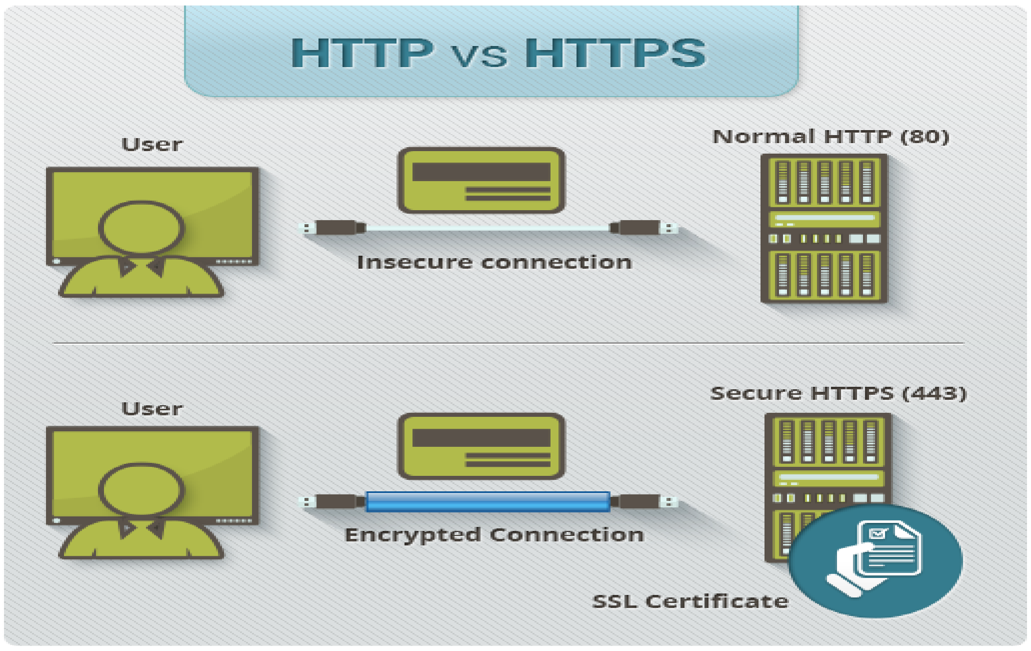
NATIONAL UNIVERSITY OF COMPUTER & EMERGING SCIENCE
Computer Network Lab (CL-307)
Lab Session 07

Application Layer Protocol

Hypertext Transfer Protocol (HTTP) is a protocol used in networking. When you type any web address in your web browser, your browser acts as a client, and the computer having the requested information acts as a server. When client requests for any information from the server, it uses HTTP protocol to do so. The server responds back to the client after the request completes. The response comes in the form of web page which you see just after typing the web address and press “Enter”.

Hypertext Transfer Protocol Secure (HTTPS) is a combination of two different protocols. It is more secure way to access the web. It is combination of Hypertext Transfer Protocol (HTTPS) and SSL/TLS protocol. It is more secure way to sending request to server from a client, also the communication is purely encrypted which means no one can know what you are looking for. This kind of communication is used for accessing those websites where security is required. Banking websites, payment gateway, emails (Gmail offers HTTPS by default in Chrome browser), and corporate sector websites are some great examples where HTTPS protocols are used.

For HTTPS connection, public key trusted and signed certificate is required for the server. These certificate comes either free or it costs few dollars depends on the signing authority. There is one other method for distributing certificates. Site admin creates certificates and loads in the browser of users. Now when user requests information to the web server, his identity can be verified easily.



Here are some major differences between HTTP and HTTPS:

HTTP	HTTPS
URL begins with “http://”	URL begins with “https://”
It uses port 80 for communication	It uses port 443 for communication
Unsecured	Secured
Operates at Application Layer	Operates at Transport Layer
No encryption	Encryption is present
No certificates required	Certificates required

Client Error

The 4xx class of status code is intended for cases in which the client seems to have erred. Except when responding to a HEAD request, the server *should* include an entity containing an explanation of the error situation, and whether it is a temporary or permanent condition. These status codes are applicable to any request method. User agents *should* display any included entity to the user.

400 Bad Request

The server cannot or will not process the request due to something that is perceived to be a client error (e.g., malformed request syntax, invalid request message framing, or deceptive request routing).

401 Unauthorized (RFC 7235)

Similar to 403 Forbidden, but specifically for use when authentication is required and has failed or has not yet been provided. The response must include a WWW-Authenticate header field containing a challenge applicable to the requested resource. See Basic access authentication and Digest access authentication.

403 Forbidden

The request was a valid request, but the server is refusing to respond to it. Unlike a 401 unauthorized response, authenticating will make no difference.

404 Not Found

The requested resource could not be found but may be available again in the future. Subsequent requests by the client are permissible.

408 Request Timeout

The server timed out waiting for the request. According to HTTP specifications: "The client did not produce a request within the time that the server was prepared to wait. The client MAY repeat the request without modifications at any later time."

Server Error

The server failed to fulfill an apparently valid request.

Response status codes beginning with the digit "5" indicate cases in which the server is aware that it has encountered an error or is otherwise incapable of performing the request. Except when responding to a HEAD request, the server *should* include an entity containing an explanation of the error situation, and indicate whether it is a temporary or permanent condition. Likewise, user agents *should* display any included entity to the user. These response codes are applicable to any request method.

500 Internal Server Error

A generic error message, given when an unexpected condition was encountered and no more specific message is suitable.

501 Not Implemented

The server either does not recognize the request method, or it lacks the ability to fulfil the request. Usually this implies future availability (e.g., a new feature of a web-service API).

502 Bad Gateway

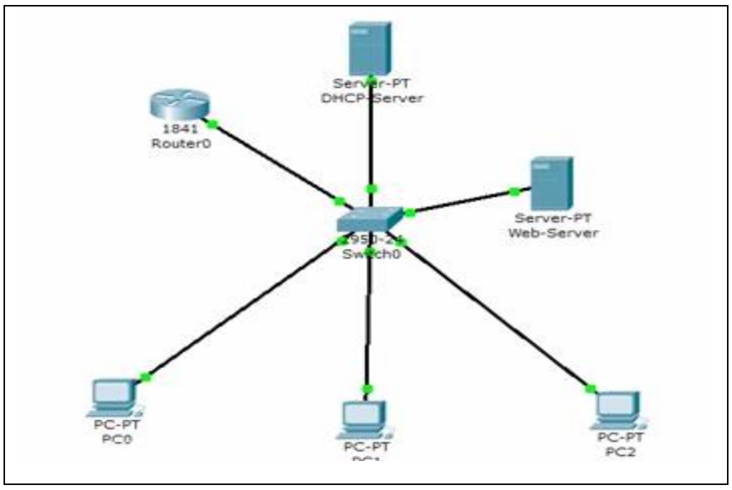
The server was acting as a gateway or proxy and received an invalid response from the upstream server.

