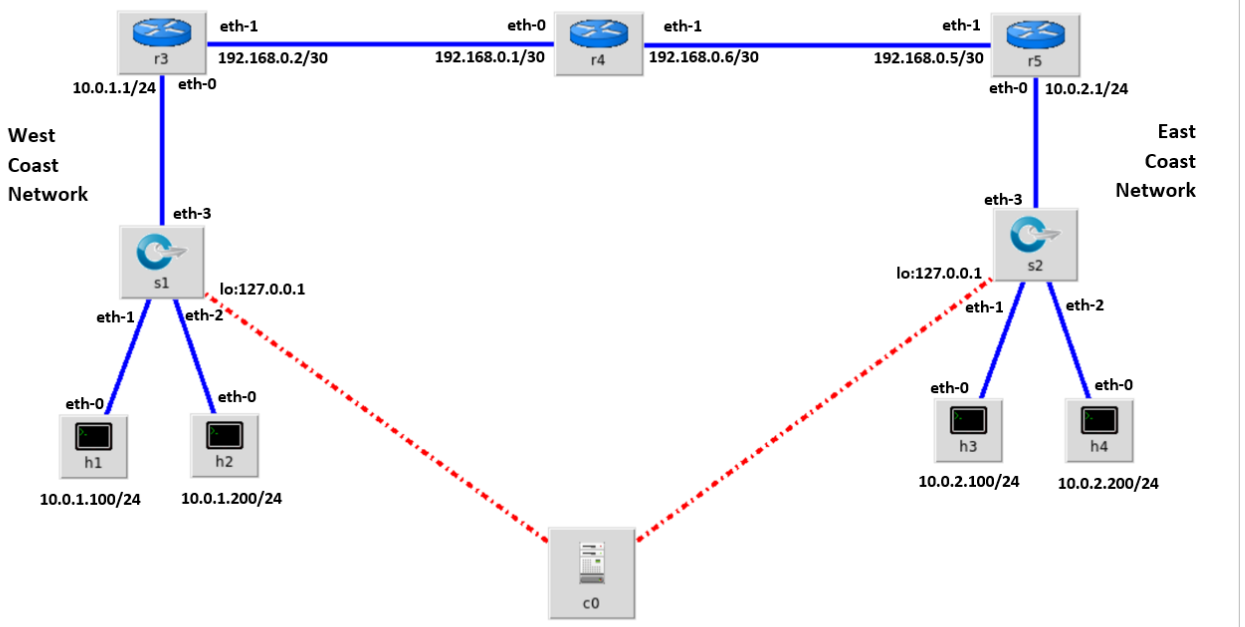
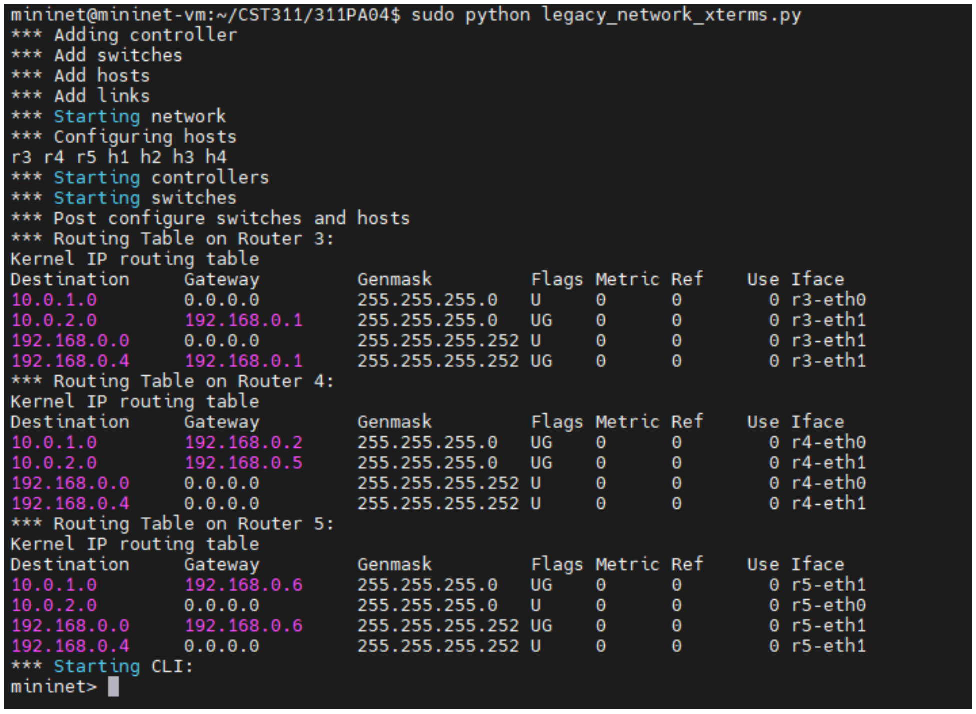
PA04

