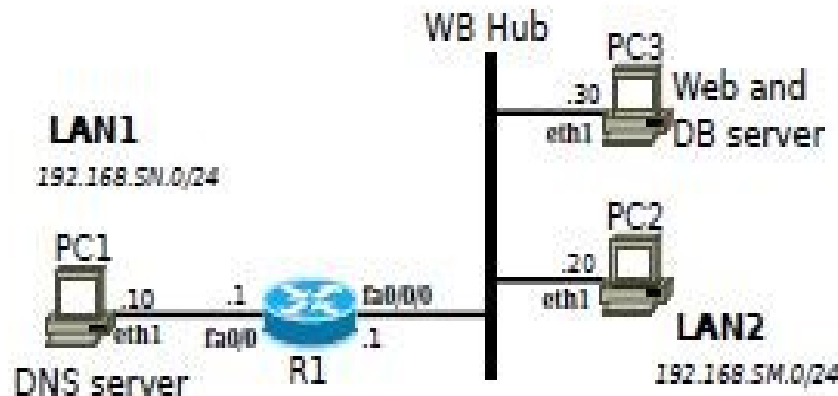


Lab 12: Application Layer
Mark Rutkowski
Arun Kumar Rajendra Kumar

Here, we discuss about setup of DNS, HTTP and HTTPS server. DNS serves as a name to the IP address thereby eliminating the need of remembering IP address. Generally, DNS depends on TCP or UDP and they are at application layer. Only when there is a data truncation, DNS prefers TCP. We set up the following network to start with the experiment.



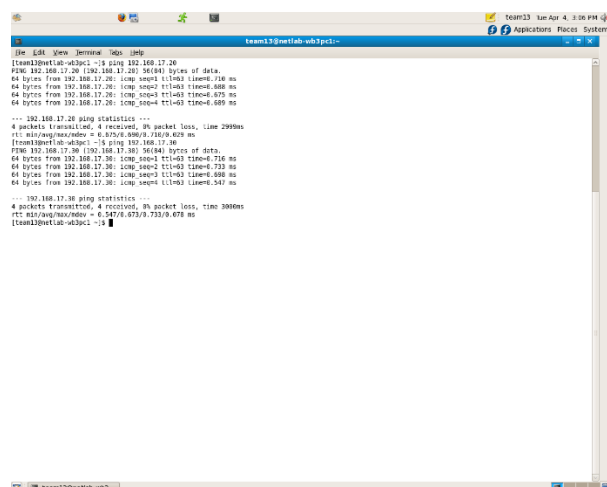
Webmin service is started on PC1 using **sudo/etc/init.d/webmin start** command & <http://127.0.0.1:10000> is accessed using the web browser. Changes are made in Address and Topology section, Zone defaults in the DNS main page. Then a master zone is created for forward lookup. Following it, master zone for reverse lookup is created for each LAN. Address record is added to each PC and then DNS server is restarted. Dig is run on PC2 for forward and reverse queries. It is checked by pinging www.wbSN.netlab.local. In the second part, an Apache web server is configured from the root directory i.e. /var/www/html. A html file called as index.html is created and then httpd is configured by making necessary changes in Listen & Server name. Again the web browser is started to access the web page set up on PC3. In the third section, we use telnet command which allows the client host to access web server. When the command for it executed, the server replies with HTTP header and HTML content. For another html file, this part is repeated.

Virtual host refers to running more than one web server on a single machine. It can be either IP-based or name-based. And here we learn how to configure virtual hosts. The old html file created is copied into the new file & some directives are added inside the configuration file. Once, this step is done, the httpd daemon is restarted & web browser is checked for its status. An additional step is performed by running the website with same IP address but different port number.

Access control allows access to a given resource only by control policies and user authentication. That feature is tested here. Initially, a html file called index.html is created and the http main configuration file is edited. Changes are made in such a way that only devices in LAN2 can access the page. Then, authentication scheme of Apache is tested by providing the necessary command. Again the web server is restarted. Now the web page is made accessible only when a valid user name and password is provided. It works on IP-based access control and various parameters like the TCP requests, GET messages are studied.

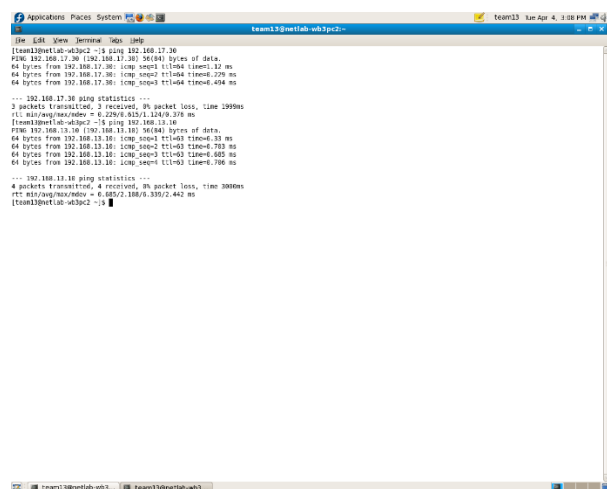
We might be using https connection in our daily lives even without our knowledge. This section describes its usage. HTTPS makes sure that there is data encryption and server authentication in a TCP connection using protocols like SSL or TLS. To play with this, we first make necessary changes in the configuration file by editing the listen directive followed by restarting the web server.

