Вінницький національний технічний університет

Факультет менеджменту та інформаційної безпеки

Кафедра менеджменту та безпеки інформаційних систем

БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

Розробка комплексної системи моніторингу та захисту SIP-телефонії на основі динамічного контролю сесій автентифікації та інтеграції протоколу TLS з криптоалгоритмом RSA

Виконав: студент 4 курсу, гр.1КІТС-21б
спеціальності 125– Кібербезпека
Освітня програма – Кібербезпека
інформаційних технологій та систем
(шифр і назва напряму підготовки, спеціальності)

Фененко В.О.
(прізвище та ініціали)

Керівник: д.ф.(PhD), доц. каф. МБІС
Салієва О.В.
(прізвище та ініціали)
«____» _____ 2025 р.

Рецензент: к.т.н., доц., доцент каф. ОТ
Крупельницький Л.В.
(прізвище та ініціали)
«____» _____ 2025 р.

Допущено до захисту
Голова секції УБ кафедри МБІС
д.т.н., проф. Яремчук Ю.Є.
“____” _____2025р.

Вінниця ВНТУ – 2025

Вінницький національний технічний університет
Факультет менеджменту та інформаційної безпеки
Кафедра менеджменту та безпеки інформаційних систем

Рівень вищої освіти І-й (бакалаврський)
Галузь знань 12 – Інформаційні технології
Спеціальність 125 – Кібербезпека
Освітньо-професійна програма - Кібербезпека інформаційних технологій та систем

ЗАТВЕРДЖУЮ
Голова секції УБ, кафедра МБІС

д.т.н., проф. Яремчук Ю.Є.
" 20 " березня 2025 р.

З А В Д А Н Н Я
на бакалаврську кваліфікаційну роботу студенту
Фененку Владиславу Олександровичу
(прізвище, ім`я, по-батькові)

1. Тема роботи Розробка комплексної системи моніторингу та захисту SIP-телефонії на основі динамічного контролю сесій автентифікації та інтеграції протоколу TLS з криптоалгоритмом RSA
Керівник роботи Саліва О.В. д.ф., доцент каф. МБІС
(прізвище, ім`я, по-батькові, науковий ступінь, вчене звання)
затверджені наказом вищого навчального закладу від " 20 " березня 2025року № 97
2. Термін подання студентом роботи "2" червня 2025 року
3. Вихідні дані до роботи Метою даної роботи є розробка комплексної системи моніторингу та захисту SIP-телефонії на основі динамічного контролю сесій автентифікації та інтеграції протоколу TLS з криптоалгоритмом RSA
4. Зміст текстової частини: У першому розділі дипломної роботи проаналізовано архітектуру SIP-телефонії, основні вразливості протоколу SIP, а також сучасні методи захисту сигналізаційного та медіа-трафіку. Розглянуто принципи безпечної автентифікації, динамічного керування сесіями, а також застосування протоколів TLS і SRTP з криптоалгоритмом RSA для забезпечення конфіденційності та цілісності передаваних даних. Проведено огляд існуючих рішень у сфері VoIP-безпеки та сформульовано задачі дослідження. У другому розділі спроєктовано архітектуру системи захисту SIP-телефонії з використанням Asterisk Realtime, описано логічну структуру бази даних та обґрунтовано вибір технологій для реалізації захисту трафіку. У третьому розділі представлено реалізацію серверної інфраструктури SIP-телефонії з підтримкою захищеного обміну, динамічного управління користувачами, а також веб-інтерфейс для моніторингу подій безпеки.

5. Перелік графічного матеріалу:
У першому розділі бакалаврської кваліфікаційної роботи наявно 1 рисунок, у другому – 2 рисунків, у третьому розділі – 7 рисунків.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Розділ I	Саліва О.В. д.ф., доцент каф. МБІС		
Розділ II	Саліва О.В. д.ф., доцент каф. МБІС		
Розділ III	Саліва О.В. д.ф., доцент каф. МБІС		

7. Дата видачі завдання 20 березня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
1	Визначення напрямку бакалаврської роботи, формулювання теми	20.03.2025	23.03.2025	Виконано
2	Аналіз предметної області обраної теми	24.03.2025	13.04.2025	Виконано
3	Розробка алгоритму роботи	14.04.2025	27.04.2025	Виконано
4	Написання бакалаврської роботи на основі розробленої теми	28.04.2025	25.05.2025	Виконано
5	Передзахист бакалаврської кваліфікаційної роботи	26.05.2025	27.05.2025	Виконано
6	Виправлення, уточнення, корегування бакалаврської дипломної роботи	28.05.2025	08.06.2025	Виконано
7	Захист бакалаврської кваліфікаційної роботи	09.06.2025	10.06.2025	Виконано

Студент _____ Фененко В.О
(підпис) (прізвище та ініціали)

Керівник роботи _____ Ольга Саліва
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Бакалаврська кваліфікаційна робота складається зі 90 сторінок формату А4, на яких наведено 10 рисунків, 3 таблиць, список використаних джерел містить 36 найменувань.

Метою роботи є розробка комплексної системи моніторингу та захисту SIP-телефонії на основі динамічного контролю сесій автентифікації та інтеграції протоколу TLS з криптоалгоритмом RSA.

У загальній частині роботи розглянуто основи SIP-протоколу, актуальні загрози для VoIP-мереж, методи автентифікації та шифрування, а також виконано аналіз існуючих рішень із безпеки IP-телефонії. Обґрунтовано потребу у створенні власної системи з механізмом виявлення аномалій.

У частині проєктування описано архітектуру програмного засобу, структури баз даних, принципи взаємодії компонентів у реальному часі, UML-діаграми, схеми роботи модулів і логіку алгоритмів виявлення загроз. Подано сценарії застосування та механізми реагування на атаки.

У частині реалізації створено PHP-модуль безпеки SIP-телефонії, що аналізує підключення, виявляє підозрілу активність, фіксує її у системі моніторингу подій і блокує джерела загроз. Реалізовано перевірку автентифікації через SIP RealTime, логування сесій, TLS/SRTP з RSA та інтеграцію з панеллю безпеки.

Ключові слова: SIP, VoIP, автентифікація, TLS, RSA, SRTP, контроль сесій.

ABSTRACT

The bachelor's thesis consists of 90 pages of A4 format, which includes 10 figures, 3 tables, and a list of references containing 36 titles.

The purpose of the work is to develop a comprehensive system for monitoring and protecting SIP telephony based on dynamic control of authentication sessions and integration of the TLS protocol with the RSA crypto algorithm.

In the general part of the paper, the basics of the SIP protocol, current threats to VoIP networks, authentication and encryption methods are considered, and an analysis of existing IP telephony security solutions is performed. The need to create an in-house system with an anomaly detection mechanism is substantiated.

The design part describes the architecture of the software, database structures, principles of real-time interaction of components, UML diagrams, module operation diagrams, and the logic of threat detection algorithms. Application scenarios and mechanisms for responding to attacks are presented.

In terms of implementation, a PHP module for SIP telephony security was created that analyzes connections, detects suspicious activity, records it in the event monitoring system, and blocks threat sources. Authentication verification via SIP RealTime, session logging, TLS/SRTP with RSA, and integration with the security panel are implemented.

Keywords: SIP, VoIP, authentication, TLS, RSA, SRTP, session control.

ЗМІСТ

ВСТУП

Актуальність. У сучасному цифровому світі голосова IP-телефонія (VoIP) стала основою комунікаційної інфраструктури у багатьох організаціях. Особливе місце в цій системі займає протокол SIP (Session Initiation Protocol), який використовується для встановлення, модифікації та завершення голосових дзвінків. Проте разом із поширенням SIP-зв'язку зростають і загрози безпеці: атаки типу "man-in-the-middle", перехоплення аудіо трафіку, сесійне викрадення, brute-force атаки на облікові дані користувачів та інші форми вторгнень.

Існуючі рішення з захисту SIP-телефонії зазвичай або недостатньо гнучкі, або не забезпечують належного рівня контролю за сесіями в режимі реального часу. Використання TLS (Transport Layer Security) та SRTP (Secure Real-time Transport Protocol) з криптоалгоритмами, такими як RSA, значно покращує конфіденційність і цілісність переданих даних, але вимагає комплексного підходу до їх інтеграції. Окрім цього, особливо актуальним стає динамічний контроль сесій автентифікації, що дозволяє оперативно виявляти та блокувати несанкціоновану активність.

Метою роботи є розробка комплексної системи моніторингу та захисту SIP-телефонії на основі динамічного контролю сесій автентифікації та інтеграції протоколу TLS з криптоалгоритмом RSA.

Для досягнення мети необхідно виконати такі задачі:

– дослідити архітектуру SIP-телефонії та сучасні загрози її безпеці;

– проаналізувати криптографічні методи захисту SIP-з'єднань;

– розробити модель динамічного контролю автентифікації в режимі реального часу;

– реалізувати захищене з'єднання SIP через TLS з RSA-сертифікатами та шифруванням аудіо через SRTP;

– побудувати систему моніторингу сесій з можливістю управління SIP-акаунтами, використовуючи Asterisk Realtime;

– провести тестування на предмет стійкості до атак та працездатності у робочому середовищі;

– створити UML-діаграми (ER, Use Case, Activity) для відображення структури та взаємодії компонентів;

Об’єкт дослідження – система захищеного голосового зв’язку на основі SIP-протоколу.

Предмет дослідження – процеси автентифікації, шифрування та моніторингу в системах SIP-телефонії.

Новизна роботи – удосконалено метод захисту SIP-телефонії шляхом впровадження:

динамічного управління сесіями автентифікації в режимі реального часу, інтеграції криптографічного протоколу TLS із алгоритмом RSA, шифрування голосового трафіку за допомогою SRTP, створення єдиної моніторингової панелі для виявлення аномальної поведінки в SIP-сесіях.

Практична цінність роботи – розроблено функціональну систему, яка дозволяє виявляти та блокувати підозрілу активність у реальному часі, управляти користувацькими SIP-акаунтами через веб-інтерфейс, забезпечити шифрування та захист SIP-сесій, моніторити та прослуховувати дзвінки (для служб безпеки), зменшити ймовірність витоку інформації через технічні канали.

1 АНАЛІЗ СУЧАСНИХ МЕТОДІВ ЗАХИСТУ ТА ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ SIP-ТЕЛЕФОНІЇ

У даному розділі проаналізовано методи захисту та забезпечення безпеки SIP-телефонії, зокрема механізми шифрування, автентифікації та протоколи захисту даних. Розглянуто особливості застосування симетричного та асиметричного шифрування для захисту голосового трафіку.

Особливу увагу приділено огляду протоколу TLS, який є ключовим для забезпечення безпеки SIP-зв’язку, а також методам інтеграції криптографічних алгоритмів у SIP-системи. Окрім цього, проаналізовано вибір технологій та засобів реалізації, що забезпечують максимальну ефективність і захищеність розроблюваної системи.

1.1 Протокол SIP – принципи функціонування, перспективи та обмеження

Протокол Session Initiation Protocol (SIP) – це сигнальний протокол, який використовується для ініціювання, підтримки, модифікації та завершення сеансів зв'язку в реальному часі між пристроями Інтернет-протоколу (IP) [1]. SIP забезпечує передачу голосу, повідомлень, відео та інших комунікаційних додатків і послуг між двома або більше кінцевими точками в IP-мережах. Його використання поширене у сфері VoIP (голос через IP), відеоконференцій, інтерактивних ігор, обміну повідомленнями та інших сервісів[2]. Завдяки своїй модульній архітектурі SIP демонструє гнучкість, що дозволяє адаптувати протокол під різні потреби, а також підтримкою різних типів медіа та можливістю інтеграції з іншими інтернет-протоколами, такими як HTTP або SMTP [3].

SIP використовує модель клієнт-сервер, де сервери (реєстрації, проксі, перенаправлення) забезпечують маршрутизацію запитів, автентифікацію та обслуговування клієнтів, а кінцеві пристрої (IP-телефони, софтфони тощо) виступають ініціаторами викликів. Основні SIP-запити – INVITE, REGISTER, ACK, BYE, CANCEL, MESSAGE – дозволяють керувати сигналізацією та встановленням мультимедійних сесій [4] .

Основною задачею SIP є керування сигналізацією – процесом ініціації, зміни та завершення сеансів зв’язку. У цьому процесі SIP працює за принципом текстових запитів і відповідей, що дозволяє легко аналізувати його роботу та налагоджувати інтеграцію з іншими мережевими технологіями [5].

Використовуючи архітектуру клієнт-сервер, протокол SIP керує програмними системами для координації, ініціювання та завершення медіа-сесій в режимі реального часу, включаючи VoIP-дзвінки, відеоконференції, обмін повідомленнями та інші види [6].

Робота SIP спирається на клієнт-серверну архітектуру, де клієнти є кінцевими точками (телефони, комп’ютери), а сервери виконують функції маршрутизації, автентифікації та перенаправлення запитів. Сервери реєстрації зберігають інформацію про місцезнаходження клієнтів, що особливо важливо для мобільних користувачів.

SIP широко використовується [7]:

– у великомасштабних корпоративних мережах для заміни традиційної телефонії;

– у контакт-центрах, де важливими є контроль сесій, запис викликів та захист персональних даних клієнтів;

– у медичних установах, енергетичних компаніях, об'єктах критичної інфраструктури – де витік або підміна голосового трафіку може створити загрозу для життя, здоров’я чи національної безпеки.

Комунікація між клієнтами зазвичай включає встановлення сесії через сервер, але протокол дозволяє і прямі з’єднання між кінцевими точками після встановлення з'єднання.

Протокол SIP надає численні функціональні можливості, які роблять його популярним вибором для мультимедійних комунікацій [8]:

1. Універсальність та масштабованість. SIP може працювати як у невеликих мережах компаній, так і в масштабах великих телекомунікаційних операторів, обслуговуючи мільйони користувачів.

2. Підтримка різних типів комунікацій. Крім голосових викликів, SIP підтримує відео, обмін миттєвими повідомленнями, а також інтерактивні сесії, наприклад, для дистанційного навчання чи роботи.

3. Мобільність та сумісність. SIP підтримує користувачів, які змінюють свої мережеві точки доступу, що робить його ефективним у сучасних умовах мобільності.

4. Інтеграція з іншими технологіями. Завдяки текстовому формату, SIP легко інтегрується з HTTP, XML та іншими стандартами, що розширює його можливості для створення складних систем комунікації.

Незважаючи на свої переваги, SIP має низку обмежень, які впливають на його впровадження у сучасних системах [9].

Попри можливість працювати у великих системах, для забезпечення продуктивності на масштабах провайдерів потрібні додаткові рішення, зокрема балансування навантаження та створення високодоступних систем.

Такий формат, з одного боку, спрощує реалізацію та інтеграцію з іншими протоколами, але з іншого – створює низку проблем з безпекою, які в умовах кіберзагроз набувають критичного значення.

Оскільки SIP здебільшого працює поверх транспортних протоколів UDP або TCP, без додаткового шифрування сигнальні дані, такі як логіни, паролі, IP-адреси користувачів, маршрутизаційна інформація, передаються у відкритому вигляді [10]. Це робить можливим здійснення низки атак: перехоплення трафіку, підміна повідомлень, викрадення облікових даних, реєстрація фальшивих пристроїв, маніпуляція маршрутами сигналізації.

Вразливість SIP-протоколу має особливу вагу у контексті його використання в критичних інформаційних середовищах [11]:

– органах державного управління (зокрема, системи внутрішнього зв’язку);

– структурах національної безпеки та оборони;

– медичних установах, де передаються персональні дані пацієнтів;

– банківських системах (наприклад, автоматизовані контакт-центри);

– транспортній інфраструктурі та енергетичному секторі.

У таких умовах перехоплення або порушення роботи SIP-систем може призвести до витоку конфіденційної інформації, зриву критично важливих процесів, фінансових втрат, репутаційних збитків або прямої загрози безпеці персоналу та користувачів.

1.2 Аналіз сучасних загроз для SIP-телефонії

Протокол SIP (Session Initiation Protocol) завдяки своїй гнучкості, відкритості та стандартизації SIP отримав широке поширення в корпоративному та споживчому секторах. Попри численні переваги (масштабованість, підтримка мобільності, інтеграція з іншими протоколами, простота реалізації), SIP має критичні вразливості, які активно експлуатуються в сучасних кіберзагрозах [12]. У відкритому варіанті SIP є вразливим до багатьох класичних атак на мережеві сервіси, що може мати серйозні наслідки в організаційному, фінансовому та репутаційному аспектах. У таблиці 1.1 розглянуто найбільш поширені загрози, їхні механізми реалізації, приклади та відповідність категоріям STRIDE [13].

Таблиця 1.1 – Аналіз потенційних загроз для SIP-телефонії за моделлю STRIDE

№	Тип загрози	Опис механізму	Наслідки	STRIDE	Приклад реалізації
1	Перехоплення RTP-трафіку	RTP (Real-time Transport Protocol), який використовується для передачі голосу, за замовчуванням передає трафік у відкритому вигляді	Витік конфіденційної розмови	Information Disclosure	Sniffing-трафіку за допомогою Wireshark на незахищеному сегменті мережі
2	Brute-force атака на автентифікацію	SIP використовує просту Digest-автентифікацію, легко піддається атакам перебору пароля	Компрометація акаунта, фінансові втрати	Elevation of Privilege	Використання інструменту SIPVicious для підбору паролів SIP-акаунтів

Продовження табл. 1.1

3	Man-in-the-Middle (MitM)	Перехоплення SIP і RTP між клієнтом і сервером із можливістю модифікації	Підміна повідомлень, підслуховування, саботаж	Tampering / Spoofing	Встановлення зловмисного проксі між Asterisk і SIP-клієнтом
4	Spoofing (Caller ID підміна)	Зловмисник надсилає SIP-запит із фальшивим Caller ID	Обман користувача, шахрайство	Spoofing	Надсилання дзвінка від імені служби безпеки банку
5	DoS / DDoS атака на порт 5060	Засипання сервера великою кількістю INVITE або REGISTER-запитів	Відмова сервісу, вивід із ладу	Denial of Service	Використання скриптів для генерації тисяч запитів/сек. на порт 5060 (UDP)

Розглянемо різноманітні типи загроз більш детальніше:

RTP (Real-time Transport Protocol) використовується для передачі аудіо та відео. У типовій SIP-сесії голос передається через протокол RTP без будь-якого шифрування у більшості реалізацій, особливо в невеликих організаціях, цей трафік не шифрується, що робить можливим його перехоплення за допомогою мережевого аналізатора[14]. Це означає, що будь-хто, хто має доступ до проміжного маршрутизатора або комутатора, може записати всю розмову. Уразливість зростає у випадках використання Wi-Fi або незахищених сегментів корпоративної мережі. Це особливо критично для комерційних або державних структур, де передається конфіденційна інформація [15].

Наприклад, зловмисник у внутрішній мережі, підключившись до порту комутатора з функцією port mirroring, записує всі RTP-потоки й зберігає розмови топменеджменту.

Через відсутність обмежень на кількість спроб входу та просту Digest-автентифікацію SIP легко піддається атакам перебору паролів, через відсутність обмежень на кількість запитів. Використовуючи спеціальні утиліти, зловмисник може за короткий час підібрати пароль до SIP-акаунта, отримати доступ до сервісу, ініціювати дзвінки від імені жертви та навіть здійснювати міжнародні дзвінки за рахунок компанії. Це особливо небезпечно в середовищах, де не впроваджено багатофакторну автентифікацію.

Наприклад, автоматизовані атаки із використанням інструментів SIPVicious, Asterisk Scanner, Hydra протягом години отримують доступ до понад 100 акаунтів на сервері без обмеження спроб.

Атаки типу Man-in-the-Middle (MitM) також є серйозною загрозою, особливо в мережах із недостатньо контрольованим доступом. У такому випадку зловмисник може перехопити SIP-сесію, змінити параметри виклику або підмінити кінцеву точку, створюючи фальшиве з'єднання між абонентами. Така атака дозволяє не тільки прослуховувати розмову, але й викрадати ідентифікаційні дані або перенаправляти дзвінки на інші номери. Зазвичай такі атаки стають можливими через відсутність шифрування на рівні сигналізації або неправильну його конфігурацію. Такі атаки особливо небезпечні, якщо SIP не захищений TLS. За відсутності використання TLS (Transport Layer Security), SIP-з'єднання можна легко перехопити або модифікувати [16].

Наприклад, при реєстрації SIP-клієнта без TLS, трафік перехоплюється та редагується, підмінюючи сервер авторизації, що дозволяє викрадати облікові дані. Перехопити розмову між лікарем і пацієнтом або між оператором і військовим підрозділом.

Спуфінг у SIP полягає в тому, що зловмисник маскується під легітимного користувача або сервер, змінюючи SIP URI, IP-адресу чи Caller ID, від імені довірених користувачів.. SIP не передбачає суворої перевірки достовірності це відкриває можливості для фішингу, шахрайства або соціальної інженерії. У SIP Caller ID задається вручну у заголовках SIP-запиту, тому зловмисник може видавати себе за будь-який інший номер. Це використовується в схемах соціальної інженерії, телефонного шахрайства або в атаках на довіру Це створює умови для шахрайських дзвінків, перенаправлення трафіку або соціальної інженерії [17].

Також варто згадати спуфінг SIP-ідентифікаторів – ситуації, коли атакувальник імітує інший SIP-ідентифікатор для введення в оману приймаючої сторони. Це дозволяє здійснювати фішингові атаки, маніпуляції з дзвінками або обходити обмеження доступу до ресурсів.

Наприклад, користувач отримує дзвінок із ідентифікатором “Служба безпеки банку”, під час якого зловмисник вивідує персональні дані або паролі, підроблений Caller ID може використовуватися для обману користувачів і отримання конфіденційної інформації.

Ще одна небезпека – атаки типу DoS (Denial of Service). SIP-сервери, як правило, обробляють велику кількість запитів та мають відкритий порт 5060 (UDP), що робить їх доступними з Інтернету, і за відсутності належних обмежень та фільтрації зловмисник може згенерувати велику кількість INVITE-запитів, що призведе до перевантаження системи, зниження якості зв’язку або повного відмовлення сервісу. У разі застосування ботнету – це призводить до повної недоступності голосового сервісу.

