Mettre en place VNC et RDP sur une Raspberry :

Pré-requis :

Nom User : user

Mot de passe User : user

Mettre à jour la Raspberry PI 4 :

Commande dans le terminal
`sudo apt update` `sudo apt upgrade libxcb1 libx11-6`

Procédure d'introduction facultative pour vérifier que VNC marche correctement :

Récupérer l'adresse IP de la Raspberry Pi 4 :

Commande dans le terminal
`ip ad`

Installer et initialiser SSH sur la Raspberry Pi 4 :

Commande dans le terminal

`sudo apt install openssh-server`
`sudo systemctl enable ssh`
`sudo systemctl start ssh`

Sur mon PC PORTABLE pour se connecter à ma Raspberry PI 4 à distance via SSH :

Commande dans le terminal
`ssh user@[IP DE LA RASPBERRY]`

Si il y a une erreur lors de la connexion SSH à cause d'un conflit de clé, écrire cette commande (seulement si la connexion SSH est refusée) :

Commande dans le terminal
`ssh-keygen -f "/home/[NOM DE L'UTILISATEUR]/.ssh/known_hosts" -R "[IP DE LA RASPBERRY]"`

Installation du serveur VNC et d'autres librairies pour y avoir accès sur la Raspberry Pi 4 :

Commande dans le terminal

`sudo apt install x11vnc`
`sudo apt install novnc`
`sudo apt install websockify`
`sudo apt install xrdp`
`sudo apt install ufw`

Activer un serveur RDP pour manipuler la Raspberry Pi 4 à distance via Remina :

Commande dans le terminal

```
sudo ufw allow 3389/tcp
sudo systemctl restart xrdp
```

Tu dois installer sur ton ordinateur la librairie "novnc" pour pouvoir tester et avoir accès à distance.

Changer le protocole d'affichage pour qu'il soit compatible avec le serveur VNC :

Commande dans le terminal

```
sudo raspi-config
```

Puis aller dans "**Advanced Options**"

Puis aller dans "**A6**" où il y a écrit "**Wayland**", il faut « select »

Puis mettre "**x11**" au lieu de "**Wayland**"

Puis appuyer sur "**Ok**"

Documentation en ligne : https://itsfoss.com/raspberry-pi-os-switch-wayland-xorg/

Puis redémarrer :

Commande dans le terminal

```
sudo reboot
```

Maintenant on va tester en local sur la Raspberry PI 4 le serveur VNC :

Écrire cette commande dans une fenêtre du terminal :

Commande dans le terminal

```
sudo x11vnc -usepw -forever -display :0 -noxdamage
```

Cette commande lancera le serveur **VNC** et affichera le [**PORT**] du serveur VNC qui faudra mettre dans la commande suivante "**websockify**"

Écrire cette commande dans une deuxième fenêtre du terminal :

Commande dans le terminal

```
websockify –web /usr/share/novnc/ 6080 localhost:[PORT]
```

Mettre dans [**PORT**], le port récupérer précédemment. Normalement c'est soit "5900" ou "5901" si le port "5900" est déjà utilisé.

Puis aller sur un navigateur :

Url dans le navigateur

```
http://[IP DE LA RASPBERRY]:6080/vnc.html
```

Exemple : http://192.168.20.26:6080/vnc.html

Créer les services automatique pour que le serveur VNC se lance automatiquement à chaque allumage de la Raspberry PI 4 :

Commande dans le terminal

```
sudo nano /etc/systemd/system/x11vnc.service
```

Ajouter ça dans le fichier :

Instruction dans le fichier

```
[Unit]
Description=Start X11VNC Server
After=display-manager.service
Wants=display-manager.service

[Service]
Type=simple
ExecStart=/usr/bin/x11vnc -usepw -forever -display :0 -noxdamage
Restart=always
User=user
Environment=DISPLAY=:0

[Install]
WantedBy=multi-user.target
```

Ajouter un mot de passe si ce n'est pas déjà fait pour assurer une connexion sécurisé :

Instruction dans le fichier

```
x11vnc -storepasswd
```

Puis sauvegarder et démarrer le service créé :

