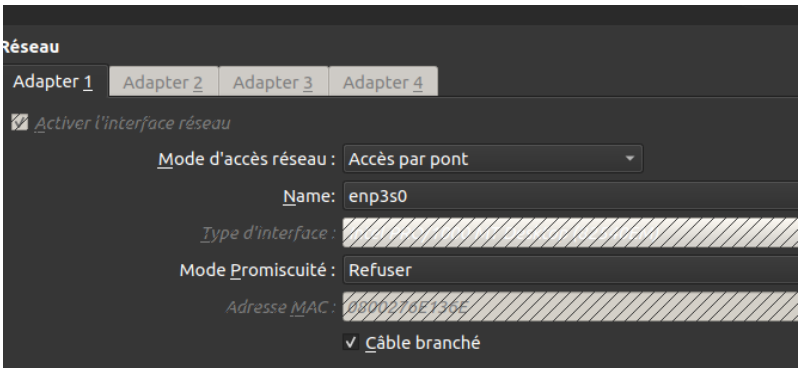
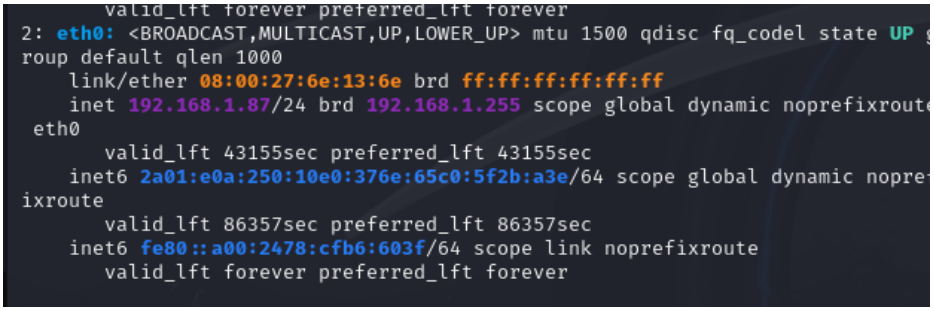
Abdel-malik FOFANA

Contrôle crypto avancé : Horodatage

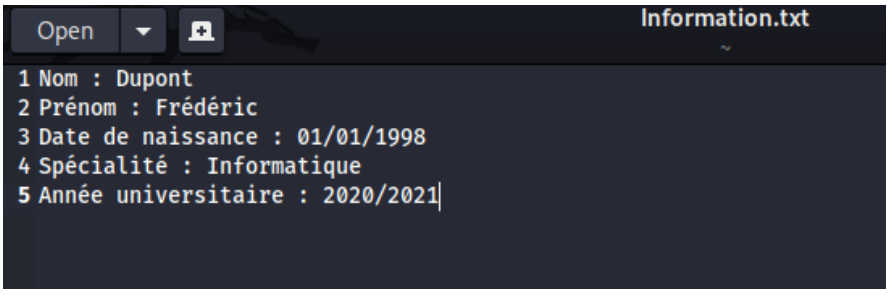
L'étape 1 : Signature du diplôme
D'abord on met notre VM en accès par pont



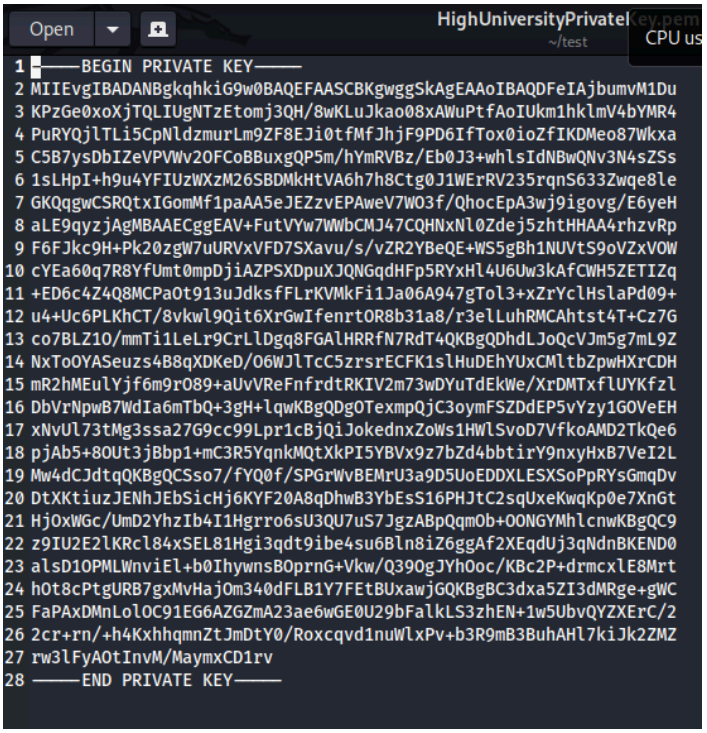
Voici l'IP donnée par le dhcp 192.168.1.87