Відсутність механізмів контролю сесій дозволяє здійснювати флудинг реєстраційними запитами, перевантажувати SIP-сервери або реєструвати фальшиві акаунти. Також можлива активація платних дзвінків на дорогі напрямки (Toll Fraud), що веде до фінансових втрат.

Наприклад, у години пік сервер SIP отримує понад 20 тисяч запитів REGISTER від ботів. Ресурси процесора вичерпуються, і система перестає відповідати. DoS-атака на SIP-сервер контакт-центру може повністю вивести з ладу обробку звернень громадян.

Також поширеною проблемою є використання базової автентифікації SIP Digest, яка, не будучи захищеною TLS, дозволяє зловмисникам легко здійснювати прослуховування (eavesdropping), перехоплення облікових даних, номерів абонентів, ID сесій тощо [18]. Наприклад, SIP-автентифікація зазвичай відбувається за допомогою HTTP Digest, що може бути перехоплено в незашифрованому трафіку.

Відомі сценарії, коли зловмисники шляхом автоматизованого перебору атакують SIP-сервери, здійснюючи спроби реєстрації на тисячі облікових записів.

Проблеми з автентичністю та цілісністю повідомлень, відсутність шифрування та захисту від повторного використання запитів роблять SIP уразливим до атак, що підривають довіру до каналу зв’язку.

Уразливості SIP мають особливо критичне значення у сферах, де надійність і конфіденційність є пріоритетними: у великих організаціях, контакт-центрах, банках, державних структурах, медичних установах та об'єктах критичної інфраструктури. У таких середовищах перехоплення або підміна трафіку може призвести не лише до витоку чутливої інформації, але й до реальних фінансових збитків або загроз національній безпеці.

SIP-повідомлення містять метадані, такі як IP-адреси, номери телефонів і логіни, які можуть використовуватися зловмисниками для аналізу мережі або організації більш складних атак.

Крім того, SIP часто використовує базову HTTP-подібну автентифікацію (Digest Authentication), яка не є захищеною від підслуховування у відкритих мережах. Без використання додаткових механізмів шифрування зловмисник може отримати облікові дані користувача шляхом перехоплення або підміни запитів [16].

Основними причинами поширення загроз є:

– відсутність вбудованого шифрування для сигналізації за замовчуванням;

– використання слабких механізмів автентифікації;

– пряме з'єднання між клієнтами та серверами без додаткового захисту;

– низький рівень моніторингу та реагування на аномалії трафіку;

– недоліки конфігурації або відсутність оновлень SIP-серверів.

Ці недоліки створюють можливості для різноманітних атак, які можуть бути реалізовані навіть без складних технічних знань.

Реалізація атак на SIP може призводити до серйозних наслідків, таких як:

– компрометація облікових даних користувачів та серверів;

– значні фінансові втрати через шахрайські виклики;

– порушення конфіденційності голосових чи відеокомунікацій;

– повне виведення з ладу системи IP-телефонії.

Сучасний стан безпеки SIP-телефонії демонструє вразливість протоколу до різних типів загроз, що робить необхідним впровадження комплексного підходу до захисту. Забезпечення надійного шифрування, розширення автентифікаційних механізмів та впровадження систем моніторингу та аномального аналізу є

критично важливими кроками для мінімізації ризиків і забезпечення надійної роботи систем SIP. Для зменшення ризиків необхідно впроваджувати сучасні криптографічні механізми, захищену автентифікацію, сегментацію мережі, а також постійний моніторинг та аудит безпеки.

1.3 Існуючі підходи до захисту SIP-телефонії

SIP (Session Initiation Protocol) став одним із найпоширеніших стандартів для організації мультимедійного зв'язку, забезпечуючи ефективність і гнучкість для корпоративних і приватних користувачів. Однак разом з широким розповсюдженням SIP-протоколу зростає і актуальність питань його захисту. Протокол за замовчуванням не забезпечує шифрування сигнального трафіку, що створює передумови для ряду вразливостей та зробили його ціллю для численних кібератак. Щоб мінімізувати ризики, були розроблені різноманітні методи захисту, які охоплюють шифрування, автентифікацію, контроль доступу, моніторинг трафіку та використання спеціалізованих рішень на кшталт Session Border Controllers (SBC) [19].

У зв'язку з постійним збільшенням витоків інформації та кібератак, уряди країн, міжнародні організації й корпоративні союзи розробили низку обов'язкових вимог до інформаційної безпеки. SIP-телефонія, яка обробляє потенційно чутливу інформацію (наприклад, персональні дані, банківські або медичні записи), повинна бути захищеною відповідно до цих вимог:

GDPR (General Data Protection Regulation) передбачає захист персональних даних, у тому числі переданих голосом, а також вимагає впровадження належних технічних і організаційних заходів безпеки. Якщо через SIP-сервіс передається будь-яка інформація, що стосується ідентифікації особи (наприклад, імена, номери телефонів, записи розмов), компанія зобов'язана гарантувати її захист. Порушення цього принципу призводить до фінансових санкцій, публічної репутаційної шкоди та юридичної відповідальності. У разі витоку таких даних передбачено серйозні штрафи – до 20 млн євро або 4% річного обороту компанії [20].

ISO/IEC 27001 – міжнародний стандарт, що визначає вимоги до систем управління інформаційною безпекою (СУІБ). У межах стандарту передбачено захист комунікацій, управління доступом, криптографічні засоби, а також аудит та моніторинг подій. Відсутність захищеної SIP-телефонії є ознакою невідповідності вимогам стандарту. Для організацій, які прагнуть сертифікувати свою систему управління інформаційною безпекою, обов'язковим є впровадження контролю доступу до комунікаційних систем, захисту каналів зв'язку та криптографічного захисту переданих даних [21].

У деяких країнах запроваджено локальні акти, які зобов'язують захищати канали передавання даних у держустановах та критичних секторах (наприклад, закон «Про критичну інфраструктуру» в Україні, NIST Cybersecurity Framework у США тощо).

У сфері оборони, державного управління, енергетики та охорони здоров'я часто діють галузеві регламенти, які прямо зобов'язують використовувати захищені канали голосового зв'язку.

В умовах гібридної війни, загрози тероризму та кібершпигунства, SIP-телефонія у державних та силових структурах розглядається як критично важлива складова інформаційного простору. Порушення цілісності системи може мати серйозні наслідки, зокрема призвести до дезорганізації управління в умовах надзвичайних ситуацій та зниження ефективності командування. Крім того, існує ризик компрометації конфіденційної інформації під час перемовин, що може створити передумови для саботажу або підміни даних у процесі передачі наказів.

У таких середовищах незахищений SIP-трафік – це потенційне «вікно» для противника, через яке можна здійснювати спостереження, блокування чи навіть повне перехоплення голосового трафіку.

Одним із найефективніших способів забезпечення захисту є впровадження шифрування. TLS (Transport Layer Security) гарантує цілісність і конфіденційність сигналізаційного трафіку, запобігаючи перехопленню даних під час їх передачі. Він також забезпечує автентифікацію сторін, що значно ускладнює атаки типу "людина посередині" (MITM). Поряд із TLS, для захисту медіа-трафіку широко

використовується SRTP (Secure Real-Time Protocol), який шифрує голосові й відео дані. Проте обидва ці протоколи вимагають ретельного налаштування та сумісності між пристроями, що може створювати труднощі при інтеграції в складні мережі [22].

Автентифікація користувачів також є ключовим компонентом захисту SIP-систем. Традиційно використовуваний метод Digest Authentication забезпечує базовий рівень безпеки, передаючи хешовані паролі замість відкритого тексту. Але його ефективність знижується у випадках, коли атакуючі мають доступ до хешів і можуть використовувати атаки перебору. Багатофакторна автентифікація (2FA), яка поєднує використання паролів із одноразовими кодами або біометричними даними, значно підвищує безпеку, але її застосування поки що є обмеженим через складність інтеграції з SIP-протоколом.

Контроль доступу та фільтрація трафіку здійснюються за допомогою брандмауерів SIP та SBC. Ці пристрої не лише блокують підозрілі запити та захищають від DoS/DDoS-атак, але й виконують роль шлюзу між внутрішньою та зовнішньою мережами. Вони також можуть забезпечувати трансляцію адрес і оптимізацію роботи мережі. Однак їх впровадження потребує значних фінансових і технічних ресурсів.

Моніторинг трафіку SIP є важливим для виявлення аномальної активності, такої як повторювані запити або спроби атак на автентифікацію. Використання інструментів аналізу поведінки дозволяє оперативно реагувати на загрози, але для ефективності таких систем потрібні регулярні оновлення правил та автоматизація.

Крім того, захист SIP-протоколу може бути підсилений за рахунок використання IPsec для шифрування всього трафіку на рівні мережевих з’єднань. IPsec забезпечує високу надійність і конфіденційність, проте його складність налаштування та затримки при обробці трафіку можуть створювати перешкоди для масового впровадження [22].

Таким чином, для досягнення максимальної безпеки SIP-телефонії необхідно впроваджувати комплексні підходи, які б включали кілька методів одночасно. Наприклад, інтеграція TLS і SRTP для шифрування, застосування SBC для

контролю доступу та моніторинг трафіку на основі поведінкових моделей можуть забезпечити всебічний захист від сучасних загроз. Це вимагає не лише технічних ресурсів, але й регулярного оновлення систем і навчання персоналу, відповідального за їх експлуатацію.

Розробка інтегрованих рішень, які поєднують новітні криптографічні протоколи, автоматизацію моніторингу та механізми реакції на загрози в режимі реального часу, є ключем до підвищення безпеки SIP-телефонії.

1.4 Порівняльний аналіз системи моніторингу та захисту SIP-телефонії

У сучасних умовах, коли забезпечення захисту SIP-телефонії є критично важливим завданням для підприємств і організацій, вибір відповідної програмно-апаратної платформи стає ключовим чинником. Рішення повинно не тільки забезпечувати базові функції маршрутизації VoIP-дзвінків, а й підтримувати сучасні механізми безпеки: шифрування трафіку, автентифікацію, виявлення та протидію атакам. На ринку існує ряд як комерційних, так і відкритих рішень, що мають різний рівень функціональності, зручності адміністрування та реалізації захисних механізмів, таких як TLS, SRTP, контроль автентифікації та інше. Розглянемо найпоширеніші системи SIP-телефонії з точки зору функціональних можливостей, зручності налаштування та безпекових характеристик у таблиці 1.2.

Таблиця 1.2 – Порівняльна характеристика популярних платформ SIP-телефонії

Платформа	Наявність шифрування (SRTP)	Підтримка TLS	Контроль автентифікації	Гнучкість	Тип ліцензії	Зручність адміністрування
FreePBX	Частково (після налаштування)	Так (необхідно вручну)	Базовий	Середня	Open Source	Висока
3CX	Так	Так	Розширений	Низька	Пропрієтарна	Дуже висока
FusionPBX	Так	Так	Гнучкий	Висока	Open Source	Середня
Asterisk	Так	Так	Повний контроль	Максимальна	Open Source	Низька

FreePBX – це популярна веб-оболонка для керування Asterisk, що надає інтуїтивно зрозумілий інтерфейс адміністрування. Вона значно полегшує створення SIP-акаунтів, налаштування дзвінків, маршрутів та інтеграцію з зовнішніми сервісами. FreePBX ідеально підходить для малого та середнього бізнесу, де важливо швидко налаштувати базову телефонну систему без глибокого занурення в конфігураційні файли [23]. Однак, незважаючи на зручність, FreePBX має низку обмежень у сфері безпеки. Підтримка TLS і SRTP наявна, але не активована за замовчуванням. Це означає, що адміністратор повинен самостійно згенерувати сертифікати, внести відповідні зміни в конфігурації SIP-профілів і протестувати стабільність шифрованих з'єднань. Також система має лише базовий контроль автентифікації, без можливостей гнучкого керування сесіями або двофакторної перевірки[24].

FreePBX не містить вбудованих механізмів активного виявлення атак або автоматичного реагування на підозрілу активність, тому для захисту рекомендовано інтегрувати сторонні засоби моніторингу, наприклад, Fail2Ban.
Переваги: зручність, швидкий запуск, велика спільнота.
Недоліки: потреба у додатковій конфігурації для шифрування, базовий контроль автентифікації.

3CX є комерційною платформою для IP-телефонії з фокусом на простоту, інтеграцію та мінімізацію вимог до технічної підготовки адміністратора та орієнтована на бізнес-сектор. Система доступна як у вигляді хмарного сервісу, так і для локального розгортання. 3CX має сучасний графічний інтерфейс, автоматичне налаштування клієнтів, підтримку мобільних додатків, інтеграцію з CRM-системами та іншими корпоративними сервісами [25].

У сфері безпеки 3CX має перевагу: TLS та SRTP вмикаються за замовчуванням, що суттєво знижує ризик перехоплення трафіку. Проте обмежена гнучкість і закрита архітектура не дозволяють адаптувати систему під нестандартні сценарії, наприклад, специфічну схему контролю автентифікації або нетипову інфраструктуру.

Оскільки 3CX є пропрієтарною системою, її використання пов'язане з ліцензійними обмеженнями, а інтеграція з зовнішніми засобами захисту може бути ускладненою[26].

Переваги: готове рішення з підтримкою безпеки, GUI, супровід. Недоліки: обмежена гнучкість, платна ліцензія, складність в адаптації під нестандартні задачі.

FusionPBX – це повнофункціональна веб-оболонка для FreeSWITCH, орієнтована на багатокористувацьке використання, підтримку мультидоменності, масштабованість, яка є потужною альтернативою Asterisk. Система підтримує SRTP та TLS, має гнучкі механізми маршрутизації дзвінків, створення політик безпеки, а також підтримку ролей користувачів, що робить її зручною для провайдерів послуг VoIP, великих організацій або контакт-центрів [19]. Проте, налаштування безпеки вимагає високої технічної кваліфікації, знань SIP-архітектури та конфігурації FreeSWITCH [27].

Система підтримує всі сучасні засоби захисту: TLS, SRTP, контроль доступу, автентифікацію на основі IP, ACL, журналювання подій, а також має гнучкий механізм створення політик безпеки. Проте адміністрування FusionPBX вимагає високого рівня знань, особливо у сфері SIP-протоколу, NAT traversal, DNS SRV, STUN/TURN.

Через складність налаштування, FusionPBX не рекомендується для швидкого впровадження без залучення досвідчених спеціалістів, але натомість дає найширші можливості з безпеки та гнучкості серед рішень із відкритим кодом.

Переваги: потужна функціональність, підтримка сучасних протоколів захисту. Недоліки: складна конфігурація, висока крива навчання.

Asterisk – це гнучкий SIP-сервер з відкритим кодом, який слугує ядром для багатьох VoIP-систем (у тому числі FreePBX). Його основна перевага – повна гнучкість. Завдяки модульності та широким можливостям конфігурації він дозволяє реалізувати будь-яку логіку обробки дзвінків [28]. Asterisk підтримує TLS, SRTP, реалізацію політик автентифікації, обмеження доступу, однак усе це реалізується вручну через редагування конфігураційних файлів. Asterisk дозволяє

вручну налаштовувати SIP-параметри, маршрути, логіку викликів, скрипти, підключення до баз даних, інтеграцію з іншими сервісами та протоколами [29].

Платформа підтримує TLS, SRTP, а також RealTime-архітектуру, що дозволяє зберігати та змінювати конфігурацію в базі даних у реальному часі. Це відкриває можливості для динамічного керування користувачами, що є критично важливим для безпеки та автоматизації процесів [30].

Проте Asterisk – це інструмент для інженерів: налаштування потребує редагування конфігураційних файлів, знання протоколів SIP і PJSIP, розуміння внутрішньої логіки роботи. У разі помилок у конфігурації – велика ймовірність виникнення вразливостей.

Переваги: максимальна гнучкість, підтримка шифрування, велика база знань. Недоліки: потребує високої технічної підготовки, ручне налаштування.

Існуючі комерційні рішення, такі як FreePBX, 3CX та Cisco Unified Communications, здебільшого забезпечують лише базовий рівень захисту, не охоплюючи потреби в глибокому аналізі подій, виявленні аномалій у реальному часі, динамічному контролі автентифікації чи використанні сучасних криптографічних протоколів, зокрема TLS, SRTP та RSA (табл. 1.2). У зв’язку з цим виникає нагальна потреба у створенні комплексних систем захисту SIP-комунікацій, які б включали багатофакторну автентифікацію, повне шифрування трафіку як сигналізації, так і медіа, контроль та моніторинг сесій у реальному часі, а також можливість автоматичної реакції на підозрілу активність. Сучасні сервіси технічної підтримки, медичних установ, банків, служб екстреного реагування та телекомунікаційних компаній використовують SIP-телефонію як основний засіб взаємодії з клієнтами. Наявність уразливостей у таких системах створює ризики:

– для безпеки життя та здоров’я людей (наприклад, у службах 112 або швидкої допомоги);

– для стабільності банківської інфраструктури (через викрадення голосових одноразових кодів чи записів дзвінків);

– для збереження довіри клієнтів (витік записів, зловмисне підключення до лінії тощо).

На практиці відомі випадки, коли хакери здійснювали викрадення SIP-ідентифікаторів контакт-центрів банків, внаслідок чого проводилися шахрайські операції із клієнтськими рахунками [21].

Попри наявність великої кількості рішень для організації SIP-телефонії, більшість із них мають суттєві обмеження з точки зору безпеки, масштабованості та адаптивності до сучасних викликів у сфері інформаційної безпеки (рис. 1.1).

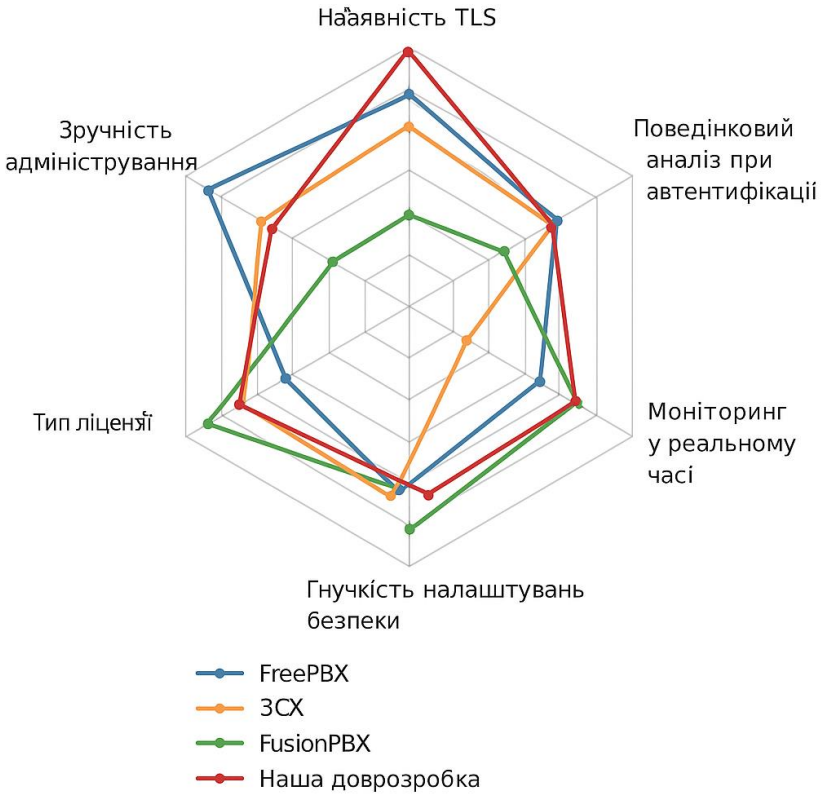


Рисунок 1.1 – Порівняльна характеристика популярних платформ SIP-телефонії

Основні проблеми, які спостерігаються при використанні наявних платформ, можна згрупувати за такими категоріями:

1. Відсутність гнучких політик контролю доступу.

Багато існуючих систем не мають вбудованих механізмів динамічного контролю автентифікації та авторизації користувачів. Як правило, контроль доступу реалізується на базовому рівні – через фіксовані логін/пароль пари або IP-фільтри. При цьому:

– неможливо встановити індивідуальні політики для окремих користувачів або груп (наприклад, часові обмеження, географічні обмеження);

– не підтримується багатофакторна автентифікація (2FA);

– відсутній механізм автоматичної деактивації облікових записів після виявлення підозрілої активності.

Це створює додаткові ризики несанкціонованого доступу, особливо у випадках, коли облікові дані компрометовано.

2. Шифрування не ввімкнено за замовчуванням.

Хоча майже всі сучасні платформи підтримують TLS (для захисту сигналізації) та SRTP (для захисту голосового трафіку), ці функції зазвичай не активовані "з коробки". Їх увімкнення вимагає:

– створення власного сертифіката або придбання сертифіката від довіреного центру сертифікації;

– ручного редагування конфігураційних файлів;

– налаштування відповідних політик на клієнтських пристроях.

Через це багато інсталяцій залишаються незашифрованими, що робить можливим прослуховування трафіку, викрадення логінів та SIP-ідентифікаторів.

3. Відсутність вбудованого моніторингу аномалій.

Більшість рішень не мають засобів виявлення аномальної активності в реальному часі. Наприклад:

– багаторазові невдалі спроби входу (brute-force);

– нетипова географія або графік дзвінків;

– нестандартні обсяги трафіку або частота реєстрацій.

4. Відсутність механізмів журналювання, аналізу поведінки користувачів та сповіщення про інциденти призводить до того, що атаки залишаються непоміченими до моменту виникнення серйозних наслідків – компрометації системи, витоку даних, або фінансових втрат (наприклад, через міжнародний SIP-фрод).

5. Обмежена масштабованість та ізольованість.

Більшість відкритих рішень (особливо FreePBX, Asterisk у базовій конфігурації) не передбачають легку масштабованість:

– відсутні готові механізми горизонтального масштабування (кластеризації);

– немає вбудованої підтримки розподілених баз даних для зберігання акаунтів, історії дзвінків тощо;

– слабка інтеграція з сучасними інструментами DevOps (Prometheus, Grafana), SIEM, системами управління подіями безпеки;

– немає API для централізованого управління ззовні.

