

Network Security

Lab 1 – Communicating with a monitor

(Anuj Shah, Chakresh Singh, Naveen Kumar Aleti, Tarun Kumar Kurra and Sahithya Ghattmaraju)

Objective:

The purpose of this assignment is to communicate with the monitor running on helios.ececs.uc.edu from our local machine.

Procedure:

To establish the communication with the monitor, we have made use of Java programming. Most of the source code is being used as is shared by Dr. Franco except for few modifications of Login() method in MessageParser.java.

Below are the steps that has been followed:

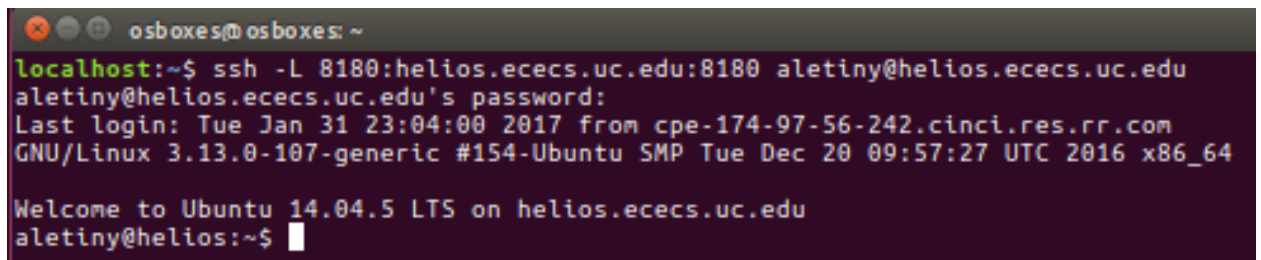
1. Establish tunnel from Localhost to Helios.

We cannot directly connect to the monitor running on port 8180 on helios.ececs.uc.edu.

To establish this connectivity an SSH tunnel must be opened from localhost to helios.

This is achieved by executing the below command in the shell:

```
$ ssh -L 8180:helios.ececs.uc.edu:8180 aletiny@helios.ececs.uc.edu
```

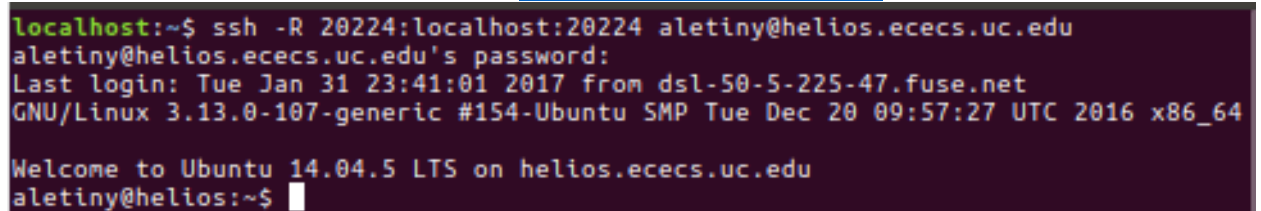


```
osboxes@osboxes: ~  
localhost:~$ ssh -L 8180:helios.ececs.uc.edu:8180 aletiny@helios.ececs.uc.edu  
aletiny@helios.ececs.uc.edu's password:  
Last login: Tue Jan 31 23:04:00 2017 from cpe-174-97-56-242.cinci.res.rr.com  
GNU/Linux 3.13.0-107-generic #154-Ubuntu SMP Tue Dec 20 09:57:27 UTC 2016 x86_64  
Welcome to Ubuntu 14.04.5 LTS on helios.ececs.uc.edu  
aletiny@helios:~$
```

