

Центральноукраїнський національний технічний університет
Центр заочної та дистанційної освіти
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
“___” _____ 2026 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за першим (бакалаврським) рівнем вищої освіти
на тему
**“Програмне забезпечення системи кібербезпеки для протидії
криптоджекінгу та ботнетам”**

Виконав здобувач вищої освіти
IV курсу, групи КБ-22з
ОПП «Кібербезпека»
спеціальності 125 «Кібербезпека»
_____ Гальченко Г.С.
«___» _____ 2026 р.

Керівник проекту
доктор технічних наук, професор
_____ Смірнов О.А.
«___» _____ 2026 р.
Рецензент _____

Центральноукраїнський національний технічний університет
Центр Заочної та дистанційної освіти
Кафедра Кібербезпеки та програмного забезпечення
Освітній ступінь бакалавр
Галузь знань 12 "Інформаційні технології"
Спеціальність 125 "Кібербезпека"
Освітньо-професійна (освітньо-наукова) програма "Кібербезпека"

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.т.н., проф.
_____ Олексій СМІРНОВ
« 27 » лютого 2026 року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ПЕРШИМ (БАКАЛАВРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Гальченку Глібу Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи *Програмне забезпечення системи кібербезпеки для протидії криптоджекінгу та ботнетам*

2. Керівник роботи *Смірнов Олексій Анатолійович, докт. техн. наук, професор*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 135-02 від 27.02.2026 року

3. Строк подання студентом роботи до захисту *23.05.2026 р.*

4. Мета та завдання випускної кваліфікаційної роботи: *Метою роботи є розробка програмного забезпечення системи кібербезпеки для протидії криптоджекінгу та ботнетам*

5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Призначення та область використання.

2. Перегляд аналогічних існуючих систем.

3. Опис і обґрунтування проектних рішень.

4. Етапи програмування системи.

5. Впровадження системи кібербезпеки в промислову експлуатацію.

6. Висновки

6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Структурна схема системи кібербезпеки *1 аркуш*

Функціональна схема системи кібербезпеки *1 аркуш*

Діаграма процесів *1 аркуш*

Блок-схема алгоритму роботи додатку *2 аркуша*

7. Дата видачі завдання « 27 » лютого 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.03.2026 р.	
2.	Постановка задачі, оформлення ТЗ	15.03.2026 р.	
3.	Розробка моделі компонента	20.03.2026 р.	
4.	Розробка структур даних	25.03.2026 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.03.2026 р.	
6.	Програмування алгоритмів	10.04.2026 р.	
7.	Оформлення ПЗ	17.04.2026 р.	
8.	Попередній захист роботи	23.05.2026 р.	

Дата видачі завдання
« 27 » лютого 2026 р.

Підпис керівника

Смірнов О.А.
(прізвище та ініціали)

Завдання прийнято до виконання
« 27 » лютого 2026 р.

Підпис здобувача

Гальченко Г.С.
(прізвище та ініціали)

АНОТАЦІЯ

Гальченко Г.С. Програмне забезпечення системи кібербезпеки для протидії криптоджекінгу та ботнетам. 125 Кібербезпека. Центральноукраїнський національний технічний університет. Кропивницький. 2026.

В даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи кібербезпеки для протидії криптоджекінгу та ботнетам.

Метою розробки є програмне забезпечення системи кібербезпеки для протидії криптоджекінгу та ботнетам.

Результат роботи – програмна реалізація системи кібербезпеки для протидії криптоджекінгу та ботнетам.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на ПЕОМ з ОС Windows 10/11.

Програму розроблено в середовищі Python.

Ключові слова: кібербезпека, криптоджекінг, ботнет

ABSTRACT

Halchenko G.S. Software for a cybersecurity system to combat cryptojacking and botnets. 125 Cybersecurity. Central Ukrainian National Technical University. Kropyvnytskyi. 2026.

In this final qualification work for the first (bachelor's) level of higher education, software has been developed that is intended for a cybersecurity system to combat cryptojacking and botnets.

The purpose of the development is software for a cybersecurity system to combat cryptojacking and botnets.

The result of the work is a software implementation of a cybersecurity system to combat cryptojacking and botnets.

In the process of working on the software model, an analysis of existing hardware and software was performed. All components of the developed software are fully described.

A convenient user interface has been developed. Instructions for working with software are provided.

The program can be used on PCs with Windows 10/11.

The program is developed in the Python environment.

Keywords: cybersecurity, cryptojacking, botnet

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	2
ВСТУП.....	3
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	5
1.1 Призначення системи.....	5
1.2 Область застосування.....	5
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	7
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти.....	7
2.2 Обґрунтування вибору засобів для побудови системи кібербезпеки та мови програмування.....	23
2.3 Розгорнута постановка завдання	26
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	27
3.1 Опис функціонування системи	27
3.2 Розробка структурної схеми.....	33
3.3 Розробка функціональної схеми	37
3.4 Розробка діаграми процесів.....	45
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	48
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	48
4.2 Захист розробленого програмного забезпечення.....	67
5 ВПРОВАДЖЕННЯ СИСТЕМИ КІБЕРБЕЗПЕКИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	69
6 ОСНОВНІ ВИСНОВКИ.....	76
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	78

					ВКРБ-125.26.0020.00.00.ПЗ			
Вим	Арк.	№ докум.	Підп.	Дата	<i>Програмне забезпечення системи кібербезпеки для протидії криптоджекінгу та ботнетам</i>	Літ.	Аркуш	Аркушів
<i>Розроб.</i>		Гальченко Г.С.				Б		
<i>Перев.</i>		Смірнов О.А.					1	86
<i>Н.контр.</i>		Коваленко А.С.				ЦНТУ КБ-22з		
<i>Затв.</i>		Смірнов О.А.						

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ЕОМ	—	електрона обчислювальна машина
КВ	—	коефіцієнт варіації
КЗ	—	канал зв'язку
НСД	—	несанкціонований доступ
ПС	—	програмна середа
СВВ	—	система виявлення вторгнень
СеМО	—	експонентна мережа масового обслуговування
СМО	—	система масового обслуговування
СПД	—	система передачі даних

К6ПЗ-2026

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		2

ВСТУП

Актуальність теми. Зловмисники в прагненні ухилитися від служб безпеки й правоохоронних органів міняють тактику зі страхаючою частотою. Їхні цифрові сліди, досить ефемерні. Гарна стратегія кібербезпеки повинна робити протилежний трюк – робити кіберзагрози видимими й запобігати зникненню критично важливих мережних ресурсів. Знання особливостей останніх погроз допомагає захиститися від них.

Криптоджекінг. При атаках такого типу шкідлива програма використовує комп'ютер жертви для майнінгу криптовалюти. Рівень поширення шкідливих програм криптомайнінгу в порівнянні з попереднім кварталом виріс більш ніж у два рази – з 13 до 28%. Шкідливі програми такого роду активно розвиваються, що утрудняє їхнє виявлення й нейтралізацію.

Шкідливі програми криптомайнінгу відрізняються незвичайною розмаїтістю для настільки молодого виду погроз. Кіберзлочинці створюють невидимі безфайлові шкідливі програми, що вбудовують свій код у браузері. Виявити його досить важко. Майнери націлюються на різні операційні системи й різні криптовалюти, у тому числі на біткойн і монеро. Методи тонкого налаштування шкідливих програм, їхньої доставки й поширення запозичаються в інших атак і враховують минулі успіхи й невдачі, що дозволяє поліпшити статистичні показники.

Коротко говорячи, злочинці йдуть за грошима й активно використовують нові можливості для досягнення своїх цілей. Вони чітко зрозуміли, що системи захвата комп'ютерів для видобутку криптовалют є досить прибутковим підприємством, тому можна чекати продовження інвестицій у цю бізнес-модель і появи тут інноваційних рішень.

Ботнети. Якщо тенденції розвитку експлойтів і шкідливих програм звичайно відбивають ситуацію з безпекою перед зараженням, то ботнети дозволяють глянути на неї вже після зараження. Будучи інфікованою, система

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

найчастіше зв'язується зі шкідливими хостами, а виявлення відповідного трафіку в корпоративному середовищі свідчить про те, що щось йде не так. Набір даних такого роду являє цінність із погляду «роботи над помилками».

У даній роботі виявили, що 58% інфікованих вузлів ботнету зберігають активність протягом усього одного дня й лише в 5% час активності перевищує тиждень. Тривалість зараження, обумовлена як число послідовних днів, протягом яких реєструються сеанси зв'язку, говорить про те, що кібергігієна не повинна обмежуватися тільки установкою відновлень. Необхідно займатися й чищенням систем. В 42% організацій заражений код не вичищається із системи протягом 1-9 днів, а в 6% він залишається в ній більше тижня.

Сьогодні всі знають, що час від часу зараження неминучі. Їм піддані навіть самі укріплені мережі. Але виявлення й швидка ліквідація цих заражень із усуненням погроз (і запобігання повторних заражень) є ознакою успішності програм кібербезпеки.

Мета й завдання дослідження. Метою роботи є програмне забезпечення системи кібербезпеки для протидії криптоджекінгу та ботнетам.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

- Огляд існуючих систем для протидії криптоджекінгу та ботнетам.
- Дослідження системи кібербезпеки для протидії криптоджекінгу та ботнетам.
- Програмна реалізація системи кібербезпеки для протидії криптоджекінгу та ботнетам.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі для протидії криптоджекінгу та ботнетам.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи кібербезпеки для протидії криптоджекінгу та ботнетам, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Ботнет Andromeda, відомий також як Win32/Gamarue, являє собою модульний HTTP-ботнет, що успішно заражає комп'ютери з моменту своєї появи в 2011 році. Andromeda продовжує з'являтися на наших радарх незважаючи на масштабну операцію по його нейтралізації, проведену правоохоронними органами в IV кварталі минулого року. По своєму обсязі й рівню поширення в I кварталі 2026 року він залишався в трійці ведучих ботнетів. На перший погляд, це говорить про те, що операція по ліквідації Andromeda була не занадто успішною. Однак подальший аналіз указує на слабку гігієну в області безпеки.

У даній роботі вивчили організації, які усе ще заражені ботнетом Andromeda, що в «дикій природі» більше не існує, і постаралися визначити, чи страждають вони й від інших погроз. Виявилося, що так воно і є. Компанії, які підверглися в I кварталі зараженню Andromeda, мали у своєму середовищі в три рази більше активних ботнет-вузлів у порівнянні з іншими організаціями. Цілком імовірно, що системи, заражені Andromeda, можуть використовуватися в кількості посередника при поганій гігієні безпеки й/або повільної реакції на виникаючі інциденти.

1.2 Область застосування

Деструктивні й проектовані атаки

Вплив деструктивних програм залишаються досить високим, особливо коли злочинці використовують їх у своїх проектованих атаках. Перед проведенням таких цілеспрямованих атак злочинці проводять ретельну розвідку організації, що сприяє підвищенню ймовірності успіху. Проникнувши в мережу,

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

що атакують обережно переміщаються по її периферії й тільки потім приступають до найбільш руйнівної частини запланованої атаки. Прикладами таких проєктованих атак з деструктивним навантаженням, що забезпечує максимально руйнівний вплив, є шкідлива програма Olympic Destroyer і здирник, що з'явився пізніше, SamSam.

Об'єднання проєктних завдань із деструктивними тенденціями приводить до створення шкідливих програм, поява яких не може не турбувати. Ключовим завданням нині вважаються виявлення й оперативна реакція. Але з обліком того, що на передній план знову виходять хробаки й деструктивні шкідливі програми, прийшов час задуматися про посилення захисту.

Прагнучи підвищити ймовірність досягнення успіху, кібезлочинці продовжують розвивати свої методи здійснення атак, починаючи від криптоджекінгу й закінчуючи ботнетами й шкідливими програмами. Факти, наведені в цьому звіті, свідчать про те, що чим більше зможемо побачити, тим простіше буде захиститися. Вони говорять про те, що не можна заспокоюватися на досягнутому й забувати про основи, зокрема про кібергігієну. І в нинішньому динамічно мінливому середовищі групи ІТ-безпеки мають набагато більше шансів захиститися від новітніх кіберсхем, коли знають, що потрібно шукати.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи кібербезпеки для протидії криптоджекінгу та ботнетам, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти

Дослідження Check Point: схований майнінг домінує серед кіберзлочинів

У новому звіті Check Point відзначається домінування криптоджекінгу серед кібератак і прогнозується ріст поширення ПЗ для схованого майнінгу криптовалют у хмарних інфраструктурах.

Відповідно до звіту Check Point з кібербезпеки, криптовалютна індустрія, як і будь-яка інша галузь, буде залишатися надзвичайно уразливою для кібератак в 2026 році. За даними за минулий рік, 28% організацій по усьому світі піддалися атакам багатоцільових ботнетів, а в 20% випадків повідомлялося про впровадження вірусів-зидників.

Як повідомляють дослідники, у найближчий рік найбільшої небезпеки піддадуться хмарні інфраструктури – вони будуть всі частіше стає жертвами криптоджекінгу або атак, націлених на схований майнінг криптовалют. Check Point відзначає, що хмарні інфраструктури – одна з основних цілей хакерів, що поширюють шкідливе ПЗ для схованого майнінгу.

У цей час більше 90% підприємств використовують хмарні сервіси, і все-таки 67% фахівців з безпеки скаржаться на недостатню прозорість і безпеку хмарної інфраструктури. Масштаби хмарних атак продовжували рости в 2026 році. Неправильне налаштування хмарних ресурсів – як і раніше першорядна причина атак. Крім того, цього року в даній роботі будемо свідками збільшення числа атак, спрямованих безпосередньо на постачальників хмарних послуг.

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

У звіті описується шкідливе ПЗ, що користується популярністю в хакерів. Наприклад, Glupteba використовується для поширення браузерних розширень для крадіжки даних або використання уразливостей маршрутизатора. Cryptoloot застосовується для схованого онлайн-майнінгу Монето, коли зловмисник заходить на веб-сторінку без відома користувача.

Крім того, існує Danabot – банківський троян для Windows, що використовується для крадіжки паролів і даних криптовалютних гаманців. Дослідники також описують, як хакери поширювали вірус-здірник Ryuk для одержання великих викупів в BTC від своїх жертв.

Спектр кіберзагроз стає усе більше велик і витонченим, як говориться у звіті. Дослідники відзначають, що організаціям необхідно реалізовувати превентивний план ефективних захисних мір, щоб устигати за діями хакерів, і постійно обновляти відомості про погрози.

Щоб запобігти атакам нульового дня, організаціям спочатку необхідно одержати інтелектуальний аналіз погроз у режимі реального часу, що надає актуальну інформацію про новітні напрямки атак і методи злому. Аналітика погроз повинна охоплювати всі області атаки, включаючи хмарні, мобільні, мережні, а також Інтернет речей, тому що саме в цих напрямках організації найчастіше стають жертвою хакерів.

Нагадаємо, що на початку минулого року Check Point заявляла, що криптоджекінг в 13-й раз посів перше місце в списку кіберзагроз.

Звіт Group-IB: фішинг і «криптоджекінг» стали найпоширенішими видами атак

Компанія Group-IB, що спеціалізується на запобіганні кібератак, предвідносила новий щорічний звіт «Hi-Tech Crime Trends 2026» у рамках конференції CyberCrimeCon 2026. Відповідно до звіту, близько 56% всіх украдених засобів були викрадені за допомогою фішингових атак.

В 2024-2026 роках основними цілями хакерів стали криптовалютні біржі – за цей період було зламано 14 криптовалютних бірж, загальний збиток яких

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

склав більше \$882 млн. Як мінімум п'ять із цих атак зв'язують із північнокорейськими хакерами із групи Lazarus, чії жертви переважно перебувають у Південній Кореї. На думку експертів компанії, слідом за Lazarus, найімовірніше, атаки на криптовалютні біржі почнуть Silence, MoneyTaker і Cobalt. При цьому основним вектором проникнення в корпоративні мережі криптовалютних бірж залишається цільовий фішинг.

Іншим розповсюдженим видом атак є «криптоджекінг» або схований майнінг, що представляє собою несанкціоноване використання пристроїв користувача для видобутку криптовалют. Цей напрямок шахрайства одержало найбільший розвиток в 2024-2026 роках. Після виходу ПЗ для схованого майнінгу Coinhive з'явилося ще сім програм подібного типу.

Експерти компанії прогнозують, що найбільші майнери можуть стати метою не тільки кіберзлочинців, але й продержавних атак. При певній підготовці це може дозволити атакуючому взяти під контроль 51% потужності для майнінгу й захопити керування криптовалютою. У першій половині 2026 року було зафіксовано відразу п'ять успішних атак 51%, при цьому сума прямого фінансового збитку склала від \$0,55 млн до \$18 млн.

Нова подвійна погроза: об'єднання криптоджекерів із програмами-здириками

Пік масштабних атак програм-здириків довівся на 2016 і початок 2024 року. Однак уже наприкінці 2024 р. для домашніх користувачів і організацій на перший план стала виходити нова погроза – незаконний видобуток криптовалют.

Acronis True Image 2025

Як і програми-здирики, програми для майнінгу (видобутку) криптовалют існують досить давно. Можливість використовувати обчислювальні ресурси комп'ютера для майнінгу біткойнів без потужного або спеціалізованого встаткування відомий як мінімум з 2011 року. Кіберзлочинці почали розробляти шкідливі програми для виконання цієї функції тільки після буму криптовалют, тобто в середині/кінці 2024 р.

У цей період з'явилися тисячі різних електронних валют на базі технології Блокчейн. Багато хто з них стрімко злетіли нагору по обсязі й капіталізації, а деякі можна було добувати за допомогою ресурсів звичайного комп'ютера. Майнінг є одним з основних принципів існування будь-якої криптовалюти. Він надає обчислювальну потужність, необхідну для перевірки попередніх транзакцій, що забезпечує цілісність електронної валюти. Майнери криптовалют використовують ресурси своїх комп'ютерів для рішення складних математичних завдань, при цьому перший хто вирішить завдання одержує винагороду в цій же криптовалюті.

Це може бути вигідним підприємством, якщо ви самі одержуєте плату за ресурси, використані для перевірки транзакцій криптовалюти (обчислювальну потужність і електрику). Але кіберзлочинці зрозуміли, що заразивши комп'ютер користувача шкідливим ПЗ для майнінгу, можна змусити його робити всю роботу, а самим тільки діставати прибуток. Помножьте цей прибуток на тисячу або мільйон заражених комп'ютерів і зрозумієте, чому шахраї поспішили скористатися можливостями схованого майнінгу криптовалют: це приносить гарний дохід, а жертви навіть не підозрюють про те, що їх обкрадають.

