

Project overview for portfolio upload

Project Title:

Designing, Implementing & Securing a VoIP Communication Network (GitHub upload)

Project Description:

Design of the architecture of a communication network using VoIP technologies. The network is implemented on a virtual network. The network was composed of three virtual machines, two machines as a client and one as a VoIP server which had been set up in **Asterisk**. This project enabled a voice call between two nodes along with some additional features. This was done through installation and configuration of Asterisk as outlined in the document.

Project Tooling:

- Ubuntu
- Asterisk
- VMWare
- Linnphone

Project processes and techniques:

- Understanding network design and architecture
- Configuration of VoIP server and clients
- Implementing security of VoIP server

Grade: B+

Appendices have been removed to avoid plagiarism from future students, however I am happy to provide additional information or discuss the work further if desired.

Disclaimer: *I do not give permission for this work to be copied, those who do may be liable for plagiarism.*

Table of Contents

Introduction	3
Methodology	3
Selection of VoIP Server (Asterisk)	3
Configuration of Virtual Machines	3
Implementation of SIP for communication	5
Results	6
Implemented	6
Challenges faced during implementation.....	7
Security	8
Vulnerabilities.....	8
Planned security implementation.	8
Conclusion	9
References	10
Appendix	11
Appendix Part 1 – Detailed Configs	11
1.1 pjsip.conf.....	11
1.2 pjsip.conf (continued)	12
1.3 extensions.conf	13
1.4 Voicemail.conf	14
1.5 musiconhold.conf	14
1.6 Endpoints	14
Appendix part 2 – Smartphone clients	15
2.1 – Lin phone interface.....	15
2.2 – Logging into Sip user.....	16
2.3 - Calling Sip user	17

Introduction

Modern online communication has been revolutionised by the Voice over Internet Protocol (VoIP) (Poole, 2005). VoIP can be credited as the protocol that has enabled organisations such as Skype, Zoom and Ring Central to be created. Working, socialising & education are all reliant on the necessity to communicate online, therefore understanding how this technology functions & how the communication between individuals can stay secured is essential for online communication to continue (Madoc-Jones and Parrott, 2005). The following report highlights how it is possible for an individual to develop the skills and knowledge to design & implement a VoIP network from universally available technology found online. For this network to remain fully secure it is important that the methodology involved in implementing security principles are highlighted accordingly.

Methodology

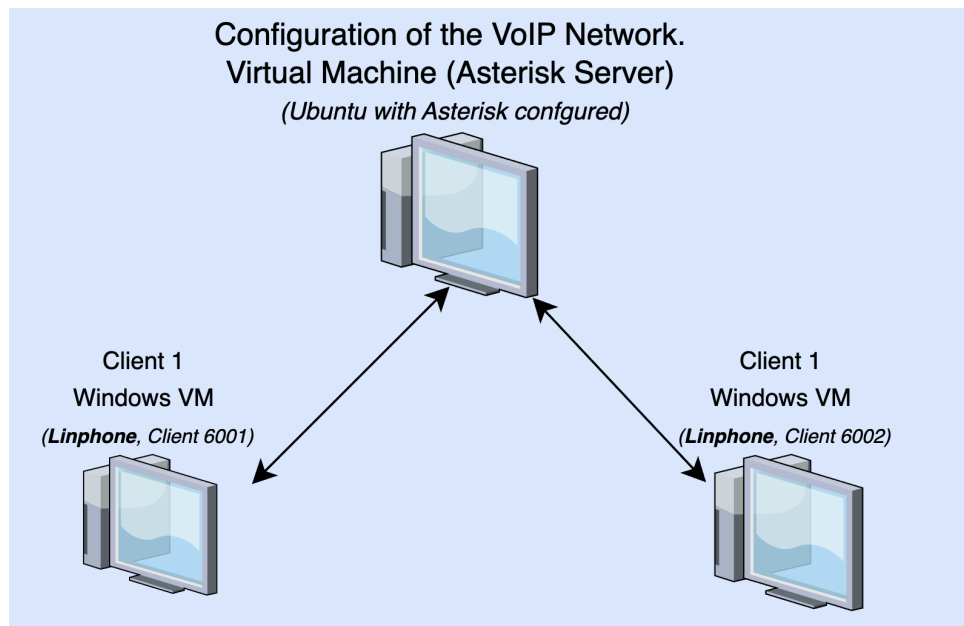
Selection of VoIP Server (Asterisk)

