МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

КАФЕДРА КОМП'ЮТЕРНИХ НАУК

ВИПУСКНА РОБОТА

на тему:

«Програмний комплекс для моніторингу процесів інформаційно-комунікаційної системи»

Завідувач випускаючої кафедри — **Довбиш А.С.**

Керівник роботи — **Лаврик Т.В.**

Студента групи КБ – 61 — **Горбась І.В.**

СУМИ 2020

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

Кафедра комп'ютерних наук

Затверджую_____

Зав. кафедрою Довбиш А.С.

"_____" _____2020 р.

ЗАВДАННЯ
до випускної роботи

Студента четвертого курсу, групи КБ-61 спеціальності "Кібербезпека" денної форми навчання Горбася Івана Володимировича.

Тема: " Програмний комплекс для моніторингу процесів інформаційно-комунікаційної системи "

Затверджена наказом по СумДУ

№_____ від _____2020 р.

Зміст пояснювальної записки: 1) аналітичний огляд існуючих систем моніторингу та їх порівняння ; 2) постановка задачі роботи; 3) опис технології мережевої телефонії , логіка робота з апаратного та програмного боку; 4) проектування моделі програмного модуля для відстеження показника затримок між послідовними пакетами; 5) порівняння технологій для реалізації та безпеки комплекса; 6) інструкція для адміністратора .

Дата видачі завдання “_____”_____2020 г.

Керівник випускної роботи _____ Лаврик Т.В.

Завдання приняв до виконання _____ Горбась І.В.

РЕФЕРАТ

Записка: 49 стор., 16 рис., 19 джерел.

Об'єкт дослідження – параметр затримки послідовних пакетів IP-мережі.

Мета роботи – розроблення програмного модуля для моніторингу процесів на сервері VoIP телефонії Askozia на базі Zabbix, який буде реалізовувати збір даних з серверу телефонії, їх обробку та аналіз.

Методи дослідження – метод аналітичного огляду, метод порівняння та аналогій, метод моделювання, методи розробки, що базуються на мові програмування bash.

Результати – розроблено програмний комплекс , який вирішує проблему відстеження якості телефонного зв’язку у ip-мережі для серверу ip-телефонії Askozia , для машин з операційної системи без системної оболонки , отримує поточні значення параметру затримки послідовних пакетів (Jitter) , налаштування сповіщень при критичному рівні Jitter , при цьому безпечно передаючи дані користувацьким AMI інтерфейсом.

СИСТЕМА МОНІТОРИНГУ ПОКАЗНИКІВ І ПРОЦЕССІВ ,
МЕРЕЖЕВА ТЕЛЕФОНІЯ, ZABBIX,LINUX БЕЗ ОБОЛОНКИ,
IP-АТС ASTERISK, РОЗРОБКА НА BASH, БЕЗПЕКА
ПЕРЕДАЧІ.

Зміст

ВСТУП

Дана робота присвячена ознайомленню з системою моніторингу та відстеження показників різноманітних сервісів комп'ютерної мережі, серверів та мережевого обладнання – ZABBIX, розробка програмного коду на скриптовій мові програмування bash .

У роботі присутній розгляд проблеми, яка полягає у неможливості відстеження параметрів зв’язку на сервері VoIP телефонії розгорнутої на базі Asterisk. На сервері Askozia , розгорнутому на операційній системі лінукс без оболонки , не дає можливості встановити клієнтську частину Zabbix – Zabbix agent. Відсутність оболонки на сервері істотно обмежує можливості користувача.

Тому, дана дипломна робота присвячується вирішенню проблеми відстеження якості зв’язку на сервері телефонії на базі Asterisk з обмеженими можливостями операційної системи за допомогою системи моніторингу ZABBIX.

1. АНАЛІЗ СИСТЕМ МОНІТОРИНГУ РЕСУРСІВ І ПОДІЙ В ІКТ СИСТЕМАХ

1.1 Загальні характеристики систем моніторингу

Однією з найважливіших частин інформаційної інфраструктури сучасних підприємств і багатьох державних організацій є корпоративні мережі передачі даних, які давно перейшли в розряд критичних для забезпечення бізнес-процесів ресурсів і вихід з ладу такої системи фактично означає зупинку діяльності всієї організації.

Корпоративні мережі мають високу складність в силу територіальної розподіленості інфраструктури, поєднання можливостей власне передачі даних з можливостями телефонії та відеоконференцзв'язку, наявності вбудованих систем підтримки інформаційної безпеки, а також резервних і дублюючих елементів, що відповідають за забезпечення надійності та доступності корпоративної мережі. Для державних структур так само, а, може бути, і значно актуальніше.

В першу чергу, підтримка мережевої інфраструктури вимагає великого штату кваліфікованих фахівців, в обов'язки яких входить підтримка доступності мережевих сервісів, оптимізація роботи мережі, налагодження та оновлення мережевого апаратного і програмного забезпечення, виявлення і усунення виникаючих інцидентів і інші аналогічні функції. При цьому, виникають такі проблеми:

- Зазначені фахівці повинні володіти навичками роботи з цілою низкою мережевих пристроїв і технологій, що підвищує вимоги до рівня підготовки співробітників.
- Рівень оплати робить непростим завдання залучення подібних фахівців на роботу в організаціях.
- Для держструктур існують бюджетні та нормативні обмеження, в силу яких загальне число найманого персоналу не може вийти за певні рамки.

Для вирішення більшої частини описаних проблем використовуються системи моніторингу та управління мережею, що потребують супроводження значно меншою кількістю фахівців[2].

Система моніторингу – це сукупність програмно-апаратних засобів, що здійснюють постійне спостереження і збір інформації в локальній обчислювальній мережі на основі аналізу статистичних даних з метою виявлення вузлів, що несправні або некоректно працюють, і оповіщення відповідальних осіб[1].

Функціонал сучасних систем моніторингу дозволяє відстежувати стан таких сервісів, як, наприклад: доступність хосту (шляхом періодичної відправки запитів ICMP Echo-Request на адресу мережевого пристрою); доступність веб серверу (шляхом відправлення HTTP запиту на отримання сторінки); доступність поштових сервісів (шляхом періодичної відправки діагностичних SMTP повідомлень). Крім того, можна виробляти заміри часу відгуку даних сервісів. Періодичні перевірки такого роду дозволяють швидко визначити на якому рівні виникла проблема і негайно приступити до її усунення.

Системи моніторингу за ціллю відстеження поділяють на:

- моніторинг мереж;
- моніторинг серверів і робочих станцій;
- моніторинг додатків і сервісів;
- моніторинг бізнес-процесів.

Компонування реалізацій усіх компонентів відстеження в один програмний додаток називається комплексною системою моніторингу .

Постачальники даної послуги виділяють 2 рівня IT моніторингу:

1. IT інфраструктури;
2. IT сервісів.

Системи моніторингу IT інфраструктури призначені для контролю над працездатністю наступних компонентів: мережеве та серверне обладнання,

бізнес програмне забезпечення . Під контролем програми моніторингу повинні знаходиться групи об'єктів, інформація про які необхідна адміністраторам[3].

Впровадження комплексної системи моніторингу IT допомагає підприємству:

- знизити час простою компонентів IT структури;
- збільшити доступність програм для бізнесу;
- здійснювати проактивний аналіз проблем;
- підвищити рівень продуктивності використання інформаційних ресурсів.

Системи моніторингу IT сервісів орієнтовані, в першу чергу, на показники ступеня доступності, а також якості надання сервісів на основі оцінок. У процесі створення системи відбувається формування каталогу IT сервісів. Визначаються показники доступності та рівня якості кожного сервісу і його залежність від інших компонентів інформаційної структури компанії. Система проводить моніторинг IT компонентів і формує показники роботи сервісів. Моніторинг IT систем корисний системним адміністраторам, IT керівникам і менеджерам IT сервісів.

Система моніторингу IT сервісів допомагає компанії:

- зробити більш високим рівень доступності сервісів IT;
- знизити витрати на їх підтримку;
- збільшити ефективність роботи IT персоналу і якості обслуговування.

Впровадження автоматизованої системи моніторингу IT і контроль роботи інформаційної інфраструктури здатне підвищити рівень якості її функціонування за допомогою швидкого виявлення та ліквідації збоїв і несправностей, а також запобігання їх виникненню в майбутньому, в першу чергу, для найбільш критичних для бізнесу компанії сервісів.

З боку проектування в процесі побудови комплексної системи моніторингу застосовують 2 підходи:

1. Підхід від інфраструктури . Цей підхід передбачає організацію спостереження за основними апаратними та програмними компонентами з

налаштуванням окремих консолей для виконання завдань різних адміністраторів на основі їх спеціалізації. Головною метою є допомога IT фахівцям в оперативному виявленні і ліквідації проблем, що з'являються при функціонуванні IT структури.

2. Підхід від IT сервісів. Застосування даного підходу полягає в формуванні каталогу послуг і відповідає методології сервісного підходу до управління IT (ITSM). Передбачається, що для кожного сервісу (послуги) повинна бути розроблена своя сервісно-ресурсна модель, що відображає взаємодію між сервісом та іншими компонентами інфраструктури, потрібними для його роботи. З використанням сервісно-ресурсної моделі проводиться процедура налаштування програми моніторингу з метою контролю функціонування IT сервісу і всіх пов'язаних з ним компонентів інфраструктури. Цей підхід сприяє тому, що системні консолі стають корисними не тільки за рахунок підтримки певних сервісів IT-фахівцями, а й диспетчерській службі, а також керівництву IT відділу.

Використання подібних систем дозволяє організації здійснювати активний моніторинг доступності, стану і продуктивності компонентів корпоративної / відомчої мережі передачі даних, аналізувати і оптимізувати їх завантаження, а також прогнозувати виникнення нештатних ситуацій. Дрібні збої, що становлять більшу частину інцидентів, за допомогою систем моніторингу та управління мережею виправляються або зовсім автоматично, або за допомогою віддаленого адміністратора. Що ж стосується великих інцидентів, то зазначені системи саме і призначені для їх недопущення або, в крайньому випадку, оперативного усунення. Для цього використовуються функції аналізу зібраної статистики та прогнозування, а також локалізації справжніх причин інцидентів. Крім того, при наявності в мережевій інфраструктурі дублюючих або резервних блоків, система просто перерозподіляє навантаження між ними, забезпечуючи «невидимість» інциденту для кінцевих користувачів[2].

Перевагами впровадження систем моніторингу та управління для бізнесу є:

- підвищення ефективності та оперативності роботи співробітників;
- підвищення рівня надійності та інформаційної безпеки IT-інфраструктури;
- зниження негативного впливу відмов системи за рахунок оперативного усунення причин аварій і їх попередження;
- централізованої координації роботи IT-інфраструктури;
- скорочення витрат на супровід IT-інфраструктури за рахунок автоматизації ряду процесів.

Впровадження систем моніторингу та управління IT-інфраструктурою особливо актуально, коли є потреба в:

- оптимізації використання інформаційних ресурсів;
- підвищенні якості IT-сервісів і швидкості усунення збоїв в роботі ПЗ і апаратної частини;
- забезпеченні надійності, безпеки і узгодженого функціонування всіх компонентів IT-інфраструктури;
- полегшенні модернізації IT-інфраструктури;
- оперативному усуненні або локалізації збоїв в IT-системі;
- активному моніторингу і запобігання збоїв.

1.2 Огляд існуючих програмних рішень

Розвиток телекомунікаційних та мережевих технологій сприяв стрімкому розвитку систем моніторингу. Розглянемо найбільш поширені системи моніторингу.

