



Polytech Dijon

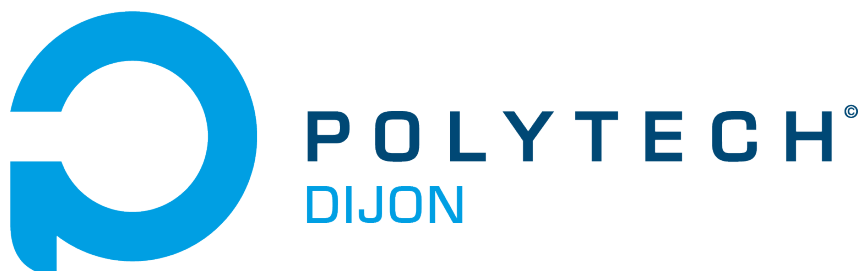
FISA IOT 5A

Programmation pour l'embarquée - TP

Aspect réseau et sécurité

Auteur :
CAMUS Pierre-Marie

Tuteur :
BARD Patrick



2024-2025

Sommaire

1	Compte Rendu	4
1.1	Introduction	4
1.2	Commandes	4
1.2.1	Adresse IP	4
1.2.2	Connexion SSH	4
1.2.3	Transfert de fichier	4
1.2.4	Programmation	5
1.2.5	Analyse	5
1.2.6	Commandes	6
1.2.7	Programmation	6
1.2.8	Analyse	8
1.3	TD RUST	8
1.4	Conclusion	10

Table des Figures

1.1	Paramètre réseau	4
1.2	Serveur.py	6
1.3	client.py	6
1.4	Wireshark réseau privé	6
1.5	Wireshark réseau privé2	7
1.6	Wireshark chiffré	8

1 Compte Rendu

1.1 Introduction

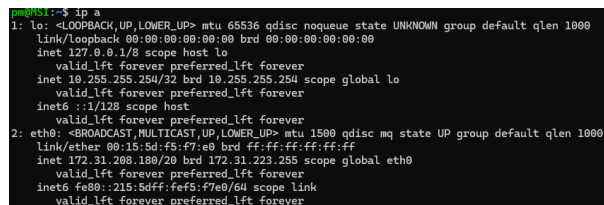
Le but de ce T.P. est de reviser les aspects réseau et sécurité avec des commandes de base

1.2 Commandes

1.2.1 Adresse IP

Avec la commande ip on obtient les paramètres des cartes réseau présent sur l'ordinateur.

```
1 ip a
```



```
root@kali:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet 10.255.255.254/32 brd 10.255.255.254 scope global lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:15:5d:f5:f7:e9 brd ff:ff:ff:ff:ff:ff
    inet 172.31.288.180/20 brd 172.31.223.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::215:5dff:fef5:f7e9/64 scope link
        valid_lft forever preferred_lft forever
```

FIGURE 1.1 – Paramètre réseau

1.2.2 Connexion SSH

Pour l'utilisation de la commande SSH il faut stipuler un utilisateur et l'adresse ip du PC a atteindre (exemple : root et 192.168.0.111)

```
1 ssh root@192.168.0.111
```

1.2.3 Transfert de fichier

Pour l'utilisation de la commande SCP il faut stipuler l'utilisateur et l'adresse ip du PC plus le chemin ou on veut copier les fichiers

```
1 scp root@192.168.0.111:/usr/
```

1.2.4 Programmation

Création du code en python coté serveur pour afficher les infos envoyé par le client. Ne disposant pas de deux machines, ici on utilise le localhost avec le port 65432 qui est dit "Dynamique et éphémères".

```
1 import socket
2
3 # Paramètres de connexion
4 HOST = "127.0.0.1"    # Localhost
5 PORT = 65432          # Port d'écoute (au choix, >1024)
6
7 # Création du socket serveur
8 with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as serveur:
9     serveur.bind((HOST, PORT))
10    serveur.listen(1)
11    print(f"Serveur en attente de connexion sur {HOST}:{PORT}...")
12
13    conn, addr = serveur.accept()
14    with conn:
15        print(f"Connecté par {addr}")
16        while True:
17            data = conn.recv(1024) # Réception de données (max 1024 octets)
18            if not data: # Si plus de données, on arrête
19                break
20            print("Reçu :", data.decode("utf-8"))
21
```