Asterisk was selected as the technology for the creation of the VoIP network. This choice was a result of the wide array of uses Asterisk has been used for such as being selected for small rural airports enabling them to have the ability to communicate (Hendrawan and Aditya, 2019). Asterisk is a feature heavy package that allows for a multitude of technologies to be built within the VoIP network. Some of the technologies available include voicemail, conference calling, music on hold & video calling, some of which are implemented in the created network. Manual installation of Asterisk was chosen over a GUI alternative to maintain a hands on, customisable approach to implementing the features.

Configuration of Virtual Machines

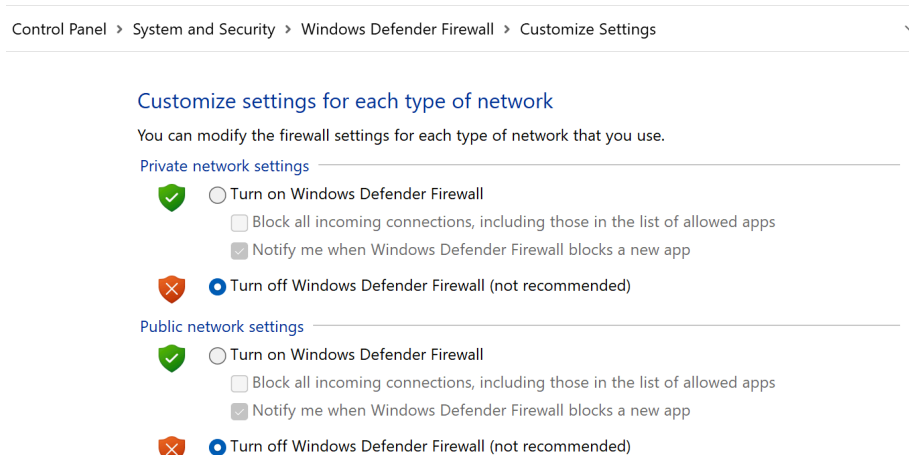
To create the network, several Virtual Machines (VM's) were deployed to simulate the server, and two clients. For the VM's to work in tandem the network adapter of each VM was required to be the same. The VM that was used to create the network was Parallels. In Parallels, the default network settings are set to a 'bridged network'. The bridged network enabled the two client VM's and the server VM to be able to ping each other, to validate the connection. Figure 1 below is a representation of how each client node will communicate with the server; and in

turn, the server in turn will relay this communication to each node. Each client relies on the Asterisk server to handle SIP signalling, the communicating being initiated.



[Figure 1: Configuration of VoIP network]

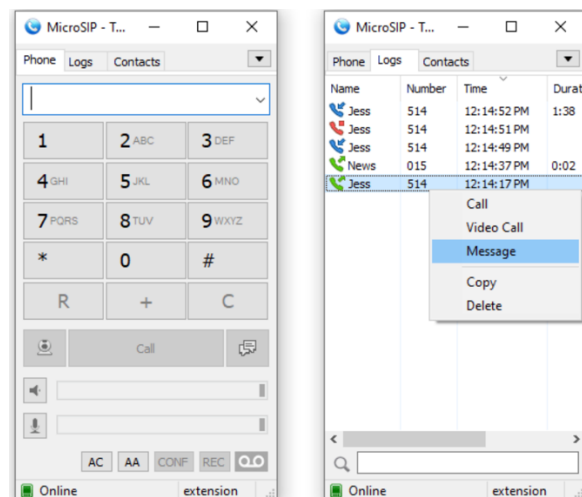
When enabling the communication between the two client nodes and the asterisk server it was important to have disabled all firewalls on the virtual machines. Firewalls can often monitor and put unnecessary restrictions on the network traffic that is allowed to pass too and from a machine. Because of this, disabling the firewalls was an essential step in ensuring the network functioned correctly. Figure 2 shows how the firewall should look on windows. The following command should be executed in Ubuntu to disable the firewall, '`sudo systemctl enable ufw`'.