Section 4.1: Topology Setup



```
team13@metlab-wb3pc1:~$ ping 192.168.17.20
PING 192.168.17.20 (192.168.17.20) 56(84) bytes of data:
64 bytes from 192.168.17.20: icmp_seq=1 ttl=63 time=0.688 ms
64 bytes from 192.168.17.20: icmp_seq=2 ttl=63 time=0.688 ms
64 bytes from 192.168.17.20: icmp_seq=3 ttl=63 time=0.689 ms
^C
--- 192.168.17.20 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2998ms
rtt min/avg/max/mdev = 0.675/0.686/0.702/0.028 ms
team13@metlab-wb3pc1:~$ ping 192.168.17.30
PING 192.168.17.30 (192.168.17.30) 56(84) bytes of data:
64 bytes from 192.168.17.30: icmp_seq=1 ttl=63 time=0.716 ms
64 bytes from 192.168.17.30: icmp_seq=2 ttl=63 time=0.733 ms
64 bytes from 192.168.17.30: icmp_seq=3 ttl=63 time=0.680 ms
^C
--- 192.168.17.30 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3000ms
rtt min/avg/max/mdev = 0.547/0.673/0.733/0.078 ms
team13@metlab-wb3pc1:~$
```

This is PC1 ping PC2 and PC3.



```
team13@metlab-wb3pc2:~$ ping 192.168.17.20
PING 192.168.17.20 (192.168.17.20) 56(84) bytes of data:
64 bytes from 192.168.17.20: icmp_seq=1 ttl=64 time=0.137 ms
64 bytes from 192.168.17.20: icmp_seq=2 ttl=64 time=0.220 ms
64 bytes from 192.168.17.20: icmp_seq=3 ttl=64 time=0.404 ms
^C
--- 192.168.17.20 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1999ms
rtt min/avg/max/mdev = 0.220/0.432/0.730/0.218 ms
team13@metlab-wb3pc2:~$ ping 192.168.17.30
PING 192.168.17.30 (192.168.17.30) 56(84) bytes of data:
64 bytes from 192.168.17.30: icmp_seq=1 ttl=63 time=0.33 ms
64 bytes from 192.168.17.30: icmp_seq=2 ttl=63 time=0.703 ms
64 bytes from 192.168.17.30: icmp_seq=3 ttl=63 time=0.685 ms
^C
--- 192.168.17.30 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3000ms
rtt min/avg/max/mdev = 0.485/0.386/0.330/0.442 ms
team13@metlab-wb3pc2:~$
```

This is PC2 ping PC1 and PC3.

```
team13@netlab-wb3pc3~$ ifconfig
eth0: flags=4096<UP,BROADCAST,MULTICAST> mtu 1500
        inet 192.168.17.20 netmask 255.255.255.0 broadcast 192.168.17.255
        ether 08:00:27:00:00:00
        txqueuelen 1000 (0 bytes)
        RX: 0 bytes 0 packets 0 errors 0 dropped 0 overruns 0 (0.0%
        TX: 0 bytes 0 packets 0 errors 0 dropped 0 overruns 0 (0.0%

team13@netlab-wb3pc3~$ ping 192.168.17.20
PING 192.168.17.20: 56(84) bytes of data:
64 bytes from 192.168.17.20: icmp_seq=1 ttl=64 time=0.218 ms
64 bytes from 192.168.17.20: icmp_seq=2 ttl=64 time=0.402 ms
64 bytes from 192.168.17.20: icmp_seq=3 ttl=64 time=0.489 ms
64 bytes from 192.168.17.20: icmp_seq=4 ttl=64 time=0.596 ms

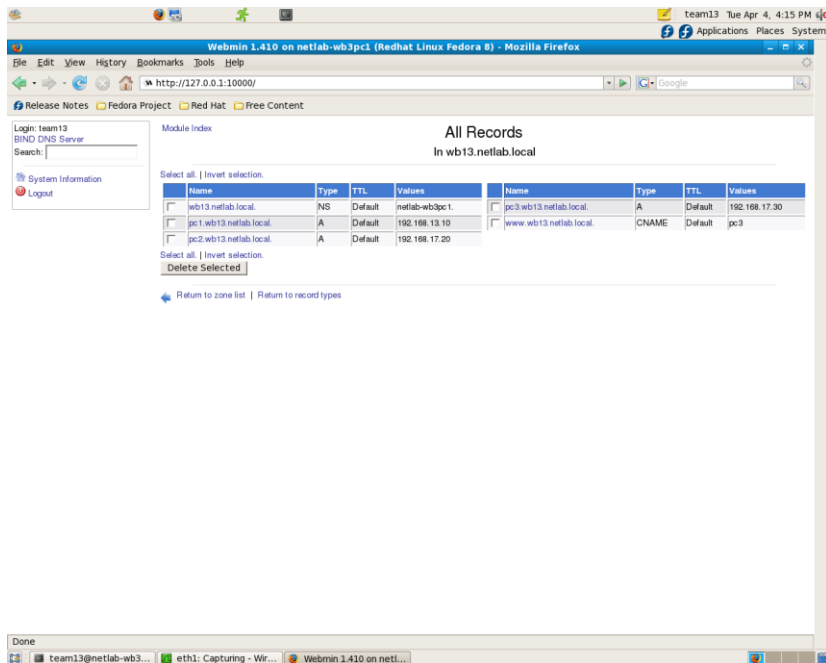
... 192.168.17.20 ping statistics ...
4 packets transmitted, 4 received, 0% packet loss, time 299ms
rtt min/avg/max/mdev = 0.218/0.422/0.596/0.121 ms

team13@netlab-wb3pc3~$ ping 192.168.13.10
PING 192.168.13.10: 56(84) bytes of data:
64 bytes from 192.168.13.10: icmp_seq=1 ttl=63 time=0.627 ms
64 bytes from 192.168.13.10: icmp_seq=2 ttl=63 time=0.612 ms
64 bytes from 192.168.13.10: icmp_seq=3 ttl=63 time=0.729 ms
64 bytes from 192.168.13.10: icmp_seq=4 ttl=63 time=0.618 ms

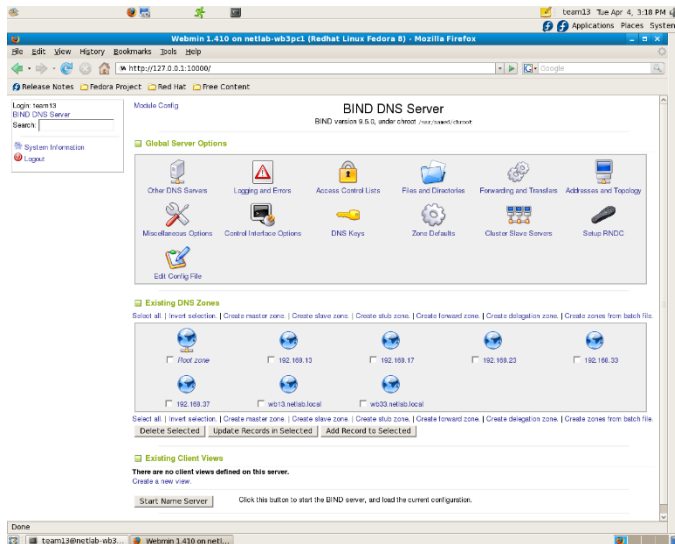
... 192.168.13.10 ping statistics ...
4 packets transmitted, 4 received, 0% packet loss, time 299ms
rtt min/avg/max/mdev = 0.612/0.646/0.729/0.054 ms
team13@netlab-wb3pc3~$
```

This is PC3 pinging PC1 and PC2.

