

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

“ ____ ” _____ 2021 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему

**“Дослідження та програмна реалізація системи кібербезпеки
для блокування вхідних вузлів Tor і серверів VPN-провайдерів”**

Виконав здобувач вищої освіти
II курсу, групи КІ-20М-1,4
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Колодяжний І.О.
« ____ » _____ 2021 р.

Керівник проекту
кандидат технічних наук, доцент
_____ Олександр ДОРЕНСЬКИЙ
« ____ » _____ 2021 р.

Рецензент _____

Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Рівень вищої освіти магістр
Галузь знань . 12 “Інформаційні технології”
Спеціальність 123 “Комп’ютерна інженерія”
Освітньо-професійна (освітньо-наукова) програма “Комп’ютерна інженерія”

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.т.н., проф.

Олексій СМІРНОВ
« 6 » вересня 2021 року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ДРУГИМ (МАГІСТЕРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Колодяжному Ігорю Олеговичу

(прізвище, ім'я, по батькові)

1. Тема роботи *Дослідження та програмна реалізація системи кібербезпеки для блокування вхідних вузлів Tor і серверів VPN-провайдерів*

2. Керівник роботи *Доренський Олександр Павлович, канд. техн. наук*

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 42-13 від 02.08.2021 року

3. Строк подання студентом роботи до захисту 10.12.2021 р.

4. Мета та завдання випускної кваліфікаційної роботи: *Метою розробки є дослідження та програмна реалізація системи кібербезпеки для блокування вхідних вузлів Tor і серверів VPN-провайдерів*

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

<i>1. Призначення та область використання.</i>	<i>7. Економічна ефективність розробленої програми.</i>
<i>2. Перегляд аналогічних існуючих систем.</i>	<i>8. Заходи з охорони праці та техніки безпеки</i>
<i>3. Опис і обґрунтування проектних рішень.</i>	<i>9. Висновки.</i>
<i>4. Етапи програмування системи.</i>	
<i>5. Впровадження системи в промислову експлуатацію</i>	
<i>6. Наукова новизна</i>	
6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)	
<i>Наукова новизна</i>	<i>1 аркуш</i>
<i>Структурна схема системи</i>	<i>1 аркуш</i>
<i>Функціональна схема системи</i>	<i>1 аркуш</i>
<i>Діаграма процесів</i>	<i>1 аркуш</i>
<i>Блок-схема алгоритму роботи додатку</i>	<i>2 аркуша</i>
<i>Показники економічної ефективності</i>	<i>1 аркуш</i>

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Савеленко Г.В.	05.10.2021	14.11.2021
Охорона праці	Оришака О.В.	06.10.2021	16.11.2021

7. Дата видачі завдання « 6 » вересня 2021 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.10.2021 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2021 р.	
3.	Розробка моделі компонента	20.10.2021 р.	
4.	Розробка структур даних	25.10.2021 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.10.2021 р.	
6.	Програмування алгоритмів	10.11.2021 р.	
7.	Розрахунок економічної ефективності	13.11.2021 р.	
8.	Розрахунки з охорони праці та техніки безпеки	15.11.2021 р.	
9.	Оформлення ПЗ	17.11.2021 р.	
10.	Попередній захист роботи	10.12.2021 р.	

Дата видачі завдання
« 6 » вересня 2021 р.

Підпис керівника

_____ (прізвище та ініціали)

Завдання прийнято до виконання
« 6 » вересня 2021 р.

Підпис здобувача

_____ (прізвище та ініціали)

АНОТАЦІЯ

Колодяжний І.О. Дослідження та програмна реалізація системи кібербезпеки для блокування вхідних вузлів Tor і серверів VPN - провайдерів. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2021.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи кібербезпеки для блокування вхідних вузлів Tor і серверів VPN-провайдерів.

Метою розробки є дослідження та програмна реалізація системи кібербезпеки для блокування вхідних вузлів Tor і серверів VPN-провайдерів.

Об'єктом дослідження є процес кібербезпеки для блокування вхідних вузлів Tor і серверів VPN-провайдерів.

Предметом дослідження є методи кібербезпеки для блокування вхідних вузлів Tor і серверів VPN-провайдерів.

Методи дослідження базуються на методах теорії захисту інформації, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи кібербезпеки для блокування вхідних вузлів Tor і серверів VPN-провайдерів.

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 архітектури IBM PC з ОС Windows XP/Vista/7/8/10.

Програму розроблено в середовищі Python, SQLite.

Ключові слова: комп'ютерна інженерія, Tor, VPN

ABSTRACT

Kolodiazhnyi I.O. Research and software implementation of cybersecurity system for blocking Tor input nodes and VPN providers' servers. 123 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2021

In this final qualification work for the second (master's) level of higher education, software has been developed that is designed for a cybersecurity system to block Tor input nodes and VPN provider servers.

The purpose of the development is to research and software implementation of a cybersecurity system to block Tor input nodes and VPN provider servers.

The object of the research is the cybersecurity process for blocking Tor input nodes and VPN provider servers.

The subject of the research is cybersecurity methods for blocking Tor input nodes and VPN provider servers.

Research methods are based on methods of information security theory, methods of mathematical statistics, methods of software development.

The result is a software implementation of a cybersecurity system to block Tor input nodes and VPN provider servers.

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.

Developed user-friendly interface. Instructions for working with software are given.

The program can be used on an IBM PC with Windows XP / Vista / 7/8/10.

The program is developed in Python, SQLite.

Keywords: computer engineering, Tor, VPN

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП.....	5
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ.....	11
1.1 Призначення системи.....	11
1.2 Область застосування.....	12
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	17
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	17
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	39
2.3 Розгорнута постановка завдання	47
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	49
3.1 Опис функціонування системи.....	49
3.2 Розробка структурної схеми	49
3.3 Розробка функціональної схеми.....	50
3.4 Розробка діаграми процесів.....	52
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ	55
4.1 Розробка блок-схем та опис алгоритмів функціонування системи	55
4.2 Захист розробленого програмного забезпечення	58
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ.....	60
6 НАУКОВА НОВИЗНА	66

БКРМ-123.21.0010.00.00.ПЗ

Вим.	Арк.	№ докум.	Підп.	Дата				
<i>Розроб.</i>		Колодяжний І.О.			<i>Дослідження та програмна реалізація системи кібербезпеки для блокування вхідних вузлів Tor і серверів VPN-провайдерів</i>	<i>Лім.</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Перев.</i>		Доренський О.П.				М	1	106
<i>Н.контр.</i>		Гермак В.С.			ЦНТУ КІ-20М-1,4			
<i>Затв.</i>		Смірнов О.А.						

7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ.....	67
7.1 Техніко економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.	67
7.2 Розрахунок трудомісткості розробки програмної продукції	69
7.3 Визначення чисельності виконавців і планового фонду зарплати	71
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника	76
7.5 Визначення собівартості розробки та ціни програмної продукції.	80
7.6 Визначення об'єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції.....	83
7.7 Визначення експлуатаційних витрат.....	83
7.8 Визначення економічної ефективності програмної продукції.....	85
7.9 Висновок.	87
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	88
8.1 Вступ	88
8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером	89
8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста ...	90
8.4 Розробка заходів з умов поліпшення охорони праці.....	93
8.5 Розрахункова частина	94
8.6 Висновки до розділу.....	96
9 ОСНОВНІ ВИСНОВКИ.....	97
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	99

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ООП – об'єктно-орієнтоване програмування;
СБУ – служба безпеки України;
Л/ДНР – терористична організація;
РФ – російська федерація;
СБ – служба безпеки;
ПЗ – програмне забезпечення;
СКС – структурована кабельна система;
БД – база даних;
СУБД – система управління базами даних;
ООСУБД – об'єктно-орієнтована система управління базами даних;
ЕОМ – комп'ютер;
ПК – персональний комп'ютер;
VPN – віртуальна приватна мережа;
Тор – система, що забезпечують анонімність в мережі Інтернет;
DNS – доменна система імен;
DES – алгоритм блочного шифрування;
UDP – протокол датаграм користувача;
TCP – протокол керування передачею;
PyQt5 – бібліотека Python для розробки графічного інтерфейсу;
GUI – графічний інтерфейс користувача;
JSON – текстовий формат обміну даними між комп'ютерами;
FLOSS – вільне та відкрите програмне забезпечення;
JVM – віртуальна машина Java;
FreeBSD – операційна система;
Macintosh – низка комп'ютерів від Apple;
Solaris – операційна система;

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

OS/2 – операційна система;

Аміга – сімейство домашніх персональних комп'ютерів і операційних систем до них;

AROS – проект по створенню кросплатформної та вільної операційної системи (ОС);

AS/400 – серія серверів IBM;

BeOS – операційна система;

OS / 390 – операційна система;

z/OS – операційна система;

Palm OS – операційна система;

QNX – операційна система;

VMS – операційна система;

Psion – англійська компанія;

Acorn RISC OS – операційна система;

VxWorks – операційна система;

PlayStation – домашня гральна консоль;

Sharp Zaurus – персональний цифровий помічник;

XML – мова програмування;

XML-RPC – протокол виклику віддалених процедур;

HTML – мова розмітки гіпертексту;

BAV – формат файлів.

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

ВСТУП

Актуальність теми. Суттєвий прогрес і поширення інформаційних технологій, глобальний характер систем масової комунікації призвели до утворення глобального інформаційного простору, який змушує світову спільноту, кожна державу швидко орієнтуватися та адаптуватися у сучасному інформаційному середовищі. Світове співтовариство в цих умовах усвідомило, що міжнародна інформаційна безпека є проблемою, розв'язання якої суттєво впливає на існування людства. Тобто з розвитком і поширенням інформаційно-комунікаційних технологій у всі сфери життєдіяльності надзвичайної значимості набувають питання забезпечення інформаційної безпеки, визнаної в нашій країні однією з найважливіших складових національної безпеки, як багаторівневої проблеми державної інформаційної політики. Відзначимо, що статтею 17 Конституції України інформаційну безпеку визначено найважливішою функцією держави, справою всього Українського народу [1].

Необхідність протистояти протиправним посяганням на інформаційний простір України, збереження інформаційних ресурсів, захисту населення від негативного інформаційного впливу тощо є одним із основних завдань забезпечення інформаційної безпеки і вкрай важливим для української держави на сучасному етапі. Окрім цього, стратегічно визнаним пріоритетом зовнішньої політики України є європейська інтеграція, що вимагає удосконалення нормативно-правової бази забезпечення інформаційної безпеки України, яке б відповідало не лише міжнародним стандартам, а передусім українським національним інтересам.

Поширення комп'ютерних вірусів, шахрайство, крадіжки коштів із банківських рахунків або електронних гаманців, викрадення персональної та комерційної інформації, порушення правил роботи комп'ютерних систем – це далеко не повний перелік кіберзлочинів, кількість яких з кожним днем

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

збільшується. Причинами цього є: анонімність, що ускладнює виявлення злочину, злочинця; простота швидкого збагачення, доступність інтернет-інформації, яка тільки дозволяє вчиняти неправомірні діяння в мережі Інтернет [7]. Так, наприклад, за підсумками 2020 року головними загрозами для бізнесу стали крадіжки облікових даних і уразливості [8]. В руки кіберзлочинців потрапляє все більше ключів, які відкривають доступ до нашого приватного життя і роботи. Їм більше не потрібно витратити час на винахід хитрих способів проникнення в компанію — злочинці можуть проникати в мережу і проводити атаки, просто використовуючи відомі методи, такі як вхід в систему з вкраденими обліковими даними [8]. Кіберзлочинці активно стимулюють соціально-політичну активність громадян, вказуючи їм конкретні напрями та завдання діяльності, підказуючи шляхи та засоби вирішення проблем, що стоять перед ними за допомогою методів переконання, які одержали потужний розвиток впродовж ХХ сторіччя. Важко знайти будь-який інший інструмент переконання чи навіювання, що порівнюється з пропагандою за ефективністю закріплення в свідомості певних поглядів та ідей.

За публічною інформацією СБУ, ворожа пропаганда активно реалізовується через соціальні мережі. Так, наприклад, на Кіровоградщині за засуджено антиукраїнського Інтернет-агітатора. Співробітники спецслужби встановили, що місцевий житель публікував у заборонених соціальних мережах матеріали із закликами до масових заворушень, збройного протистояння з правоохоронними органами, закликав приєднуватися до незаконних збройних формувань [2].

На Чернігівщині співробітники Служби безпеки України під процесуальним керівництвом прокуратури викрили адміністратора антиукраїнських спільнот у соціальних мережах. Правоохоронці встановили, що мешканець Прилуцького району створював у соцмережах акаунти, через які адміністрував антиукраїнські групи. Оперативники спецслужби задокументували, що зловмисник розповсюджував матеріали, які містили заклики до зміни меж

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

території України, повалення конституційного ладу та державної влади, пропагував діяльність терористичних організацій «Л/ДНР», поширював інформацію, що дискредитує Збройні Сили України [3].

На Дніпропетровщині СБУ викрила підготовку РФ до втручання у вибори Президента України у 2018 році [4]. Співробітники СБУ під час виконання завдань із контррозвідувального захисту інтересів держави у сфері інформаційної безпеки протягом серпня викрили та заблокували діяльність мережі Інтернет-агітаторів, яких російські спецслужби залучили до заходів із підготовки до втручання у хід проведення виборів Президента України. Оперативники СБ України встановили, що російські спецслужби залучили до «співпраці» через комунікативні можливості Інтернету мешканців міст Дніпро, Кривий Ріг та Нікополь, які є адміністраторами груп у соціальних мережах. Спецслужби РФ поставили завдання із підготовки «плацдарму» для проведення заздалегідь запланованих заходів із впливу на хід проведення майбутніх виборів Президента України через маніпулювання громадською думкою Інтернет-користувачів [4].

За матеріалами Служби безпеки України засуджено жителя Черкащини, який проводив антиукраїнську агітацію у соціальних мережах [5].

СБУ припиняє діяльність антиукраїнських адміністраторів у соцмережах. Наприклад, співробітники СБ України протягом липня 2018 року припинили у п'яти областях країни діяльність одинадцяти адміністраторів у соцмережах, які розміщували в інтернеті антиукраїнські матеріали за вказівкою кураторів з російських спецслужб. Під час реалізації комплексу заходів із забезпечення інформаційної безпеки країни співробітники Департаменту контррозвідувального захисту інтересів держави у сфері інформаційної безпеки та обласних управлінь СБ України виявили нові докази використання спецслужбами РФ соціальних мереж для шкоди державній безпеці нашої країни [6]. На Дніпропетровщині СБУ припинила діяльність мережі антиукраїнських Інтернет-провокацій, які у заборонених російських соціальних мережах провокували Інтернет-користувачів до масових заворушень у День Конституції України 2018 року [7].

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

Отже, виходячи з вище перерахованих інцидентів – вкрай важливо реалізувати систему протистояння анти-українській пропаганді, що розповсюджується за допомогою соціальних мереж.

Таким чином в Україні запроваджено протидію пропаганді в соцмережах (Указ Президента України від 15.05.2017 №133/2017 “Про рішення Ради національної безпеки і оборони України від 28 квітня 2017 року “Про застосування персональних спеціальних економічних та інших обмежувальних заходів (санкцій)”). Його суть полягає у забороні інтернет-провайдерам надавати послугу з доступу користувачів інтернету до сервісів, російського виробництва “Вконтакте”, “Однокласники” та ін. Проте, як показує практика, реалізований підхід до забезпечення інформаційної безпеки держави не є ефективним. Адже, наприклад, заблоковані веб-сайти як і раніше входять в ТОП -10 найвідвідуваніших сайтів в Україні, оскільки користувачі активно використовують VPN, Tor, Омега та інші способи “обійти” блокування [2].

Отже, запроваджений підхід до протидії розповсюдженню пропаганді в соцмережах не є ефективним та потребує удосконалення шляхом запровадження додаткового блокування Tor, VPN і сайтів з інформацією про обхід заблокованих ресурсів. Це дозволить підвищити ефективність протидії пропаганді сепаратизму і антиукраїнської ідеології у Інтернеті, в тому числі в соціальних мережах.

Отже, метою виконання магістерської дипломної роботи є підвищення ефективності протидії пропаганді сепаратизму і антиукраїнської ідеології у Інтернеті, в тому числі в соціальних мережах, за допомогою реалізації програмного забезпечення системи блокування вхідних вузлів Tor і серверів VPN-провайдерів.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи кібербезпеки для блокування вхідних вузлів Tor і серверів VPN-провайдерів.

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

– Огляд існуючих систем кібербезпеки для блокування вхідних вузлів Tor і серверів VPN-провайдерів.

– Дослідження системи кібербезпеки для блокування вхідних вузлів Tor і серверів VPN-провайдерів.

– Програмна реалізація системи кібербезпеки для блокування вхідних вузлів Tor і серверів VPN-провайдерів.

Об'єктом дослідження є процес кібербезпеки для блокування вхідних вузлів Tor і серверів VPN-провайдерів.

Предметом дослідження є методи кібербезпеки для блокування вхідних вузлів Tor і серверів VPN-провайдерів.

Методи дослідження базуються на методах теорії захисту інформації, методах математичної статистики, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

– Удосконалено метод кібербезпеки для блокування вхідних вузлів Tor і серверів VPN-провайдерів.

– Розроблено вітчизняний продукт кібербезпеки для блокування вхідних вузлів Tor і серверів VPN-провайдерів, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі кібербезпеки для блокування вхідних вузлів Tor і серверів VPN-провайдерів.

Достовірність наукових результатів підтверджена теоретичними викладеннями, даними комп'ютерного моделювання, коректними дослідженнями параметрів на функціонуючій обчислювальній мережі, а також відповідністю отриманих результатів окремим результатам, наведеним у науковій літературі.

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

Робота апробована на LV Науково-технічна конференція здобувачів вищої освіти «Наука – виробництву», 2021, основні положення випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти надруковані у статті збірника праць молодих науковців ЦНТУ, випуск №12.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи кібербезпеки для блокування вхідних вузлів Tor і серверів VPN-провайдерів, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Програмне забезпечення системи блокування вхідних вузлів Tor і серверів VPN-провайдерів призначається для підвищення ефективності протидії пропаганді сепаратизму і антиукраїнської ідеології у Інтернеті, в тому числі в соціальних мережах.

Основною функцією ПЗ є блокування адрес вхідних вузлів Tor, серверів VPN-провайдерів і сайтів з інформацією про обхід. Блокування здійснюється за допомогою внесення заборонених ресурсів в «чорний» список. Отже, при зверненні користувача до забороненого ресурсу, він перенаправляється до локального хосту.

ПЗ призначене для використання провайдерами, оскільки саме вони є постачальниками послуги Інтернет. Провайдери встановлюють ПЗ на своїх серверах. Вимоги до використання ПЗ є мінімальними: працівники мають володіти комп'ютером на рівні базового користувача для того, щоб вносити, видаляти, редагувати записи в базі даних.

ПЗ, після первинного налагодження, працює автономно та не потребує втручання працівників. Первинне налагодження включає в себе:

- 1) встановлення ПЗ на сервер;
- 2) внесення url ресурсів, доступ до яких треба обмежити;
- 3) внесення IP-адреси, яку повертає DNS-сервер у відповідь на запит користувача до заблокованого ресурсу.

ПЗ може бути використане не тільки в умовах гібридної війни провайдерами, але й у випадку необхідності обмежити доступ до ресурсів, використання яких недоцільно на робочому місці. Наприклад, приватними фірмами, які можуть доручити встановлення ПЗ своїм адміністраторам.

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

1.2 Область застосування

Використання мережі Інтернет є щоденною справою сучасного суспільства. Її використовують для: розваг, роботи, спілкування, перегляду новин та навчання. Для здійснення всього цього необхідний провідник, який свого роду виступає проміжною ланкою між клієнтом і всіма послугами, якими йому надаються в мережі інтернет. На сьогоднішній день існує досить багато Інтернет-провайдерів, при чому є як всеукраїнські провайдери, так і локальні, які діють тільки в деяких містах і прилеглих до них селах [9]. Види провайдерів в Україні наведені в таблиці 1.1.

Програмне забезпечення системи блокування вхідних вузлів Tor і серверів VPN-провайдерів призначається для підвищення ефективності боротьби з анти-українською пропагандою, що поширюється через веб-ресурси, шляхом унеможливлення використання VPN та адрес вхідних вузлів Tor. Тому областю його застосування є послуги Інтернету, надавачами яких є Інтернет-провайдери. За їх допомогою можна отримати доступ до глобальної мережі Інтернет. Вони можуть відрізнятися тарифними планами, додатковими послугами, якістю послугами, швидкістю Інтернет з'єднання та якістю усунення проблем але всі вони працюють за єдиною схемою. Використовуючи спеціальне програмне забезпечення, користувач посилає зі свого комп'ютеру запит на отримання інформації до сервера Інтернет провайдера, той у свою чергу обробляє запит та передає серверу мережі Інтернет, на якому зберігається запитувана сторінка. Якщо сервер мережі Інтернет може надати запитувану інформації, тоді він передає її серверу Інтернет провайдера, після чого користувач отримує вже потрібну йому інформацію від свого Інтернет провайдера.

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

Таблиця 1.1 – Види провайдерів в Україні

Вид провайдера	Характеристика	Замітки
Первинні провайдери	Забезпечують вихід в світовий Інтернет у світових пунктах обміну трафіком (NAP – Network Access Point; MAE – Metropolitan Area Exchange; CIX – Commercial Internet Exchange)	В Україні відсутні
Провайдери першого рівня (первинні регіональні провайдери послуг)	Компанії, що надають послуги по доступу до ресурсів Інтернету місцевим (локальним) користувачам та іншим провайдерам. В свою чергу користуються послугами закордонних компаній, що знаходяться в міжнародних пунктах обміну трафіком (національні) за допомогою внутрішніх каналів.	В Україні приблизно 20 (в Києві – 12)
Великі провайдери	Сюди, як правило, належать провайдери першого рівня, а також компанії другого рівня, що мають велику кількість користувачів.	В Україні приблизно 40
Інші (провайдери другого і третього рівня, середні, малі і т.д.)	Сюди можна віднести компанії, що орендують канали різної пропускної спроможності в інших провайдерів з десятками-сотень вхідних ліній	Більшість українських провайдерів

Найпопулярнішими провайдерами України є [25-26]:

1) ПАТ "Укртелеком" – компанія є найбільшою по розгалуженій телекомунікаційній мережі. Своєю перевагою компанія вважає доступне за

ціною надання послуг, а також якісне сервісне обслуговування. Фішкою «Укртелекому» є інтерактивне телебачення. Ця послуга включає в себе крім 120 діючих каналів ще й кінотеатр на 10000 фільмів і серіалів на будь-який смак і перевагу глядача. Оцінка провайдера 1.5 із 5 виходячи з відгуків 5549 користувачів;

2) ТОВ "Інтертелеком" – мережа CDMA охоплює 86% території України. Оцінка провайдера 2 із 5 виходячи з відгуків 5315 користувачів.

3) АО "Київстар" – домашній інтернет від Київстар має позитивні відгуки. Пояснюється це досить просто. Він має найнижчі ціни на послуги, висока їх якість, гарне обслуговування абонентів, а також швидкість підключення для нових клієнтів. Оцінка провайдера 1.5 із 5 виходячи з відгуків 5024 користувачів;

4) ТОВ "ВОЛЯ Кабель" – роботу компанії її численні клієнти оцінюють, найчастіше, позитивно. Не обходиться і без критичних зауважень, але це служить стимулом для провайдера вдосконалювати рівень сервісу, в тому числі технічної та інформаційної підтримки [2]. Оцінка провайдера 1.5 із 5 виходячи з відгуків 4389 користувачів;

5) Телекомунікаційна група "Vega" – на сьогоднішній день Vega присутня у 159 населених пунктах 22 областей України. Оцінка провайдера 2.5 із 5 виходячи з відгуків 2000 користувачів.

Найпопулярніші провайдери Кропивницького [27-29]:

1) ТОВ "ІСП ШТОРМ"- компанія надає телекомунікаційні послуги в Кіровоградській області з 1999 року. Пропонують Інтернет, телебачення і міський телефон для жителів декількох десятків населених пунктів. Їх послугами користуються організації і приватні особи в Кіровоградському, Олександрійському, Знам'янському, Новоукраїнському, Маловисківському, Новомиргородському, Новгородківському, Олександрівському, Компаніївському районах. Прокладка волоконно-оптичної мережі, рішення по IP-телефонії, системна інтеграція або СКС – входять в послуги компанії. "ІСП ШТОРМ" сертифікований дилер і авторизований партнер D-Link, Tp-link, Vinet і т. д. Вони

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

Tor і серверів VPN-провайдерів унеможливить використання VPN та адрес вхідних вузлів Tor, за допомогою яких можливо отримати доступ до заблокованих веб-ресурсів, через які поширюється анти-українська пропаганда та вплив на політичні настрої громадян України.

Використання мережі Інтернет є невід'ємною частиною кожного сучасного громадянина, за допомогою цієї технології люди розважаються, спілкуються, заробляють та навчаються. Проміжною ланкою між користувачем і інтернетом виступає постачальник послуг – провайдер. Існує досить багато Інтернет-провайдерів, які в свою чергу мають певну територію охоплення. Усі провайдери серед переваг відзначають: якісне сервісне обслуговування, низькі ціни на послуги, інноваційні технології та надійний і якісний сервіс. Проте майже всі провайдери мають оцінку 2 із 5 виходячи з відгуків. Найпопулярнішими в Україні є: «Укртелеком», «Інтертелеком», «Київстар», «ВОЛЯ Кабель», «Vega». Найпопулярнішими провайдерами Кропивницького є: «ШТОРМ», «СИЛА», «Імперіал». Отже, областю застосування системи є сфера надання інтернет послуг, постачальниками яких є Інтернет провайдери.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи кібербезпеки для блокування вхідних вузлів Tor і серверів VPN-провайдерів, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

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

Програмні засоби доступу до вебресурсів

Браузер встановлює зв'язок із веб-сервером, завантажує на комп'ютер сторінку, розміщену за вказаною користувачем адресою, та відтворює цю сторінку на екрані. Вигляд веб-сторінки (гіпертекстового документа) у вікні браузера залежить, від використаних у ній тегів – команд мови розмітки HTML. Спільне оформлення сайтів визначає також інформація, записана мовою CSS (Cascade Style Sheets – каскадні таблиці стилів) у HTML-документах чи окремих файлах. Сучасні браузери здатні відображати ілюстрації, відтворювати анімацію, відеоролики і звук. Переглядаючи з їх допомогою веб-сторінки, на яких є гіперпосилання, користувач з легкістю, лише клацаючи відповідні посилання, переходить з однієї такої сторінки на іншу [2].

