COMP8006 Final Project Test Document

COMP8006
Jake Miner
Albert Liao
March 29th, 2014

Table of Contents

Test Bench Setup 2

Test Cases 3

Verification Screenshots 8

SSH port 22 to 192.168.0.19:1337 test..8
SSH port 9999 to 192.168.0.19:1337 test 10
Apache port 4321 to 192.168.0.19:80 test 12
Apache port 4321 to 192.168.0.19:8080 test..14
httperf test port 4321 to 192.168.0.19:8080 test..15
Assignment 2 test port 7777 to 192.168.0.19:7777 16
Assignment 2 test port 7777 to 192.168.0.19:7777 with SSH port 9999 to 192.168.0.19:1337 latency test. 16

Test Bench Setup

For our testing, we will be using the computers available in the Data Communications lab at BCIT. The workstations are equipped with the following:

- 8GB of RAM
- Intel Core i5-2400 @ 3.10GHz x 4
- Fedora 19 3.1.1 64 bit.

We will be using four workstations for this set of tests, each workstation will be designated as follows:

1. 192.168.0.18 will be the machine running the port forwarder
2. 192.168.0.19 will be the machine running the services (SSHD, HTTPD, Assign 2 server)
3. 192.168.0.20 will be the test client.
4. 192.168.0.22 will be the additional test client when we test SSH latency with traffic from 192.168.0.20 going through the port forwarder simultaneously.

Test Cases

Test Description	Setup	Testing Process	Validation	Pass?
SSH port 22 to 192.168.0.19:1337 test	1. Have 192.168.0.19 be the server running SSHD listening on port 1337 and start the SSHD service 2. Setup the machine 192.168.0.18 to be the Port Forwarder and forward incoming TCP connections on port 22 to 192.168.0.19:1337	1. Log into another lab computer that isn't the port forwarder or SSHD server. 2. Attempt to SSH into the SSHD server through the Port Forwarder by using the command: "ssh 192.168.0.18". Follow the prompts to login. 3. Run the "ifconfig" command to verify we are logged into the SSHD Server and not the Port Forwarder.	See validation section below for screenshots.	PASS

SSH port 9999 to 192.168.0.19:1337 test	1. Have 192.168.0.19 be the server running SSHD listening on port 1337 and start the SSHD service 2. Setup the machine 192.168.0.18 to be the Port Forwarder and forward incoming TCP connections on port 9999 to 192.168.0.19:1337	1. Log into another lab computer that isn't the port forwarder or SSHD server. 2. Attempt to SSH into the SSHD server through the Port Forwarder by using the command: "ssh –p 9999 192.168.0.18". Follow the prompts to login. 3. Run the "ifconfig" command to verify we are logged into the SSHD Server and not the Port Forwarder.	See validation section below for screenshots.	PASS
Apache port 4321 to 192.168.0.19:80 test	1. Have 192.168.0.19 be the Apache server running on port 80. Ensure that the service is running. 2. Set the homepage to display the text "You are on machine 192.168.0.19". 3. Setup the machine 192.168.0.18 to be the Port Forwarder and forward incoming TCP connections on port 4321 to 192.168.0.19:80.	1. Log into another lab computer that isn't the port forwarder or the Apache server. 2. Open a web browser and navigate to the URL "192.168.0.18:4321" 3. Verify that the homepage is the one you set for the Apache machine.	See validation section below for screenshots.	PASS

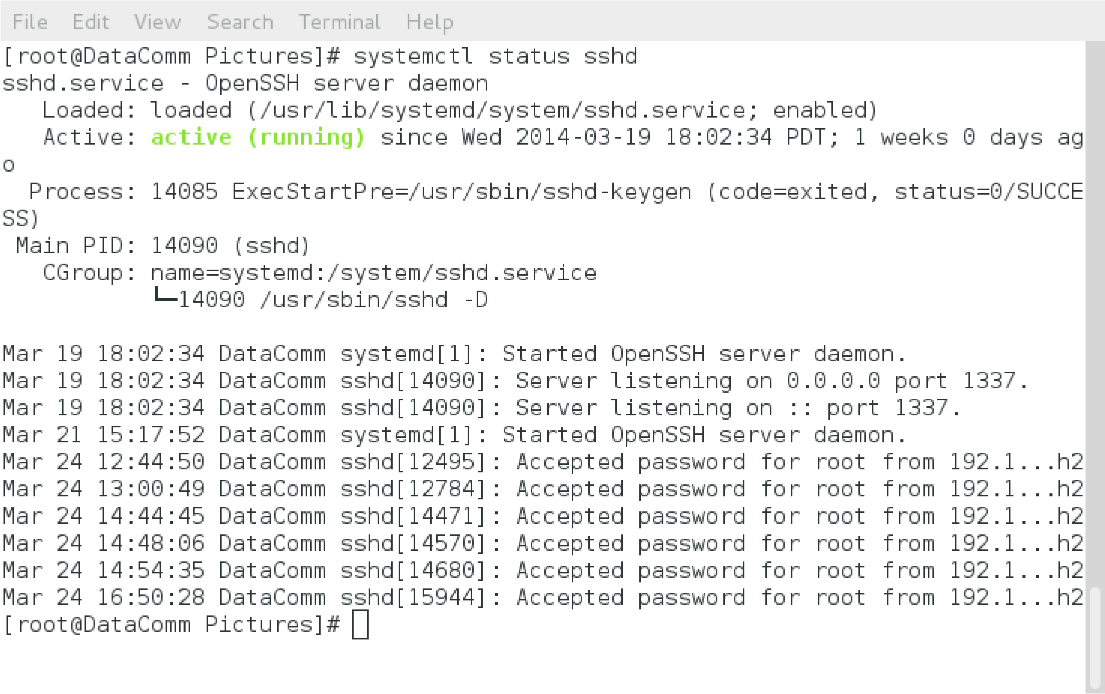
Apache port 4321 to 192.168.0.19:8080 test	1. Have 192.168.0.19 be the Apache server running on port 8080. Ensure that the service is running. 2. Set the homepage to display the text “You are on machine 192.168.0.19”. 3. Setup the machine 192.168.0.18 to be the Port Forwarder and forward incoming TCP connections on port 4321 to 192.168.0.19:8080.	1. Log into another lab computer that isn’t the port forwarder or the Apache server. 2. Open a web browser and navigate to the URL “192.168.0.18:4321” 3. Verify that the homepage is the one you set for the Apache machine.	See validation section below for screenshots.	PASS
httperf test port 4321 to 192.168.0.19:8080 test	1. Have 192.168.0.19 be the Apache server running on port 8080. Ensure that the service is running. 2. Set the homepage to display the text “You are on machine 192.168.0.19”. 3. Setup the machine 192.168.0.18 to be the Port Forwarder and forward incoming TCP connections on port 4321 to 192.168.0.19:8080.	1. Log into another lab computer that isn’t the port forwarder or the Apache server. 2. Open a console and run httperf with the following command: “httperf –-server 192.168.0.18 –-port 4321 –-rate 30 –-num-conn 3000 -–timeout 5 3. Verify that all of the connections are successful and there are no timeouts.	See validation section below for screenshots.	PASS

