Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
“____” _____ 2025 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему
“Дослідження та програмна реалізація системи контролю Інтернет шлюзів на базі ОС Ubuntu”

Виконав здобувач вищої освіти
ІІ курсу, групи КІ-24М
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Сосна О.С.
«____» _____ 2025 р.

Керівник проекту
доктор філософії (PhD)
_____ Усік П.С.
«____» _____ 2025 р.
Рецензент _____
_____

м. Кропивницький

АНОТАЦІЯ

Сосна О.С. Дослідження та програмна реалізація системи контролю Інтернет шлюзів на базі ОС Ubuntu. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2025.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи контролю Інтернет шлюзів на базі ОС Ubuntu.

Метою розробки є дослідження та програмна реалізація системи контролю Інтернет шлюзів на базі ОС Ubuntu.

Об'єктом дослідження є процес контролю Інтернет шлюзів на базі ОС Ubuntu.

Предметом дослідження є методи контролю Інтернет шлюзів на базі ОС Ubuntu.

Методи дослідження базуються на методах захисту інформації у комп'ютерних мережах, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи контролю Інтернет шлюзів на базі ОС Ubuntu.

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

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

Програма може використовуватися на ПЕОМ з ОС Ubuntu.

Програму розроблено в середовищі PHP, PERL.

Ключові слова: комп'ютерна інженерія, контроль Інтернет шлюзів, ОС Ubuntu

ABSTRACT

Sosna O.S. Research and software implementation of the Internet gateway control system based on the OS Ubuntu. 123 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2025.

In this final qualification work for the second (master's) level of higher education, software has been developed, which is intended for the Internet gateway control system based on the OS Ubuntu.

The purpose of the development is the research and software implementation of the Internet gateway control system based on the OS Ubuntu.

The object of the research is the process of controlling Internet gateways based on the OS Ubuntu.

The subject of the research is methods of controlling Internet gateways based on the OS Ubuntu.

The research methods are based on methods of information protection in computer networks, methods of mathematical statistics, methods of software development.

The result of the work is the software implementation of the Internet gateway control system based on the OS Ubuntu.

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

A user-friendly user interface has been developed. Instructions for working with the software are provided.

The program can be used on a PC with OS Ubuntu.

The program was developed in PHP, PERL.

Keywords: computer engineering, Internet gateway control, OS Ubuntu

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ 3
ВСТУП..... 4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ 6
1.1 Призначення системи..... 6
1.2 Область застосування..... 6
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ 8
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти..... 8
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування..... 21
2.3 Розгорнута постановка завдання 25
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ 27
3.1 Опис функціонування системи 27
3.2 Розробка структурної схеми..... 30
3.3 Розробка функціональної схеми 35
3.4 Розробка діаграми процесів..... 37
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ..... 39
4.1 Розробка блок-схем та опис алгоритмів функціонування системи..... 39
4.2 Захист розробленого програмного забезпечення..... 54
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ..... 58
6 НАУКОВА НОВИЗНА 62

7 МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ІТ-ПРОЄКТУ 63
7.1 Визначення цільової аудиторії кінцевого готового продукту 63
7.2 Оцінка привабливості шляхом застосування методів експертних оцінок ... 64
7.3 Вибір методу оцінки вартості ПЗ 64
7.4 Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості 65
7.5 Пропозиція алгоритму просування проєкту розробки ПЗ 67
7.6 Оптимізація каналів збуту та шляхів реалізації ПЗ 67
7.7 Визначення ключових факторів успіху конкретного проєкту 68
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ 69
8.1 Вступ 69
8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером 70
8.3 Аналіз умов праці на робочому місці програміста 71
8.4 Розрахункова частина 74
8.5 Висновки до розділу 76
9 ОСНОВНІ ВИСНОВКИ 78
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 80

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

HTML (HyperText Murkup Language) – мова розмітки гіпертекстових документів;

СУБД – системи управління базами даних;

TCP/IP (Transport Control Protocol/Internet Protocol) – протокол роботи мережі інтернет;

ГМ – глобальні мережі;

ЛВМ – локальна віртуальна мережа;

ОД – оконечне обладнання даних;

ОС – операційна система;

ПК – персональний комп'ютер;

ОЗП – основний запам'ятовуючий пристрій;

URL – universal resource locator – локатор ресурсів інтернет;

PPTP – Point-to-Point-Tunneling Protocol – протокол створення захищеного каналу при доступі віддалених користувачів через публічні мережі;

ВСЗІП – віртуальна система захисту інформаційного потоку;

ВСПІП – віртуальні системи поділу інформаційних потоків;

ВДТ – відеодисплейний термінал;

ЦО – цивільна оборона.

ПЕОМ – персональна електронно обчислювальна машина;

PHP – створює HTML-сторінки з використанням спеціальних тегів, розпізнаваних аналізатором PHP;

WWW – world wide web – всесвітня мережа інтернет;

FTP – File Transport Protocol – протокол передачі файлів.

ВСТУП

Актуальність теми. У світі мереж шлюз за замовчуванням відіграє ключову роль у спрямуванні мережевого трафіку між вашою локальною мережею та ширшим Інтернетом. Ubuntu 20, як популярний дистрибутив Linux, дозволяє користувачам ефективно налаштовувати та керувати своїми мережевими параметрами. Дана робота проведе вас через процес зміни шлюзу за замовчуванням в Ubuntu 20, забезпечуючи вам повний контроль над вашим мережевим трафіком.

Перш ніж заглибитися в процес, важливо зрозуміти, що таке шлюз за замовчуванням. Шлюз за замовчуванням – це IP-адреса маршрутизатора, який з'єднує вашу локальну мережу з ширшим Інтернетом. Коли пристрою у вашій локальній мережі потрібно зв'язатися з пристроєм в іншій мережі, він надсилає дані до шлюзу за замовчуванням, який потім пересилає їх до потрібного пункту призначення.

Перш ніж почати, переконайтеся, що у вас є такі передумови:

– Система Ubuntu 20 працює.

– Адміністративний доступ до системи.

– IP-адреса нового шлюзу за замовчуванням.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи контролю Інтернет шлюзів на базі ОС Ubuntu.

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

– Огляд існуючих систем контролю Інтернет шлюзів на базі ОС Ubuntu.

– Дослідження системи контролю Інтернет шлюзів на базі ОС Ubuntu.

– Програмна реалізація системи контролю Інтернет шлюзів на базі ОС Ubuntu.

Об'єктом дослідження є процес контролю Інтернет шлюзів на базі ОС Ubuntu.

Предметом дослідження є методи контролю Інтернет шлюзів на базі ОС Ubuntu.

Методи дослідження базуються на методах захисту інформації у комп'ютерних мережах, методах математичної статистики, методах розробки програмного забезпечення.

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

– Удосконалено метод контролю Інтернет шлюзів на базі ОС Ubuntu.

– Розроблено вітчизняний продукт контролю Інтернет шлюзів на базі ОС Ubuntu, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі контролю Інтернет шлюзів на базі ОС Ubuntu.

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 LVII Науково-технічній конференції здобувачів вищої освіти LV науково-технічної конференції «Наука в ЦНТУ: основні досягнення та перспективи розвитку» (2025 р.), основні положення випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти надруковані у статті збірника праць молодих науковців ЦНТУ, випуск №15.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи контролю Інтернет шлюзів на базі ОС Ubuntu, 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

Брандмауер вашого маршрутизатора – це ваша перша лінія захисту від шпигунів за даними та кіберзлочинців. Він стежить за будь-яким підозрілим трафіком у вашій мережі – принаймні теоретично. Але брандмауери маршрутизаторів зазвичай не справляються з двома ключовими завданнями. Вони не можуть виявити незвичайний вихідний трафік. Уявіть собі троянців, які вже влаштувалися на вашому ноутбуці та зв’язуються з центром управління польотами. І вони не можуть допомогти, коли ви перебуваєте поза домом і користуєтеся сумнівним громадським Wi-Fi.

Для повної безпеки ми рекомендуємо додати додатковий рівень захисту до ваших пристроїв: якісне антивірусне програмне забезпечення з брандмауером.

1.2 Область застосування

Варіанти використання шлюзів Інтернету речей на базі Ubuntu:

1. Розумні будинки: Ubuntu може бути основою шлюзу розумного дому, керуючи такими пристроями, як розумне освітлення, термостати та системи безпеки, одночасно забезпечуючи дистанційне керування та автоматизацію.

2. Промисловий Інтернет речей: У виробництві шлюз на базі Ubuntu може контролювати стан обладнання, оптимізувати виробничі процеси та аналізувати дані в режимі реального часу для прогнозного обслуговування.

3. Сільське господарство: Фермери можуть використовувати шлюзи Інтернету речей, побудовані на Ubuntu, для збору даних з датчиків, які контролюють стан ґрунту, погоду та здоров'я врожаю, підвищуючи продуктивність сільського господарства.

4. Охорона здоров'я: Шлюзи Інтернету речей можуть сприяти дистанційному моніторингу пацієнтів, збору та передачі даних про здоров'я медичним працівникам, покращуючи результати лікування пацієнтів.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи контролю Інтернет шлюзів на базі ОС Ubuntu, 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

Norton Antivirus – найкращий брандмауер для Android-пристроїв

Пакети Norton 360 з «Smart Firewall» орієнтовані на домогосподарства, які хочуть забезпечити повну онлайн-безпеку, не потребуючи зайвих зусиль для налаштування. Преміум-тарифи Norton, вартість яких починається від 39,99 доларів США за перший рік, також добре підходять для тих, хто планує перевантажитися передплатою, оскільки вони поєднують антивірус, VPN, батьківський контроль і менеджер паролів.

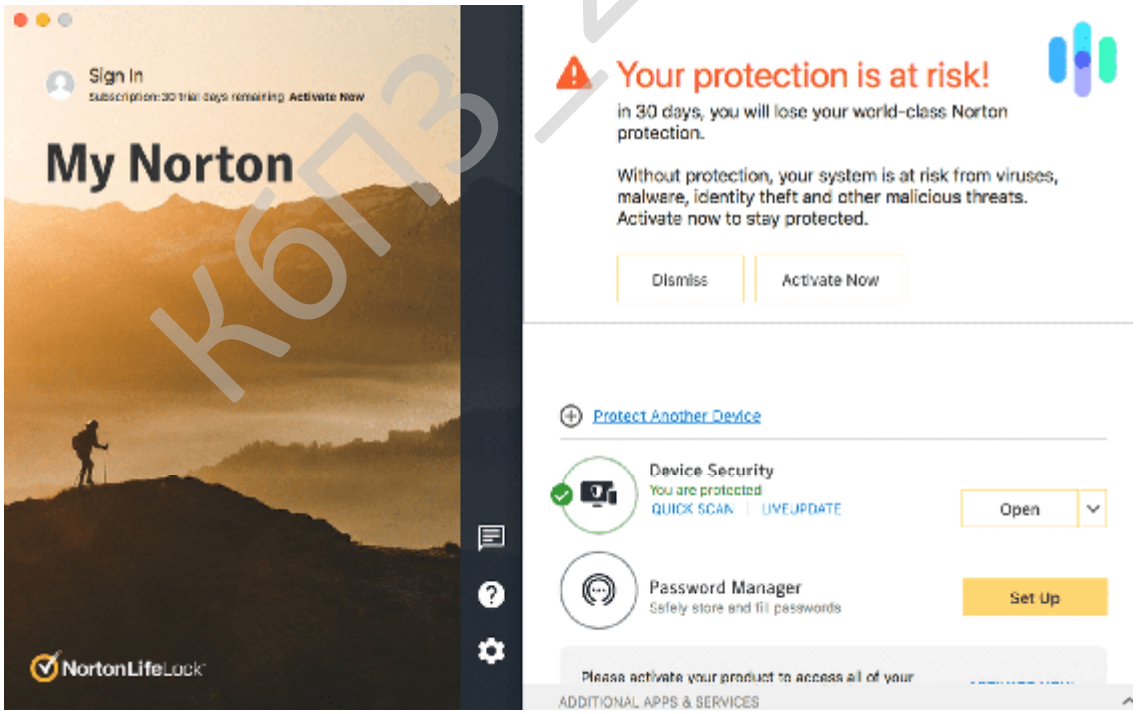

Рисунок 2.1 – Інтерфейс користувача Norton Antivirus

Norton також пропонує сімейні плани з батьківським контролем та захистом до 10 пристроїв. Якщо у вас сім'я з чотирьох або більше осіб, ми вважаємо цей варіант найкращим.

Переваги:

– Відмінний дизайн приладової панелі.
– Надійні інструменти захисту від шкідливих програм.
– Захист для до 10 пристроїв.
– Тарифні плани за розумною ціною.

Недоліки:

– Без безкоштовного плану.
– Без шредера для файлів.
– Глючне розширення браузера Chrome.

Функції брандмауера

«Розумний брандмауер» Norton – це більше, ніж просто мережа для збору шкідливих програм. Він неявно знає, як поводиться шкідливе програмне забезпечення, тому може виявляти шкідливу поведінку та блокувати джерело, навіть якщо раніше не бачив конкретної помилки. Ця розширена функція називається виявленням на основі поведінки. Нам подобається, коли вона працює у фоновому режимі, оскільки не кожен вірус, з яким ми стикаємося, буде у списку «найбільш розшукуваних» баз даних шкідливих програм. Деякі з них будуть новими, непомітними атаками, такими як експлойти нульового дня.

Інтелектуальний брандмауер також означає більш плавний щоденний досвід, оскільки Norton може розпізнавати безпечний трафік. Іншими словами, Norton не повинен запитувати нашого схвалення щоразу, коли наш додаток для продуктивності запитує вихідне з'єднання. Щоб отримати цей прогнозний захист, ми просто ввімкнули «Інтелектуальний режим» і дозволили йому виконувати свою роботу.

Щоб переглянути детальні дані про підключення окремих програм до Norton, просто натисніть на програму на вкладці моніторингу трафіку (Безпека >

Розумний брандмауер). Norton у режимі реального часу покаже вам, куди саме спрямовується ваш трафік і скільки даних він споживає. Ви можете бути шоковані.

Захист від шкідливого програмного забезпечення

Захист від шкідливого програмного забезпечення – це головна причина роботи Norton, тому ми не здивувалися, побачивши, як Norton долає наші симульовані вірусні атаки та спроби доступу до шахрайських веб-сайтів. В останньому випадку Norton не лише видав нам попередження, а й надав детальні пояснення того, яких типів шахрайства очікувати. Обидві ці функції є частиною служби Safe Web від Norton і обидві є інтелектуальними, як брандмауер Norton. Вони позначають підозрілу поведінку, а не лише підозрілі URL-адреси чи імена файлів.

Norton пропонував захист у режимі реального часу, який стежив за шкідливими вкладеннями електронної пошти. Крім того, Norton також забезпечив нам планове глибоке сканування системи, захист наших зовнішніх дисків (приємна перевага) та можливість оновлення вірусних баз даних на вимогу. Остання функція зазвичай не потрібна, але її добре мати, якщо на волі з'явився вірус нульового дня.

Для вашої інформації: у нас була можливість виключити певні файли або програми з захисту Norton у режимі реального часу. Теоретично це може бути корисним, якщо Norton неодноразово помилково позначає безпечну програму або процес як підозрілий. Однак на практиці ми рекомендуємо дозволити Norton самостійно здійснювати ці дії.

Вартість

Norton – не найдешевший антивірус. Пакети Surfshark One, які коштують 2,69 долара на місяць протягом 27 місяців, тут найкращі. Існують якісні безкоштовні антивіруси. Але Norton – це, безумовно, найповніший, готовий до використання варіант безпеки з брандмауером, який ми тестували. Навіть після поновлення підписки варіант Norton 360 Deluxe коштує розумних 8,69 долара на

місяць, що менше, ніж у багатьох якісних антивірусів із пакетами VPN, які ми розглядали.

Що стосується опцій Norton, ми виявили, що їхній тарифний план 360 Deluxe (39 доларів за перший рік, 105 доларів при продовженні) є найкращим за співвідношенням ціни та якості. Наша підписка включала повний антивірус, VPN, менеджер паролів, батьківський контроль та 10 ГБ безпечного хмарного сховища. Однак для великих сімей може бути доцільним обрати трохи дорожчий тарифний план «Преміум», оскільки на 10 доларів більше на місяць (54,99 доларів за перший рік і 144,99 доларів за наступні) ви отримуєте захист до 10 пристроїв.

Norton пропонує семиденний безкоштовний пробний період для всіх своїх преміум-планів. І на відміну від деяких сервісів, якими ми користувалися раніше, скасування підписки на Norton було безпроблемним.

Антивірус McAfee – найкращий захист системи

Плани McAfee Total Protection, ціна яких починається від 39,99 доларів США, переповнені функціями безпеки, включаючи досить простий брандмауер із детальнішим контролем для досвідчених користувачів. Захист McAfee є агресивним. Спочатку він захистив нашу «домашню» мережу брандмауером і навіть виявив потенційно шкідливий дворічний файл, захований глибоко в нашій бібліотеці, під час першого сканування. Якщо вам потрібен саме такий захист, McAfee пропонує кілька планів, на які варто звернути увагу: дорожчі варіанти «Преміум» та «Розширений» із безпекою для необмеженої кількості пристроїв.

Переваги:

– Сканування загроз у режимі реального часу.
– Ретельне сканування системи.
– Розширені параметри брандмауера для експертів.
– 30-денна пробна версія.

Недоліки:

– Брандмауер заважав інтернету.
– Сканування системи відбувається повільно.

– Незграбніший користувацький досвід.

Коли ми кажемо, що McAfee ретельно сканує систему, ми маємо на увазі саме це. Наше перше сканування тривало понад п'ять годин. З позитивного боку, McAfee все ж таки виявив деякі давно приховані потенційні загрози для нас.

