1.3 Програмні рішення для виявлення шкідливого коду

Існує багато програмних рішень для виявлення шкідливого коду, які можна використовувати для захисту комп'ютерів та мереж від загроз. Ці рішення можна розділити на кілька категорій.

Антивірусні програми.

Найпоширеніший тип програмного забезпечення для виявлення шкідливого коду, дане ПЗ сканує ПК на наявність відомих вірусів, троянських програм, черв'яків та інших типів шкідливого програмного забезпечення. Часто

використовують сигнатурні бази даних, евристичний аналіз та інші методи для виявлення загроз.

Популярні АПЗ. ESET NOD32 Antivirus, Kaspersky Anti-Virus, Bitdefender Antivirus Plus, F-Secure[1].

Мережеві брандмауери.

Захищають комп'ютери та мережі від несанкціонованого доступу та атак. Контролюють вхідний та вихідний трафік, блокуючи шкідливі з'єднання та активність. Можуть використовуватися як програмні, так і апаратні рішення.

Приклади. ZoneAlarm Free Firewall, Comodo Firewall, Windows Defender Firewall[1].

Системи запобігання вторгненням (IDS).

Системи запобігання вторгненням (СЗВ) - це програмні або апаратні комплекси, які використовуються для виявлення та запобігання несанкціонованому доступу до комп'ютерних систем і мереж. Вони є важливою частиною багаторівневої стратегії кібербезпеки, допомагаючи захищати від кібератак, шкідливого програмного забезпечення та інших загроз[14].

Як працюють СЗВ.

СЗВ використовують різні методи для виявлення та запобігання вторгненням, включаючи:

- Аналіз сигнатур**.** Цей метод порівнює трафік з мережі з базою даних відомих сигнатур шкідливого програмного забезпечення та атак. Якщо виявлено відповідність, СЗВ може заблокувати трафік або вжити інших заходів.
- Аномальний аналіз**.** Цей метод аналізує трафік на предмет незвичайної активності, яка може свідчити про атаку. Наприклад, СЗВ може шукати трафік з невідомих IP-адрес або спроби отримати доступ до захищених ресурсів.
- Поведінковий аналіз**.** Цей метод аналізує поведінку програм на комп'ютері на предмет підозрілих дій. Наприклад, СЗВ може шукати програми, які намагаються отримати доступ до конфіденційних даних або розповсюдити шкідливе програмне забезпечення[14].

Типи СЗВ:

- СЗВ на основі хоста**.** Ці СЗВ встановлюються на кожен комп'ютер у мережі. Вони можуть захищати комп'ютер від атак, які націлені на нього безпосередньо.
- СЗВ на основі мережі. Ці СЗВ встановлюються на мережевих пристроях, таких як маршрутизатори та брандмауери. Вони можуть захищати всю мережу від атак[14].

Переваги використання СЗВ:

- Підвищення рівня безпеки. СЗВ можуть допомогти захистити комп'ютерні системи та мережі від кібератак, шкідливого програмного забезпечення та інших загроз.
- Зменшення часу простою. СЗВ можуть допомогти запобігти атакам, які можуть призвести до простою комп'ютерних систем.
- Зниження витрат. СЗВ можуть допомогти знизити витрати, пов'язані з кібератаками, такими як втрата даних, крадіжка особистих даних та відновлення систем[14].

Недоліки використання СЗВ:

- Неможливість виявлення всіх загроз. Жодна СЗВ не може виявити всі загрози. Нові загрози з'являються постійно, і СЗВ можуть не встигати за ними.
- Можливість помилок. СЗВ можуть помилково ідентифікувати легітимний трафік як шкідливий, що може призвести до блокування доступу до авторизованих користувачів або програм.
- Складність налаштування та обслуговування. СЗВ можуть бути складними для налаштування та обслуговування, що може потребувати значних ресурсів[14].

Додатки СЗВ.

Snort - це безкоштовна та загальнодоступна СЗВ з відкритим кодом, яка використовується для виявлення та запобігання мережевим атакам. Вона є однією з найпопулярніших СЗВ у світі та пропонує широкий спектр функцій, включаючи:

– Аналіз сигнатур.
– Аномальний аналіз.
– Поведінковий аналіз.
– Можливість створення власних правил.
– Підтримка різних операційних систем[14].

Suricata - це ще одна безкоштовна та загальнодоступна СЗВ з відкритим кодом, яка ґрунтується на Snort. Вона пропонує багато подібних функцій, але також має деякі вдосконалення, такі як:

– Покращена продуктивність.
– Підтримка багатоядерних процесорів.
– Можливість використання з іншими інструментами кібербезпеки[14].

Cisco Secure IPS - це комерційна СЗВ від Cisco Systems. Вона пропонує широкий спектр функцій, включаючи:

– Аналіз сигнатур.
– Аномальний аналіз.
– Поведінковий аналіз.
– Захист від атак на додатки.
– Інтеграція з іншими продуктами Cisco[14].

Deepwatch Defense Platform - це хмарна СЗВ, яка пропонує широкий спектр функцій, включаючи:

– Аналіз сигнатур.
– Аномальний аналіз.
– Поведінковий аналіз.
– Захист від атак на додатки.
– Машинне навчання.
– Штучний інтелект[14].

Palo Alto Networks Next-Generation Firewall (NGFW - це брандмауер нового покоління, який також включає функції СЗВ. Він пропонує широкий спектр функцій, включаючи:

- Аналіз сигнатур.
- Аномальний аналіз.
- Поведінковий аналіз.
- Захист від атак на додатки.
- Дешифрування та аналіз SSL/TLS трафіку[14].

Система виявлення вторгнень (IDS)

Система виявлення вторгнень (IDS) - це програмне або апаратне рішення, яке використовується для виявлення несанкціонованого доступу, використання або модифікації комп'ютерних систем і мереж. IDS - це важливий компонент багаторівневої стратегії кібербезпеки, який допомагає захищати від кібератак, шкідливого програмного забезпечення та інших загроз[14].

Як працює IDS.

IDS використовує різні методи для виявлення вторгнень, включаючи:

- Аналіз сигнатур. Цей метод порівнює трафік з мережі з базою даних відомих сигнатур шкідливого програмного забезпечення та атак. Якщо виявлено відповідність, IDS може попередити адміністраторів або вжити інших заходів.
- Аномальний аналіз. Цей метод аналізує трафік на предмет незвичайної активності, яка може свідчити про атаку. Наприклад, IDS може шукати трафік з невідомих IP-адрес або спроби отримати доступ до захищених ресурсів.
- Поведінковий аналіз. Цей метод аналізує поведінку програм на комп'ютері на предмет підозрілих дій. Наприклад, IDS може шукати програми, які намагаються отримати доступ до конфіденційних даних або розповсюдити шкідливе програмне забезпечення[14].

Типи IDS:

- IDS на основі хоста. Ці IDS встановлюються на кожен комп'ютер у мережі. Вони можуть захищати комп'ютер від атак, які націлені на нього безпосередньо.

– IDS на основі мережі. Ці IDS встановлюються на мережевих пристроях, таких як маршрутизатори та брандмауери. Вони можуть захищати всю мережу від атак[14].

Переваги використання IDS.

– Підвищення рівня безпеки. IDS може допомогти захистити комп'ютерні системи та мережі від кібератак, шкідливого програмного забезпечення та інших загроз.
– Зменшення часу простою. IDS може допомогти запобігти атакам, які можуть призвести до простою комп'ютерних систем.
– Зниження витрат. IDS може допомогти знизити витрати, пов'язані з кібератаками, такими як втрата даних, крадіжка особистих даних та відновлення систем[14].

Недоліки використання IDS.

– Неможливість виявлення всіх загроз. Жодна IDS не може виявити всі загрози. Нові загрози з'являються постійно, і IDS можуть не встигати за ними.
– Можливість помилок. IDS можуть помилково ідентифікувати легітимний трафік як шкідливий, що може призвести до блокування доступу до авторизованих користувачів або програм.
– Складність налаштування та обслуговування. IDS можуть бути складними для налаштування та обслуговування, що може потребувати значних ресурсів[14].

Системи виявлення аномалій (AD).

Система виявлення аномалій (СВА) - це тип системи кібербезпеки, яка використовується для виявлення підозрілих дій або активності в комп'ютерних системах і мережах. Вона відрізняється від систем виявлення вторгнень (IDS), які зосереджуються на відомих загрозах, тим, що СВА шукає відхилення від нормальної поведінки[14].

Як працює СВА.

СВА використовують різні методи для виявлення аномалій, включаючи:

– Статистичний аналіз. Цей метод аналізує дані про систему, такі як трафік мережі, використання процесора та активність користувачів, на предмет відхилень від середніх значень або баз даних нормальної поведінки.
– Машинне навчання. Цей метод використовує алгоритми машинного навчання для автоматичного вивчення нормальної поведінки системи та виявлення відхилень від неї.
– Аналіз поведінки. Цей метод аналізує поведінку користувачів та програм на предмет підозрілих дій, таких як спроби отримати доступ до конфіденційних даних або розповсюдити шкідливе програмне забезпечення[14].

Типи СВА:

– СВА на основі хоста. Ці СВА встановлюються на кожен комп'ютер у мережі. Вони можуть захищати комп'ютер від атак, які націлені на нього безпосередньо.
– СВА на основі мережі. Ці СВА встановлюються на мережевих пристроях, таких як маршрутизатори та брандмауери. Вони можуть захищати всю мережу від атак[14].

Переваги використання СВА:

– Можливість виявлення нових та невідомих загроз. СВА не залежать від баз даних відомих сигнатур, тому вони можуть виявляти нові та невідомі загрози, які можуть пропустити IDS.
– Зменшення кількості помилкових спрацьовувань. СВА, як правило, генерують менше помилкових спрацьовувань, ніж IDS, оскільки вони не шукають специфічні сигнатури.
– Гнучкість. СВА можна налаштувати для виявлення аномалій у будь-якому типі даних, що робить їх дуже гнучкими[14].

Недоліки використання СВА:

– Складність налаштування. СВА може бути складно налаштувати для виявлення аномалій без генерації помилкових спрацьовувань.

- Можливість пропуску загроз. СВА можуть пропускати загрози, які не відхиляються значно від нормальної поведінки.
- Висока вартість. Деякі СВА, особливо ті, що ґрунтуються на машинному навчанні, можуть бути дорогими[14].

Системи пісочниць.

Система пісочниць - це тип програмного забезпечення для кібербезпеки, який використовується для ізоляції та аналізу підозрілого коду або програмного забезпечення. Вона створює віртуальне середовище, відоме як "пісочниця", де можна запускати код без ризику для основної системи. Це дозволяє аналітикам з кібербезпеки досліджувати шкідливе програмне забезпечення та вивчати його поведінку, не ставлячи під загрозу свою систему або мережу[6].

Як працюють системи пісочниць:

- Запуск підозрілого коду**.** Підозрілий код або програмне забезпечення розміщується в пісочниці.
- Ізоляція. Пісочниця ізольована від основної системи, щоб запобігти поширенню шкідливого програмного забезпечення.
- Моніторинг. Система пісочниця відстежує активність коду, збираючи дані про його дії.
- Аналіз. Аналітики з кібербезпеки аналізують зібрані дані, щоб визначити, чи є код шкідливим.
- Нейтралізація. Якщо код визнаний шкідливим, він нейтралізується та видаляється з пісочниці[6].

