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

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
“____” _____ 2021 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему
“Дослідження та програмна реалізація системи протидії шахрайським діям у мережі Інтернет”

Виконав здобувач вищої освіти
ІІ курсу, групи КІ-20МЗ
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Шкуренко К.В.
«____» _____ 2021 р.

Керівник проекту
кандидат технічних наук
_____ Тетяна СМІРНОВА
«____» _____ 2021 р.
Рецензент _____
_____

м. Кропивницький

Центральноукраїнський національний технічний університет

Факультет *Механіко-технологічний*

Центр *Заочної та дистанційної освіти*

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

Шкуренко Ксенії Всеволодівні
(прізвище, ім’я, по батькові)

1. Тема роботи *Дослідження та програмна реалізація системи протидії шахрайським діям у мережі Інтернет*

2. Керівник роботи *Смірнова Тетяна Віталіївна, канд. техн. наук*
(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 41-13 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 протидії шахрайським діям у мережі Інтернет*

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

1. Призначення та область використання.	*7. Економічна ефективність розробленої*
2.Перегляд аналогічних існуючих систем.	*програми.*
3. Опис і обґрунтування проектних рішень.	*8. Заходи з охорони праці та техніки безпеки*
4. Етапи програмування системи.	*9. Висновки.*
5. Впровадження системи в промислову	
експлуатацію	
6. Наукова новизна	

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

Наукова новизна	*1 аркуш*
Структурна схема системи	*1 аркуш*
Функціональна схема системи	*1 аркуш*
Діаграма процесів	*1 аркуш*
Блок-схема алгоритму роботи додатку	*2 аркуша*
Показники економічної ефективності	*1 аркуш*

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

Шкуренко К.В. Дослідження та програмна реалізація системи протидії шахрайським діям у мережі Інтернет. 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 протидії шахрайським діям у мережі Інтернет.

Метою розробки є дослідження та програмна реалізація системи протидії шахрайським діям у мережі Інтернет.

Об'єктом дослідження є процес протидії шахрайським діям у мережі Інтернет.

Предметом дослідження є методи протидії шахрайським діям у мережі Інтернет.

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

Результат роботи – програмна реалізація системи протидії шахрайським діям у мережі Інтернет.

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 Delphi, XUL.

Ключові слова: Комп'ютерна інженерія, шахрайські дії, Інтернет

ABSTRACT

Shkurenko K.V. Research and software implementation of the anti-fraud system on the Internet. 123 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2021

In this final qualification work on the second (master's) level of higher education the software which is intended for system of counteraction to fraudulent actions on the Internet is developed.

The purpose of development is research and software implementation of the system of counteraction to fraudulent actions on the Internet.

The object of research is the process of combating fraud on the Internet.

The subject of the research is methods of counteracting fraudulent actions on the Internet.

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 system to combat fraud on the Internet.

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 a PC IBM PC with Windows XP / Vista / 7/8/10.

The program is developed in the environment of Delphi, XUL.

Keywords: Computer Engineering, Fraud, Internet

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

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

ВСТУП

Актуальність теми. Фішинг-атаки – злочин 21-го століття. Засоби масової інформації щодня публікують списки організацій, чиї клієнти піддалися фішинговим атакам. Як тільки фішери розробляють нові прийоми атак, бізнес реагує на це розробкою нових засобів оборони, захисту персональних даних своїх клієнтів, залучає зовнішніх експертів по посиленню захисту електронної пошти. У свою чергу клієнти так само намагаються захиститися від потоку «офіційних» листів і створюють більш строгі правила спілкування [1-5].

Схований серед куп електронної макулатури, та роблячий обхід багатьох із кращих сьогоднішніх антиспамових фільтрів, новий вектор напад призначений для крадіжки конфіденційної особистої інформації. Професійні злочинці тепер використовують спеціально сформовані повідомлення, щоб заманити жертви в пастки, розроблені для крадіжки електронної тотожності користувачів [4-7].

Назва даного типу атак – Phishing (фішинг); процес обману або соціальна розробка клієнтів організацій для наступного крадіжки їхніх ідентифікаційних даних і передачі їхньої конфіденційної інформації для злочинного використання. Злочинці для свого напад використовують spam або комп'ютери-боти. При цьому розмір компанії-жертви не має значення; якість особистої інформації отриманої злочинцями в результаті напад, має значення саме по собі [8-9].

У той час як багато організацій вводять більше строгі правила у фільтрації спама, вони так само повинні приймати проактивні міри в боротьбі з фішингом. Розуміючи інструменти й методи, використовувані криміналітетом, і аналізуючи можливі діри у безпеці периметра, організації зможуть заздалегідь захиститися від багатьох популярних і успішних напрямків подібних атак.

Дана випускна кваліфікаційна робота за другим (магістерським) рівнем вищої освіти буде присвячена дослідженню деяких видів фішингу, що дозволить більш успішно захищатися від них, та розробці антифішингового програмного забезпечення.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи протидії шахрайським діям у мережі Інтернет.

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

– Огляд існуючих систем протидії шахрайським діям у мережі Інтернет.

– Дослідження системи протидії шахрайським діям у мережі Інтернет.

– Програмна реалізація системи протидії шахрайським діям у мережі Інтернет.

Об'єктом дослідження є процес протидії шахрайським діям у мережі Інтернет.

Предметом дослідження є методи протидії шахрайським діям у мережі Інтернет.

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 протидії шахрайським діям у мережі Інтернет.

– Розроблено вітчизняний продукт протидії шахрайським діям у мережі Інтернет, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі протидії шахрайським діям у мережі Інтернет.

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.

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 протидії шахрайським діям у мережі Інтернет, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Протягом сторіч, крадіжка особистості (крадіжка ідентифікаційних даних) завжди високо цінувалося злочинцями. Одержуючи доступ до чиїх-небудь особистих даних і виконуючи потім роль законного користувача, злочинець може робити анонімні злочини. Крадіжка ідентифікаційних даних ніколи не могла здійснитися більш просто, ніж у нинішнім електронному столітті.

Phishing-шахрайства продовжують рости не тільки кількісно, але і якісно з кожним місяцем. Phishing-атакам сьогодні піддається все більше число клієнтів, масове розсилання подібних листів іде на мільйони адрес електронної пошти в усьому світі. Більш того, здійснюються цілеспрямовані атаки на певні групи клієнтів. Використовуючи безліч різновидів атак, фішери можуть легко ввести в оману клієнтів для передачі їх фінансових даних (наприклад, номера платіжної карти й пароля). У той час як спам був (і продовжує бути) дратівим,
відволікаючим й обтяжуючим його одержувачів, Phishing уже показав свій потенціал, заподіюючи серйозний збиток даним і прямі втрати через шахрайське переміщення валюти.

Багато фінансових організацій і великих компаній, чий бізнес прямо пов'язаний з Інтернет ввели деякі кроки до навчання своїх клієнтів розумінню проблеми фішингу, більшість організацій, зробили дуже деяке для активної боротьби з фішерами. Однак варто розуміти, що існує багато доступних інструментальних засобів і методів для захисту від подібних атак.

Маючи високий рівень захисту від фішингових атак, організації одержать чималу вигоду від збереження лояльності своїх клієнтів.

Розвиток фішингу

Слово “phishing” спочатку виходить із аналогії, що ранні злочинці Internet використовували поштові принади для паролів і фінансових дані безлічі користувачів Internet. Використання “ph” у термінології частково загублено в літописі часу, але найбільше ймовірно пов'язане з популярними хакерськими угодами про імена типу “Phreaks”, які простежуються з історії хакерського руху, починаючи від проблеми “phreaking” – злом телефонних систем.

Термін був уперше вжитий в 1996 році хакерами, що захопили керування обліковими записами America Online (AOL) і викравши паролі користувачів AOL, які нічого не підозрювали. Перше згадування в Internet терміна фішинг було зроблено в групі новин alt.2600 hacker newsgroup у січні 1996, однак термін, можливо, використовувався навіть раніше в популярному хакерському інформаційному бюлетені "2600" [2]. До 1996 зламані облікові записи називали "phish", і до 1997 phish активно продавалися між хакерами як форма електронної валюти. Сама рання цитата, що ставиться до фішинг, була зроблена в березні 1997. Шахрайство називають **'фішинг'** – як у лові риби для вашого пароля, але записується по буквах по-іншому – сказав Татьана Го, віце-президент гарантії цілісності он-лайн сервісу [3]. Через якийсь час, визначення того, що становить фішинг-напад, було значно розширене. Термін ‘Фішинг’ тепер включав не тільки одержання подробиць облікового запису користувача, але й доступ до всіх його особистих і фінансових даних. Використовуючи спочатку повідомлення електронної пошти для одержання паролів і фінансових даних обманутих користувачів, тепер фішери створили підроблені сайти, використовували троянські програми, і атаки типу in-the-middle data proxies . У цей моме нт використання мережного шахрайства включає використання підроблених робочих місць або пропозицій роботи. Претенденти, спокушені одержанням великої кількості грошей за виконання дуже невеликої роботи, наприклад, тільки створення нового рахунку в банку, переміщення через цей рахунок грошей у

банку й пересилання їх як міжнародний грошовий переказ – класичні приклади таких атак.

Фактор соціальної інженерії

Фішинг напади покладаються на з'єднання методів технічного обману й факторів соціальної інженерії. У більшості випадків фішер переконує жертву навмисно виконати ряд дій, які забезпечать доступ до конфіденційної інформації.

У цей час фішери активно використовують популярність таких засобів зв'язку як електронна пошта, web-сторінки, IRC і служби миттєвої передачі повідомлень (IM). У всіх випадках Фішер повинен бути уособленням довіреного джерела (наприклад, служби підтримки відповідного банку й т.д.), щоб викликати довіру в жертві.

Дотепер, самі успішні фітинг-напади здійснювалися по електронній пошті – де фішер виконує роль уповноваженої особи (наприклад, імітуючи вихідну адресу електронної пошти й використовуючи впровадження відповідних корпоративних емблем). Наприклад, жертва одержує електронну пошту від *support@mybank.com* <*mailto:support@mybank.com*> (адреса – підмінена) з рядком повідомлення "модифікація захисту", у якому неї просять перейти за адресою *www.mybank-validate.info* <*http://www.mybank-validate.info*> (ім'я домену належить зловмисник, а не банку) і ввести його банківський PIN-код.

Однак фішер використовує й багато інших методів соціальної інженерії для того, щоб змусити жертву добровільно віддати конфіденційну інформацію.

Приклад. Жертва вважає, що її банківська інформація використовується ще кимось для здійснення незаконної угоди. У такому випадку жертва спробувала б увійти в контакт із відправником відповідного електронного листа й повідомити його про незаконність угоди й скасувати її. Далі, залежно від типу шахрайства, фішер попросив би (або забезпечив би мережну "безпечну" web-сторінку) для того, щоб жертва могла ввести конфіденційні подробиці (типу адреси, номера кредитної картки й т.д.), і повністю скасувати угоду. У результаті фішер одержав би досить інформації, щоб завершити реальну угоду.

Перейшовши по зазначеному посиланню, жертва попадає на сайт. Однак даний сайт, незважаючи на зовнішню повну схожість із оригінальним, призначений винятково для того, щоб жертва сама внесла конфіденційні дані.

1.2 Область застосування

Під онлайн фішингом мається на увазі, що зловмисники копіюють які-небудь сайти (найбільше часто це це інтернет-магазини онлайнової торгівлі). При цьому використовуються схожі доменні імена й аналогічний дизайн. Далі все йде за відпрацьованою схемою. Жертва, потрапляючи в такий магазин, вирішує придбати який-небудь товар. Причому число таких жертв досить велико, адже ціни в такому «неіснуючому» магазині будуть буквально непридатними, а всі підозри розсіюються через популярність копируемого сайту. Купуючи товар, жертва реєструється й уводить номер та інші дані своєї кредитної карти.

Такі способи фішингу існують уже досить давно. Завдяки поширенню знань в області інформаційної безпеки вони поступово перетворюються в неефективні способи «відібрання грошей».

Третій спосіб – комбінований. Суть даного способу полягає в тому, що створюється підроблений сайт якоїсь організації, на який потім затягуються потенційні жертви. Їм пропонується зайти на деякий сайт і там зробити якісь операції самим. Причому, як правило, використовуються методи психологічного впливу.

Численні попередження, що практично щодня з'являються в Інтернет, роблять подібні методи шахрайства усе менш і менш ефективними. Тому тепер всі частіше зловмисники прибігають до застосування key-loggers – спеціальних програм, які відслідковують натискання клавіш і відсилають отриману інформацію із заздалегідь призначених адрес.

Якщо наприкінці минулого року з'являлися одна-дві подібні програми й порядку 10-15 їхніх сайтів, що поширювали, то тепер ці показники становлять 10 і 100 відповідно.

Якщо ж ви думаєте, що фішинг -атаки актуальні лише для далекого зарубіжжя, то ви помиляєтеся.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи протидії шахрайським діям у мережі Інтернет, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

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

Електронна пошта й Spam

Найпоширеніші фішингові атаки, проведені з використанням електронної пошти. Використовуючи методи й інструментальні засоби спамерів, фішери можуть створити спеціально створені електронні повідомлення мільйонам законних адрес електронної пошти протягом декількох годин (або мінут, використовуючи розподілені бот-мережі). У багатьох випадках, списки адрес електронної пошти, купуються фішерами з тих же джерел, що й спамерами.

Використовуючи відомі недоліки в поштовому протоколі SMTP, фішери здатні створити електронні листи з підробленим рядком “Mail From:”. Заголовки в такому випадку будуть уособленням будь-якої обраної ними організації. У деяких випадках, вони можуть також установити поле "RCPT To:" на адресу обраної ними електронної пошти, внаслідок чого будь-яка відповідь клієнта на фішинговий лист буде автоматично переслана фішеру. Через все більше число повідомлень у пресі про фішингових напади, усе більше клієнтів побоюється посилати конфіденційну інформацію (типу паролів і PIN-Коду) по електронній пошті, **однак такі напади усе ще ефективні.**

Методи, використовувані фішерами при роботі з електронною поштою:

– офіційний вид листа;

– копіювання законних корпоративних адрес електронної пошти з незначними змінами URL;

– HTML, використовуваний в електронних повідомленнях, заплутує інформацію про URL;

– стандартні вкладення вірусу/хробака в повідомлення;

– використання технологій заплутування анти-спамових фільтрів;

– обробка "індивідуалізованих" або унікальних поштових повідомлень;

– підроблені відсилання поштою до популярних дощок оголошень і спискам адресатів;

– використання підробленого рядка “Mail From:” адреси й відкритих поштових релеїв маскують джерело електронної пошти.

Фішинг-атаки з використанням web-контента

Усе більш й більш популярний метод фішинг атак полягає у використанні зловмисного змісту web-сайту. Цей зміст може бути включене в сайт фішера, або сторонній сайт.

Доступні методи доставки контента включають:

– включення HTML, які маскують посилання в межах популярних сайтів;

– використання сторонніх включень або заголовків рекламних банерів для зваблення клієнтів до відвідування фішерского сайту;

– використання дефектів мережі (сховані елементи в межах сторінки – типу графічного символу нульового розміру), щоб простежити потенційного клієнта в підготовці до напад фішерів;

– використання спливаючих вікон, щоб замаскувати щире джерело фішерского повідомлення;

– впровадження зловмисного змісту в межах web-сторінки, що переглядається, що експлуатує відому уразливість у межах програмного забезпечення web-браузера клієнтів і встановлює програмне забезпечення фішера (наприклад, троянські програми).

Фальсификация рекламних банерів

Реклама за допомогою банера – дуже простий метод фішингу. Він може використовуватися для переадресації клієнта до підробленого сайту організації.

IRC і передача IM-повідомлень

Порівняно новим є використання IRC і IM-повідомлень. Однак, імовірно, цей спосіб стане популярною основою для фішинг-атак. Тому що ці канали зв'язку усе більше подобаються домашнім користувачам, і разом з тим у дане ПЗ включене велика кількість функціональних можливостей, то кількість фішинг-атак з використанням цих технологій буде різко збільшуватися. Разом з тим необхідно розуміти, що багато IRC і IM клієнтів враховують впровадження динамічного змісту (наприклад графіка, URL, мультимедіа й т.і.) для пересилання учасниками каналу, а це означає, що впровадження методів фішингу є досить тривіальним завданням. Загальне використання ботів у багатьох з популярних каналів, означає, що фішеру дуже просто анонімно послати посилання й фальсифікувати інформацію призначену потенційним жертвам.

Використання троянських програм

У той час як середовище передачі для фішинг-атак може бути різне, джерелом атаки все частіше виявляється попередньо скомпрометований домашній ПК. При цьому як частина процесу компрометації використовується установка троянського ПЗ, що дозволить фішеру (поряд зі спамерами, програмними піратами, DoS ботами й т.д.) використовувати ПК як розповсюджувачів шкідливих повідомлень. Отже, простежуючи напад фішерsв, надзвичайно складно знайти реального зловмисника.

Необхідно звернути увагу на те, що незважаючи на зусилля антивірусних компаній, число заражень троянськими програмами безупинно росте. Багато злочинних груп розробили успішні методи обману домашніх користувачів для установки в них програмного забезпечення й тепер використовують більші мережі розгорнуті за допомогою троянського ПЗ (на сьогоднішній день не рідкість становлять мережі які складаються з тисяч хостів). Дані мережі використовуються, у тому числі, для розсилання фішингових листів.

Однак не варто думати, що фішери не здатні до використання троянських програм проти конкретних клієнтів, щоб збирати конфіденційну інформацію.

Фактично, щоб зібрати конфіденційну інформацію декількох тисяч клієнтів одночасно, фішери повинні вибірково збирати записувану інформацію.

Троянські програми для вибіркового збору інформації

На початку 2004, фішери створили спеціалізований кейлоггер. Будучи впроваджений у межах стандартного повідомлення HTML (і в поштовому форматі й на декількох скомпрометованих популярних сайтах) він був кодом, що спробував запускати Java аплет, названий “javautil.zip”. Незважаючи на своє розширення zip, фактично це був файл, що виконується, що міг бути автоматично виконаний у браузерах клієнтів. Троянський кейлоггер був призначений для фіксування всіх натискань клавіш у межах вікон із заголовками різних найменувань, що включають:-commbank, Commonwealth, NetBank, Citibank, Bank of America, e-gold, e-bullion, e-Bullion, evocash, EVOCash, EVOcash, intgold, INTGold, paypal, PayPal, bankwest, Bank West, BankWest, National Internet Banking, cibc, CIBC, scotiabank and ScotiaBank.

Антифішингові технології в браузерах

Антифішинговий фільтр в Internet Explorer

В IE антифішинговий фільтр включає наступні технології:

- Вбудований фільтр. Проводить сканування відвідуваних веб-сторінок у пошуках ознак, характерних для шахрайських вузлів або фішинг-атак. Якщо користувач завантажує таку сторінку, то одержує попередження.

- Інтерактивна служба. Містить обновлювану інформацію про «шкідливі» вузли. Дуже часто фішинг-вузли з'являються на срок 24 -48 годин, тому надзвичайно важливим є процес своєчасного відновлення.

Вбудований засіб (антифішинговий фільтр) дозволяє одержувати попередження про підозрілі вузли. При цьому користувач може сам передати відомості про будь-якоі потенційно небезпечні вузли в корпорацію Microsoft для наступної перевірки. Потім підтверджена інформація додається у відповідну базу даних для захисту комп'ютерів, що використовують Internet Explorer з панеллю керування Windows Live.

У даний момент функція антифішингового фільтра доступна в оглядачі Internet Explorer. Крім того, вона є в новій панелі інструментів Windows Live [5] для IE.

Як скористатися захисними можливостями? Після завантаження й установки Internet Explorer можна включити функцію антифішингу.

Якщо при установці IE ви не активували даний захист, то зможете зробити це пізніше. Для цього в меню Tools (Сервіс) оглядача IE виберіть пункт Фішинг Filter (Антифішинг).

Фільтр розпізнає два типи «негарних» вузлів:

- веб-вузли, підозрювані в атаках;
- веб-вузли, на яких фішинг-атаки вже відбувалися.

При відвідуванні вузла, підозрюваного у фішинг-атаках, фільтр видає попередження у вигляді щита жовтого кольору. При спробі відвідати веб-вузол, на якому вживали атаки, фільтр видає щит-попередження червоного кольору й припиняє доступ до цього вузла. Введення будь-яких даних у будь-яку форму на цьому вузлі далі неможливий. Можна настроїти фішинг-фільтр на панелі інструментів Windows Live. Після інсталяції нової панелі інструментів Windows Live установите кнопку OneCare Advisor. Функція антифішингу при цьому починає роботу аналогічно IE.

Захист від фішингу в Opera

Захист від фішингу в Opera буде організована трохи по-іншому, ніж в Firefox і IE. В Opera, коли ви набирає е URL в адресному рядку, браузер буде одночасно робити запит до онлайнової бази даних Opera Software – для перевірки легітимності сайту, що ви хочете відвідати. Якщо сайт буде визначений як підставний, користувач побачить відповідне попередження. Втім, у користувача залишається можливість відвідати даний сайт, якщо йому це дійсно потрібно.

На відміну від Firefox, що звіряє адресу з відомостями, які зберігаються на ПК, Opera робить перевірку в режимі реального часу, звіряючись із постійно обновлюваною базою. Це представляється мені більше ефективним, тому що

фішингові сайти-одноденки з'являються раптово й дуже швидко зникають. Opera одержує дані про легітимність сайту з антифішингової бази від компанії GeoTrust [4].

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Embarcadero Delphi 10.2 Tokyo це найшвидший спосіб для розробки нативних крос-платформних застосунків з використанням хмарних сервісів і широкого підключення IoT. Вона надає потужні компоненти VCL для Windows 10 і забезпечує розробку на FMX для Windows, Mac і мобільних пристроїв. RAD Studio підтримує Delphi або C++ із широким спектром послуг для корпоративно-корпоративно-орієнтованого розвитку. Подивитеся на збільшений обсяг пам'яті для великих проектів, розширену підтримку декількох моніторів, поліпшений інспектор об'єктів і багато чого іншого. RAD Studio забезпечує 5-кратно збільшену швидкість розробки й розгортання на декількох настільних, мобільних, хмарних середовищах і платформах баз даних, включаючи 32-розрядні й 64-бітні ОС Windows 10.

RAD Studio 10.2, продовжуючи традицію інструментів для створення крос-платформених нативних застосунків, тепер швидше, ніж коли-або раніше, і доступна для більше широкої аудиторії розроблювачів.

Нова RAD Studio 10.2 включає:

– Перший Linux компілятор в RAD Studio для корпоративної розробки (на основі технології LLVM).

