Université de Limoges
Faculté des Sciences et Techinques

Rapport de projet TIC

Authentification, Web sécurisé et Stéganographie

Réalisé par :

Eid Maalouf
Fadwa Gardani
Adam Akkouche

8 mai 2020

1 Implémentation

Nous avons décidé de créer un fichier `library.py` qui contient toutes les fonctions nécessaires pour la gestion des requêtes demandés du WebService.

2 Fonctionnement du programme

Dans notre programme, c'est le serveur frontal qui demande la création ou pas des certificats générés par les courbes elliptiques. Ensuite, il envoie à l'utilisateur un certificat d'accès au Webservice et au Webservice les certificats d'autorisation d'accès pour le détenteur du certificat d'accès.

Lors de la requête faite par l'utilisateur au Webservice, l'attestation est créée puis sont conservées dans les données du Webservice :

- Les certificats privés d'accès au Webservice.
- Le fichier `response.tsr` contenant la réponse à la requête d'horodatage.
- La clef privée qui permet de vérifier le bloc d'information.
- La taille du bloc d'information avec le timestamp pour être géré par la suite par la fonction de stéganographie.

Dans notre programme, certains fichiers qui ne sont pas réellement importants sont conservés. Lors de la partie vérification, notamment le résultat du déchiffrement de la signature dissimulée par stéganographie, sont conservée aussi des copie de l' attestation lors respectivement de sa création et de sa vérification.

Lorsque le Webservice envoie à l'utilisateur l'attestation, celle-ci est conservée dans le répertoire `user`.

Ensuite l'employeur envoie l'attestation au Webservice. L'étudiant et l'employeur sont dans le même répertoire par commodité (nous supposons que l'étudiant a fait passer l'attestation à l'employeur de mains à mains), une vérification de l' attestation reçue est alors réalisée par le Webservice. De même l'étudiant envoie

Pour nous le timestamp qui a été dissimulé par stéganographie est comparé à la date présente dans le fichier `response.tsr`.

Lors de la vérification de l'attestation, si les dates ne correspondent pas, le Webservice se ferme automatiquement alors que si il y' a une erreur de signature il renvoie à l'employeur le message "not a good attestation". En revanche, si l' attestation est bonne il renvoie "ok".

Enfin dans notre programme nous avons choisi de faire un script bash pour les commandes curl, notre principal intérêt était de faciliter la lecture du code à l' aide de paramètres, mais le choix non judicieux des paramètres a rendu le code plus illisible.

3 Les fonctions du fichier `library.py`

— La fonction **poster** : permet de récupérer le nom et le prénom de l'utilisateur ainsi que l'intitulé de la certification.

— La fonction **sign** : permet de signer les fichiers avec l'algorithme de chiffrement RSA

— La fonction **stegano** : permet de faire la stéganographie du nom, prénom, intitulé de la certification et la date au forme timestamp

— La fonction **def-create** : permet d'intégrer le QR code qui contient la signature du bloc d'information (nom-prénom, intitulé de la cerification)

— La fonction **recup-stegano** : permet de récupérer les données dissimilées par stéganographie

— La foction **verify** : permet de vérifier si le timestamp a été modifié ou pas

— La fonction **padding** : permet d'obtenir une chaine de 64 caractère (en ajoutant des "0") du nom, prénom et intitulé certification.

— La fonction **dateTotimestamp** : permet de transformer la forme de la date en forme de timestamp.