[Figure 2]

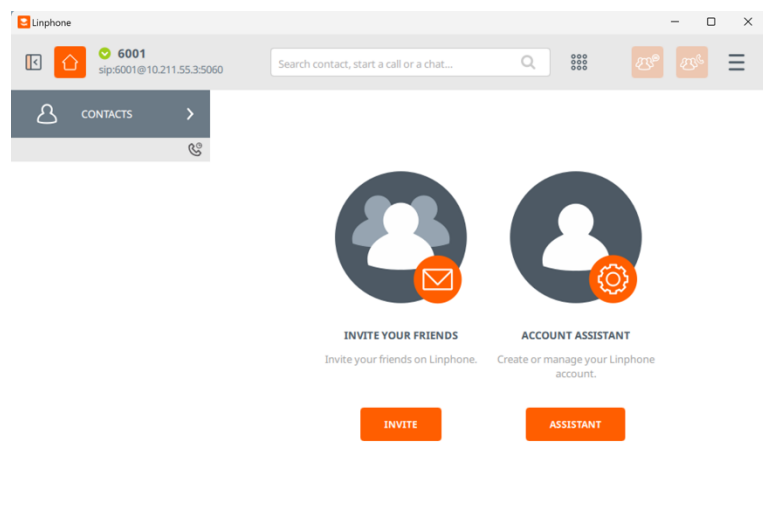
Implementation of SIP for communication

The decision for software used by the SIP nodes can be specific to the individual operating the clients. For example, different technical requirements or technical ability that can be a factor in this decision making. In this situation the developer of the network tested Zoiper, Linnphone and MicroSIP. Many users, including the network developer could decide that Micro SIP's barebones user interface is not desired as part of the communication service.



[Figure 3: MicroSIP Interface – source: <https://www.microsip.org/>]

Therefore, Zoiper or Linphone which have more modern looking interfaces might be more desired, particularly in a business environment. Regarding the creation of the VoIP network outlined by the report, Linphone's ease of use, functionality to connect directly to the sip users and overall, the graphical interface proved that Linphone was the more desirable of the choices.



[Figure 4: Linphone Client from VoIP network]

Results

Implemented

Highlighted in the accompanying video, the network allowed for two primary requirements. Firstly, and most importantly it allowed the two clients to use voice call to one another. Secondly, conference calling was enabled, allowing both users to dial in to a specified conference call room (in this case the conference room was 7000). These two features were very intertwined. The phone call required correct configuration of the *pjsips.conf* and the *extensions.conf*. The conference call required further alteration of these two config files. See figures 5, 6 and 7 below for the primary commands added within these config files to enable voice call & conference call between these two SIP clients. The outcome was a fully functional network that allowed for voice call between two SIP clients and the ability for each SIP client to dial into a conference call (with code 7000).

```
parallels@ubuntu-linux-22-04-02-desktop: /etc/asterisk
GNU nano 6.2 pjsip.conf
;
; A few more transports to pick from, and some related options below then.
;
;transport=transport-tls
;media_encryption=sdes
;transport=transport-udp-lpvo
;transport=transport-udp-nat
;direct_media=no
;
; MWI related options
;
;aggregate_mwi=yes
;mailbox=6001@default,7001@default
;mwi_from_user=6001
;
; Extension and Device state options
;
;device_state_busy_at=1
;allow_subscribe=yes
;sub_min_expiry=30
;
; STIR/SHAKEN support.
;
;stir_shaken=no
;stir_shaken_profile=ny_profile
;
[6001]
type=auth
auth_type=userpass
password=6001
username=6001
;
[6002]
type=auth
auth_type=userpass
password=6002
username=6002
;
[6001]
type=aor
max_contacts=3
contact=sip:6001@10.211.55.3:5060
;
[6002]
type=aor
max_contacts=3
contact=sip:6002@10.211.55.3:5060
;
=====ENDPOINT BEHIND NAT OR FIREWALL=====
;
; This example assumes your transport is configured with a public IP and the
;
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^J Execute  ^G Location ^U Undo
^X Exit      ^R Read File ^M Replace   ^P Paste     ^AD Justify ^L Go To Line ^RE Redo
```

```
[6001]
type=endpoint
transport=transport-udp
context=from-internal
disallow=all
allow=ulaw
allow=gsm
auth=6001
aors=6001
;
[6002]
type=endpoint
transport=transport-udp
context=from-internal
disallow=all
allow=ulaw
allow=gsm
auth=6002
aors=6002
```