503 Service Unavailable

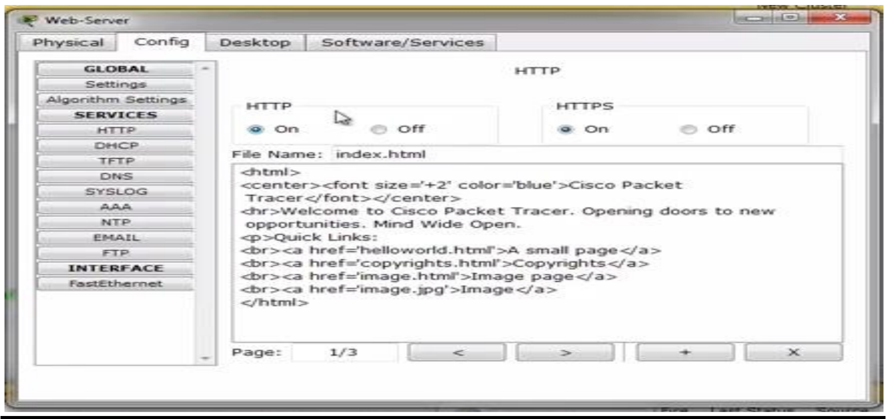
The server is currently unavailable (because it is overloaded or down for maintenance). Generally, this is a temporary state.

IMPLEMENTION OF HTTP and HTTPS:

Consider the below topology with web-server IP 192.168.1.252



Click on the web server,go to config --->services---→HTTP.Here you can see that http and https server are on.

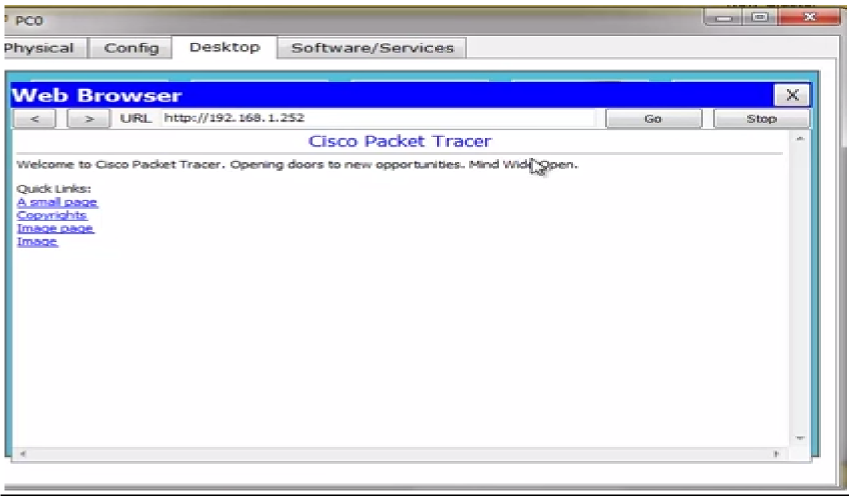


HTTP: On [✓] Off
HTTPS: On [✓] Off

File Name: index.html

```
<html>
<center><font size='+2' color='blue'>Cisco Packet
 Tracer</font></center>
<hr>Welcome to Cisco Packet Tracer. Opening doors to new
 opportunities. Mind Wide Open.
<p>Quick Links:
<br><a href='helloworld.html'>A small page</a>
<br><a href='copyrights.html'>Copyrights</a>
<br><a href='image.html'>Image page</a>
<br><a href='image.jpg'>Image</a>
</html>
```

Now click on PC0 and go to Desktop→Web Browser. Now type web-server IP 192.168.1.252



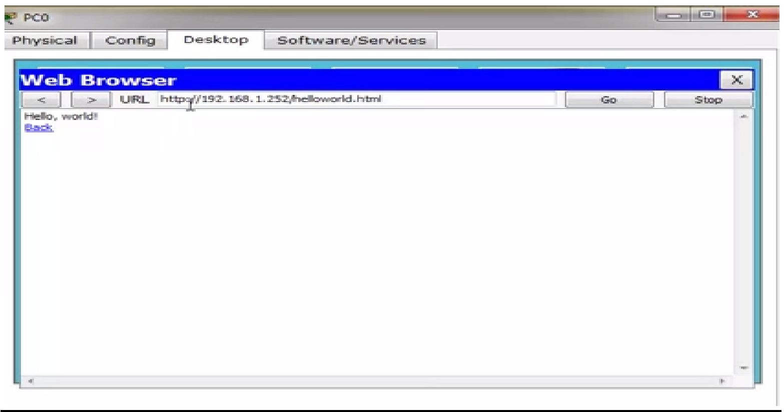
URL: http://192.168.1.252

Cisco Packet Tracer

Welcome to Cisco Packet Tracer. Opening doors to new opportunities. Mind Wide Open.

Quick Links:
A small page
Copyrights
Image page
Image

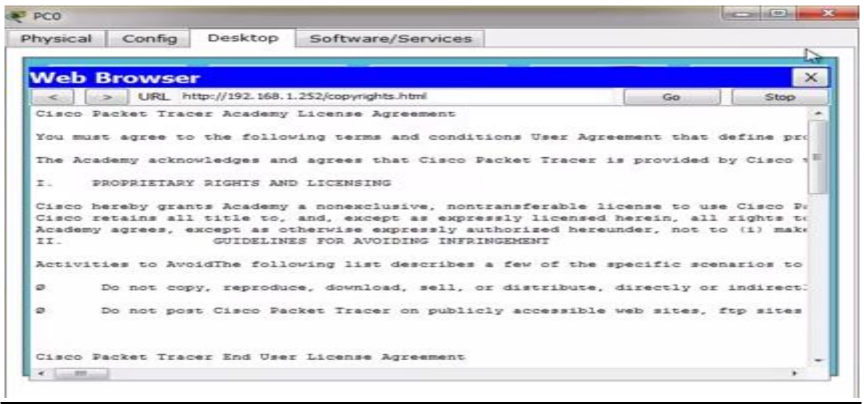
<u>Now click on a small page option on a web page you can note the URL that we move to the next page.</u>



