МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Тернопільський національний економічний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерної інженерії

Бабій Юрій Васильович

Захист програмного засобу FreePBX від зовнішніх атак / FreePBX protection from external attacks

спеціальність: 6.050102 - Комп'ютерна інженерія
освітньо-професійна програма - Комп'ютерні системи та мережі

Випускна кваліфікаційна робота

Виконав: студент групи КСМ-41
Бабій Юрій Васильович
_____

Науковий керівник:
І.Є. Романець
_____

Випускну кваліфікаційну роботу
допущено до захисту:

"___" _____ 20___ р.

Завідувач кафедри
О. М. Березький

ТЕРНОПІЛЬ - 2019

РЕЗЮМЕ

Дипломний проект містить 56 сторінок пояснюючої записки, 21 рисунок, 8 таблиць, 2 додатки. Обсяг графічного матеріалу 2 аркуші формату А3.

Метою дипломного проекту є розробка захисту програмного засобу FreePBX від зовнішніх атак.

Розглянуто задачу налаштування захисту IP-телефонії. Встановлено що для налаштування потрібно розв’язати такі задачі: визначити програмну АТС яку ми будемо використовувати, визначити вразливі, до безпеки, місця обраної АТС, проаналізувати можливі варіанти захисту, розробити та реалізувати налаштування для захисту.

Для налаштування захисту обрано відкриту комунікаційну платформу Asterisk. Захист забезпечується за допомогою внесення корективів у конфігураційні файли Asterisk та встановлення програми захисту серверів від атак – Fail2Ban разом із iptables, це також забезпечить гнучкість нашої системи.

Ключові слова: FREEPBX, ЗАХИСТ IP-ТЕЛЕФОНІЇ, ASTERISK, FAIL2BAN, ЗАХИСТ IP-АТС.

RESUME

The diploma project contains 56 pages of explanatory records, 21 drawings, 8 tables, 2 appendixes. The volume of graphic material is 2 sheets of A3 format.

The method of the diploma project is to develop the protection of FreePBX software from external attacks.

The problem of setting protection IP-telephony is considered. It is established that the following tasks need to be resolved: to determine the software IP-PBX that we will use, to identify vulnerable, to the security of the chosen PBX, to analyze the possible options for protection, to develop and implement security settings.

The Asterisk open communication platform has been selected to set up security. Protection is provided by making adjustments to the Asterisk configuration files and installing a server security attack program – Fail2Ban with iptables, which will also ensure the flexibility of our system.

Keywords: FREEPBX, PROTECT IP-TELEPHONE, ASTERISK, FAIL2BAN, PROTECT IP-PBX.

ЗМІСТ

ВСТУП

Для будь-якого сучасного підприємства заходи забезпечення безпеки локальної мережі і окремих терміналів, що працюють в ній від непередбачених атак, крадіжки конфіденційної інформації та інших важливих даних завжди стоїть на першому місці. Але більш важливо не просто захищати систему, але ще зробити певний аналіз, на основі якого можна зробити висновки про те, як цю безпеку збільшити та покращити в подальшому.

Для забезпечення надійного захисту мережі часто доводиться застосовувати велику кількість окремих компонентів, кожен з яких вимагає ретельної настройки.

У зв’язку з цим метою даної роботи є розробка функціоналу, який збільшить безпеку, гарантовану стандартними засобами тієї системи, в якій ведеться робота.

Мета роботи: розробка комплексу по автоматизації аналізу спроб зовнішніх проникнень і контролю локальних з’єднань для сервера IP-телефонії.

1 СТАН ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Дослідження предметної області

IP-телефонія – це технологія, що дозволяє використовувати Інтернет або іншу IP-мережу як засіб організації телефонних переговорів. Основна ідея полягає в тому, що людська мова у вигляді оцифрованих даних передається по мережі Інтернет, яка надає можливість доставляти дані по всьому світу за ціною, що не залежить від відстані. IP-телефонія дозволяє об'єднати телефонні мережі і мережі передачі даних і здійснювати міжнародні переговори за ціною локальних.

IP-телефонія, або VoIP (Voice over IP), поєднує в собі високу якість і зручність використання традиційного зв'язку з технологією пакетної передачі даних. При звичайному телефонному дзвінку користувач платить за всі етапи з'єднання, тому міжнародні переговори коштують дорого. Що стосується IP-телефонії, то користувач платить Інтернет-сервіс-провайдеру тільки за доступ в Інтернет і за організацію з'єднання з провайдером в іншій країні, в зв'язку з чим користуватися IP-телефонією особливо вигідно на напрямках телекомунікації. Сучасні технології дозволяють надавати послуги IP-телефонії, якість яких близько до якості традиційного телефонного зв'язку.

IP-телефонія може бути реалізована за схемами комп'ютер-комп'ютер або телефон-телефон.

При схемі комп'ютер-комп'ютер вам знадобляться звичайний ПК із звуковою картою, навушники (або колонки) і мікрофон, модем зі швидкістю підключення не менше 28 800 біт / с і відповідне ПО. На рисунку 1.1 зображено схему комп'ютер-комп'ютер [1].

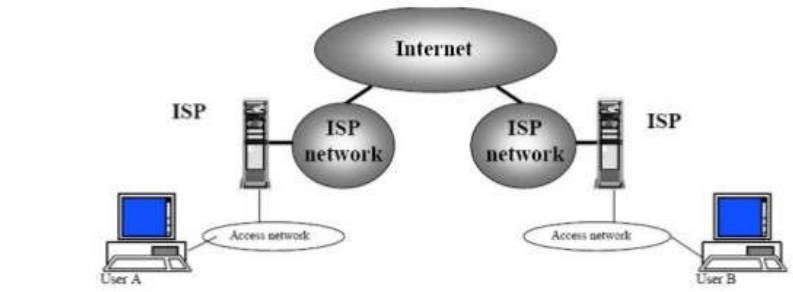

Рисунок 1.1 - Схема комп'ютер-комп'ютер

У схемі телефон-телефон дзвінок здійснюється зі звичайного апарату на телефон найближчого шлюзу IP-мережі, після чого абонента просять набрати в тоновому режимі пароль, код країни, міста і номер телефону. Шлюз приймає стандартний телефонний сигнал, оцифровує його, стискає, розбиває на IP-пакети і відправляє через Інтернет на аналогічний шлюз за місцем призначення, де відбувається зворотне перетворення цифрових пакетів в телефонну розмову і забезпечується вихід на телефонну мережу загального користування, дивитися рисунок 1.2.

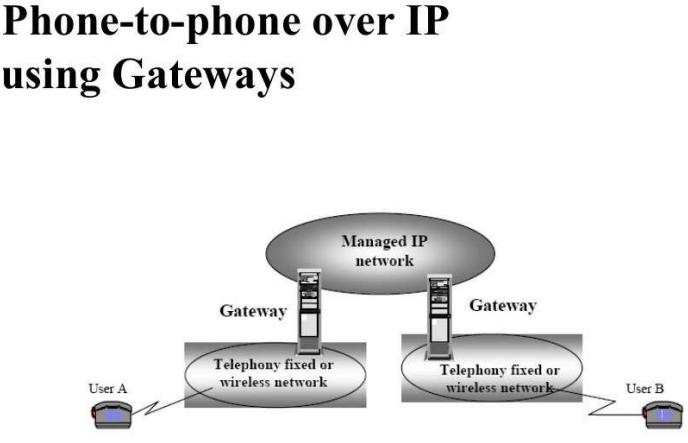

Рисунок 1.2 – Схема телефон-телефон

Крім того, можливі переговори за схемами телефон-комп'ютер і комп'ютер-телефон. IP-телефонія відкриває для користувачів нові можливості,

такі як інтерактивна електронна торгівля, голосовий зв'язок при спільній роботі на віддалених один від одного комп'ютерах і т.д. Ринок розвивається дуже швидко з 1997 року він зріс майже в 60 разів. Найбільші споживачі устаткування IP-телефонії - високорозвинені регіони світу, особливо США і Європа. До 2006 року Інтернет-телефонія складала близько 35% від усіх дзвінків, а 2010-го понад 50% всіх голосових викликів передавалися через Інтернет.

Система відео-конференцій дає можливість не тільки почути, а й побачити свого співрозмовника. Відео-конференція - це засіб спілкування територіально віддалених людей на базі використання відео в комп'ютерних мережах. Відео-конференції часто використовуються в корпоративних мережах (без Інтернету), де швидкість каналу Ethernet становить 100 Мбіт / с (цього більш ніж достатньо для проведення відео-конференції), проте саме Інтернет дозволяє з'єднувати відео-зв'язком людей, що знаходяться в будь-яких точках земної кулі, де є відповідне обладнання. Зазвичай відеозв’язок супроводжується можливістю обміну аудіо і текстовою інформацією. Сукупність аудіовізуального і текстового спілкування дозволяє розподіленим колективам працювати над спільними проектами.

Головна проблема, пов'язана з IP-телефонією і відео-конференціями, полягає в тому, що вони занадто відкриті, і зловмисники можуть відносно легко здійснювати атаки на їх компоненти. Оскільки IP-телефонія це поєднання звичайної телефонії і IP-технології, вона увібрала в себе не тільки їхні переваги, але і недоліки. Атаки, властиві звичайній телефонії, також можуть бути застосовані і для її IP-складової. Незважаючи на те, що випадки таких нападів поки не особливо поширені, хакери при бажанні в змозі їх реалізувати, тому що атаки на звичайні IP-мережі можуть бути практично без змін спрямовані і на мережі передачі оцифрованого голосу і відео. З іншого боку, схожість звичайних IP-мереж, мереж IP-телефонії та відео-конференцій підказує нам і шляхи їх захисту.

Найбільш розповсюджені реалізації IP-телефонії не підтримують криптографічне шифрування, навіть незважаючи на те, що безпечне з’єднання

набагато простіше впровадити в межах IP-технологій, ніж в стандартних телефонних лініях. Через це, за допомогою аналізатора трафіку нескладно встановити прослуховування IP-дзвінків, а або навіть змінити їхній зміст. Зловмисник, який використовує аналізатор мережевих пакетів, має можливість перехопити IP-дзвінок, якщо користувач не знаходиться у межах захищеної віртуальної мережі VPN. Ця вразливість в безпеці може привести до атак які приведуть до некоректної роботи системи (відмовами в обслуговуванні) у користувача або у того, чий номер належить тій же мережі. Ці проблеми в обслуговуванні можуть повністю знищити телефонну мережу, навантаживши її сміттєвим трафіком і створивши постійний сигнал «зайнято» що збільшить кількість роз'єднань абонентів. Ця проблема також стосується і традиційної телефонії, оскільки абсолютно захищених способів зв'язку не існує.