Group participants: Sabrina Ferras, Ian Lowe, Christian Martinez, Martin Ronquillo

Network Diagram

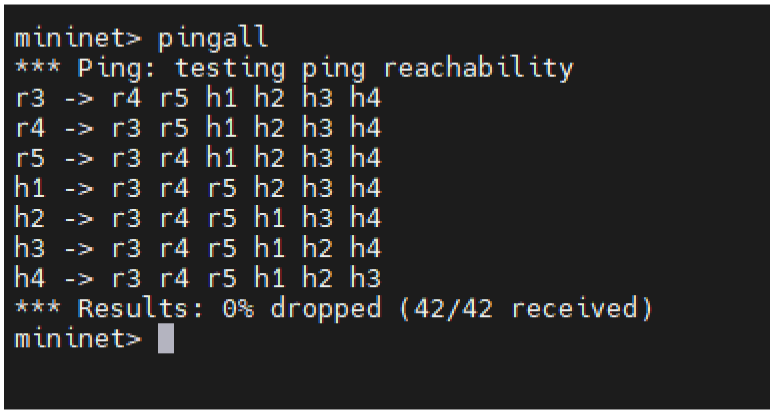


Screen capture of the program that runs with no Python errors.



```
mininet@mininet-vm:~/CST311/311PA04$ sudo python legacy_network_xterms.py
*** Adding controller
*** Add switches
*** Add hosts
*** Add links
*** Starting network
*** Configuring hosts
r3 r4 r5 h1 h2 h3 h4
*** Starting controllers
*** Starting switches
*** Post configure switches and hosts
*** Routing Table on Router 3:
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.1.0        0.0.0.0         255.255.255.0   U     0      0        0 r3-eth0
10.0.2.0        192.168.0.1     255.255.255.0   UG    0      0        0 r3-eth1
192.168.0.0     0.0.0.0         255.255.255.252 U     0      0        0 r3-eth1
192.168.0.4     192.168.0.1     255.255.255.252 UG    0      0        0 r3-eth1
*** Routing Table on Router 4:
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.1.0        192.168.0.2     255.255.255.0   UG    0      0        0 r4-eth0
10.0.2.0        192.168.0.5     255.255.255.0   UG    0      0        0 r4-eth1
192.168.0.0     0.0.0.0         255.255.255.252 U     0      0        0 r4-eth0
192.168.0.4     0.0.0.0         255.255.255.252 U     0      0        0 r4-eth1
*** Routing Table on Router 5:
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.1.0        192.168.0.6     255.255.255.0   UG    0      0        0 r5-eth1
10.0.2.0        0.0.0.0         255.255.255.0   U     0      0        0 r5-eth0
192.168.0.0     192.168.0.6     255.255.255.252 UG    0      0        0 r5-eth1
192.168.0.4     0.0.0.0         255.255.255.252 U     0      0        0 r5-eth1
*** Starting CLI:
mininet> 
```

Successful Pingall



```
mininet> pingall
*** Ping: testing ping reachability
r3 -> r4 r5 h1 h2 h3 h4
r4 -> r3 r5 h1 h2 h3 h4
r5 -> r3 r4 h1 h2 h3 h4
h1 -> r3 r4 r5 h2 h3 h4
h2 -> r3 r4 r5 h1 h3 h4
h3 -> r3 r4 r5 h1 h2 h4
h4 -> r3 r4 r5 h1 h2 h3
*** Results: 0% dropped (42/42 received)
mininet> 
```

