

Rapport de Travaux Pratiques

Sécurisation des systèmes VoIP

Djamil [Ton Nom]

25 Février 2026

1 Accès et Déploiement du Serveur

1.1 Accès SSH via Clé Publique

Conformément aux consignes, une paire de clés SSH a été générée. La clé publique a été déposée sur la plateforme de partage pour permettre l'accès au serveur.

- **Commande :** ssh-keygen -t ed25519
- **Connexion :** ssh -i .ssh/id_ed25519 admin@srv-khennouchi.avalone-formation.com

1.2 Installation de FreePBX 17

Le déploiement a été effectué sur une base Debian en suivant la procédure officielle de Sangoma :

```
cd /tmp
wget https://github.com/FreePBX/sng_freepbx_debian_install/raw/
master/sng_freepbx_debian_install.sh -O /tmp/
sng_freepbx_debian_install.sh
bash /tmp/sng_freepbx_debian_install.sh
```

Lors de l'initialisation, les services de protection ont été volontairement désactivés pour la phase de test :

```
fail2ban-client stop
fwconsole firewall stop
```

Preuve de l'installation réussie de freePBX :

Please provide the core settings that will be used to administer and update your system

Administrator User

Username: Djamil

Password:
Really Weak

Confirm Password:

System Notifications Email: khennouchidjamil5@gmail.com

Notifications Email address: khennouchidjamil5@gmail.com

System Identification: VoIP Server

Automatic Module Updates: Enabled

Automatic Module Security Updates: Enabled

Send Security Emails For Unsigned Modules: Enabled

Check for Updates every: Day Between 8am and 12pm

Activate Windows
Go to Settings to activate Windows

Setup System

FIGURE 1 – Capture d'écran prouvant l'installation de freePBX

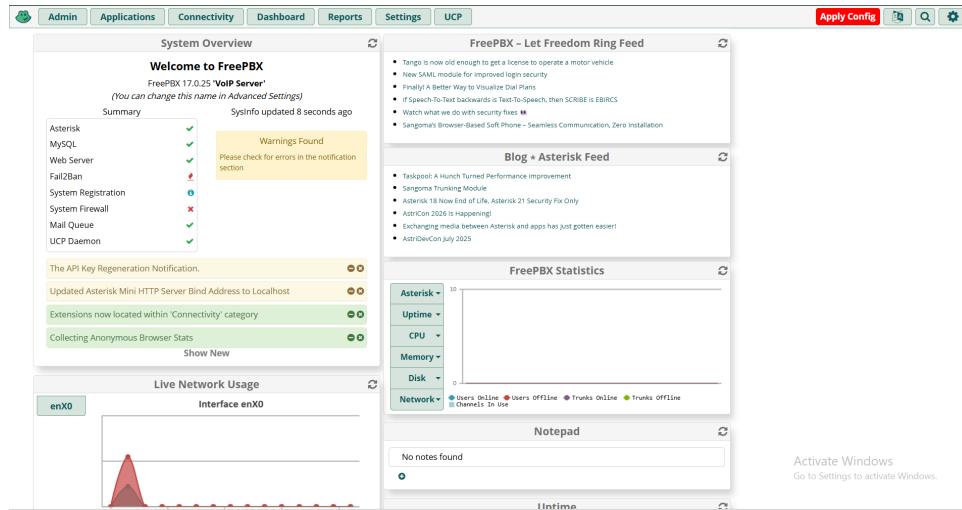


FIGURE 2 – Capture d'écran prouvant l'installation de freePBX

2 Analyse des vulnérabilités et interception

2.1 Paramétrage de base et communication SIP

Deux terminaux (Linphone sur PC et smartphone) ont été configurés pour communiquer via le serveur. Les extensions 101 et 102 ont été créées pour ces tests.

The screenshot shows the configuration interface for a User Extension. The extension number is set to 101, with a display name of "Terminal_A" and an outbound CID of "101<0101>". The secret is set to TA101 with a note "Really Weak". Under the "Language" section, the language code is set to "Default". In the "User Manager Settings" section, the user directory is set to "PBX Internal Directory", and the default user is set to "Create New User". The "Groups" section shows "All Users" selected. At the bottom right, there are buttons for "Activate", "Submit", and "Reset". A note at the bottom right says "Go to Settings to activate Windows".