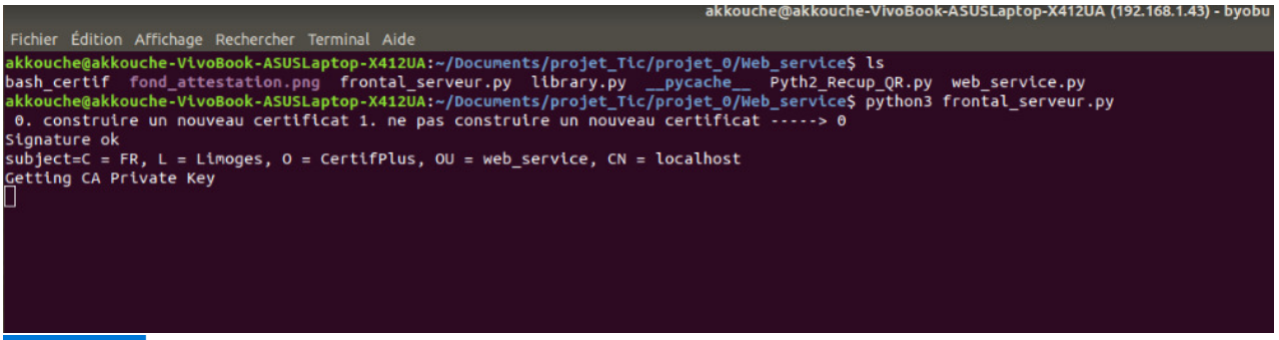
— La fonction **comparaison chaine** : permet de monter que deux chaines de caractères sont bien identiques.