Кіберзлочинці придумали хитрість, що потенційно може в кілька разів підвищити відсоток успіху й прибуток – об'єднання різних типів шкідливого ПЗ. Необережний користувач, що відкрив посилання або вкладення в листі, що було відправлено зловмисником, одержує подвійне зараження програмою для схованого майнінгу й програмою-зидником. Зловмисник не активує відразу обидва типи атак, оскільки комп'ютер, файли якого зашифровані програмою-зидником, не зможе здійснювати майнінг криптовалюти. У результаті оцінки програмної й апаратної конфігурації комп'ютера, а також системи захисту від шкідливого ПЗ, вибирається найбільш ефективна атака. ПЗ для схованого майнінгу раптово знайшло величезну популярність серед кіберзлочинців.

Зліт криптоджекерів в 2026

У четвертому кварталі 2025 р. постачальником антивірусного ПЗ McAfee було виявлено близько 400 000 зразків шкідливих програм для схованого майнінгу криптовалют. А в першому кварталі 2026 р. це число різко збільшилося на 629 відсотків і досягло небувалого значення в 2,9 мільйони. У другому кварталі воно виросло ще на 86 відсотків, додавши 2,5 мільйони нових зразків. У цей же час схожі звіти були опубліковані постачальником антивірусного ПЗ TrendMicro: у них відзначався стрибок кількості подібних атак, у результаті якого воно виросло на 956 відсотків з початку 2025 р.

Ситуація погіршується тим, що шкідливе ПЗ для майнінгу криптовалют розробляється не тільки у вигляді застосунків, що приховано працюють на заражених комп'ютерах Windows або Linux, але й у вигляді майнінгових сервісів. Злочинні організації непомітно вбудовують невеликий фрагмент коду JavaScript на веб-сайти. Цей код частково або повністю використовує обчислювальну потужність будь-якого браузера, у якому відкритий такий сайт. У такий спосіб комп'ютер бере участь у видобутку криптовалют без відома користувача й без якої-небудь винагороди за використанні системні ресурси.

Цей метод швидко набрав оберти. У листопаді 2025 р. компанією AdGuard, творцем популярного браузерного плагіна для блокування реклами, був відзначений ріст на 31 відсоток для випадків криптоджекінгу (схованого браузерного майнінгу криптовалют). У ході дослідження було виявлено більше 30 000 веб-сайтів з убудованими сценаріями для майнінгу, такими як Coinhive, від якого по різних звітах постраждала кожна п'ята організація по усьому світі. У лютому 2026 р. у звіті Bad Packets Report фігурувало 34 474 сайтів з убудованим сценарієм Coinhive. Це самий популярний майнер на базі JavaScript, що також використовується для законного майнінгу криптовалют. У липні 2026 р. звіт Check Point Software Technologies показав, що чотири з десяти самих популярних зразків шкідливого ПЗ являють собою майнери криптовалют. ПЗ для схованого майнінгу криптовалют торкнулося майже половину організацій по усьому світі й

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

обігнало програми-зидирники, ставши самої великої й розповсюдженної кібер-погрозою на сьогоднішній день.

Як працює незаконний майнінг криптовалют

Щоб шкідливе ПЗ для схованого майнінгу потрапило на цільові пристрої, кіберзлочинці використовують самі різні методи – від впровадження на окремі комп'ютери й мобільні пристрої до злому популярних веб-сайтів з метою зараження всіх відвідувачів. Фішинг і цільовий фішинг за допомогою електронної пошти, розрахований на те, що довірливий користувач відкриє шкідливе посилання або вкладення, залишається досить популярним і ефективним напрямком атак. У деяких варіантах шкідливого ПЗ як компоненти використовуються мережні хробаки, за допомогою яких воно поширюється з одного зараженого комп'ютера на інші машини, підключені до тої ж мережі. Шкідливий код EternalBlue через яке в 2025 р. поширення програми-зидирника WannaCry придбало масштаби світової епідемії, дотепер застосовується розповсюджувачами ПЗ для схованого майнінгу. Але на відміну від жертв програм-зидирників, більшість потерпілих від схованого майнінгу криптовалют не підозрюють про те, що хтось заробляє за їхній рахунок, а просто почувають, що система стала повільніше працювати.

Інший популярний метод зараження – це фіктивні відновлення ПЗ. Наприклад, шкідлива програма може бути замаскована під відновлення Adobe Flash Player і, щоб замести сліди, дійсно обновляє флеш-плеер, але попутно встановлює ПЗ для схованого майнінгу. Ще один розповсюджений метод являє собою впровадження шкідливого сценарію в нормальний веб-сайт або блокування рекламних оголошень, убудованих у багато сайтів. Коли користувач заходить на веб-сайт або браузер завантажує рекламне оголошення, запускається процес майнінгу криптовалюти, що використовує обчислювальні ресурси без відома користувача.

Розроблювачі ПЗ для схованого майнінгу враховують минулі помилки. Зараз уже рідко можна зустріти шкідливе ПЗ, що поглинає 100 відсотків

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

потужності ЦП. Це приводить до значної з роботи комп'ютераа, що користувач швидше за все помітить і спробує усунути. Нові зразки шкідливого ПЗ для майнінгу діють обережніше, щоб сховати своя присутність. Вони завантажують ЦП не більше ніж на 20 відсотків, очікують бездіяльності користувача для виконання самих ресурсномістких обчислень і т.д. У такий спосіб ці майнери криптовалют можуть незаконно використовувати ресурси комп'ютера й дуже довго залишатися непоміченими.

Більше того, зараз не потрібно мати навички розробки ПЗ, щоб зайнятися незаконним майнінгом. Як і інші набори шкідливого ПЗ, послугу криптоджекінгу можна придбати в даркнеті всього за півдолара. Високий рівень конфіденційності й анонімності, властивий деяким криптовалютам, таким як Monero і Zcash, також сильно утрудняє відстеження й піймання злочинців.

Наприклад, Monero використовує публічний реєстр для створення й відстеження обміну цифровими токенами, але у відомостях про транзакцію ховається джерело, адресат і сума переведеної криптовалюти. Недавнє дослідження показало, що убудований майнер Coinhive генерує криптовалюту Monero на суму 250 000 доларів на місяць. По даним цього ж дослідження, опублікованого Рейнско-Вестфальским технічним університетом Ахена в Німеччині, криптовалюта Monero становить 75 відсотків усього браузерного майнінгу.

Популярні криптоджекери

Smominru

Smominru – мабуть, найвідоміший ботнет, що включає більше 520 000 заражених комп'ютерів, що до січня 2026 р. приніс своїм власникам більше трьох мільйонів доларів у криптовалюті Monero, чому сприяла інтелектуальна структура мережі з постійним самовідновленням. Smominru працював на основі експлойта EternalBlue, украденого в АНБ, що також використовувався для поширення світової епідемії програми-зидника WannaCry в 2025 р.

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

BadShell

Добре продумані криптоджекери, такі як BadShell, ховаються усередині довірених процесів, наприклад Windows PowerShell, і через них виконують сховані сценарії майнінгу. Деякі традиційні антивіруси здатні виявити цю погрозу, оскільки звичайно довіряють підписаним компонентам Windows, таким як PowerShell, за замовчуванням.

Coinhive

Майнінговий код Coinhive споконвічно розроблявся (і дотепер використовується) як законний засіб монетизації веб-сайтів, але до теперішнього часу став найбільшою у світі погрозою криптоджекінгу. Цікаво, що компанія, відповідальна за Coinhive, одержує 30 відсотків із всіх операцій майнінгу, у тому числі зі зламаних екземплярів.

MassMiner

MassMiner – цікавий приклад, що незвичайний тим, що використовує безліч експлоїтів різних уразливостей в одному пакеті. Експлуатація не виправлених уразливостей в Oracle WebLogic, Windows SMB і Apache Struts принесла творцям MassMiner майже 200 000 доларів у криптовалюті Monero.

Prowli

Prowli являє собою масштабний ботнет, що включає більше 40 000 заражених веб-серверів, модемів і інших пристроїв Іо, які використовуються для майнінгу криптовалюти й перенапрямку користувачів на шкідливі веб-сайти. Частиною Prowli є мережний хробак для підбора паролів методом повного перебору, щоб прискорити поширення майнера криптовалюти Monero. У деяких випадках ботнет також установлює бекдори на заражені системи.

WinstarNssmMiner

У травні 2026 р. шкідливою програмою WinstarNssmMiner за три дні були заражені більше півмільйона систем. Цей криптоджекер не діє, якщо на цільовій машині виявлене ефективне антивірусне ПЗ, і активується тільки на системах зі слабким захистом. Крім того, при спробі видалити WinstarNssmMiner заражена система виходить із ладу.

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

Acronis True Image із захистом від криптоджекерів

У рамках постійної роботи над засобами кіберзащити Acronis спостерігає за явищем схованого майнінгу криптовалют, що перетворився в масштабну погрозу по усьому світі. Щоб захистити наших корпоративних і домашніх користувачів від шкідливого ПЗ для майнінгу криптовалют, у даній роботі розширили можливості захисту від програм-зидників Acronis Active Protection за допомогою технології, що дозволяє боротися зі схованим майнінгом.

У цій поліпшеній версії Acronis Active Protection використовуються передові методи машинного навчання для виявлення й блокування всіх відомих процесів криптоджекінгу в системах Windows. При виявленні майнера криптовалюти Acronis Active Protection повідомляє адміністратора системи про потенційно несанкціоновану діяльність. (Ця можливість споконвічно доступна в продукті для домашніх користувачів Acronis True Image, у найближчому майбутньому в даній роботі плануємо додати її в корпоративні продукти Acronis Cyber Backup.)

Боти й ботнети в 2026 році

Завдяки широкому висвітленню в пресі подій, пов'язаних з Mirai і іншими спеціалізованими ботнетами, діяльність останніх стала багато в чому асоціюватися з DDoS-атаками. Однак це тільки верхівка айсберга, і ботнети широко використовуються не тільки для здійснення DDoS, але й для крадіжки різної користувацької інформації, у тому числі фінансової. Типовий сценарій атаки звичайно наступний:

1. Виробляється спроба зараження пристрою шкідливим ПЗ (якщо ціль ботовода – фінанси користувача, то троянцем-банкером). Якщо зараження проходить успішно, пристрій разом зі зловредом стає частиною ботнету під керуванням командного центра.

2. Шкідливе ПЗ на зараженому пристрої одержує від керуючого центра команду, що містить маску цілі (наприклад, URL-адреса сервісу онлайн-банкінгу) і інші необхідні для атаки дані.

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

3. Одержавши команду, зловред відслідковує дії користувача зараженого пристрою, і коли той заходить на ресурс, що підходить під маску цілі, здійснює атаку.

Основні типи атак з використанням ботнету:

- веб-инжект;
- підміна URL;
- підміна DNS;
- збір даних при відвідуванні певного ресурсу (статистика відвідування, дані з відвідуваної сторінки, знімок екрана).

На відміну від DDoS-атак, у результаті яких страждають веб-ресурси організації-жертви, збиток від розглянутих у цьому звіті атак несе безпосередньо клієнт організації. Результатом успішної атаки може бути:

- перехоплення користувальницьких облікових даних;
- перехоплення даних банківських карт;
- підміна адресата операції (наприклад, одержувача банківської транзакції);
- здійснення якої-небудь операції потай від користувача, але від його ім'я.

Подібні сценарії справедливі не тільки для банківських аккаунтів користувача, але, як у даній роботі побачимо далі, і для інших сервісів.

Методологія

Експерти відслідковують дії ботнетів за допомогою технології Botnet Monitoring, що емулює заражені комп'ютери (ботів) для одержання оперативних даних про дії операторів ботнету.

У даний аналіз входять зареєстровані технологією Botnet Monitoring унікальні атаки за 2025 і 2026 роки, отримані на основі аналізу перехоплених конфігураційних файлів ботів і команд від керуючих серверів.

Ціль атаки – URL-маска, витягнута з конфігураційного файлу бота або перехопленої команди, наприклад, маска на URL-адресу сайту онлайн-банкінгу.

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

Під сімейством у даному звіті розуміються привселюдно відомі імена шкідливого ПЗ (наприклад, Zeus, TrickBot (Trickster), Cridex (Dridex, Feodo, Geodo, etc.), Ramnit (Nimnul)).

Приклади масок цілей, що втримуються в зареєстрованих командах

За унікальну атаку в даному аналізі приймається унікальна комбінація маски цілі й сімейства або його модифікації, що одержало команду на атаку. Інші дані про атаку (інжектуючі скрипти, правила підміни криптогаманців, URL-адрес, правила перенаправку трафіку, шаблони для перехоплення облікових даних і ін.) не враховувалися при визначенні унікальності атаки.

З аналізу виключені атаки, пов'язані з ресурсами компаній, що займаються розробкою антивірусних рішень, тому що в цьому випадку це є захисною мірою з боку шкідливого ПЗ для запобігання «лікування» зараженого пристрою (запобігання скачування захисного рішення). Крім того, з аналізу виключені атаки, у яких неможливо однозначно визначити мету, тобто з маски цілі неможливо одержати додаткову інформацію про мету (наприклад, ціль «*bank*» у банкера BetaBot не попадає в аналіз).

Також в аналізі враховується тільки кількість унікальних атак, а не загальна кількість атак конкретного сімейства, тому що частота одержання команд відрізняється від сімейства до сімейства.

Представлені результати засновані на аналізі команд від більш ніж 60 тисяч різних командних центрів, зв'язаних з 150 сімействами шкідливого ПЗ і їхніми модифікаціями.

Статистика

Загальна кількість унікальних атак на клієнтів організацій, зареєстрованих за допомогою технології Botnet Monitoring, в 2026 році знизилася на 23,46% у порівнянні з попереднім роком роком (з 20 009 шт. в 2025 році до 15 314 шт. в 2026 році).

При цьому 39,35% атак, які в даній роботі спостерігали в 2026 році, є новими, тобто така комбінація маски цілі й сімейства, що одержало команду на

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

атаку, не зустрічалася в 2025 році. Це зв'язано як з появою нових банкерів (Danabot, BackSwap), так і із прагненням авторів шкідливого ПЗ змінити кількість своїх цілей.

Географія цілей атак в 2025 році містила в собі 111 країн, в 2026 році були початі спроби атак на клієнтів організацій, що перебувають в 101 країні миру.

Цілі зловмисників

Для початку розглянемо, клієнти яких організацій є найбільш кращими цілями для зловмисників.

В 2025 році найбільше число цілей атак відносилось до категорії «Фінансові сервіси» (77,44%). У неї входять сервіси онлайн-банкінгу, мультибанкінг-сервіси, онлайн-магазини й інші ресурси, пов'язані з фінансовими операціями (за винятком криптовалют). Такий результат цілком закономірний у силу найбільшої потенційної вигоди для зловмисника, що у випадку успішної атаки фактично одержує безпосередній доступ до фінансів жертви.

На другому місці по кількості унікальних атак розташувалася категорія «Глобальні портали й соціальні мережі» (6,15%). У неї входять пошукові системи, сервіси електронної пошти й соціальні мережі. Пошукові системи включені в цю групу, тому що найчастіше саме на головній сторінці такої системи перебуває форма входу в поштову скриньку, дані якого й намагається одержати зловмисник за допомогою описаних вище типів атак.

На третім місці в нашій рейтингу розташовуються ресурси, що надають різні продукти й послуги (5,08%), але не є при цьому онлайн-магазинами. Наприклад, сюди попадають хостинг-провайдери. У їхньому випадку метою зловмисників, як і в першій категорії, є платіжні дані жертви. Такі ресурси винесені в окрему групу, тому що пропонують придбати конкретний товар або послугу, що показує, наскільки точково можуть діяти зловмисники.

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

В 2026 році відбулися невеликі зміни в першій трійці цілей атак на клієнтів різних організацій. Цікаво, що частка унікальних атак, що доводяться на фінансові сервіси, знизилася на 3,51 п. п. до 73,93%.

Практично завжди в масці цілі, одержуваної ботом, зазначений домен або його частина. Проаналізувавши домени з масок, які ставляться до фінансових організацій (банки, інвестиційні, кредитні, пенсійні установи й т.п.), у даній роботі склали карту організацій, клієнтів яких боти атакували в 2026 році. На карті зазначений діапазон числа доменів фінансових організацій, які були замічені в командах ботам. Варто відзначити, що однієї організації може належати декілька доменів, наприклад, розділених по територіях однієї країни.

Рік 2026-й ознаменувався підвищенням інтересом «ботоводів» до криптовалюти: кількість унікальних атак на користувачів пов'язаних із криптовалютою сервісів (біржі, криптовалютні гаманці й т.п.) збільшилося, їхня частка виросла більше чим утрое (на 4,95 п. п.) і склала 7,25%.

Зловмисники активно намагалися монетизувати інтерес до криптовалют і одержати у своїх жертв дані, необхідні для крадіжки засобів. У більшій частині виявлених нами атак на користувачів криптовалютних сервісів брав участь банкер Ramnit (53%). Крім того, різко збільшили кількість унікальних масок, пов'язаних із криптогаманцями й криптобіржами, банкери Chthonic і Panda, що є модифікаціями нашумілого банкера Zeus. Серйозну активність у цій сфері проявляє й троянець CarCoiner, націлений саме на подібні ресурси.

Географія цілей атак

Якщо в масці цілі зазначений TLD (top level domain, домен верхнього рівня), по якому можна визначити країну, то в статистику заноситься ця країна. Якщо по TLD не можна визначити країну (наприклад, .com), у статистику заноситься країна, де перебуває штаб-квартира організації.

В 2026 році состав учасників TOP 10 країн, до яких ставиться найбільша кількість унікальних масок цілей, залишився колишнім, змінилися лише їхні

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

позиції в нашій рейтингу. Як і роком раніше, в 2026 році найбільш частими цілями атак були клієнти організацій, що перебувають у США.

В 2026 році істотно знизилася частка унікальних атак, які довелися на клієнтів розташованих у Німеччині організацій. Це пов'язане з тим, що в 2025 році більша частина цих атак відбувалася банкером BetaBot(майже 75% всіх зареєстрованих унікальних атак), а в 2026 році його частка в цьому показнику ледь перевищує 1,5%. Навіть зареєстровані в 2026 році атаки Danabot на клієнтів німецьких банків не «допомогли» Німеччині удержати 2-е місце в нашій рейтингу.