http://192.168.1.252/helloworld.html

Hello, world!
Back

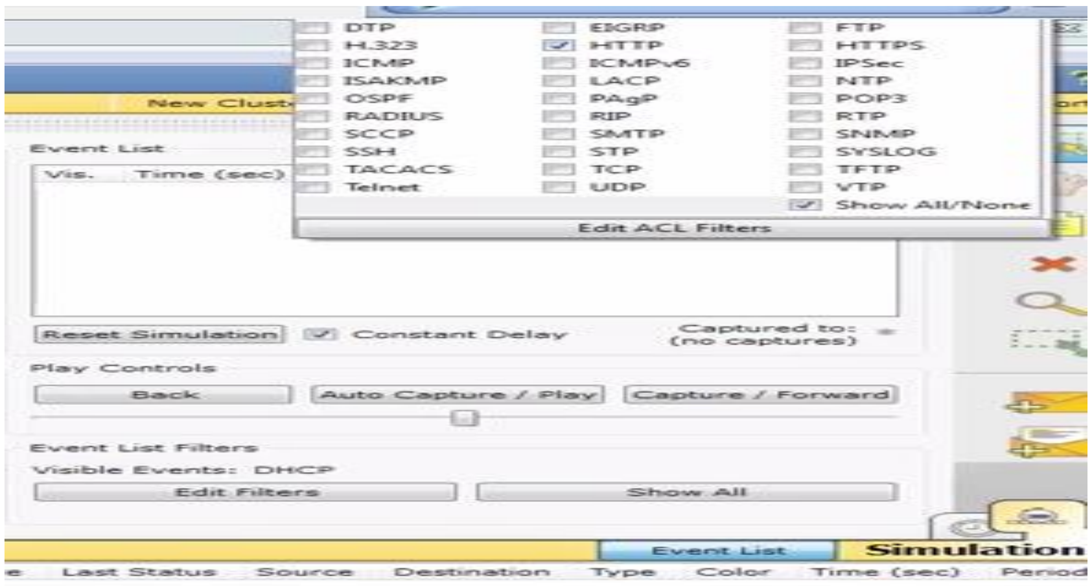
<u>Now click on a Copy rights option on a web page</u>



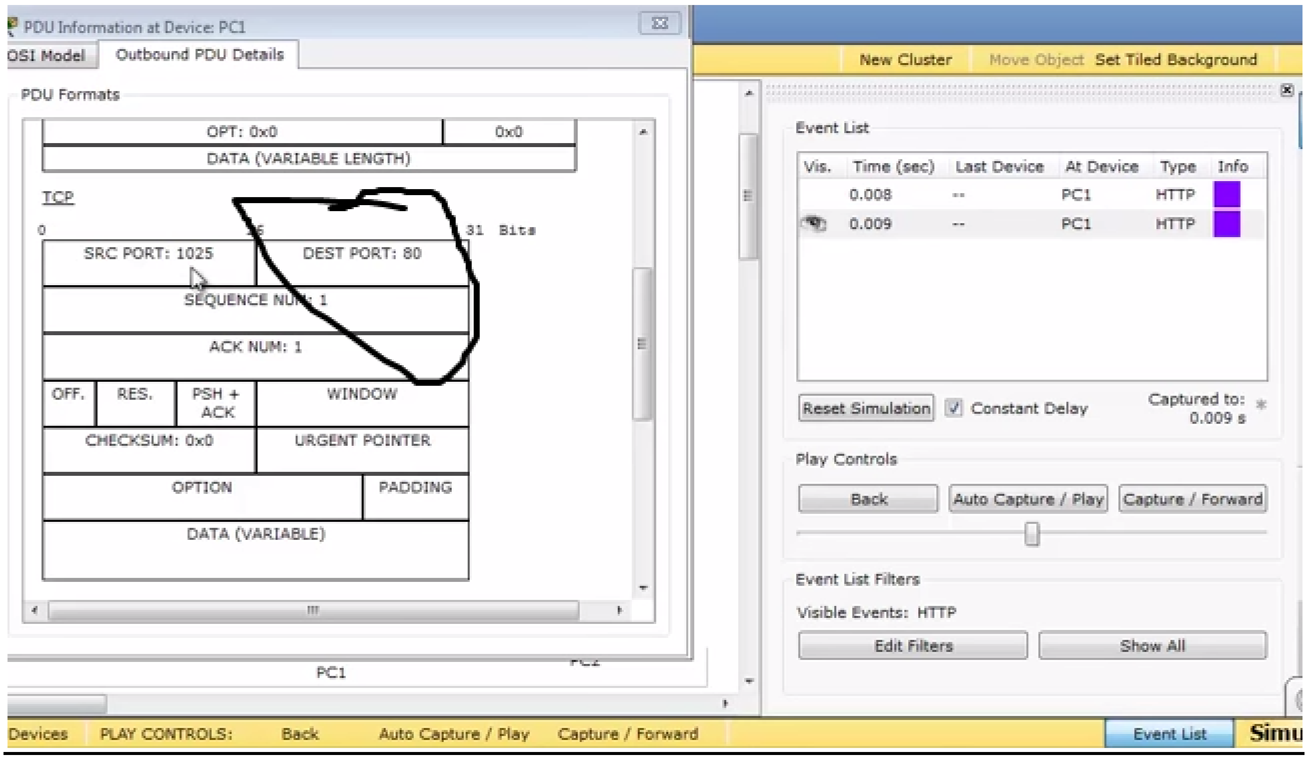
http://192.168.1.252/copyrights.html

```
Cisco Packet Tracer Academy License Agreement

You must agree to the following terms and conditions User Agreement that define pro

The Academy acknowledges and agrees that Cisco Packet Tracer is provided by Cisco t

I.      PROPRIETARY RIGHTS AND LICENSING

Cisco hereby grants Academy a nonexclusive, nontransferable license to use Cisco Pa
Cisco retains all title to, and, except as expressly licensed herein, all rights to
Academy agrees, except as otherwise expressly authorized hereunder, not to (i) make
II.                 GUIDELINES FOR AVOIDING INFRINGEMENT

Activities to AvoidThe following list describes a few of the specific scenarios to

Ø       Do not copy, reproduce, download, sell, or distribute, directly or indirectl

Ø       Do not post Cisco Packet Tracer on publicly accessible web sites, ftp sites


Cisco Packet Tracer End User License Agreement
```

<u>Now to note the http header format information go to simulation mode →edit filters and click on http check box then click on capture/forward button.</u>



Now click on the http packet, you can note that the destination port is 80.



PDU Information at Device: PC1

OSI Model | Outbound PDU Details

PDU Formats

OPT: 0x0	0x0
DATA (VARIABLE LENGTH)	

TCP

0 ... 16 ... 31 Bits

SRC PORT: 1025		DEST PORT: 80	
SEQUENCE NUM: 1			
ACK NUM: 1			
OFF.	RES.	PSH + ACK	WINDOW
CHECKSUM: 0x0		URGENT POINTER	
OPTION			PADDING
DATA (VARIABLE)			

Event List

Vis.	Time (sec)	Last Device	At Device	Type	Info
	0.008	--	PC1	HTTP	🟪
👁	0.009	--	PC1	HTTP	🟪