– Поліпшене меню IDE, що забезпечує швидку навігацію.

– Безліч поліпшень FireMonkey і ряд нових функцій.

– Нові можливості TDataSet.

– Підтримка режиму multi-tenancy в RAD Server.

– Відновлення FireDAC: нові й поліпшені засоби роботи з базами даних.

– Ряд удосконалень крос-платформеної RTL.

– Поліпшена підтримка SOAP.

– Значно поліпшена швидкодія скомпільованого C++, більше зручне налагодження й підтримка більшої робочої пам'яті для linker.

RAD Studio 10.2 є спадкоємцем версії 10.1.2 – першої IDE, що надала розроблювачам для 32- і 64-бітних Windows доступ до Windows store за допомогою Windows 10 Desktop Bridge і яка розширила підтримку AppStore для магазинів застосунків Windows, macOS, iOS і Android.

До складу RAD Studio входять Delphi, C++Builder, HTML5 Builder, а також всі перераховані нижче компоненти:

– ER/Studio 9.5 Developer Edition (тільки в редакції Architect) – ER/Studio допомагає проектувальникам баз даних виявляти, документувати й повторно використовувати інформаційні ресурси й надає зручні кошти зворотного проектування, аналізу й оптимізації існуючих баз даних.

– InterBase Developer Edition і InterBase ToGo (у редакції Professional і редакціях з більше широкими можливостями) – InterBase Developer Edition – це краща крос-платформна база даних для створення й тестування застосунків баз даних, призначених для вбудовування й підтримки малих і середніх підприємств.

– База даних IBLite для Windows, OS X, Android і iOS з ліцензією на безкоштовне розгортання (у редакції Professional з пакетом доповнень для мобільних середовищ і в редакціях Enterprise і Architect). Нова редакція InterBase, яку можна використовувати в застосунках і встановлювати безкоштовно. Розроблювачі, що бажають вмонтувати в додаток базу даних з більше широкими можливостями, можуть використовувати InterBase ToGo (продається окремо).

Додаткові засоби. RAD Studio містить широкий набір додаткових засобів, що допомагають реалізовувати недоступні раніше можливості. В основному це версії засобів, створені спеціально для середовища RAD Studio, і їхній набір можливостей може відрізнятися від повнофункціональних комерційних версій:

– FastReport VCL RAD Edition і FastReport FireMonkey Embarcadero Edition від компанії Fast Reports – швидке проектування й формування звітів.

– TeeChart Standard від компанії Steema – швидке й просте створення діаграм.

– CodeSite Express – ведення журналів для підвищення якості застосунків.

– Beyond Compare Text Compare – порівняння файлів вихідного коду.

– IntraWeb від компанії Atozed – створення веб-застосунків за принципом проектування застосунків RAD Studio.

XUL

XUL XML User Interface Language) – мова розмітки для створення графічних інтерфейсів користувача, основана на XML. XUL поширюється та розробляється в межах проекту Mozilla.

XUL розроблено для створення інтерфейсів у таких програмах, як браузери, поштовий клієнт, програма-календар, редактор HTML NVU, медіа-програвач. XUL можна ефективно використовувати для створення будь-яких програм та розширень, пов'язаних з роботою з веб-ресурсами і не тільки.

XUL, як і HTML, описує інтерфейси за допомогою мови розмітки і дозволяє задавати зовнішній вигляд програми через CSS та визначати поведінку за допомогою JavaScript. Однак, на відміну від HTML, XUL дозволяє створювати динаміку користувацького інтерфейсу набагато швидше та зручніше. XUL надає багатий набір компонентів, з можливим побудувати інтерфейс розширення чи програми.

Знання XUL – основа для створення застосунків для продуктів Mozilla, оскільки більша частина їх інтерфейсу написана на XUL.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускну кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне

забезпечення, яке призначено для системи протидії шахрайським діям у мережі Інтернет.

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

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

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

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

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

д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи 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) сформувати висновки про виконаний обсяг робіт та одержані результати.

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Напрямки фішингових атак

Фішери повинні використовувати масу методів шахрайства для того щоб здійснити успішні напади. Самі загальні включають:

– напад " людин у середині " (in-the-middle Attacks);

– напад підміни URL;

– напад, що використовують Cross-site Scripting;

– попередньо встановлені сесії атак;

– підміна клієнтських даних;

– використання уразливості на стороні клієнта.

Напад "людин у середині" (in-the-middle)

Одним із самих успішних способів одержання керування інформацією клієнта й ресурсами є напад "людин у середині". У цьому класі атак нападаючий розташовує себе між клієнтом і реальним додатком, доступним через мережу. Із цієї точки, той хто нападає може спостерігати й робити запис всіх подій.

Ця форма нападу успішна для протоколів HTTP і HTTPS. Клієнт з'єднується із сервером нападаючих, начебто з реальним сайтом, у той час як сервер нападаючих робить одночасне підключення до реального сайту. Сервер нападаючих у такому випадку відіграє роль проксі-сервера для всіх з'єднань між клієнтом і доступним через мережу прикладним сервером у реальному масштабі часу.

У випадку безпечного з'єднання HTTPS, підключення SSL установлюється між клієнтом і проксі-сервером нападаючих (отже, система нападаючих може робити запис усього трафика в незашифрованому стані), у той час як проксі-

сервер нападаючих створює своє власне підключення SSL між собою й реальним сервером.

Для проведення успішних атак "людина в середині ", той хто нападає повинен бути приєднаний прямо до клієнта замість реального сервера. Це може бути виконане за допомогою безлічі методів:

– DNS Cache Poisoning
– URL Obfuscation
– Browser Proxy Configuration

Прозорі проксі-сервери

Розташований у тім же сегменті мережі або розташований на маршруті на реальний сервер (наприклад, корпоративний гейтвей), transparent proxy service може перервати всі дані, пропускаючи весь вихідний HTTP і HTTPS через себе. У цьому випадку ніякі зміни конфігурації на стороні клієнта не потрібні.

DNS Cache Poisoning (отруєння кеша DNS)

DNS Cache Poisoning може використовуватися, щоб перервати нормальну маршрутизацію трафика, вводячи помилкові адреси IP для ключових імен домену. Наприклад, той хто нападає модифікує кеш доменної системи імен мережного міжмережного захисту так щоб весь трафик, призначений для адреси IP MyBank тепер ішов на адресу IP проксі-сервера того, хто нападає.

URL Obfuscation

Використовуючи даний метод, зловмисник змінює зв'язок замість реального сервера на з'єднання з їх проксі-сервером. Наприклад, клієнт може переходити на посилання до <http://www.mybank.com.ch/> замість <http://www.mybank.com/>

Конфігурація проксі-сервера в браузері клієнта

Даний тип атаки може бути легко замічений клієнтом при огляді настроювань браузера. У багатьох випадках зміна настроювань браузера буде здійснено безпосередньо перед фішинг-повідомленням.

Напад підміни адрес

Таємниця багатьох фішингових нападів полягає в тому, щоб змусити одержувача повідомлення випливати за линком (URL) на сервер зловмисник, не розуміючи, що він був обманутий. На жаль фішери мають доступ до все більшого арсеналу методів щоб заплутати кінцевого клієнта.

Самі звичайні методи підміни адрес включають:

– Bad domain names.
– Friendly login URL's.
– Host name obfuscation.
– URL obfuscation.

Погані імена домену

Один з найбільш тривіальних методів підміни використання поганих імен домену. Розглянемо фінансовий інститут MyBank із зареєстрованим доменом *mybank.com* і пов'язаний із клієнтом діловий сайт <*http://privatebanking.mybank.com*>. Фішер міг установити сервер, використовуючи кожне з наступних імен, щоб заплутати реальний хост адресата:

http://privatebanking.mybank.com.:

– http://mybank.privatebanking.com/
– http://privatebanking.mybonk.com/
– http://privatebanking.mybank.com/
– http://privatebanking.mybank.hackproof.com/

Важливо звернути увагу на те, що, оскільки організації реєстрації доменов рухаються в напрямку інтернаціоналізації їхніх послуг, отже, можлива реєстрація імен доменів на інших мовах і певних наборах символів. Наприклад, “o” у символах кирилиці виглядає ідентично стандартному ASCII “o”, але доменне ім'я буде іншим.

Нарешті, це варто відзначати, що навіть стандартний набір символів ASCII ураховує двозначності типу верхнього регістра “i” і нижнього регістра “L”.

Friendly Login URL's

Багато web-браузерів ураховують складний URL, що може включити розпізнавальну інформацію типу ім'я вхідного в систему й пароля. Загальний формат – URL://username:password@hostname/path.

Фішери можуть замінити ім'я користувача й поле пароля. Наприклад наступний URL установлює *ім'я користувача = mybank.com*, *пароль = ebanking*, і ім'я хоста адресата – *evilsite.com.*

<http: // mybank.com:ebanking@evilsite.com/фішинг/fakepage.htm>

Цей дружній вхід у систему URL може успішно обдурити багатьох клієнтів, які будуть уважати, що вони фактично відвідують законну MyBank сторінку. Через успіх даного методу, багато поточних версій браузерів забрали підтримку даного методу кодування URL.

Підміна імен хостів

Більшість користувачів Internet знайомо з навігацією по сайтах і послугам, використовуючи повне ім'я домену, типу www.evilsite.com <http://www.evilsite.com>. Для того щоб web-браузер міг зв'язатися з даним хостом по Internet, ця адреса повинен бути перетворений на адресу IP, типу 209.134.161.35 для www.evilsite.com <http://www.evilsite.com>. Це перетворення IP-адреси в ім'ї хоста досягається за допомогою серверів доменних імен. Фішер може використовувати адресу IP як частина URL, щоб заплутати хост і можливо обійти системи фільтрації змісту, або сховати адресат від кінцевого користувача.

Наприклад наступний URL:

<http: // mybank.com:ebanking@evilsite.com/фішинг/fakepage.htm>

міг бути заплутаним по наступному сценарію:

<http: // mybank.com:ebanking@210.134.161.35/login.htm>

У той час як деякі клієнти знайомі із класичним десятковим поданням адрес IP (000.000.000.000), більшість із них не знайомо з іншими можливими поданнями. Використовуючи ці подання IP у межах URL, можна привести користувача на фішерский сайт.

Залежно від додатка, що інтерпретує адресу IP, можливе застосування різноманітних способів кодування адрес крім класичного пунктирно-десяткового формату. Альтернативні формати включають:

– Dword – значення подвійного слова, тому що це складається по суті із двох подвійних "слів" 16 біт; але виражено в десятковому форматі,

– Восьмеричний

– Шестнадцатеричний.

Ці альтернативні формати найкраще пояснюються, використовуючи приклад. Розглянемо URL <http://www.evilsite.com/>, перетворюючи до IP-адреси 210.134.161.35. Це може інтерпретуватися як:

– Десяткове число – *<http: // 210.134.161.35/>*

– Dword – *http:// 3532038435/*

– Восьмеричний – *<http: // 0322.0206.0241.0043/>*

– Шестнадцатеричний – *<http://0x2.0x86.0x1.0x23/>* або навіть *<http:/0x286A123/>*

– У деяких випадках, можливо навіть змішати формати (наприклад *<http:// 0322.0x86.161.0043/>*).

Підміна URL

Щоб гарантувати підтримку місцевих мов у програмному забезпеченні Internet типу web-браузерів, більшість програмних забезпечень підтримує додаткові системи кодування даних.

Cross-site Scripting Attacks

Типові формати CSS ін'єкції в достовірний URL включають:

Повна заміна HTML типу:

<http://mybank.com/ebanking?URL=http://evilsite.com/phishing/fakepage.htm>

Вбудоване впровадження сценарію, типу:

*http: // mybank.com/ebanking? Page=1*client = <СЦЕНАРІЙ> evilcode <http: // mybank.com/ebanking? Page=1*client = % 3cSCRIPT%3eevilcode>...*

Наприклад, клієнт одержав наступний URL за допомогою електронного фішинг-листа:

<http: // mybank.com/ebanking? URL=http: // evilsite.com/phishing/fakepage.htm>

У той час як клієнт дійсно спрямований і пов'язаний з реальним MyBank додатком мережі, через помилкове кодування додатка банком, *ebanking* компонент прийме довільний URL для вставки в межах поля *URL* повернутої сторінки. Замість додатка, що забезпечує розпізнавальну форму MyBank, впроваджену в межах сторінки, зловмисник пересилає клієнта до сторінки під керуванням на зовнішньому сервері *(<http: // evilsite.com/phishing/fakepage.htm>)*.

Методи протидії фішинговим атакам

Як може звичайний користувач протистояти атаці фішерів? Насправді, варто задуматися над декількома правилами:

1. Ніколи не відповідайте на листи, що запитують вашу конфіденційну інформацію
2. Відвідаєте веб-сайт банку шляхом введення його URL-адреси через адресний рядок браузера
3. Регулярно перевіряйте стан своїх онлайн-рахунків
4. Перевірте рівень захисту відвідуваного вами сайту
5. Виявіть обережність, працюючи з електронними листами й конфіденційними даними
6. Забезпечте захист своєму комп'ютеру
7. Завжди повідомляйте про виявлену підозрілу активність

Розглянемо даного правила докладніше.

Ніколи не відповідайте на листи, що запитують вашу конфіденційну інформацію

Як правило, банки й фінансові компанії, що займаються електронною комерцією, розсилають персоніфіковані звернення клієнтам, а фішери – ні! Фішери часто використовують кричаще звучні заголовки листів типу

«Терміново! Ваші реквізити можуть бути викрадені!» для того, щоб змусити користувача негайно перейти по посиланню.

Варто пам'ятати, що поважаючі себе компанії ніколи не запитують у клієнтів паролі або дані рахунків за допомогою електронної пошти. Навіть, якщо вам здалося, що лист легітимний – не варто відповідати на нього, краще приїхати в офіс компанії або, у крайньому випадку, передзвонити їм по телефоні.

Варто пам'ятати про обережність при відкриванні вкладень в електронні листи або при завантаженні через Інтернет по посиланнях, незалежно від того, хто є відправником цих листів!

Відвідування веб-сайту банку або компанії

Для відвідування веб-сайту банку наберіть в адресному рядку браузера його URL-адресу.

Фішери часто використовують так звані «схожі» адреси. Однак якщо піти по такому «схожому» посиланню, ви можете потрапити на фішерский сайт замість справжнього сайту банку.

Це не дасть вам повну гарантію безпеки, однак зможе вберегти хоча б від деяких видів атак фішеров.

Регулярно перевіряйте стан своїх рахунків

У випадку виявлення підозрілої транзакції негайно зв'яжіться з вашим банком. Одним з найпростіших засобів перевірки стану рахунку є так званий SMS-банкинг.

Не менш розповсюджений спосіб на сьогоднішній день пов'язаний з лімітуванням операцій. У такому випадку клієнтові досить установити суму гранично можливого зняття готівки або платежу в торговельній точці, і банк не дозволить ні йому, ні шахраєві вийти за встановлені рамки.

Перевірте рівень захисту відвідуваного вами сайту

Перед введенням конфіденційної інформації на сторінці сайту вашого банку, не заважає провести пару перевірок, щоб переконатися у використанні банком криптографічних методів.

По-перше, перевірте веб-адресу в адресному рядку браузера. Якщо веб-сайт, що ви вирішили відвідати, розташований на захищеному сервері, то адреса повина починатися з "https://" ("s" від security), а не зі звичайного "http://".

По-друге, перевірте також стан іконки із зображенням замка в статусному рядку вашого браузера. Ви можете перевірити рівень криптозахисту, обумовлений кількістю біт, поводивши курсор мишки над цією іконкою.

Виявіть обережність, працюючи з електронними листами й конфіденційними даними

Більшість банків мають на своїх веб -сайтах сторінку з питань безпеки, де повідомляється інформація про те, як проводити транзакції в захищеному режимі, а також загальні ради по захисту конфіденційних даних: ніколи й нікому не відкривайте свої PIN-Коди або паролі, не записуйте їх і не використовуйте той самий пароль для всіх своїх онлайн-рахунків.

Не відкривайте спамові листи й не відповідайте на них, тому що цими діями ви даєте відправникові листа коштовну інформацію про те, що він роздобув діючу електронну адресу.

Користуйтеся здоровим глуздом, коли читаєте електронні листи. Якщо щось у листі вам здається неправдоподібним або настільки гарним, що не віриться, то, швидше за все, так воно і є.

Захистить свій комп'ютер

Варто пам'ятати, що найбільш ефективним захистом від троянських програм служить антивірусне ПЗ. Останнім часом деякі антивірусні компанії стали вбудовувати у свої продукти так звані антифішингові фільтри. Зокрема, антифішинговий фільтр вбудований у ПЗ від Лабораторії Касперского, Symantec і т.д. Крім того, у сучасних версіях браузерів з'явилися свої варіанти антифішингових фільтрів.

З великою часткою ймовірності можно затверджувати, що лист фішинговий, якщо:

– тема й тіло листа бідно відформатовані й містять багато помилок;

– лист починається з безликого вітання типу «Дорогий клієнт»;

– манера викладу листа така, начебто вас хочуть урятувати від якоїсь неприємності і єдиний спосіб не втратити свої гроші – повідомити конфіденційну інформацію про стан ваших рахунків.

А от рекомендації при використанні банківської картки в мережі Інтернет, розроблені українськими фахівцями:

– не відповідайте на електронні листи, у яких нібито від імені банку вас просять надати персональну інформацію. Зв'яжіться з банком по номеру телефону, що відомий вам як дійсний, щоб з'ясувати дійсність листа;

– ніколи не переходьте по посиланнях у таких листах (навіть на сайт банку), тому що вони можуть вести на шахрайські сайти;

– ніколи не розкривайте персональну інформацію або інформацію з карти (рахунку) через Інтернет;

– користуйтеся послугами тільки відомих і перевірених торговельних підприємств. Перевагу необхідно віддавати підприємствам, підключеним до програм Verified by Visa (Перевірено Візою) і Secure Code (Безпечний Код);

– перевіряйте адреси інтернет-сайтів, до яких ви підключаєтеся, тому що зловмисники можуть використовувати схожі назви для створення шахрайських ресурсів;

– уникайте користуватися послугами інтернет-ресурсів сумнівного змісту: найчастіше вони створюються спеціально для одержання інформації про банківські карти й наступного її неправомірного використання;

– контролюйте свою електронну пошту, не відкривайте повідомлення від невідомих адресатів, не передавайте свої особисті дані;

– поставте на свій комп'ютер антивірусне програмне забезпечення й регулярно робите його відновлення й відновлення інших використовуваних вами програмних продуктів;

– робіть покупки тільки зі свого комп'ютера. Не користуйтеся інтернет-кафе й іншими доступними засобами, де можуть бути встановлені програми-шпигуни, що запам'ятовують конфіденційні дані, що вводяться вами;

– вибирайте паролі, які не пов'язані з вашим днем народження або іншими персональними даними. Не записуйте паролі й нікому не повідомляйте їх.

Засоби антифішингу в Windows

Система Windows покликана стати самою безпечною з коли-або випущених версій Windows. Вона постачена новими функціями для забезпечення безпечної роботи користувачів в Інтернеті.

Захистіть свій комп'ютер

Підвищте рівень захисту від програм-шпигунів. Захисник Windows — це антишпигунська програма для системи Windows. Вона дозволяє уникнути з роботи комп'ютера, втрати конфіденційних даних і появи небажаних спливаючих рекламних вікон, викликаних програмами-шпигунами й іншими потенційно небажаними програмами.

Переглядайте веб-сторінки з більшою впевненістю в захищеному режимі оглядача Internet Explorer. Ця функція, що є тільки в системі Windows , обмежує повноваження користувача при роботі з оглядачем Internet Explorer, надаючи йому досить прав для перегляду веб-сторінок, але не дозволяючи змінювати файли й настроювання. Це забезпечує захист комп'ютера від атак через Інтернет.

Додаткові можливості захисту в центрі забезпечення безпеки. Центр забезпечення безпеки Windows повідомить користувача й допоможе вжити заходів по усуненню проблеми, якщо встановлене програмне забезпечення не оновлене або рівень безпеки занадто низок і існує потенційна небезпека. Центр забезпечення безпеки Windows у системі Windows удосконалений. У нього додані відомості про антишпигунські програми, настроювання оглядача Internet Explorer і параметрах керування обліковим записом користувача.

Одержіть додатковий контроль над роботою програм. За замовчуванням у системі Windows програми запускаються в більш безпечному режимі. При роботі більшості додатків у випадку спроби виконати потенційно небезпечну операцію, що вимагає повноважень адміністратора, у системі Windows виводиться запит на одержання згоди користувача. Це дозволяє знизити ризик проникнення вірусів, програм-шпигунів і інших погроз. Щоб захистити сімейний комп'ютер, створіть більш безпечні стандартні облікові записи користувачів для всіх членів родини й використовуйте їх при виконанні щоденних завдань. У цьому випадку, якщо дитина спробує встановити яку-небудь програму, комп'ютер запросить пароль облікового запису адміністратора. Якщо дитина його не знає, вона не зможе самостійно встановити нові програми або перевизначити параметри батьківського контролю.

Захистіть особисті дані

Використовуйте антифішинг-фільтр для захисту облікових даних. Оглядач Internet Explorer у складі системи Windows має фільтр, що при перегляді веб-вузлів повідомляє про можливість фішинг-атаки з метою розкрадання конфіденційної інформації. Фільтр виконує перевірку за списком відомих веб-вузлів фішингу, обновлюваному кілька разів у годину, а також виявляє підозрілі веб-вузли, ще не занесені в базу даних.

Видалення даних журналу користування Інтернетом одним клацанням. Інформація про відвідуваний веб-вузли й відомості, що вводяться при перегляді веб-вузлів, зберігаються в різних місцях на комп'ютері. В оглядачі Internet Explorer у системі Windows не потрібно видаляти особисті дані в різних місцях. З функцією видалення історії перегляду можна видалити всі відомості про перегляд одним клацанням миші.

Архівуйте й відновлюйте настроювання, файли й додатки. У системі Windows є більше повний і простий засіб резервного копіювання, чим базова програма архівації системи Windows XP. Нова програма архівації Windows забезпечує додаткові можливості збереження резервних копій даних.

Користувачам більше на прийде турбуватися про регулярну архівацію даних. Зручний майстер дозволяє становити розклад архівації й указувати місце збереження резервних копій.

Захистіть свою родину

Захистіть дітей, використовуючи батьківський контроль. Windows надає широкий спектр потужних засобів батьківського контролю, що допомагають спостерігати за тим, як діти користуються комп'ютером, контролювати цей процес і забезпечувати їхня безпеку.

Переглядайте докладні звіти про дії. Windows створює докладні звіти про дії дітей на комп'ютері, включаючи відомості про те, у які ігри вони грали, які веб - вузли відвідували і які додатки запускали.

