

Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
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 хмарного
сервісу протидії кіберзагрозам”**

Виконав здобувач вищої освіти
IV курсу, групи КБ-23МБ
ОПП «Кібербезпека»
спеціальності 125 «Кібербезпека»
_____ Шадурський М.Ю.
«___» _____ 2026 р.

Керівник проекту
кандидат технічних наук, доцент
_____ Смірнов С.А.
«___» _____ 2026 р.
Рецензент _____

Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
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

Шадурському Максиму Юрійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи *Програмне забезпечення системи кібербезпеки хмарного сервісу протидії кіберзагрозам*

2. Керівник роботи *Смірнов Сергій Анатолійович, канд. техн. наук, доцент*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 127-02 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 хмарного сервісу протидії кіберзагрозам*

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 С.А.
(прізвище та ініціали)

Завдання прийнято до виконання
« 27 » лютого 2026 р.

Підпис здобувача

Шадурський М.Ю.
(прізвище та ініціали)

АНОТАЦІЯ

Шадурський М.Ю. Програмне забезпечення системи кібербезпеки хмарного сервісу протидії кіберзагрозам. 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 хмарного сервісу протидії кіберзагрозам.

Метою розробки є програмне забезпечення системи кібербезпеки хмарного сервісу протидії кіберзагрозам.

Результат роботи – програмна реалізація системи кібербезпеки хмарного сервісу протидії кіберзагрозам.

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, хмарний сервіс, протидія кіберзагрозам

ABSTRACT

Shadurskyi M.Yu. Software for the cloud service cybersecurity system to counter cyber threats. 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, which is intended for the cloud service cybersecurity system to counter cyber threats.

The purpose of the development is the software for the cloud service cybersecurity system to counter cyber threats.

The result of the work is the software implementation of the cloud service cybersecurity system to counter cyber threats.

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, cloud service, countering cyber threats

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.....	15
2.3 Розгорнута постановка завдання	15
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	17
3.1 Опис функціонування системи	17
3.2 Розробка структурної схеми.....	22
3.3 Розробка функціональної схеми	29
3.4 Розробка діаграми процесів.....	30
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	32
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	32
4.2 Захист розробленого програмного забезпечення.....	46
5 ВПРОВАДЖЕННЯ СИСТЕМИ КІБЕРБЕЗПЕКИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	48
6 ОСНОВНІ ВИСНОВКИ.....	55
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	57

					ВКРБ-125.26.0040.00.00.ПЗ			
Вим	Арк.	№ докум.	Підп.	Дата	<i>Програмне забезпечення системи кібербезпеки хмарного сервісу протидії кіберзагрозам</i>	Літ.	Аркуш	Аркушів
<i>Розроб.</i>		<i>Шафурський М.Ю</i>				Б	1	63
<i>Перев.</i>		<i>Смірнов С.А.</i>						
<i>Н.контр.</i>		<i>Коваленко А.С.</i>				<i>ЦНТУ КБ-23МБ</i>		
<i>Затв.</i>		<i>Смірнов О.А.</i>						

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

DHCP	протокол, що використовується для динамічного розподілу IP-адрес.
DNS	розподілена служба Інтернет, використовувана для зіставлення логічних (доменних) імен і IP-адрес. DNS використовується для забезпечення можливості роботи зі зрозумілими й іменами, що легко запам'ятовуються, замість IP-адрес у числовому форматі
ICMP	протокол
IDS	система виявлення атак
IP	адресний протокол
LAN	локальна мережа
NAT	трансляція IP-адрес із одного адресного простору в IP-адреси іншого адресного простору
Proxu	програма-посередник, що транслює запити різних протоколів із локальної мережі в зовнішню мережу
TCP	протокол обміну даними на транспортному рівні
UDP	протокол
URL	уніфікований показник інформаційного ресурсу (стандартизований рядок символів, що вказує місцезнаходження документа в мережі Інтернет)
ЕЦП	електронний цифровий підпис
КД	ключова дискета
НСД	несанкціонований доступ
ОС	операційна система
ПЗ	програмне забезпечення
ПК	персональний комп'ютер

ВСТУП

Актуальність теми. Кіберзлочино́м є будь-яка кримінальна активність, де об'єктом як мета й/або інструмента є комп'ютер або мережний пристрій. У деяких кіберзлочинах здійснюються прямі атаки на комп'ютери або інші пристрої з метою виводу з ладу. В інших – комп'ютери використовуються для поширення вредоносів, незаконної інформації, зображень і інших матеріалів.

Існує чотири найпоширеніші способи, якими користуються кіберзлочинці. Перший, котрого бояться багато з людей – використання шкідливих програм. Імовірно, ви розумієте, що існує безліч методів експлуатації систем, і наскільки важливо використовуватися різними мірами безпеки, наприклад, установлювати довгі паролі й робити регулярні відновлення. Цей тип атак базується на зловживанні комп'ютерами й мережами.

Другий спосіб – DDoS атаки, коли зловмисник користується комунікаційним мережним протоколом для створення величезної кількості запитів до сервера або служби. У цьому типі атак головна мета – вивести з ладу об'єкт впливу.

Третій спосіб – комбінація соціальної інженерії й шкідливого коду. Найбільш відома форма подібного роду атак – фішинг, коли жертву примушують до певних дій.

Четвертий спосіб – незаконна діяльність: поширення незаконного контенту, домагання, грумінг і т.д.

Як ви могли переконатися, кіберзлочини містять у собі широкий діапазон незаконних діянь, починаючи від шахрайства й крадіжки персональної інформації й закінчуючи злочинами на ґрунті ненависті й поширення наркотиків. Між цими видами існуємо безліч перетинань, і складно провести точну границю. Наприклад, фішингова атака може бути спрямована на крадіжку персональної інформації. У той же час, підробка особистості згодом може використовуватися

					ВКРБ-125.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

для одержання грошей, контрабандистами наркотиків або навіть терористів. Важливо розуміти, що кіберзлочини не завжди асоціюються з витонченими схемами й не завжди зачіпають «глибокий інтернет». Найкращий метод захисту від кібератак – бути в курсі сучасних погроз.

У даній роботі буде розглянута така погроза як соціальна інженерія або інша назва цієї погрози – фішинг.

Мета й завдання дослідження. Метою роботи є програмне забезпечення системи кібербезпеки хмарного сервісу протидії кіберзагрозам.

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

- Огляд існуючих систем хмарного сервісу протидії кіберзагрозам.
- Дослідження системи кібербезпеки хмарного сервісу протидії кіберзагрозам.
- Програмна реалізація системи кібербезпеки хмарного сервісу протидії кіберзагрозам.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі хмарного сервісу протидії кіберзагрозам.

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

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Фішингові атаки продовжуються, оскільки вони ідеально пасують до сучасної роботи. Вони надходять у вигляді рахунків-фактур, спільних документів, скидання паролів, заявок у службу підтримки та «швидких послуг», які здаються буденними, поки не перестають бути такими. Фішингові шахрайства використовують людську психологію, тому навіть обережні команди програють, коли діють швидко.

Для бізнесу ставки здаються несправедливо перекошеними. Успішна фішингова атака може призвести до захоплення облікового запису, компрометації облікових даних, витоку даних та порушення їхньої безпеки. Фінансовими наслідками можуть бути пряма крадіжка коштів, а потім додаткові витрати на очищення, юридичні витрати, повернення коштів та відновлення особистих даних.

Довгострокову шкоду важче оцінити. Клієнти очікують, що ви будете зберігати їхню конфіденційну інформацію в безпеці, тому фішингові атаки можуть за тиждень підірвати довіру, на формування якої знадобилися роки.

1.2 Область застосування

Антифішинг – це набір рішень, які зменшують кількість фішингових атак на електронну пошту, веб-трафік та ідентифікацію вашого домену. На практиці вони поєднують кілька рівнів:

– Елементи керування електронною поштою, які фільтрують вхідні листи та виловлюють шкідливі листи, перш ніж користувачі їх побачать.

					ВКРБ-125.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

– Елементи керування веб-сайтами та системою доменних імен (DNS), які блокують фішингові сайти та шкідливі домени, коли користувачі натискають на них.

– Автентифікація домену, яка зменшує випадки підробки домену (коли зловмисник підробляє ваш домен «Від», щоб електронний лист виглядав так, ніби він надійшов від вашої компанії) та атаки під чужим ім'ям (повідомлення, які видають себе за певну особу у вашій компанії, наприклад, вашого генерального директора або фінансового директора).

– Навчання та тестування, які допомагають людям виявляти ознаки фішингу та повідомляти про підозрілі повідомлення.

Такий багаторівневий підхід має велике значення у 2026 році. Сьогодні фішингові кампанії спрямовані на електронну пошту, додатки для чату, QR-коди та видання себе за допомогою дипфейків. І один рівень рано чи пізно щось пропустить.

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

					ВКРБ-125.26.0040.00.00.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

Цей список 10 найкращих інструментів зосереджений на інструментах, що підтримують захист від фішингу для вашого бізнесу, включаючи платформи безпеки електронної пошти, інструменти ідентифікації домену та елементи керування, що блокують фішинг на рівні призначення.

Control D

Control D працює шляхом фільтрації на рівні DNS. Він блокує або перенаправляє DNS-запити для ризикованих доменів, які можуть містити фішингові сайти, шкідливе програмне забезпечення та інші потенційні загрози. Користувач може натиснути на фальшиве посилання в електронному листі, але інструмент зупиняє процес до того, як браузер відкриє сторінку. Він може захистити всю мережу одночасно за допомогою налаштувань DNS. Для роумінгових пристроїв ви можете підключити їх до конфігурації на рівні пристрою або елементів керування.

Команди можуть налаштувати це за кілька кроків. Спочатку змініть налаштування DNS вашого маршрутизатора або пристрою, щоб вони вказували на сервери Control D. Далі увійдіть на інформаційну панель і виберіть фільтри для категорій фішингу.