FIGURE 3 – Configuration du compte SIP 101

The screenshot shows the "My account" screen in Linphone. The left sidebar has "General" selected under "Account settings". The main area is titled "General" and shows the "Details" section with the SIP address set to "sip:102@35.180.71.123", display name "102", and international code "*". It also shows the account status as "Connected" and a "Disconnect my account" option. A note at the bottom right says "Activate Windows".

FIGURE 4 – Preuve de Configuration du compte SIP 102

2.2 Interception de trafic (Flux en clair)

À l'aide de Wireshark et de Asterisk, nous avons capturé le trafic entre les deux terminaux.

Vous pouvez retrouver l'intégralité des captures réseau au format .pcapng à l'adresse suivante : Lien vers les captures Wireshark.

```

-- Executing [1@zgfrom=internal:1] GotoIf("PJSIP/101-00000000", "1!=ext-local;102,1;FollowMe_Check,102,1") in new stack
-- Goto (ext-local,102,1)
Executing [102@ext-local:1] Set("PJSIP/101-00000000", "--_RINGTIMER=15") in new stack
Executing [102@ext-local:2] ExecIf("PJSIP/101-00000000", "@?Set(_C_WIGNORE=1") in new stack
Executing [102@ext-local:3] GotoIf("PJSIP/101-00000000", "@?macro-exten-vm=1;knows.102.0.0.0") in new stack
Executing [s@macro-exten-vm:1] GotoIf("PJSIP/101-00000000", "macro-user-callerid.s.1;newm.102.,,macro-exten-vm") in new stack
-- Executing [s@macro-user-callerid:1] Set("PJSIP/101-00000000", "TOUCH_MONITOR=1770398222.8") in new stack
Executing [s@macro-user-callerid:2] Set("PJSIP/101-00000000", "CHANCONTEXT=1") in new stack
Executing [s@macro-user-callerid:3] Set("PJSIP/101-00000000", "CHANCONTEXTE=1") in new stack
Executing [s@macro-user-callerid:4] Set("PJSIP/101-00000000", "CHANEXTENCONTEXT=101-00000000") in new stack
Executing [s@macro-user-callerid:5] Set("PJSIP/101-00000000", "CHANEXTEN=101-00000000") in new stack
Executing [s@macro-user-callerid:6] Set("PJSIP/101-00000000", "CALLERID(number)=101") in new stack
Executing [s@macro-user-callerid:7] Set("PJSIP/101-00000000", "AMPUSER=101") in new stack
Executing [s@macro-user-callerid:8] Set("PJSIP/101-00000000", "HOTDESCCKHAN=101-00000000") in new stack
Executing [s@macro-user-callerid:9] Set("PJSIP/101-00000000", "HOTDESCXEXTEN=101") in new stack
Executing [s@macro-user-callerid:10] Set("PJSIP/101-00000000", "HOTDESCXEXTEN=101") in new stack
Executing [s@macro-user-callerid:11] ExecIf("PJSIP/101-00000000", "@?Set(HOTDESCXCALL=101") in new stack
Executing [s@macro-user-callerid:12] ExecIf("PJSIP/101-00000000", "@?Set(CALLERID(name)=101") in new stack
Executing [s@macro-user-callerid:13] ExecIf("PJSIP/101-00000000", "@?Set(REALCALLERIDNUM=101") in new stack
Executing [s@macro-user-callerid:14] ExecIf("PJSIP/101-00000000", "@?Set(REALCALLERIDNUM=101") in new stack
Executing [s@macro-user-callerid:15] Set("PJSIP/101-00000000", "AMPUSER=101") in new stack
Executing [s@macro-user-callerid:16] GotoIf("PJSIP/101-00000000", "@?liminit") in new stack
Executing [s@macro-user-callerid:17] Set("PJSIP/101-00000000", "AMPUSERCIDNAME=Terminal_A") in new stack
Executing [s@macro-user-callerid:18] ExecIf("PJSIP/101-00000000", "@?Set(CIDMASQUERADE=TRUE)") in new stack
Executing [s@macro-user-callerid:19] GotoIf("PJSIP/101-00000000", "@?report") in new stack
Executing [s@macro-user-callerid:20] Set("PJSIP/101-00000000", "AMPUSERCID=101") in new stack
Executing [s@macro-user-callerid:21] Set("PJSIP/101-00000000", "DIAL_OPTIONS=Htr") in new stack
Executing [s@macro-user-callerid:22] Set("PJSIP/101-00000000", "CALLERID(all)=Terminal_A<101>") in new stack
Executing [s@macro-user-callerid:23] ExecIf("PJSIP/101-00000000", "@?Set(CSDIAL=102") in new stack
Executing [s@macro-user-callerid:24] ExecIf("PJSIP/101-00000000", "@?Set(CALLERID(all)=102<0102>)") in new stack
Executing [s@macro-user-callerid:25] ExecIf("PJSIP/101-00000000", "@?Set(CHANNEL(language)=en") in new stack
Executing [s@macro-user-callerid:26] ExecIf("PJSIP/101-00000000", "@?Set(GROUP(concurrent_limit)=101") in new stack
Executing [s@macro-user-callerid:27] ExecIf("PJSIP/101-00000000", "@?Set(CHANNEL(language)=en") in new stack
Executing [s@macro-user-callerid:28] NoOp("PJSIP/101-00000000", "Macro deprecated!! To keep the same line numbers") in new stack
Executing [s@macro-user-callerid:29] NoOp("PJSIP/101-00000000", "Macro deprecated !! To keep the same line numbers") in new stack
Executing [s@macro-user-callerid:30] GotoIf("PJSIP/101-00000000", "@?continue") in new stack
Executing [s@macro-user-callerid:31] ExecIf("PJSIP/101-00000000", "@?Set(_CALLEE_ACCOUNTCODE=)") in new stack
Executing [s@macro-user-callerid:32] Set("PJSIP/101-00000000", "_TTL=64") in new stack
Executing [s@macro-user-callerid:33] GotoIf("PJSIP/101-00000000", "@?continue") in new stack
Goto (macro-user-callerid.s,49)
Executing [s@macro-user-callerid:49] Set("PJSIP/101-00000000", "CALLERID(number)=101") in new stack
Executing [s@macro-user-callerid:50] Set("PJSIP/101-00000000", "CALLERID(name)=Terminal_A") in new stack
Executing [s@macro-user-callerid:51] GotoIf("PJSIP/101-00000000", "@?cnid") in new stack

