Cyber defense 2

ENSIBS Vannes

2015-2016

Report

SIP : Communications interception and encryption

Network basis

By : Valentin ALLAIRE, Dylan COIC, Guillaume COUCHARD, Kévin FAUVE

Teacher : Maël Auzias

1 Project objectives

The objectives of this project are the following :

- Set up 2 asterisks servers
- Set up 2 soft-phones on our computers or android phones.
- Intercept the SIP communication between a soft-phone and the asterisk server.
- Intercept the RTP data communication between the 2 soft-phones.
- Encrypt the SIP communication using SSL/TLS.
- Encrypt the RTP data communication using ZRTP.
- Intercept the communication while it's encrypted and prove that the encryption is efficient and reliable.
- Doing a trunk SIP between the two asterisks servers.
- Try the same things when the two clients are connected on two different asterisks servers.

The difference with the overview is that there is a new objective which is doing a voice recognition on top of the interception of the RTP data communication.

2 Why did we not meet all of our objectives ?

Mainly because of the lack of time we had and also because when we did the overview we didn't realize it will be a problem to do the encryption. We didn't even try to bypass the SSL/TLS encryption because it takes a lot of time but we think it is possible if the the version of this is SSL v3 for example so we can use the poodle vulnerability. It took us also much time to do the voice recognition because we didn't know the tools and we are not telecoms experts, although it was really cool to learn about it. We didn't meet the VPN objective also because of the lack of time but we think it can be easily done if we use openVPN for example.

3 What we wanted to learn and what we actually learned

We wanted to learn more about the VoIP, especially the SIP protocol and we did learn more about that because now we know for example that the SIP is a signaling protocol and RTP is for the data communication between the two soft-phones. After that we wanted to learn more about the encryption protocols and how it works. We used SSL/TLS for the SIP encryption so we learned a lot about this type of encryption and the certificates. We also learned how to make our own self-signed certificates thanks to this project. Now we know that to encrypt RTP, we use the SRTP protocol which uses AES 128 bits and SDES to exchange the keys and ZRTP is AES 128 bits too but is using Diffie-Hellman algorithm. We also learned what we wanted to learn which was how to configure an asterisk server, intercept a SIP communication and RTP communication and encrypt these communications. We learned how to do two asterisks server and make them communicate (which is a trunk SIP). Finally we didn't plan that but we learned how to do a voice recognition using signals analysis. What we didn't learn because we didn't have the time for it is how to settle a VPN tunnel between the 2 asterisks servers.

3.1 Challenges encountered

During our project we encountered some difficulties, the first one was how to intercept a SIP communication.Indeed, at the beginning we didn't know much about this protocol so we though SIP and RTP were the same thing .... So after we studied the protocol we were able to make the difference and then set up the right filters in Wireshark. Although that wasn't enough, we connected the two soft-phones, asterisk server and the attacker on the same network and the attacker couldn't intercept all the traffic which was a real problem. Then we understood that we needed to set up the network card into monitor mode and do some arp poisoning (we will explain that in details later on) and that solved the problem.

The next challenge was to encrypt the SIP communication because at the beginning we didn't really understand how SSL/TLS works. Also, after we understood that, we had to install the right soft-phones because some of them don't handle the SSL/TLS encryption. Once everything has been done, we had to configure the asterisk server and generate all the certificates, which was hard to do but with a lot of tutorials we managed to succeed. Finally, the big challenge was to do the voice recognition.