					ВКРБ-125.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

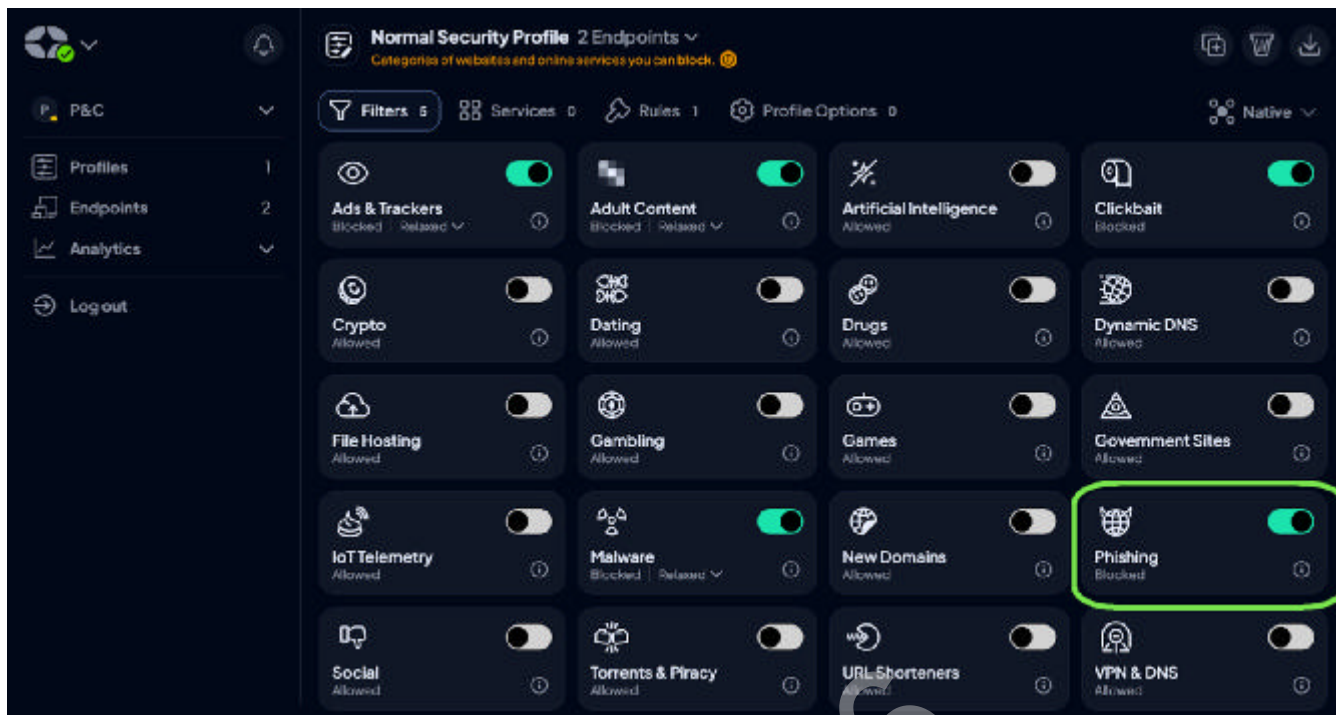


Рисунок 2.1 – Інтерфейс користувача Control D

Перевірте це за допомогою безпечних посилань, щоб підтвердити блокування. Для найкращих результатів поєднайте це з навчанням співробітників щодо виявлення підозрілих електронних листів. Потім щотижня переглядайте журнали, щоб виявити закономірності спроб та скоригувати правила.

Основні характеристики:

- Ви блокуєте фішингові сайти за групами, такими як хости шкідливого програмного забезпечення або трекери реклами, а також додаєте власні списки ризикованих доменів. Це виявляє кампанії, які часто змінюють адреси.
- Він направляє запити безпечними шляхами, щоб уникнути перехоплення, використовуючи такі протоколи, як DNS-over-TLS (метод шифрування DNS-запитів).
- Відстежуйте всі запити за допомогою позначок часу та експортуйте дані до таких інструментів, як Splunk, для глибшого аналізу. Ви можете використовувати це для виявлення повторних спроб та навчання персоналу.

Microsoft Defender for Office 365

Microsoft Defender для Office 365 чудово підходить для організацій, які вже використовують M365, завдяки вбудованим зв'язкам з іншими рішеннями безпеки Microsoft. Microsoft Defender для Office 365 захищає команди, які покладаються на Microsoft 365, скануючи електронні листи та файли на наявність ознак фішингу. Він виявляє підробку, коли зловмисники підробляють дані відправника, щоб здаватися надійними.

Інструмент також виявляє спроби видавання себе за іншу особу, такі як повідомлення, що імітують керівників, та компрометацію ділової електронної пошти, яка спрямована на шахрайство з облікових записів або крадіжку даних. Сканування виконується в режимі реального часу в таких програмах, як Outlook і Teams (хоча це може залежати від вашого плану та налаштувань), блокуючи шкідливий контент, перш ніж він потрапить до поштових скриньок.

Основні характеристики:

- Налаштуйте такі параметри, як захист від підробок та перевірка на імперсонацію, щоб зменшити кількість хибних сповіщень та одночасно закрити точки впливу. Це використовує машинне навчання для адаптації до ваших налаштувань.
- Він обробляє фішингові повідомлення в електронних листах, вкладеннях і посиланнях у таких інструментах, як SharePoint або OneDrive. Це запобігає загрозам у виконанні щоденних завдань.
- Помістіть підозрілі елементи в карантин та реєструйте події для швидкого виправлення. Інтегруйтеся з планами реагування за допомогою сповіщень та масового очищення для пришвидшення відновлення.

Proofpoint Core Email Protection

Proofpoint забезпечує безпеку електронної пошти, перевіряючи повідомлення на наявність фішингу, перш ніж вони потраплять до поштових скриньок. Він спрямований на атаки, спрямовані на людей, такі як фішингові електронні листи, що крадуть облікові дані або поширюють шкідливе програмне

					ВКРБ-125.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

забезпечення. Інструмент підходить для більших груп і розширює можливості малого та середнього бізнесу, який справляється зі складнішими загрозами. Він поєднує багаторівневі перевірки, такі як сканування репутації та аналіз поведінки, для виявлення різноманітних атак.

Основні характеристики:

- Виявляйте закономірності у поширених методах та компрометації ділової електронної пошти, коли зловмисники видають себе за лідерів, щоб запитувати передачу даних або отримувати доступ до даних. Він використовує машинне навчання для позначення аномальної поведінки відправника та зменшення кількості хибнопозитивних результатів.

- Перепишіть посилання для безпечного сканування файлів, що запускаються за принципом «клікни та запусти» в ізольованих просторах, щоб знайти приховані загрози. Це зменшує ризик появи шкідливих сайтів або заражених доповнень.

- Автоматизуйте виправлення, такі як карантин та сповіщення, щоб допомогти командам швидко реагувати на спроби. Залучайте дані до планів, щоб пришвидшити очищення та виявлення закономірностей.

Mimecast Advanced Email Security

Mimecast будує свою систему розширеної безпеки електронної пошти (Advanced Email Security) на основі чіткого аналізу фішингу, компрометації ділової електронної пошти, програм-вимагачів та атак «нульового дня». Вона застосовує машинне навчання для виявлення ризиків в електронній пошті та інструментах, таких як Microsoft Teams або Slack. Система сканує повідомлення в режимі реального часу, блокуючи шкідливий контент перед доставкою. Ця система допомагає компаніям, які обробляють великі обсяги електронної пошти, від середнього до великого бізнесу в регульованих галузях.

Основні характеристики:

- Перепишуйте посилання в електронних листах та скануйте їх при кожному кліку, виявляючи затримки переадресацій від зловмисників. Це

					ВКРБ-125.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

запобігає потраплянню користувачів на небезпечні сайти після початкової перевірки на безпеку.

- Ретельно перевіряйте файли, запускайте їх у безпечних місцях або змінюйте їх на нешкідливі типи. Це містить шкідливе програмне забезпечення з фішингових розсилок та пришвидшує безпечний доступ.

- Отримуйте глобальні дані для швидкого виявлення тенденцій фішингу. Команди використовують це для швидшого виправлення проблем, таких як карантин або сповіщення в планах реагування.

IRONSCALES

IRONSCALES використовує адаптивний штучний інтелект для захисту поштових систем від фішингу. Він вивчає звичайні схеми спілкування та позначає аномалії як потенційні загрози. Платформа виконує постійне сканування та автоматичні виправлення для обробки таких атак, як компрометація ділової електронної пошти або захоплення облікових записів. Вона підходить для середніх фірм та підприємств, яким потрібен розумний захист з мінімальними зусиллями. Інтеграція відбувається через API за допомогою таких інструментів, як Microsoft 365 або Google Workspace.

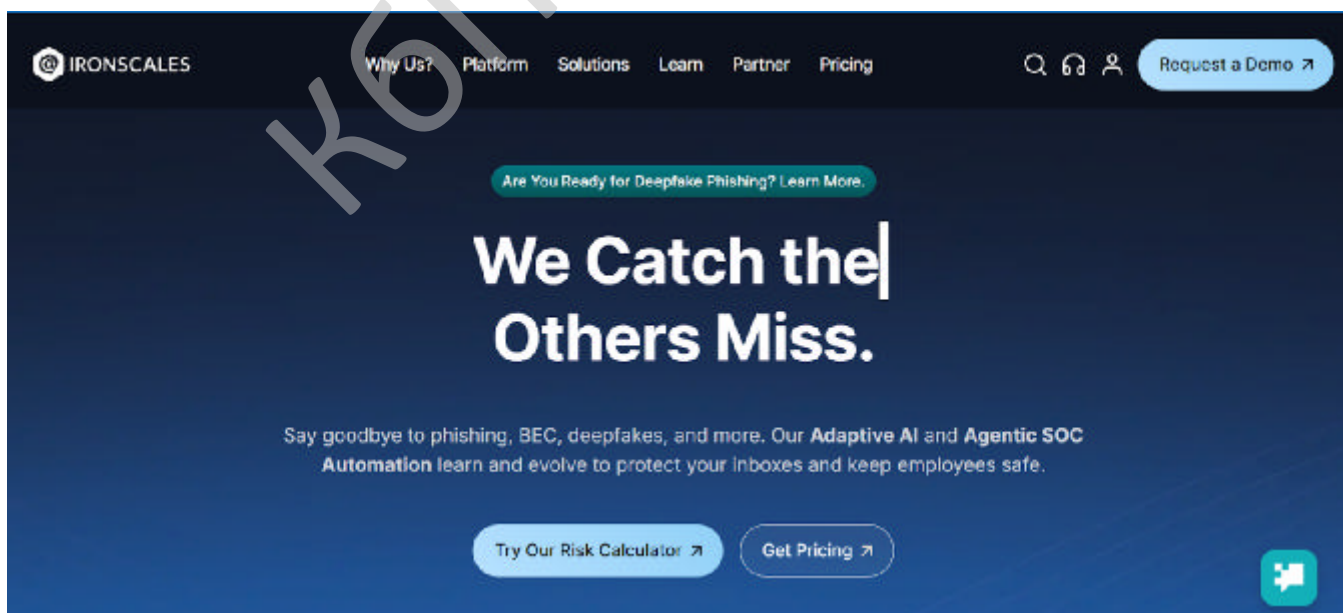


Рисунок 2.2 – інтерфейс користувача IRONSCALES

					ВКРБ-125.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

У тестах він блокує високий рівень спаму та фішингу за допомогою налаштувань у режимі реального часу. Він добре підходить для користувачів хмарної електронної пошти з навчальними додатками. Етапи навчання можуть потребувати часу, а додаткові функції, такі як симуляції, збільшують витрати, тому спочатку перевірте потреби.

Основні характеристики:

- Перевіряє метадані електронної пошти (наприклад, інформацію про відправника та позначки часу), вміст і сигнали поведінки для виявлення ризиків. Це адаптується для виявлення нових загроз без ручних оновлень.

- Виконує виправлення, такі як карантин або видалення, у разі фішингу, навіть після доставки до папки "Вхідні". Це використовує штучний інтелект для швидкої нейтралізації загроз у схожих повідомленнях.

- Використовує глобальні дані та закономірності для обробки складних атак або нових приманок, які пропускають стандартні перевірки. Команди отримують сповіщення про швидкі дії під час реагування.

Abnormal

Abnormal зосереджується на поведінковому аналізі для соціальної інженерії та компрометації ділової електронної пошти, з акцентом на атаках, яким бракує традиційних індикаторів шкідливого програмного забезпечення. Він позиціонує ВЕС як фінансово збиткову категорію, зумовлену соціальною інженерією.

Основні характеристики:

- Виявлення, спрямоване на атаки, що видаються за іншу особу, та перехоплення розмов.

- Зосередьтеся на контексті та взаємозв'язках, що допомагає, коли зловмисники використовують реалістичний тон та час.