OpenNMS

OpenNMS – програмний open source проект для моніторингу та управління корпоративною мережею підприємства.

Передбачається як «платформа програмних рішень для управління мережею»[4]. Система OpenNMS відповідає за моніторинг функціонуючих в мережевій інфраструктурі сервісів, таких як Web, DNS, DHCP, сервіси СУБД

(Oracle, MSSQL, PostgreSQL та ін.), Проте інформація про стан мережевих пристроїв також доступна.

Функціональні можливості та переваги полягають у тому , що дану систему можна:

- Використовувати за допомогою XML сценаріїв автоматизування процесів, або користуватися підтримкою XML через веб-інтерфейс користувача.
- Створює звіти високого рівня з бази даних і збирає дані про продуктивність.
- Функції забезпечення якості обслуговування OpenNMS дозволяють визначати доступність мережевих послуг (ping , відправка ICMP-запитів, тощо)[5]. На основі результатів цих запитів можна сформувати статистику чи діаграми.
- Збори відомостей дійсні за допомогою ряду протоколів , таких як: SNMP, HTTP , JMX , WMI , XMP, XML, NSClient и JDBC.
- Зрозумілий графічний інтерфейс на базі Eclipse Jetty.

Zabbix

Zabbix – програмний open source продукт для моніторингу мережевих пристроїв та параметрів мережі. Однією з головних переваг Zabbix є різноспрямований спосіб сигналізування проблем, що, врешті-решт, дозволить оперативно реагувати на можливий збій.

Моніторинг у Zabbix працює на клієнтсько-серверній архітектурі. Серверна частина , окрім термінального способу адміністрування оснащена й веб-інтерфейсом. За його допомогою можна легше й більш ефективно адмініструвати корпоративну мережу підприємства. Клієнтська частина комплексу моніторингу реалізовується за рахунок Zabbix-агентів. Агент розгортатиметься на сервері, параметри і стан якого відстежуватиметься [6].

Функціонал програмного комплексу забезпечує:

- Клієнт-серверна архітектура з підтримкою на багатьох ОС.
- Серверна частина доступна для розгортання на Linux, Solaris, HP-UX, AIX, FreeBSD, OpenBSD, OS X.

- Можливий моніторинг без агентів (ICMP , HTTP-запити для перевірки доступності або виконання скриптів з серверу Zabbix).
- Гнучка система сповіщень.
- Журнал аудиту.
- Open Source проект.
- Зручний веб-інтерфейс.
- Централізованість.

Nagios

Nagios – open source рішення, призначене для моніторингу комп'ютерних систем і мереж: спостереження, контролю стану обчислювальних вузлів і служб, оповіщення системних адміністраторів у разі припинення або відновлення роботи будь-яких служб [11].

Крім безкоштовної версії Nagios Core, існує платна версія Nagios XI з додатковими можливостями і має більш сучасний та простий в навігації web-інтерфейс, який пропонує інтерактивну інформаційну панель з оглядом хостів, сервісів і мережевих пристроїв.

Використання Nagios у мережі підприємства дозволяє:

- відстежувати мережеві служби SMTP, POP3, HTTP, NNTP, ICMP, SNMP;
- проводити моніторинг хостів в більшості ОС;
- підтримувати віддалений моніторинг за допомогою тунелів SSH і SSL;
- підтримувати паралельні служб;
- відправляти повідомлення при виникненні проблем і неполадок в роботі служби або хоста;
- автоматичну ротацію лог-файлів;
- утиліта nagiostats робить звіт по хостам, за якими проводиться моніторинг;
- огранізувати роботу відразу декількох систем моніторингу для підвищення загальної безпеки.

Graylog

Graylog – платформа з відкритим вихідним кодом для централізованого збору, зберігання, аналізу даних в локальних мережах.

Розробник [7] рекомендує перед встановленням спланувати журнал подій. Для цього необхідно визначитися зі "стратегією" використання журналів, з тим, які логи необхідні, як збирати, хто буде мати до них доступ, як і скільки зберігати.

В операційній системі за кожну секунду часу відбувається занадто багато подій, щоб записувати усі й централізовано зберігати в одному місці. Це потребує багато ресурсів у вигляді потужності центрального процесора і його багатопотоковості, швидкості та обсягу оперативної пам’яті, а також обсяг вільного місця на жорсткому диску. Навантаження збільшується чисельно під час моніторингу не одного , а декількох серверів[9].

Варіантів використання може бути безліч, тому якість моніторингу залежить лише від кваліфікованого спеціаліста, що адмініструє . Тому головне – правильно розпоряджатися існуючими ресурсами.

Головні переваги і особливості Graylog:

- Ресурсоємкий комплекс на базі MongoDB.
- Агрегація повідомлень в потоки.
- Агрегація хостів в групи. Можна об'єднати потоки з різних хостів в одну групу.
- Блеклист для логів.
- Авторотація логів. Очистка старих логів відбувається автоматично за рахунок механізму capped collections.
- OpenSource.
- Діаграми, таблиці зі статистикою (pie chart, area chart).

Splunk

Слідкувати за системою можна також за допомогою онлайн сервісів таких, як Splunk. Splunk Enterprise – це продукт, який здатен фіксувати, індексувати та корелювати дані в реальному часі у сховищі, де можна шукати.

За даними відомостями можна генерувати графіки, звіти, сповіщення, інформаційні панелі та візуалізації [10].

Сервер Splunk зберігає логи користувачів та за допомогою мови запитів SPL можна працювати з полями даних логів, тобто сортувати, агрегувати, формувати нові таблиці тощо. За допомогою даного сервісу можна створювати панелі моніторинга (dashboard-и) та аналізувати трафік. Однак, є й недоліки. Зокрема, як заявляє сам Splunk, усі логи, що зберігаються в системі за будь-який час, є доступними для запитів, тобто немає поняття архівування [8].

Система Splunk пропонує:

- здійснювати збір, пошук, моніторинг та аналіз за різними і досить великим (сотні Тб даних в день) обсягами даних в режимі реального часу.
- Splunk є універсальною платформою для машинних даних, яка забезпечує комплексний збір даних, їх обробку та аналіз. Таким чином, можливо індексувати будь-які машинні дані з відміткою про час незалежно від структури й формату.
- Splunk здійснює пошуки за часом, тобто адміністратору не потрібно заздалегідь знати структуру даних, щоб сформувати запит.
- Розгортання відбувається за короткий період часу.
- Splunk безкоштовна, але є й додаткові можливості за додаткову плату.

1.3 Постановка задачі

З розвитком IT-технологій у світі розвивається і людство цілком. Масштабність розповсюдження девайсів і мережевих технологій тепер зрозуміле і більш доступне для кожного. Ще 40 років тому люди користувалися телефонними апаратами і це була дійсно рідкість, а про комп’ютер можна було прочитати хіба що в газеті чи почути по радіо. Зараз же кожен має смартфон, комп’ютер, розумний годинник тощо. Багато процесів відбувається в житті, коли бачимо лише вершину айсбергу того, що відбувається насправді, і за легкими, на перший погляд, подіями ховається

тисячі прийнятих рішень. З’являються нові сфери в житті людини, нові професії, нові можливості, а з ними і велика відповідальність. Відповідальність за майно, свободу, особисті дані, врешті-решт за власне життя. Комп’ютерні технології настільки швидко і міцно стали пов’язані з повсякденним життям, що інше, мабуть і представити важко. Саме тому так важливо займатися мережевим моніторингом процесів і ресурсів систем.

Існує багато способів, щоб утримати актуальність показників життєдіяльності ланок IT-інфрастуктури . Не існує одного правильного і універсального рішення, яке б підійшло для усіх можливих ситуацій, мало би зрозумілий веб-інтерфейс з повним доступом таким же як у консолі, не ресурсномісткий, а ще й безкоштовний. Кожна система має свої переваги й недоліки, але для системи з операційною системою на чистому Linux, без системної оболонки, на якому розгорнута IP АТС Askozia для розгортання корпоративного серверу IP телефонії найбільш доречним рішенням є Zabbix.

В основі операційної системи лежить ядро Linux без оболонки, а це свідчить про неможливість роботи навіть з терміналом як засобом для конфігурацій, що вже казати про встановлення додаткових агентів систем моніторингу.

Тому ***метою роботи*** є розроблення програмного модуля для моніторингу процесів на сервері VoIP телефонії Askozia на базі Zabbix, який буде реалізовувати такі функціональні завдання:

- відстеження якості зв’язку;
- відстеження параметру Jitter на вході і виході до серверу;
- обмін даними сервера IP-телефонії Askozia та серверу збирача та моніторингу логів та подій Zabbix;
- вирішення проблеми неможливості встановлення Zabbix–agent на операційну систему Linux без системної оболонки.

2. ОПИС VOIP ТЕХНОЛОГІЇ

2.1 Технологічний опис

Використання та історична причина появи технології VoIP обумовлена, по-перше, високим попитом на мобільний зв’язок. За ті функції, що надає VoIP майже безкоштовно , за виключенням потреби у відповідному апаратному забезпеченні, в традиційних стільникових мережах коштують чималих грошей, а це вже друга причина.

В основі будь-якої схеми (мережі) IP-телефонії лежить мережевий комутатор (Switch). Всі пристрої (вузли мережі) IP-телефони, шлюзи, IP-АТС, персональні комп'ютери тощо з'єднуються за допомогою даного комутатора.

Мережевий комутатор (Switch) – це пристрій, призначений для об’єднання декількох вузлів локальної комп’ютерної мережі. Даний пристрій працює на канальному рівні еталонної моделі побудови мережі OSI. Оскільки IP-телефонія накладає певні вимоги (Quality of service) на комп'ютерну (локальну) мережу, виділимо параметри мережевого комутатора, на які варто звернути увагу при проектуванні мережі IP-телефонії:

- Комутаційна матриця (відповідає за швидкість передачі в мережі між вузлами).
- Кількість портів.
- PoE (Технологія, що дозволяє передавати віддаленого пристрою електричну енергію разом з даними, через стандартну виту пару в мережі Ethernet).
- VLAN.

Для організації телефонного зв'язку за допомогою IP-мереж використовується спеціальне обладнання – шлюзи IP-телефонії. Таким чином, при наборі номеру абонента йде з’єднання із мережевим шлюзом. Голос, зчитаний за допомогою мікрофона, трансформується в електричний аналоговий сигнал, який далі кодується. У шлюзі IP-телефонії сигнал оцифровується та стискається, а за допомогою спеціального кодеку відбувається поділ на пакети , які невдовзі відпраляються мережею. З іншого

боку такий же шлюз декодує отриманий набір пакетів та відбувається перетворення їх в початкову форму. За рахунок того, що передача відбувається за вкрай короткий час, реалізується дуплексний спосіб передачі сигналу [12] див. Рисунок 2.1.

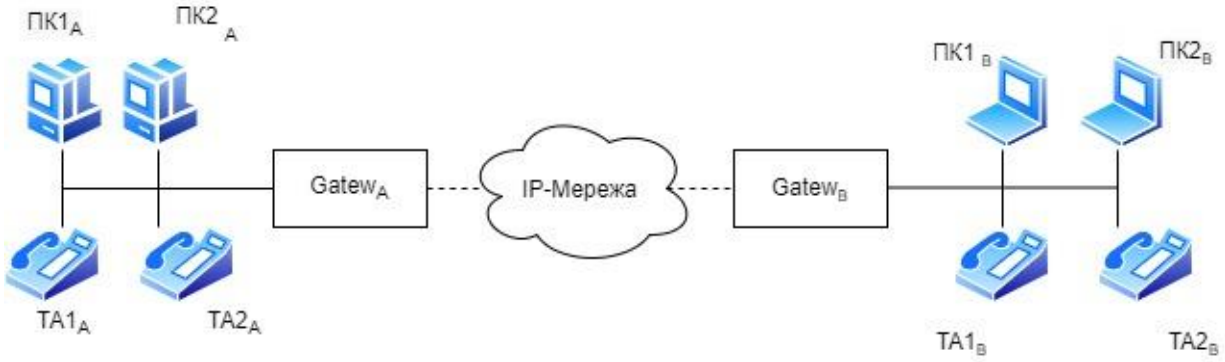

Рисунок 2.1 – Приклад VoIP мережі

Де $ПК1_A$ – перший персональний комп'ютер абонента А, $TA1_A$ – перший телефонний апарат абонента А, $Gatew_A$ – шлюз абонента А, у підмережі В аналогічно.