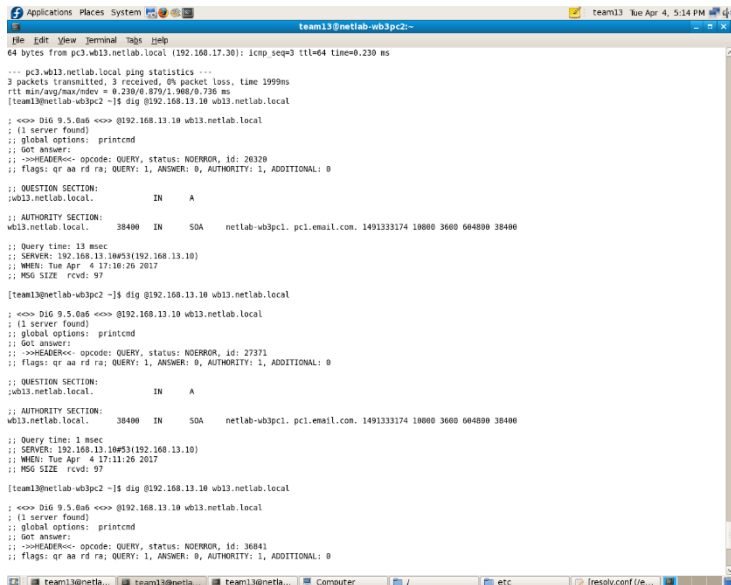
Section 4.2: DNS server configuration



These are the records we added to the forward lookup master zone.



This shows the reverse lookup master zones for 192.168.13 and 192.168.17. The other DNS zones that we did not create in this picture were deleted later.

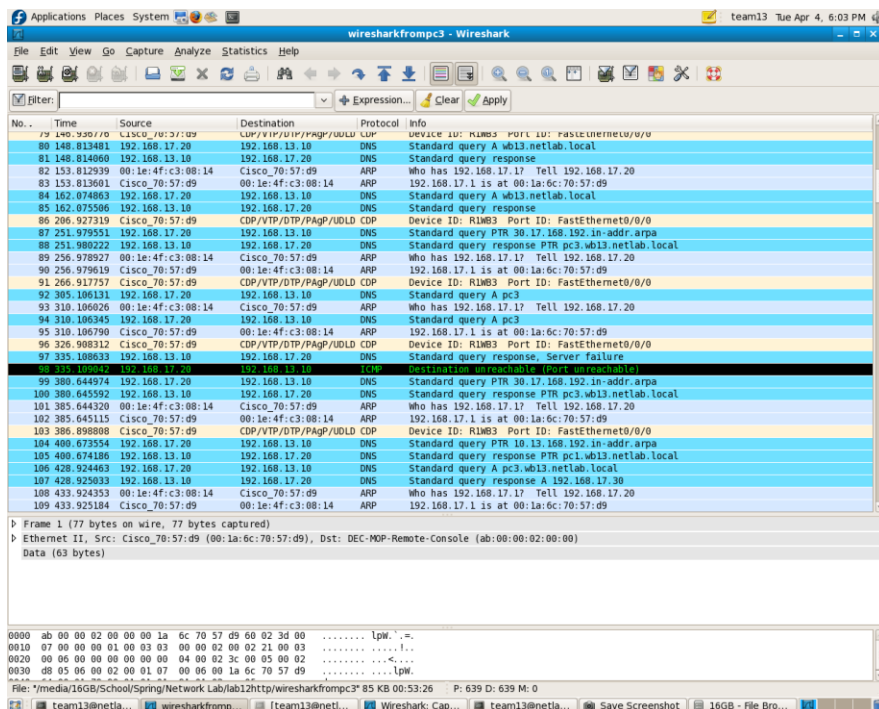


This is the dig result from executing the forward lookup query of the domain wb13.netlab.local.

```
team13@netlab-wb3pc2:~$ dig @192.168.13.10 -x 192.168.17.30
; <<> DiG 9.5.046 <<> @192.168.13.10 -x 192.168.17.30
; (1 server found)
;; global options: printcmd
;; Got answer:
;;->HEADER<- opcode: QUERY, status: NOERROR, id: 42966
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 0
;; QUESTION SECTION:
;30.17.168.192.in-addr.arpa.      IN      PTR
;; ANSWER SECTION:
30.17.168.192.in-addr.arpa. 38400 IN PTR pc3.wb13.netlab.local.
;; AUTHORITY SECTION:
17.168.192.in-addr.arpa. 38400 IN NS netlab-wb3pci.
;; Query time: 1 msec
;; SERVER: 192.168.13.10#53(192.168.13.10)
;; WHEN: Tue Apr 4 17:15:18 2017
;; MSG SIZE rcvd: 106

team13@netlab-wb3pc2:~$
```

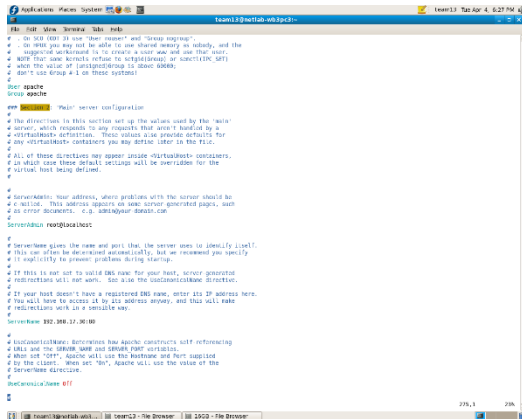
This shows the dig result from executing the reverse lookup query using PC3's IP address as 192.168.17.30.