- Покриття, яке допомагає запобігти спробам фішингу, спрямованим на фінансові робочі процеси та команди підтримки.

					ВКРБ-125.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

Cofense

Cofense відомий тим, що поєднує виявлення та реагування на фішинг зі звітністю користувачів та навчанням. Його платформа орієнтована на виявлення та усунення фішингових загроз, а також навчання співробітників розпізнаванню та повідомляння про реальні атаки, які обходять традиційні фільтри.

Основні характеристики:

- Надійні робочі процеси звітності користувачів, що дозволяють співробітникам швидко позначати підозрілі повідомлення.
- Інструменти реагування, які допомагають видаляти шкідливі електронні листи з поштових скриньок.
- Симуляція фішингу та навчання з безпеки, що відображає те, що насправді бачить ваш бізнес.

Valimail

Valimail зосереджується на ідентифікації домену, пропонуючи продукти, побудовані на основі автентифікації, звітності та відповідності повідомлень на основі домену (DMARC), рамки політики відправника (SPF) та ідентифікованої пошти DomainKeys (DKIM). Він позиціонує правозастосування як спосіб зупинити фішингові атаки та атаки, що видаються за іншу особу, а його матеріали описують використання DMARC для зменшення кількості підроблених відправників.

Основні характеристики:

- Підтримка застосування DMARC для зменшення підміни доменів та захисту довіри до бренду.
- Видимість неавторизованих відправників, що допомагає виявляти загрози електронної пошти, спрямовані на ваш домен.
- Засоби керування, які можуть зменшити кількість шкідливих електронних листів, що використовують імітацію точного домену.

					ВКРБ-125.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

ZeroFox

ZeroFox позиціонує свій захист від фішингу доменів таким чином, щоб виявляти та блокувати фішинг у першопричині, а не просто блокувати фішингові сайти. Він зосереджується на знищенні фішингової інфраструктури на рівні домену.

Основні характеристики:

- Моніторинг та реагування на фішингові домени та підроблені вебсайти.
- Дії на рівні домену, що доповнюють фільтрацію електронної пошти та блокування DNS.
- Захист бренду для компаній, які працюють з клієнтами та за яких видають себе за інших.

Sublime

Sublime збирає специфічні для організації дані, такі як схеми комунікації або списки VIP-персон, щоб виявляти загрози, такі як схожі домени або видавання себе за інших. Платформа використовує агенти штучного інтелекту, які адаптують правила в режимі реального часу, блокуючи атаки, перш ніж вони потраплять до поштових скриньок. Вона підходить для команд безпеки середніх і великих компаній, які цінують контроль і швидке налаштування порівняно з системами, прив'язаними до певних постачальників.

Основні характеристики:

- Застосовує сигнали, характерні для організації, такі як списки VIP-персон та звички відправників, для позначення імітації себе за іншу особу або спроб шахрайства.
- Сканує електронні листи на наявність схожих доменів, орфографічних помилок та тонких хитрощів. Ця адаптивна логіка обробляє QR-коди, контент, згенерований штучним інтелектом, та інші нові методи за допомогою спеціальних правил на основі коду.
- Надає повний аналіз причин спрацьовування виявлених даних, а також інструменти для швидкої сортування повідомлених електронних листів.

					ВКРБ-125.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

2.2 Обґрунтування вибору засобів для побудови системи кібербезпеки та мови програмування

Python – це потужна мова програмування, яка проста у вивченні. Він має ефективні структури даних високого рівня та простий, але ефективний підхід до об'єктно-орієнтованого програмування. Елегантний синтаксис і динамічна типізація Python разом з його інтерпретованим характером роблять його ідеальною мовою для створення сценаріїв і швидкої розробки додатків у багатьох сферах на більшості платформ.

Інтерпретатор Python і обширна стандартна бібліотека доступні у вихідному або двійковому вигляді для всіх основних платформ на веб-сайті Python <https://www.python.org/> і можуть вільно поширюватися. Цей же сайт також містить дистрибутиви та вказівники на багато безкоштовних сторонніх модулів Python, програм і інструментів, а також додаткову документацію.

Інтерпретатор Python легко розширюється за допомогою нових функцій і типів даних, реалізованих у C або C++ (або інших мовах, які можна викликати з C). Python також підходить як мова розширення для налаштовуваних програм.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи кібербезпеки хмарного сервісу протидії кіберзагрозам.

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

					ВКРБ-125.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

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-125.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Фішинг (англ. phishing, походить від fishing – рибний лов, вивудження) – це самий популярний метод в інтернет-просторі, використовуваний для злому паролів і крадіжки конфіденційної інформації. Основним елементом фішингу або фішингової атаки є процес створення дублюючої копії або клону відомого веб-сайту з метою крадіжки пароля користувача або іншої захищеної інформації. Даний метод одержав більшу популярність, тому що більшість користувачів не завжди дотримують елементарних вимог комп'ютерної безпеки.

Використовуючи різні психологічні прийоми, фішинг-шахраї спонукують користувачів вводити свої конфіденційні дані на фальшиву веб-сторінку (фішинг-сторінку), яка зовні не відрізняється від оригінального сайту, узятим шахраями за основу для копіювання.

У більшості випадків, єдина відмінність підробленої сторінки від справжньої – це її невірна адреса (URL). Найчастіше користувачі не звертають уваги на рядок адреси сторінки. А тому що зовнішній вигляд фішингової сторінки повністю копіює сторінку сайту-оригіналу, то більшість користувачів попадаються на виверт і діляться конфіденційною інформацією із шахраями. Звичайним користувачам буває досить складно ідентифікувати сторінку фішингових шахраїв через її високу вірогідність.

Як працює фішинг?

Хакери або фішинг-шахраї націлені на широке коло користувачів. Вони роблять масове розсилання фішингових листів і повідомлень із дуже високим ступенем схожості з оригіналом. Листа можуть містити офіційні логотипи установи, сайту, компанії або бренда, від імені якого йде обіг. Користувача спонукують різними причинами перейти по прикладеному фішинг-посиланню й

					ВКРБ-125.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

увести свої персональні дані. Причини можуть бути різними: усілякі конкурси й вікторини, перемога в розіграшах, неймовірно високі знижки й розпродажі, підтвердження ваших платіжних даних або ПІН-кодів, вимоги змінити ваші дані для входу в систему під приводом захисту від злому й т.д.

У результаті таких дій, користувач попадає на сторінку фішингу, практично яка не відрізняється від оригіналу, і вводить свою конфіденційну інформацію. У ту ж секунду вона ставати відомою шахраям і може бути використана для подальших незаконних дій.

Приклад фішингу

Візьмемо як приклад популярну соціальну мережу «Facebook». Допустимо, хакер створив сторінку, що ідеально схожа на сторінку входу в систему «Facebook», але змінив її URL-адресу. Наприклад, fakebook.com або faebook.com або будь-який інший URL-адреса, що дуже схожа на оригінальний. Користувач, потрапляючи на таку сторінку, може не звернути уваги на невірну адресу сторінки через схоже написання адреси сторінки з оригіналом. І може прийняти дану фішингову сторінку за справжню сторінку входу в «Facebook», і без побоювання скористається реєстраційною формою для входу в систему.

Таким чином, користувач, що не помітив різниці й скористався підробленою сторінкою, міг увести свої реєстраційні дані й дати доступ шахраєві до свого аккаунту. Одночасно із цим, для приховання шахрайства, користувач буде переспрямований на вихідну сторінку «Facebook».

Розглянемо можливий приклад з життя: Ігор працює програмістом і написав програму, що дозволяє одержати доступ до реєстраційних даних користувача. Потім він створює підроблену сторінку входу в соціальну мережу «Facebook», що містить шкідливу програму, і розміщає її на «[https://www.facebouk.com/ money-online](https://www.facebouk.com/money-online)». В Ігоря є друг Павло. І Ігор відсилає Павлу повідомлення: «Привіт, Паша, я знайшов спосіб легкого заробітку в інтернеті. Ти обов'язково повинен його побачити на [https://www.facebouk.com/ money-online](https://www.facebouk.com/money-online)». Павло переходить по посиланню й

					ВКРБ-125.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

бачить перед собою стандартну сторінку входу в систему «Facebook». Як звичайно, він уводить свої ім'я користувача й пароль. Тепер всі реєстраційні дані пересилаються Ігорю, а Павло перенаправляється на сторінку з радами по заробітку грошей у мережі інтернет <<https://www.facebook.com/money-online.html>>. От і всі, аккаунт Павла в «Facebook» був зламаний.

Як створити сторінку фішингу за лічені хвилини?

Хакери або інтернет-шахраї використовують різні варіанти створення фішинг-сторінок. Як приклад ми розглянемо один з найпростіших способів.

Шахрай переходять на сторінку сайту, аналог якого береться за основу майбутньої сторінки. Як правило береться сторінка реєстрації й уведення даних для доступу до особистого аккаунту на сайті. Потім копіює вихідний код сторінки (таку функцію надає будь-який браузер) і зберігає його в будь-якому текстовому редакторі, наприклад у блокноті. За допомогою пошуку знаходить у вихідному коді оригінальний URL-адреса для входу в систему й підмінюють його адресою програми (скрипту). У ній прописують умови підміни адрес, алгоритм дій після уведення реєстраційних даних і спосіб одержання цих даних шахраєм. Основні роботи зі створення фішингової сторінки закінчені. Далі хакеру необхідно розмістити свою облудну сторінку в мережі інтернет і перенаправляти до неї як можна більший потік користувачів.

Як розмістити фішингову сторінку в інтернеті?

Щоб помістити фішингову сторінку в інтернеті під URL-адресою, шахраєві необхідна наявність двох речей:

1. Домен.
2. Веб хостинг.

Одержите безкоштовний домен

Раніше, можна було скористатися послугами сайтів, що надають вам у користування безкоштовні доменні імена. Але тепер, власники дуже ретельно переглядають уміст надаваних безкоштовних доменів і блокують підозрілі з них. Тому шахраї перемкнулися на використання платних доменів. Або

використовують симбіоз із платного хостингу й безкоштовного домену від одне й того ж постачальника. Як варіант, такі послуги зі створення безкоштовного домену пропонує сервіс «**Bluehost**», якщо ви купите будь-який план їх платного хостингу.

Одержите веб-хостинг

Майже всі безкоштовні хостингові площадки блокують розміщення будь-яких фішинг-сторінок, або сторінок, що викликають обґрунтовану підозру. Таким чином, шахраї переходять на платні пакети хостингу. Фішингова сторінка живе обмежений час у мережі, поки не буде заблокована. Тому зломщики вибирають мінімальні пакети хостингу для розміщення своїх сторінок. Після виконання всіх процедур шахрай викладає свою фішингову сторінку в мережу інтернет і займається залученням потенційних жертв.

Як ви можете захистити себе від шахрайства?