Загальний принцип використання шлюза IP-телефонії полягає в можливості одночасної підтримки зв'язку з аналоговими телефонними лініями і за рахунок підключення до IP-мережі має змогу зв'язатися з будь-яким комп'ютером у глобальній мережі. Справа в тому, що шлюзи IP-телефонії можуть зв'язуватися не тільки через Інтернет, а й через виділену високошвидкісну мережу з комутацією комірок чи кадрів (ATM, Frame Relay), що забезпечує гарантовану якість обслуговування. Використання мережі з низькими і передбачуваними затримками забезпечить якіснішу передачу голосу, але таке рішення дорожче за звичайну Інтернет-телефонію.

Голосовий зв'язок через IP-мережу може здійснюватися різними способами. Розглядають такі чотири способи:

1. «Комп'ютер - комп'ютер». Даний варіант не є прикладом IP-телефонії, так як голос передається тільки мережею передачі даних, без виходу в телефонну мережу. Для організації передачі трафіку користувач набуває необхідне обладнання і програмне забезпечення, а також платить

провайдерові за експлуатацію каналу зв'язку. Гідність цього варіанту полягає в максимальній економії коштів. Недолік - мінімальна якість зв'язку.

2. «Телефон - Телефон». Для організації такого зв'язку необхідна наявність певних мережевих пристроїв і механізмів взаємодії. Голосовий трафік передається через IP-мережу, як правило, на окремій дорогій ділянці. Пристроями, що організують взаємодію, є шлюзи, з одного боку підключені з телефонною мережею загального користування, а з іншого - з IP-мережею. Голосовий зв'язок в такому режимі, в порівнянні з варіантом «комп'ютер - комп'ютер», коштує дорожче, проте якість її значно вище і користуватися нею зручніше. Для того, щоб скористатися цією послугою, треба подзвонити провайдеру, який обслуговує шлюз, ввести з телефонного апарату код і номер абонента, який викликається і розмовляти так само, як при звичайному телефонному зв'язку. Всі необхідні операції з маршрутизації виклику виконає шлюз.

3. «Комп'ютер - телефон». У цьому способі відкривається більше можливостей використання для корпоративних користувачів, так як найчастіше застосовується корпоративна мережа, яка обслуговує виклики від комп'ютерів до шлюзу, які вже потім передаються телефонною мережею загального користування. Корпоративні рішення з використанням зв'язку «комп'ютер-телефон» можуть допомогти заощадити гроші. Кінцевому користувачу ніякого додаткового обладнання не потрібно.

4. «WEB - телефон». Це послуга, яку надають провайдери IP-телефонії. Це дзвінок з веб-сайту або Surf & Call. Рішення компанії VocalTec в області веб-телефонії, що дозволяє здійснювати виклик, обравши зі сторінки Інтернет посилання на ім'я абонента, що викликається. Таке рішення спрямоване, насамперед, на розширення можливостей електронної комерції. Surf & Call дозволяє користувачам Інтернет безпосередньо поговорити, наприклад, з торговим представником або з фахівцем технічної підтримки, що цікавить його фірми. Встановлення телефонного з'єднання відбувається при натисканні курсором на посилання, що представляє собою, наприклад, назву компанії,

ім'я абонента, який викликається тощо. При цьому користувачеві не потрібна друга телефонна лінія або переривання роботи в Інтернет, необхідно лише завантажити невелике клієнтське програмне забезпечення, яке зазвичай можна знайти на тій же WEB-сторінці, і яке встановлюється автоматично. З іншого боку, Surf & Call дозволяє представникам компаній відповідати на питання, демонструвати WEB-сторінки, передавати необхідну інформацію, покращуючи тим самим якість послуг, що надаються [13].

Щодо пристроїв кінцевих споживачів зв’язку, то в мережах IP-телефонії це IP-телефони. Основна відмінність між аналоговим стаціонарним апаратом, це використання в якості середовища для передачі голосу IP-мережу. Телефон підключається до комп'ютерної мережі як комп'ютер за допомогою коннектора RJ-45.

Для зв’язку типу «Комп’ютер» - «Комп’ютер» або «Комп’ютер» - «IP-телефон» у реалізації з боку комп'ютера використовують програмний телефон (SoftPhone). У схемі (системі) IP-телефонії внутрішні абоненти не прив'язані до робочого місця і можуть бути як віддаленими операторами (співробітниками), так і перебувати всередині офісу.

З боку станції IP-АТС підключаються зовнішні лінії за допомогою VOIP шлюзів, що виконують перетворення аналогових FXO сигналів, GSM SIM карт та цифрових потоків E1.

2.2 Алгоритми і логіка роботи VoIP мереж

VoIP (Voice over Internet Protocol) або IP телефонія - це технологія, що дозволяє використовувати мережу Інтернет для ведення телефонних розмов і факсування в режимі реального часу. Гарантує доступність для організації з невеликим бюджетом, оскільки з економічної точки зору, використання даної технології для здійснення міжміських і міжнародних телефонних розмов або для створення розподілених корпоративних телефонних мереж є більш вигідним та раціональним рішенням [14].

Технологія VoIP розгорнута на підприємстві гарантує зменшити витрати на засоби та послуги зв’язку абонентів, що пов’язане з веденням міжнародних та міжміських розмов, а також почати процес міграції до технологій пакетної передачі мультимедійних даних. Технологія значно покращує якість зв’язку в порівнянні з попередніми версіями рішень IP-телефонії, що характеризується такими показниками як: рівень спотворення голосу; частота «зникнення» голосових пакетів; час затримки (між проголошенням фрази першого абонента і моментом, коли вона буде почута іншим).

Також ігнорується вірогідність зайнятої лінії. При неможливості абонента відповісти на телефонний дзвінок VoIP-технологія дає змогу переадресації на сервер IP-телефонії, перетворення її в пакети та спрямувати на персональний комп’ютер абонента. Якщо об’єднати віддалені підрозділи до локальної віртуальної мережі VLAN за допомогою VPN, то можна не лише організувати доступ до спільних мережевих ресурсів, а й організувати безпечну передачу голосу захищеною IP- мережею.

Загальна структура та принцип роботи класичної VoIP мережі полягає в наступному: один з абонентів передає голосові сигнали іншому абоненту, його голос проходить обробку за допомогою кодеків і пересилається через Інтернет пакетними даними в режимі реального часу. При цьому, максимальна затримка звуку становить приблизно 300-400 мілісекунд в залежності від того, скільки часу потрібно апаратному обладнанню для створення цифрового аудіосигналу. Оскільки в даний час існують технології, що дозволяють звести втрати сигналу в мережі до мінімуму і уникнути зникнення голосу, то користувач цього і не помітить. В результаті за цю розмову він заплатить набагато менше, ніж заплатив би за звичайні телекомунікації. Для передачі сигналу необхідні спеціальні пристрої - IP-шлюзи. Це пристрої, за допомогою яких здійснюється трансляція даних з одного типу мережі в мережі іншого типу. IP-шлюзи або як їх ще називають IP-сервери, з одного боку пов'язані з телефонними лініями і вдається з'єднатися з будь-яким телефоном світу, а з

іншого – з мережею Інтернет, за рахунок чого можуть зв'язуватися з будь-яким, приєднаним до інтернету комп'ютером [13].

Значна частина роботи мережевої телефонії покладена на третій мережевий рівень моделі OSI. Основним протоколом, що реалізує функції мережевого рівня, є протокол IP. Він відповідає за маршрутизацію, фрагментацію і збірку дейтаграм в робочій станції [12].

IP-протокол відповідає за дейтаграмну передачу інформації від одного абонента іншому. Саме дейтаграмну, бо передача мовного сигналу відбувається за протоколом UDP на відміну від TCP. Використання даного протоколу обумовлена вимогами мережевої телефонії. Формування маршрутів для передачі дейтаграм мережею відбувається автоматично за рахунок протоколів динамічної маршрутизації RIP , OSPF, BGP. На мережевому рівні також працює протокол контрольних повідомлень ICMP. За його допомогою реалізовано обмін службовою інформацією між програмним забезпеченням робочої станції чи маршрутизатора з іншими вузлами мережі.

Проходячи мережею, дейтаграма може бути пошкоджена або загублена. За її надійну передачу відповідає транспортний рівень, а саме, протоколи TCP, UDP та RTP. Додатки з передачі файлів використовують FTP, веб-додатки – HTTP, а усі вони базуються на TCP[1].

Користувацьке програмне забезпечення працює на прикладному рівні моделі OSI. Softphone, web-interface це усе об’єкти реалізації 7 рівня моделі.

Перша вимога системи VoIP – це протокол контролю сеансів для встановлення присутності та пошуку користувачів, а також налаштування, зміни та припинення сеансів. Сьогодні широко використовуються два протоколи. Історично використовувався протокол H.323, але протокол ініціації сеансу (SIP) швидко стає головним стандартом.

Протокол SIP

SIP визначається спеціальною групою Internet Engineering (IETF) за RFC 3261. Він був розроблений спеціально для IP-телефонії та інших Інтернет-

сервісів. Хоча він багато в чому перекриває H.323, його зазвичай вважають більш спрощеним рішенням [13].

SIP використовується з протоколом опису сеансу (SDP) для виявлення користувача; він забезпечує узгодження функцій та управління дзвінками. SDP по суті є форматом для опису параметрів ініціалізації для потокового носія під час оголошення сесії та запрошення. Пара SIP / SDP дещо аналогічна протоколу H.225 / H.245, встановленому стандартом H.323.

SIP можна використовувати в системі, що має лише дві кінцеві точки та відсутність серверної інфраструктури. Однак, у загальнодоступній мережі спеціальні сервери проксі та реєстраторів використовуються для встановлення з'єднань. У такому режимі кожен клієнт реєструється на сервері, щоб дозволити абонентам знайти його з будь-якої точки Інтернету.

Протокол H.323

H.323 – стандарт MCE, спочатку розроблений для мультимедійних конференцій у реальному часі (голосових та відеозаписів) та додаткової передачі даних. Він швидко розвивався, щоб відповідати вимогам мереж VoIP. Технічно є контейнером для багатьох стандартів мережевих та медіа-кодеків. Частина сигналізації з'єднання H.323 обробляється протоколом H.225, тоді як узгодження функції підтримується H.245.

Транспортний протокол у режимі реального часу (RTP)

RTP надає послуги з доставки пакетів аудіо та відео в режимі реального часу. Це стандартний спосіб транспортування даних у режимі реального часу через мережі IP. Протокол розміщений поверх UDP, щоб мінімізувати накладні витрати заголовка пакетів, але з ціною; немає гарантії надійності або замовлення пакетів. У порівнянні з TCP, RTP є менш надійним, але він має меншу затримку в передачі пакетів, оскільки його накладні заголовки пакетів значно менші, ніж для TCP.