Assignment 2 test port 7777 to 192.168.0.19:7777	1. Have 192.168.0.19 be the assignment 2 server running on port 7777. Ensure that the service is running. 2. Setup the machine 192.168.0.18 to be the Port Forwarder and forward incoming TCP connections on port 7777 to 192.168.0.19:7777.	1. Log into another lab computer that isn't the port forwarder or the assignment 2 server. 2. Open a console and run assignment 2's client executable by using the command:"./test.sh 192.168.0.18 100 1000 135". This will cause the client machine to attempt to run 135 clients. 3. Verify on the server side that there was no data lost (data sent received should be 200000.	See validation section below for screenshots.	PASS

Assignment 2 test port 7777 to 192.168.0.19:7777 with SSH port 9999 to 192.168.0.19:1337 latency test.	1. Have 192.168.0.19 be the assignment 2 server running on port 7777 and the SSHD server listening on port 1337. Ensure that both services are running. 2. Setup the machine 192.168.0.18 to be the Port Forwarder and forward incoming TCP connections on port 7777 to 192.168.0.19:7777 and also port 9999 to 192.168.0.19:1337.	1. Log into another lab computer that isn't the port forwarder or the assignment 2 server. 2. Open a console and run assignment 2's client executable by using the command:"./test.sh 192.168.0.18 100 1000 135". This will cause the client machine to attempt to run 135 clients. 3. Open another console window and run the command "ssh 192.168.0.18 –p 9999" 4. Verify that there is no visible latency to the user by running the command "date && ls && date" and verifying that the timestamps are within 1 second of eachother.	See validation section below for screenshots.	PASS

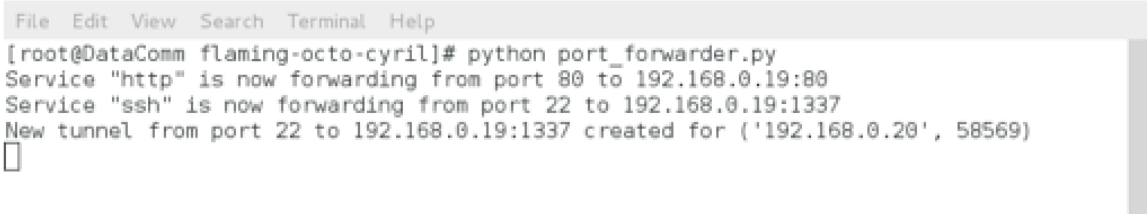
Verification Screenshots

SSH port 22 to 192.168.0.19:1337 test



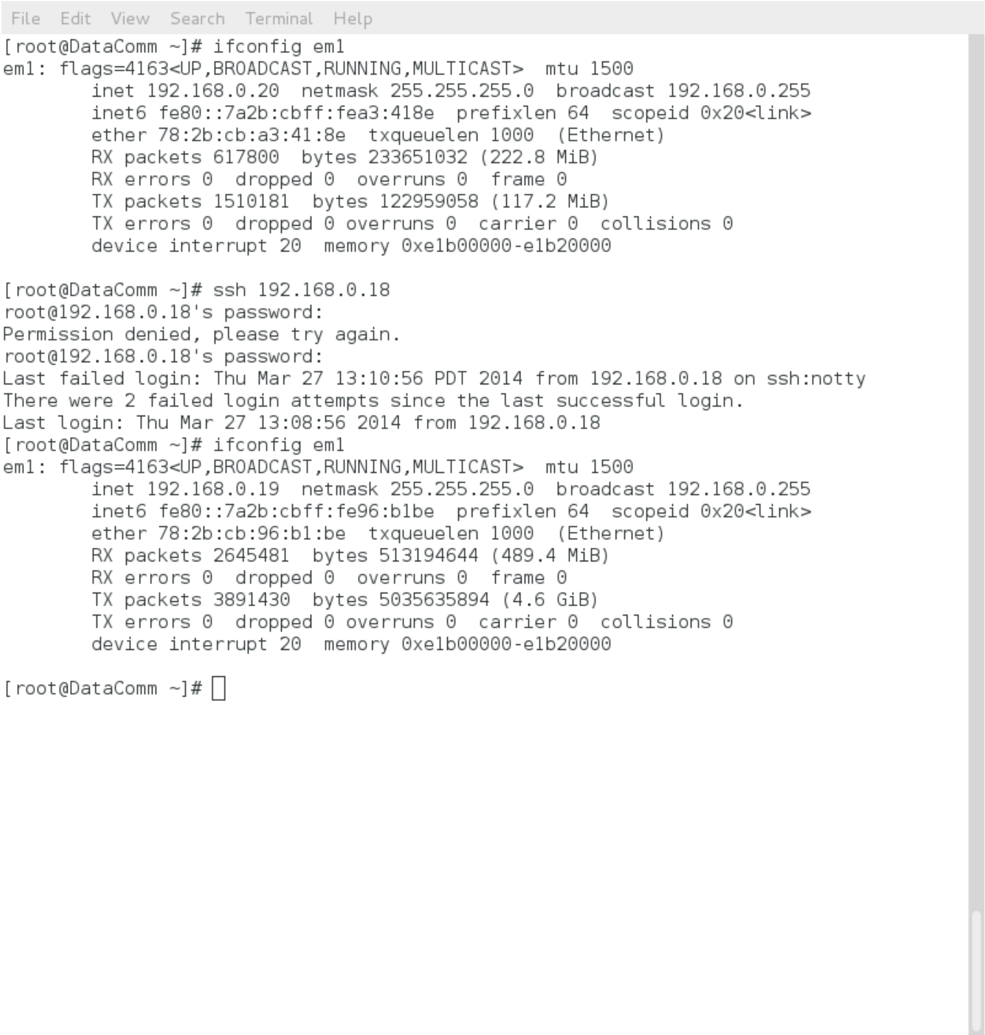
```
[root@DataComm Pictures]# systemctl status sshd
sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled)
   Active: active (running) since Wed 2014-03-19 18:02:34 PDT; 1 weeks 0 days ag
o
  Process: 14085 ExecStartPre=/usr/sbin/sshd-keygen (code=exited, status=0/SUCCE
SS)
 Main PID: 14090 (sshd)
   CGroup: name=systemd:/system/sshd.service
           └─14090 /usr/sbin/sshd -D

Mar 19 18:02:34 DataComm systemd[1]: Started OpenSSH server daemon.
Mar 19 18:02:34 DataComm sshd[14090]: Server listening on 0.0.0.0 port 1337.
Mar 19 18:02:34 DataComm sshd[14090]: Server listening on :: port 1337.
Mar 21 15:17:52 DataComm systemd[1]: Started OpenSSH server daemon.
Mar 24 12:44:50 DataComm sshd[12495]: Accepted password for root from 192.1...h2
Mar 24 13:00:49 DataComm sshd[12784]: Accepted password for root from 192.1...h2
Mar 24 14:44:45 DataComm sshd[14471]: Accepted password for root from 192.1...h2
Mar 24 14:48:06 DataComm sshd[14570]: Accepted password for root from 192.1...h2
Mar 24 14:54:35 DataComm sshd[14680]: Accepted password for root from 192.1...h2
Mar 24 16:50:28 DataComm sshd[15944]: Accepted password for root from 192.1...h2
[root@DataComm Pictures]#
```

