Міністерство освіти і науки України
Національний Університет «Одеська політехніка»
Інститут інформаційної безпеки, радіоелектроніки та телекомунікацій
Кафедра кібербезпеки та програмного забезпечення

Гайдукевич Владислав Олександрович,
студент групи РЗ-181

# КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

Розробка системи протидії використанню прихованих каналів в IP-мережах та передачі інформації по них

Спеціальність:
125 Кібербезпека

Спеціалізація, освітня програма:
Кібербезпека

Керівник:
Стопакевич Олексій Аркадійович,
к.т.н., доцент

Одеса – 2022

Міністерство освіти і науки України
Національний Університет «Одеська політехніка»
Інститут інформаційної безпеки, радіоелектроніки та телекомунікацій
Кафедра кібербезпеки та програмного забезпечення
Рівень вищої освіти перший (бакалаврський)
Спеціальність 125 – Кібербезпека
Освітня програма – Кібербезпека

ЗАТВЕРДЖУЮ
Завідувач кафедри КБПЗ

_____
д.т.н.,проф. А.А.Кобозєва
_____202_р.

**ЗАВДАННЯ**
**НА КВАЛІФІКАЦІЙНУ РОБОТУ**

*Гайдукевичу Владиславу Олександровичу*

1.Тема роботи: *Розробка системи протидії використанню прихованих каналів в IP-мережах та передачі інформації по них,*
керівник роботи *Стопакевич Олексій Аркадійович, к. т. н., доцент,*
затверджені наказом ректора від „17” 05. 2022 р. №168-в
2.Зміст роботи*: аналіз проблемної області, постановка задачі, аналіз сучасних досліджень щодо протидії передачі інформації по прихованим к аналам в* ip *мережах, розробка системи протидії передачі інформації по прихов аним каналам в ірмережах,розробка програмного забезпечення системи протид ії передачі інформації по прихованим каналам в* ip*-мережах, охорона праці.*
3. Перелік ілюстративного матеріалу: *гібридна обчислювальна система, поля заголовка протоколу tcp, поля заголовка протоколу істр, поля заголовка протоколу udp, які підходять, скріншоти робочого коду програми, слайди презентації.*

4. Консультанти розділів роботи

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|---|---|---|---|
| Охорона праці | доц. Ярова І. А. | Завдання видав | Завдання прийняв |
| | | | |

5. Дата видачі завдання “_____”_____2022 р.

## КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів кваліфікаційної роботи | Строк виконання | Примітка |
|---|---|---|---|
| *1* | *Аналіз літератури з теми випускної кваліфікаційної роботи* | 18-11-2021 | *виконано* |
| *2* | *Аналіз методів розробки системи протидії використанню прихованих каналів* | 15-01-2022 | *виконано* |
| *3* | *Розробка програмного комплексу* | 20-02-2022 | *виконано* |
| *4* | *Написання розділу з охорони праці* | 13-05-2022 | *виконано* |
| *5* | *Підготовка тексту роботи* | 15-05-2022 | *виконано* |
| *6* | *Підготовка презентації та доповіді* | 21-05-2022 | *виконано* |
| *7* | *Попередній захист* | 25-05-2022 | *виконано* |
| *8* | *Нормоконтроль, рецензування* | 17-06-2022 | *виконано* |
| *9* | *Перевірка на плагіат* | 20-06-2022 | *виконано* |

**Здобувач вищої освіти** _____ *Гайдукевич В.О.*

**Керівник роботи** _____ *Стопакевич О.А.*

**ЗАВДАННЯ**

на розробку розділу “Охорона праці”

*Гайдукевичу Владиславу Олександровичу, група РЗ-181*

Інститут інформаційної безпеки, радіоелектроніки та телекомунікацій

Кафедра кібербезпеки та програмного забезпечення

Тема роботи *Розробка системи протидії використанню прихованих каналів в IP-мережах та передачі інформації по них*

Зміст розділу:

1 Аналіз умов праці і вибір основних заходів виробничої безпеки.

2 Аналіз пожежної безпеки. Вибір заходів та засобів пожежної безпеки.

Керівник роботи

_____ (_____)

(_____)

«____» _____ 2022 р.

Консультант з охорони праці

_____

«____» _____ 2022р.

# АНОТАЦІЯ

Кваліфікаційна робота на тему “ Розробка системи протидії використанню прихованих каналів в IP-мережах та передачі інформації по них ” на здобуття першого (бакалаврського) рівня вищої освіти за спеціальністю 125 – Кібербезпека, спеціалізація, освітня програма: Забезпечення кібербезпеки, містить 15 рисунків, 1 таблицю, 1 додаток, 25 літературних джерел за переліком посилань. Робота виконана на 56 сторінках загального тексту і 52 сторінках основного тексту.

Метою роботи є підвищення рівня безпеки в мережах шляхом розробки та впровадження системи протидії прихованих каналів.

У роботі  проведено аналіз методу знаходження прихованих каналів та протидії їх для подальшої їх ліквідації.

У результаті виконання кваліфікаційної роботи розроблено проект системи протидії прихованим каналам зв’язку в IP- мережах, що дозволило знайти и ліквідувати більшість проблем в мережі, через які були витоки інформації и краще поставити роботу мережі.

Результати даної роботи можуть бути використані при побудові аналогічної системи протидії прихованим каналам зв’язку в IP- мережах.

ІНФОРМАЦІЯ, ІНФОРМАЦІЙНА БЕЗПЕКА, МЕРЕЖІ, СИСТЕМА ПРОТИДІЇ ПРИХОВАНИМ КАНАЛАМ, РИЗИК, СЕРВЕР, ХОСТ.

# ANNOTATION

Qualification work on "Development of a system to counter the use of hidden channels in IP-networks and the transmission of information on them" for the first (bachelor's) level of higher education in the specialty 125 - Cybersecurity, specialization, educational program: Cybersecurity, contains 15 figures, 1 table , 1 supplement, 25 references according to the list of references. The work is performed on 56 pages of general text and 52 pages of main text.

The aim of the work is to increase the level of security in networks by developing and implementing a system of counteraction to hidden channels.

The analysis of the method of finding hidden channels and counteracting them for their further elimination is carried out in the work.

As a result of the qualification work, a system of counteraction to hidden communication channels in IP-networks was developed, which allowed to find and eliminate most of the problems in the network, due to which there were information leaks and better network performance.

The results of this work can be used to build a similar system to counteract hidden communication channels in IP networks.

INFORMATION, INFORMATION SECURITY, NETWORKS, SYSTEM OF CONTROL OF HIDDEN CHANNELS, RISK, SERVER, HOST.

ЗМІСТ

# ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

TCP - Transmission Control Protocol (Протокол управління передачею)

IP - Internet Protocol (протокол мережевого рівня)

DNS - Domain Name System (система доменних імен)

NTFS - New Technology File System (файлова система нової технології)

MAC OS - Гропріетарна операційна система виробництва Apple

SCTP - Stream Control Transmission Protocol (протокол передачі з керування потоком)

RSTEG - Гібридний метод мережевий стеганографії

RTO - Recovery time objective (допустимий час відновлення даних)

LACK - Lost audio packets steganography (метод створення прихованих каналів)

SIP - Session Initiation Protocol (протокол передачі даних)

RTP - Real-time Transport Protocol (працює на прикладному рівні)

CPU - Central processing unit (центральне обробляє пристрій)

GPU - Graphics processing unit (окремий пристрій персонального комп'ютера або ігрової приставки, виконує графічний рендеринг)

FPGA - Field-Programmable Gate Array (програмована логічна матриця)

HTTP - HyperText Transfer Protocol (протокол передачі гіпертексту)

UDP - User Datagram Protocol (протокол призначених для користувача датаграм)

ICMP - Internet Control Message Protocol (протокол міжмережевих керуючих повідомлень)

OID - Object identifier (Механізм ідентифікації)

# ВСТУП

Актуальність роботи полягає в набутті теоретичних і практичних навиків роботи з мережею та знаходження прихованих каналів, мати змогу їх ліквідувати та будь-якими способами протидіяти їм.

Мета розробки полягає у створенні системи для будь-якої мережі в фірмі, підприємстві для протидії прихованим каналам в мережі, після вияву мати змогу виявити створення прихованих каналів та кінці розробити рішення їх ліквідування, або іншого методу протидії .

Предметом досліджень є модель мережі передачі інформації між користувачами.

Задачі. Для досягнення поставленої мети треба вирішити такі задачі:

- проаналізувати вже існуючі методи протидії прихованим канал;
- побудувати модель мережі, на основі якій буде створюватися програма;
- створити саму програму протидії прихованим каналам;
- протестувати роботу програми.

Методи дослідження. При виконанні, в основному, застосовуються методи, які реалізується за допомогою сучасного програмного забезпечення на ПК.

В даний час все джерела, які висвітлюють питання інформаційної безпеки, містять відомості про приховані канали, отримання інформації і навмисне впроваджуваних в різні технічні засоби в пристроях негласного доступу до інформації (отримання, знімання).

А що ж у нас в країні з вирішенням даної проблеми? Аналізуючи сучасну нормативну базу, можна виділити, що існує безліч рішень і методів боротьби з прихованими каналами в мережі, так само є безліч алгоритмів, які блокують передачу даних по мереж.

Виходячи з джерел і всіх стандартів визначено термін «прихований канал» - це непередбачений розробником системи інформаційних технологій і автоматизованих систем комунікаційний канал, який може бути застосований для порушення політики безпеки

При цьому необхідно відзначити, власне що вибір способів протидії загрозам, що реалізовуються за впровадженням прихованих каналів орієнтується, виходячи з персональних індивідуальностей інший захищених системи (топологія зведення системи, що застосовують протоколи інформаційної взаємодії, індивідуальностей розташування складових систем і їх взаємодії між собою, телекомунікаційних засобів і засобів захисту інформації) [13].

# 1 АНАЛІЗ СУЧАСНИХ ДОСЛІДЖЕНЬ ЩОДО ПРОТИДІЇ ПЕРЕДАЧІ ІНФОРМАЦІЇ ПО ПРИХОВАНИМ КАНАЛАМ В IP МЕРЕЖАХ

## 1.1 Лімітування пропускної можливості прихованих каналів в IP-мережах

Зазвичай, приховані канали за механізмом передачі інформації ділять на канали по пам'яті і канали за часом.