Протокол управління RTP (RTCP)

RTCP – це додатковий протокол, який використовується для передачі керуючої інформації, такої як кількість відправлених та втрачених пакетів,

описи тремтіння, затримки та кінцевих точок. Це найбільш корисно для керування базами часу сеансу та для аналізу QoS потоку RTP. Він також може забезпечити зворотний канал для обмеженої повторної передачі пакетів RTP.

Протоколи транспортного рівня

Вищевказані протоколи сигналізації відповідають за налаштування мультимедійних сеансів у мережі. Після встановлення з'єднання медіапотік між вузлами мережі встановлюється одним або декількома протоколами передачі даних, такими як UDP або TCP [2, 4].

User Datagram Protocol (UDP)

UDP – це мережевий протокол, що охоплює лише пакети, які транслюються. Немає підтвердження того, що пакет був отриманий на іншому кінці. Оскільки доставка не гарантована, передача голосу не буде працювати дуже добре лише з UDP, коли в мережі є пікові навантаження. Ось чому протокол транспортування медіа, такий як RTP, зазвичай працює над версією UDP.

Transport Control Protocol (TCP)

TCP використовує модель зв'язку клієнт / сервер. Кожен запит клієнта обробляється індивідуально, не пов'язаний з жодним попереднім. Це забезпечує доступність "безкоштовних" мережевих шляхів для використання інших каналів.

TCP створює менші пакети, які можуть передаватися через Інтернет та отримуватися шаром TCP на іншому кінці виклику, таким чином, щоб пакети були "перекомпоновані" назад у вихідне повідомлення. IP-рівень інтерпретує поле адреси кожного пакету так, щоб він потрапляв до правильного пункту призначення.

На відміну від UDP, TCP гарантує повне отримання пакетів на кінці прийому. Однак це робиться, дозволяючи повторну передачу пакетів, що додає затримки і не є корисним для даних у режимі реального часу. Для голосу

пакет, що запізнився через повторну передачу, настільки ж поганий, як і втрачений пакет. Через цю характеристику TCP, як правило, не вважається відповідним інструментом для передачі потокової інформації в реальному часі.

Кодування розмовної мови

Для передачі голосу IP-мережею, людський голос оцифровується за допомогою імпульсно-кодової модуляції, стискається (кодується) і розбивається на пакети. На приймаючій стороні, відбувається зворотна процедура – дані витягуються з пакетів, декодуются і перетворяться назад в аналоговий сигнал [15].

Кодування вносить додаткову затримку близько 15-45 мс., що виникає з наступних причин:

– Алгоритмічна затримка. Використання буфера для збору і накопичення сигналу, ведення статистики подальших відліків.

– Обчислювальна затримка. Характеризуються тим, що математичні перетворення, що виконуються над мовним сигналом, вимагають процесорного часу.

Подібна затримка з'являється і при декодуванні мови на іншій стороні. Затримку кодека необхідно враховувати при розрахунку скрізних затримок. Крім того, складні алгоритми кодування / декодування вимагають серйозніших витрат обчислювальних ресурсів системи.

Головними проблемами передачі голосу IP-мережею є: втрата пакетів при передачі; перевищення допустимого часу на доставку.

Тому це доводить неідеальність технології Інтернет телефонії, яка багато в чому потребує вдосконалення, наприклад, оптимізації затримок і прогресу алгоритмів компресування голосової мови.

2.3 Кодеки IP-телефонії

IP-телефонія базується на двох основних операціях: формування пакетів для передачі IP-мережею, а також конвертація аналогової мови в цифровий

формат у апараті кодуючого пристрою. Після отримання абонентом пакетів для відновлення вихідного повідомлення використовують ідентифікатори пакетів, що відповідають за їх порядок. У структурах, де час має не таку важливу роль, як, наприклад, отримання електронної пошти, затримка між послідовністю пакетів не відіграє вирішального значення. На відміну від мережевої телефонії, де мінімальне значення затримок, навіть ціною втрачених пакетів, значно покращує якість зв'язку. Ефективність технології покладений на кодуючий/декодуючий апарат.

Кодек G.711

Кодек G.711 з'явився раніше всіх цифрових кодеків мовних сигналів і є мінімально необхідним. Це означає, що будь-який пристрій VolP має підтримувати цей тип кодування.

Кодек G.726

Він забезпечує кодування цифрового потоку зі швидкістю 40, 32, 24 або 16 Кбіт / с, гарантуючи оцінки MOS на рівні 4.3 (32 Кбіт / с), може прийматися за еталон рівня якості телефонного зв'язку (toll quality). Однак в додатках IP-телефонії цей кодек практично не використовується, так як він не забезпечує достатньої стійкості до втрат інформації .

Кодек G.729

Кодек G.729 дуже популярний в додатках передачі мови по мережах Frame Relay. Кодек використовує кадр тривалістю 10 мс і забезпечує швидкість передачі 8 Кбіт / с. Однак, для кодека необхідний попередній аналіз сигналу тривалістю 5 мс.

Існують два варіанти кодека:

- G.729
- Спрощений варіант G.729A

Кодек GSM

GSM – найпопулярніший кодек Asterisk. Підтримується виробниками обладнання, в основному в шлюзах сотового зв'язку і VoIP-мережами. Він не обтяжений ліцензійними угодами, як G.729A, і пропонує високу

продуктивність, якщо враховувати вимоги, які він пред'являє до центрального процесора (ЦП). Якість одержуваного звуку загалом нижче, ніж забезпечує навіть спрощений G.729A.

Кодек iLBC

iLBC (Internet Low Bitrate Codec) забезпечує привабливе поєднання низького коефіцієнта використання смуги пропускання і прийнятної якості. Він особливо добре підходить для забезпечення якості в мережевих з'єднаннях з втратами.

Кодек Speex

Speex – це кодек із змінною швидкістю передачі цифрових даних (variable bitrate,VBR). Це означає, що він може динамічно змінювати швидкість передачі даних у відповідь на зміну умов мережі. Speex абсолютно безкоштовний кодек, ліцензований за версії Xiph.org ліцензії BSD. Даний кодек може використовуватися для каналів зі швидкістю передачі даних від 2,15 до 22,4 Кбіт / с завдяки його здатності змінювати свою швидкість передачі даних [15].

Asterisk Manager Interface (AMI)

Протокол взаємодії між Asterisk і клієнтом можна описати наступними характеристиками:

- Перед тим як посилати команди в сервер Asterisk, необхідно виконати сесію з'єднання клієнта з сервером Asterisk.
- Пакети можуть бути передані в будь-якій послідовності і в будь-який час після проходження процедури аутентифікації.
- Перший рядок пакета повинен містити один з таких ключів: «Action» (єдиний варіант при відправці клієнтом) і ключі «Event» (Подія) і «Response» (Відповідь) (повинні бути відправлені від Asterisk до клієнта).
- Порядок рядків в пакеті не має значення, можна використовувати будь-яку мову програмування, за допомогою якої можна формувати пакети на стороні клієнта.

- CRLF використовується для поділу кожного з рядків в пакеті та двох послідовностей CRLF (вона ж \ r \ n) для того, щоб позначити завершення передачі команди в Asterisk [14].
- AMI приймає підключення, що встановлюються на мережевий порт (за замовчуванням – TCP порт 5038). Клієнтська програма підключається до AMI через цей порт і аутентифікується, після цього Asterisk відповідатиме на запити, а також відправляти повідомлення про зміни стану заданих підсистем.

3.РОЗРОБКА ПРОГРАМНОГО КОМПЛЕКСУ ДЛЯ МОНІТОРИНГУ ПРОЦЕСІВ ІКТ-СИСТЕМИ

3.1 Проектування програмного модуля

Технології стрімко зростають на наших очах і користуватися ними стає все простіше кожного дня. Разом з усією IT-інфраструктурою зростає і попит на кваліфікований персонал, який зможе покластись на свої знання та досвід і коректно адмініструвати серверну частину програмного забезпечення, оперативно ідентифікувати і вирішувати проблеми безпеки.

Програмний комплекс – це набір програмних засобів, що працюють спільно для виконання однієї або кількох подібних завдань [16] .

Згідно ГОСТ 19781-90 програмний модуль – це програма або функціонально завершений фрагмент програми, призначений для:

-1- зберігання;

-2- трансляції;

-3- об'єднання з іншими програмними модулями;

-4- завантаження в оперативну пам'ять.

Розрізняють:

- стандартні модулі, що входять до мови програмування;
- модулі, призначені для користувача, призначені для спрощення роботи програмістів [16].

Програмний модуль, що буде розроблятися, відноситься до другого та третього типів.

На поточний момент існує актуальна задача з відстеження якості зв’язку на сервері ip-телефонії , а саме параметру Jitter, вирішення якої потребує індивідуального підходу.

На даній машині була розгорнута IP АТС Askozia. Цей варіант ідеально підходить для усіх потреб компанії для зв’язку з віддаленими підрозділами, так як розгортка буде виконуватися швидко та дієво, має зрозумілий веб-інтерфейс та готова одразу виконувати поставлені задачі.

В основі операційної системи лежить ядро Linux без оболонки, а це свідчить про неможливість роботи навіть з терміналом, що вже казати про встановлення Zabbix-агентів.

Для вирішення питання відстеження якості передачі телефонного сигналу IP-мережею, а саме показника затримки послідовних пакетів – Jitter , не відноситься до проблем , рішення якої має тільки один правильний варіант. Розв’язок даної задачі може бути вирішений декількома способами , обираючи ту чи іншу систему моніторингу , або використовуючи один з безлічі можливих мов програмування. Але вибір системи моніторингу Zabbix не випадковий , бо саме дана система дозволяє здійснювати відстеження без встановлення додаткового ПЗ.

На сервері Zabbix теж є Zabbix-агент . Саме за допомогою нього буде організовано виконання скрипта на Bash з певною частотою. Для цього на сервері в конфігураціях агента вписується ім’я і посилання на папку , де знаходиться скрипт та на сам скрипт. Через графічний інтерфейс Zabbix сервера потрібно обрати заданий ключ , який було вписано в конфігураціях агента. Далі активується і налаштовується частота виконання, при бажанні можна вказати значення, які будуть передані до скрипта при виконанні та міститься іще ряд корисних параметрів. У нашому випадку параметри для виконання будуть характеризувати значення Jitter – показник, що вказує на числове значення затримки послідовних пакетів, що передаються.