Figure 1 Verification Server is listening on port 1337



```
[root@DataComm flaming-octo-cyril]# python port_forwarder.py
Service "http" is now forwarding from port 80 to 192.168.0.19:80
Service "ssh" is now forwarding from port 22 to 192.168.0.19:1337
New tunnel from port 22 to 192.168.0.19:1337 created for ('192.168.0.20', 58569)
```

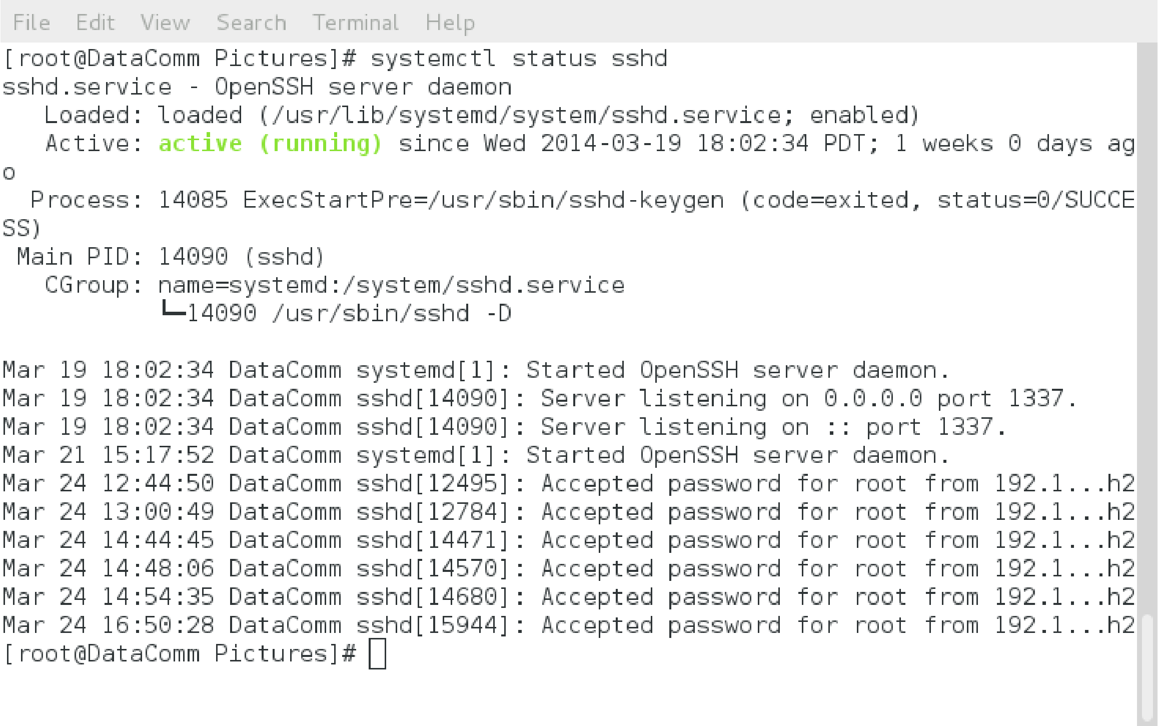
Figure 2 Verification of Port Forwarder settings



```
[root@DataComm ~]# ifconfig em1
em1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.20  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::7a2b:cbff:fea3:418e  prefixlen 64  scopeid 0x20<link>
        ether 78:2b:cb:a3:41:8e  txqueuelen 1000  (Ethernet)
        RX packets 617800  bytes 233651032 (222.8 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1510181  bytes 122959058 (117.2 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 20  memory 0xe1b00000-e1b20000

[root@DataComm ~]# ssh 192.168.0.18
root@192.168.0.18's password:
Permission denied, please try again.
root@192.168.0.18's password:
Last failed login: Thu Mar 27 13:10:56 PDT 2014 from 192.168.0.18 on ssh:notty
There were 2 failed login attempts since the last successful login.
Last login: Thu Mar 27 13:08:56 2014 from 192.168.0.18
[root@DataComm ~]# ifconfig em1
em1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.19  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::7a2b:cbff:fe96:b1be  prefixlen 64  scopeid 0x20<link>
        ether 78:2b:cb:96:b1:be  txqueuelen 1000  (Ethernet)
        RX packets 2645481  bytes 513194644 (489.4 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3891430  bytes 5035635894 (4.6 GiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 20  memory 0xe1b00000-e1b20000

[root@DataComm ~]# 
```

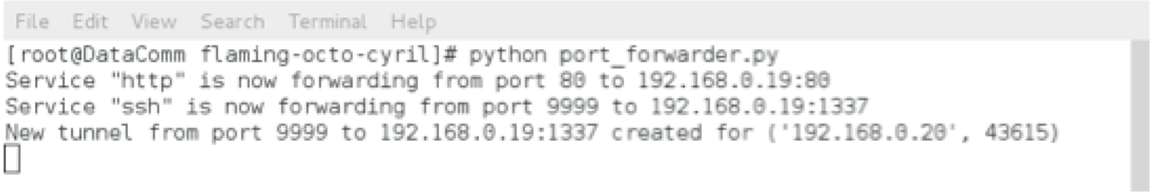
Figure 3 Verification of Client Connecting Through Port Forwarder