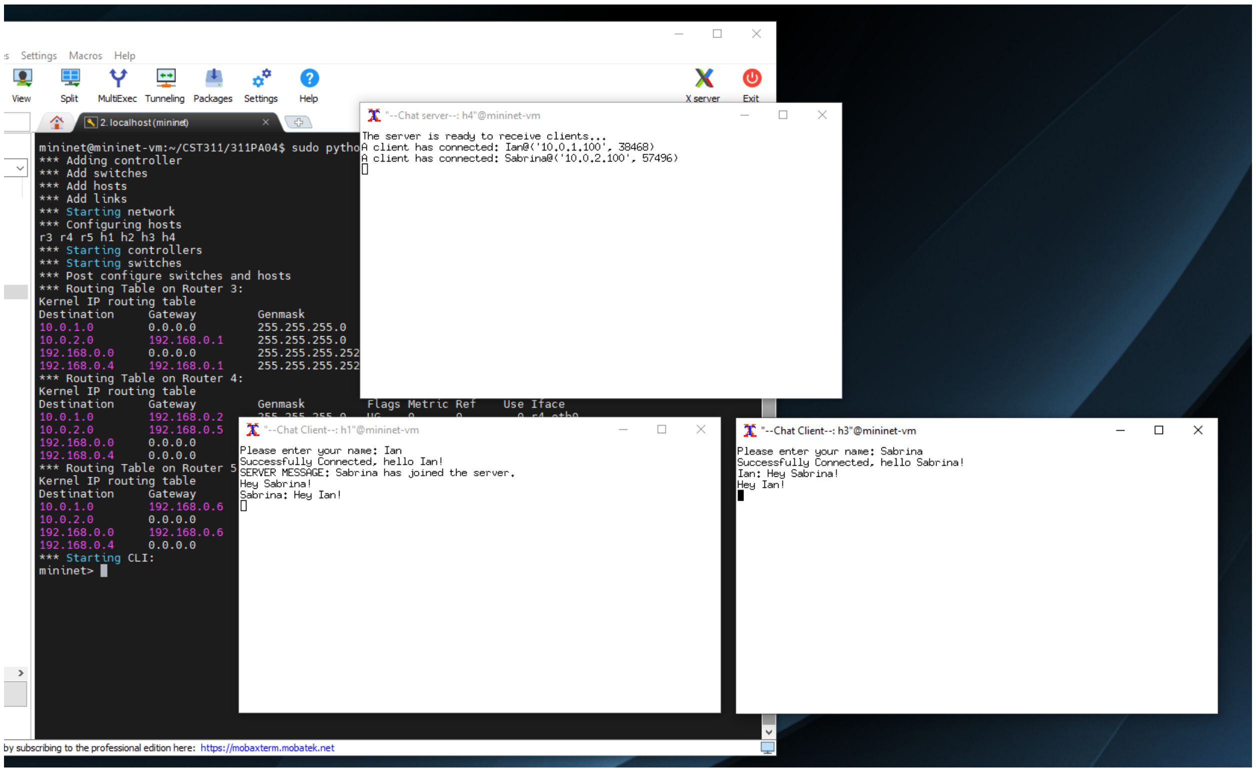
List of Lines Changed:

1. Moved lines that added switches to be executed before adding routers. This prevented the build errors.
2. Gave routers unique IP addresses instead of ‘0.0.0.0’, with CIDR numbers of /24. This was the first step necessary to set up the subnets of the network.
3. Gave hosts unique IP addresses according to the subnets to which they belong. Added default routes to the gateway routers of their subnet.
4. Added special links between r3/r4, and r4/r5 and specified that these links were on new interfaces within new subnets, each with CIDR numbers of /30. We needed that specific subnet mask because we were creating two host subnets. With two usable bits for the host portion, you can give unique IP addresses to two hosts, with the other two available addresses reserved for the network number and broadcast address.
5. Added static routes between each subnet. In total, we needed six static routes configured for all hosts to be able to ping each other. From 10.0.1.0 -> 192.168.0.0, from 192.168.0.0 -> 192.168.0.4, from 10.0.2.0 -> 192.168.0.4, from 192.168.0.4 -> 192.168.0.0, from 10.0.1.0 -> 192.168.0.4, and from 10.0.2.0 -> 192.168.0.0
6. Integrated the makeTerm() method to auto launch xterm windows for launching chat/server programs. Although this wasn’t necessary, it was an added touch to automate some of the testing.