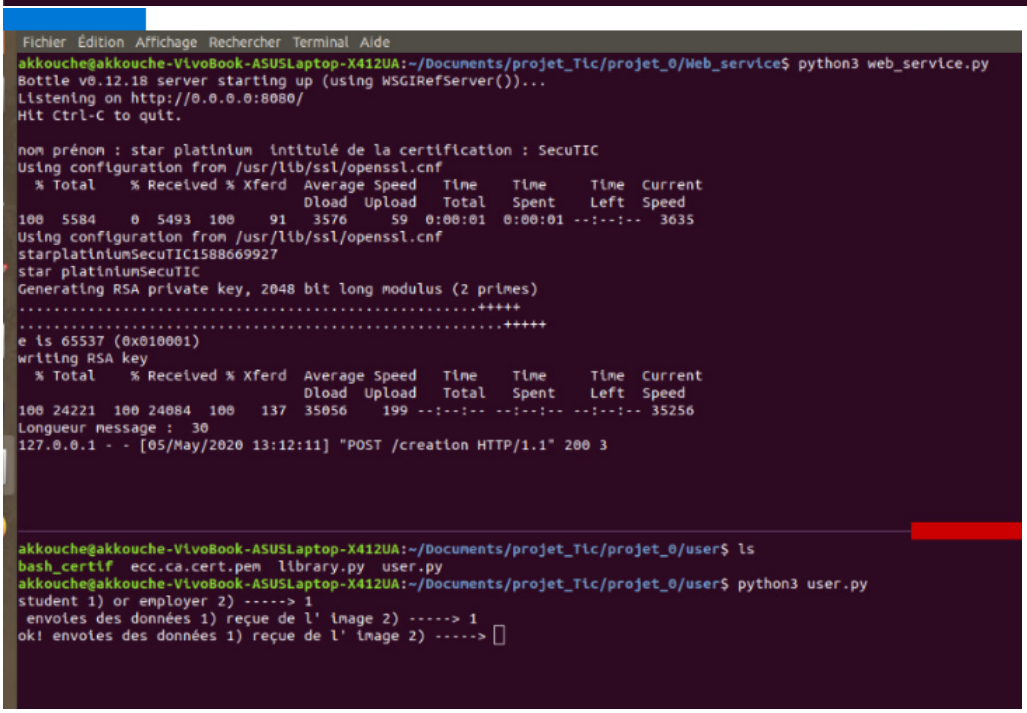
4 Lancement du programme

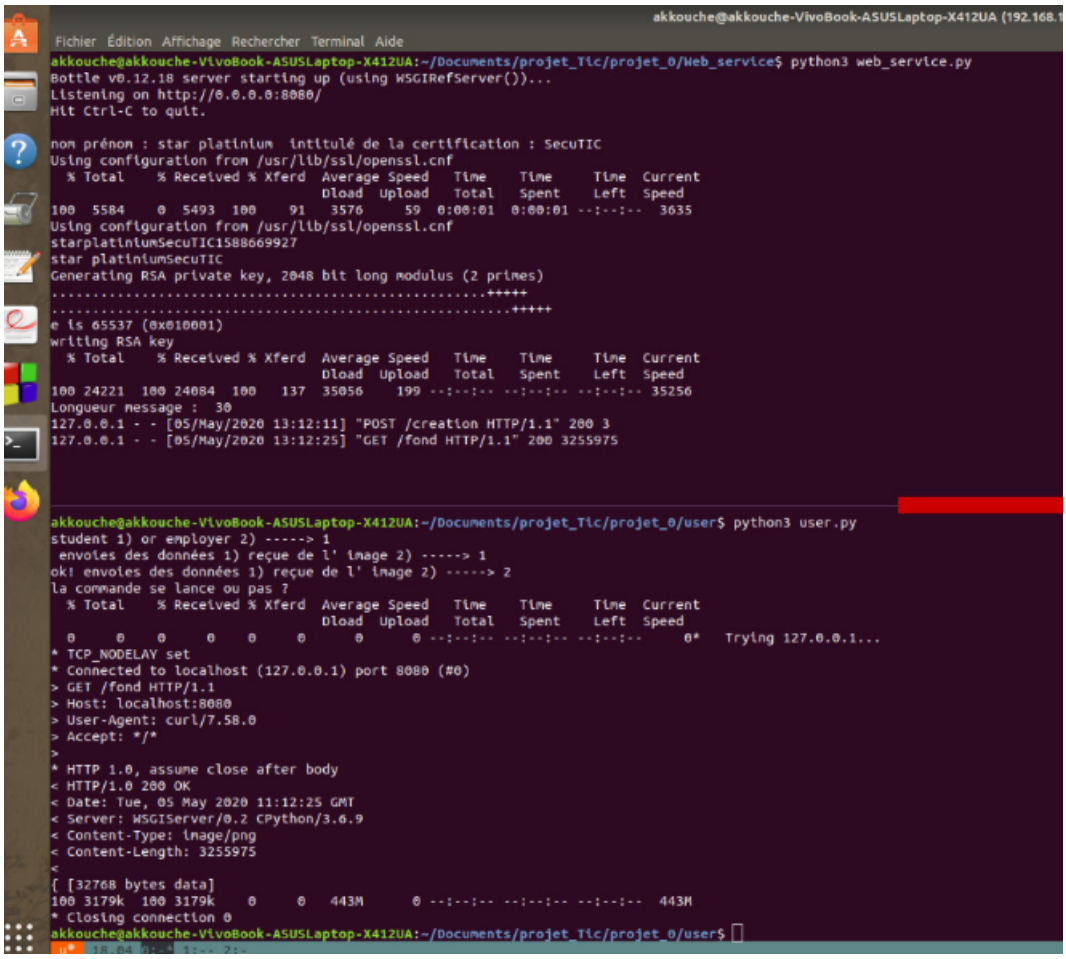
On crée un répertoire contenant 2 répertoires : `user` et `web-service`.
Le répertoire `web-service` contient les fichiers : ‘`bash-certif`, `webservice.py`, `frontalserveur.py`, `fond-attestation.png` et `library.py`.
Le répertoire `user` contient : `user.py`, `bash-certif` et `library.py`.
On ouvre trois terminaux : d'abord on lance dans l'un des terminaux en premier lieu du frontal-serveur, aprés on lance dans un second terminal `webservice.py`, enfin on lance dans le troisième `user.py`.

