applied sciences

MDPI

Article

The Development of a Secure Internet Protocol (IP) Network Based on Asterisk Private Branch Exchange (PBX)

Mubarak Yakubova [1], Olga Manankova [2,*], Assel Mukasheva [3], Alimzhan Baikenov [2] and Tansaule Serikov [4]

[1] Department of Information Technology, Almaty University of Power Engineering and Telecommunications Name after Gumarbek Daukeev, Almaty 050010, Kazakhstan
[2] Department of Telecommunications and Space Engineering, Almaty University of Power Engineering and Telecommunications Name after Gumarbek Daukeev, Almaty 050010, Kazakhstan
[3] School of Information Technology and Engineering, Kazakh-British Technical University, Almaty 050000, Kazakhstan
[4] Electronics and Telecommunication Department, S. Seifullin Kazakh AgroTechnical Research University, Astana 010011, Kazakhstan
* Correspondence: o.manankova@aues.kz or olga.manank@gmail.com

Abstract: The research problem described in this article is related to the security of an IP network that is set up between two cities using hosting. The network is used for transmitting telephone traffic between servers located in Germany and the Netherlands. The concern is that with the increasing adoption of IP telephony worldwide, the network might be vulnerable to hacking and unauthorized access, posing a threat to the privacy and security of the transmitted information. This article proposes a solution to address the security concerns of the IP network. After conducting an experiment and establishing a connection between the two servers using the WireShark sniffer, a dump of real traffic between the servers was obtained. Upon analysis, a vulnerability in the network was identified, which could potentially be exploited by malicious actors. To enhance the security of the network, this article suggests the implementation of the Transport Layer Security (TLS) protocol. TLS is a cryptographic protocol that provides secure communication over a computer network, ensuring data confidentiality and integrity during transmission. Integrating TLS into the network infrastructure, will protect the telephone traffic and prevent unauthorized access and eavesdropping.

Keywords: Asterisk; Private Branch Exchange based on Internet Protocol (IP PBX); IP telephony; security; Session Initiation Protocol (SIP); Transport Layer Security (TLS); Wireshark

Citation: Yakubova, M.; Manankova, O.; Mukasheva, A.; Baikenov, A.; Serikov, T. The Development of a Secure Internet Protocol (IP) Network Based on Asterisk Private Branch Exchange (PBX). *Appl. Sci.* **2023**, *13*, 10712. https://doi.org/10.3390/app131910712

Academic Editors: Christos Bouras, Nouman Ali and Ihsan Rabbi

Received: 4 February 2023
Revised: 9 September 2023
Accepted: 15 September 2023
Published: 26 September 2023

1. Introduction

Building a network based on IP PBX Asterisk starts with analyzing and choosing a Linux distribution, as there are many and there are distributions that can meet the needs of each. The analysis of existing distributions shows that in any distribution the main principles remain unchanged. The IP PBX Asterisk software began as a telephony system for small businesses, but over the decade since its initial release, it has become a universal tool for creating communication applications for medium-sized businesses. Today, Asterisk is used not only in IP-PBX systems, but also in VoIP gateways, call center systems, conference bridges, voice mail servers, and all types of other real-time communication applications.

Asterisk PBX has key advantages such as scalability, cost-effectiveness, customization, flexibility, etc. The Asterisk PBX software 16.16.1 can be easily scaled up or down to meet the changing needs of an organization [1]. The Asterisk PBX software is an open source solution, which means that it is free to download and use. This helps organizations save on the cost of expensive proprietary PBX systems [2]. In addition, the Asterisk PBX software is highly configurable, which means that organizations can modify and extend the functionality of the software to meet their specific needs [3,4].

As noted in [5], the Asterisk PBX software can be integrated with a wide range of telephony hardware and software, allowing organizations to use existing equipment and infrastructure. Asterisk PBX also supports voice over IP (VoIP), which means that, while using existing equipment and infrastructure, organizations can also make and receive calls over the Internet, which can help reduce the cost of long distance calls [6–8]. The Asterisk PBX software should also be used to expand the range of additional features such as call routing, voice mail, conference calls and call recording [9,10]. To ensure the security of the network built on the basis of Asterisk PBX, there is a wide range of security features, such as encryption, authentication, and authorization, which help protect against unauthorized access to the system and data [11–13]. For such setups, the Asterisk PBX software is highly available, which means that it can automatically fail over to a standby server in the event of a hardware failure or other problem, ensuring that services are always available [14–17].

Asterisk is a flexible and versatile system that is one of the most commonly programmable PBXs deployed within corporations due to its open access technology and modular and flexible design. Given the capabilities of the IP PBX Asterisk, its main criterion for choosing an office IP PBX is the ability to scale and integrate into the company's existing computer network at minimal cost. The main component of the network infrastructure of the IP network is the telephone exchange. Asterisk allows a server to be used as a telephone exchange, which can be any computer with an operating system of the Linux family installed on it (including virtual machines). Small and home offices require a minimum of 512 MB of Random Access Memory (RAM) and a 1 GHz processor (up to 10 channels). A small business system with up to 25 channels requires 1 GB of RAM and a 3 GHz processor [18,19].

This is below the capabilities of modern office personal computers, which are widely sold. The telecommunications industry is undergoing great changes, but is in no hurry to use them. Asterisk, on the other hand, is in a great hurry to not only embrace change, but actively embrace it. Asterisk significantly outperforms all PBX solutions on the market, as it has extensive programming capabilities that allow complex call processing algorithms to be implemented [20,21]. Compliance with standards has become evident over the past few years: standards change so quickly that the only way to keep up with them is to be able to quickly respond to new directions emerging in technology.

Asterisk, by virtue of being an open source system created by a community of developers, is ideal for rapid development that will ensure compliance with such rapidly changing standards.

Asterisk does not focus on profitability analysis or market research. It evolves in response to whatever the community finds interesting or necessary. Compared to the implementation of traditional PBXs in offices, using Asterisk PBX has the following advantages:

- A single network infrastructure, which Asterisk has, allows a wider range of services to be received both for office maintenance and business services, reducing the cost of technical staff;
- In traditional PBXs, the maximum message recording configuration is limited to 32 h of recording, whereas in Asterisk, the number of voice messages can be limited by the administrator and depends on the capacity of the hard disk;
- Asterisk has a powerful ability to create an interactive voice menu;
- Asterisk has a wide forwarding functionality, which includes forwarding by time of day, presence at the workplace, simultaneous forwarding to several phones, individual forwarding algorithm, etc.;
- Fax reception is carried out to all numbers with automatic sending to e-mail;
- Asterisk runs on operation system Linux, and it is possible to combine the functions of a mini-PBX and an Internet server. For example, when using an E1 channel, the data part and the voice part can be separated. Thus, through the same channel, both voice services and access to the Internet and other services from the same provider can be received [22–24].

The phone system, based on the Asterisk PBX, has great features that are not available in a traditional PBX with switching. Voice mail, conferencing, call queues and agents, music while waiting, and call parking are just some of the features provided by Asterisk [25].

The IP network architecture based on Asterisk PBX completely solves the problem of organizing multiple access from one system to another [26,27]. But, despite the many advantages of the Asterisk PBX, the problem of secure data transmission in the network, especially when it is scaled when the PBX acts as an independent service system, remains not fully understood [28–32]. Most research is currently focused on protocols and encryption [33–36], as well as the quality of the transmitted voice traffic [37–40]. This does not eliminate the main threats as the attackers are currently focused on something else [41]. The availability of Asterisk from the Internet is one of the main threats to network security. When designing an IP network based on programmable exchanges, it is necessary to take into account the network architecture and system configuration to ensure secure data transmission [42–44]. To increase the level of security at the first stage, it is necessary to design the network correctly, and then it is necessary to protect data transmitted over open communication channels from interception and listening [45–49].

From the review, we can conclude that due to the widespread use and implementation of programmable PBXs in small and medium-sized businesses, in particular Asterisk, the topic of developing a real network based on IP PBX Asterisk is relevant. In addition, an important component is to ensure the security of the designed network using modern tools and methods. The details of designing a real network, presented in this article, allow users to avoid errors when installing and configuring the Asterisk server. We decided to use hosting, with one server located in Germany and the other in the Netherlands. This approach shows the flexibility of Asterisk, i.e., that users can deploy a network even without their own equipment, and also shows the importance of ensuring the security of transmitted traffic in such design conditions. The article uses the TLS protocol to protect the transmitted traffic.

