

WHISKR

INSTALLATION DU PRODUIT



#SAE302

#Security

#Network

<https://github.com/Ykari68/SAE302/>

Cette documentation est un guide d'installation du produit Whiskr. Le produit est constitué d'un serveur, unique, et de clients.

Le serveur est en standalone, ce qui veut dire que la base de données qui lui est associée devra être installée dans la même machine.

Table des matières

I. Installation du serveur	3
1/ Installation de la base de données MySQL	3
2/ Installation du produit serveur	5
a. Prérequis.....	5
b. Installation de l'environnement python.....	5
c. Lancement du serveur	6
II. Installation du client	6
1/ Configuration du client sur Windows	6
2/ Contourner Windows Defender	7
3/ Configuration du client sur Linux	11
a. Prérequis.....	11
b. Paramétrage du client.....	11



I. Installation du serveur

1/ Installation de la base de données MySQL

Tout d'abord, nous aurons besoin d'une base de données MySQL fonctionnel pour faire fonctionner le serveur.

Pour installer MySQL :

<code>toto@debian:~\$ su -</code> <code>Mot de passe :</code>	Sur une machine Debian 11, s'accorder les droits de superutilisateur
<code>root@debian:~# cd /tmp</code> <code>root@debian:/tmp#</code>	Se rendre sur le dossier tmp
<code>wget https://dev.mysql.com/get/mysql-apt-config_0.8.29-1_all.deb</code>	Télécharger le fichier d'installation mysql
<code>root@debian:/tmp# ls</code> <code>mysql-apt-config_0.8.29-1_all.deb</code>	Vérifier que le fichier est présent
<code>root@debian:/tmp# dpkg -i mysql-apt-config_0.8.29-1_all.deb</code> Sélection du paquet mysql-apt-config précédemment désélectionné. (Lecture de la base de données... 137316 fichiers et répertoires déjà installés.) Préparation du dépaquetage de mysql-apt-config_0.8.29-1_all.deb ... Dépaquetage de mysql-apt-config (0.8.29-1) ... Paramétrage de mysql-apt-config (0.8.29-1) ...	« Déballer » le paquet installé. Une interface s'ouvrira. Par défaut mysql 8.0 est choisi, appuyer sur « Ok ».
<code>root@debian:/tmp# apt update</code>	Mettre à jour le répertoire d'apt
<code>root@debian:/tmp# apt install mysql-server</code>	Installer le serveur mysql (Un mot de passe sera demandé, il s'agit du mot de passe de l'administrateur). Choisissez ensuite les options recommandées.
<code>root@debian:/tmp# systemctl status mysql</code>	Vérifier le statut du service (active).

Désormais, le serveur MySQL est installé, mais la base de données n'y est pas. Elle se trouve dans le répertoire git du produit.

<pre>root@debian:~# apt install git</pre>	Installer git
<pre>root@debian:~# su</pre>	Assurer vous d'être dans le répertoire /root (Important pour la suite)
<pre>git clone https://github.com/Ykari68/SAE302</pre>	Cloner le répertoire SAE302 dans la machine Debian
<pre>root@debian:~# ls SAE302</pre>	Vérifier si le dossier est présent
<pre>root@debian:~# mysql -u root -p Enter password:</pre>	Se connecter au serveur MySQL
<pre>mysql> create database serveur -> ; Query OK, 1 row affected (0,00 sec)</pre>	Créer la base « serveur »
<pre>exit</pre>	Quitter mysql
<pre>mysql -u root -p serveur < SAE302/serveur/sql/save.sql</pre>	Importer la base de données sur « serveur »

MySQL est désormais installé avec la base de données utilisée par le produit.