Типи систем пісочниць:

- Системи на основі хоста. Ці системи встановлюються на кожен комп'ютер у мережі. Вони можуть захищати комп'ютер від шкідливого програмного забезпечення, яке націлене на нього безпосередньо.
- Системи на основі мережі. Ці системи встановлюються на мережевих пристроях, таких як маршрутизатори та брандмауери. Вони можуть захищати всю мережу від шкідливого програмного забезпечення, яке намагається проникнути в неї[6].

Переваги використання систем пісочниць:

- Захист від невідомих загроз. Системи пісочниць можуть виявляти нові та невідомі загрози, які можуть пропустити інші системи безпеки.
- Зменшення кількості помилкових спрацьовувань. Системи пісочниць, як правило, генерують менше помилкових спрацьовувань, ніж інші системи безпеки, оскільки вони не шукають специфічні сигнатури.
- Гнучкість. Системи пісочниць можна налаштувати для аналізу будь-якого типу коду або програмного забезпечення[6].

Недоліки використання систем пісочниць:

- Складність налаштування. Системи пісочниць може бути складно налаштувати для ефективного аналізу коду без генерації помилкових спрацьовувань.
- Висока вартість. Деякі системи пісочниць, особливо ті, що ґрунтуються на хмарних технологіях, можуть бути дорогими.
- Можливість пропуску загроз. Системи пісочниць можуть пропускати загрози, які не проявляють себе в пісочниці так само, як у реальній системі.
- Приклади. Cuckoo Sandbox, Palo Alto Networks WildFire[6].

Антивірусні програми для мобільних пристроїв.

Захищають мобільні пристрої від шкідливих програм, які можуть бути завантажені з магазинів додатків або через SMS-повідомлення.

Використовують подібні методи до антивірусних програм для комп'ютерів. Приклади програм мобільного АПЗ. ESET Mobile Security, Kaspersky Mobile Antivirus, Bitdefender Mobile Security[1].

1.4 Критерії оцінки ефективності програмного забезпечення для виявлення шкідливого коду

При виборі програмного забезпечення для виявлення шкідливого коду важливо враховувати його ефективність. Ось деякі ключові критерії, які слід використовувати для оцінки ефективності[7]:

Рівень виявлення.

Це найважливіший критерій, який визначає, наскільки добре програмне забезпечення виявляє відомі та невідомі загрози. Існують різні тести та бенчмарки, які оцінюють рівень виявлення різних програмних рішень. Важливо вибрати програмне забезпечення з високим рівнем виявлення, щоб воно могло захистити вас від широкого спектра загроз[7].

Швидкість сканування.

Це важливо, адже ви не хочете, щоб ваше ПЗ для виявлення шкідливого коду уповільнювало вашу роботу. Швидкість сканування може варіюватися залежно від типу програмного забезпечення та його конфігурації. Важливо вибрати програмне забезпечення, яке сканує швидко, але при цьому не шкодить продуктивності вашого комп'ютера[7].

Вплив на систему.

ПЗ для виявлення шкідливого коду може використовувати багато ресурсів системи, таких як процесор і пам'ять. Це може призвести до уповільнення роботи вашого комп'ютера, особливо на старих або малопотужних машинах. Важливо вибрати ПЗ, яке має мінімальний вплив на вашу систему.[7]

Легкість використання.

Програмне забезпечення для виявлення шкідливого коду має бути простим у використанні, навіть для недосвідчених користувачів. Інтерфейс користувача має бути інтуїтивно зрозумілим, а інструкції чіткими та лаконічними.

Важливо вибрати програмне забезпечення, яке легко встановити, налаштувати та використовувати[7].

Додаткові функції.

Багато програмних рішень для виявлення шкідливого коду пропонують додаткові функції, такі як:

- Захист від фішингу.
- Батьківський контроль.
- Захист хмарного сховища.
- VPN[7].

Вартість.

Програмне забезпечення для виявлення шкідливого коду може варіюватися в ціні від безкоштовного до дорогого. Важливо вибрати програмне забезпечення, яке відповідає вашому бюджету та пропонує необхідні вам функції[7].

Репутація виробника.

Перед вибором програмного забезпечення для виявлення шкідливого коду важливо провести дослідження та вибрати продукт від надійного та шанованого виробника. Прочитайте відгуки користувачів та експертів, щоб оцінити репутацію виробника та якість його продукції[7].

1.5 Висновок до першого розділу

У цьому розділі ми розглянули різні методи виявлення шкідливого коду, а також програмні рішення, які їх використовують. Ми також обговорили критерії оцінки ефективності програмного забезпечення для виявлення шкідливого коду.

На основі вищесказаного можна зробити такі висновки.

Не існує єдиного універсального методу виявлення шкідливого коду. Кожен метод має свої переваги та недоліки. Найкращий метод виявлення шкідливого коду залежить від потреб та ресурсів користувача або організації. Існує багато програмних рішень для виявлення шкідливого коду, які можна використовувати для захисту комп'ютерів та мереж від загроз. При виборі програмного забезпечення для виявлення шкідливого коду важливо враховувати його ефективність, легкість використання, додаткові функції, вартість та репутацію виробника. Важливо використовувати комплексний підхід до кібербезпеки, який включає в себе використання надійного програмного забезпечення для виявлення шкідливого коду, регулярне оновлення програмного забезпечення та операційної системи, використання надійних паролів, уникання відкриття невідомих вкладення або завантаження файлів з ненадійних джерел, а також застосування обережності при використанні електронної пошти та соціальних мереж.

2 ДОСЛІДЖЕННЯ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ВИЯВЛЕННЯ ШКІДЛИВОГО КОДУ

2.1 Вибір програмних рішень для тестування

Вибір програмного забезпечення для тестування (ПЗТ) може здатися складним завданням, адже існує безліч варіантів з різними функціями, можливостями та цінами. Щоб допомогти вам у цій задачі, я опишу ключові фактори, які слід враховувати при виборі ПЗТ, а також наведу кілька прикладів популярних рішень[8].

Визначення потреб.

Перш ніж розпочинати пошук ПЗТ, важливо чітко визначити свої потреби. Це допоможе вам звузити коло варіантів та вибрати рішення, яке найкраще відповідає вашим вимогам. Ось деякі питання, які слід собі поставити:

- Які типи тестування ви будете проводити? (наприклад, функціональне тестування, навантажувальне тестування, тестування безпеки).
- Які платформи та технології ви будете тестувати? (веб-додатки, мобільні додатки, десктопні програми).
- Скільки користувачів буде використовувати ПЗТ?
- Який у вас бюджет?
- Які функції для вас важливі? (наприклад, автоматизація тестів, генерація звітів, інтеграція з іншими інструментами)[8].

Дослідження доступних варіантів.

Після того, як ви визначили свої потреби, ви можете розпочати дослідження доступних ПЗТ. Існує багато ресурсів, які можуть вам у цьому допомогти, як от:

- Веб-сайти з відгуками про програмне забезпечення. Ці сайти надають інформацію про різні ПЗТ, включаючи їхні функції, переваги та недоліки.
- Статті та блоги про тестування. Ці ресурси можуть містити огляди ПЗТ, порівняння різних рішень та поради щодо вибору найкращого ПЗТ для ваших потреб.

– Веб-сайти розробників ПЗТ. На цих веб-сайтах ви можете знайти інформацію про продукти, демо-версії та безкоштовні пробні версії[8].

Оцінка ПЗТ.

Після того, як ви звузили коло варіантів, ви можете розпочати оцінку ПЗТ, які вас зацікавили. Ось деякі фактори, які слід враховувати:

– Функції. Чи має ПЗТ всі функції, які вам потрібні?
– Простота використання. Чи ПЗТ просте у використанні та чи має воно інтуїтивно зрозумілий інтерфейс?
– Звітність. Чи генерує ПЗТ звіти, які вам потрібні?
– Підтримка. Чи пропонує виробник ПЗТ хорошу підтримку користувачів?
– Ціна. Чи відповідає ціна ПЗТ вашому бюджету[8]?

Тестування ПЗТ.

Багато виробників ПЗТ пропонують безкоштовні пробні версії або демо-версії, які дозволяють вам протестувати їхні продукти перед покупкою. Це чудовий спосіб побачити, чи підходить вам ПЗТ, і чи воно відповідає вашим потребам[8].

Приклади популярних ПЗТ:

– Selenium. Це безкоштовний ПЗТ з відкритим кодом для автоматизованого веб-тестування.
– Appium. Це ПЗТ з відкритим кодом для автоматизованого тестування мобільних додатків.
– UFT (Unified Functional Testing**).** Це комерційне ПЗТ від Micro Focus для функціонального тестування.
– LoadRunner**.** Це комерційне ПЗТ від Micro Focus для навантажувального тестування.
– K6. Це безкоштовний ПЗТ з відкритим кодом для навантажувального тестування[8].

2.2 Створення тестового набору

Тестовий набір - це набір тест-кейсів, призначених для перевірки конкретних функцій або компонентів програмного забезпечення. Створення тестового набору є важливою частиною процесу тестування програмного забезпечення, адже воно допомагає гарантувати, що програмне забезпечення відповідає очікуванням та працює правильно[8].

Визначення обсягу тестування.

Перш ніж розпочинати створення тестового набору, важливо визначити обсяг тестування. Це включає визначення того, які функції, компоненти та інтерфейси користувача будуть тестуватися. Обсяг тестування може бути визначений на основі таких факторів, як:

- Ризики проекту. Функції та компоненти, які представляють високий ризик, повинні бути ретельніше протестовані.
- Важливість функцій. Функції, які є важливими для користувачів, повинні бути ретельніше протестовані.
- Час та бюджет. Обсяг тестування може бути обмежений доступним часом та бюджетом[8].

Розробка тест-кейсів.

Після визначення обсягу тестування можна розпочинати розробку тест-кейсів. Тест-кейс - це опис конкретного тесту, який буде виконано. Кожен тест-кейс повинен містити наступну інформацію:

- Опис тесту. Короткий опис того, що буде тестуватися.
- Умови тесту. Умови, за яких буде виконано тест.
- Кроки тесту. Покрокові інструкції з виконання тесту.
- Очікувані результати. Очікувані результати виконання тесту.
- Дійсні результати. Результати, отримані під час виконання тесту.
- Статус тесту. Пройшов тест, не пройшов або заблокований[8].

Організація тестового набору.

Тестовий набір повинен бути організований таким чином, щоб його було легко використовувати та підтримувати. Існує декілька способів організувати тестовий набір, наприклад:

- За функціональністю. Тест-кейси можна організувати за функціональністю програмного забезпечення.
- За компонентами. Тест-кейси можна організувати за компонентами програмного забезпечення.
- За пріоритетом. Тест-кейси можна організувати за пріоритетом, щоб найважливіші тести виконувалися першими.
- За датою створення. Тест-кейси можна організувати за датою створення[8].

Виконання тестування.

Після того, як тестовий набір буде створено та організовано, можна розпочинати його виконання. Тестування може виконуватися вручну або за допомогою автоматизованих інструментів.