Серед інших змін можна відзначити зниження частки клієнтів австралійських організацій, що атакуються, з 4,67% до 2,29%. Кількість унікальних масок на «австралійському напрямку» скоротили практично всі боти. Так, наприклад, серед спостережуваних нами атак банкера Gozi в 2026 році практично не було атак на клієнтів фінансових організацій Австралії, тоді як роком раніше вони становили більше 90% зареєстрованих атак цього зловреду.

Але є й погані новини. Багато хто зловреди, навпаки, розширили свою географію: в 2026 році банкер Trickster(TrickBot) додав до списку цілей відразу 11 країн, троянец SpyEye – 9, а банкер IcedID – 5 країн.

Користувачів криптовалютних сервісів найчастіше атакували в США, Люксембурзі й Китаї, що не дивно: у цих країнах зареєстрована велика кількість таких сервісів. Крім того, в 2026 році значно збільшилася кількість атак на користувачів сервісів, зареєстрованих у Великобританії, Сінгапурі, Естонії, Південній Кореї й Швейцарії.

Географія командних центрів

У даному розділі представлена статистика по географії командних центрів (C&C) ботнетів, які відправляли команди на здійснення атаки.

В 2025 році найбільше число командних центрів розташовувалося на Україні (24,25%), при цьому майже 60% з них склали командні сервери вже згаданого банкера Gozi.

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

В 2026 році лідируючу позицію по кількості командних центрів, що управляють атаками на клієнтів різних організацій, зайняла Росія (29,61%). Більше половини (54%) від цих командних центрів використовуються банкером Panda.

Найбільш активні сімейства

BetaBot

Частка унікальних атак, зроблених банкером BetaBot – 13,25% від всіх унікальних атак в 2026 році.

Основні характеристики (частки зазначені щодо кількості унікальних атак BetaBot):

- Географія цілей: 42 країни.
- Найбільш атакуємі країни: США (73,60%), Китай (6,35%), Великобританія (6,11%).
- Найбільш атакуємі категорії організацій: «Фінансові сервіси» (37,43%), «Глобальні портали й соціальні мережі» (18,16%).

Trickster (TrickBot)

Частка унікальних атак в 2026 році, зроблених банкером TrickBot, становить 12,85% від всіх унікальних атак.

Основні характеристики (частки зазначені щодо кількості унікальних атак TrickBot):

- Географія цілей: 65 країн.
- Найбільш атакуємі країни: Великобританія (11,02%), США (9,34%), Німеччина (7,99%).
- Найбільш атакуємі категорії організацій: «Фінансові сервіси» (96,97%), «Криптовалютні сервіси» (1,72%).

Panda

На банкер Panda в 2026 році довелося 9,84% всіх унікальних атак.

Основні характеристики (частки зазначені щодо кількості унікальних атак Panda):

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

- Географія цілей: 33 країни.
- Найбільш атакуємі країни: Канада (24,89%), США (22,93%), Італія (17,90%).
- Найбільш атакуємі категорії організацій: «Фінансові сервіси» (80,88%), «Криптовалютні сервіси» (10,26%).

SpyEye

Зареєстровані атаки SpyEye склали 8,05% всіх унікальних атак в 2026 році.

Основні характеристики (частки зазначені щодо кількості унікальних атак SpyEye):

- Географія цілей: 32 країни.
- Найбільш атакуємі країни: США (35,01%), Великобританія (14,38%), Німеччина (13,57%).
- Найбільш атакуємі категорії організацій: «Фінансові сервіси» (98,04%).

Ramnit

Частка унікальних атак в 2026 році, зроблених Ramnit, становить 7,97% всіх унікальних атак, зареєстрованих технологією Botnet Monitoring. Географія Ramnit вражає своїм розмахом і містить у собі 66 країн.

Основні характеристики (частки зазначені щодо кількості унікальних атак Ramnit):

- Географія цілей: 66 країн.
- Найбільш атакуємі країни: Великобританія (25,70%), США (20,12%), Китай (7,78%).
- Найбільш атакуємі категорії організацій: «Фінансові сервіси» (47,76%), «Криптовалютні сервіси» (46,83%).

Висновок

- Основні тенденції, які в даній роботі відзначили в 2026 році, проаналізувавши отримані команди атакувати клієнтів різних організацій:
- Зниження загального числа зареєстрованих унікальних атак може свідчити про прагнення зловмисників створювати маски цілей, які покривають

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

велика кількість ресурсів однієї організації й залишаються актуальними протягом довгого часу.

- Метою абсолютної більшості атак як і раніше є фінансові організації і їхні клієнти.

- Помітно виросло (у порівнянні з 2024 роком) кількість атак на клієнтів криптовалютних сервісів. Зниження кількості таких атак не варто очікувати – навпаки, можливе збільшення їхнього числа, тому що усе більше ботів включають у свій арсенал веб-інжекти для подібних ресурсів.

- Активно відбувається додавання нових цілей. Зловмисники створюють не маски, що зустрічалися раніше, а також змінюють старі, щоб «покривати» більше число сторінок, на яких можна викрасти дані користувача або його гроші.

2.2 Обґрунтування вибору засобів для побудови системи кібербезпеки та мови програмування

Python – високорівнева мова програмування, яку називають другою за популярністю в світі. Її використовують для розробки вебзастосунків, програмного забезпечення, машинного навчання. Python застосовують для вирішення робочих завдань у компаніях Google, Instagram, Facebook, IBM, NASA, Dropbox, Netflix та інших. Розробники цінують цю мову програмування за простоту у вивченні, ефективність та мультиплатформність.

Python – скриптова мова програмування з досить простим синтаксисом. Для розуміння достатньо порівняти принципи написання найпростішої програми, яка виводить на екран текстове повідомлення. Саме тому мова програмування Python більш доступна для новачків, а професіонали встигли адаптувати її для вирішення великої кількості завдань. Це мультиплатформне рішення, тому знання Python дає можливість працювати у різних сферах: від розробки мобільних застосунків до ігрової індустрії та штучного інтелекту. У мови програмування динамічна типізація: є можливість передавати до функцій будь-

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

який тип даних без попереднього вказання. Інтерпретованість дозволяє знаходити помилки у коді ще до повної збірки у робочий застосунок. При цьому Python дуже чітко дає зрозуміти, де та через що виникла помилка.

Це мова об'єктноорієнтованого програмування (ООП). Програмне забезпечення на Python оформлене у вигляді моделей, які можуть бути зібраними у пакети. Тип та структуру кожного об'єкта можна запитати під час виконання програми. Для кожного з об'єктів можна отримати всю інформацію щодо його внутрішньої структури. Окрім того:

- у мови логічний синтаксис, завдяки чому вихідний код легко читати та розуміти;
- гнучкість та масштабованість Python дозволяє адаптувати високорівневу логіку та розширяти складні застосунки, як тільки виникне така необхідність;
- розробка на Python у більшості випадків проходить швидше, ніж на інших мовах програмування;
- Python – інтерпретована мова програмування. Це значить, що код можна написати у будь-якому текстовому файлі на будь-якій платформі, і потім успішно запустити;
- у Python – колосальна спільнота однодумців. Тож будь-які складнощі конкретних розробників вирішуються колективно.

Проте є декілька особливостей, які можна віднести до недоліків. Це повільність (ця мова програмування хоч і універсальна, проте повільніша за інші), велика кількість ресурсів, необхідних для роботи та «прив'язаність» до системних бібліотек.

Мова програмування Python використовується у наступних сферах:

1. Розробка програмних застосунків будь-якого напрямку.
2. Розробка серверної частини мобільних застосунків (найпопулярніший напрямок).
3. Ігри. Багато сучасних ігор для комп'ютерів (наприклад, World of Tanks) частково чи повністю написані на Python.

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

4. Вбудовані системи для різних пристроїв. Дуже часто Python використовують для написання внутрішніх платформ управління банкоматами.

5. Скрипти та плагіни до уже реалізованих програм для автоматизації процесів чи створення інших рішень.

6. Тестування (автоматизація цього процесу).

7. Машинне навчання. – основна мова для написання алгоритмів і аналітичних застосунків у сфері Machine Learning.

Бібліотеки Python

Різні бібліотеки Python використовують для виконання конкретних завдань. Наприклад, Matplotlib підходить для відображення даних у двовимірній та тривимірній графіці. Pandas підходить для зручної роботи з даними. NumPy дозволяє створювати масиви та керувати ними. Requests використовується для веброзробки. OpenCV-Python відкриває можливості для обробки зображень з метою оптимізації систем «машинного зору».

Найвідоміші фреймворки для мови програмування Python

Фреймворки Python допомагають створити зручне та функціональне середовище для розробки. У них міститься набір інструментів, модулів та бібліотек, корисних для виконання конкретних завдань. Це значно полегшує роботу: наприклад, дає змогу не витратити час на розписування дій, які повторюються, а використати релевантний інструмент. Тож є можливість позбутися рутинних процесів та сконцентруватися на логіці проєкту.

Серед найпопулярніших фреймворків для Python:

– Django – найстаріший та найвідоміший. Створений для реалізації великих інтерактивних проєктів;

– Pyramid – зручний у налаштуваннях, і дає можливість реалізувати складні нестандартні ідеї;

– Web2py – підходить в першу чергу для вебзастосунків і може використовуватись на будь-яких архітектурах.

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускну кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи кібербезпеки для протидії криптоджекінгу та ботнетам.

В процесі розробки випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти необхідно виконати наступний обсяг роботи:

а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;

б) вибрати та обґрунтувати методику побудови системи кібербезпеки контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;

в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;

г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації в роботі технологічного обладнання;

д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи кібербезпеки в промислову експлуатацію та її подальшу успішну експлуатацію;

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

ж) розробити заходи по охороні праці при впровадженні та експлуатації системи, а також розробити заходи з цивільного захисту;

з) сформулювати висновки про виконаний обсяг робіт та одержані результати.

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Криптоджекінг

Майнінг (видобуток) криптовалюти необхідний для роботи системи. Майнінг складається із серії обчислень, здійснюваних для обробки транзакцій у Блокчейн. Він створює нову криптовалюту й підтверджує транзакцію у всій блокчейн-мережі. Щоб більше створювати криптомонет, необхідно їх добувати. Без майнінгу система може звалитися.

Багато користувачів самі стали займатися майнінгом, щоб заробити гроші. Майнери виконують математичні операції для підтвердження транзакцій, а для цього вони використовують спеціальне ПЗ. Таким чином, щоб майнінг був прибутковим, необхідно мати величезні обчислювальні потужності. Щоб робити гроші на майнінгу, кібер-злочинці стали займатися **криптоджекінгом** (cryptojacking).

Криптоджекінг полягає в несанкціонованому використанні пристроїв користувача для видобутку криптовалюти. В основному, хакери використовують шкідливе ПЗ для злому комп'ютерів, планшетів або смартфонів, після чого використовують їх для схованого майнінгу криптовалют. Можливо, користувач помітить невелике зниження швидкості роботи його пристрою, але навряд чи подумає про те, що це пов'язане зі спробою атаки на нього для майнінгу криптовалют. Одна з найпоширеніший технік полягає в тому, щоб одержати контроль над процесором пристрою жертви (CPU) або процесором його відеокарти (GPU) через відвідування якого-небудь веб-сайту, зараженого шкідливою програмою для майнінгу криптовалют [1].

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

Як відбувається схований майнінг?

Найчастіше сховано майнят піратські популярні сайти. У принципі, виявити це досить просто. При такому втручанні завантаження процесора різко збільшується практично до ста відсотків. Однак, що завантажуються торренти й без цього навантажують систему, що не дозволяє визначити майнінг[2].

Для того, щоб побачити навантаження процесора на Windows, потрібно зайти в «Диспетчер завдань» (Task Manager). В MacOS цю функцію виконує програма «Моніторинг системи»(Activity Monitor).

Уникнути схованого майнінгу можна декількома способами:

- Установити спеціальне розширення, що блокують веб-майнінг.
- Відключити JavaScript.
- Використовувати надійний антивірус. Антивірусні програми найчастіше бачать майнери, як потенційно безпечні, але при цьому їх можна використовувати в шкідливих цілях, тобто, ризиковані.

Як компанія може захистити себе від криптоджекінгу?

Такі атаки мають серйозні наслідки для підприємств. Найбільш очевидні наслідки виникають у результаті крадіжки ресурсів процесорів, що може сповільнити роботу систем і мереж, піддавши підприємство й всю систему серйозним ризикам. Більше того, після того як компанія була атакована, цілком імовірно, що буде потрібно досить багато часу й засобів для усунення даної проблеми. Інтенсивний майнінг криптовалют також може мати фінансові наслідки для компаній, тому що в результаті підвищеного використання ІТ-ресурсів повинен спостерігатися ріст споживаної потужності, а це веде до підвищених витрат на електроенергію.

Крім того, такі атаки можуть завдати шкоди корпоративним пристроям. Якщо майнінг здійснюється протягом тривалого періоду часу, то пристрою і їхніх батарей часто випробовують надмірне навантаження й перегрів, що також знижує ресурс роботи цих пристроїв.

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

Звичайно, не слід також забувати, що якщо ви стали жертвою криптоджекінгу, те це означає, що хакери змогли перебороти ваші системи безпеки й одержати контроль над корпоративними пристроями, піддавши серйозному ризику конфіденційність корпоративних даних.

Щоб захиститися від можливої атаки по майнінгу криптовалют, у даній роботі рекомендуємо вам дотримувати наступних мір безпеки:

- Періодично проводите оцінки ризиків для виявлення уразливостей.
- Регулярно обновляйте всі ваші системи й пристрої.
- Впроваджуйте рішення розширеної інформаційної безпеки, які дозволяють одержати повну видимість активності на всіх кінцевих пристроях і контролювати всі запуснені процеси.
- Створіть безпечне середовище для перегляду сайтів, установивши розширення, які перешкоджають майнінгу криптовалют.

Фахівці з компанії Check Point Software у звіті Cyber Security Report 2025 розповіли про основні інструменти, які кіберзлочинці використовували для атак на компанії по усьому світі в 2026 році. Про це стало відомо 21 січня 2025 року.

За словами експертів, серед шкідливого ПЗ домінують криптомайнери, незважаючи на той факт, що кількість криптомайнінгових атак в 2026 році скоротилося. 38% компаній по усьому світі стали жертвою даного шкідливого ПЗ. Популярність криптомайнерам забезпечують невеликі ризики й високий дохід. За результатами опитування, лише 7% українські компанії побоювалися криптомайнерів.

28% компаній були атаковані ботнетами, що на 50% перевищує даний показник за 2026 рік. Лідером серед вредоносів завдяки своїй універсальності став Emotet.

Як відзначили фахівці, кількість атак на мобільні пристрої знизилася – з 33% в 2025 році до 27% в 2026. За даними опитування, тільки 16% ІТ-експертів в Україні встановлюють або планують використовувати спеціальне ПЗ для захисту

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

мобільних пристроїв, а 52% респондентів вважають найефективнішою мірою захисту заборона на користування персональними смартфонами по роботі.

Експерти відзначили збільшення кількості підприємств, що використовують хмарні сервіси, – до 90%. Однак більше половини опитаних (67%) ІБ-фахівців поскаржилися на недостатню прозорість їхньої хмарної інфраструктури, безпеки й невідповідності вимогам. Головною причиною атак на хмарні сервіси як і раніше залишається неправильне налаштування їхніх ресурсів[3].

Ботнет

Ботнет – це мережа пристроїв, інфікованих шкідливим ПЗ й підключені до інтернету. Ботнети часто використовуються в DDoS-атаках. Крім цього, хакери часто користуються їх сукупною обчислювальною потужністю, щоб відправляти великі обсяги спаму, викрадати доступи великої кількості користувачів або шпигувати за людьми й компаніями.

Зловмисники будують ботнети, заражаючи підключені до інтернету пристрою шкідливим ПЗ й управляючи ними за допомогою командного сервера. Як тільки хакер заражає один пристрій у мережі, інші пристрої також ризикують стати інфікованими.

Атака ботнету може бути нищівною: в 2016 році ботнет Mirai відключив більшу частину інтернету, включаючи такі сайти, як Twitter, Netflix, CNN, а також українські банки й всю Ліберію. Ботнет скористався незахищеними пристроями інтернету речей, такими, як камери безпеки, і встановив на них шкідливе ПЗ, що атакувало сервери, перенаправляючи інтернет-трафік.

Приклади відомих ботнетів

Mirai

Згідно даним звіту, опублікованого компанією Fortinet у серпні 2026, Mirai був одним із самих активних ботнетів. Через 2 роки після його створення в ботнету Mirai з'явилися нові функції, – наприклад, здатність перетворювати інфіковані пристрої в комплекси шкідливого проксі ПЗ й майнінгові пристрою.

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

Reaper

Mirai – це тільки початок. Восени 2024 інженери компанії Check Point виявили новий ботнет відомий під назвами IoTroop і Reaper. Він зламує пристрої в інтернеті речей ще швидше, ніж Mirai.

Mirai заражав уразливі пристрої, які використовували дефолтні паролі й імена користувачів. Reaper пішов далі, зламуючи біля дюжини пристроїв різних виробників через уразливості – включаючи пристрою таких відомих компаній, як D-Link, Netgear і Linksys. Ботнет Reaper також гнучкий, що дозволяє хакерам легко оновлювати його код.

Reaper використовували в атаках на європейські банки цього року.

Користувачам важлива ціна, а не безпека

На жаль, технічно майже неможливо просто відключити від інтернету заражені пристрої й відстежити творців ботнетів. Коли покупці йдуть у магазин за камерою безпекою, їм важливі функції, популярність бренда й ціна. Рідко хтось бере до уваги безпека. Через те, що пристрою інтернету речей такі дешеві, імовірність того, що для них випускають регулярні відновлення, низка.

Але поки люди продовжують купувати дешеві й небезпечні пристрої, кількість уразливостей росте.

З боку виробників немає особливого бажання мінятися. Більшість із них не зіштовхується з наслідками продажу небезпечних пристроїв.

Складно/немає необхідності відслідковувати трафік при невеликій атаці

Ботнети звичайно контролюються центральним командним сервером. У теорії потрібно відключити центральний сервер, відстежити трафік до заражених пристроїв, щоб очистити їх і убезпечити. У теорії це просто, але це зовсім не так.

Коли ботнет такий великий, провайдери можуть об'єднатися, щоб зрозуміти, що відбуватися, і заблокувати трафік. Так було з ботнетом Mirai.