4 How did we share the code

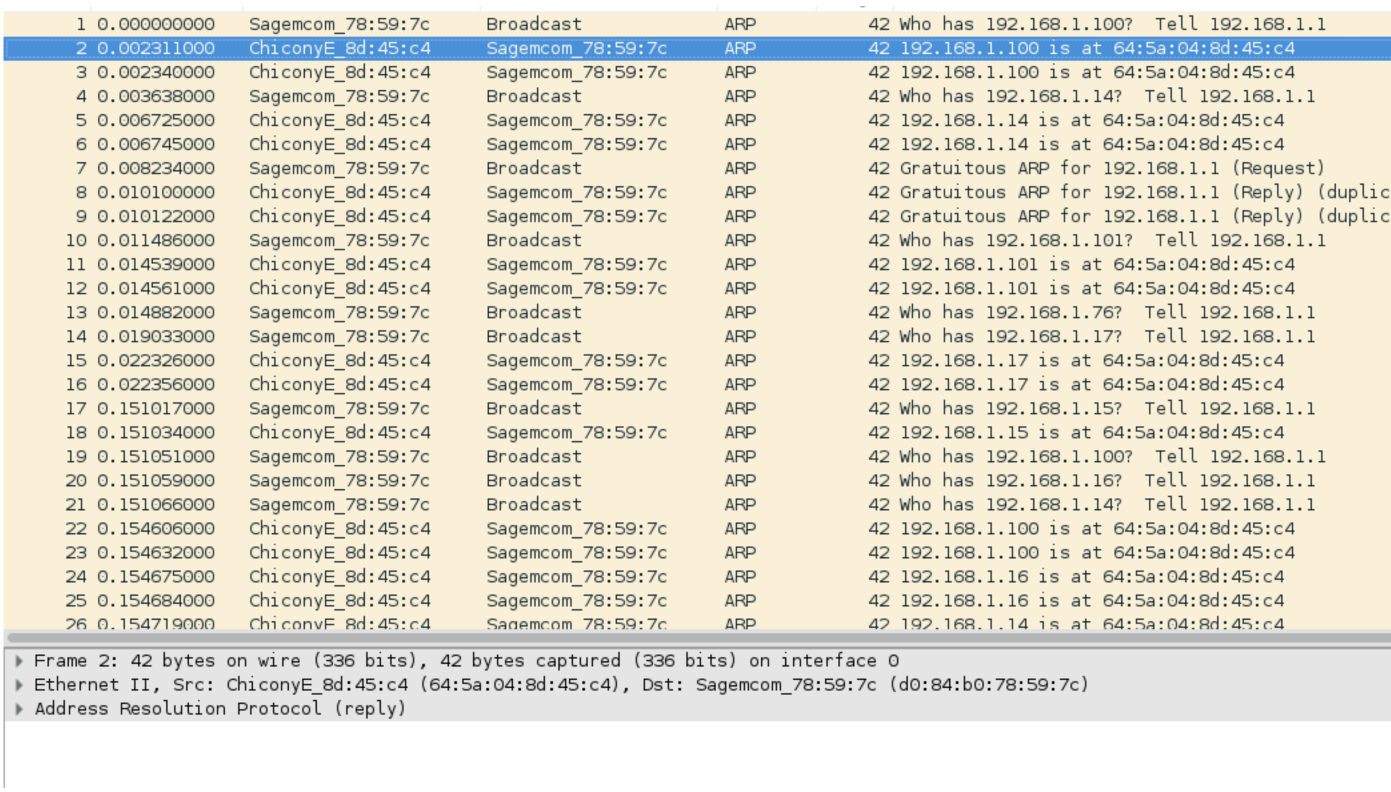
We shared the code using GitHub and e-mails sometimes, for the proof here is the GitHub link : `https://github.com/KevinF49/ProjetReseau`

5 Communications Analysis : Wireshark to the rescue !

In this section we will mainly talk about the different communications that we did and explain them to you with the help of some Wireshark captures.

5.1 ARP Communications

As the title says, we will show you a capture of some ARP communications in here. We are doing this because this is our way to explain how the attack works and how we tested the security flaw.