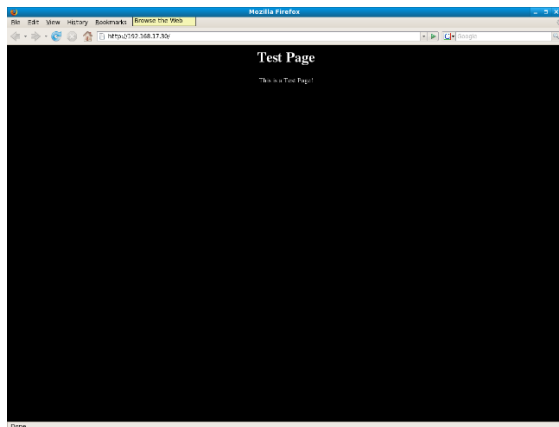
The image shows a Wireshark packet capture of DNS traffic. The packet list on the left shows several DNS queries and responses. The selected packet (No. 80) is a standard query from PC3 to the DNS server. The packet details pane shows the query for the PTR record of 192.168.17.30. The packet bytes pane shows the raw data of the query.

No.	Time	Source	Destination	Protocol	Info
79	140.350170	Cisco_70:57:d9	CPU/VTP/VDP/PagP/UDLD	CPU	Device ID: RUM83 Port ID: FastEthernet0/0/0
80	148.813481	192.168.17.20	192.168.13.10	DNS	Standard query A wb13.netlab.local
81	148.814060	192.168.13.10	192.168.17.20	DNS	Standard query response
82	153.812939	00:1e:4f:c3:08:14	Cisco_70:57:d9	ARP	Who has 192.168.17.1? Tell 192.168.17.20
83	153.813601	Cisco_70:57:d9	00:1e:4f:c3:08:14	ARP	192.168.17.1 is at 00:1a:6c:70:57:d9
84	162.074863	192.168.17.20	192.168.13.10	DNS	Standard query A wb13.netlab.local
85	162.075506	192.168.13.10	192.168.17.20	DNS	Standard query response
86	206.927319	Cisco_70:57:d9	CDP/VTP/DTP/PagP/UDLD	CDP	Device ID: RUM83 Port ID: FastEthernet0/0/0
87	251.979551	192.168.17.20	192.168.13.10	DNS	Standard query PTR 30.17.168.192.in-addr.arpa
88	251.980222	192.168.13.10	192.168.17.20	DNS	Standard query response PTR pc3.wb13.netlab.local
89	256.978927	00:1e:4f:c3:08:14	Cisco_70:57:d9	ARP	Who has 192.168.17.1? Tell 192.168.17.20
90	256.979619	Cisco_70:57:d9	00:1e:4f:c3:08:14	ARP	192.168.17.1 is at 00:1a:6c:70:57:d9
91	266.917757	Cisco_70:57:d9	CDP/VTP/DTP/PagP/UDLD	CDP	Device ID: RUM83 Port ID: FastEthernet0/0/0
92	305.106131	192.168.17.20	192.168.13.10	DNS	Standard query A pc3
93	310.106026	00:1e:4f:c3:08:14	Cisco_70:57:d9	ARP	Who has 192.168.17.1? Tell 192.168.17.20
94	310.106345	192.168.17.20	192.168.13.10	DNS	Standard query A pc3
95	310.106790	Cisco_70:57:d9	00:1e:4f:c3:08:14	ARP	192.168.17.1 is at 00:1a:6c:70:57:d9
96	326.908312	Cisco_70:57:d9	CDP/VTP/DTP/PagP/UDLD	CDP	Device ID: RUM83 Port ID: FastEthernet0/0/0
97	335.108633	192.168.13.10	192.168.17.20	DNS	Standard query response, Server failure
98	335.109042	192.168.17.20	192.168.13.10	ICMP	Destination unreachable (Port unreachable)
99	380.644974	192.168.17.20	192.168.13.10	DNS	Standard query PTR 30.17.168.192.in-addr.arpa
100	380.645592	192.168.13.10	192.168.17.20	DNS	Standard query response PTR pc3.wb13.netlab.local
101	385.644320	00:1e:4f:c3:08:14	Cisco_70:57:d9	ARP	Who has 192.168.17.1? Tell 192.168.17.20
102	385.645115	Cisco_70:57:d9	00:1e:4f:c3:08:14	ARP	192.168.17.1 is at 00:1a:6c:70:57:d9
103	386.898808	Cisco_70:57:d9	CDP/VTP/DTP/PagP/UDLD	CDP	Device ID: RUM83 Port ID: FastEthernet0/0/0
104	400.673554	192.168.17.20	192.168.13.10	DNS	Standard query PTR 10.13.168.192.in-addr.arpa
105	400.674106	192.168.13.10	192.168.17.20	DNS	Standard query response PTR pc1.wb13.netlab.local
106	428.924463	192.168.17.20	192.168.13.10	DNS	Standard query A pc3.wb13.netlab.local
107	428.925033	192.168.13.10	192.168.17.20	DNS	Standard query response A 192.168.17.30
108	433.924353	00:1e:4f:c3:08:14	Cisco_70:57:d9	ARP	Who has 192.168.17.1? Tell 192.168.17.20
109	433.925184	Cisco_70:57:d9	00:1e:4f:c3:08:14	ARP	192.168.17.1 is at 00:1a:6c:70:57:d9

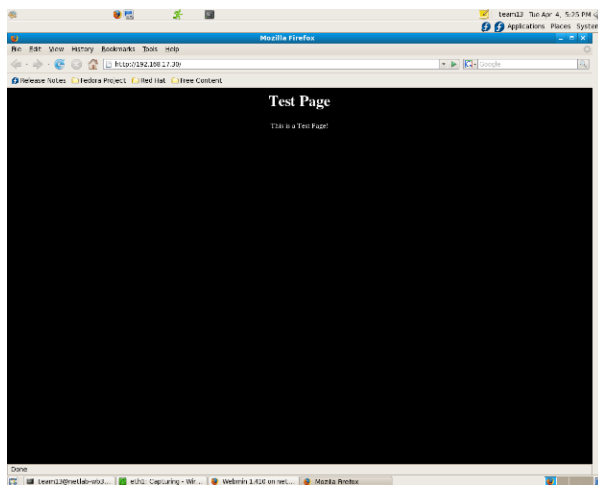
This shows the Wireshark output when we did forward and reverse lookup queries for the DNS server.