Lancement du code dans un terminal

```
1 py serveur.py
```

A l'inverse Creation du code en python coté client pour envoyer les infos au serveur

```
1 import socket
2
3 # Paramètres du serveur
4 HOST = "127.0.0.1"
5 PORT = 65432
6
7 with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as client:
8     client.connect((HOST, PORT))
9     print("Connecté au serveur. Tapez vos messages (q pour quitter) :")
10    while True:
11        msg = input(">>> ")
12        if msg.lower() == "q": # Quitter
13            break
14        client.sendall(msg.encode("utf-8"))
15
16
```

1.2.5 Analyse

En utilisant la loopback wireshark ne voit pas le reseau lo.

```

Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Installez la dernière version de PowerShell pour de nouvelles fonctionnalités
s

PS C:\Users\eltou\Documents\ProgEmbarque\TD3_Revison_Reseau> py .\serveur.py
Serveur en attente de connexion sur 127.0.0.1:65432...
Connecté par ('127.0.0.1', 51313)
Reçu : hello world

```

FIGURE 1.2 – Serveur.py

```

Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Installez la dernière version de PowerShell pour de nouvelles fonctionnalités et
s

PS C:\Users\eltou\Documents\ProgEmbarque\TD3_Revison_Reseau> py .\client.py
Connecté au serveur. Tapez vos messages (q pour quitter) :
>>> hello world
>>>

```

FIGURE 1.3 – client.py

Du coup changement de technique. On crée avec E.HELLES un réseau privé en utilisant la partage de connexion d'un meme telephone portable. Je fais le serveur et lui le client. Sur les images ci dessous on voit les echanges coté terminal et coté Wireshark. Avec le message en clair

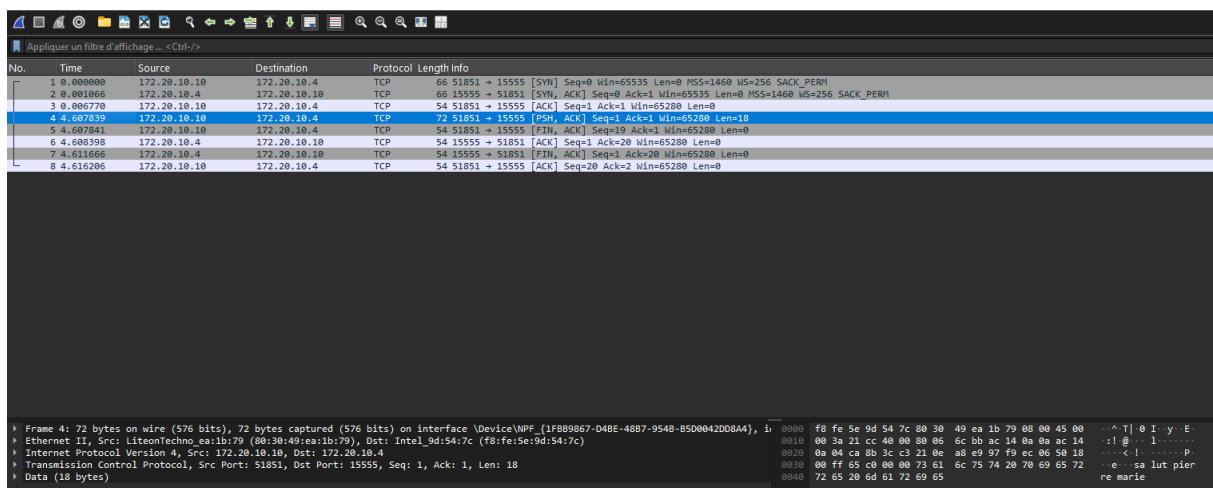


FIGURE 1.4 – Wireshark réseau privé

1.2.6 Commandes

Commande openssl pour la génération de clé RSA classique privé 2048bits.