У цей час шахраї розробили безліч усіляких хитрувань і пасток, щоб заманити довірливого користувача на свою сторінку, і змусити його ввести потрібну для хакера інформацію. Інтернет-шахраї можуть зв'язатися з вами по-різному, наприклад, по електронній пошті, особистим повідомленням, повідомленням у соціальних мережах, за допомогою оголошень на сайтах і т.д. Нажавши на будь-які посилання в таких повідомленнях, ви потрапите на сторінку входу на фішинговій сторінці. Тому щораз, коли ви одержуєте який-небудь електронний лист, що призивають вас перейти по посиланню, першою справою ви повинні звернути увагу на адресу цього посилання. Ви повинні витратити небагато часу й вивчити це посилання, звертаючи увагу на нюанси її написання. Якщо ви сумніваєтеся, по знайдіть за допомогою будь-якого пошуку (google, yandex) офіційний сайт тої організації, від якої ви одержали електронного листа, і звірте правильність написання адреси сторінки. Чи збігається адреса посилання в листі й на офіційному сайті. Перевірте також, є присутнім чи в адресному рядку символ захисту (зелений замочок, приставка https). Як бачите, варто виявити

					ВКРБ-125.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

небагато уважності, і ви можете істотно знизити ризик бути обманутими, а ваші дані – зламаними. От приклад такий фішингової сторінки **«Facebook»**.

Але існують підроблені сторінки більше складні. Більшість людей не будуть підозрювати цю сторінку (знімок, зазначений вище), тому що є префікс **«https»** з іконкою безпечного зеленого кольору й без яких-небудь помилок на www.facebook.com. Однак це фішинг-сторінка. Зверніть увагу на написання URL-адреси. Це **«https://www.facebook.com.beck.com»**. У даному виді **«www.facebook.com»** є субдоменом **«beck.com»**. Браузер **«Google Chrome»** не відрізняє піддомен і домен, на відміну від **«Firefox»**. А захисні **«SSL-сертифікати»** (HTTPS) можуть бути отримані від багатьох постачальників, деякі з них навіть надають **«сертифікат SSL»** безкоштовно протягом 1 року. Тому надто важливо завжди звертати увагу на правильність зазначеного URL-адреси.

Це приклад звичайної фішинг-сторінки **«Facebook»** з деякими змінами в написанні назви. Використовується заміна однієї букви, що при швидкому погляді може бути непомітна.

Фішинг-шахрайство

Фішинг-шахрайство – це спроби шахраїв (хакерів, зловмисників) обдурити вас, з метою вивідати вашу будь-яку конфіденційну інформацію, починаючи від імені користувача й закінчуючи банківськими даними. Як описано вище, фішинг-шахраї зосереджені на досягненні мети одержати вашу інформацію удаленно. Вони намагаються змусити вас самостійно ввести ваші реєстраційні дані й передати їх на сервер шахраєві.

Фішингова електронна пошта

Працюючи в мережі інтернет ви не застраховані від випадку бути обманутими, ви навіть випадково можете наткнутися на фішингову сторінку. Але варто пам'ятати, що в більшості випадків, фішинг-шахрайство відбувається по електронній пошті. Хакер легко підмінюють своя вихідна адреса в електронному листі на адресу будь-якого легітимного сайту або установи, використовуючи для цього всілякі програми й онлайн-сервіси. Тому користувачі, що одержали такий

					ВКРБ-125.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

лист, упевнені, що лист відправлений з офіційного сайту. Популярні поштові служби, такі як «Gmail», досить розумні, щоб ідентифікувати фішингову електронну пошту, і перенаправляють її в папку спаму. Проте, немає стовідсоткової гарантії, що всі фішингові листи будуть заблоковані. Хакери постійно вдосконалюють способи обійти блокування й знаходять можливість відправляти фішингові листи.

3.2 Розробка структурної схеми

Фішинговим атакам не потрібне екзотичне шкідливе програмне забезпечення для роботи. Їм потрібна одна мить дотримання вимог. Саме тому зловмисники продовжують інвестувати в переконання. Вони використовують фішингові методи, які виглядають як законні ділові комунікації, а потім вставляють оманливе посилання, фальшиву сторінку входу або запит на переказ грошей.

Багато **фішингових атак** все ще успішно здійснюються тими ж шляхами:

- Посилання веде на фішингові сайти, які збирають облікові дані для входу
- Вкладення розповсюджує шкідливе програмне забезпечення після того, як користувач увімкне макроси або надасть доступ
- Фейковий ланцюжок обговорень переконує когось змінити банківські реквізити або схвалити платіж

Фішинг з використанням спеціальних методів тут проявляється як «тихий» варіант. Одна людина стає мішенню, часто у сфері фінансів, кадрів чи ІТ. Електронний лист виглядає особистим та правдоподібним. Додайте текст, написаний штучним інтелектом, та імперсонацію під виглядом дипфейку, і класичний знак «поганої граматики» зникає. Саме тому заходи боротьби з фішингом у 2026 році все більше покладаються на виявлення загроз за допомогою штучного інтелекту, оскільки якість тексту більше не відрізняє

					ВКРБ-125.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

справжнє від фальшивого.

Фішингові атаки «нульового дня» додають ще один нюанс. Нові домени та свіжі сторінки можуть з'являтися ще до того, як у стрічках новин їх буде позначено як відомі фішингові сайти. Саме тут найбільше значення мають контроль цільового призначення, посилення ідентифікації та швидке реагування.

Як працює антифішингове програмне забезпечення

Програмне забезпечення для захисту від фішингу аналізує вхідні повідомлення та веб-трафік на наявність ознак фішингових спроб, а потім блокує або містить знайдені дані. Різні продукти використовують різні сигнали, тому зазвичай найкращі результати досягаються при комбінуванні інструментів. На практиці ці інструменти, як правило, поділяються на кілька основних рівнів:

Фільтрація вхідних листів у режимі реального часу

Платформи безпеки електронної пошти оцінюють вхідні електронні листи перед доставкою. Вони перевіряють репутацію відправника, результати автентифікації та підказки щодо контенту. Деякі інструменти додають банери, які попереджають користувачів про зовнішніх відправників, незвичайні адреси для відповіді або небезпечну мову. Вони можуть поміщати підозрілі повідомлення в карантин або направляти їх на перевірку, що зменшує ризики, коли зловмисники атакують фішингову кампанію.

Аналіз шкідливих посилань та URL-адрес

Більшість фішингових атак вимагають кліку, тому аналіз посилань є основним засобом контролю. Інструменти оцінюють URL-адреси, розширюють переадресації та порівнюють цільові сторінки з базами даних відомих фішингових сайтів. Вони можуть **виявляти відомі шкідливі URL-адреси**, а потім блокувати доступ. У деяких стеках це включає перезапис URL-адрес та перевірку часу кліку для зменшення затримок переадресацій.

Ви побачите кілька пов'язаних термінів у документації продукту та консольях адміністратора:

– **Шкідливі URL-адреси:** сайти, відомі крадіжкою облікових даних або

					ВКРБ-125.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

доставкою шкідливого програмного забезпечення

- **Шкідливе посилання:** одне посилання в електронному листі або повідомленні, яке призводить до шкоди
- **Відомі шкідливі URL-адреси:** URL-адреси, які вже позначені тегами у стрічках загроз
- **Підроблені веб-сайти:** схожі сторінки, призначені для крадіжки конфіденційної інформації

Сканування навісного обладнання та корисного навантаження

Фішингове програмне забезпечення сканує вкладення, щоб виявляти шкідливі файли та підозрілу поведінку файлів. Це ключовий рівень для загроз електронної пошти, які розповсюджують шкідливе програмне забезпечення. Він не ідеальний, але зменшує легкі перемоги, які зловмисники отримують за допомогою файлових приманок.

Розпізнавання образів, евристика та машинне навчання

Розширені засоби захисту від фішингу використовують розпізнавання шаблонів та евристичний аналіз для виявлення підозрілих електронних листів, які не відповідають відомим підписам. Машинне навчання допомагає, вивчаючи, як виглядає «нормальний» стан у вашій організації, а потім позначає дивну поведінку, незвичайні зв'язки відправників та аномальні моделі відповідей. Багато постачальників описують це як автоматичне виявлення, яке адаптується до розвитку атак.

Розвідка, реагування та звітність про загрози

Розвідка про загрози важлива, оскільки зловмисники швидко змінюють інфраструктуру. Коли канали загроз оновлюються швидко, інструменти можуть виявляти більше фішингових доменів і блокувати фішингові сайти на ранніх етапах кампанії.

Для бізнесу можливості звітності — це не просто марнотратна функція. Вам потрібна звітність, яка показує, що було заблоковано, що пройшло через систему та на що натискали користувачі. Ви можете використовувати ці дані для

					ВКРБ-125.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

налаштування політик, зменшення кількості хибних спрацьовувань та оцінки того, чи покращується навчання з питань безпеки.

6 додаткових способів захисту вашого бізнесу від фішингу

Вам не потрібна розгалужена програма, щоб швидко зменшити ризик. Вам потрібна послідовність, яка закриває найпоширеніші прогалини, а потім підкріплює поведінку.

Захистіть ідентифікацію свого домену за допомогою DMARC, SPF та DKIM

Зловмисники люблять спуфінг, бо він масштабується. DMARC, SPF та DKIM допомагають зупинити підроблені повідомлення, які нібито надходять від вашого бренду. Такі інструменти, як Valimail та PowerDMARC, спрямовані на спрощення цього впровадження, за допомогою робочих процесів моніторингу та забезпечення дотримання правил, які допомагають вам безпечно переходити до суворіших правил.

Посилити політики безпеки електронної пошти

Налаштуйте захист від імітації себе, сканування посилок та правила вкладень для команд, які найчастіше стають мішенями. Фінанси, підтримка та відділ кадрів зазвичай знаходяться в радіусі вибуху. Визначте, що відбувається, коли хтось повідомляє про повідомлення, і зробіть результат передбачуваним, швидким та послідовним.

Додайте елементи керування DNS-рівня для блокування фішингу зовні електронної пошти

Фільтрація електронної пошти необхідна, але зловмисники не живуть усередині електронної пошти. Фільтрація DNS допомагає блокувати спроби фішингу, які надходять через чат-додатки, спільні документи, рекламу та QR-коди. Такі інструменти, як Control D, спрощують розгортання та налаштування цього рівня. Якщо ви можете блокувати фішинг **на рівні призначення**, ви зменшуєте проблему «одного кліку», яка перетворює спробу фішингу на справжній інцидент.

					ВКРБ-125.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

Використовуйте багатофакторну автентифікацію, стійку до фішингу

Багатофакторна автентифікація (MFA) обмежує збитки після компрометації облікових даних. Це один з небагатьох засобів контролю, який може перетворити крадіжку облікових даних на контрольовану подію, а не на порушення. Навчіть користувачів, як мають виглядати запити MFA, і встановіть чіткі інструкції щодо неочікуваних запитів.

Проведіть тренінг з підвищення обізнаності щодо безпеки та симуляцію фішингу

Навчання з питань безпеки працює, коли воно базується на реальних прикладах, які бачать ваші співробітники. Робіть його коротким, повторюйте та пов'язуйте зі звітністю. Моделювання фішингу допомагає вам оцінити готовність та виявити команди, які потребують додаткової уваги. KnowBe4 та Hoxhunt – це широко використовувані платформи для моделювання фішингових атак та навчальних робочих процесів, і багато компаній використовують їх для створення повторюваної практики.

Зробіть реагування на інциденти нудним навмисно

Напишіть односторінковий посібник з реагування на інциденти фішингу. Вкажіть, хто проводить сортування, як ви підтверджуєте спробу фішингу та що робити далі: пошук та видалення, скидання пароля, скасування сеансу та зв'язок з клієнтами, якщо потрібно. Коли трапляється справжній інцидент, «нудне та повторюване» перемагає «героїчне та імпровізоване».