The structure of this paper is as follows. Section 2 presents a proposed systematic methodology for developing a secure IP network. The details of the experiment on configuration, testing, and security analysis of the IP network based on Asterisk PBX are presented in Section 3. Section 4 offers the conclusions of the observations made in Section 3.

2. Methodologies

The development of a secure IP network based on Asterisk PBX involves implementing several security measures, including network design, access control, encryption, firewall configuration, network monitoring, regular updates and patching, and user education. These measures can help ensure the security, confidentiality, and availability of data transmitted over the network.

The development of a secure IP network based on Asterisk PBX consists of several steps:

Step 1. We need to install and configure the Asterisk PBX software on servers in two cities. In our case, one server is located in Germany, the other in the Netherlands.

The IP telephony network consists of two servers on which the Asterisk software is installed, as shows in Figure 1.

To install the Asterisk software, it was decided to use hosting, and servers in Germany and the Netherlands were selected.

Thus, building a network based on Asterisk Private Branch Exchange based on Internet Protocol (IP PBX) starts with analyzing and choosing a Linux distribution. The Ubuntu operating system was chosen, which offers a well-designed platform for IP-PBX and is reliable.

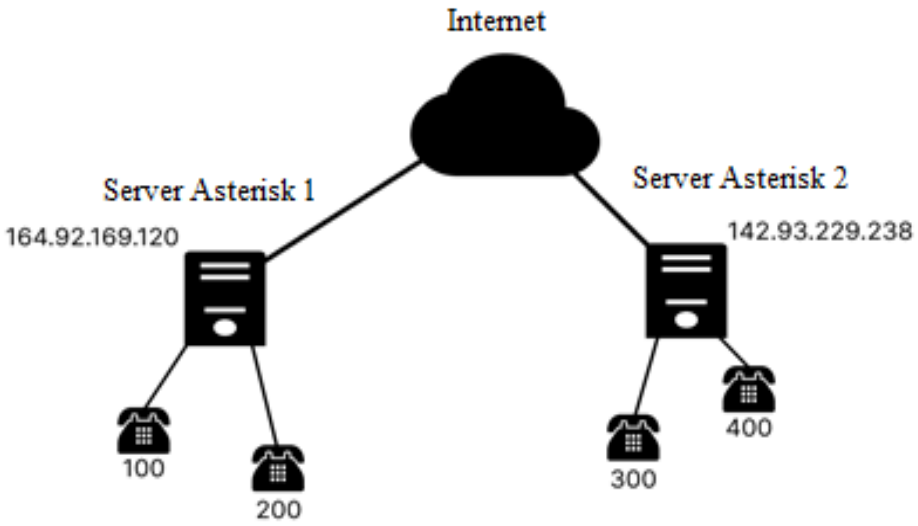

Figure 1. The developed network based on Asterisk IP PBX.

Step 2. We need to configure the IP phones that connect to the PBX. In our case, dial plan is configured on the servers. Blink is installed as a softphone. It is available for free and easy to use.

Step 3. After setting up the PBX and devices, we should test the network to make sure that it works properly and that all devices can connect and make/receive calls.

Step 4. After setting up the network, we need to analyze its security. This includes identifying potential vulnerabilities with Wireshark.

Step 5. To mitigate the impact of vulnerabilities on the network, we use the TLS protocol setting. TLS provides a secure communication channel between two endpoints by encrypting the data exchanged between them. When two endpoints establish a TLS connection, they negotiate a set of cryptographic protocols, algorithms, and parameters to use for the encryption and decryption of data. This negotiation is achieved through a process called the TLS Handshake Protocol, which ensures that both endpoints agree on the same set of parameters and that the parameters are not vulnerable to known attacks.

The TLS Handshake Protocol involves the following steps:

Client hello: The client sends a message to the server indicating its preferred encryption algorithms and other parameters.

Server hello: The server responds with a message indicating the encryption algorithms and other parameters that will be used for the connection.

Certificate exchange: The server sends its digital certificate to the client to prove its identity.

Key exchange: The client and server exchange encryption keys, which are used to encrypt and decrypt data exchanged between them.

Authentication: The client and server authenticate each other using the digital certificates exchanged earlier.

Session key generation: The client and server use the exchanged encryption keys to generate a unique session key, which is used for the remainder of the connection.

Encryption: The client and server use the session key to encrypt and decrypt data exchanged between them. This provides a high level of security for communication over the internet, even if the underlying network is untrusted or compromised.

The process of exchanging such messages is shown in Figure 2.

Overall, the TLS protocol provides a high level of security for communication over the internet, and its use has become increasingly important as the volume and sensitivity of online communication continues to grow.

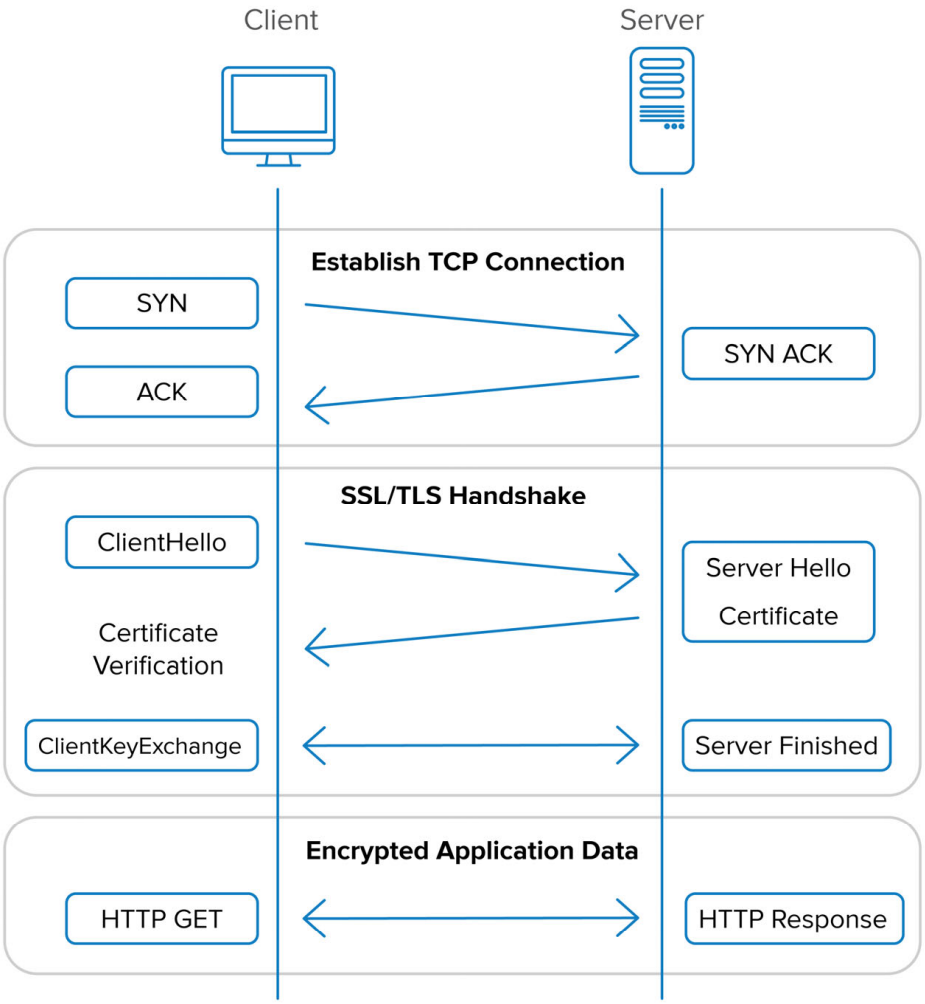

Figure 2. The TLS Handshake Protocol.

3. Results and Discussion

3.1. Development in the Configuration of a Real IP Network Based on Asterisk PBX

The most popular solution that provides a huge range of additional features and capabilities is the Asterisk IP PBX software. It is a free software package capable of acting as an IP telephony server based on popular protocols such as SIP, call center functions, and many other functions. The project is open and is constantly under development, during which the list of features of this IP PBX is expanding. Asterisk allows any computer with an operating system of the Linux family installed on it to be used as a server. The scheme of the real network on the selected servers is shown in Figure 1.

In Figure 1, 100, 200, 300 and 400 phones connected to the asterisk server which in Almaty (hosting in Germany) have the IP address 164.92.169.120, and in Astana (hosting in the Netherlands) the IP address 142.93.229.238. The networks were combined with the help of an Internet operator for a certain payment for the direction of information transfer in both directions.

