МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА

ОЧЕРЕТНИЙ СЕРГІЙ ОЛЕГОВИЧ

Допускається до захисту:
Завідувач кафедри прикладної математики та кібербезпеки, д-р філософії з математики
_____ Луценко А. В.
«__»_____20__р.

РОЗРОБКА СИСТЕМИ АНОНІМІЗАЦІЇ ІНТЕРНЕТ-ТРАФІКУ ЗА ДОПОМОГОЮ ПРОКСІ-СЕРВЕРІВ

Спеціальність 125 Кібербезпека
Кваліфікаційна (бакалаврська) робота

Керівник:
Крижановський В. Г.,
професор кафедри прикладної математики та кібербезпеки

_____
(підпис)

Оцінка : _____ / _____ / _____
(бали/за шкало ЄКТС/за національною шкалою)
Голова ЕК: _____ (підпис)

Вінниця – 2025

ЗМІСТ

АНОТАЦІЯ

Очеретний С.О. Розробка системи анонімізації інтернет-трафіку за допомогою проксі-серверів. Спеціальність 125 Кібербезпека . Донецький національний університет імені Василя Стуса. Вінниця. 2025 рік.

У межах дослідження було спроєктовано та реалізовано два прототипи системи для анонімізації мережевого трафіку з використанням проксі-серверів, VPN, Tor та інших інструментів. Проведено аналіз існуючих рішень, сформовано архітектури, здійснено налаштування, тестування та порівняння запропонованих варіантів за критеріями ефективності та рівня захисту.

Ключові слова: анонімізація, проксі-сервер, VPN, Tor, UFW,.

Структура роботи: анотація, вступ, три розділи, висновки, список використаних джерел, додатки

Рис.: 51; Бібліограф.: 43 найм.; табл.: 1.

ВСТУП

Актуальності теми дослідження: Сьогодні інформаційні технології проникають у всі сфери життя, і з кожним роком обсяг персональних даних, що циркулюють у цифровому просторі, постійно зростає. Водночас користувачі далеко не завжди мають уявлення про те, яка саме інформація про них збирається, як вона обробляється, а головне – хто має до неї доступ. Усе частіше звичайні дії в інтернеті стають підставою для створення детального профілю особи, що може бути використано як у комерційних, так і в корисних цілях, пов'язаних зі стеженням або тиском. У такій ситуації важливим завданням стає не лише захист окремих даних, а й створення умов, за яких користувач може залишатися менш ідентифікованим у мережі загалом.

Особливо гостро проблема приватності постає в контексті нестабільної політичної ситуації, воєнного стану та поширення інформаційних атак. В Україні ці виклики відчуваються особливо гостро, що зумовлює потребу в доступних і водночас надійних засобах анонімізації мережевого трафіку. Одним з ефективних рішень у цьому напрямі є використання проксі-серверів, здатних маскувати реальні технічні характеристики пристрою користувача й ускладнювати спроби його відстеження. Однак більшість стандартних конфігурацій або недостатньо ефективні в умовах підвищеної загрози, або ж вимагають глибоких технічних знань. Тому актуальність теми полягає в пошуку рішень, що дозволяють забезпечити гнучке налаштування, підвищений рівень безпеки й можливість застосування в умовах обмежених ресурсів, без втрати функціональності та з урахуванням реальних потреб користувачів.

Мета дослідження: розробка прототипу системи, що забезпечує гнучке управління проксі-серверами та підвищений рівень анонімності.

- **Завдання дослідження:** аналіз існуючих інструментів для роботи з проксі-серверами, проектування архітектури власної системи, реалізація основних функціональних можливостей, тестування розробленого рішення.

- **Об'єкт дослідження**: процес анонімізації інтернет-трафіку за допомогою проксі-серверів.
- **Предмет дослідження:** методи та інструменти для гнучкого управління проксі-серверами з метою підвищення анонімності.
- **Практичне значення:** можливість створення більш ефективного та гнучкого інструменту для забезпечення приватності користувачів в Інтернеті.

РОЗДІЛ І. ТЕХНОЛОГІЧНА АНОНІМІЗАЦІЯ ЯК ІНСТРУМЕНТ СОЦІАЛЬНОГО ЗАХИСТУ

1.1. Важливість цифрової безпеки для вразливих груп населення

У сучасних умовах стрімкої цифровізації суспільства питання захисту персональних даних набуває особливої актуальності. Особливу увагу в цьому контексті слід приділяти представникам вразливих соціальних груп, зокрема людям з обмеженими можливостями, дітям, переселенцям, військовим, а також користувачам з низьким рівнем цифрової грамотності. Ці категорії користувачів є потенційно більш вразливими до кіберзагроз, серед яких варто виокремити несанкціонований збір даних, онлайн-стеження, соціальну інженерію та інші форми інформаційного тиску.

Зазначені виклики зумовлюють потребу у впровадженні надійних технічних засобів, здатних забезпечити базовий рівень конфіденційності та анонімності під час користування мережею Інтернет. Одним із таких засобів є проксі-сервери, що виконують функції проміжної ланки між клієнтом та кінцевим ресурсом. Основним призначенням проксі-сервера є маршрутизація запитів від користувача до цільового ресурсу з можливістю маскування вихідної IP-адреси, що сприяє зниженню рівня цифрової ідентифікації користувача та зменшує ризики пов’язані з втручанням у приватне цифрове середовище [1, 2].

Проксі-сервери як технологія не є новим або маловідомим інструментом у сфері цифрової безпеки. Їх застосування має довгу історію в корпоративному середовищі, де вони використовуються для фільтрації трафіку, захисту внутрішніх мереж, оптимізації швидкості доступу до зовнішніх ресурсів та в процесі тестування програм. Однак у сучасних умовах зростання цифрових ризиків, проксі-сервери отримують інше призначення. У контексті захисту вразливих користувачів вони можуть виступати вже не лише як технічний інструмент, а як засіб забезпечення базових цифрових прав і свобод. Завдяки здатності приховувати особисту інформацію, проксі-сервери можуть бути корисними для тих, хто через обмежені технічні знання або соціальні

обставини не має змоги самостійно захищати власну приватність в інтернеті. У такому підході простежується важлива тенденція – перехід від вузько-функціонального використання до соціально-орієнтованого застосування, що відкриває нові можливості для формування інклюзивного й безпечного цифрового простору [3].

У зв’язку з цим актуалізується потреба не лише в технічному впровадженні засобів захисту, таких як проксі-сервери, а й у глибшому розумінні їхньої ролі з точки зору правових та нормативних вимог. Сучасна інформаційна безпека – це вже не лише сукупність технічних заходів, а й правове поле, в межах якого реалізується право особи на недоторканність приватного життя, безпечне зберігання та передачу даних. Цифрові права, зокрема право на конфіденційність, дедалі частіше визнаються невід’ємною складовою прав людини, і це знаходить своє відображення у низці міжнародних та національних нормативно-правових документів.

Одним із найвідоміших прикладів такого регулювання є Загальний регламент про захист персональних даних (GDPR), що набрав чинності в Європейському Союзі у 2016 році. Він встановлює жорсткі вимоги до обробки, зберігання та передачі персональних даних як у приватному, так і в державному секторах. Зокрема, відповідно до статті 25 *"Data protection privacy by design and by default"* та статті 5*"Principles relating to processing of personal data"*, організації зобов’язані впроваджувати технології, що мінімізують обсяг даних, які збираються про користувача, і зменшують ризики несанкціонованого доступу до них. У такому контексті використання проксі-серверів є одним із практичних інструментів реалізації згаданих принципів. Завдяки маскуванню IP-адреси та приховуванню місцезнаходження користувача, проксі-сервери дозволяють зменшити слід користувача в мережі, що сприяє забезпеченню анонімності й захисту ідентифікаційних даних [4, 8].

На національному рівні в Україні діють закони:

- **«Про захист персональних даних»** – регламентує обробку персональних даних, що важливо при використанні проксі-серверів, які

можуть зберігати IP-адреси та інші дані користувачів. Використання проксі має відповідати вимогам конфіденційності та безпеки персональних даних [5].

- **"Про захист інформації в інформаційно-телекомунікаційних системах"** – визначає загальні вимоги до захисту інформації, включно з організацією безпечного доступу до інформаційних ресурсів, що може включати використання проксі-серверів для контролю та фільтрації трафіку [6].
- **" Про основні засади забезпечення кібербезпеки України" –** встановлює вимоги до захисту інформаційних систем, у тому числі організаційних та технічних заходів, де проксі-сервери можуть використовуватися для моніторингу, фільтрації та захисту трафіку [7].

Крім того, у сфері цифрової трансформації активно розробляються підзаконні акти та технічні стандарти, спрямовані на забезпечення безпеки електронних інформаційних ресурсів, особливо у секторах охорони здоров’я, освіти, соціального захисту тощо. Саме в цих сферах найчастіше працюють з представниками вразливих груп населення, що підвищує відповідальність за захист їхніх даних.

У контексті згаданого правового регулювання проксі-сервери набувають значення не просто як інструмент підвищення технічної безпеки, а як складова частина комплексного підходу до реалізації законодавчих норм у сфері цифрових прав. Їх впровадження може розглядатися як підтвердження дотримання принципів добросовісної обробки даних та демонстрація зобов’язань організацій перед користувачами. Це особливо актуально у випадках, коли йдеться про роботу з особами, які з об’єктивних причин не можуть повноцінно захистити свої інтереси в цифровому середовищі.

Таким чином, проксі-сервери виступають важливою технологічною ланкою, що дозволяє поєднати вимоги інформаційної безпеки з нормативно-правовими стандартами захисту особистих даних. Їх використання не лише покращує захист інформаційних потоків, а й підсилює довіру до цифрових сервісів, у тому числі серед соціально вразливих груп населення.

1.2. Проксі-сервер як інструмент забезпечення анонімності. Порівняння з іншими методами онлайн-захисту

Проксі-сервер являє собою проміжну ланку між клієнтським запитом та сервером, який надає доступ до певних мережевих ресурсів або сервісів. Існує низка типів проксі-серверів, що застосовуються залежно від характеру та цілей клієнтських запитів. Основною функцією таких серверів є забезпечення опосередкованого доступу до інтернет-ресурсів із метою підвищення рівня безпеки та конфіденційності.

Одним з ключових призначень проксі-сервера є маскування реальної IP-адреси користувача, що сприяє досягненню анонімності під час взаємодії з мережею. У результаті зовнішні системи фіксують запити як такі, що походять безпосередньо від проксі-сервера. Окрім цього, проксі-сервери можуть змінювати віртуальну геолокацію користувача, що дозволяє обходити регіональні обмеження доступу до контенту. Така можливість є актуальною в умовах цензури або географічної фільтрації інформаційних ресурсів [9].

Механізм гешування дозволяє проксі-серверу зберігати локальні копії часто запитуваних ресурсів. У ситуаціях, коли кілька користувачів звертаються до одного й того самого веб-ресурсу, сервер може надавати його з кешу, минаючи необхідність повторного звернення до віддаленого джерела. Це суттєво зменшує затримки при завантаженні контенту, знижує загальне навантаження на мережу та оптимізує використання інтернет-каналів. Застосування даного підходу є особливо доцільним у корпоративних мережах та великих організаціях, де важливо забезпечити одночасний доступ великої кількості користувачів до однакових ресурсів за мінімальних витрат трафіку [10].

Проксі-сервери також широко використовуються для контролю за мережею через реалізацію функцій фільтрації трафіку. Ця можливість дозволяє блокувати небажаний або потенційно шкідливий контент, обмежувати доступ до небезпечних чи небажаних вебресурсів, а також

зменшувати обсяг реклами. У багатьох випадках проксі-сервери виступають засобом забезпечення політик безпеки в організаціях: наприклад, у навчальних закладах, офісах або державних установах, де існує потреба в централізованому управлінні доступом до інтернет-ресурсів. У більш складних конфігураціях ці сервери здатні виконувати глибокий аналіз переданих даних, виявляючи ознаки шкідливої активності, спроби несанкціонованого доступу або передачу конфіденційної інформації [11].

Залежно від конкретної реалізації, проксі-сервери можуть бути оснащені розширеним функціоналом, що виходить за межі базових мережевих завдань. Зокрема, одним із таких доповнень є підтримка шифрування трафіку, яке суттєво підвищує рівень безпеки переданих даних і захищає їх від потенційного перехоплення сторонніми суб'єктами. Окрім цього, проксі-сервери можуть функціонувати у складі багаторівневих систем, коли мережевий трафік послідовно проходить через кілька проміжних серверів. Такий підхід забезпечує додатковий рівень анонімності користувача, ускладнюючи виявлення джерела запиту [12].

Ці функціональні особливості роблять проксі-сервери універсальним інструментом для вирішення широкого спектра завдань – від захисту корпоративних мереж до забезпечення приватності й цифрової безпеки в умовах обмеженого доступу до інформації. Вони ефективно поєднують у собі функції маршрутизації, безпеки й анонімізації, що зумовлює їхнє активне використання як у приватному секторі, так і в державних або комерційних структурах [2, 11].

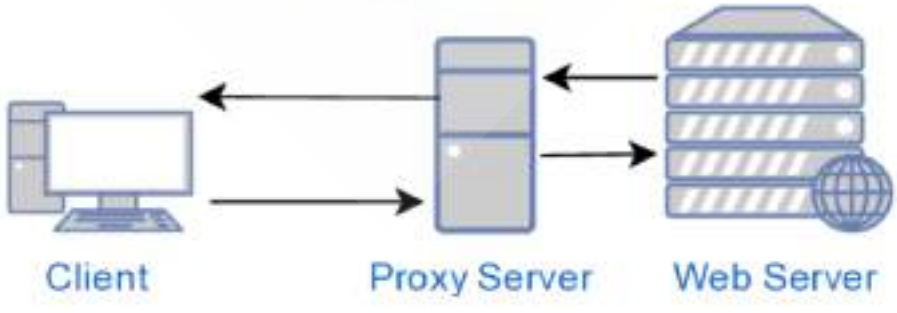


Рис.1.1 Приклад звичайної роботи проксі-сервера

Окрім проксі-серверів існують інші інструменти, завдяки яким можна здійснювати пошук та перебувати на різноманітних інтернет-порталах та більш безпечним чином. До таких методів відносять – анонімайзери та VPN (Virtual Private Network).

Віртуальні приватні мережі (VPN) являють собою один із найбільш надійних інструментів для захисту інтернет-з'єднання та забезпечення конфіденційності користувача. На відміну від проксі-серверів, які переважно виконують функції маршрутизації окремих запитів, VPN створюють повноцінне зашифроване з’єднання між пристроєм користувача та віддаленим сервером, через яке передається весь мережевий трафік. Такий підхід дозволяє ефективно захистити дані від перехоплення, забезпечуючи високий рівень інформаційної безпеки під час роботи навіть у відкритих або ненадійних мережах, наприклад, публічних точках доступу Wi-Fi.

Технологія VPN передбачає створення захищених тунелів із використанням сучасних методів шифрування та автентифікації, зокрема, за допомогою паролів, токенів або інших засобів ідентифікації користувача. Завдяки цьому забезпечується постійна цілісність і конфіденційність переданої інформації, а також мінімізується ризик несанкціонованого доступу. Крім того, VPN приховують реальну IP-адресу користувача, дозволяючи обходити географічні обмеження та регіональні блокування ресурсів.

Попри значні переваги у сфері безпеки, VPN-сервіси мають і певні технічні обмеження. Через складність обробки шифрованого трафіку можливе зниження швидкості з’єднання, особливо при використанні високорівневих протоколів шифрування або при з'єднанні через віддалені сервери. Також VPN-клієнти можуть потребувати додаткових системних ресурсів, що слід враховувати при їх інтеграції в інфраструктуру кінцевих користувачів або організацій [13].

Анонімайзери становлять собою спеціалізовані веб-сервіси, призначені для забезпечення анонімного доступу до інтернет-ресурсів шляхом опосередкування запитів користувача через власні сервери. Вони не

потребують попередньої конфігурації або встановлення додаткового програмного забезпечення, що робить їх особливо зручними для короткочасного чи одноразового використання в умовах обмеженого часу або технічних ресурсів. Завдяки простоті застосування, анонімайзери часто використовуються для обходу обмежень доступу до окремих сайтів у навчальних закладах, офісах чи публічних мережах. Однак, незважаючи на зручність, рівень безпеки, який вони забезпечують, залишається досить низьким. Здебільшого такі сервіси не підтримують шифрування трафіку, що робить передані дані вразливими до перехоплення або аналізу сторонніми особами. Крім того, велика кількість безкоштовних анонімайзерів має обмежений функціонал, нестабільну роботу або викликає сумніви щодо надійності, що ставить під питання доцільність їх використання у контексті захисту конфіденційної інформації. У зв’язку з цим, для тривалого або більш захищеного користування мережею доцільніше застосовувати більш надійні технології, такі як VPN або багаторівневі проксі-системи [14].

Кожен з цих методів має свої сфери застосування. Проксі-сервери підходять для швидкого доступу до заблокованих ресурсів, VPN забезпечують надійний захист даних і конфіденційність, а анонімайзери зручні для разового використання. Вибір між цими методами залежить від потреб користувача та рівня безпеки, який він хоче забезпечити.

Варто також враховувати, що іноді краще використовувати комбінацію цих методів. Наприклад, можна використовувати VPN для забезпечення високого рівня безпеки при роботі з конфіденційними даними, а проксі-сервери для обхід географічних обмежень при перегляді контенту.