Установіть обмеження перегляду веб-вузлів. Безкоштовна веб-служба, надавана із системою Windows, дозволяє обмежити типи веб-вузлів, які може відвідувати дитина. Можна обмежити перегляд веб-вузлів по категоріях, наприклад заблокувати всі веб-вузли з порнографічним умістом або азартними іграми, а також окремі веб-вузли по URL-адресі. Ці обмеження підтримуються більшістю веб-оглядачів.

Контролюйте гри, у які грає дитина . Система Windows дозволяє легко вказати, у які ігри можуть грати діти. З її допомогою можна:

– дозволяти або забороняти дітям грати в певні ігри;

– обмежити можливість грати в ігри, призначені для певного віку;

– блокувати ігри, що містять інформацію, небажану для перегляду або прослуховування дітьми.

Установіть межі часу роботи за комп'ютером. У системі Windows можна вказати, коли й протягом якого часу дитина може користуватися комп'ютером.

Дії у випадку фішинг-атаки

Ви можете зробити все для запобігання крадіжки особистих даних у результаті фішинг-атаки, але повну безпеку й захист не може гарантувати жодна

система й жоден методи. Якщо ви підозрюєте, що вже відповіли на фішинг - повідомлення, указавши особисту або фінансову інформацію або ввівши її на фальшивому веб-вузлі, ви ще можете звести збиток до мінімуму.

Крок 1 Заявіть про порушення безпеки

Якщо ви думаєте, що ваша особиста інформація піддалася небезпеки або була викрадена, негайно повідомите про обставини події в наступні організації:

– **У компанію по випуску кредитних карт, якщо ви подали відомості про кредитну карту.** Це потрібно зробити в першу чергу. Чим раніше в організації довідаються, що ваш рахунок піддався небезпеки, тим легше їм буде захистити вас.

– **У компанію, чиє ім'я, на вашу думку, було використано.** Помніть, що звернутися в організацію потрібно безпосередньо, а не через отримане повідомлення електронної пошти.

– **У Центр скарг на шахрайства в Інтернеті (IFCC).** Центр Internet Fraud Complaint Center (IFCC) — це результат партнерства ФБР і національного центра по боротьбі зі злочинами «білих комірців» (NW3C). Він працює по усьому світі за підтримкою правоохоронних органів і представників галузі, негайно закриває підроблені веб-вузли й виявляє за ними зловмисників, які стоять за ними.

– **Федеральна комісія з торгівлі.** Якщо ви думаєте, що ваша особиста інформація піддалася небезпеки або була викрадена, повідомте про обставини події у Федеральну комісію з торгівлі (FTC): національний центр по розкраданнях особистих даних і відвідайте її веб-вузол, щоб довідатися, як звести можливий збиток до мінімуму.

Про фішинг-атаку можна також повідомити в робочу групу Anti -phising Working Group за адресою reportphising@antiphising.org, а також у Федеральну комісію з торгівлі (FTC) за адресою spam@uce.gov. Для цього створіть нове повідомлення, адресоване цим групам, і вкладіть в нього шахрайське повідомлення електронної пошти. Можна також повністю скопіювати його й вставити в нове повідомлення. Намагайтеся не використовувати команду

«Переслати», оскільки цей формат може виключити деяку інформацію й зажадати ручної обробки повідомлення.

Крок 2 Змініть паролі для всіх облікових записів.

Якщо ви думаєте, що повідомили пароль у відповідь на фішинг-сообщение або ввели його на шахрайському веб-вузлі, переміните його якомога швидше.

Крок 3 Регулярно перевіряйте банківські звіти й звіти по кредитних картах

Роблячи це принаймні щомісяця, ви зможете піймати шахраїв і зупинити їх, перш ніж вони нанесуть вам відчутний збиток.

Крок 4 Застосовуйте новітнє антивірусне й антишпигунське програмне забезпечення

Деякі шахрайські повідомлення електронної пошти можуть містити шкідливі або небажані програми, які відслідковують дії користувача або просто сповільнюють роботу комп'ютера.

3.2 Розробка структурної схеми

На структурній схемі (рисунок 3.1) зображена розроблена під час дипломного проектування система протидії шахрайським діям у мережі Інтернет.

Розробка системи антифішингу повинна ґрунтуватися на потужній базі – браузері з розширеними можливостями. Так як найпоширеніший браузер Microsoft Internet Explorer поширюється із закритим вихідним кодом, а також має сховану структуру, що негативно впливає при написанні додаткових програм і плагінів на його основі й проаналізувавши існуючі на даний момент браузери, і їхні розподілені системи захисту , я зупинив свій вибір на браузері FireFox.

Він має величезну кількість переваг, головна з яких – надання великої кількості підпрограм що дозволяють одержати доступ до пошти, використанню Інтернет-пейджерів систем IRC, Viber, Telegram, WhatsApp, AIM.

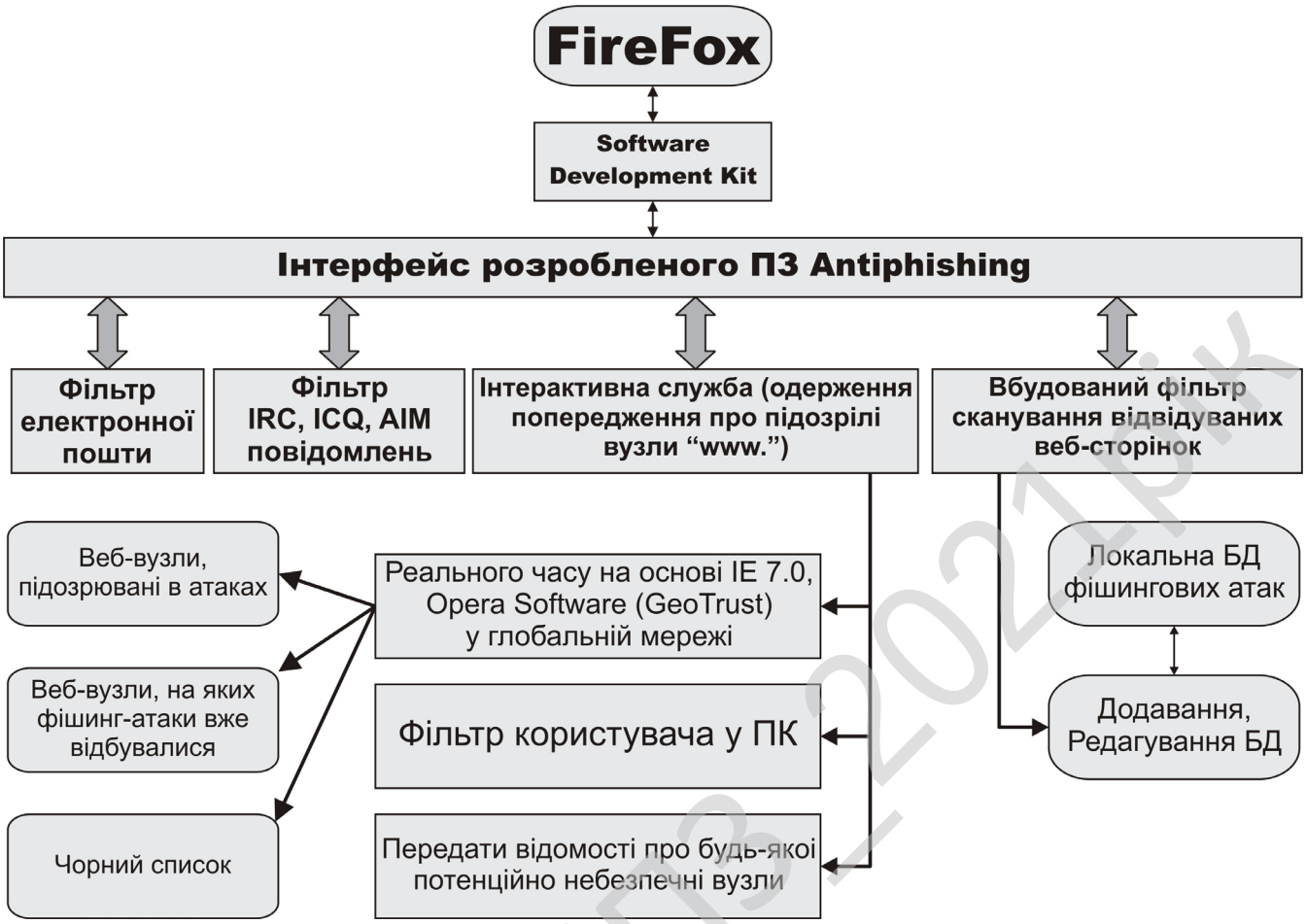


Рисунок 3.1 – Структурна схема роботи та взаємодії системи

FireFox розповсюджується із частково відкритим кодом і має розширений набір засобів (Software Development Kit) для написання й розповсюдження додатків на його основі.

Як показано на рисунку 3.1, система антифішингу заснована на браузері FireFox і робить взаємодію через Software Development Kit.

Розглянемо загальні можливості розробленого програмного забезпечення зображені на схемі. Через розроблену систему антифішингу відбуваються наступні дії:

1. Фільтр електронної пошти – дозволяє частково убезпечити поштові повідомлення від фішингових атак розширеним керуванням і контролем даних, що надходять,. Фільтр спільно працює з антивірусними програмними продуктами

й фаєрфолами (якщо такі присутні в операційній системі) не викликаючи конфліктних ситуацій і зависань тому що працює через браузер Firefox.

2. Фільтр IRC, Viber, Telegram, WhatsApp, AIM повідомлень – При використанні внутрішньої програми спілкування через Інтернет-пейджери IRC, Viber, Telegram, WhatsApp, AIM – розроблене програмне забезпечення антифішингу дозволяє контролювати процес передачі файлів і не дати зробити несанкціонований запуск шкідливої програми на ПК.

3. Інтерактивна служба (одержання попередження про підозрілі вузли “www.”) – складається із трьох підрозділів, які дозволяють інтерактивно контролювати WEB контент, який попадає на машину користувача.

3.1 Реального часу на основі IE, Opera Software (GeoTrust) у глобальній мережі – з версії браузерів IE і Opera з'явився новий безкоштовний Інтернет сервіс, який надає доступ до всесвітньої бази перевірки веб-вузлів. В Інтернеті існує величезна кількість посилань на сайти при переході на які, відбувається запуск шкідливого програмного забезпечення й крадіжка особистої інформації, у даних випадках антивирустні програми неспроможні тому що використовуючи помилки ОС шкідливі програми одержують статус перевірених. За допомогою цього безкоштовного сервісу й розробленого в дипломному проекті можна значною мірою усунути можливість запуску такого шкідливого коду.

Інтерактивний сервіс повертає наступні повідомлення:

– Веб-вузли, підозрювані в атаках;

– Веб-вузли, на яких фішинг-атаки вже відбувалися;

– Чорний список .

3.2 Фільтр користувача в ПК – локальний фільтр блокування доступу складений користувачем. Існують різні ситуації під час роботи ПК, фільтр користувача дозволяє скласти список ресурсів доступ на який буде заборонений при переадресаціях і.т.ін.

3.3 Передати відомості про будь-які потенційно небезпечні вузли – можливість послати в інтерактивну службу дані для перевірки на наявність на сайті фішингово- шкідливого коду.

4. Вбудований фільтр сканування відвідуваних веб-сторінок – розширені можливості переходу на сторінки, які відвідувалися, з перевіркою на можливу переадресацію, а також база шаблонів відомих шкідливих кодів з можливістю редагування.

4.1 Додавання, Редагування БД – можливість редагування й ручного додавання шаблона відомих шкідливих кодів.

4.2 Локальна БД фішингових атак – шаблонів відомих шкідливих кодів.

За допомогою даних засобів імовірність фішингової атаки через електронну пошту й Spam, фішинг-атаки з використанням web-контента, фальсифиция рекламних банерів, IRC і передача IM-повідомлень, використання троянських програм значно зменшується надаючі користувачеві надійну систему захисту.

3.3 Розробка функціональної схеми

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

Користувач використовуючи Інтернет браузер FireFox одержує доступ до гловальної мережі Інтернет та через розроблене програмне забезпечення Antiphishing взаємодіе з Інтернетом (Web-контент), розмовляє за допомогою Інтернет пейджерів (IRC, Viber, Telegram, WhatsApp, AIM), користується електронною поштою.

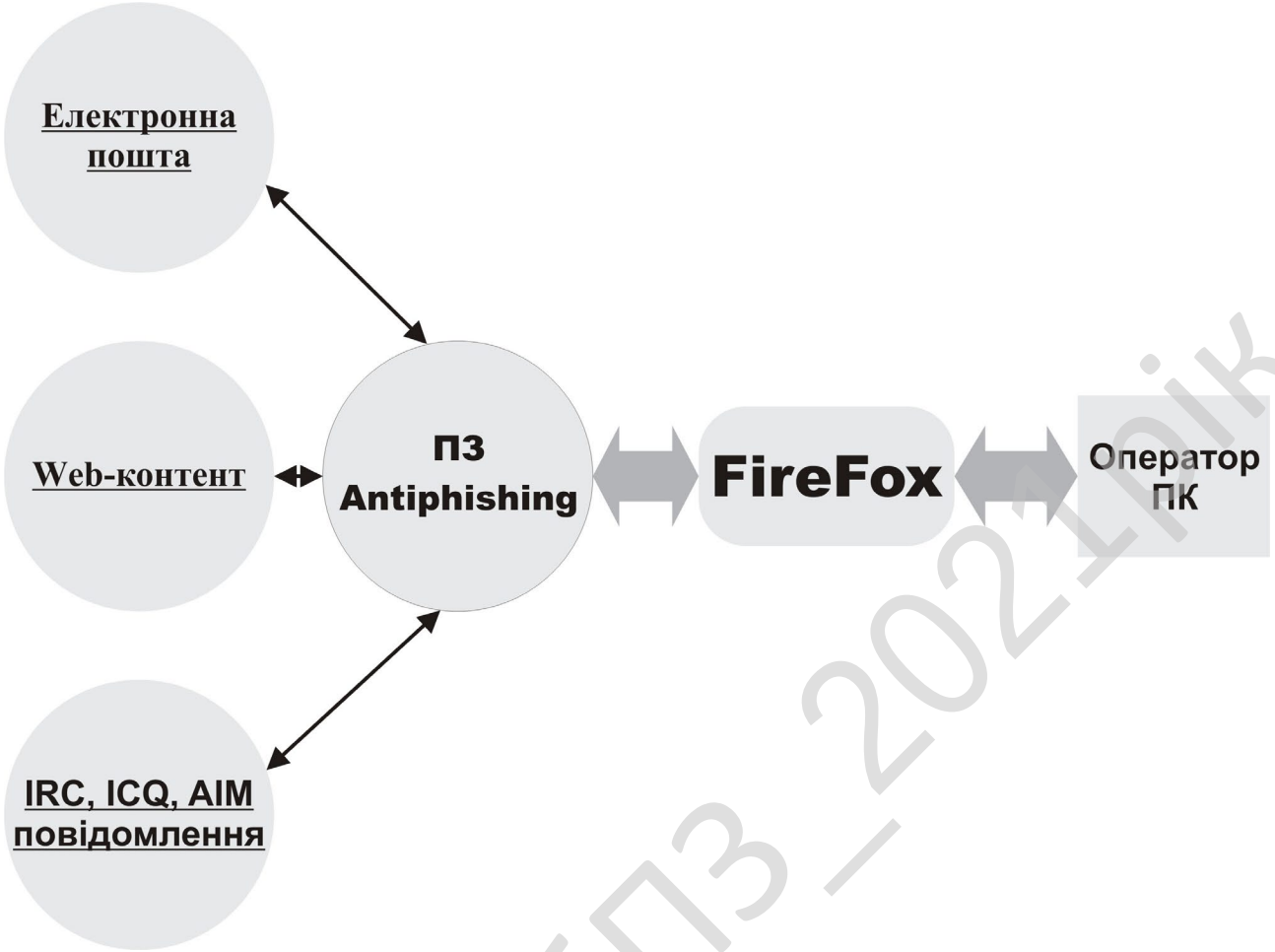


Рисунок 3.2 – Функціональна схема системи

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

Програмне забезпечення Antiphishing є системою розширення (extentions) можливостей FireFox надаючі користувачеві захист від фішингових атак.

3.4 Розробка діаграми процесів

Одним з важливих критеріїв при розробці будь-якого програмного забезпечення це грамотна розробка структури роботи системи потоків і процесів.

На рисунку 3.3 зображена діаграма роботи розробленого програмного забезпечення дипломного проектування.

Через головний блок ПЗ у якому починається й завершується програма, ми попадаємо в головне вікно браузера FireFox з доступом до головного вікна розробленого програмного забезпечення й модулю захисту й обробника помилок FireFox.

Через головне вікно розробленого програмного забезпечення й інтерфейсну частину Antiphishing реалізованого мовою framework XUL ми попадаємо в блок настроювання програми, де відбуваються дії по настроюванню – фільтру сканування відвідуваних веб-сторінок, фільтру повідомлень, фільтру електронної пошти та інтерактивної служби.

Мова framework XUL мова опису користувальницького інтерфейсу. XUL використовується у всіх основних продуктах сімейства Mozilla, що при проектуванні диплома на антифішингову тематику просто знахідка. Особливості цієї мови визначають легкість роботи з ним:

- це мова на основі XML;
- у розпорядженні розроблювача перебуває великий набір вже готових елементів керування, при цьому створювати свої елементи керування дуже можна як на основі стандартних, так і з нуля, однаково це буде нескладно, що ідеально підходить для розробки інтерфейсу антифішингового програмного забезпечення;
- при створенні інтерфейсу не потрібно думати про те, як додаток буде виглядати в який-небудь ОС, ця опція передбачена при створенні мови, при використанні стандартних елементів керування можна також не турбуватися про те, як інтерфейс буде виглядати з різними темами;
- правильно спроектований інтерфейс легко локалізуємо, забезпечує можливість маштабування й швидкого виправлення помилок, які були допущені при проектуванні антифішингового інтерфейсу дипломного проекту.

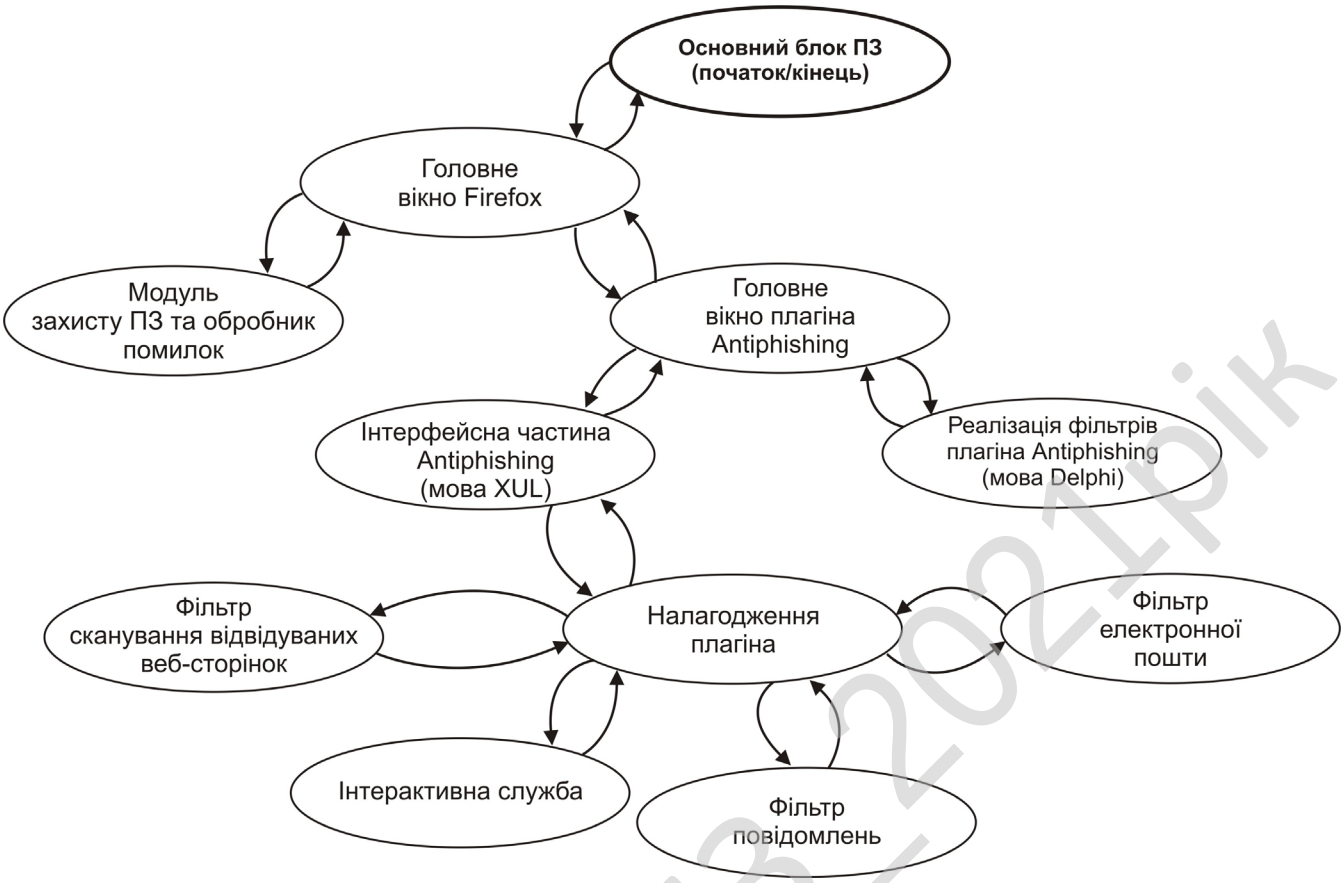


Рисунок 3.3 – Схема взаємодії процесів

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

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

4.1 Блок-схеми та опис алгоритмів функціонування системи

При складанні блок-схем програмного забезпечення і напрацювання алгоритмів взаємодії з модулями ПЗ я зіткнувся з масою проблем, які вимагали напрацювання процедур і функцій над основною проблематикою. Для чого були створені додаткові класи, типи даних і константи, що забезпечило вирішення проблем. При розробці програмного продукту я вирішував проблеми неправильної класифікації алгоритмів, що використовуються в існуючих розробках, а також нерозуміння або неправильного тлумачення методів їх реалізації, що приводить до чисельних проблем при просуванні продукту на ринок.

Розглянемо проведені першочергові дії по створенню дипломного програмного забезпечення.

Одним з найбільш корисних властивостей популярного браузера FireFox є система розширень (extentions). Завдяки розширенням, можливо одержати додаткову функціональність, відсутню в самому браузері в моєму випадку антифішингова зашита.