У результаті такі системи важко адаптувати до потреб великих організацій з десятками тисяч абонентів або з підвищеними вимогами до резервування та безперервності обслуговування.

6. Слабка адаптація до сучасних вимог безпеки

Багато SIP-систем розроблялися за часів, коли вимоги до інформаційної безпеки були значно нижчими. Сьогодні ж, у контексті загроз нульового дня, соціальної інженерії, DDoS-атак і масштабованих бот-мереж, необхідне комплексне рішення, яке б поєднувало:

– контроль поведінкових шаблонів;

– можливість виявлення загроз у режимі реального часу;

– централізований аудит та логування;

– можливість інтеграції з системами реагування (наприклад, автоматичне блокування IP-адрес або облікових записів).

Сучасні платформи VoIP часто цього не забезпечують або вимагають значних зусиль для реалізації таких функцій сторонніми засобами.

Попри широкий вибір рішень для SIP-телефонії, більшість не забезпечує належного рівня захисту в умовах сучасного кіберпростору. Вони не є комплексними та вимагають значних зусиль адміністратора навіть для налаштування базової безпеки. Це створює ризики, особливо в критичних сферах — службах підтримки, екстрених викликах, державних та військових структурах. У підрозділі розглянуто ключові недоліки типових рішень захищеної SIP-телефонії.

Поширені рішення використовують статичну автентифікацію (логін/пароль), без підтримки динамічного управління доступом за політиками, часом чи геолокацією. Механізми 2FA (TOTP, SMS) практично відсутні й потребують окремої інтеграції.

Незважаючи на наявність TLS і SRTP у багатьох системах (Asterisk, FreePBX, FusionPBX), їх використання за замовчуванням не передбачене. Активація вимагає ручного створення сертифікатів, редагування конфігурацій, оновлення клієнтів. Через це часто передається незашифрований трафік, вразливий до перехоплення.

Бракує засобів поведінкового аналізу в реальному часі: багаторазові невдалі входи, підозріла географія, нетипові дзвінки або час активності часто залишаються поза увагою. Це сприяє непомітній компрометації, SIP-фроду та витоку даних.

FreePBX і FusionPBX погано масштабуються. У великих організаціях це ускладнює централізоване керування обліковими записами, викликами, резервуванням. Відсутність API перешкоджає інтеграції з SIEM, DevOps-інструментами, обліковими системами, що унеможливлює автоматизоване реагування.

Багато реалізацій не враховують інтеграцію з Prometheus, Zabbix, ELK, IDS/IPS, що знижує ефективність виявлення загроз і ускладнює ретроспективний аналіз.

Попри розвиток VoIP, більшість платформ SIP-телефонії демонструє обмежену безпеку "за замовчуванням". В умовах зростання атак на SIP-протокол, це є критичним. Рішення фокусуються на функціональності (маршрутизація, CRM), ігноруючи автентифікацію, моніторинг, шифрування, централізоване управління й реагування.

Адміністратору часто доводиться вручну редагувати конфігурації (наприклад, pjsip.conf), генерувати TLS-сертифікати, впроваджувати 2FA. Це підвищує ризик помилок і вимагає високої кваліфікації, що не завжди доступно.

Типові SIP-системи не мають механізмів виявлення підозрілих сесій або обмеження невдалих входів, що відкриває шлях до brute-force, spraying і dictionary-

атак. Відсутність контролю сесій та блокування при перевищенні активності створює загрози, особливо при використанні слабких паролів.

Вбудовані засоби поведінкового аналізу відсутні: раптові дзвінки, незвичний час, нові IP-адреси залишаються непоміченими, що унеможливлює раннє виявлення компрометації.

Системи часто не підтримують масштабовану архітектуру та централізоване управління у великих середовищах — зокрема, через відсутність LDAP/AD.

У Asterisk, FusionPBX логування є базовим і не інтегрується із сучасними системами аналізу подій, не дозволяє гнучкий пошук чи кореляцію. Це ускладнює виявлення атак, форензик і швидке реагування.

1.5 Криптографічні алгоритми та інтеграція TLS із RSA

Криптографічні алгоритми є фундаментом забезпечення конфіденційності, цілісності та автентичності інформації, особливо в контексті SIP-телефонії, де передаються чутливі дані. Інтеграція протоколу TLS із криптографічним алгоритмом RSA відкриває широкі можливості для створення захищених каналів зв’язку, мінімізуючи ризики, пов’язані з перехопленням або підробкою даних [31].

Криптографія поділяється на два основні типи: симетрична і асиметрична. Симетричні алгоритми, такі як AES, забезпечують високу швидкість шифрування, але потребують надійного обміну ключами між сторонами. Асиметричні алгоритми, такі як RSA, дозволяють уникнути проблем із обміном ключами завдяки використанню пари відкритого та закритого ключів, але мають нижчу продуктивність. RSA заснований на математичній складності факторизації великих чисел, що робить його стійким до більшості сучасних атак за умови використання достатньої довжини ключів (2048 біт і більше) [32].

Протокол TLS є загальновизнаним стандартом для забезпечення безпеки в мережевих з’єднаннях. Його використання в SIP-телефонії дозволяє досягти кількох важливих цілей: шифрування SIP-трафіку для забезпечення конфіденційності, перевірка автентичності учасників сесії для захисту від підробок і забезпечення цілісності даних. TLS використовує RSA на етапі встановлення

з’єднання для захищеного обміну симетричними ключами, які потім застосовуються для швидкого шифрування даних у реальному часі [33].

Процес встановлення захищеного з’єднання за допомогою TLS починається з хендшейку, під час якого сервер і клієнт обмінюються сертифікатами X.509. Сертифікати містять відкриті ключі RSA та інформацію, необхідну для перевірки довіри. Після перевірки сертифікатів і автентифікації сторони обмінюються симетричними ключами, використовуючи алгоритм RSA для їх шифрування. Симетричні ключі надалі застосовуються для шифрування всіх даних сесії, забезпечуючи високу швидкість роботи й мінімальні затримки.

Інтеграція TLS із RSA у SIP-телефонії має кілька важливих переваг. По-перше, це надійний захист від атак типу "людина посередині" (MITM), оскільки перевірка сертифікатів унеможливлює підміну учасників з’єднання. По-друге, шифрування трафіку захищає від перехоплення даних, включаючи автентифікаційні дані, що є критичним для запобігання несанкціонованому доступу до SIP-системи. По-третє, використання TLS забезпечує сумісність із більшістю сучасних SIP-серверів і клієнтів, таких як Asterisk, FreeSWITCH або Cisco CallManager.

Разом із тим, існують і певні виклики, пов’язані з використанням TLS та RSA у SIP-телефонії. Обчислювальна складність RSA створює додаткове навантаження на сервери, особливо під час обробки великої кількості з’єднань. Для мінімізації цього впливу можна використовувати апаратне прискорення криптографічних операцій або оптимізувати налаштування серверів. Іншим важливим аспектом є управління сертифікатами, зокрема їх своєчасна заміна та захист від компрометації. Невчасно поновлені сертифікати або неправильна їхня конфігурація можуть спричинити помилки автентифікації та відмову в доступі до системи.

Практична реалізація TLS із RSA у SIP-телефонії може включати такі кроки: створення власного центру сертифікації (CA) для генерації та управління сертифікатами; налаштування SIP-серверів для підтримки TLS і обов’язкової перевірки сертифікатів; інтеграція SRTP для додаткового шифрування голосового

трафіку. Це дозволяє створити комплексну систему безпеки, яка захищає всі етапи передачі даних – від автентифікації до шифрування голосових повідомлень.

Інтеграція криптографічних технологій, таких як TLS із RSA, є важливим кроком у підвищенні безпеки SIP-телефонії. Вона забезпечує захищене середовище для передачі даних, мінімізуючи ризики компрометації системи. У межах цього дослідження буде розглянуто можливості оптимізації даного підходу, зокрема використання сучасних алгоритмів для пришвидшення криптографічних операцій, а також розроблено рекомендації для їх інтеграції у реальні системи.

1.6 Висновки до розділу 1 та постановка задач

У першому розділі було проведено теоретичний огляд архітектури SIP-телефонії, методів її захисту, а також сучасних підходів до забезпечення безпечної аутентифікації користувачів та шифрування голосового трафіку. Особливу увагу приділено протоколам TLS та SRTP, які забезпечують криптографічний захист сигналізації та медіа-трафіку відповідно. Проаналізовано існуючі рішення у сфері захисту VoIP, виявлено їх переваги та недоліки. Окремо розглянуто підходи до реалізації динамічного керування SIP-сесіями та зберігання параметрів у базі даних у режимі реального часу (Asterisk Realtime).

У результаті аналізу було визначено, що найбільш ефективним напрямом для подальшої розробки є поєднання:

– динамічного контролю сеансів автентифікації;

– застосування TLS з криптоалгоритмом RSA для захисту сигналізації;

– впровадження SRTP для шифрування голосового трафіку;

– розробки веб-інтерфейсу моніторингу викликів і подій безпеки.

На основі актуальності проблематики та поставленої мети були сформульовані наступні задачі дослідження:

– проаналізувати особливості протоколу SIP та вразливості його беззахисного використання;

– дослідити можливості захисту SIP через TLS та шифрування медіа-трафіку за допомогою SRTP з використанням криптоалгоритму RSA;

– реалізувати динамічний контроль автентифікації користувачів у SIP-телефонії з використанням Asterisk Realtime;

– розробити систему моніторингу викликів і подій у вигляді веб-інтерфейсу;

– протестувати систему та оцінити ефективність її захисних механізмів;

– надати економічне обґрунтування доцільності впровадження запропонованого рішення.

Виконання цих задач дозволить досягти основної мети дослідження – підвищити рівень безпеки SIP-телефонії шляхом комплексного підходу до моніторингу, автентифікації та шифрування, що є особливо важливим для об’єктів критичної інфраструктури та корпоративних інформаційних систем.

2 ПРОЄКТУВАННЯ СИСТЕМИ МОНІТОРИНГУ ТА ЗАХИСТУ SIP-ТЕЛЕФОНІЇ

Другий розділ кваліфікаційної роботи присвячений проєктуванню системи моніторингу та захисту SIP-телефонії. Метою є створення надійної системи, що забезпечує безпеку SIP-сесій, захист від атак і оперативне реагування на загрози. Розглядаються основні підходи до інтеграції протоколів шифрування та автентифікації, зокрема TLS і SRTP, а також впровадження системи моніторингу для виявлення аномальних подій і захисту голосового трафіку.

У контексті швидкого зростання кількості кіберзагроз, поширення VoIP-рішень у критично важливих сферах (державне управління, медицина, фінансовий сектор, оборонна галузь) потреба у створенні безпечної, адаптивної та масштабованої SIP-платформи стає надзвичайно актуальною. Як було зазначено у попередньому розділі, наявні системи IP-телефонії мають низку критичних недоліків: відсутність гнучких політик автентифікації, обмежене або взагалі відсутнє шифрування голосового трафіку, нестача ефективного інструментарію моніторингу та реагування на загрози.

2.1 Обґрунтування вибору підходів для розробки системи

У сучасних умовах стрімкого зростання обсягів цифрової комунікації та використання VoIP-технологій, особливо протоколу SIP (Session Initiation Protocol), питання забезпечення захищеного обміну голосовими даними набуває критичної актуальності. SIP-телефонія широко використовується в корпоративному секторі, у державних установах, військових підрозділах, службах підтримки, фінансових організаціях та елементах критичної інфраструктури, де безперебійний і безпечний обмін інформацією має вирішальне значення.

Розробка системи для захищеної SIP-телефонії вимагає врахування специфіки протоколу SIP, сучасних вимог до захисту даних і викликів у сфері кібербезпеки. Підходи спрямовані на забезпечення надійного функціонування системи за умов потенційних загроз, збереження конфіденційності інформації, що передається, та дотримання високих стандартів продуктивності й доступності.

Основною загрозою для SIP-телефонії є ризик перехоплення даних, атак на рівні сигналізації або медіа-трафіку, а також несанкціонований доступ до облікових записів користувачів. Для протидії цим викликам вибір технологій і методів базується на сучасних стандартах захисту VoIP-систем та перевірених криптографічних алгоритмах.

У зв'язку з цим доцільною є розробка удосконаленої SIP-платформи, яка враховує сучасні вимоги до інформаційної безпеки, забезпечує захист від актуальних типів атак та дозволяє адміністратору своєчасно реагувати на загрози. Основна ідея полягає у створенні рішення, що поєднує три ключові компоненти: динамічний контроль автентифікації, комплексне шифрування трафіку та інтерактивний моніторинг із засобами реагування на інциденти.

1. Динамічний контроль автентифікації.

Статичний підхід до автентифікації в SIP-системах втратив актуальність. Типові рішення базуються лише на перевірці логіна й пароля без урахування контексту: геолокації, частоти спроб, типу пристрою чи часу активності. Нова система має реалізовувати динамічні політики доступу, що реагують у реальному часі на загрози, виявляючи аномальну поведінку, блокуючи підозрілі сесії, тимчасово ізолюючи облікові записи, та застосовуючи адаптивні правила автентифікації. Обов'язковим компонентом є поведінковий аналіз, який виявляє як зовнішні, так і внутрішні загрози, забезпечуючи проактивний захист. Такий підхід знижує ризик компрометації облікових записів і підвищує стійкість до автоматизованих атак.

2. Шифрування голосового трафіку через TLS + RSA + SRTP.

Оскільки SIP за замовчуванням передає дані у відкритому вигляді, критично важливим є повне шифрування як сигналізації, так і медіа. Пропоноване рішення включає використання TLS із RSA-сертифікатами для захисту SIP-повідомлень (INVITE, REGISTER, ACK), що забезпечує їхню конфіденційність і цілісність. Для медіатрафіку застосовується SRTP, який гарантує безпеку переданих голосових даних. Застосування стійких криптоалгоритмів, як-от RSA з довжиною ключа 2048

біт і більше, значно підвищує загальний рівень захисту. Такий підхід забезпечує протидію як пасивному перехопленню, так і активним атакам на сесію.

3. Зручний інтерфейс моніторингу та реагування на інциденти.

Інформаційна безпека охоплює не лише запобігання атакам, а й своєчасне їх виявлення й реагування. Ключовим елементом запропонованої платформи є веб-інтерфейс реального часу, що забезпечує контроль за безпекою SIP-системи: перегляд активних сесій, спроб авторизації, IP-адрес, аналіз журналів викликів і RTP-трафіку, автоматичні сповіщення про інциденти. Через інтерфейс здійснюється керування списками IP-адрес (чорний/білий список) і запуск ручних або автоматизованих сценаріїв реагування, зокрема блокування IP на рівні фаєрволу. Це дозволяє оперативно виявляти загрози та мінімізувати ризики зупинки сервісу чи втрат даних.

На тлі зростання використання SIP-телефонії в державному та приватному секторах зростає потреба у комплексних рішеннях, що поєднують функціональність із високим рівнем безпеки. Сучасні системи часто зосереджені на комунікаційних можливостях, залишаючи питання безпеки на розсуд адміністратора або на післяінцидентне реагування. Це створює прогалини в захисті конфіденційних даних і безперебійній роботі сервісів.

У відповідь пропонується інноваційна SIP-платформа, яка інтегрує безпекові й функціональні рішення на рівні архітектури. Такий підхід підвищує надійність зв’язку та відповідає вимогам сучасної кібербезпеки, зокрема стандартам ISO/IEC 27001, NIST SP 800-53, рекомендаціям ENISA та положенням GDPR щодо конфіденційності комунікацій.

Також варто зазначити додаткові аспекти, що обґрунтовують розробку:

1. Висока частота атак на VoIP-сервіси. За даними міжнародних досліджень, щонайменше 1 із 4 SIP-серверів щодня зазнає спроб несанкціонованого доступу або автоматизованого сканування портів. Це вимагає впровадження розумних механізмів, здатних протидіяти не лише відомим, але й адаптивним загрозам.

2. Відсутність централізованої реакції на аномалії. Більшість реалізацій SIP-серверів не забезпечують взаємодію із зовнішніми системами SIEM/IDS або подієвими шинами. У пропонованій системі передбачено вихідні вебхуки та API-інтерфейс, які дозволяють інтегрувати платформу з інструментами моніторингу (Zabbix, Wazuh, ELK Stack) або реагування (fail2ban, firewallD, nftables).

3. Уніфікація адміністрування. Наявні рішення часто мають фрагментарний або громіздкий інтерфейс для налаштування безпеки, що веде до помилок конфігурації або її повного ігнорування. Розроблювана система має включати єдиний веб-інтерфейс управління, який дозволить зручно задавати політики доступу, керувати сертифікатами, а також бачити повну картину активності в SIP-мережі.

4. Вимоги до мобільності та захищених підключень. З огляду на популярність віддаленої роботи, все більше користувачів підключаються до SIP-мереж із зовнішніх або незахищених мереж (громадський Wi-Fi, мобільні мережі). Це вимагає надійного захищеного з'єднання через TLS + VPN або SRTP, без якого інформація може бути легко перехоплена.

Обрані підходи демонструють оптимальне поєднання сучасних технологій захисту, високу продуктивність і гнучкість у використанні. Це дозволяє створити комплексну систему, що відповідає сучасним стандартам безпеки та забезпечує надійну роботу SIP-телефонії в умовах реальних кіберзагроз.

У результаті розробки комплексної системи моніторингу та захисту SIP-телефонії очікується досягнення низки ключових переваг. Насамперед, система забезпечує підвищену стійкість до атак завдяки виявленню й блокуванню спроб підбору паролів у реальному часі, а також автоматичній реакції на зміну поведінки користувача, зокрема за параметрами геолокації, часу підключення чи типу клієнтського додатку. Водночас забезпечується повна конфіденційність комунікацій шляхом надійного шифрування як сигналізаційного, так і голосового трафіку з використанням TLS та SRTP, а також можливою інтеграцією з внутрішньою інфраструктурою відкритих ключів (PKI) організації. Система проєктується з урахуванням вимог аудиту та сертифікації: вона підтримує

генерацію детальних логів подій, журналів викликів і записів автентифікації, що робить її сумісною зі стандартами інформаційної безпеки та кіберзахисту. Окрему увагу приділено масштабованості та модульності: передбачено можливість розгортання компонентів у хмарних середовищах за допомогою Docker або Kubernetes, а гнучка архітектура системи дозволяє легко інтегрувати нові механізми контролю та алгоритми аналізу загроз відповідно до зростаючих вимог безпеки.

2.2 Архітектура комплексної системи моніторингу та захисту

Архітектура комплексної системи моніторингу та захисту SIP-телефонії спроектована з орієнтацією на надійний захист даних, високу ефективність та масштабованість. Вона забезпечує гнучке управління, зручність адміністрування та оперативне реагування на загрози в реальному часі. Кожен компонент цієї архітектури працює в тісній взаємодії, що дозволяє створити інтегровану систему, здатну забезпечити безпеку всіх етапів обробки та передачі інформації.

Проєктована система захисту SIP-телефонії базується на модульній архітектурі, що дозволяє забезпечити як безпеку передачі голосового трафіку, так і динамічний контроль автентифікації користувачів у реальному часі. Це особливо важливо для середовищ з високими вимогами до захисту інформації, таких як корпоративні мережі, державні установи, контакт-центри та об’єкти критичної інфраструктури.

Загальна архітектура системи включає наступні компоненти:

1. SIP-сервер (Asterisk). Ядро системи, що відповідає за обробку SIP-повідомлень, маршрутизацію дзвінків, взаємодію з клієнтами та шлюзами. Працює з RealTime-базою даних для динамічного керування обліковими записами.

2. Модуль контролю автентифікації. Забезпечує розширену перевірку користувачів під час встановлення сеансу. Використовує поведінкові шаблони для виявлення аномальної активності (наприклад, багаторазові спроби авторизації, підозрілі IP-адреси, часта зміна пристроїв). У разі виявлення загроз – автоматично блокує підозрілі запити або обмежує доступ.

3. TLS-модуль захисту сигнального трафіку. Всі SIP-повідомлення передаються через TLS-з'єднання із використанням криптоалгоритму RSA, що забезпечує конфіденційність, автентичність та цілісність даних на транспортному рівні.

4. SRTP-модуль шифрування голосу Застосовується для захисту RTP-трафіку (голосової інформації). Використання протоколу Secure RTP (SRTP) дозволяє шифрувати розмови в реальному часі, що унеможливлює перехоплення або підслуховування трафіку.

5. База даних RealTime. Усі облікові записи, політики доступу, записи дзвінків та інформація про сеанси зберігаються у базі даних у режимі реального часу. Це забезпечує динамічність змін без перезапуску сервісів.

6. Веб-інтерфейс моніторингу та реагування Забезпечує адміністраторам доступ до журналів подій, сповіщень про інциденти, статистики дзвінків, спроб входу, блокувань і стану системи. Доступ до панелі захищений 2FA. Надає можливість оперативного реагування на інциденти безпосередньо з інтерфейсу.

7. Система сповіщень та реагування. У разі виявлення підозрілої активності або порушень політик система автоматично повідомляє адміністратора та може запускати скрипти реагування (наприклад, відключення акаунта або блокування IP).

Одним із ключових компонентів архітектури є динамічний контроль сесій автентифікації. Цей механізм дозволяє здійснювати моніторинг усіх сеансів автентифікації в реальному часі, що важливо для забезпечення захисту від несанкціонованого доступу. Під час кожної сесії система перевіряє автентичність користувачів і пристроїв, виявляючи можливі аномалії та підозрілі активності. Це дозволяє запобігати спробам злому або несанкціонованого доступу, що може стати причиною серйозних загроз безпеці.