Споживачі можуть збільшити безпеку мережі, якщо обмежать доступ в віртуальну локальну мережу даних, сховавши свою мережу з голосовими даними від користувачів. Якщо споживач підтримує безпечний і правильно конфігурує міжмережевий інтерфейс-шлюз з контрольованим доступом, це дозволить убезпечити себе від більшості хакерських атак. Існує безкоштовне ПЗ, таке як Wireshark, воно полегшує аналіз трафіку IP-розмов. Деякі вендори використовують стиснення, щоб перехоплення інформації було важче реалізувати. Є думка, що справжня безпека мережі вимагає проведення повного криптографічного шифрування і криптографічної аутентифікації.

IP-телефонія, за деякими характеристиками, виграє у традиційної в плані безпеки, завдяки існуючим на даний момент стандартам безпеки SRTP, нового ZRTP протоколу, який доступний на деяких моделях IP-телефонів (Cisco, Yealink SNOM і ін.), аналогових телефонних адаптерах (Analog Telephone Adapters, ATAs), шлюзах, а також софтфонах. Також є можливість використовувати IPsec, щоб забезпечити безпеку P2P VoIP за допомогою застосування альтернативного шифрування (opportunistic encryption). Поєднання технології VoIP і VPN (Voice VPN) надає можливість створення безпечного голосового з'єднання для VoIP-мереж всередині компанії шляхом застосування IPSec-шифрування до оцифрованого потоку голосових даних.

Також можливо зробити багаторівневе шифрування та анонімізацію усього VoIP-трафіку (голосу, відео, службової інформації і т.д.) використовуючи мережу I2P [2].

У IP-телефонії є можливість ідентифікації абонента. Така послуга визначення номера абонента (caller ID) у різних провайдерів може відрізнятися, хоча більшість VoIP-провайдерів зараз пропонують послугу визначення ідентифікатора абонента з ім'ям на вихідні дзвінки. Коли виклик йде на номер місцевої мережі від якогось VoIP-провайдера, послуга визначення caller ID не підтримується. У деяких випадках VoIP-провайдери можуть дозволити абонентові імітувати caller ID, який йому не належить, потенційно даючи можливість демонструвати такий ID, який фактично не є номером абонента. Комерційне VoIP-обладнання та програмне забезпечення зазвичай легко дає можливість змінювати інформацію caller ID. Незважаючи на те, що ця послуга може забезпечити величезну свободу дій, вона також дає можливість для зловживань.

1.2 Опис об’єкту дослідження

PBX Asterisk це програмне забезпечення, з відкритим вихідним кодом, яке в комплексі з необхідним обладнанням має всі можливості класичної автоматичної телефонної станції.

Автоматична телефонна станція, АТС - система пристроїв, що забезпечує автоматичне (без участі оператора або телефоністок) з'єднання і підтримку телефонного зв'язку між абонентами цієї АТС [4].

Asterisk підтримує безліч VoIP-протоколів і надає багаті функції управління дзвінками, серед них:

- голосова пошта;
- конференц зв'язок;

• IVR (інтерактивне голосове меню);

• центр обробки дзвінків (постановка дзвінків в чергу і розподіл їх по абонентам, використовуючи різні алгоритми);

• Call Detail Record (докладний запис про виклик).

На рисунку 1.3 зображено IP АТС Asterisk

Рисунок 1.3 – Схема АТС

Для створення додаткової функціональності можна скористатися Asterisk Gateway Interface – гнучким та універсальним інтерфейсом для інтеграції з зовнішніми системами обробки даних або власною мовою Asterisk для написання плану нумерації, також можна написати модуль на мові C. Модулі, що виконуються через Asterisk Gateway Interface, можуть бути написані на будь-якій мові програмування. Asterisk розповсюджується на умовах подвійної ліцензії, завдяки якій одночасно з основним кодом, поширюваним по відкритій ліцензії GNU GPL, можливе створення закритих модулів, що містять ліцензований код (наприклад, модуль для підтримки кодека G.729). Asterisk розроблений для Linux, він працює на безлічі операційних систем, включаючи NetBSD, OpenBSD, FreeBSD, MacOS, і Solaris [3].

Asterisk може працювати як з аналоговими лініями (FXO/FXS-модулі), так і цифровими (ISDN, BRI і PRI-потоки T1/E1). За допомогою певних комп'ютерних плат (найбільш відомими виробниками яких є Digium, Sangoma,

OpenVox, Rhino, AudioCodes) Asterisk можна підключити до високопропускних ліній T1/E1, які дозволяють працювати паралельно з десятками телефонних з'єднань. Повний список обладнання для з'єднання з телефонною мережею загального користування визначається підтримкою обладнання в модулях ядра.

Asterisk підтримує дані протоколи:

- SIP;
- H.323;
- IAX2;
- MGCP;
- SIMPLE;
- Skinny/SCCP;
- XMPP;
- Unistim;
- DUNDi;
- OSP;
- T.38.

Підтримка широкого спектру обладнання та комп'ютерних протоколів дозволяє створити величезну кількість взаємодій між мережами, отриманням і обробку інформації.

У IP-АТС Asterisk багаторівнева система захисту. Для кожного конкретного рішення по IP-телефонії необхідно підбирати необхідні засоби захисту, які будуть відповідати компонентам рішення і його особливостям. Якщо розглядати систему безпеки IP-АТС Asterisk, то можна виділити досить потужну систему захисту від атак, що гарантує комплексний захист всіх елементів системи на найвищому рівні. Ті компоненти, які особливо часто піддаються атакам, мають додаткові рівні захисту:

- на рівні операційної системи – вбудовані засоби захисту Linux;
- на рівні Asterisk – захищена конфігурація і автоматичне оновлення ядра Asterisk;

• на рівні сторонніх компонентів, які стежать за безпекою Asterisk – системи автоматичного спостереження за логами Asterisk, Fail2Ban;

• на рівні VPN для захисту трафіка, що передається – використання OpenVPN або IPSec для передачі голосу, а також управління Asterisk [4].

Важливо також розуміти, що процес установки і розгортання самої системи IP-телефонії також багато в чому обумовлює ефективність захисту і рівня загальної безпеки деяких компонентів.

В даний час Asterisk є найпопулярнішою відкритої IP АТС в світі, займаючи майже 85% «ринку» open source PBX (а в цілому відкриті АТС займають близько 18% ринку PBX в USA).

1.3 Аналіз існуючих технічних рішень

Найбільш популярними IP АТС з відкритим кодом на даний момент є:

• Asterisk;

• FreeSWITCH;

• Yate.

Розглянемо кожні із цих систем, оскільки Asterisk уже описано в попередньому пункті, почнемо з FreeSWITCH.

FreeSWITCH - це програмний комутатор, створення якого було ініційовано одним з колишніх розробників Asterisk - Ентоні Мінессейлом (Anthony Minessale) в 2006 році. Після численних спроб використання Asterisk під високим навантаженням, Ентоні висловив ряд зауважень до базової архітектурі системи, і запропонував її змінити. Однак, автор Asterisk - Марк Спенсер, відмовився змінювати ядро. Тому Ентоні вийшов зі складу розробників Asterisk і створив «з нуля» свій продукт, який він назвав FreeSWITCH.

При розробці архітектури FreeSWITCH авторами були враховані всі проблеми існуючих відкритих програмних продуктів для IP телефонії. Тому одними з головних достоїнств нового продукту стали стабільність роботи і масштабованість, а також крос-платформенність - FreeSWITCH працює під управлінням як Linux, так і Windows. Іншою особливістю FreeSWITCH є використання SIP стека sofia-sip від Nokia, який вважається найкращою відкритою реалізацією SIP протоколу, поширюваної в вихідному коді. В Asterisk ж chan_sip реалізований з неповним дотриманням стандартів. SIP є основним протоколом роботи FreeSWITCH, хоча також підтримуються і драйвери PCI плат для інтеграції з традиційною телефонією, а також інші протоколи IP телефонії.

FreeSWITCH може використовуватися як SIP проксі і SIP реєстратор, як Session Border Controller (SBC), транскодуючий Back-to-back User Agent (B2BUA), як сервер конференцій або голосової пошти. Також FreeSWITCH підтримує і багато функцій IP PBX, такі як переклад дзвінка, перехоплення, парковка виклику, запис розмов, прослуховування та інші. Однак, на сьогоднішній день список додатків IP PBX, доступний для FreeSWITCH, програє аналогічному в Asterisk.

Основним інтерфейсом конфігурації FreeSWITCH є текстові файли у форматі XML, що ускладнює адміністрування цієї системи, тоді як в Asterisk застосовуються файли .ini які добре читаються. Для FreeSWITCH відсутні готові до використання графічні інтерфейси з управління, що також ускладнює його використання. А існуючі GUI для FreeSWITCH (WikiPBX, FusionPBX, blue.box) далекі за функціональністю від того ж FreePBX для Asterisk.

Проте, FreeSWITCH активно розвивається. Деякі експерти відкритих програмних продуктів для телекомунікацій називають FreeSWITCH «Asterisk killer app», інші стверджують, Що для обох продуктів є місце на ринку, так як у кожного з них своя унікальна специфіка [5].

Проект Yet Another Telephone Engine (Yate) було розпочато в 2004 році. Підтримуються операційні системи: Linux, BSD, Windows.

Написаний Yate на C ++. Yate не використовує зовнішніх SIP бібліотек, а реалізує SIP стек самостійно. Yate – це софтсвітч, який містить також багато PBX функції, зокрема:

- переклад, утримання і парковку виклику;
- музику на очікуванні;
- конференц зв'язок;
- черги;
- IVR;
- статистику дзвінків.

Однак, Yate в першу чергу - це мультипротокольний комутатор з дуже гнучкими правилами маршрутизації. Yate добре підтримує такі протоколи IP телефонії, як H323, IAX2, MGCP, різні рівні SS7 (MTP2, SIGTRAN), драйвера потокових цифрових плат різних виробників. Також Yate включає в себе механізм кластеризації.

Архітектурно Yate використовує модель мікро ядра і шини повідомлень, а для маршрутизації повідомлень використовуються регулярні вирази з можливістю розміщення будь-яких повідомлень на шині. Така архітектура робить простим додавання нових модулів, не зачіпаючи існуючого коду. Існує спеціальний вільний дистрибутив з Yate і WEB інтерфейсом з управління - FreeSentral, що включає в себе інтерфейс користувача, де він керує своїми налаштуваннями, такими як переадресація, голосова пошта, записна книжка, а також може переглядати статистику своїх дзвінків.

Серед усіх розглянутих продуктів Yate володіє найменшим функціоналом, однак те, що Yate вміє робити, робить дуже добре і стабільно. Ще одним недоліком є недостатня документація. Найбільш часте застосування Yate - конвертер H323-SIP сигналізації [6].

Оскільки у Yate та FreeSWITCH функціонал менший чим у Asterisk, було вирішено використовувати IP АТС Asterisk.