To install IP PBX Asterisk, we use a repository consisting of distributions and program archives, choosing from it a modern version that suits us; there are distributions that can satisfy the needs of everyone who accesses program archives. In our case, the Ubuntu distribution is selected. Let us determine the necessary distributions of Asterisk 16.16.1 to install them with the "apt search asterisk" command, as shown in Figure 3.

```
> sudo apt search asterisk
c   asterisk                                    - asterisk PBX sound files - US English
v   asterisk-abi-1fb7f5c06d7a2052e38d021b3d8ca151 - asterisk PBX sound files - en-us/g722
idA asterisk-config                            - asterisk PBX sound files - en-us/gsm
idA asterisk-core-sounds-en                    - asterisk PBX sound files - en-us/wav
p   asterisk-core-sounds-en-g722               - asterisk PBX sound files - Spanish
idA asterisk-core-sounds-en-gsm                - asterisk PBX sound files - es-mx/g722
p   asterisk-core-sounds-en-wav                - asterisk PBX sound files - es-mx/gsm
p   asterisk-core-sounds-es                    - asterisk PBX sound files - es-mx/wav
p   asterisk-core-sounds-es-g722               - asterisk PBX sound files - Canadian French
p   asterisk-core-sounds-es-gsm                - asterisk PBX sound files - fr-ca/g722
p   asterisk-core-sounds-es-wav                - asterisk PBX sound files - fr-ca/gsm
p   asterisk-core-sounds-fr                    - asterisk PBX sound files - fr-ca/wav
p   asterisk-core-sounds-fr-g722               - asterisk PBX sound files - Italian
p   asterisk-core-sounds-fr-gsm                - asterisk PBX sound files - it-it/g722
p   asterisk-core-sounds-fr-wav                - asterisk PBX sound files - it-it/gsm
p   asterisk-core-sounds-it                    - asterisk PBX sound files - it-it/wav
p   asterisk-core-sounds-it-g722               - asterisk PBX sound files - Russian
p   asterisk-core-sounds-it-gsm                - asterisk PBX sound files - ru-ru/g722
p   asterisk-core-sounds-it-wav                - asterisk PBX sound files - ru-ru/gsm
p   asterisk-core-sounds-ru                    - asterisk PBX sound files - ru-ru/wav
p   asterisk-core-sounds-ru-g722               - DAHDI devices support for the Asterisk PBX
p   asterisk-core-sounds-ru-gsm                - Development files for Asterisk
p   asterisk-core-sounds-ru-wav                - Source code documentation for Asterisk
p   asterisk-dahdi                             - Bluetooth phone support for the Asterisk PBX
p   asterisk-dev                               - loadable modules for the Asterisk PBX
p   asterisk-doc
p   asterisk-flite
p   asterisk-mobile
idA asterisk-modules
p   asterisk-moh-opsound-g722
idA asterisk-moh-opsound-gsm
```

Figure 3. Checking packages in the repository.

After the successful installation of Asterisk, we can check the status of Asterisk with the following command: system ctl status asterisk (Figure 4).

```
> systemctl status asterisk
● asterisk.service - Asterisk PBX
     Loaded: loaded (/lib/systemd/system/asterisk.service; enabled; vendor preset: enabled)
     Active: active (running) since Wed 2022-11-02 19:49:59 +06; 2min 30s ago
       Docs: man:asterisk(8)
   Main PID: 4434 (asterisk)
      Tasks: 69 (limit: 14086)
     Memory: 41.9M
        CPU: 3.391s
     CGroup: /system.slice/asterisk.service
             ├─4434 /usr/sbin/asterisk -g -f -p -U asterisk
             └─4435 astcanary /var/run/asterisk/alt.asterisk.canary.tweet.tweet.tweet 4434

ноя 02 19:49:58 unknown systemd[1]: Starting Asterisk PBX...
```

Figure 4. Running Asterisk status check.

The version can also be checked by directly accessing Asterisk with the v (version) key, as shown in Figure 5.

```
   [LD] res_pjsip_mwi.o -> res_pjsip_mwi.so
Building Documentation For: third-party channels pbx apps codecs formats cdr c
 +--------- Asterisk Build Complete ---------+
 + Asterisk has successfully been built, and +
 + can be installed by running:              +
 +                                           +
 +               make install                +
 +-------------------------------------------+
[root@localhost asterisk-16.16.1]#
```

Figure 5. Checking the installed version.

After "ecnfyjdrb lbcnhb,enbdf" it is proposed to copy the default settings that appear when installing the package. This is necessary in order to return to the original state in each case. To do this, we run the "cp" (copy) command and specify two arguments: the path to the original file and the path to the copy of the file (Figure 6).

```
root@test2:~# cp /etc/asterisk/sip.conf sip.conf
root@test2:~# ls
sip.conf
root@test2:~# cp /etc/asterisk/extensions.conf extensions.conf
root@test2:~# ls
extensions.conf  sip.conf
root@test2:~#
```

Figure 6. Copying default settings.

Now, to exchange data and connect remote users, we need to open some ports that work with Asterisk closed. Ports 5060 and 5061 are needed for the SIP protocol, 10,000–20,000 for RTP, port 4569 for IAX, and 5038 AMI, as shown in Figure 7.

```
root@test2:~# iptables -A INPUT -p udp -m udp --dport 5060 -j ACCEPT
root@test2:~# iptables -A INPUT -p udp -m udp --dport 5061 -j ACCEPT
root@test2:~# iptables -A INPUT -p udp -m udp --dport 10000:20000 -j ACCEPT
root@test2:~# iptables -A INPUT -p udp -m udp --dport 4569 -j ACCEPT
root@test2:~# iptables -A INPUT -p tcp -m tcp --dport 5038 -j ACCEPT
root@test2:~#
```

Figure 7. Opening ports for Asterisk.

Next, we edit the «sip.conf» file on the remote servers. This file is responsible for registration on other Asterisk servers; telephone numbers are also registered here. On server 1 (conditionally designated "Almaty"), telephone numbers from 100 to 200 are created, and on the second server ("Astana" server), those from 300 to 400 are created. This is necessary for the conditional separation of numbers, that is, for context. For example, when calling 302, we know that we are referring to the "Astana" server, as shown in Figure 8.

To connect to another server, we need to add a new peer. In Figure 9, this is the "Astana" peer. Type = friend means that the client can be both a receiving and a sending party. Secret is the password for authorization. Context is the name of the instruction that determines what to do with the call (these instructions are contained in the «extensions.conf» file). «Host = dynamic» means that the user can connect from any IP address.

```
  GNU nano 5.4
;Almaty asterisk server
[general]
register => almaty:almaty_pass@164.92.169.120/astana

[astana]
type = friend
secret = astana_pass
context = astana_incoming
host = dynamic
insecure = invite,port
disallow = all
allow = ulaw

[100]
type = friend
host = dynamic
context = phones
secret = 100

[200]
type = friend
host = dynamic
context = phones
secret = 200
```

Figure 8. SIP settings on remote server No. 1.

```
  GNU nano 5.4
[globals]

[general]
autofallthrough=yes

[default]

[incoming_calls]

[phones]
include => internal
include => remote

[internal]
exten => _[12]XX,1,Dial(SIP/${EXTEN})

[remote]
exten => _[34]XX,1,Dial(SIP/astana/${EXTEN})

[astana_incoming]
include => internal
```

Figure 9. Dialplan settings on remote server No. 2.

After these steps, at the very beginning of the "general" context, we add registration to another server in accordance with the peer settings.

The internal context describes the reception of incoming calls with a mask of 100 or 200. The remote context is 300 and 400, respectively. Astana incoming is responsible for receiving calls from a remote server and forwarding them to internal numbers.

We configure the second server in the same way, as shown in Figure 9.

Afterwards, we go to the Asterisk console, through the asterisk -r command, and enter reload. This will reload all settings (we changed sip.conf and extensions.conf); otherwise the changes will not take effect.

As can be seen in Figure 10, we register via soft telephony to number 100.

```
root@test2:~# asterisk -r
Asterisk 16.16.1~dfsg-1+deb11u1, Copyright (C) 1999 - 2018, Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.
=========================================================================
Connected to Asterisk 16.16.1~dfsg-1+deb11u1 currently running on test2 (pid = 543)
test2*CLI> sip show peers
Name/username             Host                                    Dyn Forcerport Comedia    ACL Port     Status

100/100                   92.47.57.249                             D  Auto (No)  No             51392    Unmonitored

200                       (Unspecified)                            D  Auto (No)  No             0        Unmonitored

astana/almaty             164.92.169.120                           D  Auto (No)  No             5060     Unmonitored

3 sip peers [Monitored: 0 online, 0 offline Unmonitored: 2 online, 1 offline]
test2*CLI>
```

Figure 10. Output of connected clients.