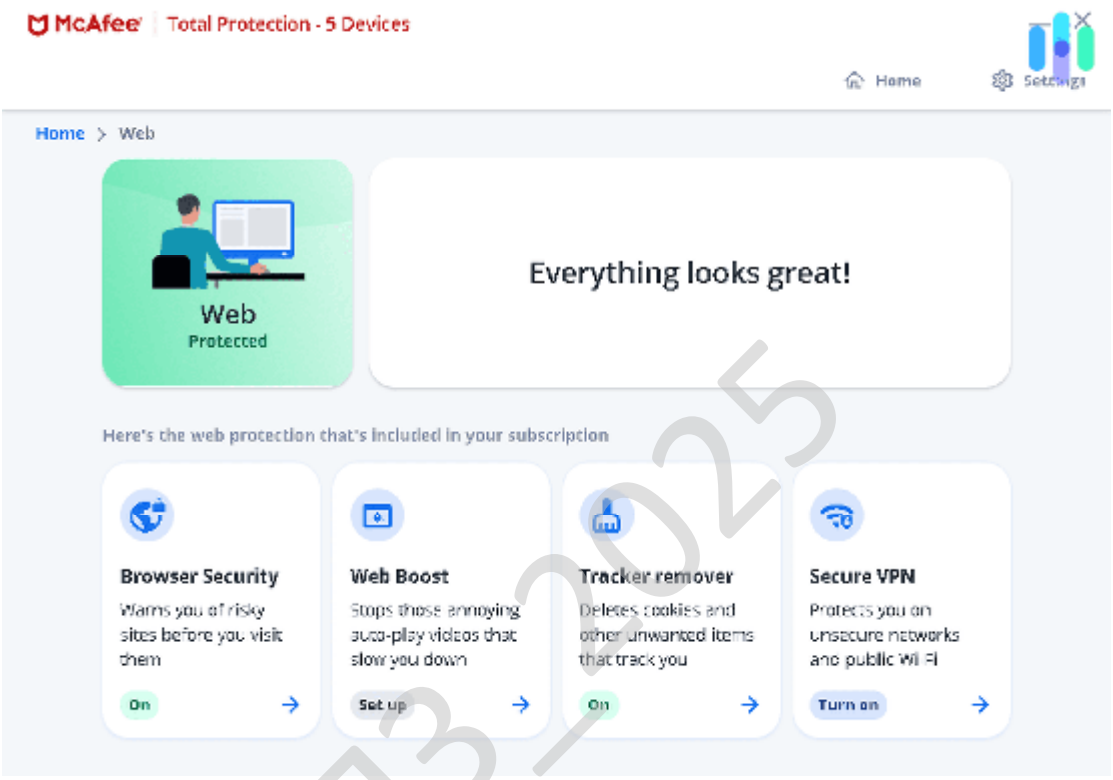

Рисунок 2.2 – Інтерфейс користувача McAfee

Функції брандмауера

Брандмауер McAfee дещо дивний тим, що McAfee працює як другий гейткіпер поряд із вбудованим брандмауером Windows. McAfee захищає вихідний трафік, тоді як Захисник Windows захищає вхідний. Зокрема, McAfee визначає, чи є вихідні запити, що надходять з наших пристроїв, підозрілими. Для цього, як і Norton, він використовує комбінацію перевірки вірусних визначень та моніторингу поведінки. Якщо є підозра на неправомірну дію, McAfee позначає запит для нашого схвалення.

Як ми вже згадували вище, McAfee схильна помилятися в плані безпеки,

що робить щоденні операції трохи більш практичними, ніж ми звикли з такими низькообслуговуваними програмами, як Norton. Так само два брандмауери, що працюють одночасно, один над одним, не були найкращим рецептом для ефективності процесора. Однак, з двома передовими брандмауерами, які контролюють наш трафік, вірусам доведеться працювати понаднормово, щоб проникнути на наш ПК.

Брандмауер McAfee для Mac був окремою функцією з розширеними налаштуваннями, яка дозволяла нам точно налаштовувати підключення портів. Можливо, це зменшило споживання системних ресурсів, але це не дуже корисно, якщо ви новачок у мережах.

Захист від шкідливого програмного забезпечення

Захист від шкідливого програмного забезпечення McAfee є більш простим. Щойно ми ввімкнули сканування в режимі реального часу, McAfee став більш-менш безшумним супутником. Хоча McAfee адекватно показав себе в наших внутрішніх тестах, споживча організація AV-Test повідомила про майже бездоганний рівень виявлення у своєму останньому дослідженні.

Для максимального захисту ми також встановили розширення для браузера McAfee WebAdvisor, яке перевіряє наявність шкідливих сайтів під час перегляду. Як не дивно, McAfee, який працює в парі з Microsoft над своїм брандмауером, не має розширення для Bing, що може бути проблемою для прихильників екосистеми Microsoft.

McAfee дозволив нам планувати глибоке сканування системи щодня, що ми рекомендуємо. Ми запускали їх вночі, коли нас не було поруч, бо це займало деякий час.

Вартість

Ціни McAfee певною мірою повторюють ціни Norton. Існує варіант нижчого рівня за 29,99 доларів США, «Базовий», для одного пристрою, якого, ймовірно, буде недостатньо. Що стосується пакету вище, то він коштує 39,99 доларів США та захищає п’ять пристроїв, тому потенційно добре підійде для

невеликих сімей. Обидва ці плани містять повноцінний антивірус, VPN та менеджер паролів.

Хоча Norton пропонує трохи дорожчий сімейний план на 10 пристроїв за 54,99 долара, план McAfee «Преміум» з необмеженою кількістю пристроїв піднімається ліфтом аж до пентхауса за 89,99 долара. І щоб було зрозуміло, ціни, які ми називали, є вступними пропозиціями. План McAfee вартістю 90 доларів стане майже 200 доларів після поновлення. Чи варто воно того? Можливо, варто – якщо ви також шукаєте якісний захист від крадіжки особистих даних, який пропонує McAfee, або у вас сім'я з чотирьох або більше осіб.

Ми були раді бачити, що McAfee пропонує щедру 30-денну безкоштовну пробну версію. Однак, за нашим досвідом, вимкнення автоматичного поновлення може бути складним, якщо не зовсім неможливим. Це був би аспект взаємодії з користувачем, який ми б одразу виправили, якби були McAfee.

Bitdefender – найкращий варіант для досвідчених користувачів

Bitdefender – це найкращий антивірус, який добре показав себе як у наших внутрішніх тестах, так і в незалежних лабораторних тестах. Bitdefender, унікальний серед антивірусів у цьому списку, також має дуже хорошу безкоштовну версію, яка надала нам усі основні функції безпеки. Однак, щоб розблокувати інші корисні функції, такі як брандмауер, VPN, батьківський контроль та низку розширених функцій, нам довелося перейти на платний план, починаючи з $59,99 за перший рік.

Переваги:

– Безкоштовна версія з основними функціями.
– Широко настроюваний брандмауер.
– Добре показав себе в тестах.
– Розширений захист від програм-вимагачів.

Недоліки:

– Немає брандмауера для Mac.
– Може бути заплутано для початківців.

– Дорожче, якщо вам потрібен безлімітний VPN.

Під час тестування Bitdefender для Mac ми не виявили брандмауера. Це не випадково. Для Mac немає брандмауера, тому, якщо ви шукаєте комплексне рішення безпеки для свого нового M3, ми рекомендуємо дотримуватися нашого першого вибору – Norton. Вам також варто ознайомитися з нашим найкращим антивірусним програмним забезпеченням для Mac.

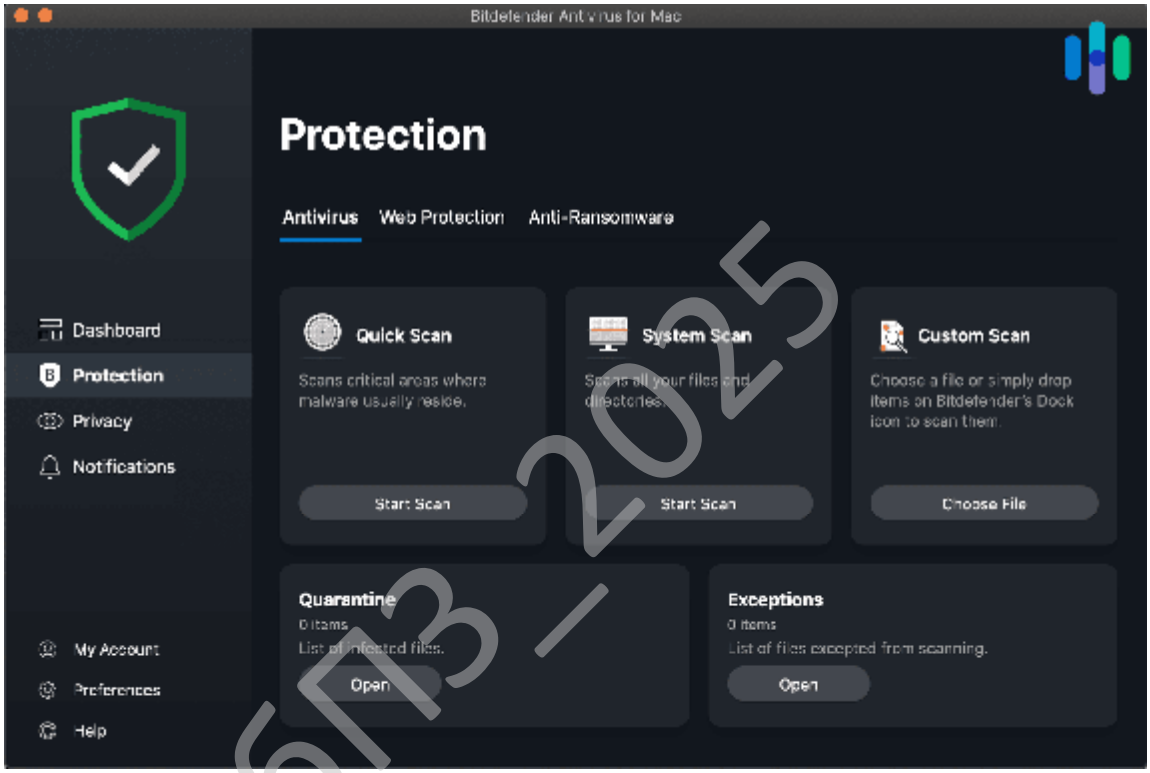

Рисунок 2.3 – Інтерфейс користувача Bitdefender

Функції брандмауера

Брандмауер Bitdefender має значно більше можливостей, ніж Norton чи McAfee. Для більшості користувачів достатньо буде ввімкнути динамічний мережевий адаптер Bitdefender під час підключення до нової мережі та ввімкнути прихований режим під час доступу до публічного Wi-Fi. Перший спосіб дозволяє Bitdefender визначити, чи потрібен певній мережі суворіший чи слабкий захист. Другий спосіб приховує вашу IP-адресу від сканування публічних мереж. В

іншому випадку, якщо налаштування DHCP-серверів не є тим, чим ви займаєтесь у вільний час, ми рекомендуємо не вмикати брандмауер.

Контроль місій Bitdefender для Mac: Брандмауер не входить до комплекту.

Якщо ви хочете ретельно контролювати весь трафік, що надходить і виходить з вашого пристрою, Bitdefender дозволяє вам це зробити. У розширених налаштуваннях увімкніть «Режим сповіщення». Просто будьте готові до великої кількості сповіщень.

Захист від шкідливого програмного забезпечення

Брандмауер Bitdefender працював разом з кількома іншими антивірусними функціями, що дублювали один одного, щоб захистити наші пристрої від помилок. Наприклад, «Advanced Threat Defense» стежив за підозрілою поведінкою програм, тоді як «Online Threat Prevention» контролював наш мережевий трафік.

Преміум-підписки Bitdefender також захищають від фішингу, криптомайнінгу та програм-вимагачів. Інструмент Bitdefender «Виправлення програм-вимагачів» особливо складний тим, що якщо він підозрює підозрілу активність на вашому пристрої, він запускає резервні копії критично важливих файлів для безпечного зберігання, доки загрозу не буде усунено. І це не кінець захисту Bitdefender.

Для вашої інформації: функція «Safepay» від Bitdefender додала додатковий рівень безпеки до наших онлайн-платежів. Замість введення конфіденційних даних у браузер, ми мали приватний браузер і клавіатуру для введення даних кредитної картки для захисту від шпигунських програм і кейлогерів. Саме це ми називаємо безпекою.

Вартість

Bitdefender пропонує безліч тарифних планів, починаючи з потужного, але спрощеного безкоштовного антивіруса. Їхній найдорожчий план, «Ultimate Security» за 89,99 доларів, постачається із захистом ідентифікаційних даних. Ми знайшли найкраще співвідношення ціни та якості в планах середнього рівня Bitdefender – «Total Security» (59,99 доларів за перший рік і 99 доларів за

наступний рік) та «Premium Security» (79,99 доларів за перший рік і 259,99 доларів за наступний рік) – хоча обидва дорожчі за аналогічні плани Norton і McAfee. І лише план «Premium Security» за 79,99 доларів постачається з необмеженим VPN.

На захист Bitdefender, якщо вам потрібен лише базовий антивірус, вони пропонують надійний безкоштовний план. А потужніші пакети Bitdefender пропонують більше захисту, ніж будь-який інший план у цьому списку.

Спеціальна сімейна пропозиція: Для сімей з безліччю пристроїв завжди є «Сімейний план» Bitdefender за $69,99. Він захищає 15 пристроїв і постачається з повним антивірусом і батьківським контролем. Однак VPN, знову ж таки, обмежений лише 200 МБ на день.

Методологія: Як ми тестуємо брандмауери

Брандмауери та антивірусні технології є одними з продуктів, які ми тестуємо, і які є найскладнішими та найважливішими для нашої цифрової безпеки. Щоб дати правильні рекомендації, наші експерти протестували майже десяток варіантів на наших персональних комп'ютерах, щоб побачити, які з них пропонують найкорисніші функції брандмауера.

Наші тести тривали понад місяць, протягом якого ми звертали увагу на ці три фактори, які мають вирішальне значення для вибору найкращого антивірусного програмного забезпечення з брандмауером:

– Захист від шкідливого програмного забезпечення: Виявлення шкідливого програмного забезпечення на рівні брандмауера має вирішальне значення для забезпечення безпеки не лише ваших пристроїв, але й усієї мережі. Ми шукали варіанти, що пропонують виявлення на основі поведінки, а не брандмауери, які шукають лише штами шкідливого коду, які вони зчитують зі списку.

– Досвід користувача: Хоча функціональність важлива, досвід користувача також важливий. Ми шукали антивіруси з брандмауерами, які також пропонують інтуїтивно зрозумілу панель інструментів та налаштовувані параметри, які ви можете налаштувати відповідно до своїх потреб.

– Бюджет: Нарешті, ми шукали варіанти, які не спустошать ваш гаманець. Антивіруси з брандмауерами не найдешевші, але це не означає, що вам потрібно витрачати на них сотні доларів щороку. Наші найкращі рекомендації мають початкові ціни від 39 до 89 доларів на перший рік, але вони часто включають додаткові послуги, такі як VPN.

Для вашої інформації: Ми часто використовуємо віртуальні приватні мережі (VPN), оскільки низка провідних антивірусних сервісів тепер включають їх до своїх пакетів безпеки. Не знаєте, що робить цей важливий елемент онлайн-захисту? Перегляньте наш нещодавно оновлений посібник покупця VPN для короткого огляду.

Чи достатньо вбудованого брандмауера мого пристрою?

Це насправді два питання. Чи достатньо брандмауера мого маршрутизатора? І чи достатньо брандмауера мого ноутбука?

Щоб відповісти на перше запитання, брандмауер, який забезпечує ваш маршрутизатор, може добре блокувати вхідний трафік. Але якщо ви заразитеся через фішингову шахрайську схему (і це один дуже поширений випадок), і вірус візьме контроль над вашим ноутбуком, а потім прорве брандмауер в іншому напрямку, ні ви, ні ваш брандмауер про це не дізнаєтесь. Існують сотні інших випадків, коли брандмауер на базі маршрутизатора не витримає.

Переходячи до другого питання, деякі наші пристрої оснащені брандмауерами. Вони є як на комп’ютерах з Windows, так і на Mac, але, як і ваш маршрутизатор, вони обробляють лише вхідні з’єднання. Тож, знову ж таки, ви будете в безпеці для будь-якої шкідливої програми, яка випадково проникла на ваш пристрій (включаючи справжніх комп’ютерних черв’яків). З іншого боку, якісний брандмауер з антивірусним компонентом а) виявить шахрайську програму, яка втручається у вашу систему, та б) виявить будь-які підозрілі вихідні запити, які вона надіслала.

Android та iPhone не мають брандмауера, головним чином тому, що мобільні пристрої не обробляють вхідні з’єднання з інших пристроїв, як це

роблять ноутбуки та настільні комп’ютери.

Чи все антивірусне програмне забезпечення постачається з брандмауером?

Багато сервісів мають цю функцію, але не всі. Наприклад, безкоштовні версії Bitdefender та Avira не мають брандмауерів. Malwarebytes – ще один найкращий антивірус без брандмауера. Ось наш повний огляд Malwarebytes. Навіть деякі з найкращих антивірусних програм, такі як TotalAV та Surfshark Antivirus, не мають цієї важливої функції.

Антивіруси без брандмауерів не обов'язково є гіршими. Ці антивіруси покладаються на подібне виявлення на основі поведінки та сканування хмарних ресурсів для захисту наших пристроїв, але замість моніторингу мережевих з'єднань вони постійно сканують пристрої на наявність аномальної активності.

Один зі способів відрізнити брандмауер від чистого антивіруса полягає в тому, що хоча брандмауер працює лише з підключенням до Інтернету, антивірус може захистити ваші пристрої без нього.

Які функції повинен мати мій брандмауер?

Незалежно від того, який антивірус ви оберете, ваш брандмауер повинен мати функцію виявлення вірусів на основі поведінки. Якщо ваш антивірус має інтелектуальний брандмауер, такий як Norton, він не просто шукає код, ідентифікований як шкідливий, а й підозрілі закономірності активності. Наприклад, якщо завантажений вами додаток починає зв’язуватися з підозрілими IP-адресами через порт, який ви зазвичай не використовуєте.

Нам також подобаються брандмауери з моніторами трафіку. Таким чином, ми можемо переглядати запити, які роблять наші програми, у режимі реального часу, а також мати запис про з’єднання, які ми блокували або схвалювали в минулому.