2/ Installation du produit serveur

Maintenant que notre base de données est installée, nous pouvons installer le serveur qui va permettre au produit de fonctionner.

a. Prérequis

Si ce n'est pas déjà fait, il faut s'assurer que le dossier SAE302 est présent dans root :

```
root@debian:~# git clone https://github.com/Ykari168/SAE302
Clonage dans 'SAE302'...
remote: Enumerating objects: 155, done.
remote: Counting objects: 100% (155/155), done.
remote: Compressing objects: 100% (97/97), done.
remote: Total 155 (delta 60), reused 141 (delta 46), pack-reused 0
Réception d'objets: 100% (155/155), 40.64 KiB | 1.27 MiB/s, fait.
Résolution des deltas: 100% (60/60), fait.
```

Si vous souhaitez installer le serveur dans un autre emplacement, alors il faudra modifier le fichier suivant :

nano SAE302/serveur/serveur.py

```
with open('SAE302/serveur/config.txt', 'r') as file:
    lines = file.readlines()
```

Modifier la partie 'SAE302/serveur/config.txt' par 'vote/emplacement/SAE302/serveur/config.txt'.

b. Installation de l'environnement python

Il est impératif de s'assurer que python soit bien installé avec les paquets nécessaires :

<pre>root@debian:~# python3 --version Python 3.7.3</pre>	Vérifier si python3 est installé
apt install python3	Sinon, installer python3
apt install python3-pip	Installer pip
pip3 install --upgrade pip	Mettre à jour pip
pip3 install -r votre/emplacement/SAE302/serveur/requirements.txt	Installer les dépendances du serveur
pip3 install mysql-connector-python pip3 install bcrypt	Si cela génère des erreurs, installer manuellement les dépendances

c. Lancement du serveur

Le serveur est installé avec la base de données et l'environnement python est correctement configuré.

Il ne reste plus que quelques étapes avant de pouvoir utiliser le serveur Whiskr.

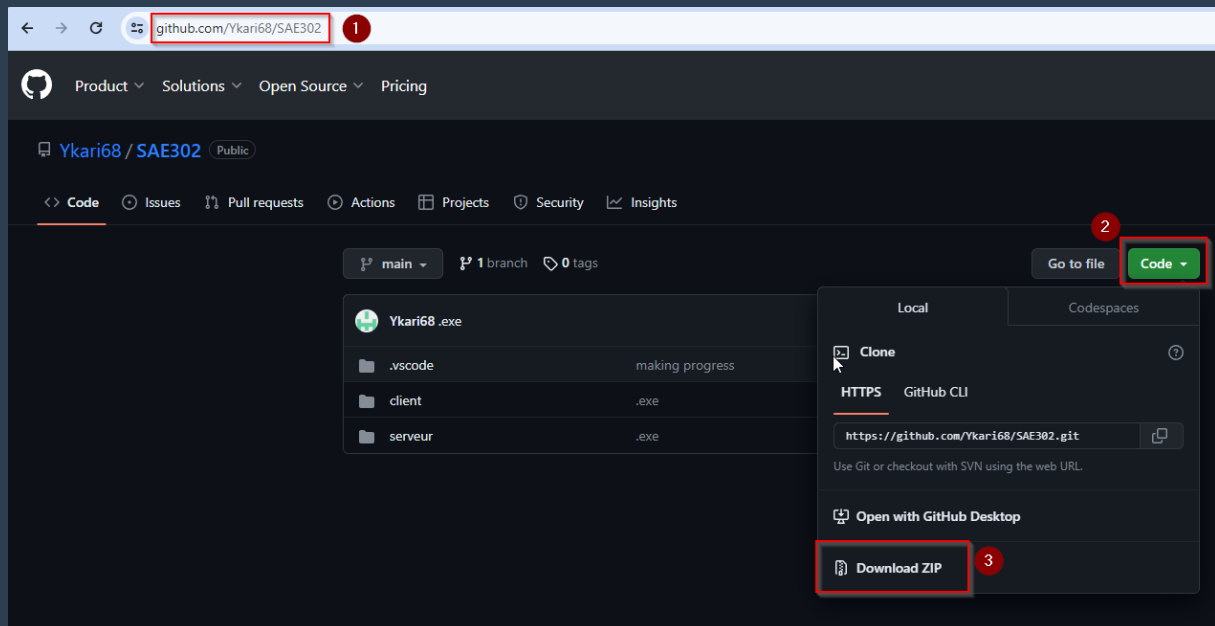
<pre>root@debian:~# nano SAE302/serveur/config.txt</pre>	Modifier ce fichier si vous souhaitez modifier le port d'écoute (par défaut 6255)
<pre>root@debian:~# chmod +x SAE302/serveur/serveur.py</pre>	Donner les droits d'exécution au fichier serveur.py
<pre>root@debian:~# python3 SAE302/serveur/serveur.py</pre>	Exécuter le fichier serveur.py