This shows changing the ServerName directive in the /etc/httpd/conf/http.conf.

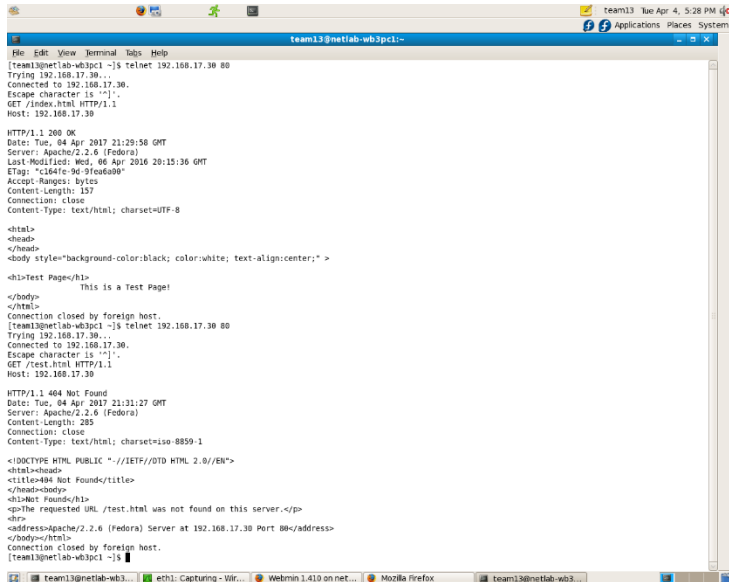


This shows PC2 accessing the web page hosted on the PC3's web server.



This shows PC1 accessing the web page hosted on the PC3's web server.

Section 4.4: Browse without browser



```
team13@netlab-wb3pci~$ telnet 192.168.17.30 80
Trying 192.168.17.30...
Connected to 192.168.17.30.
Escape character is '^['.
GET /index.html HTTP/1.1
Host: 192.168.17.30

HTTP/1.1 200 OK
Date: Tue, 04 Apr 2017 21:29:58 GMT
Server: Apache/2.2.6 (Fedora)
Last-Modified: Wed, 04 Apr 2016 20:15:36 GMT
ETag: "c164fe-9d-9fe6600"
Accept-Ranges: bytes
Content-Length: 157
Connection: close
Content-Type: text/html; charset=UTF-8

<html>
<head>
</head>
<body style="background-color:black; color:white; text-align:center;" >

<h1>Test Page</h1>
This is a Test Page!
</body>
</html>
Connection closed by foreign host.
team13@netlab-wb3pci~$ telnet 192.168.17.30 80
Trying 192.168.17.30...
Connected to 192.168.17.30.
Escape character is '^['.
GET /test.html HTTP/1.1
Host: 192.168.17.30

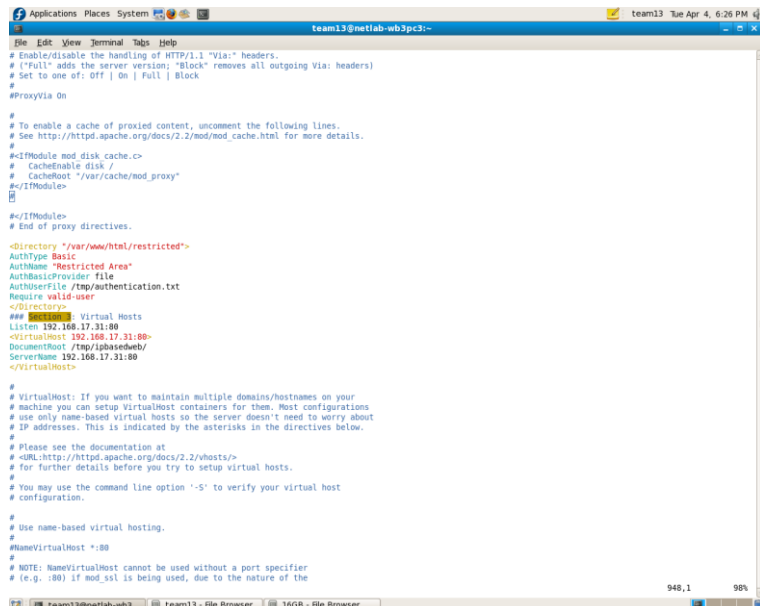
HTTP/1.1 404 Not Found
Date: Tue, 04 Apr 2017 21:31:27 GMT
Server: Apache/2.2.6 (Fedora)
Content-Length: 285
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html>
<head>
<title>404 Not Found</title>
</head>
<body>
<h1>Not Found</h1>
<p>The requested URL /test.html was not found on this server.</p>
<hr/>
<p>Address: Apache/2.2.6 (Fedora) Server at 192.168.17.30 Port 80</p>
</body>
</html>
Connection closed by foreign host.
team13@netlab-wb3pci~$
```

This shows the telnet result from PC1. The top result was successful because it used the file `index.html` in the GET command, which was located in the `/var/www/html` directory. The second result failed because it used the file `test.html` and there was no listing of that in the `/var/www/html` directory. It throws a 404 Not Found error when it can not find the file and correct HTTP header.