Зрештою, нам подобається мати хоча б певний контроль над агресивністю нашого брандмауера. Здебільшого ми дозволяємо брандмауеру керувати процесами, але бувають випадки, коли може знадобитися ручне блокування

програм або процесів. Якщо ви коли-небудь безуспішно намагалися зупинити автоматичне оновлення програми, ви точно знаєте, про що ми говоримо.

Якщо ви завантажуєте нову програму, щодо якої ви не впевнені, монітор трафіку брандмауера – чудовий спосіб її перевірити. Просто перегляньте журнали брандмауера. Якщо ви помітите, що ваше нове програмне забезпечення здійснює купу вихідних підключень до таких доменів, як adtracker.com, ви знатимете, що вас викрадають.

Підсумки

Програмні брандмауери, мабуть, останнє, про що ми думаємо, захищаючи наші пристрої від зловмисників. Зрештою, у нас є VPN, антивірус, блокувальники реклами тощо. І більшість із нас все одно мають вбудований брандмауер у маршрутизаторах і ноутбуках. Тож навіщо такий додатковий рівень безпеки? Все просто: вихідні з'єднання. У поєднанні з нашим антивірусом брандмауер може блокувати підозрілі запити. Ваш власний брандмауер – ні.

Завдяки простоті використання, ціні та безпеці, «Smart Firewall» від Norton перемагає беззаперечно. Менш ніж за дев'ять доларів на місяць при поновленні ви можете захистити п'ять пристроїв за допомогою VPN, батьківського контролю та 50 ГБ сховища. Крім того, ви можете використовувати пристрій Windows або Mac, Android або iOS.

Якщо вам потрібен трохи більше контролю та більш агресивний підхід, радимо звернути увагу на McAfee. Вартість планів середнього рівня приблизно однакова. І останнє, але не менш важливе: Bitdefender – чудовий вибір для більш досвідчених користувачів, які шукають максимального захисту. Просто спочатку переконайтеся, що у вас є ПК; Bitdefender не має брандмауера для Mac.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

PHP

Для написання програмного забезпечення була обрана мова програмування PHP.

PHP – це мова серверних скриптів (server scripting language), що вбудовується в HTML, який інтерпретується і виконується на сервері. PHP працює як частина Web-сервера, і цим самим схожий на ASP від Microsoft або Coldfusion від Allaire. Синтаксис PHP дуже схожий на синтаксис таких мов програмування C або Perl. Люди, що мають деякий досвід програмування, дуже швидко зможуть почати писати програми на PHP. У цій мові немає строгої типізації даних і немає необхідності в діях з виділення/звільнення пам'яті. Програми, написані на мові програмування PHP, читаються достатньо легко. На відміну від Perl-программ PHP-коду властива легка читабельність та зрозумілість.

PHP є препроцесором HTML.

До того, як сервер відправить файл браузеру, його проглядає препроцесор-інтерпретатор. Для того, щоб це відбувалося, файли, які піддаються обробці препроцесором, повинні мати визначене розширення (звичайно це.phtml або.php3, але ці значення можна поміняти) і містити (хоча це не обов'язкова вимога) код для препроцесора. Перед відправкою сторінки PHP-код програється на сервері і браузеру видається результат у вигляді знову таки HTML-сторінки, яка може сильно відрізнятися від тієї, що зберігається на сервері. Звичайні ж сторінки, що мають розширення.html/.htm Web-сервер буде відправляти браузеру без будь-якої обробки. Основна відмінність від CGI-скриптів, написаних на інших мовах, типу Perl або C – це те, що в CGI-програмах розробник самостійно пише HTML-код, що виводиться, а, використовуючи PHP – вбудовує свою програму-скрипт в готову HTML-сторінку, використовуючи відкриваючий і закриваючий теги (<?php та ?>). PHP називається мовою серверних скриптів на

відміну від Javascript/jscript/vbscript, які є мовами клієнтських скриптів. Це означає, що PHP-скрипт виконується на сервері, а клієнтові передається результат його роботи, тоді як в JavaScript, код повністю передається на клієнтську машину і лише там виконується браузером.

При написанні на інших мовах програмування, наприклад, на Perl або C – замість того, щоб створювати програму, яка займається формуванням HTML-коду і містить незліченну безліч призначених для цього команд, PHP-програміст має можливість створювати HTML-код з декількома упровадженими командами PHP. Код PHP відділяється спеціальними початковим і кінцевим тегами, які дозволяють процесору PHP визначати початок і кінець ділянки HTML-коду, що містить PHP-скрипт.

Значною відмінністю PHP від якого-небудь коду, що виконується на стороні клієнта, наприклад, Javascript, є те, що PHP-скрипти виконуються на сервері. PHP-скрипт розміщений на сервері, клієнт отримує тільки результат виконання скрипта, причому клієнт не має можливості з'ясувати, який саме код виконується. Існує також можливість конфігурувати власний сервер так, щоб HTML-файли оброблялися процесором PHP, так що клієнти навіть не в змозі дізнатися, чи отримують вони звичайний HTML-файл, чи це результат виконання скрипта.

Мова програмування PHP досить проста для освоєння, але разом з тим здатна задовольнити запити професійних програмістів. Хоча PHP, головним чином, призначений для роботи в середовищі web-серверів, область його застосування не обмежується тільки цим.

Можливості мови PHP дуже великі. Головним чином, область застосування PHP сфокусована на написання скриптів, що працюють на стороні сервера; таким чином, PHP здатний виконувати все те, що виконує будь-яка інша програма CGI, наприклад, обробляти дані форм, генерувати динамічні сторінки або посилати і приймати cookies. Окрім вищенаведеного PHP здатний виконувати і безліч інших завдань.

Існують такі основні області, де використовується PHP.

– Створення скриптів для виконання на стороні сервера. Саме таким чином мова програмування PHP найширше використовується. Все, що може знадобитися програмісту – це парсер PHP (у вигляді програми CGI або серверного модуля), вебсервер і браузер. Для того, щоб існувала можливість проглядати результати виконання PHP-скриптів у браузері, потрібен працюючий веб-сервер і встановлений PHP.

– Створення скриптів для виконання в командному рядку. Існує можливість створити PHP-скрипт, здатний запускатися незалежно від веб-серверу та браузеру. Все, що потрібно – це парсер PHP. Такий спосіб використання PHP ідеально підходить для скриптів, які повинні виконуватися регулярно, наприклад, за допомогою cron (на платформах Unix або Linux) або за допомогою планувальника завдань (Task Scheduler) на платформах Windows. Ці скрипти також можуть бути використані в завданнях простої обробки даних.

Існує величезна кількість документації і списків розсилки, до яких можна звернутися у разі виникнення яких-небудь питань.

Perl

Perl – високорівнева, інтерпретована, динамічна мова програмуваннязагального призначення. Perl запозичує можливості з багатьох інших мов програмування, як то C, shell scripting, AWK та sed. Мова надає потужні можливості для обробки тексту без довільних обмежень на довжину даних багатьох сучасних інструментів Unix, полегшуючи процес маніпуляції текстових файлів. Використовується для програмування графіки, системного адміністрування, у мережному програмуванні, у написанні програмного забезпечення, яке взаємодіє з базами даних, у програмуванні CGI для веб. Perl за свою гнучкість і потужність отримав прізвисько «швейцарського армійського ножа мов програмування».

Perl – мова програмування загального призначення, котра на початку розроблялась, як інструмент для обробки тексту, і тепер використається для

вирішення дуже широкого кола завдань, включно із системним адмініструванням, веб-розробкою, розробкою мережного програмного забезпечення, та програмного забезпечення з графічним інтерфейсом користувача.

Мова віддає перевагу практичному (простота у використанні, ефективність, повнота) над гарним (крихітність, елегантність, мінімалістичність). Вона має багато можливостей, включно із підтримкою кількох парадигм програмування(процедурне програмування, об'єктно-орієнтоване програмування, а також функціональне програмування), управлінням пам'яттю, вбудованою підтримкою системи обробки текстів, і величезною колекцією сторонніх модулів.

Загальна структура програм на мові Perl була запозичена у C. Perl є типовою процедурною мовою, зі змінними, виразами, присвоюваннями, дужками-розділювачами, блоками, циклами та підпрограмами.

Perl також запозичив деякі особливості з shell-програмування. Назви усіх змінних розпочинаються зі спеціальних знаків, кожен з яких характеризує певний тип даних (наприклад, скаляри, масиві та хеш-таблиці). Важливо зазначити, що це дозволяє правильно інтерпретувати назви змінних безпосередньо у текстових рядках. Perl має багато вбудованих функцій, які забезпечують програміста інструментарієм, котрий часто використовується у shell-програмуванні (хоча, багато з цих інструментів реалізовується за допомогою зовнішніх інструментів), як то сортування або виклик зовнішніх програм.

Perl запозичив списки з мови Lisp, асоціативні масиви (хеші) з AWK, і регулярні вирази з sed. Це спростило процес парсингу і обробки великих обсягів текстової інформації.

У Perl 5, була додана підтримка комплексних структур даних, функції першого класу, і об'єктно-орієнтованої моделі програмування. Що включає у себе посилання, пакунки, та інше. Серед найбільших нововведень у Perl 5 слід відмітити наявність модулів, які дозволяють повторно використовувати один і той же код.

Усі версії Perl здійснюють автоматичне типізування змінних і управління пам'яттю. Інтерпретатор знає про тип та способи зберігання усіх об'єктів даних у програмі.

Дизайн Perl можна описати, як відповідь на три основні тенденції в комп'ютерної індустрії: зниження вартості апаратного забезпечення, зростання вартості робочої сили, а також вдосконалення технології компіляторів. Багато ранніх мов програмування, як то Fortran та C, були розроблені таким чином, щоб якомога ефективніше використовувати на той час дороге апаратне забезпечення. У протилежність, Perl було розроблено для підвищення ефективності роботи дорогих в наш час програмістів.

Perl має багато можливостей, які збільшують ефективність програміста за рахунок інтенсивного використання мікропроцесора та великих обсягів оперативної пам'яті. Серед них: автоматичне управління пам'яттю; динамічна типізація; стрічки, списки, та хеші; регулярні вирази; самоаналіз; та функція eval().

Perl підтримує мовні конструкції, які є короткими і природними для людей при їхньому читанні і написанні, навіть якщо це ускладнює реалізацію інтерпретатора Perl.

Синтаксис Perl має багато спільного з синтаксисом мов Ci, Awk, Sed і Shell. Перший рядок початкового коду може починатися з «#!/Шлях/до/Perl [-ключі]» – що указує системі шлях до інтерпретатора Perl для виконання програми в *NIX системах і виконання їх на Веб-сервері.

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 контролю Інтернет шлюзів на базі ОС Ubuntu.

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

Зі зростанням Інтернету речей (IoT) потреба в надійних, гнучких та ефективних шлюзах стає дедалі важливішою. Шлюз IoT діє як міст між пристроями IoT та хмарою, забезпечуючи безперебійну передачу даних та інтелектуальну обробку. Однією з найкращих платформ для розробки та розгортання шлюзів IoT є Ubuntu, універсальна та широко використовувана операційна система з відкритим кодом. У цій статті розглядаються переваги використання Ubuntu для шлюзів IoT та те, як він може революціонізувати ваші рішення IoT.

Що таке шлюз Інтернету речей?

Шлюз Інтернету речей (IoT) – це пристрій або програмна платформа, яка забезпечує зв'язок між пристроями Інтернету речей та хмарними сервісами. Він збирає, обробляє та передає дані, підтримуючи різні протоколи та керуючи підключеннями пристроїв. Ефективний шлюз Інтернету речей забезпечує низьку затримку, підвищує безпеку та спрощує локальну обробку даних.

Чому варто обрати Ubuntu для шлюзів Інтернету речей?

1. Переваги відкритого коду

Будучи програмою з відкритим вихідним кодом, Ubuntu дозволяє розробникам налаштовувати та модифікувати ОС відповідно до потреб конкретного проекту. Ця гнучкість є критично важливою в IoT-додатках, де задіяні різноманітні пристрої та протоколи.

2. Багата екосистема

Ubuntu підтримує широкий спектр бібліотек, фреймворків та інструментів, спеціально розроблених для розробки Інтернету речей. Від Python та Node.js до бібліотек машинного навчання, розробники мають доступ до безлічі ресурсів.

3. Зручний інтерфейс

Ubuntu пропонує зручний інтерфейс, що спрощує розробникам налаштування та керування шлюзами, незалежно від того, чи використовують вони Ubuntu Server, чи Ubuntu Core, розроблені спеціально для IoT-застосунків.

4. Надійна підтримка громади

Маючи сильну спільноту розробників та користувачів, Ubuntu надає розширені ресурси, підтримку з усунення несправностей та документацію, що може бути безцінним під час подолання труднощів.

5. Функції безпеки

Ubuntu включає різні заходи безпеки, включаючи регулярні оновлення, вбудовані брандмауери та дозволи користувачів, які життєво важливі для захисту даних і пристроїв у середовищі Інтернету речей.

Налаштування шлюзу Інтернету речей за допомогою Ubuntu

Щоб проілюструвати, як налаштувати шлюз Інтернету речей за допомогою Ubuntu, давайте розглянемо кілька важливих кроків:

1. Виберіть свою версію Ubuntu

Залежно від ваших потреб, виберіть між Ubuntu Server та Ubuntu Core. Ubuntu Core особливо підходить для пристроїв Інтернету речей завдяки своїй легкості та системі керування пакетами (snap).

2. Встановлення

Завантажте вибрану версію Ubuntu та дотримуйтесь інструкцій з встановлення. Переконайтеся, що у вас є необхідне обладнання, яке відповідає специфікаціям ваших пристроїв Інтернету речей.

3. Налаштуйте параметри мережі

Налаштуйте мережеве з’єднання, щоб ваш шлюз міг взаємодіяти з пристроями Інтернету речей та Інтернетом. Це може включати налаштування параметрів Wi-Fi або Ethernet.

4. Встановлення IoT-фреймворків

Залежно від вимог вашого проекту, ви можете встановити різні IoT-

фреймворки, такі як:

Node-RED: Інструмент програмування на основі потоків для з'єднання пристроїв Інтернету речей.

MQTT: Легкий протокол обміну повідомленнями для невеликих датчиків та мобільних пристроїв, оптимізований для мереж з високою затримкою або ненадійних мереж.

Платформа Інтернету речей: Для візуалізації даних, зібраних з пристроїв Інтернету речей.

Розробка та розгортання додатків: Створюйте додатки, які використовують можливості обробки даних, аналітики та локального прийняття рішень. Python або JavaScript є популярним вибором для розробки цих додатків.

5. Захистіть свій шлюз

Впроваджуйте протоколи безпеки, такі як шифрування даних під час передачі та використання безпечних методів автентифікації, для захисту вашої екосистеми Інтернету речей.

Висновок

Використання Ubuntu для шлюзів Інтернету речей пропонує надійну, гнучку та безпечну платформу, яка може ефективно впоратися зі складнощами комунікації Інтернету речей. Завдяки відкритому коду, багатій екосистемі та сильній підтримці спільноти, Ubuntu є чудовим вибором для розробників, які прагнуть створювати інтелектуальні рішення Інтернету речей. Оскільки ландшафт Інтернету речей продовжує розвиватися, використання можливостей Ubuntu, безсумнівно, покращить підключення, ефективність та інновації в численних додатках.

Впроваджуючи та розгортаючи шлюзи Інтернету речей за допомогою Ubuntu, компанії та розробники можуть відкрити нові можливості та стимулювати майбутнє інтелектуальних технологій.

3.2 Розробка структурної схеми

Системи контролю Інтернет шлюзів на базі ОС Ubuntu – це за фактом міжмережевий екран (брандмауер), який працює під керуванням відповідної ОС. Брандмауери – це не щось привабливе. Вони рідко потрапляють у заголовки газет, а коли працюють добре, ви їх майже не помічаєте. Однак у 2025 році, коли ваш бізнес залежить від хмарних додатків, віддалених користувачів та постійно активних сервісів, скромний брандмауер все ще має велику вагу. Уявіть собі свою мережу як будівлю. У вас є двері, коридори, ліфти та постійний потік відвідувачів. Брандмауер – це стійка реєстрації та служба безпеки. Він впускає потрібних людей, не пускає не потрібних і виявляє дивну поведінку, перш ніж вона стане проблемою.

Що таке брандмауер?

По суті, брандмауер перевіряє трафік, який намагається увійти або вийти з вашого середовища, і застосовує встановлені вами правила. Ці правила можуть бути простими, наприклад, «дозволити цьому офісу доступ до цієї служби», або дуже специфічними, наприклад, «дозволити цьому користувачеві доступ до цієї програми лише в робочий час». Сучасні брандмауери йдуть далі. Вони розуміють програми та користувачів, а не лише IP-адреси та порти. Вони можуть переглядати трафік, щоб виявляти відомі загрози, підозрілі закономірності або ризикований контент. Вони ведуть детальні журнали, щоб ви могли довести, хто до чого мав доступ і коли.

Шифрування є ключовою частиною цієї історії. Більшість бізнес-трафіку зараз передається через TLS. Це чудово для конфіденційності, але також може приховувати атаки. За умови правильного проектування та політик брандмауер може розшифрувати трафік на периферії, застосовувати перевірки безпеки, а потім повторно зашифрувати його для подальшого передавання. Якщо все зроблено добре, користувач не помічає різниці, але ви усуваєте основну сліпу зону. Результатом є точка контролю, яка поєднує видимість із можливістю діяти в

режимі реального часу.

Реальні виклики 2025 року

Зловмисники швидкі, терплячі та організовані. Один фішинг може дати їм плацдарм. Звідти вони намагаються переміщатися по вашій мережі, знаходити цінні дані та непомітно викрадати їх, перш ніж з'явиться будь-яке повідомлення з вимогою викупу. Водночас ваші активи зростають. Співробітники підключаються до мережі як з дому, так і в дорозі. Партнери та постачальники інтегруються з вашими системами. Ви використовуєте поєднання хмарних та локальних сервісів. Кожне підключення – це ще один шлях, який може спробувати зловмисник.