1	0.000000000	Sagemcom_78:59:7c	Broadcast	ARP	42	Who has 192.168.1.100? Tell 192.168.1.1
2	0.002311000	ChiconyE_8d:45:c4	Sagemcom_78:59:7c	ARP	42	192.168.1.100 is at 64:5a:04:8d:45:c4
3	0.002340000	ChiconyE_8d:45:c4	Sagemcom_78:59:7c	ARP	42	192.168.1.100 is at 64:5a:04:8d:45:c4
4	0.003638000	Sagemcom_78:59:7c	Broadcast	ARP	42	Who has 192.168.1.14? Tell 192.168.1.1
5	0.006725000	ChiconyE_8d:45:c4	Sagemcom_78:59:7c	ARP	42	192.168.1.14 is at 64:5a:04:8d:45:c4
6	0.006745000	ChiconyE_8d:45:c4	Sagemcom_78:59:7c	ARP	42	192.168.1.14 is at 64:5a:04:8d:45:c4
7	0.008234000	Sagemcom_78:59:7c	Broadcast	ARP	42	Gratuitous ARP for 192.168.1.1 (Request)
8	0.010100000	ChiconyE_8d:45:c4	Sagemcom_78:59:7c	ARP	42	Gratuitous ARP for 192.168.1.1 (Reply) (duplic
9	0.010122000	ChiconyE_8d:45:c4	Sagemcom_78:59:7c	ARP	42	Gratuitous ARP for 192.168.1.1 (Reply) (duplic
10	0.011486000	Sagemcom_78:59:7c	Broadcast	ARP	42	Who has 192.168.1.101? Tell 192.168.1.1
11	0.014539000	ChiconyE_8d:45:c4	Sagemcom_78:59:7c	ARP	42	192.168.1.101 is at 64:5a:04:8d:45:c4
12	0.014561000	ChiconyE_8d:45:c4	Sagemcom_78:59:7c	ARP	42	192.168.1.101 is at 64:5a:04:8d:45:c4
13	0.014882000	Sagemcom_78:59:7c	Broadcast	ARP	42	Who has 192.168.1.76? Tell 192.168.1.1
14	0.019033000	Sagemcom_78:59:7c	Broadcast	ARP	42	Who has 192.168.1.17? Tell 192.168.1.1
15	0.022326000	ChiconyE_8d:45:c4	Sagemcom_78:59:7c	ARP	42	192.168.1.17 is at 64:5a:04:8d:45:c4
16	0.022356000	ChiconyE_8d:45:c4	Sagemcom_78:59:7c	ARP	42	192.168.1.17 is at 64:5a:04:8d:45:c4
17	0.151017000	Sagemcom_78:59:7c	Broadcast	ARP	42	Who has 192.168.1.15? Tell 192.168.1.1
18	0.151034000	ChiconyE_8d:45:c4	Sagemcom_78:59:7c	ARP	42	192.168.1.15 is at 64:5a:04:8d:45:c4
19	0.151051000	Sagemcom_78:59:7c	Broadcast	ARP	42	Who has 192.168.1.100? Tell 192.168.1.1
20	0.151059000	Sagemcom_78:59:7c	Broadcast	ARP	42	Who has 192.168.1.16? Tell 192.168.1.1
21	0.151066000	Sagemcom_78:59:7c	Broadcast	ARP	42	Who has 192.168.1.14? Tell 192.168.1.1
22	0.154606000	ChiconyE_8d:45:c4	Sagemcom_78:59:7c	ARP	42	192.168.1.100 is at 64:5a:04:8d:45:c4
23	0.154632000	ChiconyE_8d:45:c4	Sagemcom_78:59:7c	ARP	42	192.168.1.100 is at 64:5a:04:8d:45:c4
24	0.154675000	ChiconyE_8d:45:c4	Sagemcom_78:59:7c	ARP	42	192.168.1.16 is at 64:5a:04:8d:45:c4
25	0.154684000	ChiconyE_8d:45:c4	Sagemcom_78:59:7c	ARP	42	192.168.1.16 is at 64:5a:04:8d:45:c4
26	0.154719000	ChiconyE_8d:45:c4	Sagemcom_78:59:7c	ARP	42	192.168.1.14 is at 64:5a:04:8d:45:c4

Frame 2: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
Ethernet II, Src: ChiconyE_8d:45:c4 (64:5a:04:8d:45:c4), Dst: Sagemcom_78:59:7c (d0:84:b0:78:59:7c)
Address Resolution Protocol (reply)

Figure 1: ARP Flow

First of all, a scan host is established with Ettercap.Then the malicious person sends in broadcast some arp packets and matches his mac address with the ip address of the gateway. The router sends arp requests to fill his arp table. Thus attacker's mac address answers to all the requests which means his mac address is matched with every IP address on the network. Thereby, he fills his arp table and all request above the layer 2 are intercepted by him. The router notices there are two gateways and inform all the devices on the network by sending in broadcast some gratuitous arp packets. *ChiconyE_ 8d:45:c4 (64:5a:04:8d:45:c4)* is the attackers' mac address and it is spoofing every IP address on the network. There is a solution to counter/avoid that "arp poisoning" attack which is setting the arp table in static mode.

5.2 SIP Communications

In this section we will talk about the results of our project but not how we did it. In order to do that we will show you some Wireshark captures and explain them.