Answers to Questions:

1. What were any interesting findings and lessons learned?
    a. This project helped us solidify our understanding of subnets. Having the requirement that the subnets between r3/r4, r4/r5 be two host subnets aided in our understanding of broadcast addresses and network numbers. We also got great practice in various networking linux commands, such as ‘route’, ‘ifconfig’, and ‘ip’. Debugging with wireshark solidified our understanding of the ARP protocol.
2. Why didn’t the original program forward packets between the hosts?
    a. The original program didn't forward packets between hosts because they didn't know how to reach them. Each host had a default route to the gateway router of their subnet, but there were no routes in the gateway router's routing table to the subnets between the East and West coast networks.
3. Is the line ‘ r3.cmd('sysctl -w net.ipv4.ip_forward=1') ’ required?
    a. These lines are required. Without them, the packets meant for hosts on other subnets wouldn’t be forwarded.
4. Intentionally break your working program, e.g.: change a subnet length, IP address, or default route for a host. Explain why your change caused the network to break.
    a. One way to break our network would be to change the IP addresses of either r4-eth1 or r5-eth1 to 192.168.0.4 or 192.168.0.7. Although these addresses are technically within the host portion of the address space, they are reserved for the network number and broadcast address and thus wouldn’t work as assignable addresses for these interfaces.

Successful Chat Session



Settings Macros Help
View
Split
MultiExec
Tunneling
Packages
Settings
Help
X server
Exit
2. localhost (mininet)
"--Chat server--: h4"@mininet-vm
"--Chat Client--: h1"@mininet-vm
"--Chat Client--: h3"@mininet-vm
by subscribing to the professional edition here: https://mobaxterm.mobatek.net

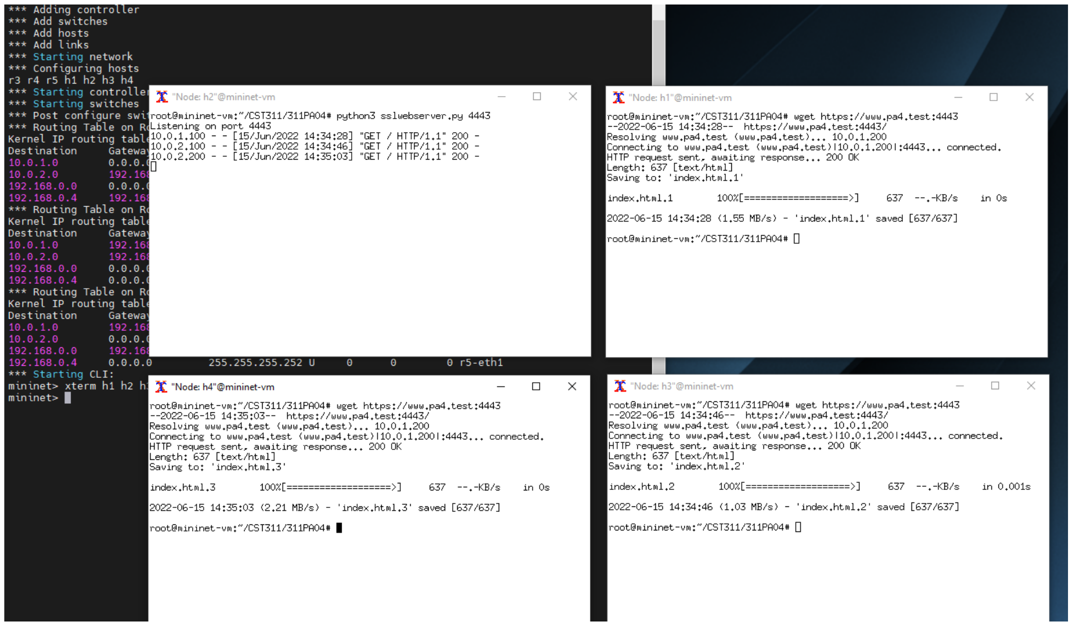
```
mininet@mininet-vm:~/CST311/311PA04$ sudo pytho
*** Adding controller
*** Add switches
*** Add hosts
*** Add links
*** Starting network
*** Configuring hosts
r3 r4 r5 h1 h2 h3 h4
*** Starting controllers
*** Starting switches
*** Post configure switches and hosts
*** Routing Table on Router 3:
Kernel IP routing table
Destination     Gateway         Genmask
10.0.1.0        0.0.0.0         255.255.255.0
10.0.2.0        192.168.0.1     255.255.255.0
192.168.0.0     0.0.0.0         255.255.255.252
192.168.0.4     192.168.0.1     255.255.255.252
*** Routing Table on Router 4:
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.1.0        192.168.0.2
10.0.2.0        192.168.0.5
192.168.0.0     0.0.0.0
192.168.0.4     0.0.0.0
*** Routing Table on Router 5
Kernel IP routing table
Destination     Gateway
10.0.1.0        192.168.0.6
10.0.2.0        0.0.0.0
192.168.0.0     192.168.0.6
192.168.0.4     0.0.0.0
*** Starting CLI:
mininet>
```

```
The server is ready to receive clients...
A client has connected: Ian@('10.0.1.100', 38468)
A client has connected: Sabrina@('10.0.2.100', 57496)
```

```
Please enter your name: Ian
Successfully Connected, hello Ian!
SERVER MESSAGE: Sabrina has joined the server.
Hey Sabrina!
Sabrina: Hey Ian!
```

```
Please enter your name: Sabrina
Successfully Connected, hello Sabrina!
Ian: Hey Sabrina!
Hey Ian!
```