SSH port 9999 to 192.168.0.19:1337 test



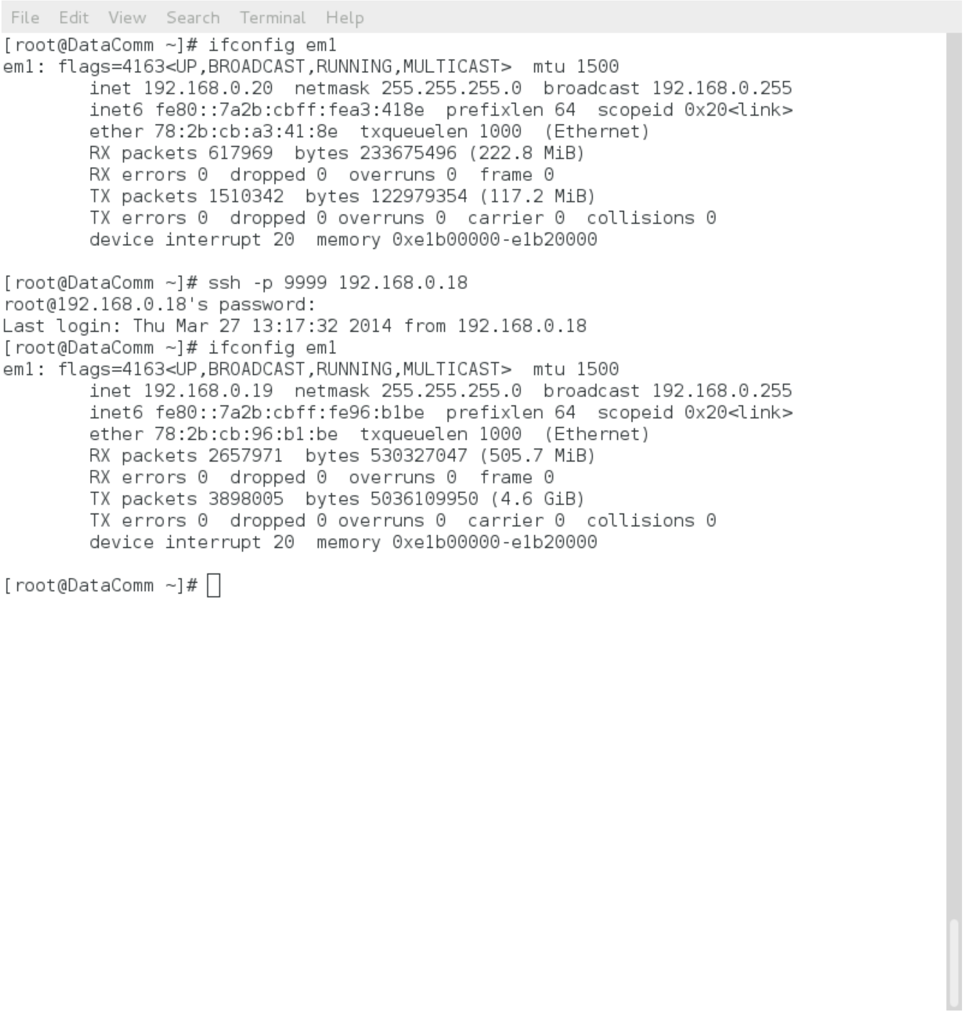
```
[root@DataComm Pictures]# systemctl status sshd
sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled)
   Active: active (running) since Wed 2014-03-19 18:02:34 PDT; 1 weeks 0 days ag
o
  Process: 14085 ExecStartPre=/usr/sbin/sshd-keygen (code=exited, status=0/SUCCE
SS)
 Main PID: 14090 (sshd)
   CGroup: name=systemd:/system/sshd.service
           └─14090 /usr/sbin/sshd -D

Mar 19 18:02:34 DataComm systemd[1]: Started OpenSSH server daemon.
Mar 19 18:02:34 DataComm sshd[14090]: Server listening on 0.0.0.0 port 1337.
Mar 19 18:02:34 DataComm sshd[14090]: Server listening on :: port 1337.
Mar 21 15:17:52 DataComm systemd[1]: Started OpenSSH server daemon.
Mar 24 12:44:50 DataComm sshd[12495]: Accepted password for root from 192.1...h2
Mar 24 13:00:49 DataComm sshd[12784]: Accepted password for root from 192.1...h2
Mar 24 14:44:45 DataComm sshd[14471]: Accepted password for root from 192.1...h2
Mar 24 14:48:06 DataComm sshd[14570]: Accepted password for root from 192.1...h2
Mar 24 14:54:35 DataComm sshd[14680]: Accepted password for root from 192.1...h2
Mar 24 16:50:28 DataComm sshd[15944]: Accepted password for root from 192.1...h2
[root@DataComm Pictures]# 
```

Figure 4 Verification of Server Listening on Port 1337



```
[root@DataComm flaming-octo-cyril]# python port_forwarder.py
Service "http" is now forwarding from port 80 to 192.168.0.19:80
Service "ssh" is now forwarding from port 9999 to 192.168.0.19:1337
New tunnel from port 9999 to 192.168.0.19:1337 created for ('192.168.0.20', 43615)
```

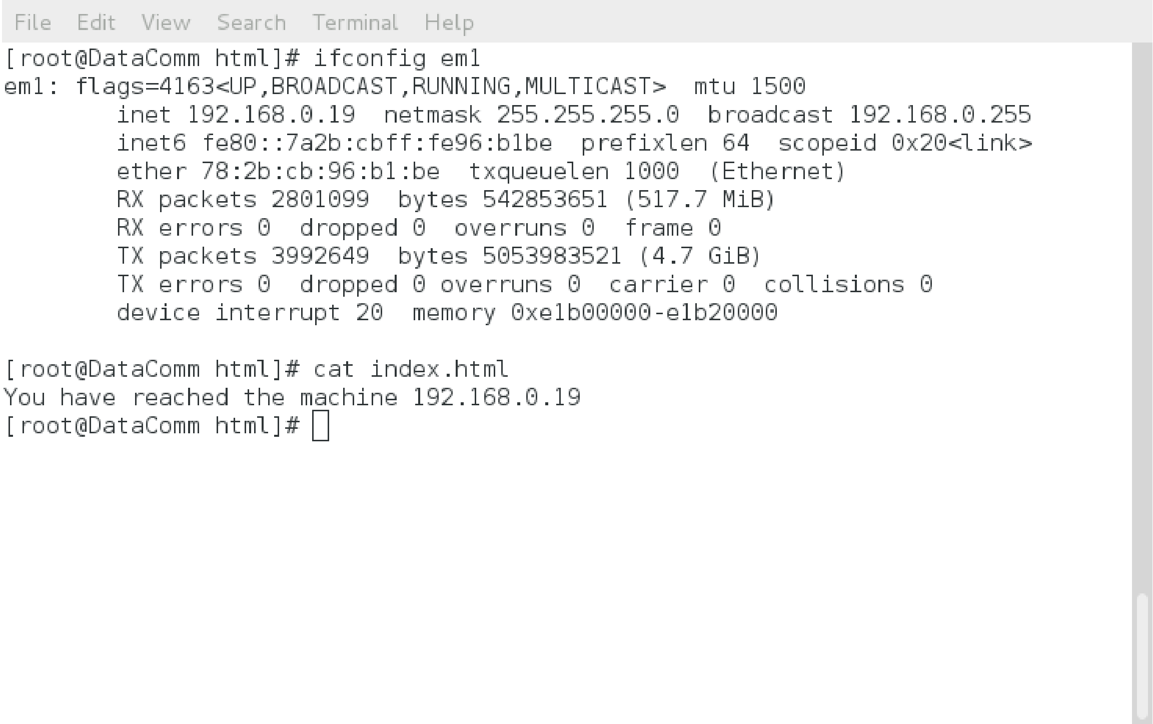
Figure 5 Verification of Port Forwarder Settings



```
[root@DataComm ~]# ifconfig em1
em1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.20  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::7a2b:cbff:fea3:418e  prefixlen 64  scopeid 0x20<link>
        ether 78:2b:cb:a3:41:8e  txqueuelen 1000  (Ethernet)
        RX packets 617969  bytes 233675496 (222.8 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1510342  bytes 122979354 (117.2 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 20  memory 0xe1b00000-e1b20000

[root@DataComm ~]# ssh -p 9999 192.168.0.18
root@192.168.0.18's password:
Last login: Thu Mar 27 13:17:32 2014 from 192.168.0.18
[root@DataComm ~]# ifconfig em1
em1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.19  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::7a2b:cbff:fe96:b1be  prefixlen 64  scopeid 0x20<link>
        ether 78:2b:cb:96:b1:be  txqueuelen 1000  (Ethernet)
        RX packets 2657971  bytes 530327047 (505.7 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3898005  bytes 5036109950 (4.6 GiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 20  memory 0xe1b00000-e1b20000

[root@DataComm ~]# 
```