Усі сучасні браузери мають приблизно однакові можливості: забезпечують високу швидкість виведення веб-сторінок, підтримують роботу як з багатьма вікнами (кожна веб-сторінка відкривається в окремому вікні браузера), так і в режимі одного вікна (кожна веб-сторінка відкривається на окремій вкладці одного вікна браузера), дозволяють з допомогою панелі інструментів вводити критерії пошуку і вибирати пошукову систему, пропонують вбудовані завантажувачі файлів і здатні забезпечити високий рівень безпеки під час роботи. Найбільшу популярність у користувачів Інтернету традиційно виборює браузер Internet Explorer, зображений на рисунку 2.1.

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

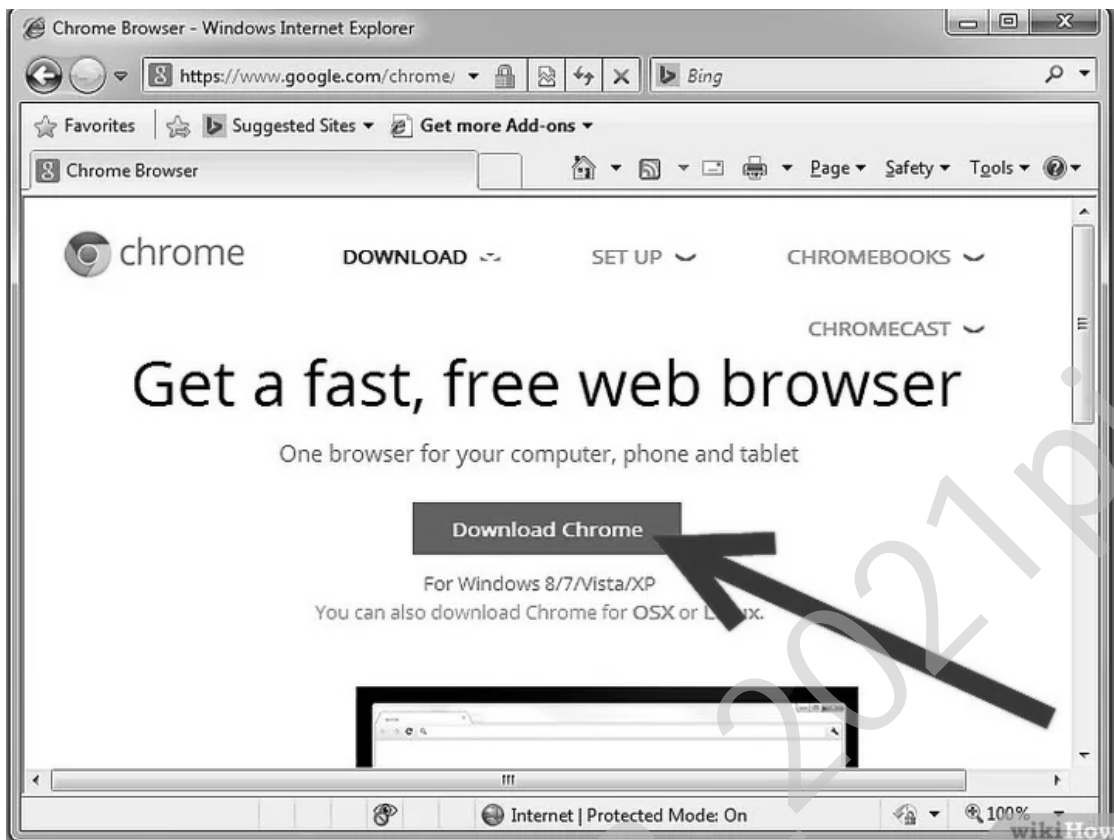


Рисунок 2.1 – Вікно браузера Internet Explorer [2]

Він не лише входить до складу кожної нової версії операційної системи Windows, а й є доступним для безкоштовного завантаження з сервера корпорації Microsoft (<http://microsoft.com/>) і встановлення в операційній системі Windows 34 попередніх версій (наприклад, у Windows XP). Слід визнати, що перші версії браузерів Internet Explorer дуже рідко використовувалися як браузери на «кожний день», хоча й були встановлені на більшості комп'ютерів. Проте в Internet Explorer версії 9 було внесено суттєві зміни, а саме перероблено інтерфейс, збільшено швидкість завантаження сторінок і покращено безпеку програми. Завдяки цьому браузер Internet Explorer 9 має хорошу функціональність [2].

Головною перевагою Internet Explorer є те, що абсолютна більшість вебсторінок відображається в ньому саме так, як задумав веб-дизайнер. Браузер має простий інтерфейс, містить засоби керування пошуком інформації в Інтернеті, надає легкий спосіб налаштування параметрів безпеки та інших

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРМ-123.21.0010.00.00.ПЗ

Арк.

18

to Firefox, яка з'являється після наведення на нього вказівника миші. Одним з найпопулярніших вважається доповнення Easy YouTube Video Downloader, яке дозволяє зберігати відеоролики з сайту YouTube на жорсткий диск [2].

Недоліком Mozilla Firefox є те, що цей браузер не вирізняється високою стабільністю роботи і потребує багато оперативної пам'яті, що інколи призводить до «гальмування» всієї системи. Інтегрований пакет програм норвезької компанії Opera Software (<http://www.opera.com/>), що зображений на рисунку 2.3, є серйозним конкурентом продуктів Microsoft і Mozilla. Він невимогливий до ресурсів комп'ютера, характеризується високою швидкістю роботи, має зручний інтерфейс, панель інструментів, що налаштовується, а також чудовий засіб для масштабування відображуваних сторінок, що містять не лише текст, а й графіку.

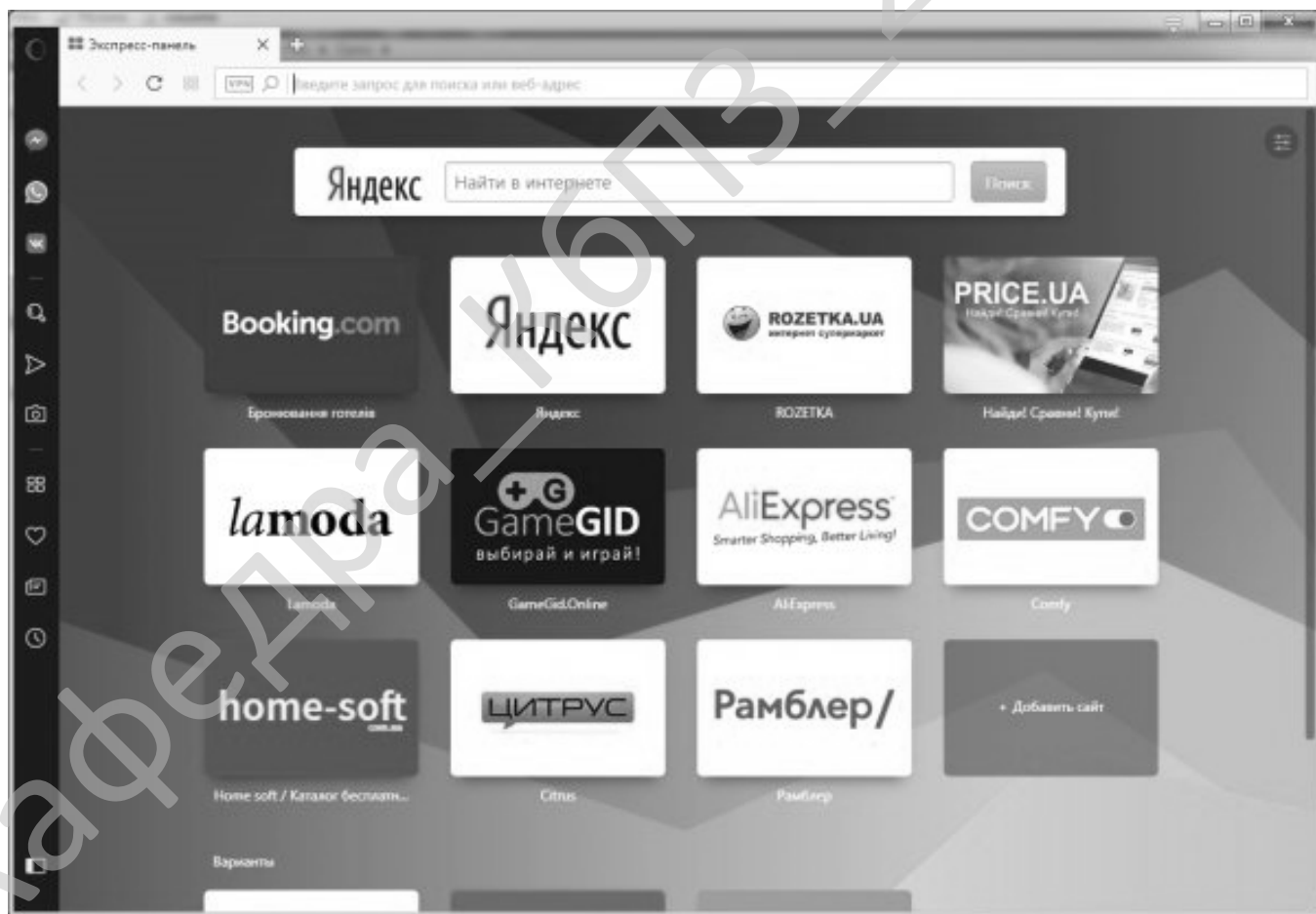


Рисунок 2.3 – Вікно браузера Опера [2]

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

Стан браузера Опера може зберігатися після виходу з його вікна та відновлюватися під час його наступного відкриття. Працюючи в режимі Turbo, можна в декілька разів пришвидшити завантаження веб-сторінок і зекономити трафік. Така можливість буде доречною насамперед тим користувачам, що мають повільне або дороге інтернет-з'єднання, наприклад мобільний доступ до мережі. Режим Turbo вмикається клацанням кнопки Turbo, що має вигляд спідометра та розташована в лівому нижньому куті вікна [2]. Саме у браузерах Опера були вперше розширені функції миші. Наприклад, для повернення на попередню сторінку достатньо, утримуючи праву кнопку миші, трохи перемістити курсор ліворуч. Як і в Mozilla Firefox, в Опера можна встановлювати розширення (мініпрограми, що збільшують кількість функціональних можливостей браузера) і теми. Щоб встановити розширення, потрібно знайти його на сайті <https://addons.opera.com/ru/addons/extensions/> і клацнути розташовану праворуч кнопку Установить. Мабуть, єдиним недоліком Опера, який іноді дає про себе знати і призводить до зависання усієї системи (на щастя, не на всіх комп'ютерах), є раптове збільшення використання ресурсів процесора до 100 % [2].

Google Chrome – браузер, випущений компанією Google, зображений на рисунку 2.4. Він позиціонується розробниками як швидкий, надійний та безпечний. Стосовно надійності він – найкращий. Справа в тому, що у разі збою закривається лише та веб-сторінка, яка стала причиною помилки, а всі інші веб-сторінки залишаються відкритими. Інші ж браузери в разі збою закривають свої вікна повністю, з усіма відкритими вкладками, тож їх доводиться завантажувати заново [2].

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

Протокол IP не намагається виправити помилки такого типу. Підтвердження отримання пакетів і повторне звернення за втраченими пакетами входять в круг завдань протоколу більш високого рівня, наприклад TCP. IP-пакет – форматований блок інформації, що передається по обчислювальній мережі. З'єднання обчислювальних мереж, які не підтримують пакети, такі як традиційні з'єднання типу «крапка-крапка» в телекомунікаціях, просто передають дані у вигляді послідовності байтів, символів або бітів. При використанні пакетного форматування мережа може передавати довгі повідомлення надійніше і ефективно [4].

UDP є одним з найпростіших протоколів транспортного рівня моделі OSI. На відміну від TCP, UDP не гарантує доставку пакету. Це дозволяє йому набагато швидше і ефективніше доставляти дані для додатків, яким потрібна велика пропускна спроможність ліній зв'язку, або потрібний малий час доставки даних. Недостатня надійність протоколу може виражатися як у втраті окремих пакетів, так і в їх дублюванні. UDP використовується при передачі потокового відео, ігор реального часу, а також деяких інших типів даних. Ненадійність протоколу UDP треба розуміти в тому сенсі, що у випадках впливу зовнішніх чинників, що приводять до збоїв, протокол UDP не передбачає стандартного механізму повторення передачі втрачених пакетів [4].

Transmission Control Protocol (TCP) виконує функції протоколу транспортного рівня моделі OSI. TCP надає потік даних, з попередньою установкою з'єднання, за рахунок цього що дає упевненість в достовірності отримуваних даних, здійснює повторний запит даних у разі втрати даних і усуває дублювання при отриманні двох копій одного пакету. На відміну від UDP, гарантує, що додаток отримає дані точно в такій же послідовності, в якій вони були відправлені, і без втрат. На відміну від традиційної альтернативи – UDP, який може відразу ж почати передачу пакетів, TCP встановлює з'єднання, які повинні бути створені перед передачею даних. TCP з'єднання можна розділити на 3 стадії: установка з'єднання, передача даних, завершення з'єднання [4].

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

DHCP дозволяє комп'ютерам автоматично отримувати IP-адрес і інші параметри, необхідні для роботи в мережі TCP/IP. Модель роботи «клієнт-сервер». Комп'ютер-клієнт на етапі конфігурації мережевого пристрою звертається до сервера DHCP, і отримує від нього потрібні параметри. Адміністратор може задати діапазон адрес, що розподіляються сервером серед комп'ютерів. Це дозволяє уникнути ручної настройки комп'ютерів і зменшує кількість помилок. Протокол DHCP використовується в більшості крупних мереж TCP/IP. DHCP є розширенням протоколу BOOTP, що використовувався раніше для забезпечення бездисккових робочих станцій IP-адресами при їх завантаженні. DHCP зберігає зворотну сумісність з BOOTP. Стандарт протоколу DHCP був прийнятий в жовтні 1993 року. Версія протоколу (березень 1997 року), що діє, описана в RFC 2131. Нова версія DHCP, призначена для використання в середовищі IPv6, носить назву DHCPv6 і визначена в RFC 3315[4].

DNS, що зображений на рисунку 2.5, здатний за запитом, що містить доменне ім'я хосту (комп'ютера або іншого мережевого пристрою), повідомити IP-адресу або іншу інформацію.

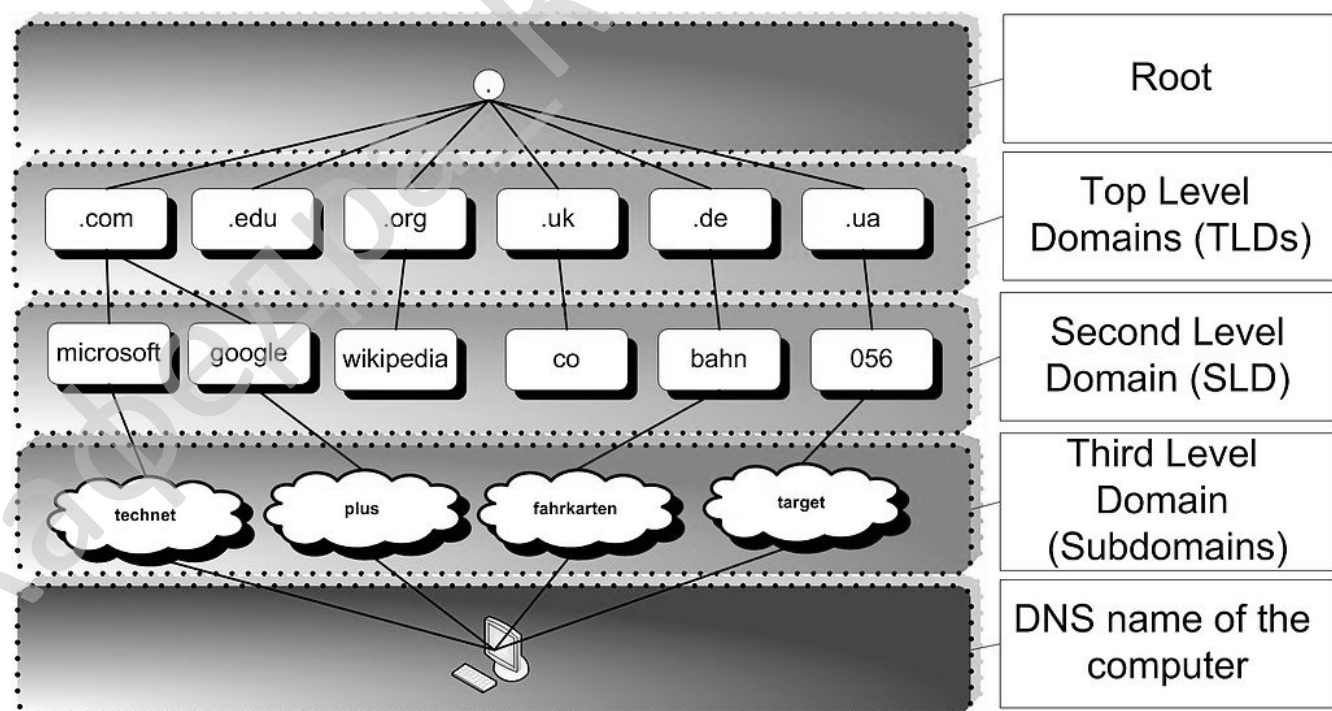


Рисунок 2.5 – Структура DNS [16]

інформація про домен. Ієрархія серверів DNS співпадає з ієрархією доменів. Ім'я і IP-адреса не тотожні – одна IP-адреса може мати безліч імен, що дозволяє підтримувати на одному комп'ютері безліч веб-сайтів (це називається віртуальний хостинг). Зворотне теж справедливо – одному імені може бути зіставлене безліч IP-адрес: це дозволяє створювати балансування навантаження. Для підвищення стійкості системи використовується безліч серверів, що містять ідентичну інформацію, а в протоколі є засоби, що дозволяють підтримувати синхронність інформації, розташованої на різних серверах. Існує 13 корневих серверів, їх адреси практично не змінюються [4].

Протокол DNS використовує для роботи TCP- або UDP-порт 53 для відповідей на запити. Традиційно запити і відповіді відправляються у вигляді однієї UDP датаграми. Припустимо, ми набрали адресу ru.wikipedia.org. Браузер питає у сервера DNS: «яка IP-адреса у ru.wikipedia.org»? Проте, сервер DNS може нічого не знати не тільки про запитане ім'я, але навіть про весь домен wikipedia.org. В цьому випадку має місце рекурсія: сервер звертається до кореневого сервера – наприклад, 198.41.0.4. Цей сервер повідомляє – «У мене немає інформації про дану адресу, але я знаю, що 204.74.112.1 є авторитетним для зони org.» Тоді сервер DNS направляє свій запит до 204.74.112.1, але той відповідає «У мене немає інформації про даний сервер, але я знаю, що 207.142.131.234 є авторитетним для зони wikipedia.org.» Нарешті, той же запит відправляється до третього DNS-серверу і отримує відповідь – IP-адреса, яка і передається клієнтові – браузеру.

В даному випадку в процесі пошуку IP по імені:

- 1) браузер відправив відомому йому DNS-серверу т.з. рекурсивний запит – у відповідь на такий тип запиту сервер зобов'язаний повернути «готовий результат», тобто IP-адреса, або повідомити про помилку;
- 2) DNS-сервер, отримавши запит від клієнта, послідовно відправляв ітеративні (нерекурсивні) запити, на які отримував від інших DNS-серверів

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

відповіді, поки не отримав авторитетну відповідь від сервера, відповідального за запитану зону [4].

Запит на визначення імені зазвичай не йде далі кеша DNS, який зберігає відповіді на запити, що проходили через нього раніше. Разом з відповіддю приходить інформація про те, скільки часу дозволяється зберігати цей запис в кеш. Зворотній DNS-запит. DNS використовується в першу чергу для перетворення символьних імен в IP-адреса, але він також може виконувати зворотний процес. Для цього використовуються вже наявні засоби DNS. Річ у тому, що із записом DNS можуть бути зіставлені різні дані, у тому числі і яке - небудь символьне ім'я. Існує спеціальний домен in-addr.arpa, записи в якому використовуються для перетворення IP-адресов в символьні імена. Наприклад, для отримання DNS імені для адреси 11.22.33.44 можна запитати у DNS-сервера запис 44.33.22.11.in-addr.arpa, і той поверне відповідне символьне ім'я. Зворотний порядок запису частин IP-адреса пояснюється тим, що в IP-адресах старші біти розташовані на початку, а в символьних DNS-іменах старші (що знаходяться ближче до кореня) частини розташовані в кінці. Зарезервовані доменні імена. Документ RFC 2606 (Reserved Top Level DNS Names – Зарезервовані імена доменів верхнього рівня) визначає назви доменів, які слід використовувати як прикладів (наприклад, в документації), а також для тестування. Окрім example.com, example.org і example.net, до цієї групи також входять test, invalid і in. Формат DNS-повідомлення. DNS має два типи повідомлення: запит і відгук. Обидва типи мають один і той же формат, згідно з рисунком 2.2.



Рисунок 2.6 – Формат DNS-повідомлення [4]

про те, чи успішно виконаний запит. Може приймати наступні значення – таблиця 2.4.

Таблиця 2.4 – Значення, які може приймати поле rCode

Значення	Коментар
0	немає помилки
1	помилка формату
2	проблема імені сервера
3	проблема посилання на домен
4	тип запиту не підтримується
5	адміністративна заборона
6-15	резервні

Лічильник питань-16-бітове поле, що містить кількість запитів. Лічильник відповідей-16-бітове поле, що містить кількість відповідей. Кількість записів повноважень-16-бітове поле, що містить кількість записів повноважень. Кількість записів додаткової інформації-16-бітове поле, що містить кількість записів додаткової інформації. Секція запиту – секція, що містить одну або більш записи. Вона міститься і в запиті, і в відповіді. Секція відповіді – секція, що містить одну або більш записи. Вона представляє тільки повідомлення відповіді. Секція включає відповідь від сервера до клієнта (повідомлення розпізнавача). Секція повноважень – секція, що містить одне або більше повідомлень джерела. Вона представлена тільки у повідомленні-відповіді. Секція дає інформацію (ім'я домена) про один або більш повноважних серверів для запиту. Секція додаткової інформації – секція, що містить одну або більше записів джерел. Вона представлена тільки у повідомленні-відповіді. Секція забезпечує додатковою інформацією, яка може допомогти розпізнавачу. Кожне доменне ім'я (кожен вузол дерева) пов'язаний із записом ресурсу. Сервер бази даних містить записи ресурсу. Записи ресурсу – це також те, що повертається сервером клієнтові. Ім'я домену – поле змінної довжини, що містить ім'я домену.

Воно дублює ім'я домена в записі запиту. DNS використовує стиснення скрізь де повторюється ім'я. Тип домену – поле таке ж, як поле типу запиту в секції запиту, виключаючи два останні типи, які не дозволені. Клас домену – поле таке ж, як поле класу запиту в секції запиту. Час життя-32-бітове поле, що визначає число секунд, при якому відповідь дійсна. Приймач може кешувати відповідь на цей період часу. Значення нуль означає, що запис ресурсів використовується тільки на один перехід і не може бути кешована.

Довжина ресурсу даних – 16-бітове поле визначає довжину ресурсу даних. Дані ресурсу – поле змінної довжини, що містить відповідь на запит (у секції відповідь), або ім'я домену повноважного сервера (у секції повноважень), або додаткову інформацію (у додатковій секції). IPv4-адрес – це ціле число з 4-х октетів, IPv6-адрес – з 16-ти октетів. Ім'я домену. Імена домена виражаються як послідовність міток. Кожна мітка позначається полем однобайтової довжини, яке визначає число символів в мітці. Кожне ім'я домену закінчується нульовою міткою, останній байт кожного імені домену – поле довжини із значенням 0. Для того, щоб відрізнити поле довжини і точку відгалуження, два старші розряди довжини поля повинні бути завжди нульовими (00). Це не створюватиме проблем, тому що довжина мітки не може бути більше 63 – максимального значення з 6-ти (111111). Точка відгалуження. Ім'я домену може бути замінене точкою відгалуження. Точка відгалуження – двобайтове поле з двома старшими бітами, встановленими в положення 1 (11). Рядок символів. Поле довжини (1 байт) + символи. Може мати довжину до 256 символів (з полем довжини). Стиснення. Коли ім'я домену повторюється, DNS вимагає, щоб воно замінювалося покажчиком зсуву. Наприклад, запис ресурсу імені домена зазвичай повторює ім'я домена в запиті. Для того, щоб уникнути дублювання, DNS визначає двобайтовий покажчик зсуву, який указує на попереднє входження імені домену. Перші два старші розряди – це дві одиниці, необхідні для того, щоб відрізнити точку-покажчик зрушення від інформації "довжина поля".) Інші 14 біт представляють число, яке указує на відповідний байт номера повідомлення. Цей

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

байт відлічується від початкового повідомлення з першим байтом, який вважається нульовим [4].

Інтернет-сервіс-провайдинг

У зв'язку з бурхливим розвитком Інтернету стрімко зростає популярність інтернет-сервіс-провайдингу. Сервіс-провайдерами, або постачальниками послуг, називають фірми, які надають послуги доступу в Інтернет. Унікальність Інтернету полягає в неосяжному наповненні інформаційними ресурсами та широкому застосуванні інформаційних технологій, таких як накопичення, зберігання, обробка інформації та подання її у формі Web-сторінок. Завдяки цьому Інтернет називають глобальною інформаційною мережею. Інформаційні послуги (Information Services) Інтернету полягають у тому, щоб забезпечити користувачів можливістю пошуку в мережі найрізноманітнішої інформації (контенту). Окрім того в Інтернеті, технологічною особливістю якого є пакетний спосіб передачі інформації, можливою є організація різних служб, зокрема з надання користувачам недорогих телекомунікаційних послуг, найпоширенішою серед яких електронна пошта [15].

