TUTORIAL USING A CISCO IAD2400 WITH FREEPBX

linkrunner · Jul 20, 2018

L

linkrunner
New Member

Jul 20, 2018 #1

A little backstory, as I know some may wonder why I'd use an ancient T1 router to provide FXS service. I do consulting IT in a small office in the Chicago area, and up until 2013, they were still running a T1 with the Cisco as CPE connected to an old Nortel Norstar. It was eventually decided by the business owner that it is cost prohibitive, and as such, was replaced by a Comcast internet line, and their VoiceEdge hosted voice service. The old T1 provider claimed that they would retrieve their equipment, but I guess not!

In any case, this assumes that the Cisco device is defaulted to an empty startup-config. If one needs to reset the passwords there are a very many tutorials on the internet detailing how to use conf-reg to ignore the startup-config.

From the clean slate:

```
en
conf t
```

This will set console privilege to enabled and enter global config mode.

After lots of hair pulling, I determined that the device will not authenticate correctly if the server and IP addresses are set from the get go, so disable both network interfaces, but set an IP address on the first for convivences' sake. Replace X.X.X.X with the ip address you want to assign, and A.A.A.A with the full subnet mask.

```
interface fa 0/0
shut
ip addr X.X.X.X A.A.A.A
interface fa 0/1
shut
```

Set the default gateway (This should not be needed, but I did it anyway).

```
ip route 0.0.0.0 0.0.0.0 X.X.X.X
```

Yet another precaution: disable the voip services on the device (This will also allow for some of the configuration options later on).

```
no sip-ua
voice service voip
sip
call service stop
```

Now, if one is running DNS to point to the PBX, set the name server:

```
ip name X.X.X.X
```

Rinse and repeat for multiple. IF a mistake is made:

```
no ip name X.X.X.X
```

To the best of my understanding, the voice service on this device treats all calls the same, incoming and outgoing. As such, the only difference between the two lies in the dial plan.
In addition, one quirk of the device is that for authentication, it requires a minimum of four digit usernames and passwords. If one uses three digit extensions, use the SIP alias feature and make sure that the CID is set to the desired extension (If someone knows of a way to do this WITHOUT modifying config files, please let me know, and I'll test it).

```
dial-peer voice XXXX pots
destination-pattern SIPUSER
port 2/0
authentication u SIPUSER p SIPPASS
```

Replace the XXXX in the dial-peer line with a number, from looking at the configs that I have access to, some use the port on the device, some use the user, some arbitrarily name them. It does not matter, the device assesses all the same.
Repeat the above for all the voice extensions/ports that need to be configured, changing the port. On some devices this may be different. Use port ? to see what IOS has available.

Configuring the outbound dial plan is a bit more intensive. Here is mine:

```
dial-peer voice 10 voip
description Extension Dial
destination-pattern ...T
session protocol sipv2
session target sip-server
codec g711ulaw

dial-peer voice 20 voip
description 10 Digit Dial
destination-pattern ..........T
session protocol sipv2
session target sip-server
```

```
codec g711ulaw

dial-peer voice 30 voip
description 11 Digit Dial
destination-pattern ...........
session protocol sipv2
session target sip-server
codec g711ulaw

dial-peer voice 100 voip
description Catch All Dial
destination-pattern .T
session protocol sipv2
session target sip-server
codec g711ulaw
```

There are many things that can be done with destination pattern. If one wishes to learn more about it, Cisco has a great wiki which explains how to use destination pattern. It does not matter which device is mentioned, all of them use the same destination-pattern rules.

Configure the connection to the PBX here:

```
sip-ua
registrar dns  bx0.excel.local expires 3600
sip-server dns  bx0.excel.local
protocol mode ipv4
```

In this case, I am using DNS to connect to my server. Change both strings to `ipv4:X.X.X.X` or `ipv6:X.X.X.X` and change protocol mode `ipv4` to `ipv6 depending on how you want to connect to you PBX.`

Now, enable sip:

```
voice service voip
sip
no call service stop
```

And enable the ethernet interface:

```
inter fa 0/0
no shut
```

Save the configuration:

```
write
```

Restart the device:

```
reload
```

If someone wants to say that IOS changes apply instantly to the running config, then they are WRONG in this case. After almost chucking this box into the trash and nearly tearing all my hair out, I realized that sip credentials will not take effect immediately and that a restart is needed to actually get registrations. In this case it may not be since the voice service was shutdown, but it is good practice. If anyone has additional insight into the behavior of these devices with IOS, please comment.

On the device after the restart you can use the `show sip register status` command to view what is happening with the device.
In my case:

```
Line peer expires(sec) registered
================================ ========== ============ ==========
2120 2120 2615 yes
2130 2130 2444 yes
2140 2140 2444 yes
2150 2150 2444 yes
2160 2160 2444 yes
2170 2170 2444 yes
2180 2180 2444 yes
2190 2190 2444 yes
```

Questions? I'll try to respond quickly, but I also have a lot on my plate right now.
And a pic of the device installed:
https://imgur.com/a/qP1DBqL