Кваліфікувати приховані канали по пам'яті і часу можливо у вигляді переліку критерій. Прихований канал вважається каналом по пам'яті при виконанні належних умов:

– Відправник і одержувач зобов'язані володіти доступом до складової спільного ресурсу;

– Відправник може поміняти цей елемент ресурсу;

– Одержувач зобов'язаний розрізнити всі зміни;

– Відправник і одержувач мають можливість використовувати канал з невисокою пропускною здатністю;

– У разі якщо не обрано особливий спосіб кодування, щоб уникнути черговості схожих знаків, відправник і одержувач зобов'язані володіти ймовірністю спочатку домовитися про тимчасову перерву, в напрямок якого одержувач стане стежити за змінами в каналі.

Подібно, прихований канал вважається каналом за часом при виконанні належних умов:

– Відправник і одержувач зобов'язані володіти доступом до складової спільного ресурсу;

– Відправник і одержувач зобов'язані ділити провідну частоту (синхроніз ація);

– Відправник зобов'язаний володіти ймовірністю змінювати час відповідного сигналу одержувача для виявлення конфігурації в складовому ресурсі;

– Відправник і одержувач можуть використовувати канал з невисокою пропускною здатністю.

Часом до прихованих каналів відносять ще канали, в яких для приховування інформації використовують фон «дані» пакета. Інформація передається в фон «дані», яке і вважається контейнером для передачі даних. З іншого боку, дані канали не сподіваються порушити правила захищеності. В наслідок цього описані канали передачі інформації не визначені до класу прихованих каналів[8].

Приховані канали по пам'яті в IP-мережах можуть бути усунені шляхом шифрування або нормалізації значень полів заголовків пакетів. Таким чином, зведення прихованих каналів наданого вигляду не можна використовувати, при наявності шифрування трафіку, власне що вважається нормальною методикою мережевої оборони[21].

Даною особливістю володіють і приховані канали по часу. Вирівнювання довжин переданих пакетів і встановлення єдиної швидкості передачі пакетів вважаються методами знищення прихованих каналів по пам'яті, і прихованих каналів за часом відповідно до цього. Хоча, ці способи призводять до істотного зниження залишкової пропускної можливості каналу зв'язку. Характеристики даних способів вибираються, як компроміс між залишковою пропускною здатністю прихованого каналу і залишкової пропускною здатністю самого каналу зв'язку [11].

Наприклад, як час проходження пакета, випадковий розмір, що має властивості, властиві гамма-розподілу, промахи при передачі інформації по прихованого каналу, заснованому на зміні довжин пакетних інтервалів, мають

всі шанси привести до різниці у часі відправника і одержувача [4]. Потреба повторюваної синхронізації, ще знижує пропускну здатність прихованого каналу.

Запропоновані такі методи підтримки синхронізації відправника і одержувача, що базуються на:

– Відправники пакетів особливого виду;

– Впровадження «інтервалів тиші» для зміни параметрів кодування;

– Впровадження «інтервалів регулювання» для зміни параметрів кодування в режимі реального часу;

– Фазового автопідстроювання частоти.

Перспективне призначення наступних досліджень, отримання кількісних даних цих способів, що дозволяють знизити пропускну здатність ймовірного прихованого каналу до сенсу, такого що функціонування прихованих каналів з найменшою пропускною здатністю є нешкідливим.

## 1.2 Виявлення прихованих каналів за часом в IP-мережах

Є безліч способів боротьби з прихованими каналами зв'язку в мережі, починаючи з програм шпигунів, які знаходять прихований канал і звідки проходить витік інформації, і забезпечуючи можливість боротьби з ними, закінчуючи кодуванням і дешифруванням як самого вмісту пакета, так і заголовка пакета.

Хотілося ще адресуватися до способів виявлення прихованих каналів, пропонується 2 способи:

– Статистичний метод;

– Сигнатурний спосіб.

Статистичний спосіб виявлення прихованих каналів передбачає збір статистичних даних про пакети, що проходять крізь захищену ділянку мережі,

без внесення в них будь-яких змін. При цьому виявлення прихованих каналів має можливість проводитися як в режимі реального часу, так і застосовуючи дані, накопичені за минулі відрізки часу.

Спосіб виявлення прихованих каналів на базі сигнатурного аналізу аналогічний методиці, що застосовується антивірусним пз для розшуку шкідливих програм. При наявності комплекту реалізацій прихованих каналів, для будь-якої з них складається сигнатура. У струмені даних ведеться розвідка цих сигнатур. За підсумками даної роботи робиться висновок про недоступність наявності прихованих каналів в системі і варіанті його реалізації боротьби з ними.

Термін «прихований канал» [1] був введений творцями в 1973 році. В ідеалі під прихованим каналом розуміється непередбачений розробником системи інформаційних технологій і автоматичних систем комунікаційний канал [2], який має можливість бути використаний для порушення захищеності. Запити довіри до захищеності інформації, намагаються звертатися до відправника та людей, які займаються безпекою в мережі[23].

Здібності протоколу IP дають можливість таємно транслювати інформацію і створювати приховані канали, модулюючи короткочасні властивості, сенсу полів заголовків і довжини переданих пакетів.

Втім знайому заходи протидії наданої небезпеки, що складаються в "нормалізації" характеристик IP-трафіку (тобто, в передачі IP-пакетів фіксованої довжини з фіксованими заголовками крізь рівні проміжки часу), призводять до істотного зниження продуктивності застосування пропускної можливості каналів зв'язку, нарощування ціни їх застосування і втрати активних ймовірностей протоколу IP. Широке поширення протоколу IP готує завдання вивчення прихованих каналів в IP-мережах актуальною.

Творцями запропонований спосіб протидії витоку інформації по прихованим каналам в IP-мережах [6], що включає в себе: ідентифікація, тест, знищення, лімітування пропускної можливості, аудит і виявлення.

Знищення здатності зведення частини прихованих каналів в IP-мережах призводить до неприпустимого лімітування залишкової пропускної можливості каналу зв'язку.

В наслідок цього згодом аналізу ймовірного прихованого каналу приймається висновок або про превентивний лімітуванні його пропускної можливості, або про виявлення прецеденту передачі інформації по цьому каналу. Надана замітка дає аналітичний огляд існуючих способів обмеження пропускної можливості та виявлення прихованих каналів в IP-мережах.

Створення прихованого каналу і втілення впливу порушника на інформаційні ресурси, які захищаються, відповідно до наведеної моделлю виповнюється прийдешнім порядком[5]:

1. У режимі штатного функціонування робота з захищеними інформаційними ресурсами ведеться в установленому порядку, суб'єкти, які мають організований доступ до них, втілять в життя обробку відповідно до встановлених правил розмежування доступу. Інспектор відображає недоступність порушень захищеності інформації.

2. У складі обробки захищених інформаційних ресурсів є в наявності завчасно злісно впроваджений розвідник порушника захищеності, яка не показує власної енергійності і жодним чином не може виявити власного наявні в наданої іт (ас).

3. У важливий для порушника момент часу агенту від порушника захищеності сервірується команда на активацію і виконання власного, активного навантаження. Команда має можливість бути подана як за штатними каналами зв'язку іт (ас), в випадки присутності можливості такого включення

(наприклад через інтернет), і впровадженням радіоканалів, при наявності подібний здатності у порушника [10].

4. Впроваджений розвідник порушника захищеності може продати власне, активне навантаження, при цьому канал інформаційної взаємодії між порушником і впровадженим агентом має можливість бути прихований від інспектора.

5. Згодом заслуги поставленої задачі робота агента закінчується автоматично , або ж по команді порушника.

В якості захисних дій пропонується застосувати:

– Зниження / обмеження пропускної можливості каналу передачі інформації (щодо прихованих каналів);

– Структура укладення зведення системи;

– Прогноз продуктивності оборони системи.

Таким чином, можна сказати, власне що ми отримуємо новий виток інформаційного протиборства «порушник - адміністратор безпеки»[15], який заносить в наше життя, як нові технології і способи нападу, так , і нові методи протидії.

Недотримання доступності і працездатності системи має можливість привести до важких результатів для країни при витоку секретної інформації. Основною проблемою вважається ще й те, власне, що переважна більшість елементної бази для цих систем виконується і поставляється через межі, а виконати абсолютний ансамбль подій, щодо пошуку ймовірних прихованих каналів і заставних приладів, для всього списку ввозяться складових, не можна на технічному рівні. А як стало відомо, технічні засоби іноземного виробництва можуть бути сповнені прикрих «сюрпризів».

Неможливо обійти стороною і повсюдне становлення мережі онлайн, і впровадження її як автотранспорту для зв'язку всіляких корпоративних і

промислових сіток, власне, що механічно дозволяє зовнішньому порушнику отримати керуючий доступ до запровадженого заставного приладу або ж модулю [3].

Є над чим думати і працювати. Питання виявлення прихованих каналів в автоматичних системах організацій робиться актуальним, за межами залежності від значення організації і її форми власності. Таємниця і вважається таємницею, внаслідок того, що її розуміє вузьке коло осіб. Плюс до цього можна додати присутність (отримання) несприятливих вражень, коли хтось зловмисно завдає шкоди вашій інформаційній інфраструктурі. І зіпсований настрій не найжахливіше, якщо при цьому бізнес-процес в організації може отримати травми.

Створення прихованих потоків є досить жорстким порушенням політики ІБ, втому потрібно досить часто проводити тест мережі на можливі витоки інформації. Перевірка мережі проводиться адміністратором, в його обов'язки входить:

– Виявлення прихованих каналів;

– Оцінка пропускної здатності прихованих каналів;

– Оцінка ризиків;

– Виділення сигналу і типу інформації, що передається по прихованим каналах;

– Протидія реалізації прихованого каналу аж до його знищення.

Добре організована мережа, захист якої побудована на принципі ешелонування - це серйозний бар'єр для зловмисника. Досить доброю системою захисту мережі вважається безпека на таких рівнях:

– Входу в корпоративну мережу;

– Сервера корпоративної мережі;

– Кінцевого пристрою користувача.

Найбільш адекватним виходом для захисту домашніх комп'ютерів є використання їх господарями провайдинг-сервісу доставки "чистого" контенту, що пройшов фільтрацію на вузлах провайдера. Більшість людей поки що скептично відноситься до такого способу. Адже система оплати трафіку не припускає виявлення шкідливого контенту - користувач платить за загальний обсяг, незалежно від того, скільки "сміття" в результаті осіло на його пристрої.

Альтернативою превентивному лімітуванню пропускної можливості прихованих каналів вважається виявлення функціонуючих каналів. Плюсом наданого розкладу - недоступність додаткового навантаження на канал зв'язку. Втім присутність ненульових промахів, а ще можливість витоку критично необхідної інформації до спрацьовування способу, виділяють важливим використовувати способи виявлення разом з способами обмеження пропускної можливості[20].