1.4 Постановка задач дипломного проекту

Метою дипломного проекту є налаштування захисту FreePBX. Для досягнення мети потрібно розв’язати наступні задачі:

- проаналізувати розвиток IP телефонії;
- проаналізувати технології АТС;
- проаналізувати вразливі місця IP-телефонії;
- розглянути різні варіанти налаштувань захисту IP-телефонії;
- розробити та реалізувати налаштування захисту IP-телефонії;
- протестувати налаштований захист.

В даному розділі було проаналізовано розвиток IP телефонії, технології АТС, вразливі місця IP-телефонії, проаналізовано можливі безкоштовні альтернативи Asterisk такі як Yate та FreeSWITCH, в результаті було визначено що у Yate та FreeSWITCH функціонал менший чим у Asterisk. Розглянуто усі можливості Asterisk.

2 РОЗРОБКА АРХІТЕКТУРИ ПРОГРАМНОЇ СИСТЕМИ

2.1 Розробка структури системи

Електронна автоматична телефонна станція призначена для роботи у міських та міжміських телефонних мережах. Завдяки стандартизованим інтерфейсам для зовнішніх і внутрішніх підключень АТС сумісна з будь-якими типами обладнання, в сучасній системі зв’язку. Завдяки модульній структурній побудові, є можливість змін технічних характеристик та функціональних можливостей в широкому діапазоні.

У загальному АТС забезпечує такі основні види обслуговування:

- внутрішньостанційний зв’язок;
- транзитний зв'язок;
- вхідний та вихідний автоматичний зв’язок на загальнодержавній телефонній мережі.

Основною особливістю структури є модульний принцип побудови АТС. Вона будується з наступних модулів:

- модуль цифрових сполучних ліній;
- модуль комутаційного поля;
- модуль абонентських ліній;
- модуль аналогових сполучних ліній;
- модуль технічного обслуговування.

Кожен модуль має вузол управління (ВУ), який містить: блок центрального процесорного пристрою (ЦПП), блок системної пам'яті (БСП), блок місцевої синхронізації (БМС), блок обміну повідомленнями (БОП). В залежності від функціонального призначення модуля ВУ він також може містити інші блоки.

З метою покращення роботи АТС два однотипних модуля з'єднуються системної шиною, утворюючи загальний мережевий вузол. При виході з ладу

блоків одного з вузлів інший ВУ може керувати обома модулями з невеликою втратою продуктивності.

Модуль комутаційного поля (МКП) забезпечує підключення абонентів між собою, абонентів на сполучну лінію, або сполучних ліній одна з іншого. Ємність МКП може нарощуватися блоками комутаційного поля.

Модуль абонентських ліній (МАЛ) відповідає за підключення абонентів телефонної мережі. Ємність МАЛ – 256 абонентських ліній. Залежно від потрібної концентрації, 1:4 або 1:8, МАЛ з'єднується з МКП двома або одним ІКМ трактом. У першому випадку абонентам може бути представлена навантаження до 0,24 ерланга на одну абонентську лінію, в другому випадку до 0,12 ерланга на одну абонентську лінію. Взаємодія з іншими модулями АТС здійснюється завдяки внутрішньостанційній мережі.

Модуль цифрових сполучних ліній (МЦСЛ) забезпечує взаємодію з іншими АТС місцевої мережі за стандартними ІКМ трактами і призначений для обробки лінійної сигналізації з різними протоколами взаємодії. Модуль цифрових сполучних ліній повинен забезпечувати взаємодію з іншими АТС використовуючи два виділених канали систем передачі ІКМ для односторонніх СЛ, а також по одному виділеному каналу систем передачі ІКМ для універсальних СЛ двосторонньої дії.

Модуль аналогових сполучних ліній забезпечує взаємодію станції з іншими АТС місцевої мережі по трьох дротових сполучних лініях. Модуль аналогових сполучних ліній забезпечує стик та обробку сигналізації до 48 трипровідних СЛ. Ємність модуля нарощується блоками по 4 сполучні лінії.

Модуль технічного обслуговування (МТО) забезпечує збір інформації про стан всіх модулів АТС, обробку інформації, тестування модулів, реконфігурацію АТС при відмові окремих блоків і діагностику відмовили блоків.

Внутрішньостанційна мережа є важливим елементом АТС, вона забезпечує обмін інформації між усіма модулями станції. Пропускна здатність мережі - 2 Мбіт/сек. У цій мережі використовується один із протоколів HDLC, він забезпечує необхідну достовірність передачі інформації.

Технічно мережеві засоби підтримуються спеціальним адаптером - блоком обміну повідомленнями (БОП). БОП призначений для організації внутрішньостанційної мережі передачі службових повідомлень між пристроями управління різних модулів АТС. БОП включає у себе такі функціональні вузли: інтерфейсну схему з'єднання з внутрішньою мережею АТС; внутрішню мікро-ЕОМ, що керує обміном по мережі; інтерфейсну схему призначену для обміну з центральним процесором свого модуля [7].

На рисунку 2.1 зображено схему міні АТС.

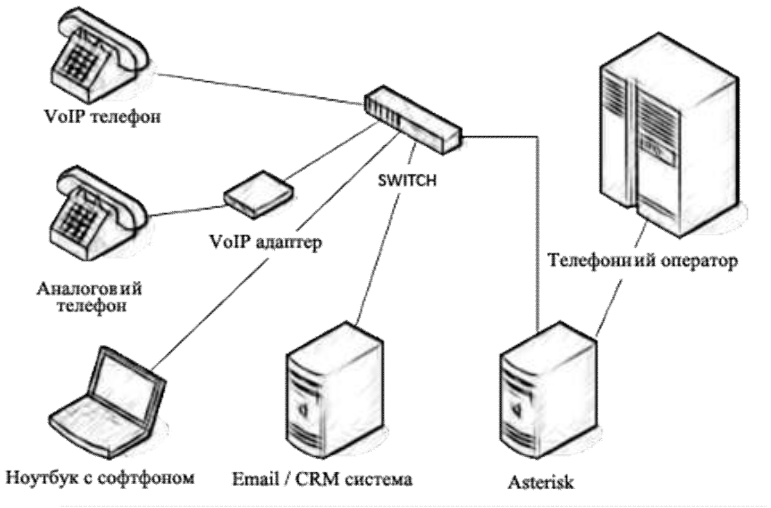

Рисунок 2.1 – Схема міні АТС

2.2 Аналіз існуючих алгоритмів розв'язання поставленої задачі

Безпека є однією з найбільш часто обговорюваних тем, але важливість забезпечення VoIP важко переоцінити. Кожен пристрій і сервіс частково відповідають за забезпечення безпечного VoIP, але є кілька різних способів розгортання безпечного VoIP. Традиційна телефонія, передбачає передачу по певному фізичному середовищу. Атаки на традиційну телефонію, такі як підслуховування, вимагають фізичної присутності з доступом до фізичних ліній.

Говорячи про безпеку рішення IP-АТС в цілому потрібно розуміти, що безпека будується не тільки на безпеці самого Asterisk, так само необхідно забезпечити безпеку оточення Asterisk.

Забезпечити захист IP-АТС можна використовуючи:

- мережевий захист;
- аналіз логів (fail2ban);
- конфігурацію Asterisk;
- захист планом маршрутизації дзвінків (dialplan);
- конфігурацію Linux;
- захист периферійних пристроїв;
- адміністративні заходи [8].

На рисунку 2.2 зображено підключення Asterisk з використанням firewall.

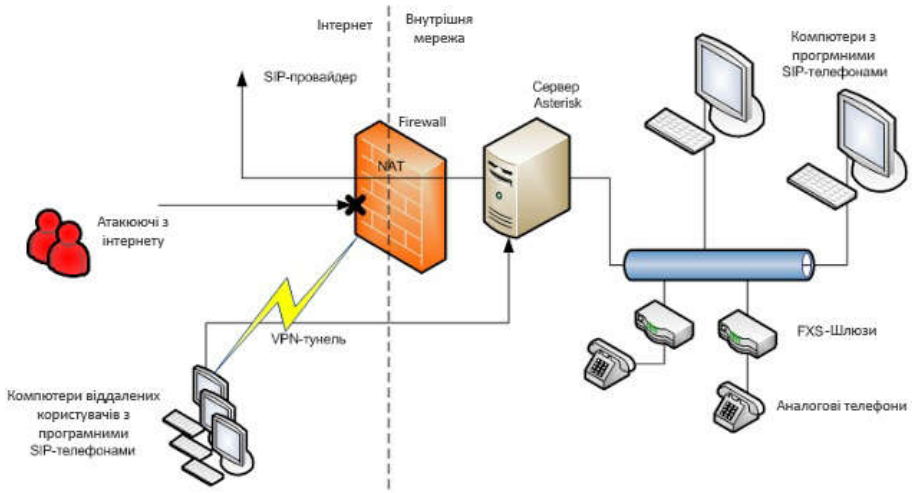

Рисунок 2.2 – Схема підключення Asterisk з використанням firewall

2.3 Алгоритмічна реалізація системи

Мережевий захист. При налаштуванні системи використовуємо мережевий екран (firewall). У Linux вбудований потужний і гнучкий інструмент IPTables, який управляється командами.

За замовчуванням IPTables налаштовуємо таким чином:

- A INPUT -m state --state ESTABLISHED, RELATED -j ACCEPT (дозволяємо пакети для вже встановлених з'єднань).

- A INPUT -p icmp -j ACCEPT (дозволяємо пакети протоколу icmp).

- A INPUT -i lo -j ACCEPT (дозволяємо трафік з інтерфейсу lo).

- A INPUT -m state --state NEW -m tcp -p tcp -dport 22 -j ACCEPT (дозволяємо нове з'єднання ssh).

- A INPUT -j REJECT --reject-with icmp-host-prohibited (забороняємо всі інші вхідні з'єднання).

- A FORWARD -j REJECT - reject-with icmp-host-prohibited (забороняємо всі транзитні з'єднання).

- COMMIT.

Дані налаштування визначають – дозволяємо ініційовані нами з'єднання, забороняємо всі вхідні з'єднання крім ssh. Далі налаштовуємо під свої завдання. Якщо трафік пробіг всі елементи ланцюжка і не потрапив ні під одне правило то це невизначений трафік, отже трафік забороняємо. Тобто, для того щоб до сервера Asterisk могли підключатися ip-телефони, софтфони з внутрішньої мережі, нам необхідно прописати правило:

- A INPUT -s xxx.xxx.xxx.xxx/24 -j ACCEPT

Дане правило дозволяє вхідні з'єднання з мережі xxx.xxx.xxx.xxx\24. Так як пакет за правилами проходить зверху вниз, треба розуміти, що дане правило буде працювати якщо ми його пропишемо перед правилами що забороняють. Таким чином, можна налаштувати мережевий екран, так що Asterisk обслуговуватиме тільки ті напрямки і тільки ті пакети, яким ми довіряємо, і зловмисникові буде практично неможливо дістатися до сервера.