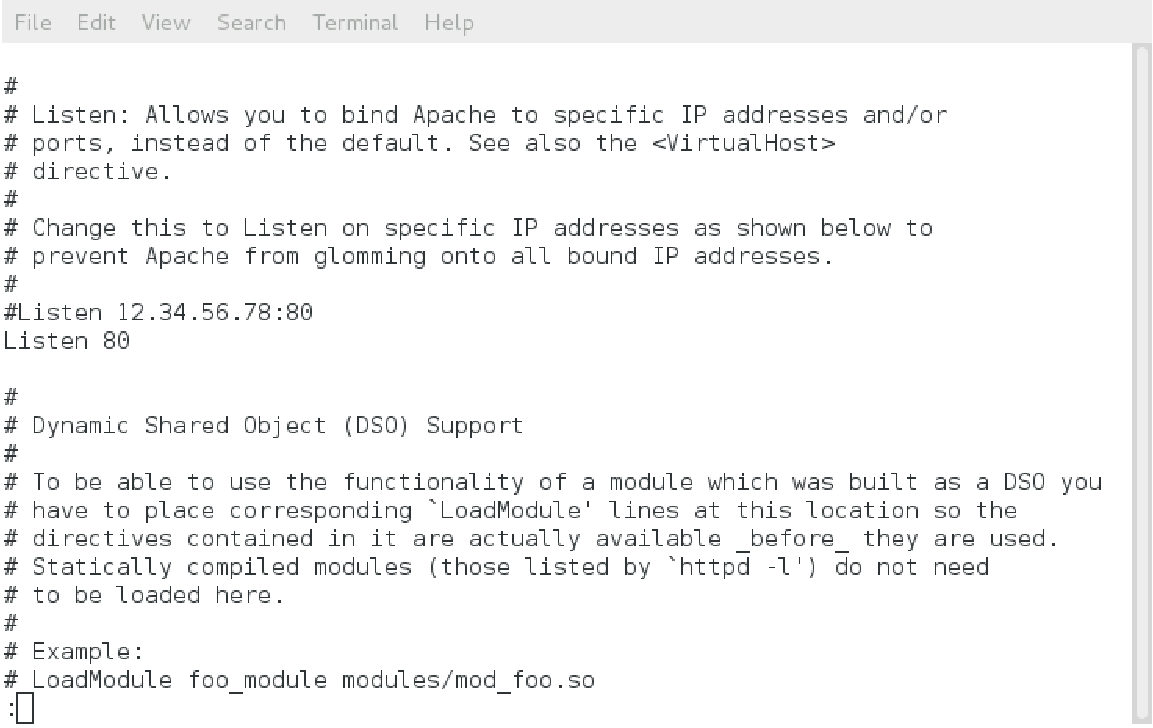
Figure 6 Verification of Client connecting through Port Forwarder

Apache port 4321 to 192.168.0.19:80 test



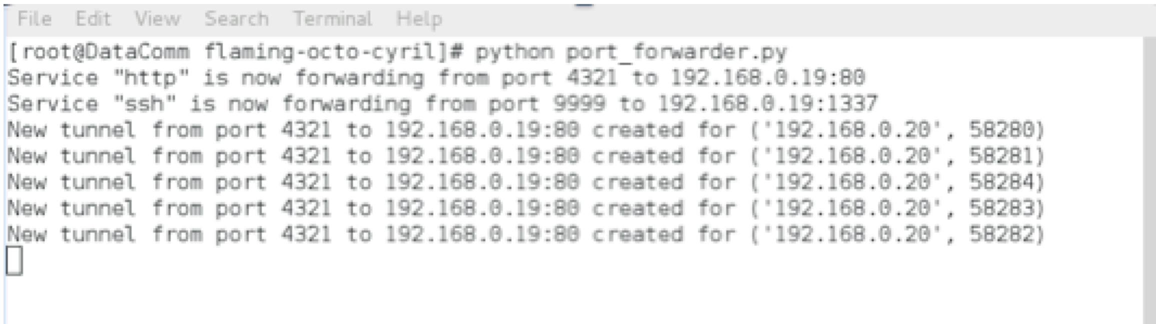
```
[root@DataComm html]# ifconfig em1
em1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.19  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::7a2b:cbff:fe96:b1be  prefixlen 64  scopeid 0x20<link>
        ether 78:2b:cb:96:b1:be  txqueuelen 1000  (Ethernet)
        RX packets 2801099  bytes 542853651 (517.7 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3992649  bytes 5053983521 (4.7 GiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 20  memory 0xe1b00000-e1b20000

[root@DataComm html]# cat index.html
You have reached the machine 192.168.0.19
[root@DataComm html]# 
```