Діяльність сервіс-провайдерів зосереджена на організації так званих сервісних вузлів (Service Nodes), за допомогою яких реалізується доступ користувачів до різних мережевих служб та інформаційних ресурсів як даного вузла, так і віддалених вузлів Інтернету. При цьому постачальники послуг (провайдери) також є споживачами телекомунікаційних послуг (послуг з транспортування інформації), які надаються мережевими операторами. Доступ користувачів до сервісного вузла, як правило, здійснюється через місцеві мережі операторів, однак окремі провайдери можуть мати власні мережі абонентського доступу. Прийнято розрізняти сервіс-провайдерів різних рівнів: місцевого рівня, регіонального рівня та національного рівня, який зображений на рисунку 2.7.

Сервісний вузол провайдера місцевого рівня (рівня III) зовнішнім каналом, орендованим у мережевого оператора, під'єднується до так званої точки

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

мережевої присутності (Point of Presence, POP), у якій розміщується обладнання мережевого доступу регіонального провайдера.

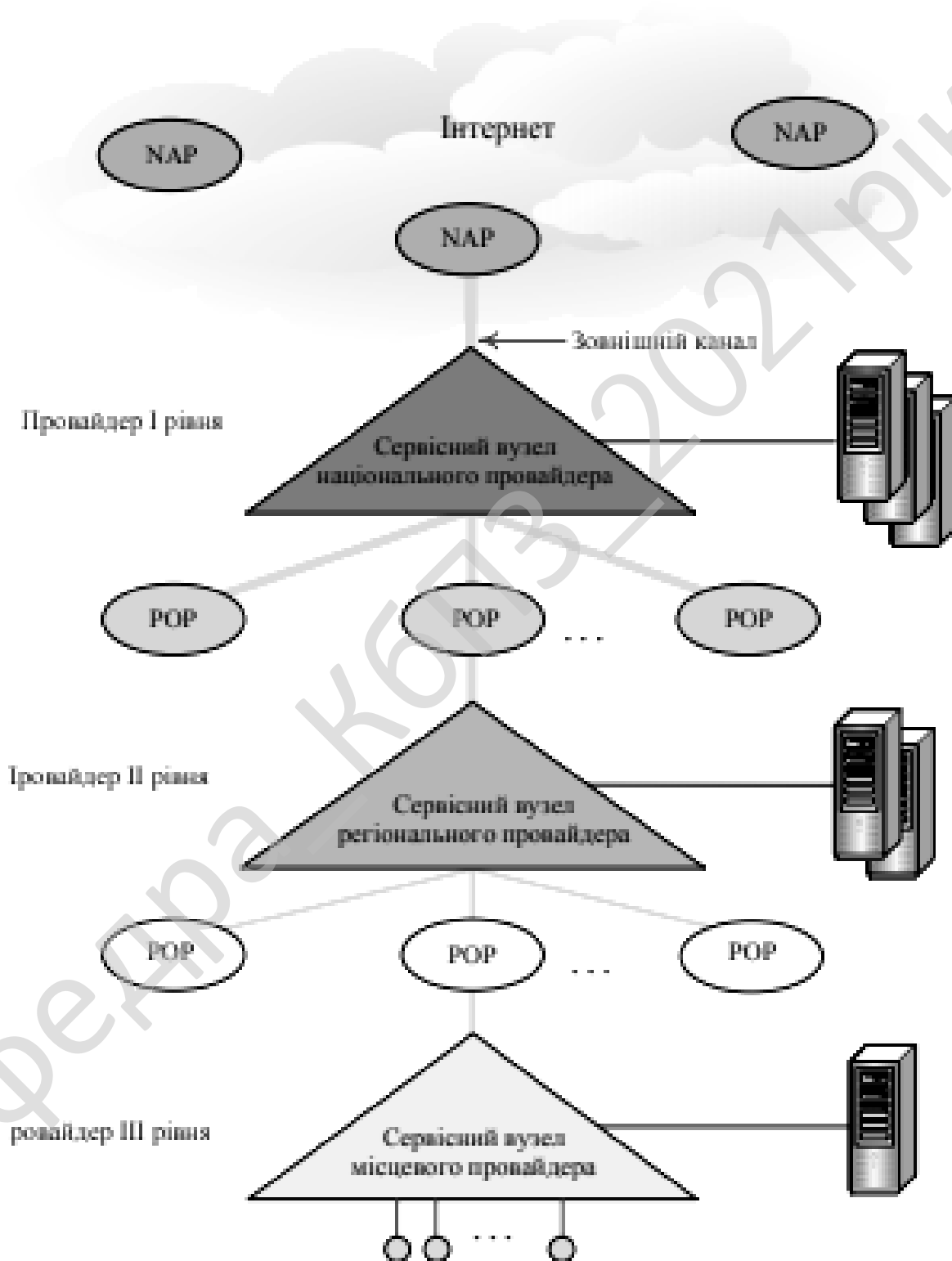


Рисунок 2.7 – Структурна схема інтернет-сервіс-провайдингу [15]

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРМ-123.21.0010.00.00.ПЗ

Арк.

33

Регіональний провайдер (рівень II) зазвичай розміщує в своєму регіоні кілька точок мережевої присутності, забезпечуючи користувачів місцевих провайдерів доступом як до своїх інформаційних ресурсів (свого сервісного вузла), так і до зовнішніх ресурсів Інтернету. Регіональний провайдер аналогічним чином, орендуючи канал у мережевого оператора, під'єднується до мережі доступу національного провайдера (рівень I). І тільки національний провайдер має право під'єднання до точки мережевого доступу (Network Access Point, NAP). NAP – це міжнаціональні точки доступу в Інтернет. Таким чином, створюється логічно функціонуюча мережа «провайдерського класу». За такою схемою сервіс-провайдер зосереджує фінансові ресурси на маркетингу та створенні потужної клієнтської бази. Орендує канали доступу та обладнання у посередника (мережевого оператора), а також використовує його служби технічної підтримки, сервіс-провайдер мінімізує свої експлуатаційні витрати. Слід зазначити, що на практиці функції вищевказаних суб'єктів діяльності часто поєднуються в одній компанії з метою збільшення прибутків [15].

На сьогодні, провайдер місцевого рівня в Україні має структуру, представлену на рисунку 2.8.

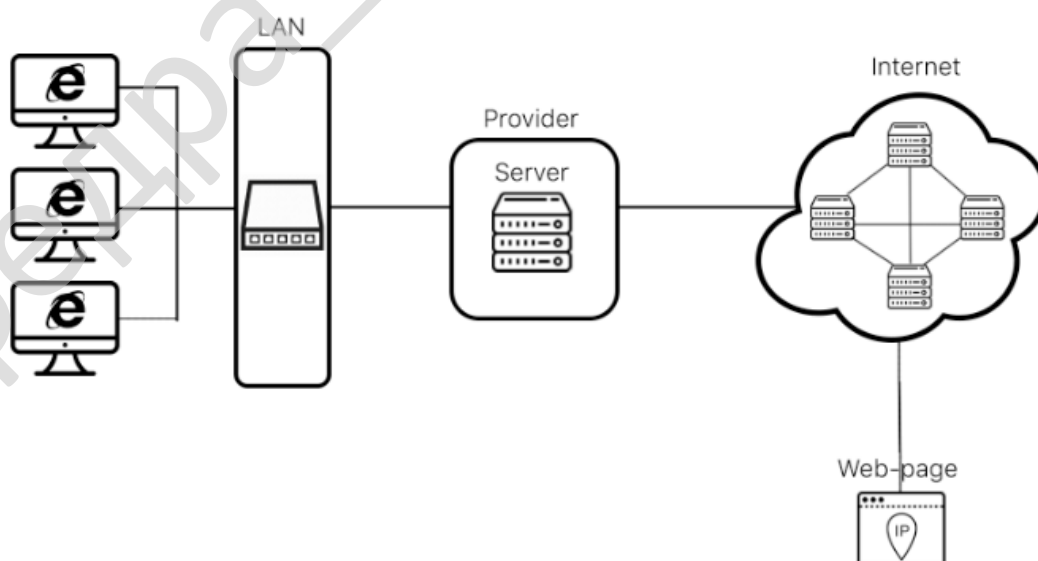


Рисунок 2.8 – Структурна схема провайдера місцевого рівня

Користувач має можливість отримати доступ до будь-якого веб-ресурсу, використавши VPN чи TOR, унеможливлення використання яких не передбачено.

Мережеві оператори

Оператором мережі (Network Operator) називається компанія, яка є власником телекомунікаційної інфраструктури та бере на себе всі витрати щодо забезпечення її працездатності з заданим рівнем якості обслуговування. Її ще називають мережевим оператором, або просто оператором, схема зображена на рисунку 2.9 [15].

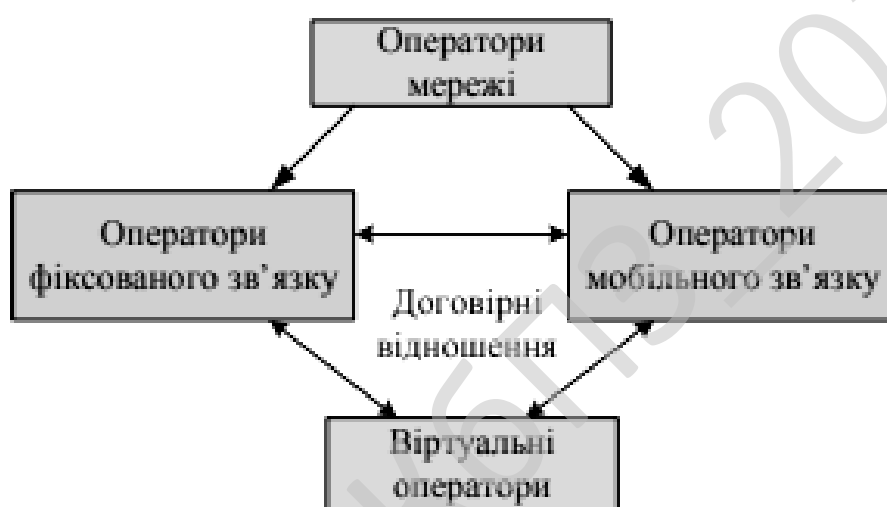


Рисунок 2.9 – Класифікаційна схема взаємодії між операторами [15]

Кінцевим продуктом діяльності мережевого оператора є надання послуг з транспортування інформації його мережею. Ці послуги називаються телекомунікаційними послугами (Telecommunication Services) та надаються як кінцевим користувачам мережі, так і іншим мережевим операторам, забезпечуючи їх транзитною можливістю з передачі трафіку через свої мережі. У зв'язку з цим мережі операторів прийнято називати телекомунікаційними мережами ("теле-" в перекладі з давньогрецької означає "далеко"). Їх основним завданням є забезпечення можливості віддаленорозташованих об'єктів

обмінюватися інформаційними повідомленнями. Створюючи мережу загального користування, оператор зобов'язаний забезпечити в будь-якому місці мережі, до якого під'єднано кінцеві пристрої, стандартний інтерфейс (точку з'єднання). Розрізняють операторів фіксованого та мобільного (стільникового) зв'язку. Оператори фіксованого зв'язку (Fixed Communication Operators) організують стаціонарні мережі, в яких комунікаційне обладнання та пристрої користувачів розміщуються в стаціонарних пунктах мережі. Оператори мобільного зв'язку (Mobile Communication Operators) створюють мережеве покриття території, розміщуючи свої базові станції за стільниковою схемою в стаціонарних або рухомих пунктах, забезпечуючи тим самим можливість вільного переміщення абонентів у зоні покриття. Серед основних тенденцій розвитку ринку стільникового зв'язку найприкметнішою є поява так званих віртуальних операторів (Virtual operators). Це компанії, які не мають власних мережевих ресурсів, займаються в основному маркетинговою діяльністю й у вигляді пакетів популярних послуг на основі гнучкої тарифної сітки реалізують їх клієнтам під своєю торговою маркою. Реалізацію ж послуг виконує мережевий оператор, з яким віртуальний оператор вступає у договірні відносини з частковою участю в прибутку від продажу послуг. Оператор, якому належить мережеве обладнання, при цьому повністю зосереджує свою діяльність на підтримці високого рівня його працездатності. У період лібералізації ринків основні інтереси всіх операторів зосереджуються на пошуку нових ринкових форм комплексних рішень щодо розширення послуг, які надаються користувачам. Для операторів фіксованих мереж таким рішенням є надання мобільного доступу своїм абонентам. Операторові мобільного зв'язку фіксована мережа дозволяє стати постачальником повного набору послуг. Доступність комбінації мобільного та фіксованого доступу, надання широкосмугового доступу, а також послуг передачі даних забезпечують ідеальні умови виживання операторів зв'язку в умовах високої конкуренції на ринку телекомунікацій.

Оскільки DNS грає ключову роль в отриманні IP домену та за його

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

допомогою користувачі мережі Інтернет отримують доступ до будь-якого веб-ресурсу – вирішено будувати систему за допомогою DNS-серверу. DNS входить у стек протоколів TCP/IP, який має наступні переваги:

1) незалежність від мережної технології. TCP/IP не залежить від обладнання, так як він визначає елемент передачі – дейтаграму та описує спосіб її руху мережею;

2) стек дозволяє будь-якій парі комп'ютерів, що його підтримують, взаємодіяти один з одним. Кожному комп'ютеру надається логічна адреса, а кожна дейтаграма, що передається, містить логічні адреси відправника та одержувача;

3) протоколи стеку TCP/IP забезпечують підтвердження правильності проходження при обміні між відправником та одержувачем;

4) протокол TCP/IP містить засоби підтримки основних прикладних програм, таких як, електронна пошта, передача файлів, віддалений доступ і т.і.

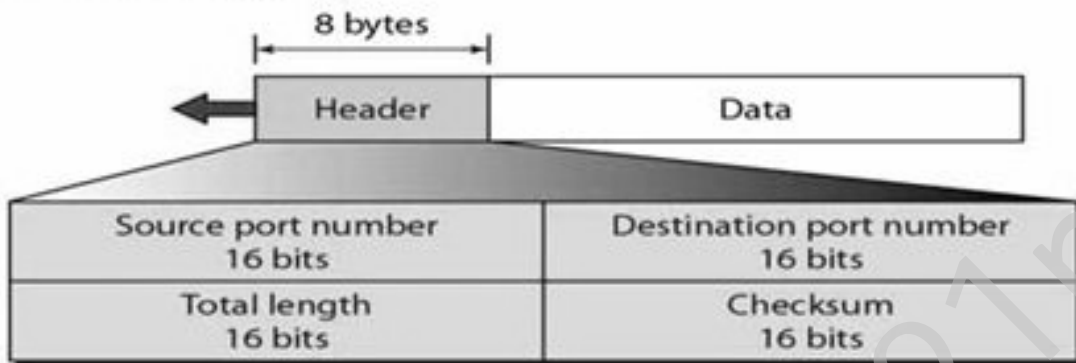
За допомогою документації, запит до DNS серверу можливо легко отримати, обробити та сформувані відповідь. DNS має зрозумілу структуру, що зображена на рисунку 2.10. Володіє важливими характеристиками, що описані в 2.10. Реалізований DNS-сервер можливо легко впровадити в експлуатацію та не потребує висококваліфікованих співробітників [22].

Протокол призначених для користувача дейтаграм є одним з основних мережеских протоколів, використовуваних для передачі інформації в Інтернеті, зображений на рисунку 2.9. Він є частиною пакета Internet Protocol Suite, який визначає, як комп'ютери взаємодіють з серверами і один з одним через мережу інтернет-протоколу (IP). При використанні протоколу призначених для користувача дейтаграм додаток може відправляти дейтаграми або повідомлення іншим хостам без попередньої настройки спеціальних каналів передачі або шляхів передачі даних. Це дозволяє швидше спілкуватися, але менш надійно, ніж інші методи.

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

User Datagram Protocol (UDP)

□ User Datagram



- The calculation of checksum and its inclusion in the user datagram are optional.

$$\text{UDP length} = \text{IP length} - \text{IP header's length}$$

Рисунок 2.10 – Протокол дейтаграм користувача [24]

Передача даних відбувається за допомогою протоколу UDP, оскільки це забезпечує низькі накладні витрати, застосовується менше службової інформації, ніж від використання протоколу TCP. Встановлення з'єднання. Немає вимоги, що одержувач доступний і готовий до приймання даних і немає вимоги щодо підтвердження отримання [23].

Фокус протоколу дейтаграми користувача полягає в передачі даних. Він не забезпечує ніяких гарантій для забезпечення фактичної доставки інформації або для захисту цілісності даних при її переміщенні. Протокол передбачає, що якщо перевірка помилок важлива, то комп'ютери, які виконують зв'язок, подбають про це. Протокол дейтаграми користувача працює без урахування стану. Це означає, що він обробляє кожен запит інформації як незалежну транзакцію. Перевага в цьому способі полягає в тому, що сервер може підтримувати більш високий трафік, тому що немає необхідності призначати простір для зберігання повідомлень. У деяких випадках протокол призначених для користувача

дейтаграм насправді краще як більш надійний спосіб зв'язку. Наприклад, він дозволяє відправляти повідомлення без затримки, викликані перевіркою помилок. З цієї причини часто використовуються додатки, що залежать від часу. Загальним прикладом такого типу додатків є потокові медіа, такі як Voice over IP. Протокол призначених для користувача дейтаграм також корисний для серверів, які обробляють величезний обсяг невеликих запитів. Оскільки сервера не потрібно призначати простір для зберігання або встановлювати з'єднання перед спілкуванням, він може швидше реагувати на кожен окремий запит. Результатом цього є більш швидке обслуговування для всіх користувачів. З цієї причини протокол призначених для користувача дейтаграм часто використовується серверами, які працюють в онлайн-іграх.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Головним завданням БД є гарантоване збереження значних обсягів інформації та надання доступу до неї користувачеві або ж прикладній програмі. БД складається з двох частин: збереженої інформації та системи управління нею. З метою забезпечення ефективності доступу записи даних організують як множину фактів (елемент даних). Існує величезна кількість різновидів баз даних, що відрізняються за критеріями. Класифікація БД за моделлю даних: ієрархічні, мережеві, реляційні, об'єктні, об'єктно-орієнтовані, об'єктно-реляційні. Класифікація БД за технологією фізичного зберігання: БД у вторинній пам'яті (традиційні); БД в оперативній пам'яті (in-memory databases); БД у третинній пам'яті (tertiary databases). Класифікація БД за вмістом: географічні, історичні, наукові, мультимедійні. Класифікація БД за ступенем розподіленості: централізовані (зосереджені); розподілені. Окреме місце в теорії та практиці займають просторові тимчасові, або темпоральні і просторово-часові БД. Ієрархічні бази даних можуть бути представлені як дерево, що складається з об'єктів різних рівнів. Верхній рівень займає один об'єкт, другий – об'єкти

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

другого рівня і т.д. Між об'єктами існують зв'язки, кожен об'єкт може включати в себе декілька об'єктів більш низького рівня. Такі об'єкти перебувають у відношенні предка (об'єкт більш близький до кореня) до нащадка (об'єкт більш низького рівня), при цьому можлива ситуація, коли об'єкт-предок не має нащадків або має їх декілька, тоді як у об'єкта -нащадка обов'язково тільки один предок. Об'єкти, що мають загального предка, називаються близнюками. Мережеві бази даних подібні до ієрархічних, за винятком того, що в них є покажчики в обох напрямках, які з'єднують споріднену інформацію. До основних понять мережевої моделі бази даних відносяться: рівень, елемент (вузол), зв'язок. Незважаючи на те, що ця модель вирішує деякі проблеми, пов'язані з ієрархічною моделлю, виконання простих запитів залишається досить складним процесом. Також, оскільки логіка процедури вибірки даних залежить від фізичної організації цих даних, то ця модель не є повністю незалежною від програми. Іншими словами, якщо необхідно змінити структуру даних, то потрібно змінити і додаток. Реляційна модель орієнтована на організацію даних у вигляді двовимірних таблиць. Кожна реляційна таблиця являє собою двовимірний масив і має наступні властивості: кожен елемент таблиці – один елемент даних; всі осередки в стовпчику таблиці однорідні, тобто всі елементи в стовпчику мають однаковий тип (числовий, символічний тощо); кожен стовпчик має унікальне ім'я; однакові рядки в таблиці відсутні; порядок проходження рядків і стовпчиків може бути довільним. Об'єктна СУБД ідеально підходить для інтерпретації складних даних, на відміну від реляційних СУБД, де додавання нового типу даних досягається ціною втрати продуктивності або за рахунок різкого збільшення термінів і вартості розробки додатків. Об'єктна база, на відміну від реляційної, не вимагає модифікації ядра при додаванні нового типу даних. Новий клас і його екземпляри просто надходять у зовнішні структури бази даних. Система управління ними залишається без змін. Результатом поєднання можливостей (особливостей) баз даних і можливостей об'єктно-орієнтованих мов програмування є об'єктно-орієнтовані системи управління базами даних (ООСУБД). ООСУБД дозволяють працювати з об'єктами баз даних також, як з

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

об'єктами у програмуванні в об'єктно-орієнтованих мовах програмування. ООСУБД розширює мови програмування, прозоро вводячи довготривалі дані, управління паралелізмом, відновлення даних, асоційовані запити й інші можливості. Об'єктно-орієнтовані бази даних звичайно рекомендовані для тих випадків, коли потрібна високопродуктивна обробка даних, які мають складну структуру. Система, яка забезпечує об'єктну інфраструктуру і набір реляційних розширювачів, називається "об'єктно-реляційною". Об'єктно-реляційні системи поєднують переваги сучасних об'єктно-орієнтованих мов програмування з такими властивостями реляційних систем як множинні представлення даних і високорівневі непроцедурні мови запитів. За технологією обробки даних бази даних поділяються на централізовані й розподілені. Централізована база даних зберігається у пам'яті однієї обчислювальної системи. Якщо ця обчислювальна система є компонентом мережі ЕОМ, можливий розподілений доступ до такої бази. Такий спосіб використання баз даних часто застосовують у локальних мережах ПК. Розподілена база даних складається з декількох, можливо пересічних або навіть дублюючих одна одну частин, які зберігаються в різних ЕОМ обчислювальної мережі. Робота з такою базою здійснюється за допомогою системи управління розподіленою базою даних. За способом доступу до даних бази даних поділяються на бази даних з локальним доступом і бази даних з віддаленим (мережевим) доступом. Системи централізованих баз даних з мережевим доступом припускають різні архітектури подібних систем: файл-сервер; клієнт-сервер [23].

База-даних, що має не менш важливе значення для повного функціонування системи, будується за допомогою sqlite. sqlite вбудована кросплатформена база-даних, популярність якої настільки велика, що кожен смартфон її використовує. Для забезпечення надійності роботи, sqlite проходить через декілька мільйонів тестів, що покриває 100 % коду. Коли додаток використовує SQLite, їх зв'язок проводиться за допомогою функціональних і прямих викликів файлів, що містять дані (наприклад, баз даних SQLite), а не якогось інтерфейсу, що підвищує швидкість і продуктивність операцій. sqlite має

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

прикладних сферах. Програми, написані на мові C ++, можна зустріти всюди: починаючи з батискафів на дні океану до космічних апаратів на поверхні Марса. Крім того, існує точний і повний міжнародний стандарт мови C ++, що не захищений патентом. Якісні та / або вільні реалізації цієї мови доступні для будь-яких комп'ютерів. Більшість концепцій програмування, які ви вивчите за допомогою мови C ++, можна безпосередньо використовувати в інших мовах, таких як C, C #, Fortran і Java [25].

Створюючи програми на більшості мов програмування, треба визначити, в якій операційній системі і на якому процесорі вони працюватимуть. Тільки визначивши це, можна долучити до програми виклик функцій з бібліотеки, призначеної для відповідної операційної системи. Наприклад: якщо розробляють програму для Windows, то можна використати бібліотеку Microsoft Foundation Classes; для роботи на платформі Macintosh – функції з Mac OS Toolbox. Після компіляції вихідних текстів отримуємо код, готовий до виконання на певному процесорі. Система Windows переважно працює на базі процесорів фірми Intel, комп'ютери Macintosh використовують процесори Motorola 680x0 або PowerPC. Створюючи програми на Java, можна не замислюватися над тим, в якій операційній системі вони працюватимуть. Java має власний набір машинно-незалежних бібліотек, які називають пакетами. Щодо процесорів ситуація аналогічна. Компілятор Java не генерує безпосередньо інструкції процесору. Він створює проміжний код – байткод для віртуальної машини Java (Java Virtual Machine – JVM). Виходячи з того, що ядро віртуальної машини Java реалізовано практично для всіх типів комп'ютерів, вважатимемо файли байткодів незалежними від платформи. Після виникнення в 1979 р. мови C++, до якої було додано порівняно з мовою C об'єктоорієнтовані засоби і збережено усі сильні сторони C, складається враження, що програмісти знайшли нарешті досконалу мову. Проте навіть могутня і популярна мова програмування має свої недоліки. Це важкі щодо розуміння і використання аспекти C++, пов'язані з керуванням пам'яттю і покажчиками. У C++ також легко зберігається процедурний стиль програмування. Згідно з думкою П.Нортон, наступним кроком у логічному

						ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			43

вільного ПЗ лежить ідея спільноти, яка ділиться своїми знаннями. Це одна з причин, за якими Python такий гарний: він був створений і постійно поліпшується співтовариством, яке хоче зробити його кращим. При написанні програми на Python ніколи не доведеться відволікатися на такі низькорівневі деталі, як управління пам'яттю, використовуваної програмою. Завдяки своїй відкритій природі, Python був імпортований на багато платформ (тобто змінений таким чином, щоб працювати на них). Всі програми зможуть запускатися на цих платформах без будь яких змін, якщо тільки уникати використання системно-залежних функцій. Python можна використовувати в GNU/Linux, Windows, FreeBSD, Macintosh, Solaris, OS/2, Amiga, AROS, AS/400, BeOS, OS / 390, z/OS, Palm OS, QNX, VMS, Psion, Acorn RISC OS , VxWorks, PlayStation, Sharp Zaurus, Windows CE і навіть на PocketPC. Програма, написана на компільованій мові програмування, як наприклад, С або С++, перетворюється з вихідної мови (тобто С або С++) в мову, зрозумілу комп'ютеру (бінарний код, тобто нулі і одиниці) за допомогою компілятора із застосуванням різноманітних прапорів і параметрів. Коли запускається така програма, компоновальник/завантажувач копіює її з диска в оперативну пам'ять і запускає її. Python не вимагає компіляції в бінарний код. Програма виконується з початкового тексту. Python сам перетворює цей вихідний текст в деяку проміжну форму, звану байткод, а потім переводить його на машинну мову і запускає. Все це помітно полегшує використання Python, оскільки немає необхідності клопотатись про компіляції програми, підключенні і завантаженні потрібних бібліотек і т.д. Разом з тим, це робить програми на Python набагато більш переносними, так як досить їх скопіювати на інший комп'ютер і вони працюють. Мова програмування Python підтримує як процедурно-орієнтоване, так і об'єктно-орієнтоване програмування. У процедурно-орієнтованих мовах програми будуються на основі процедур або функцій, які представляють собою багаторазово використовувані фрагменти програми. В об'єктно-орієнтованих мовах програмування програми будуються на основі об'єктів, які об'єднують в собі дані і функціонал. Python надає прості але потужні

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

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 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.