Також використовується поділ даних. Розділяєм голос та дані, за допомогою побудови Vlan. Комп'ютери, сервера, та інше мережеве обладнання повинні знаходитися в одних Vlan-ах, а обладнання, що працює з Asterisk (сам сервер, gate, ip-телефони ...) в іншій Vlan. Це робиться для того щоб користувачі, та можливі віруси що знаходяться на користувацьких ПК не могли нашкодити IP-АТС і навпаки.

Схема з Vlan представлена на рисунку 2.3.

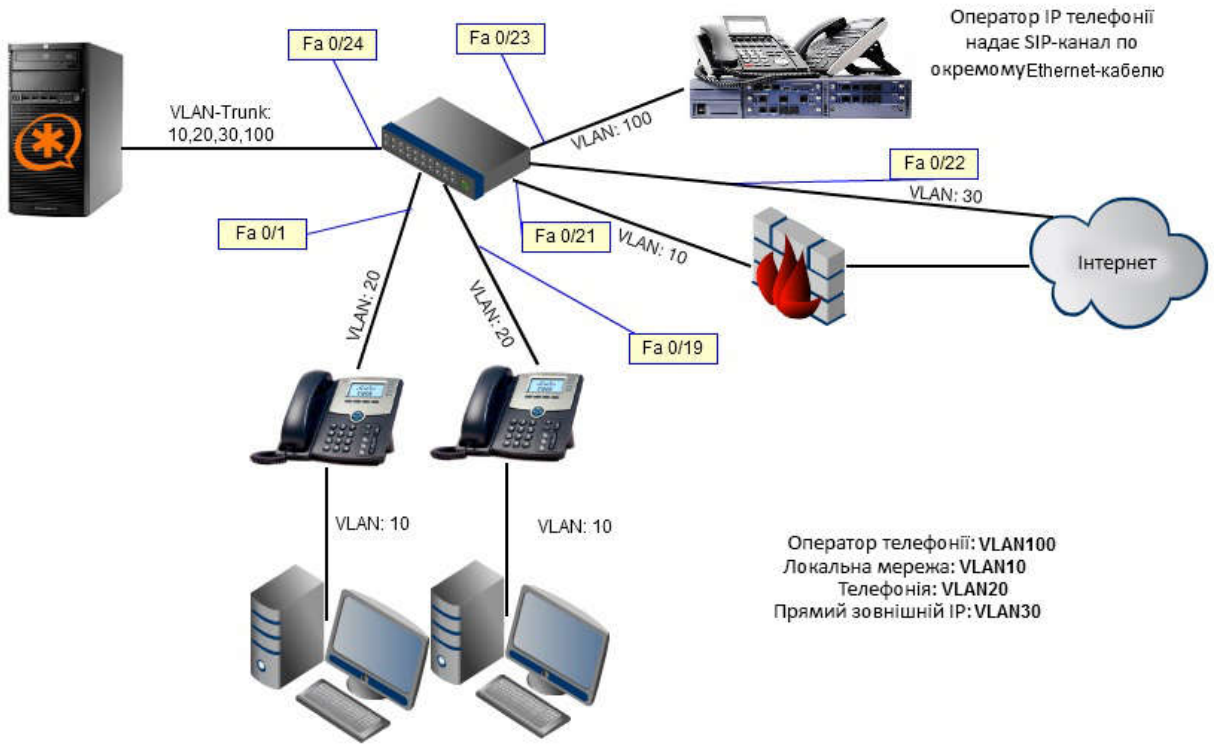

Рисунок 2.3 – Схема поділу голосу і даних в різні Vlan

Аналіз логів. В процесі роботи системи IP-АТС сам Asterisk та інші додатки записують усі свої дії в логи. Так як в лог потрапляють дані, як про санкціоновані дії, так і про несанкціоновані дії, то дуже важливо час від часу переглядати їх на предмет інцидентів. Необхідно проводити контроль підключень, контроль підбору паролів, відстеження великої кількості спроб підключень, контроль незапланованої активності.

Оскільки в Asterisk немає інструменту, який би припиняв можливість перебору паролів, використаємо зовнішню службу Fail2Ban. Служба «читає» логи, аналізує їх і при виявлення множинних спроб входу, блокує дане з'єднання на рівні IPTables. На рисунку 2.4 показана робота служби Fail2Ban.

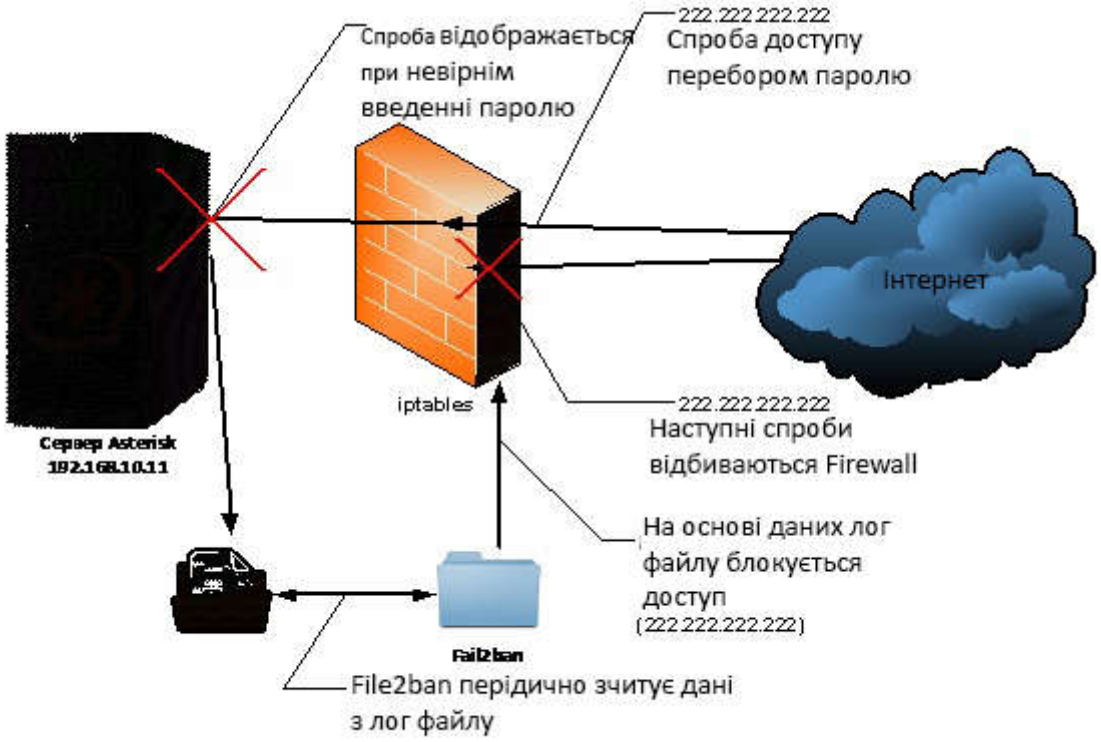

Рисунок 2.4 – Робота служби Fail2Ban

Налаштування конфігураційних файлів Linux. Як в самому Asterisk так і в ОС Linux є служби, які за замовчуванням запускаються, але вони для роботи не потрібні. Тому деякі з них можна відключити як в Asterisk, так і в Linux.

Asterisk завантажує велику кількість модулів, різні кодеки, команди, засоби для роботи з БД, канали. Відключення зайвих модулів дозволить не тільки звільнити пам'ять, але і робити IP-АТС менш вразливою.

У файлі /etc/asterisk/modules.conf прописаний параметр autoload = yes, який змушує Asterisk при завантаженні ініціалізувати і запускати всі служби, які перебувають в папці /usr/lib/asterisk/modules. Для того щоб заборонити запуск тієї чи іншої непотрібної служби, прописуємо в файлі /etc/asterisk/modules.conf наступні рядки:

noload => chan_gtalk.so (модуль каналу Gtalk).

noload => app_morsecode.so (передача заданого рядка азбукою Морзе).

noload => cdr_pgsql.so (Модуль для зберігання даних CDR в БД PostgreSQL).

Так само безпосередньо в самому Linux є служби, які завантажуються, та не потрібні для роботи IP-АТС – їх теж відключаємо. Командою chkconfig -list дивимося які служби запускаються, і визначаємо, які служби зайві.

Для адміністрування ОС і Asterisk адміністратору IP-АТС необхідний віддалений доступ. Не рекомендується використовувати протоколи для віддаленого доступу без шифрування (наприклад telnet), рекомендується використовувати SSH (Secure SHell). Служба SSH, яка використовується для віддаленого входу на сервер – це головні двері в центр управління АТС. Для підвищення рівня безпеки виконаємо наступні кроки.

- Зміна порту. Порт за замовчуванням SSH – 22-ий. Багато хакерів сканують інтернет у пошуках серверів з відкритим 22-им портом, щоб потім спробувати зламати їх. Ставимо інший порт, в діапазоні 1-65535.
- Використання протоколу SSH версії 2.
- Забороняємо прямий доступ користувача root. Це істотно ускладнить і швидше унеможливить атаку на перебір пароля, так як користувачу root буде заборонений доступ в систему, навіть якщо буде введений його коректний пароль. Використання підсистеми sudo для отримання root доступу при необхідності і тільки після віддаленого входу в систему під непрівілегійованим обліковим записом.
- Використання тимчасового обмеження по введенню пароля та сертифікати.
- Всі зазначені вище кроки заносимо в конфігураційний файл /etc/ssh/sshd_config.

```
Port 30222
AllowUsers ats admin
Protocol 2
PermitRootLogin no
LoginGraceTime 1s
```

- Також вхід в SSH виконуємо за сертифікатами. Якщо часто використовувати SSH для підключення до віддаленого хосту, для забезпечення

безпеки з'єднання застосовуємо відкритий/закритий SSH-ключ, так як при цьому по мережі не передається ніякий пароль.

Створення відкритого/закритого SSH-ключа в Linux.

- В консолі вводимо ssh-keygen -t rsa в даному випадку RSA - це асиметричний алгоритм шифрування. Так само можна використовувати і DSA (Digital Signature Algorithm).
- Далі пропонується вказати місце для збереження ключа. За замовчуванням це папка .ssh у вашій домашній директорії.
- Далі нас просять ввести ідентифікаційну фразу. Це не ідентифікаційна фраза для з'єднання з віддаленим хостом. Це ідентифікаційна фраза для розблокування закритого ключа, тому вона не допоможе вам отримати доступ до віддаленого сервера, навіть якщо на ньому зберігається ваш закритий ключ. Введення ідентифікаційної фрази не є обов'язковим. Щоб залишити її порожнюю, натискаємо «Enter».
- Наші ключі згенеровані. Переходимо в домашню директорію в папку .ssh, там повинні знаходиться згенеровані наші ключі id_rsa і id_rsa.pub.
- Правимо конфігураційний файл /etc/ssh/sshd_config.

RSAAuthentication yes

PubkeyAuthentication yes

PasswordAuthentication no

Тим самим ми дозволяємо вхід по ssh тільки за допомогою сертифіката.