Reset Simulation ✓ Constant Delay Captured to: 0.009 s

Play Controls

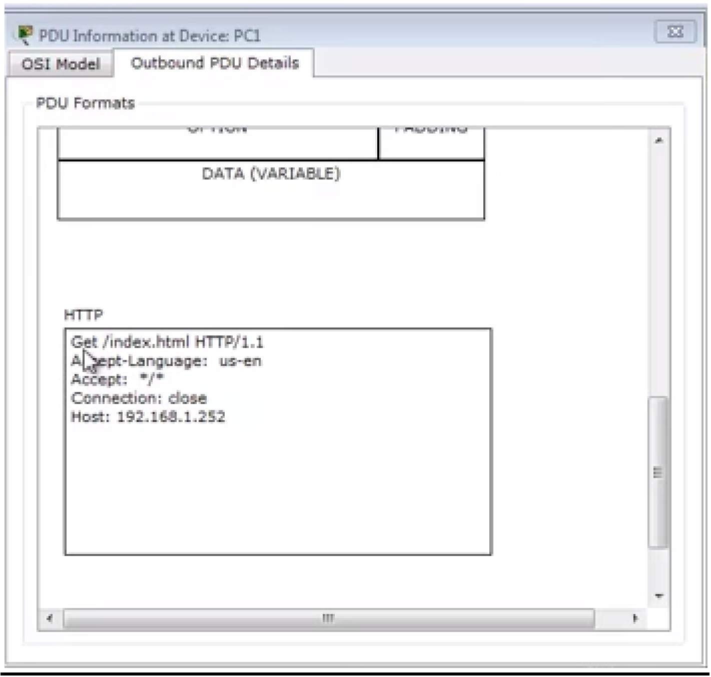
Back | Auto Capture / Play | Capture / Forward

Event List Filters

Visible Events: HTTP

Edit Filters | Show All

Now scroll the Outbound PDU Details, you can see the http protocol information.



PDU Information at Device: PC1

OSI Model | Outbound PDU Details

PDU Formats

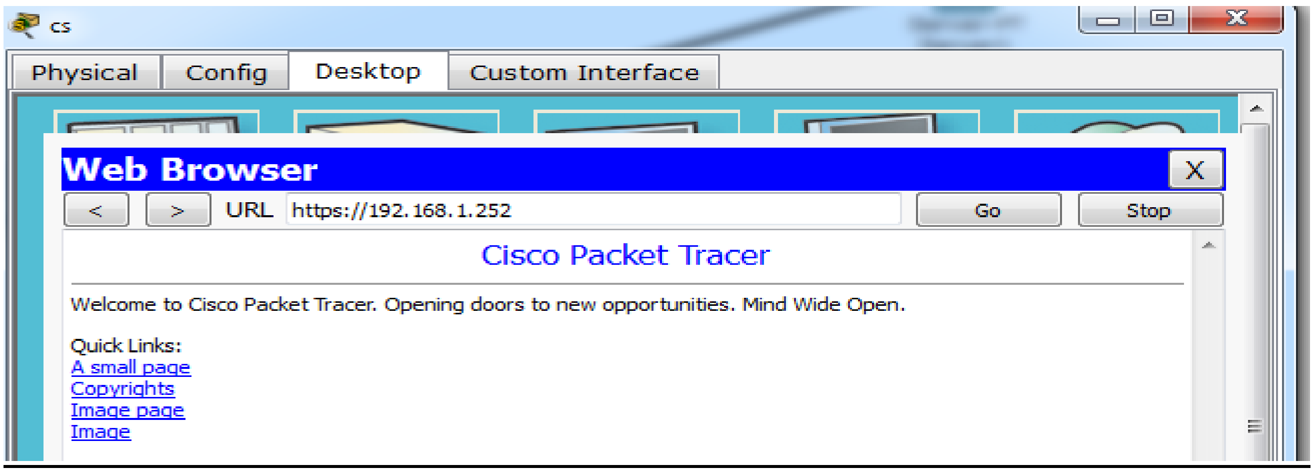
DATA (VARIABLE)

HTTP

```
Get /index.html HTTP/1.1
Accept-Language:  us-en
Accept:  */*
Connection: close
Host: 192.168.1.252
```

For HTTPS:

Now click on PC and go to Desktop→Web Browser. Now type web-server IP 192.168.1.252



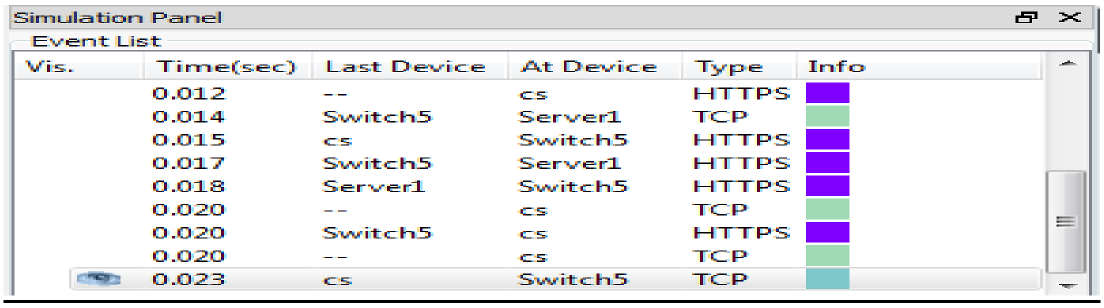
URL https://192.168.1.252

Cisco Packet Tracer

Welcome to Cisco Packet Tracer. Opening doors to new opportunities. Mind Wide Open.

Quick Links:
A small page
Copyrights
Image page
Image

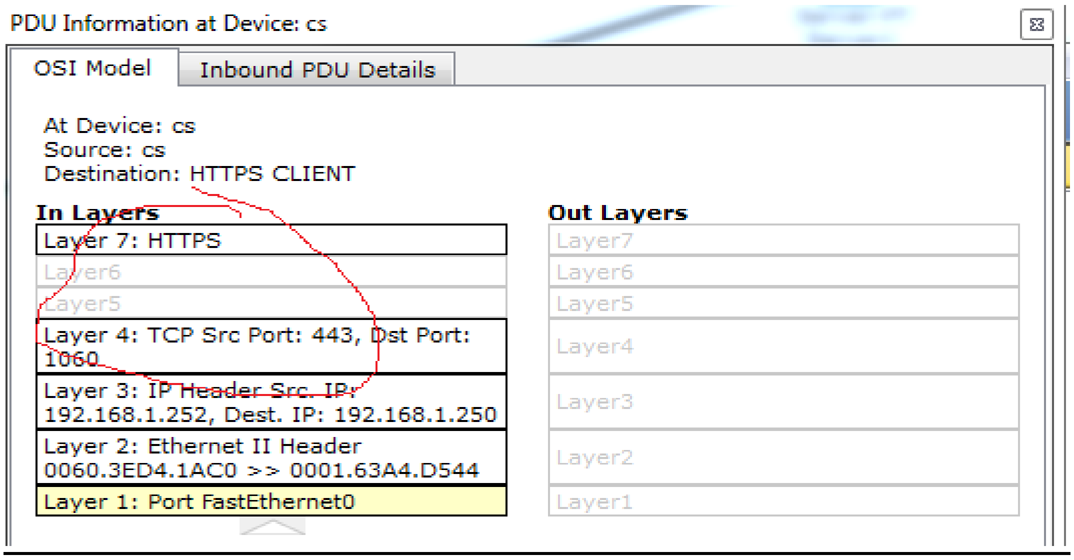
Now to note the https header format information go to simulation mode →edit filters and click on https check box then click on capture/forward button.