Заключні думки

Фішингові атаки нікуди не зникнуть. Вони продовжують працювати, оскільки націлені на людей та робочі процеси, а не лише на програмне забезпечення. Бізнес найкраще працює з багаторівневим підходом, а не з використанням одного інструменту.

Почніть із надійного захисту вхідних листів, заблокуйте ідентифікацію домену, щоб зменшити підробку домену, та використовуйте фільтрацію DNS для блокування фішингових сайтів і шкідливих доменів, коли хтось натискає на

					ВКРБ-125.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

посилання. Потім додайте резервну копію за допомогою багатфакторної автентифікації (MFA), чітких можливостей звітності та простого плану реагування на інциденти.

Найчастіші запитання про програмне забезпечення для боротьби з фішингом

Що робить антифішингове програмне забезпечення?

Програмне забезпечення для захисту від фішингу допомагає зменшити кількість фішингових атак, виявляючи, блокуючи та стримуючи спроби фішингу, перш ніж вони завдадуть реальної шкоди. На практиці воно може фільтрувати вхідні електронні листи, щоб виявляти фішингові та шкідливі листи, сканувати наявність шкідливих URL-адрес та підозрілих посилань, а також використовувати інформацію про загрози для позначення відомих фішингових сайтів. Багато антифішингових рішень також блокують фішинг під час натискання кнопки, запобігаючи доступу до фішингових сайтів та інших шкідливих сайтів, часто блокуючи шкідливі домени.

Що персонал повинен розглядати як ознаки фішингу?

Навчіть персонал уповільнювати роботу, коли вони бачать термінові запити на оплату, неочікувані запити на вхід, дивні домени відправників та невідповідні відображувані імена. Попросіть їх швидко повідомляти про підозрілі посилання та повідомлення. Швидке повідомлення дає командам безпеки час стримати фішингову кампанію, перш ніж вона пошириться.

Як виглядає «хороший» антифішинговий стек для вашого бізнесу?

Практичний стек зазвичай включає безпеку електронної пошти, фільтрацію DNS для блокування фішингових сайтів, автентифікацію домену для зменшення спуфінгу, навчання з безпеки, симуляцію фішингу та багатфакторну автентифікацію, стійку до фішингу. Додайте інструменти реагування, які пришвидшують пошук, карантин та очищення, і ви зменшите ризик без додавання робочого навантаження.

					ВКРБ-125.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

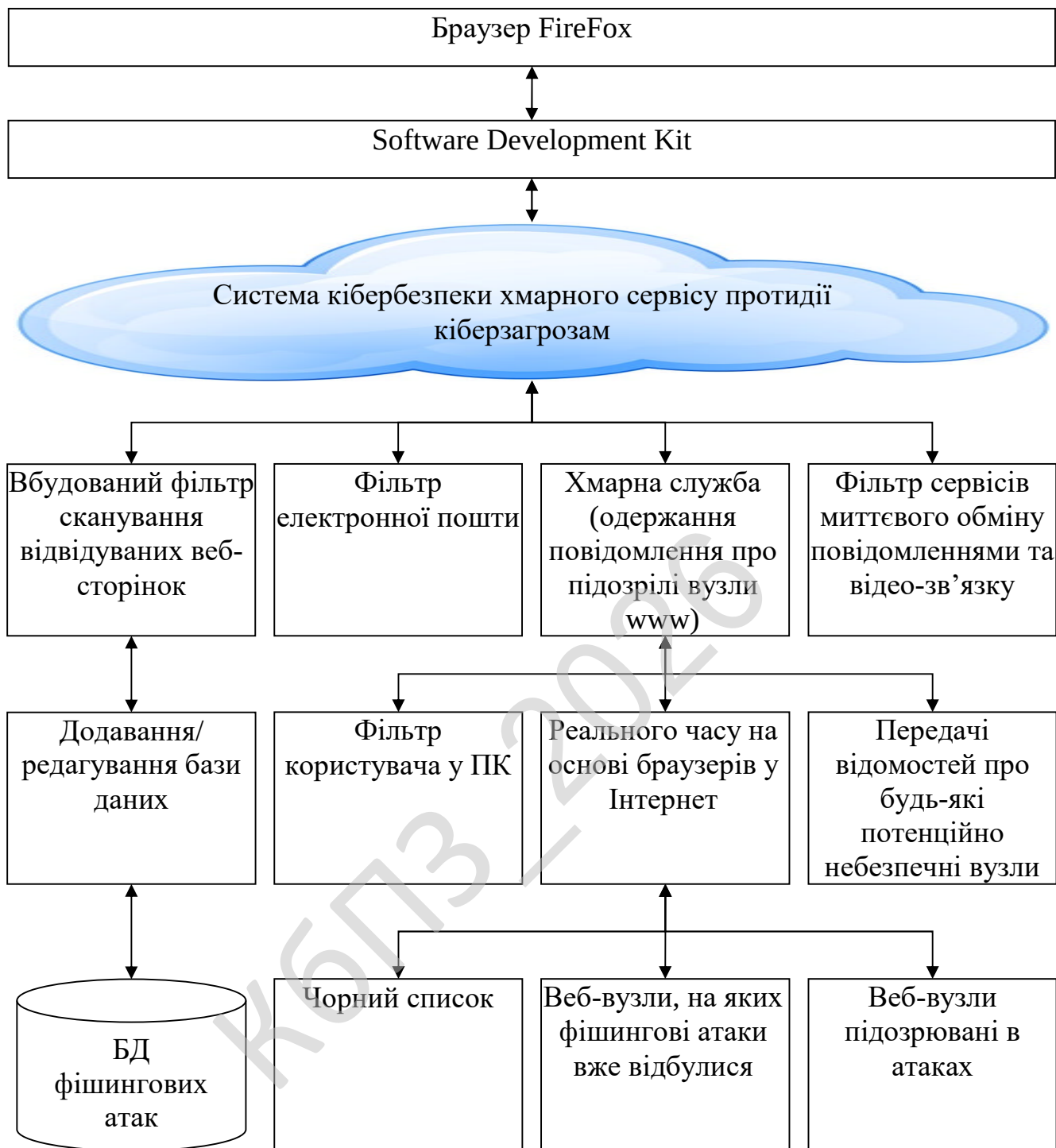


Рисунок 3.1 – Структурна схема системи

3.3 Функціональна схема роботи системи

На функціональній схемі, зображеній на рисунку 3.2, є можливість прослідкувати за шляхами проходження функціональних сигналів від одного функціонального блоку до іншого та побачити рівні надходження даних з мережі та міри їхньої обробки.



Рисунок 3.2 – Функціональна схема системи

Користувач використовуючи Інтернет браузер FireFox одержує доступ до глобальної мережі Інтернет та через розроблене програмне забезпечення системи кібербезпеки хмарного сервісу протидії кіберзагрозам взаємодіє з Інтернетом (Web-контент), розмовляє за допомогою Інтернет пейджерів (IRC, ICQ, AIM), користується електронною поштою.

Firefox є релізом наступного покоління веб-браузерів, який має безліч нагород, він містить у собі безліч спеціальних можливостей, що дозволяють зробити веб-браузер і веб-контент доступним для всіх користувачів.

Програмне забезпечення системи кібербезпеки хмарного сервісу протидії кіберзагрозам є системою розширення (extentions) можливостей FireFox надаючи користувачеві захист від фішингових атак.

Розглянувши усі блоки функціональної схеми перейдемо до розгляду діаграми взаємодії процесів, які відбуваються у системі.

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 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.

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).

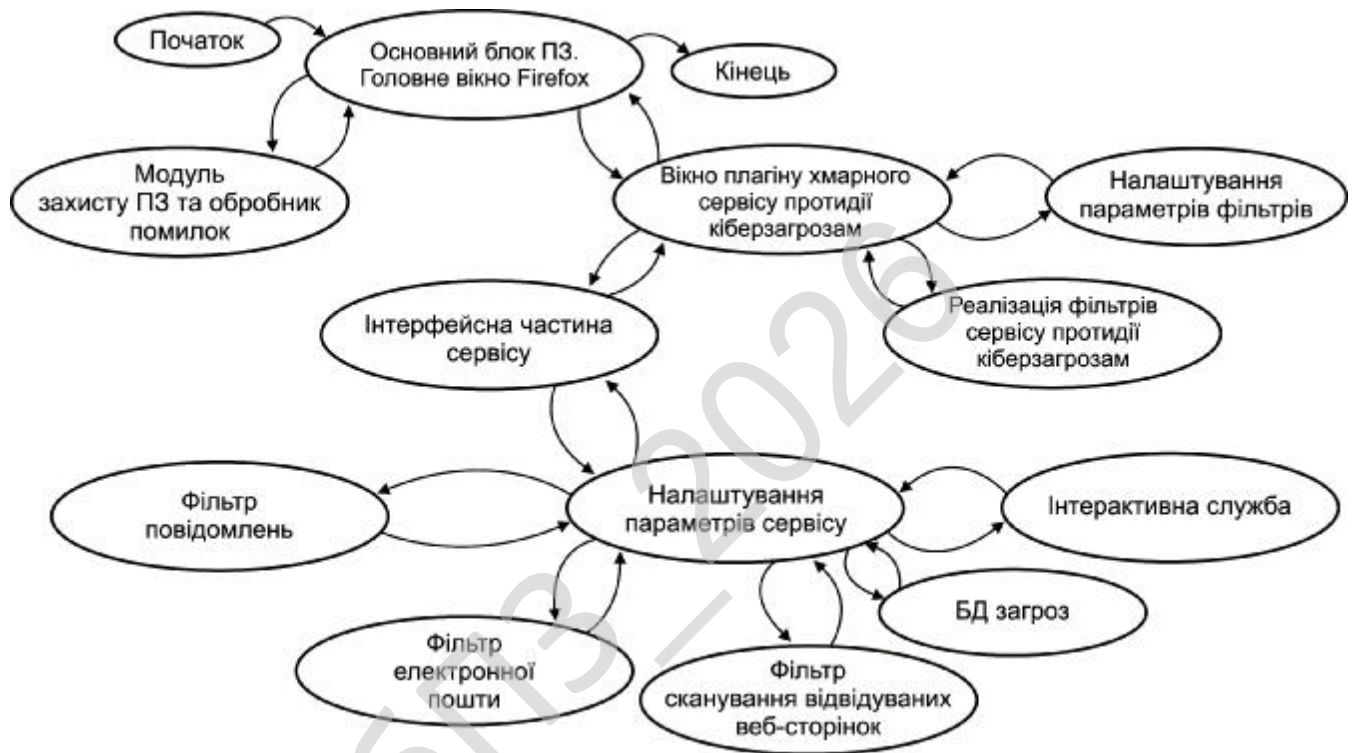


Рисунок 3.3 – Діаграма взаємодії процесів

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.

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. Перед їх розглядом необхідно провести роз'яснення який саме тип блок-схем використовується. 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.

Розглянемо проведені першочергові дії по створенню системи. Одним з найбільш корисних властивостей популярного браузера FireFox є система розширень (extentions). Завдяки розширенням, можливо одержати додаткову функціональність, відсутню в самому браузері в моєму випадку протидія кіберзагрозам.

Часто користувачі не знаходять потрібних розширень серед уже існуючих доступних у мережі Інтернет, незважаючи на готовність заплатити гроші за потрібне розширення, що породжує попит у різних розширеннях, саме через дані економічні міркування був обраний за основу браузер FireFox.