Звітність про результати тестування.

Після завершення тестування важливо задокументувати результати. Звіт про результати тестування повинен містити наступну інформацію:

- Кількість виконаних тестів. Кількість тест-кейсів, які було виконано.
- Кількість пройдених тестів**.** Кількість тест-кейсів, які пройшли успішно.
- Кількість не пройдених тестів**.** Кількість тест-кейсів, які не пройшли успішно.
- Блоковані тести. Кількість тест-кейсів, які було заблоковано.
- Дефекти. Опис дефектів, які було виявлено під час тестування[8].

2.3 Проведення тестування

Проведення тестування - це процес виконання тест-кейсів, які були розроблені під час створення тестового набору. Тестування може виконуватися вручну або за допомогою автоматизованих інструментів[8].

Ручне тестування.

Ручне тестування - це процес виконання тест-кейсів людьми. Це може бути трудомістким та потребувати багато часу, але воно може бути корисним для виявлення дефектів, які можуть бути пропущені автоматизованими інструментами.

Автоматизоване тестування.

Автоматизоване тестування - це процес виконання тест-кейсів за допомогою програмного забезпечення. Це може бути швидшим та економнішим, ніж ручне тестування, і воно може бути корисним для повторного тестування програмного забезпечення після внесення змін.

Вибір методу тестування.

Метод тестування, який використовується, залежить від декількох факторів, таких як:

- Складність програмного забезпечення. Складне програмне забезпечення може потребувати як ручного, так і автоматизованого тестування.
- Доступний час та бюджет. Ручне тестування може бути більш трудомістким та потребувати більше часу, ніж автоматизоване тестування.
- Навички тестувальників. Якщо у тестувальників немає досвіду роботи з автоматизованими інструментами, може бути краще використовувати ручне тестування[8].

Виконання тестів.

Незалежно від того, який метод тестування використовується, важливо, щоб тести виконувалися ретельно та послідовно. Тестувальники повинні дотримуватися інструкцій, задокументованих у тестових кейсах, і фіксувати всі результати тестування.

Звітність про результати тестування.

Після завершення тестування важливо задокументувати результати. Звіт про результати тестування повинен містити інформацію, описану в розділі "Створення тестового набору".

Управління процесом тестування.

Важливо ефективно управляти процесом тестування. Це включає планування тестування, відстеження прогресу, управління ризиками та забезпечення дотримання бюджету.

2.4 Аналіз результатів тестування

Рівень виявлення - це ключовий показник ефективності програмного забезпечення для виявлення шкідливого коду. Він визначає, наскільки успішно програмне рішення може ідентифікувати та позначити шкідливі файли або програми[8].

Як виміряти рівень виявлення:

- Використання тестового набору. Для вимірювання рівня виявлення використовується тестовий набір, який містить як шкідливі, так і чисті файли.
- Запуск програмного забезпечення. Програмне забезпечення для виявлення шкідливого коду запускається на тестовому наборі.
- Аналіз результатів. Програмне забезпечення позначає кожен файл у тестовому наборі як шкідливий або чистий.
- Розрахунок рівня виявлення. Рівень виявлення розраховується як відсоток шкідливих файлів, які були правильно позначені програмним забезпеченням[8].

Фактори, що впливають на рівень виявлення:

- Тип шкідливого програмного забезпечення. Різні типи шкідливого програмного забезпечення можуть мати різні рівні виявлення. Програмне забезпечення може бути більш ефективним проти одних типів загроз, ніж проти інших.
- Якість тестового набору. Тестовий набір повинен бути репрезентативним для реальних загроз, з якими може зіткнутися користувач. Використання

застарілих або нерепрезентативних зразків може призвести до неточних результатів.

- Методи виявлення шкідливого коду. Різні програмні рішення для виявлення шкідливого коду використовують різні методи, такі як сигнатурний аналіз, евристичний аналіз та аналіз поведінки. Ефективність цих методів може відрізнятися залежно від типу шкідливого програмного забезпечення.
- Оновлення програмного забезпечення. Важливо регулярно оновлювати програмне забезпечення для виявлення шкідливого коду, щоб воно могло виявляти нові загрози[8].

Інтерпретація рівня виявлення.

Високий рівень виявлення є важливим, але це не єдиний фактор, який слід враховувати при виборі програмного забезпечення для виявлення шкідливого коду. Також важливо враховувати такі фактори, як кількість помилкових спрацьовувань, вплив на продуктивність системи та простоту використання[8].

2.5 Вплив досліджень та тестувань на кібербезпеку

Результати дослідження про програмне забезпечення для виявлення шкідливого коду можуть мати значний вплив на практики кібербезпеки в різних організаціях та сферах.

Ось деякі з ключових наслідків, які слід розглянути[7]:

- Дослідження може допомогти вдосконалити методи виявлення та реагування на кіберзагрози, надаючи чітке розуміння можливостей та обмежень різних програмних рішень для виявлення шкідливого коду.
- Організації можуть використовувати цю інформацію для вибору найбільш ефективних інструментів для своїх потреб та для кращого налаштування та використання цих інструментів.

– Дослідження також може допомогти в розробці нових методів виявлення та аналізу шкідливого коду, що може призвести до більш швидкого виявлення та нейтралізації нових загроз[7].

Підвищення обізнаності та навчання:

– Результати дослідження можуть підвищити обізнаність про важливість програмного забезпечення для виявлення шкідливого коду та інших заходів кібербезпеки.
– Це може призвести до кращого навчання та підготовки співробітників щодо кіберзагроз та методів їх запобігання.
– Підвищення обізнаності та навчання також може допомогти зменшити кількість людських помилок, які часто є причиною кібератак[7].

Покращення співпраці та обміну інформацією:

– Дослідження може сприяти кращому співробітництву та обміну інформацією між організаціями, дослідницькими установами та органами державної влади щодо кіберзагроз.
– Це може призвести до більш швидкого обміну даними про нові загрози та методи їх виявлення, а також до більш скоординованих зусиль з протидії кіберзлочинності[7].

Розробка стандартів та рекомендацій.

Результати дослідження можуть бути використані для розробки стандартів та рекомендацій щодо вибору, використання та обслуговування програмного забезпечення для виявлення шкідливого коду. Ці стандарти та рекомендації можуть допомогти організаціям приймати більш обґрунтовані рішення щодо кібербезпеки та забезпечити кращий захист своїх систем та даних[7].

Зростання інвестицій у кібербезпеку**.**

Дослідження може призвести до збільшення інвестицій у кібербезпеку, оскільки організації визнають важливість захисту своїх систем та даних від кіберзагроз. Це може призвести до розробки нових технологій та рішень для кібербезпеки, а також до розширення можливостей для досліджень та освіти в цій галузі[7].

2.6 Висновок до другого розділу

У цьому розділі ми дослідили різні програмні рішення для виявлення шкідливого коду, порівнюючи їх за кількома ключовими критеріями, такими як рівень виявлення, кількість помилкових спрацьовувань, вплив на продуктивність системи, простота використання та вартість. Результати дослідження показали, що не існує єдиного універсального програмного рішення для виявлення шкідливого коду, яке б відповідало потребам усіх організацій. Найкраще програмне забезпечення для конкретної організації залежатиме від її індивідуальних вимог, бюджету та ризиків.

Важливо зазначити, що програмне забезпечення для виявлення шкідливого коду є лише одним із компонентів комплексної стратегії кібербезпеки. Організації також повинні використовувати інші заходи безпеки, такі як брандмауери, системи запобігання вторгненням та шифрування, а також проводити регулярні навчальні заняття з персоналом щодо кібербезпеки.

3 ПРАКТИЧНЕ ЗАСТОСУВАННЯ МЕТОДІВ

3.1 Сигнатурний аналіз

У цьому розділі буде продемонстровано практичне застосування декількох методів виявлення шкідливого коду, а саме сигнатурного аналізу, евристичного аналізу та аналізу поведінки.

Сигнатурний аналіз - це метод виявлення шкідливого коду, який порівнює файли або фрагменти коду з відомими зразками шкідливого програмного забезпечення. Ці зразки, які називаються "сигнатурами", зберігаються в базі даних сигнатур і постійно оновлюються, щоб включати нові загрози[5].

Сигнатурний аналіз є одним з найпоширеніших методів виявлення шкідливого коду, оскільки він простий у реалізації та ефективний проти відомих загроз. Однак він не може виявити нові або невідомі загрози, для яких ще не створено сигнатур.

Для демонстрації сигнатурного аналізу буде використано онлайн сервіс під назвою "VirusTotal"(див. рисунок 3.1). VirusTotal - це безкоштовний онлайн-сервіс, який дозволяє користувачам сканувати файли та URL-адреси на наявність шкідливого коду. VirusTotal використовує сигнатурний аналіз та інші методи для виявлення шкідливого програмного забезпечення[9].

Рисунок 3.1 – Головна сторінка сервісу

Результати VirusTotal включатимуть список антивірусних програмних продуктів(див. рисунок 3.2), які сканували файл, а також будь-які відповідності сигнатурам шкідливого програмного забезпечення. Я вирішив просканувати .rar файл на наявність вірусів. Сервіс показав що файл безпечний.

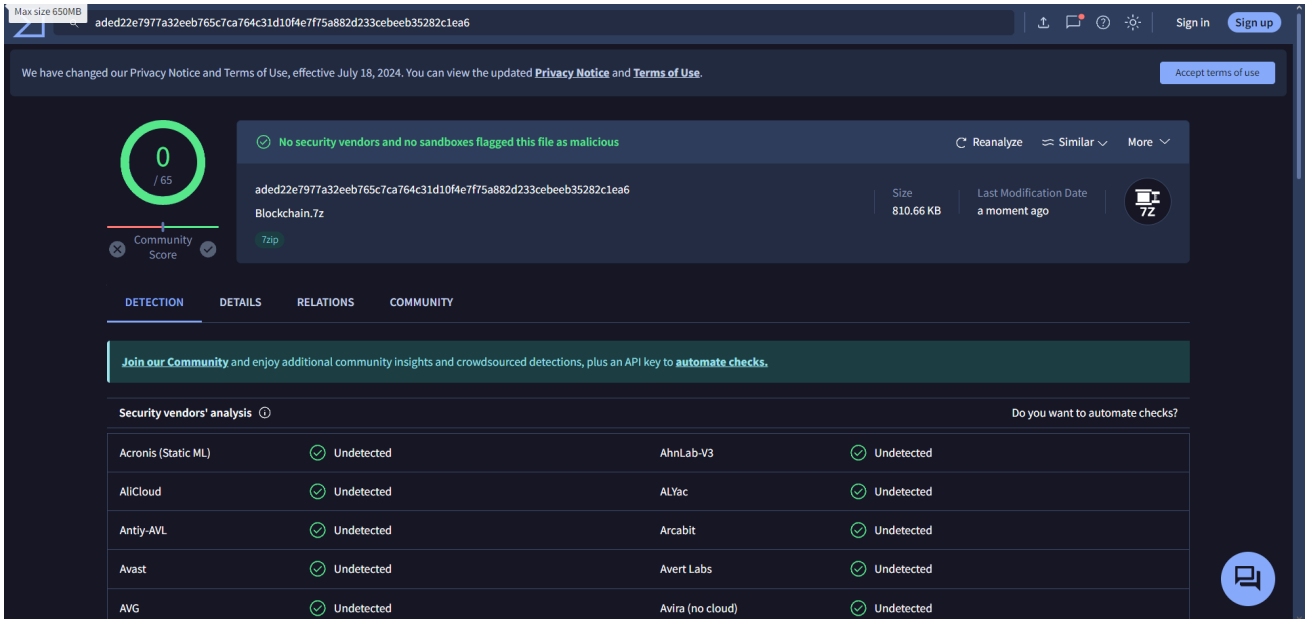

Рисунок 3.2 – Перелік антивірусних програм які сканували файл

Аналіз результатів.