To check the correct operation of the server, we write the sip show peers command, which displays all connected clients on the screen. And to check successful registration on another server, we enter the sip show registry command, as shown in Figure 11.

```
root@test2:~# asterisk -r
Asterisk 16.16.1~dfsg-1+deb11u1, Copyright (C) 1999 - 2018, Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.
=========================================================================
Connected to Asterisk 16.16.1~dfsg-1+deb11u1 currently running on test2 (pid = 543)
test2*CLI> sip show registry
Host                                dnsmgr Username       Refresh State                Reg.Time
164.92.169.120:5060                 N      almaty             105 Registered           Fri, 11 Nov 2022 18:04:31
1 SIP registrations.
test2*CLI>
```

Figure 11. Registering a remote server No. 2 on No. 1.

Blink is installed as a softphone. It is available for free and easy to use. We make a call without encryption to the subscriber and remove the traffic using Wireshark. Since we did not set up encryption, all packets are transmitted in the clear.

Let us capture traffic using the Wireshark program. Out of over 40,000 captured packets, only 30 meet our criteria. This is because the vast majority of packets exchanged are payload packets, which is voice, so they use Real-Time Transport Protocol (RTP). SIP does not carry voice, but only sends control information to establish, manage, and terminate calls.

Accordingly, the attacker sees all the data of the SIP packet. A fragment of the SIP address with possible information available to an attacker is highlighted in blue, as shown in Figure 12.

Some packages are assigned the "Request" status, while others are assigned the status as listed in the Info column. Requests are SIP messages that initiate certain functions, while message status (or responses) are messages that respond to requests and indicate the status of those requests. Typical requests include INVITE, ACK, NOTIFY, and BYE. Typical responses include Try, Call, OK, and Bad Request.

All responses display the SIP protocol, and requests have the SIP/SDP protocol in the Protocol column. The Session Description Protocol (SDP) is a companion protocol to SIP used to describe multimedia communication sessions for the purposes of session announcement, session invitation, and parameter negotiation.

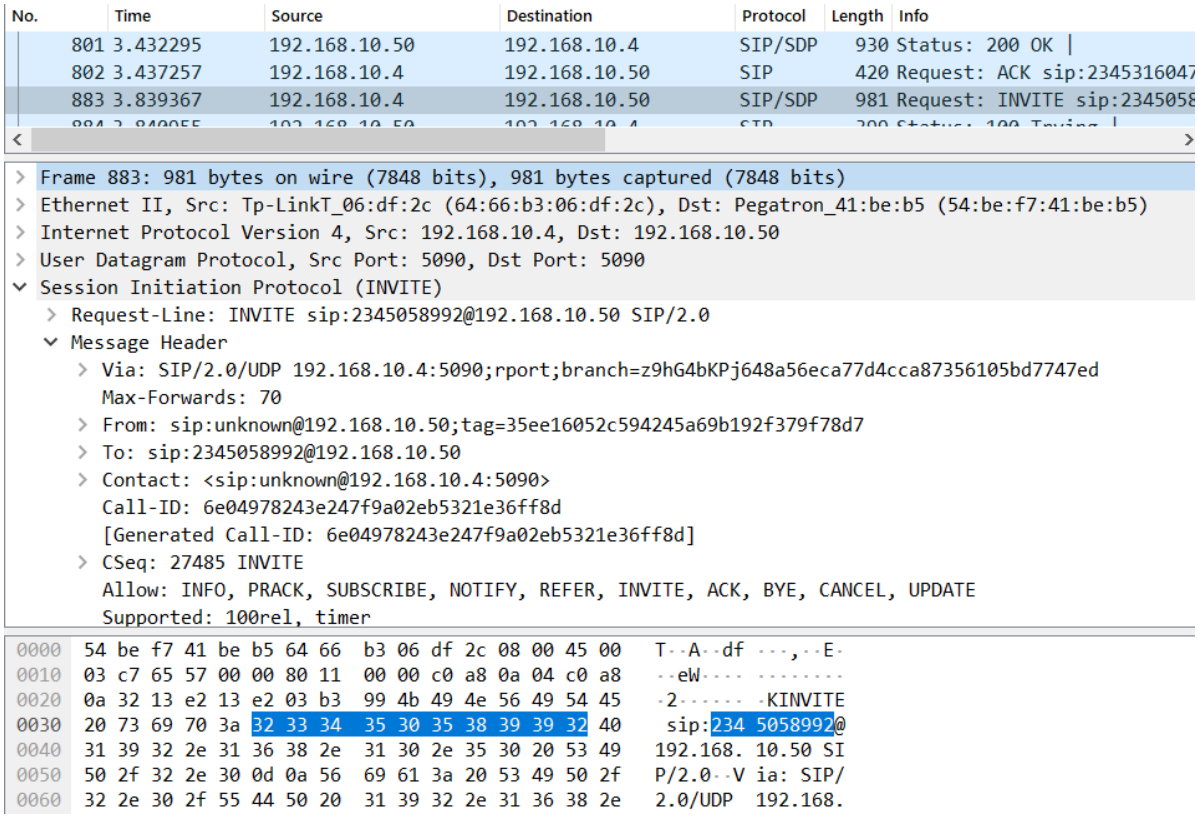

Figure 12. Captured SIP packet data.

The SIP messages filtered here are transmitted exclusively between the Blink client and the SIP server. SIP messages can also be sent between VoIP end devices participating in the data exchange.

RTP traffic is also transmitted unencrypted. Therefore, the data are also visible, as shown in Figure 13.

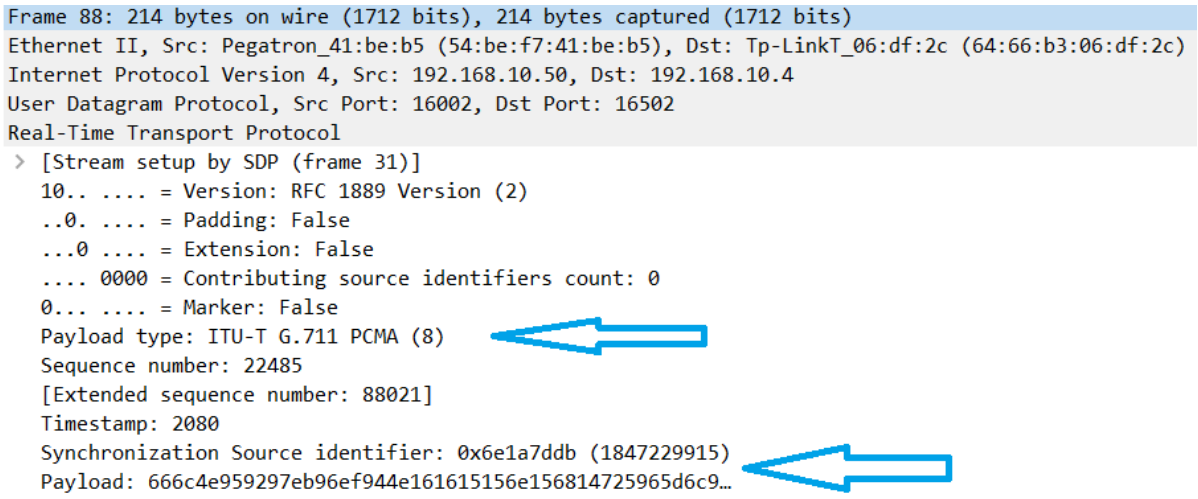

Figure 13. RTP packet data.

This packet indicates that the payload was created using the G.711 codec, as indicated by the top arrow. The bottom arrow indicates the composition of the payload. There are several fields in the RTP packet, and the protocol has two main focuses, payload identification and sequence. The recipient of the audio data must know exactly which codec was used to create the packet in order to assemble the stream back.

Figure 14 shows the RTP stream analysis.