1.3. Типи проксі-серверів

Класифікувати проксі-сервери можна кількома способами: за їхньою функціональністю і за протоколами

1) **Прямий проксі-сервер –** це вид проксі, який зазвичай спрямовує запити від клієнтів внутрішньої мережі до мережі Інтернет, використовуючи брандмауер. Прямі проксі конфігуруються для "дозволу" або "заборони"

користувацьких звернень до ресурсу, через брандмауер, з метою отримання доступу до даних в інтернеті. У випадку, коли проксі допускає запит клієнта, він переспрямовує його до веб-сервера через брандмауер. Веб-сервер відправляє відповідь на проксі. Проксі, у свою чергу, надсилає отриману відповідь назад до користувача.

Прямий проксі перш ніж отримувати інформацію з сервера, робить перевірку, чи є запитувані користувачем дані в кеші. Проксі зберігає кешовані дані сам, виключаючи необхідність звертатися до сервера. Якщо запитана інформація доступна в кеші, проксі надсилає її безпосередньо користувачеві. У разі, якщо проксі відхиляє запит клієнта, він передає йому інформацію про помилку або робить перенаправлення [15].

Рис.1.2 Принцип роботи прямого проксі-сервера

2) Зворотний проксі-сервер

Зворотний проксі-сервер (reverse proxy) є одним із ключових типів проксі-серверів, що широко використовується в архітектурі сучасних мережевих систем, зокрема для забезпечення безпечного та контрольованого доступу до внутрішніх ресурсів організацій. На відміну від прямого проксі, який діє від імені клієнта, зворотний проксі функціонує як посередник між зовнішніми користувачами з мережі Інтернет і внутрішніми серверами, прихованими за межами брандмауера. Його основна мета – перехоплення вхідних запитів і їхня переадресація на відповідні контент-сервери у приватній або інтрамережі.

Такий механізм дозволяє ефективно ізолювати внутрішні інформаційні ресурси, зменшуючи ризик прямого доступу з боку сторонніх користувачів. Це особливо актуально для захисту конфіденційних даних або критично важливих сервісів, що потребують додаткового рівня безпеки. Зворотний

проксі також виступає в ролі шлюзу безпеки, дозволяючи адміністратору здійснювати централізований контроль над трафіком, що надходить із зовнішнього середовища, та реалізовувати політики доступу до внутрішніх сервісів.

Додатковою перевагою зворотного проксі-сервера є можливість кешування відповідей від внутрішніх серверів. У разі активованого кешування сервер здатен обслуговувати повторювані запити, використовуючи збережену копію інформації, що значно знижує навантаження на серверну інфраструктуру та скорочує час відповіді. Таким чином, зворотний проксі не лише посилює захист мережі, а й оптимізує її продуктивність, забезпечуючи балансування навантаження та покращення користувацького досвіду [16, 17].

Рис. 1.2 Принцип роботи зворотнього проксі сервера

- *Проксі-серверів за місцем розташування:*

1. **Резидентний проксі-сервер.**

Резидентні проксі-сервери (residential proxies) представляють собою тип проксі-технології, при якій використовуються IP-адреси, видані користувачам провайдерами зв’язку (ISP) для звичайного домашнього доступу до мережі Інтернет. Завдяки цьому трафік, що надходить через такі проксі, має вигляд автентичного запиту звичайного користувача, що значно ускладнює його ідентифікацію як посередницького або автоматизованого. Така особливість дозволяє резидентним проксі забезпечувати високий рівень маскування, роблячи їх особливо привабливими для завдань, де важлива непомітність – зокрема, при зборі публічних даних (web scraping), обході антибот-захистів або регіональних обмежень.

Зазвичай резидентні проксі функціонують у закритому режимі доступу: лише користувачі, які мають комерційний доступ (наприклад, за передплатою), можуть використовувати IP-адреси, що входять до пулу

певного оператора або сервісу. У цьому контексті резидентні проксі часто порівнюються з відкритими проксі-серверами, оскільки мають подібні технічні властивості щодо маршрутизації трафіку. Водночас ключовою відмінністю є характер їх адміністрування – на відміну від відкритих проксі, резидентні, як правило, перебувають під централізованим керуванням і не є загальнодоступними, що підвищує їхню надійність і контрольованість з боку оператора [11].

2. Мобільний проксі-сервер

Мобільні проксі-сервери можна налаштувати за допомогою телефонної мережі 3G або 4G. Цей тип проксі-серверів працює як проміжне з'єднання з Інтернетом за допомогою стороннього постачальника послуг. Мобільні проксі-сервери можна використовувати тільки на смартфонах або планшетах, оскільки вони вимагають використання SIM-карти. Мобільний проксі-сервер використовує IP-адреси, які динамічно видаються мобільним пристроям оператором мобільного зв'язку (MNO), який також є їхнім інтернет-провайдером (ISP) [18].

- *Типи проксі-серверів за рівнем анонімності:*

1. Прозорий проксі-сервер

Прозорий проксі - це сервер-посередник, який не маскує IP-адресу клієнта. Такі проксі не гарантують анонімність, отже їх використання є недоречним для збільшення приватності в мережі. Попри це, їх не створили даремно. Прозорі проксі можуть використовуватися для кешування з метою пришвидшення завантаження контенту.

2. Анонімний проксі-сервер

Анонімний проксі-сервер приховує IP-адресу користувача від веб-сайтів. Проте, він може розкривати деякі інші дані, наприклад, тип браузера, мовні налаштування та версію операційної системи клієнта. Хоча ця інформація сама по собі не ідентифікує користувача, вона може суттєво обмежити коло потенційних кандидатів.

3. Проксі-сервер високої анонімності

Проксі-сервер з високою анонімністю маскує IP-адресу користувача та будь-яку іншу ідентифікаційну інформацію, що передається у HTTP-заголовках, наприклад, тип браузера та операційної системи. Завдяки використанню такого проксі-сервера, відстеження IP-адреси чи пристрою користувача стає набагато важчим завданням. З точки зору забезпечення приватності, це найбезпечніший тип проксі-серверів [19].

4. Ротаційний проксі-сервер

Ротаційні проксі (rotating proxies) - це різновид проксі-серверів, які працюють з динамічним набором IP-адрес, що автоматично або за певним графіком змінюють свою адресу при кожному новому запиті або через визначені проміжки часу. Цей механізм дозволяє зменшити ймовірність виявлення автоматизованих операцій та уникнути блокування на цільових серверах, адже кожен запит надсилається з іншої IP-адреси, імітуючи активність багатьох окремих користувачів.

Через цю особливість ротаційні проксі широко використовуються в тих областях, де потрібне масове генерування запитів, не втрачаючи доступу до ресурсів. Це стосується, зокрема, веб-скрейпінгу, автоматизованого збору відкритих даних, моніторингу цін, конкурентної розвідки та тестування веб-сайтів. Рівень анонімності в таких системах може бути від середнього до високого, і залежить він, здебільшого, від розміру пулу IP-адрес, частоти їх заміни та рівня управління проксі-інфраструктурою. Чим більший та різноманітніший пул IP-адрес, тим більша ймовірність уникнути фільтрації та блокування, що робить ротаційні проксі корисним інструментом для виконання масштабних і повторюваних мережевих задач [20].

- *Проксі-серверів за доступом*

1. Публічний проксі-сервер.

Публічний проксі-сервер відкритий для всіх. До нього можна отримати доступ без необхідності проходити аутентифікацію. Природно, така доступність робить публічні проксі повільними і менш надійними. Вони

мають найвищий ризик потрапляння до чорного списку і блокування веб-сайтами та пошуковими системами [8].

2. Приватні проксі-сервери.

Приватні проксі-сервери (private proxies), також відомі як виділені, - це тип проксі-інфраструктури, який надається у користування одному користувачу чи окремій організації. Технічною особливістю є виділена IP-адреса та порт, які не діляться з іншими користувачами. Це забезпечує максимальну ізоляцію мережевого трафіку, що важливо для конфіденційності, захисту даних та стабільної роботи в мережі.

З'єднання з приватними проксі зазвичай захищені аутентифікацією, що обмежує доступ лише для авторизованих користувачів. Це робить їх ідеальними для використання в критично важливих інформаційних системах або в умовах підвищених вимог до безпеки даних. На відміну від публічних чи загальнодоступних проксі, приватні мають вищу ціну. Це обумовлено ексклюзивним використанням ресурсів, а також кращим контролем, захищеністю та стабільністю з'єднання [21].

- *Проксі-серверів за протоколом:*

1. HTTP та HTTPS-проксі

HTTP-проксі-сервери є різновидом проксі-технологій, які функціонують на рівні протоколу HTTP і здійснюють посередництво між клієнтом та веб-сервером під час обміну мережевими запитами. Вони ефективно використовуються для кешування веб-контенту, що дозволяє зменшити час завантаження сторінок і знизити навантаження на вихідний сервер. Крім того, HTTP-проксі можуть виконувати функції фільтрації контенту, блокування небажаних ресурсів, а також приховування IP-адреси користувача з метою забезпечення базового рівня анонімності. Проте, через те що HTTP-проксі працюють виключно з нешифрованим текстовим трафіком, їх використання для обробки конфіденційної інформації є небажаним з міркувань безпеки.

На відміну від них, HTTPS-проксі забезпечують посередництво у передачі зашифрованих даних через протокол HTTPS, що дозволяє створити

безпечне з'єднання між клієнтом і проксі-сервером. Принцип їхньої роботи загалом подібний до HTTP-проксі, однак головною відмінністю є використання шифрування, яке запобігає доступу до вмісту переданого трафіку. Через це HTTPS-проксі не здатні зчитувати або кешувати дані, що проходять через них, проте саме ця характеристика робить їх доцільними для обробки чутливої інформації та використання в умовах підвищених вимог до конфіденційності. Таким чином, вибір між HTTP- та HTTPS-проксі має ґрунтуватися на співвідношенні між вимогами до безпеки, продуктивності та характером даних, що передаються [22].

2. SSL-проксі

SSL-проксі-сервери, що функціонують на основі протоколу захищених сокетів (Secure Sockets Layer), призначені для забезпечення шифрування даних, які передаються між клієнтом і сервером у двосторонньому напрямку. Це дозволяє значно підвищити рівень захисту інформаційного обміну, зокрема при обробці чутливих або конфіденційних даних у відкритих або потенційно небезпечних мережах. На відміну від звичайних проксі-серверів, SSL-проксі забезпечують не лише маскування IP-адреси користувача, але й збереження цілісності та конфіденційності переданої інформації за рахунок криптографічного захисту.

Завдяки високому рівню безпеки, який забезпечує даний тип проксі, SSL-проксі є доцільним вибором для корпоративних середовищ, банківських установ, медичних організацій та інших суб’єктів, що працюють із персональними або критично важливими даними. Використання таких серверів дозволяє реалізувати захищені мережеві з’єднання, мінімізувати ризики перехоплення трафіку та забезпечити відповідність вимогам інформаційної безпеки на рівні сучасних стандартів [23].

3. FTP-проксі

FTP-проксі-сервери призначені для обробки трафіку, що передається за протоколом передачі файлів (File Transfer Protocol, FTP), між клієнтськими пристроями та віддаленими серверами. Основною функціональною задачею

такого типу проксі є посередництво у процесі передачі файлів, що здійснюється через FTP-протокол, із можливістю реалізації додаткових механізмів контролю, моніторингу та фільтрації даних [24].

4. DHCP-проксі

У режимі DHCP-проксі віртуальна IP-адреса контролера використовується для всіх DHCP-транзакцій, що проходять між клієнтом і сервером, приховуючи реальну IP-адресу DHCP-сервера. Віртуальна адреса відображається в логах налагодження. Коли від зовнішніх серверів надходить кілька пропозицій, проксі вибирає першу і встановлює її в структурі даних клієнта. Усі подальші транзакції проходять через цей сервер, поки не відбудеться невдача, після чого проксі вибирає інший сервер для клієнта. DHCP-проксі за замовчуванням увімкнено, і всі контролери в списку мобільності повинні мати однакові налаштування цього режиму [25].

5. SOCKS-проксі

SOCKS-проксі-сервери спроектовані для обробки TCP/IP-трафіку, який надходить від клієнтських пристроїв. Вони мають універсальний характер і здатні взаємодіяти з будь-яким видом інтернет-трафіку, оскільки їх функціонування не обмежується специфічними мережевими протоколами. SOCKS-проксі ігнорують тип даних, що передаються, незалежно від того, чи це HTTP, FTP або інші протоколи, що значно розширює їх гнучкість у застосуванні. Залежно від версії, найактуальнішою на сьогодні є SOCKS5, яка надає розширені функції, зокрема аутентифікацію, шифрування та сумісність з IPv6.

Оскільки SOCKS-проксі працюють на низькому (мережевому) рівні, вони не змінюють і не обмежують характер трафіку, який через них проходить, що додає додаткової гнучкості та відкриває можливості для різних мережевих задач. Це робить їх практичними для анонімного веб-серфінгу, подолання обмежень доступу, і навіть для використання в складних мережевих системах, де потрібні високий рівень конфіденційності та анонімність користувача [1, 26].

6. SIP-проксі

SIP-проксі-сервери діють як посередники між пристроями, що використовують SIP (Session Initiation Protocol), зокрема, телефонами або іншими засобами комунікації. Вони відіграють ключову роль в організації телефонних розмов у мережах, опрацьовуючи запити до SIP-реєстру та направляючи виклики до потрібних пристроїв або серверів. Завдяки SIP-проксі можна ефективно контролювати процеси початку, маршрутизації та закінчення сесій зв'язку, забезпечуючи коректне з'єднання між користувачами всередині мережі.

Цей вид проксі-серверів забезпечує централізоване керування викликами, а також маршрутизацію через реєстрацію SIP-пристроїв в реєстрах. SIP-проксі також здатні фільтрувати дзвінки, контролювати доступ та надавати додаткові інструменти для гарантування безпеки та високої якості зв'язку в мережах VoIP (Voice over IP). Вони є критичними компонентами для оптимізації та конфігурування систем комунікацій в корпоративних мережах та великих інфраструктурах зв'язку [27].

1.4. Висновки до І розділу

У першому розділі дипломної роботи було розглянуто технологічну анонімізацію як засіб соціального захисту вразливих категорій користувачів у цифровому середовищі. Проаналізовано особливості застосування проксі-серверів як одного з ключових інструментів забезпечення конфіденційності та приватності в мережі Інтернет. Розкрито їхні функції, типологію, рівні анонімності, технічні можливості, а також порівняно з альтернативними методами онлайн-захисту, такими як VPN та анонімайзери.

Окрему увагу приділено правовим аспектам використання проксі-серверів, зокрема у контексті національного та міжнародного законодавства щодо захисту персональних даних. Зроблено висновок, що проксі-технології, завдяки своїй гнучкості та адаптивності, можуть бути ефективно інтегровані в інфраструктури цифрового захисту вразливих груп населення. Вони не лише

посилюють технічний рівень інформаційної безпеки, а й сприяють дотриманню цифрових прав людини в умовах зростаючих кіберзагроз.

РОЗДІЛ ІІ. ПРОЕКТУВАННЯ ПРОТОТИПУ СИСТЕМИ АНОНІМІЗАЦІЇ ІНТЕРНЕТ-ТРАФІКУ НА ОСНОВІ ПРОКСІ-СЕРВЕРУ

У цьому розділі проводиться аналіз для практичної реалізації проксі-серверу, що включатиме в себе подальше тестування. Основна увага приділятиметься архітектурі системи, завданням, які має виконувати прототип та технічним складовим, що застосовуються для розгортання проксі-серверу для досягнення певних результатів.

2.1. Складові мережевої конфіденційності

Існує чимало технологій, що мають на меті забезпечити більш безпечну роботу в мережі Інтернет. Одними із них є інструменти, що дозволяють **анонімізувати мережеві запити через проксі**. Основною задачею цієї функції є часткове або повне приховування IP-адреси користувача з метою збереження його приватності. Реальна IP-адреса користувача замінюється на IP-адресу проксі-сервера, відповідно ідентифікація фізичного місцезнаходження та інтернет-провайдера користувача ускладнюється.

Механізм анонімізації мережевих запитів із використанням проксі-серверів базується на принципах посередництва, багаторівневої маршрутизації та шифрування даних. Збільшення кількості незалежних вузлів у ланцюжку передачі, а також додаткових шарів шифрування, дозволяє досягти вищого рівня анонімності користувача. Однак така архітектура може призводити до зниження швидкості обміну даними та загальної продуктивності мережі [28, 29].

Далі, розглядаємо **аутентифікацію користувачів.** Аутентифікація дозволяє проксі-серверу перевіряти, чи має клієнт дозвіл на використання його ресурсів. Існують кілька видів:

- *Парольна аутентифікація:* Користувач вводить логін і пароль для доступу до проксі. Це стандартний метод, що забезпечує базовий контроль доступу.
- *IP-аутентифікація:* Доступ дозволяється лише для клієнтів із заздалегідь зареєстрованими IP-адресами.

- *Двофакторна аутентифікація (2FA):* Використовуються додаткові механізми, наприклад, SMS-коди або додатки для підтвердження особи.
- *Сертифікати SSL:* Використання цифрових сертифікатів для ідентифікації користувачів, що забезпечує високий рівень безпеки.
- *LDAP і Active Directory:* Інтеграція з корпоративними системами керування ідентифікацією для автоматизації доступу [23, 30].

Наступна функція – **логування та моніторинг**. Проксі-сервери можуть вести детальний облік усіх запитів, що проходять через них:

- *Журнали активності (логи):* Зберігається інформація про час, IP-адресу клієнта, відвідані ресурси, тип трафіку та обсяг переданих даних.
- *Аналітика трафіку:* Проксі-сервери надають звіти про використання ресурсів, популярні запити, час роботи тощо.
- *No-log політика:* Деякі проксі-сервіси не зберігають журнали, щоб забезпечити максимальну анонімність [1, 31].