Якщо VirusTotal не виявить жодних відповідностей сигнатурам шкідливого програмного забезпечення, то файл, ймовірно, безпечний. Однак важливо зазначити, що сигнатурний аналіз не може гарантувати, що файл є безпечним.

Якщо VirusTotal виявить відповідність сигнатурі шкідливого програмного забезпечення, то файл, ймовірно, є шкідливим. У цьому випадку рекомендується видалити файл або помістити його в карантин.

За допомогою сервісу VirusTotal можна також перевірити URL на наявність фішингу(див. рисунок 3.3).

Рисунок 3.3 – Головна сторінка вкладки для перевірки URL

Я спробував перевірити сайт дистанційного навчання ТНТУ(див. рисунок 3.4). Сервіс показав що сайт безпечний, а тепер спробую проаналізувати сайт в якому можуть бути ознаки фішингу або ж інші шахрайські дії.

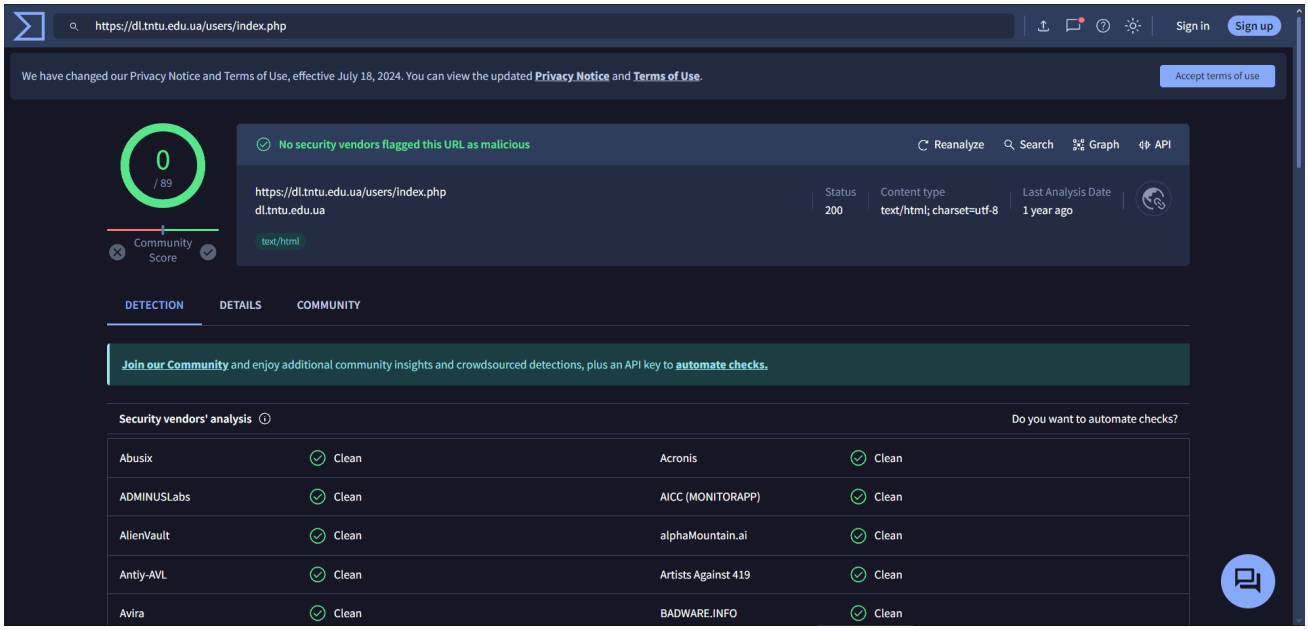

Рисунок 3.4 – Список антивірусів які перевіряли сайт

На електрону пошту прийшло спам-сповіщення, в якому є посилання на підозрілий сайт під виглядом онлайн-казино(див. рисунок 3.6).

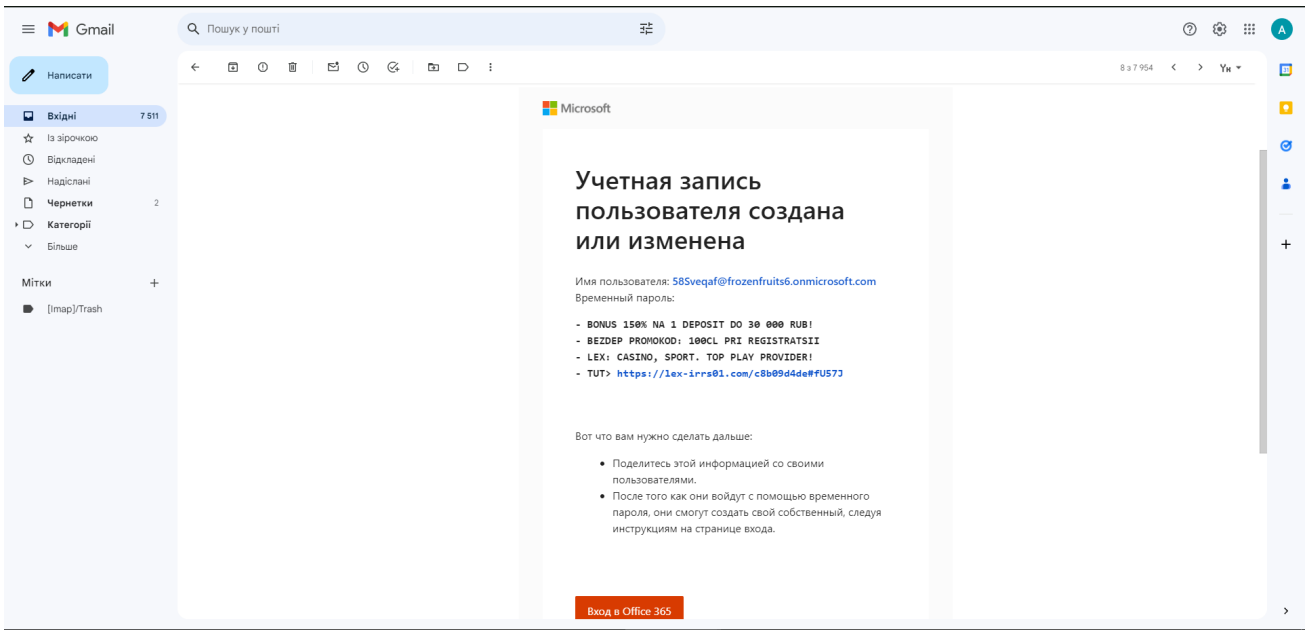

Рисунок 3.6 – Спам-повідомлення

Я скопіював посилання і перевірив його через сервіс VirusTotal результат був цікавіший а ніж попередній (див. рисунок 3.7).

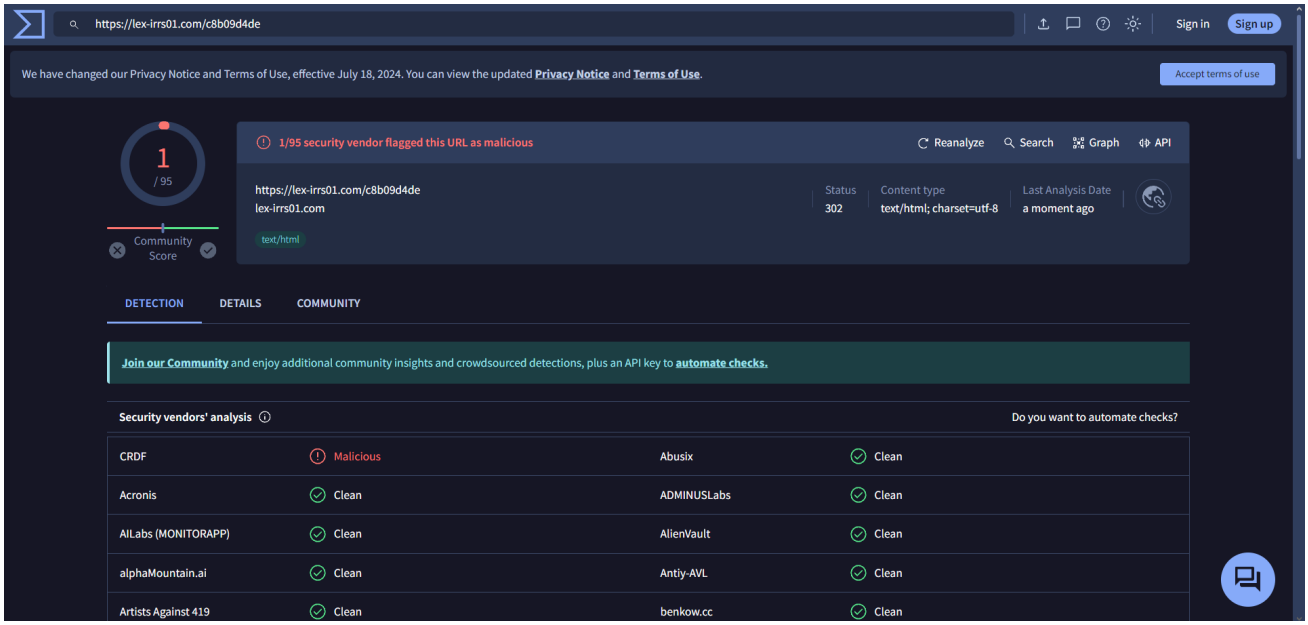

Рисунок 3.7 – Перелік антивірусів які перевіряли сайт

З 95 антивірусів лише 1 показав що сайт може бути зловмисним.

Недоліки методу:

– Не може виявити нові або невідомі загрози.
– Може генерувати помилкові спрацьовування.
– База даних сигнатур повинна бути актуальною[4].

Переваги методу:

– Сигнатурний аналіз також має ряд переваг.
– Простий у реалізації.
– Ефективний проти відомих загроз.
– Швидкий[4].

3.2 Евристичний аналіз

Евристичний аналіз використовує правила та евристики для виявлення шкідливого коду, ґрунтуючись на його поведінці та характеристиках. Цей метод може виявляти нові або невідомі загрози, які не можуть бути виявлені сигнатурним аналізом.

Однак евристичний аналіз може генерувати більше помилкових спрацьовувань, ніж сигнатурний аналіз.

Цей метод аналізу я розгляну на прикладі програми F-Secure(див. рисунок 3.8). F-Secure - це комерційний антивірусний програмний пакет, який використовує евристичний аналіз, сигнатурний аналіз та інші методи для виявлення та нейтралізації шкідливого програмного забезпечення[2].