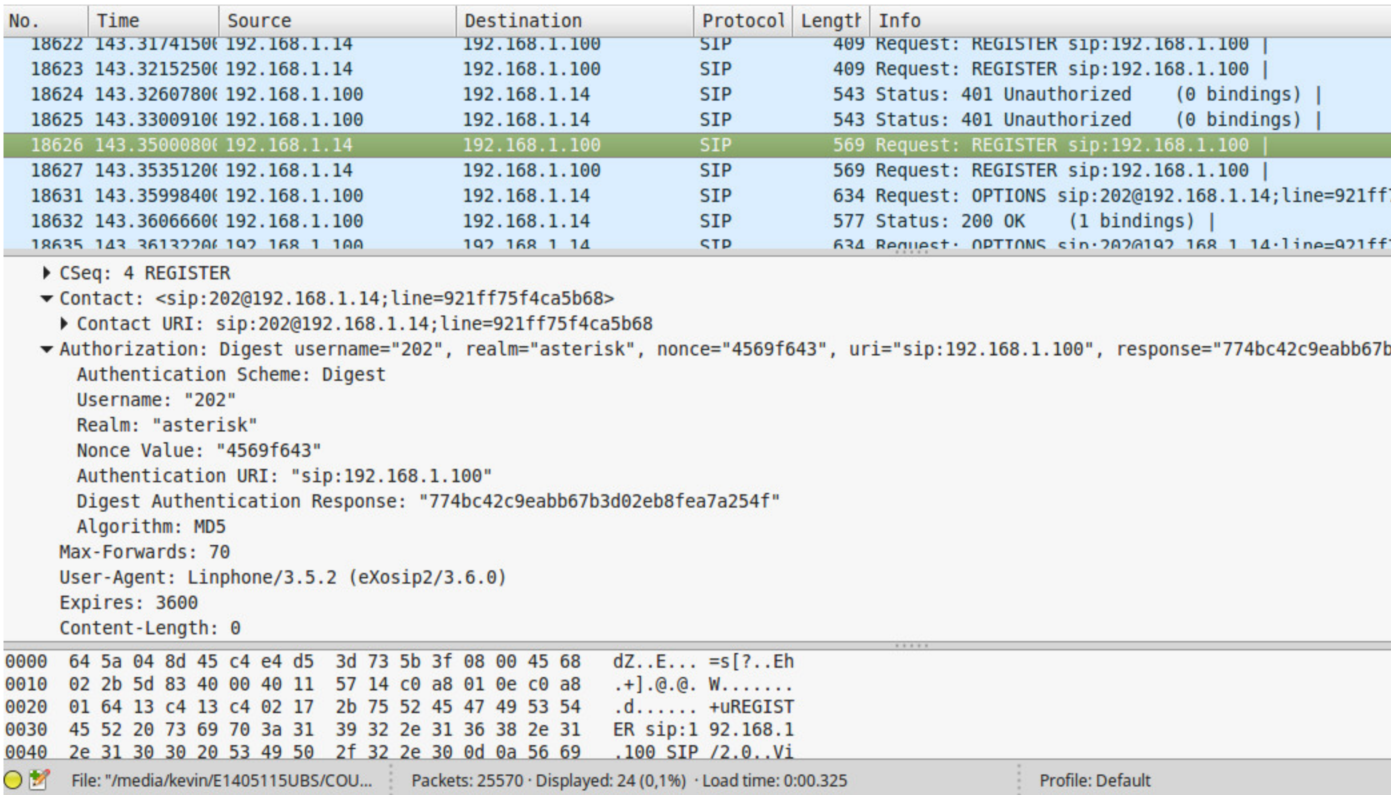


Figure 2: SIP REGISTER

On this screenshot we can see a weakness of the SIP protocol. User password is hashed with MD5 algorithm therefore it's easy to decrypt it.
With this picture only, we have all the information that we need to connect to the asterisk server:

- Username
- URI
- MD5 hash

5.3 TLS Encryption

Now that we saw that SIP is a vulnerable protocol, we will show you that it can be secured if we use SSL/TLS Encryption. For that we used the Blink soft-phone and we created certificates and private/public keys with the openSSL command.