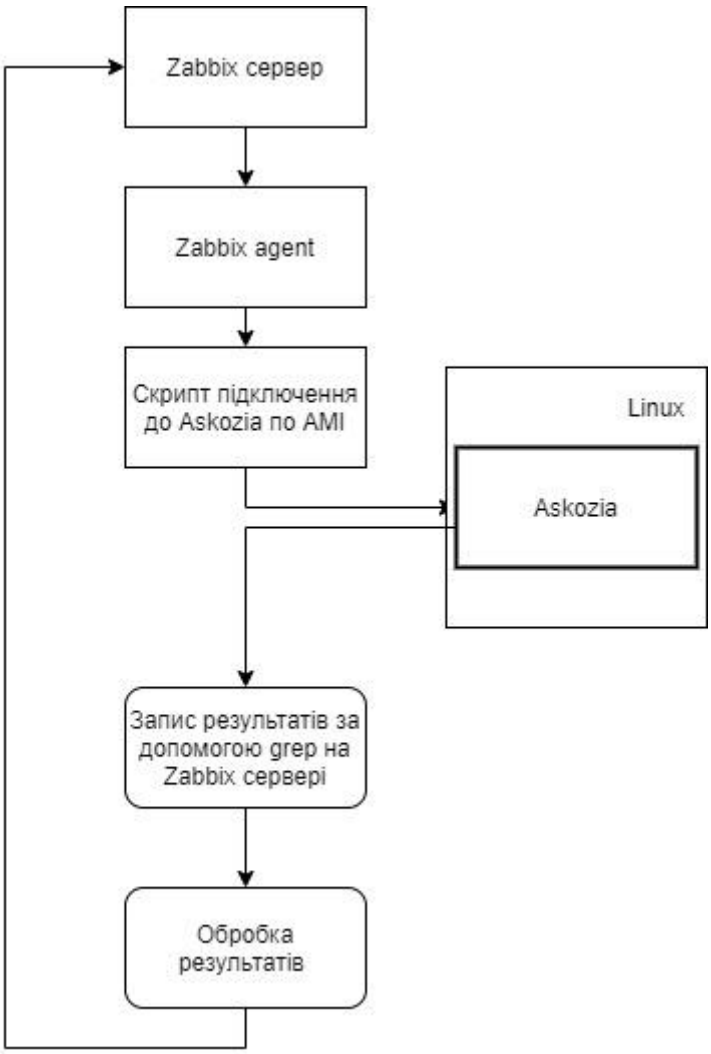

Рисунок 3.1– Блок-схема роботи модуля

За блок-схемою (рис. 3.1) на Zabbix-сервері є локальний Zabbix-агент, який саме і реалізує відстеження показників досліджуваної системи. У конфігураціях локального агенту вказується ключ та шлях до програмного модуля, для здійснення періодичного виконання. Виклик за ключем через локальний Zabbix-агент запускає скрипт за яким здійснюється підключення через AMI-інтерфейс до серверу IP-телефонії Askozia, виконує команду , що відображає поточні показники. Передані дані приводять до виду, який буде зручно обробляти на сервері Zabbix, куди вони й передаються. Усі значення формують послідовність значень і у разі високого показника Jitter за заданими тригерами виводиться повідомлення про рівень небезпеки.

Сам скрипт реалізує вхід в систему через Telnet або AMI інтерфейс , виконує ряд команд , після чого завершує сеанс . Дані передаються і

оброблюються через конвеєр Bash, де виділяється значення бажаного параметру/параметрів. Щодо базових значень , що передаються на виконання скрипта , то вони реалізують значення , при яких реалізується сортування за найбільшим параметром Jitter . Якщо вказане значення перевищує вхідний параметр , то на виході маємо відповідні значення з відповідністю «Номер абонента» - «Значення Jitter» . Відсутності підходящих параметрів не дасть результатів.

Розробка програмного модуля для відстеження якості зв’язку, а саме параметру Jitter з поєднанням системи моніторингу серверної частини Zabbix дає унікальне рішення, результатом якого є програмний комплекс, що вирішує поставлену в п.1.3 задачу.

3.2 Вибір технології для програмної реалізації

Безсумнівно проектування – це один з головних етапів розробки програмного засобу. Його вже завершено, але тепер постає питання у виборі безпосередньо технології для розробки програмного додатку. Даний підрозділ мітити: вибір мови програмування , для написання модуля; вибір сценарію і технології передачі ; написання програмного модуля та його опис.

Режим віддаленого управління, що називається також режимом термінального доступу, передбачає, що користувач перетворює свій комп'ютер у віртуальний термінал іншого комп'ютера, до якого він отримує віддалений доступ [1].

Для реалізації сценарію було запропоновано дві альтернативи для передачі команд для виконання на сервер телефонії у вигляді Telnet та SSH.

SH

SSH (від англ. "Secure Shell") – це протокол віддаленого адміністрування, розроблений для здійснення віддаленого управління операційними системами і тунелювання TCP-з'єднання. Використання цього протоколу допускає використання різних алгоритмів шифрування, що дозволяє безпечно працювати практично в будь-якому незахищеному середовищі: працювати з ПК через командну оболонку, передавати

шифрованим каналом будь-який тип даних (наприклад, відео та аудіофайли) [1].

Використання SSH підключення має ряд переваг:

- безпечна робота на віддаленому ПК з використанням командної оболонки;
- використання різних алгоритмів шифрування (симетричного, асиметричного) і хешування;
- можливість безпечного використання будь-якого мережевого протоколу, що дозволяє передавати захищений обмін файлами будь-якого розміру.

Протокол SSH використовує клієнт-серверну модель для аутентифікації віддалених систем і забезпечує шифрування даних, обмін якими відбувається в рамках віддаленого доступу.

За замовчуванням для роботи протоколу використовується TCP-22 порт: на ньому сервер (хост) очікує вхідне підключення і, після отримання команди і проведення аутентифікації, організовує запуск клієнта, відкриваючи обрану користувачем оболонку. При необхідності користувач може змінювати використовуваний порт.

Для створення SSH підключення клієнт повинен ініціювати з'єднання з сервером, забезпечивши захищене з'єднання і підтвердивши свій ідентифікатор (перевіряються відповідність ідентифікатора з попередніми записами, що зберігаються в RSA-файлі, і особисті дані користувача, необхідні для аутентифікації).

Telnet

Протокол Telnet працює в архітектурі клієнт-сервер, він забезпечує емуляцію алфавітно цифрового терміналу, обмежуючи користувача режимом командного рядка [1].

При натисканні клавіші відповідний код перехоплюється клієнтом Telnet, розміщується в TCP-повідомлення і відправляється через мережу вузла, яким користувач хоче управляти. При передачі на вузол призначення код

натиснутої клавіші витягується з TCP-повідомлення сервером Telnet і передається операційній системі вузла. ОС розглядає сеанс Telnet як один з сеансів локального користувача. Якщо ОС реагує на натискання клавіші виводу чергового символу на екран, то для сеансу віддаленого користувача цей символ також упаковується в TCP-повідомлення і мережею відправляється віддаленому вузлу. Клієнт Telnet витягує символ і відображає його у вікні свого терміналу, емулюючи термінал віддаленого вузла.

На жаль, протокол Telnet не задовольняє потреби з боку безпеки. У ньому не передбачені ні шифрування, ні перевірка автентичності переданих даних, тому він являє собою чудову мішень для кіберзлочинців. Тому сьогодні основним напрямом застосування Telnet є управління не комп'ютерами, а комунікаційними пристроями: маршрутизаторами, комутаторами і хабами. Таким чином, він вже скоріше не призначений для користувача протокол, а протокол адміністрування, тобто альтернатива SNMP.

AMI

AMI – серйозний і зручний програмний інтерфейс (API) Asterisk для управління системою із зовнішніх програм. Працює під керівництвом протоколу Telnet. Незважаючи на те , що протокол Telnet не шифрує отримані і передані дані, а лише витягує дані, отримані з TCP-пакетів, і виводить на екран, користування є безпечним. Перехопити передані до AMI дані для аутентифікації між софтфоном і інтерфейсом майже неможливо.

AMI використовується, у більшій мірі, для абонентських дзвінків без використання безпосередньо самого AMI-інтерфейсу. Посередником між ним і абонентом є веб-інтерфейс софтфона або прошивка IP-телефона. Таким чином, усі абоненти IP-мережі телефонного зв’язку є користувачами AMI-інтерфейсу.

Між сервером Asterisk і клієнтською програмою для встановлення зв'язку використовується простий порядковий протокол, кожен рядок повідомлення якого складається з двох рядків:

key – ключове слово, яке описує характер інформації, що міститься в поточному рядку. Ключове слово не є унікальним і може зустрічатися кілька разів в рамках однієї передачі.

value – значення параметра.

Ключове слово від значення відділяється двокрапкою.

Будемо використовувати термін «пакет», що буде описувати серію конструкцій «key: value», розділених CRLF і завершених додатковою послідовністю CRLF.

Виконання програмного модуля буде здійснене через Zabbix–агент на серверній частині комплексу. Тому головними інструментами для написання сценарію програми стають скриптова мова Bash та Python.

Bash

Найважливішим із призначених для користувача процесів є командна оболонка (вона ж командний інтерпретатор, або просто shell). Вона забезпечує взаємодію користувача з системою в текстовому режимі, дозволяючи вводити команди. Саме вона запускається, коли при вході у текстову консоль, і надає інтерфейс командного рядка [19]. Стандартний інтерпретатор у UNIX-системах це Bash.

Shell-скрипти дуже добре підходять для швидкого створення прототипів складних додатків, навіть не дивлячись на обмежений набір мовних конструкцій і певну "повільність". Така методу дозволяє детально опрацювати структуру майбутнього програми, виявити можливі "пастки" і лише потім приступити до кодування на C, C ++, Java, або Perl.

Скрипти повертають до класичної філософії UNIX - "розділяй і володарюй" тобто поділ складного проекту на ряд простих підзадач. Багато хто вважає такий підхід найкращим або, щонайменше, найбільш естетичним способом вирішення виникаючих проблем, ніж використання нового покоління мов - "все-в-одному", таких як Perl.

Для яких завдань застосовуються скрипти

- для ресурсномістких завдань, особливо коли важлива швидкість виконання (пошук, сортування і т.п.)
- для завдань, пов'язаних з виконанням математичних обчислень, особливо це стосується обчислень з плаваючою комою, обчислень з підвищеною точністю, комплексних чисел (для таких завдань краще використовувати C ++ або FORTRAN)
- для крос-платформного програмування (для цього краще підходить мова C)
- для великих програм, коли структурування є життєвою необхідністю (контроль за типами змінних, прототипами функцій і т.п.)
- для цільових завдань, від яких може залежати успіх підприємства.
- коли на чільне місце поставлено безпеку системи, коли необхідно забезпечити цілісність системи і захистити її від вторгнення, злому і вандалізму.
- для проектів, що містять компоненти, дуже тісно взаємодіють між собою.
- для завдань, що виконують величезний обсяг робіт з файлами.
- для завдань, які працюють з багатовимірними масивами.
- коли необхідно працювати зі структурами даних, такими як пов'язані списки або дерева
- коли необхідно надати графічний інтерфейс з користувачем (GUI)
- коли необхідний прямий доступ до апаратури комп'ютера
- коли необхідно виконувати обмін через порти введення-виведення або сокети
- коли необхідно використовувати зовнішні бібліотеки
- для пропрієтарних, "закритих" програм (скрипти вдають із себе вихідні тексти програм, доступні для загального огляду)

Python

Python є високорівневою мовою програмування, яка для виведення результатів використовує інтерпретатор. Python містить велику стандартну

бібліотеку модулів протестованого коду, які легко можна вбудувати у власні програми.

Оскільки Python орієнтований на читаність коду, в ньому часто використовуються ключові слова англійською мовою там, де інші мови програмування зазвичай використовують розділові знаки. Особлива його відмінність полягає в тому, що для угрупування інструкцій в блоці коду Python використовує відступи, а не ключові слова або знаки пунктуації.

Найбільшими досягненнями і значними особливостями є:

- Швидкість у вивченні.
- Легкість в обслуговуванні, так як має модульну структуру.
- Оперує великим об’ємом стандартної бібліотеки, що легко інтегрується.
- Його можна запустити на різних платформах і всюди він буде мати один і той же інтерфейс.
- Інтерпретованість. Компіляція не потребується.
- Не має статичного розподілу пам’яті.
- Підтримує об’єктно-орієнтований метод програмування.

Якість отриманого програмного забезпечення, швидкість розробки, можливість для інтеграції на інші мови, бібліотеки підтримки робить Python універсальним інструментом для розробки [18].

Таким чином, оптимальним варіантом для вирішення питання з відстеження зв’язку комплексом для моніторингу Zabbix буде використаний– абонентський програмний інтерфейс AMI, як спосіб передачі команд програмного модуля, виконання команд і повернення отриманих значень. Передача буде здійснюватися за протоколом Telnet, протоколом для віддаленого управління.