- Вміст файлу id_rsa.pub копіюємо до новоствореного файлу authorized_keys. Копіюємо і створюємо файл командою cat id_rsa.pub >> authorized_keys.
- Встановлюємо на файл права на читання і запис chmod 600 authorized_keys.
- Копіюємо приватний ключ id_rsa на комп'ютер, з якого будемо підключатися до сервера по SSH.

• Для того що б клієнт (Putty) зрозумів цей приватний ключ, проженемо (load) його через програму puttygen. На виході (Save private key) отримуємо приватний ключ (* .ppk).

• Тепер додамо ключ в сеанс. Запускаємо PuTTY, завантажуємо потрібний сеанс або вводимо дані для з'єднання і йдемо в «SSH – Auth», вибираємо наш приватний ключ, який був отриманий через обробку «puttygen».

• Йдемо в меню «Connection - Data» і в полі «Auto-login username» вводимо логін під яким генерували ключ.

• Зберігаємо сеанс, перезапускаємо службу SSH на сервері /etc/init.d/ssh reload.

Таким чином, ми підвищили безпеку SSH з'єднання [9].

Адміністративні заходи. Навіть в разі виконання всіх рекомендацій, все одно система може бути зламаною, оскільки немає абсолютної безпеки. Тому адміністративні заходи важливі.

Якщо міжнародні дзвінки не потрібні, то тоді можна заблокувати міжнародні напрямки на рівні оператора зв’язку. Якщо таких дзвінків дуже мало, то можна домовиться з провайдером про якийсь моніторингу даного напрямку – в разі великої кількості дзвінків на міжнародні напрямки – блокувати ці сполуки.

Також можна встановити обмеження суми рахунку за зв’язок на рівні оператора. Можна попросити оператора зв’язку встановити ліміт, який ми самі для себе визначаємо, і в разі досягнення ліміту зв’язок блокується. Це не дасть піти мінус в разі злому IP-АТС.

Один з найважливіших моментів – не забувати, що немає абсолютного захисту. Необхідний постійний контроль і аудит системи на предмет нових вразливостей, виникнення нових ризиків і ін. І в разі виявлення чого-небудь, врахувати це.

3 ТЕХНІЧНА РЕАЛІЗАЦІЯ

3.1 Функціональна структура Fail2ban

Fail2ban – це програма для блокування IP-адрес при перевищенні кількості спроб невдалого введення SSH-пароля або багаторазового запиту веб-сторінки у сервера NGINX. Він призначений для використання на UNIX-подібних операційних системах, наприклад Ubuntu, Debian, CentOS, FreeBSD. IP-адреси для блокування визначаються по log-файлах (/var/log/secure, /var/log/faillog і т.д.). Якщо з будь-якої IP-адреси здійснюється забагато спроб ввійти в захищену систему то хост з даним IP-адресом блокується на деякий час. Самий простий приклад використання – захист SSH віддаленого сервера від перебору пароля [10].

По своїй архітектурі Fail2ban представляє собою систему клієнт-сервер. Серверна частина – це багатопотокова програма, яка прослуховує Unix-сокети, очікуючи на команди та відправляє клієнту необхідну йому інформацію. Усе це відбувається в режимі реального часу, сам сервер не має ніякої інформації про даний статус конфігураційних файлів, тому при запуску знаходиться в стані "за замовчуванням", в якому не визначені налаштування. Клієнтська частина – це зовнішній компонент інтерфейсу всієї підсистеми. Клієнт встановлює з'єднання через сокет сервера і посилає через нього команди для конфігурації сервера та виконання необхідних операцій. Також клієнт може зчитувати і передавати вміст конфігураційних файлів або просто відправити на сервер одну команду, використовуючи для цього командний рядок shell-оболонки або власний інтерактивний режим.

Підсистеми Fail2ban має велику кількість файлів конфігурації. При внесенні змін в конфігурацію, зазвичай добавляться типи файлів .local. Настройки що знаходяться у .local-файлах мають переваги над настройками що є у .conf-файлах. Це означає, що спочатку зчитуються .conf-файли а після них .local-файли, тому значення раніше визначених параметрів можуть бути замінені. Таким чином, в .local-файлах можна зберігати лише ті значення

параметрів, які потрібно скорегувати. Більш того, всі необхідні зміни конфігурації слід вносити до відповідних .local-файли, а не в .conf-файли. Це допомагає підтримувати коректність загальної структури та уникнути проблем при оновленні всієї підсистеми Fail2ban.

Ведення журналів самої підсистеми Fail2ban зазвичай визначається двома параметрами:

- logtarget – задає напрямок потоку для виведення інформації про роботу Fail2ban. Даний параметр має одне із значень: STDOUT, STDERR, SYSLOG або ім'я файлу. За замовчуванням (якщо цей параметр не визначений) йому присвоюється ім'я файлу /var/log/fail2ban.log.
- loglevel – визначає подробиці виведеної інформації. Можливі значення: ERROR (лише інформація про помилки), WARN (інформація про помилки і попереджувальні повідомлення), INFO (повна інформація про роботу, стоїть за замовчуванням), DEBUG (докладний висновок, опис усіх дій, станів, помилок, необхідних для налагодження підсистеми).

Ще один параметр, що визначає функціональність Fail2ban – socket, він задає ім'я файлу, використовуваного для обміну інформацією між клієнтом та сервером. За замовчуванням даному параметру присвоюється ім'я файлу /var/run/fail2ban/fail2ban.sock.

3.2 Встановлення та налаштування Asterisk і Fail2ban

Перед встановленням Asterisk і Fail2ban нам потрібно визначити з якою linux-подібною операційною системою ми будемо працювати. Я буду використовувати Ubuntu останньої стабільної версії, яка є на даний момент (18.04). Після того як ми встановили ОС приступимо до встановлення Asterisk.

Для початку оновимо індекс пакетів APT, це база даних доступних пакетів з репозиторіїв, визначених у файлі /etc/apt/sources.list і каталозі

/etc/apt/sources.list.d. Для оновлення локального індексу пакетів до останніх змін в репозиторіях виконаємо команду:

- *apt-get update*

Asterisk встановлюємо з DAHDI (Digium Asterisk Hardware Device Interface – забезпечує інтерфейсний рівень між сервером Asterisk з одного боку, і драйвером інтерфейсу Daahdi плюс програмні ехоподавлевачі з іншого, драйвера інтерфейсів дозволяють використовувати апаратні засоби (карти) для з'єднання Asterisk з аналоговими або цифровими телефонними мережами) і LibPRI (бібліотека для роботи з потоковими TDM-інтерфейсами). Спочатку необхідно зібрати DAHDI, потім LibPRI, і тільки потім – Asterisk.

Встановлюємо пакети, необхідні для коректної збірки DAHDI і LibPRI за допомогою команди:

- *apt install make gcc*

Після того як ми встановили усі необхідні пакети завантажуємо вихідний код DAHDI, розпаковуємо його і переходимо в розпакований каталог:

- *wget https://downloads.asterisk.org/pub/telephony/dahdi-linux-complete/dahdi-linux-complete-current.tar.gz*

- *tar -xvf dahdi-linux-complete-current.tar.gz*

- *cd dahdi-linux-complete-**

Далі нам потрібно зібрати пакет і встановити його, для цього виконаємо наступні команди:

- *make*

- *make install*

- *make config*

Далі завантажуємо вихідний код LibPRI, розпаковуємо його, переходимо в розпакований каталог і також встановлюємо:

- *wget https://downloads.asterisk.org/pub/telephony/libpri/libpri-current.tar.gz*

- *tar -xvf libpri-current.tar.gz*

- *cd libpri-**

- *make*

- *make install*

Тепер можемо приступити до встановлення Asterisk, також буде встановлено останню версію, яка існує на даний момент (16.3). Завантажуємо вихідний код, розпаковуємо архів та переходимо у папку, яка з'явилася після розпакування:

- *tar -xvf asterisk-*.tar. gz*

- *cd asterisk-**

Встановлюємо залежності:

- *./contrib/scripts/install_prereq install*

- *./contrib/scripts/install_prereq install-unpackaged*

Встановлюємо бібліотеки для роботи із mp3:

- *./contrib/scripts/get_mp3_source.sh*

Запускаємо конфігурацію пакета перед компіляцією:

- *./configure*

Після виконання успішного команди *./configure* ми побачимо логотип Asterisk (рисунок 3.1).

Рисунок 3.1 – Логотип Asterisk після виконання команди *./configure*

Тепер запускаємо вибір компонентів пакета, для більшості випадків настройки можна залишити за замовчуванням:

- *make menuselect*

Тепер ми запускаємо найголовнішу частину компіляції пакета – збірку:

- *make*

Після чого встановлюємо Asterisk, створюємо конфігураційні файли та встановлюємо скрипт для автозапуску:

- *make install*

- *make samples*

- *make config*

Також налаштуємо iptables. Iptables – це брандмауер, який захищає комп'ютер від несанкціонованих підключень, як від вхідних підключень ззовні, так і несанкціонованих підключень з самого комп'ютера. Для asterisk iptables дає можливість відключити підмережі, з яких не повинно бути підключений до asterisk, а також iptables разом із fail2ban може закрити сервер від підбору паролів до сервера. Для роботи по SIP протоколу (порт UDP 5060) виконаємо команду:

- *iptables -A INPUT -p udp -m udp --dport 5060 -j ACCEPT*

Тепер можна перейти до встановлення fail2ban. Для того щоб встановити fail2ban виконаємо команду:

- *apt install fail2ban*

Для того, щоб встановлене програмне забезпечення працювало належним чином, необхідно вносити поправки у конфігураційний файл /etc/fail2ban/jail.conf. Але не рекомендують редагувати його безпосередньо, щоб уникнути ускладнень при роботі з сервером. Тому створимо локальну копію даного файлу за допомогою команди:

- *cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local*

Далі виконуємо редагування тільки файлу jail.local. Він буде підключений системою автоматично і має пріоритет вищий при виконанні ніж jail.conf. Відкриваємо його за допомогою команди *nano* та в секції [DEFAULT] налаштуємо параметри bantime (час в секундах, протягом якого IP буде заблоковано), findtime (визначає проміжок часу в секундах, протягом якого

програмою буде визначатися наявність підозрілої активності) та maxretry (максимальна кількість неуспішних спроб отримання доступу до сервера, якщо перевищено задану кількість то IP потрапляє в бан). На рисунку 3.2 зображено мої налаштування.