On crée le fichier Information.txt

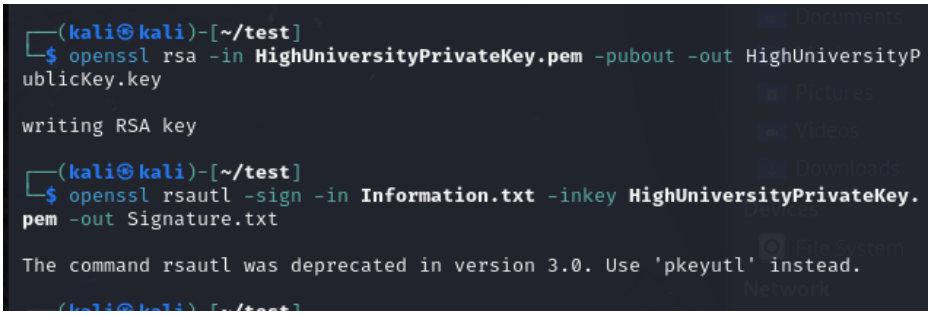


Et on crée une paire de clé privé/publig pour HighUniversity avec la commande
openssl genrsa -out HighUniversityPrivateKey.pem 2048



On extrait la clef publique et on la sauvegarde sous le nom : HighUniversityPublicKey.key avec la commande :
openssl rsa -in HighUniversityPrivateKey.pem -pubout -out HighUniversityPublicKey.key

et on signe le fichier `Information.txt` avec la clé privée :



L’étape 2 : Horodatage certifié
Ensuite on crée une requête d’horodatage avec la commande
openssl ts -query -data Signature.txt -sha256 -out Signature.txt.tsq
On envoie la requête au serveur www.freetsa.org et on récupére la réponse :
curl -H "Content-Type: application/timestamp-query" --data-binary '@Signature.txt.tsq' https://freetsa.org/tsr > Signature.tsr
Afficher la date de la signature avec la commande suivante:
#Openssl ts –reply –in Signature.tsr -text

```
┌──(kali㉿kali)-[~/test]
└─$ openssl ts -query -data Signature.txt -sha256 -out Signature.txt.tsq

Using configuration from /usr/lib/ssl/openssl.cnf

┌──(kali㉿kali)-[~/test]
└─$ curl -H "Content-Type: application/timestamp-query" --data-binary '@Signa
ture.txt.tsq' https://freetsa.org/tsr > Signature.tsr

  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Curre
nt
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--
100  1430    0  1364  100    66   2401    116 --:--:-- --:--:-- --:--:--  251
7

┌──(kali㉿kali)-[~/test]
└─$ openssl ts -reply -in Signature.tsr -text

Using configuration from /usr/lib/ssl/openssl.cnf
Status info:
Status: Granted.
Status description: unspecified
Failure info: unspecified

TST info:
Version: 1
Policy OID: tsa_policy1
Hash Algorithm: sha256
Message data:
    0000 - 61 7b aa 07 99 fc 24 e4-e4 ea 41 8e 05 8e cc 06   a{....$...A.....
    0010 - 59 a1 91 4d aa 72 26 b5-42 26 2e 01 49 92 d5 b2   Y..M.r&.B&..I...
Serial number: 0x053FF302
Time stamp: Jan 30 14:04:09 2025 GMT
Accuracy: unspecified
Ordering: yes
Nonce: 0x05EA96F1E19AACB7
TSA: DirName:/O=Free TSA/OU=TSA/description=This certificate digitally signs
documents and time stamp requests made using the freetsa.org online services/
CN=www.freetsa.org/emailAddress=busilezas@gmail.com/L=Wuerzburg/C=DE/ST=Bayer
n
Extensions:

┌──(kali㉿kali)-[~/test]
└─$
```