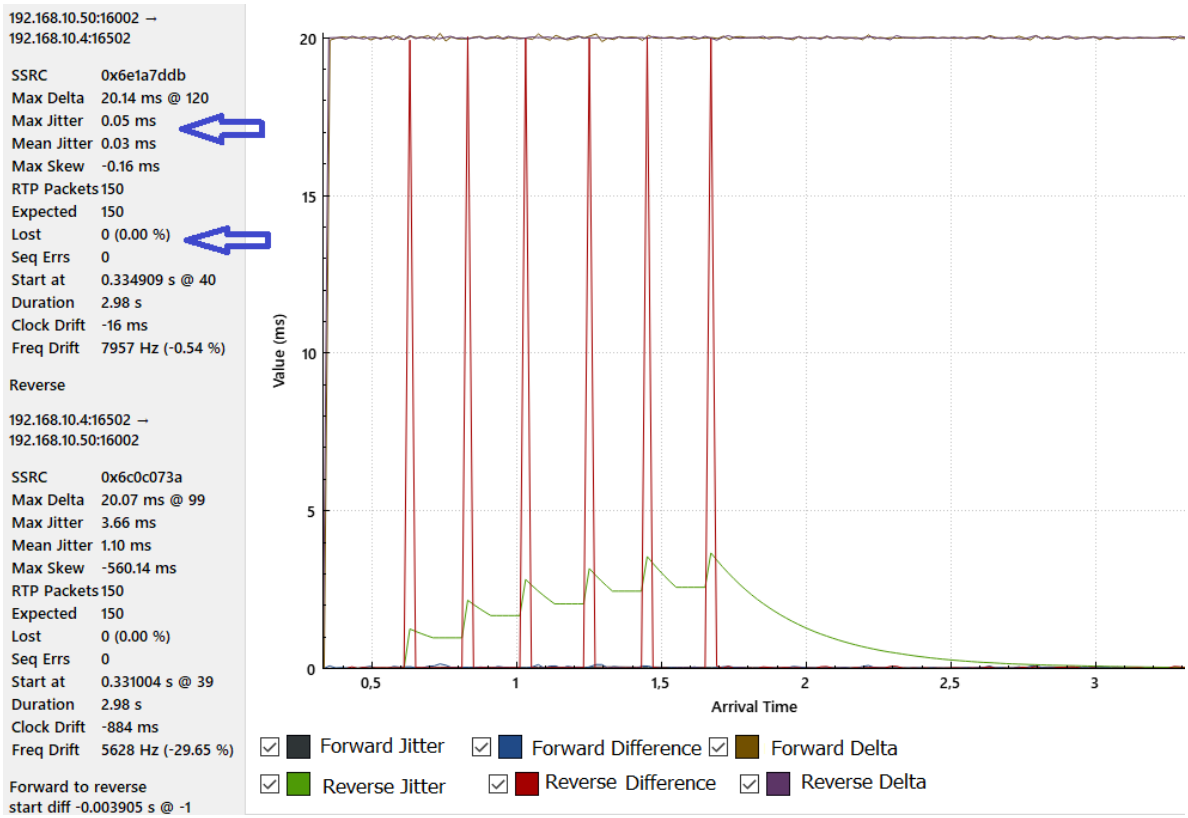

Figure 14. RTP stream analysis.

The important sections of the particular graph are the Jitter and the percentage of the expected and lost RTP packet. Packet loss can provide low-priority backdoors so cybercriminals can attack. In our case, there is no packet loss and minimal jitter, which indicates a stable connection. Figure 15 visualizes all packets as blue line and SIP with RTP as red line. This shows the total load transferred between servers.

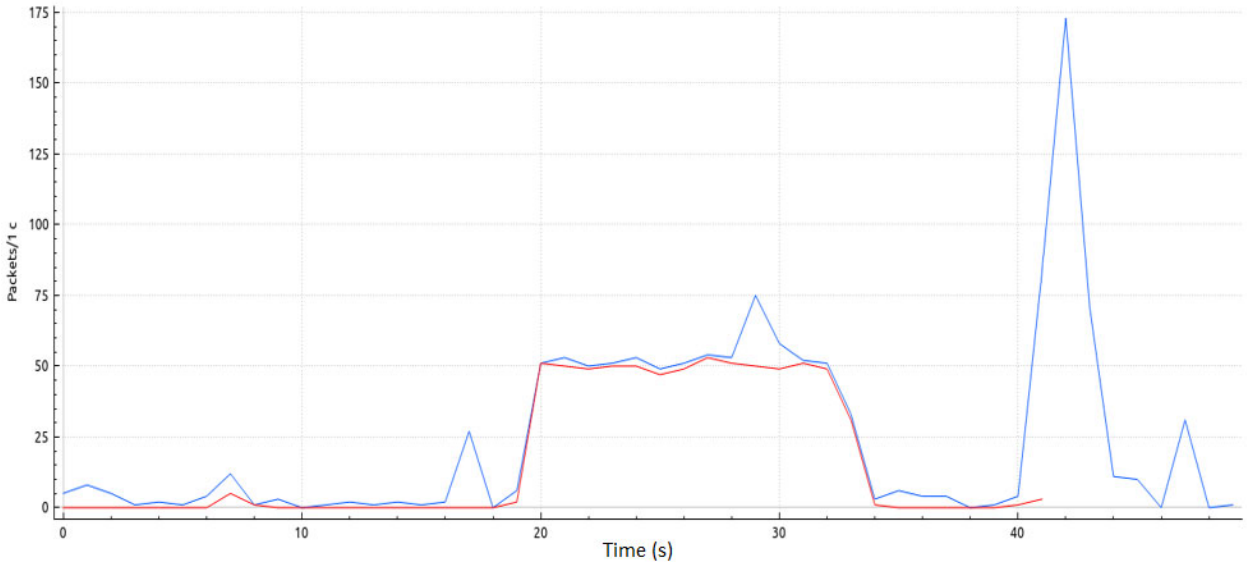

Figure 15. Total load.

By analyzing network traffic with WireShark, we get an idea of Asterisk's performance. In this case, we observe an average load during the capture period from 20 to 35 s from the start of the capture.

3.2. Development of a Real Network Encryption Method Based on the TLS Protocol

Transport Layer Security (TLS): This is a security protocol that can be used to encrypt and authenticate SIP traffic. TLS can be used to secure the signaling channel between the client and the server.

Generation of certificates and keys: As mentioned above, Asterisk uses UDP port 5060 (by default) for the SIP signaling protocol, and also transmits this information in the clear, so by intercepting traffic, we can get all the information regarding numbers and, in fact, the conversation itself. SIP over TLS will be able to help us with this problem, which in turn will encrypt signaling data using port 5061 (by default) TCP. Therefore, even by intercepting traffic, an attacker will receive information in encrypted form, but this only applies to the SIP protocol; the voice needs to be encrypted separately. SRTP will help us here. It is worth noting that using only TLS without SRTP does not make sense, since media traffic remains unencrypted.

It is worth noting that using TLS is not a panacea, and will not protect us from MITM (man-in-the-middle) attacks, if certificate authentication is not used.

First we need to generate keys for encryption. The necessary script is already in the Asterisk archive we downloaded. We go to the unpacked folder with the asterisk, and then to/contrib/scripts. We are interested in ast_tls_cert.

The certificate is generated for a year; therefore, before generating certificates, we can go into the script and correct all 365 to 3650 or any other number so that the issue period is 10 years or any necessary period.

We also need to use the hostname in the system (in other words, the computer name). If the current hostname does not suit us, then we use edit/etc/hosts and write the desired name at the end of the first line, for example, asterisk_server (Figure 16).

```
No config file specified, creating './tmp.cfg'
You can use this config file to create additional certs without
re-entering the information for the fields in the certificate
Creating certificate ./asterisk.key
Creating signing request ./asterisk.csr
Creating certificate ./asterisk.crt
Certificate request self-signature ok
subject=CN = unknown, O = EvilCorp
Enter pass phrase for ./ca.key:
Combining key and crt into ./asterisk.pem
```

Figure 16. Generating the certificate and keys of the Asterisk server.

Key C is the hostname (dns name) or IP address. O is the name of our organization. D is the certificate generation directory. We will be prompted to enter a passphrase for ca.key. This will create a ca.crt file. Next, we will be prompted to enter the passphrase again, after which the asterisk.key file will be created. The asterisk.crt file will be created automatically.

We will be prompted to enter the passphrase a third time, after which the asterisk.pem file will be created, which is a combination of the asterisk.key and asterisk.crt files. The next step is to generate client certificates and keys (Figure 17).

The "-m client" option tells the script that we need a client certificate, not a server certificate. The "-c ca.crt" option specifies which CA we are using. The "-k ca.key" switch provides the key for the above CA. The "-C" option, since we are defining a client this time, is used to determine the hostname or IP address of our SIP phone. The "-O" parameter specifies the name of our organization. The "-d" option is the output key directory. The "-o" parameter is the name of the key that we output.

```
No config file specified, creating './tmp.cfg'
You can use this config file to create additional certs without
re-entering the information for the fields in the certificate
Creating certificate ./100.key
Creating signing request ./100.csr
Creating certificate ./100.crt
Certificate request self-signature ok
subject=CN = 127.0.0.1, O = EvilCorp
Enter pass phrase for ca.key:
Combining key and crt into ./100.pem
```

Figure 17. Software for generating client certificates and keys.