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Система буде функціонувати таким чином, відправляється запит на відображення веб-ресурсу до сервера провайдера, запит, який містить доменне ім'я, приймає система та перевіряє чи наявне запитуване доменне ім'я в базі даних. У випадку, якщо доменне ім'я наявне в базі даних – система, за допомогою реалізованого DNS-серверу, повертає DNS-відповідь, яка містить в собі адресу локального серверу.

База даних «чорний список», котрий містить заборонені IP/URL. Для роботи з базою даних передбачається графічний інтерфейс. Перше, що необхідно зробити – ввести пароль, якщо пароль введений правильно – відображається головне меню, інакше – повідомлення про неправильний пароль. В головному меню є можливість додавання доменного імені та IP, пошук доменного імені, видалення доменного імені, завантаження записів, виведення інформації про розробника.

3.2 Розробка структурної схеми

Структурна схема, яка зображена на рисунку 3.1, являє собою графічний опис структури системи. На структурній схемі зображені користувачі, локальний просторій, провайдер, сукупність серверів, що складають мережу Інтернет та веб-сторінка, яка зберігається на одному з серверів в мережі Інтернет.

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

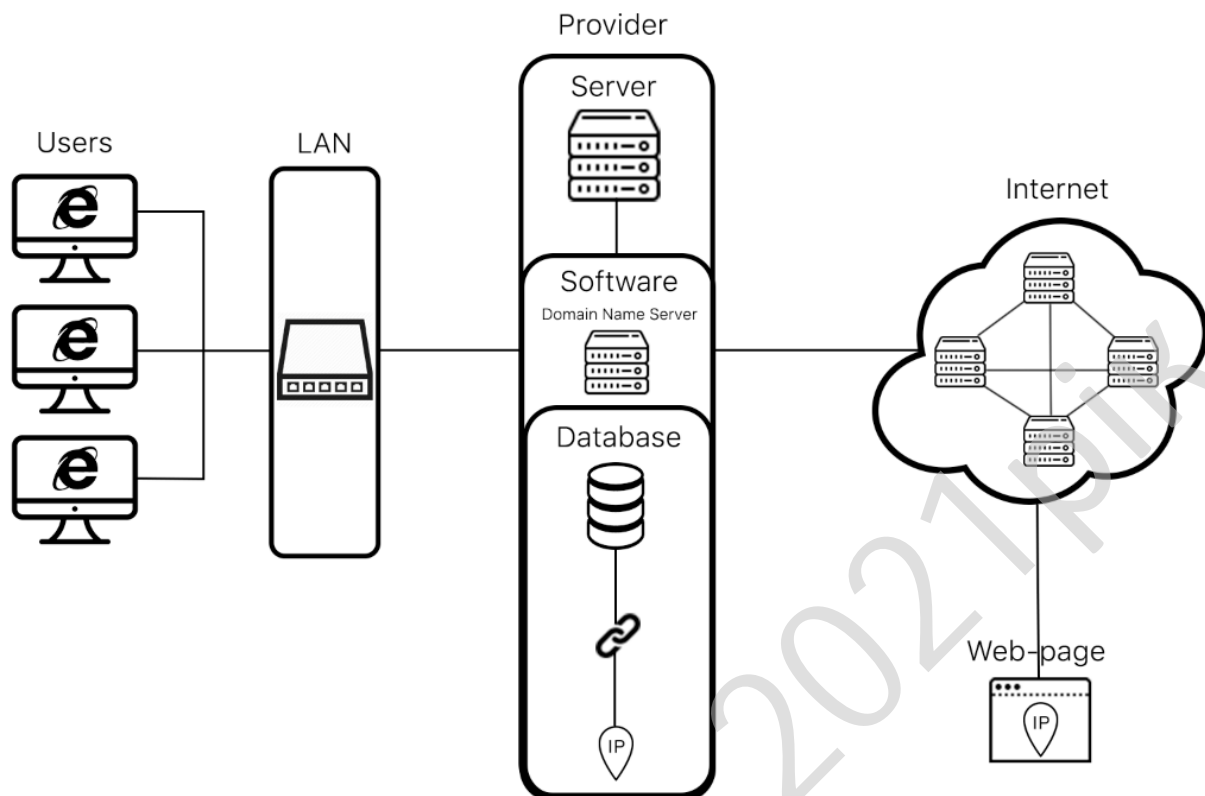


Рисунок 3.1 – Структурна схема системи

В свою чергу, провайдер містить в собі сервер, на якому встановлена власне сама система, що складається з DNS-серверу та бази даних, у якій зберігаються доменні ім'я та IP, доступ до яких повинен бути заблокований.

3.3 Розробка функціональної схеми

Функціональна схема, що зображена на рисунку 3.2, має наступні блоки: DNS-сервер та базу даних. DNS-сервер в свою чергу містить:

- відповідь DNS-серверу;
- завантаження бази даних;
- отримання зони;
- запит до DNS-серверу;
- отримання імені домену та тип запиту;
- отримання запису;

3.4 Розробка діаграми процесів

Діаграма процесів, що зображена на рисунку 3.3, включає в себе діаграму DNS-серверу, що складається з: початок – на цьому етапі створюється сокет, локальна адреса та порт 53 зв'язуються з цим сокетом.

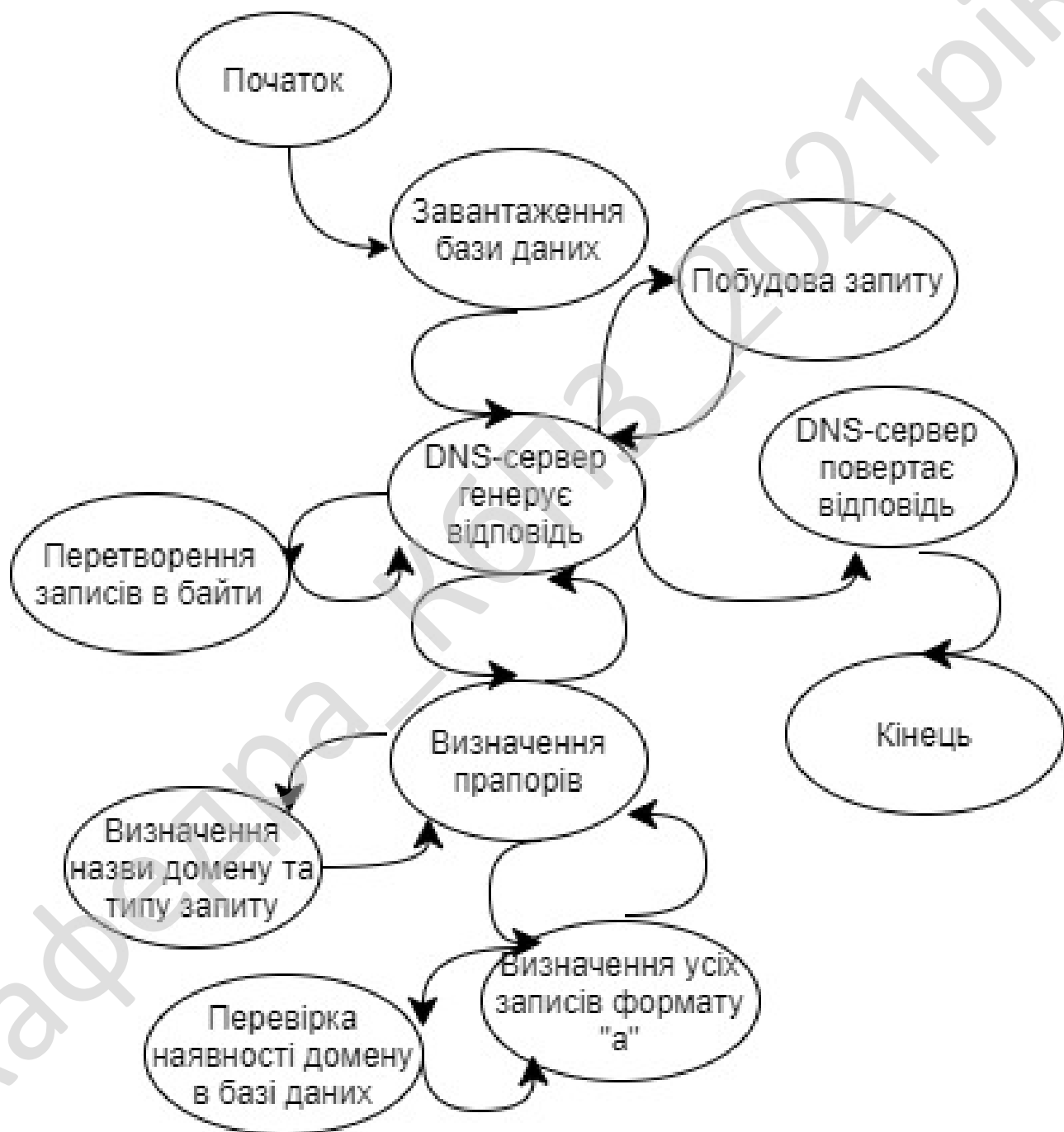


Рисунок 3.3 – Діаграма процесів системи

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРМ-123.21.0010.00.00.ПЗ

Арк.

52

Після чого починається прослухуватись порт 53; завантаження бази даних – відбувається підключення до бази даних та у циклі база даних завантажується у відповідну змінну; відповідь DNS-серверу – визначається ID транзакції, DNS-запит, прапори, кількість серверів, кількість допоміжних записів та генерується тіло DNS, що включає ім'я домену, тип запису, ttl запису, значення запису, голова DNS; отримання прапорів – визначаються прапори за допомогою методів для роботи з байтовими значеннями; отримання імені домену та типу запиту – посимвольно генерується ім'я домену та визначається тип запиту завдяки використанню зрізів; отримання записів; отримання зони – повернення зони, що відповідає доменному імені; запит до DNS-серверу – генерація запиту у вигляді байтів; перетворення записів в байти – перетворення доменного імені, ttl, тип запису, значення запису в байти для подальшого оброблення; кінець – сформована відповідь у вигляді байтів відправляється на сокет за адресою з якої прийшов запит.

Окрім DNS-серверу діаграма включає базу даних, що складається з: початок – підключення бібліотек, зокрема PyQt5, а також sys, sqlite3, time; входження в базу даних – відображається діалогове вікно з проханням ввести пароль; головне вікно – вікно, яке містить функціонал для роботи з БД та являє собою інтерфейс; пошук домену – метод за допомогою якого можна перевірити чи наявний домен в базі даних чи ні; видалення домену – метод за допомогою якого можна видалити домен із бази даних; додавання домену та адреси – метод за допомогою якого можна додати домен та IP-адресу до бази даних; оновлення даних – після того як домен був видалений чи створений, потребується оновити дані; завантаження записів – завантаження вже створених доменних імен та IP-адрес; діалогове вікно – вікно, що містить інформацію про автора.

Була розроблена і описана структурна, функціональна схема та діаграма процесів. Функціонування системи: відправляється запит на відображення веб-ресурсу до сервера провайдера, запит, який містить доменне ім'я, приймає система та перевіряє чи наявне запитуване доменне ім'я в базі даних. У випадку,

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

якщо доменне ім'я наявне в базі даних – система, за допомогою реалізованого DNS-серверу, повертає DNS-відповідь, яка містить в собі адресу локального серверу.

База даних «чорний список», котрий містить заборонені IP/URL. Для роботи з базою даних передбачається графічний інтерфейс. Перше, що необхідно зробити – ввести пароль, якщо пароль введений правильно – відображається головне меню, інакше – повідомлення про неправильний пароль. В головному меню є можливість додавання доменного імені та IP, пошук доменного імені, видалення доменного імені, завантаження записів, виведення інформації про розробника.

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.

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

Для розробки програмного забезпечення системи блокування вхідних вузлів Tor і серверів VPN-провайдерів було створено одразу дві блок-схеми. Перша блок-схема зображена на рисунку 4.1 і вона являє собою DNS-сервер. Початок – на цьому етапі створюється сокет, локальна адреса та порт 53 зв'язуються з цим сокетом. Після чого починається прослухуватись порт 53; завантаження бази даних – відбувається підключення до бази даних та у циклі база даних завантажується у відповідну змінну; відповідь DNS-серверу – визначається ID транзакції, DNS-запит, прапори, кількість серверів, кількість допоміжних записів та генерується тіло DNS, що включає ім'я домену, тип запису, ttl запису, значення запису, голова DNS; отримання прапорів – визначаються прапори за допомогою методів для роботи з байтовими значеннями; отримання імені домену та типу запиту – посимвольно генерується ім'я домену та визначається тип запиту завдяки використанню зрізів; отримання записів; отримання зони – повернення зони, що відповідає доменному імені; запит до DNS-серверу – генерація запиту у вигляді байтів; перетворення записів в байти – перетворення доменного імені, ttl, тип запису, значення запису в байти для подальшого оброблення; кінець – сформована відповідь у вигляді байтів відправляється на сокет за адресою з якої прийшов запит.

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55



Рисунок 4.1 – Блок-схема DNS-сервера

Друга блок-схема, що зображена на рисунку 4.2, відповідає за алгоритм роботи ПЗ, яке представляє собою інтерфейс для роботи з базою даних.

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРМ-123.21.0010.00.00.ПЗ

Арк.

56

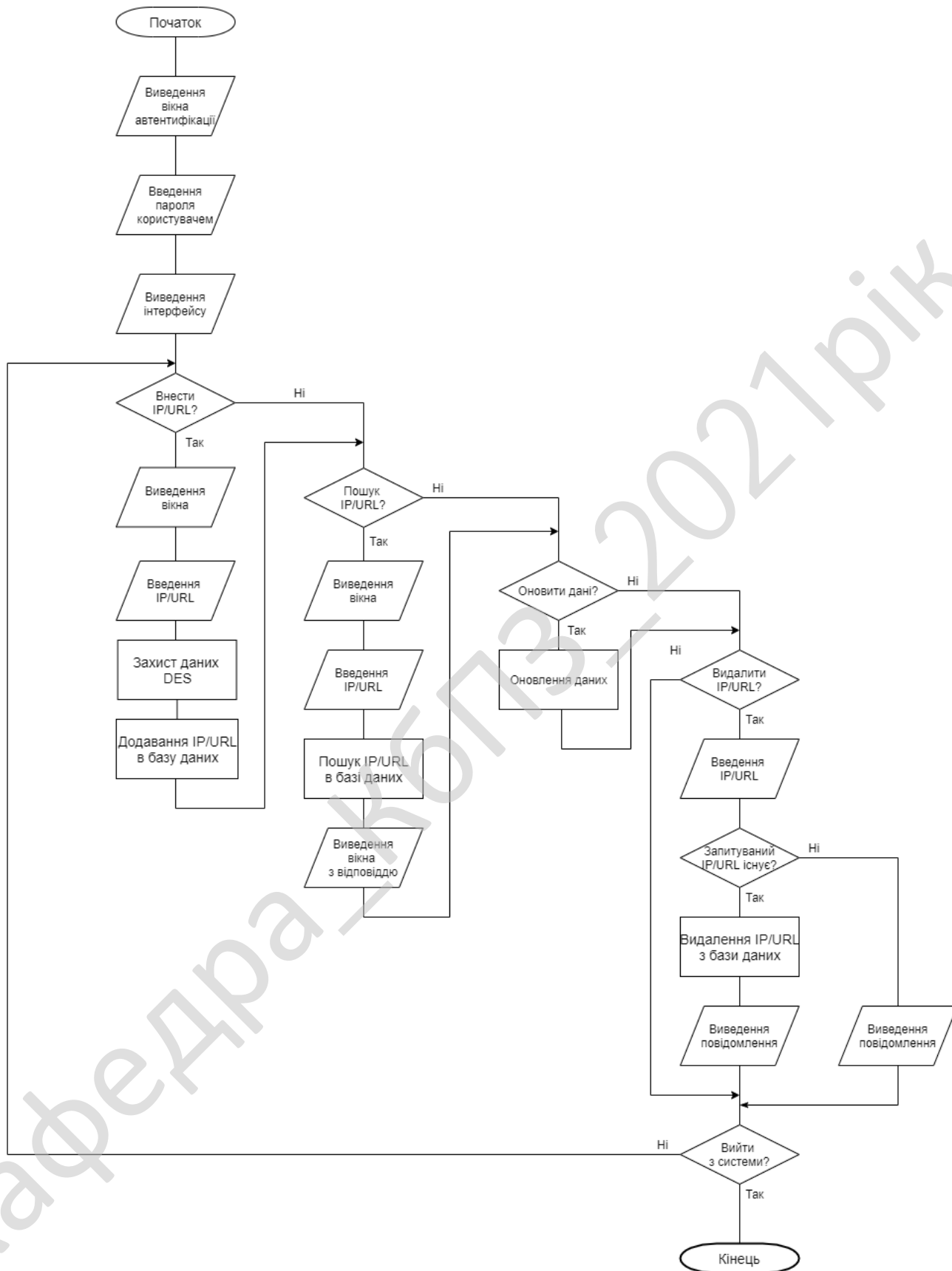


Рисунок 4.2 – Блок-схема бази даних

Початок – підключення бібліотек, зокрема PyQt5, а також sys, sqlite3, time; виведення вікна автентифікації – користувач має ввести пароль для початку роботи з БД; введення пароля користувача – перевірка введенного користувачем пароля; виведення інтерфейсу – виведення головного меню; внести IP/URL – користувач має можливість внести новий IP/URL; виведення вікна – у випадку якщо користувач вирішив внести новий IP/URL; введення IP/URL; захист даних. DES – введенні дані захищаються алгоритмом шифрування DES; додавання IP/URL в базу даних – метод за допомогою якого зашифровані дані вносяться до бази даних; пошук IP/URL – користувач має можливість перевірити чи наявний домен в базі даних чи ні; виведення вікна – у випадку якщо користувач вирішив перевірити чи наявний домен в БД; введення IP/URL; пошук домену – метод за допомогою якого перевіряється наявність IP/URL в БД; виведення вікна з відповіддю – чи був знайдений IP/URL; оновити дані – користувач має можливість оновити дані для відображення їх в ПЗ; видалення IP/URL – користувач має можливість видалити дані з БД; головне вікно – вікно, яке містить функціонал для роботи з БД та являє собою інтерфейс; пошук домену – метод за допомогою якого можна видалити домену – метод за допомогою якого можна видалити домен із бази даних; введення IP/URL – у випадку, якщо користувач вирішив видалити дані; запитуваний IP/URL існує – перевірка чи існує IP/URL, який ввів користувач; виведення повідомлення – у випадку, якщо дані не були знайдені; видалення IP/URL з бази даних – метод за допомогою якого видаляються дані з БД, якщо вони були знайдені; виведення повідомлення – про успішне видалення; вийти з системи – користувач має можливість вийти з системи.

4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використовувати алгоритм Khufu. Khufu – це 64-бітовий блоковий шифр. 64-бітовий відкритий тест спочатку розщеплюється на дві 32-бітові половини, L і R .

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

Над обома половинами й певними частинами ключа виконується операція XOR. Потім, аналогічно DES, результати проходять деяку послідовність раундів. У кожному раунді молодший значущий байт L використовується як вхід S-блоку. У кожного S-блоку 8 вхідних біт і 32 вихідних біта. Далі обраний в S-блоці 32-бітовий елемент піддається операції XOR з R . Потім L циклічно зрушується на число, кратним восьми біткам, L і R міняються місцями, і раунд завершується. Сам S-блок не статичний, він міняється кожні вісім раундів. Нарешті, по закінченні останнього раунду, над L і R виконується операція XOR з іншими частинами ключа, і половини поєднуються, утворюючи блок шифртексту.

Хоча частини ключа використовуються для операції XOR із блоком шифрування на початку й кінці виконання алгоритму, головне призначення ключа – генерація S-блоків. Ці S-блоки секретні, по суті, це частина ключа. Повний розмір ключа алгоритму Khufu дорівнює 512 біт (64 байт), алгоритм надає спосіб генерації S-блоків по ключу.

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

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

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

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

При впровадженні програмного забезпечення потрібно урахувати наступні дії:

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

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

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

Система не потребує встановлення допоміжних додатків, інструментів, бібліотек. Все необхідне знаходиться в одному каталозі, який має структуру, що зображена на рисунку 5.1.

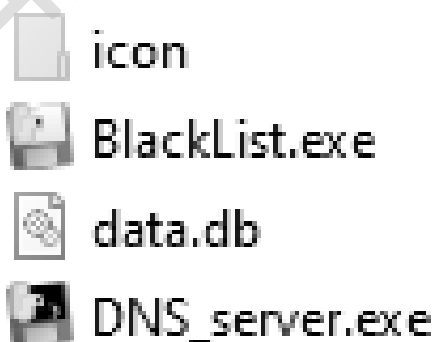
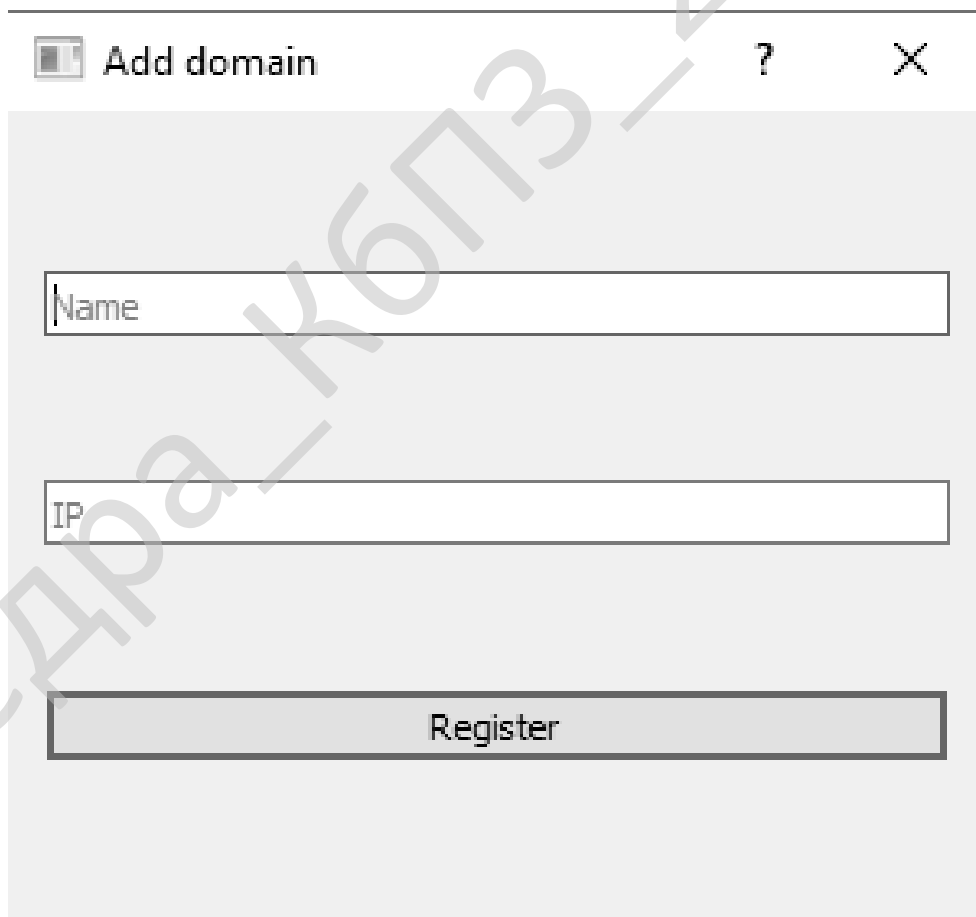


Рисунок 5.1 – Структура каталогу з програмним забезпеченням

Таким чином, для впровадження системи в експлуатацію треба перенести каталог з ПЗ на комп'ютер, а далі слідувати інструкціям використання ПЗ.

Інструкція використання програмного засобу

Для функціонування системи необхідно попередньо налаштувати базу даних. Зробити це можливо за допомогою BlackList.exe. Налаштування включає в себе додавання URL/IP, доступ до яких необхідно обмежити. Для цього програмне забезпечення містить кнопку «Додати IP/URL». По натисканню на кнопку відображається вікно з двома полями: «Name» та «IP», що зображене на рисунку 5.2. В перше поле вводиться ім'я домену, в друге вводиться IP домену. По натисканню на кнопку «Register» дані вносяться в базу даних та відображається повідомлення про успішне додавання.



The image shows a screenshot of a Windows-style dialog box titled "Add domain". The dialog box has a standard title bar with a minimize button, a question mark icon, and a close button (X). Inside the dialog, there are two text input fields. The first field is labeled "Name" and the second field is labeled "IP". Below these fields is a button labeled "Register".