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

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

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

```
+ Antiphishing/
   +- chrome/
      +- content/
```

У будь-якому розширенні FireFox повинне бути як мінімум два файли – *install.rdf* і *contents.rdf*. У першому описано, як розширення буде встановлюватися в браузер, другий файл визначає структуру розширення.

Це прості текстові файли в XML-форматі, так що їх можна створювати в будь-якому текстовому редакторі. Але для зручності, я користувався спеціальним редакторам для xml-файлів (XMLSpy). Розширення rdf – це скорочення для "Resource Description Framework".

Вміст файлу *install.rdf* – цей файл повинен розташовуватися на самому верхньому рівні нашої антифішингової файлової ієрархії.

```
+- Antiphishing /
   +- install.rdf
   +- chrome/
      +- content/
```

У цьому файлі було описане наше розширення для менеджера розширень браузера FireFox, повідомивши його різну корисну інформацію- з якими версіями браузера сумісний додаток Antiphishing.

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

```
<em:maxVersion>1.0</em:maxVersion>
</Description>
</em:targetApplication>

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

  </Description>

</RDF>
```

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

```
<em:creator> Zavorotniy S.O. </em:creator>
<em:description>Antiphishing extention</em:description>
  <em:homepageURL>
         http://www.zavorotniy.narod.ru/</em:homepageURL>
  <em:id>{65b3130e-8513-41b6-8ea8-43dbd9cc0b12}</em:id>
  <em:name>Antiphishing</em:name>
  <em:version>1.0</em:version>>
```

Перший рядок указує на авторські права коду, для чого використовується тег `<em:creator>`. Наступний рядок – короткий опис того, а для чого, властиво, це розширення призначене. Ця інформація буде видна користувачам у менеджері розширень. Третій рядок указує на домашню сторінку розширення.

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

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

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

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

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

У моєму випадку ці два значення збігаються. Поточна версія FireFox дотримується наступної угоди по версіях:

major.minor.release.build[+]

(Поля, що випливають за *major* – необов'язкові)

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

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

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

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

Уміст contents.rdf – У цьому файлі описується, як моє розширення Антифішинг зберігається.

Це текстовий файл в XML-форматі й він повинен розташовуватися в директорії *content*. Таким чином, після створення цього файлу, я одержав наступну структуру:

```
+- Antiphishing/
   +- install.rdf
   +- chrome/
      +- content/
         +- contents.rdf
```

Зміст цього файлу наступний:

```
<?xml version="1.0"?>
<RDF:RDF xmlns:RDF="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:chrome="http://www.mozilla.org/rdf/chrome#">
  <RDF:Seq about="urn:mozilla:package:root">
    <RDF:li resource="urn:mozilla:package:Antiphishing"/>
  </RDF:Seq>

  <RDF:Description about="urn:mozilla:package:Antiphishing"
    chrome:extension="true" chrome:name="Antiphishing"/>

  <RDF:Seq about="urn:mozilla:overlays">
    <RDF:li resource="chrome://browser/content/browser.xul"/>
  </RDF:Seq>

  <RDF:Seq about="chrome://browser/content/browser.xul">
    <RDF:li>chrome://Antiphishing/content/Antiphishing-Overlay.xul</RDF:li>
  </RDF:Seq>

</RDF:RDF>
```

Перші строки:

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

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

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

У наступному фрагменті коду дипломного проекту вказується, який файл chrome наше розширення буде "розширювати". У більшості випадків

розширюється тільки файл *brouser.xul*, хоча є й і нші. У випадку розробленого антифішингового додатку все просто:

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

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

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

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

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

</overlay>
```

І далі продовжував створювати інтерфейс...

Після формування антифішинговогу інтерфейсу я одержав наступну структуру файлів:

```
+- Antiphishing /
   +- install.rdf
   +- chrome/
      +- content/
         +- contents.rdf
         +- Antiphishing-Overlay.xul
```

Зібрав в один пакет з розширенням XPI і установл у браузер FireFox, після правильної установки я одержав інтерфейс зображений на рисунку 4.1.

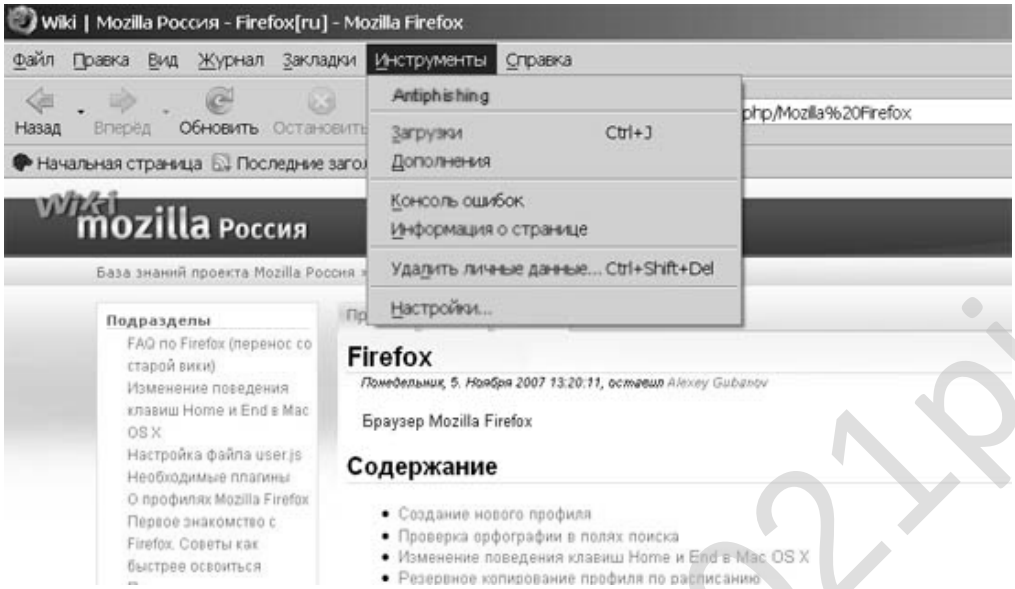


Рисунок 4.1 – Додавання пункту в меню інструментів FireFox

На рисунку 4.2 зображена основна блок-схема програми. При детальному розгляді програма розбита на дещо важливих блоків таких як:

– Блок ініціалізації початкових значень типів даних, констант, змінних, Firefox;

– Ініціалізація плагіна (antiphishing);

– Налагодження інсталяційного скрипту (antiphishing);

– Створення унікального ID та перевірка встановленої версії FF;

– Створення інтерфейсу плагіна;

– Виведення Toolbox;

– Перехід у робочий режим;

– Моніторинг WEB контенту;

– Підпрогама FL;

– Підпрограма IN;

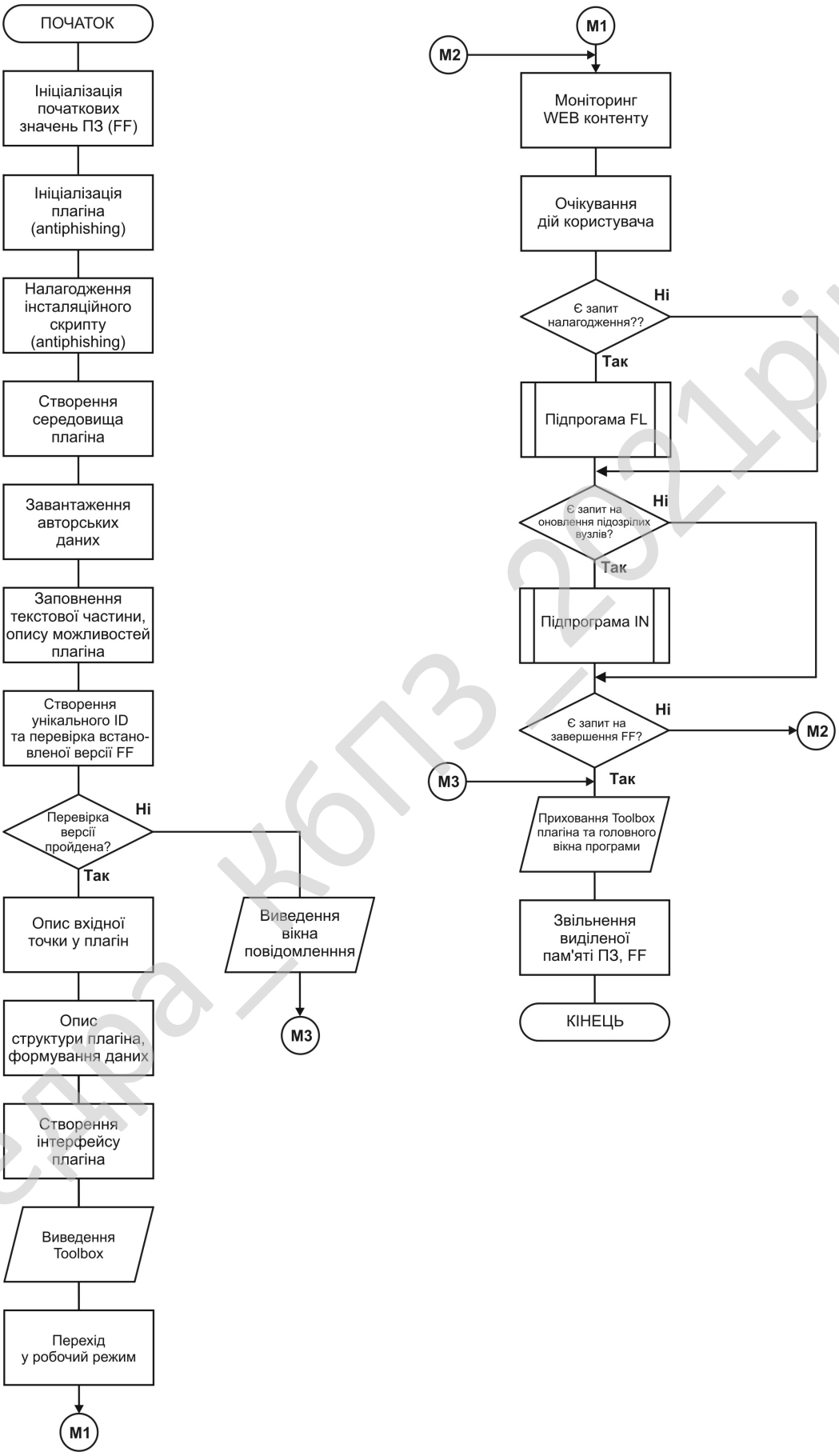
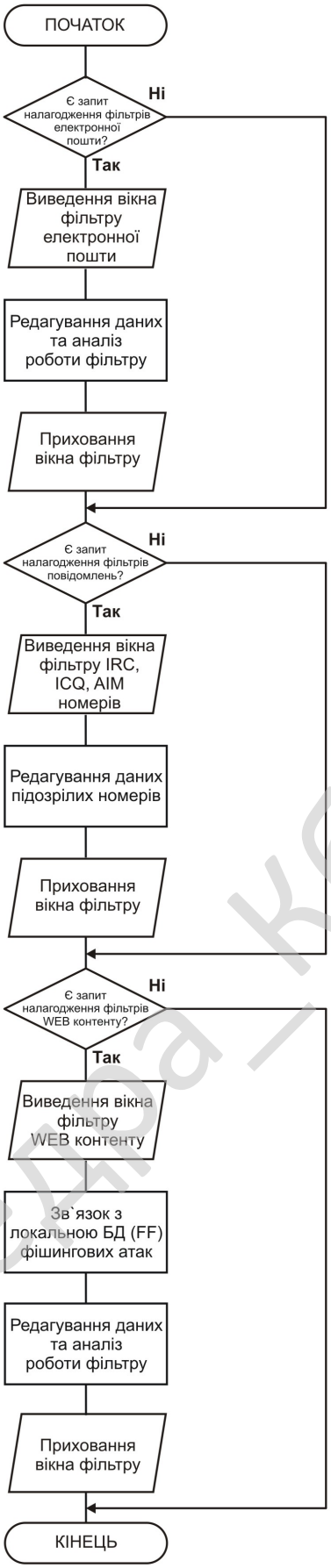


Рисунок 4.2 – Блок схема основного алгоритму програми

Підпрограма налагодження фільтрів “FL”



Підпрограма оновлення підозрілих вузлів “FL”

Рисунок 4.3 – Блок схема підпрограм

Блоки підпрограм відображені на рисунку 4.3. Підпрограма FL описує алгоритм налагодження програми, підпрограма IN оновлення списку підозрілих вузлів.

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*. 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 биткам, *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.

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Після написання програмного продукту його потрібно впровадити в промислову експлуатацію. При цьому виправляються помилки, які були помічені, система настроюється на відповідний режим роботи.

Для налаштування системи на оптимальні умови роботи та її наступне використання необхідно виконати наступні кроки:

1. Установити програмний продукт Mozilla Firefox.
2. Установити розроблене програмное забезпечення Antiphishing.
3.

Рисунок 5.1 – Запуск інсталляційного файлу

Потрібно призвести наступні дії – запустити Mozilla Firefox, через меню програми «Файл»-«Відкрити файл» запустити інсталяційний файл Antiphishing.xpi (рисунок 5.1)

Якщо інсталяційний файл розробленої антифішингового захисту цілий і правильно скомпанован, на екран виведеться вікно повідомлення установки програми (рисунок 5.2).

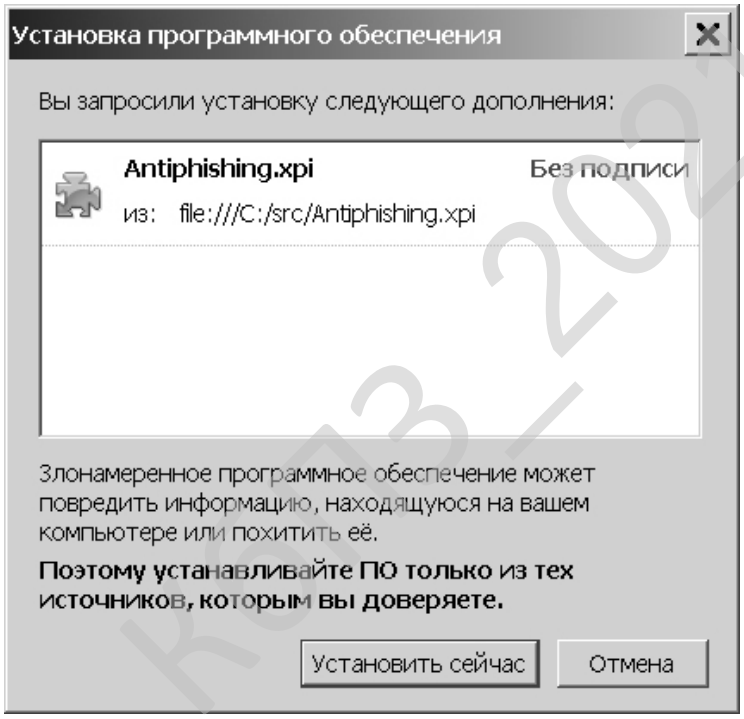


Рисунок 5.2 – Вспливаюче окно попередження

Після правильно проведених дій на екран виведеться вікно інсталяції із вказівкою дій, які проводяться. Після закінчення інсталяції необхідно перевантажити Mozilla Firefox (рисунок 5.3).

Після перезавантаження браузера на екран видається повідомлення про установку програмного забезпечення й готовності до роботи (рисунок 5.4).

Система забезпечує контроль над всіма основними можливостями які надаються браузером і Інтернетом, а саме:

– захист електронної пошти

– захист від фішинг-атак з використанням web-контента

– захист від фальсифікації рекламних банерів

– захист IRC, IRC, Viber, Telegram, WhatsApp, AIM-Повідомлень та завантаження троянських програм.

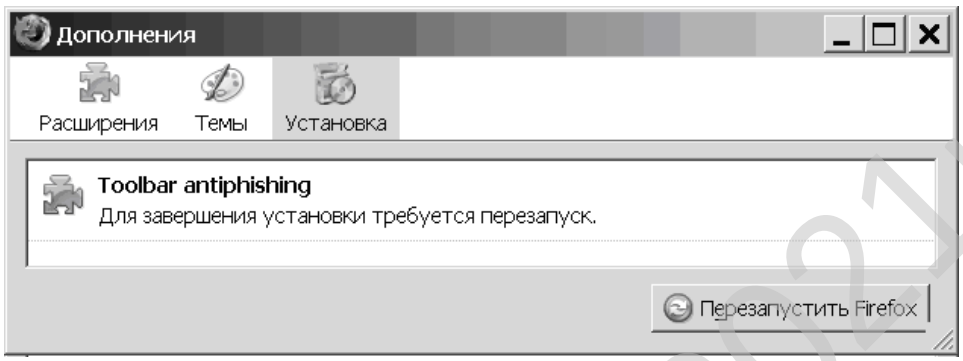


Рисунок 5.3 – Вимога перезавантаження браузера після інсталяції

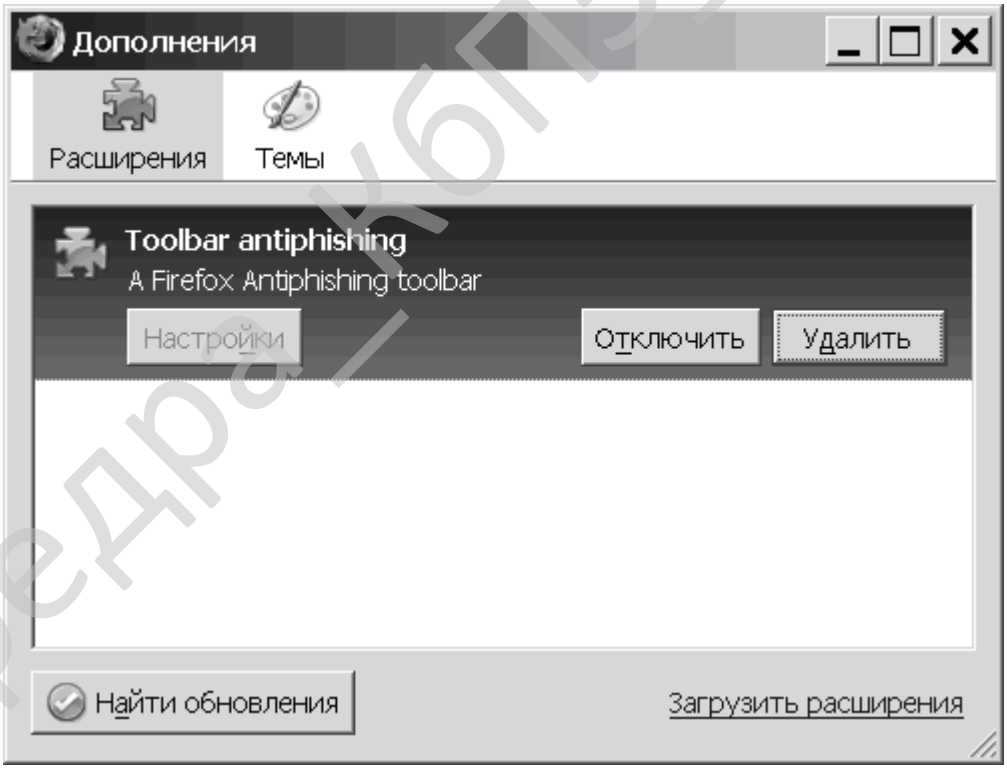


Рисунок 5.4 – Повідомлення про установку розробленого дипломного проекту – системи антифишингу

Цим самої значно зменшується ймовірність ушкодження операційної системи. На рисунку 5.5 показаний фільтр розробленої програми антифішингу. Вікно фільтра складається із двох частин – ліва частина містить список-фільтр шкідливих комбінацій і посилань на сайти з антифішинговими атаками, у правій частині вікна знаходяться кнопки управління фільтра.

Рисунок 5.5 – Фільтр вхідного контенту

За допомогою кнопок управління фільтра вхідного контенту можна робити наступні дії:

1. Додати – зробити ручне додавання шкідливих посилань і комбінацій для подальшого їхнього блокування.

4. Скачати нові шаблони – за допомогою безкоштовного Інтернет сервісу автоматично обновити список шкідливих посилань і комбінацій з доповненням нових.

5. Експортувати – зберегти список на локальний диск.

Так як вбудований Firefox пейджер Інтернет спілкування по протоколах Viber, Telegram, WhatsApp, AIM не має заборонних аркушів і захисту, розроблене вікно керування чорним списком антифішингових адрес і спам аркушів, дозволяє забезпечити систему від поширення шкідливих програм. За допомогою кнопок керування протоколами обміну Інтернет повідомленнями IRC, Viber, Telegram, WhatsApp, AIM можна робити наступні дії:

1. Додати з ігноруємого аркуша Viber, Telegram, WhatsApp – скачування игноруємих номерів із сервера Viber, Telegram, WhatsApp.

2. Видалити – видалити зі списку номер.

3. Додати – додати в ручну номер у список.

4. Експортувати – зберегти список на локальний диск.

5. Обнулити список – очистити список IRC, Viber, Telegram, WhatsApp, AIM номерів.

6. Сховати – сховати вікно програми передача керування Firefox.

На рисунку 5.5 показано форму авторського права

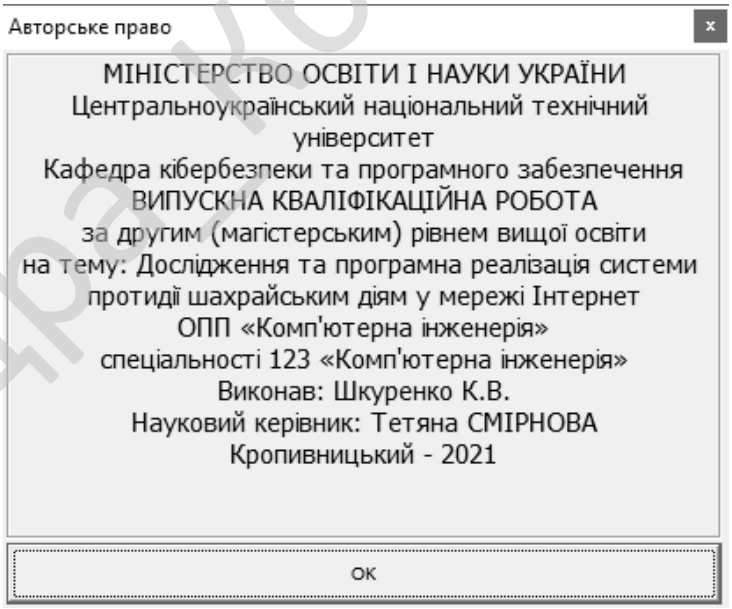


Рисунок 5.6 – Авторське право

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи протидії шахрайським діям у мережі Інтернет.

Метою розробки є дослідження та програмна реалізація системи протидії шахрайським діям у мережі Інтернет.

Об'єктом дослідження є процес протидії шахрайським діям у мережі Інтернет.

Предметом дослідження є методи протидії шахрайським діям у мережі Інтернет.

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 протидії шахрайським діям у мережі Інтернет.

– Розроблено вітчизняний продукт протидії шахрайським діям у мережі Інтернет, який має більш широкі можливості, на відміну від існуючих аналогів.

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 протидії шахрайським діям у мережі Інтернет.

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	100 (3 ост. цифри № зал)
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

Продовження таблиці 7.1

1	2	3
7. Кількість макетів вхідної інформації	–	3
8. Кількість форм вихідної інформації.	–	4
9. Мова програмування (1-6)	–	1
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

Продовження таблиці 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 ПЗ для декількох серверів одночасно (1-6)	–	2
31. Вимоги до дотримання встановленого графіка робіт (1-6)	–	2
32. Вартість ПЗ у розробника (НМА), грн.	–	100000 (3 ост. цифри № зал*10^3)
33. Норматив додаткової зарплати, % :	$Н_д$	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	50
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.:

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

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

$$B = 1{,}01 + 0{,}001\sum W_i, \qquad (7.2)$$

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

$$B = 1{,}01+0{,}001(2{,}43+3{,}64+3{,}38+3{,}95+2{,}73) = 1{,}027.$$

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

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

$$T_{уточн} = T_{ном}\, ПV_j, \qquad (7.3)$$

де: $ПV_j$ – добуток сімнадцяти додаткових коефіцієнтів, приведених в МВ додатку 3.

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

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:

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

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

$$T_{РП} = 0{,}3 \cdot 3{,}23 \cdot 9{,}37^{0{,}33+0{,}2(1{,}026-1{,}01)} \cdot 83 = 168 \text{ люд/день.}$$

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

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:

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

де: $F_{рq}$ – 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 облад-нання	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:			$З_ч$	227

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

Виходячи з цього фонд робочого часу інженерів-електронщиків складає:

$$Ф_{др}^{с} = \frac{З_ч \cdot n_{міс}}{1,2}, \quad (7.6)$$

$$Ф_{др}^{с} = \frac{227 \cdot 3}{1,2} = 567,5 \text{ год.}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

$$Ч_{ел} = \frac{Ф^{с}_{др}}{F_{др} \cdot T_{зм}}, \quad (7.7)$$

$Ч_{ел}$ = 567,5/(60·8) = 1,2 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	К-ть штатних одиниць
Адміністратор загальної мережі, аналітик	Адміністрування локальної мережі, поштового та серверу DNS (ОС FreeBSD), маршрутизатора Cisco, доменного контролеру Windows Server 2012 R2, серверу доступу ADSL (ОС Linux), налаштування ADSL, VPN, PPPoE, Frame Relay, Wi-Fi	2	0,5
	Налаштування і конфігурування базової станції безпровідного зв'язку (CMTS)	0,5	
	Розробка та впровадження проектів з організації зв'язку між віддаленими об'єктами, ЛОМ	0,5	
	Забезпечення цілодобової роботи зв'язку клієнтів до мережі Інтернет	1	
Всього		4	

Продовження таблиці 7.4

Посада	Вид роботи	Час	К-ть 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.

Таблиця 7.5 – Штатний розклад виконавців

Посада	Кількість ставок	Середньомісячний оклад, грн.	Всього за період розробки, грн.
Керівник (ІТ-менеджер)	1	6225	18675
Продакт-менеджер	0,25	6000	4500
Інженер-програміст	3,8	6000	68400
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	6000	4500
Інженер-верстальник	0,25	6000	4500
Бухгалтер-економіст	0,5	6000	9000
Всього за період розробки	$R_{сп}$ = 8,25	-	$Ф_{роб}$ = 149175

Розрахуємо середньоденну зарплату одного виконавця:

$$З_{сд} = \frac{Ф_{роб}}{R_{сп} F_{рд}}, \quad (7.8)$$

де: $Ф_{роб}$ – загальна сума зарплати за плановий період, грн.

$$З_{сд} = \frac{149175}{8,25 \cdot 60} = 301 \, грн.$$

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:

$$Б_{уд} = R^1_{сп} S_у Ц_{пл}, \quad (7.9)$$

де: $R^1_{сп}$ – кількість робочих місць виконавців, шт. Приймаємо 8 робочих місць;

$S_у$ – питома площа на одне робоче місце, $м^2$;

$Ц_{пл}$ – вартість одного квадратного метра площі, грн.

Згідно даних ТОВ науково-дослідницького консалтингового підприємства «Пекторaль» (м. Кіровоград) ціна одного квадратного метра площі новобудови, вік якої не перевищує 25 років, по місту складає 800…1600 $у.о./м^2$. Враховуючи, що курс складає 1 у.о. = 29 грн. приймаємо для розрахунку вартість одного метра квадратного рівною 29000 грн./м^2. На кожне робоче місце у середньому потрібно 8 $м^2$. З урахуванням цього:

$$Б_{уд} = 8 \cdot 8 \cdot 29000 = 1858000 \text{ грн.}$$

Вартість передавальних пристроїв складає 10% від вартості будівель, і у даному випадку вона складе: 185800 грн.

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

$$І_{нв} = R^1_{сп} \cdot Ц_м, \quad (7.10)$$

де: $Ц_м$ – ціна меблів для одного робочого місця, грн.

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

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 по прайсу фірми Комп'ютерторг за 20.10.21 – джерело http://computorg.ua/ru/price.html

Таблиця 7.6 – Специфікація

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Персональний комп’ютер		10947
Системний блок		7347
Процесор	Intel Core i7-870 / 4 (8) ядра по 2.93-3.6GHz 8Mb cache	1750
Системна плата	MACHINIST H55, MACHINIST H55, Inte H55, 1x PCI Expless x1; 1x PCI Express x16 4x SATA II (300MB/s), 6x USB; 1x HDMI; 1 VGA; 1x PS/2 keyboard; 1x PS/2 mouse; 1 RJ-45; 1x audio in\out	1200
Відеокарта	PCIeX: ATI HD SAPPHIR 512MB/128bit/DDR4/TV/DualDVI	750
Жорсткий диск	HDD: 640 Gb 7200 Serial ATA WD 32MB	1200
Оперативна пам’ять	4 GB DDR3 -2 модуля	900
DVD-привод	DVD -RW/+RW , LG SATA SuperMulti Bulk 22x, SecurDisc, black	416
Корпус	ATX Middle Tower FOXCONN Pro, 3GTLA 489, PSU 350W(FSP Brand: ATX-450PNR 12cm), black, (front bezel – black+light silver body material – 0.6mm), 80mm fan (rear 2xUSB2.0/AUDIO/MIC, Air Duct, Tool-les chassis design,Thermally Advantaged Chassis	911

Продовження таблиці 7.6

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Кардрідер внутрішній	USB 2.0 Card reader STORM CR-35U1A4-E int. 3.5", 1*USB2.0+AUDIO+1394, multi: Al Type Cards, black	220
інше	Клавіатура, мишка	Подару-нок
Монітор	22" TFT, ASUS VW223D (5ms, 300/3000: 1 170/160, D-SUB, Wide)	3600
Принтер лазерний	Canon i-SENSYS LBP6030W	2700
Принтер струминний	Epson Stylus Photo P50 (C11CA45341) + USB cable	5500
Копіювальний апарат	Canon i-SENSYS MF217W with Wi-Fi	5965

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 випробовування.	Загальна вартість, грн.
Персональні комп’ютери	15	10947	16420,5	180625,5
Принтер лаз.	2	2700	540	5940
Принтер струм.	1	5500	550	6050
Копіюв. апарат	1	5965	596,5	6561,5
Всього	–	–	–	199177

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

Таблиця 7.8 – Вартість основних фондів та амортизаційні відрахування розробника

Групи та види основних фондів	Балансова вартість, грн.	Амортизація	
		Норма, %	Відрахування, грн.
1	2	3	4
Група 3			
1. Будівлі	1858000	-	-
2. Передавальні пристрої	185800	-	-
Всього по групі	2043800	5	102190
Група 4			
3. Обчислювальна техніка	199177	-	-
Всього по групі	199177	50	99588,5
4. Нематеріальні активи	100000	10	10000
Група 5, 6			
5. Вимірювальні пристрої	9031	25	2257,75
6. Транспортні засоби	143000	20	28600
7. Господарський інвентар	28000	25	7000
Всього по групі	180031	-	37857,75
Разом	$К_р$ = 2523008		$А_р$ = 249636,25

Примітка: вартість автомобіля Sens (Standard+) взята по даним з автосалону «Кіровоград-Авто», джерело http://kirovograd-avto.ukravto.ua /catalog/tm-9/model-80/description, складає 143000 грн.

7.5 Визначення собівартості розробки та ціни програмної продукції

Визначимо основну зарплату виконавців:

$$З_о = \frac{З_{сд} \cdot Т_{пз}}{N_е}, \quad (7.11)$$

де: $N_е$ – кількість екземплярів програм, шт.

$$З_о = 301 \cdot 209/100 = 629 \text{ грн.}$$

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

$$З_д = З_о \cdot Н_q \cdot 0{,}01, \quad (7.12)$$

де: $Н_q$ – норматив додаткової зарплати, %.

$$З_д = 629 \cdot 10 \cdot 0{,}01 = 63 \text{ грн.}$$

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

$$С_{оц} = 0{,}01 \cdot Н_с (З_о + З_д), \quad (7.13)$$

де: $Н_с$ – відрахування на соціальні потреби, %.

$$С_{оц} = 0{,}01 \cdot 37(629+63) = 256 \text{ грн.}$$

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

$$Г_{осп} = З_о \cdot Н_г \cdot 0{,}01, \quad (7.14)$$

де: $Н_г$ – загальногосподарські витрати, %.

$$Г_{осп} = 629 \cdot 15 \cdot 0{,}01 = 94 \text{ грн.}$$

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

$$З_М = (З_{М1} + З_{М2} + З_{М3})/N_е, \quad (7.15)$$

де: $З_{М1}$ – вартість паперу, грн.; $З_{М2}$ – вартість запам'ятовуючих пристроїв, грн.; $З_{М3}$ – вартість фарби, картріджей, тонеру, грн.; $N_е$ – кількість екземплярів програм, шт.

Згідно виданих викладачем норм приймаємо одну пачку паперу на три місяці розробки. Тоді, враховуючи, що вартість пачки паперу складає $Ц_п$ = 103 грн., визначаємо вартість паперу за період розробки $N_м$ = 3 міс:

$$З_{М1} = Ц_п \cdot N_м. \quad (7.16)$$

$$З_{М1} = 103 \cdot 3 = 309 \text{ грн.}$$

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

$$З_{М2} = \sum Ц_д \,, \quad (7.17)$$

де: $Ц_д$ – 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.

$$З_{М2} = 12 \cdot 100 + 15 = 1215 \text{ грн.}$$

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

$$З_{М3} = \sum Ц_з., \quad (7.18)$$

де: $Ц_з$ – 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.

$$З_{М3} = 574 + 558 + 570 = 1702 \text{ грн.}$$

$$З_М = (309 + 1215 + 1702)/100 = 32 \text{ грн.}$$

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

$$О_п = З_о \cdot Н_п \cdot 0{,}01, \quad (7.19)$$

де: $Н_п$ – норматив витрат на освоєння нових мов програмування, %.

$$О_п = 629 \cdot 15 \cdot 0{,}01 = 94 \text{ грн.}$$

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ($N_е$ = 100 прим.):

$$A_м = \frac{A_р \cdot N_{міс}}{N_е \cdot 12}, \quad (7.20)$$

де: $A_р$ – загальна річна сума амортизаційних відрахувань, грн.

$$A_м = 249636 \cdot 3/(100 \cdot 12) = 624 \text{ грн.}$$

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	2	3
1. Основна зарплата виконавців	$З_о$	629
2. Додаткова зарплата виконавців	$З_д$	63
3. Відрахування на соціальні потреби	$С_{оц}$	256
4. Загальногосподарські витрати	$Г_{осп}$	94
5. Витрати на матеріали	$З_М$	32
6. Освоєння нових операційних систем, мов програмування	$О_п$	94
7. Амортизація основних фондів	$А_м$	624
8. Повна собівартість програмного забезпечення	$С_п$	1792
9. Плановий прибуток	$П_р$	896
10.Ціна підприємства $Ц_п = С_п + П_р$	$Ц_п$	2688
11.Податок на додану вартість $ПДВ = 0.01 \cdot Н_{дв} \cdot Ц_п$	$ПДВ$	537,6
12.Відпускна ціна програмної продукції $Ц = Ц_п + ПДВ$	$Ц$	3225,6

Повна собівартість ПЗ визначається як сума витрат за попередніми статтями калькуляції:

$$C_п = З_о + З_д + С_{оц} + Г_{осп} + З_м + О_п + А_м. \quad (7.21)$$

$$C_п = 629+63+256+94+32+94+624 = 1792 \text{ грн.}$$

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

Для даного програмного забезпечення рівень рентабельності складає 50%.

$$П_р = 0{,}01 \cdot P_п \cdot C_п, \quad (7.22)$$

де: $P_п$ – рівень рентабельності, %.

$$П_р = 0{,}01 \cdot 50 \cdot 1792 = 896 \text{ грн.}$$

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 пусконалагоджувальних 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	–	3226
Всього капітальних витрат	–	3226

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 технічне обслуговування)	$З_р$	15190	3165
2. Витрати на електроенергію	$З_{ел}$	218	151
3. Витрати на амортизацію	$З_{ам}$	0	807
Всього витрат за рік	$І$	15408	4123

Витрати на профілактичні роботи:

$$З_р = Т_р \cdot З_г \cdot (1 + 0{,}01 \cdot Н_q) \cdot (1 + 0{,}01 \cdot Н_c), \qquad (7.23)$$

де: $Т_р$ – кількість годин обслуговування кожного комп'ютера за рік, *год.*;

$З_г$ – заробітна плата обслуговуючого персоналу, *грн / год*.

Після купівлі нового програмного забезпечення кількість профілактичних годин робіт зменшилася з 240 годин на рік до 50 годин на рік, тому витрати на технічне обслуговування зменшилися з:

$$З_{р\,баз} = 240 \cdot 6 \cdot 1{,}1 \cdot 1{,}37 \cdot 7 = 15190 \text{ грн},$$

до:

$$З_{р\,нов} = 50 \cdot 6 \cdot 1{,}1 \cdot 1{,}37 \cdot 7 = 3165 \text{ грн.}$$

Витрати на електроенергію визначаються з урахуванням споживаємої потужності ($П_{ел}$) в кіловатах, часу експлуатації технічних засобів ($Т_р$) в годинах та ціни однієї кіловат-години ($Ц_{ел}$):

$$З_{ел} = П_{ел} \cdot Т_р \cdot Ц_{ел}. \qquad (7.24)$$

$$З_{ел\,баз} = 7 \cdot 0{,}15 \cdot 130 \cdot 1{,}6 = 218 \text{ грн.}$$

$$З_{ел\,нов} = 7 \cdot 0{,}15 \cdot 90 \cdot 1{,}6 = 151 \text{ грн.}$$

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	–	3226	–	806,5
Всього відрахувань	-	–	3226	–	806,5

7.8 Визначення економічної ефективності програмної продукції

Економічна ефективність програмного забезпечення визначається для виготовлювача і споживача за такими показниками.

Величина економічного ефекту при виготовленні програмної продукції, розраховуємо за формулою:

$$E_в = (Ц_п - C_п) \cdot N_е - \sum_{i=1}^{m} E_{p_m} \cdot K_{p_m}, \quad (7.25)$$

де: $K_р$ – балансова вартість основних фондів розробника, грн.; $E_р$ – розрахунковий коефіцієнт капіталовкладень.

$$Ев = (2688\text{-}1792) \cdot 100\text{-}(0{,}05 \cdot 2043800 + 0{,}4 \cdot 199177 + 0{,}25 \cdot 37031 +$$
$$+0{,}1 \cdot 100000 + 0{,}2 \cdot 143000) \cdot 3/12 = 32170 \; грн.$$

Визначимо період окупності додаткових капітальних вкладень у виробника програмної продукції:

$$T_в = \frac{K_p^*}{(Ц_п - C_п) \cdot N_е}, \quad (7.26)$$

де: K_p^* – балансова вартість основних фондів розробника без врахування вартості ОФ третьої групи, так як їх строк служби на порядок більший ніж період розробки ПЗ.

$$T_в = \frac{479208}{(2688\text{-}1792)\cdot 100\cdot 12/3} = 1{,}3 \text{ років}.$$

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.	100
2. Повна собівартість розробленої програми	Грн.	1792
3. Ціна розробленої програми	Грн.	2688
4. Плановий прибуток від реалізації розробленої програми	Грн.	896
5. Рентабельність програмної продукції	%	50
6. Об’єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	2523008
7. Загальний прибуток від реалізації програмної продукції	Грн.	89600
8. Величина економічного ефекту при виготовлені програмної продукції	Грн.	32170
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Років	1,3
10.Об’єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	3226
11.Величина економічного ефекту у користувача програмної продукції	Грн.	10478
12.Період окупності додаткових капітальних вкладень у користувача програмної продукції	Років	0,3

Визначимо величину економічного ефекту у користувача програмної продукції за формулою:

$$E_{сп} = (I_б - I_н) - E_н(K_н - K_б), \quad (7.27)$$

де: $I_б$, $I_н$ – величина експлуатаційних витрат за базовим и новим варіантом відповідно; $K_б$, $K_н$ – об'єм капітальних вкладень за варіантами, що порівнюються.

$$E_{сп} = (15408\text{-}4123)\text{-}0{,}25 \cdot 3226 = 10478 \text{ грн.}$$

Визначимо період окупності додаткових капітальних вкладень у споживача програмної продукції за рахунок зниження експлуатаційних витрат:

$$T_{сп} = \frac{K_н - K_б}{I_б - I_н}, \quad (7.28)$$

$$T_{сп} = \frac{3226}{15408 - 4123} = 0{,}3 \text{ року}.$$

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.

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Протягом усієї історії людство приділяє прискіпливу увагу безпеці життя [3]. Охорона праці є складовою частиною безпеки життя.

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» [4], яким затверджено нормативно-правовий акт з охорони праці НПАОП 0.00-7.15-18, «Правила охорони праці під час експлуатації електронно-обчислювальних машин», та «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98 [1].

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 іс тотне навантаженням. 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 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 [1], та «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» НПАОП 0.00-7.15-18.

Умови праці програміста включають наступні фактори:

– параметри повітряного середовища в приміщенні;

– 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

Програміст працює з електронно-обчислювальною машиною (ЕОМ) та іншим обладнанням, яке є джерелом небезпеки ураження електричним струмом. Так як робота програміста характеризується істотним зоровим навантаженням, то вимагає належного освітлення. Так як програміст постійно перебуває в приміщенні, тому для комфортних умов праці в цьому приміщенні необхідно створити належний мікроклімат.

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;

– недостатня, або надмірна освітленість робочого місця;

– монотонність праці;

– електромагнітні (у т.ч. високочастотні) електромагнітні випромінювання (коливання);

– несприятливі мікрокліматичні умови;

– нервово-емоційна напруженість праці;

– інтелектуальні навантаження;

– невідповідність ергономічних показників робочого місця діючим вимогам;

– шуми;

– статичні навантаження на кістково-м’язовий апарат.

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.

Таблиця 8.1 – Розміри приміщення

Найменування	Значення, м
Ширина	8
Довжина	8,2
Висота	3,1

Таблиця 8.2 – Площа та обсяг приміщення, на одного працюючого*

Геометрична характеристика	Одиниця виміру	Нормативне значення*	Фактичне значення
Площа, S	$м^2$	не менше 6.0	6,56
Обсяг, V	$м^3$	не менше 20.0	20,34

* Згідно ДСанПіН 3.3.2.007-98 (Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин) [1].

У зазначеному приміщенні працює 10 осіб. 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 відповідають нормативним вимогам (Наказу Міністерства соціальної політики України № 207, від 14.02.2018 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями», ДСанПіН 3.3.2-007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» [1] та НПАОП 0.00 -1.28-10 «Правила охорони праці під час експлуатації електронно-обчислювальних машин»).

Температура повітря в приміщенні визначається впливом температури зовнішнього повітря і тепловою енергією, яка виділяється всередині приміщення. Джерелами виділення теплоти в даному приміщенні є електроустаткування, освітлювальні прилади, а також люди. У світлий час доби джерелом надлишкового тепла є сонячна радіація. Згідно Постанови Головного державного санітарного лікаря України [5], робота, яка виконується в даному приміщенні, відноситься до категорії Іа. В цьому випадку людина витрачає енергії до 120 ккал у годину. Вологість повітря у приміщенні визначається 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.

Таблиця 8.3 – Оптимальні і фактичні значення параметрів мікроклімату

Пора року	Оптимальні для Іа			Фактичні		
	Температура, °С	Воло-гість,%	Швидкість повітря, м/с	Температура, °С	Воло-гість%	Швидкість повітря, м/с
Холодна	22-24	40-60	0,1	22-23	40-55	0,1
Тепла	23-25	50-70	0,1	23-24	52-70	0,11

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

В літню пору застосовується кондиціонер.

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

У приміщенні знаходяться наступні джерела шуму: принтер HP Laser 107a, 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 пропозиція буде така: дотримання позитивної психологічної атмосфери в колективі та регламентованого режиму праці та відпочинку, організація робочого місця з урахуванням ергономічних вимог.

Рекомендовані заходи: регулярні періодичні наочні огляди персоналом шляхів для евакуації людей із приміщення, відповідно до плану евакуації (який повинен розташовуватись на видному місці у приміщенні), включення до колективного договору мінімально можливого вмісту аптечок з обов’язковою наявністю масок-клапанів, або іншого спорядження для штучного дихання. Регулярна періодична перевірка параметрів заземлення та занулення (вимірювання опору ланцюга).

8.5 Розрахункова частина

Проведемо розрахунок штучного освітлення за методом коефіцієнта використання світлового потоку для приміщення ширина якого складає 8 м, довжина – 8,2 м, висота – 3,1 м.

Для того, щоб визначити потрібну кількість світильників, які повинні забезпечити нормований рівень освітленості, визначимо світловий потік, що падає на робочу поверхню за формулою:

$$F=ESKZ/n,$$

де:

F – світловий потік, що розраховується, Лм;

E – нормована мінімальна освітленість, Лк; E = 300 Лк;

S – площа освітлюваного приміщення (у нашому випадку S=8 х 8,2 = 65,6 м^2);

K – коефіцієнт запасу, що враховує зменшення світлового потоку лампи в результаті забруднення світильників в процесі експлуатації (його значення залежить від типу приміщення і характеру робіт, що проводяться в ньому, в нашому випадку K = 1,5 [5]);

Z – відношення середньої освітленості до мінімальної (зазвичай приймається рівним 1.1... 1.2, в нашому випадку Z = 1,1);

n – коефіцієнт використання світлового потоку, (відношення світлового потоку, 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% [5].

Обчислимо індекс приміщення за формулою:

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

де:

S – площа приміщення, S = 65,6 м^2;

h – розрахункова висота підвісу, h = 3,1 м (співпадає з висотою стелі, т.я.

лампи освітлення закріплюються на стелі);

A – ширина приміщення, A = 8 м;

B – довжина приміщення, B = 8,2 м.

Підставимо всі значення у формулу та визначемо індекса приміщення:

$$i=1,3.$$

Знаючи індекс приміщення (i), за знаходимо n = 0,5 (з табличних даних коефіцієнтів використання світлового потоку (n) світильників з відповідним типом ламп [5]). Підставимо всі значення у формулу, визначемо світловий потік: F=64944 Лм.

Для штучного освітлення приміщення використовуються *LED* світильники *PANEL-B2B-595*, світловий потік яких $F_л$ = 3500 Лм.

Число світильників визначається по формулі:

$$N=F/F_л$$

де:

F – світловий потік,

$F_л$ – світловий потік однієї лампи.

Підставимо всі значення у формулу та визначемо індекса приміщення: N= 64944/ 3500=18,55 шт.

Для забезпечення нормованої мінімальної освітленості приймаємо необхідну кількість світильників 19 шт.

8.6 Висновки до розділу

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

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.

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи протидії шахрайським діям у мережі Інтернет.

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 протидії шахрайським діям у мережі Інтернет.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем протидії шахрайським діям у мережі Інтернет.

– Досліджена система протидії шахрайським діям у мережі Інтернет.

– На основі отриманих результатів досліджень створена програмна реалізація системи протидії шахрайським діям у мережі Інтернет.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання протидії шахрайським діям у мережі Інтернет.

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.

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

Програма реалізована на мові високого рівня Delphi. 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 10478 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 0,3 роки.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Шкуренко К.В. Дослідження та програмна реалізація системи протидії шахрайським діям у мережі Інтернет // Збірник праць молодих науковців ЦНТУ. – Вип. 12. – Кропивницький: ЦНТУ, 2022.

2. http://www.anti-phishing.org

3. 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.

4. 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.

5. Смирнов С.А. Метод безопасной маршрутизации метаданных в облачные антивирусные системы / А.К. Дидык, С.А. Смирнов // Информационные технологии в управлении, образовании, науке и промышленности: монография / Под редакцией профессора В.С. Пономаренко. – Х.: Видавець Рожко С.Г., 2016. – 566 с.

6. 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.

7. 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.

8. 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.

9. 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.

10. 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.

11. Смирнов С. А. Анализ и исследование методов управления сетевыми ресурсами для обеспечения антивирусной защиты даннях / 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.

12. 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.

13. 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.

14. 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.

15. 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.

16. 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.

17. 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.

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

19. Смирнов С. А. Исследование математических моделей технологии распространения компьютерных виру сов / 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.

20. 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.

21. 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.

22. 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.

23. 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.

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

25. 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.

26. Смирнов С. А. Метод управления доступом к облачным телекоммуни-кационным ресурсам для обеспечения защиты даннях / 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.

27. 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.

28. 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.

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

30. Смирнов С. А. gert-модели технологии облачной антивирусной защиты / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Безпека 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.

31. 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.

32. 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.

33. 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.

34. 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.

35. 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.

36. 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 «ХПІ», 2016. – С. 14.

37. 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.

38. 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.

39. 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 ІІІ міжнар. наук.-практ. конф., м. Харків, 28-30 кві. 2016 р. – Х.: ХННІ ДВНЗ «УБС», 2016. – С. 178-182.

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

41. 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.

42. 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.

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

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

45. Телекоммуникационные системы и сети: учебное пособие. В 3 томах / [В.В. Величко, Е.А. Субботин, В.П. Шувалов, А.Ф. Ярославцев]; под ред. В.П. Шувалова. – М.: Горячая линия-Телеком, 2005, т. 3 – 592 с.

46. Хайкин С. Нейронные сети: полный курс / С. Хайкин. – М.:Вильямс, 2006. – 1103 с.

47. Шелухин О.И. Фрактальные процессы в телекоммуникациях: моногр. / О.И. Шелухин, А.М. Тенякшев, А.В. Осин – М.: Радиотехника, 2003. – 480 с.

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

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

50. Сакулин В.П., Шептовицкий В.М. Безопасность труда при монтаже и эксплуатации электроустановок / В.П.Сакулин, В.М.Шептовицкий. – Л. : “Колос”, 1973. – 238 с.

51. Охорона праці. Ч. 1. Захисне заземлення: метод. вказ. до викон. розрахунків з викор. персон. ЕОМ ІВМ сумісного типу / Кіровоград. ін-т с.-г. машинобуд.; [укл. О. В. Оришака, Є. К. Солови х, В. О. Оришака]. – Кіровоград: КІСМ, 1997. – 20 с. Режим доступу до ресурсу: http://dspace.kntu.kr.ua/jspui/handle/123456789/4358

Додаток А
(обов'язковий)

Технічне завдання

Зміст

					ВКРМ-123.21.0100.00.00.ТЗ		
Вим.	Арк.	№ документа	Підпис	Дата			
Розробив		Шкуренко К.В.			Дослідження та програмна реалізація системи протидії шахрайським діям у мережі Інтернет	Літ. М	Аркуш 1 / Аркушів 6
Перевірив		Смірнова Т.В.					
Н. Контр.		Гермак В.С.				ЦНТУ КІ-20МЗ	
Затв.		Смірнов О.А.					

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи протидії шахрайським діям у мережі Інтернет.

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. № 41-13 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 протидії шахрайським діям у мережі Інтернет.

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;

– 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 протидії шахрайським діям у мережі Інтернет;

– 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.

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 Delphi, XUL.

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 повинна бути розглянута розробка заходів з умов поліпшення охорони праці.

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 – 95 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.

Додаток Б

(обов'язковий)

Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

Керівник випускної кваліфікаційної роботи за

другим (магістерським) рівнем вищої освіти

_____ Смірнова Т.В.

Дослідження та програмна реалізація

системи протидії шахрайським діям у мережі Інтернет

Лістинг програми

Код документу 12

Носій: CD/DVD-диск

Загальна кількість аркушів: 48

Літера: РП

Кропивницький – 2021 року

Головний файл взаємодії з FireFox install.rdf

(*..rdf - Resource Description Framework)