Figure 7 Verification of Web Server Files



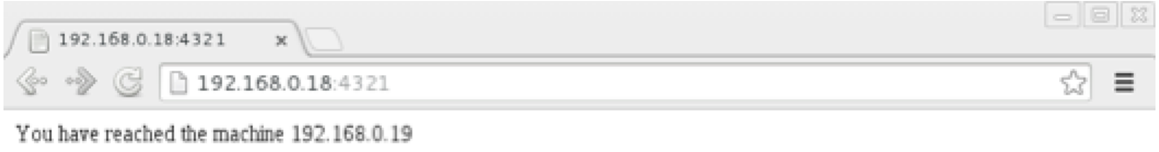
```
#
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses.
#
#Listen 12.34.56.78:80
Listen 80

#
# Dynamic Shared Object (DSO) Support
#
# To be able to use the functionality of a module which was built as a DSO you
# have to place corresponding `LoadModule' lines at this location so the
# directives contained in it are actually available _before_ they are used.
# Statically compiled modules (those listed by `httpd -l') do not need
# to be loaded here.
#
# Example:
# LoadModule foo_module modules/mod_foo.so
:
```

Figure 8 Verification of Web Server Port Settings



```
[root@DataComm flaming-octo-cyril]# python port_forwarder.py
Service "http" is now forwarding from port 4321 to 192.168.0.19:80
Service "ssh" is now forwarding from port 9999 to 192.168.0.19:1337
New tunnel from port 4321 to 192.168.0.19:80 created for ('192.168.0.20', 58280)
New tunnel from port 4321 to 192.168.0.19:80 created for ('192.168.0.20', 58281)
New tunnel from port 4321 to 192.168.0.19:80 created for ('192.168.0.20', 58284)
New tunnel from port 4321 to 192.168.0.19:80 created for ('192.168.0.20', 58283)
New tunnel from port 4321 to 192.168.0.19:80 created for ('192.168.0.20', 58282)
```

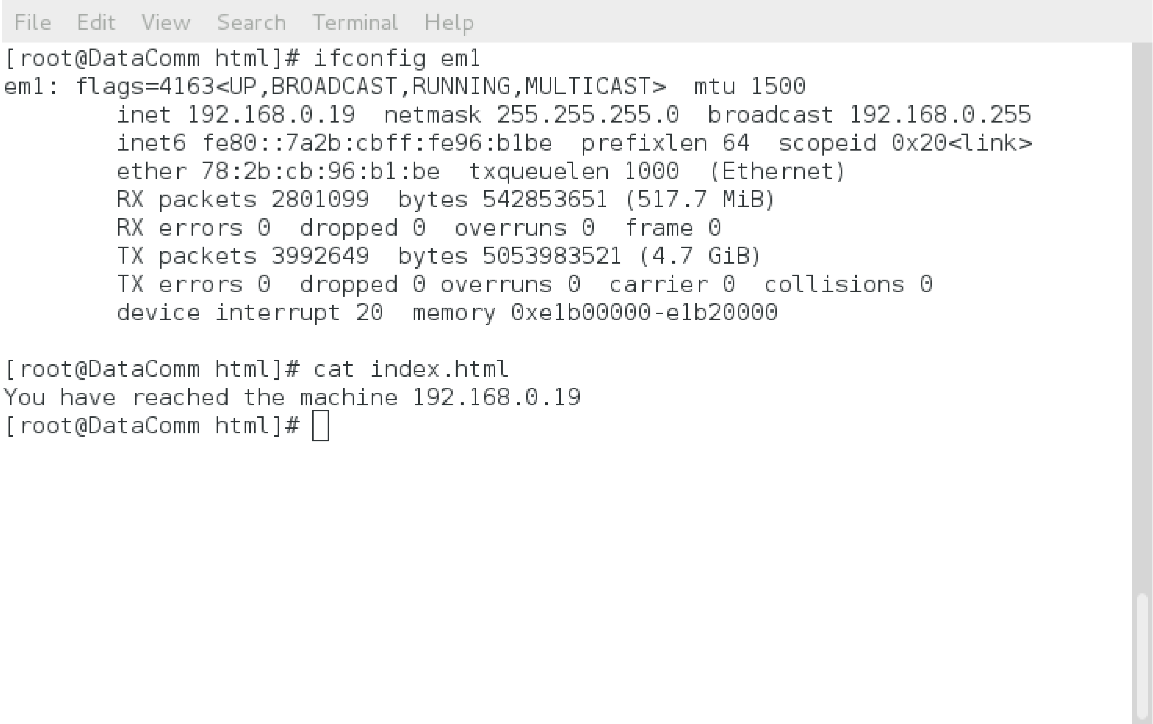
Figure 9 Verification of Port Forwarder Logs and Settings



You have reached the machine 192.168.0.19

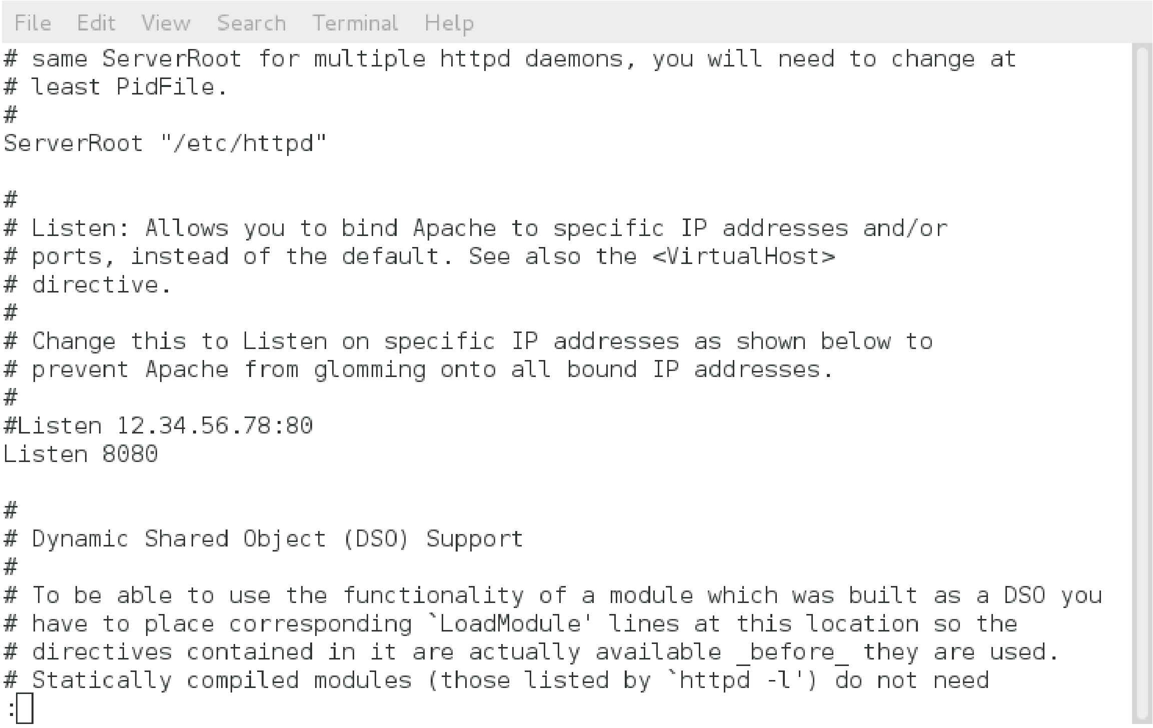
Figure 10 Verification of Client Connection

