Using Asterisk at Home

An introduction to open source telephony

What is Asterisk

- Asterisk is an open-source toolkit that allows you to software that interacts with telephone systems.
- Originally designed as a PBX (Private Branch eXchange) - an internal telephone system - but now can do much more.
- Created by Mark Spencer (Huntsville native.) He needed a phone system for his Linux support business but couldn’t afford a system from 3Com or another vendor.
- “It’s just software!” - Mark

What Can It Do?

- Just about anything involving telephones!
- Easiest things to set up are still internal phone systems, but can be used all the way up to carrier-grade VoIP call handling.
- Can interact with the PSTN (Public Switched Telephone Network) using WinModems, ISDN or T1-style PC cards.
- Can interact with VoIP providers.
- Can be extended to do virtually anything using scripting.

Why Use Asterisk At Home?

- My use case was creating an internal intercom system so people would quit yelling.
- Could use it for your own home phone system.
- Create a private calling system for families that works over the internet - no long distance charges. Great for elderly people who have problems understanding cellphones.

Some Abbreviations

- SIP - Session Initiation Protocol. It’s the call-control protocol most widely used by VoIP phones. Looks a LOT like HTTP. :)
- IAX - Inter-Asterisk eXchange - another less widely used protocol usually used for connecting multiple Asterisk installations together.
- TDM Card - Time Division Multiplexing. Used to connect to Asterisk to a digital telecom network (ISDN, T1/3, etc)
- FXO/FXS - Foreign eXchange Office/Station. Used to define whether one is a receiver or originator of analog telecom calls. “Who provides dialtone” and it’s backwards from what you think:
    - FXO - The end unit/customer. Telephone, fax machine, PC Card, etc
    - FXS - The head unit/provider. Ma Bell!

Let’s Build an Intercom System!

Getting Phones

- My total investment in this project was $0.
- Obviously, I got mine from the office when we were moving. :)
- But they can be had pretty cheaply (around $20 each) currently on eBay.
- These old Polycoms are unsupported, but they still work fine and there are plenty of resources our there on the Internet for working with them.

Start with Installing Asterisk

- Unfortunately, the version in the apt tree is way out of date, so your best bet is to compile from source.
- After running make menuselect, be sure that you install at least one sound package, usually u-law format.
- Install the MP3 decoder if you want to use MP3 sounds.

Configure Phone Accounts

- Any IP phone that you want to connect to Asterisk will need to have an account.
- These are configured in pjsip.conf
- Next is an example

```
[phoneprov_defaults](!)
type=phoneprov
PROFILE=polycom
SERVER=<your server name or IP address>
TIMEZONE=America/Chicago

[103]
type=endpoint
context=default
disallow=all
allow=ulaw
auth=103-auth
outbound_auth=103-auth
transport=transport-udp
aors=103

[103-auth]
type=auth
auth_type=userpass
password=103
username=103

[103]
type=aor
max_contacts=1

[103](phoneprov_defaults)
endpoint=103
MAC=<mac address goes here, all lowercase>
DISPLAY_NAME=Office
LABEL=103 Office
CALLERID=103
```

Provisioning Phones

- Provisioning is the process of configuring the phone itself to connect to Asterisk.
- On most modern IP phones it can be done manually, either on the phone itself or on a web interface the phone provides.
- … but Asterisk can also provision phones for us!

```
# In /etc/asterisk/http.conf

enabled = yes
bindaddr = 0.0.0.0

# In /etc/asterisk/manager.conf

[general]
enabled = yes
webenabled = yes

# Fetch Polycom configs and sources

$ mkdir -p /var/lib/asterisk/phoneprov/configs
$ cd /var/lib/asterisk/phoneprov/configs
$ wget https://downloads.polycom.com/voice/voip/uc/
Polycom_UC_Software_3_3_5_release_sig_combined.zip --no-check-certificate
$ wget https://downloads.polycom.com/voice/voip/uc/
Polycom_UC_Software_3_3_5_release_sig_split.zip --no-check-certificate
$ unar -D Polycom*
$ rm -rf *.zip
```

Touchless Provisioning

- Most phones have the ability to “touchlessly provision” meaning that you don’t even need to take the phone out of the box to provision it!
- This is meant for large-scale deployments where many phones will be coming and going and you’re not going to have the time to configure each one.
- Uses DHCP options:
  - Option 160: http://<server IP or host name>:8080/phoneprov/
  - Option 101: America/Chicago (or your timezone)
  - Option 100: CST6CDT,M3.2.0,M11.1.0 (or your time code)

Using Queues

- Now we have our phones connecting to Asterisk, we need to configure some queues, because we're going to use those for the Intercom system.
- Queues are just what they sound like: a queue of incoming calls that queue members can answer. Think like in a call center.
- Could also be used in a standard home phone setup: incoming calls go into the queue, the first phone to answer the ring wins.