```

FIGURE 5 – Ecoute de l'appel entre les deux compte SIP sur Asterisk

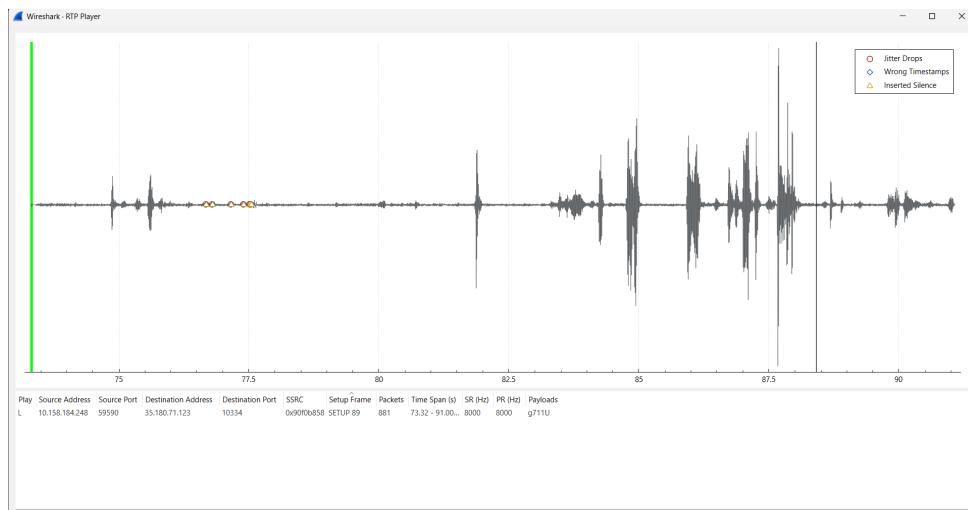


FIGURE 6 – Ecoute de l'appel entre les deux compte SIP sur Wireshark

- **Observation :** Les messages SIP circulent en UDP sur le port 5060.
- **Résultat :** Le flux RTP n'étant pas chiffré, l'outil "RTP Player" de Wireshark permet de reconstituer et d'écouter l'intégralité de la conversation.

3 Sécurisation des communications

3.1 Activation du SRTP

Le protocole SRTP a été activé pour chiffrer les flux médias.