Simulation Panel

Event List

Vis.	Time(sec)	Last Device	At Device	Type	Info
	0.012	--	cs	HTTPS	🟪
	0.014	Switch5	Server1	TCP	🟩
	0.015	cs	Switch5	HTTPS	🟪
	0.017	Switch5	Server1	HTTPS	🟪
	0.018	Server1	Switch5	HTTPS	🟪
	0.020	--	cs	TCP	🟩
	0.020	Switch5	cs	HTTPS	🟪
	0.020	--	cs	TCP	🟩
👁	0.023	cs	Switch5	TCP	🟦

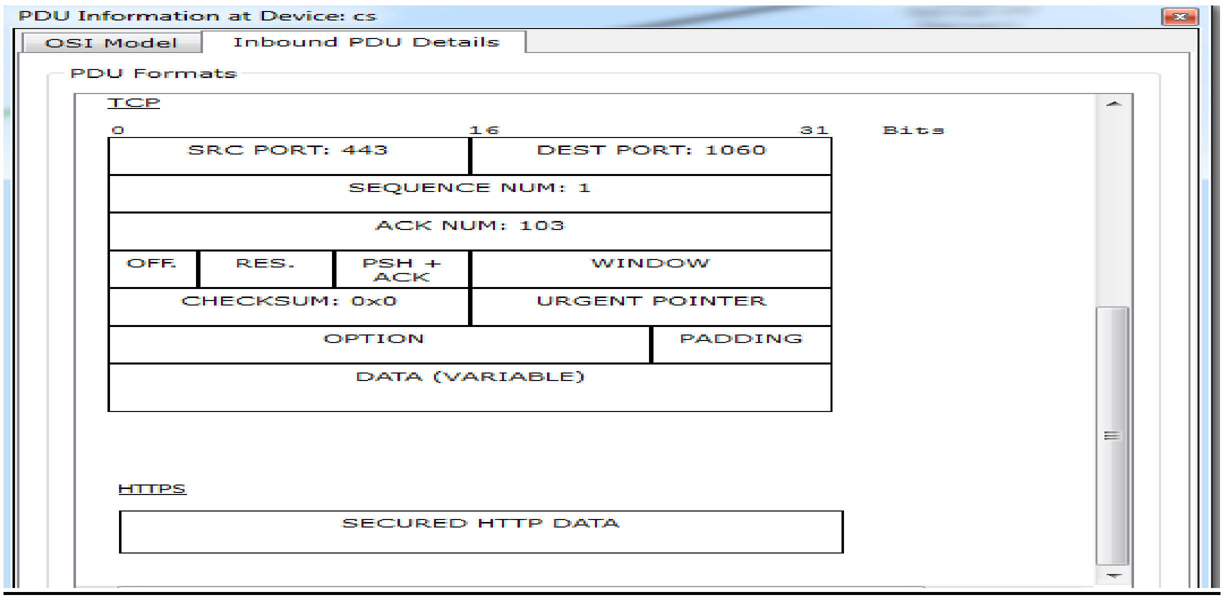
Now click on the https packet, you can note that the destination port is 443.


PDU Information at Device: cs

At Device: cs
Source: cs
Destination: HTTPS CLIENT

In Layers	Out Layers
Layer 7: HTTPS	Layer7
Layer6	Layer6
Layer5	Layer5
Layer 4: TCP Src Port: 443, Dst Port: 1060	Layer4
Layer 3: IP Header Src. IP: 192.168.1.252, Dest. IP: 192.168.1.250	Layer3
Layer 2: Ethernet II Header 0060.3ED4.1AC0 >> 0001.63A4.D544	Layer2
Layer 1: Port FastEthernet0	Layer1

Now scroll the Outbound PDU Details, you can see the https PDU.


PDU Information at Device: cs

OSI Model | Inbound PDU Details

PDU Formats

TCP

0		16	31 Bits
SRC PORT: 443		DEST PORT: 1060	
SEQUENCE NUM: 1			
ACK NUM: 103			
OFF.	RES.	PSH + ACK	WINDOW
CHECKSUM: 0x0		URGENT POINTER	
OPTION			PADDING
DATA (VARIABLE)			

HTTPS

SECURED HTTP DATA

File Transfer Protocol

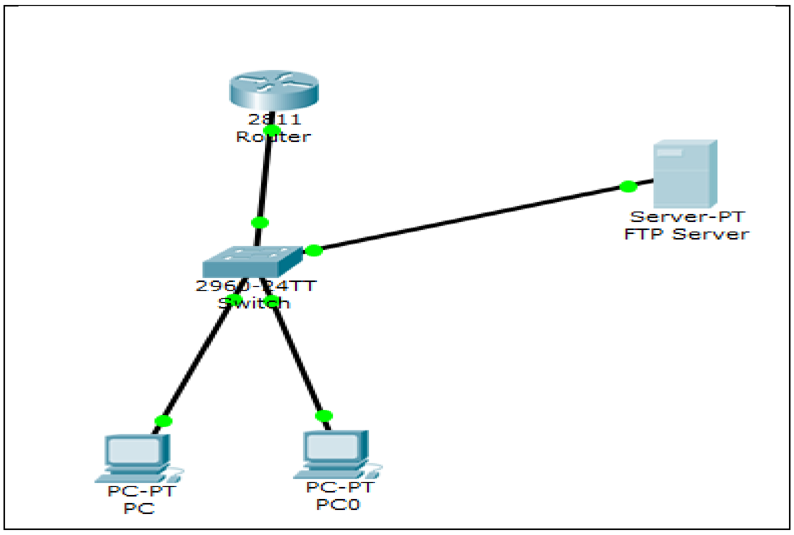
The File Transfer Protocol (FTP) is a standard network protocol used to transfer computer files between a client and server on a computer network. FTP is built on a client-server model architecture and uses separate control and data connections between the client and the server.

Objectives: In this activity, you will configure FTP services. You will then use the FTP services to transfer files between clients and the server.