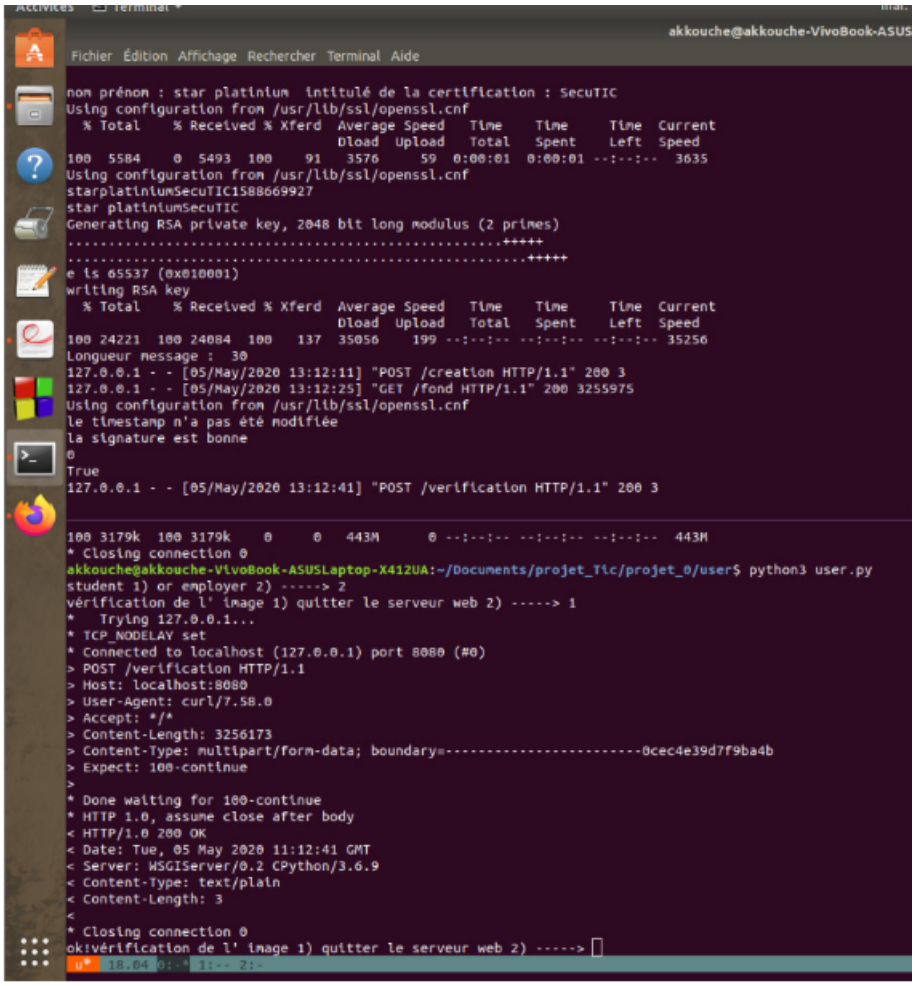
5 Jeu d'essai

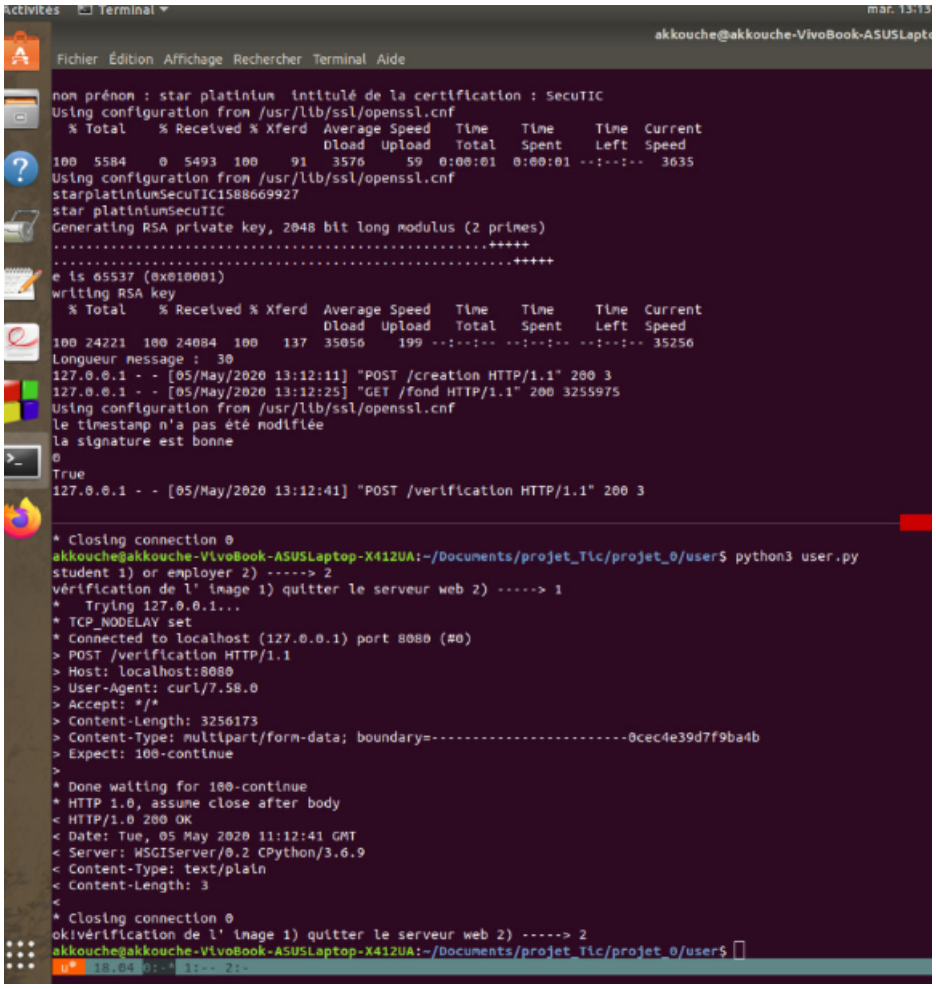
Voici des photos pour tester nos programme.