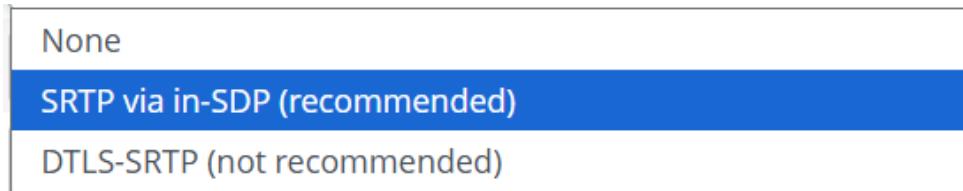


FIGURE 7 – Activation du SRTP sur FreePBX

- **Démonstration :** L'écoute n'est plus possible (bruit blanc ou données illisibles).
- **Analyse de la clé :** Bien que les médias soient chiffrés, la clé de chiffrement est visible dans le message SIP INVITE sous l'attribut `a=crypto`. La sécurité est donc partielle.

```
> Session Attribute (a): rtcp-xr:rcvr-rtt=all:10000 stat-summary=loss,dup,jitt,TTL voip-metrics
> Session Attribute (a): record:off
> Media Description, name and address (m): audio 57076 RTP/SAVP 96 97 98 0 8 18 101 99 100
> Media Attribute (a): rtpmap:96 opus/48000/2
> Media Attribute (a): ftmp:96 useinbandfec=1
> Media Attribute (a): rtpmap:97 speex/16000
> Media Attribute (a): ftmp:97 vbr=on
> Media Attribute (a): rtpmap:98 speex/8000
> Media Attribute (a): ftmp:98 vbr=on
> Media Attribute (a): ftmp:18 annexb=yes
> Media Attribute (a): rtpmap:101 telephone-event/48000
> Media Attribute (a): rtpmap:99 telephone-event/16000
> Media Attribute (a): rtpmap:100 telephone-event/8000
> Media Attribute (a): crypto:1 AEAD_AES_128_GCM inline:pSD6YYDgKxTxXdUD+41Pc2jcqMtTUWHnA8sAHw=
> Media Attribute (a): crypto:2 AES_CM_128_HMAC_SHA1_80 inline:Mosw8LD1kZY7zYkpro4pODWL9b1j/uWD
> Media Attribute (a): crypto:3 AEAD_AES_256_GCM inline:ITUcvkZj1CYDfV5+TQww1DGXbRWYdQr4d97ncq
> Media Attribute (a): crypto:4 AES_256_CM_HMAC_SHA1_80 inline:4ZoRno/Ek+IzzanUb2E5D40Iaj/A5bJz
> Media Attribute (a): rtcp-fb:* trr-int 1000
> Media Attribute (a): rtcp-fb:* ccm tmmbr
[Generated Call-ID: ac6qCr5ri0]
```

FIGURE 8 – Visualisation de la clé de chiffrement SRTP dans le paquet SIP

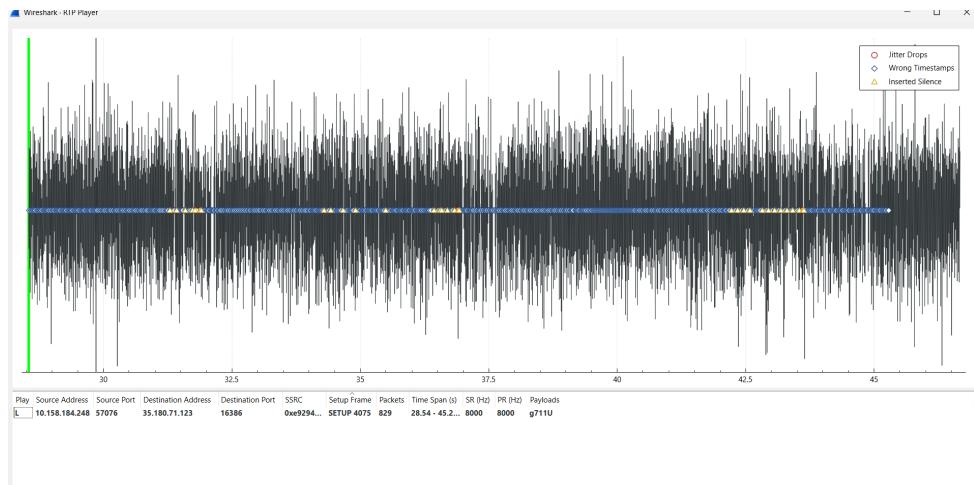


FIGURE 9 – Ecoute de l'appel en SRTP

3.2 Activation de SIP sur TLS

Pour rendre la sécurité robuste, nous avons encapsulé la signalisation dans un tunnel TLS.