Configuring Queues

- For our queue, we are going to be using the “ringall” queue type, which rings all lines.
- Add each member to the queue using member => PJSIP/<id>

```
[default]
strategy = ringall
timeout = 10
retry = 5
maxlen = 1
joinempty = yes
leavewhenempty = no
context = default
periodic-announce-frequency = 60
periodic-announce = calling

member => PJSIP/101
member => PJSIP/102
# Add each member here
```

Now let’s make it work!

extensions.conf

- extensions.conf is probably the single most important file in any Asterisk installation. It is also called the dialplan.
- It tells Asterisk what to do when a call comes in.
- Uses pattern matching and ladder logic to determine call flow.
- Remember, this is just software!

Dialplan for Intercom

- For our queue, we are going to be using the “ringall” queue type, which rings all lines.
- Add each member to the queue using member => PJSIP/<id>
- Extens starting with _ are pattern matches.
- You can use “n” as the second argument for “next”, makes editing it easer.

```
[default]

exten => 0,1,Answer
exten => 0,n,Wait(1)
exten => 0,n,Queue(default)

exten => 0000,1,Answer
exten => 0000,n,Wait(1)
exten => 0000,n,Page($
{STRREPLACE(QUEUE_MEMBER_LIST(default),",","&")},b(ring-answer^addheader^1))

exten => _0000[12]XX,1,Answer
exten => _0000[12]XX,n,Wait(1)
exten => _0000[12]XX,n,Page(PJSIP/${EXTEN:4},b(ring-answer^addheader^1))

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

[ring-answer]
exten => addheader,1,Set(PJSIP_HEADER(add,Alert-Info)=Paging)
exten => addheader,n,Return
```

Making Phones Auto-Answer

- Phones need to automatically answer for an intercom system to work. Otherwise, you're just calling the phones. :)
- With Polycom phones, you can send a specific SIP header and configure the phones to auto-answer on speaker phone when they receive a call with that header.

```
# In /var/lib/asterisk/phoneprov/configs/sip.cfg

<alertInfo
    voIpProt.SIP.alertInfo.1.value="Paging"
    voIpProt.SIP.alertInfo.1.class="custom1" />

<se.rt>
    <se.rt.custom1
        se.rt.custom1.name="Paging"
        se.rt.custom1.ringer="ringer15"
        se.rt.custom1.timeout="800"
        se.rt.custom1.type="ring-answer">
    </se.rt.custom1>
    <se.rt.default se.rt.default.timeout="5000">
    </se.rt.default>
</se.rt>
```

Making the Intercom Soft Button

- Most Polycom phones have “soft buttons” on them. These are the “Callers” and “Dir” buttons at the top.
- These are configurable in the config!
- Would be nice to have a one-touch “intercom” button…

```
# In /var/lib/asterisk/phoneprov/configs/sip.cfg

<feature
    feature.directedCallPickup.enabled="1"
    feature.enhancedFeatureKeys.enabled="1" />

<softkey
    softkey.1.label="Intrcom"
    softkey.1.action="$FLine1$0000$Tinvite$"
    softkey.1.enable="1"
    softkey.1.use.idle="1"
    softkey.2.label="Intrcom"
    softkey.2.action="$FLine1$$FDialpad0$$FDialpad0$$FDialpad0$$FDialpad0$
$Tinvite$"
    softkey.2.enable="1"
    softkey.2.use.dialtone="1"
    softkey.feature.newcall="0"
    softkey.feature.endcall="0" />
```

Demonstration

Beyond Intercoms

- Connecting a SIP provider for having an actual home phone is rather easy, it just feeds into the 0 extension, which will ring all phones.
- Make your own IVR using play and read extensions.
- Use the Asterisk Gateway Interface (AGI) to do literally anything you can think of.
- “It’s just software!”

Resources

- https://www.asterisk.org/
- https://computingforgeeks.com/how-to-install-asterisk-pbx-on-ubuntu/
- https://www.robpeck.com/2022/03/creating-a-home-intercom-system-using-asterisk-and-cheap-used-phones/
- https://www.voip-info.org/
- Slides will be on my website.