F-Secure пропонує безкоштовну 30-денну пробну версію свого антивірусного програмного забезпечення(див. рисунок 3.9), яку можна завантажити з веб-сайту компанії, https://www.f-secure.com/en.

Після встановлення F-Secure (див. рисунок 3.10), ви можете виконати повне сканування системи, щоб виявити потенційні загрози. Сканування використовуватиме евристичний аналіз, а також інші методи, такі як сигнатурний аналіз та аналіз поведінки, для виявлення шкідливого програмного забезпечення[10].

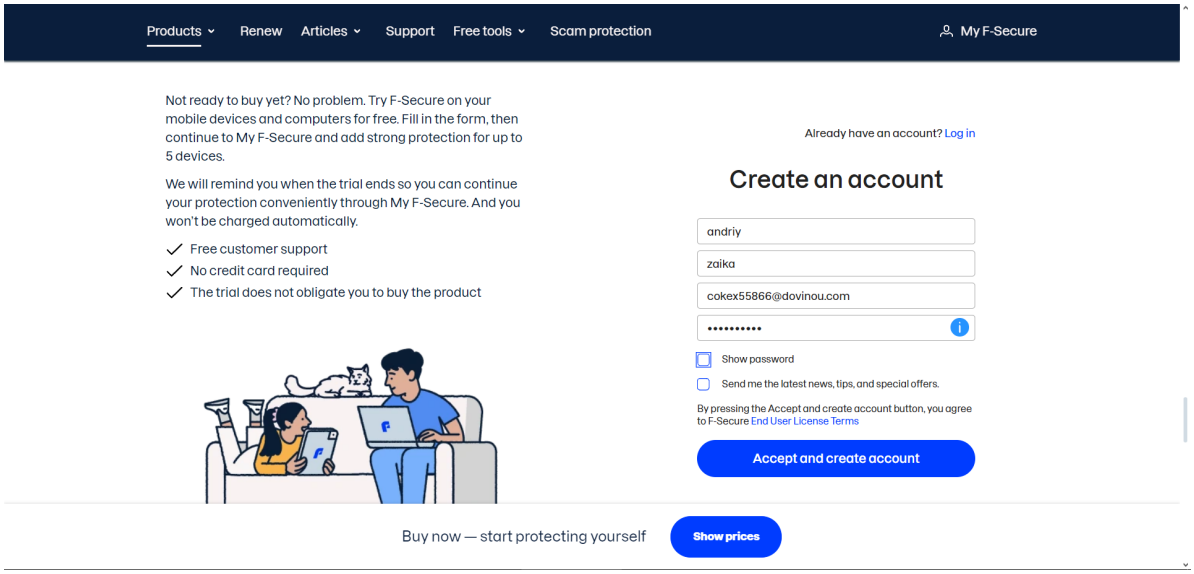

Рисунок 3.8 – Реєстрація на сайті

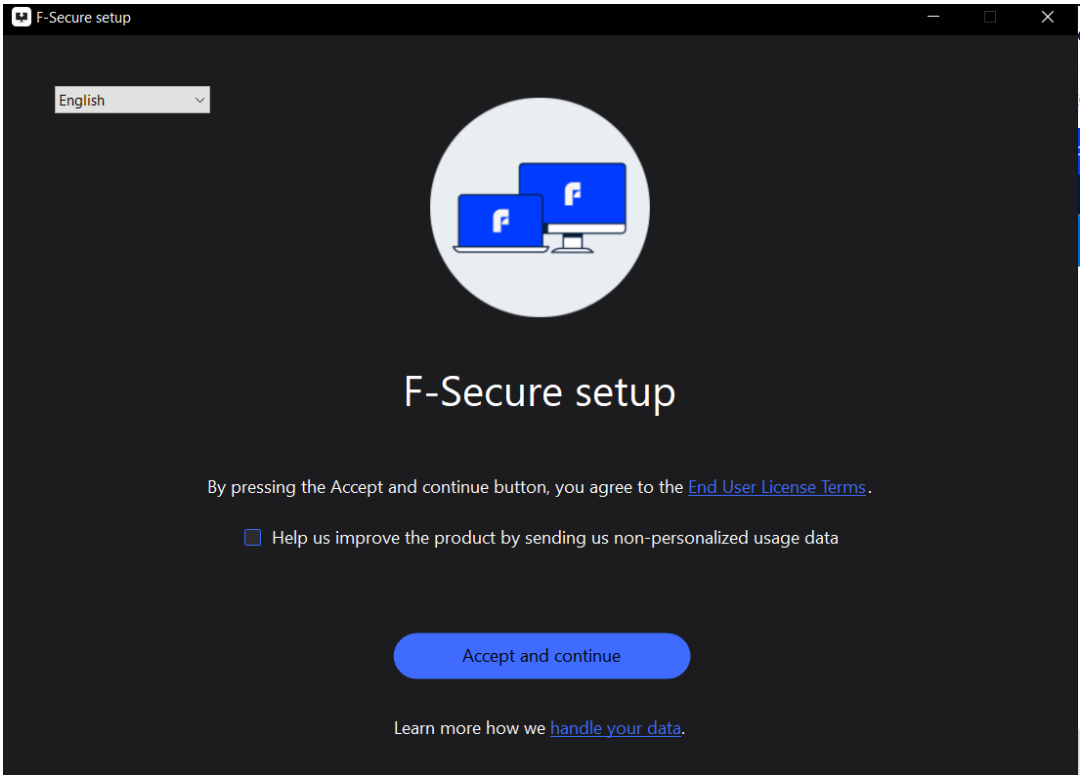

Рисунок 3.9 – Початок встановлення

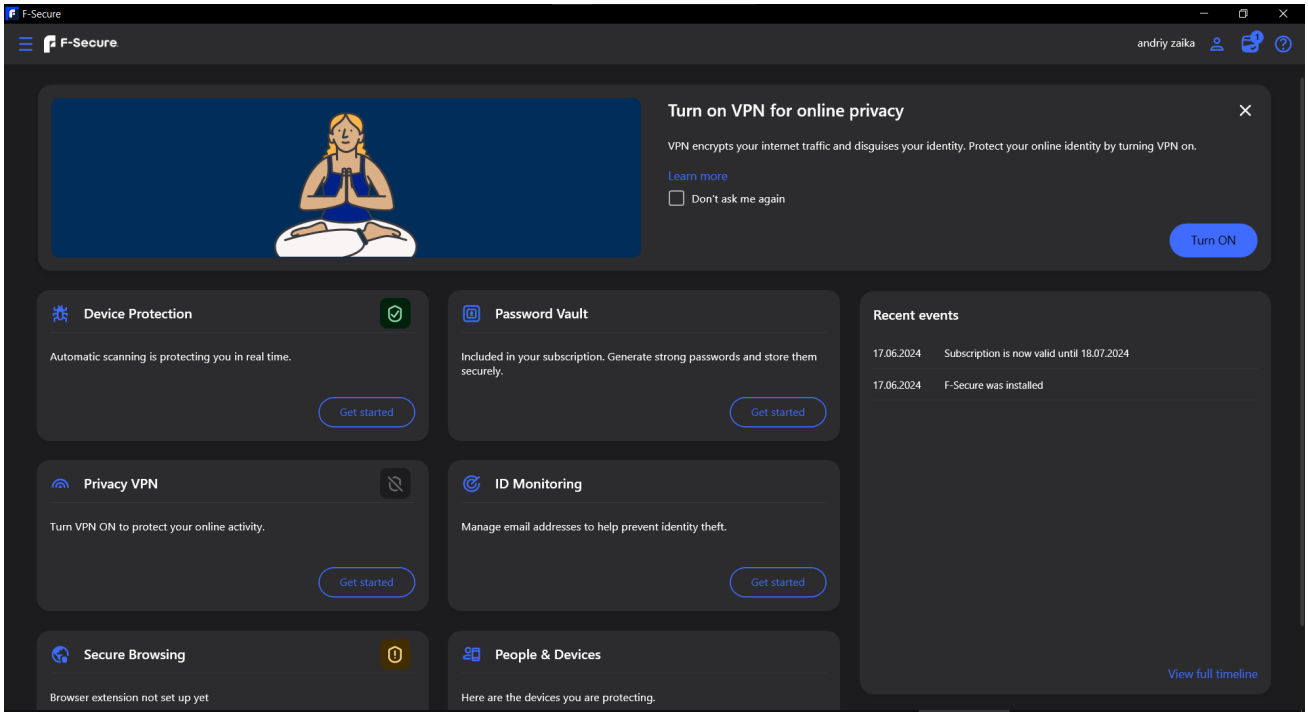

Рисунок 3.10 – Головне меню антивірусу

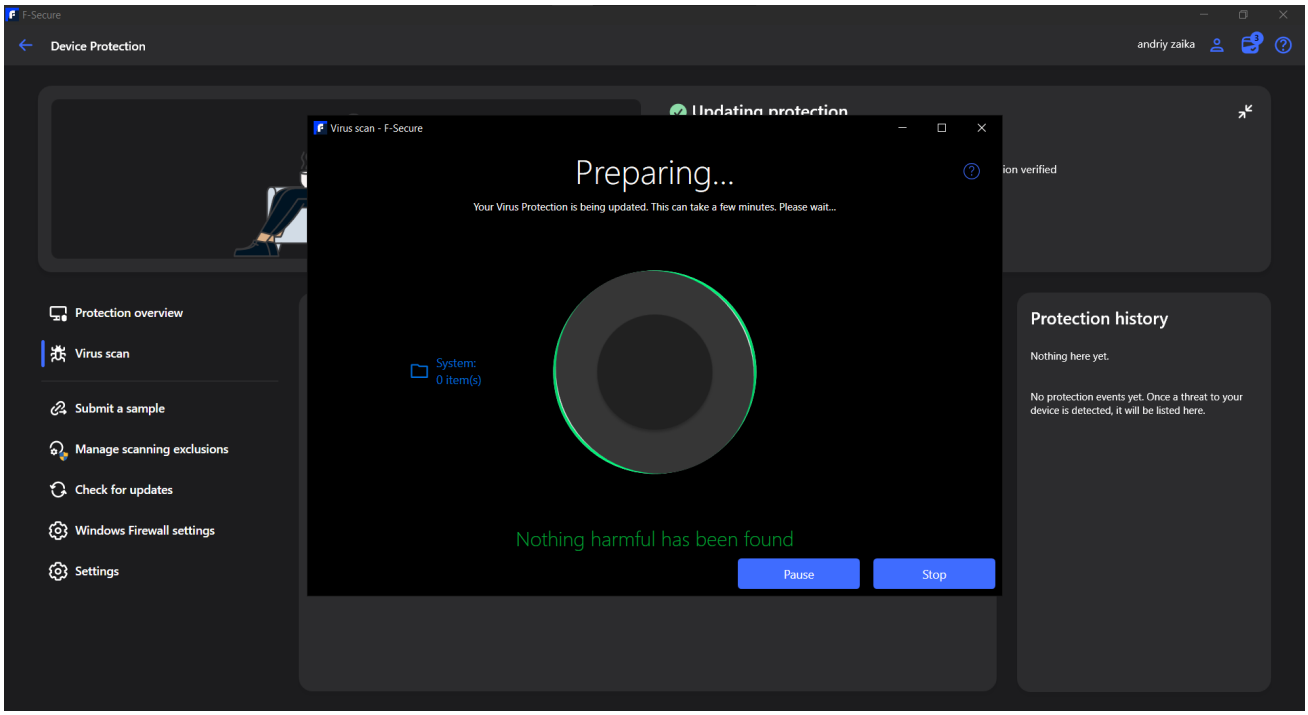

Рисунок 3.11 – Початок сканування системи на наявність вірусів

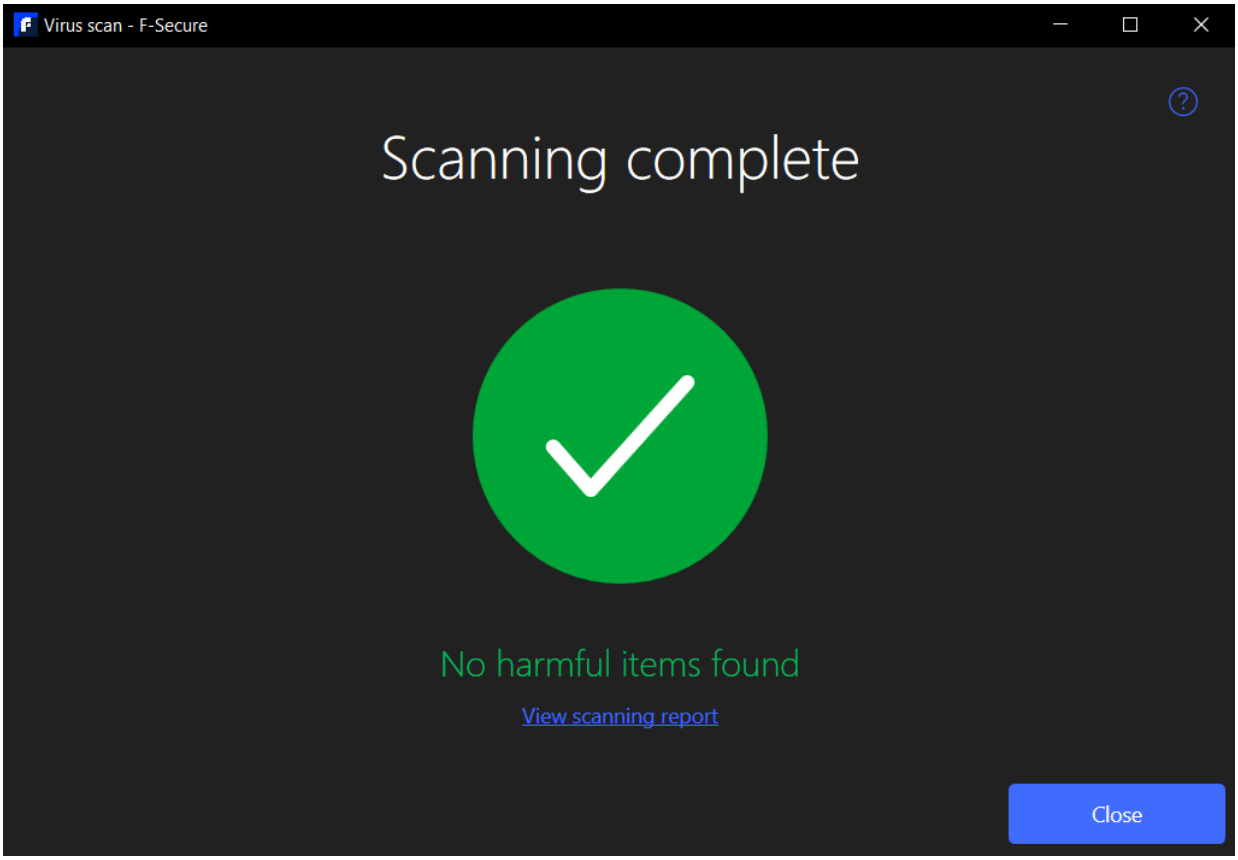

Рисунок 3.12 – Сканування завершено, вірусів не знайдено

Якщо F-Secure виявить загрозу(див. рисунок 3.11), то файл, ймовірно, є шкідливим. F-Secure запропонує вам можливість видалити файл, помістити його в карантин або проігнорувати його.

Важливо зазначити, що евристичний аналіз може генерувати помилкові спрацьовування.

Якщо F-Secure виявить загрозу(див. рисунок 3.12), але ви не впевнені, чи є файл шкідливим, ви можете завантажити його на веб-сайт VirusTotal для сканування іншими антивірусними програмними продуктами.

Недоліки методу:

- Може генерувати більше помилкових спрацьовувань, ніж сигнатурний аналіз.
- Може бути менш ефективним проти деяких типів шкідливого програмного забезпечення.
- Може бути повільнішим, ніж сигнатурний аналіз[4].

Переваги методу:

- Може виявляти нові або невідомі загрози.
- Не залежить від баз даних сигнатур.
- Може адаптуватися до нових загроз.

Додаткові можливості F-Secure.