akkouche@akkouche-VivoBook-ASUSLaptop-X412UA (192.168.1.43) - byobu
Fichier Édition Affichage Rechercher Terminal Aide
akkouche@akkouche-VivoBook-ASUSLaptop-X412UA:~/Documents/projet_Tic/projet_0/Web_service$ ls
bash_certif fond_attestation.png frontal_serveur.py library.py __pycache__ Pyth2_Recup_QR.py web_service.py
akkouche@akkouche-VivoBook-ASUSLaptop-X412UA:~/Documents/projet_Tic/projet_0/Web_service$ python3 frontal_serveur.py
0. construire un nouveau certificat 1. ne pas construire un nouveau certificat -----> 0
Signature ok
subject=C = FR, L = Limoges, O = CertifPlus, OU = web_service, CN = localhost
Getting CA Private Key


Fichier Édition Affichage Rechercher Terminal Aide
akkouche@akkouche-VivoBook-ASUSLaptop-X412UA:~/Documents/projet_Tic/projet_0/Web_service$ python3 web_service.py
Bottle v0.12.18 server starting up (using WSGIRefServer())...
Listening on http://0.0.0.0:8080/
Hit Ctrl-C to quit.
nom prénom : star platinium intitulé de la certification : SecuTIC
Using configuration from /usr/lib/ssl/openssl.cnf
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 5584 0 5493 100 91 3576 59 0:00:01 0:00:01 --:--:-- 3635
Using configuration from /usr/lib/ssl/openssl.cnf
starplatiniumSecuTIC1588669927
star platiniumSecuTIC
Generating RSA private key, 2048 bit long modulus (2 primes)
.....+++++
.....+++++
e is 65537 (0x010001)
writing RSA key
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 24221 100 24084 100 137 35056 199 --:--:-- --:--:-- --:--:-- 35256
Longueur message : 30
127.0.0.1 - - [05/May/2020 13:12:11] "POST /creation HTTP/1.1" 200 3
akkouche@akkouche-VivoBook-ASUSLaptop-X412UA:~/Documents/projet_Tic/projet_0/user$ ls
bash_certif ecc.ca.cert.pem library.py user.py
akkouche@akkouche-VivoBook-ASUSLaptop-X412UA:~/Documents/projet_Tic/projet_0/user$ python3 user.py
student 1) or employer 2) -----> 1
envoies des données 1) reçue de l' image 2) -----> 1
ok! envoies des données 1) reçue de l' image 2) ----->