Розширення Firefox мають власну структуру файлів і щоб розширення правильно працювало, потрібно створити кореневий каталог, де будуть розміщатися всі файли розширення антифішингового захисту.

Створюються вкладені шаблонів директорії. Їхні імена повинні бути chrome (обов'язково в нижньому регістрі) і content (знову в нижньому регістрі). Таким чином, створюється файлова структура наступного виду:

```
<?xml version="1.0"?>
<RDF xmlns="http://www.w3.org/1999/02/22-rdf-syntax-
ns#"xmlns:em="http://www.mozilla.org/2004/em-rdf#">
  <Description about="urn:mozilla:install-manifest">
    <em:description>Servise extention</em:description>
    <em:homepageURL>http://www.adfs.ru/</em:homepageURL>
    <em:id>{65b3130e-8513-41b6-8ea8-43dbd9cc0b12}</em:id>
    <em:name>Servise</em:name>
    <em:version>1.0</em:version>
    <em:targetApplication>
      <Description>
        <em:id>{ec8030f7-c20a-464f-9b0e-13a3a9e97384}</em:id>
        <em:minVersion>1.0</em:minVersion>
        <em:maxVersion>1.0</em:maxVersion>
      </Description>
    </em:targetApplication>    <em:file>
      <Description about="urn:mozilla:extension:file: Servise.jar">
        <em:package>content</em:package>
      </Description>
    </em:file>    </Description>
  </RDF>
```

У перших рядках написано, що цей файл - ніщо інше, як стандартний XML-Файл. Для менеджера розширень надаються наступні дані.

```
<em:description>Servise extention</em:description>
  <em:homepageURL>
http://www.der4g.ru/</em:homepageURL>
  <em:id>{65b3130e-8513-41b6-8ea8-43dbd9cc0b12}</em:id>
  <em:name>Servise</em:name>
  <em:version>1.0</em:version>>
```

Перший рядок указує на авторські права коду, для чого використовується тег <em:creator>.

					ВКРБ-125.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

Наступний рядок - короткий опис того, а для чого, властиво, це розширення призначене. Ця інформація буде видна користувачам у менеджері розширень. Третій рядок указує на домашню сторінку розширення.

Четвертий рядок у цій секції, у тезі `<em:id>`, це найбільш важлива інформація у всьому файлі. Тут вказується унікальний ідентифікатор антифішингового розширення. Існує багато способів генерації унікальних ідентифікаторів.

Я використовував програму GUID Generator від Andy Hoskinson <http://www.hoskinson.net/webservices/guidgeneratorclient.aspx>.

Після створення унікального ідентифікатора я помістив його в тег `<em:id>`. У наступних рядках файлу описується, для якого додатка призначене це розширення:

```
<em:targetApplication>
  <Description>
    <em:id>{ec8030f7-c20a-464f-9b0e-13a3a9e97384}</em:id>
    <em:minVersion>1.0</em:minVersion>
    <em:maxVersion>1.0</em:maxVersion>
  </Description>
</em:targetApplication>
```

У моєму випадку в тезі `<em:name>` зазначений унікальний код браузера Firefox і його значення не повинне змінюватися. Тут же я встановив два тега для вказівки мінімальної й максимальної версії додатка, з якими дане розширення сумісне.

У моєму випадку ці два значення збігаються. Поточна версія Firefox дотримується наступної угоди по версіях: `major.minor.release.build[+]` (Поля, що впливають за *major* – необов'язкові).

У кінцевій частині опису файлу я вказав розташування JAR-Файлу (розглянутий нижче):

```
<em:file>
  <Description about="urn:mozilla:extension:file: Servise.jar">
  <em:package>content</em:package>
</Description>
</em:file>
```

					ВКРБ-125.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

В атрибуті about тегу <Description> вказується ім'я. У моєму випадку це посилання на файл "Servise.jar".

Для зручності, я назвав ім'я файлу співпадаюче з ім'ям розширення. В елементі <em:package> вказується розташування JAR-Файлу. Тому що в нас у директорії *chrome* є тільки один фолдер *content*, те його він і вказується.

```
<?xml version="1.0"?>
<RDF:RDF xmlns:RDF="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:chrome="http://www.mozilla.org/rdf/chrome#">
  <RDF:Seq about="urn:mozilla:package:root">
    <RDF:li resource="urn:mozilla:package:Servise"/>
  </RDF:Seq>
  <RDF:Description about="urn:mozilla:package:Servise" chrome:extension="true"
chrome:name="Servise"/>
  <RDF:Seq about="urn:mozilla:overlays">
    <RDF:li resource="chrome://browser/content/browser.xul"/>
  </RDF:Seq>
  <RDF:Seq about="chrome://browser/content/browser.xul">
    <RDF:li>chrome://Servise/content/Servise-Overlay.xul</RDF:li>
  </RDF:Seq>
</RDF:RDF>
```

Перші строки:

```
<RDF:Seq about="urn:mozilla:package:root">
  <RDF:li resource="urn:mozilla:package:Servise"/>
</RDF:Seq>
```

Указую назву пакета в першому атрибуті. У другому атрибуті повідомляємо, що так, цей додаток є розширенням протидії кіберзагрозам. І, нарешті, вказування ім'я нашого пакета. Необхідно щоб значення атрибута *chrome:name* повинне бути зазначене в нижньому регістрі.

Всі видимі елементи в браузері FireFox створені за допомогою framework XUL (визначення розглянуте в раніше) і всі разом вони називаються *chrome*.

У наступному фрагменті коду дипломного проекту вказується, який файл *chrome* наше розширення буде "розширювати". У більшості випадків розширюється тільки файл *brouser.xul*, хоча є й інші.

У випадку розробленої протидії кіберзагрозам додатку все просто:

```
<RDF:Seq about="urn:mozilla:overlays">
```

					ВКРБ-125.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

```
<RDF:li resource="chrome://browser/content/browser.xul"/>
</RDF:Seq>
```

Для розробки користувацького інтерфейсу розширень Firefox використовується мова розмітки XUL, що я вже згадував ледве вище. Краса XUL виражається в тім, що називається "динамічні оверлеї" (dynamic overlays). Вони дозволяють змінювати поведження інтерфейсу без необхідності зміни оригінального коду.

Для початку, я створив створив у меню Tools пункт "Servise". Для цього створив overlay-файл із назвою, зазначеною в тезі <RDF:li>. Він розташовується в тій же директорії що й *contents.rdf*.

У файлі Servise-Overlay.xul я вказав наступні дані:

```
<?xml version="1.0"?>
<overlay id="ServiseOverlay"
xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">
<menupopup id="menu_ToolsPopup">
<menuitem label="Servise" position="1" />
<menuitem label=" About" position="2" />
. . . . .
</menupopup>
</overlay>
```

Блок-схеми показують весь процес роботи системи з підсистемами та частково доказують правильність вибраних проектних рішень. Тому від точності і детальної блок-схеми залежить результат всієї програми.

При виборі початкової точки відліку при побудові схем було враховано, що виходячи з вибору мови програмування і інших технічних засобів, програма буде об'єктно-орієнтована що вимагає високого рівня декомпозиції задач на класи.

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

					ВКРБ-125.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

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.

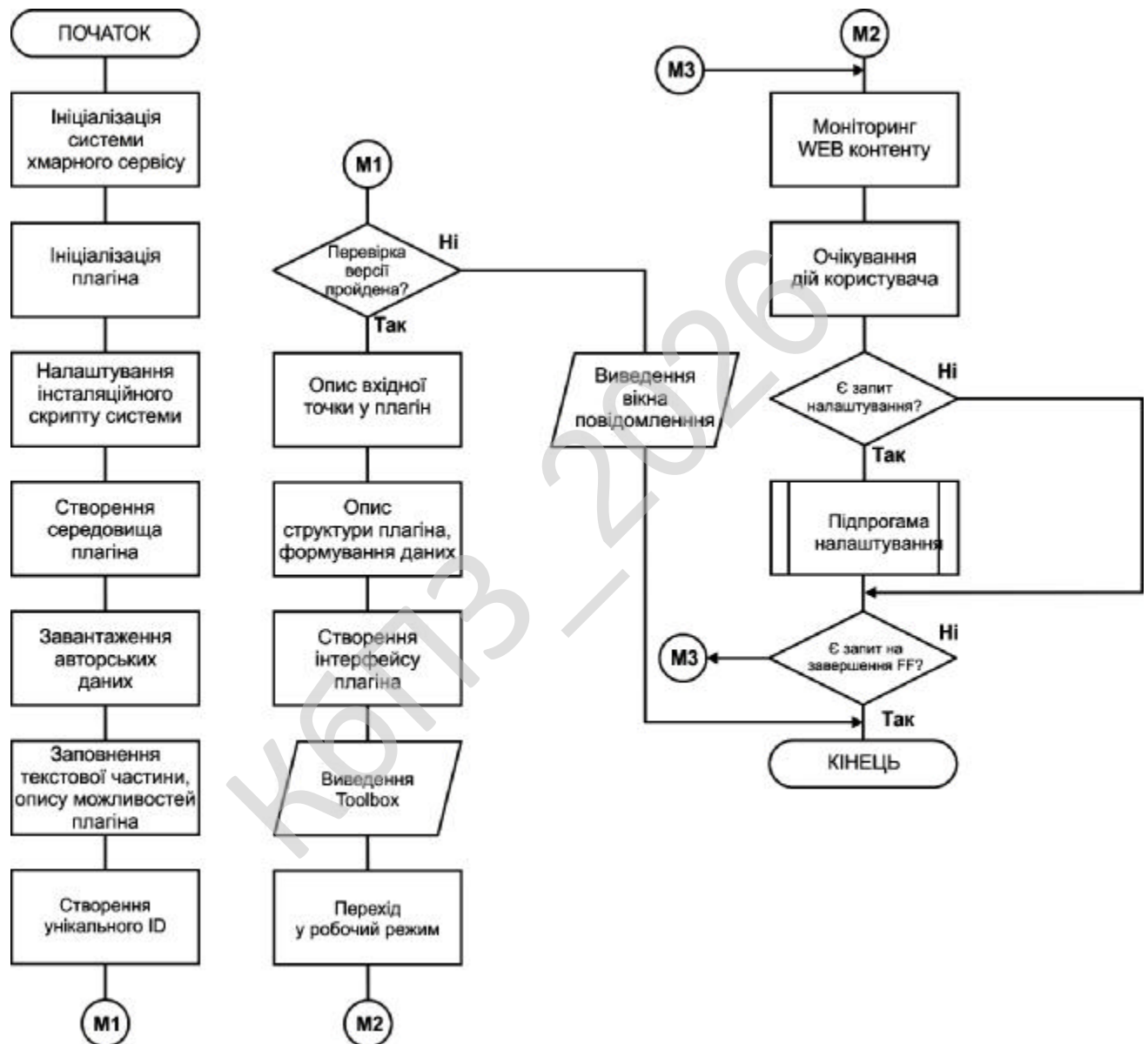


Рисунок 4.1 – Блок-схема основної програми

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.