Part 1: Configure FTP Services on Servers

Part 2: Upload a File to the FTP Server

Part 3: Download a File from the FTP Server



Part 1: Configure FTP Services on Servers

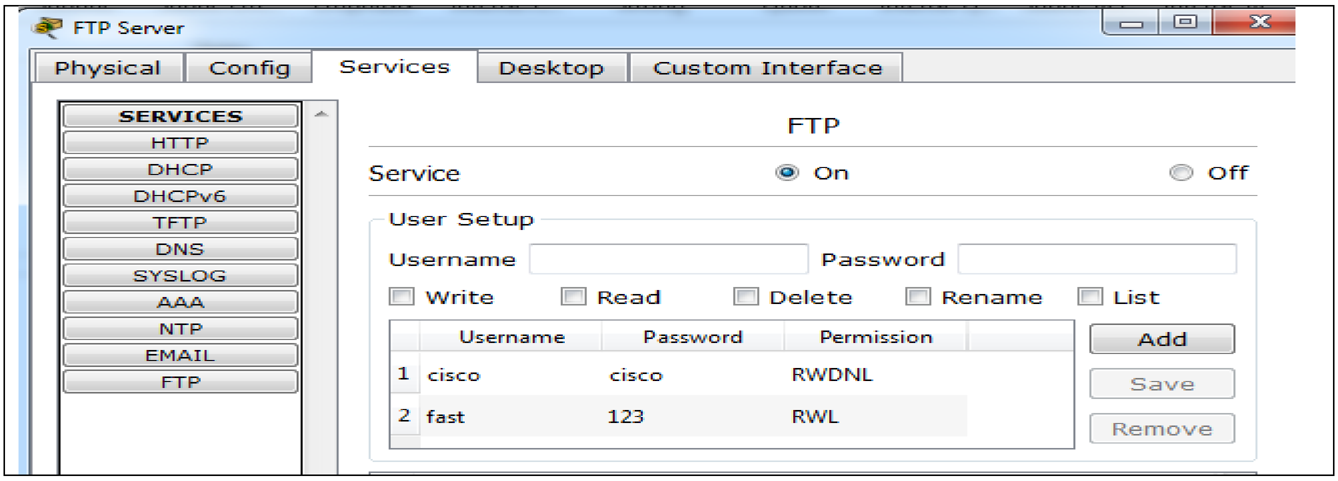
Step 1: Configure the FTP service on Server.

a. Click Server > Config tab > FTP.

b. Click On to enable FTP service.

c. In User Setup, create the following user accounts. Click the + button to add the account:

Username	Password	Permissions
fast	123	limited to Read, write and List



Service [✓ On] Off

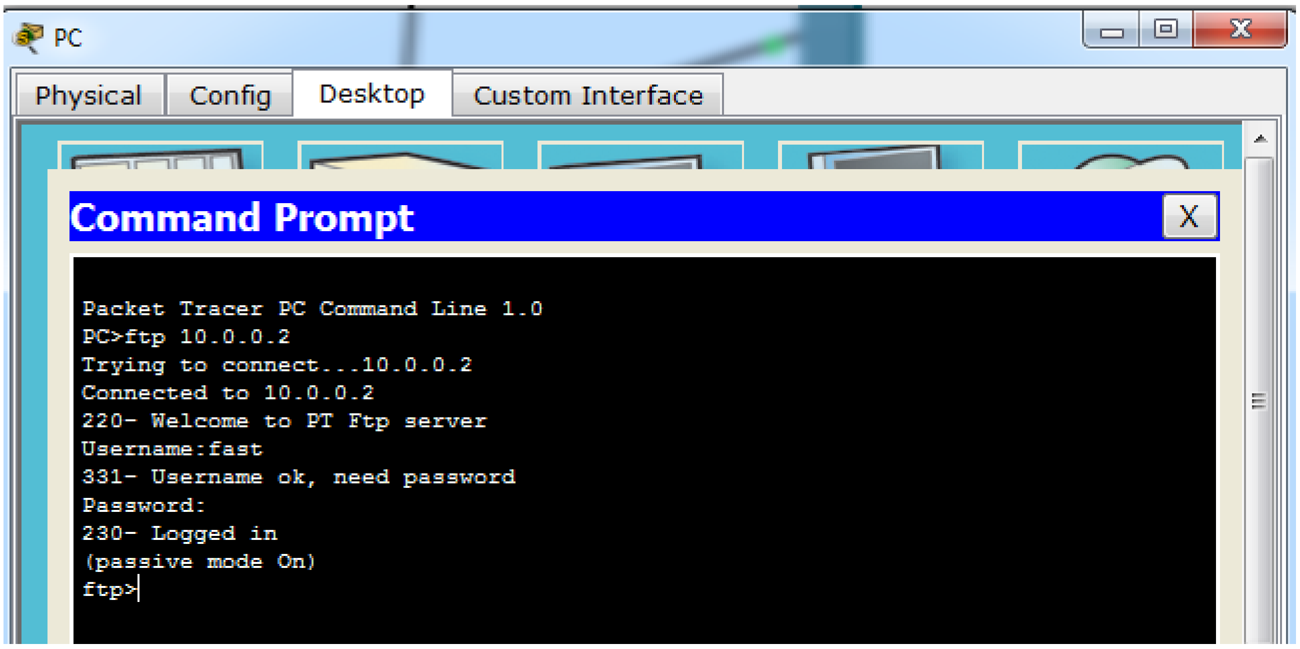
User Setup

Username Password

Write Read Delete Rename List

	Username	Password	Permission
1	cisco	cisco	RWDNL
2	fast	123	RWL

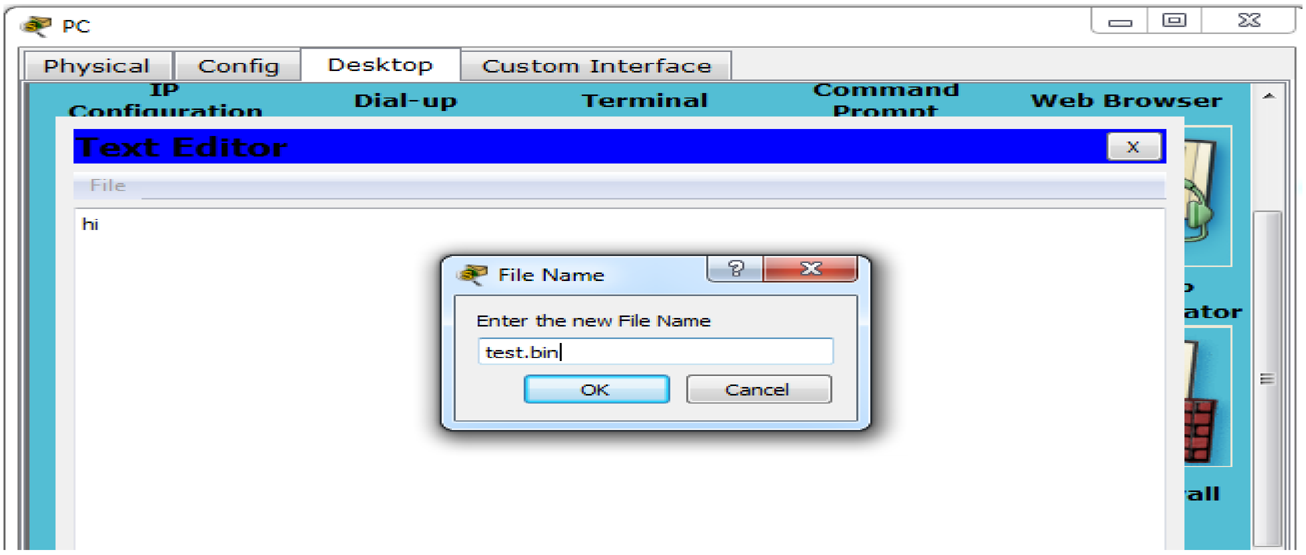
Now go to PC→Desktop →command prompt



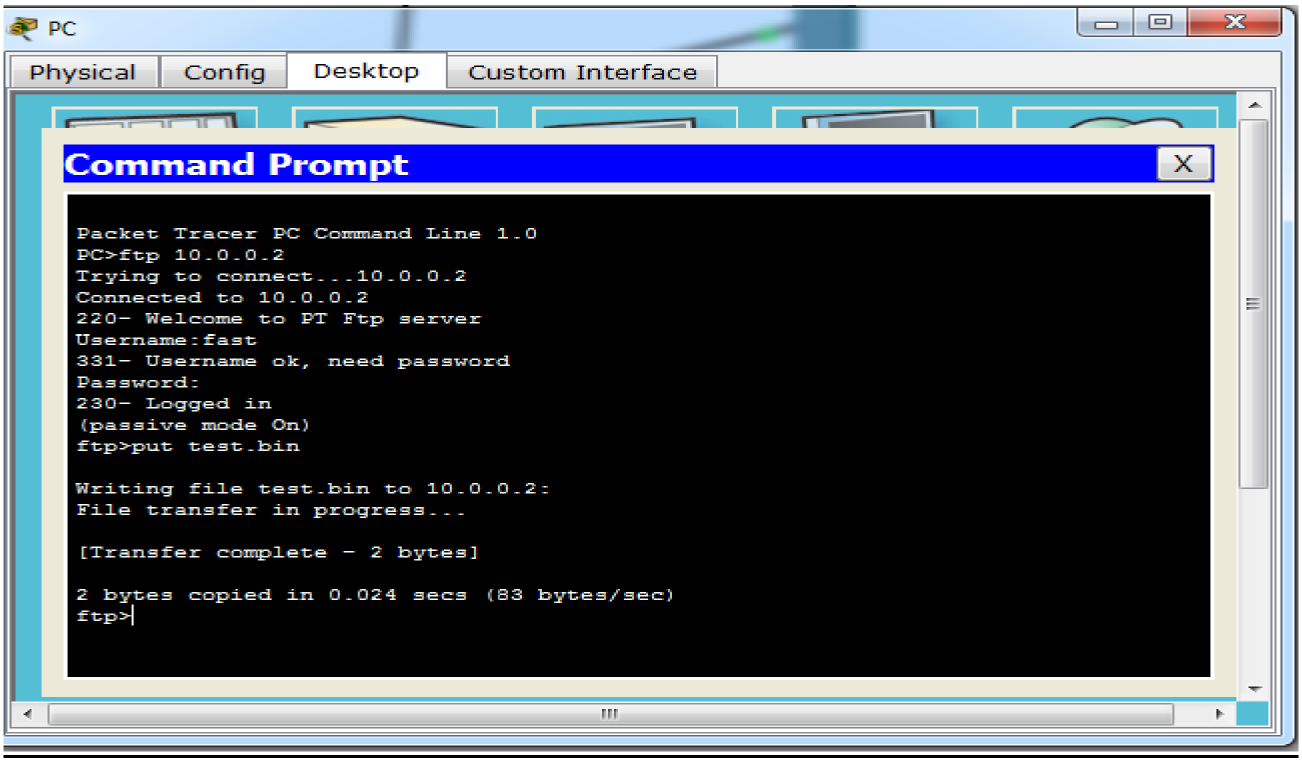
```
Packet Tracer PC Command Line 1.0
PC>ftp 10.0.0.2
Trying to connect...10.0.0.2
Connected to 10.0.0.2
220- Welcome to PT Ftp server
Username:fast
331- Username ok, need password
Password:
230- Logged in
(passive mode On)
ftp>
```

Part 2: Upload a File to the FTP Server

Now go to PC→Desktop →text editor→create file named test.bin



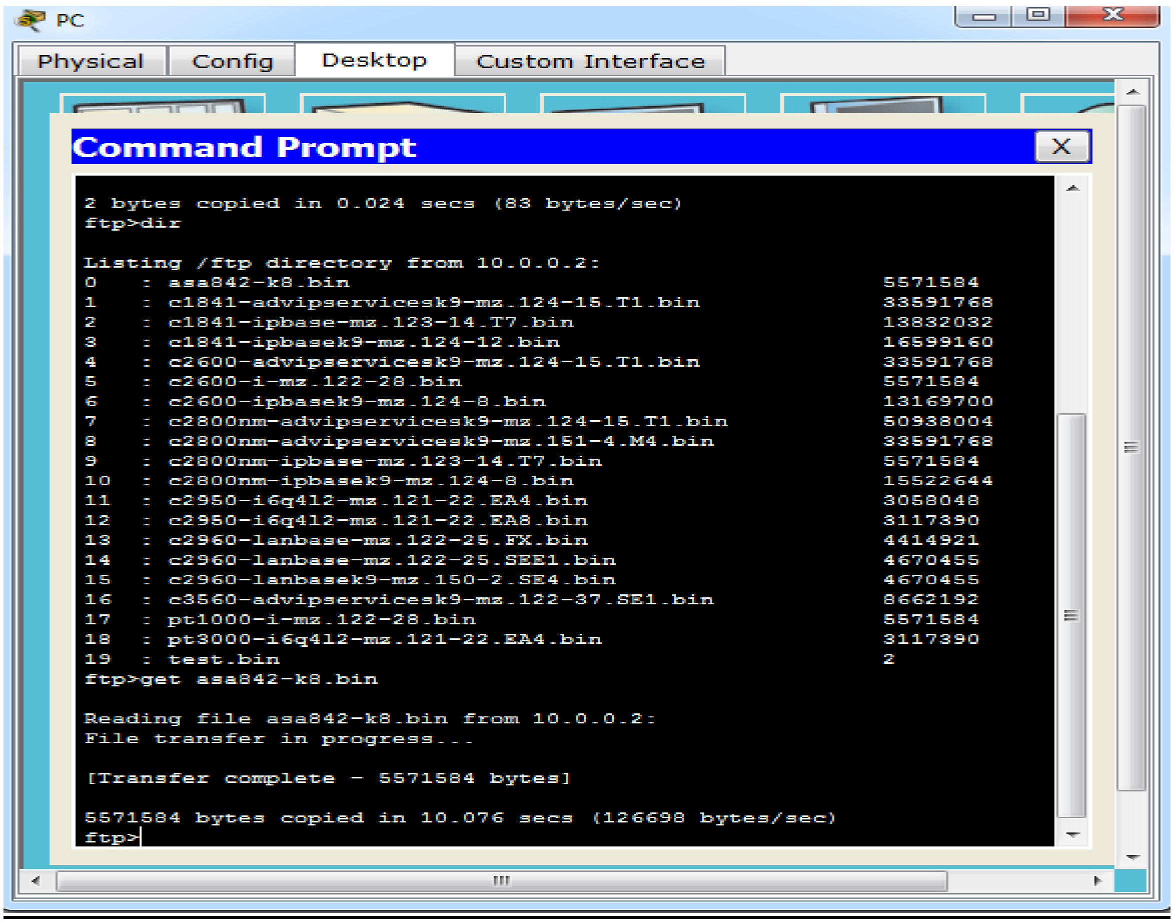
Now go to PC→Desktop →command prompt



```
Packet Tracer PC Command Line 1.0
PC>ftp 10.0.0.2
Trying to connect...10.0.0.2
Connected to 10.0.0.2
220- Welcome to PT Ftp server
Username:fast
331- Username ok, need password
Password:
230- Logged in
(passive mode On)
ftp>put test.bin

Writing file test.bin to 10.0.0.2:
File transfer in progress...

[Transfer complete - 2 bytes]

2 bytes copied in 0.024 secs (83 bytes/sec)
ftp>
```