Окрім евристичного аналізу, F-Secure пропонує ряд інших функцій для захисту вашої системи від кіберзагроз, таких як:

- Захист в режимі реального часу**.** F-Secure може сканувати файли на наявність шкідливого програмного забезпечення, перш ніж вони будуть відкриті або запущені на вашому комп'ютері.
- Захист веб-браузера. F-Secure може блокувати шкідливі веб-сайти та фішингові атаки.
- Захист від програм-вимагачів. F-Secure може захистити ваші файли від шифрування програм-вимагачів.
- Батьківський контроль. F-Secure може допомогти вам контролювати, до чого ваші діти мають доступ в Інтернеті.

3.3 Аналіз поведінки

Аналіз поведінки спостерігає за поведінкою програмного забезпечення під час його виконання, щоб виявити підозрілу активність, яка може свідчити про те, що воно є шкідливим. Цей метод може виявляти нові або невідомі загрози, які не можуть бути виявлені сигнатурним аналізом або евристичним аналізом[3].
Однак аналіз поведінки може генерувати більше помилкових спрацьовувань, ніж інші методи, і може бути більш ресурсомістким.

Для демонстрації аналізу поведінки буде використано програмне забезпечення (див. рисунок 3.13) під назвою "Comodo Behavioral Defense". Comodo Behavioral Defense - це безкоштовний програмний пакет для захисту кінцевих точок, який використовує аналіз поведінки, а також інші методи для виявлення та нейтралізації

шкідливого програмного забезпечення[11]. В головному меню програми можна побачити функції які пропонує сервіс (див. рисунок 3.14).

Рисунок 3.13 – Початок встановлення антивірусу

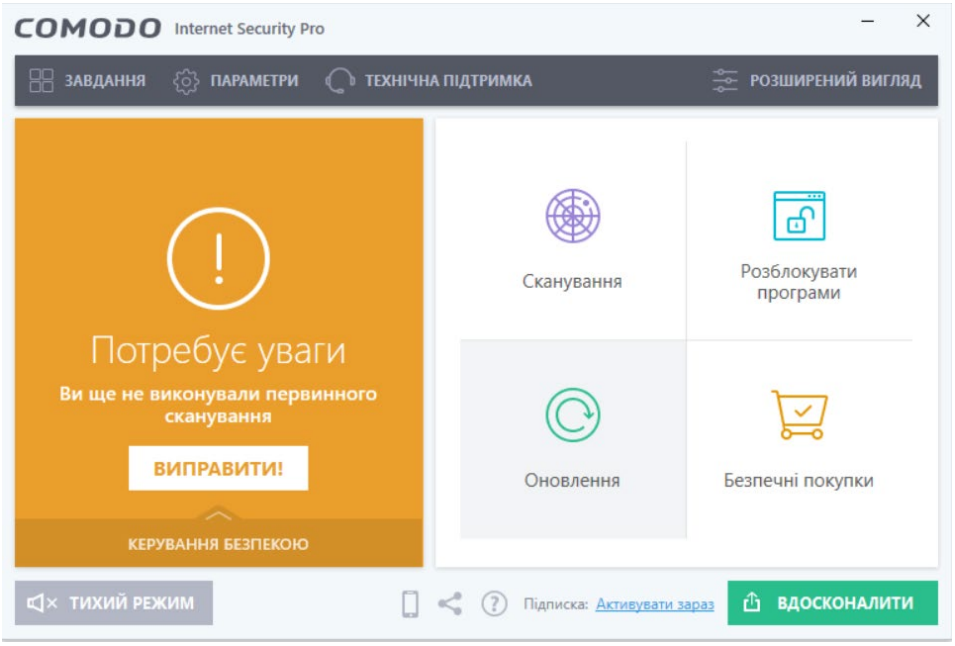

Рисунок 3.14 – Головне меню програми

За допомогою вкладки «Переглянути активність» (див. рисунок 3.15) ми можемо спостерігати в реальному часі активність програм, їх поведінку та

навантаження на систему. Для цього нам потрібно було додатково встановити “Comodo KILL Switch”(див. рисунок 3.16).