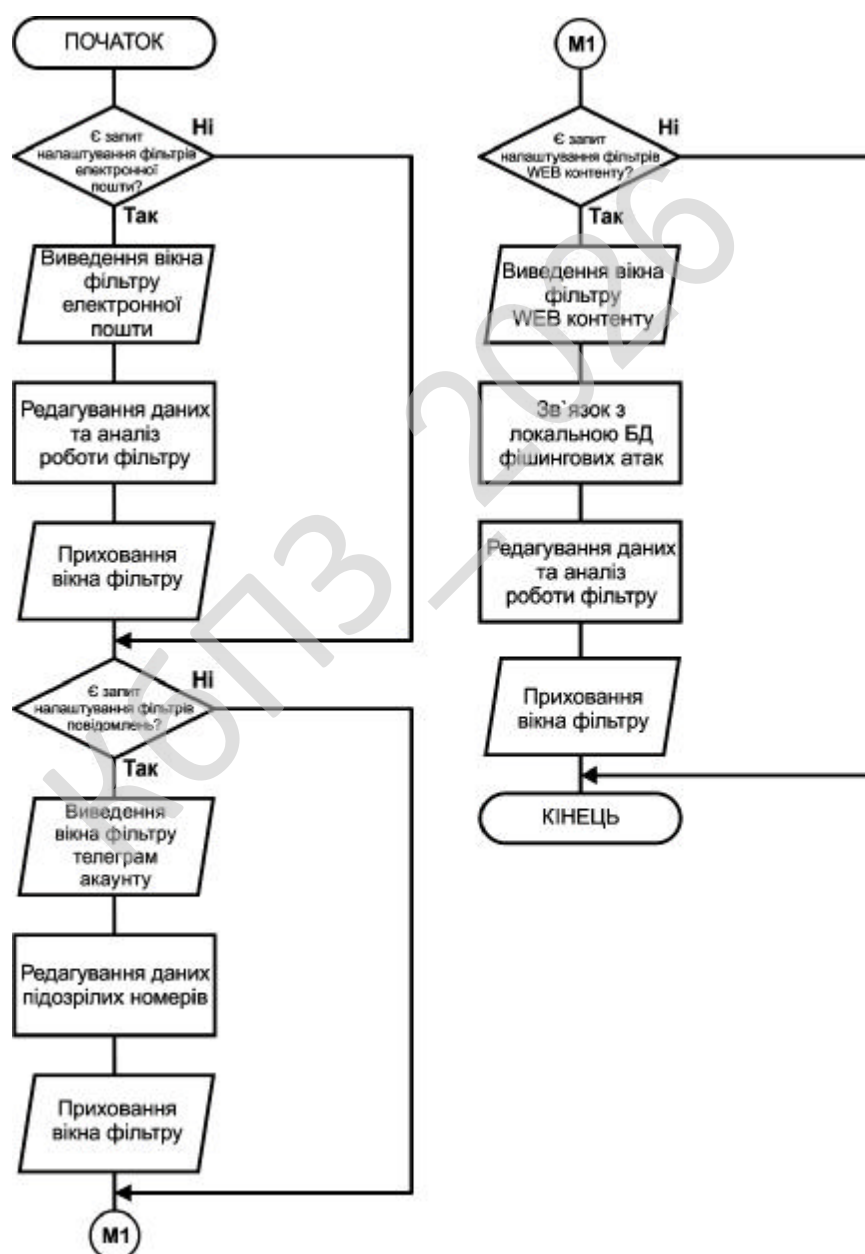


Рисунок 4.2 – Блок-схема роботи підпрограми

Було використано наступні підходи UML: Діаграма об'єктів; Діаграма розгортання.

Діаграма об'єктів в UML це діаграма, що відображає об'єкти та їх зв'язки в певний момент часу. Діаграма об'єктів може розглядатись як окремий випадок діаграми класів, на якій можуть бути представлені як класи, так і екземпляри (об'єкти) класів. Схожою за змістом є діаграма взаємодії (collaboration diagram).

Діаграми об'єктів не мають власної нотації. Оскільки діаграми класів можуть відображати об'єкти, то діаграма класів, на якій відображено лише об'єкти, та не відображено класи, може вважатись діаграмою об'єктів.

Діаграма об'єктів відображає об'єкти та зв'язки в певний момент роботи програми. Об'єкти можуть містити інформацію про власні значення а не про описання. Для відображення загальних шаблонів об'єктів та зв'язків, що можуть багаторазово створюватись під час роботи програми, слід використовувати діаграму взаємодії, яка може відображати характеристики об'єктів та зв'язків. Екземпляр діаграми взаємодії створює діаграму об'єктів.

Діаграма об'єктів не відображає еволюцію системи під час роботи. Натомість, слід використовувати діаграми взаємодії з повідомленнями, або діаграми послідовності.

Діаграма розгортання (deployment diagram) це діаграма в UML, на якій відображаються обчислювальні вузли під час роботи програми, компоненти, та об'єкти, що виконуються на цих вузлах. Компоненти відповідають представленню робочих екземплярів одиниць коду. Компоненти, що не мають представлення під час роботи програми на таких діаграмах не відображаються; натомість, їх можна відобразити на діаграмах компонент. Діаграма розгортання відображає робочі екземпляри компонент, а діаграма компонент, натомість, відображає зв'язки між типами компонент.

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

					ВКРБ-125.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

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-125.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

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": "А",
  "lastName": "Б",
  "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.

					ВКРБ-125.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

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-125.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

регулярних виразів. У RFC, що визначає JSON пропонується використовувати такий код для перевірки його відповідності формату JSON

```
const my_JSON_object = !(/[^\s,:{}\[\]0-9.\-+Eaeflnr-u \n\r\t]/.test(
text.replace(/"(\\"|.)*"/g, ''))) &&
eval('(' + testedData + ')');
```

Як безпечніша альтернатива eval() була запропонована нова функція parseJSON(), здатна обробляти тільки JSON-дані.

Вбудований JSON

Останні версії веб-браузерів мають вбудовану підтримку JSON і здатні його обробляти без виклику функції eval(), що призводить до описаної проблеми. Обробка JSON у такому разі зазвичай здійснюється швидше. Так у червні 2009 року вбудовану підтримку JSON мали такі браузери: Mozilla Firefox 3.5+; SeaMonkey 2; Thunderbird 3; Microsoft Internet Explorer 8; Opera 10.5; Браузери, засновані на WebKit (наприклад, Google Chrome, Apple Safari).

Принаймні дві популярні бібліотеки JavaScript використовують вбудований JSON у разі його доступності: jQuery; Dojo.

Підробка крос-доменного запиту

Непродумане використання JSON робить сайти вразливими до підробки міжсайтових запитів (CSRF або XSRF). Оскільки тег <script> допускає використання джерела, що не належить до того ж домену, що і використовуваний ресурс, це дозволяє виконувати код даних, представлених у форматі JSON, в контексті довільної сторінки, що робить можливою компрометацію паролів або іншої конфіденційної інформації користувачів, що пройшли авторизацію на іншому сайті.

Це є проблемою тільки у разі вмісту в JSON-даних конфіденційної інформації, яка може бути компрометована третьою стороною і якщо сервер розраховує на політику одного джерела, блокуючи доступ до даних при виявленні зовнішнього запиту. Це не є проблемою, якщо сервер визначає допустимість запиту, надаючи дані тільки у разі його коректності.

					ВКРБ-125.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

HTTP cookie не можна використовувати для визначення цього. Виключне використання HTTP cookie використовується підрубкою 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 веб-сервісів.

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-125.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

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

Для захисту розробленого програмного забезпечення запропоновано використовувати алгоритм ДСТУ 8845:2019 – алгоритм симетричного потокового перетворення. В основі ДСТУ 8845:2019 лежить класична схема підсумовуючого генератора подібна генератору SNOW 2.0, SNOW 3.0 та SNOW V. В основі ДСТУ 8845:2019 збережені всі базові операції шифрів сімейства SNOW, а раунд шифрування AES замінений на функцію нелінійної підстановки T, що реалізовує перестановку елементів скінченного поля $GF(2^{64})$ за допомогою компонентів національного стандарту симетричного криптоперетворення ДСТУ 7624:2014.

ДСТУ 8845:2019 використовує 256-бітний вектор ініціалізації IV та 256-бітний або 512-бітний секретний ключ K і забезпечує високий та надвисокий рівень стійкості із врахуванням можливого застосування квантового криптографічного аналізу. При розробці алгоритму ДСТУ 8845:2019 орієнтувалися на сучасні 64-бітні обчислювальні системи, тому розмір слова обрано рівним 8 байт. запису байтів застосовують подання від старшого до молодшого. Генератор ключових потоків ДСТУ 8845:2019 у режимі генерації гами шифру схематично приведено у стандарті.

Як впливає з генератора ключових потоків ДСТУ 8845:2019 у режимі генерації гами шифру основними компонентами генератору є регістр зсуву з лінійним зворотнім зв'язком та скінчений автомат на базі якого виконується нелінійне перетворення T. Вхідні дані (ключ шифрування K та вектор ініціалізації IV) використовуються для ініціалізації змінної стану $S_i (i \geq 0)$, яка складається із двох компонент до складу яких входять [12]:

					ВКРБ-125.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

- 16 змінних $s^{(i)}$ – комірок регістра зсуву з лінійним зворотнім зв'язком: $s^{(i)} = (s_{15}^{(i)}, s_{14}^{(i)}, s_{13}^{(i)}, s_{12}^{(i)}, s_{11}^{(i)}, s_{10}^{(i)}, s_9^{(i)}, s_8^{(i)}, s_7^{(i)}, s_6^{(i)}, s_5^{(i)}, s_4^{(i)}, s_3^{(i)}, s_2^{(i)}, s_1^{(i)}, s_0^{(i)})$;
- Два регістри скінченного автомату $r^{(i)} : r^{(i)} = (r_2^{(i)}, r_1^{(i)})$. На виході отримуємо ключовий потік (гамма), який формується з 8-байтних слів Z_i .

З рисунку слідує, що відводи регістра зсуву з лінійним оберненим зв'язком побудовані за примітивним над полем $GF(2^{64})$ поліномом: $f(x) = x^{16} + x^{13} + \alpha^{-1}x^{11} + \alpha$, де α є коренем примітивного над полем $GF(2^8)$ поліному $gz(z) = z^8 + \beta^{170}z^7 + \beta^{166}z^6 + \beta^2z^5 + \beta^{224}z^4 + \beta^{70}z^3 + \beta^2$. Поле $GF(2^8)$ як і в ДСТУ 7624:2014 побудовано за примітивним на полем $GF(2)$ поліномом $p(y) = y^8 + y^4 + y^3 + y^2 + 1$, а коефіцієнти $g(z)$ подаються через ступінь примітивного елементу β поля $GF(2^8)$, тобто β корінь поліному $p(y)$. Тобто, у нас є вежа полів: $GF(2) \subset GF(2^8) \subset GF(2^{64}) \subset GF(2^{1024})$, де:

- поле $GF(2^{1024})$ задається відводами зворотного зв'язку як фактор кільце $GF(2^{64})[x]/(f(x))$;
- поле $GF(2^{16424})$ задається як фактор кільце $GF(2^8)[z]/(g(z))$;
- поле $GF(2^{1024})$ задається як фактор кільце $GF(2)[y]/(p(y))$.

З вищезазначеного, слідує що період вихідної послідовності становить 2^{1024} .