2. Establish tunnel between participant's server and Helios:

The monitor will not be able to connect to our server running on port 20224 (Fixed port configured in Homework.java). Hence the below command is executed from a new terminal to establish a tunnel between Participant's server and the monitor.

```
$ ssh -R 20224:localhost:20224 aletiny@helios.ececs.uc.edu
```



```
localhost:~$ ssh -R 20224:localhost:20224 aletiny@helios.ececs.uc.edu  
aletiny@helios.ececs.uc.edu's password:  
Last login: Tue Jan 31 23:41:01 2017 from dsl-50-5-225-47.fuse.net  
GNU/Linux 3.13.0-107-generic #154-Ubuntu SMP Tue Dec 20 09:57:27 UTC 2016 x86_64  
Welcome to Ubuntu 14.04.5 LTS on helios.ececs.uc.edu  
aletiny@helios:~$
```

3. Changes to the sourcecode: There has been two major changes to Homework.java and MessageParser.java. The main method in Homework class has been modified to accept

“programMode” as the fourth input argument that will determine if the ActiveClient (when programMode=0) thread needs to be started or the passive server (when programMode=1).

The login() Method In MessageParser class has been updated to automatically execute RegistrationTransaction and ARE_YOU_ALIVE transaction upon launch. In addition to this, if the program is launched in ActiveClient mode (programMode = 0) then an interactive shell session is provided for the user to enter any commands.

Refer to the java files for the source code, submitted along with the report.

4. Run Passive server in background: The passive server is launched as a background process using the below command. The output will be redirected to nohup.out
\$ nohup java Homework localhost 8180 SYNERGY 1 &
nohup will run the program in the background and will return a process ID that can be managed separately from other shells.

```
localhost:~/Downloads$ nohup java Homework localhost 8180 SYNERGY 1 &
[1] 30522
localhost:~/Downloads$ nohup: ignoring input and appending output to 'nohup.out'
```

```
localhost:~/Downloads$ tail -f nohup.out
Project Begin:
  Monitor: localhost random port: 20224 monitor port: 8180
Starting login from Server..

  COMMENT: Monitor Version 2.2.1 PARTICIPANT_PASSWORD_CHECKSUM: 28604e8040a26f20ff842984a1d232f7997b5990 REQUIRE: IDENT WAITING:
expectedInput = IDENT
MessageParser [SendIt]: sent:
  IDENT SYNERGY
Test
Response from Monitor : REQUIRE: ALIVE WAITING:
expectedInput = ALIVE
MessageParser [SendIt]: sent:
  ALIVE K74USE233VSRH042PZU
Test
Response from Monitor : RESULT: ALIVE Identity has been verified. REQUIRE: QUIT WAITING:
expectedInput = QUIT
MessageParser [SendIt]: sent:
  QUIT
Test
Response from Monitor :
Outside while loop
1
Success Value Login = true
ConnectionHandler [run]: success Logged In!
```

5. Run ActiveClient: A Client session with the monitor can be launched by executing the below command.

\$ java Homework localhost 8180 SYNERGY 0

The client session can be launched and terminated any number of times without impacting the server.

```
^Clocalhost:~/Downloads$ java Homework localhost 8180 SYNERGY 0
Project Begin:
  Monitor: localhost random port: 20224 monitor port: 8180
Active Client: trying monitor: localhost port: 8180...completed.

      COMMENT: Monitor Version 2.2.1 REQUIRE: IDENT WAITING:
expectedInput = IDENT
MessageParser [SendIt]: sent:
      IDENT SYNERGY
Test
Response from Monitor : REQUIRE: ALIVE WAITING:
expectedInput = ALIVE
MessageParser [SendIt]: sent:
      ALIVE K74USE233V5RH042PZU
Test
Response from Monitor : RESULT: ALIVE Identity has been verified. WAITING:
Outside while loop
0
GET_GAME_IDENTS
MessageParser [SendIt]: sent:
      GET_GAME_IDENTS
Response from Monitor: RESULT: GET_GAME_IDENTS TESTINGACCOUNT4 VINJANDA KRAIGER TESTINGACCOUNT3 IDENT TESTINGACCOUNT2 TESTINGACCOUNT1 IRONMAN UNDEFINED TESTINGACCOUNT AVENGERS SYNERGY JASH STRANGERS TESTI
NGACCOUNT6 TESTINGACCOUNT5 WAITING:
█
```