Будь-хто може розширити можливості Zabbix-агента, створивши скрипти за допомогою таких мов програмування, як скрипт командної оболонки, Perl, Python, Ruby і будь-яких інших мов, які можуть бути виконані [6]. Тому вибір для виконання поставленої задачі залишається за розробником.

У якості інструменту для написання програмного модуля буде використано скриптову мову Bash.

На вхід даного скрипта подається два значення . Одне з них – це значення Jitter, при якому буде виведений результат з поміткою «danger», інший – «alarm», з відповідною градацією небезпеки.

Скрипт на bash:

```
#!/bin/bash
(
echo "Action: login"
echo "Username: zabbix"
echo "Secret: as6d5gf4356raes1g321e32gqaw3413gfasde32avf"
echo "Events: off"
echo
sleep 1
echo "Action: Command"
echo "Command: pjsip show channelstats"
echo
echo "Action: logoff"
echo
sleep 1
)| telnet 10.10.2.21 5038 | grep 'Output:  ' | sed 's/Output://g' | sort -rnk 5 | grep -v "^$\|PROVIDER\|ChannelId\|Receive\|$
 | awk '{ if (NF==13) print $2,$8,$12 }' \
 | awk -v danger_alert=$1 -v alarm_alert=$2 \
 '{ if($2>=danger_alert && $2<alarm_alert || $3 >=danger_alert \
 && $3<alarm_alert) print $1,$2,$3 " danger"; \
  if($2>=alarm_alert || $3>=alarm_alert  ) \
 print $1,$2,$3 " alarm"  }'
```

3.3 Результат роботи програмного модуля

Спосіб виконання програмного модуля з боку Zabbix буде відбуватися за допомогою AMI–інтерфейсу.

Наведені команди і даний синтаксис актуальний лише для інтерфейсу AMI. У консолі, при підключенні по SSH, синтаксис буде відрізнятися.

Справа від команди через двокрапку буде коротке пояснення, а в дужках (необхідні для виконання привілеї).

AMI має обмежений список команд (рис. 3.2) для виконання їх усі можна переглянути зайшовши у систему з командою:

Action: login

Username: zabbix

Secret: as6d5gf4356raes1g321e32gqaw3413gfasde32avf

```
robot@mail:~$ telnet 10.10.2.21 5038
Trying 10.10.2.21...
Connected to 10.10.2.21.
Escape character is '^]'.
Asterisk Call Manager/5.0.1
Action: login
Username: zabbix
Secret: as6d5gf4356raes1g321e32gqaw3413gfasde32avf
Events: off

Response: Success
Message: Authentication accepted
```

Рисунок 3.2 – Вхід на сервер Askozia , використовуючи AMI інтерфейс

*Events: off – не обов’язкова команда, але без неї усі поточні розмови будуть відображатися в консолі.

Список доступних команд (для поточного користувача з поточним набором привілеїв) можна при виконанні Action: ListCommands. Скріншот виконання приведений на рис. 3.3.

```
robot@mail: ~
Action: ListCommands

Response: Success
AbsoluteTimeout: Set absolute timeout.  (Priv: system,call,all)
Atxfer: Attended transfer.  (Priv: call,all)
BlindTransfer: Blind transfer channel(s) to the given destination  (Priv: call,all)
Bridge: Bridge two channels already in the PBX.  (Priv: call,all)
BridgeDestroy: Destroy a bridge.  (Priv: <none>)
BridgeInfo: Get information about a bridge.  (Priv: <none>)
BridgeKick: Kick a channel from a bridge.  (Priv: <none>)
BridgeList: Get a list of bridges in the system.  (Priv: <none>)
```

Рисунок 3.3 – Виконання на сервері Askozia команди ListCommands, використовуючи AMI інтерфейс

Поточні розмови можна переглянути увівши команду (рис. 3.4):

Action: Command

Command: pjsip show channelstats

```
Action: Command
Command: pjsip show channelstats

Response: Success
Message: Command output follows
Output:
Output:                                              ..........Receive..... .....Transmit.....
Output:  BridgeId ChannelId ..... UpTime.. Codec.   Count    Lost Pct  Jitter   Count    Lost Pct  Jitter RTT....
Output:  ========================================================================================================
Output:
Output:           1607-00000796      00:00:57 alaw     2751       0    0   0.000   2700       0    0   0.002   0.001
Output:           1635-00000798      00:00:10 alaw      408       0    0   0.000    357       0    0   0.000   0.002
Output:  2a14f7e8 1649-00000784      00:02:28 alaw     7281      27    0   0.000   7054      19    0   0.002   0.011
Output:  2a14f7e8 SIP-PROVIDER-25C21 00:02:28 alaw     7073       0    0   0.000   7260      48    0   0.000   0.004
Output:  3cd26402 SIP-PROVIDER-25C21 00:01:00 alaw     2667       0    0   0.000   2892       0    0   0.000   0.004
Output:  48c82aa5 SIP-PROVIDER-25C21 00:02:02 alaw     5356       0    0   0.000   5348       8    0   0.001   0.003
Output:  6167563b 1632-00000789      00:02:01 alaw     5362       8    0   0.000   5356       2    0   0.001   0.012
Output:  62393854 1631-0000078e      00:01:14 alaw     3545       8    0   0.000   3207       1    0   0.002   0.012
Output:  6f340e48 SIP-PROVIDER-25C21 00:01:11 alaw     3223       0    0   0.000   3446       0    0   0.000   0.003
Output:  76a73161 SIP-PROVIDER-25C21 00:01:19 alaw     3656       0    0   0.000   3885       0    0   0.001   0.004
Output:  81bfc0b9 1648-00000790      00:01:02 alaw     2998       5    0   0.000   2652       0    0   0.001   0.017
Output:  a93ac3f4 1641-0000078c      00:01:22 alaw     3993       1    0   0.000   3641       0    0   0.001   0.016
Output:
Output: Objects found: 12
Output:
```

Рисунок 3.4 -Вивід результату команди pjsip show channelstats, через інтерфейс AMI

Action: Command – дає змогу виконати команду з інтерфейсу AMI з синтаксисом роботи в терміналі.

Для виходу з інтерфейсу виконується команда:

Action: logoff

```
Action: logoff

Response: Goodbye
Message: Thanks for all the fish.

Connection closed by foreign host.
```

Рисунок 3.5 – Вихід з інтерфейсу AMI

Для виконання зазначених команд використовуватиметься користувач Zabbix, серверу телефонії Askozia з даними привілеями:

Read: command.

Write: .

Налаштування модуля буде виконано у декілька етапів:

1. Створення вузла для роботи з IP АТС Askozia.

2. Створення елемента даних, який буде звертатись до агента.
3. Налаштування тригерів.
4. Моніторинг результатів.

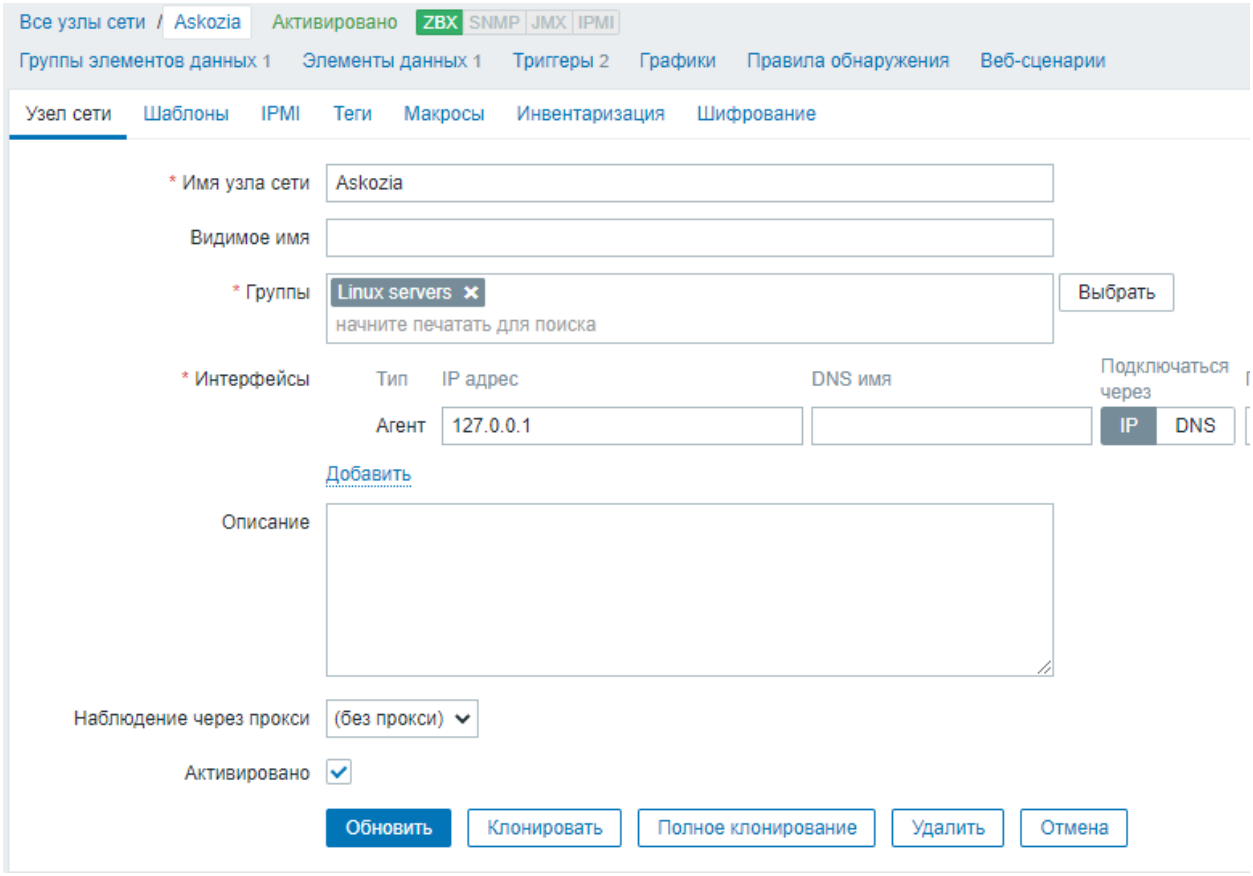

Рисунок 3.6 – Панель додавання вузла відстеження на панелі Zabbix

На рис. 3.6 додається вузол. До стандартних налаштувань додається лише належність групі для подальшої зручності. Параметри агента вказують на те, що звернення буде відправлено локальному агенту на сервері Zabbix. До новоствореного вузла додається елемент даних (рис.3.7), функціонал якого саме і полягає у відстеженні параметра Jitter.

Виконання за вказаним ключем remote команди channelstats_jitter[$1,$2] (рис. 3.7) передає локальному Zabbix-агенту граничні показники jitter.