Операції також перебувають під тиском. Команди зайняті, інструментів безпеки багато, а зміни накопичуються. Зі зміщенням акценту на безпеку до сучасніших рішень, таких як XDR, SSE, CNAPP та CTEM, брандмауери часто страждають від проблеми «встановив і забув». Правила швидко додаються для вирішення бізнес-проблеми, а потім ніколи не очищаються. З часом ви отримуєте захаращені політики, тіньові правила та широкі «тимчасові» дозволи. Це послаблює безпеку та може уповільнити продуктивність. Якщо сам брандмауер вийде з ладу, наслідки можуть бути негайними. Персонал не може отримати доступ до потрібних програм. Клієнти не можуть зв'язатися з вами. Для багатьох організацій збій брандмауера є збоєм у роботі бізнесу.

Відповідність додає ще один рівень. Вам може знадобитися продемонструвати відповідність вимогам CIS Controls, ISO 27001, PCI DSS або контрактним вимогам. Аудитори очікують побачити чіткі граничні засоби контролю, історію змін та докази того, що зашифрований трафік не є неконтрольованою прогалиною. Ніщо з цього не є складним окремо, але вимагає регулярної уваги та турботи.

Основна функція брандмауера

Основна функція брандмауера – охороняти вашу безпечну внутрішню мережу та публічний Інтернет. Ця роль є вирішальною для пояснення функції брандмауера в захисті системи та інформаційних активів, оскільки вона гарантує,

що проходить лише авторизований трафік. Брандмауери захищають цінні цифрові активи, зосереджуючись на трьох основах інформаційної безпеки:

– Конфіденційність: Запобігання несанкціонованому доступу для збереження конфіденційності даних.

– Цілісність: Блокування шкідливого програмного забезпечення, яке може змінити або пошкодити дані.

– Доступність: Захист від атак, таких як відмова в обслуговуванні (DoS), метою яких є зробити системи недоступними.

Невпинно моніторячи мережевий трафік, брандмауери запобігають проникненню хакерів, вірусів і черв'яків у вашу мережу або викраданню конфіденційних даних.

Основні переваги впровадження брандмауера

Впровадження надійного брандмауера надає численні переваги, які пояснюють функцію брандмауера в захисті системи та інформаційних активів:

– Захист від загроз: Брандмауери діють як щит, запобігаючи несанкціонованому доступу, шкідливим програмам та кібератакам, що потрапляють у вашу приватну мережу.

– Фільтрація трафіку: Вони ретельно перевіряють усі пакети даних, регулюючи вхідний та вихідний трафік, щоб забезпечити потік лише легітимних даних, захищаючи від вірусів та спроб фішингу.

– Контроль доступу: Забезпечуючи дотримання правил безпеки, брандмауери визначають, хто може отримувати доступ до мережевих ресурсів, запобігаючи несанкціонованому доступу.

– Безпечний віддалений доступ: Брандмауери мають вирішальне значення для захисту підключень до віртуальної приватної мережі (VPN), що дозволяє віддаленим співробітникам безпечно отримувати доступ до корпоративних мереж.

– Сегментація мережі: Вони можуть розділити мережу на ізольовані сегменти, тому, якщо одна частина порушена, пошкодження локалізується і не

може поширюватися.

– Відповідність нормативним вимогам: Багато нормативних актів (таких як HIPAA або PCI-DSS) вимагають використання брандмауерів для захисту конфіденційних даних, допомагаючи підприємствам уникати штрафів та шкоди репутації.

Як працюють брандмауери: огляд основних механізмів інспекції

Щоб пояснити функцію брандмауера в захисті системи та інформаційних активів, ми повинні зрозуміти, як вони перевіряють дані. Брандмауери – це інтелектуальні контролери трафіку, які постійно, за частки секунди, приймають рішення про те, що дозволяти або блокувати, на основі детального набору правил безпеки.

Ця фільтрація на основі правил діє як контрольний список для відхилення, визначаючи, що потрапляє, а що відхиляється. Для кожного пакета даних брандмауер приймає рішення про дозвіл або блокування, контролюючи як вхідний, так і вихідний трафік. Це запобігає проникненню загроз і запобігає виходу конфіденційних даних без дозволу.

Крім того, брандмауери надають можливості ведення журналу та аудиту, зберігаючи детальний облік мережевого трафіку. Ці журнали безцінні для розслідування підозрілої активності, підтвердження відповідності нормативним вимогам та вдосконалення політик безпеки.

Основні механізми: фільтрація пакетів, перевірка стану та проксі-сервери

Сучасні брандмауери використовують три основні методи перевірки, кожен з яких пропонує різний рівень захисту. Розуміння цих методів допомагає пояснити функцію брандмауера у захисті системи та інформаційних активів.

Фільтрація пакетів – це найбазовіша технологія брандмауера. Вона швидка та ефективна, перевіряючи заголовки пакетів на наявність такої інформації, як IP-адреси, номери портів та протоколи. Це як швидка перевірка ідентифікатора біля дверей, але їй бракує усвідомлення ширшого контексту

з’єднання.

Перевірка стану є більш інтелектуальною. Вона використовує відстеження стану з’єднання, щоб запам’ятовувати активні, легітимні з’єднання. Розуміючи контекст потоку трафіку, вона знає, коли дозволити відповідь на запит, зроблений вашою мережею, що робить її набагато ефективнішою у блокуванні спроб несанкціонованого доступу.

Проксі-сервіси пропонують найретельнішу перевірку, працюючи на рівні додатків. Вони діють як посередники, отримуючи та повторно передаючи дані від імені вашої мережі. Ця перевірка на рівні додатків дозволяє їм перевіряти фактичний вміст комунікацій, забезпечуючи найвищий рівень безпеки для критично важливих додатків.

Запобігання шкідливому програмному забезпеченню та іншим кіберзагрозам

Коли ми пояснюємо функцію брандмауера для захисту системи та інформаційних активів, це зводиться до запобігання загрозам. Брандмауери працюють цілодобово, щоб зупинити різні кіберзагрози, перш ніж вони завдадуть шкоди.

– Блокування шкідливого коду: Брандмауери розпізнають підозрілі моделі трафіку, пов'язані з вірусами та черв'яками, зупиняючи їх на периметрі мережі до того, як може статися зараження.

– Запобігання DoS-атак: Коли зловмисники перевантажують мережу трафіком, брандмауери можуть виявляти ці незвичайні закономірності та блокувати джерело шкідливого програмного забезпечення, дозволяючи при цьому продовжувати легітимний бізнес-трафік.

– Зменшення спроб фішингу: Хоча брандмауери не є повним рішенням для боротьби з фішингом, вони можуть блокувати доступ до відомих шкідливих веб-сайтів, посилання на які містяться у фішингових електронних листах, забезпечуючи важливу систему безпеки.

– Зупинка несанкціонованого витоку даних: моніторячи вихідний трафік,

брандмауери можуть виявляти та блокувати спроби шкідливого програмного забезпечення або інсайдерів надсилати конфіденційну інформацію з вашої мережі.

На рисунку 3.1 зображена структурна схема системи. На який розглянута розроблена система в цілому.

Рисунок 3.1 – Структурна схема системи

3.3 Розробка функціональної схеми

На рисунку 3.2 зображена функціональна схема системи. На який детально розглянута розроблена система. Розглянемо роботу розробленої системи

на прикладі запиту Інтернету користувача № N.

Користувач N через локальну мережу через WEB браузер робочої станції посилає запит до мережі з даними IP адреси. На сервері розроблене ПЗ проводить автентифікацію робочої станції за допомогою протоколу тунелювання та далі проводе запит сторінки через брандмауер сервера. В цей час розроблене ПЗ проводе зчитування трафіку локальної мережі, через демон PPTP, керування відбувається через відповідний модуль. Всі дані роботи користувача у Інтернеті заносяться до бази даних сервера. Основні таблиці це «Таблиця Кошти» де зберігаються дані витрачених коштів, «Таблиця Користувачі» де знаходиться інформація користувача, «Таблиця Статистика» статистичні дані та «Таблиця Група» де зберігаються встановлені привілеї користувача.

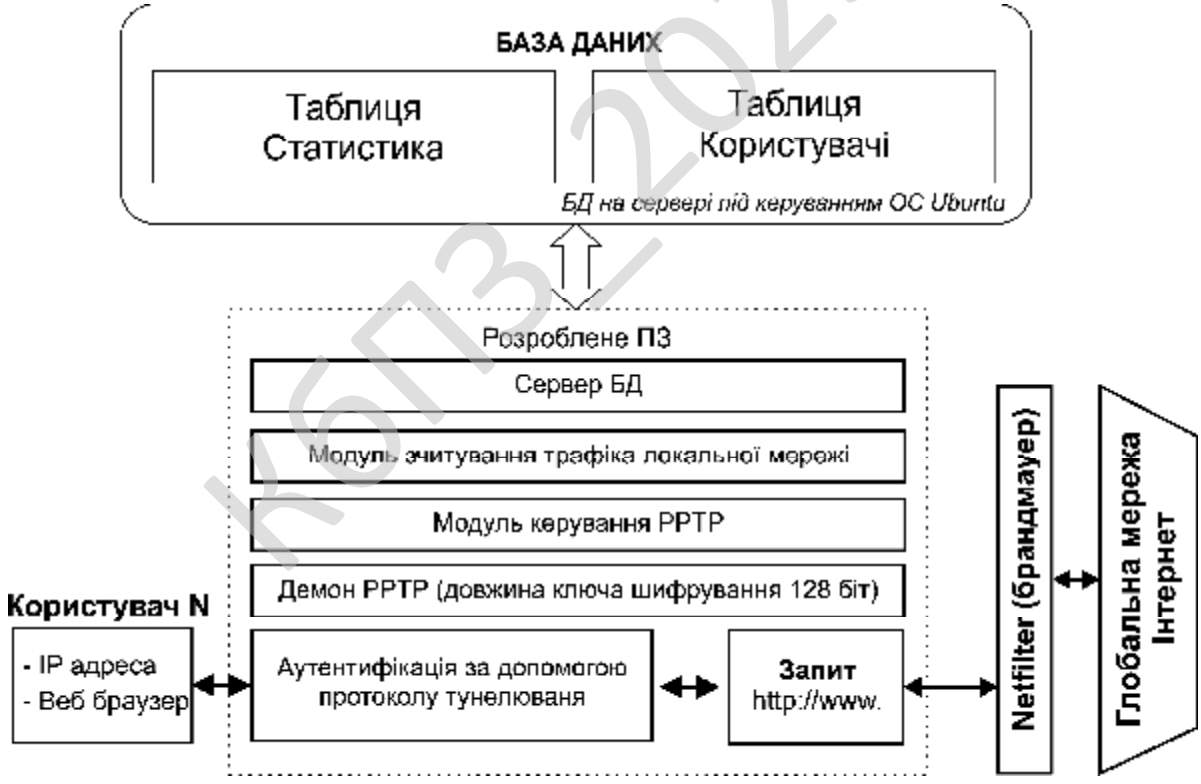

Рисунок 3.2 – Функціональна схема системи

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

3.4 Розробка діаграми процесів

Діаграма взаємодії процесів системи, розробленої у результаті виконання магістерської роботи, наведена на рисунку 3.3.

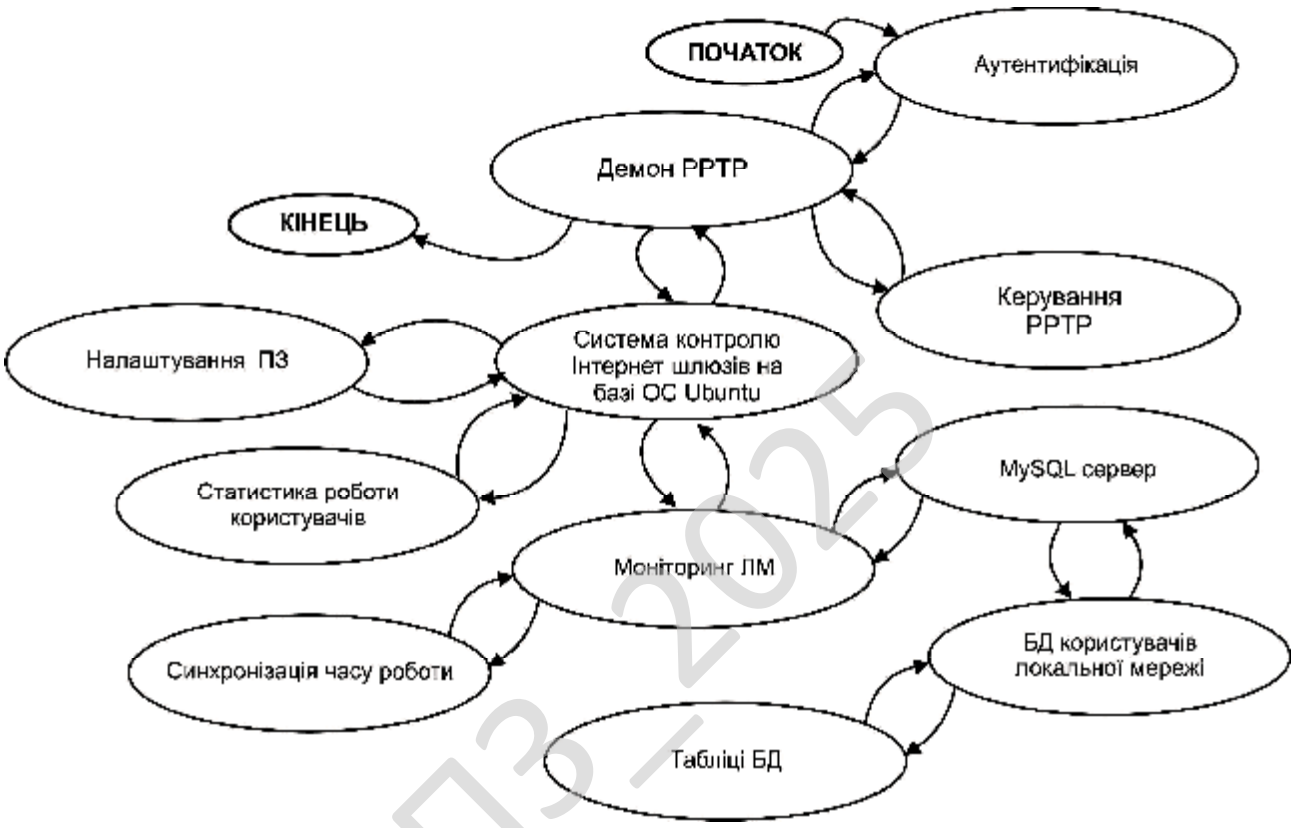

Рисунок 3.3 – Діаграма взаємодії процесів

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

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

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

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

Діаграми потоків даних містять чотири типи елементів:

– Процеси які являють собою трансформацію даних в рамках описуваної системи.

– Сховища даних (репозиторії).

– Зовнішні по відношенню до системи сутності.

– Потоки даних між елементами трьох попередніх типів.

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 ПРАВИЛЬНІСТЬ ПРОЕКТНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

Під час роботи над магістерською роботою було створено блок-схеми. Перед їх розглядом необхідно провести роз’яснення який саме тип блок-схем використовується.

Блок-схема це представлення задачі для її аналізу або розв'язування за допомогою спеціальних символів (геометричних образів), які позначають такі елементи, як операції, потік, дані тощо. Блок вхідних та вихідних даних прийнято позначати паралелограмом, блок обчислень (обробки) даних – прямокутником, блок прийняття рішень – ромбом, еліпсом – початок та кінець алгоритму.

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

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

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

Основні елементи схем алгоритму це термінатор, процес, рішення, зумовлений процес (підпрограма), дані та з'єднувач.

Термінатор це елемент відображає вхід із зовнішнього середовища або вихід з неї (найчастіше застосування – початок і кінець програми). Всередині фігури записується відповідна дія.

Процес це виконання однієї або кількох операцій, обробка даних будь-якого виду (зміна значення даних, форми подання, розташування). Всередині фігури записують безпосередньо самі операції.

Рішення це показує рішення або функцію перемикального типу з одним входом і двома або більше альтернативними виходами, з яких тільки один може бути обраний після обчислення умов, визначених всередині цього елементу. Вхід в елемент позначається лінією, що входить зазвичай у верхню вершину елементу. Якщо виходів два чи три то зазвичай кожен вихід позначається лінією, що виходить з решти вершин (бічних і нижній). Якщо виходів більше трьох, то їх слід показувати однією лінією, що виходить з вершини (частіше нижній) елемента, яка потім розгалужується. Відповідні результати обчислень можуть записуватися поруч з лініями, що відображають ці шляхи.

Зумовлений процес (підпрограма) це символ відображає виконання процесу, що складається з однієї або кількох операцій, що визначені в іншому місці програми (у підпрограмі, модулі). Всередині символу записується назва процесу і передані в нього дані.

Дані це перетворення у форму, придатну для обробки (введення) або відображення результатів обробки (виведення). Цей символ не визначає носія даних (для вказівки типу носія даних використовуються специфічні символи).

З'єднувач це символ відображає вихід в частину схеми і вхід з іншої частини цієї схеми. Використовується для обриву лінії та продовження її в іншому місці (приклад: поділ блок-схеми, що не поміщається на листі). Відповідні сполучні символи повинні мати одне (при тому унікальне) позначення.