Рисунок 5.2 – Вікно реєстрації URL/IP

Після того, як усі необхідні IP/URL були внесені в базу даних, файл під назвою data.db, налаштування вважається здійсненим. Залишається увімкнути DNS-сервер. Це можливо зробити за допомогою DNS_server.exe. Якщо після вмикання відображається командний рядок – при запуску ніяких проблем не виникло.

На рисунку 5.3 зображено вікно автентифікації, що потребує пароль від користувача для подальшої роботи з розробленим програмним забезпеченням системи блокування вхідних вузлів Tor і серверів VPN-провайдерів.

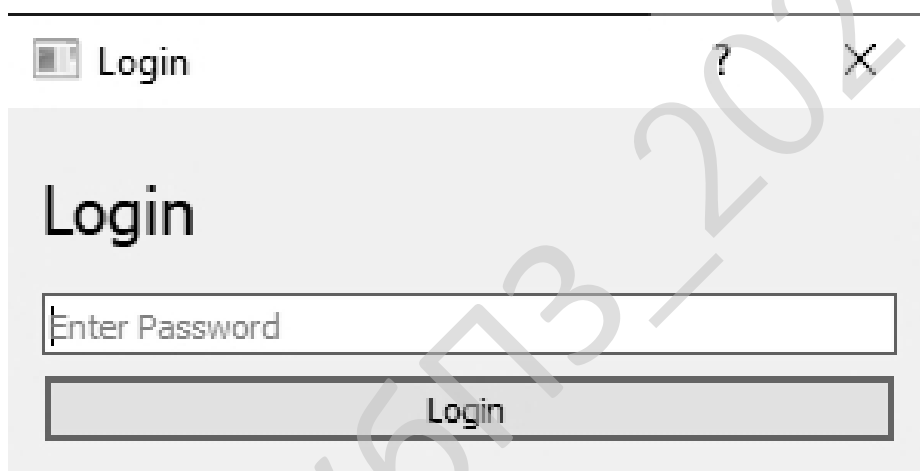


Рисунок 5.3 – Вікно автентифікації

Після автентифікації можна побачити головне меню програмного забезпечення, яке зображене на рисунку 5.4. Інтерфейс головного вікна розподілено на наступні функціональні розділи:

- 1) меню: файл; довідка;
- 2) кнопки: додати; оновити; пошук; видалити;
- 3) кнопки управління вікном: згорнути; розгорнути; вихід.

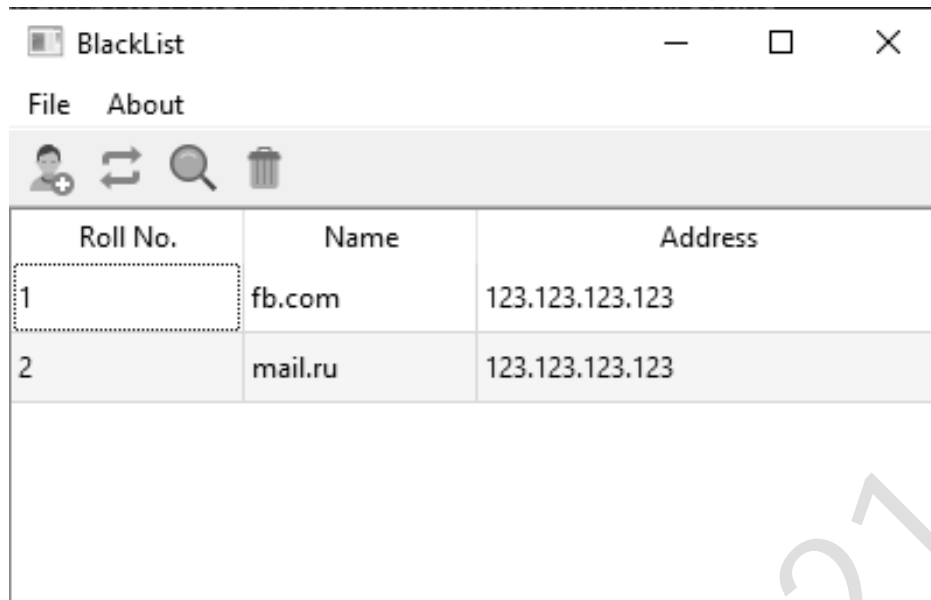


Рисунок 5.4 – Головне вікно ПЗ

На рисунку 5.5 зображено авторські дані розробленого програмного забезпечення.

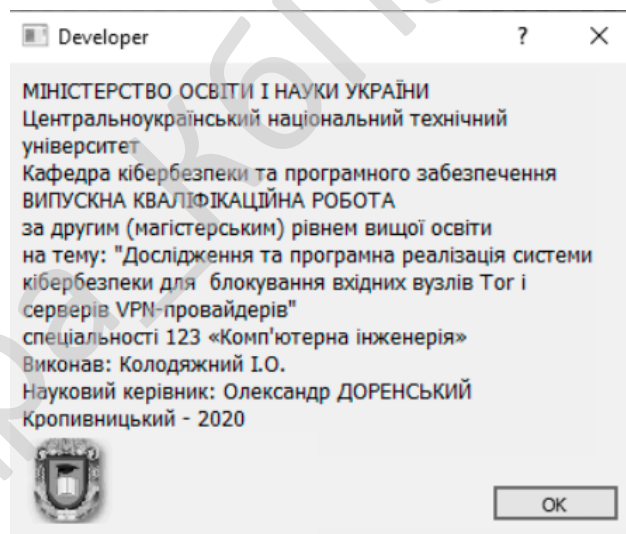


Рисунок 5.5 – Авторське право

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.

Система постачається з усім необхідним для роботи. Для впровадження системи необхідно розмістити каталог з ПЗ в зручному для користувача місці на комп'ютері/сервері.

Після чого систему необхідно налаштувати. Налаштування включає в себе додавання в БД IP/URL ресурсів, доступ до яких треба обмежити. Після вдалого налаштування користувач має можливість увімкнути DNS-сервер. Для роботи з DNS-сервером є виконуваний файл DNS_server.exe, в свою чергу для роботи з БД – BlackList.exe.

Кафедра КБПЗ – 2021 рік

					VKPM-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи кібербезпеки для блокування вхідних вузлів Tor і серверів VPN-провайдерів.

Метою розробки є дослідження та програмна реалізація системи кібербезпеки для блокування вхідних вузлів Tor і серверів VPN-провайдерів.

Об'єктом дослідження є процес кібербезпеки для блокування вхідних вузлів Tor і серверів VPN-провайдерів.

Предметом дослідження є методи кібербезпеки для блокування вхідних вузлів Tor і серверів VPN-провайдерів.

Методи дослідження базуються на методах теорії захисту інформації, методах математичної статистики, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

– Удосконалено метод кібербезпеки для блокування вхідних вузлів Tor і серверів VPN-провайдерів.

– Розроблено вітчизняний продукт кібербезпеки для блокування вхідних вузлів Tor і серверів VPN-провайдерів, який має більш широкі можливості, на відміну від існуючих аналогів.

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

7 ДАНІ ПРО ЕКОНОМІЧНУ ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ

7.1 Техніко-економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти

Після ознайомлення з підприємством та засобами розробки програмної продукції був розроблений план розробки програми. Був підрахований необхідний час для розробки та впровадження програми. Цей час склав 60 днів (три місяці).

В магістерській роботі було проведене дослідження та виконана програмна реалізація системи кібербезпеки для блокування вхідних вузлів Tor і серверів VPN-провайдерів. Розроблене програмне забезпечення має достатню надійність і задовольняє усім поставленим умовам, а саме:

- а) невеликий розмір;
- б) невеликі системні потреби;
- в) незалежність від встановлених на комп'ютері баз даних;
- г) зручність у користуванні та надійність

Таблиця 7.1 – Початкові дані

Показники	Позначення	Характеристика або величина
1	2	3
1. Кількість розроблених програм період, шт	N	1
2. Кількість екземплярів програм, шт	Ne	40 (Варіант №10)
3. Запланований термін розробки, днів	Fpq	60 (3 місяці)
4. Група задачі підсистеми управління (1-6)	–	1
5. Ступінь новизни задачі (А, Б, В, Г)	–	Б
6. Складність алгоритму (1, 2, 3)	–	2
7. Кількість макетів вхідної інформації	–	3

Продовження табл. 7.1

1	2	3
8. Кількість форм вихідної інформації.	–	4
9. Мова програмування (1-6)	–	4
10. Попередній досвід (1-6)	–	3
11. Гнучкість проекту ПП (1-6)	–	3
12. Детальність проекту ПП (1-6)	–	2
13. Рівень спрацьованості колективу (1-6)	–	2
14. Ступінь вимірності процесів (1-6)	–	3
15. Необхідна надійність програмного забезпечення (1-6)	–	2
16. Розмір бази даних (порівняно з розміром програми) (1-6)	–	2
17. Складність кінцевого програмного продукту (1-6)	–	2
18. Необхідний рівень забезпечення повторного використання (1-6)	–	2
19. Документованість відповідно до планованого життєвого циклу (1-6)	–	2
20. Вимоги до швидкодії ПП (1-6)	–	2
21. Обмеження на розміри основного сховища даних (1-6)	–	2
22. Різноманітність використовуваних обчислювальних платформ (1-6)	–	2
23. Професійний рівень аналітиків (1-6)	–	2
24. Професійний рівень програмістів (1-6)	–	2
25. Постійність складу команди розробників (1-6)	–	2
26. Досвід розробки додатків (1-6)	–	2
27. Досвід роботи з обчислювальною платформою (1-6)	–	2

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРМ-123.21.0010.00.00.ПЗ

Арк.

68

Продовження табл. 7.1

1	2	3
28. Досвід роботи з мовою і інструментами середовища розробки (1-6)	–	2
29. Досвід роботи з програмними інструментами розробки (1-6)	–	3
30. Розробка ПО для декількох серверів одночасно (1-6)	–	2
31. Вимоги до дотримання встановленого графіка робіт (1-6)	–	2
32. Вартість ПЗ у розробника (НМА), грн	–	40000 (Варіант №10)
33. Норматив додаткової зарплати, % :	Нд	10
34. Норматив відрахувань у соціальні фонди, %	Нс	37
35. Норматив загальногосподарських витрат, %	Нг	15
36. Норматив витрат на освоєння нових мов програмування, %	Нп	15
37. Рівень рентабельності програмної продукції, %	Ре	55
38. Ставка податку на додану вартість, %	Ндв	20

7.2 Розрахунок трудомісткості розробки програмної продукції

Значення трудомісткості розробки програмного забезпечення для стадій ТЗ, ЕК, ТП та ВП визначаємо по типовим нормам часу приведеним в додатках МВ. Стадія РП є найбільш тривалою і трудомісткою, що робить значний вплив на інші стадії проекту.

Визначимо трудомісткість розробки ПЗ для стадії РП.

Обчислюємо номінальні трудовитрати, люд-міс.:

$$T_{ном} = A \text{ Size}^B, \quad (7.1)$$

де A – коефіцієнт Боєма, $A=2,45$; $Size$ – загальний об'єм відлагодженого програмного коду, тис. рядків; B – показник ступеня, що визначається співвідношенням

$$B = 1,01 + 0,001 \sum W_i \quad (7.2)$$

де W_i – сумарне значення п'яти показників (МВ, додаток 2), що відображають особливості розробки проекту програмного продукту (ПП) і колективу розробників.

$$B = 1,01 + 0,001(2,43 + 3,64 + 3,38 + 3,95 + 2,73) = 1,026$$

$$T_{ном} = 2,45 \cdot 2,7^{1,026} = 6,78 \text{ люд-міс.}$$

Визначаємо уточнені (з урахуванням приведених в МВ додатку 3 сімнадцяти додаткових коефіцієнтів) трудовитрати, люд-міс.:

$$T_{уточн} = T_{ном} \prod V_j, \quad (7.3)$$

де $\prod V_j$ – добуток сімнадцяти додаткових коефіцієнтів, приведених в МВ додатку 3.

$$T_{уточн} = 6,78 \cdot (0,88 \cdot 0,93 \cdot 0,88 \cdot 0,91 \cdot 0,95 \cdot 1,1 \cdot 0,87 \cdot 1,22 \cdot 1,16 \cdot 1,1 \cdot 1,1 \cdot 1,12 \cdot 1,1 \cdot 1,1 \cdot 1,1) = 9,37 \text{ люд-міс.}$$

Ці коефіцієнти дозволяють диференційовано оцінювати результати роботи програмістів, беручи до уваги швидкодію програми, використання різноманітних обчислювальних платформ і інструментів розробки, взаємодію декількох серверів, вимоги до об'ємів баз даних і ін.

Визначаємо підсумкові трудовитрати по стадії робочий проект, люд-дні:

$$T_{РП} = 0,3 C T_{уточн}^{0,33+0,2(B-1,01)} S, \quad (7.4)$$

де C – визначений емпірично коефіцієнт, запропонований авторами методики, (МВ, додаток 4); S – коефіцієнт стиснення (або подовження) графіка робіт %, що дозволяє коректувати терміни розробки ПО згідно встановленим вимогам. Вибираємо в межах (25...350)%

$$T_{РП} = 0,3 \cdot 2,66 \cdot 9,37^{0,33+0,2(1,026-1,01)} \cdot 100 = 168 \text{ люд/день}$$

Для зручності визначення загальної трудомісткості на розробку програмного забезпечення результати розрахунків по стадіям зводимо до таблиці 7.2.

						ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			70

Таблиця 7.2 – Визначення трудомісткості розробки програмного забезпечення

Стадії розробки	Трудомісткість за типовими нормами та розрахунками	
	Величина, люд/дні	Підстава
Технічне завдання	9	Д5
Ескізний проект	10	Д6
Технічний проект	9	Д7
Робочий проект	168	Ф 7.1-7.4
Впровадження	13	Д13
Всього	209	–

7.3 Визначення чисельності виконавців і планового фонду зарплати

Чисельність ставок інженерів-програмістів для розробки програмного забезпечення визначається за формулою

$$Ч = \frac{T_{пз} \cdot N}{F_{рч} - H_{ев}}, \quad (7.5)$$

де $F_{рч}$ – плановий фонд робочого часу одного спеціаліста, днів,
 $T_{пз}$ – трудомісткість розробки програмного забезпечення люд-дні,

$$Ч = \frac{209 \cdot 1}{60 - 5} = 3,8 \text{ ставки}$$

Чисельність інженерів-електронщиків для проведення технічного обслуговування та ремонту комп'ютерних мереж визначається в залежності від наявності технічних засобів і норм витрат часу на виконання профілактичних робіт на протязі року.

Визначаємо затрати часу на виконання профілактичних робіт по обслуговуванню обладнання за період розробки. Результати розрахунку зводимо до таблиці 7.3

Таблиця 7.3 – Затрати часу на виконання профілактичних робіт по обслуговуванню обладнання за розрахунковий період

Найменування обладнання	Профілактичне обслуговування			
	Кількість хв. на один. обл.	Кількість обладнан ня	Затрати часу в хв.	Затрати часу в год.
Системний блок ПК	385	12	4620	77
Монітор	160	12	1920	32
Клавіатура	140	12	1680	28
Маніпулятор «мишка»	30	12	360	6
Принтер матричний	185	1	185	3
Принтер лазерний	355	2	710	12
Принтер струминний	300	1	300	5
Сканер	155	2	310	5
Концентратор-маршрутизатор	155	2	310	5
Кабельні господарства ЛОМ на 1 м. п.	2,5	100	250	4
Кабельне господарство електромережі	48	50	2400	40
Копіювальний апарат	285	2	570	10
Усього за рік:			3 _ч	227

Час на профілактику обладнання в загальному балансі робочого часу інженерів-електронщиків не повинен складати більше 10%

Виходячи з цього фонд робочого часу інженерів-електронщиків складає:

$$\Phi_{op}^c = \frac{3_{ч} \cdot n_{mic}}{1,2} \quad (7.6)$$

$$\Phi_{op}^c = \frac{227 \cdot 3}{1,2} = 567,5 \text{ год}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

$$Ч_{ел} = \frac{\Phi_{др}^c}{F_{др} \cdot T_{зм}} \quad (7.7)$$

$$Ч_{ел} = 567,5 / (60 \cdot 8) = 1,2 \text{ ставки}$$

Для забезпечення нормального технічного обслуговування засобів ТО та мереж, необхідно прийняти найбільше ціле значення розрахункової чисельності інженерів – електронщиків.

Чисельність інженерів-системотехніків, адміністраторів мережі, дизайнерів WEB вузлів, системних програмістів (аналітиків), бухгалтерів-економістів визначається за потребою в залежності від функціональних обов'язків. Після визначення чисельності персоналу складається штатний розклад.

Таблиця 7.4 – Розрахунок чисельності штатного персоналу сектору системного та адміністративного обслуговування засобів ОТ та комп'ютерних мереж

Посада	Вид роботи	Час	Кількість штатних одиниць
Адміністратор загальної мережі, аналітик	Адміністрування локальної мережі, поштового та серверу DNS (ОС FreeBSD), маршрутизатора Cisco, серверу доступу АДСЛ (ОС Linux), Wi-Fi налаштування ADSL, VPN, PPPoE, Frame Relay	2	0,5
	Налаштування і конфігурування базової станції безпроводного зв'язку (СМТS)	0,5	
	Розробка та впровадження проектів з організації зв'язку між віддаленими об'єктами, ЛОМ	0,5	
	Забезпечення цілодобової роботи зв'язку клієнтів д мережі Інтернет	1	
Всього		4	

Продовження таблиці 7.4

Посада	Вид роботи	Час	Кількість штатних одиниць
Продакт-менеджер	Презентації нової продукції, пошук каналів збуту	1	0,25
	Підтримка постійних клієнтів	0,5	
	Оформлення договорів, ведення тендерів	0,25	
	Контроль взаєморозрахунків з постачальниками	0,25	
Всього		2	
Дизайнер WEB	Розробка концепції оформлення та інтерфейсу сайту, оптимізація дизайну існуючих, проектує їх структуру та навігацію	1	0,25
	Створення графічних і стилістичних елементів сайту	0,5	
	Оформлення банерів і промо-сторінок	0,25	
	Розміщення графіки і контенту на Інтернет сторінках	0,25	
Всього		2	
Інженер верстальник	Розробка та верстка макетів рекламної продукції та технічної документації	1	0,25
	Верстка друкованих видань	0,5	
	Додрукова підготовка макетів	0,25	
	Розміщення графіки і контенту на Інтернет сторінках	0,25	
Всього		2	

Складемо штатний розклад виконавців:

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

Таблиця 7.5 – Штатний розклад виконавців

Посада	Кількість ставок	Середньо-місячний оклад, грн.	Всього за період розробки, грн.
Керівник (ІТ-менеджер)	1	12000	36000
Продакт-менеджер	0,25	8000	6000
Інженер-програміст	3,8	8000	91200
Інженер-електронщик	1,2	6000	21600
Інженер-системотехнік	0,25	6000	4500
Адміністратор мережі	0,5	6000	9000
Системний програміст	0,25	6000	4500
Дизайнер WEB	0,25	8000	6000
Інженер-верстальник	0,25	6000	4500
Бухгалтер-економіст	0,5	10000	15000
Всього за період розробки	$R_{cn}=8,25$	-	$\Phi_{роб}=198300$

Розрахуємо середньоденну зарплату одного виконавця:

$$z_{cd} = \frac{\Phi_{роб}}{R_{cn} \cdot F_{pq}}, \quad (7.8)$$

де $\Phi_{роб}$ – загальна сума зарплати за плановий період, грн.

$$z_{cd} = \frac{198300}{8,25 \cdot 60} = 400 \text{ грн.}$$

7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника

Балансова вартість будівель визначається з урахуванням кількості робочих місць виконавців, питомої площі на одне робоче місце, та вартості одного квадратного метра виробничої площі

$$B_{y\delta} = R_{cn}^1 S_y C_{пл}, \quad (7.9)$$

де R_{cn}^1 – кількість робочих місць виконавців, шт. Приймаємо 8 робочих місць. S_y – питома площа на одне робоче місце, m^2 ; $C_{пл}$ – вартість одного квадратного метра площі, грн.

Згідно даних ТОВ науково-дослідницького консалтингового підприємства «Пектораль» ціна одного квадратного метра площі новобудови, вік якої не перевищує 25 років, по місту складає 800...1600 у.о./ m^2 . Враховуючи, що курс складає 1 у.о. = 25 грн. приймаємо для розрахунку вартість одного метра квадратного рівною 20000 грн./ m^2 . На кожне робоче місце у середньому потрібно 8 m^2 . З урахуванням цього:

$$B_{y\delta} = 8 \cdot 8 \cdot 20000 = 1280000 \text{ грн.}$$

Вартість передавальних пристроїв складає 10% від вартості будівель, і у даному випадку вона складе: 128000 грн.

Балансова вартість інвентарю розраховується за нормою 3500 грн на одне робоче місце. Тобто

$$I_{нв} = R_{cn}^1 \cdot C_m, \quad (7.10)$$

де C_m – ціна меблів для одного робочого місця, грн.

$$I_{нв} = 8 \cdot 3500 = 28000 \text{ грн}$$

Балансова вартість обчислювальної техніки визначається по оптовим цінам постачальника з врахуванням витрат на транспортування.

Специфікація на обчислювальну техніку наведена в таблиці 7.7. Дані по оптовій ціні на обладнання та комплектуючі вибирались за прайсом фірми Компбест за 06.11.21 – джерело <https://compbest.com.ua/>.

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

Таблиця 7.6 – Специфікація

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Персональний комп'ютер		10947
Системний блок		7347
Процесор	Intel Core i5-3470: 4 ядра (4 потоки) по 3.2 – 3.6 GHz, 6mb Cache.	-
Системна плата	Intel, USB, VGA, DVI, RJ-45, audio, LGA1155	-
Відеокарта	GIGABYTE GV-N710 GeForce GT710 1 GB DDR4 64-bit D-Sub+HDMI+DVI	-
Жорсткий диск	240 SSD+HDD Seagate Barracuda 750 Gb 7200 32Mb SATAII ST3750528AS (ST3750528AS)	-
Оперативна пам'ять	DIMM 4Gb DDR3 , non-Reg., no-ECC , CL 9 (2 модулі)	-
DVD-привод	DVD -RW/+RW , LG SATA SuperMulti Bulk 22x, SecurDisc, black	-
Корпус	GRESSO GE-7525, 500W (120mm big fan), 2xIDE, full-ATX,БЖ 2xSATA, 1xFDD, Air Duct, 2xUSB 2.0, Mic+Audio, silver/black	-
Кардрідер внутрішній	USB 2.0 Card reader STORM CR -35U1A4-B, int. 3.5", 1*USB2.0+AUDIO+1394, multi: All Type Cards, black	220
інше	Клавіатура, мишка	-

Продовження таблиці 7.6

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Монітор	LG W2363V-WF Wide LCD 2ms, 70 000:1, 300кд/м2, 170/160, D-Sub / Glossy White	3600
Принтер лазерний	Canon i-SENSYS LBP6030W	2700
Принтер струменевий	Epson Stylus Photo P50 (C11CA45341) + USB cable	5500
Копіювальний апарат	Canon i-SENSYS MF217W with Wi-Fi	5965

Витрати на транспорт, монтаж та випробування можуть бути прийняті в межах до 10% від оптової ціни.

Для визначення необхідної кількості капітальних вкладень складемо таблицю 7.8.

Таблиця 7.7 – Балансова вартість обчислювальної техніки

Найменування обчислювальної техніки	Кількість, шт.	Ціна за одиницю, грн.	Витрати на транспортування, монтаж та випробування.	Загальна вартість, грн.
Персональні комп'ютери	8	10947	8757,6	96333,6
Принтер лаз.	2	2700	540	5940
Принтер струм.	1	5500	550	6050
Копіюв. апарат	1	5965	596,5	6561,5
Всього	–	–	–	114885,1

Таблиця 7.8 – Вартість основних фондів та амортизаційні відрахування розробника

Групи та види основних фондів	Балансова вартість, грн.	Амортизація	
		Норма, %	Відрахування, грн.
Група 3			
1. Будівлі	1280000	-	-
2. Передавальні пристрої	128000	-	-
Всього по групі	1408000	5	70400
Група 4			
3. Обчислювальна техніка	114885	-	-
Всього по групі	114885	50	57442,5
Група 5			
4. Вимірювальні пристрої	5190	-	-
5. Господарський інвентар	28000	-	-
Всього по групі	33190	25	8297,5
Нематеріальні активи			
6. Нематеріальні активи	40000	10	4000
Разом	$K_p = 1596075$		$A_p = 140140$

7.5 Визначення собівартості розробки та ціни програмної продукції

Визначимо основну зарплату виконавців

$$z_o = \frac{z_{cd} \cdot T_{nz}}{N_e}, \quad (7.11)$$

де N_e – Кількість екземплярів програм, шт.

$$Z_o = 400 \cdot 209 / 40 = 2090 \text{ грн}$$

Визначимо додаткову зарплату (оплата відпусток, виконання державних та суспільних обов'язків) на рівні 10%

$$Z_d = Z_o \cdot H_q \cdot 0,01, \quad (7.12)$$

де H_q – норматив додаткової зарплати, %

$$Z_d = 2090 \cdot 10 \cdot 0,01 = 209 \text{ грн}$$

Відрахування на соціальні потреби за нормативом $H_c=37\%$ від суми основної та додаткової зарплати

$$C_{oc} = 0,01 \cdot H_c (Z_o + Z_d), \quad (7.13)$$

де H_c – відрахування на соціальні потреби, %

$$C_{oc} = 0,01 \cdot 37 (2090 + 209) = 851 \text{ грн}$$

Визначимо загальногосподарські витрати (електроенергію, ремонт і утримання приміщень і т.д) за нормативом $H_g=15\%$ від основної зарплати

$$G_{ocn} = Z_o \cdot H_g \cdot 0,01, \quad (7.14)$$

де H_g – загальногосподарські витрати, %

$$G_{ocn} = 2090 \cdot 15 \cdot 0,01 = 314 \text{ грн}$$

Визначимо витрати на матеріали для розробки програмної продукції за нормами споживання та діючими цінами за одиницю виміру:

$$Z_M = (Z_{M1} + Z_{M2} + Z_{M3}) / N_e, \quad (7.15)$$

де Z_{M1} – вартість паперу, грн., Z_{M2} – вартість запам'ятовуючих пристроїв, грн., Z_{M3} – вартість фарби, картриджей, тонеру, грн., N_e – кількість екземплярів програм, шт.