Ще одну з ключову роль відіграє **шифрування.** Дозволяє захистити дані від несанкціонованого доступу під час їх передавання каналами зв’язку. Суть цього процесу полягає у перетворенні відкритої інформації в зашифрований вигляд за допомогою криптографічних алгоритмів. Розшифрування таких даних можливе лише за наявності відповідного ключа, що гарантує контроль над доступом до змісту переданої інформації.

Серед основних функцій шифрування у контексті мережевої безпеки можна виокремити: забезпечення конфіденційності даних, перевірку автентичності джерела інформації та контроль цілісності переданих повідомлень. У сучасних системах використовуються як симетричні алгоритми (наприклад, AES), так і асиметричні (зокрема, RSA), кожен з яких має свої особливості й сфери застосування [32, 33].

Шифрування є базовим елементом багатьох сучасних технічних рішень, таких як:

- *Tor* – система багаторівневої маршрутизації, яка забезпечує анонімність користувача завдяки застосуванню багатошарового шифрування [34];
- *VPN* – створюють захищені канали зв’язку між користувачем і сервером, використовуючи протоколи OpenVPN, WireGuard, IPsec та інші [13];
- *TLS (HTTPS)* – протокол, що шифрує дані при взаємодії клієнта з веб-сервером і забезпечує захищене веб-з’єднання [34].

Також однією із важливих та необхідних складових для забезпечення мережевої безпеки – є **інтеграція з іншими інструментами захисту** для формування комплексної системи оборони мережі. Такий підхід дозволяє ефективно протидіяти широкому спектру загроз, забезпечуючи не лише конфіденційність переданих даних, а й контроль доступу, моніторинг активності та виявлення аномалій:

- *Фаєрволи (мережеві екрани)* – контролюють вхідний і вихідний трафік на основі визначених правил доступу. Інтеграція з шифруванням дозволяє створювати правила для зашифрованих каналів та обмежувати підозрілу активність, навіть якщо вміст передавання недоступний для аналізу [33].
- *Системи виявлення та запобігання вторгненням (IDS/IPS)* – аналізують мережевий трафік з метою виявлення потенційно небезпечної поведінки. Шифрування може ускладнювати роботу таких систем, тому часто використовується комбінація – наприклад, розміщення IDS до моменту шифрування (на клієнтському рівні) або використання проксі-серверів для розшифрування трафіку перед аналізом [33, 35].

2.2. Архітектури систем

Для забезпечення різних сценаріїв використання, а також підвищення анонімності, контролю доступу та безпеки, було реалізовано дві альтернативні архітектури проксі-серверів. Обидві моделі базуються на модульному підході,

що дозволяє масштабувати систему та адаптувати її до конкретних потреб користувача чи організації.

2.2.1. Архітектура І проксі-серверу

Перша архітектура поєднує класичний проксі-сервер з VPN-клієнтом та фаєрволом. Така конфігурація дозволяє забезпечити контрольований доступ до зовнішніх ресурсів із додатковим рівнем шифрування та маршрутизації трафіку.

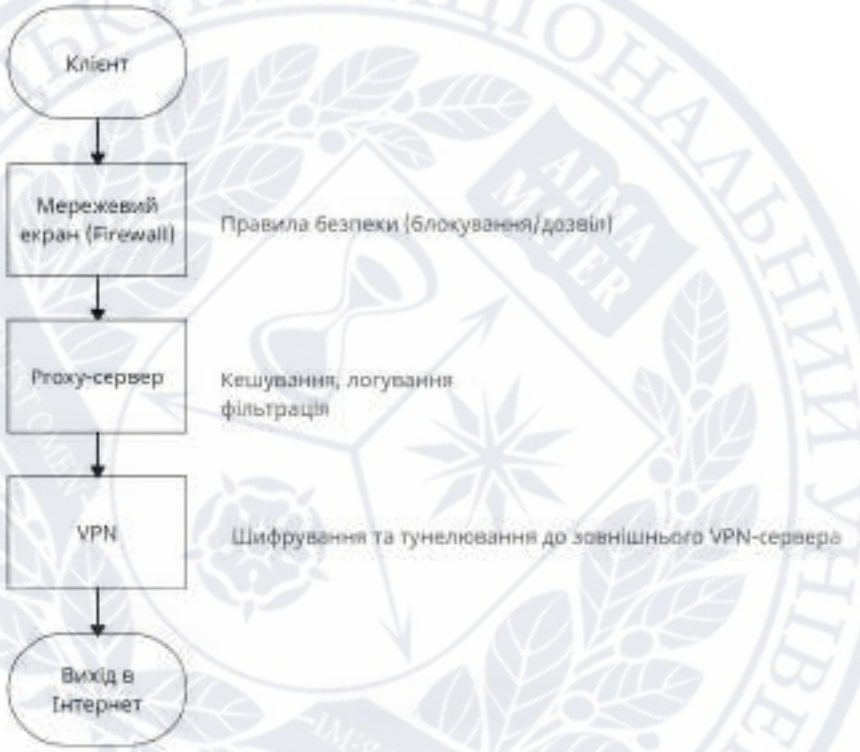


Рис. 2.1. Схема архітектури проксі серверу в поєднанні із мережевим екраном та VPN

Структура системи передбачається відповідно до Рис.2.1.:

- *Фаєрвол* – застосовується для обмеження мережевої активності, зокрема для контролю доступу за IP-адресами, портами чи протоколами.
- *Проксі-сервер* – виконує функції проміжної ланки між клієнтом та зовнішнім сервером, включаючи кешування, фільтрацію вмісту, а також логування запитів.

- *VPN-клієнт* – відповідає за створення захищеного тунелю з віддаленим сервером, що підвищує конфіденційність і знижує ризик перехоплення даних.

Завдяки використанню VPN забезпечується захищене з’єднання із зовнішніми сервісами.

2.2.2. Архітектура II проксі-серверу

Друга архітектура орієнтована на досягнення максимальної анонімності користувача та підвищеної керованості процесом обробки трафіку. Вона поєднує декілька технологій для багаторівневої маршрутизації й гнучкого налаштування:

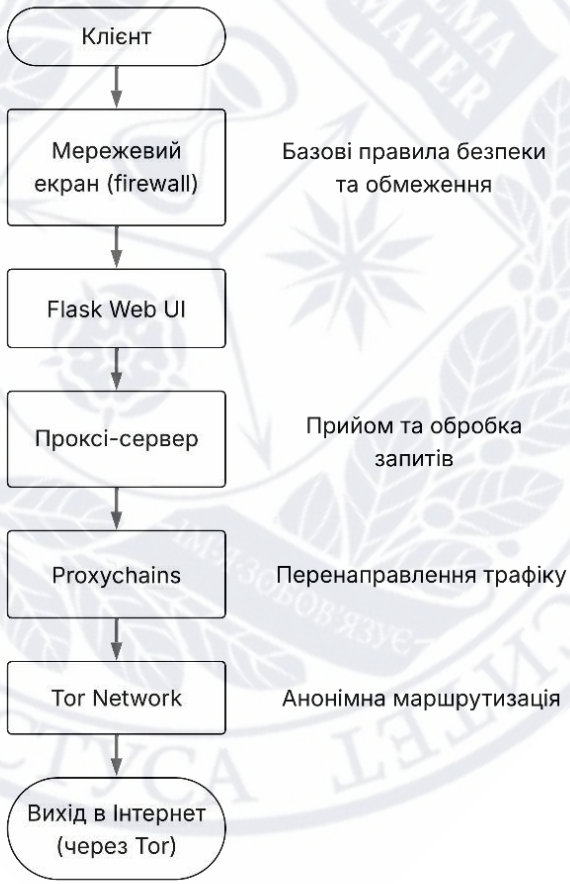


Рис. 2.2. Схема архітектури II проксі серверу

- *Фаєрвол* – забезпечує первинний рівень захисту системи.

- *Flask Web UI* – реалізує інтерфейс взаємодії з користувачем для конфігурації системи, моніторингу та керування мережею через веб-браузер [36].
- *Проксі-сервер* – приймає та обробляє HTTP/HTTPS/SOCKS-запити, реалізує політики доступу та зберігає журнали активності.
- *ProxyChains* – дозволяє направляти мережеві запити через визначений ланцюг проксі-серверів [37].
- *Tor* – виконує багатоступеневу маршрутизацію трафіку через децентралізовану мережу вузлів для забезпечення високого рівня анонімності [34].
- *SSL* – гарантує безпечну передачу даних у зашифрованому вигляді між компонентами системи та з користувачем [23].

Ця архітектура підходить для середовищ, де пріоритетом є анонімність, гнучкість маршрутизації та зручність адміністрування.

2.3. Технічні інструменти для побудови системи

Для розгортання проксі-серверів, було обрано таке програмне забезпечення:

1) **Ubuntu OS** – це безкоштовний дистрибутив операційної системи Linux, заснований на Debian Linux. Система має відкритий вихідний код, що дозволяє користувачам вільно завантажувати, використовувати та модифікувати його. Операційна система Ubuntu підтримує широкий спектр програмного забезпечення. Вона також відзначається високим рівнем безпеки завдяки регулярним оновленням і тестуванню на вразливості [38].

2) **Dante** – це продукт, що складається з SOCKS-сервера і SOCKS-клієнта. Це гнучкий продукт, який можна використовувати для забезпечення зручного та безпечного підключення до мережі.

Після встановлення SOCKS-сервер Dante у більшості випадків може бути налаштований таким чином, щоб бути прозорим для клієнтів, виконуючи функції, які частково відповідають характеристикам так званого непрозорого

маршрутизатора четвертого рівня моделі OSI. Для користувачів, які прагнуть здійснювати контроль та моніторинг мережевих з'єднань як всередині, так і за межами своєї мережі, Dante надає низку переваг. Серед них – підвищений рівень безпеки за рахунок завершення TCP/IP-з'єднань (що виключає прямий контакт між внутрішніми й зовнішніми хостами), можливість обмеження ресурсів (зокрема, пропускної здатності та кількості сеансів), а також розширене журналювання з фіксацією інформації про джерело запиту та обсяг переданих даних [40].

3) **3Proxy** – компактним багатоплатформним програмним комплексом проксі-серверів, що підтримується в операційних системах Linux/Unix та Windows, включаючи 64-бітні версії. Цей інструмент надає широкі можливості для роботи з різними типами мережевого трафіку, включаючи HTTP-проксі з підтримкою протоколів HTTPS і FTP, SOCKS-проксі (версій v4, v4.5 та v5), а також спеціалізовані проксі для POP3, SMTP, AIM/ICQ, MSN Messenger і Live Messenger.

У додаток до цього, 3proxy включає в себе DNS-проксі з функцією кешування, а також засоби для TCP- та UDP-перенаправлення портів. Комплекс містить також універсальний проксі-сервер, який забезпечує розширене керування мережею, зокрема підтримку механізмів обмеження пропускної здатності та трафіку на основі денних, тижневих або місячних лімітів, можливість переспрямування з’єднань і побудови ланцюгів проксі, а також інтеграцію з системами журналювання через ODBC та syslog [41].

4) **UFW** – це стандартний інструмент для налаштування брандмауера в операційній системі Ubuntu, який спрощує роботу з низькорівневими правилами iptables. Він розроблений як зручний інтерфейс для конфігурації мережевого фільтрування на рівні хоста з підтримкою як IPv4, так і IPv6. За замовчуванням UFW неактивний, що передбачає його ручне увімкнення та налаштування відповідно до потреб користувача або адміністратора системи [39].

5) **Flask Web UI –** це засіб взаємодії з користувачем, що забезпечує можливість налаштування системи, здійснення моніторингу та керування мережевими процесами за допомогою веб-браузера [36].

6) **Проксі-ланцюги (Proxychains) –** це програма для UNIX, яка дозволяє нам маскувати нашу IP-адресу, перенаправляючи мережевий трафік. Вона спрямовує наш TCP-трафік через різні проксі-сервери, зокрема TOR, SOCKS і HTTP [37].

7) **Tor** – це безкоштовний мережевий протокол з відкритим вихідним кодом, який забезпечує анонімність і конфіденційність інтернет-користувачів. Він працює шляхом маршрутизації інтернет-трафіку через мережу волонтерських серверів, які називаються «вузлами» або ретрансляторами.

Трафік шифрується кілька разів, коли він проходить через різні вузли, причому кожен шар шифрування знімається на наступному вузлі. Це ускладнює для будь-кого, хто спостерігає за мережевим трафіком, з'ясування того, звідки він прийшов або куди йде [34].

8) **OpenSSL** – Бібліотека криптографії OpenSSL надає доступ до широкого спектру криптографічних алгоритмів, що використовуються в різних стандартах Інтернету. Сервіси, що надаються цією бібліотекою, використовуються в OpenSSL реалізаціях TLS і CMS, а також були використані для реалізації багатьох інших продуктів і протоколів сторонніх розробників.

Функціонал включає симетричне шифрування, криптографію з відкритим ключем, узгодження ключів, обробку сертифікатів, криптографічні хеш-функції, криптографічні генератори псевдовипадкових чисел, коди автентифікації повідомлень (MAC), функції отримання ключів (KDF) та різні утиліти [42].

9) **OpenVPN** – це засіб створення захищених з’єднань, який базується на використанні протоколів SSL/TLS для шифрування даних і формування віртуального тунелю між двома кінцевими точками. Однією з характерних особливостей цього інструменту є те, що він не залежить від

використання веб-браузера на клієнтському пристрої. Натомість для забезпечення повноцінної роботи OpenVPN вимагається встановлення відповідного програмного забезпечення як на серверній стороні, так і на клієнтській, що дозволяє реалізувати повністю контрольований і безпечний обмін даними [43].

2.4. Постановка задач для проєктування

Розробка програми має на меті реалізувати вищезазначені технології для забезпечення анонімності та безпеки користувачів, що використовуватиме сервер.

Перелік вимог для розробки має складатися з таких пунктів:

- Розробити проксі-сервер за схемою, що зображено на рис.2.1.:
  - Встановити мережевий екран UFW, налаштувати та увімкнути;
  - Встановити проксі клієнт 3Proxy, створити та налаштувати конфігураційний файл;
  - Інсталювати OpenVPN, скачати та встановити конфігураційний файл для коректної роботи;
  - Вимкнути зберігання log-файлів;
  - Налаштувати приховування реальної IP-адреси
  - Написати правило, яке не допустить DNS-витоку
  - Додатково: реалізація автоматичного оновлення UFW-дозволів для змінних IP-адрес, що проходять через VPN-тунель у проксі-сервері.
- Розробити проксі-сервер за схемою, що зображено на рис.2.2.:
  - Встановити мережевий екран UFW, налаштувати та увімкнути;
  - Встановити проксі клієнт Dante, створити та налаштувати конфігураційний файл;
  - За допомогою bash-скрипту встановити Flask Web-інтерфейс;
  - Інсталювати скрипт OpenSSL та згенерувати пару відкритого та приватного (закритого) ключів для підвищення безпеки з’єднання;
  - Встановити Tor та Proxychains і налаштувати їх

- Налаштувати авторизацію для користувачів серверу;
- Вимкнути зберігання log-файлів;
- Налаштувати приховування IP-адреси та місця звернення;
- Додатково: вдосконалити Flask web-інтерфейс

- Порівняти дві системи та визначити їхні переваги і недоліки.

2.5. Висновки до II розділу

У розділі були визначені головні аспекти проєктування прототипу системи анонімізації інтернет-трафіку на основі проксі-серверу. Розглянуто складові мережевої конфіденційності, зокрема, методи анонімізації, аутентифікації, логування, моніторингу та шифрування, що є важливими для забезпечення безпеки та приватності користувачів в Інтернет-просторі.

Надано дві альтернативні архітектури проксі-серверів, кожна з яких має свої переваги та орієнтована на різні сценарії використання. Перша архітектура поєднує проксі-сервер з VPN-клієнтом і фаєрволом для забезпечення контрольованого доступу та додаткового рівня шифрування. Друга архітектура робить акцент на максимальній анонімності та гнучкості управління трафіком, використовуючи багаторівневу маршрутизацію через ProxyChains та Tor.

Також, визначений вибір технічних інструментів для реалізації системи, включаючи операційну систему Ubuntu, проксі-сервери Dante та 3Proxy, фаєрвол UFW, веб-інтерфейс Flask Web UI, інструменти Proxychains, Tor, бібліотеку OpenSSL та засіб для створення захищених з'єднань OpenVPN..

Сформульовано конкретні задачі для проєктування системи, що включають розробку проксі-серверів за двома запропонованими схемами, налаштування компонентів системи, забезпечення авторизації, приховування IP-адрес та інше.