Le serveur Whiskr est désormais installé sur votre machine, il ne reste plus qu'à lancer les clients pour avoir un produit complet et fonctionnel.

II. Installation du client

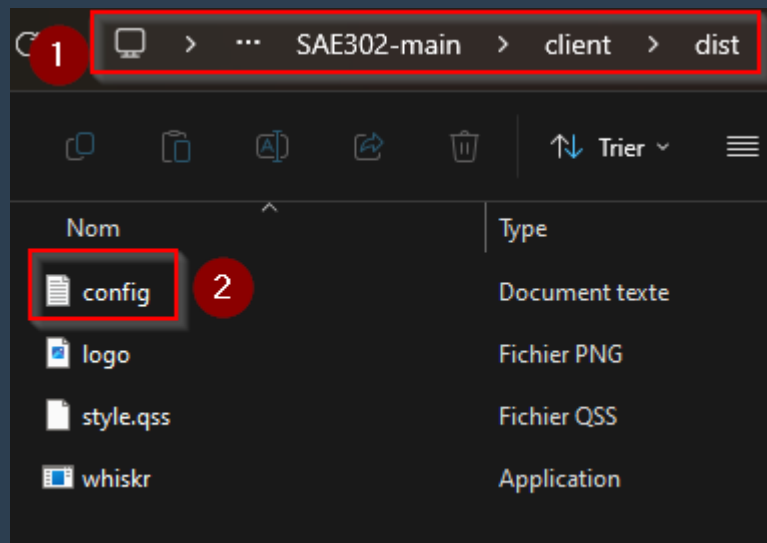
1/ Configuration du client sur Windows

Afin d'avoir un client fonctionnel sur Windows, il faut télécharger le dossier SAE302 sur github.



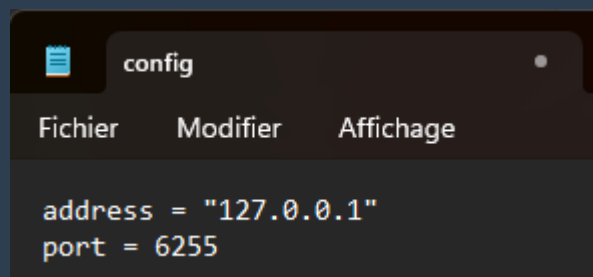
The screenshot shows the GitHub repository page for Ykari68/SAE302. The browser address bar displays the URL 'github.com/Ykari68/SAE302', which is highlighted with a red box and a red circle containing the number 1. The repository page shows the 'Code' button with a red box and a red circle containing the number 2. The 'Download ZIP' option is highlighted with a red box and a red circle containing the number 3.

Ensuite, ouvrir le fichier **config.txt** se trouvant dans le dossier **dist** du dossier **client**.



Dans ce fichier, il y a deux valeurs à renseigner :

- L'adresse du serveur (Renseigner l'adresse du serveur whiskr)
- Le port d'écoute du serveur (Renseigner le port mis dans le fichier config.txt du serveur whiskr)

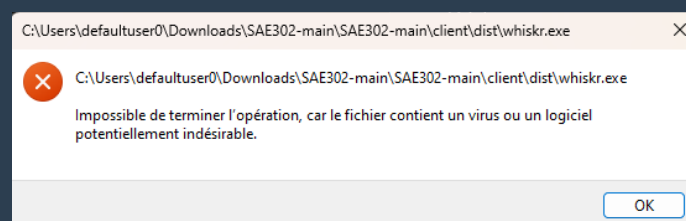


Par défaut, l'adresse du serveur est l'adresse locale et le port est 6255.