L'étape 3 : Ajout de la signature et la preuve d'horodatage à une image

Cacher le fichier info.zip dans l'image diplmo.jpg avec la commande :
#steghide embed -cf Diplome.jpg –ef info.zip

```
┌──(kali㉿kali)-[~/test]
└─$ zip info.zip Signature.txt Signature.txt.tsq Signature.tsr

updating: Signature.txt (stored 0%)
updating: Signature.txt.tsq (stored 0%)
updating: Signature.tsr (deflated 16%)

┌──(kali㉿kali)-[~/test]
└─$ steghide embed -cf Diplome.jpg -ef info.zip

Enter passphrase:
Re-Enter passphrase:
embedding "info.zip" in "Diplome.jpg"... done

┌──(kali㉿kali)-[~/test]
└─$
```

Et on peut voir que l'on a bien du contenue caché dans l'image

```
┌──(kali㉿kali)-[~/test]
└─$ steghide info Diplome.jpg
"Diplome.jpg":
  format: jpeg
  capacity: 3.7 KB
Try to get information about embedded data ? (y/n) y
Enter passphrase:
  embedded file "info.zip":
    size: 1.9 KB
    encrypted: rijndael-128, cbc
    compressed: yes
```

L'étape 4 : Vérification de la date de la signature

On a telecharger les certificat venant de freetsa.org
wget https://freetsa.org/files/cacert.pem
wget https://freetsa.org/files/tsa.crt

Puis on extrait les données cachées de l'image Diplome.jpg , décompresse le fichier info.zip , et on vérifie l'horodatage avec OpenSSL

```
┌──(kali㉿kali)-[~/test/New Folder]
└─$ steghide extract -sf Diplome.jpg

Enter passphrase:
wrote extracted data to "info.zip".

┌──(kali㉿kali)-[~/test/New Folder]
└─$ unzip info.zip

Archive:  info.zip
 extracting: Signature.txt
 extracting: Signature.txt.tsq
  inflating: Signature.tsr

┌──(kali㉿kali)-[~/test/New Folder]
└─$ openssl ts -verify -in Signature.tsr -queryfile Signature.txt.tsq -CAfile cacert.pem -untrusted tsa.crt

Using configuration from /usr/lib/ssl/openssl.cnf
Warning: certificate from 'tsa.crt' with subject '/O=Free TSA/OU=TSA/description=This certificate digitally signs documents and time stamp
ail.com/L=Wuerzburg/C=DE/ST=Bayern' is not a CA cert
Verification: OK

┌──(kali㉿kali)-[~/test/New Folder]
└─$
```

L'étape 5 : vérification de la signature de HighUniversity
Enfin on vérifiee la signature avec la clé publique avec la commande
`openssl rsautl -verify -in Signature.txt -pubin -inkey HighUniversityPublicKey.key -out hashSignature.txt`

Et comme on peut voir hashSignature.txt est bien pareil que Information.txt

```
kali@kali: ~/test
File  Actions  Edit  View  Help
┌──(kali㉿kali)-[~/test]
└─$ openssl rsautl -verify -in Signature.txt -pubin -inkey HighUniversityPublicKey.key -out hashSignature.txt

The command rsautl was deprecated in version 3.0. Use 'pkeyutl' instead.
```

Open | Infor... ~/test | Save

```
1 Nom : Dupont
2 Prénom : Frédéric
3 Date de naissance : 01/01/1998
4 Spécialité : Informatique
5 Année universitaire : 2020/2021
```

Open | hashSignature.txt ~/test

```
1 Nom : Dupont
2 Prénom : Frédéric
3 Date de naissance : 01/01/1998
4 Spécialité : Informatique
5 Année universitaire : 2020/2021
```