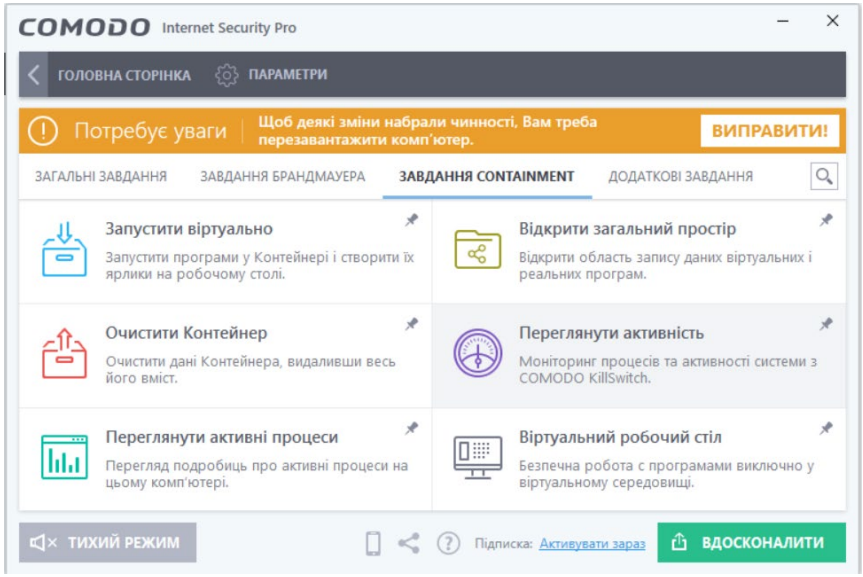

Рисунок 3.15 – Перелік функцій антивірусу

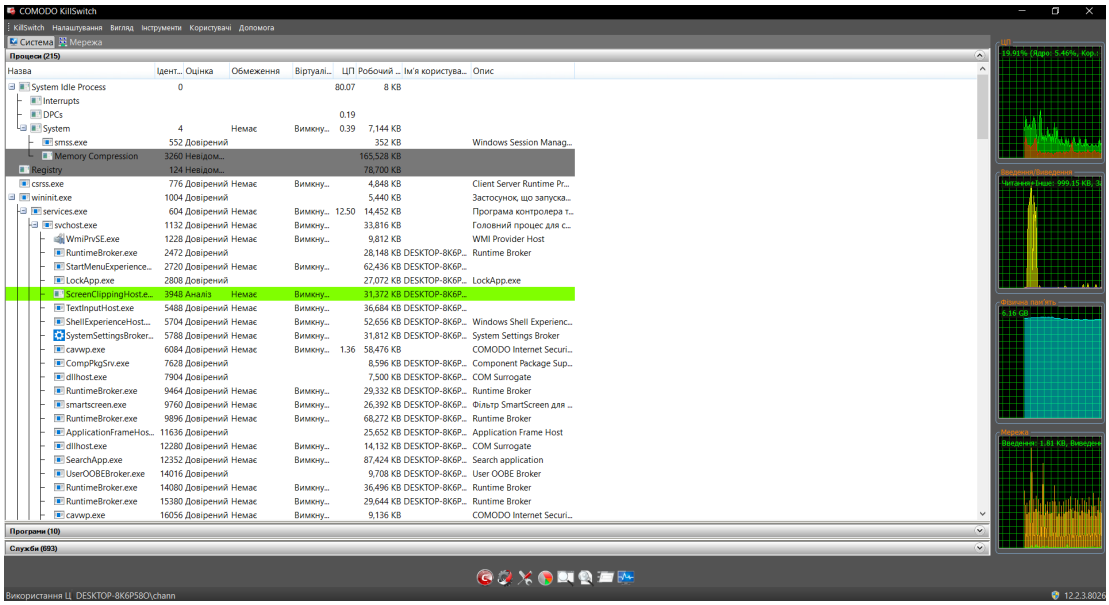

Рисунок 3.16 – Робочий вигляд програми

Аналіз результатів.

Якщо Comodo Behavioral Defense виявить підозрілу активність, він відобразить сповіщення. Ви можете переглянути деталі сповіщення, щоб дізнатися більше про активність, яка була виявлена.

Comodo Behavioral Defense дозволяє вам вирішити, чи слід дозволити або заблокувати підозрілу активність.

3.4 Порівняльний аналіз сервісів

Провести порівняльний аналіз трьох методів використання сервісів "VirusTotal", "F-Secure" і "Comodo Kill Switch" можна за кількома ключовими критеріями, функціональність, зручність використання, ефективність у виявленні загроз та додаткові можливості.

За функціональністю їх порівнювати важко тому що, VirusTotal онлайн сервіс який дозволяє користувачам завантажувати файли або вказувати URL для аналізу з використанням понад 70 антивірусних програм та інструментів аналізу. Він забезпечує швидкий доступ до результатів та можливість перегляду детальної інформації про кожну загрозу.

F-Secure в свою чергу це повноцінне антивірусне ПЗ, яке надає комплексний захист від вірусів, шкідливих програм, фішингу та інших кіберзагроз. Воно використовує хмарні технології для оновлення баз даних і забезпечення актуального захисту.

А Comodo Kill Switch, це інструмент моніторингу та управління процесами, що дозволяє користувачам контролювати активні програми, мережеву активність і швидко зупиняти підозрілі процеси. Він також надає можливості для аналізу безпеки та виявлення аномалій який і використовує в своїх перевірках Аналіз Поведінки.

Тобто, F-Secure за функціональністю буде зрозумілим та зручнішим.

Далі порівняємо їх за зручністю використання. VirusTotal**.** Веб-інтерфейс простий у використанні, не вимагає встановлення додаткового програмного забезпечення. Користувачі можуть легко завантажувати файли або вводити URL для аналізу.

F-Secure. Інтерфейс інтуїтивно зрозумілий, але потребує встановлення програмного забезпечення на пристрій. Включає автоматичні оновлення і просту в налаштуванні панель управління.

Comodo Kill Switch. Цей інструмент має більш складний інтерфейс, орієнтований на досвідчених користувачів. Вимагає деякого часу на вивчення всіх функцій та можливостей.

Опираючись на ці факти можна зробити висновок що, VirusTotal в плані зручності буде кращим варіантом. Тобто використовуючи його одноразово щоб перевірити файл чи URL. Але як для повноцінного антивірусного ПЗ, то F-Secure підходить краще.

І для підведення підсумків який ж з трьох сервісів краще підходить для використання, проведемо порівняння по ефективності виявлення загроз.

VirusTotal. Використання великої кількості антивірусних движків забезпечує високий рівень виявлення загроз. Однак, він не надає захисту в режимі реального часу.

F-Secure**.** Ефективно захищає від більшості типів загроз завдяки регулярним оновленням баз даних та використанню хмарних технологій.

Comodo Kill Switch**.** Здатен виявляти та зупиняти підозрілі процеси в реальному часі, але основний акцент зроблено на моніторинг і управління, а не на повноцінний антивірусний захист.

Для доповнення цього порівняння можна ще додати про додаткові можливості

VirusTotal. Можливість інтеграції з іншими інструментами через API, надання звітів та аналізу для дослідників безпеки.

F-Secure. Включає додаткові функції, такі як VPN, захист від фішингу та функції батьківського контролю.

Comodo Kill Switch. Підтримка детального моніторингу системи, можливість віддаленого управління та інтеграція з іншими інструментами безпеки від Comodo.

Порівнюючи ці сервіси, можна зробити висновок, що кожен з них має свої унікальні переваги і призначений для різних цілей. VirusTotal підходить для швидкого і багатоаспектного аналізу файлів, F-Secure надає комплексний захист для користувачів, а Comodo Kill Switch є потужним інструментом для управління та моніторингу системи в реальному часі.

3.5 Висновок до третього розділу

У цьому розділі були детально розглянуті три ключових методи виявлення шкідливого коду, сигнатурний аналіз, евристичний аналіз та аналіз поведінки. Кожен з них володіє унікальними характеристиками, перевагами та недоліками, роблячи їх придатними для різних сценаріїв кібербезпеки. Та провели порівняльний аналіз.

Сигнатурний аналіз ґрунтується на порівнянні зразків шкідливого коду з відомими сигнатурами, що робить його простим та ефективним для виявлення вже знайомих загроз. Однак його обмеженість полягає в нездатності розпізнавати нові або мутовані варіації шкідливого програмного забезпечення.

Евристичний аналіз, навпаки, здатний виявляти нові загрози, аналізуючи підозрілу поведінку програмного забезпечення. Це робить його цінним інструментом для активного захисту від невідомих зловмисників. Проте, евристичний аналіз може генерувати більше помилкових спрацьовувань, що потребує ретельного налаштування та аналізу результатів.

Аналіз поведінки йде ще далі, відстежуючи поведінку програмного забезпечення в ізольованому середовищі (пісочниці) для виявлення шкідливих дій. Цей метод забезпечує глибокий аналіз та може розпізнавати складні загрози, що обходять інші методи. Проте, аналіз поведінки потребує значних ресурсів та може бути складнішим у налаштуванні та інтерпретації результатів.

Порівняльний аналіз показав, що кожен сервіс унікальний та має свої переваги і недоліки. VirusTotal підходить для швидкої перевірки, F-Secure підходить для повноцінного антивірусного ПЗ для ПК, а Comodo Kill Switch, підходить більш для досвідчених користувачів ПК.

4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Економічне значення заходів щодо покращенню умов та охорони праці.

Покращення умов та охорони праці (УОП) має значний економічний вплив на всі аспекти діяльності підприємства, держави та суспільства в цілому[17].

На рівні підприємства:

- Підвищення продуктивності праці. Здорові, мотивовані та безпечні працівники зможуть працювати продуктивніше, що призведе до кращих результатів та зростання прибутку.
- Зниження витрат. Зменшення нещасних випадків на виробництві, професійних захворювань та плинності кадрів веде до економії коштів, пов'язаних з лікуванням, виплатами компенсацій, навчанням нових співробітників тощо.
- Покращення іміджу. Підприємства, які піклуються про своїх працівників та забезпечують їм безпечні та комфортні умови праці, мають кращий імідж на ринку праці та серед клієнтів.
- Зниження ризиків. Дотримання норм УОП допомагає уникнути штрафів, санкцій та судових позовів, які можуть призвести до значних фінансових втрат[17].

На рівні держави:

- Зростання ВВП. Здорові та працездатні громадяни більше заробляють та роблять більший внесок у розвиток економіки.
- Зниження витрат на охорону здоров'я. Зменшення професійних захворювань та травм на виробництві знижує навантаження на систему охорони здоров'я.
- Соціальна стабільність**.** Безпечні та гідні умови праці сприяють зменшенню соціальної напруги та конфліктів[17].

На рівні суспільства:

- Підвищення якості життя. Здорові та щасливі люди живуть довше, мають кращий рівень життя та більше можливостей для саморозвитку.
- Розвиток людського капіталу. Здорове та освічене населення є основою для розвитку конкурентоспроможної економіки.
- Збереження навколишнього середовища. Впровадження екологічно безпечних технологій та практик на виробництві сприяє збереженню довкілля[17].

Інвестиції в УОП є не лише гуманітарним актом, але й економічно вигідним вкладенням.

Підприємства, які приділяють належну увагу цій сфері, отримують значні конкурентні переваги та сприяють розвитку стійкої та соціально відповідальної економіки. Важливо зазначити, що економічні вигоди від покращення УОП не обов'язково є короткостроковими.

Інвестиції в цю сферу можуть окупитися лише через деякий час, але в довгостроковій перспективі вони призводять до значного покращення економічних показників як на рівні підприємства, так і на рівні держави та суспільства в цілому[18].

Окрім вищезазначених економічних переваг, покращення УОП також має низку інших позитивних наслідків:

- Підвищення морального духу та мотивації працівників.
- Зниження ризиків вигоряння працівників.
- Залучення та утримання кваліфікованих кадрів.
- Створення більш позитивної та продуктивної робочої атмосфери[18].

Впровадження заходів щодо покращення УОП є не лише законом, але й мудрим економічним рішенням, яке призведе до кращого майбутнього для всіх. Важливим аспектом економічного значення УОП є також їх вплив на інвестиційний клімат. Інвестори, як правило, віддають перевагу вкладенню коштів у країни та компанії, які дбають про безпеку та здоров'я своїх працівників. Це пов'язано з тим, що такі країни та компанії, як правило, мають більш стійку

економіку, нижчі ризики для інвестицій та більш кваліфіковану робочу силу. Крім того, покращення УОП може сприяти розвитку інновацій та нових технологій. Здорові та мотивовані працівники більше схильні до творчого мислення та генерування нових ідей[18]. Це може призвести до розробки нових продуктів та послуг, що стимулюватиме економічне зростання.

Загалом, можна зробити висновок, що покращення УОП має значний позитивний вплив на економіку на всіх рівнях[18].

Інвестиції в цю сферу є не лише гуманітарним актом, але й економічно вигідним вкладенням, яке призводить до кращого життя для всіх. Однак, важливо зазначити, що ефективне впровадження заходів щодо покращення УОП потребує системного підходу. Це передбачає співпрацю держави, роботодавців та працівників, а також забезпечення належного фінансування та ресурсів[19].