РОЗДІЛ ІІІ. ПРАКТИЧНА РЕАЛІЗАЦІЯ СИСТЕМИ АНОНІМІЗАЦІЇ ІНТЕРНЕТ-ТРАФІКУ НА ОСНОВІ ПРОКСІ-СЕРВЕРІВ

3.1. Створення та налаштування системи анонімізації через проксі, мережевий екран та VPN

Створимо bash-скрипт *install_and_run_3proxy.sh*, що зможе встановити необхідний пакет для налаштування проксі-серверу(3Proxy) та який запускатиме його автоматично при умові, що працює VPN. (Додаток А)

Після збереження скрипту, надаємо дозвіл на його запуск та виконуємо його.

```
diplom@diplom:~$ nano install_and_run_3proxy.sh
diplom@diplom:~$ chmod +x install_and_run_3proxy.sh
diplom@diplom:~$ sudo ./install_and_run_3proxy.sh
[sudo] password for diplom:
Hit:1 http://ua.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://ua.archive.ubuntu.com/ubuntu noble-updates InRelease [126 k
Hit:3 http://ua.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:5 https://repo.protonvpn.com/debian stable InRelease
```

Рис.3.1. Встановлення п акетів 3Proxy та налаштування

Надалі встановлюємо пакети OpenVPN. Потім переходимо на сайт ***VPNBOOK***, скачуємо zip-архів, розпаковуємо його та обираємо один із чотирьох файлів, що там знаходяться. До прикладу візьмемо файл *vpnbook-de220-tcp80.ovpn* з папки *vpnbook-openvpn-de220*.

```
diplom@diplom:~$ sudo apt install openvpn
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openvpn is already the newest version (2.6.12-0ubuntu0.24.04.3).
The following packages were automatically installed and are no longer required:
  libllvm17t64 python3-netifaces
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
diplom@diplom:~$
```

Рис.3.2. Встановлення OpenVPN

Розпаковуємо архів і перекидаємо необхідний файл в папку, що знаходиться за шляхом */etc/openvpn/server/*.

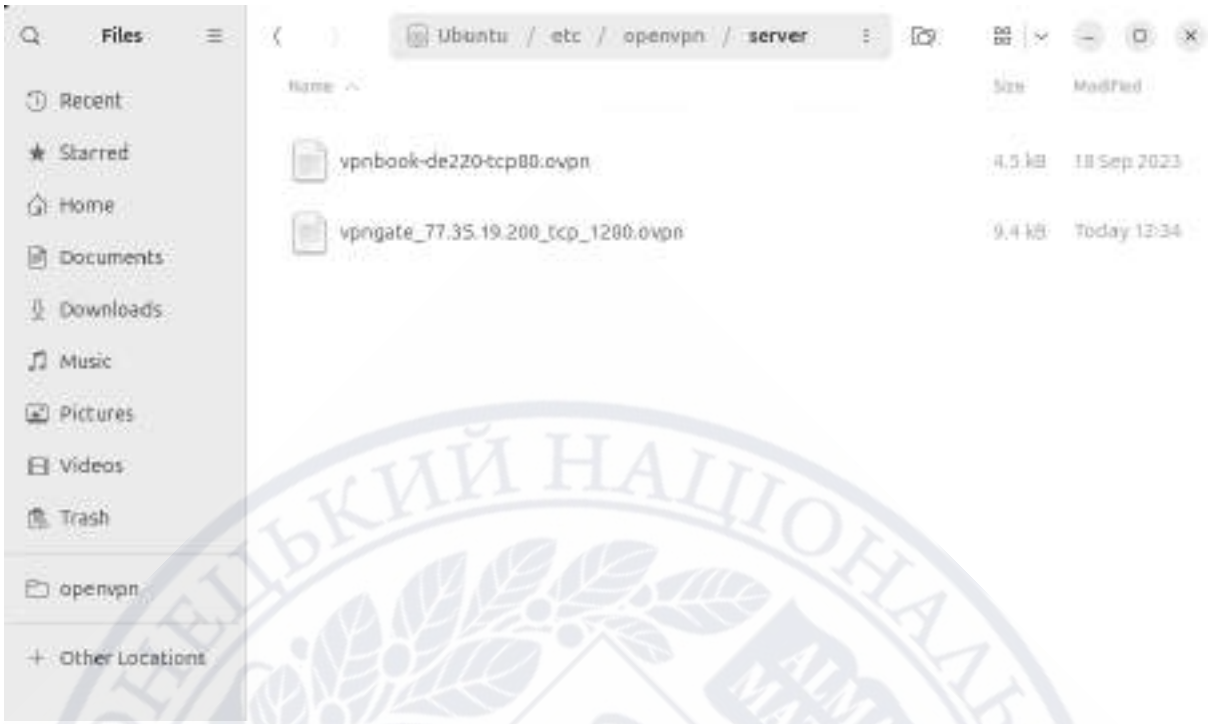


Рис.3.3. Файли для підключення VPN

Запускаємо VPN, вводимо логін та пароль для підключення до VPN і перевіряємо результати використання проксі-серверу та VPN.

Рис.3.4. Запуск та авторизація до серверу

У результаті VPN змінив справжню IP-адресу та інформацію, звідки йде звернення.

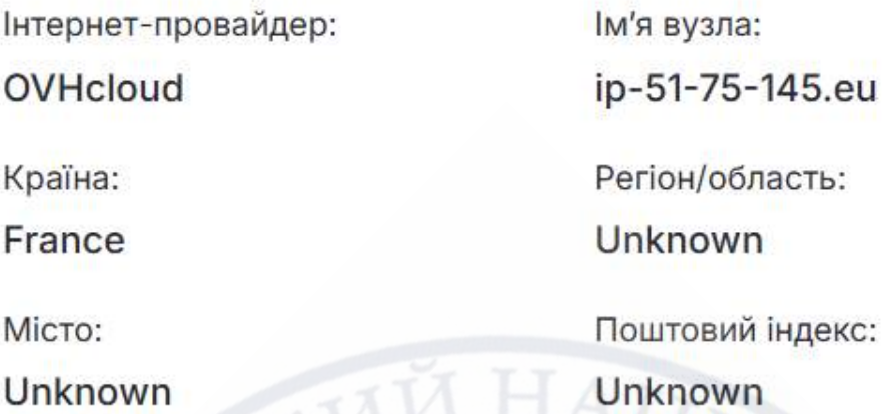


Рис.3.5. Результат використання проксі серверу та VPN

Встановимо, ще один VPN-клієнт – Proton, що буде заміною для OpenVPN. Для подальшої реалізації завдань є необхідність в більш гнучкому клієнті.

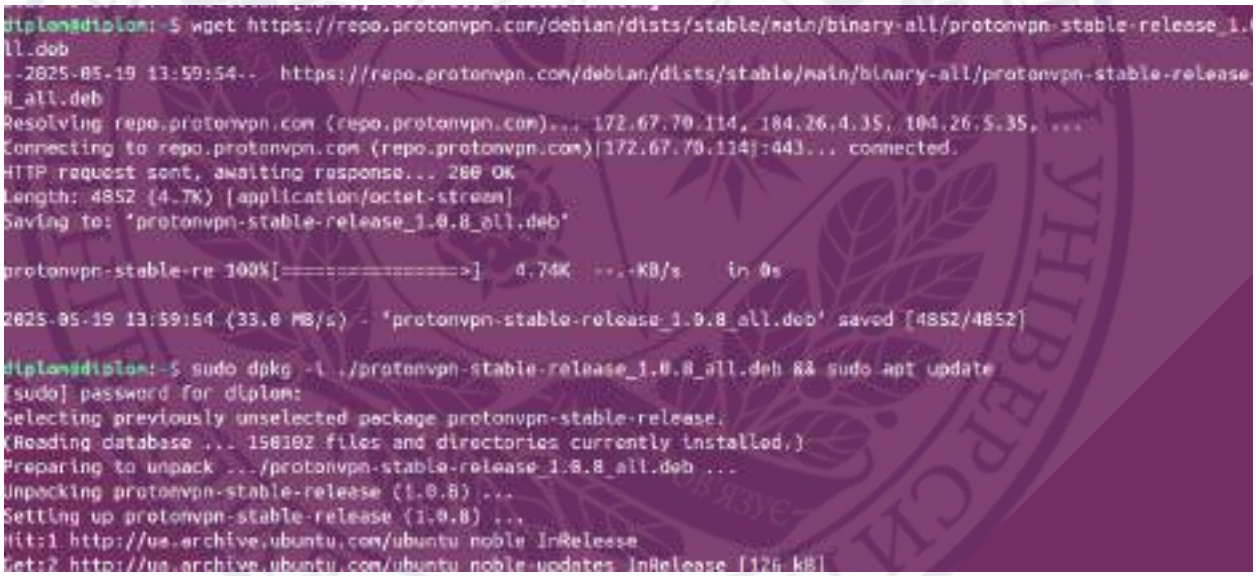


Рис.3.6. Встановлення Proton

Відключаємо попередній клієнт та під'єднуємся до серверів Proton та перевіряємо результат.

Рис.3.6. Приклад роботи з Proton

Тепер перезапускаємо bash-скрипт (Додаток А), що оновить роботу проксі-серверу та дасть змогу отримати нові результати.

```
* Connection #2 to host ifconfig.me left intact
2001:ec0:40:27::11diplom@diplom:~$ curl -v socks5-hostname diplom:diplom@127.0.0.1:1080 http://ifconfig.me
* Could not resolve host: socks5-hostname
* Closing connection
curl: (6) Could not resolve host: socks5-hostname
*   Trying 127.0.0.1:1080...
* Connected to 127.0.0.1 (127.0.0.1) port 1080
* Server auth using Basic with user 'diplom'
> GET / HTTP/1.1
> Host: 127.0.0.1:1080
> Authorization: Basic ZGlwbG9tOmRpcGxvbQ==
> User-Agent: curl/8.5.0
> Accept: */*
>
* Received HTTP/0.9 when not allowed
* Closing connection
curl: (1) Received HTTP/0.9 when not allowed
* Host ifconfig.me:80 was resolved.
* IPv6: 2600:1901:0:b2bd::
* IPv4: 34.160.111.145
*   Trying [2600:1901:0:b2bd::]:80...
*   Trying 34.160.111.145:80...
* Connected to ifconfig.me (2600:1901:0:b2bd::) port 80
```

Рис.3.7. Виконання з’єднання проксі та VPN

Спочатку доступ на з'єданання був закритий, оскільки не було ввімкнуто VPN. При повторному з’єднанні до серверу, здійснювалась авторизація (diplom:diplom), яка було виконато успішно. DNS-запити також правильно передаються (через socks5-hostname). Вихідний трафік клієнтів через 3proxy – проходить через VPN та IP не витікає, але це лише для конкретної IP-адреси.

При зміні VPN серверу, трафік перестане проходити, IP витікатиме і проксі сервер працювати не коректно.

Рис.3.8. Виконання з’єднання проксі та VPN

Оновимо правила через мережевий екран (UFW), які видалятимусь самостійно на порті 1080 при динамічних змінах IP-адрес. Надамо дозвіл на запуск цього скрипта та робим його виконуваним.

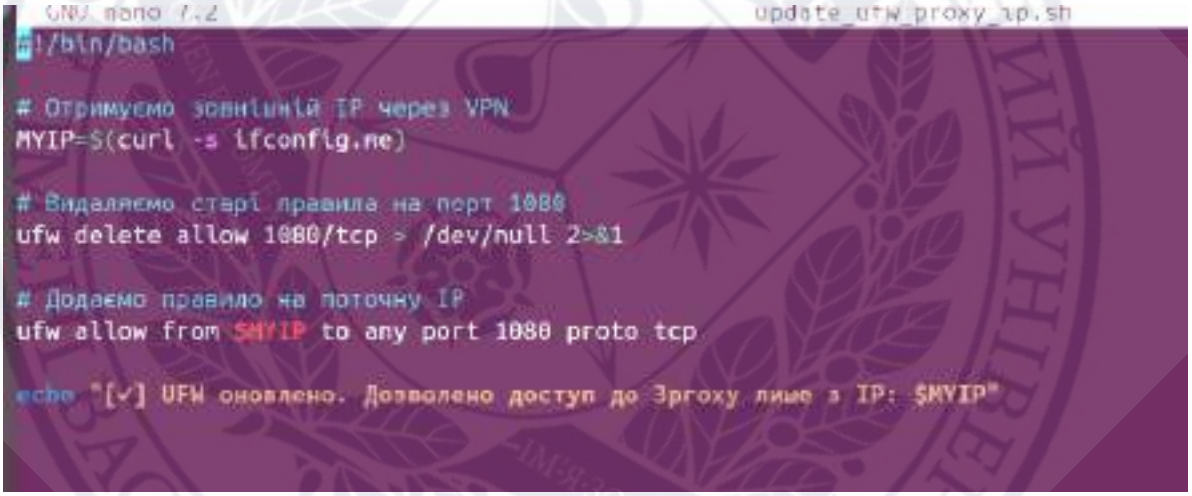


Рис.3.9. Bash-скрипт з необхідними правилами

Потім переходимо до скрипта *crontab -e*, завдяки якому ми можем налаштувати таймер, що оновлюватиме через кожні 5 хвилин UFW-правила для поточної IP-адреси.

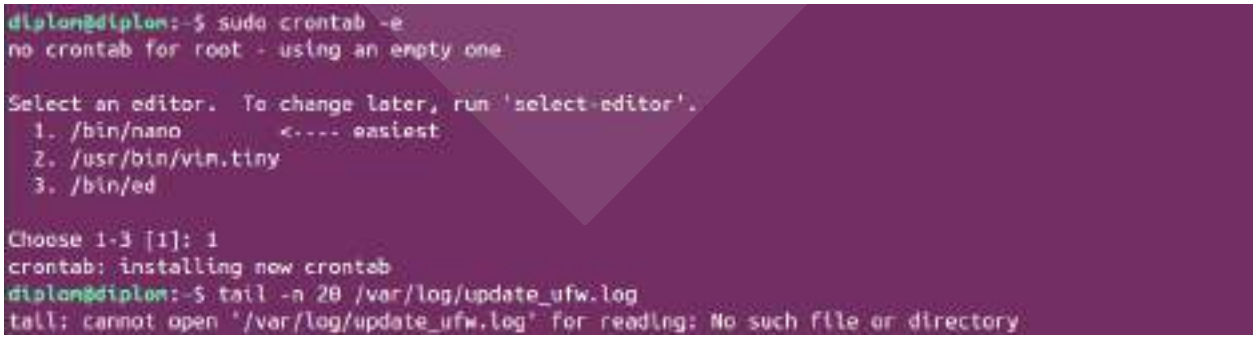


Рис.3.10. UFW-правило з таймером

Далі забороняємо всі вхідні підключення та загальний доступ до проксі.

```
diplom@diplom:~$ # Заборонити всі вхідні підключення за замовчуванням (опціонально)
sudo ufw default deny incoming


sudo ufw allow ssh
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
Rule added
Rule added (v6)
diplom@diplom:~$ # Заборонити загальний доступ до проксі
sudo ufw deny 1080/tcp
Rule added
Rule added (v6)
diplom@diplom:~$ sudo ufw status numbered
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] 22/tcp                     ALLOW IN    Anywhere
[ 2] 1080/tcp                   DENY IN     Anywhere
[ 3] 1080/tcp                   ALLOW IN    2001:ac8:40:27::11
[ 4] 22/tcp (v6)                ALLOW IN    Anywhere (v6)
[ 5] 1080/tcp (v6)              DENY IN     Anywhere (v6)
```

Рис.3.11. Додаткові UFW-правила

Перевіряємо чи є якісь збережені логи, добавляємо додаткові правила, для поточних VPN IPv4 та IPv6. які будуть відображатися, інші IP – приховані.

```
diplom@diplom:~$ tail -n 20 /var/log/update_ufw.log
/bin/sh: 1: /usr/local/bin/update_ufw_proxy_ip.sh: not found
/bin/sh: 1: /usr/local/bin/update_ufw_proxy_ip.sh: not found
/bin/sh: 1: /usr/local/bin/update_ufw_proxy_ip.sh: not found
diplom@diplom:~$ sudo ufw deny 1080/tcp
Skipping adding existing rule
Skipping adding existing rule (v6)
diplom@diplom:~$ # Дозволити поточний IPv4
sudo ufw allow from $(curl -s4 ifconfig.me) to any port 1080 proto tcp

# Дозволити поточний IPv6
sudo ufw allow from $(curl -s6 ifconfig.me) to any port 1080 proto tcp
Rule added
Rule added (v6)
```

Рис.3.12. Дозволи лише на поточні VPN IPv4 та IPv6. Перевірка наявностей log-файлів

Тепер перевіряємо чи додані правила вплинули на роботу системи. Виконуємо команду *sudo ufw status numbered.*

```
diplom@diplom:~$ sudo ufw status numbered
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] 22/tcp                     ALLOW IN    Anywhere
[ 2] 1080/tcp                   DENY IN     Anywhere
[ 3] 1080/tcp                   ALLOW IN    149.102.244.103
[ 4] 1080/tcp                   ALLOW IN    2001:ac8:40:27::11
[ 5] 22/tcp (v6)                ALLOW IN    Anywhere (v6)
[ 6] 1080/tcp (v6)              DENY IN     Anywhere (v6)
[ 7] 1080/tcp                   ALLOW IN    2a02:6ea0:ce14:3033::11
```

Рис.3.13. Перевірка активних правил фаєрвола

Оскільки помилок ніяких не виникло, значить правила дійсні. На рис.3.13. видно наступне:

1) [1] 22/tcp allow in – діє правило, що порт відкритий і використовується для SSH-з'єднання;
2) [2] 1080/tcp deny all – дія для порту, що закриває реальну IP-адресу користувача та VPN IP, яке вже не дійсне;
3) [3] 1080/tcp allow in – дія лише для поточного VPN IP(змінюється протягом кожних 5хв), що підмінює справжню IP та місце звернення;
4) [4] та [7] 1080/tcp allow in – приховані та вимкнуті VPN IP, які вже не дійсні;
5) [5] 22/tcp (v6) allow in та [6] 1080/tcp (v6) deny all – всі ті самі дії, які були налаштовані за допомогою правило що і для [1] та [2], але IPv6 протоколу.