Successful wget



```
*** Adding controller
*** Add switches
*** Add hosts
*** Add links
*** Starting network
*** Configuring hosts
r3 r4 r5 h1 h2 h3 h4
*** Starting controlle
*** Starting switches
*** Post configure swi
*** Routing Table on R
Kernel IP routing table
Destination     Gateway
10.0.1.0        0.0.0.
10.0.2.0        192.168
192.168.0.0     0.0.0.
192.168.0.4     192.168
*** Routing Table on R
Kernel IP routing table
Destination     Gateway
10.0.1.0        192.168
10.0.2.0        192.168
192.168.0.0     0.0.0.
192.168.0.4     0.0.0.
*** Routing Table on R
Kernel IP routing table
Destination     Gateway
10.0.1.0        192.168
10.0.2.0        0.0.0.
192.168.0.0     192.168
192.168.0.4     0.0.0.0         255.255.255.252 U     0      0        0 r5-eth1
*** Starting CLI:
mininet> xterm h1 h2 h
mininet>
```

```
root@mininet-vm:~/CST311/311PA04# python3 sslwebserver.py 4443
Listening on port 4443
10.0.1.100 - - [15/Jun/2022 14:34:28] "GET / HTTP/1.1" 200 -
10.0.2.100 - - [15/Jun/2022 14:34:46] "GET / HTTP/1.1" 200 -
10.0.2.200 - - [15/Jun/2022 14:35:03] "GET / HTTP/1.1" 200 -
```

```
root@mininet-vm:~/CST311/311PA04# wget https://www.pa4.test:4443
--2022-06-15 14:34:28--  https://www.pa4.test:4443/
Resolving www.pa4.test (www.pa4.test)... 10.0.1.200
Connecting to www.pa4.test (www.pa4.test)|10.0.1.200|:4443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 637 [text/html]
Saving to: 'index.html.1'

index.html.1        100%[===================>]     637  --.-KB/s    in 0s

2022-06-15 14:34:28 (1.55 MB/s) - 'index.html.1' saved [637/637]

root@mininet-vm:~/CST311/311PA04#
```

```
root@mininet-vm:~/CST311/311PA04# wget https://www.pa4.test:4443
--2022-06-15 14:35:03--  https://www.pa4.test:4443/
Resolving www.pa4.test (www.pa4.test)... 10.0.1.200
Connecting to www.pa4.test (www.pa4.test)|10.0.1.200|:4443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 637 [text/html]
Saving to: 'index.html.3'

index.html.3        100%[===================>]     637  --.-KB/s    in 0s

2022-06-15 14:35:03 (2.21 MB/s) - 'index.html.3' saved [637/637]

root@mininet-vm:~/CST311/311PA04#
```