У той час як стандартні рішення на кшталт Fail2Ban або SNORT можуть частково виконувати функції блокування, їх обмеження у контексті SIP-сигналізації та труднощі адаптації під специфіку VoIP-мереж зумовлюють потребу у створенні вузькоспеціалізованого інструменту захисту. Розроблений модуль

інтегрується із сервером Asterisk і здійснює моніторинг у реальному часі, аналізуючи журнали подій та таблиці бази даних RealTime. Він спроектований таким чином, щоб виконувати автоматичний збір інформації про спроби автентифікації, характер вхідних запитів, типи User-Agent, частоту повторних звернень до сервера та інші показники, що можуть вказувати на потенційно шкідливу активність. На основі заданих порогових значень кількості запитів у визначений часовий інтервал модуль визначає ступінь загрози та приймає рішення щодо подальшої реакції. Якщо виявлено перевищення допустимого порогу авторизаційних спроб з однієї IP-адреси або ознаки сканування, система автоматично додає цю адресу до списку блокування, оновлює правила фаєрволу та ініціює відповідні дії в Asterisk. Функціонал модуля також включає інтерактивну веб-панель, яка надає візуалізацію активності в системі, статистику виявлених атак, а також дозволяє вручну керувати списком заблокованих адрес.

Алгоритм роботи модуля (рис. 2.1):

Крок 1. Збір подій з VoIP-системи. Модуль ініціює періодичне зчитування логів із SIP-сервера Asterisk або отримує події із журналу full-лог Asterisk. Збір може бути реалізований через cron.

Крок 2. Парсинг логів та формування сесій активності. Інформація з логів структурується за IP-адресами. Аналізуються ключові події: невдалі спроби реєстрації, частота запитів, помилки автентифікації, тип User-Agent тощо.

Крок 3. Виявлення аномалій. На основі заздалегідь встановлених порогових значень виявляються підозрілі шаблони активності. Такі події класифікуються як потенційні атаки brute-force, DoS або SIP scanning.

Крок 4. Реєстрація події безпеки. Після виявлення аномалії, в таблицю security_events додається запис із детальним описом події: IP-адреса, тип інциденту, час виявлення, додаткові дані.

Крок 5. Внесення IP до списку блокування. Якщо подія критична, модуль автоматично додає IP-адресу до таблиці ip_block_list. Для повторюваних IP також можливе оновлення часу блокування або підвищення рівня небезпеки.

Крок 6. Відправка інформації до веб-інтерфейсу адміністратора. Інформація про інцидент передається до інтерфейсу адміністрування (через API, push або оновлення бази). Адміністратор може переглянути подію та прийняти рішення щодо блокування.

Крок 7. Реалізація блокування. Блокування IP здійснюється автоматично або вручну в веб-інтефейсі, що змінює правила iptables/firewalld або оновлює параметри доступу в Asterisk RealTime.

Крок 8. Журналювання та аудит. Усі події, дії адміністратора та зміни станів IP зберігаються в таблицях бази даних для подальшого аналізу, формування звітів і виявлення нових шаблонів атак.

Додатково передбачено механізм примусового завершення SIP-сесії та можливість блокування через зміну політик доступу в конфігураційних параметрах Asterisk. Усі події фіксуються в журналі системи безпеки, а адміністратор отримує повідомлення про зафіксовану загрозу.

Для реалізації такої функціональності в системі створена модель централізованого зберігання та обробки SIP-профілів. Всі компоненти SIP-сервера працюють у взаємозв’язку з базою даних у режимі реального часу. Це означає, що додавання, видалення або зміна параметрів користувачів відбувається миттєво без необхідності перезапуску сервера.

База даних реалізується на основі MySQL і застосовується для зберігання параметрів SIP-клієнтів, їх автентифікації, контролю доступу, управління сесіями та журналювання подій. Вона дозволяє динамічно обробляти підключення користувачів, перевіряти облікові дані в режимі реального часу та забезпечує основу для реалізації поведінкового аналізу і виявлення аномалій. У контексті безпеки база даних є ядром механізму автентифікації, контролю доступу та зберігання мета-інформації про SIP-сеанси.

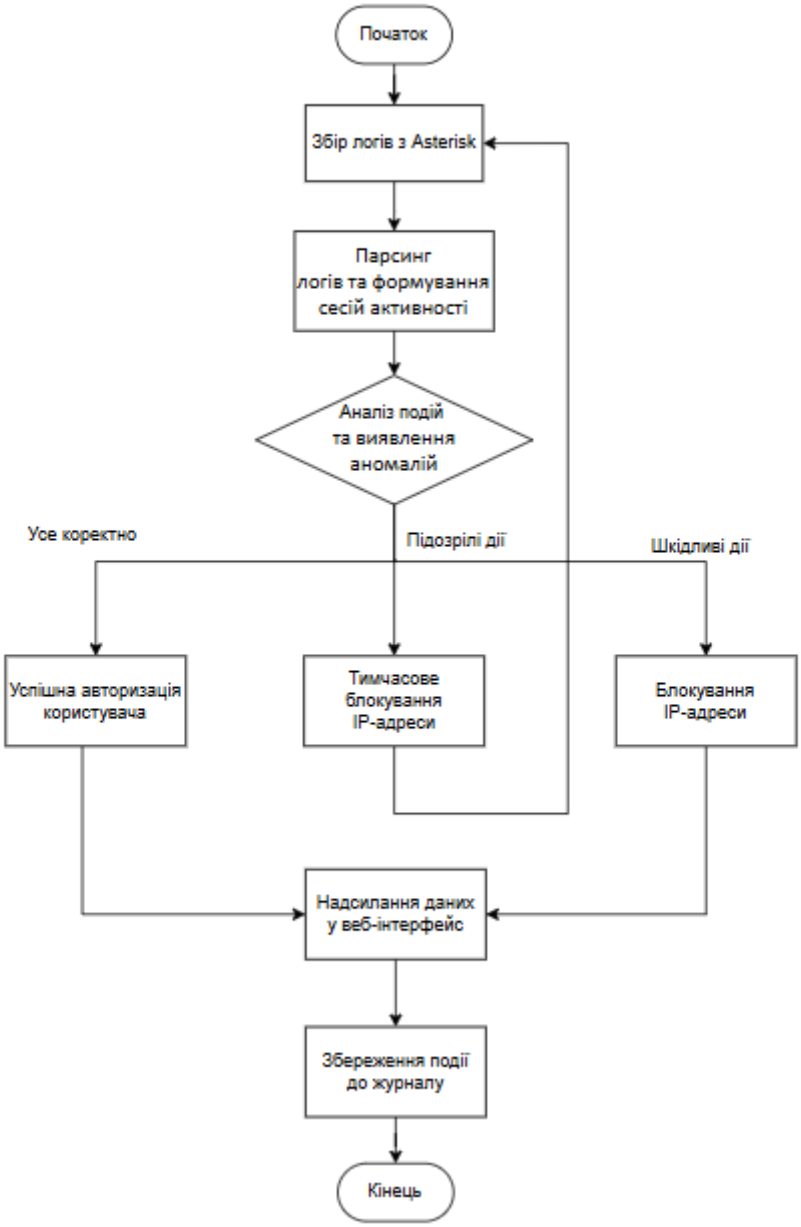


Рисунок 2.1 – Блок-схема роботи модуля контролю автентифікації

Розглянемо основні таблиці бази даних:

ps_endpoints – зберігає конфігурацію для кожного SIP-клієнта (endpoint), включаючи логіни, дозволені транспортні протоколи, шифрування (TLS/SRTP), дозволені IP-адреси, контексти дзвінків. Таблиця є центральною у взаємодії з іншими.

ps_aors – відповідає за адреси доступності (Address of Record) для SIP-клієнтів. Визначає, куди Asterisk повинен відправляти виклики для endpoint'a, скільки з'єднань дозволено одночасно, тайм-аути реєстрацій.

ps_auths – містить облікові дані для автентифікації: логін, пароль, тип автентифікації. Саме з цієї таблиці береться інформація при перевірці REGISTER-запитів користувача.

ps_contacts – зберігає активні сесії клієнтів, включаючи IP-адресу, порт, User-Agent, час останньої активності. Ця таблиця оновлюється автоматично під час кожної SIP-реєстрації або зняття реєстрації.

ps_endpoint_id_ips – дозволяє створити політики дозволу доступу до певних endpoint'ів тільки з певних IP-адрес. Це реалізує фільтрацію на основі мережевої політики та підвищує захист від зовнішніх підключень.

ip_block_list – таблиця модуля контролю безпеки, яка містить список IP-адрес, заблокованих через виявлену підозрілу активність. Зберігає причину блокування, час, статус блокування.

security_events – веде облік безпекових подій, таких як спроби несанкціонованого доступу, перебір паролів, порушення політик. Містить час, IP-адресу, тип події, опис, зв’язок із записом у ip_block_list

Структура взаємозв'язків між основними таблицями баз даних представлена на UML-діаграмі (рисунок 2.2).

Ще одним важливим елементом архітектури є інтеграція TLS (Transport Layer Security) з криптоалгоритмом RSA. Ця комбінація забезпечує захист сигналізації SIP і гарантує конфіденційність переданих даних. Протокол TLS шифрує весь канал зв'язку, захищаючи його від атак "людина посередині", що є однією з основних загроз для SIP-систем. Використання алгоритму RSA для шифрування ключів забезпечує високий рівень безпеки, оскільки тільки ті учасники комунікації, які володіють відповідними приватними ключами, можуть розшифрувати дані.

Завдяки такій інтеграції SIP-системи можуть передавати чутливі дані, включаючи аутентифікаційні дані та ключі для захисту голосового трафіку, з високим рівнем надійності та безпеки. Цей підхід мінімізує ризики, пов'язані з перехопленням даних або маніпуляцією ними, і дозволяє зберігати конфіденційність як сигналізації, так і голосових потоків.

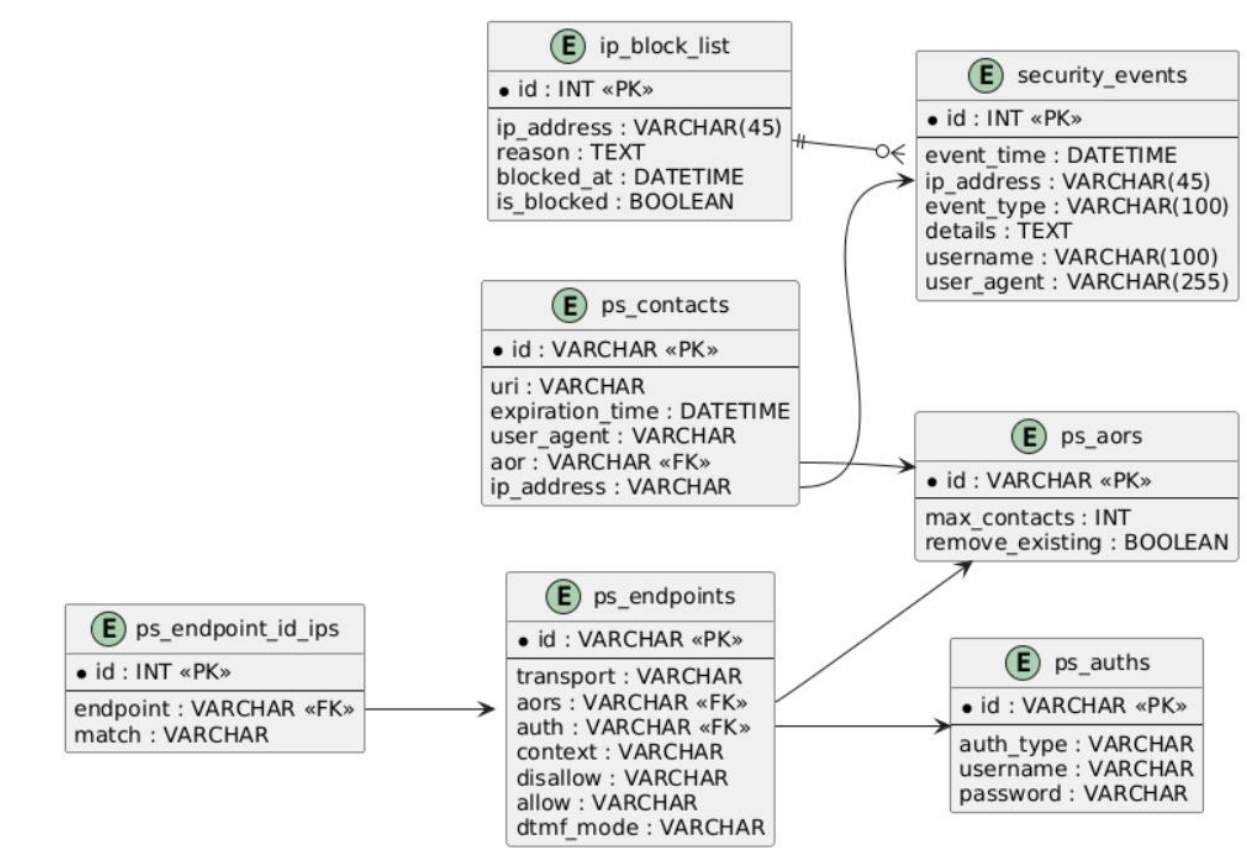


Рисунок 2.2 – UML модель взаємозв'язку між основними таблицями баз даних

Веб-інтерфейс є центральним елементом управління системою та забезпечує зручне адміністрування. Він дозволяє здійснювати централізовану настройку та моніторинг системи SIP-телефонії, зокрема керувати користувачами, відслідковувати активні сесії та контролювати стан шифрування каналів. Інтерфейс надає адміністраторам доступ до інформації про активні з’єднання, автентифікаційні спроби та використання ключів. Це дозволяє швидко виявляти проблеми, що можуть виникнути, та забезпечувати належний рівень захисту без необхідності глибокого втручання в систему.

Крім того, через веб-інтерфейс адміністратори можуть моніторити різноманітні події, що відбуваються в системі, таких як підключення з невідомих IP-адрес або спроби несанкціонованого доступу до акаунтів. Система автоматично генерує сповіщення про такі події, що дозволяє своєчасно реагувати на потенційні загрози. Інтерфейс також дозволяє змінювати налаштування безпеки, додавати чи видаляти користувачів, а також управляти параметрами шифрування.

Крім базового моніторингу безпеки, веб-інтерфейс також дозволяє отримувати статистику щодо використання ресурсів системи, що є важливим для ефективного управління навантаженням і масштабуванням системи при необхідності. Можливість отримувати в режимі реального часу звітність і аналіз подій дає змогу своєчасно виявляти аномалії та зловмисні дії, що дозволяє оперативно приймати рішення щодо подальших дій.

Розглянемо принцип роботи системи, коли користувач ініціює з'єднання, SIP-запит спочатку надходить до SIP-сервера, який передає інформацію модулю TLS для перевірки шифрування. Далі система автентифікації звіряє облікові дані з RealTime-базою, аналізує активність користувача та приймає рішення про допуск до сеансу. Якщо автентифікація пройшла успішно, система ініціює захищене голосове з'єднання через SRTP. Усі події реєструються у базі, а інформація про поточні сесії відображається у веб-інтерфейсі.

Алгоритм функціонування системи включає такі етапи (рис. 2.3):

Крок 1. Ініціація з'єднання. Користувач надсилає SIP-запит типу INVITE через порт 5061, що підтримує TLS. Завдяки використанню TLS у поєднанні з RSA-шифруванням забезпечується конфіденційність і достовірність сигнального каналу.

Крок 2. Первинна перевірка. SIP-сервер аналізує структуру запиту, IP-джерело, User-Agent, ідентифікатор облікового запису. Далі запит спрямовується до RealTime-бази для перевірки облікових даних.

Крок 3. Автентифікація з поведінковим аналізом. Окрім звичайної перевірки логіна і пароля, система перевіряє контекст сеансу: кількість спроб входу за останній період, зміну географічного розташування, IP-репутацію тощо. У разі виявлення підозрілої активності – обліковий запис або IP-адреса блокується, адміністратор отримує повідомлення.

Крок 4. Установлення медіасесії з SRTP. Після успішної автентифікації встановлюється RTP-сесія між абонентами, яка шифрується за допомогою SRTP. Ключі шифрування обмінюються через SDES або DTLS. Це виключає можливість прослуховування трафіку навіть при доступі до мережевого середовища.

Крок 5. Моніторинг активної сесії. У реальному часі відображається вся інформація про дзвінок: IP, статус шифрування, тривалість, кодеки. Логування подій дозволяє аналізувати історію викликів та реагувати на порушення безпеки.

Крок 6. Реагування на інциденти. Система автоматично блокує IP у разі підозрілих дій (brute-force, spoofing, DoS). Інформація надсилається до веб-інтерфейсу, а також через інтегровані канали повідомлень (наприклад, Telegram).

Крок 7. Завершення сесії. Після завершення виклику сесія фіксується в базі даних з усіма параметрами (тривалість, шифрування, маршрут, статус перевірок). У разі інциденту – запис позначається для подальшого аналізу.

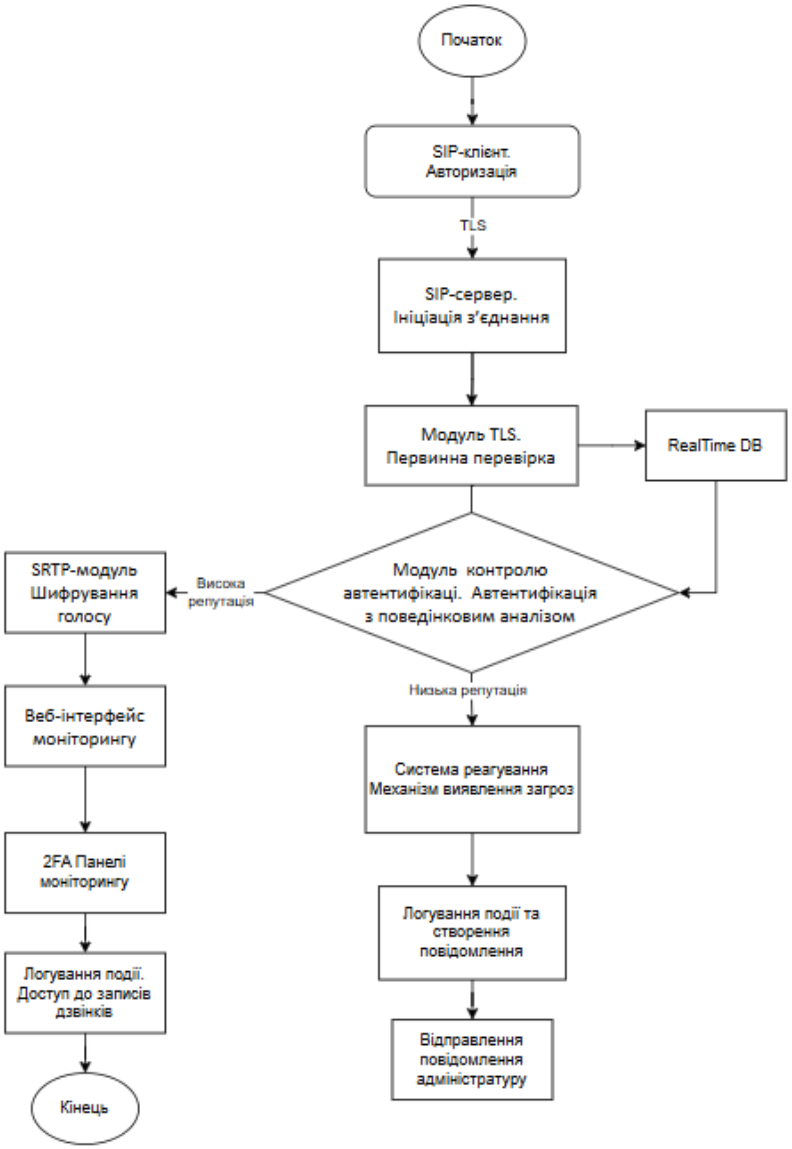


Рисунок 2.3 – Блок-схема системи

Запропонована система захищеної SIP-телефонії побудована за модульною архітектурою з урахуванням вимог до конфіденційності, цілісності, доступності та моніторингу. Основною метою архітектури є не лише забезпечення шифрування голосового трафіку, а й впровадження динамічного контролю автентифікації, виявлення аномальної активності та оперативного реагування на інциденти безпеки. Система інтегрує компоненти з відкритим кодом, що дозволяє адаптувати її під конкретні потреби організації.

2.3 Моделювання сценаріїв використання системи захисту

Моделювання сценаріїв використання системи захисту SIP-телефонії є важливим етапом, що дозволяє виявити потенційні загрози та оцінити ефективність застосованих механізмів захисту. Для цього розглядаються кілька реалістичних сценаріїв, у яких система може бути піддана різним типам атак, а також визначаються методи реагування для забезпечення безпеки та стабільної роботи системи.

Розглянуті сценарії охоплюють як штатну взаємодію користувача із системою, так і ситуації, пов’язані з порушенням політик безпеки або несанкціонованим доступом. У штатному режимі авторизований користувач надсилає SIP-запит REGISTER через порт 5061 з TLS-шифруванням, після чого система перевіряє облікові дані в RealTime-базі, виконує аналіз контексту (IP, User-Agent, кількість спроб), встановлює SRTP-сесію для передачі голосу і передає дані до веб-інтерфейсу моніторингу. У випадку атаки типу brute-force система фіксує масові невдалі спроби авторизації, блокує IP-адресу, надсилає адміністратору повідомлення через Telegram і журналює подію. Якщо здійснюється підключення з нового пристрою, модуль поведінкового аналізу фіксує відхилення від шаблону, блокує запит або вимагає підтвердження, інформуючи користувача, а адміністратор має змогу підтвердити або заборонити доступ. При спробі перехоплення трафіку безпеку забезпечує передача SIP-пакетів через TLS і голосових даних через SRTP, що робить дешифрування без приватного ключа неможливим. У випадку виявлення аномальної активності система автоматично

генерує сповіщення, адміністратор блокує обліковий запис або IP-адресу через веб-інтерфейс, а інцидент фіксується для подальшого аналізу, забезпечуючи швидке реагування на загрози.

Ці сценарії охоплюють ключові аспекти роботи системи захисту SIP-телефонії, зокрема автентифікацію, поведінковий аналіз, захист трафіку, а також моніторинг і реагування на інциденти. Їх моделювання дозволяє оцінити ефективність розроблених засобів безпеки в умовах як штатної експлуатації, так і під час потенційних атак. Реалізація цих заходів дозволяє ефективно захищати систему від перехоплення даних та несанкціонованого доступу, забезпечуючи захист конфіденційної інформації та стабільну роботу SIP-мережі.