3.2. Створення та налаштування системи анонімізації через проксі, Tor, Proxychains, SSL, UFW та Flask

За допомогою bash-скрипта, встановлюємо необхідні інструменти для подальшого налаштування та роботи з ними. Надаємо дозвіл на запуск та виконуємо. (Додаток Б.1)

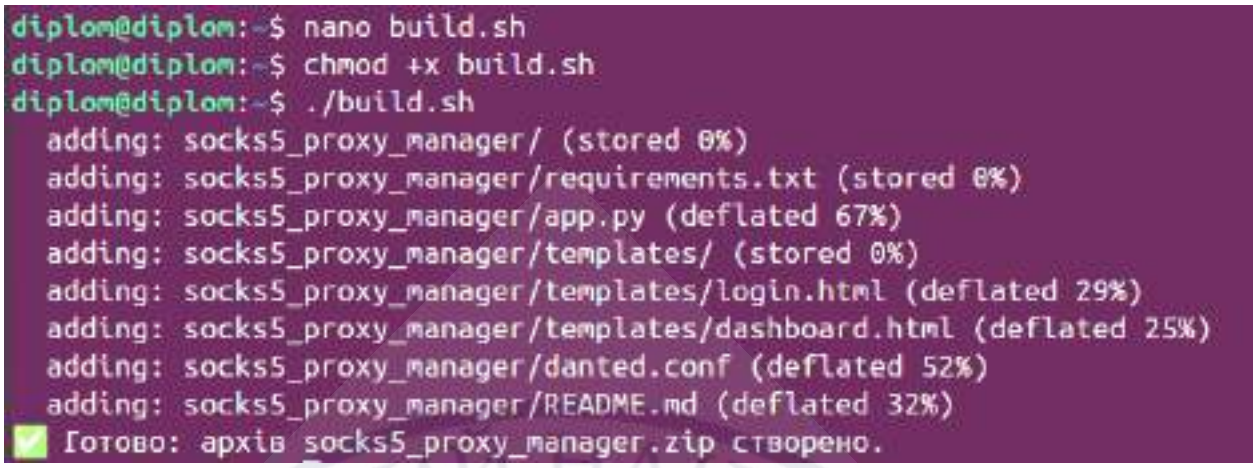


Рис.3.14. Встановлення інструментів через bash-скрипт

Перш за все після інсталяції скрипту, перевіряємо працездатність конфігураційних файлів проксі серверу.

Рис.3.15. Перевірка стану проксі серверу Dante

Тепер потрібно створити сертифікат та ключ для подальшого шифрування при з’єднанні і передачі пакетів через проксі сервер, Tor та UFW. (Додаток Б.2)

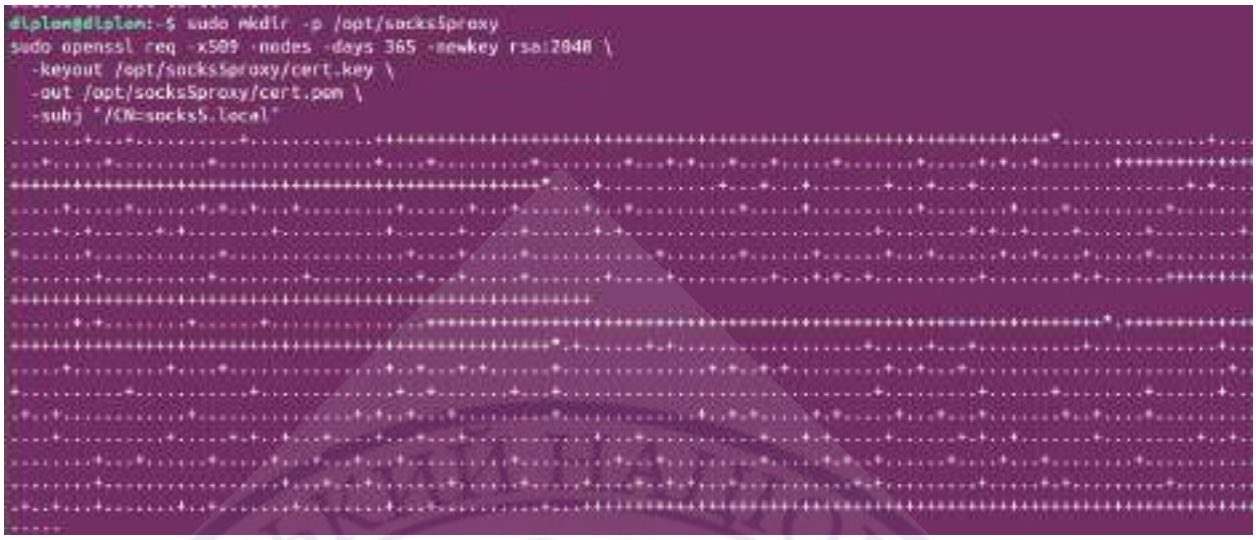


Рис.3.16. Створення SSL/TLS-сертифіката

Тепер переходимо до редагування конфігураційних файлів. Налаштуємо спочатку файл, який відповідає за підключення веб-інтерфейсу.

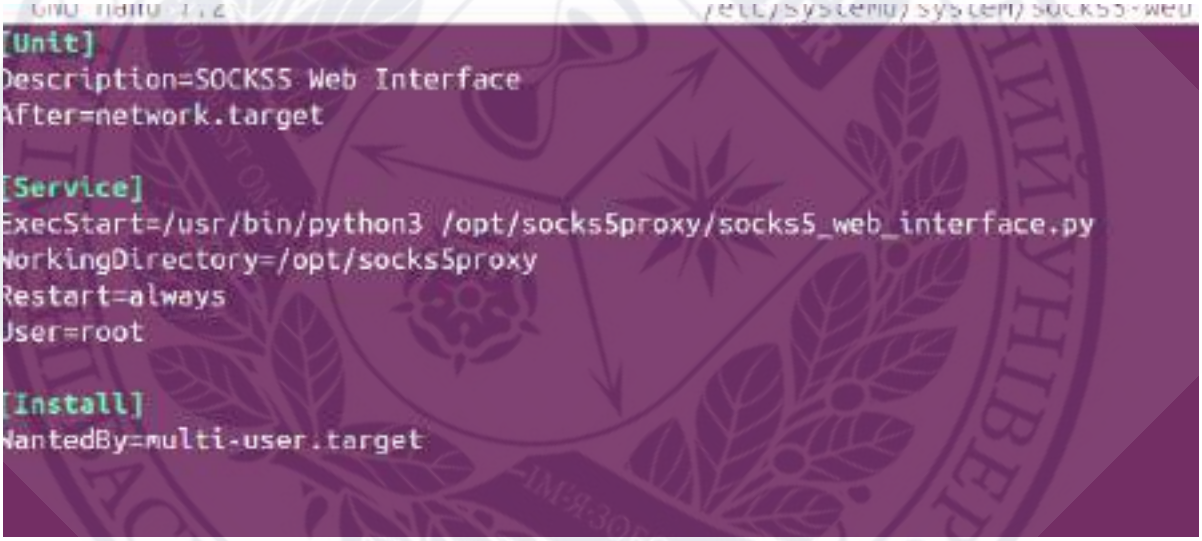


Рис.3.17. Редагування файлу socks5-web-interface.service

Далі редагуємо файл, що відповідає запуску саме socks5 та який є залежним від роботи та налаштувань конфігураційного файлу danted.

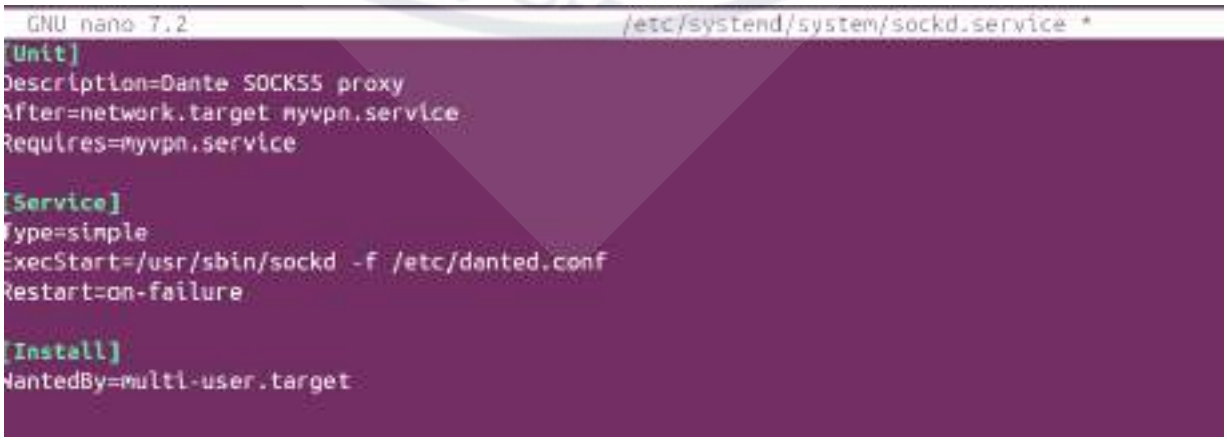


Рис.3.18. Редагування файлу sockd.service

Зробивши усі необхідні кроки, перезапускаємо демона та веб-сервіс.

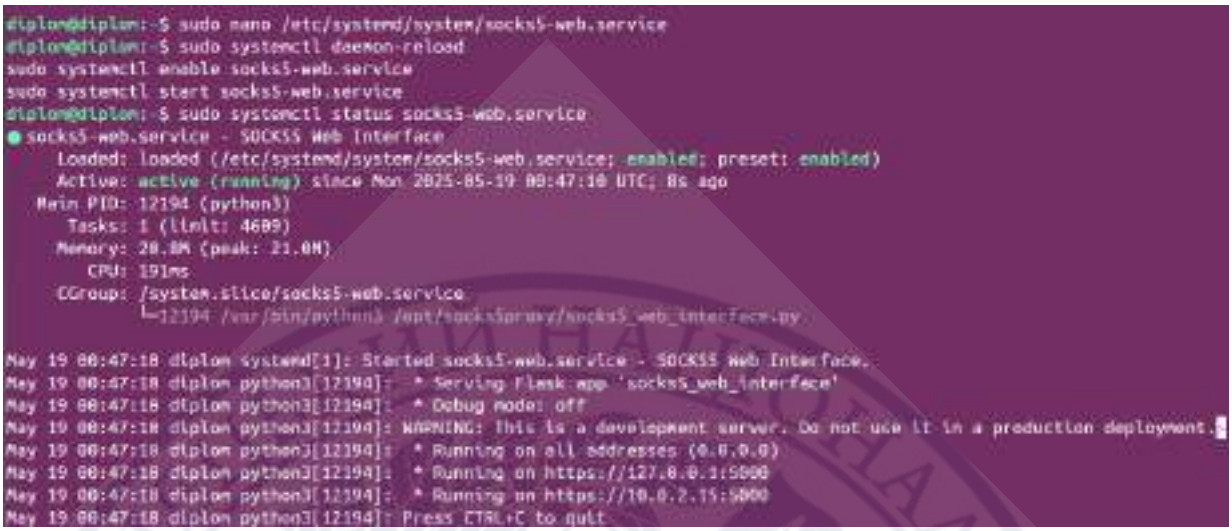


Рис.3.19. Перезавантаження системи

Перейшовши на веб-ресурс, можна побачити, що було реалізовано автентифікація за логіном та паролем на сайт для користувача.

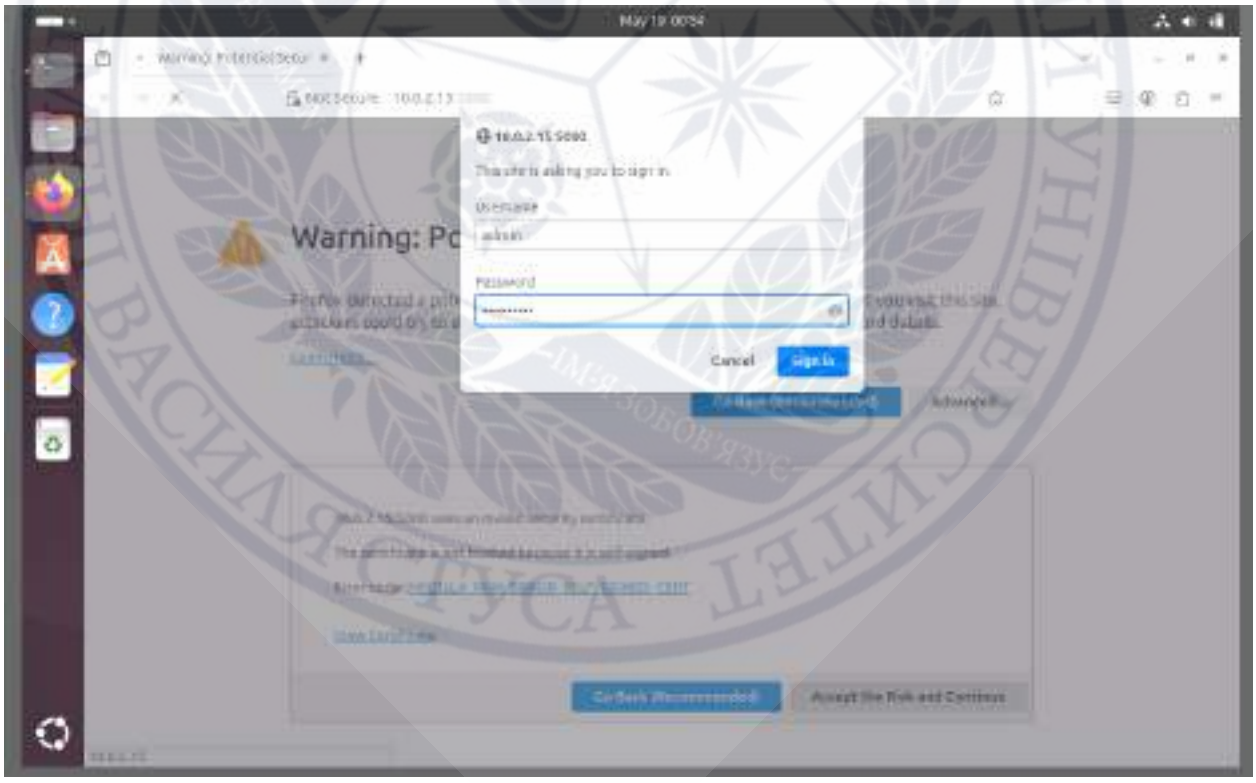


Рис.3.20. Результат використання Dante, Flask

Встановлюємо та включаємо інструменти для налаштування SSH-тунелю.

```
diplom@diplom:~$ sudo systemctl enable --now ssh
[sudo] password for diplom:
Synchronizing state of ssh.service with SysV service script with /usr/lib/system
d/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable ssh
diplom@diplom:~$ ssh -f -D -N 1080 -p 2222 diplom@127.0.0.1
Bad dynamic forwarding specification '-N'
diplom@diplom:~$ ssh -f -N -D 1080 -p 2222 diplom@127.0.0.1
ssh: connect to host 127.0.0.1 port 2222: Connection refused
diplom@diplom:~$ ssh -f -N -D 1080 -p 22 diplom@127.0.0.1
diplom@127.0.0.1's password:
Permission denied, please try again.
diplom@127.0.0.1's password:
diplom@diplom:~$ sudo systemctl restart danted
diplom@diplom:~$ sudo nano /etc/proxychains4.conf
```

Рис.3.21. Використання SSH-тунелювання

Вписуємо та оновлюємо порти, які можна використовувати з проксі сервером.

```
diplom@diplom:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
1080/tcp                   ALLOW       Anywhere
22/tcp                     ALLOW       Anywhere
9050/tcp                   ALLOW       Anywhere
1080/tcp (v6)              ALLOW       Anywhere (v6)
22/tcp (v6)                ALLOW       Anywhere (v6)
9050/tcp (v6)              ALLOW       Anywhere (v6)
```

Рис.3.22. Статус мережевого екрана

Також перевстановлюємо та налаштовуємо пакети Tor та Proxychains. Оновлюємо порти, які проксі сервер пропускатиме для подальшого використання. Приклад використання динамічних проксі ланцюгів.

```
diplom@diplom:~$ sudo systemctl restart tor
diplom@diplom:~$ sudo ufw reload
Firewall reloaded
diplom@diplom:~$ proxychains4 curl https://ifconfig.me
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
[proxychains] Dynamic chain  ...  127.0.0.1:9050  ...  127.0.0.1:1080 <--socket error or ti
meout!
[proxychains] Dynamic chain  ...  127.0.0.1:9050  ...  ifconfig.me:443  ...  OK

<html><head>
<meta http-equiv="content-type" content="text/html;charset=utf-8">
<title>403 Forbidden</title>
</head>
<body text=#000000 bgcolor=#ffffff>
<h1>Error: Forbidden</h1>
<h2>Your client does not have permission to get URL <code>/</code> from this server.</h2>
<h2></h2>
</body></html>
```

Рис.3.23. Використання динамічних проксі ланцюгів

Редагуємо конфігураційний файл для коректного функціонування інструмента Proxychains із сервісом Tor.