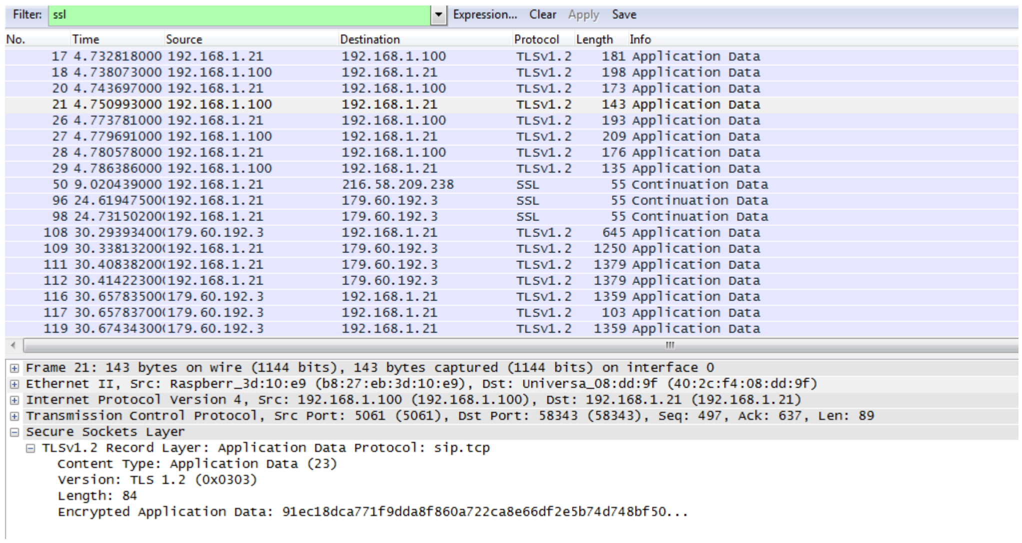


Figure 3: TLS REGISTER SUCCEED

In this capture you can see that all the packets are encrypted with the TLS Encryption. Just for the record, this is a capture when the user authenticates, which means you could see the right password with SIP but now you just can't see anything because it is encrypted.

Although, this is only the signalization in this conversation, which means we can still intercept the data packets between the two users, carried by the RTP protocol.

5.4 RTP Communications

Like we said in the last subsection, the RTP protocol is used to transmit data packets in a VoIP communication. Concretely, it means that voice is carried by this protocol. Which is interesting because we wanted to see if we can intercept these packets and then extract the voice to replay the conversation, and we did.