2.4 Вибір середовища розробки та технологій

Одним із ключових елементів є налаштування SIP-телефонії. Для цього було обрано Asterisk – потужну та гнучку платформу з відкритим вихідним кодом, яка дозволяє створювати повноцінні VoIP-рішення. Asterisk підтримує протокол SIP, інтеграцію із шифруванням SRTP та роботу з TLS, що робить його оптимальним вибором для захищених комунікацій. Для адміністрування та конфігурації використовується стандартна CLI-консоль, що значно спрощує налаштування.

У контексті забезпечення шифрування сигналізації SIP-сесій було прийнято рішення інтегрувати протокол TLS з використанням криптоалгоритму RSA. Для реалізації цього функціоналу були обрані наступні бібліотеки та фреймворки:

OpenSSL – набір криптографічних інструментів, що підтримує реалізацію TLS-з'єднань, генерацію сертифікатів X.509, обмін ключами за допомогою RSA та шифрування трафіку. OpenSSL використовується як на серверній стороні (для налаштування Asterisk), так і при конфігурації клієнтів [34].

PJSIP – бібліотека з підтримкою SIP, SDP, STUN, TURN, ICE та SRTP. Вона дозволяє гнучко керувати параметрами сигналізації та медіа, включаючи TLS-захист SIP-сесій.

LibSRTP – бібліотека для реалізації протоколу Secure RTP, що відповідає за шифрування медіа-трафіку. Вона інтегрується з Asterisk та іншими VoIP-движками, забезпечуючи захист голосового потоку.

Для створення веб-інтерфейсу управління системою та моніторингу було використано стек LAMP (Linux, Apache, MySQL, PHP), а також додаткові інструменти, такі як Bootstrap для побудови адаптивного інтерфейсу та Chart.js для візуалізації статистичних даних.

Для керування користувачами та авторизацією реалізовано бекенд частину, що працює з LDAP або базою даних MySQL, залежно від конфігурації. Це забезпечує централізоване управління обліковими записами та зберігання історії автентифікації.

Окрему увагу приділено безпечному зберіганню ключів і сертифікатів. Для цього використовується сховище з обмеженим доступом на основі PKI (Public Key Infrastructure), яке забезпечує надійне управління сертифікатами та контроль їхнього життєвого циклу.

Загальна архітектура середовища (табл. 2.1):

– Серверна частина розгортається на Linux-дистрибутиві Ubuntu 22;

– Asterisk виконує функції обробки SIP-запитів, RTP-трафіку та захищених з’єднань через TLS/SRTP;

– MySQLзберігає динамічні дані про SIP-користувачів, правила контролю, історію сесій;

– PHP відповідає за бізнес-логіку веб-інтерфейсу та моніторинг;

– OpenSSL реалізує TLS-захист і сертифікацію;

– iptables + Fail2ban формують базову захисну політику;

– Веб-інтерфейс надає адміністратору засоби перегляду активних з’єднань, сповіщення про підозрілу активність та можливість миттєвого реагування.

Таблиця 2.1 – Ключові елементи середовища розробки

Інструмент / технологія	Призначення
Asterisk (з PJSIP)	SIP-сервер з підтримкою шифрування (TLS/SRTP), реєстрації абонентів, обробки дзвінків
MySQL	Реляційна база даних для зберігання інформації про SIP-користувачів, сесії, журнали подій
OpenSSL	Генерація та управління TLS-сертифікатами, криптографічні операції для захисту трафіку
PHP (8.x)	Серверна мова для розробки веб-інтерфейсу моніторингу та керування SIP-акаунтами
Apache	Веб-сервер для розгортання адміністративної панелі
iptables / nftables	Організація міжмережевого екрану, блокування підозрілих IP-адрес вручну або через скрипти
Fail2ban	Автоматичне блокування атак типу brute-force на SIP-порти

Asterisk із підтримкою PJSIP був обраний як основа системи завдяки своїй здатності працювати з сучасним SIP-стеком, який реалізовано на базі модуля PJSIP. На відміну від застарілого модуля chan_sip, він підтримує гнучкі механізми автентифікації, у тому числі через RealTime-базу, дозволяє використовувати різні транспортні протоколи (UDP, TCP, TLS) та має вбудовану підтримку SRTP для шифрування голосового трафіку. У якості сховища облікових записів, шаблонів автентифікації та сесійної інформації використовується MySQL, що в поєднанні з RealTime-режимом Asterisk забезпечує динамічне керування конфігурацією без потреби у перезапуску сервісів. Це також дозволяє централізовано зберігати дані для подальшої аналітики та впроваджувати складні механізми контролю через SQL-запити та тригери.

Реалізація TLS і SRTP у конфігурації Asterisk гарантує наскрізне шифрування як сигналізації, так і голосового трафіку між вузлами, що значно підвищує рівень безпеки комунікацій. У межах загальної архітектури додатково інтегруються такі інструменти, як Fail2ban — для виявлення й блокування brute-force атак на SIP-інтерфейси, та iptables або nftables — для застосування адаптивних політик фільтрації трафіку на мережевому рівні.У контексті побудови захищеної SIP-телефонії, де комунікації мають конфіденційний та часто критично важливий

характер, безпека середовища розробки набуває фундаментального значення. Тому кожен обраний інструмент оцінювався насамперед за критеріями підтримки захищених протоколів (TLS, SRTP), стабільності, гнучкості налаштувань та можливості масштабування у великих корпоративних мережах.

Завдяки такій конфігурації, запропоноване середовище забезпечує повний цикл обробки та контролю SIP-сесій – від автентифікації до аналізу подій безпеки – в умовах підвищених вимог до захищеності зв'язку.

2.5 Обгрунтування вибору мови програмування

У процесі розробки системи моніторингу та захисту SIP-телефонії було прийнято рішення використовувати мову програмування PHP для створення серверної логіки веб-інтерфейсу управління користувачами, а також для реалізації функцій моніторингу та обробки подій безпеки.

Вибір саме PHP зумовлений низкою технічних і практичних переваг. Насамперед, PHP є однією з найпоширеніших мов для створення веб-додатків, що забезпечує широке коло підтримуваних фреймворків, бібліотек та інструментів для швидкої розробки та інтеграції з іншими системами. Зокрема, використання таких бібліотек як phpseclib або розширень OpenSSL для PHP дозволяє ефективно працювати з криптографічними операціями, включаючи реалізацію TLS та RSA.

Мову PHP обрано для розробки серверної частини веб-інтерфейсу завдяки її широким технічним можливостям, серед яких — вбудована підтримка криптографії через OpenSSL для обробки TLS-з'єднань, перевірки сертифікатів та шифрування повідомлень[34]. PHP також забезпечує ефективну взаємодію з базами даних, зокрема MySQL, за допомогою розширень PDO або MySQLi, що дозволяє працювати з таблицями RealTime. Завдяки великій кількості бібліотек і фреймворків, розробка здійснюється швидко й структуровано. PHP легко інтегрується з Asterisk через AMI або REST API, що дає змогу реалізувати керування сесіями та отримання інформації в режимі реального часу.

PHP відзначається високою сумісністю з LAMP-стеком (Linux, Apache, MySQL, PHP), що робить його оптимальним вибором для створення надійного,

масштабованого та безпечного веб-інтерфейсу. Це дозволяє організувати керування доступом, зберігати логи подій, реалізувати автентифікацію та візуалізувати стан системи. Окрім цього, PHP забезпечує зручну роботу з різними базами даних, включаючи MySQL і PostgreSQL, для зберігання даних про сесії, логи доступу, користувачів і параметри безпеки. Завдяки підтримці підготовлених SQL-запитів, зменшується ризик атак типу SQL-ін'єкцій, що підвищує загальний рівень безпеки.

PHP також добре підходить для розробки REST API, які можуть використовуватись зовнішніми системами для інтеграції з SIP-сервером, системами журналювання або сервісами автентифікації. Це забезпечує модульність і розширюваність архітектури проєкту, дозволяючи легко підключати сторонні сервіси або власні клієнтські застосунки [36].

У контексті забезпечення безпеки, PHP дає змогу реалізувати такі засоби, як:

– контроль сесій користувача із застосуванням токенів доступу (access tokens);

– захист від CSRF- та XSS-атак;

– гнучке управління правами доступу на основі ролей (RBAC).

У процесі розробки система на PHP дозволяє гнучко реалізувати ключові компоненти веб-інтерфейсу, а також механізми взаємодії із SIP-сервером. Завдяки підтримці різноманітних бібліотек, таких як OpenSSL, phpseclib, JWT (JSON Web Token) та GuzzleHTTP, можливо реалізувати захищені протоколи комунікації, цифровий підпис даних, генерацію токенів автентифікації, а також інтеграцію з зовнішніми API.

Використання об'єктно-орієнтованого програмування в PHP дало змогу побудувати модульну архітектуру системи, яка спрощує супровід, масштабування та тестування. Кожен клас виконує окрему функцію: роботу з конфігураціями SIP-акаунтів, управління користувачами, автентифікацію, логування дій, моніторинг з'єднань у реальному часі, збереження та шифрування журналів подій для захисту інформації. Особливу увагу приділено розробці інтуїтивного і безпечного адміністративного інтерфейсу, що забезпечує зручну візуалізацію логів, статистики

підключень і аномалій за допомогою PHP у поєднанні з HTML, CSS і JavaScript. Серверна логіка обробляє події в режимі реального часу: при спробах несанкціонованого доступу або зміні налаштувань SIP система автоматично фіксує інцидент, надсилає повідомлення адміністраторам і блокує підозрілу сесію.

2.6 Висновки до розділу 2

У другому розділі було здійснено проєктування архітектури комплексної системи моніторингу та захисту SIP-телефонії. Розглянуто загальну структуру інформаційної системи, її основні компоненти, взаємозв'язки та потоки даних. Особливу увагу приділено питанням безпечної аутентифікації користувачів, захисту сигналізації та медіа-трафіку.

Було розроблено логічну структуру бази даних системи із використанням підходу Asterisk Realtime, яка забезпечує динамічне створення та керування SIP-акаунтами. Також наведено опис компонентів бази даних і обґрунтовано її вибір для забезпечення гнучкості та масштабованості проєкту.

Проаналізовано алгоритми та технології, що використовуються для підвищення рівня безпеки: інтеграція протоколу TLS із криптоалгоритмом RSA для захисту сигналізації, використання SRTP для шифрування голосового трафіку, а також механізми динамічного контролю автентифікації. Окремо розглянуто роль системи моніторингу дзвінків та подій у забезпеченні прозорості й вчасного виявлення аномальної активності.

Таким чином, розділ закладає теоретичну та технічну основу для реалізації програмної частини системи в наступному етапі роботи. Спроєктовані компоненти забезпечують гнучку й надійну платформу для реалізації сучасної та безпечної SIP-телефонії з вбудованими засобами моніторингу та захисту.

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ СИСТЕМИ МОНИТОРИНГУ ТА ЗАХИСТУ SIP-ТЕЛЕФОНІЇ

У цьому розділі представлено етапи практичної реалізації програмно-апаратного комплексу, призначеного для моніторингу та захисту SIP-телефонії. В основі системи лежить поєднання сучасних технологій VoIP-зв'язку з криптографічними методами захисту інформації та інструментами для візуалізації та адміністрування подій.

Описано процес розгортання SIP-серверної інфраструктури, налаштування автентифікації користувачів, інтеграцію протоколу TLS із криптоалгоритмом RSA для забезпечення захищеної сигналізації, а також розробку веб-інтерфейсу для керування та оперативного реагування на інциденти безпеки. Особливу увагу приділено тестуванню системи в умовах, наближених до реального середовища, що дозволило перевірити її надійність, стабільність та здатність протидіяти типовим загрозам, таким як перехоплення трафіку або несанкціонований доступ.

Розділ демонструє, як на основі теоретичних напрацювань і обраних технологій було створено цілісне рішення, яке забезпечує не лише захист, а й зручне керування SIP-інфраструктурою у динамічному середовищі.

3.1 Розгортання серверної інфраструктури SIP-телефонії

На цьому етапі було здійснено початкове розгортання серверної частини інфраструктури SIP-телефонії з метою створення стабільного середовища для забезпечення безпечного та ефективного обміну голосовими повідомленнями. Серверна частина є основною платформою, яка відповідає за маршрутизацію дзвінків, автентифікацію користувачів, збереження параметрів сеансів та підключення до бази даних, у якій зберігаються конфігурації.

Розгортання здійснювалося на основі операційної системи Ubuntu linux 22, яка була обрана завдяки стабільності, довготривалій підтримці (LTS) та сумісності з більшістю серверних рішень корпоративного рівня. Для реалізації SIP-сервера використано програмне забезпечення Asterisk, яке забезпечує гнучке налаштування SIP-телефонії та підтримує інтеграцію з зовнішніми базами даних.

Перед початком інсталяції було підготовлено віртуальне серверне середовище з такими параметрами:

Процесор: 2 vCPU

Оперативна пам’ять: 4 ГБ

Диск: 40 ГБ SSD

Операційна система: Ubuntu Linux 22 (x86_64)

Системні конфігураційні файли було налаштовано відповідно до специфіки задачі: створено облікові записи SIP-користувачів із надійною автентифікацією, визначено маршрути дзвінків, а також встановлено обмеження на підключення з незареєстрованих IP-адрес.

Після цього було проведено встановлення Asterisk зі збиранням його з вихідного коду. Такий підхід надав змогу уникнути зайвих модулів, які могли б стати потенційними точками вразливості, і зосередитися лише на тих компонентах, що безпосередньо забезпечують SIP-комунікацію та взаємодію з TLS для максимальної гнучкості та можливості включення/виключення модулів. Після інсталяції були налаштовані конфігураційні файли, зокрема:

pjsip.conf – для опису SIP-ендпоінтів та транспортів;

extensions.conf – для логіки обробки дзвінків;

res_odbc.conf та extconfig.conf – для інтеграції з MySQL через RealTime API.

У конфігураційних файлах не вказується статичне визначення користувачів, оскільки їх збереження здійснюється в базі даних.

Для збереження параметрів користувачів, реєстрацій, інформації про дзвінки було налаштовано підключення до СУБД MySQL. Всі таблиці створено згідно з моделлю Asterisk RealTime та структурою PJSIP.

Було перевірено підключення до бази даних через odbc.ini та res_odbc.conf, а також протестовано зчитування конфігурації ендпоінтів в режимі реального часу.

Система передбачає автентифікацію користувачів під час реєстрації SIP-клієнтів. Дані автентифікації (логін, хеш пароля, типи аутентифікації) зберігаються в таблиці ps_auths. Додатково реалізована прив’язка до таблиці ps_aors, яка дозволяє обслуговувати множинні реєстрації одного користувача.

Відповідність між SIP-ідентифікаторами, IP-адресами, портами та іншими параметрами контролюється через таблицю ps_endpoint_id_ips, яка забезпечує можливість обмеження доступу за IP.

У ході розгортання також здійснювалося первинне тестування усіх функцій системи, включно з реєстрацією SIP-клієнтів, ініціалізацією дзвінків, обробкою маршрутизації та перевіркою журналів активності. Сервер було адаптовано до майбутньої інтеграції з компонентами моніторингу, такими як веб-інтерфейс керування та система динамічного контролю автентифікації.

Додатковим важливим аспектом під час розгортання серверної інфраструктури стало врахування питань продуктивності та масштабованості. Оскільки SIP-телефонія є системою, чутливою до затримок, було здійснено оптимізацію параметрів ядра операційної системи та параметрів мережевої взаємодії. Зокрема, було налаштовано черги пакетів, збільшено розмір буферів і введено пріоритетну обробку VoIP-трафіку. Це дозволило знизити ймовірність виникнення затримок у голосовій комунікації навіть при навантаженні на мережу.

Іншою важливою складовою стало логування та зберігання системної інформації. До інфраструктури було додано модулі журналювання, що ведуть облік усіх дій користувачів, системних викликів та спроб доступу. Це дозволяє не лише забезпечити аудит подій, але й оперативно виявляти аномалії в поведінці користувачів або системи загалом. Логи передаються до централізованої системи обробки подій, яка у подальшому аналізується модулями моніторингу.

3.2 Інтеграція модулів шифрування та захисту трафіку

У межах реалізації комплексної системи захисту SIP-телефонії одним із пріоритетних напрямів стало впровадження криптографічних механізмів захисту сигналізаційного і голосового трафіку. Основна мета полягала у створенні стійкого до атак середовища передачі даних, яке забезпечувало б конфіденційність, цілісність та автентичність інформації, що передається в межах VoIP-комунікацій.

Захист сигналізаційного трафіку був реалізований шляхом використання протоколу TLS (Transport Layer Security), який дозволяє створювати зашифровані

канали між SIP-клієнтами та сервером. У процесі встановлення з'єднання виконується обмін цифровими сертифікатами, що дозволяє автентифікувати сторони з'єднання та гарантувати цілісність переданих даних. Кожен клієнт перевіряє цифровий сертифікат сервера перед початком сеансу, а у випадку невідповідності або відсутності підпису з'єднання блокується.

Для забезпечення високої продуктивності системи та уникнення затримок під час шифрування медіа та сигнального трафіку було проведено оптимізацію налаштувань TLS: вибрано найефективніший протоколу TLS 1.2, встановлення криптографічно стійкого шифру RSA, а також використання стиснення лише в безпечних контекстах.

Для TLS-з'єднання на стороні Asterisk було налаштовано PJSIP-транспорт transport-tls з відповідними параметрами:

```
[transport-tls]
type=transport
protocol=tls
bind=0.0.0.0:5061
cert_file=/etc/asterisk/keys/server.crt
priv_key_file=/etc/asterisk/keys/server.key
method=tlsv1_2
```

SSL-сертифікат створено за допомогою OpenSSL з самопідписаним центром сертифікації, а також підтримано можливість інтеграції з Let's Encrypt. Було протестовано реєстрацію клієнтів (наприклад, Linphone та MicroSIP) з TLS-шифруванням, що підтвердило захист обміну сигналізаційними SIP-повідомленнями.

Шифрування голосового трафіку було реалізовано шляхом впровадження протоколу SRTP (Secure Real-Time Transport Protocol), який забезпечує конфіденційність, цілісність і захист від повторного відтворення аудіоданих у реальному часі. Обидві сторони використовують однакові ключі шифрування, отримані під час ініціалізації з'єднання.

Після встановлення TLS-сеансу відбувається обмін ключами за допомогою криптоалгоритму RSA. Сервер передає клієнту відкритий ключ, яким той шифрує симетричний ключ, необхідний для шифрування медіатрафіку. Сервер, у свою чергу, використовуючи приватний ключ, розшифровує отримане повідомлення та

ініціалізує SRTP-сеанс. Таким чином досягається безпечна передача голосових даних із використанням симетричного шифрування, ключі до якого передаються в захищеному вигляді (рис. 3.1).

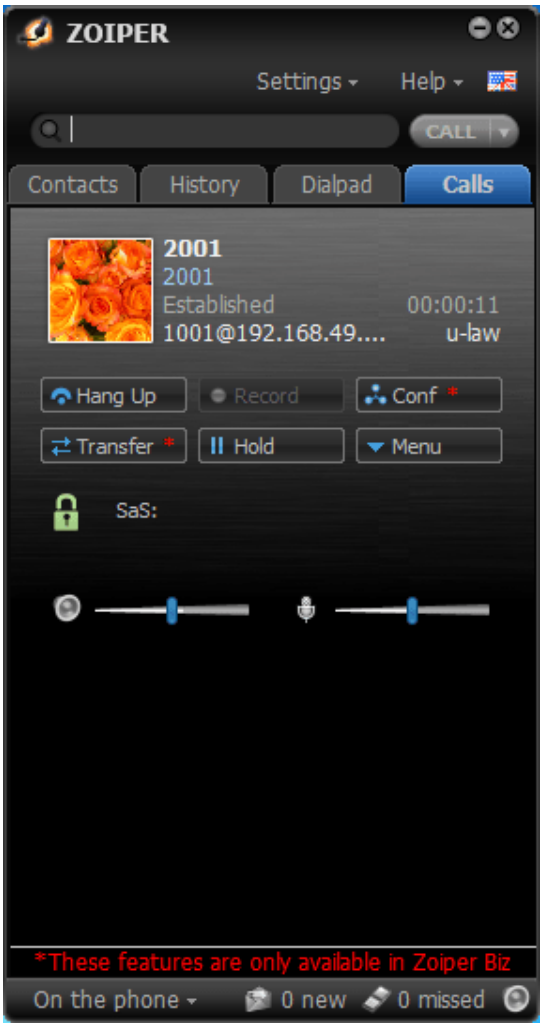


Рисунок 3.1 – Передача голосових даних під час SRTP-сеансу у програмі Zoiper

Важливо, що при спробі підключення без використання TLS або з недійсним сертифікатом, сервер автоматично фіксує інцидент та ініціює захисні заходи, зокрема блокування IP-адреси джерела за допомогою системи Fail2Ban.

Реалізація модуля контролю безпеки SIP-телефонії з виявленням аномальної активності ґрунтується на активному моніторингу сигналізаційного трафіку, аналізі динаміки підключень та реагуванні на виявлені відхилення за заданими критеріями. Основне завдання модуля – виявляти потенційно небезпечні дії на рівні SIP-протоколу, такі як несанкціоновані спроби автентифікації, масові реєстрації з одного джерела, зміна User-Agent'a або інші нетипові шаблони поведінки, ще до

того, як вони можуть порушити працездатність системи або скомпрометувати дані користувача.