[Figure 5 & 6 – Configuring the *pjsips.conf* for voice call]

```

[default]
exten => 6001,1,Dial(SIP/6001,30)
    same => n,VoiceMail(6001,u)
    same => n,Hangup()

exten => 6002,1,Dial(SIP/6002,30)
    same => n,VoiceMail(6002,u)
    same => n,Hangup()

[from-internal]
exten => 6001,1,Dial(PJSIP/6001,30,U)
exten => 6002,1,Dial(PJSIP/6002,30,U)

;voicemail
exten => *98,1,VoiceMailMain()
exten => *98,n,Hangup()

;conference
exten => 6001,1,Dial(PJSIP/6001,30)
exten => 6001,n,ConfBridge(1000)

exten => 6002,1,Dial(PJSIP/6002,30)
exten => 6002,n,ConfBridge(1000)

exten => 7000,1,ConfBridge(1000)

;music on hold
exten => 6001,1,Dial(PJSIP/6001,30)
exten => 6001,n,MusicOnHold(default)

exten => 6002,1,Dial(PJSIP/6002,30)
exten => 6002,n,MusicOnHold(default)

```

[Figure 7 – Highlighting key inputs on *extensions.conf* for the conference call]

Challenges faced during implementation.

During the implementation of the VoIP network, obtaining a deep understanding of the functionality of Asterisk and SIP clients proved difficult. Firstly, the configuration files on Asterisk. The documentation found on the Asterisk website was highly relied upon during the first configuration of the server. Furthermore, grappling the difficulties of utilising the asterisk configs to set up and register SIP users with the clients proved to be time consuming due to the nomenclature of the configurations file, lengthy experimentation was required.

However, this experimentation and research phase did not help deliver solutions to the failed implementation of music on hold & half implementation of the voicemail. The 'Music on Hold' config file was altered but did not play when a client put another on hold.

Additionally, the voicemail service, whilst it allowed users to phone their voicemail box it did not enable them to leave a voicemail with another user. The challenges faced unfortunately left many features to be implemented fully at a further date.

Security

Vulnerabilities

Developing security for the VoIP network is vital in ensuring that sensitive communication data is kept private to maintain the integrity of the system. To correctly secure the VoIP network created, it is essential to learn of the vulnerabilities. Eavesdropping is one of the biggest vulnerabilities faced by a VoIP network. Eavesdropping is when an outside, undesired party intercepts & listens into the communication between clients (Keromytis, 2010).

Caller ID spoofing is becoming a regular occurrence today and as a result there has been a great shift towards using AI for detecting this type of spoofing (Tianxiang Chen, Avrosh Kumar, Parav Nagarsheth, Ganesh Sivaraman, Elie Khoury, 2020). However, it is difficult for smaller businesses and less technologically orientated individuals to be on the lookout for this.

Finally, a vulnerability which is commonplace, particularly in networks like the network created is default settings. Default settings is when the creator of the network will leave the settings that have been implemented on start-up of the software used. Default settings in config files are common due to the complex nature of such tasks, creating a large security risk for the server (Shah and Sandvig, 2008).

Planned security implementation.

To counteract the vulnerabilities outlined, it is essentially to deploy correct security implementation. This necessity for security is only increased as the network grows. One of the most effective ways of implementing security procedures is through implementing the 'Fail2Ban' framework onto a network. This security framework is designed to counteract any attempt of brute forcing passwords. Deploying the Fail2Ban framework provides a level of security to stop intrusion onto private systems (Ford *et al.*, 2016).

In addition, to Fail2Ban, implementing SSL & SRTP protocols can be a modern and important strategy for mitigating the potential leak of sensitive communication data. SRTP, or Secure

Real-Time Transport Protocol is very relevant today. This is because of a large focus on multiple clients connecting to a server, whether it be for a livestream or conference call. SRTP can allow for packets to be sent quickly ensuring that if the VoIP server is updated to include video call as a feature, then it would be safely encrypted.