Apache port 4321 to 192.168.0.19:8080 test



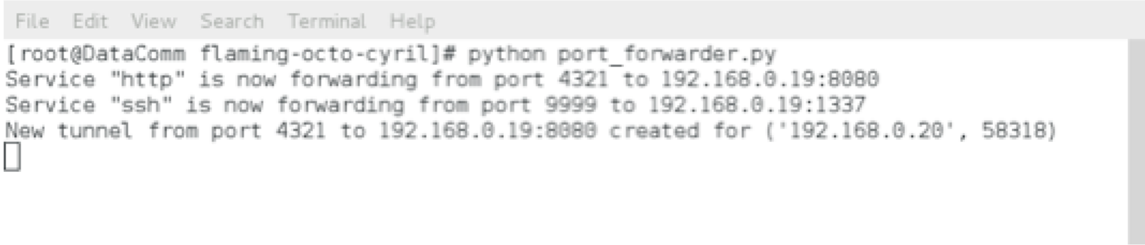
```
[root@DataComm html]# ifconfig em1
em1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.19  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::7a2b:cbff:fe96:b1be  prefixlen 64  scopeid 0x20<link>
        ether 78:2b:cb:96:b1:be  txqueuelen 1000  (Ethernet)
        RX packets 2801099  bytes 542853651 (517.7 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3992649  bytes 5053983521 (4.7 GiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 20  memory 0xe1b00000-e1b20000

[root@DataComm html]# cat index.html
You have reached the machine 192.168.0.19
[root@DataComm html]# 
```

Figure 11 Verification of Web Server Files



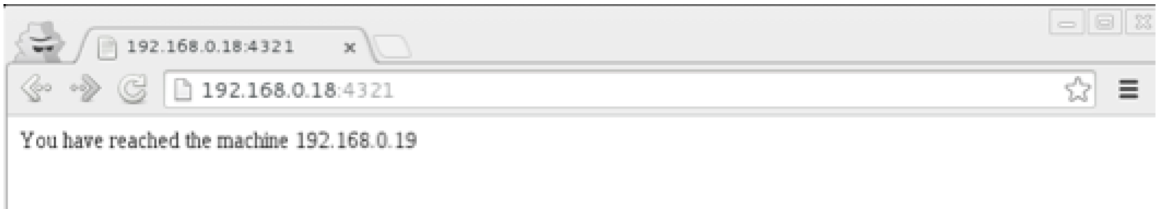
```
# same ServerRoot for multiple httpd daemons, you will need to change at
# least PidFile.
#
ServerRoot "/etc/httpd"

#
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses.
#
#Listen 12.34.56.78:80
Listen 8080

#
# Dynamic Shared Object (DSO) Support
#
# To be able to use the functionality of a module which was built as a DSO you
# have to place corresponding `LoadModule' lines at this location so the
# directives contained in it are actually available _before_ they are used.
# Statically compiled modules (those listed by `httpd -l') do not need
:
```

Figure 12 Verification of Web Server Port Settings



```
[root@DataComm flaming-octo-cyril]# python port_forwarder.py
Service "http" is now forwarding from port 4321 to 192.168.0.19:8080
Service "ssh" is now forwarding from port 9999 to 192.168.0.19:1337
New tunnel from port 4321 to 192.168.0.19:8080 created for ('192.168.0.20', 58318)
```

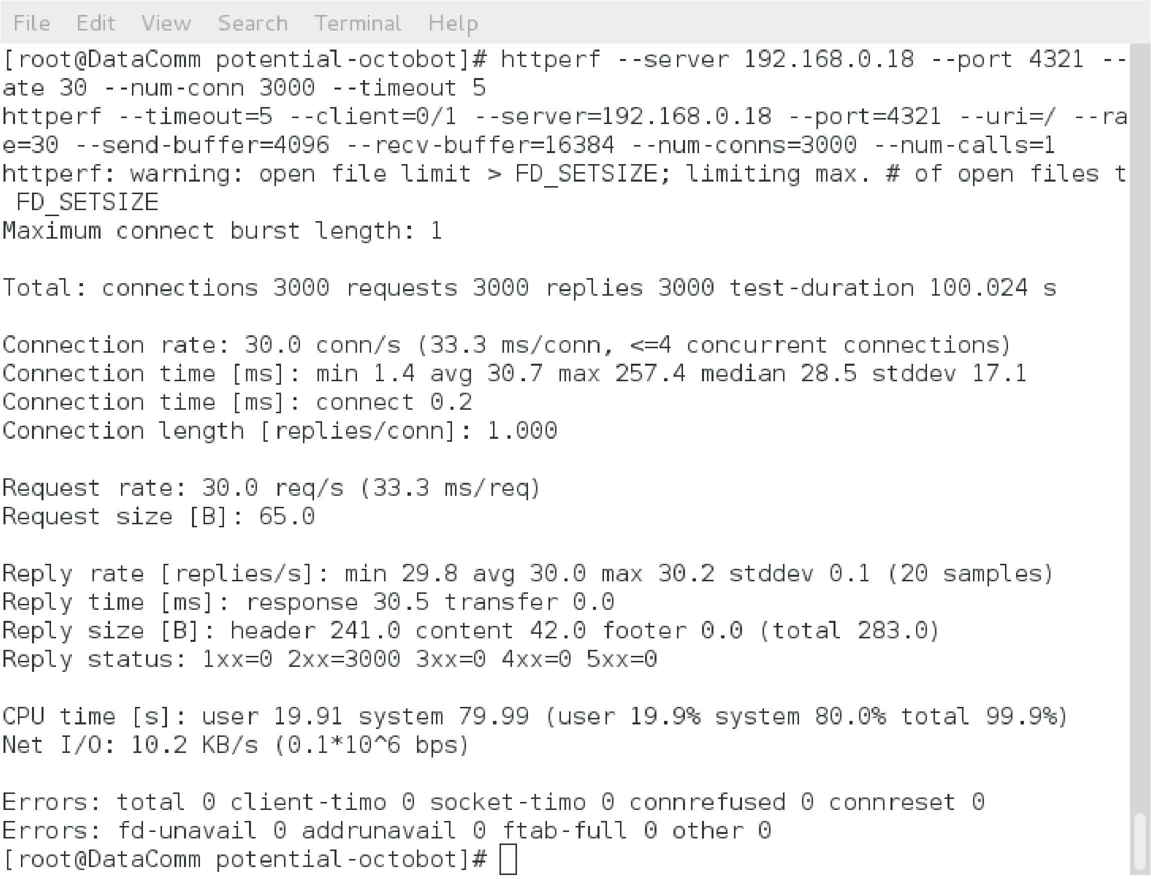
Figure 13 Verification of Port Forwarder Logs