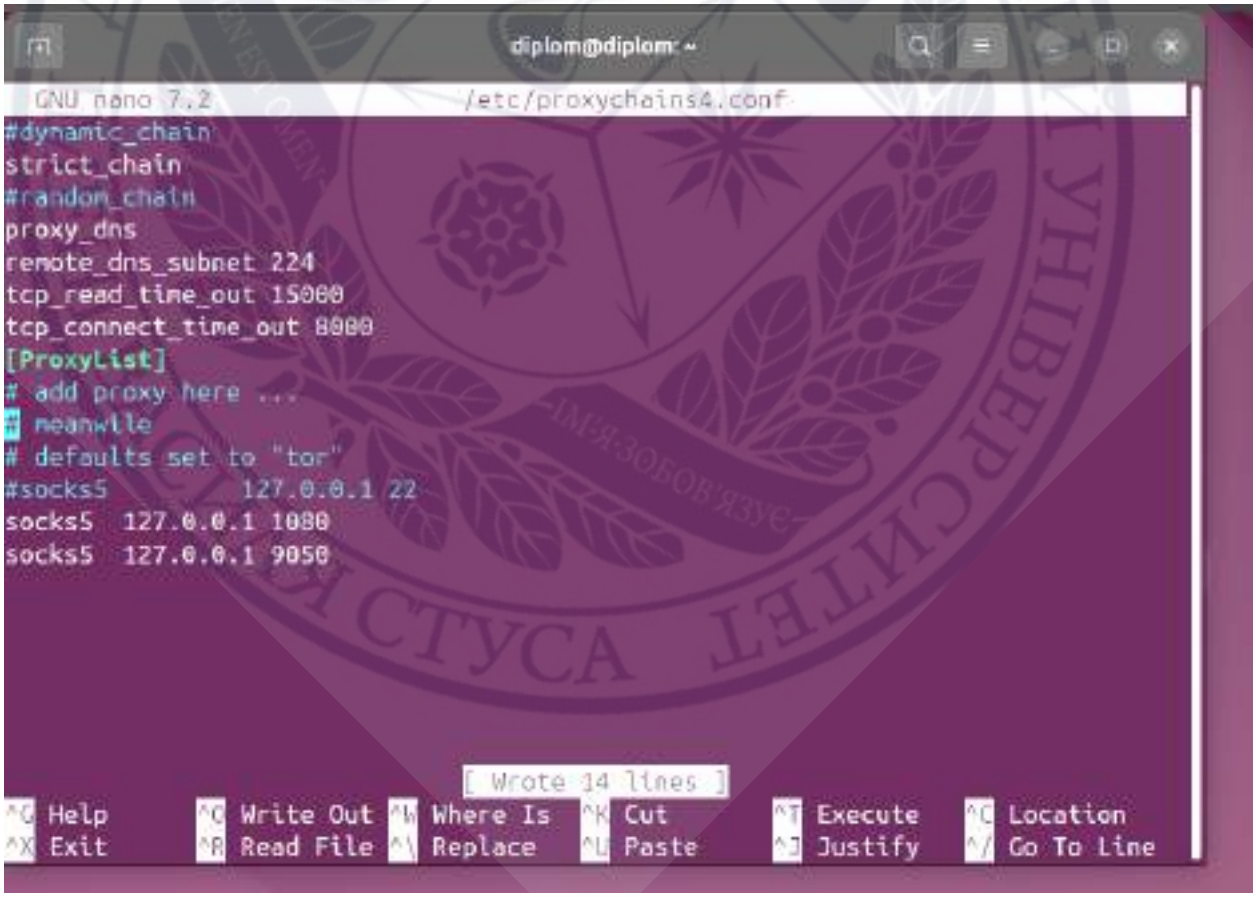


Рис.3.24. Зміна типу ланцюжку

Змінивши динамічні ланцюжки на прямі, проксі сервер почав працював належним чином. UFW також працює належним чином, оскільки не дає підключитись до порта 5555

Рис.3.25. Перевірка коректності роботи проксі серверу

Відповідь від сервера – це JSON від httpbin.org, де поле "origin" показує, яка IP-адреса дійшла до нього. В даному випадку це IP-адреса Tor-exit node.

Запит успішно пройшов через обидва проксі (Dante → Tor) і дійшов до httpbin.org. Сервер побачив не локальний IP, а IP вихідного Tor-вузла – 192.42.116.182. Тобто proxychains у режимі strict_chain з Dante і Tor налаштовані коректно.

Тепер зробимо приховування місця запиту. Для початку перевіримо чи фаєрвол працює, і чи активний Tor.

Рис.3.26. Статус роботи Tor

Перевіряємо дозволи SSH/Dante/Tor, забороняємо всі вихідні з’єднання, крім тих, що дозволено.

```
diplom@diplom:~$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To                         Action      From
--                         ------      ----
1080/tcp                   ALLOW IN    Anywhere
22/tcp                     ALLOW IN    Anywhere
9050/tcp                   ALLOW IN    Anywhere
1080/tcp (v6)              ALLOW IN    Anywhere (v6)
22/tcp (v6)                ALLOW IN    Anywhere (v6)
9050/tcp (v6)              ALLOW IN    Anywhere (v6)

diplom@diplom:~$ sudo ufw default deny outgoing
Default outgoing policy changed to 'deny'
(be sure to update your rules accordingly)
```

Рис.3.27. Перевірка статусу фаєрволу

Далі добавляємо вихід тільки на локальні проксі-порти та локальний трафік по інтерфейсу lo. Перезавантаєумо фаєрвол

```
diplom@diplom:~$ # Dante (локально)
sudo ufw allow out to 127.0.0.1 port 1080 proto tcp

# Tor (локально)
sudo ufw allow out to 127.0.0.1 port 9050 proto tcp
Rule added
Rule added
diplom@diplom:~$ sudo ufw allow out to any port 53 proto udp
sudo ufw allow out to any port 53 proto tcp
Rule added
Rule added (v6)
Rule added
Rule added (v6)
diplom@diplom:~$ sudo ufw allow out on lo
sudo ufw allow in on lo
Rule added
Rule added (v6)
Rule added
Rule added (v6)
diplom@diplom:~$ sudo ufw reload
Firewall reloaded
```

Рис.3.28. Нові правила для UFW

Із входу видно ALLOW IN лише для портів 22 (SSH), 1080 (Dante) і 9050 (Tor) –Tor слухає лише на localhost, тож і цей «ALLOW IN 9050» нікого не пропустить. Відповідно не може статися виходу в інтернет напряму, завдяки цим правилам, тільки через SOCKS-проксі, в інакшому плані – реальне місце звернення буде захищеним.

Якщо у вас щось спробує вийти напряму –UFW його заблокує. Таким чином жодна програма не зможе «витекти» в інтернет крім через SOCKS-проксі, і місце звернення (ваше реальне IP/геолокація) буде приховано.

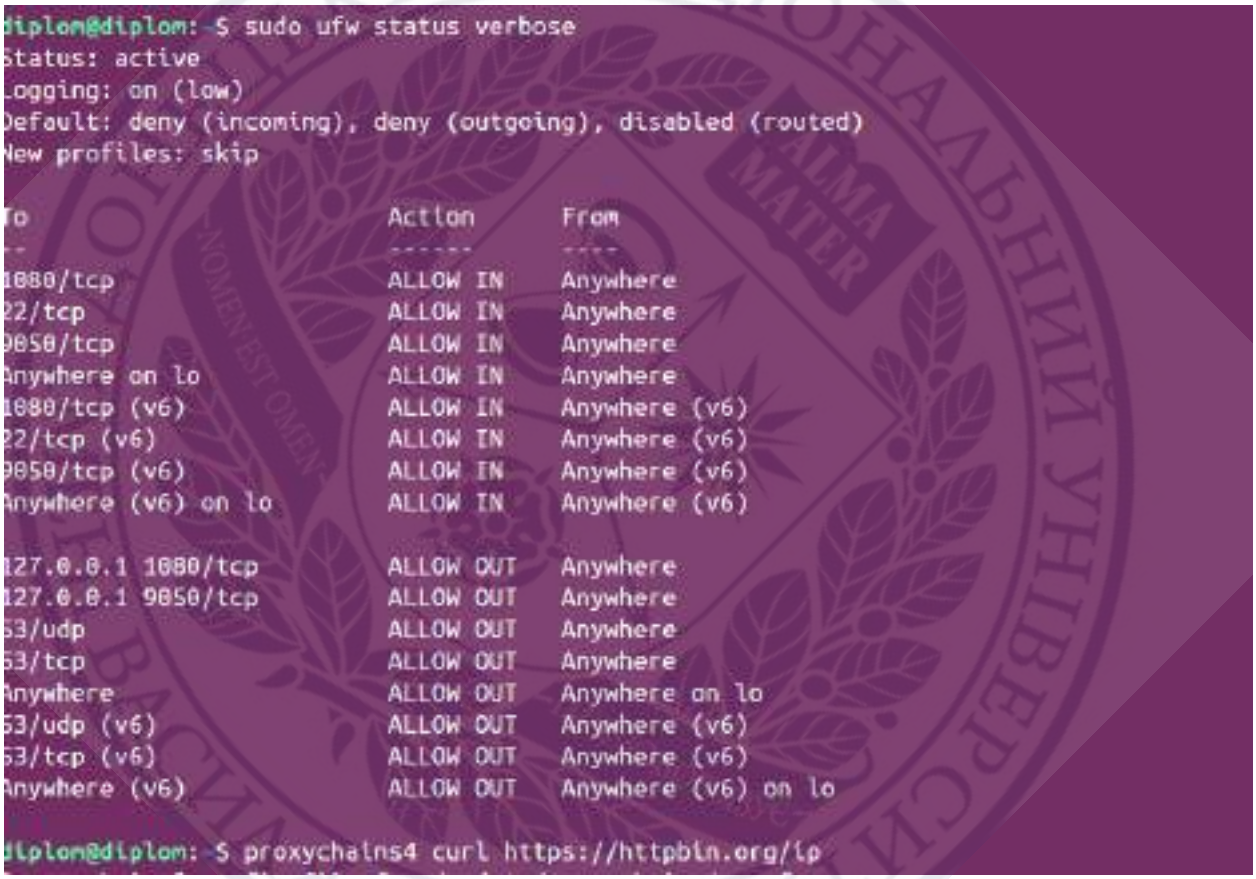


Рис.3.29. Перевірка статусу UFW після додавання нових правил

Тепер протеестуємо ланцюжок UFW→Dante→Tor та proxychains4

Рис.3.30. Тестування системи в загальному

Вот яким чином виглядає результат підміни місця запиту.

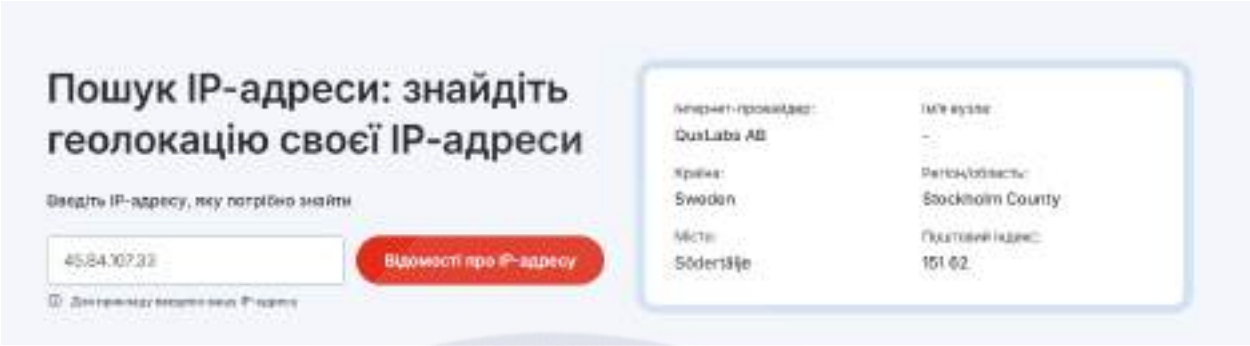


Рис.3.31. Результат підміни місця запиту

Якщо щось намагатиметься вийти напряму – UFW його заблокує. Таким чином жодна програма не зможе «витекти» в інтернет, крім як через SOCKS-проксі, і місце звернення (ваше реальне IP/геолокація) буде приховано.

3.3. Порівняння двох створених систем для анонімізації трафіку

При порівнянні двох систем для анонімізації трафіку – на основі VPN та на основі мережі Tor з ProxyChains – потрібно розуміти для яких цілей використовувати їх необхідно. Обидві технології мають на меті захистити дані користувача від зовнішнього втручання, проте реалізують це за різними принципами і з неоднаковим рівнем складності в налаштуванні та використанні.

Система, що базується на VPN, функціонує шляхом створення захищеного тунелю між клієнтським пристроєм і віддаленим сервером. У цьому випадку весь трафік, що виходить із мережі користувача, шифрується й передається через обраний вузол, що надає можливість приховати справжню IP-адресу та зменшити ризики перехоплення інформації. Це ефективно у контексті централізованого захисту з'єднання, однак він потребує довіри до провайдера VPN-послуг, оскільки, теоретично, має доступ до переданих даних у незашифрованому вигляді на стороні виходу з тунелю.

Інша система побудована за принципом багаторівневої маршрутизації через децентралізовану мережу Tor, де трафік проходить через випадкову послідовність вузлів, кожен з яких бачить лише попередній та наступний етап маршруту. Разом з ProxyChains така структура дозволяє формувати складні

ланцюги проксі, що значно ускладнює спроби відстеження джерела трафіку. Цей варіант анонімізації забезпечує вищий рівень приватності, проте має свої обмеження, зокрема повільнішу швидкість з'єднання та певну складність у налаштуванні, особливо у випадках, коли потрібна стабільна робота з додатковими сервісами, такими як веб-інтерфейси чи мережеві протоколи, які не підтримують проксі-ланцюги.

У підсумку можна зазначити, що вибір між цими двома системами має базуватися не лише на технічних характеристиках, а й на конкретних завданнях користувача. Якщо пріоритетом є швидкість і простота – доцільним буде використання VPN. Якщо ж головним є високий рівень анонімності, заради якого можна пожертвувати зручністю та продуктивністю – краще обрати систему з мережею Tor у зв'язці з ProxyChains.

Таблиця 3.1. Порівняння двох систем для анонімізації трафіку

Компонент	**Система I (VPN)**	**Система II (Tor та ProxyChains)**
Анонімність	Середня (залежить від VPN)	Висока (Tor маршрутизація)
Продуктивність	Вища	Нижча (Tor – повільніший)
Гнучкість	Менша	Вища (тонке налаштування ProxyChains)
Інтерфейс	CLI/конфіги	Flask Web Interface

3.4. Висновки до III розділу

Було розроблено дві системи анонімізації через проксі-сервер. Одна базується на використанні та налаштуванні VPN та UFW. Інша система працює завдяки мережі Tor з ProxyChains.

У цьому розділі приведено етапи побудови цих системи і результати, які отримували в кожному із випадків.

Проведено порівняння на основі отриманого практичного досвіду.

ВИСНОВКИ

У межах дипломної роботи було створено два прототипи систем анонімізації трафіку: один із використанням проксі-сервера, VPN і фаєрвола UFW, другий – на базі Tor і ProxyChains. Самостійно виконано налаштування всіх компонентів, а також реалізовано захист від DNS-витоків і перевірено працездатність систем.

Наукова й практична новизна роботи полягає в розробці конфігурації, яка має на меті забезпечити необхідну анонімність залежно від мети користувача та може бути розгорнута на звичайному сервері без потреби у складних технічних знаннях.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. SYSEL, Martin; DOLEŽAL, Ondřej. An educational http proxy server. *Procedia Engineering*, 2014, 69: 128-132. Режим доступу: https://www.researchgate.net/publication/261104026_An_Educational_HTTP_Proxy_Server (Дата звернення 18.04.2025).

2. Abiona, O., Oluwaranti, A., Oluwatope, A., Bello, S., Onime, C., Sanni, M., & Kehinde, L. (2014). Proxy server experiment and network security with changing nature of the web. International Journal of Communications, Network and System Sciences, 7(12), 519. Режим доступу: https://www.scirp.org/journal/paperinformation?paperid=52071 (Дата звернення 18.04.2025).

3. Kothapalli, Sai Charitha. *Measurement, Analysis, and System Implementation of Internet Proxy Servers*. MS thesis. University of Minnesota, 2023. (Дата звернення 18.04.2025).

4. REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL. Режим доступу: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679 (Дата звернення 18.04.2025).

5. Закон України «Про захист персональних даних» від 01.06.2010 № 2297-VI. Режим доступу: https://zakon.rada.gov.ua/laws/show/2297-17?dark=1#Text (Дата звернення 18.04.2025).

6. Закон України «Про захист інформації в інформаційно-телекомунікаційних системах» від 05.07.1994 № 80/94-ВР. Режим доступу: https://zakon.rada.gov.ua/laws/show/80/94-%D0%B2%D1%80#Text (Дата звернення 18.04.2025).

7. Закон України «Про основні засади забезпечення кібербезпеки України» від 05.10.2017 № 2163-VIII. Режим доступу: https://zakon.rada.gov.ua/laws/show/2163-19#Text (Дата звернення 18.04.2025).

8. Ambhore, Premchand B., and K. A. Wankhade. "Proxy server FOR intranet security." IOSR Journal of Computer Engineering 20.2 (2018): 1-14. Режим

доступу: https://www.academia.edu/download/56747830/A2002010114.pdf (Дата звернення 20.04.2025).