Finally, server handling is one security method that can directly solve the issues of default settings. Server hardening involves taking a deeper look at the settings of a server. This can help the server administrator identify the server settings that will allow them to increase a server's overall security. For Example, introducing a password policy that includes regular changes, minimum characters and a password not relevant to the network itself.

Conclusion

Developing a VoIP network using Asterisk can be a difficult challenge for administrators who have not previously developed or maintained a similar type of server. However, by learning the importance of configuration files & maintaining correct, organised naming practice within these configuration files it can be rewarding to develop a highly customisable and expandable network. Whilst some features remain still to be implemented, understanding how the network can be weak to attacks & how to protect against attacks through methods such as Fail2Ban, SRTP or server handling is highly advantageous.

References

- Ford, M., Mallery, C., Palmasani, F., Rabb, M., Turner, R., Soles, L. and Snider, D. (2016) 'A process to transfer Fail2ban data to an adaptive enterprise intrusion detection and prevention system ', *Proceedings of IEEE Southeastcon*, , pp. 1-4. doi: 10.1109/SECON.2016.7506771 <https://ieeexplore.ieee.org/document/7506771>.
- Hendrawan, H. and Aditya, B. (2019) 'Asterisk and Radio Over IP Integration at Voice Communication System Air Traffic Control', *2019 IEEE 13th International Conference on Telecommunication Systems, Services, and Applications (TSSA)*, , pp. 271-276. doi: 10.1109/TSSA48701.2019.8985483 <https://ieeexplore.ieee.org/document/8985483>.
- Keromytis, A.D. (2010) 'Voice-over-IP Security: Research and Practice', *IEEE security & privacy*, 8(2), pp. 76-78. doi: 10.1109/MSP.2010.87 <https://ieeexplore.ieee.org/document/5439534>.
- Madoc-Jones, I. and Parrott, L. (2005) 'Virtual Social Work Education—Theory and Experience ', *Social work education*, 24(7), pp. 755-768. doi: 10.1080/02615470500238678 <https://www.tandfonline.com/doi/abs/10.1080/02615470500238678>.
- Poole, I. (2005) 'What exactly is VoIP?', *Communications engineer*, 3(2), pp. 44-45. doi: 10.1049/ce:20050209 .
- Shah, R.C. and Sandvig, C. (2008) 'SOFTWARE DEFAULTS AS DE FACTO REGULATION The case of the wireless internet', *Information, Communication & Society*, 11(1), pp. 25-46. doi: 10.1080/13691180701858836 <https://doi.org/10.1080/13691180701858836>.
- Tianxiang Chen, Avrosh Kumar, Parav Nagarsheth, Ganesh Sivaraman, Elie Khouury (2020) 'Generalization of Audio Deepfake Detection ', , pp. 132-137. doi: 0.21437/Odyssey.2020-19 .

Appendix

Appendix Part 1 – Detailed Configs

1.1 pjsip.conf

```
parallels@ubuntu-linux-22-04-02-desktop: /etc/asterisk
GNU nano 6.2 pjsip.conf
;type=aor
;max_contacts=1

;=====ENDPOINT CONFIGURED FOR USE WITH A SIP PHONE=====
;
; This example includes the endpoint, auth and aor configurations. It
; requires inbound authentication and allows registration, as well as references
; a transport that you'll need to uncomment from the previous examples.
;
; Uncomment one of the transport lines to choose which transport you want. If
; not specified then the default transport chosen is the first compatible transport
; in the configuration file for the contact URL.
;
; Modify the "max_contacts=" line to change how many unique registrations to allow.
;
; Use the "contact=" line instead of max_contacts= if you want to statically
; define the location of the device.
;
; If using the TLS enabled transport, you may want the "media_encryption=sdes"
; option to additionally enable SRTP, though they are not mutually inclusive.
;
; If this endpoint were remote, and it was using a transport configured for NAT
; then you likely want to use "direct_media=no" to prevent audio issues.

[6001]
type=endpoint
transport=transport-udp
context=from-internal
disallow=all
allow=ulaw
allow=gsm
auth=6001
aors=6001

[6002]
type=endpoint
transport=transport-udp
context=from-internal
disallow=all
allow=ulaw
allow=gsm
auth=6002
aors=6002

;
; A few more transports to pick from, and some related options below them.
;
;transport=transport-tls
;media_encryption=sdes

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line  M-E Redo
```

