מטלה 3

סטודנט ראשון : אילן מאיר סופיר, ת"ז : **342615648**

סטודנט שני : בן כהן , ת"ז : **207029786**

<u>In our ZIP folder we have :</u>

- the python code : Web_Server.py

- the html file : HelloWorld.html

- Two wireshark captures :

-> The first one (capture1.pcapng) which contains the correct capture when we write : http://192.168.1.35:6789/HelloWorld.html

-> The seconde (capture1.pcapng) which contains the wrong capture when we write : http://192.168.1.35:6789/aaa.html

<u>How the web server works with our python code ?</u>

We need 2 computers connected to the same network, wireshark, and a web browser (google chrome for us).

The adress of the <u>first computer</u> is 192.168.1.35 which is the computer where i run the python code.

The adress of the <u>second computer</u> is 192.168.1.38 which is the computer where i need to write the URL on the web browser.

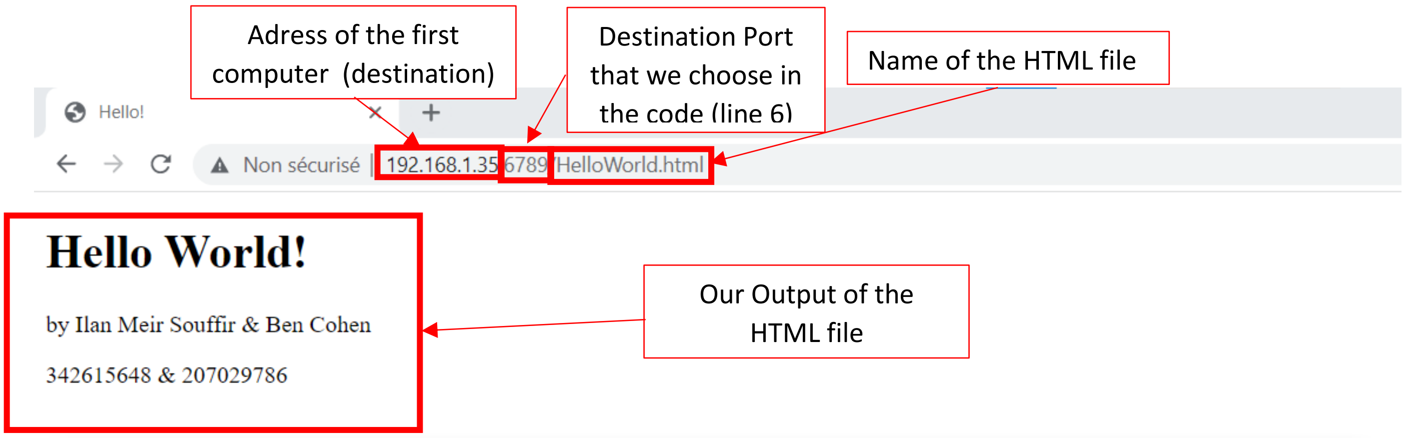
<u>Steps for the First manipulation :</u>

1. Open Wireshark and start the capture on the first computer.

2. Run the server python code (Web_Server.py) on the first computer.

3. Open google Chrome on the second computer and write the URL : http://192.168.1.35:6789/HelloWorld.html

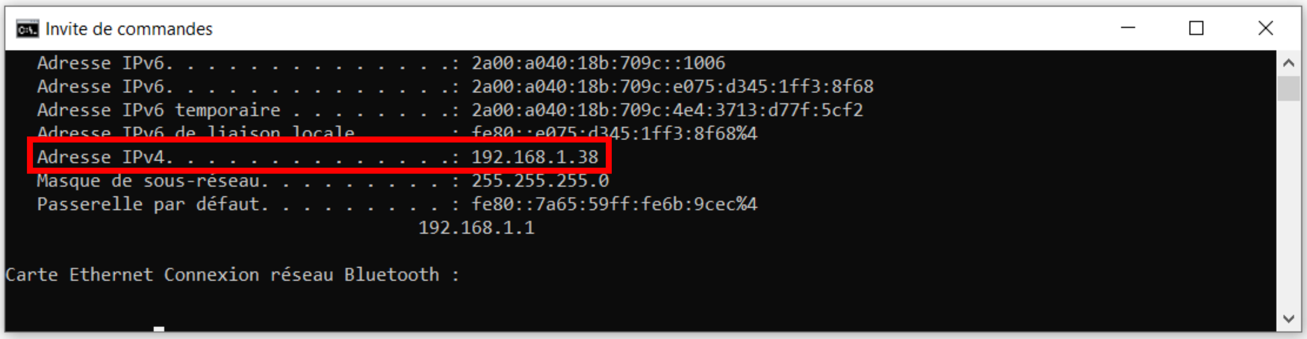
4. We can see that we have this page which is displayed :