Далі будуть розглянуті способи виявлення прихованих каналів за часом в IP-мережах, що базуються на модуляції довжин пакетних інтервалів, наприклад як популярні схеми передачі інформації, що базуються на модуляції довжин переданих пакетів. З іншого боку, канали по пам'яті, що базуються на зміні бітів заголовків пакетів, можуть бути усунені шляхом шифрування трафіку, або ж нормалізації значень полів заголовків.

Зауважимо, власне, що підтверджена вірогідність зведення невидимого прихованого каналу, в разі якщо порушнику відома схема виявлення. Хоча ці підсумки мають аналітичний характер: зважаючи на обмежений діапазон методів зведення прихованих каналів і невеликої кількості способів їх виявлення практична застосовність цих результатів при аналізі прихованих каналів в IP-мережах незначна [7].

Наприклад, мають місце бути способи виявлення, які неможливо вважати надійними: у всіх способах наявних ненульової помилки, при побудові

прихованого каналу з шумом, тест існуючих методів протидії витоку інформації по прихованим каналах в IP-мережах існує 15 схем кодування.

Втім характеристики даних способів невідомі: відсутні кількісні характеристики (пропускна спроможність, ступінь промахів) прихованого каналу, стійкого до наявних способів виявлення. Багатообіцяючим напрямком подальшої роботи вважається тест здатності зведення прихованих каналів, які не виявляються існуючими способами виявлення, і отримання кількісних даних цих каналів.

## 1.3 Аналіз моделей і класифікація атак прихованих каналів

Широке поширення атак використовують уразливість виду «прихованого каналу» можна зрозуміти, як широке розповсюдження великої кількості програмних засобів різної спрямованості, за допомогою яких можливо створити необхідний прихований канал, наприклад[9]:

- Графічні редактори.
- Аудіо редактори.
- Текстові редактори.
- ОС.
- Сайти, що містять додатки для створення прихованих каналів.

Для захисту від прихованих каналів в автоматизованій системі необхідно знати способи їх реалізації та механізми роботи. Найбільш поширені типи прихованих каналів наведені нижче. Приховані канали можна розбити на 4 основні класи, за методом впливу:

- Стеганографічні (внесення змін до графіку, аудіо, файли).
- Мережеві (внесення змін до TCP / IP пакетів).
- Текстові маніпуляції (словесні маніпуляції, заміни).

– Маніпуляції з механізмами операційних систем (приховування даних, незадекларовану передачу інформації).

– Маніпуляції з відкритими даними (розташування деяким, не випадковим чином, поява в певній послідовності і ін.).

Слід навести перелік факторів, що сприяють поширенню прихованих каналів в автоматизованих банківських системах:

– Широке використання автоматизованих банківських систем (практично будь-який комерційний банк має системи дистанційного банківського обслуговування).

– Наявність сучасних програмних і високопродуктивних апаратних засобів у зловмисників.

– Поширене використання дистанційного банківського обслуговування аж до інтернет ресурсів з персональними пропозиціями для кожного клієнта.

Широке поширення на увазі простоти організації отримали приховані канали на основі широко поширених мережевих протоколів.

Найбільш поширеними є протоколи TCP / IP і DNS. У протоколі TCP / IP організувати прихований канал можна використовуючи заголовки пакетів.

Таким способом реалізується прихований канал з низькою пропускною здатністю. Слід зазначити, що виявлення даного каналу вельми складно через низку причин:

– Канал може бути задіяний вкрай нетривалий час.

– Відібрати даний запит з сотень, а то і тисяч запитів, які проходять через DNS досить важко.

– Запит є по суті звичайним, і нічим не виділяється, і з першого погляду не несе ніякої інформативною навантаження.

Використання перестановок. Основна ідея: відстеження послідовностей звернень до певних ресурсів мережі інтернет. Наприклад, якщо зловмисник

контролює сайти (певні розділи сайтів), то сформована послідовність звернень, що є перестановкою, є кодом переданого повідомлення. Також можуть використовуватися гіперпосилання всередині певного документа на інші розділи сайту[19].

Як видно, приховані канали на основі відкритих широко поширених мережевих протоколів, незважаючи на те, що є канали з низькою пропускною здатністю, мають ряд переваг:

– Вкрай низька вартість створення.

– Відносно висока складність виявлення.

Використання даних прихованих каналів виправдано в ролі керуючих, при гібридної атаки, за допомогою декількох типів прихованих каналів, або у зв'язку з іншими типами атак.

Приховані канали в графічних файлах базуються на внесення певної кількості змін в початковий файл. В даному методі необхідно враховувати, що чим більше внесених змін, тим простіше виявити даний канал. У загальному випадку частка внесеної інформації не повинна перевищувати 20-25% від вихідної. Якщо використовувати графічний файл розміром 1 мегабайт, то він здатний нести в собі до 200 кілобайт початкових даних. Ймовірність виявлення різко падає в разі використання групи файлів. Також зростає пропускна здатність даного каналу[18].

Програма дозволяє приховати дані розміром до 256  мегабайт в безлічі файлів з розширеннями аудіо, відео, графічних файлів, а також в файлах flv, swf, pdf. Даний метод може бути застосований в реалізації прихованих каналів в автоматизованих банківських системах, тому що у кожного банку існує інтернет ресурс дистанційного обслуговування, в якому міститься безліч малюнків аж до персональних пропозицій для кожного клієнта[12].

Приховування даних з використанням недоліків файлових систем. Будь-яка сучасна файлова система, будь то ext3 або NTFS, дозволяє додавати до файлів деяку службову інформацію.

Так, наприклад в файлової системі NTFS можна привласнити файлу атрибут «архівний» або «прихований», і якщо привласнити цей атрибут не випадковим, а заздалегідь певним способом, то за характером розподілу можна задати певний інформаційне повідомлення. Також в файлової системі NTFS введені додаткові атрибути, для забезпечення сумісності з MAC OS, використовуючи їх також можливо передати інформацію.

В цьому випадку пропускна здатність прихованого каналу залежить від кількості використовуваних файлів і використовуваних полів. Але необхідно враховувати, що більш повне наповнення файлів різними атрибутами підвищить ймовірність виявлення даного каналу. Також, при збереженні даних файлів в іншій файловій системі, всі приховані дані будуть загублені. В цілому даний канал зв'язку можна використовувати більше для управління, ніж для передачі великої кількості даних. Даний вид прихованого каналу також популярний в рамках нашої роботи, тому що банк обмінюється з клієнтами різної документацією в різних форматах.

Складність виявлення безпосередньо залежить від вибору сценарію і умов стороннього спостерігача (наприклад, його розташування). З недоліків варто згадати той факт, що даний метод важко реалізувати. Потрібно з'ясувати, якій код використовує програма для голосового зв'язку, підібрати код з найменшою різницею втрати якості мови, при цьому дають більше місця для вкладення стеганограмми [8]. При стисненні втрачається якість переданої мовної інформації.

Методи SCTP-стеганографії можна розбити на три групи:

– Методи, в яких змінюється вміст SCTP-пакетів.

– Методи, в яких змінюється послідовність передачі SCTP-пакетів.

– Методи, які об'єднують два попередніх та впливають на зміст пакетів і на їх порядок при передачі (гібридний метод).

Методи зміни вмісту SCTP-пакетів засновані на факті, що кожен stcp-пакет складається з частин, і кожна з цих частин може містити змінні параметри. Незалежно від реалізації, статистичний аналіз адрес мережевих карт, використовуваних для пересланих блоків, може допомогти у виявленні прихованих каналів.

Модифікація пакетів з використанням гібридного методу може бути представлена на прикладі системи HICCUPS (hidden communication system for corrupted networks), яка використовує недосконаліть передачі даних в мережевому оточенні, такі як перешкоди і шум в середовищі зв'язку, а також звичайну схильність даних до спотворення.

HICCUPS є стеганографічної системою з розподілом пропускної спроможності в громадській телекомунікаційної мережевому середовищі [14]. Бездротові мережі (wi-fi, Bluetooth, wimax), на відміну від провідних (ETHERNET), більш сприйнятливі до можливого спотворення даних, і тому використання перешкод і шуму в каналі зв'язку під час роботи системи виглядає стандартно, а значить непомітно. «прослуховування» всіх кадрів з передаються даними в середовищі поширення і можливість повторної відправки пошкоджених кадрів з невірно відкоригованими кодовими значеннями - дві найважливіші мережеві особливості, абсолютно необхідні для реалізації HICCUPS. Зокрема, бездротові мережі передачі даних в АІС використовують повітряне середовище поширення і з'єднання з динамічною частою помилок в бітах (BER), що забезпечує можливість штучно вводити пошкоджені кадри в передачу. Даний метод має досить низьку пропускну здатність (залежить від мережі), складною і громіздкою реалізацією, але низькою стеганографічної

вартістю і досить високою складністю виявлення. Але тим не менш, вірогідність виявлення є, адже якщо провести аналіз кадрів з помилками з (невірної контрольною сумою), то з'ясується що в кадрах з помилками простежується логіка, періодичність, і це може привести до виявлення прихованого каналу зв'язку[16].

Метод RSTEG базується на механізмі повторної передачі пакетів, суть якого полягає в тому, що: коли відправник а посилає пакет, то одержувач б, не дивлячись на отримання пакета від а, не відповідає пакетом з прапором підтвердження, отже повинен спрацювати механізм повторної відправки пакета, і на цьому етапі надсилається пакет із запровадженою стеганограммой, тобто організовується прихований канал. Але і в цей раз, одержувач приймає пакет і не виробляє відсилання підтвердження його отримання. Пакет висилається повторно ще раз, але в цей раз, вже початковий, без впровадженої стеганограмми, і одержувач отримавши його, відсилає підтвердження отримання. Сеанс передачі закривається, в результаті чого санкціоновані дані були отримані адресатом без змін вмісту, і додатково була проведена передача даних по таємному, несанкціонованому каналі, що реалізується в легальному каналі зв'язку. Продуктивність (пропускна здатність каналу зв'язку, що реалізується за допомогою методу RSTEG) залежить від сукупності декількох чинників, таких як деталі процедур зв'язку (в частотності, розмір корисного навантаження пакета, частота, з якою генеруються сегменти і так далі).

RSTEG на основі RTO характеризується низькою ймовірністю виявлення і низькою пропускною здатністю, а SACK має максимальну для RSTEG пропускною здатністю. Застосування RSTEG з використанням TCP протоколу є обґрунтованим вибором для IP-мереж. З недоліків даного методу слід виділити те, що даний метод досить складно реалізувати, особливо ті його варіації, які ґрунтуються на перехопленні і виправленні переданих санкціонованими

користувачами АІС пакетів. До того ж у зв'язку з різким зростанням кількості пересилаються (ретранслюються) пакети, або появи нестандартних, не властивих даному каналу зв'язку затримок, що виникають внаслідок передачі стеганограмми, можуть викликати підозри у адміністраторів АІС [17].