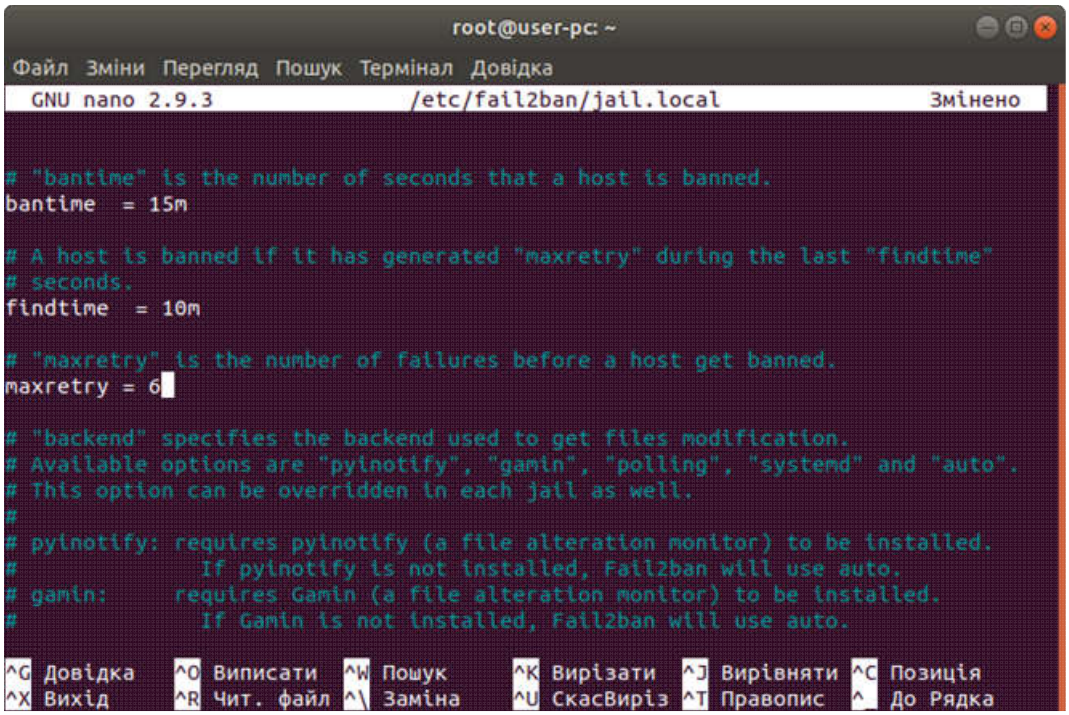

Рисунок 3.2 – Налаштування параметрів bantime, findtime та maxretry

Тепер переходимо до налаштувань правил фільтрації. Необхідно створити фільтр, який буде витягувати із загального потоку повідомлень Asterisk потенційно небезпечні (невірний логін/пароль, спроба входу від несанкціонованого IP адреси, і т.д.). При цьому нам необхідно не тільки виявляти такі потенційно небезпечні події, а й виокремлювати звідти IP адреса. Правила фільтрації можна прописати в файлі /etc/fail2ban/filter.d/asterisk.conf (рисунок 3.3).

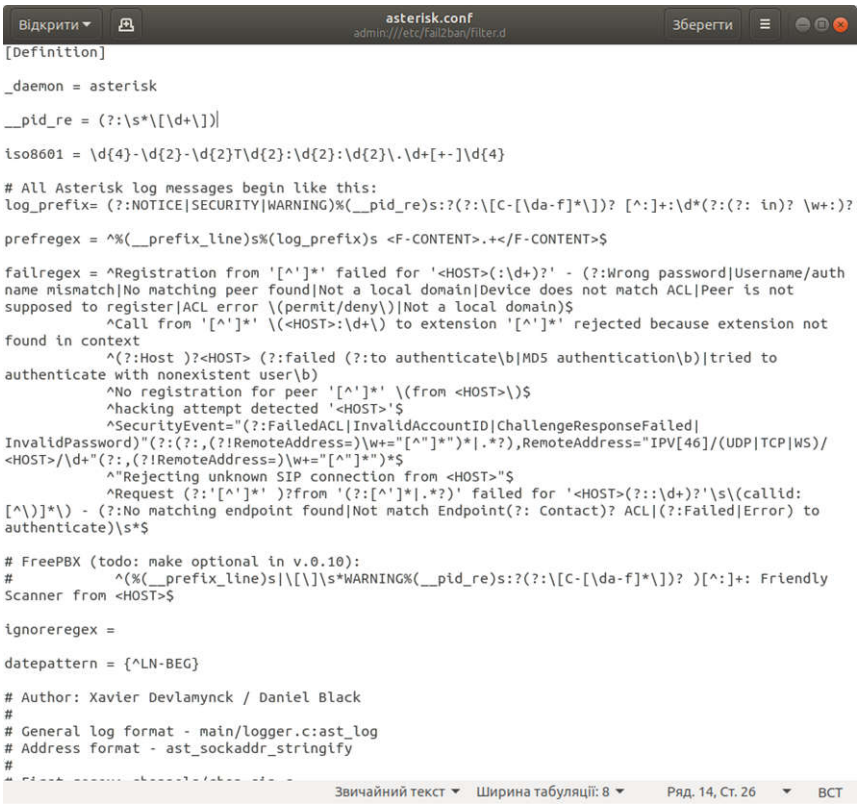

Рисунок 3.3 – Файл /etc/fail2ban/filter.d/asterisk.conf

Далі нам необхідно створити опис так званих "ізоляторів" (jails) для fail2ban, тобто "прив'язати" наші фільтри до fail2ban. Для цього відкриваємо файл /etc/fail2ban/jail.local та прописуємо там два правила (дивитися додаток А) (рисунок 3.4).

```
[asterisk-iptables]
enabled  = true
# фільтр, яким буде користуватися правило, називається asterisk
# (назва фільтру - це ім'я файлу в каталозі /etc/fail2ban/filter.d):
filter   = asterisk
# до якого файлу (логам астеріска) застосовувати фільтр для пошуку потенційно небезпечних подій:
logpath  = /var/log/asterisk/messages
# кількість потенційно небезпечних подій, знайдених фільтром, для спрацьовування дії:
maxretry = 3
# на який період часу (в секундах) застосовувати дію action:
bantime = 86400
# за який період часу (в секундах) шукати в logpath потенційно небезпечні події:
findtime = 3600
# що робити, якщо фільтр виявив атаку (за період findtime секунд в логах logpath виявлено
# Maxretry потенційно небезпечних дій з однієї IP адреси) - блокуємо всі порти
# Для цього IP і посилаємо лист для root:
action   = iptables-allports[name=ASTERISK, protocol=all]
           sendmail-whois[name=ASTERISK, dest=root, sender=fail2ban@asterisk]
# список IP адрес / підмереж, для яких всі потенційно небезпечні події ігноруються:
ignoreip = 127.0.0.1/8
```

Рисунок 3.4 – Перше правило

На рисунку 3.5 зображено друге правило, яке буд працювати якщо включено ведення логів security.

```
# Налаштовуємо ізолятори fail2ban для подій безпеки asterisk:
[asterisk-security-iptables]
# Правило включено:
enabled = true
# Фільтр, яким буде користуватися правило, називається asterisk-security
# (Назва фільтру - це ім'я файлу в каталозі /etc/fail2ban/filter.d):
filter = asterisk-security
# До якого файлу (логам астеріска) застосовувати фільтр для пошуку потенційно небезпечних подій:
logpath = /var/log/asterisk/security
# Кількість потенційно небезпечних подій, знайдених фільтром, для спрацьовування дії:
maxretry = 3
# На який період часу (в секундах) застосовувати дію action:
bantime = 86400
# За який період часу (в секундах) шукати в logpath потенційно небезпечні події:
findtime = 3600
# Що робити, якщо фільтр виявив атаку (за період findtime секунд в логах logpath виявлено
# Maxretry потенційно небезпечних дій з однієї IP адреси) - блокуємо всі порти
# Для цього IP і посилаємо лист для root:
action   = iptables-allports[name=ASTERISK-security, protocol=all]
           sendmail-whois[name=ASTERISK-security, dest=root, sender=fail2ban@asterisk]
# Список IP адрес / підмереж, для яких всі потенційно небезпечні події ігноруються:
ignoreip = 127.0.0.1/8
```

Рисунок 3.5 – Друге правило

Тепер потрібно перезавантажити fail2ban, для цього виконаємо команду:

- *fail2ban-client reload*

Далі налаштуємо SIP клієнта, для того щоб мати змогу перевірити роботу fail2ban. Для цього відкриваємо файл /etc/asterisk/sip.conf, та прописуємо там наступне (рисунок 3.6).

```
[1001]
type=friend
regexten=1001
secret=1234
context=outcaling
host=dynamic
callerid="1001" <1001>
disallow=all
allow=alaw
allow=ulaw
language=ru
callgroup=1
pickupgroup=1
qualify=yes
canreinvite=yes
call-limit=4
nat=no
```

Рисунок 3.6 – Додаємо Sip користувача

Створюємо правило обробки виклику, у файлі extensions.conf (рисунок 3.7).

```
[outcaling]
exten => _XXXX,1,Dial(SIP/${EXTEN},,m)
```

Рисунок 3.7 – Правило обробки виклику

Перезавантажуємо Asterisk.

3.3 Тестування та верифікація

Почнемо із перевірки роботи Asterisk за допомогою команди *systemctl status asterisk* (рисунок 3.8).

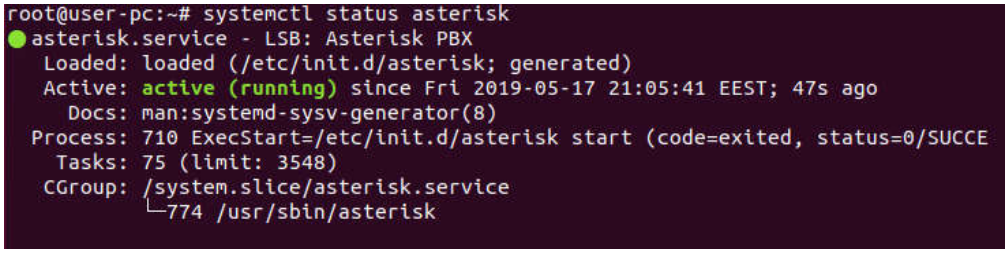

Рисунок 3.8 – Статус Asterisk

Тепер перевіримо чи працюють правила fail2ban. Для перевірки, що fail2ban запущений успішно і правило завантажено, виконуємо команди:

- *fail2ban-client status asterisk-iptables*
- *fail2ban-client status asterisk-security-iptables*

На рисунках 3.9 зображено перевірку правила asterisk-iptables.