Next, go to the sip.conf settings and add these lines to the general field:
tlsenable = yes
tlsbindaddr = 0.0.0.0:5061
tlscertfile = /etc/asterisk/keys/asterisk.pem
tlscafile = /etc/asterisk/keys/ca.crt
tlscipher = ALL
tlsclientmethod = tlsv1
tlsdontverifyserver = no

Here we activate TLS, specify which interfaces and which port to listen on, specify the Asterisk certificate, and also the CA certificate (if we use self-signed certificates for peers), and enable all types of encryption. In the last option, we either allow verification of the peer certificate, or not.

Of course, it is better to use the tlsdontverifyserver = no option, but in practice there are many problems with this and the option does not work quite as intended; even if the certificate does not match the verification by ca.crt, Asterisk still allows us to make calls, even though there will be throw up errors. Therefore, it is recommended to use tlsdontverifyserver = yes.

Next, we go to the Asterisk console and reboot using the reload command (Figure 18). We can check that the TLS port is working with the command openssl s_client -connect 127.0.0.1:5061.

Now, let us make settings in Asterisk on the peer to work with TLS and SRTP, activate tls for the peer in sip.conf, and enable srtp (encryption). To do this, we add the line data in the client fields:

transport = tls
encryption = yes

After the call has been made, another icon appears, a padlock on an orange background, which indicates media traffic encryption (SRTP). Let us check these data in Wireshark (Figure 19).

It can be seen from Figures 19 and 20 that the main service information is now transmitted in encrypted form. Several criteria were identified by which it is possible to assess the degree of security of an IP network based on Asterisk PBX and, following them, increase the security of this network:

1. Authentication and authorization: in our case, users are configured with an access password, so that only authorized users have access to the PBX system and its functions.
2. Encryption: it uses the secure SIP protocol setting over TLS to secure communication between devices.
3. Firewall: the implementation of a firewall blocks unauthorized access to the PBX system.
4. Network segmentation: the segmentation of the PBX network from other networks minimizes the attack surface.

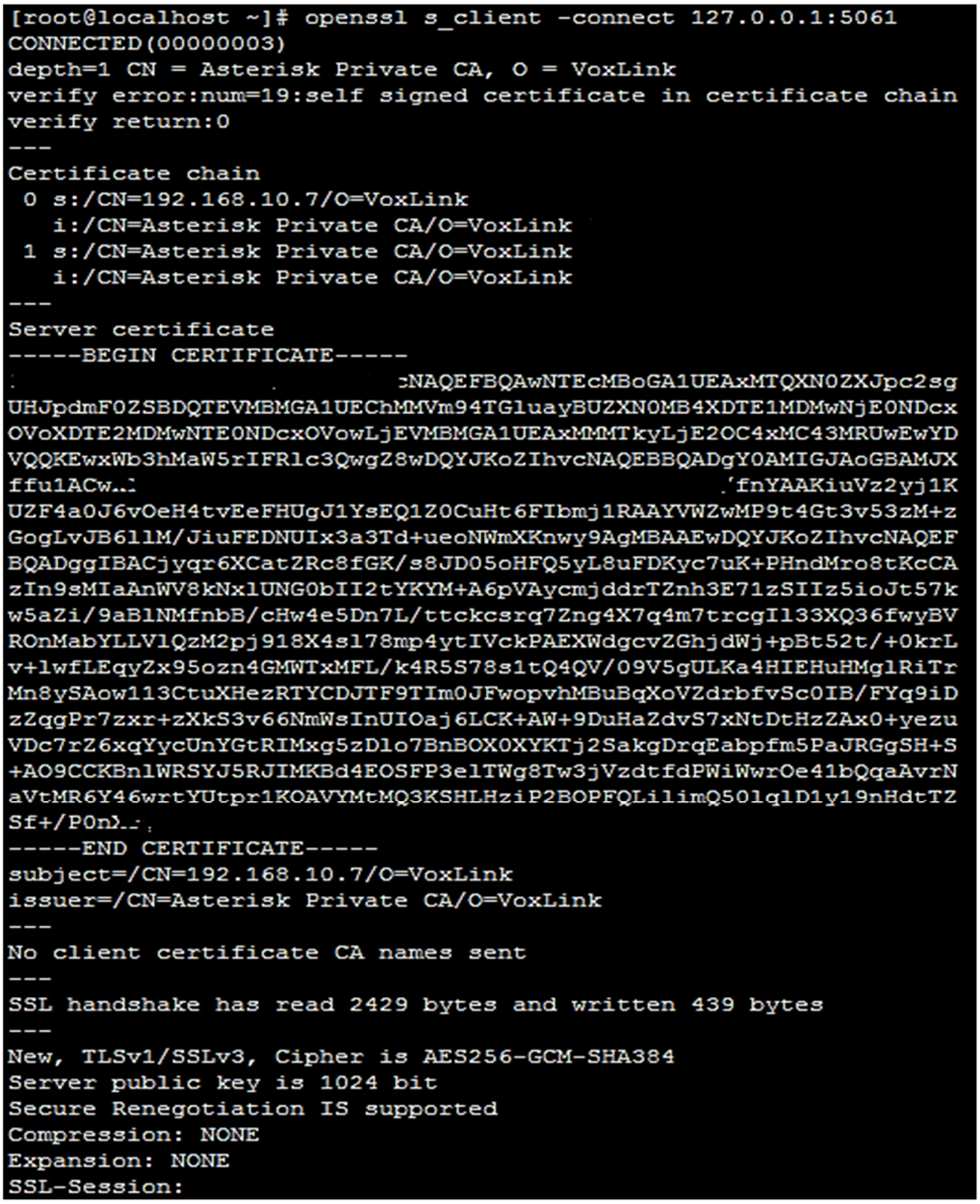

Figure 18. Certificate information software.

When operating an IP network, we recommend regular updates and maintenance, including keeping the PBX system and its components up to date with the latest security patches and configurations; monitoring and logging system activity to detect and respond to security incidents; regularly scanning the network and PBX systems for vulnerabilities and taking measures to eliminate them; and having a plan for quickly restoring service in the event of a security incident or other disruption.

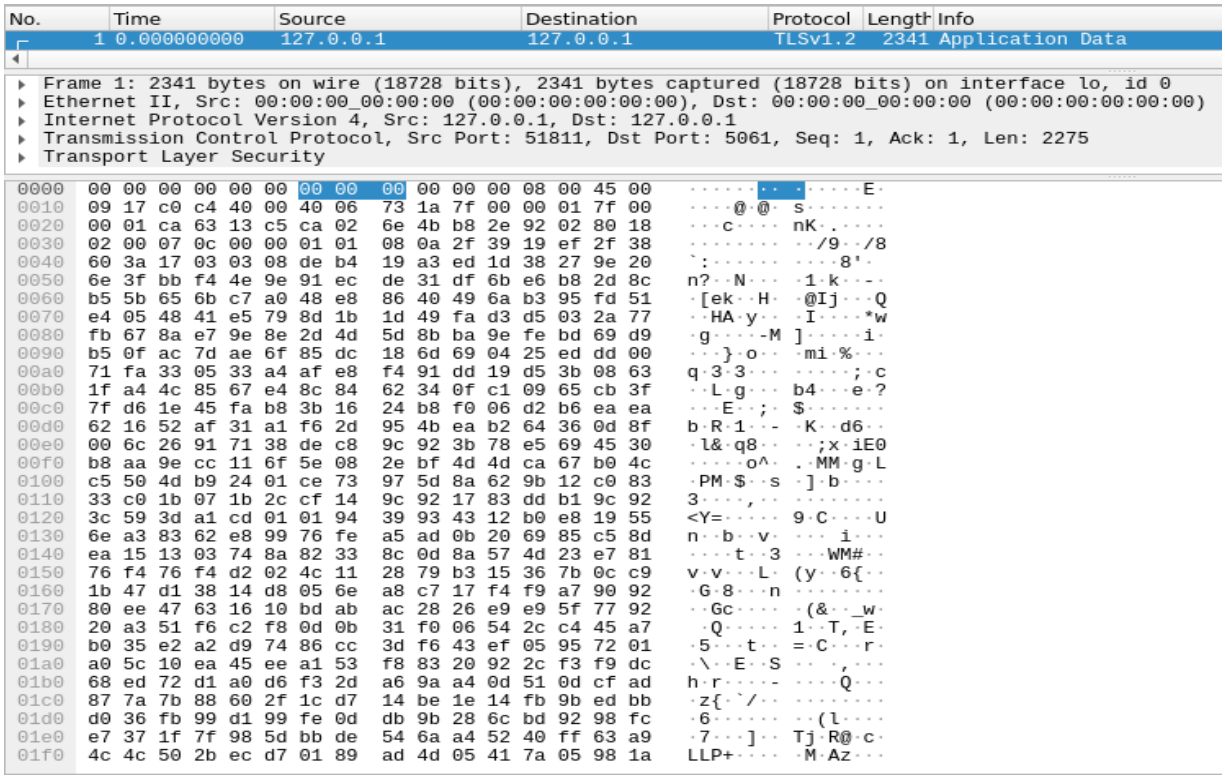

Figure 19. Transfer of encrypted session information.