Commande dans le terminal

```
sudo systemctl daemon-reload
sudo systemctl enable x11vnc
sudo systemctl start x11vnc
```

Créer les services automatique pour que le serveur Websockify (c'est ce qui envoi le flux VNC en temps-réel au navigateur Web) :

Commande dans le terminal

```
sudo nano /etc/systemd/system/websockify.service
```

Ajouter ça dans le fichier (attention : il faut mettre le [PORT] trouvé précédemment soit "5900", soit "5901" ou soit un autre) :

Instruction dans le fichier

```
[Unit]
Description=Start Websockify Server
After=x11vnc.service
Wants=x11vnc.service

[Service]
Type=simple
ExecStart=/usr/bin/websockify --web /usr/share/novnc/ 6080 localhost:[PORT]
Restart=always
User=user

[Install]
WantedBy=multi-user.target
```

Puis sauvegarder et démarrer le service créé :

Commande dans le terminal

```
sudo systemctl daemon-reload
sudo systemctl enable websockify
sudo systemctl start websockify
```

Maintenant on va vérifier les statuts des services créé pour voir si ils ont bien démarré sans bug :

Commande dans le terminal

```
sudo systemctl status x11vnc
sudo systemctl status websockify
```

Configuration VNC pour le lié le serveur VNC au Back-Office d'Acenstream :

Créer un dossier websockify dans /etc/ pour configurer et sécurisé notre serveur websockify :

Commande dans le terminal

```
sudo mkdir /etc/websockify
cd /etc/websockify
```

Puis créer un fichier "auth-source.json" :

Commande dans le terminal

```
sudo nano auth-source.json
```

Contenu :

Commande dans le terminal

```
{
    "header": "X-Websockify-Auth-Token",
    "value": "EpKnRna39p8TiqapNjRiMA2Ev1lmw6X38LXpEI/OTsxwnkywN09e1g"
}
```

Puis créer un fichier "CustomHeaderAuth.py" :

Commande dans le terminal

```
sudo nano CustomHeaderAuth.py
```

Contenu :

Commande dans le terminal

```
import json


class AuthenticationError(Exception):
    def __init__(self, log_msg=None, response_code=403, response_headers={}, response_msg=None):
        self.code = response_code
        self.headers = response_headers
        self.msg = response_msg

        if log_msg is None:
            log_msg = response_msg

        super().__init__('%s %s' % (self.code, log_msg))


class CustomHeaderAuth():
    """Verifies Custom Auth header. Specify src as JSON object {"header": "HEADER NAME", "value": "HEADER
VALUE"}"""

    def __init__(self, src=None):
        with open(src) as config:
            self.data = json.load(config)

    def authenticate(self, headers, target_host, target_port):
        auth_header = headers.get(self.data['header'])
        if not auth_header:
            self.demand_auth()

        if auth_header != self.data['value']:
            self.auth_error()

    def auth_error(self):
        raise AuthenticationError(response_code=403)

    def demand_auth(self):
        raise AuthenticationError(
            response_code=401,
            response_msg='Authentication is required'
        )
```

Puis créer un fichier "cert.pem" :

Commande dans le terminal

```
sudo nano cert.pem
```

Contenu :

