Loud_Listener - A High Speed SIP/rtp session recorder

WHY

To satisfy various regulatory lawful Intercept requirements, most voice carriers require a capability to record an individual telephone call.

There are many commercial tools available but the manufacturers charge a small fortune for this software. But what do you get:

1. A gui
2. a voice recorder;
3. a bit of encryption to protect the target and
4. a ETSI XML interpreter

Point 1, 4 and 3 is a luxury (ETSI XML requests are not a legal requirement for HI1 in most places – although you'd never believe it by reading manufacturers documentation) – and can be knocked up in a rainy Sunday afternoon when you cant fly your model helicopter.

So all we really need is a voice packet recorder, right? Now I keep on reading articles, everywhere – in serious Linux journals and security periodicals, that says how insecure VOIP is. Particularly, how easy it is to sniff a voice call.

And that's true – to an extent. If you create a SPAN port on your 100mb office lan switch – boot knoppix and type the command

```
$ tcpdump  -n –nn –w voice.out UDP
```

You are bound to capture all the voce conversations taking place – plus Gigabytes of other old tosh. Load the file into ethereal or cain&able - wait while it sorts out all the junk you've recorded and the you can listen to all of the calls. One by one.

We can't really be more specific because:

- There is are two or more VOIP protocol sets (SIP/RTP, or H323 or skinny)
- Even if we focus on SIP/RTP (the most modern), we discover that there is no well known port for the content streams. The ports are specified in a separate protocol (SDP) imbedded in SIP

Recording all voice calls on a network with out specific permission and then listening to them would not be "lawfully". So I investigated shareware sniffers.

There aren't many but the few were very good. However, I could not find one that could be armed to record one particular phone call – Most recorded all voice passing the Interface which, as I said before, really could not be used "lawfully".

So we decided to write one.

Speed is also an issue, the code will manage 1Gbs of traffic on a standard Intel/Linux platform – To work at 10 or 40Gbs, the Linux code will need to be on an optimised platform (providing your target isn't generating more traffic than a PCI-X bus can handle)

HOW

This "Beta" or “Proof of concept” program does the following:

1 - Search all packets on UDP 5060 on the assumption that it is a SIP packet. Obviously, SIP can appear on a range of ports, this must be enhanced in the future.

2 - If it is a SIP packet

- Check to see if the packet is an INVITE packet,
- Check to see if the packet includes an SDP Header
- Check if the packet contains the search string which should ideally be a telephone number or a sip url. (obviously for testing purpose the search string “SIP” will pass the condition on every packets and so record the first valid SIP call)

4 - Extract the Call-ID from the SIP packet
5 - Extract RTP port and RTP ip address from the SDP header
5 - Generate RTCP port
6 - Log all packets with the RTP port or RTCP port that comes to or from the RTP address
8 - Bail if

- the RTP SSRC field is not correct or
- the time-out is reached or
- A SIP “bye” or “cancel” with matching Call-ID is reached.