akkouche@akkouche-VivoBook-ASUSLaptop-X412UA (192.168.1
Fichier Édition Affichage Rechercher Terminal Aide
akkouche@akkouche-VivoBook-ASUSLaptop-X412UA:~/Documents/projet_Tic/projet_0/Web_service$ python3 web_service.py
Bottle v0.12.18 server starting up (using WSGIRefServer())...
Listening on http://0.0.0.0:8080/
Hit Ctrl-C to quit.
nom prénom : star platinium intitulé de la certification : SecuTIC
Using configuration from /usr/lib/ssl/openssl.cnf
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 5584 0 5493 100 91 3576 59 0:00:01 0:00:01 --:--:-- 3635
Using configuration from /usr/lib/ssl/openssl.cnf
starplatiniumSecuTIC1588669927
star platiniumSecuTIC
Generating RSA private key, 2048 bit long modulus (2 primes)
.....+++++
.....+++++
e is 65537 (0x010001)
writing RSA key
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 24221 100 24084 100 137 35056 199 --:--:-- --:--:-- --:--:-- 35256
Longueur message : 30
127.0.0.1 - - [05/May/2020 13:12:11] "POST /creation HTTP/1.1" 200 3
127.0.0.1 - - [05/May/2020 13:12:25] "GET /fond HTTP/1.1" 200 3255975
akkouche@akkouche-VivoBook-ASUSLaptop-X412UA:~/Documents/projet_Tic/projet_0/user$ python3 user.py
student 1) or employer 2) -----> 1
envoies des données 1) reçue de l' image 2) -----> 1
ok! envoies des données 1) reçue de l' image 2) -----> 2
la commande se lance ou pas ?
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
0 0 0 0 0 0 0 0 --:--:-- --:--:-- --:--:-- 0* Trying 127.0.0.1...
* TCP_NODELAY set
* Connected to localhost (127.0.0.1) port 8080 (#0)
> GET /fond HTTP/1.1
> Host: localhost:8080
> User-Agent: curl/7.58.0
> Accept: */*
>
* HTTP 1.0, assume close after body
< HTTP/1.0 200 OK
< Date: Tue, 05 May 2020 11:12:25 GMT
< Server: WSGIServer/0.2 CPython/3.6.9
< Content-Type: image/png
< Content-Length: 3255975
<
{ [32768 bytes data]
100 3179k 100 3179k 0 0 443M 0 --:--:-- --:--:-- --:--:-- 443M
* Closing connection 0
akkouche@akkouche-VivoBook-ASUSLaptop-X412UA:~/Documents/projet_Tic/projet_0/user$


akkouche@akkouche-VivoBook-ASUS
Fichier Édition Affichage Rechercher Terminal Aide
nom prénom : star platinium intitulé de la certification : SecuTIC
Using configuration from /usr/lib/ssl/openssl.cnf
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 5584 0 5493 100 91 3576 59 0:00:01 0:00:01 --:--:-- 3635
Using configuration from /usr/lib/ssl/openssl.cnf
starplatiniumSecuTIC1588669927
star platiniumSecuTIC
Generating RSA private key, 2048 bit long modulus (2 primes)
e is 65537 (0x010001)
writing RSA key
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 24221 100 24084 100 137 35056 199 --:--:-- --:--:-- --:--:-- 35256
Longueur message : 30
127.0.0.1 - - [05/May/2020 13:12:11] "POST /creation HTTP/1.1" 200 3
127.0.0.1 - - [05/May/2020 13:12:25] "GET /fond HTTP/1.1" 200 3255975
Using configuration from /usr/lib/ssl/openssl.cnf
le timestamp n'a pas été modifiée
la signature est bonne
0
True
127.0.0.1 - - [05/May/2020 13:12:41] "POST /verification HTTP/1.1" 200 3
100 3179k 100 3179k 0 0 443M 0 --:--:-- --:--:-- --:--:-- 443M
* Closing connection 0
akkouche@akkouche-VivoBook-ASUSLaptop-X412UA:~/Documents/projet_Tic/projet_0/user$ python3 user.py
student 1) or employer 2) -----> 2
vérification de l' image 1) quitter le serveur web 2) -----> 1
* Trying 127.0.0.1...
* TCP_NODELAY set
* Connected to localhost (127.0.0.1) port 8080 (#0)
> POST /verification HTTP/1.1
> Host: localhost:8080
> User-Agent: curl/7.58.0
> Accept: */*
> Content-Length: 3256173
> Content-Type: multipart/form-data; boundary=------------------------0cec4e39d7f9ba4b
> Expect: 100-continue
>
* Done waiting for 100-continue
* HTTP 1.0, assume close after body
< HTTP/1.0 200 OK
< Date: Tue, 05 May 2020 11:12:41 GMT
< Server: WSGIServer/0.2 CPython/3.6.9
< Content-Type: text/plain
< Content-Length: 3
<
* Closing connection 0
ok!vérification de l' image 1) quitter le serveur web 2) ----->