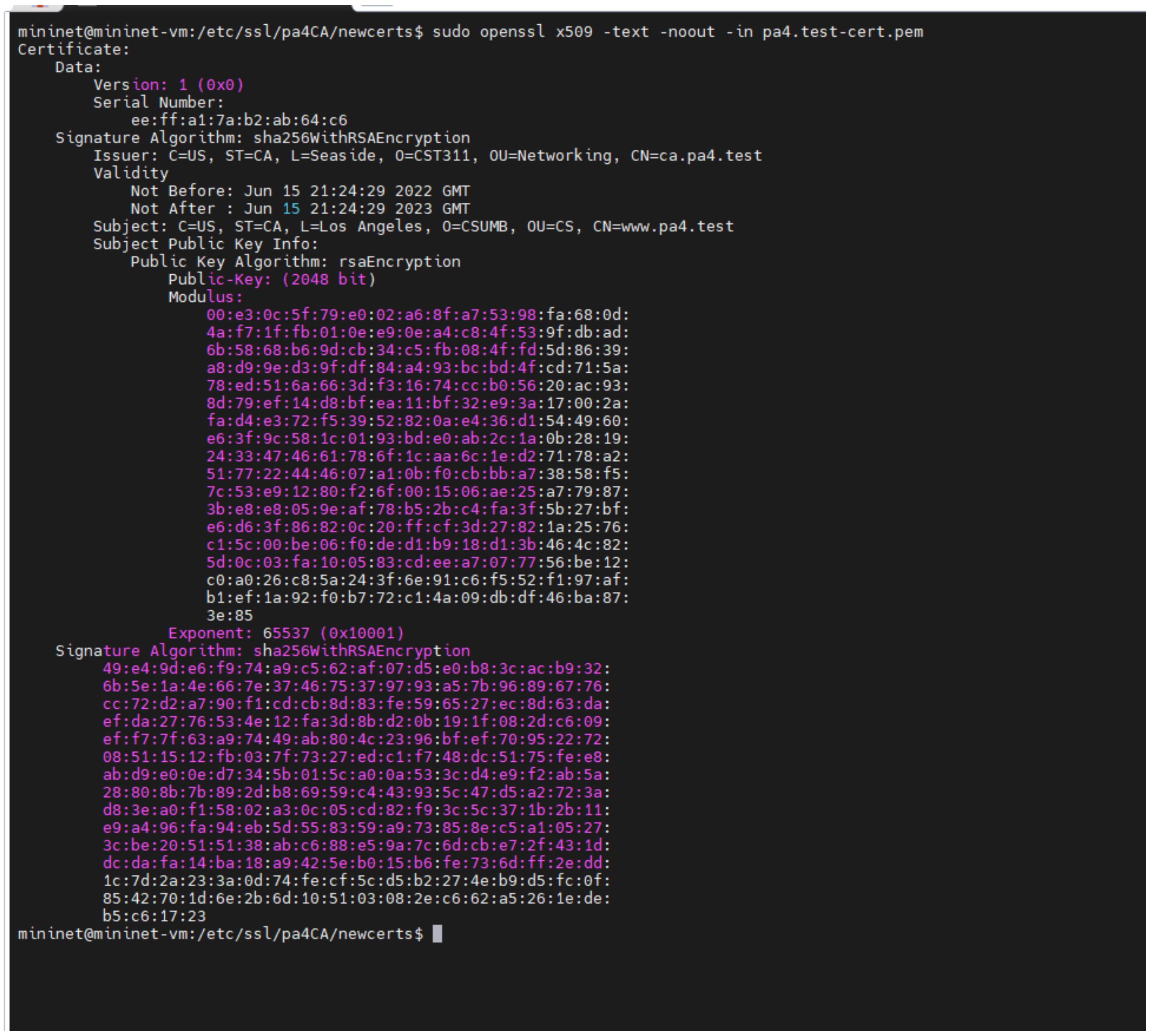
```
root@mininet-vm:~/CST311/311PA04# wget https://www.pa4.test:4443
--2022-06-15 14:34:46--  https://www.pa4.test:4443/
Resolving www.pa4.test (www.pa4.test)... 10.0.1.200
Connecting to www.pa4.test (www.pa4.test)|10.0.1.200|:4443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 637 [text/html]
Saving to: 'index.html.2'

index.html.2        100%[===================>]     637  --.-KB/s    in 0.001s

2022-06-15 14:34:46 (1.03 MB/s) - 'index.html.2' saved [637/637]

root@mininet-vm:~/CST311/311PA04#
```