Модуль реалізовано у вигляді PHP-скрипту, що взаємодіє з базою даних, яка зберігає інформацію про всі події, спроби реєстрації, автентифікації, джерела підключення та результат перевірки.

```
function get_ip_history($mysqli, $ip) {
  $stmt = $mysqli->prepare("SELECT COUNT(*) as events, MAX(timestamp) as last_seen FROM
security_events WHERE ip_address = ?");
  $stmt->bind_param("s", $ip);
  $stmt->execute();
  $stmt->bind_result($events, $last_seen);
  $stmt->fetch();
  $stmt->close();
  $geo_data = get_stored_geo($ip);
  return [
    'events' => $events,
    'last_seen' => $last_seen,
    'geo_data' => $geo_data];}
```

Ключову роль у модулі відіграє таблиця security_events, у якій фіксуються всі підозрілі дії з вказанням часу, IP-адреси, типу події (наприклад, "Excessive auth failures" або "Suspicious User-Agent"), та детальною інформацією. Це дозволяє формувати історію поведінки з джерелами ризику, і в разі повторення – автоматично активувати механізм блокування.

Під час кожного нового з'єднання система ініціалізує запит до бази даних, витягує історичну інформацію про IP-адресу клієнта, User-Agent, логін, кількість попередніх спроб, географічну приналежність та інші параметри, які дозволяють сформувати повну картину поведінки.

```
  while ($row = $result->fetch_assoc()) {
    $history['events']++;
    if (!$history['last_seen']) {
      $history['last_seen'] = $row['event_time'];}
    $history['logins'][] = $row['username'];
    $history['user_agents'][] = $row['user_agent'];}
  $stmt->close();
  $history['geo'] = get_geo_data($ip);
  return $history;}
```

Таблиця ip_block_list виконує функцію чорного списку адрес: якщо IP потрапляє під щонайменше один з критичних критеріїв, він додається до списку з

відміткою про час, причину блокування та статус активності. При наступних запитах модуль першочергово перевіряє наявність IP-адреси в цій таблиці, і в разі збігу – негайно завершує сеанс без передачі виклику в Asterisk.

```
$stmt = $mysqli->prepare("SELECT status FROM ip_block_list WHERE ip_address = ? AND status =
'blocked'");
    $stmt->bind_param("s", $ip);
    $stmt->execute();
    $stmt->store_result();
    if ($stmt->num_rows > 0) {
      $stmt->close();
      continue;}
    $stmt->close();
```

Для кожного SIP-повідомлення система оцінює низку параметрів, серед яких:

Частота запитів з однієї IP-адреси.

```
foreach ($ip_attempts as $ip => $count) {
  if ($count >= $threshold) {
    block_ip($ip, "Excessive auth failures: $count attempts in $interval sec", $mysqli);
    log_event($mysqli, $ip, 'failed_auth_threshold', "Detected $count failed attempts in $interval
sec");}}
```

Кількість невдалих спроб входу.

```
if (stripos($line, 'authentication failed') !== false || stripos($line, 'failed') !== false) {          if
(!isset($ip_attempts[$ip])) $ip_attempts[$ip] = 0;
        $ip_attempts[$ip]++;        }
```

Географічна аномалія підключення.

```
 $current_geo = get_geo_info($ip);
  $geo_str = $current_geo['country'] . ', ' . $current_geo['city'];
  $previous_geo = $history['geo_location'] ?? null;
  $geo_anomaly = false;
  if ($previous_geo && $previous_geo !== $geo_str) {
    $geo_anomaly = true;
    log_event($mysqli, $ip, 'geo_anomaly', "New region for same login: $login from $geo_str,
previous: $previous_geo");}
```

Відсутність шифрування при підключенні до TLS-порту.

```
if (!$is_tls && stripos($line, '5061') !== false) {
        log_event($mysqli, $ip, 'unencrypted_tls_connection', "Unencrypted connection to TLS
port");}
```

Незвична поведінка одного endpoint'а – наприклад, реєстрації з кількох адрес.

```
$suspicious_agents = ['sipvicious', 'friendly-scanner', 'iWar'];
      foreach ($suspicious_agents as $sus_ua) {
        if (stripos($user_agent, $sus_ua) !== false) {
          log_event($mysqli, $ip, 'scan_detected', "Suspicious User-Agent: $user_agent");
          break;}}
```

Окрім фіксації подій та блокування, модуль забезпечує логічну взаємодію з панеллю адміністратора, що дозволяє вручну переглядати список подій, редагувати чорний список, аналізувати IP-адреси та створювати виключення.

Зібрані дані зберігаються у базі даних та візуалізуються через веб-інтерфейс адміністратора (рис 3.2).

SIP Security Panel

IP Address	Reason	Status	Timestamp	Actions
192.168.1.100	Suspicious activity	blocked	2025-05-30 16:57:46	Allow Delete
10.0.0.5	Multiple failed login attempts	blocked	2025-05-30 16:57:46	Allow Delete
172.16.0.22	Port scanning detected	blocked	2025-05-30 16:57:46	Allow Delete
203.0.113.45	Suspicious activity	blocked	2025-05-30 16:57:46	Allow Delete
198.51.100.12	Brute force attack	blocked	2025-05-30 16:57:46	Allow Delete
192.0.2.1	Multiple failed login attempts	blocked	2025-05-30 16:57:46	Allow Delete
192.168.50.25	Manual block for testing	blocked	2025-05-30 16:57:46	Allow Delete
203.0.113.55	Port scanning detected	blocked	2025-05-30 16:57:46	Allow Delete
198.51.100.22	Brute force attack	blocked	2025-05-30 16:57:46	Allow Delete

Рисунок 3.2 – Вигляд веб-інтерфейс журналу безпеки адміністратора

Рисунок 3.2 демонструє журнал безпеки, де фіксуються події, пов’язані з автентифікацією, помилками TLS-шифрування, невдалими спробами реєстрації та виявленими відхиленнями у трафіку.

Продовжуючи реалізацію захищеного середовища SIP-телефонії, основна увага була зосереджена на впровадженні системи моніторингу SRTP-сеансів, політик шифрування та засобів виявлення потенційних атак, зокрема атак типу «людина посередині» (MITM). Забезпечення постійного контролю криптозахищених каналів дозволяє не лише оперативно реагувати на відхилення у поведінці трафіку, а й підтримувати актуальний рівень захисту на основі динамічного аналізу сесій.

Для зниження ризику компрометації ключів були реалізовані обмеження на час життя сеансу та регулярна ротація симетричних ключів у рамках SRTP. На рівні політик шифрування було запроваджено примусове використання лише стійких алгоритмів, зокрема RSA. Усі спроби встановити з’єднання із застарілими

параметрами або відсутністю шифрування блокувалися ще до завершення фази автентифікації.

Ще одним рівнем контролю стала перевірка збоїв у процесі встановлення TLS-з'єднань. Якщо протягом короткого інтервалу часу від однієї IP-адреси фіксувалася значна кількість невдалих спроб встановити TLS-сеанс, це трактувалося як потенційна атака перебором або аналізом сертифікатів. Такі дії блокувалися за допомогою автоматичного занесення адреси до чорного списку на рівні фаєрвола.

3.3 Реалізація веб-інтерфейсу для моніторингу та керування SIP

З метою забезпечення зручного доступу до функціональностей системи моніторингу та адміністрування SIP-телефонії було прийнято рішення про розробку власного веб-інтерфейсу. Такий підхід дозволив об’єднати засоби спостереження за активністю користувачів, контроль за станом з’єднань, перегляд подій безпеки, керування обліковими записами. Веб-інтерфейс став ключовим елементом інтегрованої архітектури системи, що забезпечує централізоване керування у режимі реального часу.

Для реалізації інтерфейсу було використано сучасний стек технологій, орієнтований на високу продуктивність, безпеку та масштабованість. Бекенд-сервер було реалізовано на основі PHP з використанням фреймворку Laravel, що забезпечило розширювану структуру проєкту, підтримку аутентифікації, обробку запитів REST API та підключення до бази даних SIP RealTime.

Важливим етапом проєктування інтерфейсу стало визначення функціональних модулів. У першій черзі було реалізовано модулі для перегляду активних SIP-сесій, перегляду статусу реєстрації користувачів, моніторингу зашифрованих з’єднань, виявлення аномалій у трафіку, а також фіксації подій, пов’язаних з автентифікацією, помилками TLS і збоїв SRTP-сеансів. Візуалізація даних була реалізована у вигляді інтерактивних дашбордів, що оновлюються в реальному часі.

Крім спостереження, веб-інтерфейс дозволяє адміністратору безпосередньо впливати на конфігурацію SIP-компонентів. Зокрема, через панель керування можна змінювати параметри облікових записів користувачів, налаштовувати правила маршрутизації дзвінків, активувати або деактивувати SIP-об’єкти в режимі реального часу, а також застосовувати зміни до параметрів шифрування без необхідності перезапуску SIP-сервера.

У першу чергу було реалізовано модулі для перегляду активних SIP-сесій, статусу реєстрації користувачів та моніторингу зашифрованих з’єднань. На рисунку 3.3 представлено головну панель адміністратора веб-інтерфейсу, де відображаються ключові показники в режимі реального часу: кількість зареєстрованих користувачів, правила маршрутизації дзвінків, параметри облікових записів користувачів. Інтерфейс реалізовано у вигляді дашборду, що дозволило досягти сучасного візуального оформлення та адаптивності для різних розмірів екрану.

Список існуючих SIP-користувачів

ID	Username	Password	Transport	Context	Allow	Max Contacts	Allowed IP
1001	1001	secret123	transport-tls	internal	ulaw,alaw	1	
1002	1002	secret1002	transport-udp	internal	alaw	1	192.168.1.100
1004	1004	pass1004!	transport-udp	internal	alaw	1	192.168.100.204
2001	2001	pass2001!	transport-udp	internal	alaw	1	192.168.100.201
5021	5021	pass5021!	transport-udp	internal	alaw	1	192.168.100.205
6001	6001	pass6001!	transport-udp	internal	alaw	1	192.168.100.202
6002	6002	pass6002!	transport-udp	internal	alaw	1	192.168.100.203

Рисунок 3.3 – Вигляд головної панель адміністратора

У межах інтерфейсу реалізовано модуль керування користувачами, який дозволяє додавати, редагувати або блокувати облікові записи, встановлювати політики автентифікації та контролювати права доступу до ресурсів системи.

Візуалізація відбувається через інтерактивні компоненти, що автоматично оновлюються, з відображенням стану кожного TLS-з’єднання та інформацією про відповідні сертифікати, їх строк дії та хеши. Це особливо важливо в контексті підтримки безперервного шифрування і швидкого реагування на проблеми в каналах захищеного зв’язку.

Інтерфейс панелі керування дозволяє також здійснювати перегляд поточних подій, у тому числі успішна авторизація, підозріла активність та небезпечні події (рис 3.4).

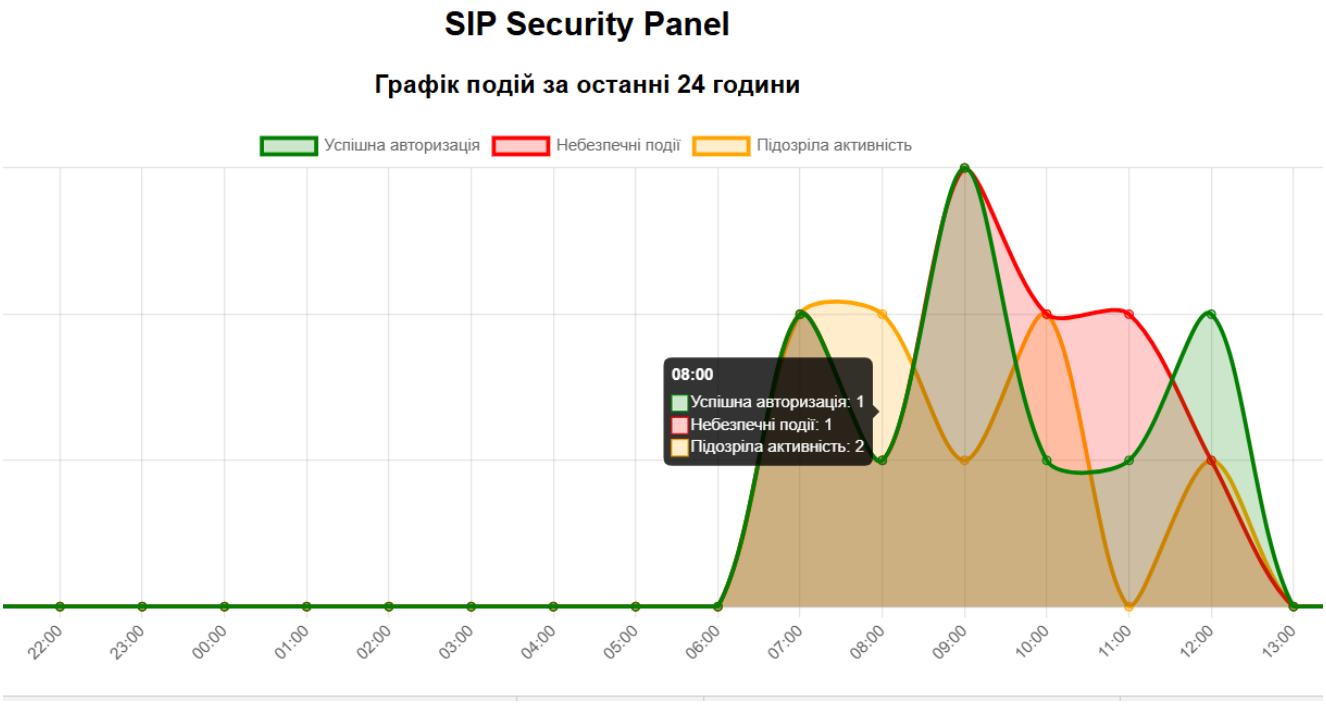


Рисунок 3.4 – Графік подій за останні 24 години

Крім спостереження, веб-інтерфейс також дозволяє виконувати керування обліковими записами та конфігурацією SIP-серверу. На рисунку 3.5 зображено форму редагування параметрів користувача, де адміністратор може змінити тип автентифікації, включити або вимкнути TLS-режим для конкретного облікового запису, а також задати політику шифрування голосового трафіку. Ці зміни зберігаються до бази RealTime без необхідності перезапуску сервісу.

```
$message = '';
if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['add_user'])) {
    $id = $_POST['id'];
    $username = $_POST['username'];
    $password = $_POST['password'];
    $context = $_POST['context'] ?? 'from-internal';
    $transport = $_POST['transport'] ?? 'transport-udp';
    if ($id && $username && $password) {
        $check = $pdo->prepare("SELECT id FROM ps_endpoints WHERE id = ?");
        $check->execute([$id]);
        if ($check->fetch()) {
            $message = "Користувач з таким ID вже існує.";
```

```
    } else {
        $pdo->prepare("INSERT INTO ps_auths (id, auth_type, username, password) VALUES (?,
'userpass', ?, ?)")
            ->execute([$id, $username, $password]);
      $pdo->prepare("INSERT INTO ps_aors (id, max_contacts) VALUES (?, 1)")
            ->execute([$id]);
        $pdo->prepare("INSERT INTO ps_endpoints (id, transport, aors, auth, context, disallow, allow,
direct_media)
  VALUES (?, ?, ?, ?, ?, 'all', 'ulaw,alaw', 'no')")  ->execute([$id, $transport, $id, $id, $context]); }
```

Рисунок 3.5 – Форма редагування параметрів користувача

Завдяки реалізованій перевірці у реальному часі адміністратор отримує сповіщення про дублювання імен користувачів, помилки в параметрах TLS-сертифікатів або конфлікти в налаштуваннях AOR (рис 3.6).

Після створення чи редагування облікового запису зміни автоматично записуються в таблиці ps_endpoints, ps_auths, ps_aors, а також – за потреби – у ps_endpoint_id_ips, що дозволяє забезпечити високий рівень динамічності та гнучкості керування.

В реалізації веб-інтерфейсу було зосереджено увагу на впровадженні функцій керування обліковими записами, налаштуванням політик безпеки, а також організації журналювання адміністративних дій. Особливістю даного модуля є його тісна інтеграція з базою даних RealTime, що забезпечує миттєве відображення

змін у конфігурації SIP-користувачів та їхніх атрибутів без необхідності ручного редагування конфігураційних файлів або перезапуску сервісів Asterisk.

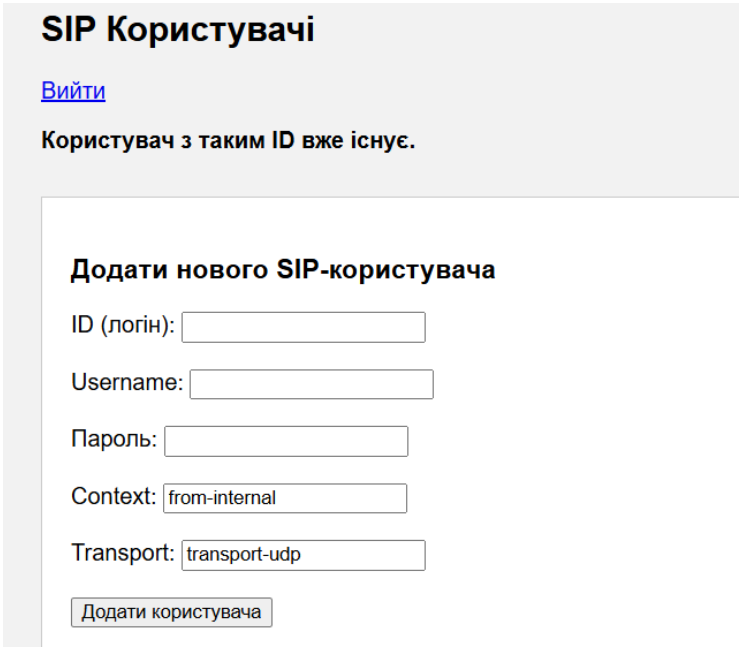


Рисунок 3.6 – сповіщення про некоректність даних

Для забезпечення безпечного доступу до самого веб-інтерфейсу впроваджено механізм автентифікації. Цей механізм забезпечує додатковий рівень захисту для адміністративного доступу, знижуючи ймовірність компрометації облікових записів навіть у випадку перехоплення основного пароля (рис. 3.7).

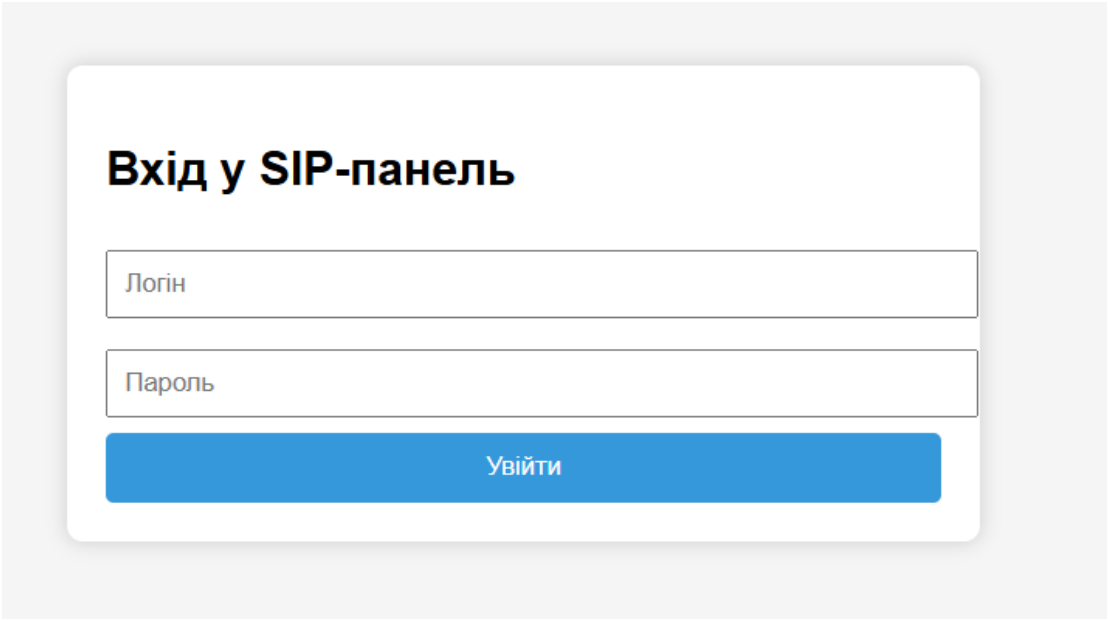


Рисунок 3.7 – Форма автентифікації до адміністративного доступу

Інтерфейс спроектовано таким чином, щоб максимально спростити роботу системного адміністратора або оператора безпеки. Через веб-інтерфейс користувач має змогу в реальному часі переглядати стан активних SIP-сесій, перевіряти автентифікацію користувачів, спостерігати за подіями, пов’язаними з безпекою, такими як спроби несанкціонованого доступу, а також контролювати стан TLS-з'єднань і використання сертифікатів.

Крім цього, веб-інтерфейс інтегровано з базою даних, яка зберігає історію всіх дій, транзакцій, спроб входу та інших важливих подій, що дозволяє проводити ретроспективний аналіз і виявляти потенційні шаблони загроз. Така аналітика є критично важливою для формування поведінкових профілів користувачів і подальшого автоматизованого виявлення аномалій.

3.4 Інструкція адміністратора та користувача системи SIP-моніторингу

Після завершення розгортання та налаштування програмно-апаратної частини системи захищеної SIP-телефонії виникає необхідність у створенні експлуатаційної документації, яка регламентує порядок взаємодії з елементами розробленої системи для двох категорій користувачів: адміністратора та кінцевого споживача послуг IP-телефонії. Зазначена інструкція визначає алгоритм дій під час виконання типових сценаріїв роботи із системою, а також містить опис основних інструментів адміністрування, механізмів контролю доступу та засобів моніторингу поточних подій.

Адміністратор системи, відповідальний за підтримку працездатності та безпеки SIP-інфраструктури, здійснює доступ до веб-інтерфейсу через захищене TLS-з’єднання. Така реалізація забезпечує додатковий рівень захисту при доступі до інтерфейсу керування.

Після успішного входу адміністратору доступна головна інформаційна панель системи, яка містить дані про активні SIP-сесії, поточні станції користувачів, журнал подій, а також індикатори стану криптографічного захисту з’єднань. Через відповідні розділи інтерфейсу адміністратор може виконувати базові операції, зокрема створення, редагування та блокування SIP-облікових

записів у режимі реального часу. При створенні нового користувача формується запис у таблицях бази даних ps_endpoints, ps_auths і ps_aors.