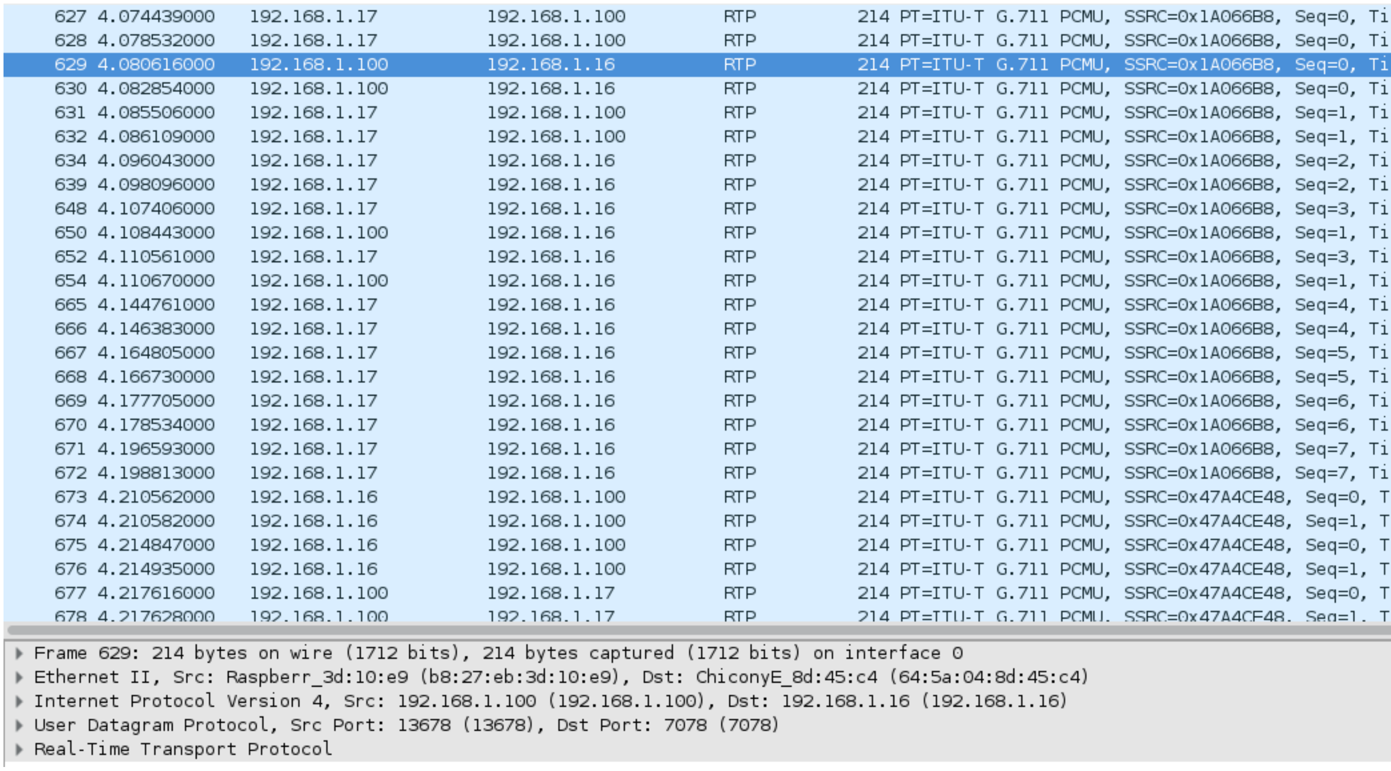


Figure 4: RTP

This capture shows the conversation between two soft-phones with the RTP protocol. We can see that the host 192.168.1.17 wants to speak with 192.168.1.16. But when the host believes that he is sending the packets at the asterisk server, he actually sends the packets to the pirate. That's the same when the real host answers, the packets are for the attacker. To sum up, communications are intercepted by the malicious person. This is a hyperlink to a conversation that we intercepted : click to play the sound!

5.5 SRTP Communications

We saw before that RTP is a weak protocol because it is not encrypted, so now we will encrypt it. For that we used SRTP or ZRTP protocols, both are extensions to RTP that incorporates enhanced security features. SRTP for example encrypts RTP using AES 128 bits.

14233	274.016610000	192.168.1.100	192.168.1.71	SRTP	224 PT=ITU-T G.711 PCMU, SSRC=0x5F58D097, Seq=27
14234	274.016749000	192.168.1.100	192.168.1.71	SRTP	224 PT=ITU-T G.711 PCMU, SSRC=0x5F58D097, Seq=27
14235	274.018675000	192.168.1.100	192.168.1.71	SRTP	224 PT=ITU-T G.711 PCMU, SSRC=0x5F58D097, Seq=27
14236	274.018748000	192.168.1.100	192.168.1.71	SRTP	224 PT=ITU-T G.711 PCMU, SSRC=0x5F58D097, Seq=27
14237	274.027551000	192.168.1.100	192.168.1.71	SRTP	224 PT=ITU-T G.711 PCMU, SSRC=0x5F58D097, Seq=27
14238	274.030624000	192.168.1.100	192.168.1.71	SRTP	224 PT=ITU-T G.711 PCMU, SSRC=0x5F58D097, Seq=27
14239	274.050116000	192.168.1.100	192.168.1.71	SRTP	224 PT=ITU-T G.711 PCMU, SSRC=0x5F58D097, Seq=27

Figure 5: SRTP Flow

In this capture the principle is the same that the one above but in this one the conversation is encrypted with SRTP protocol. We won't detail how it works here because this isn't the purpose of this report but basically, it counters our attack because even if we intercept all the SRTP packets, and then we replay the conversation, we will only hear noise because of the encryption, so it's impossible here to hear the voices of the people in a conversation as you can see in this audio file

Conclusion

As you saw when you read this report, it is not about how we configured the servers and soft-phones to get them communicate using VoIP but more about how they do it. For that we had to analyze some Wireshark captures and that is what we explained in here. To end this report, we would say that SIP along with RTP are really vulnerable therefore if someone wants to set up a ToIP infrastructure they have to secure these protocols using at least SSL/TLS and ZRTP/STRP. Indeed, not only they are vulnerable but the attack is quite easy to do considering it's a simple man in the middle. We didn't do the best secure network because on top of what we did a VPN can be settled between the two asterisk servers and after that we think it is pretty secure.