Commande dans le terminal

```
-----BEGIN CERTIFICATE-----
MIIDlzCCAn+gAwIBAgIUFG4pbD+pYuuowxTuIIOFJX4UN3wwDQYJKoZIhvcNAQEL
BQAwWzELMAkGA1UEBhMCRlIxDzANBgNVBAgMBkZyYW5jZTETMBEGA1UEBwwKQ291
cmJldm9pZTEPMA0GA1UECgwGQWNlbnNpMRUwEwYDVQQDDAxwbGF5ZXIubG9jYWww
HhcNMjQwNDI5MTQyODQ0WhcNMzQwNDI3MTQyODQ0WjBbMQswCQYDVQQGEwJGUjEP
MA0GA1UECAwGRnJhbmNlMRMwEQYDVQQHDApDb3VyYmV2b2llMQ8wDQYDVQQKDAZB
Y2Vuc2kxFTATBgNVBAMMDHBsYXllci5sb2NhbDCCASIwDQYJKoZIhvcNAQEBBQAD
ggEPADCCAQoCggEBANtZB9HFwBrU361PR81ivuIBTViLICIdWfCtaPe6EJ1FLRvM
Q9P5eQNUuCs0Hz3Mcxq/E1S8CeFvM+FodrZphgs4VHbgmCIPDn/zsCVJEn6+MoQt
WKb36LGQ74d8PYEvbRAlD3v1iCkq52xEPULEPDoddMbQzDbhMdSQyZq0NiO0Mt7D
HYTyu48AFriWpVSAlJMhKx38vNtSVnsYe2Ieux0Z14q99EUuzpYeu+D5dhfpJ9MJ
AtTVqBf37yr7ZrITR26nG9L5xX7p4ocOEKb7mTVOQ5GDSxgcsZdHvqe+Z8RxbO0S
u21so3W+nDq9tzSznU0/uY5f5Cgjx+Hh45Vg1lkCAwEAAaNTMFEwHQYDVR0OBBYE
FL1aQxuzwALRcqx5uv0ytqvX+sqUMB8GA1UdIwQYMBaAFL1aQxuzwALRcqx5uv0y
tqvX+sqUMA8GA1UdEwEB/wQFMAMBAf8wDQYJKoZIhvcNAQELBQADggEBAK6RZ36Q
tbbjmcJ5Is6g3zwLQTqhYYW/nF4Ix2kcPNQRQ3PRAOC/5f/OB9RvluHfdzdePAnz
EiLNfmf1GhKUxuEvrha8b+q+3BtTOO63+ZhcVoxv6kSY+0L4nXVCkwmts5J9Y/U0
xtexHBk7rNggkCx6Bh80gJOf8DbbVzMOEZXN3WDKgVjozsbG7Qy52rVi7uRYmoCX
jTTqw6rwpwZ+zpQjmJXF78Fj/ym1+iNijwCmBsXwN13vQI9M7N7hFWQk8DmqbWEl
/qskn8/j5Bc5fxQ9vbiuTGPO04CBFuAmN+GevmTj76qyFPhl1Eo+4yXsNETLUXk6
0Qb/cL8KABDLnuc=
-----END CERTIFICATE-----
```

Puis créer un fichier "key.pem" :

Commande dans le terminal

```
sudo nano key.pem
```

Contenu :