Але коли це щось невелике типу спаму, провайдери звичайно не обертають на цю увагу. Деякі провайдери попереджають користувачів, але це

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

такий невеликий масштаб, що попередження ніяк не зачіпають ботнет. Крім того, дуже складно відстежити трафік ботнету.

Проблематично обновляти уразливі пристрої

Деякі компанії безпеки намагаються працювати з інтернет-провайдерами, щоб виявляти заражені пристрої. Наприклад, компанія CrowdStrike співробітничав з інтернет-провайдерами по усьому світі й повідомляє їм, у яких пристроях є уразливості. Провайдери повинні зв'язуватися із власниками таких пристроїв і призивати їх полагодити.

Але проблема в тому, що таких пристроїв можуть бути мільйони, і на них потрібно встановлювати патчі безпеки. Плюс вилучена опція апгрейда часто відсутній. Крім цього деякі пристрої більше не підтримуються або зроблені так, що на них не можна встановити патчі. Тому навіть якщо пристрій інфікований, але він і раніше працює, власники не особливо хочуть позбутися від нього й купувати новий

Як попереджати атаки ботнетів

Установлювати відновлення для захисту від ботнетів

Ботнети використовують уразливості, щоб переходити від пристрою до пристрою й заподіювати найбільша шкода компанії. Найважливіше – регулярно обновляти всі системи, і краще, щоб це відбувалося автоматично.

Автоматично повинні обновлятися не тільки операційні системи й додатки, але й апаратне встаткування. ПЗ й апаратне забезпечення, для якого вже не випускаються відновлення, потрібно перестати використовувати.

Блокувати доступ

Всі компанії повинні використовувати мультифакторну авторизацію й принцип найменших привілеїв, при якому користувач одержує тільки необхідний мінімум доступів для досягнення цілей.

Один з ефективних кроків для безпеки – використовувати фізичні ключі безпеки. Google, наприклад, зобов'язала всіх працівників використовувати такі ключі в 2025 році. З тих пор жодна обліковий запис працівника не була зламана.

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

Але, на жаль, не всі компанії можуть потягнути це фінансово. Крім того, що впровадити саму технологію коштує недешево, є ще ризик того, що працівники втратять ключі.

Двофакторна автентифікація на смартфоні може бути непоганою заміною фізичним ключам. Це недорого коштує й додатково захищає дані.

3.2 Розробка структурної схеми

Блок захисту від криптоджекінгу

Криптоджекінг – це використання чужих комп'ютерів для майнінгу криптовалюти без відома власників цих комп'ютерів. Говорячи по-простому, це шкідливий майнінг: негарні люди тим або іншим способом непомітно встановлюють на чужі комп'ютери програми-майнери (в ідеалі – масово) і одержують весь прибуток від видобутку криптовалюти, не несучи при цьому ніяких витрат – ні на встаткування, ні на електрику.

Якщо ви ще не чули про криптоджекінг, то зараз найкращий час познайомитися із цим явищем, і от чому.

Може здатися, що падіння курсів криптовалют повинне було привести до зниження популярності криптоджекінгу – адже прибуток від майнінгу знизилася. Але насправді всі зовсім навпаки. На RSA Conference 2026 аналітики Cisco Umbrella розповіли про те, що пов'язаний із криптомайнінгом трафік організацій по усьому світі за рік різко виріс (приблизно в 200 разів), і це прямо пов'язане з ростом популярності криптоджекінгу. Швидше за все, справа в тому, що за період росту курсів криптовалют багато з людей встигли звикнути до високих доходів. І після падіння ринку, коли майнити із використанням власного встаткування, чесно оплачуючи рахунку за електрику, стало не вигідно, вони вирішили перемкнутися на альтернативний спосіб заробітку – криптоджекінг.

Основна втрата від криптоджекінгу несуть малі й середні бізнеси – на перші доводиться біля третини криптомайнінгового трафіку, на другі – половина.

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

Великі організації відповідають лише за 18% трафіку. Можна припустити, що це пов'язане з набагато більше серйозним відношенням великого бізнесу до інформаційної безпеки й більше розслабленим відношенням до даного питання в середніх і невеликих організацій.

Цікавий розподіл криптомайнінгового трафіку по індустріях: перше місце займає енергетичний сектор (34%), за ним ідуть утворення (26%) і охорона здоров'я (7%).

Судячи зі статистики Cisco Umbrella, головний удар криптожуликів довівся на Північну Америку: США відповідає за 62% криптомайнінгового трафіку й ще 2% доводиться на Канаду. На другому місці регіон ЕМЕА з більше скромними 6%. Найбільшою несподіванкою для дослідників стало те, що на третім місці виявилася ПАР – близько 2% пов'язаного з майнінгом трафіку доводиться на цю країну.

Не слід недооцінювати збиток від криптоджекінгу. Так, діяльність шкідливих майнерів набагато менш руйнівна, чим криптовимагачів. Але, по-перше, це в кожному разі прямі збитки – жертвам доводиться оплачувати обчислювальні потужності й електрика, і це може вилитися в чималі суми.

По-друге, сам факт, що на комп'ютерах вашої компанії відбувається якась нелегальна діяльність, не слід ігнорувати. Адже це означає, що у вашої ІТ-інфраструктурі є уразливості. І сьогодні їх використовують для криптоджекінгу – а завтра, коли він перестане бути вигідним, почнуть використовувати для чогось іншого:

- Використовуйте надійні захисні рішення. Переважно з технологіями, споконвічно розробленими для виявлення криптоджекінгу.

- На embedded-пристроях (табло, термінали й так далі) також використовуйте захисні рішення й по можливості включайте режим заборони за замовчуванням (Default Deny).

- Стежте за трафіком: часті звертання до доменам популярного криптомайнінгових пулів – явна ознака того, що хтось майнить за ваш рахунок.

– В ідеалі заблокуйте звертання до таким доменам для всіх комп'ютерів у вашій мережі – списки доменів можна знайти в Мережі. Не забувайте оновлювати цей список час від часу – нові домени з'являються постійно.

– Перевіряйте платформи хмарних обчислень, такі як Amazon EC2, Microsoft Azure і так далі, на аномальний ріст використання процесора.

Блок захисту від ботнетів

Термін «бот» – є скороченням від слова «робот». Зловмисники поширюють шкідливі програми, які можуть перетворити ваш комп'ютер у бот або зомбі-комп'ютер. У цьому випадку по команді через Інтернет ваш комп'ютер може виконувати певні завдання непомітно для вас.

За допомогою бот-мереж зловмисники можуть розсилати спам, поширювати віруси, атакувати комп'ютери й сервери, а також робити інші злочини й види шахрайства. Якщо комп'ютер стає учасником бот-мережі, його продуктивність може падати, а ви можете стати мимовільним співучасником злочинців.

Проведення DDoS-атак

DDoS-атака (від англ. Distributed Denial-of-Service) – це атака на комп'ютерну систему, наприклад на веб-сайт, метою якої є доведення системи до «падіння», тобто стану, коли вона більше не зможе приймати й обробляти запити легітимних користувачів.

Збір конфіденційної інформації

Конфіденційна інформація, що зберігається на комп'ютерах користувачів, завжди буде залучати зловмисників. Найбільший інтерес представляють номери кредитних карт, фінансова інформація й паролі до різних служб: поштовим скринькам, FTP-серверам, «месенджерам» і ін.

Розсилання спаму

Щодня по усьому світі курсують мільйони спам-повідомлень. Розсилання незапитаної пошти є однієї з основних функцій сучасних ботнетів. Близько 80% усього спаму розсилається через зомбі-мережі.

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

Створення пошукового спаму

Ще один варіант використання ботнетів – підвищення популярності сайтів у пошукових системах. Працюючи над пошуковою оптимізацією, адміністратори ресурсів намагаються підвищити позицію сайту в результатах пошуку. Спеціально створена програма завантажується на зомбі-комп'ютер і від імені його власника залишає на популярних ресурсах коментарі з посиланнями на сайт, що розкручується.

Злочинці можуть включити ваш комп'ютер у бот-мережу в такий спосіб:

- Передавати шкідливе ПЗ в складі завантажень, які, на вашу думку, є картинками або фільмами, або через посилання, які можна знайти в повідомленнях електронної пошти, миттєвих повідомленнях або на сайтах соціальних мереж.

- Злякати вас, так щоб ви нажали кнопку або клацнули посилання, що втримується в помилковому попередженні про комп'ютерний вірус.

У такий спосіб можна зробити вивід, що й у тім і іншому випадку їсти зміст аналізувати трафік на аномальність.

Структурна схема розробленої системи зображена на рисунку 3.1. На ній показано структуру системи виявлення дестабілізуючого широкомовного трафіку на основі аналізу мережної статистики.

Вона складається з наступних блоків:

- Блок моніторингу мережі.
- Блок аналізу мережної статистики.
- Блок визначення топології мережі.
- Блок виявлення аномального поведіння трафіку.
- Блок визначення криптоджекінгу.
- Блок визначення ботнету.
- Блок протидії криптоджекінгу та ботнетам.
- Блок зберігання результатів.

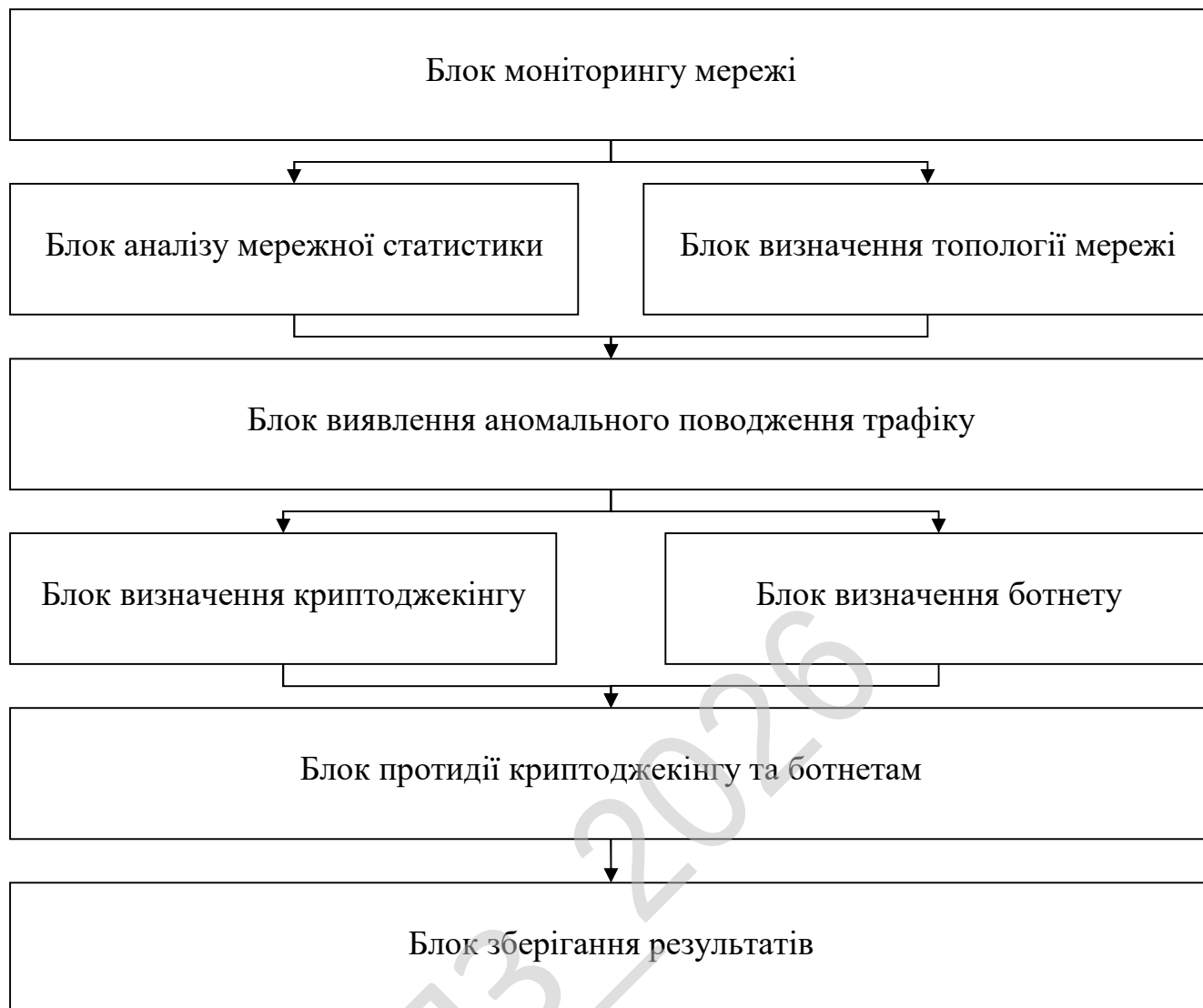


Рисунок 3.1 – Структурна схема системи

3.3 Розробка функціональної схеми

Функціональна схема розробленої системи зображена на рисунку 3.2. Так, як функціональна схема є більш подібним описом функціональних можливостей структурної схеми, то вона буде представляти собою, більш детальний варіант структурної схеми.

З рисунку видно, що розроблена система складається з наступних частин:

- Блок моніторингу мережі.
- Блок аналізу мережної статистики.

- Блок визначення топології мережі.
- Блок виявлення аномального поведіння трафіку.
- Блок визначення виду атаки.
- Блок зберігання результатів.

Блок аналізу мережної статистики

Блок збирання наступної інформації:

- HTTP.
- IP-адреса (IP address).
- Довжина пакету (Packet Length).
- Тип порту (Port Type).
- Основна статистика (Summary).
- Ієрархія протоколу (Protocol Hierachy).
- Сеанси обміну пакетами (Conversations).
- Точки призначення (Endpoints).
- Графіки I/O (IO Graphs).
- Список сеансів обміну пакетами (Conversation List).
- Список точок призначення (Endpoint List).
- Час чекання відповіді від сервісу (Service Response Time).
- RTP.
- SIP.
- Виклики VoIP (VoIP Calls).
- Призначення (Destination).
- Графік потоку (Flow Graph).

Розпишемо їх більш детально.

1. HTTP. HTTP (Hypertext Transfer Protocol, протокол передачі гіпертексту) – це протокол типу «клієнт-сервер», який використовується для передачі HTML-файлів. HTTP-клієнт (у більшості випадків це web-браузер) відсилає HTTP-запит до web-серверу із полем «URL», який допомагає знайти

потрібний файл. Web-сервер відповідає HTTP-пакетом та забезпечує клієнт необхідною web-сторінкою.

Меню «HTTP» містить три підменю:

- «Load Distribution» (Розподіл пакетів).
- «Packet Counter» (Лічильник пакетів).
- «Requests» (Запити).

2. IP-адреса. Відображення IP-адреси джерела або призначення мережових пакетів.

3. Довжина пакету.

4. Тип порту. Відображення статистики портів TCP або UDP.

5. Основна статистика. Доступні такі елементи основної статистики, як:

- Властивості захоплених файлів.
- Час захвату.
- Інформація про фільтр захвату.
- Інформація про фільтр відображення.

6. Ієрархія протоколу. Статистика ієрархії протоколу допомагає аналізувати пакети, розбиваючи відображені дані, які належать чинному рівню OSI.

7. Сеанси обміну пакетами. Якщо ви використовуєте протокол TCP/IP або програму, яка працює із цим протоколом, ви маєте побачити чотири активних вкладок для обміну пакетами за допомогою Ethernet, IP, TCP та UDP. «Діалог» між комп'ютерами відображає трафік між двома активними хостами. Номер, зазначений на вкладці після назви протоколу, означає кількість «діалогів» між хостами. Номер, зазначений на вкладці після назви протоколу, означає кількість «діалогів» між хостами, наприклад, «Ethernet:6».

8. Точки призначення. Точки призначення забезпечують статистику даними про відправку та прийом пакетів. Номер, зазначений на вкладці після назви протоколу, вказує на кількість точок призначення. Наприклад, «Ethernet:6».

9. Графіки I/O. Основний графік може бути отриманий за допомогою команди «IO graphs» (Графіки I/O). Ще декілька графіків можуть бути додані у тому ж вікні на основі фільтрів відображення.

10. Час чекання відповіді від сервісу. 13 протоколів доступні для глибокого аналізу.

11. RTP. RTP (Real-time Transport Protocol, протокол передачі у реальному часі, RFC 3550) – це протокол для передачі звука та відео через IP-мережу. Він працює у початку протоколу дейтаграм користувача (User Datagram Protocol, UDP). Він часто використовується у сукупності з протоколами SIP або H.233, забезпечуючи виконання сигнальних завдань.

12. SIP. SIP (Session Initiation Protocol, протокол встановлення сесії, RFC 3261) – це сигнальний протокол, який оголошує відео– або VoIP-сесії. Він працює разом із протоколом RTP, який використовується для передачі мультимедійних даних.

13. Виклики VoIP.

VoIP (Voice over IP, голосовий зв'язок за допомогою Інтернету) взагалі використовує два типи протоколів:

- сигнальні протоколи, такі, як SIP або H.323
- переносні протоколи, наприклад, RTP

14 Призначення. Відображення усіх IP-адрес призначення мережевих пакетів.

15. Графік потоків. Графіки потоків забезпечує послідовний аналіз TCP-з'єднань. Перші три строки містять оголошення TCP– з'єднання з послідовностями «SYN», «SYN ACK» та «ACK».

Блок визначення топології мережі

Блок визначення топології мережі:

– Блок використання відомостей із загальної системи моніторингу мережі, а не опитування пристрою додатково.

– Блок складання списку пристроїв у мережі, автоматично, ґрунтуючись на дані системи моніторингу.

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

- Блок побудови топології мережі, за станом на задану дату й відстеження змін у топології протягом часу.
- Блок автоматичного визначення рівнів ієрархії пристроїв у мережі, з виділенням периферійних, проміжних і центральних вузлів;
- Блок побудови топології мережі, незалежно від використовуваної системи моніторингу й програмно-апаратних платформ;
- Блок комбінувати показників, на основі яких визначаються зв'язки між пристроями, і при їхньому обчисленні виконувати перевірку на значимість із використанням статистичних критеріїв.

Блок виявлення аномального поведження трафіку

Блок виявлення аномального поведження трафіку:

- Блок визначення профілю поведження нормального трафіку.
- Блок заміни направлення трафіку.
- Блок усунення аномального трафіку.