# 2 РОЗРОБКА СИСТЕМИ ПРОТИДІЇ ПЕРЕДАЧІ ІНФОРМАЦІЇ ПО ПРИХОВАНИМ КАНАЛАМ В IP МЕРЕЖАХ

## 2.1 Визначення пропускної здатності прихованих каналів та виявлення факторів, які впливають їх пропускну здатність

Для початку, аби проаналізувати канал в мережі та зробити його модель, потрібно визначення пропускну здатність прихованих каналів та виявлення факторів, які впливають на пропускну здатність.

Пропускна здатність каналу дорівнює кількості інформації, яке може передаватися по ньому в одиницю часу.

Зазвичай пропускна здатність вимірюється в бітах в секунду (біт/с) і кратних одиницях кбіт/с і мбіт/с. Однак іноді в якості одиниці використовується байт в секунду (байт / с) і кратні йому одиниці.

Співвідношення між одиницями пропускної здатності каналу передачі інформації такі ж, як між одиницями вимірювання кількості інформації:

1байт / с = 23біт / с = 8біт / с;

1кбит / с = 210біт / с = 1024біт / с;

1 мбіт / с = 210кбіт / с = 1024кбіт / с;

1гбіт / с = 210мбіт / с = 1024мбіт / с.

## 2.2 Розробка гібридної формальної моделі розподіленої обчислювальної системи.

Розподілені обчислювальні системи - це сформована сфера високопродуктивних обчислень, що може володіти своєю специфікою, яскраво вираженим класом вирішуваних завдань та способом їх вирішення. Розробляються нові концепції побудови розподілених систем, розширюється

коло розв'язуваних ними завдань, спрощується процес організації, розробляються прості методи використання ресурсів простими користувачами.

В свою чергу гібридна обчислювальна система з гетерогенної апаратної обчислювальної структурою. Комбінація будь-яких обчислювальних пристроїв або блоків, наприклад обчислення за допомогою CPU і GPU. Завдяки цим двом особливостям дана архітектура більше адаптується до сучасніх умов, так як досягла своєї межі в сучасному аналізі і можливості побудови систем. В кінцевому підсумку мі отримаємо систему, яка багатофункціональна і швидко обраховане, в сучасних умовах.

Для прикладу ось така схема:

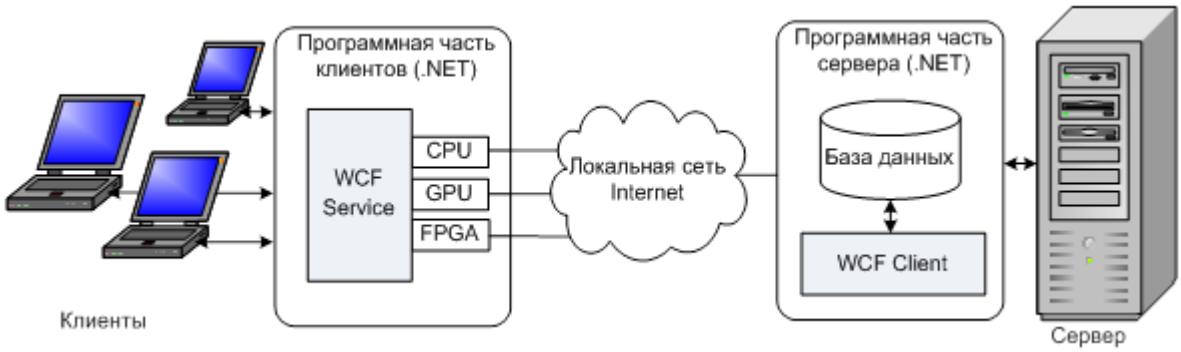


Рисунок 2.1.1 — Гібридна обчислювальна система

На ній можна побачити як перед з'єднанням до серверу клієнти відправляють інформацію через CPU і GPU механізми. В свою чергу серверна частина являє собою WCF-клієнт, який по локальній мережі або через Інтернет звертається до обчислювальних ресурсів клієнтів. Основними функціями серверної частини розробленої гібридної системи є: оцінка продуктивності клієнтів і швидкості з'єднання; моніторинг клієнтів і процесу обчислення; остаточна обробка результатів обчислень, отриманих від клієнтів.

Основні переваги розробленої гібридної реконфігурованою системи для високопродуктивних обчислень, в порівнянні з існуючими аналогами:

– Об'єднання в єдину систему обчислювальних вузлів різної архітектури (CPU, GPU і FPGA).

– Простота розгортання системи. Щоб підключити нового клієнта до обчислень, необхідно зайти на веб-сторінку проекту в Інтернеті, або в мережевому оточенні. Потім на стороні клієнта буде встановлена остання версія системи, і він буде готовий до обчислень.

– Можливість комунікації на основі різних протоколів, в тому числі і з шифруванням.

– Можливість динамічного додавання і відключення як клієнтів, так і окремих обчислювальних блоків клієнта під час обчислень.

– Розробка обчислювальних задач на сучасних мовах програмування, таких як C #, VB.NET, Python і ін.

Для моєї роботи, так як я вибрав мову програмування Python ця гібридна обчислювальна система підходить, для подальшого представлення моєї системи в роботі мережі.

## 2.3 Способи організації прихованих каналів передачі інформації шляхом модифікації пакетів

Провідна ідея при застосуванні методів трансформації пакетів охоплюється у використанні деяких полів заголовків при внесенні в їх стеганограмми. Це можливо внаслідок особливостям певних полів заголовків, які не задіються при передачі наданих або змінюються певним чином, не порушуючи роботу самого пакета.

На рисунку 2.3.1 показаний обсяг заголовка пакетів протоколу TCP. Переданий документ видається одним з основних протоколів передачі наданих

в мережі. Механізм TCP надає функціонал, за допомогою якого можна здійснювати передачу з гарантованою цілісністю, шляхом виконання вторинних запитів наданих при можливості втрати або зміни самого пакета. Переданий документ функціонує на транспортному рівні TCP / IP і намагається передати пакет в повній цілісності, наприклад, переданий документ може використовуватися для протоколів прикладного рівня HTTP або HTTPS, за допомогою яких може повідомлятися значима інформація, наприклад, інформація аутентифікації або інформація про банківські операції.


Рисунок 2.3.1 — Поля заголовка протоколу TCP, які підходять для розміщення стеганограмм

Поля допустимо можуть вживатися заради розміщення в них стеганограмм. Поле "Порт відправника" має розмір в 2 байта, а поле "Послідовний номер" - 4 байта. Передача з підтримкою наданих полів імовірна один раз при встановленні з'єднання. Отже, за один показ ймовірно буде 6 байт інформації.

На рисунку 2.3.2 показаний обсяг заголовка пакету UDP. Переданий документ видається одним з ключових протоколів TCP / IP. Внаслідок своєї простоті він використовується при передачі потокових відео і аудіо, в мережевих комп'ютерних іграх і т.д. UDP має на увазі, що перевірка цілісності або непотрібна, або відповідно обстежена в самому додатку.


Рисунок 2.3.2 — Поля заголовка протоколу UDP, які підходять для розміщення стеганограмм

Представлені на малюнку поля допустимо можуть вживатися заради розміщення в них стеганограмм. У протоколі присутнє поле, яке зберігає габарит корисного навантаження (довжина датаграми). Отже, цей габарит корисного навантаження потенційно можна використовувати при прихованій передачі.

На рисунку 2.3.3 показаний обсяг заголовка пакета ICMP. Особливість наданого протоколу в тому, що він не спеціалізується на передачі даних. Як правило ICMP використовується для передачі повідомлень при погрішностях і інших незвичайних ситуаціях, що утворилися при передачі даних, наприклад, запитувана допомога недоступна, хост, або маршрутизатор не відповідають. Так для ICMP покладаються деякі сервісні функції. Також, близького вживання в мережі протоколу IPv6, документ ICMPv6 представляється неодмінною і без нього правильна дія на жаль неможливо. За цих причин створення прихованого каналу передачі для протоколу ICMP є більш цікавою.


Рисунок 2.3.3 — Поля заголовка протоколу ICMP, які підходять для розміщення стеганограмм

# 3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ПРОТИДІЇ ПЕРЕДАЧІ ІНФОРМАЦІЇ ПО ПРИХОВАНИМ КАНАЛАМ В IP МЕРЕЖ

## 3.1 Розробка методу парирування прихованого каналу

Для виявлення прихованих каналів я вирішив використовувати метод виявлення за часом і поліпшити його, для подальшої зручної роботи з програмою.

Методи виявлення прихованих каналів за часом в IP-мережі за принципом роботи розділені на три групи:

– виявлення приватного класу прихованих каналів, заснованих на зміні довжин міжпакетних інтервалів;

– виявлення шляхом аналізу закономірностей в трафіку;

– виявлення шляхом порівняння з «еталонної» моделлю трафіку.

Робота методів другої групи заснована на припущенні, що при наявності прихованого каналу трафік стає більш передбачуваним. Особливий інтерес представляє дослідження методів першої і другої груп, так як дані методи спеціальним чином спроектовані для виявлення прихованих каналів за часом в мережах пакетної передачі інформації.

Тобто, проаналізувавши дану інформацію можна прийти до такого висновку, що для подальшої роботи по виявленню прихованих каналів зв'язку і створення програмного забезпечення потрібно звернути увагу на час проходять пакетів по мережі, а точніше інтервал між пакетами, і аналіз закономірностей трафіку по мережі. Як на мене, найкращим рішенням буде звернути велику увагу на інтервал між пакетами і час витрачений на їх доставку. Так як для проходження великого обсягу інформації вимагає великої кількості часу для його доставки одержувачу. Звичайно це залежить від самого відправника і

вмісту пакета, але в більшості випадків, саме повільна доставка пакета уповільнює роботу сервера і мережі в цілому.

Розробка програмної частини для обраного методу хороша на вибір більшості програмних інструментів. Мова С представляється переважно знаменитим, перевірених у своїй області, внаслідок його поширеність серед всіляких архітектур і платформ. Програмний код, написаний для С має великий стрімкістю роботи, втім генеральним вадою даного мови представляється велике розмаїття допустимо ризикованих компонентів і конструкцій, що тримаються в мові. Наприклад, основними проблемами є витік пам'яті, зважаючи на відсутність в мові механізму збору сміття, наявність вразливих функцій (printf, strcpy і т.д.).

Мова C++ це компільований мову загального призначення з типом статичності. Утримує такі парадигми програмування, як об'єктно і загальне програмування. Синтаксис мови успадкований від С і в більшості своїй є з ним сумісним. В ядрі самої мови, також як і в його стандартній бібліотеці відсутній механізм збору сміття, що накладає на розробника додаткові завдання по контролю своєчасного звільнення пам'яті.