Section 4.5: Virtual host configuration

After adding another IP address to PC3 and modifying the `httpd` configuration file after the Section 3: Virtual Hosts section, we could test whether we can access that web page running on the web server.



```
team13@netlab-wb3pci~$ cat /etc/httpd/conf/httpd.conf
# Enable/disable the handling of HTTP/1.1 "Via:" headers.
# "Full" adds the server version; "Block" removes all outgoing Via: headers)
# Set to one of: Off | On | Full | Block
#
#ProxyVia On

# To enable a cache of proxied content, uncomment the following lines.
# See http://httpd.apache.org/docs/2.2/mod/mod_cache.html for more details.
#
<#Module mod_cache.c>
# Cacheable disk /
# CacheRoot "/var/cache/mod_proxy"
#</Module>

<#IfModule>
# End of proxy directives.

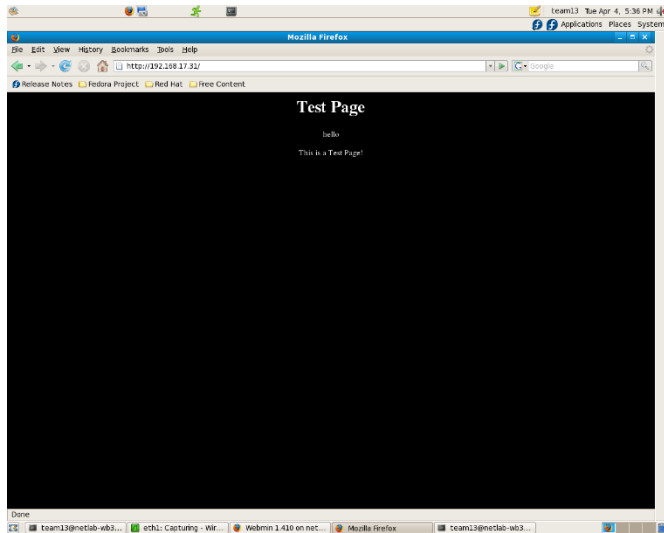
<Directory "/var/www/html/restricted">
AuthType Basic
AuthName "Restricted Area"
AuthBasicProvider file
AuthUserFile /tmp/authentication.txt
Require valid-user
</Directory>

### BEGIN SSL Virtual Hosts
Listen 192.168.17.31:80
<VirtualHost 192.168.17.31:80>
DocumentRoot /tmp/localhostweb/
ServerName 192.168.17.31:80
</VirtualHost>

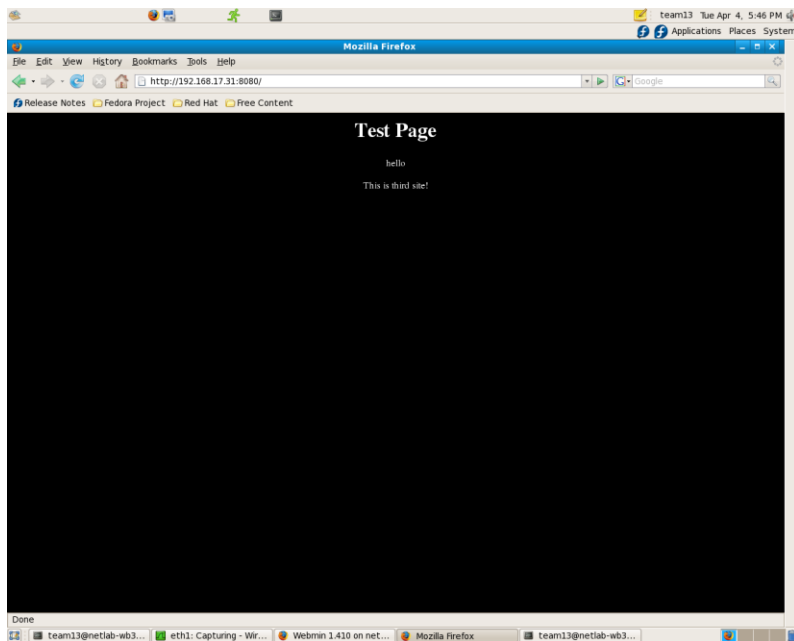
# VirtualHost: If you want to maintain multiple domains/hostnames on your
# machine you can setup VirtualHost containers for them. Most configurations
# use only name-based virtual hosts so the server doesn't need to worry about
# IP addresses. This is indicated by the asterisks in the directives below.
#
# Please see the documentation at
# <URL>http://httpd.apache.org/docs/2.2/vhosts</URL>
# for further details before you try to setup virtual hosts.
#
# You may use the command line option '-S' to verify your virtual host
# configuration.

#
# Use name-based virtual hosting.
#
#NameVirtualHost *:80
#
# NOTE: NameVirtualHost cannot be used without a port specifier
# (e.g. :80) if mod_ssl is being used, due to the nature of the
```

This shows the modifications of the Section 3 Virtual Hosts section in the httpd configuration file.



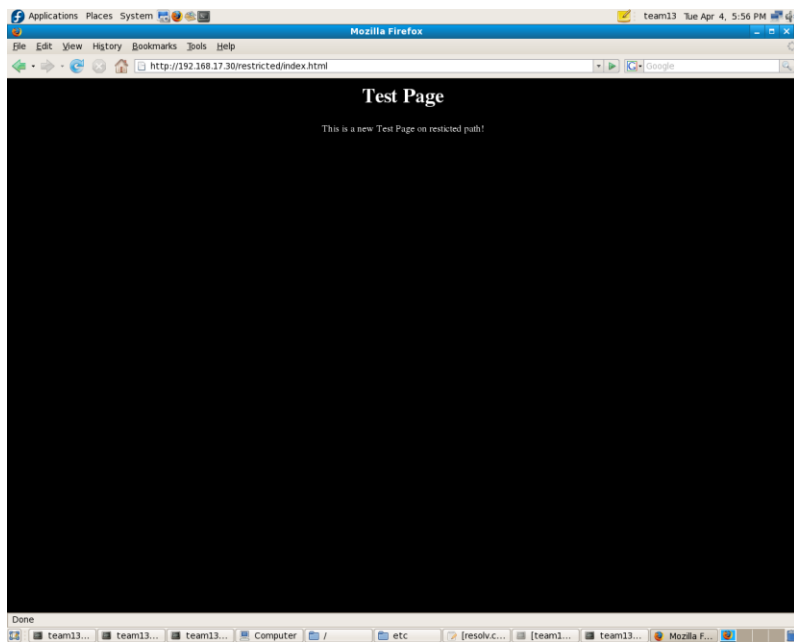
This is PC1 accessing the new website at 192.178.17.31.



