

CSC 4200

PA3_README

Jonathan Leonard (jcleonard42)

Kevin Pathana (kspathana42)

Work Process Description:

We had chosen to write the server and client in different languages but managed to get them to communicate properly as specified in the assignment. Kevin Pathana had chosen to write the lightserver in C while Jonathan Leonard had chosen to write the lightclient program in Python. What we have is a simple socket server and client that communicate with each other flawlessly. We have included a Windows compilation for the server as it was first written for Windows, then ported over to Linux with the change of libraries and function calls. As such, the compilation command requires the `-lws2_32` tag for Windows while the Linux compilation does not. As for the Python client, it does not need to have any differences between the Windows and Linux version, as it works on both with no problem. We have tested them in the Google VM and it does indeed work with the Google VMs. We did have to install GCC and other GNU libraries through ``sudo apt-get install gcc`` because for some reason the Google VMs did not come preloaded with them.

Windows Compilation:

Server command:

Compilation: `gcc -o lightserver lightserver.c -lws2_32`

Execution: `.\lightserver <port number> .\<log file>`

Client Command:

`python lightclient.py -s <ip address> -p <port number> -l .\<log file>`
(or python3)

Linux Compilation:

Server command:

Compilation: `gcc -o lightserver lightserver.c`

Execution: `./lightserver <port number> ./<log file>`

Client Command:

python lightclient.py -s <ip address> -p <port number> -l .\<log file>
(or python3)

PCAP Capture:

We used termshark (terminal ui for tshark, which in turn uses TCPDUMP) to capture the data desired from client and server. (Screenshot is not accurate data, the .pcap files will have the data).

Google VMs (Ubuntu 23.04)

V1 = Server, V2 = Client

The image displays four screenshots of a terminal environment, likely a Google Cloud VM, showing SSH connections and network traffic capture using termshark.

The top-left screenshot shows the SSH-in-browser interface with the following output:

```
EXECUTING SUPPORTED COMMAND: LIGHTON
Connection from 10.128.0.3:54316 is closed
Received connection from 10.128.0.3:60818
Received Data: version: 17 type: 1 length: 5
Sent SUCCESS response
VERSION ACCEPTED
Received Data: version: 17 type: 2 length: 7
Sent SUCCESS response
EXECUTING SUPPORTED COMMAND: LIGHTON
Connection from 10.128.0.3:60818 is closed
Received connection from 10.128.0.3:60830
Received Data: version: 17 type: 1 length: 5
Sent SUCCESS response
VERSION ACCEPTED
Received Data: version: 17 type: 2 length: 7
Sent SUCCESS response
EXECUTING SUPPORTED COMMAND: LIGHTON
Connection from 10.128.0.3:60830 is closed
```

The top-right screenshot shows the termshark 2.2.0 interface with the filter `ip.addr == 10.128.0.3` applied. The captured traffic is as follows:

No.	Time	Source	Destination	Protocol	Length	Info
3704	120.532	10.128.0.3	10.128.0.2	TCP	66	60830 → 8080 [FIN, ACK] Seq=41 A
3705	120.532	10.128.0.2	10.128.0.3	TCP	66	8080 → 60830 [FIN, ACK] Seq=41 A
3707	120.532	10.128.0.3	10.128.0.2	TCP	66	60830 → 8080 [ACK] Seq=42 Ack=42

The bottom-left screenshot shows the SSH-in-browser interface with the following output:

```
2023-10-20 22:40:30,085 Received Data: version: 17 type: 1 length: 7
2023-10-20 22:40:30 VERSION ACCEPTED
2023-10-20 22:40:30 Received Message SUCCESS
2023-10-20 22:40:30,086 Sending command
2023-10-20 22:40:30,127 Received Data: version: 17 type: 1 length: 7
2023-10-20 22:40:30,127 VERSION ACCEPTED
2023-10-20 22:40:30,127 Received Message SUCCESS
Closing socket
kpathana3495@v2:~$ python3 ./lightclient.py -s 10.128.0.2 -p 8080 -l ./clientlog.txt
2023-10-20 22:40:32,438 Sending HELLO Packet
2023-10-20 22:40:32,438 Received Data: version: 17 type: 1 length: 7
2023-10-20 22:40:32 VERSION ACCEPTED
2023-10-20 22:40:32 Received Message SUCCESS
2023-10-20 22:40:32,439 Sending command
2023-10-20 22:40:32,483 Received Data: version: 17 type: 1 length: 7
2023-10-20 22:40:32,483 VERSION ACCEPTED
2023-10-20 22:40:32,483 Received Message SUCCESS
Closing socket
kpathana3495@v2:~$
```

The bottom-right screenshot shows the termshark 2.2.0 interface with the filter `ip.addr == 10.128.0.2` applied. The captured traffic is as follows:

No.	Time	Source	Destination	Protocol	Length	Info
3704	120.532	10.128.0.3	10.128.0.2	TCP	66	60830 → 8080 [FIN, ACK] Seq=41 A
3705	120.532	10.128.0.2	10.128.0.3	TCP	66	8080 → 60830 [FIN, ACK] Seq=41 A
3707	120.532	10.128.0.3	10.128.0.2	TCP	66	60830 → 8080 [ACK] Seq=42 Ack=42