Python це високорівнева мова програмування загального призначення, орієнтований на підвищення продуктивності розробника і читання коду. Синтаксис ядра Python мінімалістичний. В той же час Стандартна бібліотека включає великий обсяг корисних функцій. Python підтримує декілька парадигм програмування, в тому числі структурний, об'єктно-орієнтоване, функціональне, імперативне і аспектноорієнтірованное. Основні архітектурні риси - динамічна типізація, автоматичне керування пам'яттю, повна інтроспекція, механізм обробки виключень, підтримка багатопоточних обчислень і зручні високорівневі структури даних [24]

Pcap це бібліотека, що дозволяє створювати програми аналізу мережевих даних, що надходять на мережеву карту комп'ютера. прикладом програмного забезпечення, що використовує бібліотеку Pcap, служить програма Wireshark. Різноманітні програми моніторингу та тестування мережі, сніфери використовують цю бібліотеку. Вона призначена для використання спільно з мовами C / C ++, а для роботи з бібліотекою на інших мовах, таких як Java, .NET, використовують обгортки. Для Unix-подібних систем це бібліотека libpcap, а для Microsoft Windows - WinPcap. Програмне забезпечення мережевого моніторингу може використовувати libpcap або WinPcap, щоб захопити пакети, подорожують по мережі, і (в новіших версіях) для передачі пакетів в мережі. Libpcap і WinPcap також підтримують збереження захоплених пакетів в файл і читання файлів, що містять збережені пакети. Програми, написані на основі libpcap або WinPcap, можуть захопити мережевий трафік, аналізувати його. Файл захопленого трафіку зберігається в форматі, зрозумілому для додатків, що використовують Pcap [25].

## 3.2 Розробка програми захисту від передачі інформації по прихованим каналам в IP мереж

У попередніх розділах ми розглянули якою мовою краще буде розробити програму, для нашого подальшого методу виявлення прихованих каналів, в результаті ми зупинилися на Python. Так як дана мова дуже підходить по функціоналу для роботи з мережею і дуже простий у використанні, як для адміністратора, так і для звичайного користувача. Так само в ньому немає тих проблем, які є в інших мовах.

Для роботи з Python спочатку нам буде потрібно його встановити. Після установки робота проходить в консолі, але для нашої програми потрібна робота

з графічним інтерфейсом. На рисунку 3.2.1 показані 3 основні модулі, які потрібні встановити для того щоб далі працювати з мережею і IP-адресами.

```
pip3 install pysnmp

pip3 install datetime

pip3 install ipaddress
```

Рисунок 3.2.1– Модулі для роботи з мережею і IP-адресами

Є багато модулів які справляються з даної роботи, ми вибрали ті які зазвичай використовуються для точного визначення роботи хоста, які в підсумку отримують інформацію від його OID.

На рисунку 3.2.2 показано скрипт, який повинен вимагати hostname.

```
from pysnmp.hlapi import *
from ipaddress import *
from datetime import datetime

# var section

#snmp
community_string = 'derfnutfo'  # From file
ip_address_host = '192.168.88.1'  # From file
port_snmp = 161
OID_sysName = '1.3.6.1.2.1.1.5.0'  # From SNMPv2-MIB hostname/sysname

# function section

def snmp_getcmd(community, ip, port, OID):
    return (getCmd(SnmpEngine(),
                   CommunityData(community),
                   UdpTransportTarget((ip, port)),
                   ContextData(),
                   ObjectType(ObjectIdentity(OID))))

def snmp_get_next(community, ip, port, OID):
    errorIndication, errorStatus, errorIndex, varBinds = next(snmp_getcmd(communit
y, ip, port, OID))
    for name, val in varBinds:

        return (val.prettyPrint())

#code section

sysname = (snmp_get_next(community_string, ip_address_host, port_snmp, OID_sysNam
e))
print('hostname= ' + sysname)
```

Рисунок 3.2.2 – Скрипт, який вимагає hostname

У першій частині ми бачимо що відбувається проце присвоєння порту, адреси хоста і т.д. У другій частині ж відбувається вже процес використання вже самого імені хоста, так само запитується порт через який він підключений і його OID. І в підсумку виводиться ім'я хоста. Звичайно можна використовувати сам адресу, але для точної перевірки і безпеки використовується ім'я.

Розберемо скрипт детальніше. Спочатку ми імпортуємо необхідні модулі:

1. pysnmp - забезпечує роботу скрипта з хостом по SNMP;

2. ipaddress - забезпечує роботу з адресами. Перевірка адрес на коректність, перевірка на входження адреси на адресу мережі тощо;

3. datetime- отримання поточного часу. У цьому завданню потрібен для організації логів.

Потім заводимо чотири змінних:

1. community

2. адреса хоста

3. порт SNMP

4. значення OID

Дві функції:

1. snmp_getcmd

2. snmp_get_next

Перша функція надсилає запит GET вказаною хосту, через порт,, з зазначеним comminity і OID.

Друга функція це генератор snmp_getcmd. Напевно розбивати на дві функції було не зовсім правильно, але вже так вийшло :)

В цьому скрипті не вистачає деяких речей:

1. У скрипт необхідно завантажити ip адреси хостів. Наприклад, з текстового файлу. При завантаженні необхідно перевірити завантаження адреса на коректність, інакше pysnmp може дуже сильно здивуватися і скрипт

зупиниться з traceback. Неважливо, звідки ви будете брати адреси з файлу, з бази даних, але ви повинні бути впевнені, що адреси, які ви отримали - коректні. І так, джерело адрес текстовий файл, один рядок - один адреса в десяткового формі.

2. Мережеве обладнання може бути виключене на момент опитування, може бути неправильно налаштоване, в результаті pysnmp видасть в цьому випадку абсолютно не те, що ми чекаємо і при подальшій обробці отриманої інформації отримаємо зупинку скрипта з traceback. Потрібен обробник помилок для нашої взаємодії по SNMP.

3. Потрібен лог файл, в який будуть записуватися оброблені помилки.

Нижче показаний фрагмент коду, на якому ми бачимо, що програма зчитує адреси, заздалегідь прописані або задокументовані, так як для кожної мережі потрібно свої адреси. Після зчитування файлу з адресами программа перевіряє їх на правильність і можливість роботи в мережі, і при наявності помилок створює лог файл в який записує час і які помилки були виявлені, для подальшої ліквідації їх в мережі, для повної її безпеки.

```
filename_of_ip = 'ip.txt' # имя файла с Ip адресами
#Log
filename_log = 'zone_gen.log' #

def check_ip(ip): # проверка ip адреса корректность
    try:
        ip_address(ip)
    except ValueError:
        return False
    else:
        return True

def get_from_file(file, filelog): # выбирает ip адреса из файла. одна строка - один
адрес в десятичной форме
    fd = open(file,'r')
    list_ip = []
    for line in fd:
       line=line.rstrip('\n')
       if check_ip(line):
           list_ip.append(line)
       else:
            filed.write(datetime.strftime(datetime.now(), "%Y.%m.%d %H:%M:%S") + ':
Error Мусор в источнике ip адресов ' + line)
            print('Error Мусор в источнике ip адресов ' + line)
```

```
    fd.close()
    return list_ip

#code section

#открываем лог файл
filed = open(filename_log,'w')

# записываем текущее время
filed.write(datetime.strftime(datetime.now(), "%Y.%m.%d %H:%M:%S") + '\n')

ip_from_file = get_from_file(filename_of_ip, filed)

for ip_address_host in ip_from_file:
    sysname = (snmp_get_next(community_string, ip_address_host, port_snmp, OID_sysN
ame))
    print('hostname= ' + sysname)

filed.write(datetime.strftime(datetime.now(), "%Y.%m.%d %H:%M:%S") + '\n')
filed.close()
```

У функції snmp_get_next вже є висновок помилок errorIndication, errorStatus, errorIndex, varBinds. У varBinds вивантажуються отримані дані, в змінні, що починаються з error, вивантажується інформація по помилках. Це тільки потрібно правильно обробити. Так як в подальшому в скрипті буде ще кілька функцій по роботі з snmp, має сенс обробку помилок винести в окрему функцію (див. рис. 3.2.3).

```
def errors(errorIndication, errorStatus, errorIndex, ip, file):
    #обработка ошибок В случае ошибок возвращаем False и пишем в файл
    if errorIndication:
        print(errorIndication, 'ip address ', ip)
        file.write(datetime.strftime(datetime.now(), "%Y.%m.%d %H:%M:%S") + ' : ' +
str(errorIndication) + ' = ip address = ' + ip + '\n')
        return False
    elif errorStatus:
        print(datetime.strftime(datetime.now(), "%Y.%m.%d %H:%M:%S") + ' : ' + '%s
at %s' % (errorStatus.prettyPrint(), errorIndex and varBinds[int(errorIndex) - 1]
[0] or '?'))
        file.write(datetime.strftime(datetime.now(), "%Y.%m.%d %H:%M:%S") + ' : ' +
'%s at %s' % (errorStatus.prettyPrint(), errorIndex and varBinds[int(errorIndex) -
1][0] or '?' + '\n'))
        return False
    else:
        return True
```

Рисунок 3.2.3 – Фрагмент коду, для обробки помилок (перша функція)

І тепер додаємо в функцію snmp_get_next обробку помилок і запис в лог файл. Функція тепер повинна повертати не тільки дані, але і повідомлення про те, чи були помилки (див. рис. 3.2.5).

```
def snmp_get_next(community, ip, port, OID, file):
    errorIndication, errorStatus, errorIndex, varBinds = next(snmp_getcmd(communit
y, ip, port, OID))
    if errors(errorIndication, errorStatus, errorIndex, ip, file):
        for name, val in varBinds:
            return (val.prettyPrint(), True)
    else:
        file.write(datetime.strftime(datetime.now(), "%Y.%m.%d %H:%M:%S") + ' : Err
or snmp_get_next ip = ' + ip + ' OID = ' + OID + '\n')
        return ('Error', False)
```

Рисунок 3.2.4 - Фрагмент коду, для обробки помилок (друга функція)