9. Wenceslao, J. L. B., and R. B. Wenceslao. "Network Performance of Proxy-Enabled Server Using Three Configurations." Indian Journal of Science and Technology 15.18 (2022): 850-856. Режим доступу: https://sciresol.s3.us-east-2.amazonaws.com/IJST/Articles/2022/Issue-18/IJST-2021-1551.pdf (Дата звернення 21.04.2025)

10. RAUNAK, Mohammad S., et al. Implications of proxy caching for provisioning networks and servers. ACM SIGMETRICS Performance Evaluation Review, 2000, 28.1: 66-77. Режим доступу: https://dl.acm.org/doi/pdf/10.1145/345063.339357 (Дата звернення 21.04.2025)

11. CHOI, Jinchun, et al. Understanding the proxy ecosystem: A comparative analysis of residential and open proxies on the internet. IEEE Access, 2020, 8: 111368-111380. Режим доступу: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9115039 (Дата звернення 21.04.2025)

12. PANNU, Mandeep, et al. Exploring proxy detection methodology. In: 2016 IEEE International Conference on Cybercrime and Computer Forensic (ICCCF). IEEE, 2016. p. 1-6. Режим доступу: https://ieeexplore.ieee.org/abstract/document/7740438 (Дата звернення 21.04.2025)

13. Павличек А., Судзіна Ф. Інтернетова безпека та приватність у VPN. Журнал технологій мережі, 2018. №9 Режим доступу: https://vbn.aau.dk/en/publications/internet-security-and-privacy-in-vpn (Дата звернення: 24.04.2025р.)

14. Li, Bingdong, et al. "An analysis of anonymizer technology usage." Traffic Monitoring and Analysis: Third International Workshop, TMA 2011, Vienna, Austria, April 27, 2011. Proceedings 3. Springer Berlin Heidelberg, 2011. Режим доступу: https://www.researchgate.net/profile/Paolo-Bolletta/publication/221151246_Limits_in_the_Bandwidth_Exploitation_in_Passi

ve_Optical_Networks_Due_to_the_Operating_Systems_Poster/links/00b495386d6aa48764000000/Limits-in-the-Bandwidth-Exploitation-in-Passive-Optical-Networks-Due-to-the-Operating-Systems-Poster.pdf#page=118 (Дата звернення: 24.04.2025р.)

15. Lakhno, Valerii, et al. "WAF ЗАХИСТУ ВНУТРІШНІХ СЕРВІСІВ У СТРУКТУРІ ZERO TRUST." Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка» 1.13 (2021): 81-91. Режим доступу: https://www.csecurity.kubg.edu.ua/index.php/journal/article/download/283/243 (Дата звернення: 24.04.2025р.)

16. Shiwani, Savita, et al. "Performance measurements: Proxy server for various operating systems." 2014 International Conference on Contemporary Computing and Informatics (IC3I). IEEE, 2014. Режим доступу: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=17fe6e7fd902b889cbbaaad21decc06854f47769 (Дата звернення: 24.04.2025р.)

17. Nappa, Antonio, et al. "RevProbe: detecting silent reverse proxies in malicious server infrastructures." Proceedings of the 32nd Annual Conference on Computer Security Applications. 2016. Режим доступу: https://software.imdea.org/~juanca/papers/revprobe_acsac16.pdf (Дата звернення: 24.04.2025р.)

18. Mi, Xianghang, et al. "Your phone is my proxy: Detecting and understanding mobile proxy networks." Proceeding of ISOC Network and Distributed System Security Symposium (NDSS), 2021. 2021. Режим доступу: https://www.ndss-symposium.org/wp-content/uploads/ndss2021_3B-2_24008_paper.pdf (Дата звернення: 24.04.2025р.)

19. Chhabra, Yogita. "A Study of Recent Research Trends of Proxy Server." International Journal of Advanced Technology in Engineering and Science 3.01 (2015): 159-164. Режим доступу: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=17fe6e7fd902b889cbbaaad21decc06854f47769 (Дата звернення: 24.04.2025р.)

20. Sawwashere, Dr Supriya, et al. "Integration of an Autonomous system for AI Infusion, Mailing with Raspberry Pi and Python Using Proxy Server Configuration." (March 11, 2024) (2024) Режим доступу:. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4754945 (Дата звернення: 24.04.2025р.)

21. Dube, Shruti. "Peer-to-Peer File Sharing Across Private Networks Using Proxy Servers." Department of computer science and engineering, Indian Institute of Technology, Kanpur (2008). Режим доступу: https://cse.iitk.ac.in/users/dheeraj/mtech/shruti-dube.pdf (Дата звернення: 24.04.2025р.)

22. Wu, Tina, Frank Breitinger, and Stephen Niemann. "IoT network traffic analysis: Opportunities and challenges for forensic investigators?." Forensic Science International: Digital Investigation 38 (2021): 301123. Режим доступу: https://dfrws.org/wp-content/uploads/2021/01/2021_APAC_paper-iot_network_traffic_analysis_opportunities_and_challenges_for_forensic_investigators.pdf (Дата звернення: 24.04.2025р.)

23. Wang, Y., Yang, K., & Zhang, Y. (2007). Research and realization of security proxy based on SSL protocol. In 2007 8th International Conference on Electronic Measurement and Instruments, ICEMI (pp. 2264-2267). Article 4350668 (2007 8th International Conference on Electronic Measurement and Instruments, ICEMI). Режим доступу: https://doi.org/10.1109/ICEMI.2007.4350668 (Дата звернення: 24.04.2025р.)

24. Junichi FUNASAKA, Masato BITO, Kenji ISHIDA, Kitsutaro AMANO, "An FTP Proxy System to Assure Providing the Latest Version of Replicated Files" in IEICE TRANSACTIONS on Communications, vol. E86-B, no. 10, pp. 2948-2956, October 2003, doi . Режим доступу: https://globals.ieice.org/en_transactions/communications/10.1587/e86-b_10_2948/_p (Дата звернення: 24.04.2025р.)

25. Configuring DHCP Proxy. Cisco systems. Режим доступу: https://www.cisco.com/c/en/us/td/docs/wireless/controller/7-

4/configuration/guides/consolidated/b_cg74_CONSOLIDATED/m_configuring_dhcp_proxy.pdf (Дата звернення: 24.04.2025р.)

26. Maltz, David A., and Pravin Bhagwat. "TCP Splice for application layer proxy performance." Journal of High Speed Networks 8.3 (1999): 225-240. Режим доступу: https://elibrary.kubg.edu.ua/id/eprint/50568/1/V_Osadchyi_PITS_2024_FEU.pdf (Дата звернення: 24.04.2025р.)

27. Montazerolghaem, Ahmadreza, et al. "A load scheduler for SIP proxy servers: design, implementation and evaluation of a history weighted window approach." (2015). Режим доступу: https://www.researchgate.net/profile/Ahmadreza-Montazerolghaem/publication/276527759_A_load_scheduler_for_SIP_proxy_servers_Design_implementation_and_evaluation_of_a_history_weighted_window_approach/links/5f8819ab92851c14bcc8efd9/A-load-scheduler-for-SIP-proxy-servers-Design-implementation-and-evaluation-of-a-history-weighted-window-approach.pdf (Дата звернення: 24.04.2025р.)

28. MOGHADDAM, Hooman Mohajeri; MOSENIA, Arsalan. Anonymizing masses: Practical light-weight anonymity at the network level. arXiv preprint arXiv:1911.09642, 2019. Режим доступу: https://arxiv.org/pdf/1911.09642 (Дата звернення: 24.04.2025р.)

29. BACKES, Michael, et al. Introducing accountability to anonymity networks. arXiv preprint arXiv:1311.3151, 2013. Режим доступу: https://arxiv.org/pdf/1311.3151 (Дата звернення: 24.04.2025р.)

30. Appl. Math. Inf. Sci. 9, No. 2L, 483-492 (2015) Applied Mathematics & Information Sciences An International Journal Proxy Server Authentication for Blocking HTTP-Cache-Poisoning Attacks Режим доступу: https://www.naturalspublishing.com/files/published/m44qh96phk6433.pdf (Дата звернення: 24.04.2025р.)

31. FEI, Bennie, et al. Analysis of web proxy logs. In: Advances in Digital Forensics II: IFIP international Conference on Digital Forensics, National Center for

Forensic Science, Orlando, Florida, January 29–February 1, 2006. Springer US, 2006. p. 247-258. Режим доступу: https://opendl.ifip-tc6.org/db/conf/ifip11-9/df2006/FeiEOV06.pdf (Дата звернення: 24.04.2025р.)

32. SINGH, Avanish Kumar; PATHAK, Amit Kumar; RAO, Ashutosh Kumar. Encryption Techniques as Security Tools: A Technical Review. International Journal of Advanced Research in Computer Science, 2012, 3.6. Режим доступу: https://www.ijarcs.info/index.php/Ijarcs/article/view/1456/1444 (Дата звернення: 24.04.2025р.)

33. Miller, S., Curran, K., & Lunney, T. (2015). Securing the internet through the detection of anonymous proxy usage. In Unknown Host Publication World Congress on Internet Security. Режим доступу: https://pure.ulster.ac.uk/ws/portalfiles/portal/11555353/WorldCIS2015.pdf (Дата звернення: 24.04.2025р.)

34. DINGLEDINE, Roger, et al. Tor: The second-generation onion router. In: USENIX security symposium. 2004. p. 303-320. Режим доступу: https://www.usenix.org/legacy/publications/library/proceedings/sec04/tech/full_papers/dingledine/dingledine.pdf (Дата звернення: 24.04.2025р.)

35. Policy-Based Security Configuration Management. Application to Intrusion Detection and Prevention. / Issam Aib, Khalid Alsubhi, Jerome Francois, and Raouf Boutaba. // David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, ON, Canada. Technical Report CS-2008-24 Режим доступу: https://cs.uwaterloo.ca/research/tr/2008/CS-2008-24.pdf (Дата звернення: 24.04.2025р.)

36. Flask’s documentation. Режим доступу: https://flask.palletsprojects.com/en/stable/ (Дата звернення: 24.04.2025р.)

37. NONIK, Oleksandr, et al. Approaches to Solving Proxy Performance Problems for HTTP and SOCKS5 Protocols for the Case of Multi-Port Passwordless Access. CPITS 2024-Cybersecurity Providing in Information and Telecommunication Systems, 2024, 3654: 189-200. Режим доступу:

https://elibrary.kubg.edu.ua/id/eprint/50568/1/V_Osadchyi_PITS_2024_FEU.pdf (Дата звернення: 24.04.2025р.)

38. Ubuntu OS. Режим доступу: [Електронний ресурс] https://ubuntu.com/ (Дата звернення: 24.04.2025р.)

39. UFW - Uncomplicated Firewall. Режим доступу: [Електронний ресурс] https://help.ubuntu.com/community/UFW (Дата звернення: 24.04.2025р.)

40. Dante – A free SOCKS server. Режим доступу: [Електронний ресурс] https://www.inet.no/dante/ (Дата звернення: 24.04.2025р.)

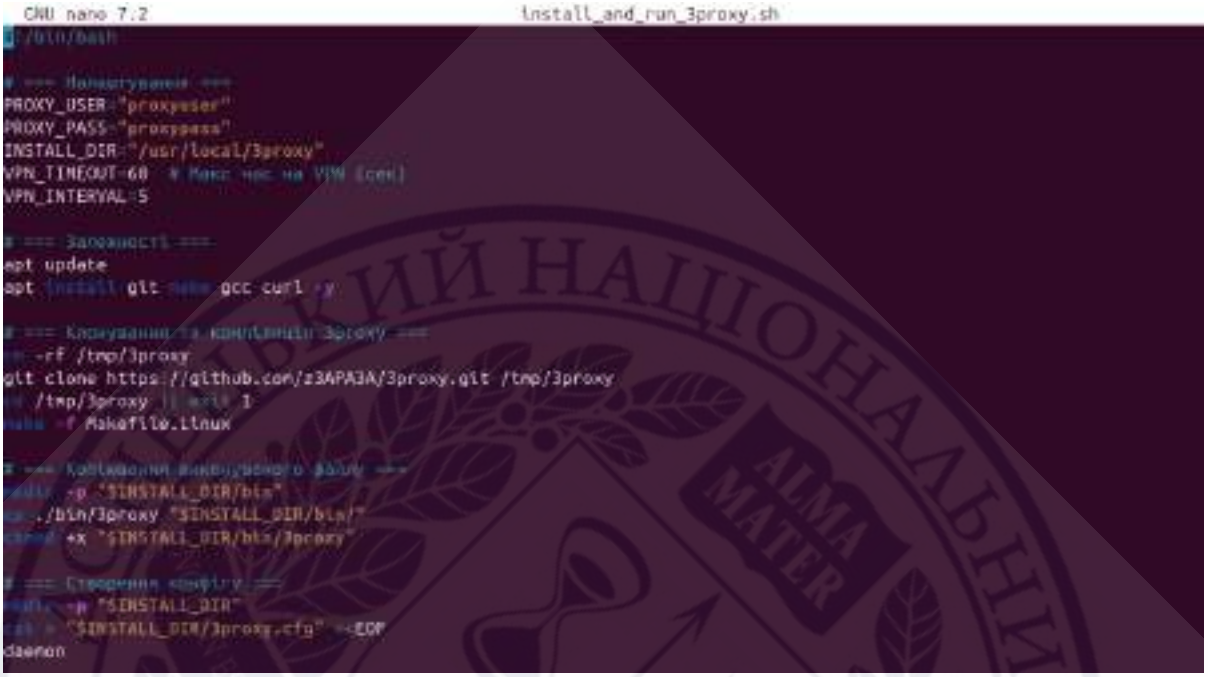
41. 3proxy tiny free proxy server Режим доступу: [Електронний ресурс] https://3proxy.ru/?l=EN (Дата звернення: 24.04.2025р.)

42. OpenSSL Documentation Режим доступу: https://docs.openssl.org/master/man7/ossl-guide-libcrypto-introduction/ (Дата звернення: 24.04.2025р.)

43. SKENDZIC, A.; KOVACIC, B. Open source system OpenVPN in a function of Virtual Private Network. In: IOP Conference Series: Materials Science and Engineering. IOP Publishing, 2017. p. 012065. Режим доступу: https://iopscience.iop.org/article/10.1088/1757-899X/200/1/012065/pdf (Дата звернення: 24.04.2025р.)

ДОДАТОК А

Bash-скрипт:



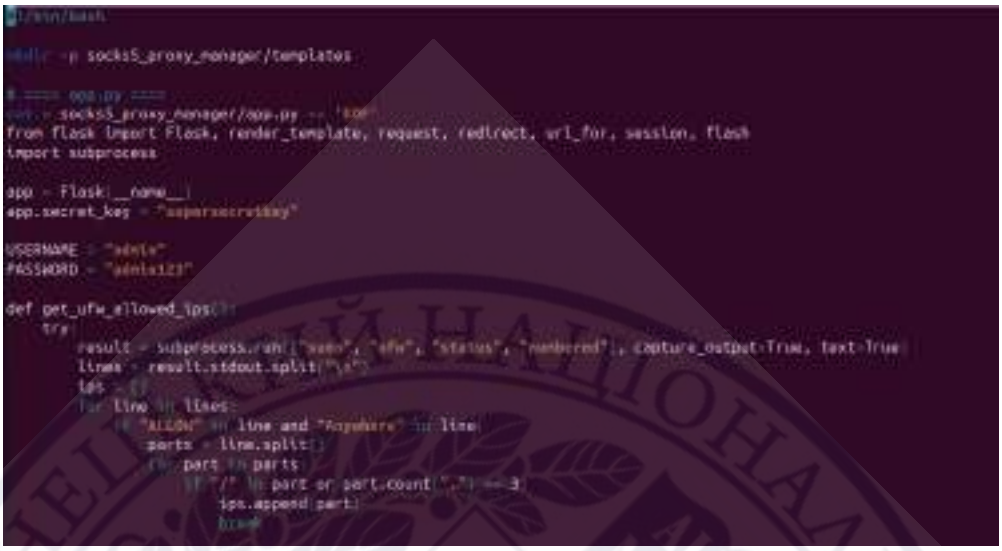
```
  GNU nano 7.2                                  install_and_run_3proxy.sh
#!/bin/bash

# === Налаштування ===
PROXY_USER="proxyuser"
PROXY_PASS="proxypass"
INSTALL_DIR="/usr/local/3proxy"
VPN_TIMEOUT=60  # Макс час на VPN (сек)
VPN_INTERVAL=5

# === Залежності ===
apt update
apt install git make gcc curl -y

# === Клонування та компіляція 3proxy ===
rm -rf /tmp/3proxy
git clone https://github.com/z3APA3A/3proxy.git /tmp/3proxy
cd /tmp/3proxy || exit 1
make -f Makefile.Linux

# === Копіювання виконуваного файлу ===
mkdir -p "$INSTALL_DIR/bin"
cp ./bin/3proxy "$INSTALL_DIR/bin/"
chmod +x "$INSTALL_DIR/bin/3proxy"

# === Створення конфігу ===
mkdir -p "$INSTALL_DIR"
cat > "$INSTALL_DIR/3proxy.cfg" <<EOF
daemon
```

```
maxconn 100
nscache 65536
timeouts 1 5 30 60 180 1800 15 60
users $PROXY_USER:CL:$PROXY_PASS
auth strong
allow $PROXY_USER
socks -p1080
log
logformat "- - - - - - - -"
nolog
EOF

# === Systemd юніт ===
cat > /etc/systemd/system/3proxy.service <<EOF
[Unit]
Description=3proxy Proxy Server
After=network.target

[Service]
ExecStart=$INSTALL_DIR/bin/3proxy $INSTALL_DIR/3proxy.cfg
Restart=always
User=root

[Install]
WantedBy=multi-user.target
EOF

# === Очікування VPN ===
```

```
# === Очікування VPN ===
echo "[INFO] Чекаємо VPN з'єднання..."
REAL_IP=$(curl -s ifconfig.me)
for ((i = 0; i < VPN_TIMEOUT / VPN_INTERVAL; i++)); do
    CUR_IP=$(curl -s ifconfig.me)
    if [[ "$CUR_IP" != "$REAL_IP" ]]; then
        echo "[OK] VPN активний. IP: $CUR_IP"
        systemctl daemon-reload
        systemctl enable 3proxy
        systemctl start 3proxy
        echo "[✅] 3proxy запущено на порті 1080 з IP $CUR_IP"
        exit 0
    fi
    echo "[...] VPN не активний (поточний IP: $CUR_IP). Очікування..."
    sleep $VPN_INTERVAL
done

echo "[❌] VPN не підключено. 3proxy НЕ буде запущено."
exit 1
```