```
<?xml version="1.0"?>

<RDF xmlns="http://www.w3.org/2021/09/22-rdf-syntax-
ns#"xmlns:em="http://www.mozilla.org/2021/em-rdf#">

<Description about="urn:mozilla:install-manifest">
<em:creator> Shkurenko K.V. </em:creator>
<em:description>Antifraud extention</em:description>
<em:homepageURL>http://www.shkurenko.kntu.kr.ua/</em:homepageURL>
  <em:id>{65b3130e-8513-41b6-8ea8-43dbd9cc0b12}</em:id>
  <em:name>Antifraud</em:name>
  <em:version>1.0</em:version>
  <em:targetApplication>
  <Description>
  <em:id>{ec8030f7-c20a-464f-9b0e-13a3a9e97384}</em:id>
  <em:minVersion>1.0</em:minVersion>
  <em:maxVersion>1.0</em:maxVersion>
  </Description>
  </em:targetApplication>

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

Опис знаходження файлів Delphi contents.rdf

(*..rdf - Resource Description Framework)

```
<?xml version="1.0"?>

<RDF:RDF xmlns:RDF="http://www.w3.org/2021/09/22-rdf-syntax-ns#"
     xmlns:chrome="http://www.mozilla.org/rdf/chrome#">

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

  <RDF:Description about="urn:mozilla:package:Antifraud"
    chrome:extension="true" chrome:name="Antifraud"/>

  <RDF:Seq about="urn:mozilla:overlays">
    <RDF:li resource="chrome://browser/content/browser.xul"/>
  </RDF:Seq>

  <RDF:Seq about="chrome://browser/content/browser.xul">
<RDF:li>chrome://Antifraud/content/Antifraud-Overlay.xul</RDF:li>
  </RDF:Seq>
 </RDF:RDF>
```

Створення розділів меню Antifraud-Overlay.xul

(*.xul User-interface Language, опис інтерфейсу користувача)

```
  <?xml version="1.0"?>
  <overlay id="AntifraudOverlay"
 xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">
  <menupopup id="menu_ToolsPopup">
  <menuitem label="Antifraud" position="1" />
  <menuitem label=" About" position="2"/>
  </menupopup>
<?xml version="1.0"?>
<?xml-stylesheet href="chrome://global/skin/" type="text/css"?>
<?xml-stylesheet href="findfile.css" type="text/css"?>
<!ENTITY findWindow.title "Find Files">
<!ENTITY fileMenu.label "File">
<!ENTITY editMenu.label "Edit">
<!ENTITY fileMenu.accesskey "f">
<!ENTITY editMenu.accesskey "e">
<!ENTITY openCmd.label "Open Search...">
<!ENTITY saveCmd.label "Save Search...">
<!ENTITY closeCmd.label "Close">
<!ENTITY openCmd.accesskey "o">
<!ENTITY saveCmd.accesskey "s">
<!ENTITY closeCmd.accesskey "c">
<!ENTITY cutCmd.label "Cut">
<!ENTITY copyCmd.label "Copy">
<!ENTITY pasteCmd.label "Paste">
<!ENTITY cutCmd.accesskey "t">
<!ENTITY copyCmd.accesskey "c">
<!ENTITY pasteCmd.accesskey "p">
<!ENTITY cutCmd.commandkey "X">
<!ENTITY copyCmd.commandkey "C">
<!ENTITY pasteCmd.commandkey "V">
<!ENTITY openCmdToolbar.label "Open">
<!ENTITY saveCmdToolbar.label "Save">
<!ENTITY searchTab "Search">
<!ENTITY optionsTab "Options">
<!ENTITY findDescription "Enter your search criteria below and select the Find
button to begin the search.">
<!ENTITY findCriteria "Search Criteria">
<!ENTITY type.name "Name">
<!ENTITY type.size "Size">
<!ENTITY type.date "Date Modified">
<!ENTITY mode.is "Is">
<!ENTITY mode.isnot "Is Not">
<!ENTITY casesensitive "Case Sensitive Search">
```

```
<!ENTITY matchfilename "Match Entire Filename">
<!ENTITY results.filename "Filename">
<!ENTITY results.location "Location">
<!ENTITY results.size "Size">
<!ENTITY bytes.before "">
<!ENTITY bytes.after "bytes">
<!ENTITY button.find "Find">
<!ENTITY button.cancel "Cancel">
<window
  id="findfile-window"
  title="&findWindow.title;"
  persist="screenX screenY width height"
  orient="horizontal"
onload="initSearchList()"xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">
<script src="findfile.js"/>
<popupset>
   <popup id="editpopup">
     <menuitem label="Cut" accesskey="&cutCmd.accesskey;"/>
     <menuitem label="Copy" accesskey="&copyCmd.accesskey;"/>
<menuitem label="Paste"  accesskey="&pasteCmd.accesskey;"disabled="true"/>
   </popup>
</popupset>
<keyset>
   <key id="cut_cmd" modifiers="accel" key="&cutCmd.commandkey;"/>
   <key id="copy_cmd" modifiers="accel" key="&copyCmd.commandkey;"/>
   <key id="paste_cmd" modifiers="accel" key="&pasteCmd.commandkey;"/>
   <key id="close_cmd" keycode="VK_ESCAPE" oncommand="window.close();"/>
</keyset>

<vbox flex="1">

 <toolbox>

  <menubar id="findfiles-menubar">
   <menu id="file-menu" label="&fileMenu.label;"
       accesskey="&fileMenu.accesskey;">
     <menupopup id="file-popup">
       <menuitem label="&openCmd.label;"
                 accesskey="&openCmd.accesskey;"/>
       <menuitem label="&saveCmd.label;"
                 accesskey="&saveCmd.accesskey;"/>
       <menuseparator/>
       <menuitem label="&closeCmd.label;"

accesskey="&closeCmd.accesskey;" key="close_cmd" oncommand="window.close();"/>
```

```
      </menupopup>
    </menu>
    <menu id="edit-menu" label="&editMenu.label;"
          accesskey="&editMenu.accesskey;">
      <menupopup id="edit-popup">
        <menuitem label="&cutCmd.label;"
                  accesskey="&cutCmd.accesskey;" key="cut_cmd"/>
        <menuitem label="&copyCmd.label;"
                  accesskey="&copyCmd.accesskey;" key="copy_cmd"/>
        <menuitem label="&pasteCmd.label;"
                  accesskey="&pasteCmd.accesskey;" key="paste_cmd"
disabled="true"/>
      </menupopup>
    </menu>
  </menubar>
  <toolbar id="findfiles-toolbar">
    <toolbarbutton id="opensearch" label="&openCmdToolbar.label;"/>
    <toolbarbutton id="savesearch" label="&saveCmdToolbar.label;"/>
  </toolbar>
 </toolbox>
 <tabbox>
  <tabs>
    <tab label="&searchTab;" selected="true"/>
    <tab label="&optionsTab;"/>
  </tabs>
  <tabpanels>

   <tabpanel id="searchpanel" orient="vertical" context="editpopup">
   <description>
     &findDescription;
   </description>

   <spacer class="titlespace"/>

   <groupbox orient="horizontal">
     <caption label="&findCriteria;"/>
     <menulist id="searchtype">
       <menupopup>
         <menuitem label="&type.name;"/>
         <menuitem label="&type.size;"/>
         <menuitem label="&type.date;"/>
       </menupopup>
     </menulist>
   <spacer class="springspace"/>
     <menulist id="searchmode">
       <menupopup>
```

```
          <menuitem label="&mode.is;"/>
          <menuitem label="&mode.isnot;"/>
        </menupopup>
      </menulist>
    <spacer class="springspace"/>
<menulist id="find-text" flex="1"
editable="true"
datasources="file:///mozilla/recents.rdf">
      <template>
        <menupopup>
        </menupopup>
      </template>
    </menulist>
    </groupbox>
  </tabpanel>
  <tabpanel id="optionspanel" orient="vertical">
      <checkbox id="casecheck" label="&casesensitive;"/>
      <checkbox id="wordscheck" label="&matchfilename;"/>
     </tabpanel>
  </tabpanels>
 </tabbox>
 <tree id="results" style="display: none;" flex="1">
   <treecols>
     <treecol id="name" label="&results.filename;" flex="1"/>
     <treecol id="location" label="&results.location;" flex="2"/>
     <treecol id="size" label="&results.size;" flex="1"/>
   </treecols>

   <treechildren>
     <treeitem>
       <treerow>
         <treecell label="mozilla"/>
         <treecell label="/usr/local"/>
         <treecell label="&bytes.before;2520&bytes.after;"/>
       </treerow>
     </treeitem>
   </treechildren>
 </tree>
 <splitter id="splitbar" resizeafter="grow" style="display: none;"/>
 <spacer class="titlespace"/>
 <hbox>
<progressmeter id="progmeter" value="50%" style="display: none;"/>
   <spacer flex="1"/>
   <button id="find-button" label="&button.find;"
           oncommand="doFind()"/>
   <button id="cancel-button" label="&button.cancel;"
```

```
            oncommand="window.close();"/>
    </hbox>
    </vbox>
    </window>
```

Головний файл prAddFilter.dpr

```
program prAddFilter;

uses
  Forms,
  frmIdGlobal  in 'IdGlobal.pas' {Form1},
  frmFILTER_ALL in 'frmFILTER_ALL.pas' {Form2},
  frmICQ in 'frmICQ.pas' {Form3};
  frmfrmForm1 in 'frmForm1.pas' {Form4};
  frmfrmForm2 in 'frmForm2.pas' {Form5};
{$R *.res}

begin
  Application.Initialize;
  Application.CreateForm(TForm1, Form1);
  Application.CreateForm(TForm2, Form2);
  Application.CreateForm(TForm2, Form3);
  Application.CreateForm(TForm4, Form4);
  Application.CreateForm(TForm5, Form5);
  Application.Run;
end.
```