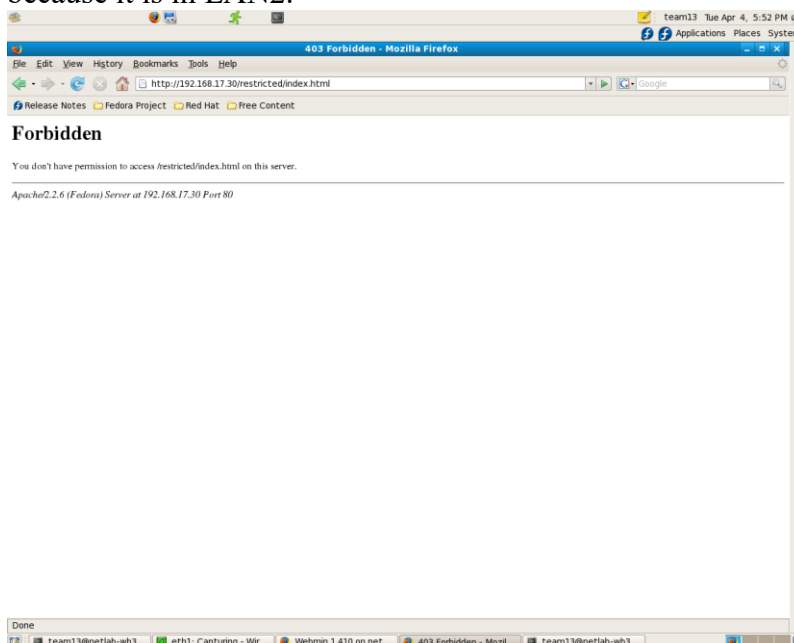
This is PC1 accessing the third website using the same IP address as before 192.168.17.31, but now using port 8080 instead of port 80.

Section 4.6: Access control and basic authentication

After creating the `/var/www/html/restricted` directory and modifying the `httpd` configuration file after the Section 3: Virtual Hosts section to allow any devices in LAN2 to access the website but deny all other devices in other LANs, we could test if the rules we set worked.



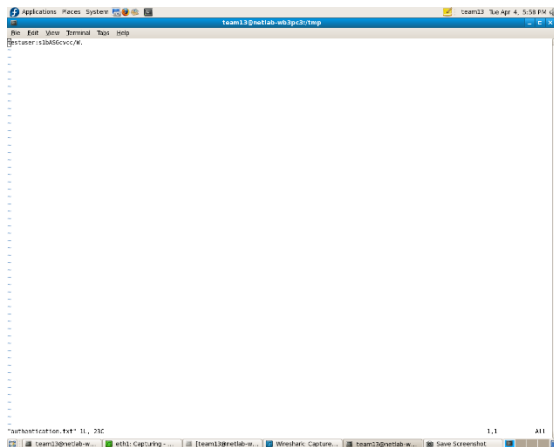
This is PC2 accessing the website at `192.168.17.30/restricted/index.html` and it is allowed because it is in LAN2.



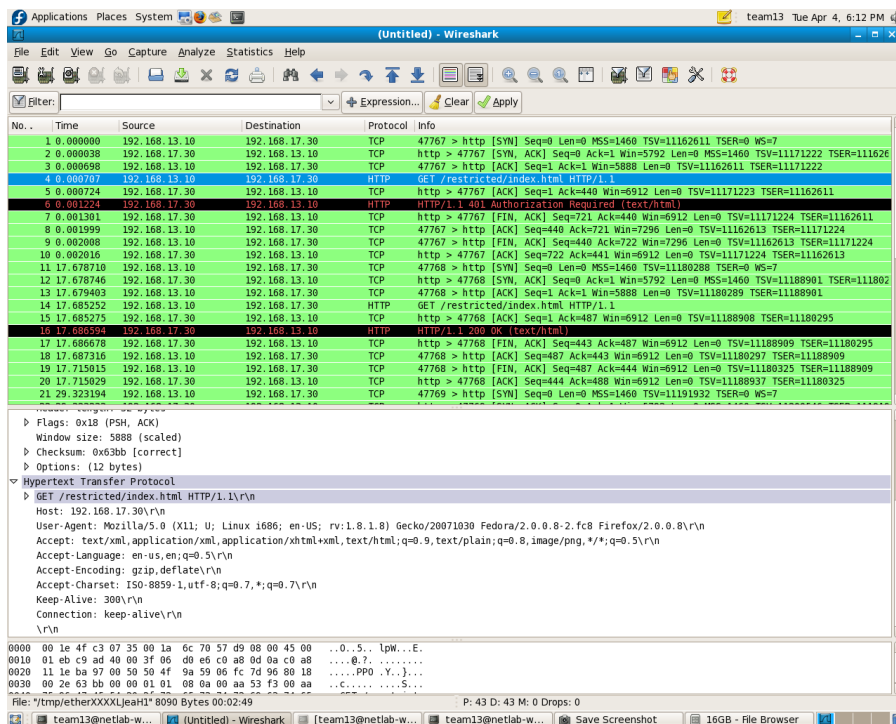
This is PC1 accessing the website at `192.168.17.30/restricted/index.html` and it is not allowed because it is in LAN1 and not LAN2.

Instead of forbidding and allowing certain users, we can allow users to authenticate themselves in order to prove whether they can access the website or not. Using `htpasswd` to create a user and

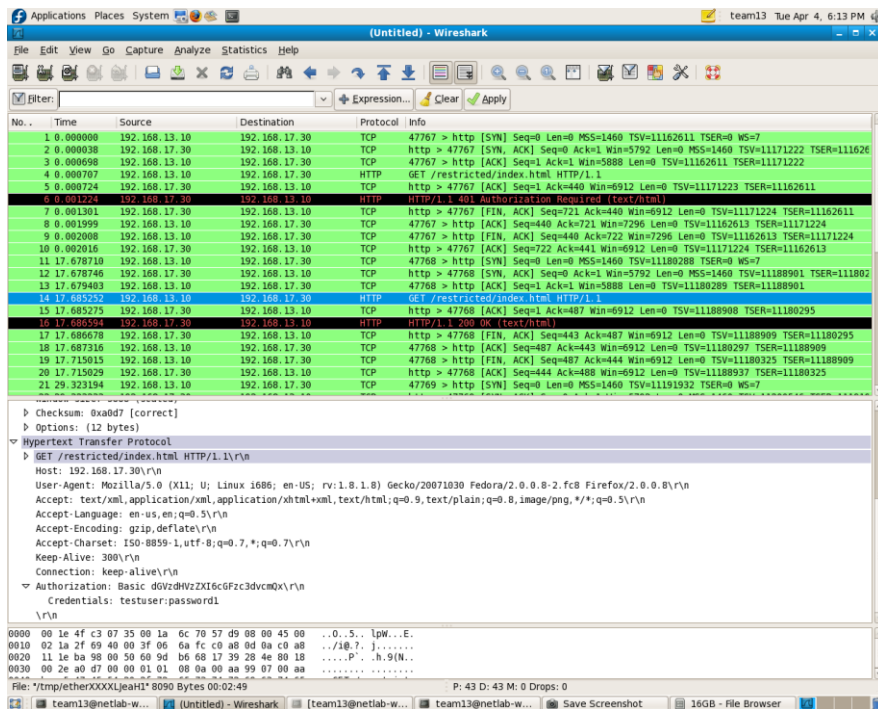
password file in order to access the website and telling the httpd configuration file where to look in order to authenticate the user input, we could verify the test results.