- 1 `openssl genpkey -algorithm RSA -out private.pem -pkeyopt rsa_keygen_bits:2048`

Commande openssl pour la extraite la clé RSA publique.

- 1 `openssl rsa -pubout -in private.pem -out public.pem`

1.2.7 Programmation

On garde la même configuration, je transfère ma clé publique issue de ma clé privée à E.HELLE pour qu'il puisse chiffrer les données et que je sois le seul à déchiffrer le message.

- 1 `import socket`
- 2 `from cryptography.hazmat.primitives.asymmetric import padding`

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SERIAL MONITOR

PS C:\Users\eltou\Documents\ProgEmbarque\TD3_Revision_Reseau> py .\serveur.py
data = conn.recv(1024) # Réception de données (max 1024 octets)
ConnectionResetError: [WinError 10054] Une connexion existante a dû être fermée par l'hôte distant
PS C:\Users\eltou\Documents\ProgEmbarque\TD3_Revision_Reseau> py .\serveur.py
Serveur en attente de connexion sur 172.20.10.4:15555...
Connecté par ('172.20.10.10', 51836)
Traceback (most recent call last):
  File "C:\Users\eltou\Documents\ProgEmbarque\TD3_Revision_Reseau\serveur.py", line 17, in <module>
    data = conn.recv(1024) # Réception de données (max 1024 octets)
ConnectionResetError: [WinError 10054] Une connexion existante a dû être fermée par l'hôte distant
PS C:\Users\eltou\Documents\ProgEmbarque\TD3_Revision_Reseau> py .\serveur.py
Serveur en attente de connexion sur 172.20.10.4:15555...
Connecté par ('172.20.10.10', 51841)
Reçu : bjr
PS C:\Users\eltou\Documents\ProgEmbarque\TD3_Revision_Reseau> py .\serveur.py
Serveur en attente de connexion sur 172.20.10.4:15555...
Connecté par ('172.20.10.10', 51851)
Reçu : salut pierre marie
PS C:\Users\eltou\Documents\ProgEmbarque\TD3_Revision_Reseau>

```

FIGURE 1.5 – Wireshark réseau privé2

```

3  from cryptography.hazmat.primitives import serialization, hashes
4
5  HOST = "172.20.10.4"
6  PORT = 15555
7
8  # Charger la clé privée depuis un fichier PEM
9  with open("privatePM.pem", "rb") as f:
10     private_key = serialization.load_pem_private_key(f.read(), password=None)
11     print("Clé privée chargée")
12
13  with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as serveur:
14     serveur.bind((HOST, PORT))
15     serveur.listen(1)
16     print(f"Serveur en attente de connexion sur {HOST}:{PORT}...")
17
18     conn, addr = serveur.accept()
19     with conn:
20         print(f"Connecté par {addr}")
21         while True:
22             data = conn.recv(256) # Limité à un bloc RSA
23             if not data:
24                 break
25
26             try:
27                 # Déchiffrement RSA
28                 message = private_key.decrypt(
29                     data,
30                     padding.OAEP(
31                         mgf=padding.MGF1(algorithm=hashes.SHA256()),
32                         algorithm=hashes.SHA256(),
33                         label=None
34                     )
35                 )

```



```
12
13     return 4 * inside_circle / nb_points
14
15 # Exemple avec 1 million de points
16 print("Approximation de pi :", approx_pi(1_000_000))
```

[Lien Github](#)

1.4 Conclusion

Ce TP a permis de se refamiliariser avec les commandes de bases et l'aspect sécurité des messages