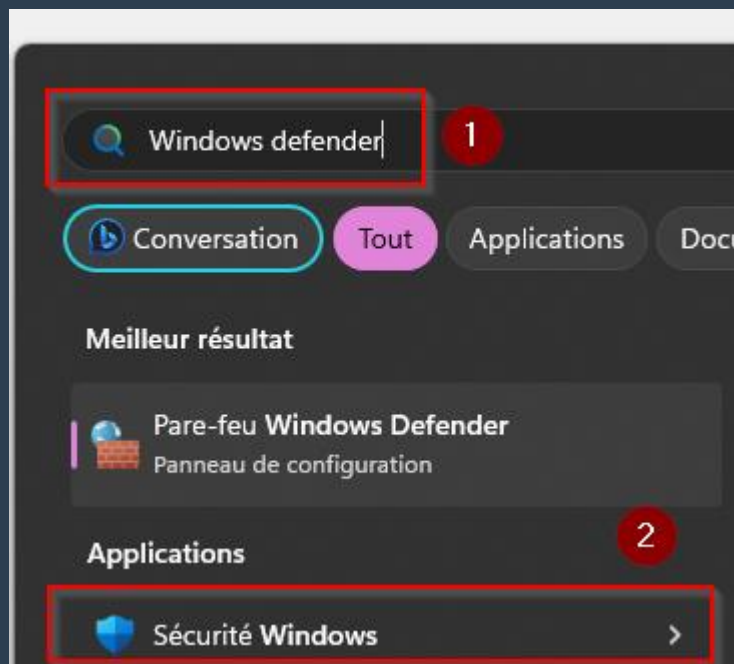
2/ Contourner Windows Defender

La configuration du client est désormais terminée, le client est fonctionnel et prêt à être lancé.

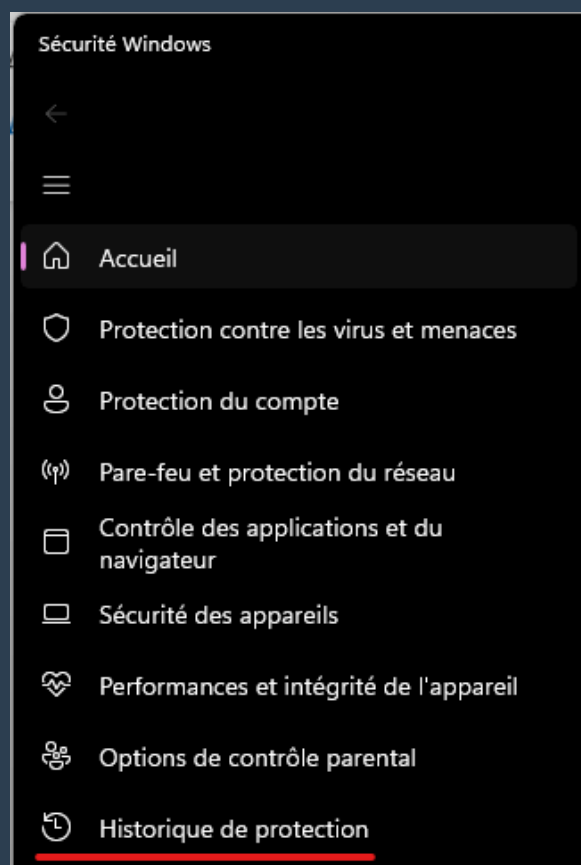
Cependant, il y a de forte chance que Windows Defender mette le fichier sous quarantaine, avec le message d'alerte suivant :



Lorsque cela arrive, il faudra alors ouvrir dans les paramètres Windows Defender :



Ensuite, ouvrir l'historique de protection




Trouver la menace de mise en quarantaine correspondante (le fichier mis en quarantaine est spécifié dans l'alerte), puis le restaurer sous « Actions ».


Historique de protection

Consultez les dernières actions de protection et les recommandations de Sécurité Windows.