1.2 pjsip.conf (continued)

```
parallels@ubuntu-linux-22-04-02-desktop: /etc/asterisk
GNU nano 6.2 pjsip.conf
;
; A few more transports to pick from, and some related options below them.
;
;transport=transport-tls
;media_encryption=sdes
;transport=transport-udp-ipv6
;transport=transport-udp-nat
;direct_media=no
;
; MWI related options
;aggregate_mwi=yes
;mailboxes=6001@default,7001@default
;mwi_from_user=6001
;
; Extension and Device state options
;
;device_state_busy_at=1
;allow_subscribe=yes
;sub_min_expiry=30
;
; STIR/SHAKEN support.
;
;stir_shaken=no
;stir_shaken_profile=my_profile

[6001]
type=auth
auth_type=userpass
password=6001
username=6001

[6002]
type=auth
auth_type=userpass
password=6002
username=6002

[6001]
type=aor
max_contacts=3
contact=sip:6001@10.211.55.3:5060

[6002]
type=aor
max_contacts=3
contact=sip:6002@10.211.55.3:5060

;=====ENDPOINT BEHIND NAT OR FIREWALL=====
;
; This example assumes your transport is configured with a public IP and the
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location  M-U Undo
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line M-E Redo
```

1.3 extensions.conf

```
[default]

exten => 6001,1,Dial(SIP/6001,30)
    same => n,VoiceMail(6001,u)
    same => n,Hangup()

exten => 6002,1,Dial(SIP/6002,30)
    same => n,VoiceMail(6002,u)
    same => n,Hangup()

[from-internal]

exten => 6001,1,Dial(PJSIP/6001,30,U)

exten => 6002,1,Dial(PJSIP/6002,30,U)

;voicemail
exten => *98,1,VoiceMailMain()
exten => *98,n,Hangup()

;conference
exten => 6001,1,Dial(PJSIP/6001,30)
exten => 6001,n,ConfBridge(1000)

exten => 6002,1,Dial(PJSIP/6002,30)
exten => 6002,n,ConfBridge(1000)

exten => 7000,1,ConfBridge(1000)

;music on hold

exten => 6001,1,Dial(PJSIP/6001,30)
exten => 6001,n,MusicOnHold(default)

exten => 6002,1,Dial(PJSIP/6002,30)
exten => 6002,n,MusicOnHold(default)
```

1.4 Voicemail.conf

```
GNU nano 6.2
;
; Voicemail Configuration
;

[general]

spool => ~/Desktop/

[default]

6001 => 1111,JohnDoe
6002 => 2222,JaneDoe
```

1.5 musiconhold.conf

```
[default]
mode=files
directory=/var/lib/asterisk/holdmusic/holdmusic.mp3

; Music on Hold -- Sample Configuration
;

[general]
```