- Configuration : Utilisation du port TCP/5061 avec un certificat Let's Encrypt.

```

apt install certbot python3-certbot-apache
certbot --apache -d srv-khennouchi.avalone-formation.
com
cp /etc/letsencrypt/archive/srv-masset.avalone-
formation.com/privkey1.pem /etc/asterisk/keys/
default.key
cp /etc/letsencrypt/archive/srv-khennouchi.avalone-
formation.com/fullchain1.pem /etc/asterisk/keys/
default.crt
chown asterisk:asterisk /etc/asterisk/keys/default.*
#permet d'installer un certificat valide sur FreePBX

```

- Dans le fichier `/etc/asterisk/pjsip.transports.conf` on as rajouter ceci :

```

[0.0.0.0-tls]
type=transport
protocol=tls
bind=0.0.0.0:5061
external_media_address=82.65.56.108
external_signaling_address=82.65.56.108
ca_list_file=/etc/ssl/certs/ca-certificates.crt
cert_file=/etc/asterisk/keys/default.crt
priv_key_file=/etc/asterisk/keys/default.key
method=tlsv1_2
verify_client=yes
verify_server=yes
allow_reload=no
tos=cs3
cos=3

```

```
local_net=172.31.0.0/20
```

- **Résultat :** Wireshark ne peut plus déchiffrer les messages SIP. La clé SRTP n'est plus exposée. La communication est désormais sécurisée de bout en bout contre l'interception et l'homme du milieu (MITM).

4 Travaux pratiques

4.1 Fail2ban et Pare-feu

Après les phases de test, Fail2ban et le pare-feu ont été réactivés pour protéger le serveur contre les attaques par force brute (Brute Force).

```
root@srv-khennouchi:~# systemctl status fail2ban
● fail2ban.service - Fail2Ban Service
    Loaded: loaded (/lib/systemd/system/fail2ban.service; disabled; )
      Active: active (running) since Wed 2026-02-04 16:00:57 UTC; 15s
        Docs: man:fail2ban(1)
     Main PID: 1693 (fail2ban-server)
       Tasks: 19 (limit: 2345)
      Memory: 17.6M
         CPU: 293ms
        CGroup: /system.slice/fail2ban.service
                  └─1693 /usr/bin/python3 /usr/bin/fail2ban-server -xf sta

Feb 04 16:00:57 srv-khennouchi systemd[1]: Started fail2ban.service →
Feb 04 16:00:58 srv-khennouchi fail2ban-server[1693]: 2026-02-04 16:0
Feb 04 16:00:58 srv-khennouchi fail2ban-server[1693]: Server ready
... skipping...
● fail2ban.service - Fail2Ban Service
    Loaded: loaded (/lib/systemd/system/fail2ban.service; disabled; )
      Active: active (running) since Wed 2026-02-04 16:00:57 UTC; 15s
        Docs: man:fail2ban(1)
     Main PID: 1693 (fail2ban-server)
       Tasks: 19 (limit: 2345)
      Memory: 17.6M
         CPU: 293ms
        CGroup: /system.slice/fail2ban.service
                  └─1693 /usr/bin/python3 /usr/bin/fail2ban-server -xf sta

Feb 04 16:00:57 srv-khennouchi systemd[1]: Started fail2ban.service →
Feb 04 16:00:58 srv-khennouchi fail2ban-server[1693]: 2026-02-04 16:0
```

FIGURE 10 – Réactivation du service fail2ban

4.2 Teste d'attaque

Afin de s'assurer de l'activation du fail2ban sur FreePBX nous avons tenter de se connecter a un compte SIP en utilisant un faux mot de passe et après plusieurs tentatives voici le résultat afficher sur Asterisk grace a la commande :

```
fail2ban-client status asterisk-iptables
```

```
root@srv-khennouchi:~# fail2ban-client status asterisk-iptables
Status for the jail: asterisk-iptables
|- Filter
| |- Currently failed: 0
| |- Total failed:    0
| `-' File list:        /var/log/asterisk/fail2ban
`- Actions
  |- Currently banned: 1
  |- Total banned:    1
  `-' Banned IP list:  149.7.98.83
root@srv-khennouchi:~#
```

FIGURE 11 – Affichage de l’IP bannie

5 Conclusion

Ce TP a permis de démontrer les risques d’une infrastructure VoIP non sécurisée. Le passage au couple TLS/SRTP, complété par Fail2ban, assure une protection efficace contre les interceptions et les intrusions.