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

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

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

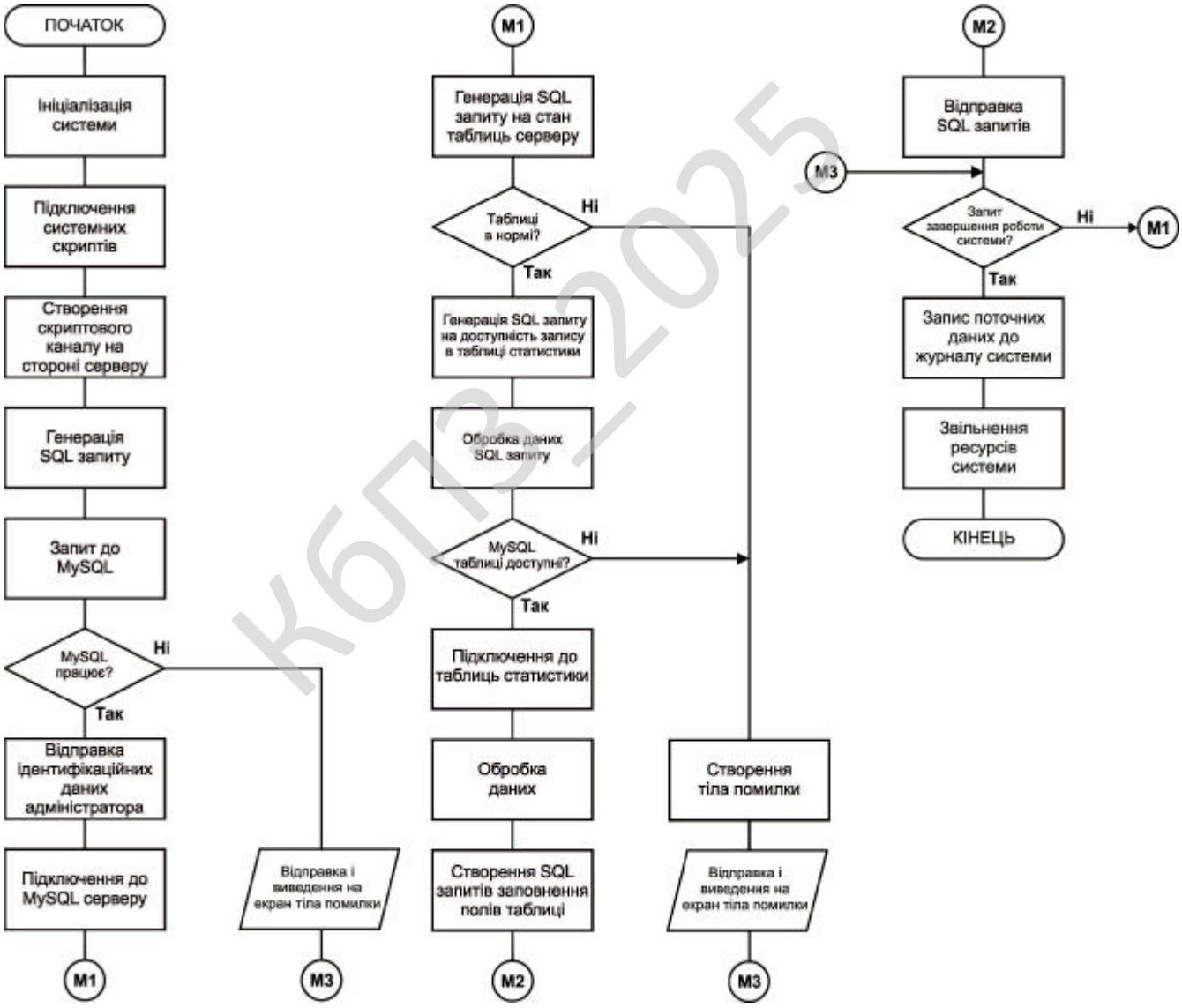

Рисунок 4.1 – Блок-схема основної програми

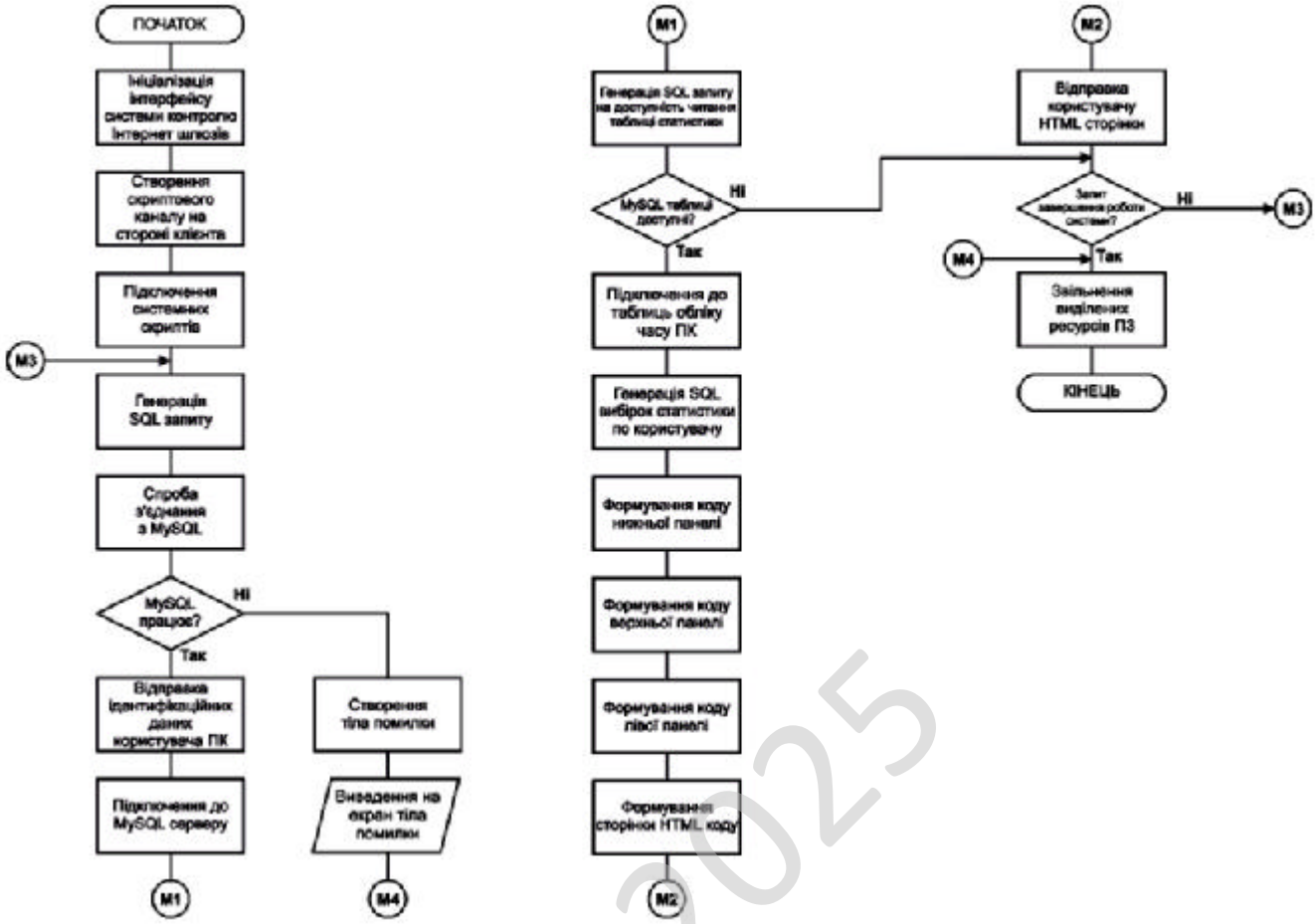

Рисунок 4.2 – Блок-схема роботи підпрограми

При виборі початкової точки відліку при побудові схем було враховано, що виходячи з вибору мови програмування і інших технічних засобів, програма буде об’єктно-орієнтована що вимагає оптимізації програми високого рівня, також те, що при розробці програми слід надати особливу увагу модулю системи контролю Інтернет шлюзів на базі ОС Ubuntu.

Було використано підходи з використанням UML, це уніфікована мова моделювання, використовується у парадигмі об'єктно-орієнтованого програмування. Є невід'ємною частиною уніфікованого процесу розробки програмного забезпечення. UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, називаної UML-моделлю. UML був створений для визначення,

візуалізації, проектування й документування в основному програмних систем. UML не є мовою програмування, але в засобах виконання UML-моделей як інтерпретованого коду можлива кодогенерація.

UML може бути застосовано на всіх етапах життєвого циклу аналізу бізнес-систем і розробки прикладних програм. Різні види діаграм які підтримуються UML, і найбагатший набір можливостей представлення певних аспектів системи робить UML універсальним засобом опису як програмних, так і ділових систем.

Діаграми дають можливість представити систему (як ділову, так і програмну) у такому вигляді, щоб її можна було легко перевести в програмний код. Основною причиною використання мови UML є спілкування розробників між собою.

Крім того, UML спеціально створювалася для оптимізації процесу розробки програмних систем, що дозволяє збільшити ефективність їх реалізації у кілька разів і помітно поліпшити якість кінцевого продукту.

UML прекрасно зарекомендувала себе в багатьох успішних програмних проектах. Засоби автоматичної генерації кодів дозволяють перетворювати моделі мовою UML у вихідний код об’єктно-орієнтованих мов програмування, що ще більш прискорює процес розробки. Практично усі CASE-засоби (програми автоматизації процесу аналізу і проектування) мають підтримку UML. Моделі розроблені в UML, дозволяють значно спростити процес кодування і направити зусилля програмістів безпосередньо на реалізацію системи.

Діаграми підвищують супроводжуваність проекту і полегшують розробку документації.

UML необхідний:

– Керівникам проектів, які керують розподілом завдань і контролем за проектом.

– Проектувальникам інформаційних систем які розробляють технічні завдання для програмістів.

– Бізнес-аналітикам, які досліджують реальну систему і здійснюють інжиніринг і реінжиніринг бізнесу компанії.

– Програмістам які реалізовують модулі інформаційної системи.

При модифікації системи об'єктний підхід дозволяє легко включати в систему нові об'єкти і виключати застарілі без істотної зміни її життєздатності. Використання побудованої моделі при модифікаціях системи дає можливість усунути небажані наслідки змін, оскільки вони не ламають структури системи, а тільки змінюють поведінку об'єктів.

Розглянемо розроблені алгоритми на мовах PHP та PERL.

Читання й запис файлів і каталогів є чи не головною перевагою Perl. Практично будь-який Perl – скрипт використовує або запис у файли, або пошук певних рядків у файлі, або запис або завантаження файлів у каталоги.

Файл можна відкривати для читання, запису, додавання або пошуку певних рядків:

1. Відкрити файл для читання – open (FILE,"$file").

2. Відкрити файл для запису – open (FILE,">$file").

3. Відкрити файл для додавання – open (FILE,">>$file").

4. Відкрити файл для читання й додавання – open (FILE,"+<$file").

Де:

– FILE це дескриптор або описатель файлу, тобто ім'я, під яким він фігурує в програмі. Може бути будь-яким рядком символів, рекомендується набирати прописними буквами щоб уникнути неоднозначностей;

– $file змінна для файлу, містить звичайне ім'я файлу й шлях до нього. Переважно оголосити цю змінну на початку скрипта. Це зручно, якщо файл перебуває десь глибоко в підкаталогах, приміром, незручно весь час набирати /usr/local/htdocs/main/pages/file.html.

Зміст файлу можна читати порядково або в масив:

```
#!/usr/local/bin/perl
# прочитати один рядок з файлу.
open (STAT,"$file");
$count=;
```

```
close (STAT);

#!/usr/local/bin/perl
# прочитати файл у масив.
open (STAT,"$file");
@count=;
close (STAT);
```

Розглянемо сценарій реєстрації користувача на веб-сервері. Ім'я користувача і його пароль записуються в текстовий файл і використовуються для його наступної автентифікації.

```
#!/usr/local/bin/perl
# об’ява глобальних змінних
$request=$ENV{'REQUEST_METHOD'};
$content=$ENV{'CONTENT_LENGTH'};
$basedir="http://www.mydomain.com/~";
$userdir="f:/home";
# Підпрограми для декодування даних з форми.
sub urldecode {
local($val)=@_;
$val=~ s/\+/ /g;
$val=~ s/%[0-9a-ha-h] {2}/pack('C',hex($1))/ge;
return $val;
}
sub strhtml {
local($val)=@_;
$val=~s//>/g;
$val=~s/(http:\/\/\+S)/<A href="$1">$1<\/A>/g;
return $val;
}
#
if ($request eq 'GET') {
$query=$ENV{'QUERY_STRING'};
}
else {
sysread(STDIN,$query,$content);
}
# Генеруємо форму,якщо ніякі дані не введені.
print "Content-type:text/html\n\n";
print <<HTML_gen;
<HTML><BODY bgcolor="e6e8fa">
HTML_gen
```

```
        if ($query eq '') {
        print <<HTML;
        <h2 align=center><font color="ff0000">Registration.</font></h2>
        <p><font face="serif" size=2> Please,fill in the form below.
        <p>After registration you will receive your personal directory</font>
        <p><FORM ACTION="../cgi-bin/addlogin.cgi" METHOD="POST" name="reg">
        <center><TABLE BGCOLOR="bfbfbf">
        <TR><td><font color="ff0000">*</font>
        <TD><b>Login:</b><TD><INPUT TYPE="text" NAME="login" SIZE="20">
        <TR><td><font color="ff0000">*</font>
        <TD><b>Password:</b>
        <TD><INPUT TYPE="password" NAME="pass" SIZE="20">
        <TR><td><font color="ff0000">*</font><TD><b>E-mail:</b>
        <TD><INPUT TYPE="text" NAME="email" SIZE="20">
        <TR><TD colspan=3><p><center>
        <INPUT TYPE="submit" VALUE="Submit"></center>
        </TABLE></center>
        </FORM>
# HTML
# Декодуємо поля форми
        else {
        foreach (@fields=split(/&/,$query)) {
        if (/login=(.*)/) { $login=&urldecode ($1); }
        if (/pass=(.*)/) { $password=&urldecode ($1); }
        if (/email=(.*)/) { $email=&urldecode ($1); }
        }
# Перевіряємо, не чи існує дане ім'я в системі.
        open(INFO,"login.txt") ||die;
# читаємо рядок в масив.
        @data=<INFO>;
        close(INFO);
        foreach $string(@data) {
        @item=split(/&/,$string);
# Розбиваємо рядок на частини
        foreach (@item) {
        if ($item[0] eq $login) {
# Порівнюємо отримане ім'я з першим полем файлу
# для кожного рядка і якщо таке знайдене видаємо
# помилку.
        print <<HTML;
        <h2 align=center><font color="ff0000">Error!</font></h2>
```

```
        <p><center><b>The name <font color="ff0000">$login</font> already
          exists in the system. <p>Please,go back and choose another name.</b>
        <p><form><input type="button" value="Back" onclick=
          "history.back()"></form>
        </center>
        HTML
        exit;
        }
        }
        }
# Якщо ім'я не знайдене,відкриваємо базу даних і додаємо інформацію.
        if ($item[0] ne $login) {
        open(DATA,">>login.txt");
        $string=join('&',$login,$password,$email,scalar localtime,$ENV
                    {'REMOTE_ADDR'};
        print DATA "$string\n";
        close(DATA);
# Створюємо домашній каталог користувача й переходимо в нього.
        mkdir("$userdir/$login",0700);
        chdir("$userdir/login");
        opendir(USER,"$userdir/$login");
# Поміщаємо файл index.html у каталог користувача.
        open(IN,">$userdir/$login/index.html");
        print IN "This is the test!\n";
        close(IN);
        closedir(USER);
# Зміст файлу може бути будь-яким,це тільки для прикладу.
# Генеруємо відповідь користувачеві.
        print <<HTML;
        <p><h1 align=center><font color="ff0000">Congratulations!</font></h1>
        <p><b>Your registration was successful and your data were added to
        our database.Thank you for your time.</b>
        <p><center><b><font color="ff0000">
        You entered:</font>(print this page and keep it in safe place)</b>
        <p><table>
        <tr><td><b>Your login name:</b><td><font color="0000ff">$login</font>
        <tr><td><b>Your password:</b><td><font color="0000ff">$password</font>
        <tr><td><b>Your e-mail address:</b>
        <td><font color="0000ff">$email</font>
        </table></center>
        HTML
        }
```

```
}
```

Скрипт видає відповідь у вигляді html сторінки, що містить всю інформацію, введену користувачем.

Розглянемо відкриття й читання каталогу й виведення списку файлів, що втримуються в ньому. Початок скрипта аналогічно попередньому коду.