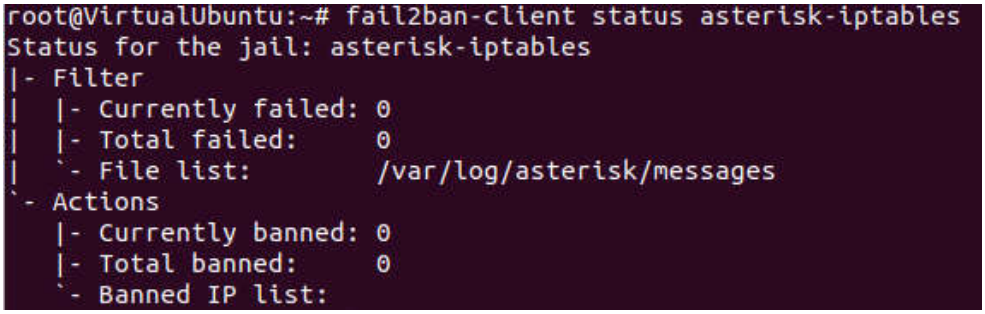

Рисунок 3.9 – Результат виконання команди *fail2ban-client status asterisk-iptables*

На рисунку 3.10 зображено перевірку правила asterisk-security-iptables.

```
root@VirtualUbuntu:~# fail2ban-client status asterisk-security-iptables
Status for the jail: asterisk-security-iptables
|- Filter
|   |- Currently failed: 0
|   |- Total failed:     0
|   `- File list:        /var/log/asterisk/security
`- Actions
    |- Currently banned: 0
    |- Total banned:     0
    `- Banned IP list:
```

Рисунок 3.10 – Результат виконання команди *fail2ban-client status asterisk-security-iptables*

Використовуючи безкоштовний софтфон X-Lite спробуємо підключитися до Asterisk використовуючи невірний пароль до користувача «1001». Вводимо необхідні дані для того щоб підключитися до Asterisk (рисунок 3.11).

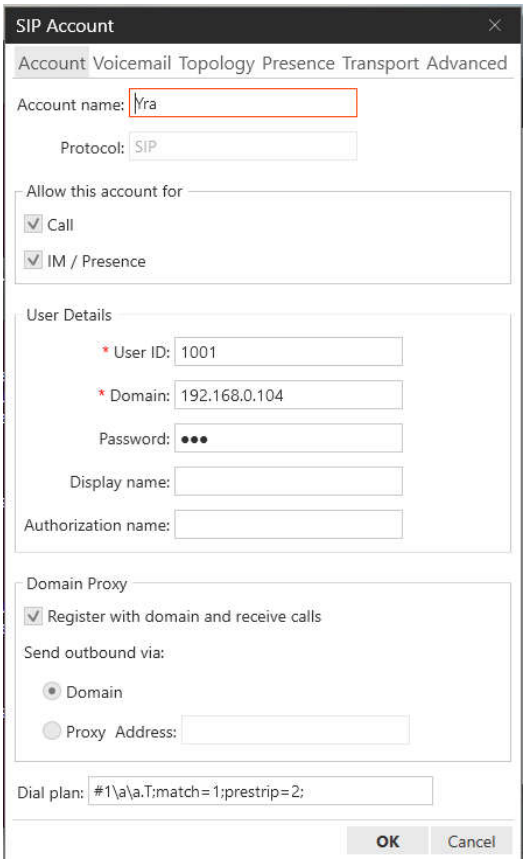

Рисунок 3.11 – Підключаємося до Asterisk

Після трьох спроб введення невірного паролю на сервері Asterisk у нас виводить такі сповіщення (рисунок 3.12).

Рисунок 3.12 – Сповіщення на сервері Asterisk

Для того щоб перевірити чи спрацював fail2ban ще раз перевіряємо правила asterisk-iptables та asterisk-security-iptables (рисунок 3.13 та 3.14).

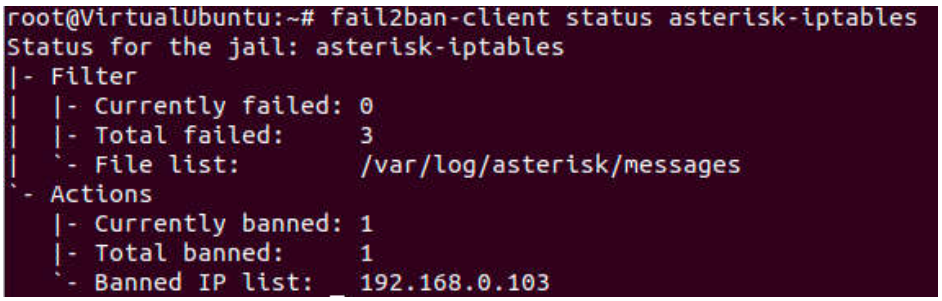

Рисунок 3.13 – Перевірка asterisk-iptables

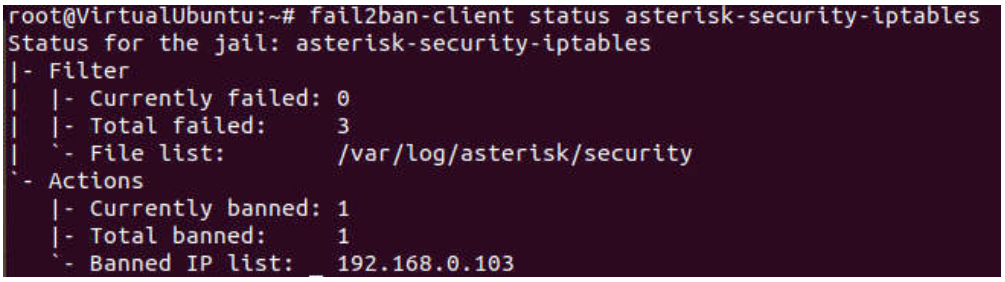

Рисунок 3.14 – Перевірка asterisk-security-iptables

Як бачимо з вище наведених рисунків fail2ban заблокував IP після трьох спроб введення невірного паролю.

4 ТЕХНІКО-ЕКОНОМІЧНИЙ РОЗДІЛ

В даному розділі дипломного проекту проводиться економічне обґрунтування доцільності розробки проектного рішення. Зокрема, здійснюється розрахунок витрат на виконання проектного рішення, експлуатаційних витрат, ціни споживання проектного рішення. В заключній частині визначаються показники економічної ефективності нового проектного рішення, обґрунтовуються відповідні висновки.

Розроблене проектне рішення призначене для захисту IP-телефонії.

4.1 Розрахунок витрат на виконання проектного рішення

Витрати на виконання проектного рішення розраховуються шляхом складання калькуляції кошторисної вартості за наступними статтями:

- витрати на оплату праці;
- відрахування на соціальні заходи;
- матеріальні витрати;
- витрати на використання комп’ютерної техніки;
- витрати на використання спецобладнання для наукових (експериментальних) робіт;
- накладні витрати;
- інші витрати.

Розрахунок витрат на оплату праці. Витрати на оплату праці включають заробітну плату (ЗП) всіх категорій працівників, безпосередньо зайнятих на всіх етапах проектування. Розмір ЗП обчислюється на основі трудоємності

відповідних робіт у людино-годинах та середньої ЗП відповідних категорій працівників.

У розробці проектного рішення задіяні наступні спеціалісти - розробники, а саме: керівник проекту (К); студент-дипломант (С); консультант техніко-економічного розділу (КТЕО).

Таблиця 4.1 – Вихідні дані для розрахунку витрат на оплату праці

№ п/п	Посада виконавців	Місячний оклад (стипендія), *грн*.
1	Керівник ДП, доцент	4920,00
2	Консультант техніко-економічного розділу, доцент	6086,00
3	Студент	1300,00

Витрати на оплату праці розробників проекту визначаються за формулою:

$$B_{ОП} = \sum_{i=1}^{N} \sum_{j=1}^{M} n_{ij} \cdot t_{ij} \cdot C_{ij} \,, \qquad (4.1)$$

де n_{ij} – чисельність розробників *i*-ої спеціальності *j*-го тарифного розряду, *осіб*; t_{ij} – затрачений час на розробку проекту співробітником *i*-ої спеціальності *j*-го тарифного розряду, *год*; C_{ij} – годинна ставка працівника *i*-ої спеціальності *j*-го тарифного розряду, *грн*.,

Середньо годинна ставка працівника може бути розрахована за формулою:

$$C_{ij} = \frac{C_{ij}^{0}(1+h)}{РЧ_{i}}, \qquad (4.2)$$

де C_{ij} – основна місячна заробітна плата розробника *i*-ої спеціальності *j*-го тарифного розряду, *грн*.; *h* – коефіцієнт, що визначає розмір додаткової заробітної плати (при умові наявності доплат); $РЧ_i$ - місячний фонд робочого часу працівника *i*-ої спеціальності *j*-го тарифного розряду, *год.* (приймаємо *168 год.*). Результати розрахунку записують до таблиці 4.2.

Таблиця 4.2 – Розрахунок витрат на оплату праці

№ п/п	Посада виконавців	Час розробки, *год*	Погодинна заробітна плата, *грн/год.*	Витрати на розробку, *грн*
1	Керівник ДП, доцент	16	61,5	984,00
2	Консультант техніко-економічного розділу, доцент	2	76	152,00
3	Студент	130	8	1040,00
Разом				2176

Відрахування на спеціальні державні фонди. Величну відрахувань у спеціальні державні фонди визначають у відсотковому співвідношенні від суми основної та додаткової заробітних плат. Згідно діючого нормативного законодавства сума відрахувань у спеціальні державні фонди складає 20,5% від суми заробітної плати:

$$B_{\Phi} = \frac{20,5}{100} \cdot 2176 = 446.00 \ \ грн.$$

Розрахунок витрат на матеріали та комплектуючі. У таблиці 4.3 наведений перелік купованих виробів і розраховані витрати на них.

Таблиця 4.3 – Розрахунок витрат на матеріали та комплектуючі

№ п/п	Найменування купованих виробів	Одиниця виміру	Ціна, *грн*	Кількість купованих виробів	Сума, *грн*	Транспортні витрати (10% від суми)	Загальна сума, *грн*
1	Папір (формат А4)	уп	80,00	2	160,00	16,00	176,00
2	Ручка кулькова	шт	6,00	1	6,00	0,60	6,60

Продовження таблиці 4.3

3	Олівець простий	шт	4,00	2	3,00	0,30	3,30
4	Зошит, 48 арк	шт	10,00	1	10,00	1,00	11,00
5	Тонер для принтера	уп	50,00	1	50,00	5,00	55,00
Разом							251,90

Витрати на використання комп'ютерної техніки Витрати на використання комп'ютерної техніки включають витрати на амортизацію комп'ютерної техніки, витрати на користування програмним забезпеченням, витрати на електроенергію, що споживається комп'ютером. За даними обчислювального центру ТНЕУ для комп'ютера типу IBM вартість години роботи становить 5,2 грн. Середній щоденний час роботи на комп'ютері – 2 години. Розрахунок витрат на використання комп'ютерної техніки приведений в таблиці 4.4.

Таблиця 4.4 – Розрахунок витрат на використання комп'ютерної техніки

№ п/п	Назва етапів робіт, при виконанні яких використовується комп'ютер	Час використання комп'ютера, *год.*	Витрати на використання комп'ютера *грн.*
1	Проведення досліджень та оформлення їх результатів	87	452,40
2	Оформлення техніко-економічного розділу	3	15,60
4	Оформлення ДП	10	52,00
Разом		100	520,00

Накладні витрати проектних організацій включають три групи видатків: витрати на управління, загальногосподарські витрати, невиробничі витрати.