При первісному розгортанні рішення по DDoS адміністратор створює профіль поведження нормального трафіку. Цей процес іменується навчанням. Компанія використовує додатки звичайним образом протягом 24 годин протягом одного тижня, і трафік додатка проходить через Детектор аномалій трафіку. У період навчання Детектор аномалій трафіку збирає базову інформацію для розуміння нормальної роботи мережі, куди входять:

- Інтенсивність пакетів для кожного типу пакетів, обмірювана як кількість пакетів у секунду (pps).
 - Співвідношення пакетів, наприклад, співвідношення пакетів SYN і пакетів FIN.
 - Кількість одночасних TCP-з'єднань, відкритих одним джерелом.
- Базова інформація збирається по кожній цільовій адресі хост-ПК, цільовій підмережі, вихідній адресі хост-ПК і вихідній підмережі.

Після закінчення періоду навчання Детектор аномалій трафіку переводиться в режим моніторингу, а Блок усунення аномального трафіку – у резервний режим готовності. Доти, поки немає атаки, що активно розвивається,

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

вхідний трафік з мережі Інтернет проходить через комутатор без якого-небудь втручання з боку Блоку усунення аномального трафіку. Копія вхідного трафіку посилає для аналізу на Детектор аномалій трафіку через зовнішній аналізатор протоколів (SPAN) або віртуальні списки ACL. Якщо Детектор аномалій трафіку виявляє аномальне в порівнянні з базовою інформацією поведження трафіку, починається процес усунення:

- Детектор аномалій трафіку направляє в Блок усунення аномального трафіку команду почати процес зміни напрямку.

- Блок усунення аномального трафіку відхиляє (“захоплює”) трафік, адресований на атакуєму IP-адресу, переадресуючи його на самого себе.

- Блок усунення аномального трафіку піддає трафік багатоступінчастому аналізу й застосовує контрзаходи для відділення благонадійних джерел від джерел атаки. Цей процес іменується очищенням або вичищенням.

- Блок усунення аномального трафіку скидає трафік атаки й пересилає благонадійний трафік назад на нормальний маршрут проходження трафіку до мети. Цей процес іменується ін'єкцією.

Детектор аномалій трафіку

Детектор аномалій трафіку – це пасивний пристрій моніторингу, що постійно виявляє ознаки, що вказують на присутність атаки DDoS, спрямованої проти захищеного місця призначення, також іменованого зоною. Це може бути сервер, інтерфейс міжмережного екрана або інтерфейс маршрутизатора. Детектор аномалій трафіку аналізує копії всього вхідного трафіку, адресуємого в захищені зони, через SPAN або відгалуження пасивної мережі. Цей аналіз включає зіставлення поточного поведження трафіку з базовими граничними параметрами, які також іменуються зональною політикою, для виявлення аномального поведження трафіку. Якщо аномальне поведження виявлене й виглядає як можлива атака, Детектор аномалій трафіку через позаполосну управлінську мережу Ethernet посилає в Блок усунення аномального трафіку сигнал про початок аналізу й усунення атаки.

Блок усунення аномального трафіку

Блок усунення аномального трафіку – це автономний пристрій аналізу й фільтрації трафіку. Починаючи прийом трафіку, адресованого в конкретну зону, що, очевидно, піддається атаці, Блок усунення аномального трафіку проводить точний аналіз цього трафіку. Якщо результати аналізу підтверджують, що трафік злочинний, Блок усунення аномального трафіку застосовує контрзаходи, наприклад, механізми анти-спуфінга й фільтрацію різного рівня (таблиця 3.1). Кінцевий результат полягає в тому, що трафік зі злочинних джерел скидається, а трафік із благонадійних джерел пересилається в передбачений пункт призначення.

Можливі варіанти зміни напрямку трафіку

Фахівці з ІТ можуть використовувати описані нижче варіанти зміни напрямку трафіку з його пересиланням з мережі, розташованого вище лежачого оператора зв'язку, на Блок усунення аномального трафіку. Цей процес також іменується “захватом” трафіку:

- Повідомлення прикордонного шлюзового протоколу (Border Gateway Protocol, BGP) із Блок усунення аномального трафіку на маршрутизатори, розташовані у вище лежачого оператора зв'язку, з інформацією про те, що трафік, адресований на захищену адресу призначення, буде переспрямований на Блок усунення аномального трафіку.

- Використання зовнішніх механізмів зміни напрямку трафіку, наприклад, маршрутизаторів віддаленого відновлення BGP.

- Повідомлення про ін'єкцію очищеного трафіку на маршруті (Route Health Injection, RHI) від Блок усунення аномального трафіку для процесу маршрутизації в Catalyst серії 6500 або в систему нагляду серії 7600. Ці повідомлення поміщають статичний маршрут у глобальну таблицю маршрутизації, у якій модуль Блок усунення аномального трафіку позначений як наступний вузол.



Рисунок 3.2 – Функціональна схема системи

Можливі варіанти ін'єкції трафіку

Ін'єкція трафіку – це процес, застосовуваний у Блок усунення аномального трафіку для пересилання очищеного благонадійного трафіку в точку призначення, що піддається атаці. Рішення підтримує різні варіанти ін'єкції трафіку. У варіанті 2-ого рівня топології, очищений трафік пересилається із Блок усунення аномального трафіку на статично-конфігуруєму наступну адресу заходу. Ця адреса перебуває на маршрутизаторі, розташованому нижче й з'єднаним з тої ж VLAN або підмережею, що й інтерфейс/VLAN ін'єкції трафіку. Ін'єкцію трафіку на 2-му рівні найпростіше конфігурувати, оскільки тут не потрібно вносити які-небудь істотні зміни в конфігурацію маршрутизатора, розташованого нижче.

Варіанти ін'єкції трафіку 3-го рівня:

- Маршрутизація й пересилання по VPN (VPN Routing and Forwarding, VRF).
- Маршрутизація на основі політики (Policy-Based Routing, PBR).
- Транкінг VLAN (VLAN Trunking).
- Інкапсуляція по загальній маршрутизації (GRE) або інкапсуляція IP у тунелі IP (IPIP).

Блок визначення виду атаки

Блок визначення виду атаки:

- Криптоджекінг.
- Ботнет.

У розділах вище були описані основні дії для протидії цим видам атак.

3.4 Розробка діаграми процесів

Розглянемо розроблену діаграму процесів яка зображена на рисунку 3.3. Основна будова діаграми процесів полягає у графічному представленні складу сукупностей даних, що характеризуються як співвідношення різних частин

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

кожної з сукупностей. Склад статистичної сукупності графічно може бути представлений як за допомогою абсолютних, так і відносних показників. Графічне зображення складу сукупності по абсолютними і відносними показниками сприяє проведенню більш глибокого аналізу і дозволяє проводити аналіз системи.



Рисунок 3.3 – Діаграма взаємодії процесів

Діаграма взаємодії процесів використовується для візуалізації процесів обробки даних (структурне проектування).

Для розробника вважається звичним спочатку креслити діаграму взаємодії процесів даних рівня контексту, завдяки чому буде показано взаємодію системи.

Ця діаграма в подальшому підлягає уточненню шляхом деталізації процесів та потоків даних з метою показати систему що розробляється.

При детальному її розгляді можна побачити як саме проходить взаємодія у розробленій системі. Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі.

Діаграми потоків даних містять чотири типи елементів:

- Зовнішні по відношенню до системи сутності.
- Потоки даних між елементами трьох попередніх типів.
- Процеси які являють собою трансформацію даних в рамках описуваної системи.
- Сховища даних (репозиторії).

Таким чином, розглянувши опис системи, структурну, функціональну схеми системи, та діаграму взаємодії процесів перейдемо до опису блок-схем основної програми, та підпрограм, які використовуються, для реалізації системи.

К6ПЗ_2026

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

4 РЕАЛІЗАЦІЯ ПРОЕКТУ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ПРАВИЛЬНІСТЬ ПРОЕКТНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

Розглянемо реалізацію бакалаврської дипломної роботи. Були проведені розрахунки і підібрані набори тестових даних для перевірки правильності реалізації проектних рішень.

Було створено блок-схеми роботи системи. Блок-схема це представлення задачі для її аналізу або розв'язування за допомогою спеціальних символів (геометричних образів), які позначають такі елементи, як операції, потік, дані тощо.

Блок вхідних та вихідних даних прийнято позначати паралелограмом, блок обчислень (обробки) даних – прямокутником, блок прийняття рішень – ромбом, еліпсом – початок та кінець алгоритму.

У інформаційних технологіях функціональна схема складається з функціональних блоків, які являють собою конструктивно відособлені частини (елементи або пристрої) автоматичних систем, які виконують певні функції. Функціональні блоки на схемі позначають прямокутниками, всередині яких надписують їх найменування відповідно до функцій, що виконуються. Зв'язки між функціональними блоками (внутрішні впливи) позначаються лініями зі стрілками, які вказують напрям впливів.

Функціональні схеми можуть виконуватися в укрупненому і розгорненому вигляді. У першому випадку на схемі зображають найважливіші блоки системи і зв'язки між ними.

У другому варіанті схема відображається більш детально, що полегшує її читання та ілюструє принцип роботи.

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

Основні елементи схем алгоритму це термінатор, процес, рішення, зумовлений процес (підпрограма), дані та з'єднувач. Термінатор це елемент відображає вхід із зовнішнього середовища або вихід з неї (найчастіше застосування – початок і кінець програми). Всередині фігури записується відповідна дія.

Процес це виконання однієї або кількох операцій, обробка даних будь-якого виду (зміна значення даних, форми подання, розташування). Всередині фігури записують безпосередньо самі операції. Рішення це показує рішення або функцію перемикального типу з одним входом і двома або більше альтернативними виходами, з яких тільки один може бути обраний після обчислення умов, визначених всередині цього елемента.

Вхід в елемент позначається лінією, що входить зазвичай у верхню вершину елемента. Якщо виходів два чи три то зазвичай кожен вихід позначається лінією, що виходить з решти вершин (бічних і нижній). Якщо виходів більше трьох, то їх слід показувати однією лінією, що виходить з вершини (частіше нижній) елемента, яка потім розгалужується.

Відповідні результати обчислень можуть записуватися поруч з лініями, що відображають ці шляхи. Зумовлений процес (підпрограма) це символ відображає виконання процесу, що складається з однієї або кількох операцій, що визначені в іншому місці програми (у підпрограмі, модулі). Всередині символу записується назва процесу і передані в нього дані. Дані це перетворення у форму, придатну для обробки (введення) або відображення результатів обробки (виведення). Цей символ не визначає носія даних (для вказівки типу носія даних використовуються специфічні символи). З'єднувач це символ відображає вихід в частину схеми і вхід з іншої частини цієї схеми.

Використовується для обриву лінії та продовження її в іншому місці (приклад: поділ блок-схеми, що не поміщається на листі). Відповідні сполучні символи повинні мати одне (при тому унікальне) позначення.

На рисунку 4.1 зображена основна блок-схема програми, на рисунку 4.2 зображено роботу підпрограми.

З яких видно що робота основної програми складається з початкових етапів ініціалізації ПЗ, перевірки наявності ресурсів системи, блоку початку основного циклу з чеканням запиту від користувача в якому відбувається виклик підсистеми та останньої стадії – перевірка поточного стану з завершенням роботи розробленого ПЗ.

При роботі підпрограми виконується основний функціонал системи з циклічними послідовностями, перевіркою поточного стану та поверненням в основну програму прапорів стану виконання.

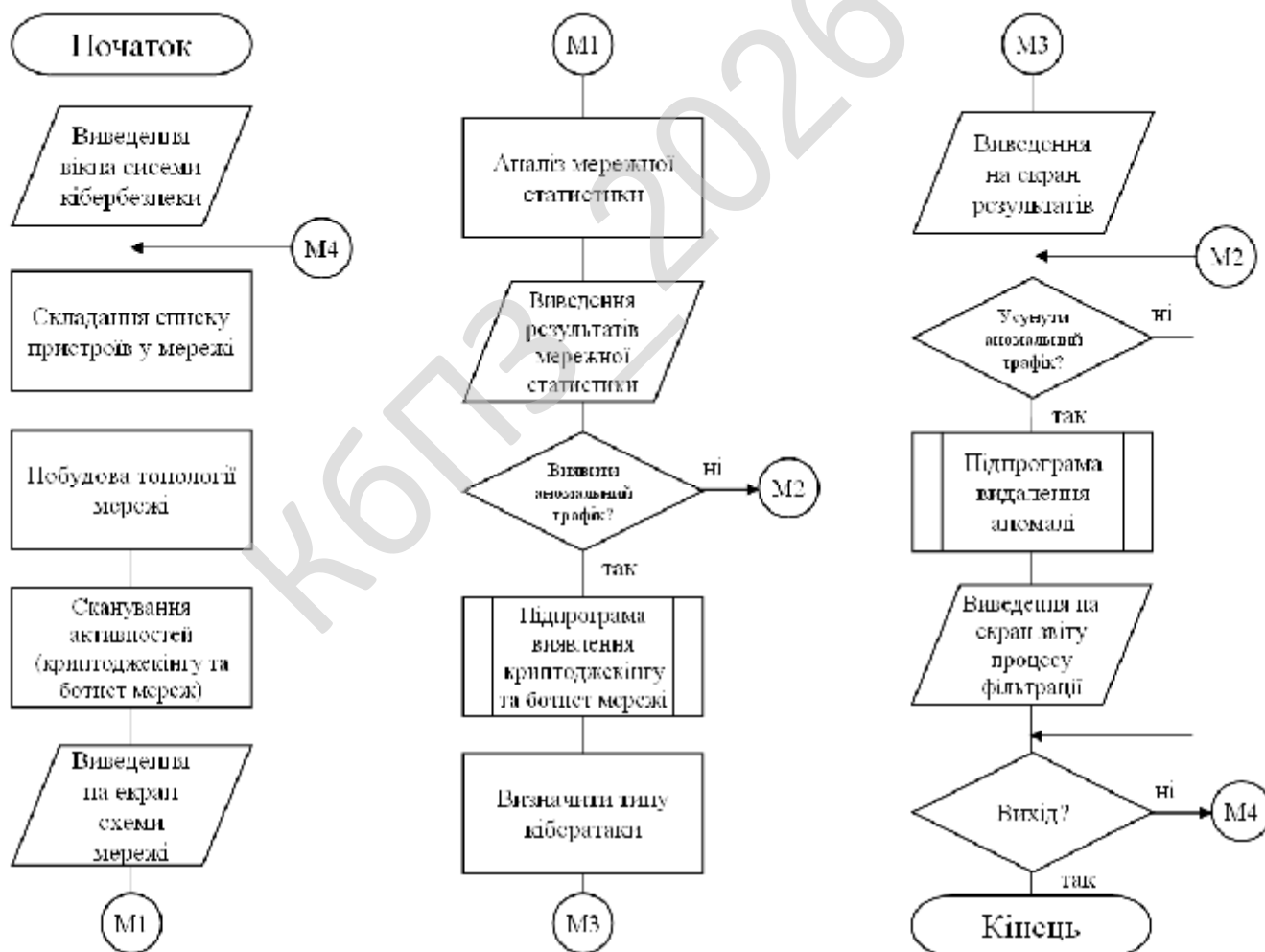


Рисунок 4.1 – Блок-схема основної програми

Оскільки наша програма буде мати графічний інтерфейс, те й писати її найкраще на Delphi. Версія середовища розробки значення не має.

Одержати інформацію про трафік можна декілька способами. Ми скористаємося функцією GetIfTable з бібліотеки IPHLPAPI.DLL (у модулях, що йдуть разом з Delphi, її опису немає).

Ця функція дозволяє з легкістю одержати інформацію про трафік й мережні інтерфейси.

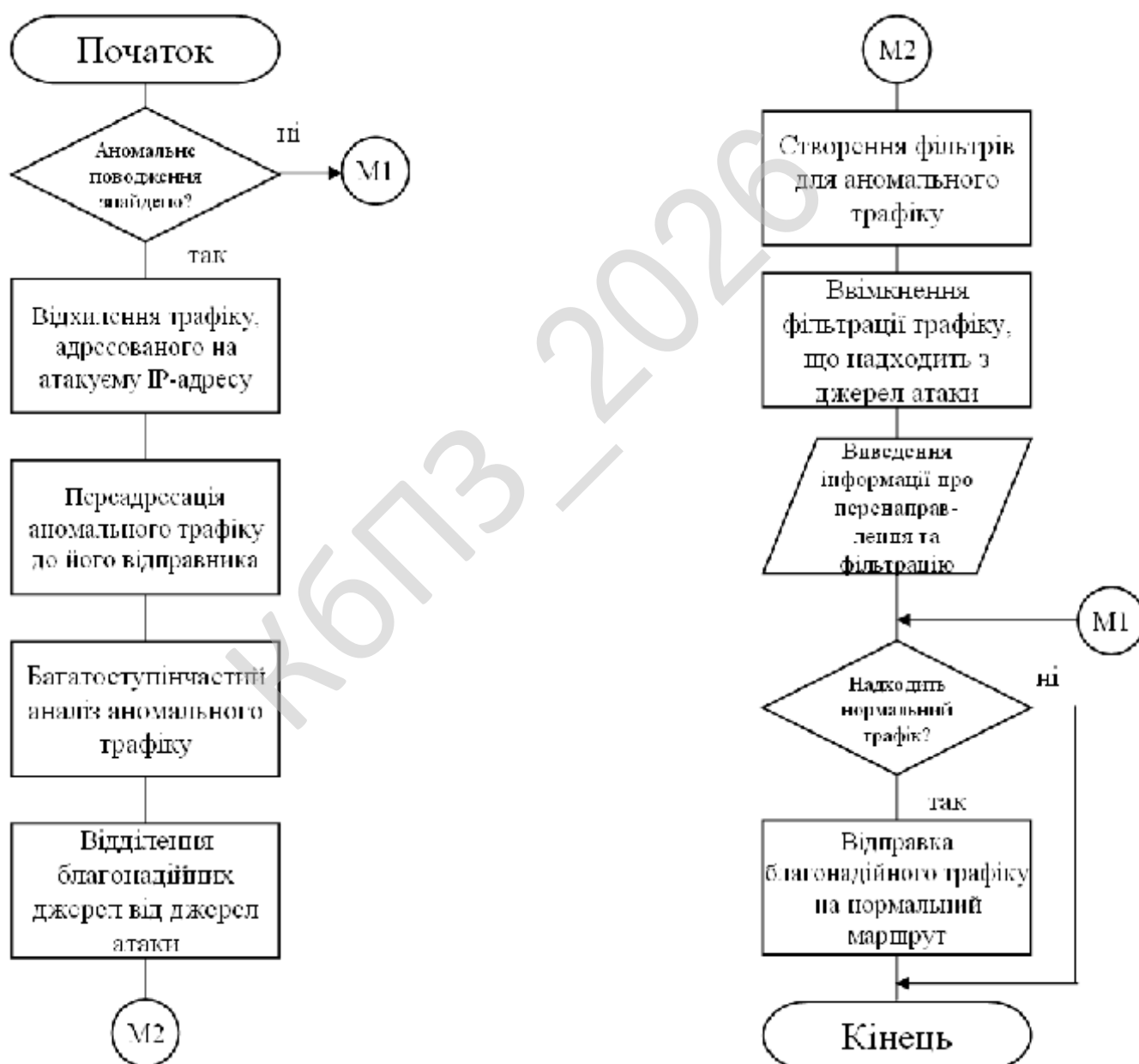


Рисунок 4.2 – Блок-схема роботи підпрограми

Працювати з нею дуже просто, а бібліотека, крім її, містить ще масу функцій для одержання всілякої інформації про роботу мережі й т.д. Ця бібліотека існує Windows, де вона відрізняється появою нових функцій (зміна структур).