Важливими інструментами для вдосконалення системи УОП в Україні є:

- Законодавство. Потрібно постійно вдосконалювати законодавство про УОП, приводячи його у відповідність до міжнародних стандартів.
- Контроль. Державні органи повинні забезпечувати ефективний контроль за дотриманням законодавства про УОП на підприємствах.
- Інформування та навчання. Важливо проводити інформаційні кампанії та навчання з питань УОП для роботодавців та працівників.
- Соціальний діалог. Роботодавці та працівники повинні мати можливість вести конструктивний діалог з питань УОП та спільно шукати рішення проблем[19].

Впровадження цих заходів дозволить Україні створити більш безпечну, здорову та продуктивну робочу атмосферу, що сприятиме зростанню економіки та покращенню життя всіх громадян.

4.2 Проведення інструктажів з охорони праці

Інструктажі з охорони праці (ОП) є важливою складовою системи заходів, спрямованих на забезпечення безпечних та здорових умов праці[15].

Їх метою є ознайомлення працівників з вимогами ОП, надання їм знань та навичок, необхідних для безпечного виконання роботи.

Види інструктажів з ОП:

- Вступний інструктаж. Проводиться для всіх новоприйнятих працівників, а також для тих, хто переведений на нову роботу або в новий підрозділ.
- Первинний інструктаж. Проводиться на робочому місці безпосередньо перед початком роботи.
- Повторний інструктаж. Проводиться періодично, але не рідше одного разу на три місяці для робіт з підвищеною небезпекою та одного разу на шість місяців – для інших робіт.
- Цільовий інструктаж. Проводиться перед виконанням робіт з підвищеною небезпекою, а також перед участю у роботах з ліквідації аварій та стихійних лих[15].

Проведення інструктажу з ОП.

Інструктаж проводить:

- керівник робіт.
- особа, уповноважена керівником робіт.
- інструктор з ОП[15].

Інструктаж може проводитися.

- Особисто.
- з використанням засобів навчання (фільми, слайди, комп'ютерні програми)[15].

Під час інструктажу необхідно.

- довести до відома працівників вимоги ОП, що стосуються їх роботи.
- ознайомити з небезпечними та шкідливими факторами, які можуть виникнути під час роботи.
- навчити працівників прийомам безпечного виконання роботи.
- роз'яснити порядок застосування засобів індивідуального та колективного захисту.
- проінструктувати про дії у разі аварії або нещасного випадку[15].

Після закінчення інструктажу працівник.

- повинен розписатися у журналі реєстрації інструктажів з ОП.
- повинен пройти перевірку знань та навичок з питань ОП[15].

Важливість інструктажів з ОП.

Інструктажі з ОП допомагають знизити рівень травматизму та профзахворювань на виробництві. Інструктажі сприяють підвищенню обізнаності працівників з питань ОП. Інструктажі допомагають покращити культуру безпеки на виробництві[16].

Відповідальність за проведення інструктажів з ОП.

За проведення інструктажів з ОП відповідає роботодавець. Роботодавець повинен забезпечити проведення інструктажів з ОП згідно з вимогами законодавства. Особи, які проводять інструктаж з ОП, повинні мати відповідну кваліфікацію[16].

Рекомендації щодо покращення проведення інструктажів з ОП.

Використовувати різні методи навчання (лекції, бесіди, демонстрації, практичні заняття). Залучати до проведення інструктажів фахівців з ОП. Використовувати сучасні засоби навчання (фільми, слайди, комп'ютерні програми). Регулярно проводити перевірку знань та навичок працівників з питань ОП. Створити на підприємстві культуру безпеки, в якій кожен працівник буде зацікавлений у дотриманні вимог ОП[15].

Проведення інструктажів з охорони праці (ОП) є важливим заходом, який допомагає:

- Знизити рівень травматизму та профзахворювань на виробництві.
- Підвищити обізнаність працівників з питань ОП.
- Покращити культуру безпеки на виробництві.
- Зменшити ризики для життя та здоров'я працівників.
- Поліпшити імідж підприємств.
- Знизити витрати на лікування травм та профзахворювань.
- Підвищити продуктивність праці[15].

Нехтування проведенням інструктажів з ОП може призвести до:

- Зростання рівня травматизму та профзахворювань на виробництві.
- Штрафних санкцій до роботодавця.
- Погіршення іміджу підприємства.
- Виникнення нещасних випадків на виробництві.
- Підвищення ризиків для життя та здоров'я працівників[15].

Тому роботодавцю важливо:

- Забезпечити проведення інструктажів з ОП згідно з вимогами законодавства.
- Призначити відповідальних осіб за проведення інструктажів з ОП.
- Створити умови для проведення інструктажів з ОП.
- Надати працівникам необхідну інформацію та навчальні матеріали з питань ОП.
- Провести перевірку знань та навичок працівників з питань ОП.
- Регулярно проводити повторні інструктажі з ОП.
- Впровадити систему заохочення працівників до дотримання вимог ОП.[16]

Впровадження комплексного підходу до проведення інструктажів з ОП дозволить роботодавцю не лише виконати вимоги законодавства, але й створити на підприємстві культуру безпеки, в якій кожен працівник буде зацікавлений у дотриманні вимог ОП.[16]

ВИСНОВКИ

Тема даної дипломної роботи полягала в аналізі ефективності програмного забезпечення для виявлення шкідливого коду.

У першому розділі були розглянуті теоретичні основи аналізу ефективності програмного забезпечення для виявлення шкідливого коду. Була представлена класифікація шкідливого програмного забезпечення, а також методи його виявлення. Крім того, були описані програмні рішення для виявлення шкідливого коду та критерії оцінки їх ефективності.

У другому розділі були розглянуті методи проведення дослідження та тестування програмного забезпечення для виявлення шкідливого коду. Були визначені критерії вибору програмних рішень для тестування, проведено аналіз ринку програмного забезпечення та відібрано програмні рішення для тестування. Також визначили найбільш ефективні програмні рішення для різних типів користувачів та організацій

У третьому розділі були розглянуті три програми які використовували методи аналізу та пошуку шкідливого ПЗ, а саме Сигнатурний аналіз, Аналіз поведінки, та Евристичний аналіз. Також провели порівняльний аналіз щодо вибору та використання програмного забезпечення для виявлення шкідливого коду.

У ході виконання роботи було виконано три поставлених завдання. Провів аналіз ефективності різних програмних рішень, для виявлення шкідливого коду та визначив найбільш ефективні програмні рішення, для різних типів користувачів та організацій, і останнє завдання було провести порівняльний аналіз щодо вибору та використання програмного забезпечення для виявлення шкідливого програмного забезпечення.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Що таке антивіруси – UA5.org.UA5.org – Матеріали з інформаційних технологій. URL: https://ua5.org/virus/52-shho-take-antivrusi.html (дата звернення: 24.06.2024).

2. Евристичний аналіз. StudFiles. URL: https://studfile.net/preview/7190364/page:4/ (дата звернення: 24.06.2024).

3. Аналіз поведінки користувачів та суб`єктів, UEBA. URL: https://iitd.com.ua/analiz-povedinki-koristuvachiv-ta-sub-iektiv-ueba/ (дата звернення: 24.06.2024)

4. Бекер, І., Тимощук, В., Маслянка, Т., & Тимощук, Д. (2023). МЕТОДИКА ЗАХИСТУ ВІД ПОВІЛЬНИХ ТА ШВИДКИХ BRUTE-FORCE АТАК НА IMAP СЕРВЕР. Матеріали конференцій МНЛ, (17 листопада 2023 р., м. Львів), 275-276.

5. Malware Detection: 10 Techniques - CrowdStrike.crowdstrike.com. URL:https://www.crowdstrike.com/cybersecurity-101/malware/malware-detection/(date of access: 24.06.2024).

6. Іваночко, Н., Тимощук, В., Букатка, С., & Тимощук, Д. (2023). РОЗРОБКА ТА ВПРОВАДЖЕННЯ ЗАХОДІВ ЗАХИСТУ ВІД UDP FLOOD АТАК НА DNS СЕРВЕР. Матеріали конференцій МНЛ, (3 листопада 2023 р., м. Вінниця), 177-178.

7. Azeem M., Khan D. Analyzing and comparing the effectiveness of malware detection: A study of machine learning approaches. ScienceDirect. URL: https://www.sciencedirect.com/science/article/pii/S2405844023107821/.

8. Тимощук, В., Долінський, А., & Тимощук, Д. (2024). СИСТЕМА ЗМЕНШЕННЯ ВПЛИВУ DOS-АТАК НА ОСНОВІ MIKROTIK. Матеріали конференцій МЦНД, (17.05. 2024; Ужгород, Україна), 198-200. https://doi.org/10.62731/mcnd-17.05.2024.008

9. VirusTotal Learning resources. VirusTotal. URL: https://www.virustotal.com/getstarted/ (date of access: 24.06.2024).

10. F-Secure User Guides. F-Secure User Guides. URL: https://help.f-secure.com/product.html#home/total-windows/latest/en (date of access: 24.06.2024).

11. Skorenkyy, Y., Kozak, R., Zagorodna, N., Kramar, O., & Baran, I. (2021, March). Use of augmented reality-enabled prototyping of cyber-physical systems for improving cyber-security education. In Journal of Physics: Conference Series (Vol. 1840, No. 1, p. 012026). IOP Publishing.

12. What is Dynamic Analysis? Importance & Purpose | Appknox. #1 Mobile Application Security Testing Solutions | Appknox. URL: https://www.appknox.com/cyber-security-jargons/dynamic-analysis (date of access: 24.06.2024).

13. Kharchenko, A., Halay, I., Zagorodna, N., & Bodnarchuk, I. (2015). Trade-off optimal decision of the problem of software system architecture choice. In Proceedings of the International Conference on Computer Sciences and Information Technologies, CSIT 2015 (pp. 198-205)

14. Демчук, В., Тимощук, В., & Тимощук, Д. (2023). ЗАСОБИ МІНІМІЗАЦІЇ ВПЛИВУ SYN FLOOD АТАК. Collection of scientific papers «SCIENTIA», (November 24, 2023; Kraków, Poland), 130-130.

15. Види та порядок проведення інструктажів з охорони праці - Охорона праці і пожежна безпека. Охорона праці і пожежна безпека. URL: https://oppb.com.ua/news/vydy-ta-poryadok-provedennya-instruktazhiv-z-ohorony-praci' (дата звернення: 24.06.2024).

16. Profiteh. Інструктажі з охорони праці в Україні – види й порядок проведення | Профітех. ПРОФІТЕХ. URL: https://profiteh.ua/instruktazhi-z-okhorony-pratsi-v-ukraini/ (дата звернення: 24.06.2024).

17. Економічне значення заходів щодо покращення умов та охоронипраці. StudFiles. URL: https://studfile.net/preview/9364384/ (дата звернення: 24.06.2024).

18. ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ ВПРОВАДЖЕННЯ ЗАХОДІВ З ОХОРОНИ ПРАЦІ | К М Дідур | Агросвіт №5 2020 стр 43-49. Журнал Агросвіт - наукове фахове видання з питань економіки. URL: http://www.agrosvit.info/?op=1&z=3116&i=6 (дата звернення: 24.06.2024).