Вони розраховуються за встановленими відсотками до витрат на оплату праці. Середньостатистичний відсоток накладних витрат в організації складає 150%.

$$H = 1{,}5 \cdot 2176 = 3264{,}00 \ (грн.)$$

Інші витрати є витратами, які не враховані в попередніх статтях. Вони складають 10% від заробітної плати:

$$I = 2176 \cdot 0{,}1 = 217{,}60 \ (грн.)$$

На основі отриманих даних складається калькуляція планової собівартості проектного рішення і зводиться до таблиці 4.6.

Таблиця 4.6 – Кошторис витрат

№ п/п	Найменування елементів витрат	Сума витрат, *грн*.
1	Витрати на оплату праці	2176,00
2	Відрахування у спеціальні державні фонди	446,00
3	Витрати на куповані вироби	207,90
4	Витрати на використання комп'ютерної техніки	450,00
5	Накладні витрати	3264,00
6	Інші витрати	217,60
Разом		6761,50

4.2 Розрахунок договірної ціни та прибутку

Договірна ціна встановлюється по домовленості між замовником та виконавцем і попередньо розраховується за формулою:

$$Ц_Д = C \cdot (1 + p),$$

де *Ц* – договірна ціна, *С* – собівартість проектного рішення, *р* – рівень рентабельності витрат.

Тоді очікуваний прибуток від реалізації проекту розраховують за формулою:

$$П = Ц_Д - С .$$

Для даного дипломного проекту рівень рентабельності приймемо 0,3, тоді ціна дорівнюватиме:

$$Ц_Д = 6761{,}50 \cdot (1 + 0{,}3) = 8789.95 \ (грн.).$$

Прибуток дорівнюватиме:

$$П = 8789{,}95 - 6761{,}50 = 2028{,}45 \ (грн.).$$

4.3 Оцінка результативності проектного рішення

Оцінка наукової та науково-технічної ефективності проводиться за допомогою коефіцієнта результативності, який обчислюються за наступною формулою:

$$k_{р.} = \sum_{i=1}^{m} k_{зн.,i} \cdot k_{д.,i} \ ,$$

де $k_{зн.,i}$ – коефіцієнт значимості i-го фактору, використовуваного для оцінки; $k_{д.,i}$ – коефіцієнт досягнутого рівня i-го фактору; m – кількість факторів результативності проекту.

При оцінці результативності використовуються різні фактори, які впливають на її кількісну оцінку. В якості факторів при оцінюванні можуть бути прийняті новизна отриманих чи прогнозованих результатів, глибина наукового опрацювання, ступінь ймовірності успіху (при незавершеності роботи), перспективність використання результатів, масштаб можливої реалізації результатів, завершеність отриманих результатів.

По кожному із факторів експертним шляхом встановлюється числове значення коефіцієнта вагомості. При цьому сума цих коефіцієнтів повинна бути рівною 1. Коефіцієнт досягнутого рівня фактору також встановлюється експертним шляхом, а його числове значення визначається в межах від 0 до 1. Чим ближчі значення до 1, тим більша результативність НДР, яка проводиться.

З таблиці 4.7 слід вибрати фактори, які характеризують результативність даного проекту .

Таблиця 4.7 – Характеристика факторів результативності проектного рішення

Фактор результативності	Коефіцієнт вагомості фактора	Оцінка фактора	Характеристика фактора	Коефіцієнт досягнутого рівня
Новизна отриманих або передбачуваних результатів	0,35	Висока	Отримані принципово нові результати, невідомі раніше науці, створена нова теорія, відкрита нова закономірність	1,0
		Середня	Встановлені деякі загальні закономірності, методи, способи, що дозволяють створити принципово нові види техніки	0,7
		Недостатня	Позитивне рішення поставлених завдань на основі простих узагальнень, аналіз зв'язків між фактами, поширення відомих принципів на нові об’єкти	0,5
		Тривіальна	Опис окремих елементарних фактів, передача і поширення раніше отриманих результатів, реферативні огляди	0,2

Продовження таблиці 4.7

Глибина наукового опрацювання	0,25	Висока	Виконані складні теоретичні розрахунки, результати перевірені на великій кількості експериментальних даних	1,0
		Середня	Складність теоретичних розрахунків невисока, результати перевірені на обмеженій кількості експериментальних даних	0,6
		Недостатня	Теоретичні розрахунки прості, експериментальна перевірка не проводилася	0,3
Перспектив-ність використання результатів	0,20	Першоряд-на важливість	Результати можуть бути використані в багатьох наукових напрямах, мають значення для розвитку зв'язаних наук	1,0
		Важлива	Результати будуть використані в конкретному науковому напрямі при розробці нових технічних рішень, спрямованих на істотне підвищення продуктивності громадської праці в народному господарстві	0,8

Продовження таблиці 4.7

		Корисна	Результати будуть використані при проведенні наступних НДР, при розробці нових технічних рішень у конкретній галузі народного господарства	0,5
Масштаб можливої реалізації результатів	0,2	Народно-господар-ський	Час реалізації, років: до 3 » 5 » 10 понад 10	 1,0 0,8 0,6 0,4
		Галузевий	Час реалізації, років: до 3 » 5 » 10 понад 10	 1,0 0,7 0,5 0,3
		Окремі організації і підприєм-ства	Час реалізації, років: до 3 » 5 » 10 понад 10	 1,0 0,4 0,2 0,1

Для оцінки результативності даного проекту використані фактори, наведені в таблиці 4.8.

Таблиця 4.8 – Характеристики факторів та ознак результативності проектного рішення

Фактор результативності	Коефіцієнт значимості фактору	Якість фактору	Характеристика фактору	Коефіцієнт досягнутого рівня
Перспектив-ність використання результатів	0,2	Корисна	Результати будуть використані при проведенні наступних НДР, при розробці нових технічних рішень у конкретній галузі народного господарства	0,5
Масштаб можливої реалізації результатів	0,2	Окремі організації і підприєм-ства	Час реалізації: до трьох років	1,0

Згідно із наведеними даними розраховуємо коефіцієнт результативності:

$$k_{p.} = 0.35 \cdot 0.5 + 0.2 \cdot 1 = 0.375 .$$

Як видно з розрахунків, отримано достатній коефіцієнт оцінювання проектного рішення.

В даному розділі проведено розрахунки витрат на виконання проектного рішення. Також обчислені витрати на заробітну плату, єдиний соціальний внесок, витрати на куповані вироби, накладні та інші витрати. Як можна побачити із розрахунків, основними є накладні витрати та витрати на оплату праці. Оскільки коефіцієнт результативності 0,375 та терміні окупності до трьох років дане проектне рішення є доцільним та економічно вигідним.

ВИСНОВКИ

Розглянуто задачу налаштування захисту IP-телефонії. Проаналізовано розвиток IP телефонії, технології АТС, вразливі місця IP-телефонії та можливі варіанти захисту IP-телефонії. Розроблено та реалізовано налаштування захисту IP-АТС Asterisk. Також було проаналізовано можливі безкоштовні альтернативи Asterisk такі як Yate та FreeSWITCH, в результаті було визначено що у Yate та FreeSWITCH функціонал менший чим у Asterisk.

Захист забезпечується за допомогою внесення корективів у конфігураційні файли Asterisk та Linux, встановленням програми захисту серверів від атак – Fail2Ban разом із iptables.

Для налаштування захисту IP-АТС було використано відкриту комунікаційну платформу Asterisk на операційній системі Linux (Ubuntu).

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Електронна енциклопедія Вікіпедія: IP-телефония [Електронний ресурс] Режим доступу: https://ru.wikipedia.org/wiki/IP-%D1%82%D0%B5%D0%BB%D0%B5%D1%84%D0%BE%D0%BD%D0%B8%D1%8F.

2. Електронна енциклопедія Вікіпедія: IP-АТС [Електронний ресурс] Режим доступу: https://ru.wikipedia.org/wiki/IP-%D0%90%D0%A2%D0%A1.

3. Електронна енциклопедія Вікіпедія: FreePBX [Електронний ресурс] Режим доступу: https://ru.wikipedia.org/wiki/FreePBX.

4. Електронна енциклопедія Вікіпедія: Asterisk [Електронний ресурс] Режим доступу: https://ru.wikipedia.org/wiki/Asterisk.

5. Електронна енциклопедія Вікіпедія: FreeSWITCH [Електронний ресурс] Режим доступу: https://ru.wikipedia.org/wiki/FreeSWITCH.

6. Електронна енциклопедія Вікіпедія: Yate [Електронний ресурс] Режим доступу: https://ru.wikipedia.org/wiki/Yate.

7. Олифер В.Г. Компьютерные сети. Принципы, технологии, протоколы. 4 изд. / В.Г. Олифер, Н.А. Олифер – СПб: Питер, 2010. – 944 с.

8. Яковина В.С. Основи безпеки комп'ютерних мереж: Навчальний посібник / В.С. Яковина. – Львів: НВФ "Українські технології", 2008. – 396 с.

9. Bryant R. Asterisk™: The Definitive Guide [Електронний ресурс] / R. Bryant, J. Meggelen, L. Madsen. – 2011. – Режим доступу до ресурсу: http://www.asteriskdocs.org/en/3rd_Edition/asterisk-book-html-chunk/index.html.

10. Електронна енциклопедія Вікіпедія: Fail2ban [Електронний ресурс] Режим доступу: https://uk.wikipedia.org/wiki/Fail2ban

11. Використання IP-телефонії: переваги та можливості [Електронний ресурс] Режим доступу: http://www.klaster-plus.ua/ua/stati-i-obzory/preimushchestva-i-vidy-ip-telefonii/.

12. Електронна енциклопедія Вікіпедія: Linux [Електронний ресурс] Режим доступу: https://uk.wikipedia.org/wiki/Linux.

13. Шоттс У. Командная строка Linux. Полное руководство – Питер, 2017 – 480 с.

14. Методичні рекомендації до виконання дипломного проекту з освітньо-кваліфікаційного рівня “Бакалавр” напряму підготовки 6.050102 «Комп’ютерна інженерія» фахового спрямування «Комп’ютерні системи та мережі» / О.М. Березький, Л.О.Дубчак, Р.Б. Трембач, Г.М. Мельник, Ю.М. Батько, С.В. Івасьєв / Під ред. О.М. Березького. - Тернопіль: ТНЕУ, 2013.–65с.

15. Методичні вказівки до написання техніко-економічного розділу дипломних проектів освітньо-кваліфікаційного рівня «бакалавр» напряму підготовки 6.050102 комп’ютерна інженерія / І.Р. Паздрій – Тернопіль: ТАНГ, 2014. – 37 с.

16. Типові вимоги до оформлення дипломних робіт за освітньо-кваліфікаційними рівнями «спеціаліст» і «магістр» / За ред. Г.П. Журавля – Тернопіль: ТНЕУ, 2007. – 32 с.