Adress of the first computer (destination)
Destination Port that we choose in the code (line 6)
Name of the HTML file
Our Output of the HTML file
Hello!
Non sécurisé
192.168.1.35:6789/HelloWorld.html

Hello World!

by Ilan Meir Souffir & Ben Cohen

342615648 & 207029786



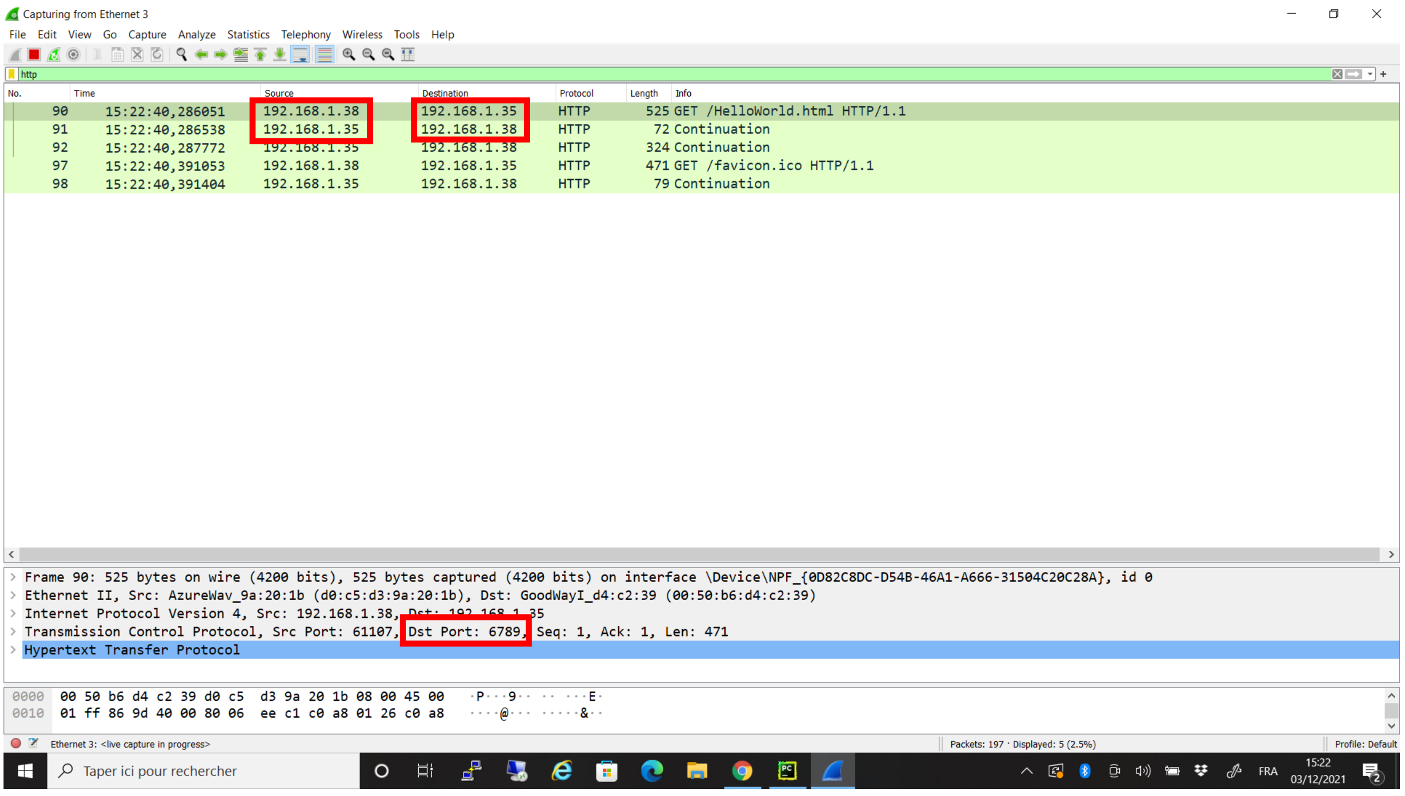
```
   Adresse IPv6. . . . . . . . . . . . . . .: 2a00:a040:18b:709c::1006
   Adresse IPv6. . . . . . . . . . . . . . .: 2a00:a040:18b:709c:e075:d345:1ff3:8f68
   Adresse IPv6 temporaire . . . . . . . . .: 2a00:a040:18b:709c:4e4:3713:d77f:5cf2
   Adresse IPv6 de liaison locale. . . . . .: fe80::e075:d345:1ff3:8f68%4
   Adresse IPv4. . . . . . . . . . . . . . .: 192.168.1.38
   Masque de sous-réseau. . . . . . . . . . : 255.255.255.0
   Passerelle par défaut. . . . . . . . . . : fe80::7a65:59ff:fe6b:9cec%4
                                       192.168.1.1

Carte Ethernet Connexion réseau Bluetooth :
```