Відправимо на час налагодження різний рівень логування, щоб потім не виколупувати по всьому скрипту зайві повідомлення (див. рис. 3.2.6).

```
for ip_address_host in ip_from_file:
    #  получаем sysname hostname+domainname, флаг ошибки
    sysname, flag_snmp_get = (snmp_get_next(community_string, ip_address_host, port
_snmp, OID_sysName, filed))

    if flag_snmp_get:
        # Всё хорошо, хост ответил по snmp
        if sysname == 'No Such Object currently exists at this OID':
            #  а community неверный.надо пропускать хост, иначе словим traceback. П
ричём мы никак не поймаешь, что проблема в community, поэтому всегда надо запрашива
ть hostname, который отдают все устройства
            print('ERROR community', sysname, ' ', ip_address_host)
            filed.write(datetime.strftime(datetime.now(), "%Y.%m.%d %H:%M:%S") + '
: ' + 'ERROR community sysname = ' + sysname + '  ip = ' + ip_address_host + '\n')
        else:
            if log_level == 'debug':
                filed.write(datetime.strftime(datetime.now(), "%Y.%m.%d %H:%M:%S")
+ ' : ' + '  sysname ' + sysname + ' type ' + str(type(sysname)) + ' len ' + str(le
n(sysname)) + ' ip ' + ip_address_host + '\n')
            if len(sysname) < 3
                if log_level == 'debug' or log_level == 'normal':
                    filed.write(datetime.strftime(datetime.now(), "%Y.%m.%d %H:%M:%
S") + ' : ' + 'Error sysname  3  = ' + sysname + '  ip = ' + ip_address_host +
'\n')
            if sysname.find(domain) == -1:
                # что-то отдало hostname без домена, например Huawei или Catos
                sysname = sysname + '.' + domain
                  if log_level == 'debug' or log_level == 'normal':
                    filed.write("check domain     : " + sysname + " " + ip_address_
host + " " + "\n")

        print('hostname= ' + sysname)
```

Рисунок 3.2.5 – Фрагмент коду, для обробки помилок (третя функція)

В результаті запуску програми ми отримуємо готовий призначений для користувача інтерфейс, в якому є: кнопка для вибору файлу з заздалегідь приготованими IP-адресами і поле виведення результату аналізу даних адрес (див. рис. 3.2.6).

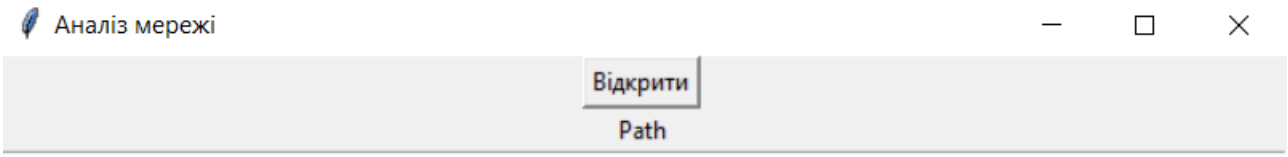


Рисунок 3.2.6 – Інтерфейс програми.

Після натискання кнопки ми можемо вибрати будь-який файл з IP-адресами для перевірки. Наприклад, у мене використано простий текстовий файл, але можна використовувати будь-який формат (див. рис 3.2.7).

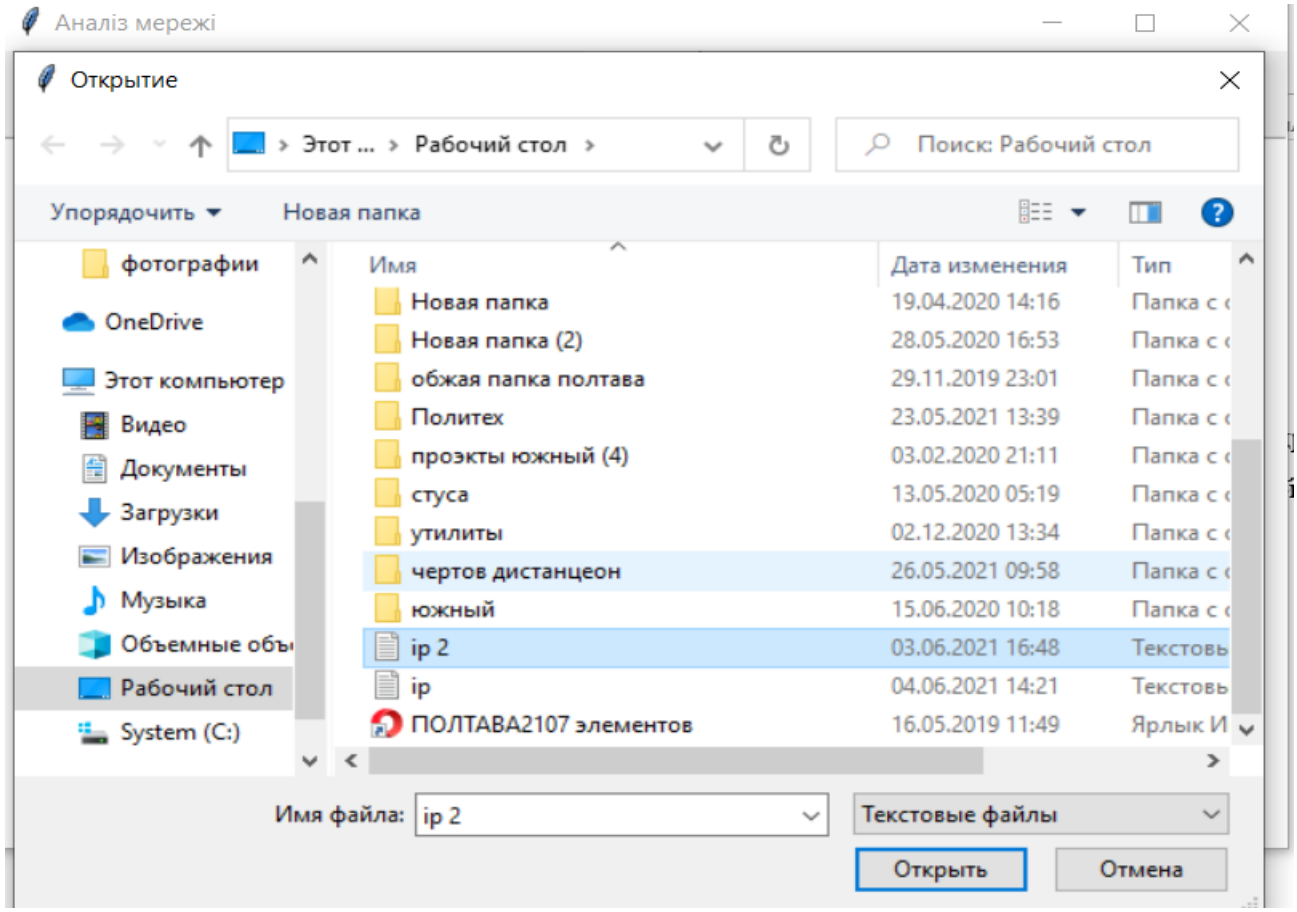


Рисунок 3.2.7 – Вибір адрес для аналізу

Після вибору файлу, починається аналіз. Чим більше IP-адрес будуть в список, тим більше часу знадобиться для аналізу, так як програма відстежує час відправлення запиту від користувача до сервера. Після аналізу ми бачимо два типи IP-адрес: з помилкою і без помилки. З помилкою це IP- адреси які написані не правильно і програма їх не розпізнає, або час запиту перевищує проходження по мережі, коли він був відправлений від користувача і до того часу коли він надійде на сервер, з цього дана програма вважає такі IP-адреси не правильними. А другі успішно пройшли перевірку і можуть далі використовуватися в мережі (див. рис 3.2.8).

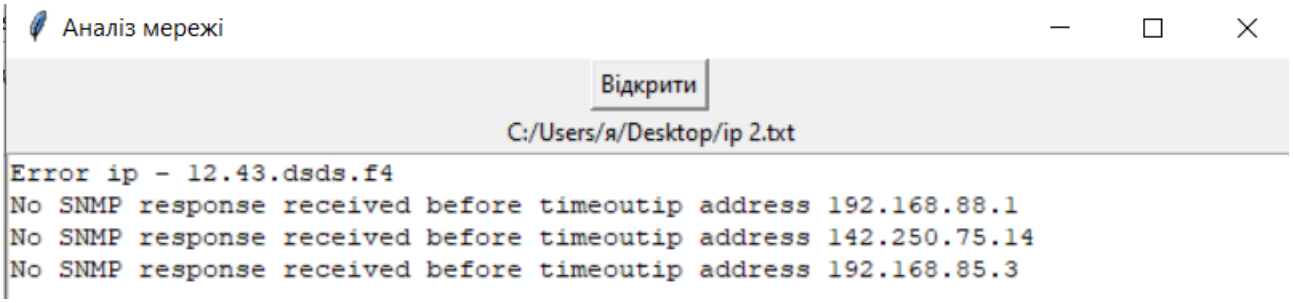


Рисунок 3.2.8 – Результат аналізу.

Після закриття програми дані автоматично записуються в лог-файл. Після кожного сеансу аналізу туди вноситься час перевірки, час який пішов на перевірку кожної адреси і адреси які вибили помилку і не справні, і вимагають втручання (див. рис 3.2.10).

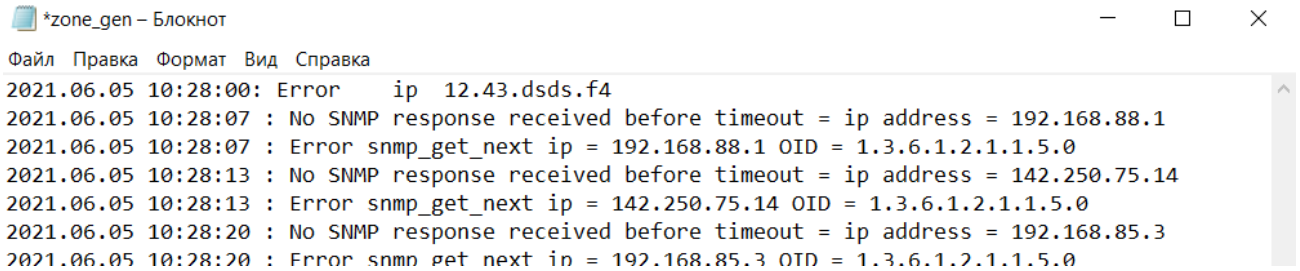


Рисунок 3.2.9 - Результат перевірки в окремому файлі.

# 4 ОХОРОНА ПРАЦІ

Питання охорони праці людини необхідно вирішувати на всіх стадіях трудового процесу незалежно від виду професійної діяльності.

Забезпечення безпечних і здорових умов праці в значній мірі залежить від правильної оцінки небезпечних, шкідливих виробничих факторів. Однакові по складності зміни в організмі людини можуть бути викликані різними причинами. Це можуть бути фактори виробничого середовища, надмірне фізичне і розумове навантаження, нервово-емоційна напруга, а також різне сполучення цих причин.

У даному розділі вирішується питання охорони праці веб-розробника на стадії розробки ним програмно-апаратної частини системи управління навчанням для кафедри університету.