Припустимо, що користувач, зареєстрований на веб-сервері, прагне увійти у свій домашній каталог.

```
#!/usr/local/bin/perl
# Повідомляємо глобальні змінні.
$request=$ENV{'REQUEST_METHOD'};
$content=$ENV{'CONTENT_LENGTH'};
$basedir="http://www.mydomain.com/~";
$file="login.txt";
$url="http://www.mydomain.com";
$dir="f:/home/";
$cgi="f:/usr/local/apache/cgi-bin";
# Підпрограми для декодування даних з форми.
sub urldecode {
local($val)=@_;
$val=~ s/\+/ /g;
$val=~ s/%[0-9a-ha-h] {2}/pack('C',hex($1))/ge;
return $val;
}
sub strhtml {
local($val)=@_;
$val=~s//>/g;
$val=~s/(http:\/\/\+S)/<A href="$1">$1<\/A>/g;
return $val;
}
#
if ($request eq 'GET') {
$query=$ENV{'QUERY_STRING'};
}
else {
sysread(STDIN,$query,$content);
}

# Генеруємо форму, якщо ніякі дані не введені.
print "Content-type:text/html\n\n";
```

```
        print <<HTML_gen;
        <HTML><BODY bgcolor="e6e8fa">
        HTML_gen
        if ($query eq '') {
        print "Content-type:text/html\n\n";
        print <<HTML;
        <HTML><HEAD>
        </HEAD><BODY bgcolor="e6e8fa">
        <FORM ACTION="../cgi-bin/fileman.cgi"  name="form1" METHOD="POST">
        <h2 align=center><font color="ff0000">System login.</font></h2>
        <p><center>Please,enter your login name and password:
        <p><TABLE BGCOLOR="cccccc">
        <tr><td colspan=2 align=center bgcolor="99cccc"><b>
        <font color="ff0000"> I am registered user</font></b>
        <TR><TD><p><b>Login:</b><TD><INPUT TYPE="text" NAME="login" SIZE="20">
        <TR><TD><p><b>Password:</b><TD>
        <INPUT TYPE="password" NAME="pass" SIZE="20">
        <tr><td colspan=2 align=center>
        <input type=submit value="Submit"></center>
#HTML
        }
# Якщо інформація отримана, декодуємо поля форми.
        else {
        foreach (@fields=split(/&/,$query)) {
        if (/login=(.*)/) { $login=&urldecode ($1); }
        if (/pass=(.*)/) { $password=&urldecode ($1); }
        }
# Відкриваємо базу даних і перевіряємо логін і пароль.
        open(INFO,$file) ||die;
        @data=;
        close(INFO);
        foreach $string(@data) {
        @item=split(/&/,$string);
        foreach (@item) {
        if (($item[0] eq $login) && ($item[1] eq $password)) {
# Якщо все нормально, переходимо в користувацький каталог.
        print "Content-type:text/html\n\n";
        print <<HTML;
        <html><body bgcolor="e6e8fa">

# Вітаємо користувача.
        <p><h2 align=center><font color="ff0000">Hello,$login!</font></h2>
```

```
        <p><center>Welcome to your home directory!
        <p>Your URL is <a href="$basedir$login">$basedir$login.</a></center>
        HTML
# Список директорії
        $userdir=$dir.$login;
        chdir ("$userdir");
# Відкриваємо каталог і читаємо список файлів у масив.
        opendir(DIR,"$userdir") || die "Cannot open $userdir!";
        while (@files=readdir(DIR)) {
# Якщо каталог містить підкаталоги, виводимо їх
# окремо, а також не показуємо
# каталоги "." і ".." Друкуємо шапку таблиці.
        print <<HTML;
        <p><center>
        <table bgcolor=\"bfbfbf\" width=600 border cellspacing=0
        cellpadding=0 nowrap>
        <tr><td colspan=5 align=center nowrap>
        <b><font color="ff0000">Directories</font></b></td></tr>
        <tr><td>.</td><td align=center><b>List</b>
        </td><td><b>Size</b>
        <td><td><b>Last accessed</b></td><td><b>Last modified</b></td>
#HTML
        foreach $file(@files)  {
# Статистика файлів – розмір, час останнього звернення й модифікації.
        $size=(stat("$userdir/$file"))[7];
        $atime=localtime((stat("$userdir/$file"))[8]);
        $mtime=localtime((stat("$userdir/$file"))[9]);
# друкуємо список підкаталогів.
        if ( -d "$userdir/$file" && "$file" ne "." && "$file" ne "..") {
        print "<tr><td width=30><img src=\"$url/image/folder.gif\">
        </td><td width=100 align=left>$file</td>\n";
        print "<td width=50>",$size,"</td><td width=200>",$atime,"</td>
        <td width=200>",$mtime,"</td></tr>\n";
        }
        }
        print "</table>\n";
#  Files list  #
# Ту ж операцію проводимо для файлів. Друкуємо шапку таблиці.
        print <<HTML;
        <p><table bgcolor=\"bfbfbf\" width=600 border
        cellspacing=0 cellpadding=0>
        <tr><td colspan=5 align=center><b><font color="ff0000">Files</font>
```

```
        </b></td></tr>
        <tr><td>.</td><td><b>List</b><td><b>Size</b>
        <td><b>Last accessed</b><td><b>Last modified</b></tr>
#HTML
        foreach $file(@files)  {
        $size=(stat("$userdir/$file"))[7];
        $atime=localtime((stat("$userdir/$file"))[8]);
        $mtime=localtime((stat("$userdir/$file"))[9]);
        if (!-d "$userdir/$file" && "$file" ne "." && "$file" ne "..") {
# поміщаємо знайдені файли в масив
        push (@dir,"$userdir/$file");
# підраховуємо їх кількість.
        $number=@dir;
# Видаємо інформацію.
        print "<tr><td width=30><img src=\"$url/image/page.gif\"></td>
        <td width=100><a href=\"$basedir$login/$file\">",$file,"</a></td>\n";
        print "<td width=50>",$size,"</td>\n";
        print "<td width=200>",$atime,"</td>
        <td width=200>",$mtime,"</td></tr>\n";
        }
        }
        print "</table>\n";
        print "<p><center><b><font color=\"0000ff\">There are",
        $number," files in this directory.</b></font></center>\n";
        }
```

При роботі із системою моніторингу трафіка локальної мережі підприємства. Виникла проблема роботи з XML. Інформація на сервері зберігається саме в цьому форматі. Тому був розроблений перетворювач XML в HTML на мові програмування PHP.

Як неважко припустити по його назві, мова HTML (Hypertext Markup Language) ставиться до числа так званих мов розмітки тексту (markup languages).

Під терміном "розмітка" розуміється загальна службова інформація, яка не виводиться разом з документом, але визначає; як повинні виглядати ті або інші фрагменти документа.

Наприклад можна зажадати, щоб яке-небудь слово виводилося жирним або курсивним шрифтом, вивести окремий абзац особливим шрифтом або оформляти заголовки збільшеним шрифтом. Текстовий редактор теж використовує особливу

форму розмітки для вистави тих атрибутів форматування, які я вибираю. Таким чином, у ньому теж використовується особливий різновид мови розмітки.

У наші дні існує безліч різних мов розмітки. Наприклад, у комунікаційних програмах особлива форма розмітки визначає зміст кожного пакета з нулів і одиниць, що пересилається в Інтернеті. Коли ми підкреслюємо слова в книзі, це теж можна вважати свого роду розміткою. Втім, будь-яка мова розмітки повинен вирішувати два важливі завдання:

1. Мова визначає синтаксис розмітки. Наприклад, у відповідності зі специфікацією HTML конструкція text визначає синтаксично правильну розмітку тексту, а конструкція <xr5t>text</x4rt> уважається неправильної через розбіжність відкриваючого й закриваючого тегів.

2. Мова визначає зміст розмітки. Звичайно, ви знаєте, що команда text виводить слово text жирним шрифтом. У цьому випадку визначається зміст, пов'язаний з оголошенням деякого компонента документа.

Стрімкий розвиток Web за останні кілька років наочно показує, що самою популярною мовою розмітки тексту є HTML.

Перетворення XML в HTML.

```
<?
Class XMLHTML {
VAR $xmlparser;
VAR $tagcolor ="#800000";
VAR $datacolor ="#0000ff";
function XMLHTML() {
$this->xmlparser = xml_parser_create();
xml_set_object($this->xmlparser, $this);
xml_set_element_handler($this->xmlparser, "starttag", "endtag");
xml_set_character_data_handler($this->xmlparser, "characterdata");
}
// Функція відповідає за обробку всіх відкриваючих тегів.
function starttag($parser, $tagname, $attributes)
{
GLOBAL $tagcolor;
print "<font size=\"-2\" color=\"$this->
tagcolor\" face=\"arial,verdana\">&lt; $tagname&gt; </font><br>";
}
```

```
// Функція відповідає за обробку всіх символьних даних.
        function characterdata($parser, $characterdata)
        {
        GLOBAL $datacplor;
        print "<font size=\"-2\" color=\"$this->
        datacolor\" face=\"arial,verdana\">   
          $characterdata</font><br>";
        }
// Функція відповідає за обробку всіх закриваючих тегів.
        function endtag($parser, $tagname) {
        GLOBAL $tagcolor;
        print "<font size=\"-2\" color=\"$this->tagcolor\"
        face=\"arial, verdana\">&lt;/$tagname&gt;</font><br>";
        }
        function parse($fp) {
        xml_parse($this->xmlparser,$data);
// Обробити файл XML
        while ( $line = fread($fp, 4096) ):
// При виникненні помилки перервати обробку
// і вивести повідомлення про помилку.
        if ( ! xml_parse($this->xmlparser, $line, feof($fp))):
        die(sprintf("XML error: %s at line %d",
        xml_error_sthng(xml_get_error_code($this->xmlparser)),
        xml_get_curren_line_number($this->xml_parser)));
        endif;
        endwhile;
        }
        }
// Відкрити файл XML для обробки
        $xml_file = "bookmarks.xml";
        $fp = fopen ($xml_file, "r");
// Створити новий об'єкт
        $xml_parser = new XMLHTML;
// Обробити $xml_file
        $xml_parser->parse($fp);
        ?>
```

4.2 Захист розробленого програмного забезпечення

Захист розробленого програмного забезпечення буде відбуватися за допомогою IDEA – симетричний блоковий алгоритм шифрування даних, запатентований швейцарською фірмою Ascom. Відомий тим, що застосовувався в пакеті програм шифрування PGP. У листопаді 2000 року IDEA був представлений як кандидат у проекті NESSIE в рамках програми Європейської комісії IST (англ. Information Societes Technology, інформаційні громадські технології).

Першу версію алгоритму розробили в 1990 році Лай Сюецзя (Xuejia Lai) і Джеймс Мессі (James Massey) зі Швейцарського інституту ETH Zürich (за контрактом з Hasler Foundation, яка пізніше влилася в Ascom-Tech AG) як заміна DES (англ. Data Encryption Standard, стандарт шифрування даних) і назвали її PES (англ. Proposed Encryption Standard, запропонований стандарт шифрування). Потім, після публікації робіт Біхамом і Шаміра по диференціальному криптоаналізу PES, алгоритм був поліпшений з метою посилення криптостійкості і названий IPES (англ. Improved Proposed Encryption Standard, покращений запропонований стандарт шифрування). Через рік його перейменували в IDEA (англ. International Data Encryption Algorythm).

Так як IDEA використовує 128-бітний ключ і 64-бітний розмір блоку, відкритий текст розбивається на блоки по 64 біт. Якщо таке розбиття неможливо, останній блок доповнюється різними способами певною послідовністю біт. Для уникнення витоку інформації про кожному окремому блоці використовуються різні режими шифрування. Кожен вихідний незашифрований 64 – біт ний блок ділиться на чотири підблока по 16 біт кожен, так як всі алгебраїчні операції, що використовуються в процесі шифрування, відбуваються над 16-бітними числами. Для шифрування і розшифрування IDEA використовує один і той же алгоритм.

Позначення операцій:

- $\boxplus$ Додавання за модулем 2^{16}.
- $\odot$ Множення за модулем $2^{16}+1$.

– $\oplus$ Побітова виключна диз'юнкція.

Фундаментальним нововведенням в алгоритмі є використання операцій з різних алгебраїчних груп, а саме:

Додавання за модулем 2^{16}.

Множення за модулем $2^{16}+1$.

Побітова виключна диз'юнкція (XOR).

Ці три операції несумісні в тому сенсі, що ніякі дві з них не задовольняють дистрибутивному закону, тобто:

$$a \odot (b \oplus c) \neq (a \odot b) \oplus (a \odot c).$$

Застосування цих трьох операцій ускладнює криптоаналіз IDEA в порівнянні з DES, який базується виключно на операції виключає АБО, а також дозволяє відмовитися від використання S-блоків і таблиць заміни. IDEA є модифікацією мережі Фейстеля.

Генерація ключів

З 128-бітного ключа для кожного з восьми раундів шифрування генерується по шість 16-бітних подключів, а для вихідного перетворення генерується чотири 16-бітних підключа. Всього буде потрібно 52 = 8 x 6 + 4 різних подключів по 16 біт кожен. Процес генерації п'ятдесяти двох 16-бітних ключів полягає в наступному:

Насамперед, 128-бітний ключ розбивається на вісім 16-бітних блоків. Це будуть перші вісім подключів по 16 біт кожен – $(K_1^{(1)}K_2^{(1)}K_3^{(1)}K_4^{(1)}K_5^{(1)}K_6^{(1)}K_1^{(2)}K_2^{(2)})$

Потім цей 128-бітний ключ циклічно зсувається вліво на 25 позицій, після чого новий 128-бітний блок знову розбивається на вісім 16-бітних блоків. Це вже наступні вісім подключів по 16 біт кожен – $(K_3^{(2)}K_4^{(2)}K_5^{(2)}K_6^{(2)}K_1^{(3)}K_2^{(3)}K_3^{(3)}K_4^{(3)})$

Процедура циклічного зсуву і розбивки на блоки триває до тих пір, поки не будуть згенеровані всі 52 16-бітних підключа.

Шифрування

Структура алгоритму IDEA показана на рисунку 4.3.

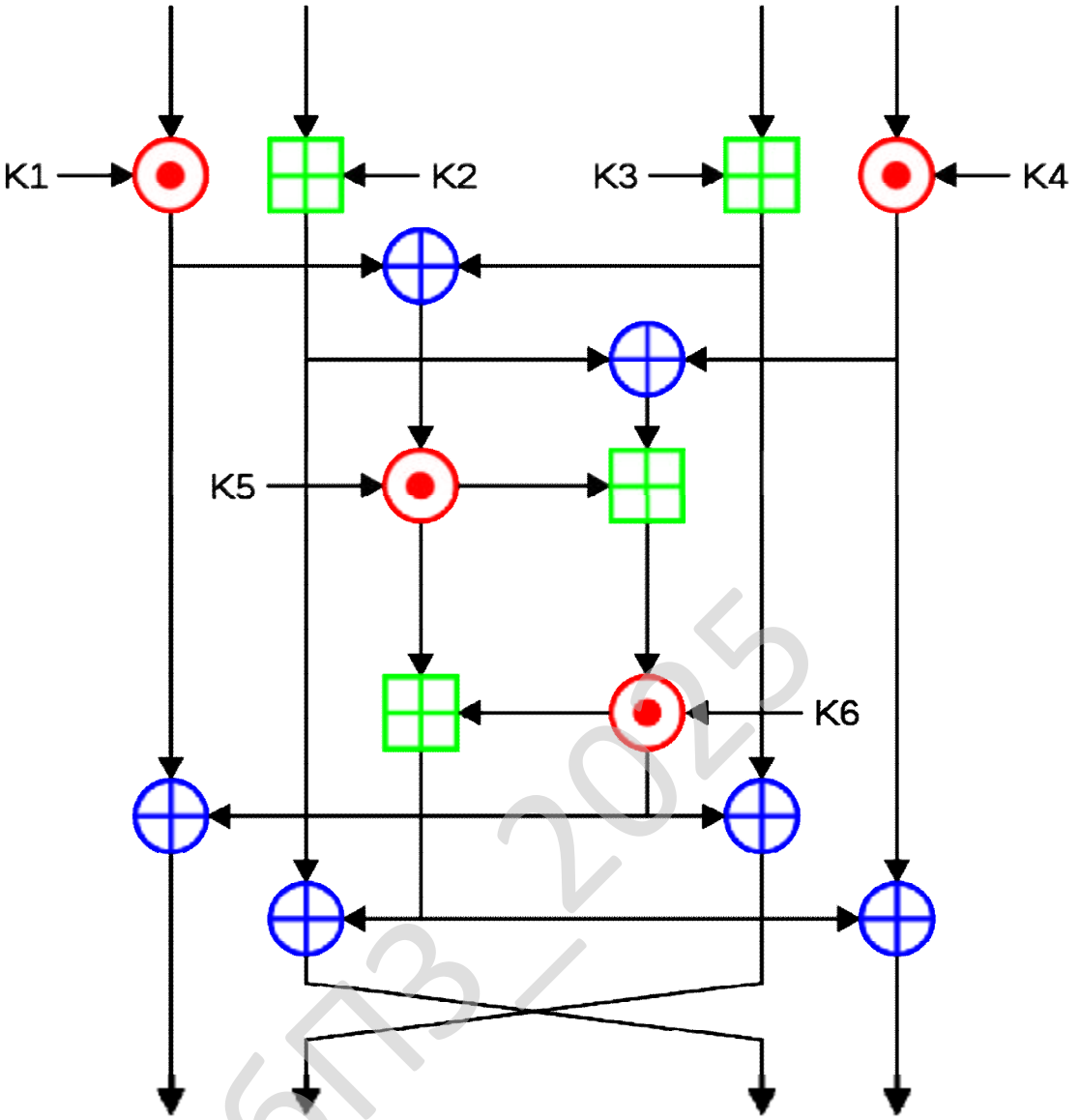

Рисунок 4.3 – Структура алгоритму IDEA

Процес шифрування складається з восьми однакових раундів шифрування і одного вихідного перетворення. Вихідний незашифрований текст ділиться на блоки по 64 біта. Кожен такий блок ділиться на чотири підблока по 16 біт кожен. На рисунку ці підблоки позначені D_1, D_2, D_3, D_4. У кожному раунді використовуються свої підключі згідно з таблицею підключів. Над 16-бітними підключами і підблока незашифрованого тексту проводяться наступні операції:

– Множення за модулем $2^{16}+1 = 65537$, причому замість нуля використовується 2^{16}.

– Додавання за модулем 2^{16}.

– Побітове виключне АБО.

В кінці кожного раунду шифрування є чотири 16-бітних підблоки, які потім використовуються як вхідні підблоки для наступного раунду шифрування. Вихідна перетворення являє собою скорочений раунд, а саме, чотири 16-бітних підблоки на виході восьмого раунду і чотири відповідних підключа піддаються операціям:

– Множення за модулем $2^{16}+1$.

– Додавання за модулем 2^{16}.

Після виконання вихідного перетворення конкатенація підблоків D_1', D_2', D_3' і D_4' являє собою зашифрований текст. Потім береться наступний 64-бітний блок незашифрованого тексту і алгоритм шифрування повторюється. Так продовжується до тих пір, поки не зашифрують всі 64-бітові блоки вихідного тексту.

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Розроблене ПЗ немає графічного інтерфейсу, поточні результати її роботи зберігаються у файлі system-data.tt як це можна побачити на рисунку 5.1. Під час роботи розроблене ПЗ системи контролю Інтернет шлюзів на базі ОС Ubuntu проводить контроль наступних підсистем:

– Контроль доступу до локального WEB-серверу.

– Контроль доступу до локального FTP-серверу.

– Декілька каналів вхідного трафіку.

– Міжмережний екран.

– Локальний сервер електронної пошти.

– Контроль доступу до Інтернету.

– Контроль телефонних дзвінків.

– Модуль біллінгу.