5. Stop Web_Server.py

6. Stop the Wireshark capture.

We can see on Wireshark that we have the GET from 192.168.1.38 to 192.168.1.35 and the answer with the opposite direction of the adresses :


Capturing from Ethernet 3
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http
Ethernet 3: <live capture in progress>
Packets: 197 · Displayed: 5 (2.5%)
Profile: Default
Taper ici pour rechercher
FRA
15:22
03/12/2021

```
No.   Time              Source          Destination     Protocol  Length Info
   90 15:22:40,286051   192.168.1.38    192.168.1.35    HTTP        525 GET /HelloWorld.html HTTP/1.1
   91 15:22:40,286538   192.168.1.35    192.168.1.38    HTTP         72 Continuation
   92 15:22:40,287772   192.168.1.35    192.168.1.38    HTTP        324 Continuation
   97 15:22:40,391053   192.168.1.38    192.168.1.35    HTTP        471 GET /favicon.ico HTTP/1.1
   98 15:22:40,391404   192.168.1.35    192.168.1.38    HTTP         79 Continuation

> Frame 90: 525 bytes on wire (4200 bits), 525 bytes captured (4200 bits) on interface \Device\NPF_{0D82C8DC-D54B-46A1-A666-31504C20C28A}, id 0
> Ethernet II, Src: AzureWav_9a:20:1b (d0:c5:d3:9a:20:1b), Dst: GoodWayI_d4:c2:39 (00:50:b6:d4:c2:39)
> Internet Protocol Version 4, Src: 192.168.1.38, Dst: 192.168.1.35
> Transmission Control Protocol, Src Port: 61107, Dst Port: 6789, Seq: 1, Ack: 1, Len: 471
> Hypertext Transfer Protocol

0000  00 50 b6 d4 c2 39 d0 c5  d3 9a 20 1b 08 00 45 00   ·P···9·· ·· ···E·
0010  01 ff 86 9d 40 00 80 06  ee c1 c0 a8 01 26 c0 a8   ····@··· ·····&··
```

Steps for the Second manipulation :