Щоб не хвилюватися за працездатність своєї програми, раджу звернутися до MSDN і зробити в коді перевірку версії Windows

Основні функції системи

1. Збір інформації про процеси та навантаження, формування короткого звіту для адміністратора.
2. Виявлення ознак криптоджекінгу.
3. Виявлення ознак ботнет активності.
4. Розрахунок рівня ризику.
5. Збереження інцидентів у базі даних.

Опис системи

Програмне забезпечення системи кібербезпеки для протидії криптоджекінгу та ботнетам працює як локальний модуль моніторингу комп'ютера або серверного вузла. Система аналізує активність процесів, рівень завантаження процесора, мережеві з'єднання, підозрілі домени, частоту звернень до зовнішніх адрес та ознаки прихованого майнінгу.

Основна мета системи полягає у виявленні процесів, які споживають надмірні ресурси, звертаються до підозрілих вузлів або поведуться як частина ботнету. Після виявлення загрози система формує запис у журналі, присвоює рівень ризику та передає подію адміністратору.

Архітектура системи складається з модуля збору даних, модуля аналізу процесів, модуля аналізу мережевої активності, модуля оцінювання ризику, бази подій SQLite та модуля звітності.

Розглянемо фрагмент вихідного коду з описом основних функцій. Цей код демонструє базову логіку системи. Він приймає події моніторингу, аналізує ознаки криптоджекінгу та ботнет активності, визначає рівень ризику, зберігає інциденти у SQLite та формує звіт для адміністратора

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

```

import sqlite3
import time
from datetime import datetime

DB_NAME = "cyber_guard.db"
class CyberGuardDatabase:
    def __init__(self, db_name=DB_NAME):
        self.db_name = db_name
        self.create_tables()
    def create_tables(self):
        connection = sqlite3.connect(self.db_name)
        cursor = connection.cursor()
        cursor.execute("""
            CREATE TABLE IF NOT EXISTS incidents (
                id INTEGER PRIMARY KEY AUTOINCREMENT,
                created_at TEXT NOT NULL,
                process_name TEXT NOT NULL,
                cpu_usage REAL NOT NULL,
                network_target TEXT NOT NULL,
                threat_type TEXT NOT NULL,
                risk_level TEXT NOT NULL,
                description TEXT NOT NULL
            )
        """)
        connection.commit()
        connection.close()
    def save_incident(self, incident):
        connection = sqlite3.connect(self.db_name)
        cursor = connection.cursor()
        cursor.execute("""
            INSERT INTO incidents (
                created_at,
                process_name,
                cpu_usage,
                network_target,
                threat_type,
                risk_level,
                description
            )
            VALUES (?, ?, ?, ?, ?, ?, ?)
        """, (
            incident["created_at"],

```

```

        incident["process_name"],
        incident["cpu_usage"],
        incident["network_target"],
        incident["threat_type"],
        incident["risk_level"],
        incident["description"]
    ))
    connection.commit()
    connection.close()

def get_report(self):
    connection = sqlite3.connect(self.db_name)
    cursor = connection.cursor()
    cursor.execute("""
SELECT created_at, process_name, threat_type, risk_level, description
    FROM incidents
    ORDER BY id DESC
    """)
    rows = cursor.fetchall()
    connection.close()
    return rows

class ThreatAnalyzer:
    def __init__(self):
        self.suspicious_domains = [
            "pool",
            "miner",
            "xmr",
            "cryptonight",
            "botnet",
            "command"
        ]

    def analyze_event(self, event):
        score = 0
        threat_type = "невизначена активність"
        if event["cpu_usage"] >= 75:
            score += 40
            threat_type = "можливий криптоджекінг"
        if self.has_suspicious_network_target(event["network_target"]):
            score += 35
            threat_type = "можлива ботнет активність"

```

```

        if event["connection_count"] >= 25:
            score += 25
            threat_type = "можлива ботнет активність"
        risk_level = self.define_risk_level(score)
        return {
            "created_at": datetime.now().strftime("%Y-%m-%d %H:%M:%S"),
            "process_name": event["process_name"],
            "cpu_usage": event["cpu_usage"],
            "network_target": event["network_target"],
            "threat_type": threat_type,
            "risk_level": risk_level,
            "description": self.create_description(event, threat_type, risk_level)
        }

    def has_suspicious_network_target(self, target):
        target_lower = target.lower()
        for marker in self.suspicious_domains:
            if marker in target_lower:
                return True
        return False

    def define_risk_level(self, score):
        if score >= 70:
            return "високий"
        if score >= 40:
            return "середній"
        return "низький"

    def create_description(self, event, threat_type, risk_level):
        return (
            f"Процес {event['process_name']} має навантаження CPU "
            f"{event['cpu_usage']} відсотків та звертається до "
            f"{event['network_target']}. Тип загрози {threat_type}. "
            f"Рівень ризику {risk_level}."
        )

class CyberGuardSystem:
    def __init__(self):
        self.database = CyberGuardDatabase()
        self.analyzer = ThreatAnalyzer()

    def scan_events(self, events):

```

```

        for event in events:
            incident = self.analyzer.analyze_event(event)
            if incident["risk_level"] in ["середній", "високий"]:
                self.database.save_incident(incident)
                print("Виявлено інцидент")
                print(incident["description"])

    def show_report(self):
        rows = self.database.get_report()
        print("\nЗвіт системи кібербезпеки")
        for row in rows:
            print(f"{row[0]} | {row[1]} | {row[2]} | {row[3]} | {row[4]}")

#Демонстраційні події імітують роботу модуля моніторингу процесів та мережі
sample_events = [
    {
        "process_name": "browser.exe",
        "cpu_usage": 18.5,
        "network_target": "update.service.local",
        "connection_count": 4
    },
    {
        "process_name": "hidden_worker.exe",
        "cpu_usage": 91.2,
        "network_target": "xmr.pool.example",
        "connection_count": 12
    },
    {
        "process_name": "system_task.exe",
        "cpu_usage": 48.0,
        "network_target": "command.node.example",
        "connection_count": 31
    }
]

#Запуск системи виконує аналіз подій, збереження інцидентів та формування звіту
if __name__ == "__main__":
    system = CyberGuardSystem()
    system.scan_events(sample_events)
    time.sleep(1)
    system.show_report()

```

Було використано підходи з використанням UML, це уніфікована мова моделювання, використовується у парадигмі об'єктно-орієнтованого програмування. Є невід'ємною частиною уніфікованого процесу розробки програмного забезпечення. UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, називаної UML-моделлю. UML був створений для визначення, візуалізації, проектування й документування в основному програмних систем. UML не є мовою програмування, але в засобах виконання UML-моделей як інтерпретованого коду можлива кодогенерація.

Архітектура клієнт-сервер є одним із архітектурних шаблонів програмного забезпечення та є домінуючою концепцією у створенні розподілених мережних програм і передбачає взаємодію та обмін даними між ними. Вона передбачає такі основні компоненти:

- набір серверів, які надають інформацію або інші послуги програмам, які звертаються до них;
- набір клієнтів, які використовують сервіси, що надаються серверами;
- мережа, яка забезпечує взаємодію між клієнтами та серверами.

Сервери є незалежними один від одного. Клієнти також функціонують паралельно і незалежно один від одного. Немає жорсткої прив'язки клієнтів до серверів. Більш ніж типовою є ситуація, коли один сервер одночасно обробляє запити від різних клієнтів; з іншого боку, клієнт може звертатися то до одного сервера, то до іншого. Клієнти мають знати про доступні сервери, але можуть не мати жодного уявлення про існування інших клієнтів.

Дуже важливо ясно уявляти, хто або що розглядається як «клієнт». Можна говорити про клієнтський комп'ютер, з якого відбувається звернення до інших комп'ютерів. Можна говорити про клієнтське та серверне програмне забезпечення. Нарешті, можна говорити про людей, які бажають за допомогою відповідного програмного та апаратного забезпечення отримати доступ до тієї чи іншої інформації.

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

Загальноприйнятим є положення, що клієнти та сервери – це перш за все програмні модулі. Найчастіше вони знаходяться на різних комп'ютерах, але бувають ситуації, коли обидві програми – і клієнтська, і серверна, фізично розміщуються на одній машині; в такій ситуації сервер часто називається локальним.

Модель клієнт-серверної взаємодії визначається перш за все розподілом обов'язків між клієнтом та сервером. Логічно можна відокремити три рівні операцій:

- рівень представлення даних, який по суті являє собою інтерфейс користувача і відповідає за представлення даних користувачеві і введення від нього керуючих команд;
- прикладний рівень, який реалізує основну логіку ПЗ і на якому здійснюється необхідна обробка інформації;
- рівень управління даними, який забезпечує зберігання даних та доступ до них.

Дворівнева клієнт-серверна архітектура передбачає взаємодію двох програмних модулів – клієнтського та серверного. В залежності від того, як між ними розподіляються наведені вище функції, розрізняють:

- модель тонкого клієнта, в рамках якої вся логіка ПЗ та управління даними зосереджена на сервері. Клієнтська програма забезпечує тільки функції рівня представлення;
- модель товстого клієнта, в якій сервер тільки керує даними, а обробка інформації та інтерфейс користувача зосереджені на стороні клієнта. Товстими клієнтами часто також називають пристрої з обмеженою потужністю: кишенькові комп'ютери, мобільні телефони та ін.

Типовим прикладом клієнт-серверної взаємодії є WWW. Існує величезна кількість веб-серверів, на яких розміщується та чи інша інформація. У найпростішому випадку ця інформація являє собою набір веб-сторінок, які можуть зберігатися на сервері у вигляді файлів, розмічених за допомогою мови

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

розмітки HTML. Але ситуація, як правило, є складнішою; значна частина веб-ресурсів на сучасному етапі є динамічними, тобто вони не існують в заздалегідь підготовленому вигляді, а створюються безпосередньо в процесі обробки запиту від користувача.

Для того, щоб людина, яка працює в Інтернеті, могла переглянути ту чи іншу сторінку, на її комп'ютері повинно бути встановлено відповідне програмне забезпечення. Програми для перегляду веб-сторінок називаються браузером.

Але, крім браузерів, до серверів можуть звертатися і інші клієнти, а саме – автономні програми. Вони можуть передбачати взаємодію з людиною, а можуть працювати в цілком автоматичному режимі. Типовим класом таких програм є роботи, призначені для автоматичного перегляду веб-ресурсів. Зокрема, роботи є важливим елементом пошукових систем і використовуються ними для перегляду сторінок і збору інформації про них.

Для запиту до веб-сервера клієнтська програма повинна задати місцезнаходження комп'ютера, на якому розміщується серверна програма, назву потрібного документа і, можливо, інші дані, які специфікують запит. Мережа забезпечує знаходження сервера і передачу йому клієнтського запиту. Серверні програми обробляють цей запит, відповідь пересилається по мережі клієнтові.

Трирівнева клієнт-серверна архітектура, яка почала розвиватися з середини 90-х років, передбачає відділення прикладного рівня від управління даними. Відокремлюється окремий програмний рівень, на якому зосереджується прикладна логіка ПЗ. Програми проміжного рівня можуть функціонувати під управлінням спеціальних серверів ПЗ, але запуск таких програм може здійснюватися і під управлінням звичайного веб-сервера. Нарешті, управління даними здійснюється сервером даних.

Для роботи з системою користувач використовує стандартне програмне забезпечення –звичайний браузер. Це позбавляє його необхідності завантажувати та інсталиувати спеціальні програми (хоча інколи така необхідність все-таки виникає).

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

Але користувачеві слід надати в розпорядженні інтерфейс, який дозволяв би йому взаємодіяти з системою і формувати запити до неї. Форми, що визначають цей інтерфейс, розміщуються на веб-сторінках та завантажуються разом з ними.

Веб-оглядач формує запит та пересилає його до сервера, який здійснює обробку. При необхідності сервер викликає серверні програмні модулі, які забезпечують обробку запиту і в разі потреби звертаються до сервера даних. Сервер даних здійснює операції з даними, що зберігаються в системі та складають її інформаційну основу. Зокрема, він може здійснити вибірку з інформаційної бази відповідно до запиту та передати її модулю проміжного рівня для подальшої обробки. Дані, з якими працює сервер даних, найчастіше організовані як реляційна база даних.

Найчастіше веб-сервер і серверні модулі проміжного рівня розміщуються на одному комп'ютері, хоч і являють собою окремі і логічно незалежні програмні модулі.

На сучасному етапі для програмування модулів проміжного рівня використовується мова серверних сценаріїв PHP, а для управління даними – СУБД MySQL. Таким чином, зв'язку PHP-MySQL слід розглядати як стандартний інструмент для створення порівняно простих інтерактивних веб-сайтів та систем електронної комерції; близько 90% комерційних систем сьогодні створюється саме на цій основі. Водночас як засоби управління даними, так і middleware-засоби можуть бути найрізноманітнішими. Так, для створення серверних програм, крім PHP, широко застосовуються Java, Perl, Python, Delphi.

Взагалі, технології створення розподілених, зокрема веб-програм, стрімко розвиваються. Слід згадати про технології EJB (Enterprise Java Beans), CORBA, а також про .NET – порівняно нову ініціативу компанії Microsoft. Для зберігання даних та їх передачі часто використовується так звана розширювана мова розмітки XML (Extensible Markup Language).

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

Розглянемо формат що використовується – JSON (JavaScript Object Notation, укр. запис об'єктів JavaScript, вимовляється джейсон) – це текстовий формат обміну даними між комп'ютерами.

JSON базується на тексті, може бути прочитаним людиною. Формат дозволяє описувати об'єкти та інші структури даних. Цей формат головним чином використовується для передачі структурованої інформації через мережу (завдяки процесу, що називають серіалізацією). Розробив і популяризував формат Дуглас Крокфорд.

JSON знайшов своє головне призначення у написанні веб-програм, а саме при використанні технології AJAX. JSON, що використовується в AJAX, виступає як заміна XML (використовується в AJAX) під час асинхронної передачі структурованої інформації між клієнтом та сервером. При цьому перевагою JSON перед XML є те, що він дозволяє складні структури в атрибутах, займає менше місця і прямо інтерпретується за допомогою JavaScript в об'єкти.

JSON з'явився через необхідність обміну даними з сервером у реальному часі без використання плагінів для браузерів, flash-додатків або Java-апплетів, які використовувались скрізь на початку 2000-х років. Дуглас Крокфорд був тим, хто активно просував новий на той час формат. Він з колегами хотів створити технологію, яка використовувала б можливості звичайного браузера давала б веб-розробникам можливість створювати веб-додатки із постійним двостороннім зв'язком із веб-сервером. JSON вперше був використаний в проєкті в Communities.com для Cartoon Network, він дозволяв обмінюватись повідомленнями і одночасно маніпулювати DHTML-елементами. Веб-сайт JSON.org було запущено 2002 року. У грудні 2005-го року Yahoo! почав переводити деякі зі своїх веб-сервісів на роботу з JSON. Google взялася до роботи з технологією у своєму веб-протоколі GData у грудні 2006-го року.

Використання

За рахунок своєї лаконічності в порівнянні з XML, формат JSON може бути більш придатним для серіалізації складних структур.

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

Якщо говорити про веб-застосунки, в такому ключі він доречний в задачах обміну даними як між браузером і сервером (AJAX), так і між самими серверами (програмні HTTP-інтерфейси).

Формат JSON так само добре підходить для зберігання складних динамічних структур в реляційних базах даних або файловому кеші.

Приклад використання JSON (JavaScript)

```
const ajaxData = '{"name": "wiki", "fname": "pedia", "rates": [1,4,5,6]}'
const ajaxObj = JSON.parse(ajaxData)
console.log(`${ajaxObj.name}_${ajaxObj.rates[2]}`) /* Виведе "wiki_5"*/
```

Синтаксис

JSON будується на двох структурах:

1. Набір пар назва/значення. У різних мовах це реалізовано як об'єкт, запис, структура, словник, хеш-таблиця, список з ключем або асоціативним масивом.

2. Впорядкований список значень. У багатьох мовах це реалізовано як масив, вектор, список, або послідовність.

Це універсальні структури даних. Теоретично всі сучасні мови програмування підтримують їх у тій чи іншій формі. Оскільки JSON використовується для обміну даними між різними мовами програмування, то є сенс будувати його на цих структурах.

У JSON використовуються такі їхні форми:

1. Об'єкт – це послідовність пар назва/значення. Об'єкт починається з символу { і закінчується символом } . Кожне значення слідує за : і пари назва/значення відділяються комами.

2. Масив – це послідовність значень. Масив починається символом [і закінчується символом]. Значення відділяються комами.

3. Значення може бути рядком в подвійних лапках, або числом, або логічними true чи false, або null, або об'єктом, або масивом. Ці структури можуть бути вкладені одна в одну.

4. Рядок – це послідовність з нуля або більше символів юнікода, обмежена подвійними лапками, з використанням escape-послідовностей, що починаються зі зворотної косої риски \. Символи представляються простим рядком. Тип Рядок (String) дуже схожий на String в мовах C і Java. Число теж дуже схоже на C- або Java-число, за винятком того, що вісімкові та шістнадцяткові формати не використовуються. Пропуски можуть бути вставлені між будь-якими двома лексемами.