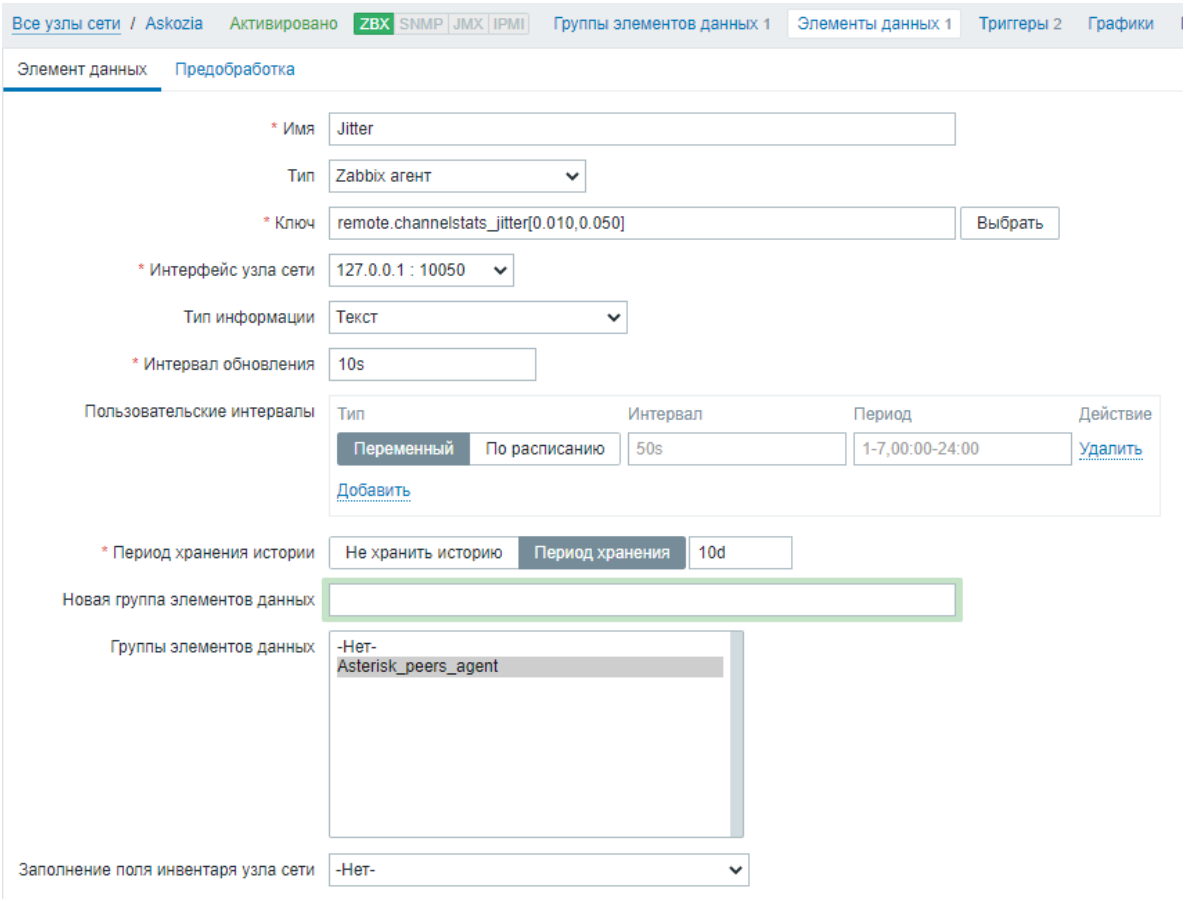

Рисунок 3.7 – Панель додавання елемента даних для вузла Askozia на панелі Zabbix

Зазначений ключ записаний в конфігураціях локального агента на сервері Zabbix.

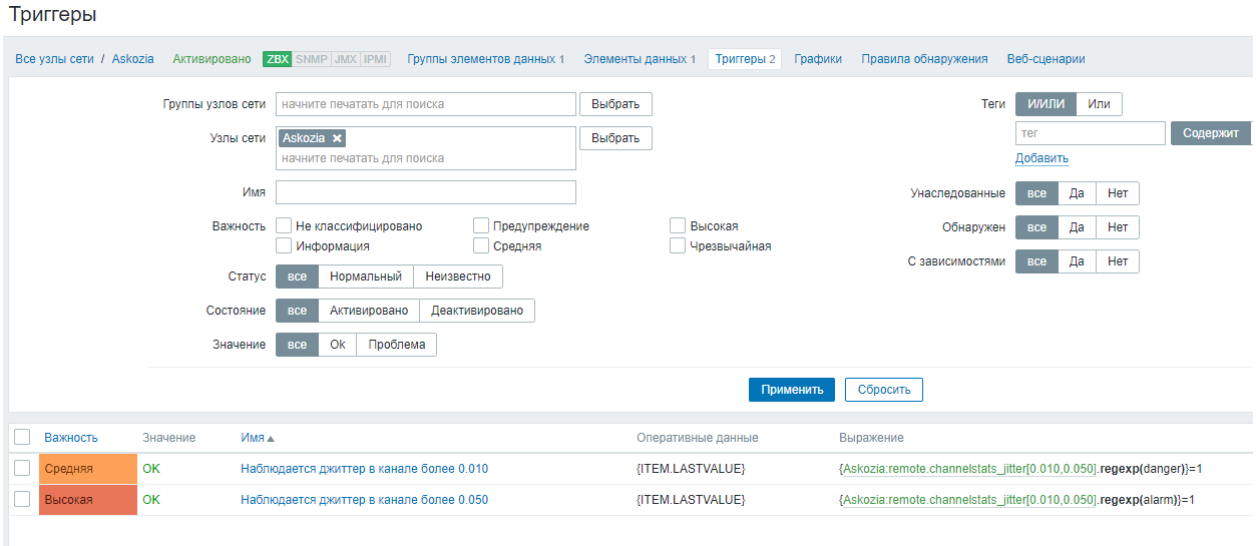

Рисунок 3.8 – Панель додавання тригера для вузла Askozia на панелі Zabbix

Відповідно до рис. 3.8, тригери захоплюють результати, які задовольняють умовам, що при значенні Jitter:

> 0.010 – ризик середній.

> 0.050 – ризик високий.

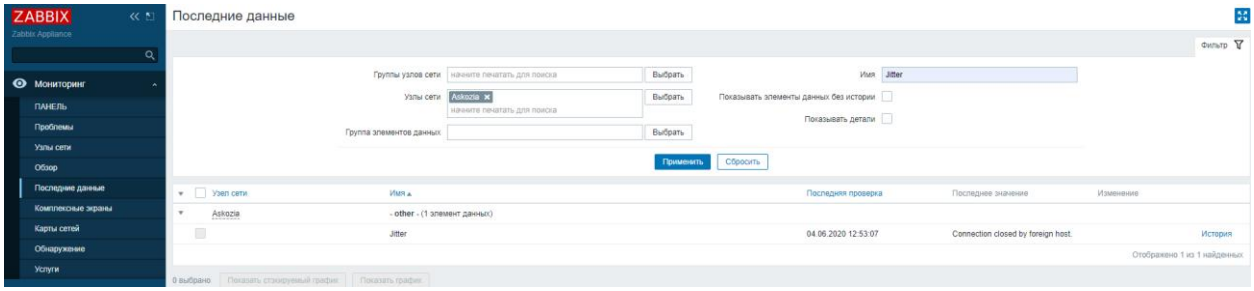

Рисунок 3.9 – Панель відстеження елемента даних Jitter для вузла Askozia на панелі Zabbix

На панелі навігації, в розділі «Последние данные», можна знайти ведення обліку параметра, заповнивши відповідні фільтри як на рис. 3.9.

Головною причиною, чому актуальність програмного модуля існує в даному контексті і чому на нього є попит це тому , що:

- Програмне забезпечення (ПЗ) вирішує питання моніторингу якості зв’язку у IP-мережах.
- Гарантує безпеку.

А головне, що комбінацію ПЗ з системою моніторингу Zabbix, можна використовувати не лише у випадках з ОС без оболонки. У іншому випадку даний програмний сценарій буде використовуватися дещо інакше, як показано у прикладі нижче.

Розгорнуто сервер телефонії на базі FreePBX. На ньому встановлено Zabbix-agent. У налаштуваннях агента дописується ключ зі вказівником на шлях до скрипта, що приведений нижче на рис. 3.10.

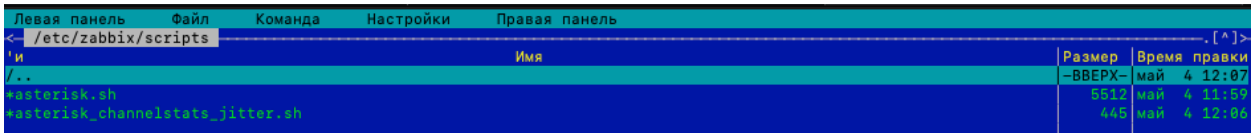

Рисунок 3.10 – Уміст каталогу /etc/zabbix/scripts серверу телефонії FreePBX

Установлений агент автоматично створює конфігураційний файл вміст і параметри якого характеризують вузол (рис. 3.11).

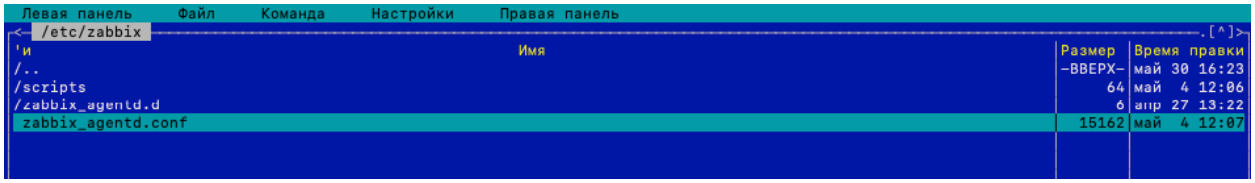

Рисунок 3.11 – Уміст каталогу /etc/Zabbix і до конфігураційного файлу агента серверу телефонії FreePBX

Використання нестандартних засобів для моніторингу потребують запису типу:

UserParameter = <ключ[параметри]>, <шлях_до_модуля> $1_параметр $2_парметр …$n_парметр

Приклад використання приведений нижче на рисунку 3.12.

```
UserParameter=brut.password,/etc/zabbix/scripts/asterisk.sh brut.password
UserParameter=calls.active,/etc/zabbix/scripts/asterisk.sh calls.active
UserParameter=calls.processed,/etc/zabbix/scripts/asterisk.sh calls.processed
UserParameter=calls.longest, /etc/zabbix/scripts/asterisk.sh calls.longest
UserParameter=channels.active,/etc/zabbix/scripts/asterisk.sh channels.active
UserParameter=status,/etc/zabbix/scripts/asterisk.sh status
UserParameter=status.crashes,/etc/zabbix/scripts/asterisk.sh status.crashes
UserParameter=status.reload,/etc/zabbix/scripts/asterisk.sh status.reload
UserParameter=status.uptime,/etc/zabbix/scripts/asterisk.sh status.uptime
UserParameter=status.version,/etc/zabbix/scripts/asterisk.sh status.version
UserParameter=iax.register.time,/etc/zabbix/scripts/asterisk.sh iax.register.time
UserParameter=iax.trunk.down,/etc/zabbix/scripts/asterisk.sh iax.trunk.down
UserParameter=sip.register.time,/etc/zabbix/scripts/asterisk.sh sip.register.time
UserParameter=sip.trunk.down,/etc/zabbix/scripts/asterisk.sh sip.trunk.down
UserParameter=sip.peers,/etc/zabbix/scripts/asterisk.sh sip.peerss
UserParameter=asterisk.channelstats_jitter[*],/etc/zabbix/scripts/asterisk_channelstats_jitter.sh $1 $2
```

Рисунок 3.12 –Вказівка ключа asterisk.channelstats_jitter[*] для виконання в параметрах Zabbix agent на сервері телефонії FreePBX

З боку Zabbix додається вузол з IP даного серверу (рис. 3.13). Відшукується агент. Його статус буде вказано зеленим індикатором.