This shows the encrypted password of the user testuser we created.



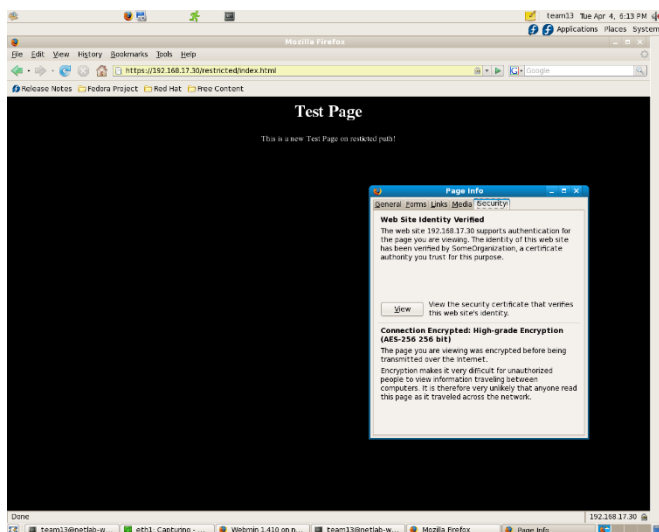
This shows the Wireshark output when PC1 attempted to access the website after entering in the proper credentials. This is before we entered in any credentials and it does not show any authentication in the GET message.



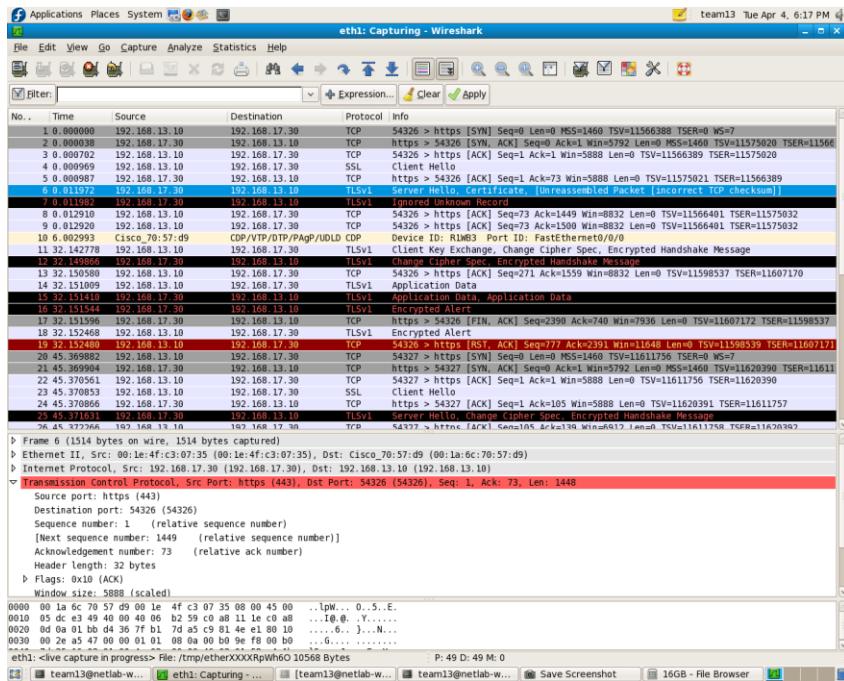
It shows that authentication is required and once the correct username and password were entered, it would be allowed access even though it is in LAN1. It also shows the credentials under the authorization in plaintext with the username: testuser and the password: password1.

Section 4.7: Using https

We changed the configuration file in /etc/httpd/conf.d/ssl.conf to include the IP address of PC3 192.168.17.30 under the Listen directive and to use port 443, which is the default port for secure http access.



This shows PC1 accessing the website using https after entering in the credentials. It shows that it is using AES 256 bit encryption when accessing the website.



This shows the Wireshark output when accessing the website from PC1 using https. It shows the protocols TLS and SSL. It shows the TLS handshake when the client first sends the Hello message. The server will respond with a Hello message acknowledging that it received the client's message. The certificate and key exchange will be established between PC1 and the web server. Then the server will send a Hello message back saying it is done with setting up the certificate and encryption. After this, the data can be transferred and PC1 can access the website from PC3's web server.

R1WB3#show run
Building configuration...

Current configuration : 1149 bytes

!

version 12.4

service timestamps debug datetime msec

service timestamps log datetime msec

no service password-encryption

!

hostname R1WB3

!

boot-start-marker

boot system flash:c2800nm-adventerprisek9-mz.124-24.T8.bin

boot-end-marker

!

logging message-counter syslog

!

```
no aaa new-model
!
!
!
dot11 syslog
ip source-route
!
!
ip cef
!
!
no ip domain lookup
no ipv6 cef
!
multilink bundle-name authenticated
!
!

!
voice-card 0
!

!
!
archive
 log config
  hidekeys
!
!
interface Loopback0
 no ip address
!
interface FastEthernet0/0
 ip address 192.168.13.1 255.255.255.0
 ip virtual-reassembly
 duplex auto
 speed auto
!
interface FastEthernet0/1
 no ip address
 ip virtual-reassembly
 duplex auto
 speed auto
!
interface FastEthernet0/0/0
 ip address 192.168.17.1 255.255.255.0
```

```
ip virtual-reassembly
duplex auto
speed auto
!
no ip forward-protocol nd
no ip http server
no ip http secure-server
!
!
!
ip access-list extended NP
!

!
control-plane
!
!
!
line con 0
  exec-timeout 0 0
  logging synchronous
line aux 0
line vty 0 4
  login
!
scheduler allocate 20000 1000
end

R1WB3#
```

This is the show run output for R1.