Наступний приклад показує JSON представлення об'єкта, що описує людину. У об'єкті є рядкові поля імені і прізвища, об'єкт, що описує адресу, і масив, що містить список телефонів.

```
{
  "firstName": "Іван",
  "lastName": "Іванович",
  "address": {
    "streetAddress": "вул. Гру 14, кв.101",
    "city": "Київ",
    "postalCode": 21000
  },
  "phoneNumbers": [
    "044 123-1234",
    "050 123-4567"
  ]
}
```

Використання JSON в AJAX

Наступний фрагмент коду JavaScript показує, як клієнт може використати XMLHttpRequest для запиту об'єктів в форматі JSON з серверу. (Серверна частина коду пропущена, вона просто повертає на запит URL рядок у JSON форматі).

Треба відзначити, що тут використання XMLHttpRequest не є кросс-браузерним (за деталями звертайтеся на сторінку XMLHttpRequest), і код зазнаватиме незначних модифікацій в різних версіях оглядачів Internet Explorer, Opera, Safari або Mozilla. Використання запиту XMLHttpRequest обмежено правилом одного джерела (same origin policy): URL, що відповідає на запит, має посилатися на той же сайт, що обслуговує сторінку, що ініціювала запит.

Оглядачі можуть також використовувати тег `<iframe>` для асинхронного запиту JSON-даних в кросс-браузерному варіанті, або використати просте перенаправлення `<form action="url_to_cgi_script" target="name_of_hidden_iframe">`. Такий підхід був поширений до приходу популярного нині запиту XMLHttpRequest.

Динамічний тег `<script>` також можна використати для підвантаження JSON-даних. Ця техніка можлива, щоб обійти надто суворе правило одного джерела, але вона не є безпечною. Запит XMLHttpRequest пропонується, як безпечніша альтернатива.

Питання безпеки

Хоча JSON призначений для передачі даних в серіалізованому вигляді, його синтаксис відповідає синтаксису JavaScript і це створює низку проблем безпеки. Часто для обробки даних, отриманих від зовнішнього джерела у форматі JSON, до них застосовується функція `eval()` без якої-небудь попередньої перевірки.

JavaScript eval()

Оскільки JSON представляється синтаксично правильним фрагментом коду JavaScript, природним способом розбору JSON-даних в JavaScript-програмі є використання вбудованої в JavaScript функції `eval()`, яка призначена для обчислення JavaScript-виразів. При цьому підході відпадає необхідність у використанні додаткових парсерів.

Техніка використання `eval()` робить систему вразливою, якщо джерело JSON-даних, що використовуються, не відноситься до надійних. Такими даними може виступати шкідливий JavaScript-код для атак за допомогою ін'єкції коду. За допомогою цієї вразливості можливо здійснювати крадіжку даних, підробку автентифікації. Проте, вразливість можна усунути за рахунок використання додаткових засобів перевірки даних на коректність. Наприклад, до виконання `eval()` отримані від зовнішнього джерела дані можуть перевірятися за допомогою

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

використовувати для визначення цього. Виключне використання HTTP cookie використовується підтримкою міжсайтових запитів.

Розширення, JSONP

JSONP або «JSON з підкладкою» є розширенням JSON, коли назва функції зворотного виклику вказується як вхідний аргумент. Спочатку ідея була запропонована в блозі MacPython в 2005 році, і в наш час використовується багатьма Web 2.0 застосунками, такими, як Dojo Toolkit Applications, Google Toolkit Applications і zanox Web Services.

Подальші розширення цього протоколу були запропоновані з урахуванням введення додаткових аргументів, як, наприклад, у разі JSONPP за підтримки S3DB вебсервісів.

Оскільки JSONP використовує скрипт-теги, виклики по суті відкриті світові. З цієї причини JSONP може бути недоречними для зберігання конфіденційних даних.

Включення скриптових тегів від віддалених сайтів дозволяє їм передати будь-який контент на сайті. Якщо віддалений сайт має вразливості, які дозволяють виконати ін'єкції JavaScript, то початковий сайт також може бути ними зачеплений.

Розширення, BSON

BSON – це бінарна форма представлення простих структур даних і асоціативних масивів (які називають об'єктами або документами). Назва «BSON» заснована на визначенні JSON і неофіційно означає «Binary JSON» (бінарний JSON).

JSON Reference

Стандарт JSON не описує посилання на інші об'єкти або частини, але існує чернетка стандарту IETF для посилань на об'єкти на основі JSON. Посилання дозволяють здійснювати трансклюзію – вставляти одні документи в інші.

JSON Reference – це JSON-об'єкт з ключем "\$ref" (всі інші ключі ігноруються) і значенням стрічкового типу що містить URI, наприклад:

```
{ "$ref": "http://example.com/example.json#/foo/bar" }
```

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

Якщо URI містить ідентифікатор фрагмента (в прикладі вище "/foo/bar"), він інтерпретується як JSON Pointer.

Модуль `dojox.json.ref` в Dojo toolkit, забезпечує підтримку декількох форм JSON Reference.

4.2 Захист розробленого програмного забезпечення

Дані які використовуються у даній роботі захищаються алгоритмом ДСТУ 9041:2020. Його повна назва: ДСТУ 9041:2020. Інформаційні технології. Криптографічний захист інформації. Алгоритм шифрування коротких повідомлень, що ґрунтується на скручених еліптичних кривих Едвардса.

Цей алгоритм призначений для шифрування коротких (до 616 біт) повідомлень для будь-яких алгоритмів шифрування, в тому числі визначених національними стандартами України.

Як і стандарт цифрового підпису ДСТУ 4145:2002, новий алгоритм використовує криптографічні перетворення у групі точок еліптичних кривих, використовуючи замість кривих у формі Вейерштрасса найновітніші розробки у галузі еліптичної криптографії – криві у формі Едвардса. Це дає суттєві переваги у швидкодії більш ніж у 3 рази. Новий стандарт розроблений з урахуванням усіх найсучасніших вимог до стійкості криптографічних алгоритмів. Так, нижня межа стійкості криптосистем у цьому стандарті дорівнює 2127 (≈ 1042) (це більш ніж у півтора рази вище, ніж у ДСТУ 4145) і можуть бути обрані інші рівні, такі як 2255 (≈ 1085), 2383 (≈ 10127) та 2767 (≈ 10255); крім того, строго обґрунтована його стійкість як до атак на відновлення відкритого тексту, так і до розрізняючих атак.

Проект алгоритму шифрування, що ліг в основу цього стандарту, пройшов апробацію як в Україні, так і за її межами (Центрально-Європейська конференція з криптографії (червень 2020 року) – форум ведучих криптологів з усього світу). Стандарт ДСТУ-9041 узгоджений з усіма діючими в Україні національними

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

стандартами. Новиною стандарту є його сфера застосування – інкапсуляція ключів, найсучасніший математичний апарат, а також новий алгоритм генерації псевдовипадкових послідовностей, який, на відміну від аналогічного алгоритму генерації з ДСТУ 4145, використовує виключно національні криптографічні алгоритми національних стандартів та не містить посилань на відповідні пост-радянські стандарти, термін дії яких вже практично вичерпався.

Новий стандарт не належить до так званих постквантових стандартів. Але його стійкість буде під загрозою лише тоді, коли з'являться квантові комп'ютери з 700 і більше кубітами (на даний час кількість "робочих" кубітів, які вдалося створити, – близько 50). Його перевагою перед постквантовими алгоритмами є відносно невелика довжина ключа (у десятки або навіть у сотні разів менша, ніж у постквантових алгоритмах).

КБПЗ-2026

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ КІБЕРБЕЗПЕКИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено розроблене у бакалаврської дипломної роботі система кібербезпеки для протидії криптоджекінгу та ботнетам. З рисунку можна побачити що інтерфейс головного вікна розподілено на наступні функціональні розділи: Функціональних кнопок ПЗ; Навігаційного меню яке викликається натисканням правої клавіші маніпулятора миші; Верхнього меню; Розділу обрання групи; Розділу виведення результату роботи системи.

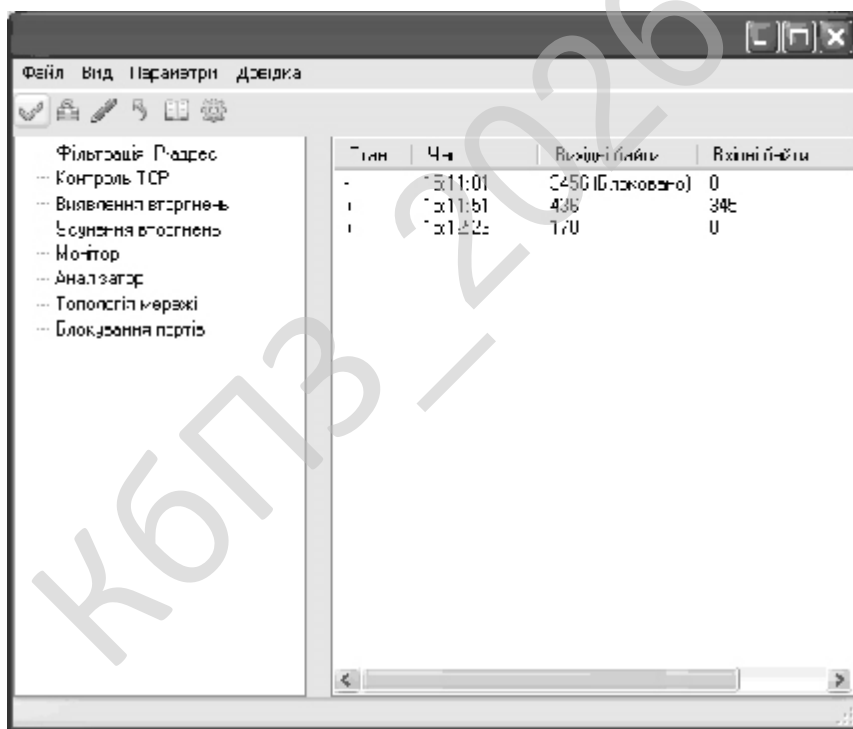


Рисунок 5.1 – Головне вікно ПЗ

Вплив деструктивних програм залишаються досить високим, особливо коли злочинці використовують їх у своїх проєктованих атаках. Перед проведенням таких цілеспрямованих атак злочинці проводять ретельну розвідку організації, що сприяє підвищенню ймовірності успіху. Проникнувши в мережу,

що атакують обережно переміщаються по її периферії й тільки потім приступають до найбільш руйнівної частини запланованої атаки. Прикладами таких проєктованих атак з деструктивним навантаженням, що забезпечує максимально руйнівний вплив, є шкідлива програма Olympic Destroyer і здирник, що з'явився пізніше, SamSam.

Об'єднання проектних завдань із деструктивними тенденціями приводить до створення шкідливих програм, поява яких не може не турбувати. Ключовим завданням нині вважаються виявлення й оперативна реакція. Але з обліком того, що на передній план знову виходять хробаки й деструктивні шкідливі програми, прийшов час задуматися про посилення захисту.

Прагнучи підвищити ймовірність досягнення успіху, кіберзлочинці продовжують розвивати свої методи здійснення атак, починаючи від криптоджекінгу й закінчуючи ботнетами й шкідливими програмами. Факти, наведені в цьому звіті, свідчать про те, що чим більше зможемо побачити, тим простіше буде захиститися. Вони говорять про те, що не можна заспокоюватися на досягнутому й забувати про основи, зокрема про кібергігієну. І в нинішньому динамічно мінливому середовищі групи ІТ-безпеки мають набагато більше шансів захиститися від новітніх кіберсхем, коли знають, що потрібно шукати.

Розроблена програма має дуже простий і зрозумілий інтерфейс з користувачем. Кожен, хто в достатньому обсязі володіє операційним середовищем Windows без особливих складностей освоїть і цю програму, оскільки її інтерфейс інтуїтивно зрозумілий. Якщо програма не видала ніяких помилок, і працює, то можна використовувати, інакше слід слідувати інструкціям, які пропонує програма.

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення.

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

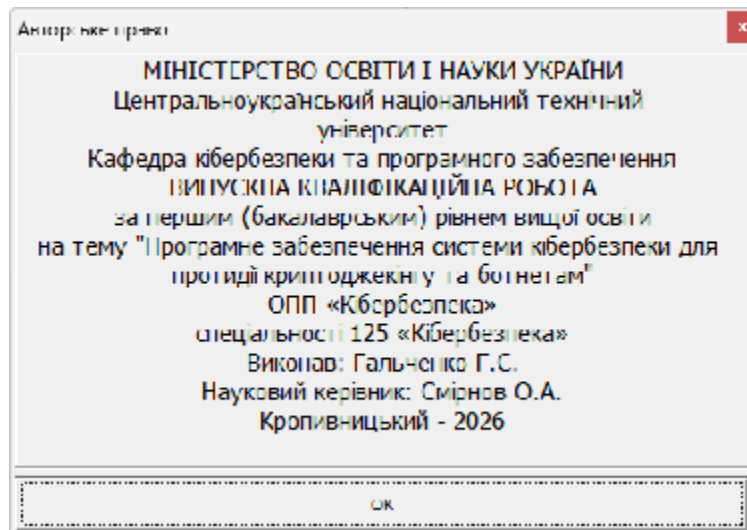


Рисунок 5.2 – Авторське право

Розглянемо процес впровадження програмного забезпечення, це процес налаштування програмного забезпечення під певні умови використання, а також навчання користувачів роботі з програмним продуктом. Впровадження програмного забезпечення це усі дії, що роблять розроблену програмну систему готовою до використання. Даний процес є частинною життєвого циклу програмного забезпечення.

Загалом процес розгортання складається з кількох взаємопов'язаних дій із можливими переходами між ними. Ця активність може відбуватися як з боку виробника так і з боку споживача. Оскільки кожна програмна система є унікальною, то усі процеси та процедури під час розгортання важко передбачити. Тому, "розгортання" можна трактувати як загальний процес відповідно до певних вимог та характеристик. Розгортання може здійснюватись програмістом і в процесі розробки програмного забезпечення.

До діяльностей пов'язаних із розгортанням програмного забезпечення відносять:

- Випуск.
- Встановлення та активація.
- Деактивація.

- Адаптація.
- Обновлення.
- Вмонтування.
- Відстежування версій.
- Видалення.
- Вилучення з обігу.

При впровадженні програмного забезпечення потрібно урахувати наступні дії:

– Виділення критичних, з точки зору загального результату, процедур в діяльності організації. Коли набір таких процедур визначений, необхідно в першу чергу використовувати ІТ рішення для автоматизації операцій усередині саме цих процедур. Таким чином, розроблене ІТ рішення автоматично стає життєво важливим і затребуваним для організації, а також буде забезпечена публічність процесу впровадження;

– Розширення нормативної бази організації шляхом включення до неї регламентів, що описують порядок виконання процедур автоматизованих процесів. В іншому випадку є небезпека виникнення неузгодженості між автоматизованими процедурами та іншими процесами організації.

– Виконання робіт з загальної стандартизації існуючої діяльності організації, коли виділяються кращі практики виконання процедур і включаються в ІТ рішення за принципом найбільшої корисності для більшості учасників. Відсоток таких процедур щодо загального обсягу автоматизації може бути невеликий, але це надає процесу побудови рішення вагу в організації за рахунок збільшення його необхідності.

Під час роботи над програмою було проведено тестування програмного забезпечення, тобто технічне дослідження, призначене для виявлення інформації про якість продукту відносно контексту, в якому воно має використовуватись.

Тестування включає як процес пошуку помилок або інших дефектів, так і випробування програмних складових з метою їх оцінки.

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

Проводилась оцінка:

- відповідності поставленим вимогам;
- правильна відповідь для усіх можливих вхідних даних;
- виконання функцій за прийнятний час;
- практичність;
- сумісність з ОС та стороннім ПЗ.

Оскільки число можливих тестів для програмних компонент практично нескінченне, тому стратегія тестування полягала в тому, щоб провести всі можливі тести з урахуванням наявного часу та ресурсів.

Як результат ПЗ тестувалось стандартним виконанням програми з метою виявлення помилок або інших дефектів. Проводилось тестування чорної скриньки. Основне місце програми тестів «чорної скриньки» – інтерфейс ПЗ. Відомі: функції програми. Досліджується: робота кожної функції на всій області визначення.

Ці тести демонструють:

- Як виконуються функції програми.
- Як приймаються вихідні дані.
- Як виробляються результати.
- Як зберігається цілісність зовнішньої інформації.

При тестуванні «чорної скриньки» розглядаються системні характеристики програм, ігнорується їхня внутрішня логічна структура. Вичерпне тестування, як правило, неможливе.

Наприклад, якщо в програмі 10 вхідних величин і кожна приймає по 10 значень, то кількість тестових варіантів становитиме 10^{10} . Тестування «чорної скриньки» не реагує на багато особливостей програмних помилок.

Тестування «чорної скриньки» (функціональне тестування) дозволяє отримати комбінації вхідних даних, які забезпечують повну перевірку всіх функціональних вимог до програми.

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

Програмний виріб тут розглядається як «чорна скринька», чію поведінку можна визначити тільки дослідженням його входів та відповідних виходів. При такому підході бажано мати:

- Набір, утворений такими входними даними, які призводять до аномалій у поведінці програми (назвемо його ІТс).

- Набір, утворений такими входними даними, які демонструють дефекти програми (назвемо його ОТ).

Будь-який спосіб тестування «чорної скриньки» повинен:

- Виявити такі входні дані, які з високою ймовірністю належать набору ІТс;
- Сформулювати такі очікувані результати, які з високою ймовірністю є елементами набору ОТ.

Принцип «чорної скриньки» не альтернативний принципу «білої скриньки». Скоріше це доповнює підхід, який виявляє інший клас помилок.

Тестування «чорної скриньки» забезпечує пошук наступних категорій помилок:

- Некоректних чи відсутніх функцій;
- Помилки інтерфейсу;
- Помилки у зовнішніх структурах даних або в доступі до зовнішньої бази даних;
- Помилки характеристик (необхідна ємність пам'яті і т.д.);
- Помилки ініціалізації та завершення.

Обрано умови розповсюдження – Shareware.

Під умовно-безплатним програмним забезпеченням можна розуміти спосіб або метод розповсюдження комерційного ПЗ на ринку (тобто на шляху до кінцевого користувача), при якому випробувачеві пропонується обмежена за можливостями (не повнофункціональна або демонстраційна версія), терміном дії (тріал версія) або версія з вбудованим набридливим нагадуванням про необхідність оплати використання програми.

В угоді про використання (ліцензії для кінцевого користувача, EULA) також може бути обумовлена заборона на комерційне або професійне (не тестове) її використання.