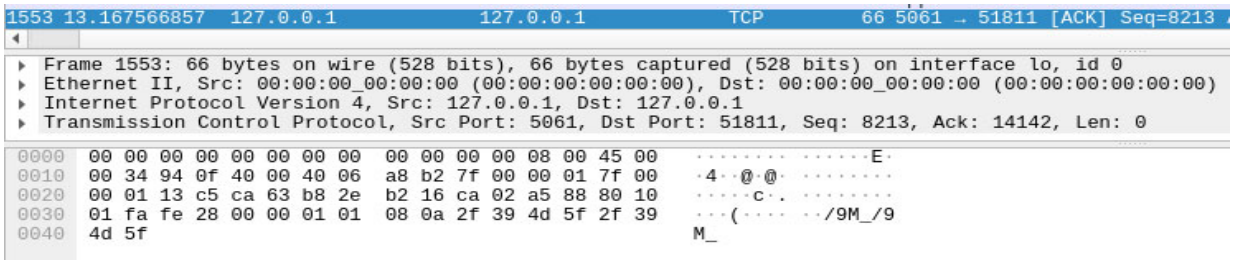

Figure 20. Transfer of encrypted media traffic.

4. Conclusions

This research aimed to address the challenges encountered while setting up a telecommunications IP network based on Asterisk PBX. The identified problems ranged from selecting the appropriate distribution kit and configuring the PBX and user equipment to ensuring secure transmission methods. These challenges can significantly hinder communication and development for small and medium-sized businesses.

The solution proposed was a configuration that utilizes Blink SIP clients and an Asterisk Private Branch Exchange (PBX) server for communication. The results of this study demonstrated that this configuration is effective and cost-efficient for small and medium-sized businesses since it eliminates the need to purchase additional equipment. However, to ensure the security of data transmission, this study recommends the implementation of the TLS protocol, which provides encryption and authentication, thereby safeguarding the communication traffic.

Solving issues related to Asterisk PBX configuration and offering a secure and cost-effective solution for small and medium-sized businesses, this research provides valuable information for improving the telecommunications infrastructure. The proposed configuration using Blink and Asterisk PBX can help enterprises expand their communication

capabilities without significant financial investment. The integration of the TLS protocol safeguarding confidential information against unauthorized access and potential threats.

This research contributes to the field of telecommunications by presenting a practical and secure approach to organizing IP networks, benefiting small and medium-sized businesses seeking efficient communication solutions. The results and recommendations of this research can assist decision-makers and specialists in choosing suitable communication setups and security measures, ultimately promoting business growth and productivity in the rapidly evolving landscape of modern telecommunications.

Author Contributions: Conceptualization, M.Y. and O.M.; methodology, M.Y.; software, M.Y.; validation, M.Y. and O.M.; formal analysis, A.M.; investigation, M.Y.; resources, A.B.; data curation, T.S.; writing—original draft preparation, M.Y., O.M. and T.S.; writing—review and editing, M.Y., O.M. and T.S.; visualization, O.M.; supervision, T.S.; project administration, T.S.; funding acquisition, M.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Science Committee of the Ministry of Science and Higher Education of the Republic of Kazakhstan, AP14871745, «Development of a method for improving the security of a telecommunications network based on IP-PBX Asterisk».

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Danylchenko, V.M.; Mykolaychuk, V.R.; Tkalenko, O.M.; Didkivskyy, A.S. Initial setup of PBX server based on Asterisk. *Connectivity* **2020**, *148*. [CrossRef]
2. Kumar, S.S.; Dhivyalekshmi, B.; Preethi, S.; Rengaraju, P. PBX implementation in LAN using Asterisk open source software. *Int. J. Appl. Eng. Res.* **2015**, *10*, 66–69.
3. Khan, B.M.; Fahad, M.; Bilal, R.; Khan, A.H. Performance Analysis of Raspberry Pi 3 IP PBX Based on Asterisk. *Electronics* **2022**, *11*, 3313. [CrossRef]
4. Al-Saadoon, G.M.W. Asterisk Open Source to Implement Voice over Internet Protocol. *Int. J. Comput. Sci. Netw. Secur.* **2009**, *9*, 39.
5. Rahman, M.M.; Islam, N.S. VoIP Implementation Using Asterisk PBX. *J. Bus. Manag.* **2014**, *15*, 47–53.
6. Chinna Rao, R.; Lakshmi, K.; Raja, C.; Varma, P.; Rao, G.R.K.; Patibandla, A. Real-Time Implementation and Testing of VoIP Vocoders with Asterisk PBX Using Wireshark Packet Analyzer. *J. Intercon. Netw.* **2022**, *22*, 2141030. [CrossRef]
7. Nuno, P.; Suarez, C.; Suarez, E.; Bulnes, F.G.; Calle, F.J.; Granda, J.C. A Diagnosis and Hardening Platform for an Asterisk VoIP PBX. *Secur. Commun. Netw.* **2020**, *2020*, 8853625. [CrossRef]
8. Martin, A.; Gamess, E.; Urribarri, D.; Gomez, J. A proposal for a high availability architecture for VoIP telephone systems based on open source software. *Int. J. Adv. Comput. Sci. Appl.* **2018**, *9*, 1–11. [CrossRef]
9. Maar, M.; Sitarova, J.; Orgon, M. Enterprise network with software Asterisk PBX based on the PLC technology. *Int. J. Adv. Telecommun. Electrotech. Sig. Syst.* **2017**, *6*, 1–10. [CrossRef]
10. Yessenbayev, Z.; Saparkhojayev, N.; Tibeyev, T. Implementation of the Intelligent Voice System for Kazakh. *J. Phys. Conf. Ser.* **2014**, *495*, 012043. [CrossRef]
11. Saenger, J.; Mazurczyk, W.; Keller, J.; Caviglione, L. VoIP network covert channels to enhance privacy and information sharing. *Fut. Gener. Compu. Syst.* **2020**, *111*, 96–106. [CrossRef]
12. Zhang, L.; Hu, X.; Rasheed, W.; Huang, T.; Zhao, C. An Enhanced Steganographic Code and Its Application in Voice-Over-IP Steganography. *IEEE Access* **2019**, *7*, 97187–97195. [CrossRef]
13. Lomotey, R.K.; Deters, R. Intrusion Prevention in Asterisk-Based Telephony System. In Proceedings of the 2014 IEEE International Conference on Mobile Services, Anchorage, AK, USA, 27 June–2 July 2014; pp. 116–123. [CrossRef]
14. Khan, N.A.; Chan, A.S.; Saleem, K.; Bhutto, Z.; Ayaze, H. VoIP QoS analysis over asterisk and Axon servers in LAN environment. *Int. J. Adv. Comput. Sci. Appl.* **2019**, *10*, 548–554. [CrossRef]
15. Kim, W.; Song, T.; Kim, T.; Park, H.; Pack, S. VoIP Capacity Analysis in Full Duplex WLANs. *IEEE Transact. Vehicular Technol.* **2017**, *66*, 11419–11424. [CrossRef]
16. Sanchez-Iborra, R.; Cano, M.-D.; Garcia-Haro, J. Performance Evaluation of BATMAN Routing Protocol for VoIP Services: A QoE Perspective. *IEEE Transact. Wirel. Commun.* **2014**, *13*, 4947–4958. [CrossRef]
17. Ali, S.R. Reliability Analysis of VoIP System. In *Next Generation and Advanced Network Reliability Analysis: Using Markov Models and Software Reliability Engineering*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 211–244. [CrossRef]
18. Yeh, W.C. Search for MC in modified networks. *Comput. Operat. Res.* **2001**, *28*, 177–184. [CrossRef]