Decrypted Web Server Certificate



```
mininet@mininet-vm:/etc/ssl/pa4CA/newcerts$ sudo openssl x509 -text -noout -in pa4.test-cert.pem
Certificate:
    Data:
        Version: 1 (0x0)
        Serial Number:
            ee:ff:a1:7a:b2:ab:64:c6
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=US, ST=CA, L=Seaside, O=CST311, OU=Networking, CN=ca.pa4.test
        Validity
            Not Before: Jun 15 21:24:29 2022 GMT
            Not After : Jun 15 21:24:29 2023 GMT
        Subject: C=US, ST=CA, L=Los Angeles, O=CSUMB, OU=CS, CN=www.pa4.test
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:e3:0c:5f:79:e0:02:a6:8f:a7:53:98:fa:68:0d:
                    4a:f7:1f:fb:01:0e:e9:0e:a4:c8:4f:53:9f:db:ad:
                    6b:58:68:b6:9d:cb:34:c5:fb:08:4f:fd:5d:86:39:
                    a8:d9:9e:d3:9f:df:84:a4:93:bc:bd:4f:cd:71:5a:
                    78:ed:51:6a:66:3d:f3:16:74:cc:b0:56:20:ac:93:
                    8d:79:ef:14:d8:bf:ea:11:bf:32:e9:3a:17:00:2a:
                    fa:d4:e3:72:f5:39:52:82:0a:e4:36:d1:54:49:60:
                    e6:3f:9c:58:1c:01:93:bd:e0:ab:2c:1a:0b:28:19:
                    24:33:47:46:61:78:6f:1c:aa:6c:1e:d2:71:78:a2:
                    51:77:22:44:46:07:a1:0b:f0:cb:bb:a7:38:58:f5:
                    7c:53:e9:12:80:f2:6f:00:15:06:ae:25:a7:79:87:
                    3b:e8:e8:05:9e:af:78:b5:2b:c4:fa:3f:5b:27:bf:
                    e6:d6:3f:86:82:0c:20:ff:cf:3d:27:82:1a:25:76:
                    c1:5c:00:be:06:f0:de:d1:b9:18:d1:3b:46:4c:82:
                    5d:0c:03:fa:10:05:83:cd:ee:a7:07:77:56:be:12:
                    c0:a0:26:c8:5a:24:3f:6e:91:c6:f5:52:f1:97:af:
                    b1:ef:1a:92:f0:b7:72:c1:4a:09:db:df:46:ba:87:
                    3e:85
                Exponent: 65537 (0x10001)
    Signature Algorithm: sha256WithRSAEncryption
         49:e4:9d:e6:f9:74:a9:c5:62:af:07:d5:e0:b8:3c:ac:b9:32:
         6b:5e:1a:4e:66:7e:37:46:75:37:97:93:a5:7b:96:89:67:76:
         cc:72:d2:a7:90:f1:cd:cb:8d:83:fe:59:65:27:ec:8d:63:da:
         ef:da:27:76:53:4e:12:fa:3d:8b:d2:0b:19:1f:08:2d:c6:09:
         ef:f7:7f:63:a9:74:49:ab:80:4c:23:96:bf:ef:70:95:22:72:
         08:51:15:12:fb:03:7f:73:27:ed:c1:f7:48:dc:51:75:fe:e8:
         ab:d9:e0:0e:d7:34:5b:01:5c:a0:0a:53:3c:d4:e9:f2:ab:5a:
         28:80:8b:7b:89:2d:b8:69:59:c4:43:93:5c:47:d5:a2:72:3a:
         d8:3e:a0:f1:58:02:a3:0c:05:cd:82:f9:3c:5c:37:1b:2b:11:
         e9:a4:96:fa:94:eb:5d:55:83:59:a9:73:85:8e:c5:a1:05:27:
         3c:be:20:51:51:38:ab:c6:88:e5:9a:7c:6d:cb:e7:2f:43:1d:
         dc:da:fa:14:ba:18:a9:42:5e:b0:15:b6:fe:73:6d:ff:2e:dd:
         1c:7d:2a:23:3a:0d:74:fe:cf:5c:d5:b2:27:4e:b9:d5:fc:0f:
         85:42:70:1d:6e:2b:6d:10:51:03:08:2e:c6:62:a5:26:1e:de:
         b5:c6:17:23
mininet@mininet-vm:/etc/ssl/pa4CA/newcerts$
```