Приміщення, в якому працює програміст, має загальну площу 8 м$^2$, висоту стелі 3 м. У приміщенні знаходиться 1 робоче місце з ноутбуком. Робоче місце обладнане робочим столом площею 1,2 м$^2$, стільцем та ноутбуком, що складається з ноутбуку, зарядного пристрою, миші.

Так, як робота програміста за енерговитратами належить до категорії легких робіт Іа, Іб, тому мають виконуватись наступні вимоги згідно з ДСН 3.3.6.042-99:

- оптимальна температура повітря – 22°С (допустима – 20-24°С);
- оптимальна відносна вологість – 40-60% (допустима – не більш 75%);
- швидкість руху повітря не більш 0,1 м/с.

Нормованим параметром природного освітлення згідно ДБН В.2.5–28 – 2006 є коефіцієнт природного освітлення (КПО). КПО встановлюється в залежності від розряду виконуваних зорових робіт. Робота програміста відноситься до робіт середньої точності (IV розряд зорових робіт, мінімальний розмір об'єкту розрізнення складає 0,5-1,0мм), для яких при використанні

бокового освітлення КПО=1,5%. Для штучного освітлення нормованим параметром виступає Емін – мінімальний рівень освітленості, та Кп – коефіцієнт пульсації світлового потоку, який не повинний бути більшим ніж 20%. Мінімальна освітленість встановлюється в залежності від розряду виконуваних зорових робіт. Для IV розряду зорових робіт вона складає 300-500 лк.

Робоче місце веб-розробника складається з: ноутбуку, зарядного пристрою для ноутбуку, дротової миші, офісного стільця. Біля робочого місця знаходиться мережевий фільтр на 6 розеток по 220В кожна. Біля виходу з приміщення знаходиться вогнегасник та план евакуації.

Вплив шуму на веб-розробника. Під впливом шуму знижується концентрація уваги, порушуються фізіологічні функції, з'являється втома у зв'язку з підвищеними енергетичними витратами і нервово-психічним напруженням, погіршується мовна комутація. Все це знижує працездатність людини і її продуктивність, якість і безпеку праці. Згідно з ДСН 3.3.6.037-99 «Санітарні норми виробничого шуму, ультразвуку та інфразвуку» рівень шуму у робочому приміщенні, з урахуванням важкості та напруженості праці,, не має перевищувати 50дБА. У приміщенні рівень шуму знаходиться у діапазоні від 40дБА до 60дБА. Для зменшення шуму здійснюється ослаблення причин шуму, а саме збільшення відстані до джерела шуму, та використання навушників.

Як міри по зниженню шуму можна додатково запропонувати:

– облицювання стелі і стін звукопоглинаючим матеріалом (знижують шум на 6-8 дб);

– екранування робочого місця (постановкою перегородок, діафрагм);

– установка в комп'ютерних приміщеннях устаткування, що робить мінімальний шум;

– раціональне планування приміщення.

Ерогонімічні аспекти. Ергономічними аспектами проектування відеотермінальних робочих місць, зокрема, є: висота робочої поверхні, розміри простору для ніг, вимоги до розташування документів на робочому місці (наявність і розміри підставки для документів, можливість різного розміщення документів, відстань від очей користувача до екрану, документа, клавіатури і т.д.), характеристики робочого крісла, вимоги до поверхні робочого столу, регульованість елементів робочого місця.

Головними елементами робочого місця веб-розробника є стіл і крісло. Основним робочим положенням є положення сидячи.

Робоча поза сидячи викликає мінімальне стомлення програміста. Раціональне планування робочого місця передбачає чіткий порядок і сталість розміщення предметів, засобів праці і документації. Те, що потрібно для виконання робіт частіше, розташоване в зоні легкої досяжності робочого простору.

Робочий стіл задовольняє наступні умови:

- висота столу вибрана з урахуванням можливості сидіти вільно, в зручній позі, при необхідності спираючись на підлокітники;
- нижня частина столу сконструйована так, що веб-розробник може зручно сидіти, не змушений підбирати ноги;
- поверхня столу має властивості, що виключають появу відблисків у поле зору веб-розробника;
- висота робочої поверхні знаходиться в межах 720мм.

При роботі з персональним комп'ютером дуже важливу роль грає дотримання правильного режиму праці та відпочинку. В іншому випадку у персоналу наголошуються значна напруга зорового апарату з появою скарг на незадоволеність роботою, головні болі, дратівливість, порушення сну, втому і хворобливі відчуття в очах, в поясниці, в області шиї і руках. Перерви та

робочий час виконувались відповідно до СанПіН 2.2.2 542-96 «Гігієнічні вимоги до відеодисплейних терміналів, персональних електронно-обчислювальних машин і організації робіт ». Денний робочий час становив 6 годин, 70 хвилин – загальний час перерв. Під час перерв виконувалась виробнича гімнастика, фізкультхвилинки, з двох-трьох вправ потягування з глибоким диханням, обертання тулуба, присідання тощо.

Електробезпека. Статична електрика. Приміщення за небезпекою ураження електричним струмом можна віднести до 1 класу, тобто це приміщення без підвищеної небезпеки (сухе, без пилу, з нормальною температурою повітря, ізольованими підлогами і малим числом заземлених приладів).

На робочому місці веб-розробника немає металевих деталей, корпус ноутбуку виконаний із пластику, деталі самого ноутбуку щільно закриті всередині та не мають прямого доступу до них.

Основні причини ураження людини електричним струмом на робочому місці:

- дотик до металевих неструмоведучих частин (внутрішніх деталей ноутбуку), що можуть виявитися під напругою в результаті ушкодження ізоляції та у разі пошкодження корпусу ноутбука;
- нерегламентоване використання електричних приладів;
- відсутність інструктажу співробітників з правил електробезпеки.

Пожежна безпека являє собою комплекс організаційних заходів щодо забезпечення нормального протипожежного стану об’єкта нерухомості — житлового, виробничого, складського, офісного або торгово-розважальної споруди, приміщення або комплексу приміщень.

Основним законодавчим документом, що регламентує вимоги щодо пожежної безпеки, є Закон України "Про пожежну безпеку", згідно з яким

необхідно огляд приміщень, та профілактики пожежі, надзвичайно важлива правильна оцінка пожежонебезпеки будинку, визначення небезпечних факторів і обґрунтування способів і засобів пожежопопередження і захисту.

До виникнення пожеж, в приміщеннях пов’язаних із веб-розробкою, найчастіше призводять:

- куріння;
- порушення правил користування електроприладами;
- перенавантаження електромережі;
- необережне поводження з вогнем.

Комп’ютерне обладнання повинне підключатися до електромережі лише за допомогою справних штепсельних з’єднань і електророзеток заводського виготовлення.

У штепсельних з’єднаннях та електророзетках, крім контактів фазового та нульового робочого провідників, мають бути спеціальні контакти для підключення нульового захисного провідника. Їх конструкція має бути такою, щоб приєднання нульового захисного провідника відбувалося раніше, ніж приєднання фазового та нульового робочого провідників. Порядок роз’єднання при відключенні має бути зворотним.

Не допускається підключати комп’ютерну техніку до звичайної двопровідної електромережі, зокрема з використанням перехідних пристроїв.

На підприємствах необхідно організувати вивчення всіма працівниками правил пожежної безпеки і дій у разі виникнення пожежі. Осіб, які не пройшли інструктаж із пожежної безпеки, не можна допускати до роботи. Кожен працівник зобов’язаний виконувати ці вимоги, а також вживати заходи щодо усунення порушень правил пожежної безпеки, ліквідації пожеж і загорянь.

Згідно із Правилами пожежної безпеки в України, громадяни України зобов’язані:

- виконувати вимоги правил пожежної безпеки, забезпечу-вати будівлі, які їм належать на правах особистої власно-сті, первинними засобами гасіння пожеж і протипожежним
- інвентарем, виховувати в дітей обережність у поводженні з вогнем;
- повідомити пожежну охорону про виникнення пожежі та вжити заходи для її ліквідації, рятування людей і майна.

Основними завданнями пожежної безпеки є: контроль за дотриманням протипожежних вимог, запобігання пожеж і нещасних випадків від них, гасіння пожеж, рятування людей і надання допомоги в ліквідуванні наслідків аварій, катастроф і стихійного лиха.

Гасіння пожежі електроустановок під напругою здійснюється за виконання таких обов’язкових умов:

- не допускається наближення пожежних до струмопровідних частин електроустановок на відстань менше 4 метрів;
- маршрути руху пожежних на бойові позиції КГП повинен погоджувати з черговим персоналом енергооб’єкта і конкретно вказувати кожному пожежнику під час інструктажу;
- пожежні і водії пожежних автомобілів, які забезпечують подачу вогнегасних речовин, повинні працювати в діелектричних рукавицях і взутті;
- подавання вогнегасних речовин необхідно проводити після заземлення ручних пожежних стволів і пожежних автомобілів;
- перестановку сил і засобів, зміну бойових позицій тощо КГП повинен виконувати після узгодження зі старшою посадовою особою з присутнього інженерно-технічного персоналу енергетичного об’єкта.

Під час гасіння пожежі електроустановок під напругою забороняється:

- використання усіх видів піни;

- проводити будь-які відключення та інші операції з електричним обладнанням особовому складу пожежних підрозділів;
- використовувати воду зі змочувачами при подаванні компактних струменів води, як для гасіння, так і для охолодження електрообладнання та будівельних конструкцій;
- наближатися до машин і механізмів, які застосовуються для подачі води (вогнегасних речовин) на електроустановки під напругою, особам, безпосередньо не зайнятим на гасінні пожежі.

У даному розділі дипломної роботи були викладені вимоги до робочого місця веб-розробника. Створені умови повинні забезпечувати комфортну роботу. Були зазначені оптимальні розміри робочого столу, робочої поверхні. Дотримання умов, що визначають оптимальну організацію робочого місця веб-розробника, дозволить зберегти гарну працездатність протягом усього робочого дня, підвищить як в кількісному, так і в якісному відносинах продуктивність праці веб-розробника , що в свою чергу сприятиме якнайшвидшій розробці і налагодженню програмного продукту.

# ВИСНОВКИ

В рамках випускної кваліфікаційної роботи бакалавра розроблений програмний комплекс для дослідження способів прихованої передачі інформації в IP-мережах. Зазначений комплекс забезпечує можливість ознайомлення студентів з таким поняттям, як приховані канали передачі інформації і забезпечує можливість виконання лабораторних робіт і практикумів з таки[ дисциплін, як "Безпека мереж" і "Інформатика", що вивчаються на кафедрі кібербезпеки та програмного забезпечення, Державного Університету «Одеська політехніка» .