Tous les éléments récents Filtres ▾




Les paramètres permettant de bloquer les applications...
28/12/2023 21:57



Menace bloquée
28/12/2023 21:52

Grave



Menace mise en quarantaine
28/12/2023 21:52

1 Grave ▴


Détecté : Trojan:Win32/Wacatac.B!ml
État : En quarantaine
Les fichiers en quarantaine se trouvent dans une zone sécurisée dans laquelle ils ne peuvent endommager votre appareil. Ils seront supprimés automatiquement.

Date : 28/12/2023 21:52
Détails : Ce programme est dangereux et il exécute des commandes émanant d'une personne malveillante.

Éléments affectés :

file: C:\Users\defaultuser0\Downloads\SAE302-main\SAE302-main\client\dist\whiskr.exe

[En savoir plus](#)



Menace bloquée
28/12/2023 21:52

2 Actions ▾

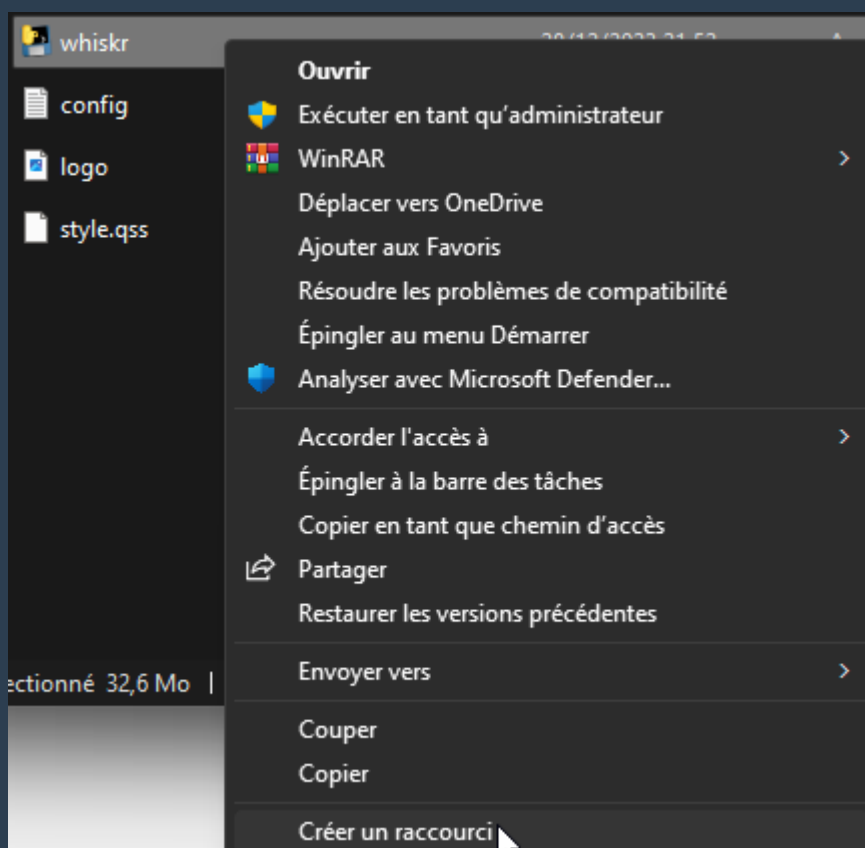
Restaurer

Supprimer

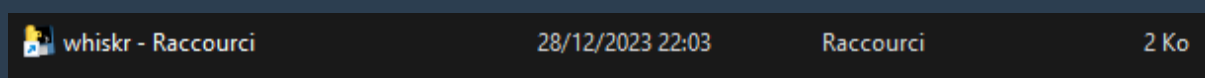
Le client whiskr est fonctionnel et désormais utilisable. Afin de ne pas avoir à naviguer dans les dossiers à chaque lancement, il est recommandé de créer un raccourci pour l'Application, ou alors de l'ajouter à la barre des tâches :

Raccourci :

Click droit sur l'application, puis créer un raccourci.