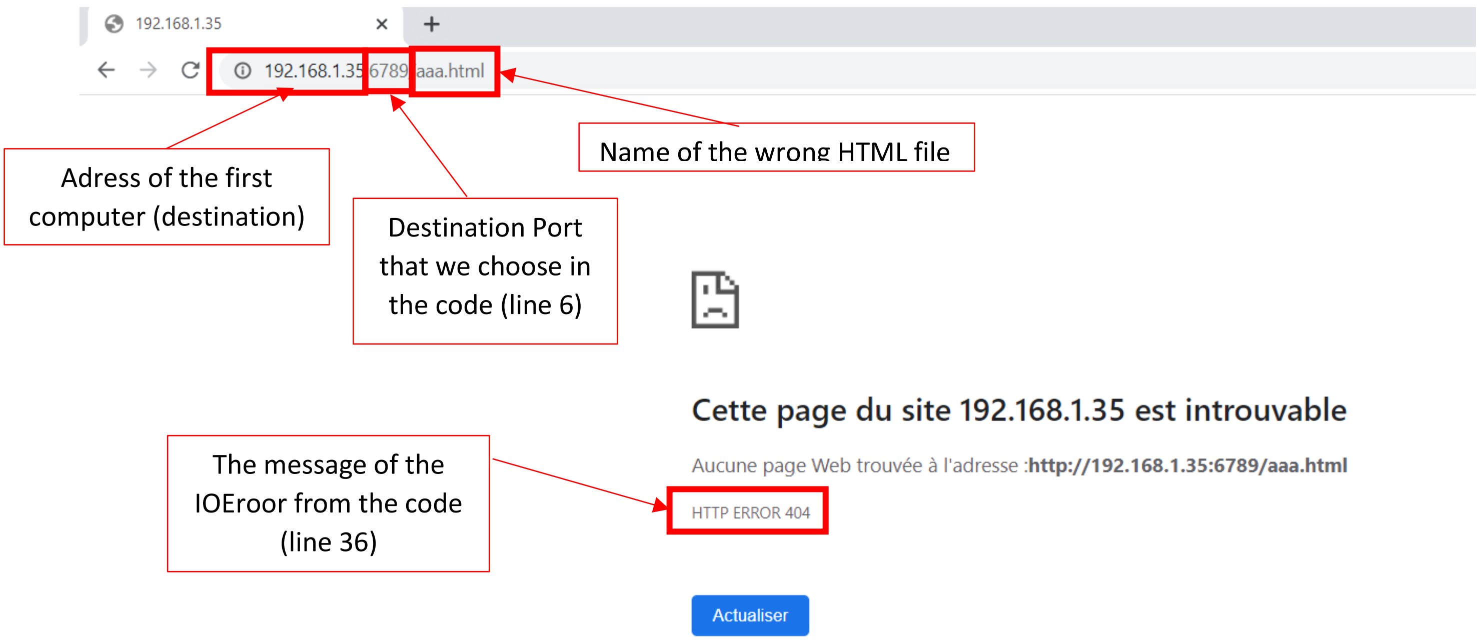
1. Open Wireshark and start the capture on the first computer.

2. Run the server python code (Web_Server.py) on the first computer.

3. Open google Chrome on the second computer and write the URL : http://192.168.1.35:6789/aaa.html

Wich is a html file that doesn't exist.

4. We can see that we have this page which is displayed :



192.168.1.35
192.168.1.35:6789/aaa.html
Adress of the first computer (destination)
Destination Port that we choose in the code (line 6)
Name of the wrong HTML file
The message of the IOEroor from the code (line 36)
Actualiser

Cette page du site 192.168.1.35 est introuvable

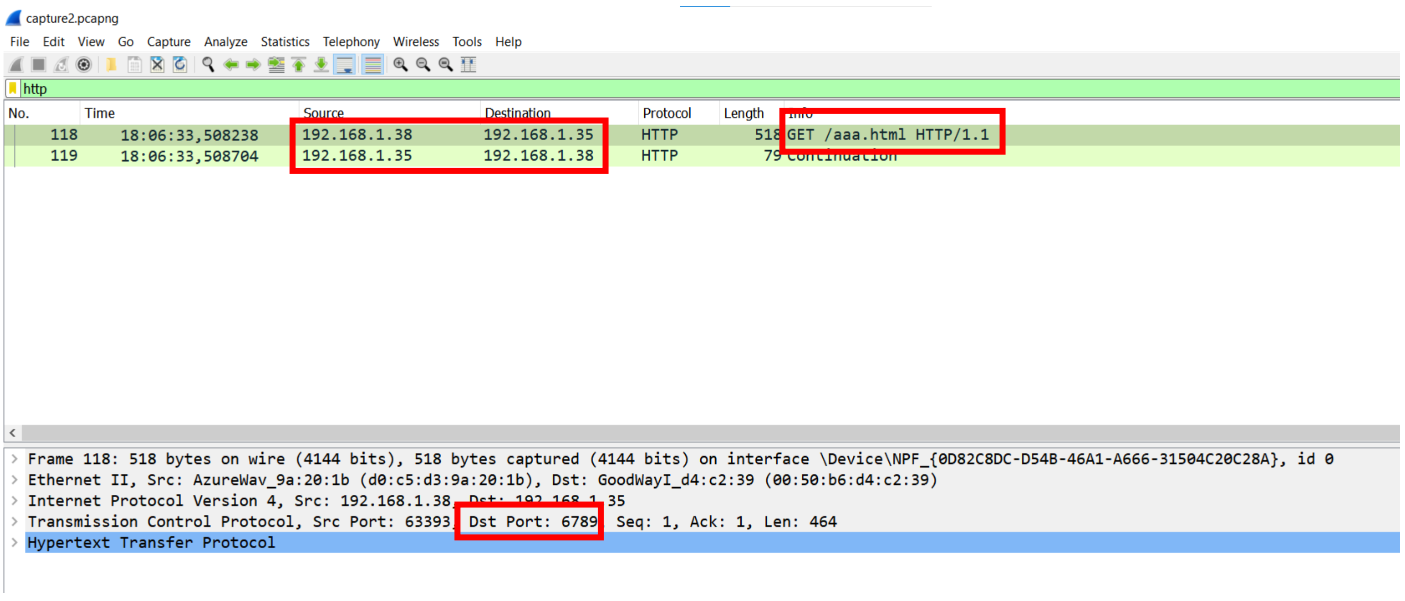
Aucune page Web trouvée à l'adresse :**http://192.168.1.35:6789/aaa.html**