Основний принцип умовно-безплатного ПЗ – «спробуй, перш ніж купити» (try before you buy). ПЗ що поширюється як умовно-безплатний, надається користувачам безоплатно. Звичайно користувач платить тільки за час завантаження файлів через Інтернет або за носій (CD диск, флешку, ключ). Протягом певного терміну, що становить зазвичай тридцять днів, він може користуватися програмою, тестувати її, освоювати її можливості.

Якщо після закінчення цього терміну користувач вирішить продовжити використання ПЗ, він зобов'язаний купити його (zareestruvatisya), заплативши авторіві певну суму.

В іншому випадку користувач повинен припинити використання ПЗ та видалити його зі свого комп'ютера.

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти, призначено для системи кібербезпеки для протидії криптоджекінгу та ботнетам.

В межах України в недостатній мірі представлені вітчизняні розробки в цій області.

Рішення завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем для протидії криптоджекінгу та ботнетам.
- Досліджена система для протидії криптоджекінгу та ботнетам.
- На основі отриманих результатів досліджень створена програмна реалізація системи кібербезпеки для протидії криптоджекінгу та ботнетам.

Розроблені під час виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання для протидії криптоджекінгу та ботнетам.

Розроблене програмне забезпечення має простий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань.

При створенні програмного забезпечення було використано об'єктно-орієнтований підхід, що відповідає сучасним тенденціям у галузі розробки комерційних програмних систем.

Програма реалізована на мові високого рівня Python. Дана мова програмування дозволяє найбільш ефективно обробляти дані призначені для системи кібербезпеки для протидії криптоджекінгу та ботнетам. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи кібербезпеки й включає програми, що реалізують її функції.

Програма призначена для виконання під управлінням багатозадачної операційної системи Windows 10/11.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм ДСТУ 9041:2020.

В цілому створене програмне забезпечення підтверджує правильність використаних проектних рішень та повністю відповідає вимогам технічного завдання. Створене програмне забезпечення має потенційну можливість для подальшого вдосконалення і застосування у різних галузях.

К6ПЗ-2026

					ВКРБ-125.26.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Kuznetsov, O., Atzeni, G., Arnesano, M., Randieri, C., Smirnov, O. «Secure IoT-based smart wheelchair system: From implementation to security enhancement strategy». *Security and Privacy of Cyber Physical Systems Emerging Trends Technologies and Applications*, 2025, pp. 225–257.

2. Kuznetsov, O., Smirnov, O., Kuznetsova, T., Shaikhanova, A., Svatowsky, I. «Privacy-utility trade-offs in IoT networks: A comparative analysis of differential privacy mechanisms for sensor data aggregation». *Security and Privacy of Cyber Physical Systems Emerging Trends Technologies and Applications*, 2025, pp. 589–622.

3. Lakhno, V., Malyukov, V., Smirnov, O., Bebeshko, B., Chubaievskiy, V., Zhumadilova, M., Malyukova, I., Smirnov, S. «Multifactorial Model for Targeted Attacks Counteracting Within the Framework of a Multi-Step Quality Game with Fuzzy Information». *8th International Symposium on Intelligent Informatics, ISI 2023*, 2025. vol 389. pp 377-389. Springer, Singapore.

4. Kuznetsov O., Frontoni E., Kuznetsova Y., Smirnov O., Moskovchenko I. «Trust-Based Security Architecture for Edge Computing: A Simulation Study of Dynamic Trust Evolution and Attack Detection». *CEUR Workshop Proceedings*, 2024, 3909, pp. 227–241.

5. Kuznetsov, O., Frontoni, E., Kryvinska, N., Chevardin, V., Smirnov, O. «Wireless Network Encryption Stream Ciphers, Computational Modeling, and Security Analysis». *Computational Modeling and Simulation of Advanced Wireless Communication Systems*, 2024, pp. 379–402.

6. Kuznetsov, O., Frontoni, E., Kryvinska, N., Smirnov, O., Imoize, G.L. «Computational Modeling of Enhanced Spread Spectrum Codes for Asynchronous Wireless Communication». *Computational Modeling and Simulation of Advanced Wireless Communication Systems*, 2024, pp. 403–447.

7. Ткаченко, О., Ільєнко, А., Улічев, О., Мелешко, Є., Смірнов, О. «Правові засади поширення інформаційних впливів в соціальних мережах». *Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка»*, 2024. № 2(26), С. 170–188.

8. Смірнова Т.В., Коноплицька-Слободенюк О.К., Буравченко К.О., Смірнов С.А., Кравчук О.В., Козірова Н.Л., Смірнов О.А. «Дослідження технологій забезпечення кібербезпеки хмарних сервісів IaaS, PaaS та SaaS». *Кібербезпека: освіта, наука, техніка*. 2024. №4(24), С. 6-27.

9. Вінтенко, Б., Миронець, І., Смірнов, О., Кравчук, О., Козірова, Н., Савеленко, Г., Коваленко, А. «Дослідження вимог та аналіз кібербезпеки програмного забезпечення інформаційно-керуючих систем АЕС, важливих для безпеки». *Кібербезпека: освіта, наука, техніка*. 2024. №3(23), С. 111-131.

10. Батрак О., Смірнова Т., Гнатюк В., Одарченко Р., Смірнов О. «Дослідження показників ефективності функціонування та перспектив розвитку систем ІР-телефонії». *Підводні технології*, 2024, № 13, с. 28-35.

11. Kuznetsov, O., Kryvinska, N., Ilchenko, O., Smirnova, T., Ulianovska, Y. «Comparative Analysis of Cryptocurrency Trading Platforms Using the Analytic Hierarchy Process». *CEUR Workshop Proceedings*, 2023, 3628, pp. 106-115.

12. Akhalaia, G., Iavich, M., Iashvili, G., Prysiashnyy, D., Smirnova, T. «Secure Encrypted Connection on Georgian Website». *CEUR Workshop Proceedings*, 2023, 3550, pp. 313-320.

13. Al-Mudhafar Aqeel, A.M., Smirnova, T., Buravchenko, K., Smirnov, O. «The method of assessing and improving the user experience of subscribers in software-configured networks based on the use of machine learning». *Advanced Information Systems*, 2023, 7(2), pp. 49-56

14. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yenchев, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». *CEUR Workshop Proceedings*, Volume 3530, 2023, pp. 256-265.

15. Kuznetsov, O., Kandiy, S., Frontoni, E., Smirnov, O. «Trade-offs in Post-Quantum Cryptography: A Comparative Assessment of BIKE, HQC, and Classic McEliece». *CEUR Workshop Proceedings*, Volume 3504, 2023, pp. 1-11.

16. Smirnov, O., Lakhno, V., Akhmetov, B., Chubaievskyi, V., Khorolska, K., Bebesko, B. «Selection of a Rational Composition of Information Protection Means Using a Genetic Algorithm». In: *Rajakumar, G., Du, KL., Vuppalapati, C., Beligiannis, G.N. (eds) Intelligent Communication Technologies and Virtual Mobile Networks. Lecture Notes on Data Engineering and Communications Technologies*, vol 131. 2023. Springer, Singapore. pp. 21-34.

17. Смірнова Т.В., Гнатюк С.О., Бердибаєв Р.Ш., Сидоренко В.М., Жигаревич О.К., «Система корелювання подій та управління інцидентами кібербезпеки на об'єктах критичної інфраструктури». *Кібербезпека: освіта, наука, техніка*, №3(19), 2023, С. 176-196.

18. Смірнов О.А. Козлов Я.О., Смірнова Т.В. «Дослідження застосування SIEM-систем для забезпечення кібербезпеки та захисту інформації». *II Міжнародна науково-практична Інтернет-конференція «Інновації та перспективні шляхи розвитку інформаційних технологій (ІПШРІТ-2023)»* м.Черкаси 6 грудня 2023 року – Черкаси: ЧДТУ.– 2023. – С.251-252.

19. Козлов Я.О., Смірнова Т.В., Смірнов О.А. «Дослідження SIEM-систем для забезпечення кібербезпеки». *VII міжнародна науково-практична конференція “Інформаційна безпека та комп'ютерні технології” до 30-ти річчя кафедри кібербезпеки та програмного забезпечення*, м. Кропивницький. 1 листопада 2023 р. – Кропивницький: ЦНТУ. – 2023. – С. 26.

20. Козлов Я.О., Козірова Н.Л., Смірнов О.А. «Дослідження структури та принципу роботи SIEM-системи». *VII міжнародна науково-практична конференція “Інформаційна безпека та комп'ютерні технології” до 30-ти річчя кафедри кібербезпеки та програмного забезпечення*, м. Кропивницький. 1 листопада 2023 р. – Кропивницький: ЦНТУ. – 2023. – С. 59.

21. Вінтенко Б.Ю., Смірнов О.А., Коваленко О.В., Смірнов С.А., Коваленко А.С. «Дослідження нормативних документів та галузевих стандартів розробки програмного забезпечення комп'ютерних систем управління АЕС, важливих для безпеки». *Системи управління, навігації та зв'язку*, 2023, вип. 2(72), С. 170-178.

22. Smirnov, O., Neskorodieva, T., Fedorov, E., Rudakov, K., Neskorodieva, A. «Method Detection Audit Data Anomalies on Basis Restricted Cauchy Machine» *CEUR Workshop Proceedings*, Volume 3187, 2022,

23. Smirnov O.A., Al-Oraiqat A.M., Ulichev O.S., Meleshko Ye.V., Al-Rawashdeh H.S., Polishchuk L.I. «Modeling strategies for information influence dissemination in social networks». *Journal of Ambient Intelligence and Humanized Computing* Volume 13, Issue 5. Springer, Cham. 2022, pp. 2463-2477.

24. Смірнова Т.В., Гнатюк С.О., Сидоренко В.М., Юдін О.Ю., Сидоренко С.Ю., «Модель визначення критичності галузевих інформаційно-телекомунікаційних систем». *Проблеми інформатизації та управління*, № 2(70). 2022. С. 28-37.

25. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Смірнов С.А., Поліщук Л.І., «Дослідження стійкості до диференціального криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку*, 2022, № 3(69). С. 93-98.

26. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Поліщук Л.І., Смірнов С.А. «Дослідження статистичної стійкості та швидкісних характеристик запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Вісник Хмельницького національного університету. Серія: «Технічні науки»*, № 2 (307). С. 46-52. 2022.

27. Смірнов О.А., Смірнова Т.В., Константинова Л.В., Смірнов С.А., Якименко Н.М., «Дослідження стійкості до лінійного криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного

захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку*, 2022, № 1(67). С. 84-89.

28. Smirnov O., Kuznetsov A., Zhora V., Onikiychuk A., Pieshkova O. «Hiding Messages in Audio Files Using Direct Spread Spectrum». *11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2021*, Cracow, Poland, 22-25 September 2021. P. 414-418

29. Smirnov O., Kuznetsov A., Lokotkova I., Kuznetsova T., Florov S., Lebid O. «Using Orthogonal Signals to Hide Information in Images». *4 IEEE International Conference on Advanced Information and Communication Technologies (AICT) - 2021*, Lviv, Ukraine, September 21-25, 2021. P. 255-260.

30. Smirnov O., Kuznetsov A., Girzheva O., Kiian A., Nakisko O., Kuznetsova T. «Advanced Code-Based Electronic Digital Signature Scheme». *2020 IEEE International Conference on Problems of Infocommunications Science and Technology, PIC S and T 2020*, Kharkiv, 6 October 2020-9 October 2020, P. 358-362.

31. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova K. «Data hiding scheme based on spread sequence addressing». *CEUR Workshop Proceedings Volume 2805*, 2020, Pages 44-58.

32. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». *International Journal of Computing*; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256.

33. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». *CEUR Workshop Proceedings. Volume 2740*, 2020, Pages 102-114.

34. Smirnov O., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». *Journal of theoretical and applied information technology* Vol.98. No 21, 2020, P. 3334-3346.

35. Smirnov O., Kuznetsov A., Arischenko A., Chepurko I., Onikiychuk A., Kuznetsova T. «Pseudorandom sequences for spread spectrum image steganography». *CEUR Workshop Proceedings* Volume 2654, 2020, Pages 122-131.
36. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New technique for data hiding in cover images using adaptively generated pseudorandom sequences». *CEUR Workshop Proceedings* Volume 2654, 2020, Pages 1-14.
37. Smirnov O., Lutsenko M., Kuznetsov A., Kiian A., Kuznetsova T., «Biometric cryptosystems: overview, state-of-the-art and perspective directions». *Lecture Notes in Networks and Systems*, vol 152. Springer, Cham. 2021, pp 66-84.
38. Smirnov O., Kuznetsov A., Onikiychuk A., Makushenko T., Anisimova O., Arischenko A. «Adaptive pseudo-random sequence generation for spread spectrum image steganography». *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Ukraine, Kyiv, May 14-18. 2020. P. 161-165.
39. Smirnov O., Kuznetsov A., Kiian A., Babenko V., Perevozova I., Chepurko I. «New Approach to the Implementation of Post-Quantum Digital Signature Scheme». *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Ukraine, Kyiv, May 14-18. 2020. P. 166-171.
40. Smirnov O., Kuznetsov A., Kiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-quantum application». *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Ukraine, Kyiv, May 14-18. 2020. P. 172-177.
41. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) *Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies*, vol 48. Springer, Cham. 2021. pp 557-587.

42. Smirnov, O., Markovets, O. Vovk, N., Turchyn, Y., «Model of informational support for social network administrators' content creation». *CEUR Workshop Proceedings* Volume 2616, 2020, Pages 125-136.

43. Smirnov, O., Shekhanin, K., Kuznetsov, A., Krasnobayev, V. «Detecting Hidden Information in FAT». *International Journal of Computer Network and Information Security (IJCNIS)*. Vol. 12, No. 3, 2020. PP.33-43.

44. Smirnov, O., Kuznetsov, A., Gorbacheva, L., Babenko, V., «Hiding data in images using a pseudo-random sequence», *CEUR Workshop Proceedings* Volume 2608, 2020, Pages 646-660.

45. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». *International Journal of Computing*; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407.

46. Smirnov, O., Kuznetsov, A., Reshetniak, O., Ivko, N., Katkova, T., Kuznetsova, T., «Generators of Pseudorandom Sequence with Multilevel Function of Correlation». *2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T)*, Kyiv, Ukraine, 8 – 11 October 2019 . P.517-522.

47. Smirnov, O., Ulichev, O., Meleshko, Y., Khokh, V., Goncharenko, I. «Method of Choosing Objects for Informational Influence in Social Networks during Information Campaign Based on the Analytic Hierarchy Process». *CEUR Workshop Proceedings*, Vol 2588, P. 215-227, 2019.

48. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». *CEUR Workshop Proceedings*, Vol 2588, P. 90-106, 2019.

49. Smirnov, O., Kuznetsov, A., Kiian, A., Gorbenko, Y., Cherep, O., Bexhter L. «Code-based Pseudorandom Generator for the Post-Quantum Period», *2019 IEEE International Conference on Advanced Trends in Information Theory (IEEE ATIT 2019)*. 18.12.19-20.12.19 Kyiv Ukraine. P. 204 – 209.

50. Smirnov, O., Kuznetsov, A., Nariezhnii, O., Stelnyk, S., Kokhanovska, T., Kuznetsova T., «Side Channel Attack on a Quantum Random Number Generator», *10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2019*; Metz; France; 18 - 21 September 2019. P.713-718.

51. Smirnov, O., Kuznetsov, A., Kiian, A., Pushkar'ov, A., Mialkovskyi, D., Kuznetsova, T., «Code-Based Schemes for Post-Quantum Digital Signatures», *10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2019*; Metz; France; 18-21 September 2019. P. 707-712.

52. Smirnov, O., Kuznetsov, A., Stefanovych, O., Gorbenko, Y., Krasnobaeв, V., Kuznetsova K. «Information Hiding Using 3D-Printing Technology», *10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2019*; Metz; France; 18-21 September 2019. P.701-706.

53. Smirnov, O., Hu, Z., Vasiliu, Y., Sydorenko, V., Polishchuk, Y., «Abstract Model of Eavesdropper and Overview on Attacks in Quantum Cryptography Systems», *10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2019*; Metz; France; 18-21 September 2019. P.399-405.

54. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Averchev, A., Pastukhov, M., Kuznetsova, K., «Formation of Pseudorandom Sequences with Special Correlation Properties», *2019 3rd International Conference on Advanced Information and Communications Technologies, AICT -2019/ Lviv, Ukraine, 2-6 July, 2019*, P. 395-399.

55. Smirnov, O., Kuznetsov, A., Kiian, A., Babenko, B., Zhosan, H., Prokopovych-Tkachenko, D., «Soft Decoding Method for Turbo-Productive Codes», *2019 3rd International Conference on Advanced Information and Communications Technologies, AICT 2019, Lviv, Ukraine, 2-6 July, 2019*, P. 129-134.

56. Smirnov, O., Kuznetsov, A., Kiian, A., Zamula, A., Rudenko, S., Hryhorenko, V., «Variance Analysis of Networks Traffic for Intrusion Detection in Smart Grids», *2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS)*, Kyiv, Ukraine April 17-19, 2019 P. 353-358.

57. Smirnov, O., Kuznetsov, A., Kavun, S., Babenko, B., Nakisko, O., Kuznetsova, K., «Malware Correlation Monitoring in Computer Networks of Promising Smart Grids», *2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS)*, Kyiv, Ukraine April 17-19, 2019 P. 347-352.

58. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Pastukhov, M., Kuznetsova, K., Prokopovych-Tkachenko, D., «Discrete Signals with Special Correlation Properties», *CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019*, Pages 618-629.

59. Smirnov, O., Kuznetsov, A., Kiian, A., Kuznetsova, K., Ivko, T., Prokopovych-Tkachenko, D., «Soft Decoding Based on Ordered Subsets of Verification Equations of Turbo-Productive Codes», *CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019*, Pages 873-884.

60. Smirnov A.A., Kuznetsov A.A., Danilenko D.A., Berezovsky A., «The statistical analysis of a network traffic for the intrusion detection and prevention systems», *Telecommunications and Radio Engineering*. – Volume 74, Issue 1. – Begel House Inc. – 2015. – P. 61-78.

61. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New Technique for Hiding Data in Cover Images Using Adaptively Generated Pseudorandom Sequences». *CEUR Workshop Proceedings Volume 2732, 2020*, Pages 214-227.

62. Смірнов О.А., Коноплицька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В., Книшук А.В. «Вступ до кібербезпеки»: навчальний посібник – Кропивницький: ЦНТУ – 2022. – 968 с.