19. Pal, D.; Triyason, T.; Vanijja, V. Asterisk server performance under stress test. In Proceedings of the IEEE 17th International Conference on Communication Technology (ICCT), Chengdu, China, 27–30 October 2017; pp. 1967–1971. [CrossRef]
20. Costa, L.R.; Nunes, L.S.N.; Bordim, J.L.; Nakan, K. Asterisk PBX Capacity Evaluation. In Proceedings of the IEEE International Parallel and Distributed Processing Symposium Workshop, Hyderabad, India, 25–29 May 2015; pp. 519–524. [CrossRef]
21. Rughinis, R.V.; Iconaru, C. A Practical Analysis of Asterisk SIP Server Performance. In Proceedings of the 7th RoEduNet Internaional Conference, Cluj-Napoca, Romania, 28–30 August 2008; pp. 61–64.
22. Muntaka, S.A.; Hussein, F.; Sarfo, P. Implementation of an IP Telephony System Based on Asterisk PBX. *Int. J. Comput. Appl.* **2019**, *177*, 8887. [CrossRef]
23. Murkute, P.V.; Deshmukh, V.M. Implementing the VOIP Communication Principles Using Raspberry Pi as Server. *Int. J. Comput. Appl.* **2015**, *124*, 34–38. [CrossRef]
24. Karapantazis, S.; Pavlidou, F.-N. VoIP: A comprehensive survey on a promising technology. *Comput. Netw.* **2009**, *53*, 2050–2090. [CrossRef]
25. Konshin, S.V.; Yakubova, M.Z.; Nishanbayev, T.N.; Manankova, O.A. Research and development of an IP network model based on PBX asterisk on the opnet mod eler simulation package. In Proceedings of the International Conference on Information Science and Communications Technologies, ICISCT 2020, Tashkent, Uzbekistan, 4–6 November 2020. [CrossRef]
26. Voznak, M.; Rezac, F. Threats to Voice over IP communications systems. *WSEAS Transact. Comput.* **2010**, *9*, 1348–1358.
27. Ganesan, V.; Msk, M. A scalable detection and prevention scheme for voice over internet protocol (VoIP) signaling attacks using handler with Bloom filter. *Int. J. Netw. Manag.* **2018**, *28*, e1995. [CrossRef]
28. Rehman, U.U.; Abbasi, A.G. Secure layered architecture for Session Initiation Protocol based on SIPSSO: Formally proved by Scyther. In Proceedings of the 12th International Conference on Information Technology—New Generations (ITNG 2015), Las Vegas, NV, USA, 13–15 April 2015; pp. 185–190. [CrossRef]
29. Jung, S. CAPTCHA-based DDoS defense system of call centers against zombie smart-phone. *Int. J. Secur. Appl.* **2012**, *6*, 29–36.
30. Barison, D.; Miani, R.S.; De Souza Mendes, L. Evaluation of quality and security of a VoIP network based on asterisk and Open VPN. In Proceedings of the International Conference on Security and Cryptography 2009, Milan, Italy, 7–10 July 2009; pp. 144–147. [CrossRef]
31. Abualhaj, M.M.; Al-Tahrawi, M.M.; Al-Khatib, S.N. Performance evaluation of VoIP systems in cloud computing. *J. Eng. Sci. Technol.* **2019**, *14*, 1398–1405.
32. Wu, H.; Zhu, C.; Cheng, G. Real-Time Application Identification of RTC Media Streams via Encrypted Traffic Analysis. In Proceedings of the International Conference on Computer Communications and Networks (ICCCN 2022), Honolulu, HI, USA, 25–28 July 2022. [CrossRef]
33. Shen, C.; Nahum, E.; Schulzrinne, H. The impact of TLS on SIP server performance. In Proceedings of the IPTComm 2010—Principles, Systems and Applications of IP Telecommunications 2010, Munich, Germany, 2–3 August 2010. [CrossRef]
34. Bhujangaa Rao, S.; Apoorva, C. Implementation of RFC 5359 SIP (VoIP) services on asterisk PBX. *Int. J. Innov. Technol. Expl. Eng.* **2019**, *8*, 15–21.
35. Kulin, M.; Kazaz, T.; Mrdovic, S. SIP server security with TLS: Relative performance evaluation. In Proceedings of the 9th International Symposium on Telecommunications, BIHTEL 2012—Proceedings, Sarajevo, Bosnia and Herzegovina, 25–27 October 2012; pp. 1–6. [CrossRef]
36. Lara-Cueva, R.A.; Pazmino, S.; Acosta, F. Performance evaluation of an Asterisk PBX prototype Beaglebone Black based. In Proceedings of the 17th Iberian Conference on Information Systems and Technologies—CISTI 2022, Madrid, Spain, 22–25 June 2022. [CrossRef]
37. Asif, K.; Eshtiak, A.; Sami, A.; Bharanidharana, S.; Pronab, G. Mitigating the Latency Induced Delay in IP Telephony through an Enhanced De-Jitter Buffer. In *Mobile Computing and Sustainable Informatics*; Springer: Singapore, 2022; pp. 1–16. [CrossRef]
38. Manankova, O.A.; Yakubov, B.M.; Serikov, T.G.; Yakubova, M.Z.; Mukasheva, A.K. Analysis and research of the security of a wireless telecommunications network based on the IP PBX Asterisk in an Opnet environment. *J. Theoret. Appl. Inform. Technol.* **2021**, *99*, 3617–3630.
39. Serikov, T.G.; Yakubova, M.Z.; Mekhtiev, A.D.; Yugay, V.V.; Muratova, A.K.; Razinkin, V.P.; Okhorzina, A.V.; Yurchenko, A.V.; Alkina, A.D. The analysis and modeling of efficiency of the developed telecommunication networks on the basis of IP PBX asterisk now. In Proceedings of the 11th International Forum on Strategic Technology, IFOST 2016, Novosibirsk, Russia, 1–3 June 2016; pp. 510–515. [CrossRef]
40. Zhang, Z.; De Luca, G.; Archambault, B.; Chavez, J.; Rice, B. Traffic Dataset and Dynamic Routing Algorithm in Traffic Simulation. *J. Artif. Intell. Technol.* **2022**, *2*, 111–122. [CrossRef]
41. Samanta, R.K.; Sadhukhan, B.; Samaddar, H.; Sarkar, S.; Koner, C.; Ghosh, M. Scope of machine learning applications for addressing the challenges in next-generation wireless networks. *CAAI Trans. Intell. Technol.* **2022**, *7*, 395–418. [CrossRef]
42. Montazerolghaemm, A. Softwarization and virtualization of VoIP networks. *J. Supercomput.* **2022**, *78*, 14471–14503. [CrossRef]
43. Dong, F.; Deng, B.; Yu, H.; Xie, W.; Xu, H.; Gu, Z. An Asterisk-shaped Patch Attack for Object Detection. In Proceedings of the 7th IEEE International Conference on Data Science in Cyberspace, DSC 2022, Guilin, China, 11–13 July 2022; pp. 126–133. [CrossRef]
44. Surasak, T.; Scott, H.C.-H. Enhancing VoIP Security and Efficiency Using VPN. In Proceedings of the International Conference on Computing, Networking and Communications, ICNC 2019, Honolulu, HI, USA, 18–21 February 2019; pp. 180–184. [CrossRef]

45. Hooshmand, M.K.; Doreswamy, H. Network anomaly detection using deep learning techniques. *CAAI Trans. Intell. Technol.* **2022**, *7*, 228–243. [CrossRef]
46. Romanets, I.; Sachenko, A.; Dubchak, L. Method of Protection against Traffic Termination in VoIP. In Proceedings of the 10th International Conference on Electronics, Computers and Artificial Intelligence, ECAI 2018, Iasi, Romania, 28–30 June 2018. [CrossRef]
47. Jama, A.M.; Khalifa, O.; Subramaniam, O.; Kumar, N. Novel Approach for IP-PBX Denial of Service Intrusion Detection Using Support Vector Machine Algorithm. *Int. J. Commun. Netw. Inf. Secur.* **2021**, *13*, 249–257. [CrossRef]
48. Wang, S.; Tang, W.; Liu, C.; Li, B.; Tang, M.; Wen, M. Digital Image Correlation Measurement of the Deformation and Failure in PBX Brazilian Discs Reinforced with CFRP Patches. *Propellants Explosives Pyrotechnics* **2021**, *46*, 548–554. [CrossRef]
49. McInnes, N.; Wills, G. The VoIP PBX Honeypot Advance Persistent Threat Analysis. In Proceedings of the International Conference on Internet of Things, Big Data and Security, IoTBDS—2021, Online Streaming, 23–25 April 2021; pp. 70–80. [CrossRef]