Файл системи захисту (Antifphishing)

```
unit IdGlobal;

interface

uses
Windows,
Classes,
IdException,
SyncObjs, SysUtils;

const
IdTimeoutDefault = -1;
IdTimeoutInfinite = -2;

IdFetchDelimDefault = ' ';
IdFetchDeleteDefault = true;
IdFetchCaseSensitiveDefault = true;

LWS = [TAB, CHAR32];
wdays: array[1..7] of string = ('Sun', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri'
, 'Sat');
monthnames: array[1..12] of string = ('Jan', 'Feb', 'Mar', 'Apr', 'May'
, 'Jun',  'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec');
IdHexDigits: array [0..15] of Char = '0123456789ABCDEF';

GPathDelim = '/';
INFINITE = LongWord($FFFFFFFF);
tpIdle = 19;
tpLowest = 12;
tpLower = 6;
tpNormal = 0;
tpHigher = -7;
tpHighest = -13;
tpTimeCritical = -20;

GPathDelim = '\';
GOSType = otWindows;
type
THandle = LongWord;
TIdThreadPriority = -20..19;
THandle = Windows.THandle;
TIdThreadPriority = TThreadPriority;
TIdMaxLineAction = (maException, maSplit);
TIdReadLnFunction = function: string of object;
```

```
TStringEvent = procedure(ASender: TComponent; const AString: String);
TPosProc = function(const Substr, S: string): Integer;
TIdReuseSocket = (rsOSDependent, rsTrue, rsFalse);

TIdCardinalBytes = record
 case Integer of
 0: (
   Byte1: Byte;
   Byte2: Byte;
   Byte3: Byte;
   Byte4: Byte;);
 1: (Whole: Cardinal);
 2: (CharArray : array[0..3] of Char);
end;

TIdLocalEvent = class(TEvent)
public
 constructor Create(const AInitialState: Boolean = False;
  const AManualReset: Boolean = False); reintroduce;
 function WaitFor: TWaitResult; overload;
end;

TIdGlobalMimeTable = class(TObject)
protected
 FOnBuildCache: TNotifyEvent;
 FMIMEList: TStringList;
 FFileExt: TStringList;
 procedure BuildDefaultCache; virtual;
public
 procedure BuildCache; virtual;
 procedure AddMimeType(const Ext, MIMEType: string);
 function GetFileMIMEType(const AFileName: string): string;
 function GetDefaultFileExt(Const MIMEType: string): string;
 procedure LoadFromStrings(AStrings: TStrings; const MimeSeparator: Char =
'=');
 procedure SaveToStrings(AStrings: TStrings; const MimeSeparator: Char =
'=');
 constructor Create(Autofill: boolean=true); virtual;
 destructor Destroy; override;
 property  OnBuildCache: TNotifyEvent read FOnBuildCache write FOnBuildCache;
end;

TIdReadMemoryStream = class (TCustomMemoryStream)
public
 procedure SetPointer(Ptr: Pointer; Size: Longint);
 function Write(const Buffer; Count: Longint): Longint; override;
```

```
 End;

 TIdCharSet = (csGB2312, csBig5, csIso2022jp, csEucKR, csIso88591);

 PByte =^Byte;
 PWord =^Word;
 TIdPID = Integer;
 TIdPID = LongWord;
 TIdWin32Type = (Win32s, WindowsNT40, Windows95, Windows95OSR2,
 Windows98, Windows98SE,Windows2000, WindowsMe, WindowsXP);
 EIdFailedToRetreiveTimeZoneInfo = class(EIdException);
 EIdCorruptServicesFile = class(EIdException);
 EIdExtensionAlreadyExists = class(EIdException);
 function  AnsiMemoryPos(const ASubStr: String; MemBuff: PChar; MemorySize:
Integer): Integer;
 function  AnsiPosIdx(const ASubStr,AStr: AnsiString; AStartPos: Cardinal=0):
Cardinal;
 function  AnsiSameText(const S1, S2: string): Boolean;
 procedure FreeAndNil(var Obj);
 function GetFileCreationTime(const Filename: string): TDateTime;
 function BreakApart(BaseString, BreakString: string; StringList: TStrings):
TStrings;
 procedure CommaSeparatedToStringList(AList: TStrings; const Value:string);
 function CopyFileTo(const Source, Destination: string): Boolean;
 function CurrentProcessId: TIdPID;
 function DateTimeToGmtOffSetStr(ADateTime: TDateTime; SubGMT: Boolean):
string;
 Function DateTimeToInternetStr(const Value: TDateTime; const AIsGMT : Boolean
= False) : String;
 procedure DebugOutput(const AText: string);
 function DomainName(const AHost: String): String;
 function FileSizeByName(const AFilename: string): Int64;
 function GetMIMETypeFromFile(const AFile: TFileName): string;
 function GetSystemLocale: TIdCharSet;
 function GetThreadHandle(AThread: TThread): THandle;
 function GetTickCount: Cardinal;
 function iif(ATest: Boolean; const ATrue: Boolean; const AFalse: Boolean):
Boolean; overload;
 function IncludeTrailingSlash(const APath: string): string;
 function IsDomain(const S: String): Boolean;
 function IsFQDN(const S: String): Boolean;
 function IsHostname(const S: String): Boolean;
 function IsNumeric(AChar: Char): Boolean; overload;
 function IsNumeric(const AString: string): Boolean; overload;
 function IsTopDomain(const AStr: string): Boolean;
 function IsValidIP(const S: String): Boolean;
```

```
 function InMainThread: boolean;
 function Max(AValueOne,AValueTwo: Integer): Integer;
 function MakeMethod (DataSelf, Code: Pointer): TMethod;
 function MakeTempFilename(const APath: String = ''): string;
 function Min(AValueOne, AValueTwo: Integer): Integer;
 function RightStr(const AStr: String; Len: Integer): String;
 function ROL(AVal: LongWord; AShift: Byte): LongWord;
 function ROR(AVal: LongWord; AShift: Byte): LongWord;
 function SetLocalTime(Value: TDateTime): boolean;
 procedure Sleep(ATime: cardinal);
 function StrToCard(const AStr: String): Cardinal;
 function StrInternetToDateTime(Value: string): TDateTime;
 function StrToDay(const ADay: string): Byte;
 function StrToMonth(const AMonth: string): Byte;
 function MemoryPos(const ASubStr: String; MemBuff: PChar; MemorySize:
Integer): Integer;
 function TimeZoneBias: TDateTime;
 function UpCaseFirst(const AStr: string): string;
 function Win32Type : TIdWin32Type;

 var
 FilterPos: TPosProc = nil;
 GOffsetFromUTC: TDateTime = 0;
 GSystemLocale: TIdCharSet = csIso88591;
 GTimeZoneBias: TDateTime = 0;

 FilterFalseBoolStrs : array of String;
 FilterTrueBoolStrs : array of String;

 implementation

 uses
 Libc,
 IdStack,
 IdStackWindows,
 Registry,
 IdStack, IdResourceStrings, IdURI;

 const
 WhiteSpace = [#0..#12, #14..' '];
 var
 FIdPorts: TList;
 ATempPath: string;

 function Win32Type: TIdWin32Type;
 begin
```

```
if Win32MajorVersion >= 5 then begin
 if Win32MinorVersion >= 1 then begin
   Result := WindowsXP;
 end
 else begin
   Result := Windows2000;
 end;
end
else begin
  if Win32MajorVersion > 3 then begin
   if Win32Platform = VER_PLATFORM_WIN32_NT then begin
     Result := WindowsNT40;
   end
   else begin
     Win32BuildNumber := Win32BuildNumber and $FFFF;
     if Win32MinorVersion >= 90 then begin
       Result := WindowsMe;
     end
     else begin
       if Win32MinorVersion >= 10 then begin
         {Windows 98}
         if Win32BuildNumber >= 2222 then begin
           Result := Windows98SE
         end
         else begin
           Result := Windows98;
         end;
       end
       else begin {Windows 95}
         if Win32BuildNumber >= 1000 then begin
           Result := Windows95OSR2
         end
         else begin
           Result := Windows95;
         end;
       end;
     end;
   end;//if VER_PLATFORM_WIN32_NT
 end
 else begin
   Result := Win32s;
 end;
end;//if Win32MajorVersion >= 5
end;

function GetThreadHandle(AThread : TThread) : THandle;
```

```
begin
Result := AThread.ThreadID;
Result := AThread.Handle;
end;

function RawStrInternetToDateTime(var Value: string): TDateTime;
var
i: Integer;
Dt, Mo, Yr, Ho, Min, Sec: Word;
sTime: String;
ADelim: string;

Procedure ParseDayOfMonth;
begin
 Dt :=  StrToIntDef( Fetch(Value, ADelim), 1);
 Value := TrimLeft(Value);
end;

Procedure ParseMonth;
begin
 Mo := StrToMonth( Fetch ( Value, ADelim )  );
 Value := TrimLeft(Value);
end;
begin
Result := 0.0;
Value := Trim(Value);
if Length(Value) = 0 then begin
 Exit;
end;

try
 if StrToDay(Copy(Value, 1, 3)) > 0 then begin
   Fetch(Value);
   Value := TrimLeft(Value);
 end;

 if (FilterPos('-', Value) > 1) and (FilterPos('-', Value) < FilterPos(' ',
Value)) then begin
   ADelim := '-';
 end
 else begin
   ADelim := ' ';
 end;
 if (StrToMonth(Fetch(Value, ADelim,False)) > 0) then
 begin
   {Month}
```

```
    ParseMonth;
    {Day of Month}
    ParseDayOfMonth;
  end
  else
  begin
    {Day of Month}
    ParseDayOfMonth;
    {Month}
    ParseMonth;
  end;
  {Year}

  sTime := Fetch(Value);
  Yr := StrToIntDef(sTime, 1900);
  if Yr = 1900 then begin
    Yr := StrToIntDef(Value, 1900);
    Value := sTime;
  end;
  if Yr < 80 then begin
    Inc(Yr, 2000);
  end else if Yr < 100 then begin
    Inc(Yr, 1900);
  end;

  Result := EncodeDate(Yr, Mo, Dt);
  i := FilterPos(':', Value);
  if i > 0 then begin
    sTime := fetch(Value, ' ');
    {Hour}
    Ho  := StrToIntDef( Fetch ( sTime,':'), 0);
    {Minute}
    Min := StrToIntDef( Fetch ( sTime,':'), 0);
    {Second}
    Sec := StrToIntDef( Fetch ( sTime ), 0);
    {The date and time stamp returned}
    Result := Result + EncodeTime(Ho, Min, Sec, 0);
  end;
  Value := TrimLeft(Value);
except
  Result := 0.0;
end;
end;

function IncludeTrailingSlash(const APath: string): string;
begin
```

```
Result := IncludeTrailingBackSlash(APath);
 Result :=  IncludeTrailingPathDelimiter(APath);
 Result := APath;
 if not IsPathDelimiter(Result, Length(Result)) then begin
   Result := Result + GPathDelim;
 end;
end;

function AnsiSameText(const S1, S2: string): Boolean;
begin
Result := CompareString(LOCALE_USER_DEFAULT, NORM_IGNORECASE, PChar(S1)
, Length(S1), PChar(S2), Length(S2)) = 2;
end;

procedure FreeAndNil(var Obj);
var
P: TObject;
begin
if TObject(Obj) <> nil then begin
 P := TObject(Obj);
 TObject(Obj) := nil;  // clear the reference before destroying the object
 P.Free;
end;
end;

function CreateTRegistry: TRegistry;
begin
 Result := TRegistry.Create;
end;
function CreateTRegistry: TRegistry;
begin
 Result := TRegistry.Create(KEY_READ);
end;

function Max(AValueOne,AValueTwo: Integer): Integer;
begin
if AValueOne < AValueTwo then
begin
 Result := AValueTwo
end //if AValueOne < AValueTwo then
else
begin
 Result := AValueOne;
end; //else..if AValueOne < AValueTwo then
end;
```

```
 function Min(AValueOne, AValueTwo : Integer): Integer;
 begin
 If AValueOne > AValueTwo then
 begin
  Result := AValueTwo
 end //If AValueOne > AValueTwo then
 else
 begin
  Result := AValueOne;
 end;
 end;
function DateTimeToInternetStr(const Value: TDateTime; const AIsGMT : Boolean =
False) : String;
 var
 wDay,
 wMonth,
 wYear: Word;
 begin
 DecodeDate(Value, wYear, wMonth, wDay);
 Result := Format('%s, %d %s %d %s %s',
                 [wdays[DayOfWeek(Value)], wDay, monthnames[wMonth],
                  wYear, FormatDateTime('HH":"NN":"SS', Value),
                  DateTimeToGmtOffSetStr(OffsetFromUTC, AIsGMT)]);
 end;

 function StrInternetToDateTime(Value: string): TDateTime;
 begin
 Result := RawStrInternetToDateTime(Value);
 end;


 function GetInternetFormattedFileTimeStamp(const AFilename: String):String;
 const
 wdays: array[1..7] of string = ('Sun', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri',
'Sat');
 monthnames: array[1..12] of string = ('Jan', 'Feb', 'Mar', 'Apr', 'May',
 'Jun',
  'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec');
 var
 DT1, DT2 : TDateTime;
 wDay, wMonth, wYear: Word;
 begin
 DT1 := GetFileCreationTime(AFilename);
 DecodeDate(DT1, wYear, wMonth, wDay);
 DT2 := TimeZoneBias;
```

```
 Result := Format('%s, %d %s %d %s %s', [wdays[DayOfWeek(DT1)], wDay,
monthnames[wMonth],
 wYear, FormatDateTime('HH":"NN":"SS', DT1),
DateTimeToGmtOffSetStr(DT2,False)]);
 end;

 function GetFileCreationTime(const Filename: string): TDateTime;
 var
 Data: TWin32FindData;
 H: THandle;
 FT: TFileTime;
 I: Integer;
 begin
 H := FindFirstFile(PCHAR(Filename), Data);
 if H <> INVALID_HANDLE_VALUE then begin
  try
    FileTimeToLocalFileTime(Data.ftLastWriteTime, FT);
    FileTimeToDosDateTime(FT, LongRec(I).Hi, LongRec(I).Lo);
    Result := FileDateToDateTime(I);
  finally
    Windows.FindClose(H);
  end
 end else begin
  Result := 0;
 end;
 end;


 function BreakApart(BaseString, BreakString: string; StringList: TStrings):
TStrings;
 var
 EndOfCurrentString: integer;
 begin
 repeat
  EndOfCurrentString := Pos(BreakString, BaseString);
  if (EndOfCurrentString = 0) then
  begin
    StringList.add(BaseString);
  end
  else
    StringList.add(Copy(BaseString, 1, EndOfCurrentString - 1));
  delete(BaseString, 1, EndOfCurrentString + Length(BreakString) - 1);
  Copy(BaseString, EndOfCurrentString + length(BreakString),
length(BaseString) - EndOfCurrentString);
 until EndOfCurrentString = 0;
 result := StringList;
```

```
end;

procedure CommaSeparatedToStringList(AList: TStrings; const Value:string);
var
iStart,
iEnd,
iQuote,
iPos,
iLength : integer ;
sTemp : string ;
begin
iQuote := 0;
iPos := 1 ;
iLength := Length(Value) ;
AList.Clear ;
while (iPos <= iLength) do
begin
 iStart := iPos ;
 iEnd := iStart ;
 while ( iPos <= iLength ) do
 begin
   if Value[iPos] = '"' then
   begin
     inc(iQuote);
   end;
   if Value[iPos] = ',' then
   begin
     if iQuote <> 1 then
     begin
       break;
     end;
   end;
   inc(iEnd);
   inc(iPos);
 end ;
 sTemp := Trim(Copy(Value, iStart, iEnd - iStart));
 if Length(sTemp) > 0 then
 begin
   AList.Add(sTemp);
 end;
 iPos := iEnd + 1 ;
 iQuote := 0 ;
end ;
end;

function CopyFileTo(const Source, Destination: string): Boolean;
```

```
var
SourceStream: TFileStream;
begin
Result := false;
if not FileExists(Destination) then begin
 SourceStream := TFileStream.Create(Source, fmOpenRead); try
   with TFileStream.Create(Destination, fmCreate) do try
     CopyFrom(SourceStream, 0);
   finally Free; end;
 finally SourceStream.free; end;
 Result := true;
end;
end;

function CopyFileTo(const Source, Destination: string): Boolean;
begin
Result := CopyFile(PChar(Source), PChar(Destination), true);
end;

function TempPath: string;
var
     i: integer;
begin
SetLength(Result, MAX_PATH);
     i := GetTempPath(Length(Result), PChar(Result));
     SetLength(Result, i);
IncludeTrailingSlash(Result);
end;

function MakeTempFilename(const APath: String = ''): string;
Begin
Result := tempnam(nil, 'Filter');
SetLength(Result, MAX_PATH + 1);
if APath > '' then begin
 GetTempFileName(PChar(IncludeTrailingSlash(APath)), 'Filter', 0,
PChar(Result));
end
else begin
 GetTempFileName(PChar(ATempPath), 'Filter', 0, PChar(Result));
end;
Result := PChar(Result);
End;

function RPos(const ASub, AIn: String; AStart: Integer = -1): Integer;
var
i: Integer;
```

```
LStartPos: Integer;
LTokenLen: Integer;
begin
result := 0;
LTokenLen := Length(ASub);
// Get starting position
if AStart = -1 then begin
 AStart := Length(AIn);
end;
if AStart < (Length(AIn) - LTokenLen + 1) then begin
 LStartPos := AStart;
end else begin
 LStartPos := (Length(AIn) - LTokenLen + 1);
end;
// Search for the string
for i := LStartPos downto 1 do begin
 if AnsiSameText(Copy(AIn, i, LTokenLen), ASub) then begin
   result := i;
   break;
 end;
end;
end;

function GetSystemLocale: TIdCharSet;
begin
Result := GSystemLocale;
case SysLocale.PriLangID of
 LANG_CHINESE:
   if SysLocale.SubLangID = SUBLANG_CHINESE_SIMPLIFIED then
     Result := csGB2312
   else
     Result := csBig5;
 LANG_JAPANESE: Result := csIso2022jp;
 LANG_KOREAN: Result := csEucKR;
 else
  Result := csIso88591;
end;
end;

function FileSizeByName(const AFilename: string): Int64;
begin
with TFileStream.Create(AFilename, fmOpenRead or fmShareDenyNone) do
try
 Result := Size;
finally Free; end;
end;
```

```
Function RightStr(const AStr: String; Len: Integer): String;
var
LStrLen : Integer;
begin
LStrLen := Length (AStr);
if (Len > LStrLen) or (Len < 0) then begin
 Result := AStr;
end  //f ( Len > Length ( st ) ) or ( Len < 0 ) then
else begin
 Result := Copy(AStr, LStrLen - Len+1, Len);
end;
end;

function OffsetFromUTC: TDateTime;
begin
Result := GOffsetFromUTC;
end;
function OffsetFromUTC: TDateTime;
var
iBias: Integer;
tmez: TTimeZoneInformation;
begin
Case GetTimeZoneInformation(tmez) of
 TIME_ZONE_ID_INVALID:
   raise EIdFailedToRetreiveTimeZoneInfo.Create(RSFailedTimeZoneInfo);
 TIME_ZONE_ID_UNKNOWN  :
    iBias := tmez.Bias;
 TIME_ZONE_ID_DAYLIGHT :
   iBias := tmez.Bias + tmez.DaylightBias;
 TIME_ZONE_ID_STANDARD :
   iBias := tmez.Bias + tmez.StandardBias;
 else
   raise EIdFailedToRetreiveTimeZoneInfo.Create(RSFailedTimeZoneInfo);
end;
if iBias > 0 then begin
 Result := 0 - Result;
end;
end;

function StrToCard(const AStr: String): Cardinal;
begin
Result := StrToInt64Def(Trim(AStr),0);
end;

function TimeZoneBias: TDateTime;
```

```
  begin
  Result := GTimeZoneBias;
  end;

  function TimeZoneBias: TDateTime;
  var
  ATimeZone: TTimeZoneInformation;
  begin
  case GetTimeZoneInformation(ATimeZone) of
   TIME_ZONE_ID_DAYLIGHT:
  Result := ATimeZone.Bias + ATimeZone.DaylightBias;
  TIME_ZONE_ID_STANDARD:
  Result := ATimeZone.Bias + ATimeZone.StandardBias;
  TIME_ZONE_ID_UNKNOWN:
     Result := ATimeZone.Bias;
   else
     raise EIdException.Create(SysErrorMessage(GetLastError));
  end;
  Result := Result / 1440;
  end;

  function GetTickCount: Cardinal;
  var
  tv: timeval;
  begin
  gettimeofday(tv, nil);

  Result := int64(tv.tv_sec) * 1000 + tv.tv_usec div 1000;
  end;

function GetTickCount: Cardinal;
  begin
  Result := Windows.GetTickCount;
  end;

 function GetTickDiff(const AOldTickCount, ANewTickCount : Cardinal):Cardinal;
  begin
   if ANewTickCount >= AOldTickCount then begin
     Result := ANewTickCount - AOldTickCount;
   end else begin
     Result := High(Cardinal) - AOldTickCount + ANewTickCount;
   end;
  end;

  function FilterStrToBool(const AString : String) : Boolean;
  var
```

```
LCount : Integer;
begin
for LCount := Low(FilterFalseBoolStrs) to High(FilterFalseBoolStrs) do
begin
 if AnsiSameText(AString, FilterFalseBoolStrs[LCount]) then
 begin
   result := false;
   exit;
 end;
end;

for LCount := Low(FilterTrueBoolStrs) to High(FilterTrueBoolStrs) do
begin
 if AnsiSameText(AString, FilterTrueBoolStrs[LCount]) then
 begin
   result := true;
   exit;
 end;
end;
LCount := StrToInt(AString);
if LCount = 0 then
begin
 result := false;
end else
begin
 result := true;
end;
end;

function SetLocalTime(Value: TDateTime): boolean;
begin
result := False;
end;


function SetLocalTime(Value: TDateTime): boolean;
var
dSysTime: TSystemTime;
buffer: DWord;
tkp, tpko: TTokenPrivileges;
hToken: THandle;
begin
Result := False;
if SysUtils.Win32Platform = VER_PLATFORM_WIN32_NT then
begin
```

```
  if not Windows.OpenProcessToken(GetCurrentProcess(), TOKEN_ADJUST_PRIVILEGES
or TOKEN_QUERY,
    hToken) then
  begin
    exit;
  end;
  Windows.LookupPrivilegeValue(nil, 'SE_SYSTEMTIME_NAME',
tkp.Privileges[0].Luid);
  tkp.PrivilegeCount := 1;
  tkp.Privileges[0].Attributes := SE_PRIVILEGE_ENABLED;
  if not Windows.AdjustTokenPrivileges(hToken, FALSE, tkp, sizeof(tkp), tpko,
buffer) then
  begin
    exit;
  end;
 end;
 DateTimeToSystemTime(Value, dSysTime);
 Result := Windows.SetLocalTime(dSysTime);
 if SysUtils.Win32Platform = VER_PLATFORM_WIN32_NT then
 begin
  Windows.AdjustTokenPrivileges(hToken, FALSE,tpko, sizeOf(tpko), tkp,
Buffer);
  Windows.CloseHandle(hToken);
 end;
 end;

 function IdPorts: TList;
 var
 sLocation, s: String;
 idx, i, iPrev, iPosSlash: integer;
 sl: TStringList;
 begin
 if FIdPorts = nil then
 begin
  FIdPorts := TList.Create;
  SetLength(sLocation, MAX_PATH);
  SetLength(sLocation, GetWindowsDirectory(pchar(sLocation), MAX_PATH));
  sLocation := IncludeTrailingSlash(sLocation);
  if Win32Platform = VER_PLATFORM_WIN32_NT then begin
    sLocation := sLocation + 'system32\drivers\etc\';
  end;
  sl := TStringList.Create;
  try
    sl.LoadFromFile(sLocation + 'services');
    iPrev := 0;
    for idx := 0 to sl.Count - 1 do
```

```
      begin
        s := sl[idx];
        iPosSlash := FilterPos('/', s);
if (iPosSlash > 0) and (not (FilterPos('#', s) in [1..iPosSlash])) then
        begin
          i := iPosSlash;
          repeat
            dec(i);
            if i = 0 then begin
              raise EIdCorruptServicesFile.CreateFmt(RSCorruptServicesFile,
 [sLocation + 'services']);
            end;
          until s[i] in WhiteSpace;
          i := StrToInt(Copy(s, i+1, iPosSlash-i-1));
          if i <> iPrev then begin
            FIdPorts.Add(TObject(i));
          end;
          iPrev := i;
        end;
      end;
    finally
      sl.Free;
    end;
  end;
  Result := FIdPorts;
  end;

  function FetchCaseInsensitive(var AInput: string; const ADelim: string =
 IdFetchDelimDefault;
  const ADelete: Boolean = IdFetchDeleteDefault): String;
  var
  LPos: integer;
  begin
  if ADelim = #0 then begin
   LPos := Pos(ADelim, AInput);
  end else begin
   LPos := FilterPos(UpperCase(ADelim), UpperCase(AInput));
  end;
  if LPos = 0 then begin
   Result := AInput;
   if ADelete then begin
     AInput := '';
   end;
  end else begin
   Result := Copy(AInput, 1, LPos - 1);
   if ADelete then begin
```

```
    AInput := Copy(AInput, LPos + Length(ADelim), MaxInt);