1.6 Endpoints

```
=====
Running as user 'parallels'
Running under group 'asterisk'
Connected to Asterisk 18.20.1 currently running on ubuntu-linux-22-04-02-desktop (pid = 1546)
ubuntu-linux-22-04-02-desktop*CLI> show pjsip endpoints
No such command 'show pjsip endpoints' (type 'core show help show pjsip' for other possible commands)
ubuntu-linux-22-04-02-desktop*CLI> pjsip show endpoints

Endpoint: <Endpoint/CID.....> <State.....> <Channels.>
  I/OAuth: <AuthId/UserName.....>
  Aor: <Aor.....> <MaxContact>
  Contact: <Aor/ContactUri.....> <Hash....> <Status> <RTT(ms)..>
Transport: <TransportId.....> <Type> <cos> <tos> <BindAddress.....>
Identify: <Identify/Endpoint.....>
  Match: <criteria.....>
  Channel: <ChannelId.....> <State.....> <Time.....>
  Exten: <DialedExten.....> CLCID: <ConnectedLineCID.....>
=====

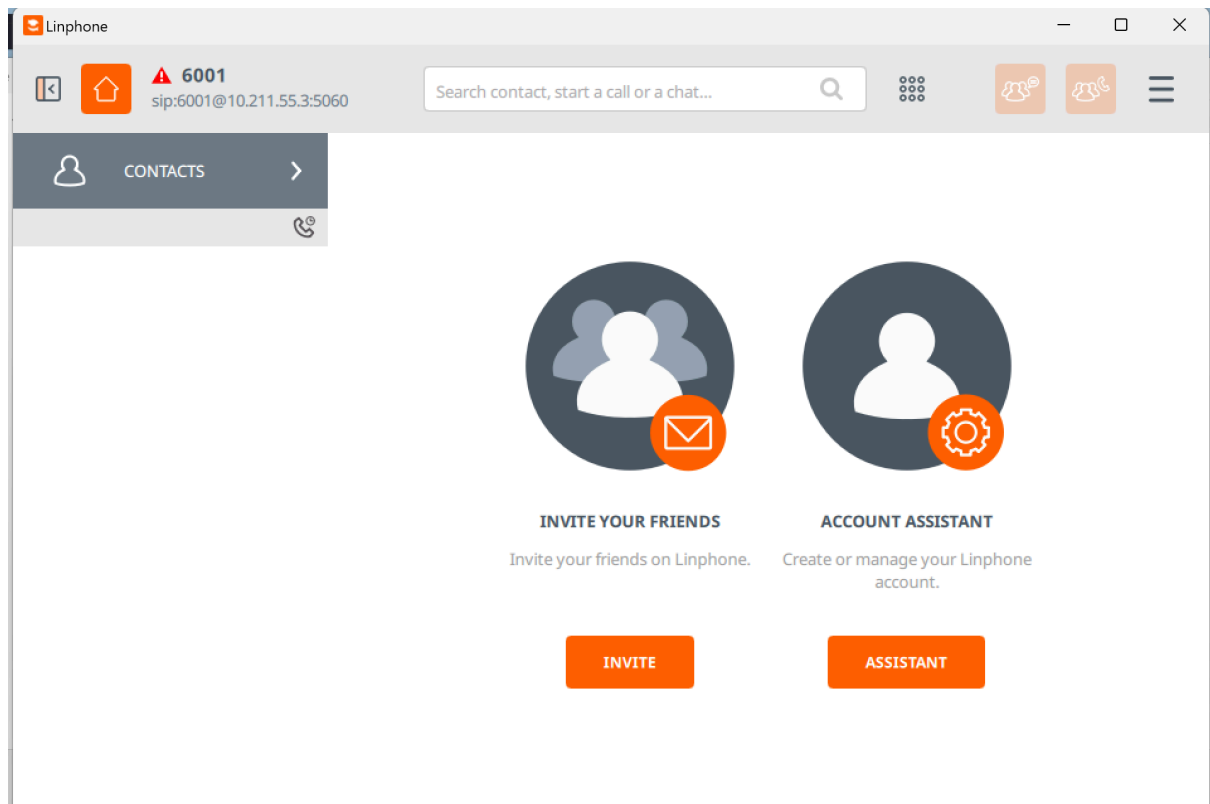
Endpoint: 6001                                Not in use    0 of inf
  InAuth: 6001/6001
  Aor: 6001
  Contact: 6001/sip:6001@10.211.55.3:5060      6e12207bd9 NonQual    nan
Transport: transport-udp                      udp          0          0 0.0.0.0:5060

Endpoint: 6002                                Not in use    0 of inf
  InAuth: 6002/6002
  Aor: 6002
  Contact: 6002/sip:6002@10.211.55.3:5060      8a9b98a9c0 NonQual    nan
Transport: transport-udp                      udp          0          0 0.0.0.0:5060

Objects found: 2
```


Appendix part 2 – Smartphone clients

2.1 – Lin phone interface



2.2 – Logging into Sip user


Presence status

 Available

▼

Active account

sip:jamie_rice@[fdb2:2c26:f4e4:0:eda2:42d2:b...



sip:6001@10.211.55.3:5060

OK

2.3- Calling Sip user