Окрему увагу приділено можливості оперативного керування політиками шифрування для окремих облікових записів. У процесі додавання або редагування користувача передбачено опцію обмеження доступу за IP-адресами, що забезпечує фільтрацію небажаного трафіку на прикладному рівні.

Також, увагу зосереджено на описі функціоналу кінцевого користувача, що взаємодіє з системою SIP-телефонії в межах захищеного середовища. Користувач отримує доступ до SIP-сервісів через надані облікові дані, які створюються адміністратором та зберігаються в базі даних у вигляді параметрів автентифікації, ідентифікатора SIP-клієнта, параметрів шифрування, а також зареєстрованих IP-адрес пристроїв.

Типовий процес взаємодії користувача з системою розпочинається із конфігурування SIP-клієнта (наприклад, Zoiper, Linphone, або IP-телефону), де необхідно вказати адресу SIP-сервера, порт TLS-з'єднання 5061, ім’я користувача (SIP URI), пароль та шлях до кореневого сертифіката, який необхідний для перевірки SSL-з’єднання. При коректному налаштуванні встановлюється зашифрований канал зв’язку, що гарантує захист трафіку від стороннього перехоплення.

З міркувань безпеки, користувач позбавлений можливості змінювати власні параметри без залучення адміністратора. Це стосується зокрема зміни пароля, IP-обмежень, сертифікатів чи параметрів SRTP. У разі потреби внести зміни, користувач звертається до адміністратора через інтегровану систему запитів, яка також доступна з веб-інтерфейсу.

3.5 Висновки до розділу 3

У третьому розділі було реалізовано практичну частину проєкту комплексної системи моніторингу та захисту SIP-телефонії. Згідно з проєктною архітектурою, налаштовано SIP-сервер на базі Asterisk із підтримкою технології Asterisk Realtime,

що дозволило забезпечити динамічне керування обліковими записами через базу даних.

Було впроваджено захист сигналізаційного трафіку за допомогою протоколу TLS у поєднанні з криптоалгоритмом RSA, що гарантує цілісність та конфіденційність передаваних даних. Для забезпечення захисту медіа-трафіку реалізовано шифрування голосу за допомогою протоколу SRTP.

Окремо реалізовано модуль моніторингу активності користувачів із веб-інтерфейсом, що дозволяє виявляти аномальну активність. Система забезпечує прозорість функціонування SIP-телефонії, а також дозволяє швидко реагувати на загрози або підозрілі дії завдяки інтеграції механізмів сповіщення.

ВИСНОВОК

У процесі виконання дипломної роботи розроблено та реалізовано комплексну систему моніторингу й захисту SIP-телефонії, орієнтовану на підвищення інформаційної безпеки в корпоративних і критичних інформаційних системах.

У теоретичній частині досліджено архітектуру SIP-протоколу, його вразливості та сучасні підходи до захисту VoIP-зв'язку. Особливу увагу приділено методам безпечної автентифікації, шифруванню сигналізаційного й медіа-трафіку через TLS і SRTP. Проаналізовано наявні рішення у сфері VoIP-безпеки, виявлено їхні недоліки, що дозволило обґрунтувати створення більш захищеної архітектури з динамічним керуванням обліковими записами.

У проєктній частині побудовано логічну архітектуру системи з урахуванням вимог до масштабованості, гнучкості та захищеності. Розроблено структуру бази даних Asterisk Realtime для динамічного керування SIP-акаунтами. Інтегровано криптографічні засоби захисту: TLS з RSA для сигналізації та SRTP для голосового трафіку. Обґрунтовано вибір сучасних інструментів безпеки.

У практичній частині реалізовано повний цикл впровадження SIP-сервера з Asterisk Realtime, налаштовано криптографічний захист, створено веб-інтерфейс для моніторингу активності користувачів. Система виявляє підозрілу активність і загрози в реальному часі, забезпечуючи підвищений контроль над інфраструктурою. Механізми сповіщення дають змогу оперативно реагувати на інциденти.

Загалом, результати дослідження підтверджують доцільність застосування комплексного підходу до захисту SIP-телефонії, який об'єднує криптографічні протоколи, динамічну автентифікацію, централізоване управління обліковими даними та інтерактивний моніторинг у єдину захищену інформаційну систему. Розроблене рішення може бути використане як основа для впровадження захищеного VoIP-зв'язку на підприємствах різних масштабів.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Abdul, J. Session Initiation Protocol (SIP) Deep Dive. UC Solution Architect, 2024. 17 p.

2. А. Б. Фещенко, Л. В. Борисова, В. О. Собина, Д. В. Тарадуда, М. О. Дement, І. М. Автоматизовані системи управління та зв'язок: підручник: НУЦЗУ, 2021. 288 с.

3. Черкасов Д. Основи технології VoIP та IP-телефонії. ТЕЛЕКОМ ВІЙСЬКОВИЙ ЗВ'ЯЗОК. 2020. № 2. С. 98-104.

4. Іващенко В.Л. Сучасні медіакомунікативні технології: навчальний посібник. Київ: ТОВ «Видавництво “Розумники”», 2021. 192 с.

5. Іващенко В.Л. Сучасні медіакомунікативні технології: навчальний посібник. Київ: ТОВ «Видавництво “Розумники”», 2021. 192 с.

6. Сабурова С.О., Новіченко Є.О. дослідження методів реалізації технології ip-телефонії на сервері cisco unified communications manager. Кафедра «Інфокомунікаційної інженерії ім. В.В. Поповського», Харківський національний університет радіоелектроніки, Україна. 2022. № 8. С. 10-13.

7. Варіс І., Кравчук О. Цифрова трансформація бізнесу: вибір, впровадження та вдосконалення crm-систем. Маркетинг і цифрові технології. 2021. № 5. С. 48-66.

8. І.Б. Гобир, Т.М. Паневник, С.А. Побігун, Л.С. Войтків, І.В. Мельничук, Х.В. Обельницька, В.Ю. Кравченко. Креативна економіка, діджиталізація бізнесу і сталий розвиток : методичні вказівки до теоретико-практичного курсу: Івано-Франківськ, 2023. 69 с.

9. Що таке SIP-телефонія та навіщо вона потрібна бізнесу?: веб–сайт. URL: https://streamtele.com/uk/shho-take-sip-telefoniya-ta-navishho-vona-potribna-biznesu (дата звернення 13.02.2025)

10. Все, що вам потрібно знати про SIP: веб–сайт. URL: https://blog.ringostat.com/uk/prostymy-slovamy-shcho-take-sip-telefoniia/ (дата звернення 13.02.2025)

11. Простими словами: що таке SIP-телефонія: веб–сайт. URL: https://www.oki-toki.ua/blog/vibir-sip-postachalnika (дата звернення 15.02.2025)

12. What is a SIP Protocol and How Does it Work?: веб–сайт. URL: https://getvoip.com/blog/sip-protocol/ (дата звернення 16.02.2025)

13. What is Session Initiation Protocol (SIP)? A Complete Guide: веб–сайт. URL: https://www.yeastar.com/blog/sip-protocol/ (дата звернення 16.02.2025)

14. SIP-телефонія: веб–сайт. URL: https://www.datagroup.ua/smb/telefoniya/sip-telefoniya (дата звернення 19.02.2025)

15. What's VoIP's Big Security Problem? SIP: веб–сайт. URL: https://www.pcmag.com/news/voips-big-security-problem-its-sip (дата звернення 20.02.2025)

16. Security Issues of SIP: веб–сайт. URL: https://www.diva-portal.org/smash/get/diva2:833435/FULLTEXT01.pdf (дата звернення 20.02.2025)

17. Security Enhancement of SIP Protocol in VoIP Communication: веб–сайт. URL: https://jicts.udsm.ac.tz/index.php/udsm/article/view/32/19 (дата звернення 25.02.2025)

18. How do you implement SIP authentication and encryption methods and what are the trade-offs?: веб–сайт. URL: https://www.linkedin.com/advice/3/how-do-you-implement-sip-authentication (дата звернення 26.03.2025)

19. Що таке SBC (Session Border Controller)?: веб–сайт. URL: https://www.euromobile.com.ua/sbc-session-border-controller-r (дата звернення 27.02.2025)

20. Стаття 28 Загального регламенту про захист даних (GDPR). Брюссель: Європейський парламент та Рада ЄС, 2016. 5 с.

21. ISO/IEC 27001. Інформаційні технології. Методи забезпечення безпеки. Системи управління інформаційною безпекою. Вимоги. Женева: Міжнародна організація зі стандартизації, 2022. 35 с.

22. How to Set Up an IPsec Connection with NAT (with SIP): веб–сайт. URL: https://www.ingate.com/appnotes/Ingate_Setting_Up_IPsec_With_NAT-en.pdf (дата звернення 05.03.2025)

23. FreePBX 2.5 Powerful Telephony Solutions: веб–сайт. URL: https://senatelecom.com/wp-content/uploads/2024/08/FreePBX-2.5.pdf (дата звернення 20.03.2025)

24. InstructionsFreePBX IPAuthentication: веб–сайт. URL: https://portal.bulkvs.com/assets/images/Instructions-FreePBX-Bulkvs-IPAuthentication.pdf (дата звернення 20.03.2025)

25. 3CX. Innovating Communications: веб–сайт. URL: https://www.tdlsystems.com/_webedit/uploaded-files/All%20Files/3CX-Brochure-lo-res.pdf (дата звернення 20.03.2025)

26. 3CX PHONE SYSTEM: веб–сайт. URL: http://www.telephonesandcables.com/documents/3cx.pdf (дата звернення 20.03.2025)

27. FusionPBX Documentation: веб–сайт. URL: https://app.readthedocs.org/projects/fusionpbx-docs-pt-br/downloads/pdf/latest/ (дата звернення 20.03.2025)

28. Asterisk - The Basics: веб–сайт. URL: https://www.pacnog.org/pacnog6/VoIP/PacNOG6_voip_asterisk_basic.pdf (дата звернення 22.03.2025)

29. Asterisk The Definitive Guide: веб–сайт. URL: https://asterisk-service.com/downloads/Asterisk-The-Definitive-Guide-Edition.pdf (дата звернення 22.03.2025)

30. Asterisk & Dnake User Manual: веб–сайт. URL: https://www.dnake-global.com/uploads/Asterisk-DNAKE-User-Manual.pdf (дата звернення 22.03.2025)

31. Fighting SIP Trunk Hacking: веб–сайт. URL: https://www.channelfutures.com/unified-communications/fighting-sip-trunk-hacking (дата звернення 25.03.2025)

32. Гапак О. М. Посібник з курсу «Компютерна криптографія» призначено для студентів інженерно-технічного факультету ДВНЗ «УжНУ» спеціальності 123-«комп'ютерна інженерія»: навч. посіб. для студентів Ужгород. 2021. 93с.

33. Customizing TLS Protocol Version: веб–сайт. URL: https://community.rsa.com/s/article/Customizing-TLS-Protocol-Version-089c86f4 (дата звернення 25.03.2025)

34. Openssl: веб–сайт. URL: https://www.openssl.org/ (дата звернення 27.03.2025)

35. Svitlana S., Volodymyr M. Overview: PHP and MySQL Features for Creating Modern Web Projects. International Journal of Academic Information Systems Research. 2023. № 7. С. 11-17.

36. Web Development with Php and MySQL: веб–сайт. URL: https://edc.iba-suk.edu.pk/assets/course%20outlines/wd.pdf (дата звернення 27.03.2025)

ДОДАТКИ

Додаток А. Технічне завдання

Вінницький національний технічний університет

Факультет менеджменту та інформаційної безпеки

Кафедра менеджменту та безпеки інформаційних систем

ЗАТВЕРДЖУЮ

Голова секції "Управління інформаційною безпекою" кафедри МБІС
д.т.н., професор

_____ Юрій ЯРЕМЧУК
" 20 " березня 2025 р.

ТЕХНІЧНЕ ЗАВДАННЯ

до бакалаврської кваліфікаційної роботи на тему:
Розробка комплексної системи моніторингу та захисту SIP-телефонії на основі динамічного контролю сесій автентифікації та інтеграції протоколу TLS з криптоалгоритмом RSA
08-72.БКР. 024.00.090 ТЗ

Керівник бакалаврської кваліфікаційної роботи
д.ф.(PhD), доц. каф. МБІС
Салієва О.В.
" _____ " _____

Вінниця – 2025 р.

1. Найменування та область застосування

Програмний засіб для комплексного захисту SIP-телефонії на основі динамічного контролю сесій автентифікації та шифрування сигналізації.

Область застосування: захист VoIP-інфраструктури в інформаційно-телекомунікаційних системах шляхом моніторингу поведінки користувачів, виявлення аномальної активності, блокування загроз і забезпечення конфіденційності трафіку через шифрування.

2.Підстава для розробки

Розробка виконується на основі наказу ректора ВНТУ № 97 від 20.03.2025 р.

3. Мета та призначення розробки

3.1 Мета: розробка програмного засобу для підвищення рівня безпеки SIP-телефонії через впровадження динамічного контролю сесій автентифікації, поведінкового аналізу та шифрування TLS/RSA.

3.2 Призначення: розроблений програмний засіб забезпечує виявлення та фіксацію аномальної активності, блокування потенційних загроз, шифрування SIP-сигналізації та голосового трафіку з метою захисту IP-телефонії від атак і несанкціонованого доступу

4. Джерела розробки

4.1 What is a SIP Protocol and How Does it Work?: веб–сайт. URL: https://getvoip.com/blog/sip-protocol/ (дата звернення 16.02.2025)

4.2 6. Сабурова С.О., Новіченко Є.О. Дослідження методів реалізації технології ip-телефонії на сервері cisco unified communications manager. Кафедра «Інфокомунікаційної інженерії ім. В.В. Поповського», Харківський національний університет радіоелектроніки, Україна. 2022. № 8. С. 10-13.

4.3 Asterisk - The Basics: веб–сайт. URL: https://www.pacnog.org/pacnog6/VoIP/PacNOG6_voip_asterisk_basic.pdf (дата звернення 22.03.2025)

4.4 Svitlana S., Volodymyr M. Overview: PHP and MySQL Features for Creating Modern Web Projects. International Journal of Academic Information Systems Research. 2023. № 7. С. 11-17.

5. Вимоги до програми

5.1 Вимоги до функціональних характеристик:

5.1.1 Програмний засіб повинен мати зручний веб-інтерфейс адміністратора для перегляду подій безпеки та керування блокуваннями;

5.1.2 Реалізація не повинна вимагати спеціального ліцензійного програмного забезпечення;

5.1.3 Програмний засіб повинен здійснювати динамічний контроль сесій автентифікації та виявляти підозрілу активність користувачів.

5.2 Вимоги до надійності:

5.2.1 Програмний засіб повинен стабільно функціонувати в реальному часі та у випадку помилок виводити інформативні повідомлення;

5.2.2 База даних подій безпеки повинна підтримувати регулярне створення резервних копій;

5.2.3 Програмний засіб повинен коректно виконувати моніторинг, аналіз, блокування загроз і логування подій.

5.3 Вимоги до складу і параметрів технічних засобів:

– процесор – не нижче Intel Core i3 або аналогічний;

– оперативна пам’ять – не менше 2 ГБ;

– середовище функціонування – серверна ОС (Linux, наприклад Ubuntu) з підтримкою PHP, MariaDB, Asterisk, OpenSSL;

– вимоги до техніки безпеки при роботі з програмою повинні відповідати чинним стандартам безпеки інформаційних систем.

6. Вимоги до програмної документації

6.1 Обов’язкова поетапна інструкція для майбутніх користувачів, наведена у пункті 3.4.

7. Вимоги до технічного захисту інформації

7.1 Необхідно забезпечити захист програмного засобу від несанкціонованого використання через систему автентифікації;

7.2 Неможливість доступу до функцій моніторингу та керування без авторизованого входу в адміністративну панель.

8. Техніко-економічні показники

8.1 Цінність результатів використання даного проекту повинна перевищувати витрати на його реалізацію;

8.2 Програмний засіб має бути сумісним з поширеним серверним та клієнтським програмним забезпеченням, що забезпечує можливість його інтеграції в існуючу інфраструктуру без додаткових витрат.

9. Стадії та етапи розробки

№ з/п	Назва етапів бакалаврської кваліфікаційної роботи	Початок	Закінчення
1.	Визначення напрямку бакалаврської роботи, формулювання теми	20.03.2025	23.03.2025
2.	Аналіз предметної області обраної теми	24.03.2025	13.04.2025
3.	Розробка алгоритму роботи	14.04.2025	27.04.2025
4.	Написання бакалаврської роботи на основі розробленої теми	28.04.2025	25.05.2025
5.	Передзахист бакалаврської кваліфікаційної роботи	26.05.2025	27.05.2025
6.	Виправлення, уточнення, корегування бакалаврської дипломної роботи	28.05.2025	02.06.2025
7.	Захист бакалаврської кваліфікаційної роботи	09.06.2025	10.06.2025

10. Порядок контролю та прийому

10.1 До приймання бакалаврської кваліфікаційної роботи надається:

– ПЗ до бакалаврської кваліфікаційної роботи;

- демонстрація результату бакалаврської кваліфікаційної роботи;
- презентація;
- відзив керівника роботи;
- відзив рецензента

Технічне завдання до виконання прийняв _____ Фененко В.О.

Додаток Б. Лістинг коду серверної частини системи захисту SIP-телефонії

```
<?php
date_default_timezone_set('Europe/Kyiv');
require_once "db.php";

$log_file = '/var/log/asterisk/full';
$threshold = 5;
$interval = 60; // секунд
$ip_attempts = [];
$now = time();

function get_geolocation($ip) {
    return [
        'country' => 'UA',
        'region' => 'Kyiv',
        'city' => 'Kyiv'
    ];
}

function get_ip_history($mysqli, $ip) {
    $stmt = $mysqli->prepare("SELECT COUNT(*) as events, MAX(timestamp) as last_seen FROM
security_events WHERE ip_address = ?");
    $stmt->bind_param("s", $ip);
    $stmt->execute();
    $stmt->bind_result($events, $last_seen);
    $stmt->fetch();
    $stmt->close();

    $geo_data = get_stored_geo($ip);

    return [
        'events' => $events,
        'last_seen' => $last_seen,
        'geo_data' => $geo_data
    ];
}

function log_event($mysqli, $ip, $type, $details) {
    $now = date('Y-m-d H:i:s');
    $stmt = $mysqli->prepare("INSERT INTO security_events (event_time, ip_address, event_type,
details) VALUES (?, ?, ?, ?)");
    $stmt->bind_param("ssss", $now, $ip, $type, $details);
    $stmt->execute();
    $stmt->close();
}

function block_ip($ip, $reason, $mysqli) {
```

```
    $stmt = $mysqli->prepare("SELECT COUNT(*) FROM ip_block_list WHERE ip_address = ? AND
status = 'blocked'");
    $stmt->bind_param("s", $ip);
    $stmt->execute();
    $stmt->bind_result($exists);
    $stmt->fetch();
    $stmt->close();

    if ($exists == 0) {
        $now = date('Y-m-d H:i:s');
        $stmt = $mysqli->prepare("INSERT INTO ip_block_list (ip_address, reason, timestamp, status)
VALUES (?, ?, ?, 'blocked')");
        $stmt->bind_param("sss", $ip, $reason, $now);
        $stmt->execute();
        $stmt->close();

        log_event($mysqli, $ip, "block_ip", $reason);
    }}

$lines = file($log_file, FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);
$lines = array_slice($lines, -1000); // останні 1000 рядків

foreach ($lines as $line) {
    $timestamp = strtotime(substr($line, 0, 15));

    if (preg_match('/(SIP\/\d\.\d)\s+(\w+)\s+.*from\s+(\d{1,3}(?:\.\d{1,3}){3})/', $line, $match)) {
        $proto = $match[1];
        $action = $match[2];
        $ip = $match[3];

        if (($now - $timestamp) <= $interval) {
            if (stripos($line, 'authentication failed') !== false || stripos($line, 'failed') !== false) {
                if (!isset($ip_attempts[$ip])) $ip_attempts[$ip] = 0;
                $ip_attempts[$ip]++;
            }

$is_tls = (stripos($line, '5061') !== false || stripos($line, 'TLS') !== false);

            preg_match('/User-Agent:\s*(.+)/i', $line, $ua_match);
            $user_agent = $ua_match[1] ?? 'Unknown';

            preg_match('/(SIP\/\d\.\d)\s+(\w+)\s+(\d+)@/', $line, $login_match);
            $login = $login_match[3] ?? 'unknown';

            $geo = get_geolocation($ip);
            $geo_str = "{$geo['country']}, {$geo['region']}, {$geo['city']}";

            $history = get_ip_history($mysqli, $ip);
            if ($history['events'] > 0 && $history['last_seen']) {
```

```
            $geo_anomaly = false;
            if ($geo_anomaly) {
               log_event($mysqli, $ip, 'geo_anomaly', "New region for same login: $login from
$geo_str");
            }   }

         if (!$is_tls && stripos($line, '5061') !== false) {
            log_event($mysqli, $ip, 'unencrypted_tls_connection', "Unencrypted connection to TLS
port");}

         $suspicious_agents = ['sipvicious', 'friendly-scanner', 'iWar'];
         foreach ($suspicious_agents as $sus_ua) {
            if (stripos($user_agent, $sus_ua) !== false) {
               log_event($mysqli, $ip, 'scan_detected', "Suspicious User-Agent: $user_agent");
               break;
            } } } }}

foreach ($ip_attempts as $ip => $count) {
   if ($count >= $threshold) {
      block_ip($ip, "Excessive auth failures: $count attempts in $interval sec", $mysqli);
      log_event($mysqli, $ip, 'failed_auth_threshold', "Detected $count failed attempts in $interval
sec");
   }
}
?>
```