Структурно в алгоритмі симетричного потокового перетворення ДСТУ 8845:2019 виділяють три основні функції:

- функція ініціалізації, яка приймає в якості вхідних даних 256-бітний вектор ініціалізації IV та 256-бітний або 512-бітний секретний ключ K, і виробляє початкове значення змінної стану $S_0 = (s^{(0)}, r^{(0)})$;
- функція наступного стану Next, яка приймає на вхід зміну стану $S_i = (s^{(i)}, r^{(i)})$ та виробляє наступне значення змінної стану $S_{i+1} = (s^{(i+1)}, r^{(i+1)})$;
- функція ключового потоку Strm, що приймає на вході змінну стану $S_i = (s^{(i)}, r^{(i)})$ та виробляє на виході 64-бітний ключовий потік Z^i .

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ КІБЕРБЕЗПЕКИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено розроблене у бакалаврської дипломної роботі програмне забезпечення системи кібербезпеки хмарного сервісу протидії кіберзагрозам. 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.
- Розділу обрання групи.
- Розділу виведення результату роботи системи.
- Функції представлені у графічному вигляді (іконки).

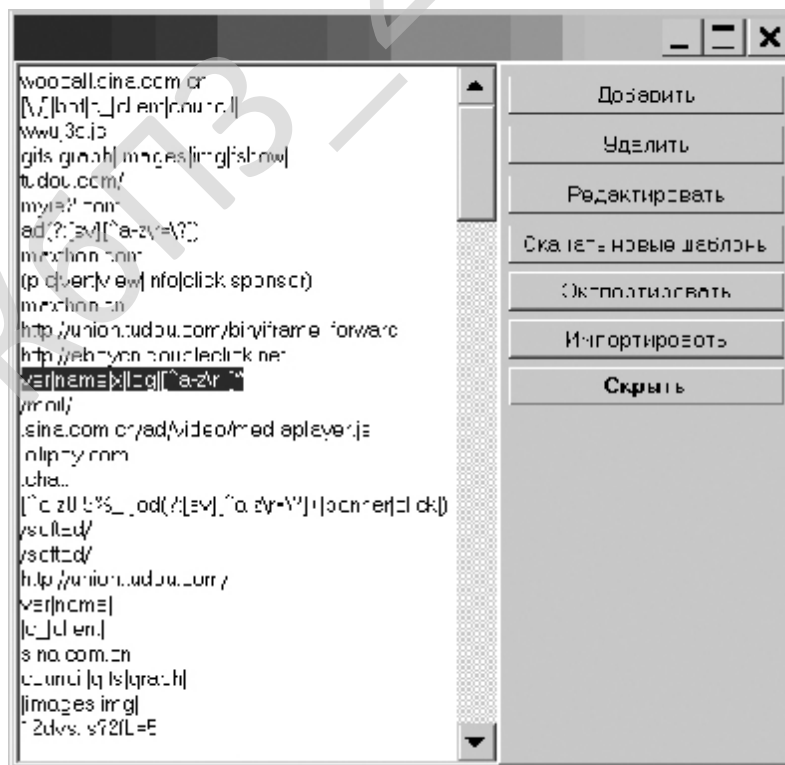


Рисунок – Головне вікно ПЗ

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.

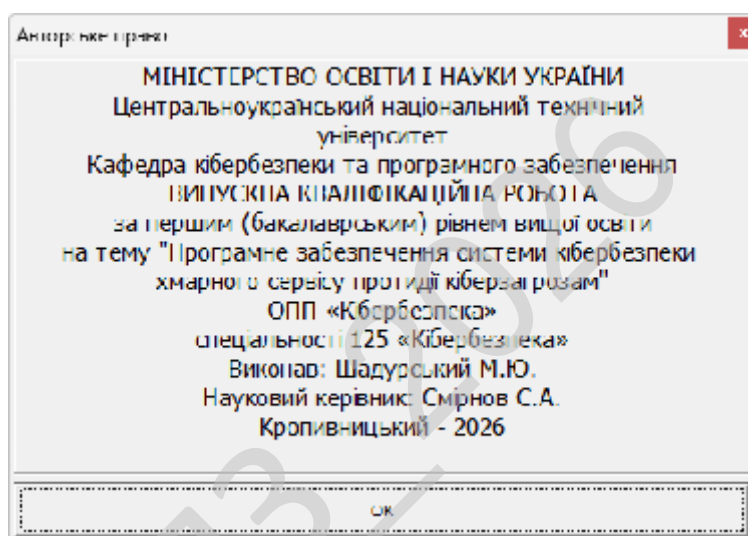


Рисунок 5.2 – Авторське право

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

					ВКРБ-125.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

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.
- Оновлення.
- 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

					ВКРБ-125.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

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.

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 форматом білої скриньки засноване на аналізі керуючої структури програми. Програма вважається повністю перевіреною, якщо проведено вичерпне тестування маршрутів (шляхів) її графа управління.

У цьому випадку формуються тестові варіанти, в яких:

- Гарантується перевірка всіх незалежних маршрутів програми.
- Знаходяться гілки True, False для всіх логічних рішень.
- Виконуються всі цикли (у межах їхніх кордонів та діапазонів).
- Аналізується правильність внутрішніх структур даних.

Недоліки тестування "білої скриньки":

- Кількість незалежних маршрутів може бути дуже велика.
- Повне тестування маршрутів не гарантує відповідності програми вихідним вимогам до неї.

- У програмі можуть бути пропущені деякі маршрути.

- Не можна виявити помилки, поява яких залежить від даних.

Переваги тестування "білої скриньки" пов'язані з тим, що принцип «білої скриньки» дозволяє врахувати особливості програмних помилок:

- Кількість помилок мінімально в «центрі» і максимально на «периферії» програми.

- Попередні припущення про ймовірність потоку керування або даних у програмі часто бувають некоректними. У результаті типовим може стати маршрут, модель обчислень за яким опрацьована слабо.

- При записі алгоритму програмного забезпечення у вигляді тексту на мові програмування можливе внесення типових помилок трансляції (синтаксичних та семантичних).

- Деякі результати в програмі залежать не від вихідних даних, а від внутрішніх станів програми.

Обрано умови розповсюдження – proprietary software.

Програмне забезпечення, на яке зберігаються як немайнові, так і майнові авторські права. Отримавши або придбавши таке програмне забезпечення, користувач отримує обмежені права користування ним: може бути заборонено або закрито доступ до коду (вивчення), внесення змін, тиражування, розповсюдження та перепродаж. Програмне забезпечення вважається власницьким, якщо наявне хоча б одне з перелічених обмежень.

Найчастіше основним методом захисту майнових прав на власницьке ПЗ, поза ліцензійною угодою, власник обирає закриття сирцевого коду, захищаючи свій продукт від модифікації і вбудовуючи системи обмеження користування через авторизацію. Таке програмне забезпечення називається закритим. Проте,

					ВКРБ-125.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

код власницького продукту може бути і відкритим, але власник може обмежити права користувача умовами користувацької ліцензії.

Власницьке програмне забезпечення та комерційне програмне забезпечення не є синонімами – власницьким може бути і безплатне (тобто, некомерційне) програмне забезпечення.

На противагу власницькому ПЗ існує вільне програмне забезпечення, автори і власники якого дозволяють вивчати, модифікувати і поширювати свій продукт. Саме визначення власницького програмного забезпечення виникло в результаті діяльності громадського руху вільного програмного забезпечення (представленого Фондом вільного програмного забезпечення та іншими організаціями) і осмислення умов свободи користування програмами. Визначенням власницького програмного забезпечення є не невідповідність хоча б одній з базових умов вільного програмного забезпечення. Сама назва власницьке ПЗ підкреслює визначальне значення власника у способі використання і можливостях розвитку цього програмного забезпечення.

					ВКРБ-125.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

6 ОСНОВНІ ВИСНОВКИ

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

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

Рішення завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем хмарного сервісу протидії кіберзагрозам.
- Досліджена система хмарного сервісу протидії кіберзагрозам.
- На основі отриманих результатів досліджень створена програмна реалізація системи кібербезпеки хмарного сервісу протидії кіберзагрозам.

Розроблені під час виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання хмарного сервісу протидії кіберзагрозам.

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 хмарного сервісу протидії кіберзагрозам. 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-125.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

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 8845:2019.

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.

КБПЗ-2026

					ВКРБ-125.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Derek Fisher. Application Security Program Handbook. Manning Publications. 2021. 155 p.
2. Cameron Wyatt PH.D. Kali Linux Tutorial. Independently published. 2021. 60 p.
3. Kuznetsov O., Frontoni E., Kuznetsova Y., Chevardin V., Smirnov O. «Architectural foundations for adaptive security in edge computing systems». *Cybersecurity Defensive Walls in Edge Computing*, 2025. pp. 21-61.
4. Петрик В.М., Присяжнюк М.М., Аль-Файюмі Халед та ін. «Системи інформаційної зброї та технології інформаційної війни»: підручник / Петрик В.М., Присяжнюк М.М., Аль-Файюмі Халед, Жарков Я.М., Смірнов О.А, Буравченко К.О., Давидюк А.В., Кононович В.Г., Корчинский В.В., Кудирко В.М., Фесенко А.О.; за заг. ред. В.М. Петрика, М.М. Присяжнюка.– К.: Видавничий центр “Кафедра”, 2025.– 320 с.
5. Усік, П.С., Смірнова, Т.В., Буравченко, К.О., Смірнов, О.А., Улічев, О.С., Смірнов, С.А. «Дослідження технологій забезпечення кібербезпеки банківських систем з використанням штучного інтелекту». *Кібербезпека: освіта, наука, техніка*. 2025. Том 1 № 29. С.704–716, 2025
6. Kuznetsov, O., Frontoni, E., Kuznetsova, K., Arnesano, M., Smirnov, O. «A secure biometric authentication architecture for blockchain-driven cyber-physical systems». *Security and Privacy of Cyber Physical Systems Emerging Trends Technologies and Applications*, 2025, pp. 193–224.
7. 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.

8. 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.

9. 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.

10. 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.

11. 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.

12. 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.

13. 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.

14. 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.

					ВКРБ-125.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

15. 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.
16. Батрак О., Смірнова Т., Гнатюк В., Одарченко Р., Смірнов О. «Дослідження показників ефективності функціонування та перспектив розвитку систем IP-телефонії». *Підводні технології*, 2024, № 13, с. 28-35.
17. 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.
18. Akhalaia, G., Iavich, M., Iashvili, G., Prysiashnyy, D., Smirnova, T. «Secure Encrypted Connection on Georgian Website». *CEUR Workshop Proceedings*, 2023, 3550, pp. 313-320.
19. 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
20. 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.
21. 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.
22. 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.

23. 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.

24. 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.

25. 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.

26. 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.

27. Вінтенко Б.Ю., Смірнов О.А., Коваленко О.В., Смірнов С.А., Коваленко А.С. «Дослідження нормативних документів та галузових 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.

28. 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,

29. 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.

30. 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.

31. 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.

32. 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.

33. 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.

34. 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

35. Smirnov O., Kuznetsov A., Lokotkova I., Kuznetsova T., Florov S., Lebid O. «Using Orthogonal Signals to Hide Information in Images». *4 IEEE*

					ВКРБ-125.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

International Conference on Advanced Information and Communication Technologies (AICT) - 2021, Lviv, Ukraine, September 21-25, 2021. P. 255-260.

36. 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.*

37. 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.*

38. 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.*

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

40. 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.*

41. 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.*

42. 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.*

43. 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.*

44. 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.
45. 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.
46. 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.
47. 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.
48. 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.
49. 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.
50. 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.

51. 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.

52. 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.

53. 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.

54. 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.

55. 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.