ДОДАТОК Б

1. Bash-скрипт

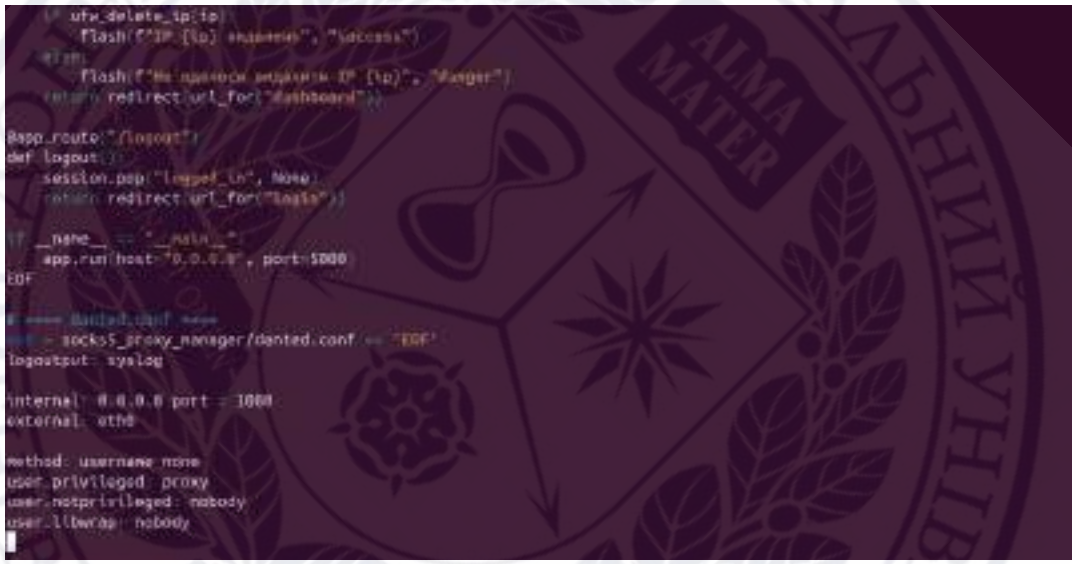


```
#!/bin/bash

mkdir -p socks5_proxy_manager/templates

# ==== app.py ====
cat > socks5_proxy_manager/app.py << 'EOF'
from flask import Flask, render_template, request, redirect, url_for, session, flash
import subprocess

app = Flask(__name__)
app.secret_key = "supersecretkey"

USERNAME = "admin"
PASSWORD = "admin123"

def get_ufw_allowed_ips():
    try:
        result = subprocess.run(["sudo", "ufw", "status", "numbered"], capture_output=True, text=True)
        lines = result.stdout.split("\n")
        ips = []
        for line in lines:
            if "ALLOW" in line and "Anywhere" in line:
                parts = line.split()
                for part in parts:
                    if "/" in part or part.count(".") == 3:
                        ips.append(part)
                        break
```

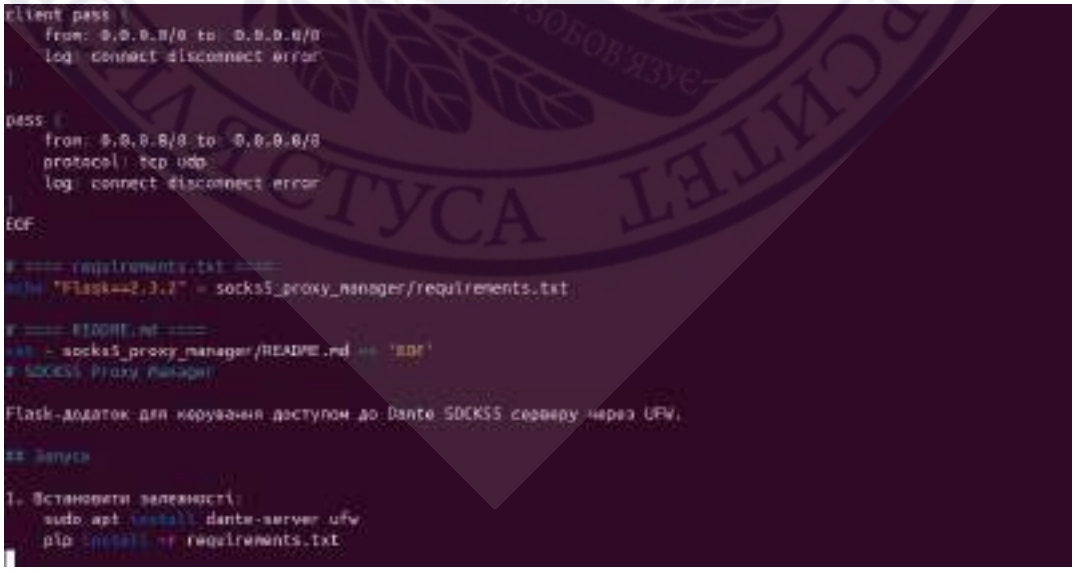
```
        return ips
    except Exception:
        return []

def ufw_add_ip(ip):
    try:
        subprocess.run(["sudo", "ufw", "allow", "from", ip, "to", "any", "port", "1080"], check=True)
        subprocess.run(["sudo", "ufw", "reload"], check=True)
        return True
    except Exception:
        return False

def ufw_delete_ip(ip):
    try:
        result = subprocess.run(["sudo", "ufw", "status", "numbered"], capture_output=True, text=True)
        lines = result.stdout.split("\n")
        number = None
        for line in lines:
            if ip in line and "ALLOW" in line and "1080" in line:
                idx1 = line.find("[")
                idx2 = line.find("]")
                if idx1 != -1 and idx2 != -1:
                    number = line[idx1+1:idx2]
                    break
        if number:
            subprocess.run(["sudo", "ufw", "delete", number], check=True, input="y\n", text=True)
            subprocess.run(["sudo", "ufw", "reload"], check=True)
```

```
        return True
    except Exception:
        return False

def ufw_delete_ip(ip):
    try:
        result = subprocess.run(["sudo", "ufw", "status", "numbered"], capture_output=True, text=True)
        lines = result.stdout.split("\n")
        number = None
        for line in lines:
            if ip in line and "ALLOW" in line and "1080" in line:
                idx1 = line.find("[")
                idx2 = line.find("]")
                if idx1 != -1 and idx2 != -1:
                    number = line[idx1+1:idx2]
                    break
        if number:
            subprocess.run(["sudo", "ufw", "delete", number], check=True, input="y\n", text=True)
            subprocess.run(["sudo", "ufw", "reload"], check=True)
            return True
        return False
    except Exception:
        return False

@app.route("/", methods=["GET", "POST"])
def login():
    if request.method == "POST":
```

```
        username = request.form.get("username")
        password = request.form.get("password")
        if username == USERNAME and password == PASSWORD:
            session["logged_in"] = True
            return redirect(url_for("dashboard"))
        else:
            flash("Невірний логін або пароль", "danger")
    return render_template("login.html")

@app.route("/dashboard", methods=["GET", "POST"])
def dashboard():
    if not session.get("logged_in"):
        return redirect(url_for("login"))
    if request.method == "POST":
        new_ip = request.form.get("new_ip")
        if new_ip:
            if ufw_add_ip(new_ip):
                flash(f"IP {new_ip} додано", "success")
            else:
                flash(f"Не вдалося додати IP {new_ip}", "danger")
    ips = get_ufw_allowed_ips()
    return render_template("dashboard.html", ips=ips)

@app.route("/delete/<ip>")
def delete_ip(ip):
    if not session.get("logged_in"):
        return redirect(url_for("login"))
```



```
    if ufw_delete_ip(ip):
        flash(f"IP {ip} видалено", "success")
    else:
        flash(f"Не вдалося видалити IP {ip}", "danger")
    return redirect(url_for("dashboard"))

@app.route("/logout")
def logout():
    session.pop("logged_in", None)
    return redirect(url_for("login"))

if __name__ == "__main__":
    app.run(host="0.0.0.0", port=5000)
EOF

# ==== danted.conf ====
cat > socks5_proxy_manager/danted.conf << 'EOF'
logoutput: syslog

internal: 0.0.0.0 port = 1080
external: eth0

method: username none
user.privileged: proxy
user.notprivileged: nobody
user.libwrap: nobody
```



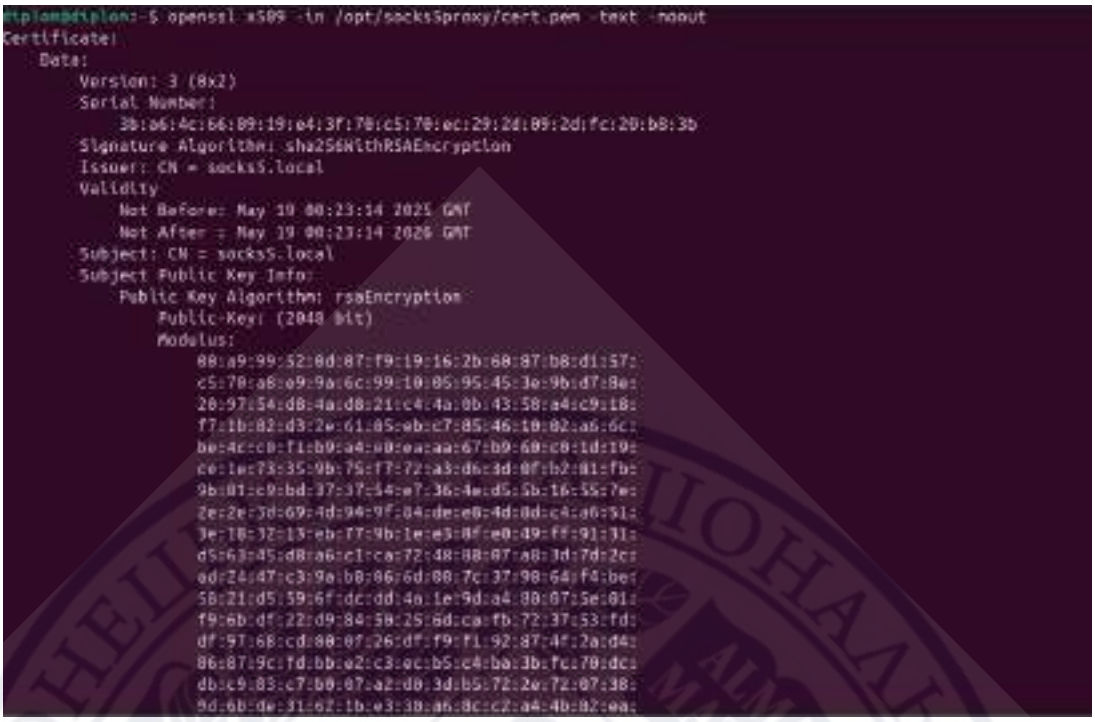
```
client pass {
    from: 0.0.0.0/0 to: 0.0.0.0/0
    log: connect disconnect error
}

pass {
    from: 0.0.0.0/0 to: 0.0.0.0/0
    protocol: tcp udp
    log: connect disconnect error
}
EOF

# ==== requirements.txt ====
echo "Flask==2.3.2" > socks5_proxy_manager/requirements.txt

# ==== README.md ====
cat > socks5_proxy_manager/README.md << 'EOF'
# SOCKS5 Proxy Manager

Flask-додаток для керування доступом до Dante SOCKS5 серверу через UFW.

## Запуск

1. Встановити залежності:
    sudo apt install dante-server ufw
    pip install -r requirements.txt
```

```
2. Запустити Dante:
    sudo danted -f danted.conf

3. Запустити Flask:
    python3 app.py

4. Перейти на http://localhost:5000
EOF

# ==== templates/login.html ====
cat > socks5_proxy_manager/templates/login.html << 'EOF'
<!doctype html>
<html lang="uk">
<head><meta charset="utf-8"><title>Вхід</title></head>
<body><form method="post"><input name="username"><input name="password" type="password"><button>Login</button></for
</html>
EOF

# ==== templates/dashboard.html ====
cat > socks5_proxy_manager/templates/dashboard.html << 'EOF'
<!doctype html>
<html lang="uk">
<head><meta charset="utf-8"><title>Панель</title></head>
<body>
<form method="post"><input name="new_ip"><button>Додати</button></form>
<ul>
{% for ip in ips %}
```

```
  <li>{{ ip }} <a href="{{ url_for('delete_ip', ip=ip) }}">Видалити</a></li>
{% endfor %}
</ul>
</body>
</html>
EOF

# ==== Створення архіву ====
zip -r socks5_proxy_manager.zip socks5_proxy_manager

echo "Готово: архів socks5_proxy_manager.zip створено."
```

2. SSL/TLS - сертифікат

```
diplom@diplom:~$ sudo mkdir -p /opt/socks5proxy
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
  -keyout /opt/socks5proxy/cert.key \
  -out /opt/socks5proxy/cert.pem \
  -subj "/CN=socks5.local"
```



```
ftplamp@tplon:~$ openssl x509 -in /opt/socks5proxy/cert.pem -text -noout
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            3b:a6:4c:66:09:19:e4:3f:70:c5:70:ec:29:2d:09:2d:fc:20:b8:3b
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: CN = socks5.local
        Validity
            Not Before: May 19 00:23:14 2025 GMT
            Not After : May 19 00:23:14 2026 GMT
        Subject: CN = socks5.local
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:a9:99:52:0d:87:f9:19:16:2b:60:87:b8:d1:57:
                    c5:70:a8:e9:9a:6c:99:10:05:95:45:3e:9b:d7:8e:
                    20:97:54:d8:4a:d8:21:c4:4a:0b:43:58:a4:c9:18:
                    f7:1b:82:d3:2e:61:85:eb:c7:85:46:10:02:a6:6c:
                    be:4c:c0:f1:b0:a4:e0:ea:aa:67:b9:60:c0:1d:19:
                    ce:1e:73:35:9b:79:f7:72:a3:d6:3d:0f:b2:81:fb:
                    9b:07:c9:bd:37:37:54:e7:36:4e:d5:5b:16:55:7e:
                    2e:2e:3d:69:4d:94:9f:04:de:e0:4d:0d:c4:a6:51:
                    3e:18:32:13:eb:f7:9b:1e:e3:0f:e0:49:ff:91:31:
                    d5:63:45:d8:a6:c1:ca:72:48:88:07:a8:3d:7d:2c:
                    ed:24:47:c3:9a:b0:86:6d:00:7c:37:90:64:f4:be:
                    58:21:d5:59:6f:dc:dd:4a:1e:9d:a4:80:07:5e:01:
                    f9:6b:df:22:d9:84:50:25:6d:ca:fb:72:37:53:fd:
                    df:97:68:cd:80:0f:26:df:f9:f1:92:87:4f:2a:d4:
                    86:87:9c:fd:bb:e2:c3:ec:b5:c4:ba:3b:fc:70:dc:
                    db:c9:83:c7:b0:07:a2:d0:3d:b5:72:2e:72:07:38:
                    9d:6b:de:31:62:1b:e3:30:a6:0c:c2:a4:4b:82:ea:
                    27:7d
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Subject Key Identifier:
                73:19:48:59:21:D9:2C:39:9A:0C:57:0F:8C:55:90:7F:8F:7B:C8:4A
            X509v3 Authority Key Identifier:
                73:19:48:59:21:D9:2C:39:9A:0C:57:0F:8C:55:90:7F:8F:7B:C8:4A
            X509v3 Basic Constraints: critical
                CA:TRUE
    Signature Algorithm: sha256WithRSAEncryption
    Signature Value:
        13:15:f9:b5:40:4e:b4:80:b4:3c:8a:d3:14:d0:75:44:42:43:
        c0:f1:ed:fb:dc:b8:f5:85:df:b5:82:0f:3b:c6:a8:2f:d0:6d:
        dd:b7:49:5c:37:cf:cc:b8:78:85:6d:51:96:69:d9:92:b1:a0:
        ae:ee:1e:3a:6c:12:81:04:a3:54:1c:29:1f:d2:80:02:aa:aa:
        b6:f6:0e:eb:d5:2f:7c:57:de:46:bd:36:bc:3b:fa:f3:cd:03:
        8b:f5:0e:34:10:c4:79:3b:8b:fd:16:3a:a6:97:e0:05:85:32:
        88:37:9b:73:ab:4f:e4:44:2d:1f:fb:8b:2f:ef:a1:b9:55:10:
        3d:0b:3a:80:01:05:c5:40:63:76:43:b0:69:c1:41:86:98:77:
        c1:3c:76:db:f2:9c:db:3e:96:0d:8f:58:2e:ff:ca:16:92:a2:
        ff:4e:21:6e:f5:33:73:9f:55:60:33:32:46:4f:c7:0f:9f:43:
        d2:e7:d8:7d:a3:d0:08:d4:90:20:bd:88:9c:e5:87:32:ce:d2:
        55:42:c3:c6:a7:18:2a:c2:58:a5:b3:d1:d6:29:ab:0e:15:71:
        64:14:54:50:90:19:51:0f:5e:1e:1d:45:59:28:73:43:bf:32:
        59:c7:87:1e:2f:82:16:ad:11:6b:56:b8:fc:89:6c:e1:78:47:
        35:44:e3:87
```