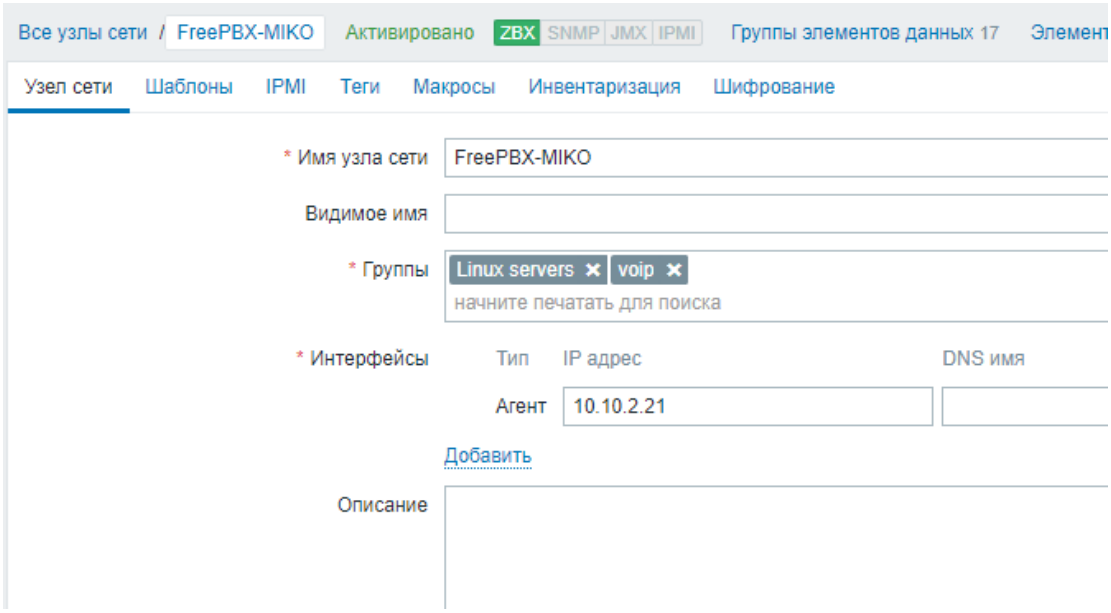

Рисунок 3.13 – Налаштування вузла для відстеження за допомогою Zabbix агента серверу FreeBPX-MIKO

Додається елемент даних з ключем, який вказали на рис. 3.12 і передали йому показники Jitter, для подальшого спрацювання тригеру та визначення показника ризику у відповідності: 0,010 – середній, 0,040 – небезпечний, аналогічно до елемента даних серверу Askozia (рис. 3.14).

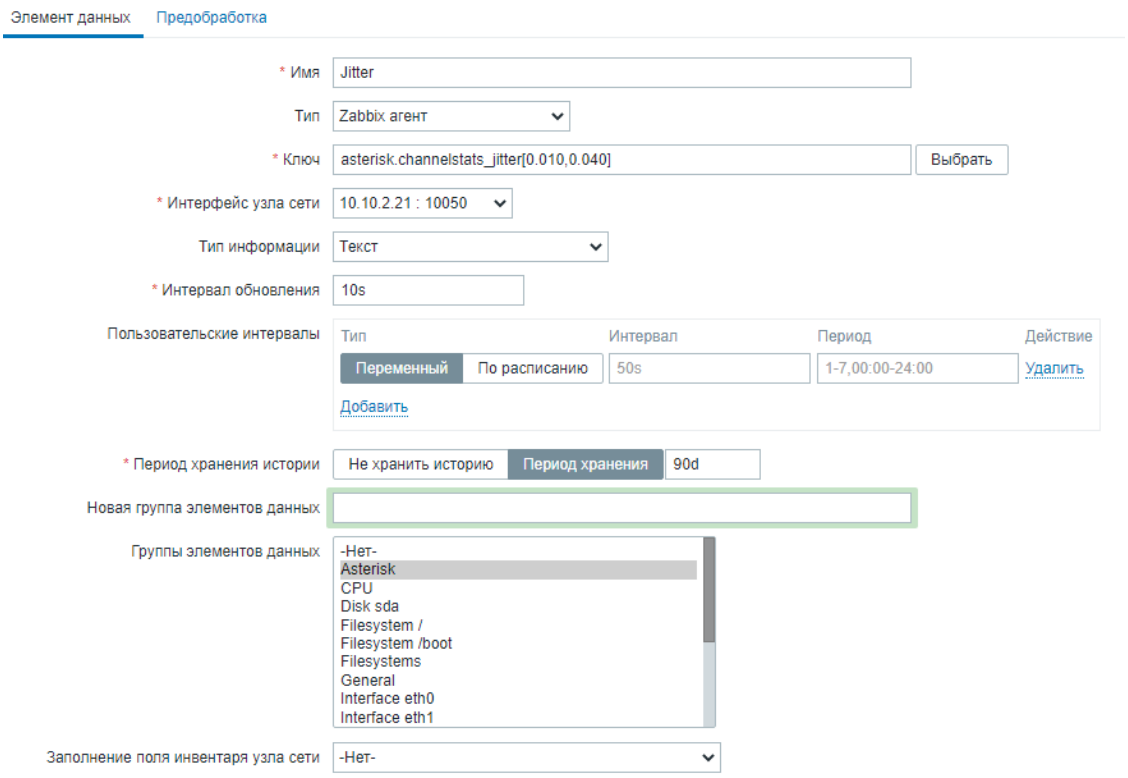

Рисунок 3.14 - Панель додавання елемента даних для вузла FreeBPX-MIKO на панелі Zabbix

Налаштування тригерів для подальшого виведення на інформаційну панель Zabbix(рис. 3.15).

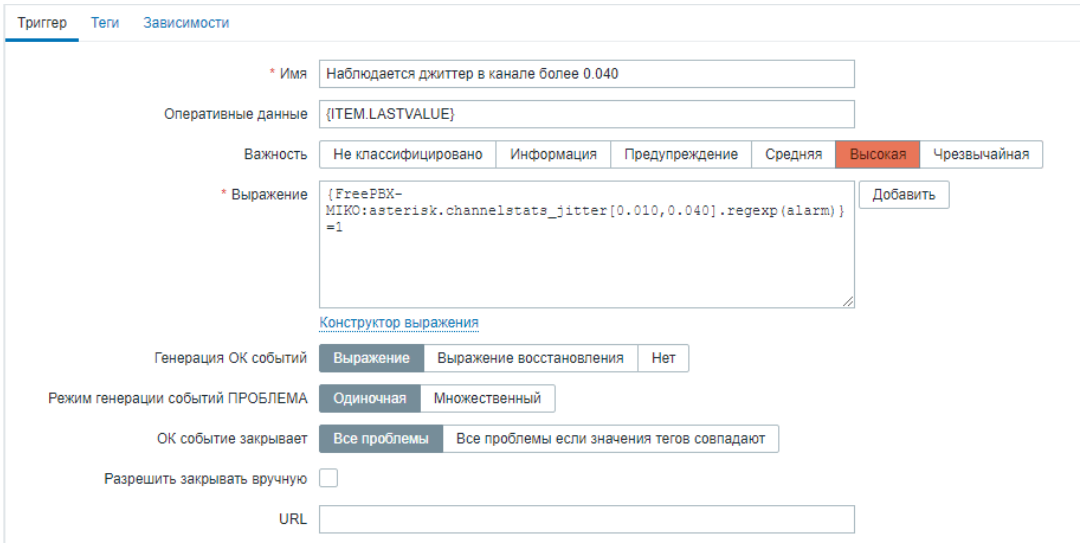

Рисунок 3.15 - Тригер для сповіщення про високий рівень ризику.

ВИСНОВКИ

Дана робота практично доводить значущість використання систем моніторингу у корпоративних мережах. Завдяки їй автоматизація типових , на перший погляд , процесів переходить на інший рівень. Буденність стає легшою, а життя простішим.

Було розроблено програмний комплекс, який вирішує проблему відстеження якості телефонного зв’язку в IP-мережі для серверу IP-телефонії Askozia, для машин з операційною системою без системної оболонки, отримує поточні значення параметру затримки послідовних пакетів (Jitter), налаштування сповіщень при критичному рівні Jitter, при цьому безпечно передаючи дані користувацьким AMI інтерфейсом. Даний програмний комплекс можна використовувати не лише для операційних систем без системної оболонки. Він добре себе показав для IP-АТС Asterisk, що входить до складу серверу IP-телефонії FreePBX і при звичайному використанні за допомогою Zabbix-агента.

Кожна проблема потребує індивідуального підходу та індивідуального рішення, але незважаючи на це одностайно правильного розв’язку може і не бути. Завжди вірогідні альтернативні підходи з більшою або меншою складністю, із різним рівнем навантаження та показником самостійності. Але даний програмний комплекс вирішує поставлену задачу, гарантує її працездатність для серверів мережевої телефонії з IP-АТС Asterisk і являється оптимальним рішенням для вирішення проблеми.

СПИСОК ЛІТЕРАТУРИ

1. В. Олифер Компьютерные сети. Принципы, технологии, протоколы /Н. Олифер – [5-е изд.]

2. Стандарт інформаційної безпеки Інформаційні технології — Технології безпеки — Практичні правила менеджменту інформаційної безпеки ISO/IEC 17799: 2005

3. Ланде Д.В. Пошук знань в Internet - М .: Діалектика, 2005. - 272 с.

4. The OpenNMS Project. Opennms.org. Retrieved 2014-06-16.

5. Page Sequence Monitor (PSM) Setup - OpenNMS | The OpenNMS Project". OpenNMS. 2014-05-22. Retrieved 2014-06-16.

6. Zabbix Documentation 5.0. Zabbix.com . Retrivied 01.07.2013.

7. Graylog attracts $2.5 mln. penhub.com. Archived from the original on 31 May 2015. Retrieved 4 February 2015.

8. Top 10 Series: Splunk. Eventide Asset Management, LLC. Archived from the original on 14 October 2019. Retrieved 14 October 2019. Splunk's platform interprets big data for business intelligence and is trusted by 90 of the Fortune 100 companies. (Splunk is based in San Francisco, CA.)

9. Review: Graylog delivers open source log management for the dedicated do-it-yourselfer. cio.com. Retrieved 9 November 2015.

10. Start-Ups Aim to Help Tame Corporate Data, Pui-Wing Tam, Wall Street Journal, September 08, 2009

11. Open Source Monitoring: Icinga vs Nagios Sos open source. Sosopensource.com. 2010-11-25.

12. И.В. Баскаков IP-телефония в компьютерных сетях / А.В. Пролетарский, С.А. Мельников, Р.А. Федотов

13. Б. С. Гольдштейн Протокол SIP. Справочник / А. А. Зарубин, В. В. Саморезов.

14. Меггелен Дж. Asterisk: будущее телефонии / Дж. Меггелен, Л. Мадсен, Дж. Смит. –[2-е изд.] – Пер. с англ. – СПб: Символ-Плюс, 2009. – 656 с.

15. Б. С. Гольдштейн IP-Телефония / А. В. Пинчук, А. Л. Суховицкий. - М. : Радио и связь, 2001. - 334, [1] с. : ил., портр., табл.; 25 см.; ISBN 5-256-01585-0

16.Фінансовий тлумачний словник https://dic.academic.ru/contents.nsf/fin_enc/

17. “Zabbix Documentation 5.0”. Zabbix.com . Retrivied 01.07.2013.

18. Майк МакГрат. Программирование на Python для начинающих. – М.: Эксмо, 2015. – 192 с.

19. Колисниченко Д.Н., Аллен Питер В. LINUX: полное руководство. — СПб: Наука и Техника, 2006. — 784 с: ил. Под редакцией М.В. Финкова.