Commande dans le terminal

```
-----BEGIN PRIVATE KEY-----
MIIEvgIBADANBgkqhkiG9w0BAQEFAASCBKgwggSkAgEAAoIBAQDbWQfRxcAa1N+t
T0fNYr7iAU1YiyAiHVnwrWj3uhCdRS0bzEPT+XkDVLgrNB89zHMavxNUvAnhbzPh
aHa2aYYLOFR24JgiDw5/87AlSRJ+vjKELVim9+ixkO+HfD2BL20QJQ979YgpKuds
RD1CxDw6HXTG0Mw24THUkMmatDYjtDLewx2E8ruPABa4lqVUgJSTISsd/LzbUlZ7
GHtiHrsdGdeKvfRFLs6WHrvg+XYX6SfTCQLU1agX9+8q+2ayE0dupxvS+cV+6eKH
DhCm+5k1TkORg0sYHLGXR76nvmfEcWztErttbKN1vpw6vbc0s51NP7mOX+QoI8fh
4eOVYNZZAgMBAAECggEAUUnfz3pxNKBiWoiQjAmFwx0HbIJ2wJ3PCJZDWdc/NAGA
c5ZSJrsEAVuCuyQgVdcwax+XyuW65aw/9k+v+u8FaaAREVMvniOlVH7bbx3y+DyF
XR2vBoXGWxPi6Pt+kaMhUIPjhqVaNiN7y1BIwbcWuNTUw596pvXy8RtNU4Nosvbz
3G7LhQLpDXWzUN+OQ9PKh3E9VQxgtes+34E4Na+z1RP60xOYQ86InR7JMCx40ImO
1b8BM9AXyi5YELKIt4VYRyutOjdGf4eGQFqoOXiLa8oR0O55Y4XyYZwUXt3lprU7
vEjYAfWsb2c0ILqYoTCge5ru37T3j3By317pjV965wKBgQDoLCDRb4cUQJWa1YMM
adRZ6uwuy6feFaH93L5LGrzJ7fppwwb1VkV4TWFZMzfaMclYZFDQ6MqULCL+PjPZ
4AdEmdQm3oh/wxnmN+fw5xlnnZairL7taXrY9MiEwaWCTbuyviVO1B6K4Q9WNv6W
4O4mZ/11tvMxvZYOB72Px5SCzwKBgQDx2/YT/VzgDqTT0c1cIv6sIe9r3h25sXex
lmNXNfcq6+M1D8fIeNcLhlQStDCD5J+j7dakOV1S0BqW6NIN80vETN0n4G0cqOW/
oR4Ijy8fvSw2+nd4aEmM4AGC0WCU2CbeWOSpfM/8iuLKFNB84SWkzp0aN0qNw1HL
lT3XWbH+VwKBgD6BguRfzlIZgS2KRhnVdCIxKR0AH1WfJVmfUGwP6rvCoiRPKVRr
wZA95POK4oKP3NE7ULattbxmlZU1UehadBp3/7HnP4CGGjbs3jjxPtmyh5eMQmrQ
dw5p+IIlUFXgb6QTpR+v4/kn1mUzsR8Vt4gHfylt89N0Yy2rz9Ehvk3pAoGBAOv6
+ja21o+muhwScJSdcG7zDW/p4Ib2KRsU6QiGlwtB7QYwx3VKskeEYTN+rHeD8l84
CF1juzK56bekoo8J+7skZAQsbUmxYbl/spYXXptRMHrUFw1cc6cISXEiaWHRJNt9
UF4ucaCuwmj5thKieuA61/5kRFHdaUmB7DzaP1kxAoGBAKd0KkEdrwpR8at0sY+a
QNtfLgSomK5GZR1qudHHupVPpb61aLyzuaTecO3litq9LJYFOQMDFY3Je5dNrlaj
k9AcukhDZv36/BDq/GjqdWwrsbHJ97K2Q5FglnnWwAZAx35sJ3/l4qQVZ2xGZhdQ
5yheMNbf1hMlbsEVh0is+8WZ
-----END PRIVATE KEY-----
```

Puis ajouter cette variable pour python :

Commande dans le terminal

```
export PYTHONPATH=/etc/websockify/CustomHeaderAuth.py
```

Créer le fichier qui lancera automatiquement le serveur VNC sécurisé et les fichiers qui stockeront les logs :

Commande dans le terminal

```
sudo touch /home/user/start_vnc.sh
sudo touch /var/log/x11vnc.log
sudo touch /var/log/websockify.log
```

Aller dans le fichier "start_vnc.sh" puis mettez ceci :

Commande dans le terminal

```
// Se qu'il faut mettre dans le file
#!/bin/bash

# Exporter la variable d'affichage
export DISPLAY=:0

# Attendre quelques secondes pour s'assurer que l'affichage est prêt
sleep 5

# Démarrer x11vnc
sudo x11vnc -shared -forever -display :0 -nopw -listen 0.0.0.0 -auth guess -noxdamage >> /var/log/x11vnc.log
2>&1 &

# Exporter le chemin PYTHONPATH pour websockify
export PYTHONPATH=/etc/websockify:$PYTHONPATH

# Démarrer websockify avec SSL
python3 -m websockify -v --auth-plugin=CustomHeaderAuth.CustomHeaderAuth --auth-source /etc/websockify/auth-
source.json --cert=/etc/websockify/cert.pem --key=/etc/websockify/key.pem --ssl-only 443 127.0.0.1:5900 >> /var
/log/websockify.log 2>&1 &
```

Accorder les permissions à l'utilisateur "user" au fichier :