You have reached the machine 192.168.0.19

Figure 14 Verification of Client Connection

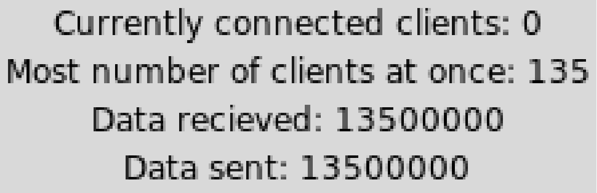
httperf test port 4321 to 192.168.0.19:8080 test



```
[root@DataComm potential-octobot]# httperf --server 192.168.0.18 --port 4321 --
ate 30 --num-conn 3000 --timeout 5
httperf --timeout=5 --client=0/1 --server=192.168.0.18 --port=4321 --uri=/ --ra
e=30 --send-buffer=4096 --recv-buffer=16384 --num-conns=3000 --num-calls=1
httperf: warning: open file limit > FD_SETSIZE; limiting max. # of open files t
 FD_SETSIZE
Maximum connect burst length: 1

Total: connections 3000 requests 3000 replies 3000 test-duration 100.024 s

Connection rate: 30.0 conn/s (33.3 ms/conn, <=4 concurrent connections)
Connection time [ms]: min 1.4 avg 30.7 max 257.4 median 28.5 stddev 17.1
Connection time [ms]: connect 0.2
Connection length [replies/conn]: 1.000

Request rate: 30.0 req/s (33.3 ms/req)
Request size [B]: 65.0

Reply rate [replies/s]: min 29.8 avg 30.0 max 30.2 stddev 0.1 (20 samples)
Reply time [ms]: response 30.5 transfer 0.0
Reply size [B]: header 241.0 content 42.0 footer 0.0 (total 283.0)
Reply status: 1xx=0 2xx=3000 3xx=0 4xx=0 5xx=0

CPU time [s]: user 19.91 system 79.99 (user 19.9% system 80.0% total 99.9%)
Net I/O: 10.2 KB/s (0.1*10^6 bps)

Errors: total 0 client-timo 0 socket-timo 0 connrefused 0 connreset 0
Errors: fd-unavail 0 addrunavail 0 ftab-full 0 other 0
[root@DataComm potential-octobot]# 
```

Figure 15 Httperf results. Server in identical state as above test case.

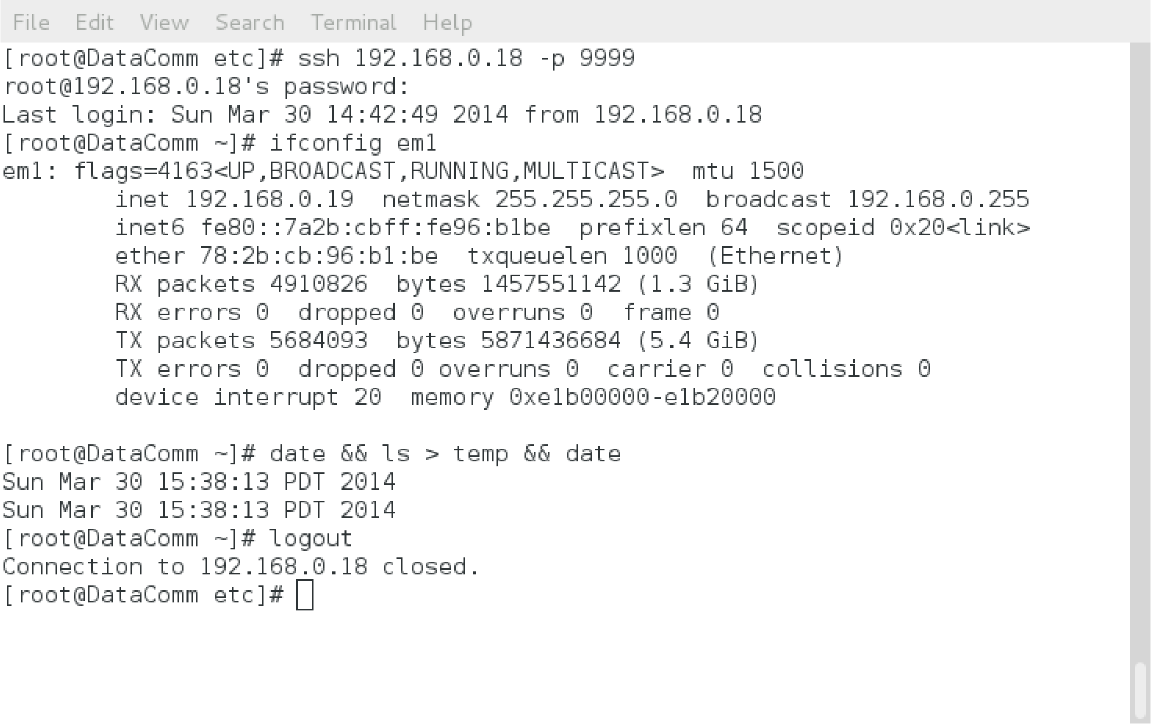
Assignment 2 test port 7777 to 192.168.0.19:7777



```
Currently connected clients: 0
Most number of clients at once: 135
Data recieved: 13500000
Data sent: 13500000
```

Figure 16 Verification of client connections through port forwarder.

Assignment 2 test port 7777 to 192.168.0.19:7777 with SSH port 9999 to 192.168.0.19:1337 latency test.



```
[root@DataComm etc]# ssh 192.168.0.18 -p 9999
root@192.168.0.18's password:
Last login: Sun Mar 30 14:42:49 2014 from 192.168.0.18
[root@DataComm ~]# ifconfig em1
em1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.19  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::7a2b:cbff:fe96:b1be  prefixlen 64  scopeid 0x20<link>
        ether 78:2b:cb:96:b1:be  txqueuelen 1000  (Ethernet)
        RX packets 4910826  bytes 1457551142 (1.3 GiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 5684093  bytes 5871436684 (5.4 GiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 20  memory 0xe1b00000-e1b20000

[root@DataComm ~]# date && ls > temp && date
Sun Mar 30 15:38:13 PDT 2014
Sun Mar 30 15:38:13 PDT 2014
[root@DataComm ~]# logout
Connection to 192.168.0.18 closed.
[root@DataComm etc]# 
```

Figure 17 Verification of SSH latency test. Ran concurrently to the above test case.