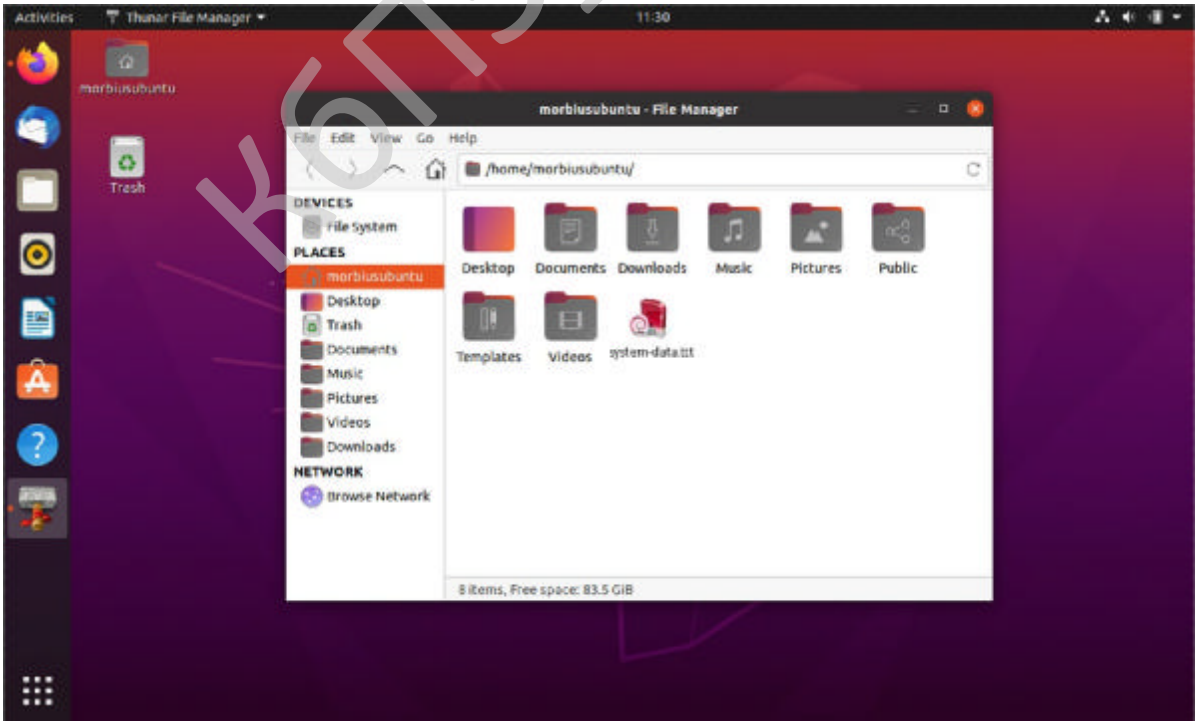

Рисунок 5.1 – Результат роботи ПЗ

Налаштування розробленої системи контролю Інтернет шлюзів на базі ОС Ubuntu знаходиться у файлі settings.conf.

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

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

Проводилась оцінка:

– відповідності поставленим вимогам;

– правильна відповідь для усіх можливих вхідних даних;

– виконання функцій за прийнятний час;

– практичність;

– сумісність з ОС та стороннім ПЗ.

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

Як результат ПЗ тестувалось стандартним виконанням програми з метою виявлення помилок або інших дефектів.

Проводилось тестування форматом чорної скриньки. Основне місце програми тестів «чорної скриньки» – інтерфейс ПЗ. Відомі: функції програми. Досліджується: робота кожної функції на всій області визначення.

Ці тести демонструють:

– Як виконуються функції програми.

– Як приймаються вихідні дані.

– Як виробляються результати.

– Як зберігається цілісність зовнішньої інформації.

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

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

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

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

– Набір, утворений такими вхідними даними, які призводять до аномалій у поведінці програми (назвемо його ITc).

– Набір, утворений такими вхідними даними, які демонструють дефекти програми (назвемо його OT).

Будь-який спосіб тестування «чорної скриньки» повинен:

– Виявити такі вхідні дані, які з високою ймовірністю належать набору ITc;

– Сформулювати такі очікувані результати, які з високою імовірністю є елементами набору OT.

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

Тестування «чорної скриньки» забезпечує пошук наступних категорій помилок:

– Некоректних чи відсутніх функцій;

– Помилок інтерфейсу;

– Помилок у зовнішніх структурах даних або в доступі до зовнішньої бази даних;

– Помилок характеристик (необхідна ємність пам'яті і т.д.);

– Помилок ініціалізації та завершення.

Обрано умови розповсюдження – Freeware.

Це власницьке програмне забезпечення, котре можна Безоплатно використовувати протягом необмеженого терміну без обмежень у функціональності, і поширюване без сирцевих кодів.

Автори такого програмного забезпечення, як правило, хочуть «дати щось спільноті», але хочуть також контролювати його подальшу розробку. Іноді, коли програмісти вирішують припинити розробку, вони передають сирцевий код іншим програмістам, або ж спільності як вільне програмне забезпечення.

Дуже часто плутають поняття «безплатне програмне забезпечення» та «вільне програмне забезпечення», хоча вони суттєво відрізняються.

Безплатне програмне забезпечення можна безоплатно встановлювати та використовувати (іноді з певними обмеженнями, як, наприклад, «безплатне для домашнього або некомерційного вжитку»), в той час як вільне програмне забезпечення можна продавати за будь-яку суму, але при тому, у користувача, котрий його отримує, повинні бути права на вивчення, модифікацію та поширення сирцевих кодів одержаної програми.

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи контролю Інтернет шлюзів на базі ОС Ubuntu.

Метою розробки є дослідження та програмна реалізація системи контролю Інтернет шлюзів на базі ОС Ubuntu.

Об'єктом дослідження є процес контролю Інтернет шлюзів на базі ОС Ubuntu.

Предметом дослідження є методи контролю Інтернет шлюзів на базі ОС Ubuntu.

Методи дослідження базуються на методах захисту інформації у комп'ютерних мережах, методах математичної статистики, методах розробки програмного забезпечення.

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

– Удосконалено метод контролю Інтернет шлюзів на базі ОС Ubuntu.

– Розроблено вітчизняний продукт контролю Інтернет шлюзів на базі ОС Ubuntu, який має більш широкі можливості, на відміну від існуючих аналогів.

7 МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ІТ-ПРОЄКТУ

7.1 Визначення цільової аудиторії кінцевого готового продукту

Результати дослідження і практичної реалізації системи контролю Інтернет-шлюзів на базі ОС Ubuntu насамперед можуть зацікавити державні установи, освітні заклади, бізнес-компанії та ІТ-підприємства, які прагнуть підвищити рівень безпеки своєї мережі, зменшити витрати на ліцензійне програмне забезпечення і мати повний контроль над мережевим трафіком. Для державних структур важливою є не лише економія, а й незалежність від іноземних розробників, тому використання відкритої операційної системи Ubuntu забезпечує високий рівень кіберсуверенітету.

Також такий проєкт буде корисний для навчальних закладів – університетів і технікумів, де він може використовуватись як навчальний приклад для студентів спеціальностей з інформаційних технологій, кібербезпеки чи комп'ютерних систем. Він демонструє практичне поєднання відкритих технологій і прикладного адміністрування мереж.

Бізнес-компанії середнього і малого масштабу, особливо ті, що мають обмежений бюджет на ІТ, зможуть зацікавитись цим рішенням, адже Ubuntu є безкоштовною, гнучкою та надійною платформою. Для них важливо, що така система може бути налаштована під конкретні потреби підприємства, без зайвих витрат на дорогі корпоративні продукти. Таким чином, результати дослідження мають як практичну, так і навчальну цінність, відкриваючи перспективи для широкого кола користувачів.

7.2 Оцінка привабливості шляхом застосування методів експертних оцінок

Щоб оцінити привабливість впровадження системи контролю Інтернет-шлюзів, можна провести експертне опитування серед фахівців у галузі IT-інфраструктури, кібербезпеки та адміністрування мереж. Наприклад, група з десяти експертів може оцінити проект за критеріями: вартість реалізації, гнучкість налаштування, рівень безпеки, простота обслуговування та масштабованість. Кожен експерт виставляє оцінку від 1 до 10, після чого обчислюється середнє значення за кожним показником.

Припустимо, за результатами оцінки проект отримав 9 балів за гнучкість, 8 за безпеку, 9 за економічність, 7 за простоту впровадження і 8 за масштабованість. Середній підсумковий бал 8,2 свідчить про високу привабливість проєкту з точки зору ефективності та потенційної користі. Така експертна оцінка допомагає не лише підтвердити практичну цінність розробки, а й визначити напрями вдосконалення системи.

Подібний підхід до оцінки може використовуватись при подачі проєкту на державне фінансування або при пошуку партнерів серед IT-компаній, що займаються безпекою даних. Таким чином, метод експертних оцінок дозволяє об’єктивно виміряти перспективність і конкурентоспроможність системи на ринку.

7.3 Вибір методу оцінки вартості ПЗ

Для оцінки вартості впровадження системи контролю Інтернет-шлюзів на базі Ubuntu доцільно застосувати метод повної вартості володіння (TCO – Total Cost of Ownership). Цей метод дозволяє врахувати не лише початкові інвестиції у закупівлю обладнання чи оплату праці фахівців, а й довгострокові витрати – на підтримку, оновлення, резервне копіювання даних та навчання персоналу. Саме

цей підхід найточніше відображає реальну економічну картину експлуатації системи.

Оскільки Ubuntu є безкоштовною, головні витрати зосереджуються у сфері технічного впровадження та адміністрування. Однак, на відміну від комерційних систем, тут немає потреби у регулярних ліцензійних платежах, що істотно знижує витрати у довгостроковій перспективі.

Комбінація ТСО з методом порівняльного аналізу вигод (Cost-Benefit Analysis) дозволить також оцінити, наскільки впровадження системи підвищить ефективність роботи організації. Таким чином, можна порівняти витрати з очікуваними вигодами, наприклад, зі зменшенням кількості простоїв мережі, економією часу адміністраторів і зниженням ризиків кібератак. Це робить оцінку реалістичною і зрозумілою навіть для нефахівців.

7.4 Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості

Економічна ефективність від впровадження системи контролю Інтернет-шлюзів на базі ОС Ubuntu для середньої компанії з 200 співробітниками, що активно використовують корпоративну мережу та доступ до Інтернету для роботи може бути розрахована наступним чином. Вхідні дані зафіксовано в таблиці 7.1.

Розрахунок економічного ефекту демонструє наступне: економія на відмові від комерційних ліцензій – 120 000грн, зростання продуктивності працівників – 420 000грн, зменшення витрат на ліквідацію наслідків кібератак – 140 000грн, оптимізація пропускної здатності каналів зв’язку – 50 000грн, загальний річний ефект – 730 000 грн, витрати на обслуговування системи – 30 000 грн, чистий річний економічний ефект – 700 000 грн, термін окупності (Payback Period) – 0,26 року (~3 місяці), окупність інвестицій – 389%.

Таблиця 7.1 – Вихідні дані для розрахунку

Показник	**До впровадження**	**Після впровадження**
Кількість співробітників, які користуються Інтернетом	200	200
Витрати на комерційне програмне забезпечення для шлюзів (щороку)	120 000 грн	—
Втрати продуктивності через нецільове використання Інтернету	10% робочого часу	3%
Кількість випадків витоку даних / зараження вірусами за рік	8	1
Середня вартість відновлення роботи після інциденту	20 000 грн	5 000 грн
Заробітна плата системного адміністратора (щомісяця)	40 000 грн	40 000 грн
Початкові інвестиції у впровадження системи (налаштування, сервери)	—	180 000 грн
Річні витрати на обслуговування	—	30 000 грн

Додаткові нефінансові ефекти: підвищення рівня інформаційної безпеки компанії – зменшення ризику витоку даних клієнтів і внутрішніх документів, прозорість використання Інтернету – керівництво отримує чітку аналітику активності користувачів і навантаження на мережу, гнучкість модернізації – завдяки відкритій архітектурі Ubuntu можна інтегрувати нові модулі без купівлі додаткових ліцензій, надійність і стабільність роботи – система базується на перевіреній платформі з активною спільнотою підтримки.

Впровадження системи контролю Інтернет-шлюзів на базі Ubuntu дає змогу компанії значно скоротити операційні витрати, підвищити безпеку й

продуктивність персоналу. Уже через три місяці після запуску система повністю окупає себе, а річний економічний ефект перевищує 700 000 грн.

Окрім фінансової вигоди, компанія отримує стратегічну перевагу – незалежність від дорогих комерційних рішень, можливість кастомізації під власні потреби та зниження кіберризиків. Таким чином, цей проєкт є не лише економічно доцільним, а й технологічно перспективним для побудови сучасної, безпечної ІТ-інфраструктури..

7.5 Пропозиція алгоритму просування проєкту розробки ПЗ

Просування такого проєкту має починатися з демонстрації практичних результатів – створення тестового середовища, у якому потенційні користувачі можуть на власні очі побачити, як система працює, які функції має та як підвищує ефективність мережевої безпеки. Це допоможе сформувати довіру до продукту ще до його масштабного запуску.

Наступним етапом варто презентувати проєкт на спеціалізованих ІТ-форумах, конференціях та виставках, орієнтованих на системних адміністраторів, кібербезпекові служби та ІТ-директорат компаній. Участь у таких заходах не лише популяризує продукт, а й допомагає налагодити контакти з потенційними партнерами та замовниками.

Важливим кроком є розробка офіційного веб-сайту або платформи, де детально описані функціональні можливості, інструкції та приклади впровадження системи. Це сприятиме поширенню знань серед ІТ-фахівців і підвищить впізнаваність проєкту в ІТ-спільноті.

7.6 Оптимізація каналів збуту та шляхів реалізації ПЗ

Оптимізація збуту такої системи може відбуватись через створення партнерської мережі серед компаній, які займаються постачанням ІТ-рішень і

обладнання для корпоративних мереж. Це дозволить швидко охопити ринок, не витрачаючи значних ресурсів на власний відділ продажів. Крім того, можна пропонувати систему як частину комплексного рішення із забезпечення кібербезпеки, поєднуючи її з іншими відкритими інструментами захисту. Ще одним напрямом є реалізація системи через модель SaaS (Software-as-a-Service), що дозволяє організаціям орендувати систему замість купівлі. Такий підхід знижує бар'єр входу для клієнтів і забезпечує стабільний грошовий потік для розробників. Для підвищення впізнаваності продукту можна запустити освітню кампанію – серію вебінарів і тренінгів, де буде продемонстровано, як Ubuntu можна ефективно використовувати для управління шлюзами. Це не лише допоможе залучити нових клієнтів, а й сформує спільноту користувачів, готових підтримувати розвиток продукту.

7.7 Визначення ключових факторів успіху конкретного проєкту

Основними факторами успіху проєкту є стабільність і надійність системи, а також здатність адаптуватись до різних масштабів підприємств. Система повинна працювати без перебоїв, бути захищеною від зовнішніх загроз і легко інтегруватись у вже наявну IT-інфраструктуру клієнта. Це забезпечить довіру користувачів і сформує позитивну репутацію рішення. Другим ключовим фактором є відкритість і гнучкість. Завдяки Ubuntu система має змогу швидко оновлюватись, отримувати підтримку від світової спільноти та розвиватись відповідно до нових технологічних трендів. Це створює додаткову перевагу над комерційними аналогами, які часто залежать від політики постачальника. Не менш важливим чинником є якісна підтримка користувачів – швидка допомога при налаштуванні, консультації та постійне вдосконалення продукту на основі зворотного зв'язку. Якщо ці елементи реалізовані грамотно, система контролю Інтернет-шлюзів на базі Ubuntu має всі шанси стати популярним, конкурентним і затребуваним рішенням у сфері корпоративної безпеки.

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Впровадження комп'ютерних технологій принципово змінило характер праці різних категорій фахівців. Працівники, використовують комп'ютерну техніку, на своєму досвіді оцінили її величезні можливості. Одночасно виникла певна безтурботність при її експлуатації.

Характерною ознакою сучасного науково-технічного прогресу практично у всіх сферах діяльності людини є широке застосування комп'ютерних технологій, заснованих на використанні електронно-обчислювальних машин (ЕОМ). Сьогодення, а тим більше, майбутнє, вже важко уявити без комп'ютерів та іншої електронної техніки. Адже саме завдяки їм стала можливою швидка переробка величезних обсягів інформації, проведення необхідних розрахунків, виконання різних видів робіт, пов'язаних обробкою текстових та ілюстраційних зображень, організація оперативного отримання та передачі інформації, збереження її значних обсягів електронним способом.

Недотримання вимог безпеки призводить до того, що й через кілька днів роботи за комп'ютером співробітник починає відчувати певний дискомфорт: в нього виникає головний біль і різь у власних очах, з'являються почуття виснаження й дратівливість. В окремих людей порушується сон, погіршується зір, занедужують руки, шия, поперек тощо.

До недоліків умов праці користувачів комп’ютерної техніки можна віднести:

– недостатню площу і обсяг виробничого приміщення;

– недотримання вимог, мікроклімату на робочих місцях;

– низький рівень освітленості у приміщеннях і на робочих поверхнях апаратури;

– підвищений рівень низькочастотних магнітних полів від моніторів;

– порушення вимог організації робочих місць;

– недотримання вимог до режимам праці та відпочинку;

– надмірне виробничу навантаження працівників;

– відсутність навичок зниження впливу психоемоційного напруги.

Відповідно до ст.14 Закону «Про охорони праці» [3] на роботодавця покладено обов'язок забезпечити: безпеку працівників при експлуатації устаткування; застосування коштів індивідуальної захисту працівників; відповідні вимоги охорони праці, умови праці в кожному робоче місце; дотримання режиму праці та відпочинку працівників; навчання безпечним методам і прийомам виконання; інструктаж з охорони праці; організацію контролю над станом умов праці в робочих місць; проведення атестації робочих місць в умовах праці.

Максимально зменшити кількість шкідливих впливів на людину при високій продуктивності праці, створити комфортні умови для роботи людей – ось одна з головних задач охорони праці.

8.2 Шкідливі і небезпечні фактори при роботі з комп’ютером

Електронно-обчислювальні машини (ЕОМ) та інше обладнання є джерелами небезпеки ураження електричним струмом. Оскільки робота програміста характеризується істотним зоровим навантаженням, то вимагає належного освітлення. У приміщенні, в якому працюють програмісти, необхідно створити належний мікроклімат, параметри якого регламентуються Державними санітарними правилами і нормами, зокрема ДСанПіН 3.3.2.007-98 [2].