Додаток В. Лістинг коду адміністративної панелі безпеки SIP-системи

```
<?php
//sip_security_panel
$host = 'localhost';
$dbname = 'sip_security';
$username = 'sipadmin';
$password = 'password';

try {
   $pdo = new PDO("mysql:host=$host;dbname=$dbname;charset=utf8mb4", $username,
$password);
   $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
} catch (PDOException $e) {
   die("DB connection failed: " . $e->getMessage());
}

if (isset($_GET['toggle_status'])) {
   $id = intval($_GET['toggle_status']);
   $stmt = $pdo->prepare("SELECT status, ip_address FROM ip_block_list WHERE id = ?");
   $stmt->execute([$id]);
   $row = $stmt->fetch(PDO::FETCH_ASSOC);

   if ($row) {
      $new_status = ($row['status'] === 'blocked') ? 'allowed' : 'blocked';
      shell_exec("sudo iptables " . ($new_status === 'blocked' ? "-A" : "-D") . " INPUT -s
{$row['ip_address']} -j DROP");

      $update = $pdo->prepare("UPDATE ip_block_list SET status = ? WHERE id = ?");
      $update->execute([$new_status, $id]);
   }

   header("Location: panel.php");
   exit;
}

if (isset($_GET['delete'])) {
   $id = intval($_GET['delete']);
   $stmt = $pdo->prepare("SELECT ip_address FROM ip_block_list WHERE id = ?");
   $stmt->execute([$id]);
   $row = $stmt->fetch(PDO::FETCH_ASSOC);

   if ($row) {
      shell_exec("sudo iptables -D INPUT -s {$row['ip_address']} -j DROP");
      $delete = $pdo->prepare("DELETE FROM ip_block_list WHERE id = ?");
      $delete->execute([$id]);
   }
```

```
    header("Location: panel.php");
    exit;
}

$stmt = $pdo->query("SELECT * FROM ip_block_list ORDER BY timestamp DESC");
$ipList = $stmt->fetchAll(PDO::FETCH_ASSOC);

$eventTypes = ['auth_success', 'failed_auth', 'scan_detected'];
$labels = [];

for ($i = 23; $i >= 0; $i--) {
    $labels[] = date('H:00', strtotime("-$i hour"));
}

$eventData = [];
foreach ($eventTypes as $type) {
    foreach ($labels as $hour) {
        $eventData[$type][$hour] = 0;
    }
}

$stmt = $pdo->query("
    SELECT event_type, DATE_FORMAT(event_time, '%H:00') AS hour_slot, COUNT(*) AS total
    FROM security_events
    WHERE event_time >= NOW() - INTERVAL 1 DAY
    GROUP BY event_type, hour_slot
");

foreach ($stmt as $row) {
    $type = $row['event_type'];
    $hour = $row['hour_slot'];
    $count = $row['total'];

    if (isset($eventData[$type][$hour])) {
        $eventData[$type][$hour] = (int)$count;}}

$chartDatasets = [];
foreach ($eventTypes as $type) {
    $chartDatasets[$type] = array_values($eventData[$type]);}
?>
<!DOCTYPE html>
<html>
<head>
    <title>SIP Security Panel</title>
    <style>
        body { font-family: Arial; }
        table { width: 90%; border-collapse: collapse; margin: 20px auto; }
        th, td { border: 1px solid #ccc; padding: 10px; text-align: center; }
        th { background-color: #f2f2f2; }
```

```
    h2, h3 { text-align: center; }
    .btn { padding: 5px 10px; margin: 2px; text-decoration: none; }
    .blocked { background-color: #f44336; color: white; }
    .allowed { background-color: #4CAF50; color: white; }
    canvas { max-width: 100%; height: 400px; }
  </style>
</head>
<body>
<h2>SIP Security Panel</h2>
<h3>Графік подій за останні 24 години</h3>

<div style="width: 95%; margin: auto;">
  <canvas id="attackChart"></canvas>
</div>

<table>
  <tr>
    <th>IP Address</th>
    <th>Reason</th>
    <th>Status</th>
    <th>Timestamp</th>
    <th>Actions</th>
  </tr>
  <?php foreach ($ipList as $row): ?>
    <tr>
      <td><?= htmlspecialchars($row['ip_address']) ?></td>
      <td><?= htmlspecialchars($row['reason']) ?></td>
      <td><?= $row['status'] ?></td>
      <td><?= $row['timestamp'] ?></td>
      <td>
        <a class="btn <?= $row['status'] === 'blocked' ? 'allowed' : 'blocked' ?>"
          href="?toggle_status=<?= $row['id'] ?>">
          <?= $row['status'] === 'blocked' ? 'Allow' : 'Block' ?>
        </a>
        <a class="btn" href="?delete=<?= $row['id'] ?>" onclick="return confirm('Delete this
IP?');">Delete</a>
      </td>
    </tr>
  <?php endforeach; ?>
</table>
<script>
  const ctx = document.getElementById('attackChart').getContext('2d');
  const chart = new Chart(ctx, {
    type: 'line',
    data: {
      labels: <?= json_encode($labels) ?>,
      datasets: [
        {
          label: 'Успішна авторизація',
```

```
            data: <?= json_encode($chartDatasets['auth_success']) ?>,
            borderColor: 'green',
            backgroundColor: 'rgba(0, 128, 0, 0.2)',
            fill: true,
            tension: 0.3
          },
          {
            label: 'Небезпечні події',
            data: <?= json_encode($chartDatasets['failed_auth']) ?>,
            borderColor: 'red',
            backgroundColor: 'rgba(255, 0, 0, 0.2)',
            fill: true,
            tension: 0.3
          },
          {
            label: 'Підозріла активність',
            data: <?= json_encode($chartDatasets['scan_detected']) ?>,
            borderColor: 'orange',
            backgroundColor: 'rgba(255, 165, 0, 0.2)',
            fill: true,
            tension: 0.3
          }]},
      options: {
        responsive: true,
        maintainAspectRatio: false,
        scales: {
          y: {
            beginAtZero: true,
            ticks: { stepSize: 1 }
          },
          x: {
            ticks: {
              autoSkip: false,
              maxRotation: 90,
              minRotation: 45
            }}},
        plugins: {
          legend: { position: 'top' },
          tooltip: { mode: 'index', intersect: false }
        }} });
</script>

</body>
</html>

// user verification
<?php
session_start();
if (isset($_SESSION['logged_in']) && $_SESSION['logged_in'] === true) {
```

```
    header('Location: panel.php');
    exit();
}

define('USERNAME', 'admin');
define('PASSWORD', 'admin123');

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $login = $_POST['username'] ?? '';
    $pass = $_POST['password'] ?? '';

    if ($login === USERNAME && $pass === PASSWORD) {
        $_SESSION['logged_in'] = true;
        header('Location: panel.php');
        exit();
    } else {
        $error = "Невірний логін або пароль.";
    }
}
?>

<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Авторизація</title>
    <style>
        body { font-family: sans-serif; background: #f5f5f5; display: flex; justify-content: center; align-
items: center; height: 100vh; }
        .login-form { background: #fff; padding: 20px; border-radius: 8px; box-shadow: 0 0 10px #ccc; }
        input[type="text"], input[type="password"] { width: 100%; padding: 8px; margin: 8px 0; }
        input[type="submit"] { width: 100%; padding: 10px; background: #3498db; border: none; color:
white; border-radius: 4px; cursor: pointer; }
    </style>
</head>
<body>
    <form class="login-form" method="POST">
        <h2>Вхід у SIP-панель</h2>
        <?php if (!empty($error)) echo "<p style='color:red;'>$error</p>"; ?>
        <input type="text" name="username" placeholder="Логін" required>
        <input type="password" name="password" placeholder="Пароль" required>
        <input type="submit" value="Увійти">
    </form>
</body>
</html>

// sip control panel
<?php
session_start();
```

```
if (!isset($_SESSION['logged_in'])) {
   header('Location: index.php');
   exit;
}
require_once 'db.php';

$message = '';
if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['add_user'])) {
   $id = $_POST['id'];
   $username = $_POST['username'];
   $password = $_POST['password'];
   $context = $_POST['context'] ?? 'from-internal';
   $transport = $_POST['transport'] ?? 'transport-udp';

   if ($id && $username && $password) {
      $check = $pdo->prepare("SELECT id FROM ps_endpoints WHERE id = ?");
      $check->execute([$id]);
      if ($check->fetch()) {
         $message = "Користувач з таким ID вже існує.";
      } else {

         $pdo->prepare("INSERT INTO ps_auths (id, auth_type, username, password) VALUES (?,
'userpass', ?, ?)")
            ->execute([$id, $username, $password]);

         $pdo->prepare("INSERT INTO ps_aors (id, max_contacts) VALUES (?, 1)")
            ->execute([$id]);

         $pdo->prepare("INSERT INTO ps_endpoints (id, transport, aors, auth, context, disallow, allow,
direct_media)
                  VALUES (?, ?, ?, ?, ?, 'all', 'ulaw,alaw', 'no')")
            ->execute([$id, $transport, $id, $id, $context]);

         $message = " Користувача $id успішно додано!";}
   } else {
      $message = "Заповніть всі поля.";
   }}

$sql = "SELECT
         e.id AS endpoint_id,
         e.transport,
         e.context,
         e.allow,
         a.username,
         a.password,
         ao.max_contacts,
         ip.match_ip
      FROM ps_endpoints e
      LEFT JOIN ps_auths a ON e.auth = a.id
```

```
    LEFT JOIN ps_aors ao ON e.aors = ao.id
    LEFT JOIN ps_endpoint_id_ips ip ON e.id = ip.endpoint";
$stmt = $pdo->query($sql);
$users = $stmt->fetchAll(PDO::FETCH_ASSOC);
?>

<!DOCTYPE html>
<html>
<head>
  <title>SIP Панель керування</title>
  <style>
    body { font-family: Arial; background: #f2f2f2; padding: 20px; }
    table { width: 100%; border-collapse: collapse; background: #fff; margin-top: 20px; }
    th, td { padding: 8px 12px; border: 1px solid #ccc; }
    th { background: #eee; }
    form { margin-top: 30px; background: #fff; padding: 20px; border: 1px solid #ccc; }
    .message { margin-top: 10px; font-weight: bold; }
  </style>
</head>
<body>
  <h2>SIP Користувачі</h2>
  <p><a href="logout.php">Вийти</a></p>

  <?php if ($message): ?>
    <div class="message"><?= htmlspecialchars($message) ?></div>
  <?php endif; ?>

  <form method="post">
    <input type="hidden" name="add_user" value="1">
    <h3>Додати нового SIP-користувача</h3>
    <label>ID (логін): <input type="text" name="id" required></label><br><br>
    <label>Username: <input type="text" name="username" required></label><br><br>
    <label>Пароль: <input type="text" name="password" required></label><br><br>
    <label>Context: <input type="text" name="context" value="from-internal"></label><br><br>
    <label>Transport: <input type="text" name="transport" value="transport-
udp"></label><br><br>
    <button type="submit">Додати користувача</button>
  </form>

  <h3>Список існуючих SIP-користувачів</h3>
  <table>
    <thead>
      <tr>
        <th>ID</th>
        <th>Username</th>
        <th>Password</th>
        <th>Transport</th>
        <th>Context</th>
        <th>Allow</th>
```

```
            <th>Max Contacts</th>
            <th>Allowed IP</th>
          </tr>
        </thead>
        <tbody>
          <?php foreach ($users as $user): ?>
            <tr>
              <td><?= htmlspecialchars($user['endpoint_id']) ?></td>
              <td><?= htmlspecialchars($user['username']) ?></td>
              <td><?= htmlspecialchars($user['password']) ?></td>
              <td><?= htmlspecialchars($user['transport']) ?></td>
              <td><?= htmlspecialchars($user['context']) ?></td>
              <td><?= htmlspecialchars($user['allow']) ?></td>
              <td><?= htmlspecialchars($user['max_contacts']) ?></td>
              <td><?= htmlspecialchars($user['match_ip']) ?></td>
            </tr>
          <?php endforeach; ?>
        </tbody>
      </table>
    </body>
    </html>
```

Додаток Г. Ілюстративний матеріал

РОЗРОБКА КОМПЛЕКСНОЇ СИСТЕМИ МОНІТОРИНГУ ТА ЗАХИСТУ SIP-ТЕЛЕФОНІЇ НА ОСНОВІ ДИНАМІЧНОГО КОНТРОЛЮ СЕСІЙ АВТЕНТИФІКАЦІЇ ТА ІНТЕГРАЦІЇ ПРОТОКОЛУ TLS З КРИПТОАЛГОРИТМОМ RSA

Виконав студент групи 1КІТС-216:
Фененко В.В.
Керівникд.ф.(PhD), доц. каф. МБІС:
Салієва О.В

Актуальність, Мета, завдання

Актуальність. VoIP-телефонія, зокрема на базі SIP-протоколу, стала основою корпоративних комунікацій, проте супроводжується зростанням кіберзагроз — від перехоплення трафіку до brute-force атак. Існуючі засоби захисту часто не забезпечують гнучкого контролю в реальному часі, тому актуальним є впровадження TLS і SRTP з RSA та динамічного контролю автентифікаційних сесій для своєчасного виявлення й блокування несанкціонованої активності.

Мета. Розробка системи захисту SIP-телефонії з динамічним контролем сесій автентифікації та інтеграцією TLS із RSA для підвищення безпеки зв'язку.

Основне завдання. Дослідити методи захисту SIP-телефонії, розробити алгоритм динамічного контролю автентифікації з використанням шифрування TLS та SRTP

ПРОТОКОЛ SIP: ПРИНЦИПИ, ПЕРСПЕКТИВИ ТА ОБМЕЖЕННЯ

Протокол Session Initiation Protocol (SIP) – це сигнальний протокол, який використовується для ініціювання, підтримки, модифікації та завершення сеансів зв'язку в реальному часі між пристроями Інтернет-протоколу (IP)

SIP працює за моделлю клієнт-сервер, де сервери (реєстрації, проксі, перенаправлення) маршрутизують запити, автентифікують і обслуговують клієнтів, а кінцеві пристрої (IP-телефони, софтфони) ініціюють виклики. Основні запити SIP — INVITE, REGISTER, ACK, BYE, CANCEL, MESSAGE — керують сигналізацією та мультимедійними сесіями. Головна задача SIP — керувати сигналізацією, тобто ініціацією, зміною та завершенням сеансів, використовуючи текстові запити і відповіді для зручності аналізу та інтеграції.

СУЧАСНІ ЗАГРОЗИ ДЛЯ SIP-ТЕЛЕФОНІЇ

01 **Перехоплення RTP-трафіку**

RTP, який використовується для передачі голосу, за замовчуванням передає трафік у відкритому вигляд

02 **Brute-force атака на автентифікацію**

SIP використовує просту Digest-автентифікацію, легко піддається атакам перебору пароля

03 **Man-in-the-Middle (MitM)**

Перехоплення SIP і RTP між клієнтом і сервером із можливістю модифікації

04 **Spoofing (Caller ID підміна)**

Зловмисник надсилає SIP-запит із фальшивим Caller ID

05 **DoS / DDoS**

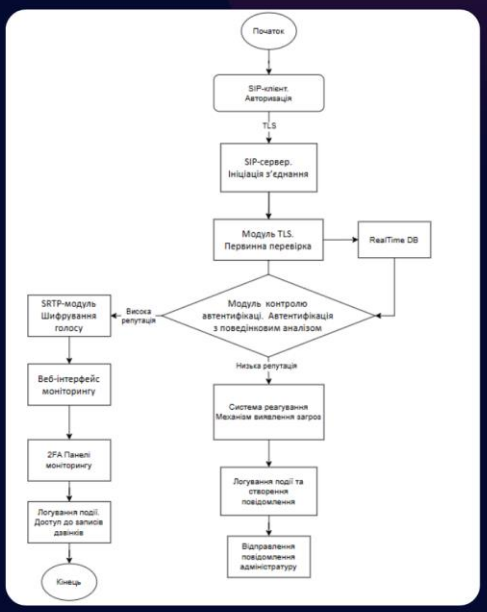
Засипання сервера великою кількістю INVITE або REGISTER-запитів

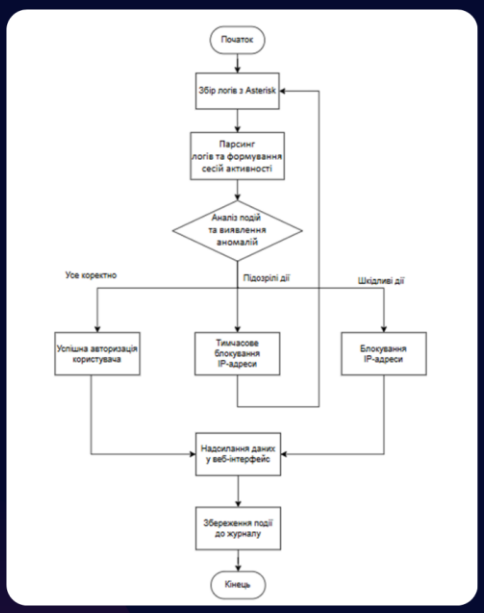
ІНТЕГРАЦІЯ TLS З КРИПТОАЛГОРИТМОМ RSA

Інтеграція TLS із RSA у SIP-телефонії має кілька важливих переваг. По-перше, це надійний захист від атак типу "людина посередині" (MITM), оскільки перевірка сертифікатів унеможливлює підміну учасників з'єднання. По-друге, шифрування трафіку захищає від перехоплення даних, включаючи автентифікаційні дані, що є критичним для запобігання несанкціонованому доступу до SIP-системи. По-третє, використання TLS забезпечує сумісність із більшістю сучасних SIP-серверів і клієнтів, таких як Asterisk, FreeSWITCH або Cisco CallManager.

АРХІТЕКТУРА СИСТЕМИ МОНІТОРИНГУ

SIP-запит спочатку надходить до SIP-сервера, який передає інформацію модулю TLS для перевірки шифрування. Далі система автентифікації звіряє облікові дані з RealTime-базою, аналізує активність користувача та приймає рішення про допуск до сеансу. Якщо автентифікація пройшла успішно, система ініціює захищене голосове з'єднання через SRTP. Усі події реєструються у базі, а інформація про поточні сесії відображається у веб-інтерфейсі.



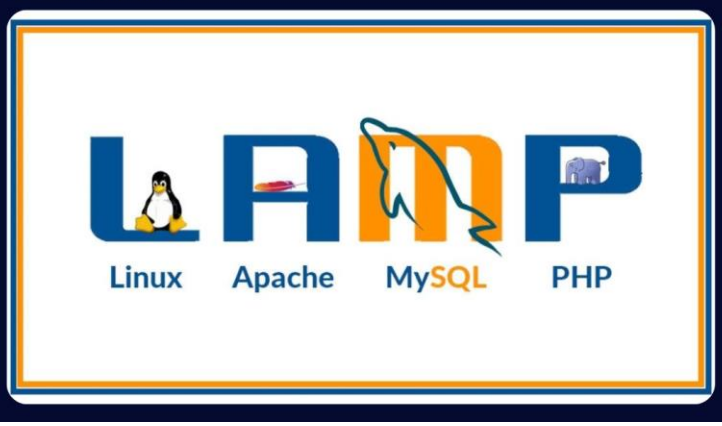


МОДУЛЬ КОНТРОЛЮ АВТЕНТИФІКАЦІЇ

Модуль автоматично збирає дані про спроби автентифікації, типи User-Agent, частоту запитів та інші ознаки можливої атаки. За перевищенням порогів активності визначається рівень загрози. Якщо виявлено підозрілу IP-адресу, її додають до списку блокування, оновлюють правила фаєрволу та виконують дії в Asterisk.

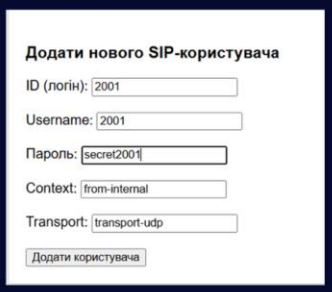
Технології та середовище розробки системи

Для створення веб-інтерфейсу управління системою та моніторингу було використано стек LAMP (Linux, Apache, MySQL, PHP), а також додаткові інструменти, такі як Bootstrap для побудови адаптивного інтерфейсу та Chart.js для візуалізації статистичних даних.

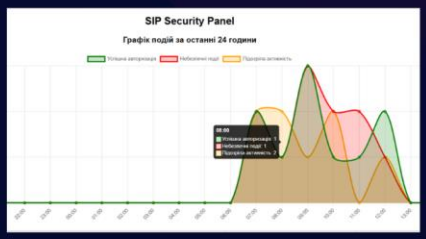


ІНТЕРФЕЙС АДМІНІСТРАТОРА

Головна панель адміністратора веб-інтерфейсу, де відображаються ключові показники в режимі реального часу



Форма редагування параметрів користувача



Інтерфейс панелі керування дозволяє також здійснювати перегляд поточних подій, у тому числі успішна авторизація, підозріла активність та небезпечні події

ДЯКУЮЗАУВАГУ!

Додаток Д. Протокол перевірки на антиплагіат

90

Додаток Д. Протокол перевірки на антиплагіат

ПРОТОКОЛ ПЕРЕВІРКИ НАВЧАЛЬНОЇ (КВАЛІФІКАЦІЙНОЇ) РОБОТИ

Назва роботи:

Розробка комплексної системи моніторингу та захисту SIP-телефонії на основі динамічного контролю сесій автентифікації та інтеграції протоколу TLS з криптоалгоритмом RSA

Тип роботи: бакалаврська кваліфікаційна робота

Підрозділ Кафедра менеджменту на безпеки інформаційних систем
Факультет менеджменту та інформаційної безпеки
Гр. 1КІТС-21б

Керівник доцент Салієва О.В. _____

Показники звіту подібності

Strike Plagiarism	
Оригінальність	99,73%
Загальна схожість	0,27%

Аналіз звіту подібності (відмітити потрібне)

- ☐ **Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.**
- ☐ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її автора. Роботу направити на доопрацювання.
- ☐ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Заявляю, що ознайомлений (-на) з повним звітом подібності, який був згенерований Системою щодо роботи (додається)

Автор _____ Фененко В.О.
(підпис) (прізвище, ініціали)

Опис прийнятого рішення

☐ **Допустити до захисту**

Особа, відповідальна за перевірку _____ Коваль Н.П.
(підпис) (прізвище, ініціали)

Експерт (за потреби) _____ _____
(підпис) (прізвище, ініціали, посада)