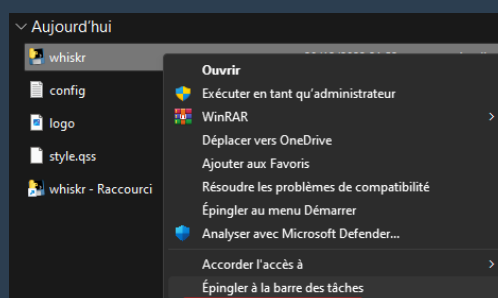


Le dossier créer pourra alors être déplacé où l'on veut sur le poste Windows.



Barre des tâches :

Click-droit sur l'application, puis « Epingler à la barre des tâches »



3/ Configuration du client sur Linux

a. Prérequis

Il est impératif de s'assurer que python soit bien installé avec les paquets nécessaires :

<code>root@debian:~# python3 --version</code> Python 3.7.3	Vérifier si python3 est installé
<code>apt install python3</code>	Sinon, installer python3
<code>apt install python3-pip</code>	Installer pip
<code>pip3 install --upgrade pip</code>	Mettre à jour pip
<code>pip3 install PyQt6</code>	Installer les dépendances du serveur

b. Paramétrage du client

Afin d'avoir le client sur Linux, il faut suivre les étapes suivantes :

<code>toto@debian:~\$ git clone https://github.com/Ykari68/SAE302</code> Clonage dans 'SAE302'... remote: Enumerating objects: 198, done. remote: Counting objects: 100% (198/198), done. remote: Compressing objects: 100% (123/123), done. remote: Total 198 (delta 72), reused 184 (delta 58), pack-reused 0 Réception d'objets: 100% (198/198), 34.09 Mio 23.65 Mio/s, fait. Résolution des deltas: 100% (72/72), fait.	Cloner le dossier SAE302 sur votre répertoire utilisateur
<code>toto@debian:~\$ ls</code> Bureau Images Musique <u>SAE302</u> Vidéos Documents Modèles Public Téléchargements	Vérifier la présence du dossier
<code>toto@debian:~/SAE302/client\$ cd SAE302/client/</code>	Se déplacer dans le dossier client
<code>toto@debian:~/SAE302/client\$ nano client2.py</code>	Modifier client2.py

<pre>GNU nano 5.4 client2.py import socket import threading import sys import time from PyQt6 import * from PyQt6.QtWidgets import QApplication, QWidget, QPushButton, QLineEdit, QLabel from PyQt6.QtCore import Qt, pyqtSignal, pyqtSlot, QThread from PyQt6.QtGui import QPixmap, QIcon, QGuiApplication, QTextCursor import os # Récupérer le chemin absolu du répertoire de l'exécutable executable_directory = os.path.dirname(sys.executable) # Construire le chemin relatif des fichiers par rapport à l'exécutable config_file_path = '/home/toto/SAE302/client/dist/config.txt' logo_file_path = '/home/toto/SAE302/client/dist/logo.png' style_file_path = '/home/toto/SAE302/client/dist/style.qss'</pre>	<p>Il faut modifier les chemins d'accès en fonction de votre structure.</p>
<pre>toto@toto:~/SAE302/client\$ nano dist/config.txt</pre>	<p>Modifier le fichier config.txt</p>
<pre>address = "127.0.0.1" port = 6255</pre>	<p>Il faudra y renseigner l'adresse du serveur whiskr ainsi que son port d'écoute.</p>
<pre>root@debian:~# chmod 777 /home/toto/SAE302/client/client2.py root@debian:~# S</pre>	<p>(Optionnel) En tant qu'administrateur ou en sudo, donner les droits d'exécution à tous les utilisateurs.</p>
<pre>toto@toto:~/Desktop\$ nano /home/toto/SAE302/client/dist/whiskr.sh</pre>	<p>Modifier le fichier whiskr.sh</p>
<pre>#!/bin/sh python3 /home/toto/SAE302/client/client2.py</pre>	<p>Mettre le chemin absolu du client2.py</p>
<pre>toto@toto:~/Desktop\$ bash whiskr.sh</pre>	<p>Le fichier whiskr.sh peut être déplacé où l'on souhaite, et être lancé avec bash.</p>