Activités
Terminal
mar. 13:13
akkouche@akkouche-VivoBook-ASUSLapto
Fichier Édition Affichage Rechercher Terminal Aide
nom prénon : star platinium intitulé de la certification : SecuTIC
Using configuration from /usr/lib/ssl/openssl.cnf
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 5584 0 5493 100 91 3576 59 0:00:01 0:00:01 --:--:-- 3635
Using configuration from /usr/lib/ssl/openssl.cnf
starplatiniumSecuTIC1588669927
star platiniumSecuTIC
Generating RSA private key, 2048 bit long modulus (2 primes)
.....+++++
.....+++++
e is 65537 (0x010001)
writing RSA key
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 24221 100 24084 100 137 35056 199 --:--:-- --:--:-- --:--:-- 35256
Longueur message : 30
127.0.0.1 - - [05/May/2020 13:12:11] "POST /creation HTTP/1.1" 200 3
127.0.0.1 - - [05/May/2020 13:12:25] "GET /fond HTTP/1.1" 200 3255975
Using configuration from /usr/lib/ssl/openssl.cnf
le timestamp n'a pas été modifiée
la signature est bonne
0
True
127.0.0.1 - - [05/May/2020 13:12:41] "POST /verification HTTP/1.1" 200 3
* Closing connection 0
akkouche@akkouche-VivoBook-ASUSLaptop-X412UA:~/Documents/projet_Tic/projet_0/user$ python3 user.py
student 1) or employer 2) -----> 2
vérification de l' image 1) quitter le serveur web 2) -----> 1
* Trying 127.0.0.1...
* TCP_NODELAY set
* Connected to localhost (127.0.0.1) port 8080 (#0)
> POST /verification HTTP/1.1
> Host: localhost:8080
> User-Agent: curl/7.58.0
> Accept: */*
> Content-Length: 3256173
> Content-Type: multipart/form-data; boundary=------------------------0cec4e39d7f9ba4b
> Expect: 100-continue
>
* Done waiting for 100-continue
* HTTP 1.0, assume close after body
< HTTP/1.0 200 OK
< Date: Tue, 05 May 2020 11:12:41 GMT
< Server: WSGIServer/0.2 CPython/3.6.9
< Content-Type: text/plain
< Content-Length: 3
<
* Closing connection 0
ok!vérification de l' image 1) quitter le serveur web 2) -----> 2
akkouche@akkouche-VivoBook-ASUSLaptop-X412UA:~/Documents/projet_Tic/projet_0/user$

6 Analyse de Risques

- Nous aurions pu restreindre l'accès au serveur frontal avec l'insertion d'un mot de passe par exemple. Mais le problème peut être bénin si l'on limite l'accès au serveur frontal. (De plus la création du certificat ECC se fait depuis le serveur frontal pour faciliter la démonstration,en pratique il est bien entendue préférable de faire cette étape au préalable et de réserver le serveur frontal pour la connexion DCP).

- Concernant la clef privée de chiffrement, une nouvelle clef est créée à chaque fois par l'AC. Durant la création de l'attestation, nous aurions pu créer préalablement la clef et la conserver dans l'AC mais le choix semble alors plus risqué.

- Nous aurions aussi pu avoir deux répertoires différents gérant l'étudiant et l'employeur. Le principale avantage est de pouvoir alors disposer de deux certificats différents : l'un autorisant la création et l'autre la vérification.

- L'un des principaux défauts de notre programme est que celui-ci ne gère pas les connexions multiples au serveur frontal. Ainsi c'est un euphémisme de dire que notre programme est sensible aux attaques par dénie de service et c'est à notre avis son principal défaut.

- Dans le programme, il faut impérativement créer les deux répertoires. De plus, ceux-ci doivent être dans le même répertoire car le chemin qui lie Webservice à user n'est pas absolue et dépend de la configuration des répertoires.

- Le serveur web peut être fermé par un simple ctrl + c, ce risque peut être facilement de niveau 4 et doit être impérativement traité.