Commande dans le terminal

```
sudo chown user:user /home/user/start_vnc.sh
sudo chown user:user /etc/websockify/CustomHeaderAuth.py
sudo chown user:user /etc/websockify/auth-source.json
sudo chown user:user /etc/websockify/cert.pem
sudo chown user:user /etc/websockify/key.pem
sudo chown user:user /etc/websockify/
sudo chown user:user /home/user/.Xauthority
sudo chown user:user /var/log/websockify.log
sudo chown user:user /var/log/x11vnc.log
sudo chmod +x /home/user/start_vnc.sh
```

Accorder les droits à l'utilisateur "user" de faire écouter un serveur python sur un port inférieur à 1024 ici 443 :

Commande dans le terminal

```
COMMAND: which python3
RESPONSE: /usr/bin/python3

COMMAND: ls -l $(which python3)
RESPONSE: lrwxrwxrwx 1 root root 10 Apr  9  2023 /usr/bin/python3 -> python3.11

COMMAND: sudo setcap cap_net_bind_service=+ep /usr/bin/python3.11

COMMAND: getcap /usr/bin/python3.11
RESPONSE: /usr/bin/python3.11 = cap_net_bind_service+ep
```

Création d'un tache automatique (choisir l'option 1 "/bin/nano") :

Commande dans le terminal

```
crontab -e
```

Et rajouter à la fin du fichier ceci :

Commande dans le terminal

```
@reboot /home/user/start_vnc.sh
```

Créer un fichier qui lancera automatiquement le script à chaque démarrage :

Commande dans le terminal

```
sudo nano /etc/rc.local
```

Ajouter ceci au fichier :

Commande dans le terminal

```
/home/user/start_vnc.sh
```

Avant de redémarrer il faut que l'utilisateur "user" est bien tout les droits :

Commande dans le terminal

```
sudo usermod -aG video user
```

Aller dans le fichier root pour ajouter ou juste vérifier que l'utilisateur "user" est bien tout les droits :

Commande dans le terminal

```
sudo visudo
```

Puis ajouter cette ligne dans le fichier :

Commande dans le terminal

```
user ALL=(ALL:ALL) ALL
```

Aller dans le fichier "lightdm.comf" (fichier qui configure l'affichage de l'écran) :

Commande dans le terminal

```
sudo nano /etc/lightdm/lightdm.conf
```

Ajouter dans la section "[Seat:*]" :

Commande dans le terminal

```
[Seat:*]
xserver-allow-tcp=true
```

Et enfin ajouter cette permission (facultatif) :

Commande dans le terminal

```
sudo chmod 777 /var/run/lightdm/root/:0
```

Puis redémarrer :

Commande dans le terminal

```
sudo reboot
```

Puis quand elle a redémarrer regarder si les ports 5900 (serveur x11vnc) et 443 (serveur websockify) ont bien démarrés et sont en train d'écouter (ils doivent avoir le "state" en "LISTEN") :

Commande dans le terminal

```
sudo netstat -tuln
```

Désactiver le mode vieille sur la Raspberry PI :

Documentation en plus : https://raspberrytips.fr/desactiver-mode-veille-raspberry-pi/

> Attention, si le mode veille n'est pas désactivée sur la Raspberry, cela provoque l'extinction de l'affichage des flux envoyées par Acenstream.

Commande dans le terminal

```
export DISPLAY=:0
xset s off
xset -dpms
xset s noblank
```

Commande dans le terminal

```
sudo raspi-config
```

Puis : Aller dans "Display Options" Sélectionner "Screen Blanking" Désactiver-le :

Commande dans le terminal

```
sudo nano /etc/lightdm/lightdm.conf
```

Dans la section `[Seat:*]`, ajouter ou modifier cette ligne :

Commande dans le terminal

```
xserver-command=X -s 0 -dpms
```

Forcer le HDMI pour qu'il soit toujours actif :

Commande dans le terminal

```
sudo nano /boot/firmware/config.txt
```

Ajoute à la fin du fichier :

Commande dans le terminal

```
hdmi_ignore_edid=0xa5000080
hdmi_force_hotplug=1
hdmi_group=1
hdmi_mode=4
config_hdmi_boost=7
hdmi_blanking=0
hdmi_channel=0
hdmi_drive=2
hdmi_safe=1
```

Puis reboot la Raspberry Pi :

Commande dans le terminal

```
sudo reboot
```