   end;
  end;
  end;

  function Fetch(var AInput: string; const ADelim: string =
IdFetchDelimDefault;
  const ADelete: Boolean = IdFetchDeleteDefault;
  const ACaseSensitive: Boolean = IdFetchCaseSensitiveDefault): String;
  var
  LPos: integer;
  begin
  if ACaseSensitive then begin
   if ADelim = #0 then begin
     LPos := Pos(ADelim, AInput);
   end else begin
     LPos := FilterPos(ADelim, AInput);
   end;
   if LPos = 0 then begin
     Result := AInput;
     if ADelete then begin
       AInput := '';
     end;
   end
   else begin
     Result := Copy(AInput, 1, LPos - 1);
     if ADelete then begin
       //slower Delete(AInput, 1, LPos + Length(ADelim) - 1);
       AInput:=Copy(AInput, LPos + Length(ADelim), MaxInt);
     end;
   end;
  end else begin
   Result := FetchCaseInsensitive(AInput, ADelim, ADelete);
  end;
  end;

  begin
  for Result := Low(Contents) to High(Contents) do begin
   if CaseSensitive then begin
     if SearchStr = Contents[Result] then begin
       Exit;
     end;
   end else begin
     if ANSISameText(SearchStr, Contents[Result]) then begin
       Exit;
     end;
```

```
 end;
end;
Result := Low(Contents) to High(Contents) do
end;

function IsCurrentThread(AThread: TThread): boolean;
begin
result := AThread.ThreadID = GetCurrentThreadID;
end;

function IsNumeric(AChar: char): Boolean;
begin
Result := AChar in ['0'..'9'];
end;

function IsNumeric(const AString: string): Boolean;
var
LCode: Integer;
LVoid: Integer;
begin
Val(AString, LVoid, LCode);
Result := LCode = 0;
end;

function StrToDay(const ADay: string): Byte;
begin
Result := Succ(PosInStrArray(Uppercase(ADay),
 ['SUN','MON','TUE','WED','THU','FRI','SAT']));
end;

function StrToMonth(const AMonth: string): Byte;
begin
Result := Succ(PosInStrArray(Uppercase(AMonth),
 ['JAN','FEB','MAR','APR','MAY','JUN','JUL','AUG','SEP','OCT','NOV','DEC']));
end;

function UpCaseFirst(const AStr: string): string;
begin
Result := LowerCase(TrimLeft(AStr));
if Result <> '' then begin
 Result[1] := UpCase(Result[1]);
end;
end;

function DateTimeToGmtOffSetStr(ADateTime: TDateTime; SubGMT: Boolean):
string;
```

```
 var
 AHour, AMin, ASec, AMSec: Word;
 begin
 if (ADateTime = 0.0) and SubGMT then
 begin
  Result := 'GMT';
  Exit;
 end;
 DecodeTime(ADateTime, AHour, AMin, ASec, AMSec);
 Result := Format(' %0.2d%0.2d', [AHour, AMin]);
 if ADateTime < 0.0 then
 begin
  Result[1] := '-';
 end
 else
 begin
  Result[1] := '+';
 end;
 end;

 procedure BuildMIMETypeMap(dest: TStringList);
 begin
 raise EIdException.Create('BuildMIMETypeMap not implemented yet.');
 end;
 var
 Reg: TRegistry;
 slSubKeys: TStringList;
 i: integer;
 begin
 Reg := CreateTRegistry; try
  Reg.RootKey := HKEY_CLASSES_ROOT;
  Reg.OpenKeyreadOnly('\MIME\Database\Content Type');
  slSubKeys := TStringList.Create;
  try
    Reg.GetKeyNames(slSubKeys);
    reg.Closekey;
    for i := 0 to slSubKeys.Count - 1 do
    begin
Reg.OpenKeyreadOnly('\MIME\Database\Content Type\' + slSubKeys[i]);
      dest.Append(LowerCase(reg.ReadString('Extension')) + '=' +
slSubKeys[i]);
      Reg.CloseKey;
    end;
  finally
    slSubKeys.Free;
  end;
```

```
 finally
  reg.free;
 end;
 end;

 function GetMIMETypeFromFile(const AFile: TFileName): string;
 var
 MIMEMap: TIdMIMETable;
 begin
 MIMEMap := TIdMimeTable.Create(true);
 try
  result := MIMEMap.GetFileMIMEType(AFile);
 finally
  MIMEMap.Free;
 end;
 end;

 function GmtOffsetStrToDateTime(S: string): TDateTime;
 begin
 Result := 0.0;
 S := Copy(Trim(s), 1, 5);
 if Length(S) > 0 then
 begin
  if s[1] in ['-', '+'] then
  begin
    try
      Result := EncodeTime(StrToInt(Copy(s, 2, 2)), StrToInt(Copy(s, 4, 2)),
0, 0);
      if s[1] = '-' then
      begin
        Result := -Result;
      end;
    except
      Result := 0.0;
    end;
  end;
 end;
 end;

 function GMTToLocalDateTime(S: string): TDateTime;
 var
 DateTimeOffset: TDateTime;
 begin
 Result := RawStrInternetToDateTime(S);
 if Length(S) < 5 then begin
  DateTimeOffset := 0.0
```

```
end else begin
 DateTimeOffset := GmtOffsetStrToDateTime(S);
end;
if DateTimeOffset < 0.0 then begin
 Result := Result + Abs(DateTimeOffset);
end else begin
 Result := Result - DateTimeOffset;
end;
// Apply local offset
Result := Result + OffSetFromUTC;
end;


procedure Sleep(ATime: cardinal);
begin
if (not Assigned(GStack)) then begin
 GStack := TIdStack.CreateStack;
end;
GStack.WSSelect(nil, nil, nil, ATime);
Windows.Sleep(ATime);
end;

var
i: Integer;
begin
SetLength(result, 32);
for i := 1 to 32 do
begin
 if ((Value shl (i-1)) shr 31) = 0 then
   result[i] := '0'
 else
   result[i] := '1';
end;
end;

function CurrentProcessId: TIdPID;
begin
Result := getpid;


Result := GetCurrentProcessID;

end;

function InMainThread: boolean;
begin
```

```
Result := GetCurrentThreadID = MainThreadID;
end;

procedure LoadMIME(const AFileName : String; AMIMEList : TStringList);
var
KeyList: TStringList;
i, p: Integer;
s, LMimeType, LExtension: String;
begin
If FileExists(AFileName) Then
Begin
 KeyList := TStringList.Create;
 try
   KeyList.LoadFromFile(AFileName);
   for i := 0 to KeyList.Count -1 do begin
     s := KeyList[i];
     p := FilterPos('#', s);
     if (p>0) then
     begin
       setlength(s, p-1);
     end;
     if s <> '' then
     begin
       s := Trim(s);
       LMimeType := Fetch(s);
       if LMimeType <> '' then
       begin
          while (s<>'') do
          begin
            LExtension := Fetch(s);
            if LExtension <> '' then
            try
              AMIMEList.Values['.'+LExtension]:= LMimeType;
            except
              on EListError do {ignore} ;
            end;
          end;
       end;
     end;
   end;
 except
   on EFOpenError do {ignore} ;
 end;
End;
end;
```

```
procedure FillMimeTable(AMIMEList : TStringList);
var
reg: TRegistry;
KeyList: TStringList;
i: Integer;
s: String;
begin
if not Assigned(AMIMEList) then
begin
 Exit;
end;
if AMIMEList.Count > 0 then
begin
 Exit;
end;

AMIMEList.Duplicates := dupError;

with AMIMEList do
begin
//resurses
 Add('.aiff=audio/x-aiff');
 Add('.au=audio/basic');
 Add('.mid=midi/mid');
 Add('.mp3=audio/x-mpg');
 Add('.m3u=audio/x-mpegurl');
 Add('.qcp=audio/vnd.qcelp');
 Add('.ra=audio/x-realaudio');
 Add('.wav=audio/x-wav');
 Add('.gsm=audio/x-gsm');
 Add('.wax=audio/x-ms-wax');
 Add('.wma=audio/x-ms-wma');
 Add('.ram=audio/x-pn-realaudio');
 Add('.mjf=audio/x-vnd.AudioExplosion.MjuiceMediaFile');


 { Image }
 Add('.bmp=image/bmp');
 Add('.gif=image/gif');
 Add('.jpg=image/jpeg');
 Add('.jpeg=image/jpeg');
 Add('.jpe=image/jpeg');
 Add('.pict=image/x-pict');
 Add('.png=image/x-png');
 Add('.svg=image/svg-xml');
```

```
 { Text }
 Add('.323=text/h323');
 Add('.xml=text/xml');
 Add('.uls=text/iuls');
 Add('.txt=text/plain');
 Add('.rtx=text/richtext');
 Add('.wsc=text/scriptlet');
 Add('.rt=text/vnd.rn-realtext');
 Add('.htt=text/webviewhtml');
 Add('.htc=text/x-component');
 Add('.vcf=text/x-vcard');


 { video/ }
 Add('.avi=video/x-msvideo');
 Add('.flc=video/flc');
 Add('.mpeg=video/x-mpeg2a');
 Add('.mov=video/quicktime');
 Add('.rv=video/vnd.rn-realvideo');
 Add('.ivf=video/x-ivf');
 Add('.movie=video/x-sgi-movie');

 { application/ }
 Add('.wmd=application/x-ms-wmd');
 Add('.wms=application/x-ms-wms');
 Add('.wmz=application/x-ms-wmz');
 Add('.p12=application/x-pkcs12');
 Add('.p7b=application/x-pkcs7-certificates');
 Add('.p7r=application/x-pkcs7-certreqresp');
 Add('.qtl=application/x-quicktimeplayer');
 Add('.rtsp=application/x-rtsp');
 Add('.swf=application/x-shockwave-flash');
 Add('.sit=application/x-stuffit');
 Add('.tar=application/x-tar');
 Add('.man=application/x-troff-man');
 Add('.urls=application/x-url-list');
 Add('.zip=application/x-zip-compressed');
 Add('.cdf=application/x-cdf');
end;

Reg := CreateTRegistry; try
 KeyList := TStringList.create;
 try
   Reg.RootKey := HKEY_CLASSES_ROOT;
   if Reg.OpenKeyReadOnly('\') then
```

```
    begin
      Reg.GetKeyNames(KeyList);
    end;
    for i := 0 to KeyList.Count - 1 do
    begin
      if Copy(KeyList[i], 1, 1) = '.' then
      begin
        if reg.OpenKeyReadOnly(KeyList[i]) then
        begin
          s := Reg.ReadString('Content Type');
        if Reg.ValueExists('Content Type') then
        begin
         FFileExt.Values[KeyList[i]] := Reg.ReadString('Content Type');
        end;
          if Length(s) > 0 then
          begin
            AMIMEList.Values[KeyList[i]] := s;
          end;
        end;
      end;
    end;
    if Reg.OpenKeyreadOnly('\MIME\Database\Content Type') then
    begin
      KeyList.Clear;
      Reg.GetKeyNames(KeyList);
      reg.Closekey;
      for i := 0 to KeyList.Count - 1 do
      begin
        if Reg.OpenKeyreadOnly('\MIME\Database\Content Type\' + KeyList[i])
then
        begin
          s := reg.ReadString('Extension');
          AMIMEList.Values[s] := KeyList[i];
          Reg.CloseKey;
        end;
      end;
    end;
   finally
     KeyList.Free;
   end;
  finally
   reg.free;
  end;
  end;

  procedure TIdMimeTable.AddMimeType(const Ext, MIMEType: string);
```

```
var
LExt,
LMIMEType: string;
begin
LExt := AnsiLowerCase(Ext);
if Length(LExt) = 0 then
begin
 raise EIdException.Create(RSMIMEExtensionEmpty);
end
else
begin
if LExt[1] <> '.' then
begin
  LExt := '.' + LExt;
end;
end;
LMIMEType := AnsiLowerCase(MIMEType);
if Length(LMIMEType) = 0 then
 raise EIdException.Create(RSMIMEMIMETypeEmpty);

if FFileExt.IndexOf(LExt) = -1 then
begin
 FFileExt.Add(LExt);
 FMIMEList.Add(LMIMEType);
end
else
 raise EIdException.Create(RSMIMEMIMEExtAlreadyExists);
end;

procedure TIdMimeTable.BuildCache;
begin
if Assigned(FOnBuildCache) then
begin
 FOnBuildCache(Self);
end
else
begin
 if FFileExt.Count = 0 then
 begin
   BuildDefaultCache;
 end;
end;
end;

procedure TIdMimeTable.BuildDefaultCache;
var LKeys : TStringList;
```

```
begin
LKeys := TStringList.Create;
try
 FillMIMETable(LKeys);
 LoadFromStrings(LKeys);
finally
 FreeAndNil(LKeys);
end;
end;

constructor TIdMimeTable.Create(Autofill: boolean);
begin
FFileExt := TStringList.Create;
FFileExt.Sorted := False;
FMIMEList := TStringList.Create;
FMIMEList.Sorted := False;
if Autofill then begin
 BuildCache;
end;
end;

destructor TIdMimeTable.Destroy;
begin
FreeAndNil(FMIMEList);
FreeAndNil(FFileExt);
inherited Destroy;
end;

function TIdMimeTable.getDefaultFileExt(const MIMEType: string): String;
var
Index : Integer;
LMimeType: string;
begin
 Result := '';
LMimeType := AnsiLowerCase(MIMEType);
 Index := FMIMEList.IndexOf(LMimeType);
if Index <> -1 then
begin
 Result := FFileExt[Index];
end
else
begin
 BuildCache;
 Index := FMIMEList.IndexOf(LMIMEType);
 if Index <> -1 then
```

```
    Result := FFileExt[Index];
  end;
  end;

  function TIdMimeTable.GetFileMIMEType(const AFileName: string): string;
  var
  Index : Integer;
  LExt: string;
  begin
  LExt := AnsiLowerCase(ExtractFileExt(AFileName));
  Index := FFileExt.IndexOf(LExt);
  if Index <> -1 then
  begin
   Result := FMIMEList[Index];
  end
  else
  begin
   BuildCache;
   Index := FFileExt.IndexOf(LExt);
   if Index = -1 then
   begin
     Result := 'application/octet-stream'
   end
   else
   begin
     Result := FMIMEList[Index];
   end;
  end;
  end;

  procedure TIdMimeTable.LoadFromStrings(AStrings: TStrings;const
MimeSeparator: Char = '=');
  var
  I   : Integer;
  Ext : string;
  begin
  FFileExt.Clear;
  FMIMEList.Clear;
  for I := 0 to AStrings.Count - 1 do
  begin
   Ext := AnsiLowerCase(Copy(AStrings[I], 1, Pos(MimeSeparator, AStrings[I]) -
1));
   if Length(Ext) > 0 then
     if FFileExt.IndexOf(Ext) = -1 then
       AddMimeType(Ext, Copy(AStrings[I], Pos(MimeSeparator, AStrings[I]) + 1,
Length(AStrings[I])));
```

```
 end;
 end;


 procedure TIdMimeTable.SaveToStrings(AStrings: TStrings;
 const MimeSeparator: Char);
 var
 I : Integer;
 begin
 AStrings.Clear;
 for I := 0 to FFileExt.Count - 1 do
  AStrings.Add(FFileExt[I] + MimeSeparator + FMIMEList[I]);
 end;

 procedure SetThreadPriority(AThread: TThread; const APriority:
TIdThreadPriority; const APolicy: Integer = -MaxInt);
 begin
 if (getpriority(PRIO_PROCESS, 0) < APriority) or (geteuid = 0) then begin
  setpriority(PRIO_PROCESS, 0, APriority);
 end;
 AThread.Priority := APriority;
 end;

 function SBPos(const Substr, S: string): Integer;
 begin
 Result := Pos(Substr, S);
 end;

 function MemoryPos(const ASubStr: String; MemBuff: PChar; MemorySize:
Integer): Integer;
 var
 LSearchLength: Integer;
 LS1: Integer;
 LChar: Char;
 LPS,LPM: PChar;
 begin
 LSearchLength := Length(ASubStr);
 if (LSearchLength = 0) or (LSearchLength > MemorySize) then begin
  Result := 0;
  Exit;
 end;

 LChar := PChar(Pointer(ASubStr))^; //first char
 LPS:=PChar(Pointer(ASubStr))+1;//tail string
 LPM:=MemBuff;
 LS1:=LSearchLength-1;
```

```
LSearchLength := MemorySize-LS1;//MemorySize-LS+1
if LS1=0 then begin //optimization for freq used LF
 while LSearchLength>0 do begin
   if LPM^= LChar then begin
     Result:=LPM-MemBuff+1;
     EXIT;
   end;
   inc(LPM);
   dec(LSearchLength);
 end;//while
end else begin
 while LSearchLength>0 do begin
   if LPM^= LChar then begin
     inc(LPM);
     if CompareMem(LPM,LPS,LS1) then begin
       Result:=LPM-MemBuff;
       EXIT;
     end;
   end
   else begin
     inc(LPM);
   end;
   dec(LSearchLength);
 end;
end;
Result:=0;
End;

function FilterGetHostName: string;
var
LHost: array[1..255] of Char;
i: LongWord;
begin
if GetHostname(@LHost[1], 255) <> -1 then begin
 i := FilterPos(#0, LHost);
 SetLength(Result, i - 1);
 Move(LHost, Result[1], i - 1);
end;
end;
function FilterGetHostName: string;
var
i: LongWord;
begin
SetLength(Result, MAX_COMPUTERNAME_LENGTH + 1);
i := Length(Result);
if GetComputerName(@Result[1], i) then begin
```

```
 SetLength(Result, i);
end;
end;


function IsValidIP(const S: String): Boolean;
var
j, i: Integer;
LTmp: String;
begin
Result := True;
LTmp := Trim(S);
for i := 1 to 4 do begin
 j := StrToIntDef(Fetch(LTmp, '.'), -1);
 Result := Result and (j > -1) and (j < 256);
 if NOT Result then begin
   Break;
 end;
end;
end;


function IsHostname(const S: String): Boolean;
begin
Result := ((FilterPos('.', S) = 0) or (S[1] <> '.')) and NOT IsValidIP(S);
end;

function IsTopDomain(const AStr: string): Boolean;
Var
i: Integer;
S1,LTmp: String;
begin
i := 0;

LTmp := AnsiUpperCase(Trim(AStr));
while FilterPos('.', LTmp) > 0 do begin
 S1 := LTmp;
 Fetch(LTmp, '.');
 i := i + 1;
end;

Result := ((Length(LTmp) > 2) and (i = 1));
if Length(LTmp) = 2 then begin  // Country domain names
 S1 := Fetch(S1, '.');
 if LTmp = 'UK' then begin
   if S1 = 'CO' then result := i = 2;
```

```
    if S1 = 'COM' then result := i = 2;
   end;

   if LTmp = 'TW' then begin
    if S1 = 'CO' then result := i = 2;
    if S1 = 'COM' then result := i = 2;
   end;
  end;
  end;

  function IsDomain(const S: String): Boolean;
  begin
  Result := NOT IsHostname(S) and (FilterPos('.', S) > 0) and NOT
IsTopDomain(S);
  end;

  function DomainName(const AHost: String): String;
  begin
  result := Copy(AHost, FilterPos('.', AHost), Length(AHost));
  end;

  function IsFQDN(const S: String): Boolean;
  begin
  Result := IsHostName(S) and IsDomain(DomainName(S));
  end;


  function ProcessPath(const ABasePath: string;
  const APath: string;
  const APathDelim: string = '/'): string;
  var
  i: Integer;
  LPreserveTrail: Boolean;
  LWork: string;
  begin
  if FilterPos(APathDelim, APath) = 1 then begin
   Result := APath;
  end else begin
   Result := '';
   LPreserveTrail := (Copy(APath, Length(APath), 1) = APathDelim) or
(Length(APath) = 0);
   LWork := ABasePath;
   if (Length(LWork) > 0) and (Copy(LWork, Length(LWork), 1) <> APathDelim)
then begin
    LWork := LWork + APathDelim;
   end;
```

```
    LWork := LWork + APath;
    if Length(LWork) > 0 then begin
      i := 1;
      while i <= Length(LWork) do begin
        if LWork[i] = APathDelim then begin
          if i = 1 then begin
            Result := APathDelim;
          end else if Copy(Result, Length(Result), 1) <> APathDelim then begin
            Result := Result + LWork[i];
          end;
        end else if LWork[i] = '.' then begin
 if (Copy(Result, Length(Result), 1) = APathDelim) and (Copy(LWork, i, 2) =
 '..') then begin
            Delete(Result, Length(Result), 1);
            while (Length(Result) > 0) and (Copy(Result, Length(Result), 1) <>
 APathDelim) do begin
              Delete(Result, Length(Result), 1);
            end;
            Inc(i);
          end else begin
            Result := Result + LWork[i];
          end;
        end else begin
          Result := Result + LWork[i];
        end;
        Inc(i);
      end;
    end;
 if (Result <> APathDelim) and (Copy(Result, Length(Result), 1) = APathDelim)
      and (LPreserveTrail = False) then begin
       Delete(Result, Length(Result), 1);
     end;
   end;
   end;

 constructor TIdLocalEvent.Create(const AInitialState: Boolean = False;
   const AManualReset: Boolean = False);
   begin
   inherited Create(nil, AManualReset, AInitialState, '');
   end;

function TIdLocalEvent.WaitFor: TWaitResult;
   begin
   Result := WaitFor(Infinite);
   end;
```

```
function iif(ATest: Boolean; const ATrue: Integer; const AFalse: Integer):
Integer;
 begin
 if ATest then begin
   Result := ATrue;
 end else begin
   Result := AFalse;
 end;
 end;

function iif(ATest: Boolean; const ATrue: string; const AFalse: string):
string;
 begin
 if ATest then begin
   Result := ATrue;
 end else begin
   Result := AFalse;
 end;
 end;

 function iif(ATest: Boolean; const ATrue: Boolean; const AFalse: Boolean):
Boolean;
 begin
 if ATest then begin
   Result := ATrue;
 end else begin
   Result := AFalse;
 end;
 end;

 procedure TIdReadMemoryStream.SetPointer(Ptr: Pointer; Size: Integer);
 Begin
 inherited SetPointer(Ptr, Size);
 Seek(0,0);//Position:=0;
 End;
function TIdReadMemoryStream.Write(const Buffer; Count: Integer): Longint;
 begin
 Result := 0;
 End;
function  AnsiPosIdx_ (const ASubStr: AnsiString; AStr: PChar; L1: Cardinal;
AStartPos: Cardinal=0): Cardinal;
 var
 L2: Cardinal;
 ByteType : TMbcsByteType;
 Str, SubStr, CurResult: PChar;
 Begin
```

```
 Result:= 0; //not found
 L1 := Length(AStr);
 L2 := Length(ASubStr);
 if (L2=0) or (L2>L1) then Exit;
 Str:=Pointer(AStr);
 SubStr:=Pointer(ASubStr);
 if AStartPos>0 then begin
   Str := Str + AStartPos - 1;
   L1  := L1 + 1 - AStartPos;
 end;
 if L1<=0 then EXIT;
 CurResult := StrPos(Str, SubStr);
 while (CurResult <> nil) and ((L1 - Cardinal(CurResult - Str)) >= L2) do
begin
   ByteType := StrByteType(Str, Integer(CurResult-Str));
   if (ByteType <> mbTrailByte) and
     (Windows.CompareString(LOCALE_USER_DEFAULT, 0, CurResult, L2, SubStr, L2)
= 2) then begin
     Result:=CurResult-Pointer(AStr)+1;
     Exit;
   end;
   if (ByteType = mbLeadByte) then Inc(Result);
   if (ByteType <> mbTrailByte) and
     (strncmp(CurResult, SubStr, L2) = 0) then begin
     Result:=CurResult-Pointer(AStr)+1;
     Exit;
   end;
   Inc(Result);
   CurResult := StrPos(CurResult, SubStr);
 end;
 End;
function  AnsiPosIdx(const ASubStr,AStr: AnsiString; AStartPos: Cardinal=0):
Cardinal;
 Begin
 Result:=AnsiPosIdx_(ASubStr, Pointer(AStr), Length(AStr), AStartPos);
 End;

 function  AnsiMemoryPos(const ASubStr: String; MemBuff: PChar; MemorySize:
Integer): Integer;
 Begin
 Result:=AnsiPosIdx_(ASubStr, MemBuff, MemorySize, 0);
 End;

 Function  PosIdx (const ASubStr,AStr: AnsiString; AStartPos: Cardinal):
Cardinal;
 var
```

```
lpSubStr,lpS: PChar;
LenSubStr,LenS: Integer;
LChar: Char;
Begin
LenSubStr:=Length(ASubStr);
LenS:=Length(AStr);
if (LenSubStr=0) or (LenSubStr>LenS) then begin
  Result:=0;
  EXIT;
end;
lpSubStr:=Pointer(ASubStr);
lpS:=Pointer(AStr);
if AStartPos>0 then begin
  lpS:=lpS+AStartPos-1;
  LenS:=LenS+1-Integer(AStartPos);
end;
LChar :=lpSubStr[0];
lpSubStr:=lpSubStr  +1;
LenSubStr:=LenSubStr-1;
LenS:=LenS-LenSubStr;
if LenS<=0 then begin
  Result:=0;
  EXIT;
end;
while LenS>0 do begin
  if lpS^= LChar then begin
    inc(lpS);
    if CompareMem(lpS,lpSubStr,LenSubStr) then begin
      Result:=lpS-Pointer(AStr);//+1 already here
      EXIT;
    end;
  end
  else begin
    inc(lpS);
  end;
  dec(LenS);
end;
Result:=0;
End;

function MakeMethod (DataSelf, Code: Pointer): TMethod;
Begin
Result.Data := DataSelf;
Result.Code := Code;
End;
```

```
initialization
GStackClass := TIdStack;
ATempPath := TempPath;
GStackClass := TIdStackWindows;
if LeadBytes = [] then begin
  FilterPos := SBPos;
end else begin
  FilterPos := AnsiPos;
end;

SetLength(FilterFalseBoolStrs, 1);
FilterFalseBoolStrs[Low(FilterFalseBoolStrs)] := 'FALSE';
SetLength(FilterTrueBoolStrs, 1);
FilterTrueBoolStrs[Low(FilterTrueBoolStrs)] := 'TRUE';

finalization
FreeAndNil(FIdPorts);
end.
```