HTTP ERROR 404

5. Stop Web_Server.py

6. Stop the Wireshark capture.

We can see on Wireshark that we have the GET from 192.168.1.38 to 192.168.1.35 with the wrong file, and the answer with the opposite direction of the adresses :



capture2.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http

```
No.   Time              Source          Destination     Protocol  Length  Info
  118   18:06:33,508238   192.168.1.38    192.168.1.35    HTTP        518  GET /aaa.html HTTP/1.1
  119   18:06:33,508704   192.168.1.35    192.168.1.38    HTTP         79  Continuation

> Frame 118: 518 bytes on wire (4144 bits), 518 bytes captured (4144 bits) on interface \Device\NPF_{0D82C8DC-D54B-46A1-A666-31504C20C28A}, id 0
> Ethernet II, Src: AzureWav_9a:20:1b (d0:c5:d3:9a:20:1b), Dst: GoodWayI_d4:c2:39 (00:50:b6:d4:c2:39)
> Internet Protocol Version 4, Src: 192.168.1.38, Dst: 192.168.1.35
> Transmission Control Protocol, Src Port: 63393, Dst Port: 6789, Seq: 1, Ack: 1, Len: 464
> Hypertext Transfer Protocol
```

How does the code run ?

1. we import the soket module that we need to create a soket.

2. creation of a soket with a port.
line 5 : `serverSocket = socket(AF_INET, SOCK_STREAM)`

3. Prepare a sever socket with port 6789.
lines 6 and 9 :
```
SERVER_ADDRESS = ('', 6789)
serverSocket.bind(SERVER_ADDRESS)
```
-> Listen to all IP’s on this computer with port destination 6789.

4. Server only handles one connection at a given time. ‘1’ is a number of incoming connections are waiting in the queue.
Line 10 : `serverSocket.listen(1)`

5. Then we etablish the connection and with an infinite loop we wait for client connection :
We come out of the loop with a end of the code.
This line 16 is for the waiting connection of the client :
```
connectionSocket, addr = serverSocket.accept()
```

6. Then if we have a correct HTML file, we try the lines 18 to 32 which will display the html file on the second computer.
```
message = connectionSocket.recv(4096).decode()
```
-> Reading houses from the socket
```
connectionSocket.send("\nHTTP/1.1 200 OK\n\n".encode())
```
-> Send one HTTP header line into socket

```
for i in range(0, len(outputdata)):
    connectionSocket.send(outputdata[i].encode())
connectionSocket.send("\r\n".encode())
```
-> Send the content of the requested file to the client
```
connectionSocket.close()
```
Closing the connection to a specific customer.
And then we are in the infinite loop so we repeat this step until the end of the run.

7. Then if we wrote a wrong HTML file in the seconde computer (in the search bar for URL)
We enter in the part of the code from line 33 to line 41.
And if we wrote a wrong html file (file note found), there is a message on the screen http ERROR 404 that corresspond with this line :
```
connectionSocket.send("\nHTTP/1.1 404 Not Found\n\n".encode())
```

Then we close the client socket :
```
connectionSocket.close()
```

8. We close the Server socket to end the infinite loop with :
```
serverSocket.close()
```

9. At the End, we terminate the program after sending the corresponding data with :
```
sys.exit()
```