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;

– електромагнітні (у тому числі високочастотні) випромінювання (коливання);

– несприятливі мікрокліматичні умови;

– нервово-емоційна напруженість праці;

– інтелектуальні навантаження;

– монотонність праці;

– невідповідність ергономічних показників робочого місця діючим вимогам;

– шум;

– статичні навантаження на кістково-м’язовий апарат.

8.3 Аналіз умов праці на робочому місці програміста

Робота програміста пов'язана з постійною роботою на ЕОМ, яка відбувається у кімнаті розмірами 4,8 м×7,2 м×2,8 м. Одна з її більших стін має шість двостулкових вікон, розмірами 2,1 м×1,9 м, які виходять на північний схід. Вікна розташовані рівномірно по всій довжині стіни. Підлога в кімнаті покрита лінолеумом, всі стіни пофарбовані у світло оранжевий колір до висоти 2,8 м, а далі підвісна стеля. Уздовж стін розташовані комп'ютерні столи. На них розташовуються 2 персональні комп'ютери й інша оргтехніка (сканер, принтери, телефони й ксерокс). Столи мають пластикове покриття. Габарити їхньої робочої поверхні 1245 мм×840 мм. Висота столів 750 мм. Висота стільців від рівня підлоги становить 425 мм.

Згідно НПАОП 0.00-1.28-10 «Правила охорони праці під час експлуатації електронно-обчислювальних машин» площа повинна задовольняти умові – не

менш 6 м^2 на одне робоче місце. Кратність повітрообміну в приміщенні також регламентується ДСанПіН 3.3.2.007-98 [2], вона повинна становити 20 м^3/годину на одне місце. Виконання даних вимог забезпечить підтримку в приміщенні оптимального значення вологості й складу повітря.

Відповідно ДБН В.2.5-28-2006 [1] роботу програміста 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, хто працює, на робочу поверхню. Для такого типу приміщень і розряду зорової роботи нормоване значення коефіцієнта природної освітленості (КПО) робочої поверхні (при сполученому освітленні), повинен становити 0,5%, освітленість при штучному освітленні повинна становити 300 лк.

За результатами виміру освітленості відділом охорони праці величина освітленості від системи загального штучного висвітлення лежить у межах 200-250 лк, що не відповідає вимогам, які пред'являються до приміщення.

Відповідно ДСанПіН 3.3.2.007-98 [2] рівні звукового тиску в робочому приміщенні не повинні перевищувати в октавних смугах із середньо геометричними частотами наступних значень, наведених у таблиці 8.1.

Таблиця 8.1 – Допустимі спектри рівнів звукового тиску

Робоче місце	Рівень звукового тиску, дБ, в октавних смугах із середньогеометричними частотами, Гц								Рівень звуку і еквівалентний рівень звуку, дБА
	63	125	250	500	1000	2000	4000	8000	
Приміщення програмістів обчислювальних машин	71	61	54	49	45	42	40	38	50

У приміщенні перебувають наступні джерела шуму: електродвигуни внутрішнього вентилятора ЕОМ; працюючі принтери; працюючі дисководи. Шум, вироблений вентилятором можна класифікувати як постійний, всі інші джерела шуму, як імпульсні. Відповідно паспорта на приміщення рівень звуку, Дб(А), обмірюваний за шкалою (А) шумоміра досяг величини 28,3 Дб(А) при роботі всього устаткування, включаючи й ксерокс. Це дозволяє зробити висновок про відповідність рівня звуку в приміщенні вимогам нормативних актів.

Ергономічні вимоги до робочого місця працюючого з ВДТ ЕОМ і ПЕОМ нормуються НПАОП 0.00-1.28-10**.** Оптимальне положення тіла того, що працює забезпечується відповідною конструкцією робочого місця, а також регуляцією висоти робочої поверхні, сидіння, простори й підставки для ніг. Даного місця програміста не мають регульованих параметрів. Відмінності реальних параметрів робочого місця від параметрів відповідні вимоги нормативного акту дані в таблиці 8.2.

Таблиця 8.2 – Відмінності реальних параметрів робочого місця від параметрів, відповідних вимогам нормативного акту

Зріст людини, см	Висота робочої поверхні мм,	Висота простору для ніг, мм	Висота робочого сидіння, мм
175	765(740)	655(600)	450(440)

У дужках зазначені реальні значення параметрів робочого місця; всі вони не відповідають параметрам, зазначеним у стандарті.

Параметри мікроклімату можуть мінятися в широких межах, тоді як необхідною умовою життєдіяльності людини є підтримка сталості температури тіла завдяки властивості терморегуляції, тобто здатності організму регулювати віддачу тепла в навколишнє середовище.

У приміщеннях, де встановлені комп'ютери, повинні дотримуватися певні параметри мікроклімату. У санітарних нормах ДСН 3.3.6.042-99 встановлені величини параметрів мікроклімату, що створюють комфортні умови. Ці норми

встановлюються в залежності від пори року, характеру трудового процесу і характеру виробничого приміщення (табл. 8.3).

Таблиця 8.3 – Параметри мікроклімату для приміщень, де встановлені комп'ютери

Період року	Параметр мікроклімату	Величина
Холодний	Температура повітря в приміщенні Відносна вологість Швидкість руху повітря	22 – 24°С 40 – 60% до 0,1 м/с
Теплий	Температура повітря в приміщенні Відносна вологість Швидкість руху повітря	23 – 25°С 40 ... 60% 0,1 ... 0,2 м/с

8.4 Розрахункова частина

Для захисного штучного заземлення застосовуються вертикальні електроди: металевий куток 63х63х6 мм (згідно з ДСТУ 2251-93 «Кутики сталеві гарячекатані рівнополичні. Сортамент») довжиною L=1,7 м, та горизонтальний електрод – металева полоса з перетином 60х5 мм. Напруга – 220/380 В. Розрахункова схема розташування заземлюючих електродів – по контуру прямокутником (рис. 8.1).

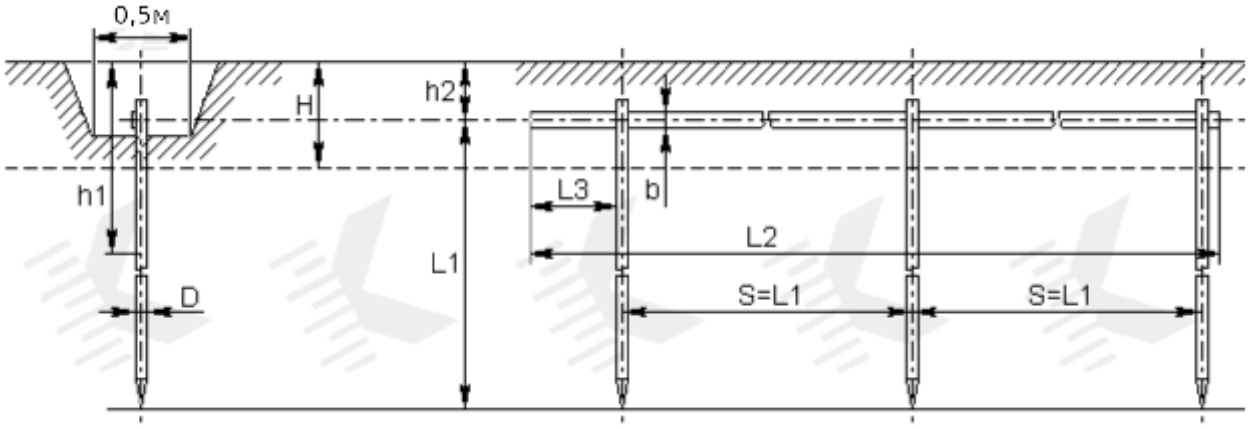

Рисунок 8.1 – Схема штучного заземлення

Розрахунок проведемо за допустимим опором розтіканню струму заземлювача.

Початкові дані для розрахунку захисного заземлення: тип верхнього шару ґрунту – чорнозем, нижнього шару ґрунту – глина (питомий опір ρ_2 = 40 Ом·м). Умовна товщина верхнього шару ґрунту H=0,5 м. Відстань між вертикальними заземлювачами (електродами) A=3 м. Глибина закладення горизонтального контура заземлення t=0,6 м. Опір заземлювача, який нормується, $R_{ЗН}$=4 Ом. Необхідно визначити необхідну кількість вертикальних заземлювачів та довжину полоси (горизонтального заземлювача).

Виконаємо розрахунок.

Відстань від центра вертикального заземлювача до поверхні землі:

$$T = t + L/2 = 0,6 + 1,7/2 = 1,45 \text{ м.}$$

Розрахунковий питомий опір ґрунту (з врахуванням того, що фактично вся конструкція заземлювача розташовується у нижньому шарі ґрунту):

$$\rho = \psi \cdot \rho = 1,36 \cdot 40 = 54,5 \text{ Ом·м.}$$

де ψ = 1,36 – табличне значення коефіцієнту сезонності для відповідної кліматичної зони у багатошаровому ґрунті [12]; ρ_2= 40 Ом·м. – табличне значення питомого опору нижнього шару ґрунту (глина) [12].

Еквівалентний діаметр вертикального електрода (кутка) [12]:

$$D_в = 0,95 \cdot К = 0,95 \cdot 63 = 59,85 \text{ мм} = 0,0598 \text{ м.}$$

де К= 63 мм. – розмір металевого кутка (заданий).

Відношення A/L=3/1,7 =1,76.

Опір розтіканню електричного струму одного електрода вертикального заземлювача з урахуванням заглиблення заземлювача [12]:

$$R_О = 0,366 \cdot (\rho/L) \cdot [\lg(2L/D_в) + (1/2) \cdot \lg((4T+L)/(4T-L))] =$$

$$= 0,366 \cdot (54,5/1,7) \cdot [(\lg(2 \cdot 1,7/0,0598) + (1/2) \cdot \lg((4 \cdot 1,45+1,7)/(4 \cdot 1,45 - 1,7))] = 22 \text{ Ом.}$$

Визначаємо коефіцієнт екранування вертикальних електродів $К_{ев}$=0,62 при орієнтовній кількості вертикальних електродів 5 [12].

Визначаємо необхідну кількість вертикальних електродів заземлювача (без врахування горизонтального заземлювача), при $R_{ЗН}$ = 4 Ом:

$$N=R_О /(К_{ев}\cdot R_{ЗН}) = 22 / (0,62\cdot 4) = 8,9 \approx 9 \text{ шт.}$$

Визначаємо довжину з'єднуючої полоси:

$$L_П =1,05\cdot A\cdot N= 1,05\cdot 3\cdot 9=28,35\approx 28 \text{ м.}$$

Опір розтіканню електричного струму з'єднуючої полоси з урахуванням кліматичного коефіцієнта питомого опору ґрунту $K_П$ [12]:

$$R_П = 0,366\cdot(\rho\cdot K_П/L_П)\cdot \lg(2\cdot L_П^{\,2}/(B\cdot t)) =$$

$$= 0,366\cdot(40\cdot 5/28)\cdot \lg((2\cdot 28^{\,2})/(0,06\cdot 0,6)) = 12,1 \text{ Ом.}$$

де $K_П$=5 – табличне значення кліматичного коефіцієнту питомого опору ґрунту для відповідної кліматичної зони для з'єднуючої полоси [12]:

B = 60 мм = 0,06 м – ширина з'єднуючої полоси (задана).

Загальний опір розтіканню електричного струму заземлювача [12]:

$$R = (R_О\cdot R_П)/(R_О\cdot \eta_П + N\cdot R_П\cdot Кев) =$$

$$= (22\cdot 12,1)/(22\cdot 0,6 + 9\cdot 12,1\cdot 0,62) = 3,35 \text{ Ом.}$$

де $\eta_П$ = 0,6 – табличне значення коефіцієнту екранування з'єднуючої полоси [12].

Умова $R \le R_{ЗН}$ виконується ($3,35 \le 4$).

Оскільки при 9 вертикальних електродах R суттєво менше $R_{ЗН}$, зменшимо кількість вертикальних електродів N до 8 і виконаємо перерахунок. У результаті остаточно отримали: R = 3,7 Ом. при кількості вертикальних електродів N = 8.

8.5 Висновки до розділу

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

З цих міркувань було здійснено аналіз умов праці на робочому місці програміста, проведено розгляд небезпечних та шкідливих факторів, що негативно впливають на програмістів під час роботи.

Тільки повна усвідомленість працівника про можливу шкоду та небезпеку, що можуть підстерігати його на робочому місці, та дотримання вимог нормативних актів з питань охорони праці та відповідних рекомендацій фахівців, дозволять значною мірою знизити негативний вплив шкідливих та небезпечних факторів при роботі з комп’ютером на організм людини.

Виконано розрахунок захисного штучного заземлення, як одного з ключових факторів безпеки програміста.

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи контролю Інтернет шлюзів на базі ОС Ubuntu.

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 контролю Інтернет шлюзів на базі ОС Ubuntu.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем контролю Інтернет шлюзів на базі ОС Ubuntu.

– Досліджена система контролю Інтернет шлюзів на базі ОС Ubuntu.

– На основі отриманих результатів досліджень створена програмна реалізація системи контролю Інтернет шлюзів на базі ОС Ubuntu.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання контролю Інтернет шлюзів на базі ОС Ubuntu.

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 PHP, PERL. 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 Ubuntu.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм IDEA.

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.

Проведено маркетингове та економічне обґрунтування ІТ-проєкту, що дозволило визначити ключові фактори успіху даного проекту.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Сосна О.С. Дослідження та програмна реалізація системи контролю Інтернет шлюзів на базі ОС Ubuntu // Збірник праць молодих науковців ЦНТУ. – Вип. 15. – Кропивницький: ЦНТУ, 2025.

2. Loren Kohnfelder. Designing Secure Software. No Starch Press. 2022. 332 p.

3. Samir Kumar Rakshit. Ethical Hacker's Penetration Testing Guide. BPB Online. 2022. 509 p.

4. Corey J. Ball. Hacking APIs. No Starch Press. 2022. 353 p.

5. Kevin Beaver. Hacking for Dummies. John Wiley & Sons. 2022. 419 p.

6. Mark S. Merkow. Practical Security for Agile and DevOps. CRC Press. 2022. 236 p

7. Derek Fisher. Application Security Program Handbook. Manning Publications. 2021. 155 p.

8. Cameron Wyatt PH.D. Kali Linux Tutorial. Independently published. 2021. 60 p.

9. Alex Matrosov, Eugene Rodionov, Sergey Bratus. Rootkits and Bootkits. No Starch Press. 2019. 450 p.

10. Петрик В.М., Присяжнюк М.М., Аль-Файюмі Халед та ін. «Системи інформаційної зброї та технології інформаційної війни»: підручник / Петрик В.М., Присяжнюк М.М., Аль-Файюмі Халед, Жарков Я.М., Смірнов О.А, Буравченко К.О., Давидюк А.В., Кононович В.Г., Корчинский В.В., Кудирко В.М., Фесенко А.О.; за заг. ред. В.М. Петрика, М.М. Присяжнюка.– К.: Видавничий центр “Кафедра”, 2025.– 320 с.

11. Усік, П.С., Смірнова, Т.В., Буравченко, К.О., Смірнов, О.А., Улічев, О.С., Смірнов, С.А. «Дослідження технологій забезпечення кібербезпеки банківських систем з використанням штучного інтелекту». *Кібербезпека: освіта, наука, техніка.* 2025. Том 1 № 29. С.704–716, 2025

12. Kuznetsov, O., Frontoni, E., Kuznetsova, K., Arnesano, M., Smirnov, O. «A secure biometric authentication architecture for blockchain-driven cyber-physical systems». *Security and Privacy of Cyber Physical Systems Emerging Trends Technologies and Applications*, 2025, pp. 193–224.

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

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

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

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

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

18. Kuznetsov, O., Frontoni, E., Kryvinska, N., Smirnov, O., Imoize, G.L. «Computational Modeling of Enhanced Spread Spectrum Codes for Asynchronous

Wireless Communication». *Computational Modeling and Simulation of Advanced Wireless Communication Systems*, 2024, pp. 403–447.

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

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

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

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

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

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

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

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

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

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

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

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

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

32. Козлов Я.О., Козірова Н.Л., Смірнов О.А. «Дослідження структури та принципу роботи SIEM-системи». *VII міжнародна науково-практична конференція "Інформаційна безпека та комп'ютерні технології" до 30-ти річчя*

кафедри кібербезпеки та програмного забезпечення, м. Кропивницький. 1 листопада 2023 р. – Кропивницький: ЦНТУ. – 2023. – С. 59.

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

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

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

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

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

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

39. Смірнов О.А., Смірнова Т.В., Константинова Л.В., Смірнов С.А., Якименко Н.М., «Дослідження стійкості до лінійного криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку*, 2022, № 1(67). С. 84-89.

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

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

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

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

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

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

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

47. Smirnov O., Kuznetsov A., Arischenko A., Chepurko I., Onikiychuk A., Kuznetsova T. «Pseudorandom sequences for spread spectrum image steganography». *CEUR Workshop Proceedings* Volume 2654, 2020, Pages 122-131.

48. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New technique for data hiding in cover images using adaptively generated pseudorandom sequences». *CEUR Workshop Proceedings* Volume 2654, 2020, Pages 1-14.

49. Smirnov O., Lutsenko M., Kuznetsov A., Kiian A., Kuznetsova T., «Biometric cryptosystems: overview, state-of-the-art and perspective directions». *Lecture Notes in Networks and Systems*, vol 152. Springer, Cham. 2021, pp 66-84.

50. Smirnov O., Kuznetsov A., Onikiychuk A., Makushenko T., Anisimova O., Arischenko A. «Adaptive pseudo-random sequence generation for spread spectrum image steganography». *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Ukraine, Kyiv, May 14-18. 2020. P. 161-165.

51. Smirnov O., Kuznetsov A., Kiian A., Babenko V., Perevozova I.,Chepurko I. «New Approach to the Implementation of Post-Quantum Digital Signature Scheme». *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT),* Ukraine, Kyiv, May 14-18. 2020. P. 166-171.

52. Smirnov O., Kuznetsov A., Kiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-quantum application». *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT),* Ukraine, Kyiv, May 14-18. 2020. P. 172-177.