Згідно виданих викладачем норм n_{mic} приймаємо 0,33 пачки паперу на місяць розробки. Тоді, враховуючи, що вартість пачки паперу складає $C_n=105$ грн., визначаємо вартість паперу за період розробки $N_m=3$ міс:

$$Z_{M1} = C_n \cdot N_m \cdot n_{mic}. \quad (7.16)$$

$$Z_{M1} = 105 \cdot 3 \cdot 0,33 = 105 \text{ грн.}$$

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		80

Згідно виданих викладачем норм до вартості запам'ятовуючих пристроїв входить вартість CD дисків в кількості, що дорівнює кількості екземплярів програм та одного DVD диска для збереження резервної копії програми:

$$Z_{M2} = \sum C_{d.}, \quad (7.17)$$

де C_d – вартість дисків CD/DVD: CDR TDK 700Mb, 80Min, 52x Cake box – 2 грн/шт., DVD-R LG 4,7Gb, 16x speed Cake box – 2 грн/шт.

$$Z_{M2} = 41 \cdot 12 = 492 \text{ грн.}$$

Згідно виданих викладачем норм одноразовій заправці підлягають усі друкуючі пристрої і становить:

$$Z_{M3} = \sum C_{z.}, \quad (7.18)$$

де: C_z – вартість розхідних матеріалів друкуючих пристроїв: відновлення та заправка картриджу для Canon i-SENSYS LBP6030W – 574 грн.; картридж для Epson Stylus Photo P50 – 558 грн.; відновлення картриджу для MF217W – 570 грн.

$$Z_{M3} = 574 + 558 + 570 = 1702 \text{ грн.}$$

$$Z_M = (105 + 492 + 1702) / 40 = 57 \text{ грн.}$$

Визначимо витрати на освоєння нових мов програмування або операційних систем за нормативом ($H_n = 15\%$) від основної зарплати виконавців

$$O_n = Z_o \cdot H_n \cdot 0,01, \quad (7.19)$$

де H_n – норматив витрат на освоєння нових мов програмування, %

$$O_n = 2090 \cdot 15 \cdot 0,01 = 314 \text{ грн}$$

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ($N_e = 40$ прим.)

$$A_m = \frac{A_p \cdot N_{mic}}{N_e \cdot 12}, \quad (7.20)$$

де A_p – загальна річна сума амортизаційних відрахувань, грн.

$$A_m = 140140 \cdot 3 / (40 \cdot 12) = 876 \text{ грн}$$

Повна собівартість ПЗ визначається як сума витрат за попередніми статтями калькуляції

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		81

$$C_n = Z_o + Z_d + C_{oc} + \Gamma_{ocn} + Z_m + O_n + A_m. \quad (7.21)$$

$$C_n = 2090 + 209 + 851 + 314 + 57 + 314 + 876 = 4711 \text{ грн.}$$

Визначимо плановий прибуток за рівнем рентабельності (Рп) програмної продукції, яка залежить від складності програми та ступеня новизни задачі.

Для даного програмного забезпечення рівень рентабельності складає 55%

$$P_p = 0,01 \cdot P_n \cdot C_n, \quad (7.22)$$

де P_c – рівень рентабельності, %

$$P_p = 0,01 \cdot 55 \cdot 4711 = 2591 \text{ грн.}$$

Величини ціна підприємства, податок на додану вартість, відпускна ціна програмної продукції визначаються за формулами, приведеними в таблиці 7.9

Таблиця 7.9 – Нормативна калькуляція собівартості розробки програмного забезпечення задачі

Найменування статей витрат	Позначення	Величина, грн.
1. Основна зарплата виконавців	Z_o	2090
2. Додаткова зарплата виконавців	Z_d	209
3. Відрахування на соціальні потреби	C_{oc}	851
4. Загальногосподарські витрати	Γ_{ocn}	314
5. Витрати на матеріали	Z_m	57
6. Освоєння нових операційних систем, мов програмування	O_n	314
7. Амортизація основних фондів	A_m	876
8. Повна собівартість програмного забезпечення	C_n	4711
9. Плановий прибуток	P_p	2591
10. Ціна підприємства $C_n = C_n + P_p$	C_n	7302
11. Податок на додану вартість $ПДВ = 0,01 \cdot H_{ов} \cdot C_n$	$ПДВ$	1460,4
12. Відпускна ціна програмної продукції $C = C_n + ПДВ$	C	9893

Вим.	Арк.	№ докум.	Підпис	Дата
------	------	----------	--------	------

ВКРМ-123.21.0010.00.00.ПЗ

Арк.

82

7.6 Визначення об'єму капітальних вкладень у споживача програмної продукції

Об'єм капітальних вкладень у споживача програмної продукції визначаємо на основі балансової вартості основних фондів, яка враховує ціну, транспортно-заготівельні витрати, вартість будівель, монтажних та пусконаладжувальних робіт, а також витрати на випробування у виробничих умовах. Результати розрахунків зводимо у таблицю 7.10.

Таблиця 7.10 – Розрахунок об'єму капітальних вкладень у споживача програмної продукції

Найменування капітальних вкладень	Сума за варіантами, грн	
	Базовий	Новий
Вартість програмної продукції	–	9893
Всього капітальних витрат	–	9893

7.7 Визначення експлуатаційних витрат

Експлуатаційні витрати у споживача програмної продукції визначаємо при умові роботи підсистеми на протязі року. Результати зводимо до таблиці 7.11.

Таблиця 7.11 – Розрахунок експлуатаційних витрат у споживача програмної продукції

Найменування статей витрат	Позначення	Сума витрат за варіантами, грн.	
		Базовий	Новий
1. Витрати на оплату праці	Z_p	19290	3617
2. Витрати на електроенергію	$Z_{ел}$	2099	1049
3. Витрати на амортизацію	$Z_{ам}$	0	2473
Всього витрат за рік	I	21389	7139

Вим.	Арк.	№ докум.	Підпис	Дата
------	------	----------	--------	------

ВКРМ-123.21.0010.00.00.ПЗ

Арк.

83

Витрати на оплату праці:

$$Z_p = T_p \cdot Z_z \cdot (1 + 0,01 \cdot H_q) \cdot (1 + 0,01 \cdot H_c), \quad (7.23)$$

де T_p – кількість годин обслуговування системи за рік, год.; Z_z – заробітна плата обслуговуючого персоналу, грн/год.

Після купівлі нового програмного забезпечення кількість профілактичних годин робіт зменшилася з 800 годин на рік до 150 годин на рік, тому витрати на технічне обслуговування зменшилися з

$$Z_{p \text{ баз}} = 800 \cdot 16 \cdot 1,1 \cdot 1,37 = 19290 \text{ грн.}$$

до

$$Z_{p \text{ нов}} = 150 \cdot 16 \cdot 1,1 \cdot 1,37 = 3617 \text{ грн.}$$

Витрати на електроенергію визначаються з урахуванням споживаємої потужності ($P_{ел}$) в кіловатах, часу експлуатації технічних засобів (T_p) в годинах та ціни однієї кіловат-години ($C_{ел}$).

$$Z_{ел} = P_{ел} \cdot T_p \cdot C_{ел}. \quad (7.24)$$

$$Z_{ел \text{ баз}} = 0,475 \cdot 2455 \cdot 1,8 = 2099 \text{ грн}$$

$$Z_{ел \text{ нов}} = 0,475 \cdot 1227 \cdot 1,8 = 1049 \text{ грн}$$

Витрати по амортизації визначаються на основі норм амортизаційних відрахувань, вартості програмної продукції і основних фондів. Для розрахунку складаємо таблицю 7.12.

Таблиця 7.12 – Розрахунок амортизаційних відрахувань

Групи основних фондів	Норма амортизації %	Балансова вартість, грн., за варіантами		Сума відрахувань, грн за варіантами	
		Базовий	Новий	Базовий	Новий
Програмна продукція	25	–	9893	–	2473,25
Всього відрахувань	-	–	9893	–	2473,25

$$T_{cn} = \frac{K_n - K_0}{I_0 - I_n} \quad (7.28)$$

$$T_{cn} = \frac{9893}{21389 - 7139} = 0,8 \text{ року}$$

Показники економічної ефективності програмної продукції зводимо до таблиці 7.13.

Таблиця 7.13 – Показники економічної ефективності програмної продукції

Найменування показників	Одиниця виміру	Величина
1. Кількість екземплярів програми	Прим.	40
2. Повна собівартість розробленої програми	Грн	4711
3. Ціна розробленої програми	Грн.	7302
4. Плановий прибуток від реалізації розробленої програми	Грн.	2591
5. Рентабельність програмної продукції	%	55
6. Об'єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	1596075
7. Загальний прибуток від реалізації програмної продукції	Грн.	103640
8. Величина економічного ефекту при виготовлені програмної продукції	Грн.	68605
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Років	0,5
10. Об'єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	9893
11. Величина економічного ефекту у користувача програмної продукції	Грн.	11777
12. Період окупності додаткових капітальних вкладень у користувача програмної продукції	Років	0,8

7.9 Висновки

Розроблена програма економічно вигідна. За рахунок впровадження програмного забезпечення досягається скорочення часу обробки інформації, підвищується культура праці, підвищення якості приймаючих управлінських рішень.

Кафедра КБПЗ – 2021 рік

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		87

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Охорона праці – це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини в процесі трудової діяльності.

Охорона праці є складовою частиною безпеки життєдіяльності [3].

Законом України “Про охорону праці” [2] регламентуються загальні положення державної політики в галузі охорони праці, а конкретизуються ці положення нормативно-правовими актами про охорону праці, зокрема Наказом Міністерства соціальної політики України 14.02.2018 № 207, який зареєстровано в Міністерстві юстиції України 25 квітня 2018 р. за №508/31960 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями», НПАОП 0.00-1.28-10 «Правила охорони праці під час експлуатації електронно-обчислювальних машин», та ДСанПіН 3.3.2-007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин».

Програмісти у процесі роботи мають негативний вплив на органи зору, а також мають значну розумову напругою і нервово-емоційне навантаження. Руки (суглоби пальців та м'язи рук) при роботі з клавіатурою мають теж істотне навантаження. До шкідливих факторів, які впливають на робітників галузі інформаційних технологій (ІТ) спеціалісти відносять високочастотні електромагнітні коливання (випромінювання) роботи апаратної частини ЕОМ та виділення шкідливих газів.

Ці шкідливі фактори можуть привести до професійних захворювань.

При розгляді шкідливих чинників роботи програмістів та інших

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		88

спеціалістів ІТ будемо керуватись наступними нормативно-правовими актами: «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98, та «Правила охорони праці під час експлуатації електронно-обчислювальних машин» НПАОП 0.00-1.28-10,

Умови праці програміста включають наступні фактори:

- параметри монітора (випромінювання);
- вентиляція приміщення;
- освітлення приміщення;
- параметри повітряного середовища в приміщенні, тощо.

Щоб запропонувати заходи щодо зменшення впливу комп'ютера на організм програміста визначимо фактори, які можуть викликати професійне захворювання і впливають на працездатність програміста.

8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером

При роботі з використанням ЕОМ відзначають наступні небезпечні та шкідливі фактори:

- ризик виникнення надзвичайних ситуацій природного або штучного характеру на об'єкті або території.
- ризик виникнення пожежі;
- негативний вплив на органи зору людини;
- ризики ураження електричним струмом;
- електромагнітні (у т.ч. високочастотні) випромінювання (коливання);
- несприятливі мікрокліматичні умови;
- нервово-емоційна напруженість праці;
- інтелектуальні навантаження;
- невідповідність ергономічних показників робочого місця діючим вимогам;

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		89

- шуми;
- статичні навантаження на кістково-м'язовий апарат;
- недостатня, або надмірна освітленість робочого місця;
- монотонність праці.

Результати досліджень показали, що найбільшою мірою негативний фізіологічний вплив на операторів ПК пов'язано з дискомфортними зоровими умовами через неправильно спроектованого освітлення: пряма і відбита від екранів бляклість, несприятливий розподіл яскравості в полі зору, невірна орієнтація робочого місця щодо світлоприймачів.

Розташовувати обладнане дисплеєм робоче місце необхідно таким чином, щоб в поле зору оператора не потрапляли вікна або освітлювальні прилади. Вони не повинні перебувати і безпосередньо за спиною оператора. Слід домагатися зменшення відображень на екрані від різних джерел штучного і денного світла. Коли штучне світло змішується з природним, рекомендується використовувати лампи, по спектрального складу найбільш близькі до сонячного світла. Співвідношення яскравості екрана і безпосередніх найближчого оточення не повинно перевищувати 3:1.

Рекомендована відносна вологість повітря у приміщенні 55%. Рекомендована швидкість руху повітря не більше 0.1 м / с. Оптимальні значення температури повітря в приміщенні повинні бути у межах 19-23 С.

8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста

Розглянемо умови праці у приміщенні, в якому працюють програмісти. Геометричні розміри приміщення наведено у таблиці 8.1.

У зазначеному приміщенні працює 1 особа. За даними, які наведено у табл. 8.1, та табл. 8.2, можна зробити висновок, що площа та об'єм приміщення у розрахунку на одно робоче місце програміста істотно перевищують нормативні

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		90

впливом багатьох факторів, серед яких: вологість атмосферного повітря, виділення вологи людьми (при диханні та випарами з поверхні шкіри).

Мікроклімат повітряного середовища в приміщенні характеризується запиленістю та загазованістю повітря. Мікроклімат приміщення визначається діючим на організм людини поєднанням, вологості, температури, швидкості руху повітря та інтенсивності теплового випромінювання. Аналіз мікроклімату складається з визначення зазначених вище факторів і порівняння результатів із встановленими нормами.

У таблиці 8.3 наведено оптимальні та фактичні значення параметрів мікроклімату як для категорії ваги робіт Іа, так і розглянутого приміщення. У приміщеннях, де встановлено ЕОМ, рекомендується застосування тільки оптимальних значень показників мікроклімату.

Проведений аналіз показує, що показники мікроклімату в приміщенні відповідають установленим нормам. Штучне опалення застосовується у холодний період року.

Таблиця 8.3 – Оптимальні і фактичні значення параметрів мікроклімату

Пора року	Оптимальні для Іа			Фактичні		
	Температура, °С	Вологість, %	Швидкість повітря, м/с	Температура, °С	Вологість, %	Швидкість повітря, м/с
Холодна	22-24	40-60	0,1	23-24	45-55	0,1
Тепла	23-25	50-70	0,1	23-24	55-65	0,11

В літню пору застосовується кондиціонер.

Для боротьби з пилом робляться регулярні провітрювання та вологі прибирання приміщенні.

У приміщенні знаходяться наступні джерела шуму: принтер HP 1100, електродвигуни вентиляторів ЕОМ.

Одним з найважливіших факторів, які впливають на ефективність трудової діяльності людини, та попереджають травматизм і професійні захворювання програмістів є освітлення на робочому місці.

Працю працівника, який постійно працює за комп'ютером, згідно ДБН В.2.5 – 28 – 2006 р можна віднести до роботи з малою точністю (найменший розмір об'єкта розрізнення від 1 до 5 мм) V-го розряду зорової роботи, з великою контрастністю об'єкта розрізнення (символів на екрані дисплея), з темним тлом (під розряд зорової роботи В). Приміщення можна віднести до 1-ої групи приміщень, у яких проводиться розрізнення об'єктів зорової роботи при фіксованому напрямку лінії зору того, що працює на робочу поверхню. Для такого типу приміщень і розряду зорової роботи нормоване значення коефіцієнта природної освітленості (КПО) робочої поверхні (при поєднаному, спільному освітленні), повинен становити не більше 1,5%, освітленість при штучному висвітленні повинна становити 300 лк. Крім того все поле зору повинне бути освітлено достатньо рівномірно – ця основна гігієнічна вимога. Так як яскраве світло на ділянці периферійного зору значно збільшує напруженість очей і, як наслідок, призводить до їх швидкої стомлюваності, ступінь освітлення приміщення і яскравість екрану комп'ютера повинні бути приблизно однаковими.

8.4 Розробка заходів з умов поліпшення охорони праці

Згідно аналізу умов праці в розглянутому приміщенні, ми одержали наступні результати:

– розмірі приміщення, у розрахунку на одному працюючого, відповідають нормативам;

– мікроклімат відповідає нормативному значенню;

– акустичні умови роботи не перевищують нормативних значень;

Таким чином можна припустити, що основною причиною можливого зниження працездатності програміста є психофізіологічний фактор, тому основна

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		93

поток, що падає на розрахункову поверхню, до сумарного потоку всіх ламп і обчислюється в долях одиниці; залежить від характеристик світильника, розмірів приміщення, забарвлення стін і стелі, що характеризуються коефіцієнтами відбиття від стін ($\rho_{стін}$) і стелі ($\rho_{стелі}$), значення коефіцієнтів дорівнюють $\rho_{стін} = 50\%$ і $\rho_{стелі} = 50\%$.

Обчислимо індекс приміщення за формулою:

$$i = S / (h(A+B)),$$

де:

S – площа приміщення, $S = 30 \text{ м}^2$;

h – розрахункова висота підвісу, $h = 2,5 \text{ м}$ (співпадає з висотою стелі, т.я. лампи освітлення закріплюються на стелі);

A – ширина приміщення, $A = 5 \text{ м}$;

B – довжина приміщення, $B = 6 \text{ м}$.

Підставимо всі значення у формулу та визначимо індекса приміщення:
 $i = 1,09$.

Знаючи індекс приміщення, за знаходимо $n = 0,46$ (з табличних даних коефіцієнтів використання світлового потоку (n) світильників [4]). Підставимо всі значення у формулу, визначимо світловий потік: $F = 32282,6 \text{ Лм}$.

Будемо використовувати світильник TL-F 12 36W LED 6000K IP65, світловий потік яких $F_n = 3000 \text{ Лм}$.

Число світильників визначається по формулі:

$$N = F / F_n$$

де:

F – світловий потік,

F_n – світловий потік одного світильника.

Підставимо всі значення у формулу та визначимо індекс приміщення:

$$N = 32282,6 / 3000 = 10,8 \text{ шт.}$$

Приймаємо необхідну кількість світильників 11 шт.

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		95

8.6 Висновки до розділу

Дотримання всіх необхідних умов праці не лише сприяє збереженню здоров'я працівників, а також підвищує ефективність виробництва в цілому.

З цих міркувань було здійснено аналіз приміщення, призначеного для праці програмістів, проведено розгляд небезпечних та шкідливих факторів, що негативно впливають на програмістів під час роботи. Виконано розрахунок штучного освітлення, як одного з ключових факторів впливу на працездатність та здоров'я програміста. Розроблено заходи з охорони праці.

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		96

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи кібербезпеки для блокування вхідних вузлів Tor і серверів VPN-провайдерів.

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

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти наведені теоретичне узагальнення й рішення наукового завдання дослідження методів кібербезпеки для блокування вхідних вузлів Tor і серверів VPN-провайдерів.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем кібербезпеки для блокування вхідних вузлів Tor і серверів VPN-провайдерів.
- Досліджена система кібербезпеки для блокування вхідних вузлів Tor і серверів VPN-провайдерів.
- На основі отриманих результатів досліджень створена програмна реалізація системи кібербезпеки для блокування вхідних вузлів Tor і серверів VPN-провайдерів.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання кібербезпеки для блокування вхідних вузлів Tor і серверів VPN - провайдерів.

Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		97

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, SQLite. Дана мова програмування дозволяє найбільш ефективно обробляти дані. 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 техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

Програма призначена для виконання під управлінням багатозадачної операційної системи Windows XP/Vista/7/8/10.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм Khufu.

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.

Розроблена програма має реальний економічний ефект від її впровадження у виробництво у сумі 11777 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 0,8 роки.

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		98

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Колодяжний І.О. Дослідження та програмна реалізація системи кібербезпеки для блокування вхідних вузлів Tor і серверів VPN-провайдерів // Збірник праць молодих науковців ЦНТУ. – Вип. 12. – Кропивницький: ЦНТУ, 2022.

2. Kolodiazhnyi I., Dorenskyi O. A Software Tool For Increasing Effectiveness Of Countering Of Anti-Ukrainian Propaganda On Internet // Інформаційні технології – 2020 : VII всеукр. наук.-практ. конф. молодих науковців, м. Київ / Київський університеті імені Бориса Грінченка. – Київ, 2020.

3. Степанов В. Ю. Інформаційна безпека як складова державної інформаційної політики / В. Ю. Степанов // Державне будівництво: наук. журн. – 2016. – № 2. – С. 1-9.

4. Г.Г. Швачич, В.В. Толстой, Л.М. Петречук, Ю.С. Іващенко, О.А. Гуляєва, Соболенко О.В. Сучасні інформаційно-комунікаційні технології: Навчальний посібник. – Дніпро: НМетАУ, 2017. –230 с.

5. Смирнов А.А. Анализ и сравнительное исследование перспективных направлений развития цифровых телекоммуникационных систем и сетей / А.А.Смирнов, В.В.Босько, Е.В.Мелешко // Системи обробки інформації. – Х.: ХУ ПС, 2008. – Вип.7(74). – С.120-123.

6. Смирнов А.А. Усовершенствование метода управления очередями в многопротокольных узлах телекоммуникационной сети / А.А.Смирнов, Е.В.Мелешко // Збірник тез та доповідей другої всеукраїнської науково-практичної конференції «Системний аналіз. Інформатика. Управління». Запоріжжя. Тези доповідей. Запоріжжя: КПУ, 2011.

7. Смирнов С.А. Метод безопасной маршрутизации метаданных в облачные антивирусные системы / А.К. Дидык, С.А. Смирнов // Информационные технологии в управлении, образовании, науке и

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		99

промышленности: монография / Под редакцией профессора В.С. Пономаренко. – Х.: Видавець Рожко С.Г., 2016. – 566 с.

8. Смирнов С. А. Сравнительные исследования математических моделей технологии распространения компьютерных вирусов в информационно-телекоммуникационных сетях / Мохамад Абу Таам Гани, А. А. Смирнов, А. В. Коваленко, С. А. Смирнов // Системи обробки інформації: зб. наук. праць. – Х.: ХУПС, 2014. – Вип. 9(125). – 105-110.

9. Смирнов С. А. Математическая модель интеллектуального узла коммутации с обслуживанием информационных пакетов различного приоритета / Мохамад Абу Таам Гани, А. А. Смирнов, Н. С. Якименко, С. А. Смирнов // Збірник наукових праць Харківського університету Повітряних Сил. – Харків: ХУПС, 2014. – Вип. 4 (41). – С. 48-52.

10. Смирнов С. А. Исследование показателей качества функционирования интеллектуальных узлов коммутации в телекоммуникационных системах и сетях / Мохамад Абу Таам Гани, А. А. Смирнов, Н. С. Якименко, С. А. Смирнов // Наука і техніка Повітряних Сил Збройних Сил України: наук. журн. – Х.: ХУПС, 2014. – № 4(17). – С. 90-95.

11. Смирнов С. А. Усовершенствованный алгоритм управления доступом к «облачным» телекоммуникационным ресурсам / Мохамад Абу Таам Гани, А. А. Смирнов, Н. С. Якименко, С. А. Смирнов // Системи обробки інформації: зб. наук. праць. – Х.: ХУПС, 2015. – Вип. 1(126). – С. 150-153.

12. Smirnov S.A. Method of controlling access to intellectual switching nodes of telecommunication networks and systems / A.A. Smirnov, Mohamad Abou Taam, S.A. Smirnov // International Journal of Computational Engineering Research (IJCER). – Volume 5, Issue 5. – India. Delhi. – 2015. – P. 1-7.

13. Смирнов С. А. Анализ и исследование методов управления сетевыми ресурсами для обеспечения антивирусной защиты данных / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Системи озброєння і військова техніка: наук. журн. – Х.: ХУПС, 2015. – № 3(43). – С. 100-107.

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		100

14. Смирнов С. А. Исследование эффективности метода управления доступом к облачным антивирусным телекоммуникационным ресурсам / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Наука і техніка Повітряних Сил Збройних Сил України: наук. журн. –Х.: ХУПС, 2015. – № 3(20). – С. 134-141.

15. Смирнов С. А. Комплекс gert-моделей технологии облачной антивирусной защиты телекоммуникационной системы / А. А. Смирнов, А. К. Дидык, А. Н. Дреев, С. А. Смирнов // Безпека інформації: наук. – практ. журн. – К.: НАУ, 2015. – Т. 21, № 3. – С. 251-262.

16. Смирнов С. А. Метод безопасной маршрутизации метаданных в облачные антивирусные системы / А. А. Смирнов, А. К. Дидык, С. А. Смирнов // Системи озброєння і військова техніка: наук. журн. –Х.: ХУПС, 2016. – № 2 (46). – С. 146-149.

17. Смирнов С. А. Модели системы нейросетевых экспертов безопасной маршрутизации в облачных антивирусных системах / А. А. Смирнов, А. К. Дидык, А. Н. Дреев, С. А. Смирнов // Системи обробки інформації: зб. наук. праць. – Х.: ХУПС, 2016. – Вип. 3 (140). – С. 36-39.

18. Смирнов С. А. Метод безопасной маршрутизации на базовом множестве путей передачи метаданных в облачные антивирусные системы / В. Л. Бурячок, С. А. Смирнов // Системи управління, навігації та зв'язку. – Полтава, 2016. – Вип. 4(40). – С. 57-62.

19. Смирнов С. А. Способ контроля линий связи телекоммуникационной системы облачного антивируса / А. А. Смирнов, А. К. Дидык, А. Н. Дреев, С. А. Смирнов // Збірник наукових праць Харківського університету Повітряних Сил. – Харків: ХУПС, 2016. – № 2 (47). – С. 148-152.