У дослідницькому розділі наведено різні методи знаходження та протидії прихованим каналам в мережах, а також методи передачі інформації по цим каналам. Розглянуто особливості протоколів, які можуть передаватися по прихованим каналам. Також розглянуті класифікація і способи організації прихованих каналів передачі інформації.

У практичному розділі була розглянута модель мережі, на основі якої було розроблено програмний продукт. Розглянуто розробка програмного комплексу: процес вибору мови програмування, методу та системи по якій буде відбуватися розробка програми.

# ПЕРЕЛІК ПОСИЛАНЬ

1. Безукладников И.И., Кон Е.Л. Скрытые каналы в распределенных автоматизированных системах. *Вестник УГАТУ*. 2010. Т. 14, №2. С. 245-250.

2. Безукладников И.И. Кон Е.Л. Проблема скрытых каналов в промышленных информационно-управляющих и коммуникационных сетях. *Вестник УГАТУ*. 2011. №7. С. 61-64.

3. Безукладников И.И. Кон Е. Л. Проблема скрытых каналов в промышленных управляющих системах. *Вестник УГАТУ*. 2012 №2. С. 56-62.

4. Безукладников И.И., Кон Е.Л. Скрытые каналы в распределенных информационно-управляющих системах. *Вестник УГАТУ*. 2012. №3. С. 126-133

5. Lampson B.W. A Note on the Confinement Problem. Communications of the *ACM*. 1973. V.16, N 10. P. 613-615.

6. Архангельская А.В. Когосов К. О подходе к противодействию утечки информации по скрытым каналам. *Безопасность информационных технологий*. 2013. Т.20, № 4. С.10-20.

7. Амосов А.А. Дубинский Ю.А. Копченова Н.В. Вычислительные методы для инженеров. Москва: Высшая школа. 1994. 544 с.

8. Андрианов В.И. Соколов А.В. Средства мобильной связи. Санкт – Петербург: БВХ Петербург, 1998. 256 с.

9. Анин Б.Ю. Защита компьютерной информации. Санкт – Петербург: БВХ Петербург, 2000. 384 с.

10. Баранов В.М. Защита информации в системах и средствах информатизации и связи. Санкт – Петербург: БВХ Петербург, 1996. 111с.

11. Барсуков В.С., Водолазский В.В. Интегральная безопасность информационно-вычислительных и телекоммуникационных сетей. *Технологии электронных коммуникаций.* 1993. Т.34. Ч.1. С.1-146 с.

12. Барсуков В.С., Водолазкий В.В. Современные технологии безопасности. М.: Нолидж, 2000. 496с.

13. Барсуков В.С. Дворянкин С.В. Шеремет И.А. Безопасность связи в каналах телекоммуникаций. Москва: Мир, 1992. 154 с.

14. Батурин Ю.М. Жодзишский А.М. Компьютерная преступность и компьютерная безопасность. Москва: Юрид. Литература, 1991. 160 с.

Березин Б.И. Березин С.Б. Начальный курс С и С ++. Москва: Диалог МИФИ, 2001. 288 с.

15. Берлекемп Е. Алгебраическая теория кодирования. Москва: Мир,1971. 477 с.

16. Варфоломеев А.А. Жуков А.Е. Мельников А.Б. Устюжанин Д.Д. Блочные криптосистемы. Основные свойства и методы анализа устойчивости. Москва: МИФИ, 1998..

17. Матвеев С.В. Меры защиты от скрытых каналов в автоматизированных системах и их пропускная. *Доклады ТУСУР*. 2012. № 1(25). Ч. 2. С. 74–77.

18. Архангельская А.В. Когос К.Г. Пропускная способность скрытых каналов, основанных на модуляции длин передаваемых пакетов, при увеличении длин пакетов случайным образом. *Безопасность информационных технологий*. 2015. Т. 22, № 3. С.10-16.

19. Edekar S., Goudar R. Capacity boost with data security in Network Protocol Covert Channel. *Computer Engineering and Intelligent Systems*. 2013. V.4, No.5, P. 55-59.

20. Ященко В.В. Введение в криптографию. Москва: Мир, 1999. 272 с.

21. Водолазкий І.М. Комерційні системи шифрування: основні алгоритми і їх реалізація. Рівне, 1992.

22. Кащеев В.И. Мониторинг телефонной сети. *Системы безопасности*. 1995, № 1. С 23-26.

23. Мафтик С. Механизмы защиты в сетях ЭВМ. М.: Мир, 1993. 216 с.

24. Learn Python – Programming Tutorials For Beginners. URL: https://www.programiz.com/python-programming

25. Packet capture library. URL: https://wiki.wireshark.org/libpcap

# Додаток А. Листінг програми

```
# import section
from tkinter import INSERT

from pysnmp.hlapi import *
from ipaddress import *
from datetime import datetime
import tkinter as tk
from tkinter.filedialog import askopenfilename

# var section

# snmp
community_string = 'derfnutfo'
ip_address_host = '192.168.88.1'
port_snmp = 161
OID_sysName = '1.3.6.1.2.1.1.5.0'  # From SNMPv2-MIB
hostname/sysname
filename_of_ip = 'C:\\Users\\я\\Desktop\\ip.txt'  # Ip
# log
filename_log = 'zone_gen.log'  # для лог файла
log_level = 'debug'

domain = 'mydomain.ua'


# function section

def snmp_getcmd(community, ip, port, OID):
    # type class 'generator' errorIndication, errorStatus,
errorIndex, result[3] - список
    # метод get получаем результат обращения к устойстройству по
SNMP с указаным OID
    return (getCmd(SnmpEngine(),
                   CommunityData(community),
                   UdpTransportTarget((ip, port)),
                   ContextData(),
                   ObjectType(ObjectIdentity(OID))))


def snmp_get_next(community, ip, port, OID, file):
    # метод обрабатывает class generator от def snmp_get
    # обрабатываем errors, выдаём тип class
'pysnmp.smi.rfc1902.ObjectType' с OID (в name) и значением  (в val)
    # получаем одно скалярное значение
```

```
    errorIndication, errorStatus, errorIndex, varBinds =
next(snmp_getcmd(community, ip, port, OID))

    if errors(errorIndication, errorStatus, errorIndex, ip, file):
        for name, val in varBinds:
            return (val.prettyPrint(), True)
    else:
        file.write(datetime.strftime(datetime.now(),
                                     "%Y.%m.%d %H:%M:%S") + ' :
Error snmp_get_next ip = ' + ip + ' OID = ' + OID + '\n')
        return ('Error', False)


def get_from_file():
    file = askopenfilename(
        filetypes=[("Текстовые файлы", "*.txt"), ]
    )
    lbl_path['text'] = file
    # Загрузка ip адресов из файла file, запись ошибок в filelog
    fd = open(file, 'r')
    ip_from_file = []
    for line in fd:
        line = line.rstrip('\n')
        if check_ip(line):
            ip_from_file.append(line)
        else:
            filed = open(filename_log, 'w')

            # записываем текущее время

            filed.write(datetime.strftime(datetime.now(),
                                          "%Y.%m.%d %H:%M:%S") + ':
Error    ip  ' + line)
            txt_field.insert(INSERT, 'Error ip - ' + line + '\n')
    fd.close()
    for ip_address_host in ip_from_file:
        # получаем sysname hostname+domainname, флаг ошибки
        sysname, flag_snmp_get = (snmp_get_next(community_string,
ip_address_host, port_snmp, OID_sysName, filed))

        if flag_snmp_get:
            # Всё хорошо, хост ответил по snmp
            if sysname == 'No Such Object currently exists at this
OID':
                # а community неверный.надо пропускать хост, иначе
словим traceback. Причём ты никак не поймаешь, что проблема в
community, поэтому всегда надо запрашивать hostname, который отдают
все устройства
```

```
                txt_field.insert(INSERT, 'ERROR community' +
sysname + ' ' + ip_address_host)
                filed.write(datetime.strftime(datetime.now(),
                                                "%Y.%m.%d %H:%M:%S")
+ ' : ' + 'ERROR community sysname = ' + sysname + '  ip = ' +
ip_address_host + '\n')
            else:

                if log_level == 'debug':
                    filed.write(datetime.strftime(datetime.now(),
                                                    "%Y.%m.%d
%H:%M:%S") + ' : ' + '  sysname ' + sysname + ' type ' + str(
                        type(sysname)) + ' len ' +
str(len(sysname)) + ' ip ' + ip_address_host + '\n')
                if len(sysname) < 3:
                    sysname = 'None_sysname'
                    if log_level == 'debug' or log_level ==
'normal':

filed.write(datetime.strftime(datetime.now(),
                                                          "%Y.%m.%d
%H:%M:%S") + ' : ' + 'Error sysname  3  = ' + sysname + '  ip = ' +
ip_address_host + '\n')
                if sysname.find(domain) == -1:
                    # что-то отдало hostname без домена, например
Huawei или Catos
                    sysname = sysname + '.' + domain
                    if log_level == 'debug' or log_level ==
'normal':
                        filed.write("check domain     : " + sysname
+ " " + ip_address_host + " " + "\n")

            print('hostname= ' + sysname)
            filed.close()


def check_ip(ip):
    #  Проверка ip адреса на корректность. False проверка не
пройдена.
    try:
        ip_address(ip)
    except ValueError:
        return False
    else:
        return True


def errors(errorIndication, errorStatus, errorIndex, ip, file):
```

```
    #  обработка ошибок в случае ошибок возвращаем False и пишем в
файл file
    if errorIndication:
        txt_field.insert(INSERT, str(errorIndication) + 'ip address
' + ip + '\n')
        file.write(datetime.strftime(datetime.now(), "%Y.%m.%d
%H:%M:%S") + ' : ' + str(
            errorIndication) + ' = ip address = ' + ip + '\n')
        return False
    elif errorStatus:
        print(datetime.strftime(datetime.now(), "%Y.%m.%d
%H:%M:%S") + ' : ' + '%s at %s' % (
            errorStatus.prettyPrint(),
            errorIndex and varBinds[int(errorIndex) - 1][0] or
'?'))
        file.write(datetime.strftime(datetime.now(), "%Y.%m.%d
%H:%M:%S") + ' : ' + '%s at %s' % (
            errorStatus.prettyPrint(),
            errorIndex and varBinds[int(errorIndex) - 1][0] or '?'
+ '\n'))
        return False
    else:
        return True


window = tk.Tk()
window.title("Аналіз мережі")
btn_open = tk.Button(text="Відкрити", command=get_from_file)
lbl_path = tk.Label(text="Path")
txt_field = tk.Text()
btn_open.pack()
lbl_path.pack()
txt_field.pack()
window.mainloop()
```