20. Смирнов С. А. Дослідження та реалізація GERT-моделі технології розповсюдження комп'ютерних вірусів для захисту телекомунікаційних систем / В. Л. Бурячок, Мохамад Абу Таам Гани, С. А. Смирнов // Інформаційні технології

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		101

та комп'ютерна інженерія: зб. тез доп. наук.-практ. конф., м. Кіровоград, 4 грудня 2014 р. – Кіровоград: КНТУ, 2014. – С. 168.

21. Смирнов С. А. Исследование математических моделей технологии распространения компьютерных вирусов / А. А. Смирнов, Мохамад Абу Таам Гани, С. А. Смирнов // Актуальні питання забезпечення кібернетичної безпеки та захисту інформації: зб. наук. праць міжнар. наук.-практ. конф., м. Київ, 25-28 лютого 2015 р. – К.: Європейський університет, 2015. – С. 90-91.

22. Смирнов С. А. Метод управления доступом к «облачным» ресурсам для защиты телекоммуникационных систем / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Всеукраїнська науково-практична конференція «Інформаційна безпека держави, суспільства та особистості», м. Кіровоград, 16 квітня 2015 р.: зб. тез доп. – Кіровоград: КНТУ, 2015. – С. 50-52.

23. Смирнов С. А. Разработка метода управления доступом в интеллектуальных узлах коммутации / А. А. Смирнов, Мохамад Абу Таам Гани, С. А. Смирнов // Проблеми і перспективи розвитку ІТ-індустрії: зб. тез VII міжнар. наук.-практ. конф., м. Харків, 17-18 квітня 2015 р. – Х.: ХНЕУ, 2015. – С. 14.

24. Смирнов С.А. Реализация метода управления доступом в интеллектуальных узлах коммутации / А.А. Смирнов, Мохамад Абу Таам Гани, С.А. Смирнов // Збірник тез XVII міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування». м. Кіровоград. 17-18 квітня 2015 р. – Кіровоград: КНТУ. – 2015. – С. 91-92.

25. Смирнов С. А. технология передачи сигнатур в облачные антивирусные системы для обеспечения защищенности телекоммуникационных сетей / А. А. Смирнов, С. А. Смирнов // Збірник тез V міжнародної науково-технічної конференції «ITSEC», Київ, 19-22 травня 2015 р. – К.: НАУ 2015. – С. 12-13.

26. Смирнов С. А. Реализация математической модели интеллектуального узла коммутации для обеспечения защищенности

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		102

телекоммуникационной сети / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Інформаційна та економічна безпека (INFECO-2015): зб. тез II Міжнар. наук.-практ. Інтернет-конф., м. Харків, 21-22 травня 2015 р. – Х.: ХІБС УБС НБУ, 2015. – С. 20-24.

27. Смирнов С. А. Разработка математической модели технологии распространения компьютерных вирусов в информационно-телекоммуникационных сетях / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Сборник тезисов XI международной конференции «Стратегия качества в промышленности и образовании», г. Варна, Болгария, 01-06 июня 2015 г. – Варна: ТУВ, 2015. – С. 488-491.

28. Смирнов С. А. Метод управления доступом к облачным телекоммуникационным ресурсам для обеспечения защиты данных / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Комп'ютерні технології та інформаційна безпека: зб. тез доп. міжнар. наук.-практ. конф., м. Кіровоград, 2-3 липня 2015 р. – Кіровоград: КНТУ, 2015. – С. 4-5.

29. Смирнов С. А. Имитационная модель системы управления доступом к облачным антивирусным телекоммуникационным ресурсам / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Збірник тез першої всеукраїнської науково-практичної конференції «Перспективні напрями захисту інформації» (м. Затока, 7-9 вересня 2015 р.). – Одеса: ОНАЗ, 2015. – С. 90-94.

30. Смирнов С. А. Разработка комплекса gert-моделей технологии облачной антивирусной защиты телекоммуникационной системы / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Інформаційні технології та взаємодії» (IT & I): зб. тез II міжнар. наук. -практ. конф., м. Київ, 3-5 листопада 2015 р. – К.: КНУ ім. Тараса Шевченка, 2015. – С. 65-67.

31. Смирнов С. А. Разработка моделей телекоммуникационной системы формирования и обработки метаданных в облачных антивирусных системах / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Информационные и телекоммуникационные технологии: образование, наука, практика: сб. тезисов II

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		103

международ. научно-практ. конф., г. Алматы, Казахстан, 3-4 декабря 2015 г. – Алматы: КазНИТУ им. К.И. Сатпаева, 2015. – С. 309-313.

32. Смирнов С. А. gert-модели технологии облачной антивирусной защиты / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Безопаска українського суспільства в концепції вступу в постіндустріальне суспільство ЄС: зб. тез Круглого столу, м. Київ, 16 грудня 2015 р. – К.: Європейський університет, 2015. – С.41-43.

33. Смирнов С. А. Алгоритмы формирования множества маршрутов передачи метаданных в облачные антивирусные системы / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Актуальні питання забезпечення кібернетичної безпеки та захисту інформації: зб. наук. праць II Міжнар. наук.-практ. конф., м. Київ, 24-27 лютого 2016 р. – К.: Європейський університет, 2016. – С. 140-142.

34. Смирнов С. А. Разработка и реализация метода безопасной маршрутизации метаданных в облачные антивирусные системы / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Securitea informationala 2015-2016: Conferenta internationala (editia a XII-a), Chisinau, Moldova, 3 martie 2016. – Chisinau: ADSEM, 2016. – С. 90-96.

35. Смирнов С. А. Алгоритм формирования базового множества маршрутов передачи метаданных в облачные антивирусные системы / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Інформатика та системні науки (ІСН-2016): зб. тез VII всеукр. наук.-практ. конф., м. Полтава, 10-12 березня 2016 р. – Полтава: ПУЕТ, 2016. – С. 261-263.

36. Смирнов С. А. Система обработки и формирования начального состояния маршрутизации метаданных в облачные антивирусные системы / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Проблеми кібербезпеки інформаційно-телекомунікаційних систем: зб. тез наук.-практ. конф., м. Київ, 10-11 березня 2016 р. – К.: КНУ ім. Тараса Шевченка, 2016. – С. 81-82.

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		104

37. Смирнов С. А. Алгоритм безопасной маршрутизации на базовом множестве путей передачи метаданных в программный сервер облачной антивирусной системы / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Інформаційна безпека та комп'ютерні технології (IS&CT): зб. тез міжнар. наук.-практ. конф., м. Кіровоград, 24-25 березня 2016 р. – Кіровоград: КНТУ, 2016. – С. 73.

38. Смирнов С. А. Исследование способа контроля линий связи телекоммуникационной системы для облачных антивирусов / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Збірник тез першої міжнародної науково-практичної конференції «Проблеми науково-технічного та правового забезпечення кібербезпеки у сучасному світі» (ПНПЗК-2016), м. Харків, 30 березня – 1 квітня 2016 р. – Х.: НТУ «ХП», 2016. – С. 14.

39. Смирнов С. А. Разработка способа контроля линий связи телекоммуникационной системы для облачных антивирусов / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Матеріали XVIII міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування» (м. Кіровоград, 15-16 квітня 2016 р.). – Кіровоград: КНТУ, 2016. – С. 182-186.

40. Смирнов С. А. Разработка и исследование способа контроля линий связи телекоммуникационных сетей для облачных антивирусных систем / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Проблеми і перспективи розвитку ІТ-індустрії: VIII міжнар. наук.-практ. конф., м. Харків, 28-29 квітня 2016 р.: зб. тез. – Х.: ХНЕУ, 2016. – С. 48.

41. Смирнов С. А. Модель системы нейросетевых экспертов безопасной маршрутизации для облачных антивирусных систем / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Інформаційна та економічна безпека (INFECO-2016): зб. тез III міжнар. наук.-практ. конф., м. Харків, 28-30 кві. 2016 р. – Х.: ХННІ ДВНЗ «УБС», 2016. – С. 178-182.

42. Смирнов С. А. Метод безопасной маршрутизации метаданных в облачные антивирусные системы / А. А. Смирнов, С. А. Смирнов, А. К. Дидык //

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		105

Сборник тезисов XII международной конференции «Стратегия качества в промышленности и образовании» (г. Варна, Болгария, 30 мая – 02 июня 2016 г.). – Варна: ТУВ, 2016. – С. 581-585.

43. Смирнов С. А. Оценка эффективности метода безопасной маршрутизации метаданных в облачные антивирусные системы / А. А. Смирнов, С. А. Смирнов, А. С. Коваленко // РадіоЕлектроніка та ІнфоКомунікації: зб. тез першої наук. – техн. конф., м. Київ, 11-16 вересня 2016 р. – К.: НТУУ «КПІ», 2016. – С. 17.

44. Современные телекоммуникации. Технологии и экономика / [В.Л. Банкет, О.В. Бондаренко, П.П. Воробьенко и др.]; под ред. С.А. Довгого. – М.: Эко-Трендз, 2003. – 320 с.

45. Столлингс В. Современные компьютерные сети / Вильям Столлингс. –СПб.: Питер, 2003. – 778 с.

46. Таненбаум Э. Компьютерные сети / Эндрю Таненбаум; пер. с англ. А. Леонтьев. – СПб.: Питер, 2002. – 848 с.

47. Зеркалов Д. В. Охорона праці в Галузі: Загальні вимоги: навч. посіб. Київ: Основа. 2011. 551 с.

48. Про охорону праці: Закон України від 14.10.1992 р. № 2694-ХІІ. Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/2694-12#Text>

49. Вакцинирование, как составляющая безопасности жизнедеятельности во время пандемии Sars-Cov-2 (УДК 331.45, 614.8) / О.В. Орышака, К.Н. Марченко, А.К. Марченко // Журнал “Соціальна фармація в охороні здоров’я” / Національний фармацевтичний університет (НфаУ) – Харків, 2021. – Том 7 № 2 (2021), – С. 27-33. Режим доступу до ресурсу: <https://doi.org/10.24959/sphhcj.21.226>

50. Центр післядипломної освіти та підвищення кваліфікації. Режим доступу до ресурсу: <https://cpo.stu.cn.ua/>

					ВКРМ-123.21.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		106

Додаток А
(обов'язковий)

Технічне завдання

Зміст

1	Найменування та область застосування.....	2
2	Підстава для розробки.....	2
3	Мета та призначення розробки.....	2
4	Джерела розробки.....	2
5	Технічні вимоги.....	2
5.1	Вміст проекту.....	2
5.2	Показники призначення.....	3
5.3	Вимоги до функціональних характеристик.....	3
5.4	Вимоги до архітектури.....	3
5.5	Вимоги до надійності.....	3
5.6	Умови експлуатації.....	4
5.7	Вимоги до складу та параметрів технічних засобів.....	4
5.8	Вимоги до інформаційної і програмної сумісності.....	4
5.8.1	Обладнання.....	4
5.8.2	Мова програмування.....	4
5.8.3	Вхідні дані.....	5
5.8.4	Вихідні дані.....	5
6	Вимоги до програмної документації.....	5
7	Економічні вимоги.....	5
8	Вимоги щодо охорони праці.....	5
9	Перелік документів, що розробляються.....	6
10	Етапи розробки.....	6
11	Порядок контролю та приймання.....	6

					ВКРМ-123.21.0010.00.00.ТЗ			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Колодяжний І.О.				<i>Дослідження та програмна реалізація системи кібербезпеки для блокування вхідних вузлів Tor і серверів VPN-провайдерів</i>	Літ.	Аркуш	Аркушів
Перевірів	Доренський О.П.					М	1	6
Н. Контр.	Гермак В.С.				ЦНТУ КІ-20М-1,4			
Затв.	Смірнов О.А.							

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи кібербезпеки для блокування вхідних вузлів Tor і серверів VPN-провайдерів.

2 Підстава для розробки

Підставою для розробки служить завдання на випускну кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 42-13 від 02.08.2021 року).

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи кібербезпеки для блокування вхідних вузлів Tor і серверів VPN-провайдерів.

4 Джерела розробки

Джерелом цієї випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є стосовна до теми література і існуючі аналоги.

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

- вибір і обґрунтування методів реалізації проекту;
- розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;

					ВКРМ-123.21.0010.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

- техніко-економічне обґрунтування доцільності прийнятого до розробки програмного забезпечення;
- аналіз умов праці;
- розробка програми, що реалізує спроектовані алгоритми роботи системи.

5.2 Показники призначення

Система повинна забезпечувати:

- програмну реалізацію системи кібербезпеки для блокування вхідних вузлів Tor і серверів VPN-провайдерів;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

Розроблене програмне забезпечення не повинно мати обмежень на версію драйверів та операційної системи.

5.4 Вимоги до архітектури

Компонент, що розробляється повинен використовувати системні засоби та апаратні засоби, що на даному етапі розвитку обчислювальної техніки найбільше поширені.

5.5 Вимоги до надійності

Програмні модулі написані по всім правилам, які стосуються стандартних викликів процедур, функцій, методів і форм, визначених технічною документацією на середовище розробки.

					ВКРМ-123.21.0010.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

- температура повітря: 19-20 град. по Цельсію;
- відносна вологість повітря до 80%;
- атмосферний тиск 107 кПа.

5.7 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення повинно бути реалізоване на ПЕОМ архітектури IBM PC, працювати в ОС Windows XP/Vista/7/8/10 і з сумісними з цією платформою пристроями і прикладним програмним забезпеченням.

5.8 Вимоги до інформаційної і програмної сумісності

Переносність програмного забезпечення повинна бути забезпечена за рахунок його реалізації стандартного інтерфейсу взаємодії з ОС, що працюють під управлінням ОС Windows XP/Vista/7/8/10.

5.8.1 Обладнання

Комп'ютер Intel® Celeron/8 Mb/1.2 Gb/SVGA 14" 1Mb або сумісні з ним.

5.8.2 Мова програмування

Середовище Python, SQLite.

					ВКРМ-123.21.0010.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

Програмна продукція повинна бути представлена у виді опису структури даних, схем та опису алгоритму, а також текстів вихідних модулів програмного забезпечення згідно ЄСПД .

7 Економічні вимоги

7.1 Для ПЗ необхідно виробити функціонально-вартісний аналіз варіантів розробки.

7.2 Виконати розрахунок витрат показників економічного ефекту з урахуванням цін на 3 вересня 2021 року.

8 Вимоги щодо охорони праці

В частині охорони праці випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти повинні бути розглянуті шкідливі і небезпечні фактори при роботі з комп'ютером.

					ВКРМ-123.21.0010.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

9 Перелік документів, що розробляються

- Наукова новизна – 1 аркуш.
- Структурна схема системи – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма процесів – 1 аркуш.
- Блок-схема алгоритму роботи програми – 2 аркуша.
- Показники економічної ефективності – 1 аркуш.
- Пояснювальна записка – 106 аркушів.

10 Етапи розробки

10.1 Збір і обробка інформації по темі випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти. Постановка задачі на виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти (складання ТЗ).

10.2 Проведення досліджень або експериментальних робіт для уточнення основних положень випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.

10.3 Розробка функціональних схем, блок схем алгоритмів роботи програмного забезпечення.

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Розрахунок з техніко-економічного обґрунтування.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю та приймання

11.1 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на попередній захист 10.12.2021 р.

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 22.12.2021 р.

					ВКРМ-123.21.0010.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

Керівник випускної кваліфікаційної роботи за
другим (магістерським) рівнем вищої освіти

_____ Доренський О.П.

*Дослідження та програмна реалізація
системи кібербезпеки для блокування вхідних вузлів Tor і серверів VPN-
провайдерів*

Лістинг програми

Код документу 12

Носій: CD/DVD-диск

Загальна кількість аркушів: 20

Літера: РП

Кропивницький – 2021 року

Файл DNS_server.py основної програми

```

import socket, glob, json
import sqlite3 as lite
import sys
from Crypto.Cipher import DES

port = 53
ip = '127.0.0.1'

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.bind((ip, port))

"""
Bisection algorithms.

This module provides support for maintaining a list in sorted order without
having to sort the list after each insertion. For long lists of items with
expensive comparison operations, this can be an improvement over the more
common approach.
"""

def des_decryption(text):
    """
    bisect_left(a, x[, lo[, hi]]) -> index

    Return the index where to insert item x in list a, assuming a is sorted.

    The return value i is such that all e in a[:i] have e < x, and all e in
    a[i:] have e >= x. So if x already appears in the list, i points just
    before the leftmost x already there.

    Optional args lo (default 0) and hi (default len(a)) bound the
    slice of a to be searched.
    """

    key = b'abcdefgh'
    des = DES.new(key, DES.MODE_ECB)

    res = str(des.decrypt(text))[2:-1].replace(' ', '')
    return res

def load_zones():
    """
    bisect_right(a, x[, lo[, hi]]) -> index

    Return the index where to insert item x in list a, assuming a is
    sorted.

    The return value i is such that all e in a[:i] have e <= x, and all e
    in
    a[i:] have e > x. So if x already appears in the list, i points just
    beyond the rightmost x already there

    Optional args lo (default 0) and hi (default len(a)) bound the
    slice of a to be searched.
    """

    con = lite.connect('data.db')
    cur = con.cursor()
    cur.execute("SELECT * FROM students")
    rows = cur.fetchall()

    l = []

```

```

for row in rows:
    l.append([des_decryption(row[1]) + '.', des_decryption(row[2])])

res = {}
d = {}
for cur_list in range(len(l)):
    d.update({'$origin': l[cur_list][0], 'a': []})
    for cur_elem in range(1, len(l[cur_list])):
        temp = {}
        temp = {'name': '@', 'ttl': 400, 'value': l[cur_list][cur_elem]}
        d['a'].append(temp)
    zonenumber = d["$origin"]
    res.update({'zonenumber': d.copy()})

return res

zonedata = load_zones()

def getflags(flags):
    """
    insert_left(a, x[, lo[, hi]])

    Insert item x in list a, and keep it sorted assuming a is sorted.

    If x is already in a, insert it to the left of the leftmost x.

    Optional args lo (default 0) and hi (default len(a)) bound the
    slice of a to be searched.
    """
    byte1 = bytes(flags[:1])
    byte2 = bytes(flags[1:2])

    rflags = ''

    QR = '1'

    OPCODE = ''
    for bit in range(1, 5):
        OPCODE += str(ord(byte1) & (1 << bit))

    AA = '1'

    TC = '0'

    RD = '0'

    # Byte 2
    RA = '0'

    Z = '000'

    RCODE = '0000'

    return int(QR + OPCODE + AA + TC + RD, 2).to_bytes(1, byteorder='big') +
        int(RA + Z + RCODE, 2).to_bytes(1,
        byteorder='big')

def getquestiondomain(data):
    """
    insert_right(a, x[, lo[, hi]])

    Insert item x in list a, and keep it sorted assuming a is sorted.

    If x is already in a, insert it to the right of the rightmost x.

```

```

Optional args lo (default 0) and hi (default len(a)) bound the
slice of a to be searched.
"""
state = 0
expectedlength = 0
domainstring = ''
domainparts = []
x = 0
y = 0
for byte in data:
    if state == 1:
        if byte != 0:
            domainstring += chr(byte)
            x += 1
            if x == expectedlength:
                domainparts.append(domainstring)
                domainstring = ''
                state = 0
                x = 0
            if byte == 0:
                domainparts.append(domainstring)
                break
        else:
            state = 1
            expectedlength = byte
            y += 1

questiontype = data[y:y + 2]

return (domainparts, questiontype)

def getzone(domain):
    """
    Meta path import for built-in modules.

    All methods are either class or static methods to avoid the need to
    instantiate the class.
    """
    global zonedata
    zone_name = '.'.join(domain)
    try:
        return zonedata[zone_name]
    except KeyError:
        pass

def getrecs(data):
    """
    Find the built-in module.

    If 'path' is ever specified then the search is considered a
    failure.

    This method is deprecated. Use find_spec() instead.
    """
    domain, questiontype = getquestiondomain(data)
    qt = ''
    if questiontype == b'\x00\x01':
        qt = 'a'
    zone = getzone(domain)
    try:
        return (zone[qt], qt, domain)
    except TypeError:
        pass

def buildquestion(domainname, rectype):

```

```

qbytes = b''

for part in domainname:
    length = len(part)
    qbytes += bytes([length])

    for char in part:
        qbytes += ord(char).to_bytes(1, byteorder='big')

if rectype == 'a':
    qbytes += (1).to_bytes(2, byteorder='big')

qbytes += (1).to_bytes(2, byteorder='big')

return qbytes

```

```

def rectobytes(domainname, rectype, recttl, recval):
    """

```

an adaptation of the MySQL C API (mostly)

You probably are better off using MySQLdb instead of using this module directly.

In general, renaming goes from `mysql_*` to `_mysql.*`. `_mysql.connect()` returns a connection object (MYSQL). Functions which expect `MYSQL *` as an argument are now methods of the connection object. A number of things return result objects (MYSQL_RES). Functions which expect `MYSQL_RES *` as an argument are now methods of the result object. Deprecated functions (as of 3.23) are NOT implemented.

```

    """
    rbytes = b'\xc0\x0c'

    if rectype == 'a':
        rbytes = rbytes + bytes([0]) + bytes([1])

    rbytes = rbytes + bytes([0]) + bytes([1])

    rbytes += int(recttl).to_bytes(4, byteorder='big')

    if rectype == 'a':
        rbytes = rbytes + bytes([0]) + bytes([4])

        for part in recval.split('.'):
            rbytes += bytes([int(part)])

    return rbytes

```

```

def buildresponse(data):
    """

```

Returns a MYSQL connection object. Exclusive use of keyword parameters strongly recommended. Consult the MySQL C API documentation for more details.

host
string, host to connect

user
string, user to connect as

passwd
string, password to use

db
string, database to use

port
integer, TCP/IP port to connect to

```

unix_socket
    string, location of unix_socket (UNIX-ish only)

conv
    mapping, maps MySQL FIELD_TYPE.* to Python functions which
    convert a string to the appropriate Python type

connect_timeout
    number of seconds to wait before the connection
    attempt fails.

compress
    if set, gzip compression is enabled

named_pipe
    if set, connect to server via named pipe (Windows only)

init_command
    command which is run once the connection is created

read_default_file
    see the MySQL documentation for mysql_options()

read_default_group
    see the MySQL documentation for mysql_options()

client_flag
    client flags from MySQLdb.constants.CLIENT

load_infile
    int, non-zero enables LOAD LOCAL INFILE, zero disables
"""
# Transaction ID
TransactionID = data[:2]

# Get the flags
Flags = getflags(data[2:4])

# Question Count
QDCOUNT = b'\x00\x01'

# Answer Count
try:
    ANCOUNT = len(getrecs(data[12:])[0]).to_bytes(2, byteorder='big')
except TypeError:
    pass
# Nameserver Count
NSCOUNT = (0).to_bytes(2, byteorder='big')

# Additional Count
ARCOUNT = (0).to_bytes(2, byteorder='big')
try:
    dnsheader = TransactionID + Flags + QDCOUNT + ANCOUNT + NSCOUNT +
ARCOUNT
except UnboundLocalError:
    pass
# Create DNS body
dnsbody = b''

# Get answer for query
try:
    records, rectype, domainname = getrecs(data[12:])
except TypeError:
    pass
try:
    dnsquestion = buildquestion(domainname, rectype)
except UnboundLocalError:
    pass
try:

```

```

        for record in records:
            dnsbody += rectobytes(domainname, rectype, record["ttl"],
record["value"])
            return dnsheader + dnsquestion + dnsbody
        except UnboundLocalError:
            pass

```

```

while 1:

```

```

"""

```

```

Returns a MYSQL connection object. Exclusive use of
keyword parameters strongly recommended. Consult the
MySQL C API documentation for more details.

```

```

host

```

```

    string, host to connect

```

```

user

```

```

    string, user to connect as

```

```

passwd

```

```

    string, password to use

```

```

db

```

```

    string, database to use

```

```

port

```

```

    integer, TCP/IP port to connect to

```

```

unix_socket

```

```

    string, location of unix_socket (UNIX-ish only)

```

```

conv

```

```

    mapping, maps MySQL FIELD_TYPE.* to Python functions which
    convert a string to the appropriate Python type

```

```

connect_timeout

```

```

    number of seconds to wait before the connection
    attempt fails.

```

```

compress

```

```

    if set, gzip compression is enabled

```

```

named_pipe

```

```

    if set, connect to server via named pipe (Windows only)

```

```

init_command

```

```

    command which is run once the connection is created

```

```

read_default_file

```

```

    see the MySQL documentation for mysql_options()

```

```

read_default_group

```

```

    see the MySQL documentation for mysql_options()

```

```

client_flag

```

```

    client flags from MySQLdb.constants.CLIENT

```

```

load_infile

```

```

    int, non-zero enables LOAD LOCAL INFILE, zero disables

```

```

"""

```

```

data, addr = sock.recvfrom(512)

```

```

r = buildresponse(data)

```

```

try:

```

```

    sock.sendto(r, addr)

```

```

except TypeError:pass

```

Файл BlackList.py - інтерфейс бази даних

```

from PyQt5.QtCore import *
from PyQt5.QtWidgets import *
from PyQt5.QtGui import *
from PyQt5.QtWebEngineWidgets import QWebEngineView
from PyQt5.QtPrintSupport import *
import sys,sqlite3,time
from Crypto.Cipher import DES

import os

key = b'abcdefgh'
des = DES.new(key, DES.MODE_ECB)

def des_encryption(text):
    """
    Changes the user and causes the database specified by db to
    become the default (current) database on the connection
    specified by mysql. In subsequent queries, this database is
    the default for table references that do not include an
    explicit database specifier.

    This function was introduced in MySQL Version 3.23.3.

    Fails unless the connected user can be authenticated or if he
    doesn't have permission to use the database. In this case the
    user and database are not changed.

    The db parameter may be set to None if you don't want to have
    a default database.
    """
    text = text.encode()
    def pad(text):
        """
        Returns a dict with information about the current character set:

        collation
            collation name
        name
            character set name
        comment
            comment or descriptive name
        dir
            character set directory
        mbminlen
            min. length for multibyte string
        mbmaxlen
            max. length for multibyte string

        Not all keys may be present, particularly dir.

        Non-standard.
        """
        while len(text) % 8 != 0:
            text += b' '
        return text
    padded_text = pad(text)
    encrypted_text = des.encrypt(padded_text)

    return encrypted_text

class InsertDialog(QDialog):
    def __init__(self, *args, **kwargs):
        """
        Returns the ID generated for an AUTO_INCREMENT column by the previous
        query. Use this function after you have performed an INSERT query into a

```

table that contains an `AUTO_INCREMENT` field.

Note that this returns 0 if the previous query does not generate an `AUTO_INCREMENT` value. If you need to save the value for later, be sure to call this immediately after the query that generates the value.

The ID is updated after `INSERT` and `UPDATE` statements that generate an `AUTO_INCREMENT` value or that set a column value to `LAST_INSERT_ID(expr)`. See section 6.3.5.2 Miscellaneous Functions in the MySQL documentation.

Also note that the value of the SQL `LAST_INSERT_ID()` function always contains the most recently generated `AUTO_INCREMENT` value, and is not reset between queries because the value of that function is maintained in the server.

```

"""
super(InsertDialog, self).__init__(*args, **kwargs)

self.QBtn = QPushButton()
self.QBtn.setText("Register")

self.setWindowTitle("Add domain")
self.setFixedWidth(300)
self.setFixedHeight(250)

self.QBtn.clicked.connect(self.addstudent)

layout = QVBoxLayout()

self.nameinput = QLineEdit()
self.nameinput.setPlaceholderText("Name")
layout.addWidget(self.nameinput)

# self.branchinput = QComboBox()
# self.branchinput.addItem("Mechanical")
# self.branchinput.addItem("Civil")
# self.branchinput.addItem("Electrical")
# self.branchinput.addItem("Electronics and Communication")
# self.branchinput.addItem("Computer Science")
# self.branchinput.addItem("Information Technology")
# layout.addWidget(self.branchinput)

# self.seminput = QComboBox()
# self.seminput.addItem("1")
# self.seminput.addItem("2")
# self.seminput.addItem("3")
# self.seminput.addItem("4")
# self.seminput.addItem("5")
# self.seminput.addItem("6")
# self.seminput.addItem("7")
# self.seminput.addItem("8")
# layout.addWidget(self.seminput)

# self.mobileinput = QLineEdit()
# self.mobileinput.setPlaceholderText("Mobile")
# self.mobileinput.setInputMask('99999 99999')
# layout.addWidget(self.mobileinput)

self.addressinput = QLineEdit()
self.addressinput.setPlaceholderText("IP")
layout.addWidget(self.addressinput)

layout.addWidget(self.QBtn)
self.setLayout(layout)

```

```
def addstudent(self):
```

```

"""
Checks whether or not the connection to the server is
working. If it has gone down, an automatic reconnection is
attempted.

This function can be used by clients that remain idle for a
long while, to check whether or not the server has closed the
connection and reconnect if necessary.

New in 1.2.2: Accepts an optional reconnect parameter. If True,
then the client will attempt reconnection. Note that this setting
is persistent. By default, this is on in MySQL<5.0.3, and off
thereafter.

Non-standard. You should assume that ping() performs an
implicit rollback; use only when starting a new transaction.
You have been warned.
"""

name = ""
#branch = ""
#sem = -1
#mobile = -1
address = ""

name = des_encryption(self.nameinput.text())
# branch = self.branchinput.itemText(self.branchinput.currentIndex())
# sem = self.seminput.itemText(self.seminput.currentIndex())
# mobile = self.mobileinput.text()
address = des_encryption(self.addressinput.text())
try:
    self.conn = sqlite3.connect("data.db")
    self.c = self.conn.cursor()
    self.c.execute("INSERT INTO students (name,address) VALUES
(?,?)", (name,address))
    self.conn.commit()
    self.c.close()
    self.conn.close()
    QMessageBox.information(QMessageBox(), 'Successful', 'Student is added
successfully to the database.')
    self.close()
except Exception:
    QMessageBox.warning(QMessageBox(), 'Error', 'Could not add student
to the database.')

class SearchDialog(QDialog):
"""
Returns a MYSQL connection object. Exclusive use of
keyword parameters strongly recommended. Consult the
MySQL C API documentation for more details.

host
    string, host to connect

user
    string, user to connect as

passwd
    string, password to use

db
    string, database to use

port
    integer, TCP/IP port to connect to

unix_socket
    string, location of unix_socket (UNIX-ish only)

```

```

conv
    mapping, maps MySQL FIELD_TYPE.* to Python functions which
    convert a string to the appropriate Python type

connect_timeout
    number of seconds to wait before the connection
    attempt fails.

compress
    if set, gzip compression is enabled

named_pipe
    if set, connect to server via named pipe (Windows only)

init_command
    command which is run once the connection is created

read_default_file
    see the MySQL documentation for mysql_options()

read_default_group
    see the MySQL documentation for mysql_options()

client_flag
    client flags from MySQLdb.constants.CLIENT

load_infile
    int, non-zero enables LOAD LOCAL INFILE, zero disables
    """

def __init__(self, *args, **kwargs):
    super(SearchDialog, self).__init__(*args, **kwargs)

    self.QBtn = QPushButton()
    self.QBtn.setText("Search")

    self.setWindowTitle("Search domain")
    self.setFixedWidth(300)
    self.setFixedHeight(100)
    self.QBtn.clicked.connect(self.searchstudent)
    layout = QVBoxLayout()

    self.searchinput = QLineEdit()
    # self.onlyInt = QIntValidator()
    # self.searchinput.setValidator(self.onlyInt)
    self.searchinput.setPlaceholderText("Roll No.")
    layout.addWidget(self.searchinput)
    layout.addWidget(self.QBtn)
    self.setLayout(layout)

def searchstudent(self):
    """
    Returns a MYSQL connection object. Exclusive use of
    keyword parameters strongly recommended. Consult the
    MySQL C API documentation for more details.

host
    string, host to connect

user
    string, user to connect as

passwd
    string, password to use

db
    string, database to use

port

```

```

integer, TCP/IP port to connect to

unix_socket
    string, location of unix_socket (UNIX-ish only)

conv
    mapping, maps MySQL FIELD_TYPE.* to Python functions which
    convert a string to the appropriate Python type

connect_timeout
    number of seconds to wait before the connection
    attempt fails.

compress
    if set, gzip compression is enabled

named_pipe
    if set, connect to server via named pipe (Windows only)

init_command
    command which is run once the connection is created

read_default_file
    see the MySQL documentation for mysql_options()

read_default_group
    see the MySQL documentation for mysql_options()

client_flag
    client flags from MySQLdb.constants.CLIENT

load_infile
    int, non-zero enables LOAD LOCAL INFILE, zero disables
    """

    searchrol = ""
    searchrol = self.searchinput.text()
    try:
        self.conn = sqlite3.connect("data.db")
        self.c = self.conn.cursor()
        result = self.c.execute("SELECT * from students WHERE name="+ '\'+
searchrol + '\')
        row = result.fetchone()
        serachresult = "Rollno : "+str(row[0])+'\n'+ "Name :
"+str(row[1])+'\n'+ "Address : "+str(row[2])
        QMessageBox.information(QMessageBox(), 'Successful', serachresult)
        self.conn.commit()
        self.c.close()
        self.conn.close()
    except Exception:
        QMessageBox.warning(QMessageBox(), 'Error', 'Could not Find student
from the database.')

class DeleteDialog(QDialog):
    """
    CSV parsing and writing.

```

This module provides classes that assist in the reading and writing of Comma Separated Value (CSV) files, and implements the interface described by PEP 305. Although many CSV files are simple to parse, the format is not formally defined by a stable specification and is subtle enough that parsing lines of a CSV file with something like `line.split(",")` is bound to fail. The module supports three basic APIs: reading, writing, and registration of dialects.

DIALECT REGISTRATION:

Readers and writers support a dialect argument, which is a convenient

handle on a group of settings. When the dialect argument is a string, it identifies one of the dialects previously registered with the module. If it is a class or instance, the attributes of the argument are used as the settings for the reader or writer:

```
class excel:
    delimiter = ','
    quotechar = '"'
    escapechar = None
    doublequote = True
    skipinitialspace = False
    lineterminator = '\r\n'
    quoting = QUOTE_MINIMAL
```

SETTINGS:

- * quotechar - specifies a one-character string to use as the quoting character. It defaults to '"'
- * delimiter - specifies a one-character string to use as the field separator. It defaults to ','.
- * skipinitialspace - specifies how to interpret whitespace which immediately follows a delimiter. It defaults to False, which means that whitespace immediately following a delimiter is part of the following field.
- * lineterminator - specifies the character sequence which should terminate rows.
- * quoting - controls when quotes should be generated by the writer. It can take on any of the following module constants:

csv.QUOTE_MINIMAL means only when required, for example, when a field contains either the quotechar or the delimiter
 csv.QUOTE_ALL means that quotes are always placed around fields.
 csv.QUOTE_NONNUMERIC means that quotes are always placed around fields which do not parse as integers or floating point numbers.

csv.QUOTE_NONE means that quotes are never placed around fields.

- * escapechar - specifies a one-character string used to escape the delimiter when quoting is set to QUOTE_NONE.
- * doublequote - controls the handling of quotes inside fields. When True, two consecutive quotes are interpreted as one during read, and when writing, each quote character embedded in the data is written as two quotes

```
def __init__(self, *args, **kwargs):
    super(DeleteDialog, self).__init__(*args, **kwargs)
```

```
self.QBtn = QPushButton()
self.QBtn.setText("Delete")

self.setWindowTitle("Delete Student")
self.setFixedWidth(300)
self.setFixedHeight(100)
self.QBtn.clicked.connect(self.deletestudent)
layout = QVBoxLayout()
```

```
self.deleteinput = QLineEdit()
self.onlyInt = QIntValidator()
self.deleteinput.setValidator(self.onlyInt)
self.deleteinput.setPlaceholderText("Roll No.")
layout.addWidget(self.deleteinput)
layout.addWidget(self.QBtn)
self.setLayout(layout)
```

```
def deletestudent(self):
```

```
    delrol = ""
    delrol = self.deleteinput.text()
    try:
        self.conn = sqlite3.connect("data.db")
```

```

        self.c = self.conn.cursor()
        self.c.execute("DELETE from students WHERE roll="+str(delrol))
        self.conn.commit()
        self.c.close()
        self.conn.close()
        QMessageBox.information(QMessageBox(), 'Successful', 'Deleted From
Table Successful')
        self.close()
    except Exception:
        QMessageBox.warning(QMessageBox(), 'Error', 'Could not Delete
student from the database.')

class LoginDialog(QDialog):
    def __init__(self, *args, **kwargs):
        super(LoginDialog, self).__init__(*args, **kwargs)

        self.setFixedWidth(300)
        self.setFixedHeight(120)

        layout = QVBoxLayout()

        self.passinput = QLineEdit()
        self.passinput.setEchoMode(QLineEdit.Password)
        self.passinput.setPlaceholderText("Enter Password")
        self.QBtn = QPushButton()
        self.QBtn.setText("Login")
        self.setWindowTitle('Login')
        self.QBtn.clicked.connect(self.login)

        title = QLabel("Login")
        font = title.font()
        font.setPointSize(16)
        title.setFont(font)

        layout.addWidget(title)
        layout.addWidget(self.passinput)
        layout.addWidget(self.QBtn)
        self.setLayout(layout)

    def login(self):
        if(self.passinput.text() == ""):
            self.accept()
        else:
            QMessageBox.warning(self, 'Error', 'Wrong Password')

class AboutDialog(QDialog):
    def __init__(self, *args, **kwargs):
        super(AboutDialog, self).__init__(*args, **kwargs)

        self.setFixedWidth(370)
        self.setFixedHeight(280)

        QBtn = QDialogButtonBox.Ok # No cancel
        self.buttonBox = QDialogButtonBox(QBtn)
        self.buttonBox.accepted.connect(self.accept)
        self.buttonBox.rejected.connect(self.reject)

        layout = QVBoxLayout()

        #title = QLabel("Колодяжний I.O.")
        #font = title.font()
        #font.setPointSize(20)
        #title.setFont(font)

        labelpic = QLabel()

```

```

pixmap = QPixmap('icon/logol.png')
pixmap = pixmap.scaledToWidth(50)
labelpic.setPixmap(pixmap)
labelpic.setFixedHeight(50)

#layout.addWidget(title)

layout.addWidget(QLabel("Міністерство освіти і науки України"))
layout.addWidget(QLabel("Центральноукраїнський національний технічний
університет"))
layout.addWidget(QLabel("Факультет механіко-технологічний"))
layout.addWidget(QLabel("Тема: Програмне забезпечення системи
кібербезпеки"))
layout.addWidget(QLabel("для блокування вхідних вузлів Tor і серверів
VPN-провайдерів"))

layout.addWidget(QLabel(""))

layout.addWidget(QLabel("Спеціальність: 125 \"Кібербезпека\""))
layout.addWidget(QLabel("Освітній ступінь: бакалавр"))
layout.addWidget(QLabel("Виконав: Колодяжний І.О. "))
layout.addWidget(QLabel("Науковий керівник: Доренський О.П. "))
layout.addWidget(QLabel("Кропивницький - 2020"))
layout.addWidget(labelpic)

layout.addWidget(self.buttonBox)

self.setLayout(layout)
self.setWindowTitle("Developer")

```

```

class MainWindow(QMainWindow):
"""

```

This module provides access to the Windows registry API.

Functions:

CloseKey() - Closes a registry key.
ConnectRegistry() - Establishes a connection to a predefined registry handle on another computer.
CreateKey() - Creates the specified key, or opens it if it already exists.
DeleteKey() - Deletes the specified key.
DeleteValue() - Removes a named value from the specified registry key.
EnumKey() - Enumerates subkeys of the specified open registry key.
EnumValue() - Enumerates values of the specified open registry key.
ExpandEnvironmentStrings() - Expand the env strings in a REG_EXPAND_SZ string.
FlushKey() - Writes all the attributes of the specified key to the registry.
LoadKey() - Creates a subkey under HKEY_USER or HKEY_LOCAL_MACHINE and stores registration information from a specified file into that subkey.
OpenKey() - Opens the specified key.
OpenKeyEx() - Alias of OpenKey().
QueryValue() - Retrieves the value associated with the unnamed value for a specified key in the registry.
QueryValueEx() - Retrieves the type and data for a specified value name associated with an open registry key.
QueryInfoKey() - Returns information about the specified key.
SaveKey() - Saves the specified key, and all its subkeys a file.
SetValue() - Associates a value with a specified key.
SetValueEx() - Stores data in the value field of an open registry key.

Special objects:

HKEYType -- type object for HKEY objects
error -- exception raised for Win32 errors

Integer constants:

Many constants are defined - see the documentation for each function to see what constants are used, and where.

```

"""
def __init__(self, *args, **kwargs):
    super(MainWindow, self).__init__(*args, **kwargs)

    self.conn = sqlite3.connect("data.db")
    self.c = self.conn.cursor()
    self.c.execute("CREATE TABLE IF NOT EXISTS students(roll INTEGER PRIMARY
KEY AUTOINCREMENT ,name TEXT,address TEXT)")
    self.c.close()

    file_menu = self.menuBar().addMenu("&File")

    help_menu = self.menuBar().addMenu("&About")
    self.setWindowTitle("BlackList")

    self.setMinimumSize(400, 600)

    self.tableWidget = QTableWidgetItem()
    self.setCentralWidget(self.tableWidget)
    self.tableWidget.setAlternatingRowColors(True)
    self.tableWidget.setColumnCount(3)
    self.tableWidget.horizontalHeader().setCascadingSectionResizes(False)
    self.tableWidget.horizontalHeader().setSortIndicatorShown(False)
    self.tableWidget.horizontalHeader().setStretchLastSection(True)
    self.tableWidget.verticalHeader().setVisible(False)
    self.tableWidget.verticalHeader().setCascadingSectionResizes(False)
    self.tableWidget.verticalHeader().setStretchLastSection(False)
    self.tableWidget.setHorizontalHeaderLabels(("Roll No.",
"Name", "Address"))

    toolbar = QToolBar()
    toolbar.setMovable(False)
    self.addToolBar(toolbar)

    statusbar = QStatusBar()
    self.setStatusBar(statusbar)

    btn_ac_adduser = QAction(QIcon("icon/add.png"), "Add Student", self)
    btn_ac_adduser.triggered.connect(self.insert)
    btn_ac_adduser.setStatusTip("Add Student")
    toolbar.addAction(btn_ac_adduser)

    btn_ac_refresh = QAction(QIcon("icon/refresh.png"), "Refresh", self)
    btn_ac_refresh.triggered.connect(self.loaddata)
    btn_ac_refresh.setStatusTip("Refresh Table")
    toolbar.addAction(btn_ac_refresh)

    btn_ac_search = QAction(QIcon("icon/search.png"), "Search", self)
    btn_ac_search.triggered.connect(self.search)
    btn_ac_search.setStatusTip("Search User")
    toolbar.addAction(btn_ac_search)

    btn_ac_delete = QAction(QIcon("icon/trash.png"), "Delete", self)
    btn_ac_delete.triggered.connect(self.delete)
    btn_ac_delete.setStatusTip("Delete User")
    toolbar.addAction(btn_ac_delete)

    adduser_action = QAction(QIcon("icon/add.png"), "Insert Student", self)
    adduser_action.triggered.connect(self.insert)
    file_menu.addAction(adduser_action)

    searchuser_action = QAction(QIcon("icon/search.png"), "Search Student",
self)
    searchuser_action.triggered.connect(self.search)
    file_menu.addAction(searchuser_action)

```

```

deluser_action = QAction(QIcon("icon/trash.png"), "Delete", self)
deluser_action.triggered.connect(self.delete)
file_menu.addAction(deluser_action)

about_action = QAction(QIcon("icon/info.png"), "Developer", self)
about_action.triggered.connect(self.about)
help_menu.addAction(about_action)

def loaddata(self):
    """
    Return a compressor object.

    level
        The compression level (an integer in the range 0-9 or -1; default is
        currently equivalent to 6). Higher compression levels are slower,
        but produce smaller results.
    method
        The compression algorithm. If given, this must be DEFLATED.
    wbits
        +9 to +15: The base-two logarithm of the window size. Include a zlib
        container.
        -9 to -15: Generate a raw stream.
        +25 to +31: Include a gzip container.
    memLevel
        Controls the amount of memory used for internal compression state.
        Valid values range from 1 to 9. Higher values result in higher memory
        usage, faster compression, and smaller output.
    strategy
        Used to tune the compression algorithm. Possible values are
        Z_DEFAULT_STRATEGY, Z_FILTERED, and Z_HUFFMAN_ONLY.
    zdict
        The predefined compression dictionary - a sequence of bytes
        containing subsequences that are likely to occur in the input data.
    """
    self.connection = sqlite3.connect("data.db")
    query = "SELECT * FROM students"
    result = self.connection.execute(query)
    self.tableWidget.setRowCount(0)
    for row_number, row_data in enumerate(result):
        self.tableWidget.insertRow(row_number)
        for column_number, data in enumerate(row_data):
            print(str(data))
            #if(str(data).isdigit()): continue
            if(str(data).isdigit() != True):
                self.tableWidget.setItem(row_number,
                column_number, QTableWidgetItem(str(des.decrypt(data))[2:-1].replace(' ', '')))
            else:
                self.tableWidget.setItem(row_number,
                column_number, QTableWidgetItem(str(data)))
        self.connection.close()

    def handlePaintRequest(self, printer):
        """
        Windows: mmap(fileno, length[, tagname[, access[, offset]])

        Maps length bytes from the file specified by the file handle fileno,
        and returns a mmap object. If length is larger than the current size
        of the file, the file is extended to contain length bytes. If length
        is 0, the maximum length of the map is the current size of the file,
        except that if the file is empty Windows raises an exception (you cannot
        create an empty mapping on Windows).

        Unix: mmap(fileno, length[, flags[, prot[, access[, offset]])

        Maps length bytes from the file specified by the file descriptor fileno,
        and returns a mmap object. If length is 0, the maximum length of the map
        will be the current size of the file when mmap is called.
        flags specifies the nature of the mapping. MAP_PRIVATE creates a

```

private copy-on-write mapping, so changes to the contents of the mmap object will be private to this process, and MAP_SHARED creates a mapping that's shared with all other processes mapping the same areas of the file. The default value is MAP_SHARED.

To map anonymous memory, pass -1 as the fileno (both versions).

```

"""
    document = QTextDocument()
    cursor = QTextCursor(document)
    model = self.table.model()
    table = cursor.insertTable(
        model.rowCount(), model.columnCount())
    for row in range(table.rows()):
        for column in range(table.columns()):
            cursor.insertText(model.item(row, column).text())
            cursor.movePosition(QTextCursor.NextCell)
    document.print_(printer)

def insert(self):
    """
    Functional tools for creating and using iterators.

    Infinite iterators:
    count(start=0, step=1) --> start, start+step, start+2*step, ...
    cycle(p) --> p0, p1, ... plast, p0, p1, ...
    repeat(elem [,n]) --> elem, elem, elem, ... endlessly or up to n times

    Iterators terminating on the shortest input sequence:
    accumulate(p[, func]) --> p0, p0+p1, p0+p1+p2
    chain(p, q, ...) --> p0, p1, ... plast, q0, q1, ...
    chain.from_iterable([p, q, ...]) --> p0, p1, ... plast, q0, q1, ...
    compress(data, selectors) --> (d[0] if s[0]), (d[1] if s[1]), ...
    dropwhile(pred, seq) --> seq[n], seq[n+1], starting when pred fails
    groupby(iterable[, keyfunc]) --> sub-iterators grouped by value of
    keyfunc(v)
    filterfalse(pred, seq) --> elements of seq where pred(elem) is False
    islice(seq, [start,] stop [, step]) --> elements from
        seq[start:stop:step]
    starmap(fun, seq) --> fun(*seq[0]), fun(*seq[1]), ...
    tee(it, n=2) --> (it1, it2, ... itn) splits one iterator into n
    takewhile(pred, seq) --> seq[0], seq[1], until pred fails
    zip_longest(p, q, ...) --> (p[0], q[0]), (p[1], q[1]), ...

    Combinatoric generators:
    product(p, q, ... [repeat=1]) --> cartesian product
    permutations(p[, r])
    combinations(p, r)
    combinations_with_replacement(p, r)
    """
    dlg = InsertDialog()
    dlg.exec_()

def delete(self):
    """
    Functions to convert between Python values and C structs.
    Python bytes objects are used to hold the data representing the C struct
    and also as format strings (explained below) to describe the layout of data
    in the C struct.

    The optional first format char indicates byte order, size and alignment:
    @: native order, size & alignment (default)
    =: native order, std. size & alignment
    <: little-endian, std. size & alignment
    >: big-endian, std. size & alignment
    !: same as >

    The remaining chars indicate types of args and must match exactly;
    these can be preceded by a decimal repeat count:
    x: pad byte (no data); c:char; b:signed byte; B:unsigned byte;

```

```

?:_Bool (requires C99; if not available, char is used instead)
h:short; H:unsigned short; i:int; I:unsigned int;
l:long; L:unsigned long; f:float; d:double; e:half-float.
Special cases (preceding decimal count indicates length):
s:string (array of char); p: pascal string (with count byte).
Special cases (only available in native format):
n:ssize_t; N:size_t;
P:an integer type that is wide enough to hold a pointer.
Special case (not in native mode unless 'long long' in platform C):
q:long long; Q:unsigned long long
Whitespace between formats is ignored.

```

The variable `struct.error` is an exception raised on errors.

```
"""
```

```

    dlg = DeleteDialog()
    dlg.exec_()

```

```
def search(self):
```

```
"""
```

```

S_IFMT: file type bits
S_IFDIR: directory
S_IFCHR: character device
S_IFBLK: block device
S_IFREG: regular file
S_IFIFO: fifo (named pipe)
S_IFLNK: symbolic link
S_IFSOCK: socket file
S_IFDOOR: door
S_IFPORT: event port
S_IFWHT: whiteout

S_ISUID: set UID bit
S_ISGID: set GID bit
S_ENFMT: file locking enforcement
S_ISVTX: sticky bit
S_IREAD: Unix V7 synonym for S_IRUSR
S_IWRITE: Unix V7 synonym for S_IWUSR
S_IEXEC: Unix V7 synonym for S_IXUSR
S_IRWXU: mask for owner permissions
S_IRUSR: read by owner
S_IWUSR: write by owner
S_IXUSR: execute by owner
S_IRWXG: mask for group permissions
S_IRGRP: read by group
S_IWGRP: write by group
S_IXGRP: execute by group
S_IRWXO: mask for others (not in group) permissions
S_IROTH: read by others
S_IWOTH: write by others
S_IXOTH: execute by others

UF_NODUMP: do not dump file
UF_IMMUTABLE: file may not be changed
UF_APPEND: file may only be appended to
UF_OPAQUE: directory is opaque when viewed through a union stack
UF_NOUNLINK: file may not be renamed or deleted
UF_COMPRESSED: OS X: file is hfs-compressed
UF_HIDDEN: OS X: file should not be displayed
SF_ARCHIVED: file may be archived
SF_IMMUTABLE: file may not be changed
SF_APPEND: file may only be appended to
SF_NOUNLINK: file may not be renamed or deleted
SF_SNAPSHOT: file is a snapshot file

ST_MODE
ST_INO
ST_DEV
ST_NLINK
ST_UID

```

```
ST_GID
ST_SIZE
ST_ETIME
ST_MTIME
ST_CTIME

FILE_ATTRIBUTE_*: Windows file attribute constants
                  (only present on Windows)
"""
    dlg = SearchDialog()
    dlg.exec_()

def about(self):
    dlg = AboutDialog()
    dlg.exec_()

app = QApplication(sys.argv)
passdlg = LoginDialog()
if(passdlg.exec_() == QDialog.Accepted):
    window = MainWindow()
    window.show()
    window.loaddata()
sys.exit(app.exec_())
```

Кафедра КБПЗ – 2021 рік