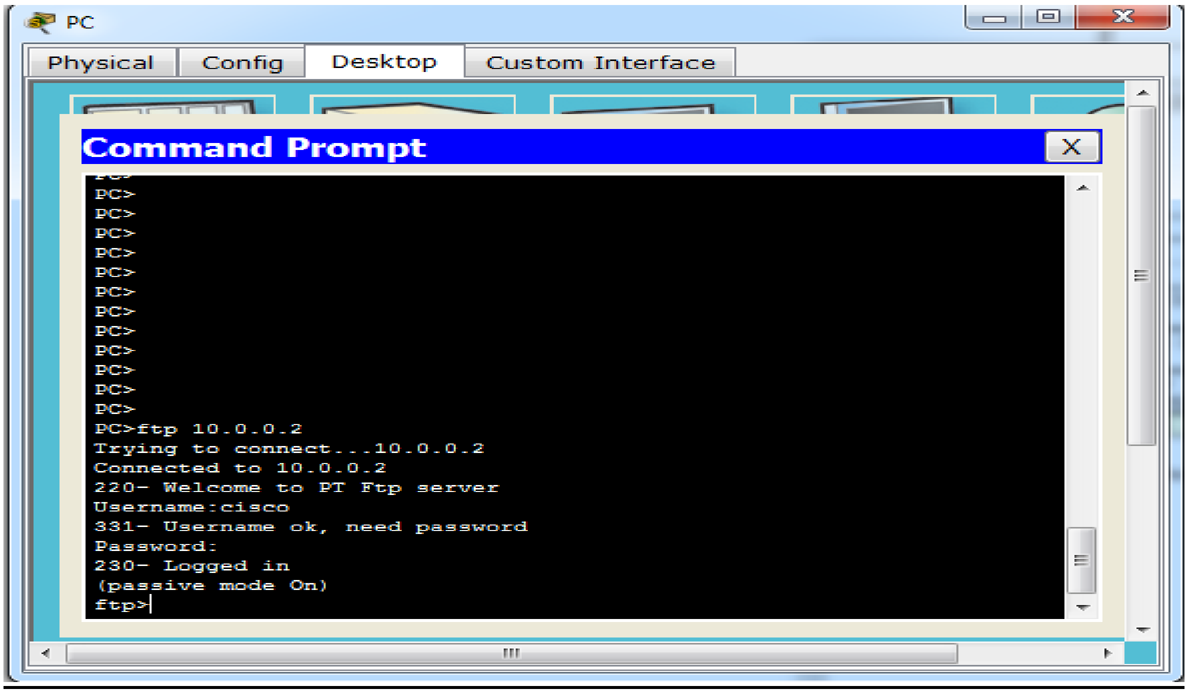
Part 3: Download a File from the FTP Server

Now go to PC→Desktop →command prompt



```
2 bytes copied in 0.024 secs (83 bytes/sec)
ftp>dir

Listing /ftp directory from 10.0.0.2:
0    : asa842-k8.bin                                         5571584
1    : c1841-advipservicesk9-mz.124-15.T1.bin                33591768
2    : c1841-ipbase-mz.123-14.T7.bin                         13832032
3    : c1841-ipbasek9-mz.124-12.bin                          16599160
4    : c2600-advipservicesk9-mz.124-15.T1.bin                33591768
5    : c2600-i-mz.122-28.bin                                 5571584
6    : c2600-ipbasek9-mz.124-8.bin                           13169700
7    : c2800nm-advipservicesk9-mz.124-15.T1.bin              50938004
8    : c2800nm-advipservicesk9-mz.151-4.M4.bin               33591768
9    : c2800nm-ipbase-mz.123-14.T7.bin                       5571584
10   : c2800nm-ipbasek9-mz.124-8.bin                         15522644
11   : c2950-i6q4l2-mz.121-22.EA4.bin                        3058048
12   : c2950-i6q4l2-mz.121-22.EA8.bin                        3117390
13   : c2960-lanbase-mz.122-25.FX.bin                        4414921
14   : c2960-lanbase-mz.122-25.SEE1.bin                      4670455
15   : c2960-lanbasek9-mz.150-2.SE4.bin                      4670455
16   : c3560-advipservicesk9-mz.122-37.SE1.bin               8662192
17   : pt1000-i-mz.122-28.bin                                5571584
18   : pt3000-i6q4l2-mz.121-22.EA4.bin                       3117390
19   : test.bin                                              2
ftp>get asa842-k8.bin

Reading file asa842-k8.bin from 10.0.0.2:
File transfer in progress...

[Transfer complete - 5571584 bytes]

5571584 bytes copied in 10.076 secs (126698 bytes/sec)
ftp>
```

<u>SIMULATION:</u> Now click on PC and go to Desktop→command prompt. Now type ftp 10.0.0.2

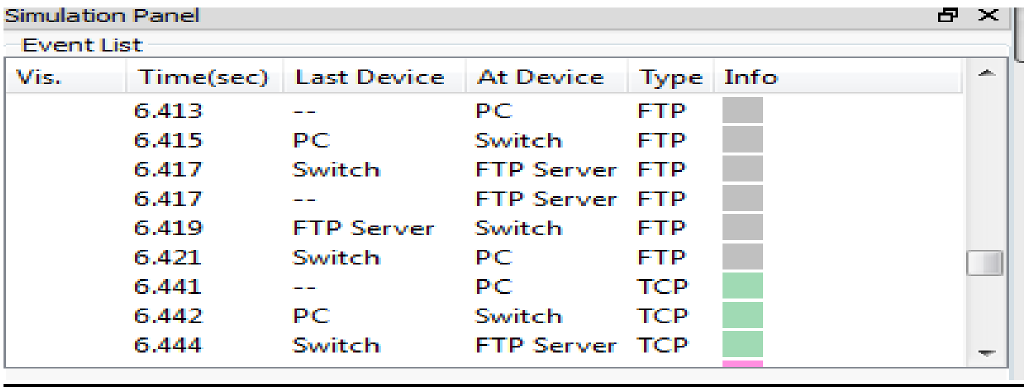


```
PC>
PC>
PC>
PC>
PC>
PC>
PC>
PC>
PC>
PC>
PC>
PC>
PC>ftp 10.0.0.2
Trying to connect...10.0.0.2
Connected to 10.0.0.2
220- Welcome to PT Ftp server
Username:cisco
331- Username ok, need password
Password:
230- Logged in
(passive mode On)
ftp>
```

SIMULATION

Now to note the FTP header format information go to simulation mode →edit filters and click on FTP check box then click on capture/forward button.

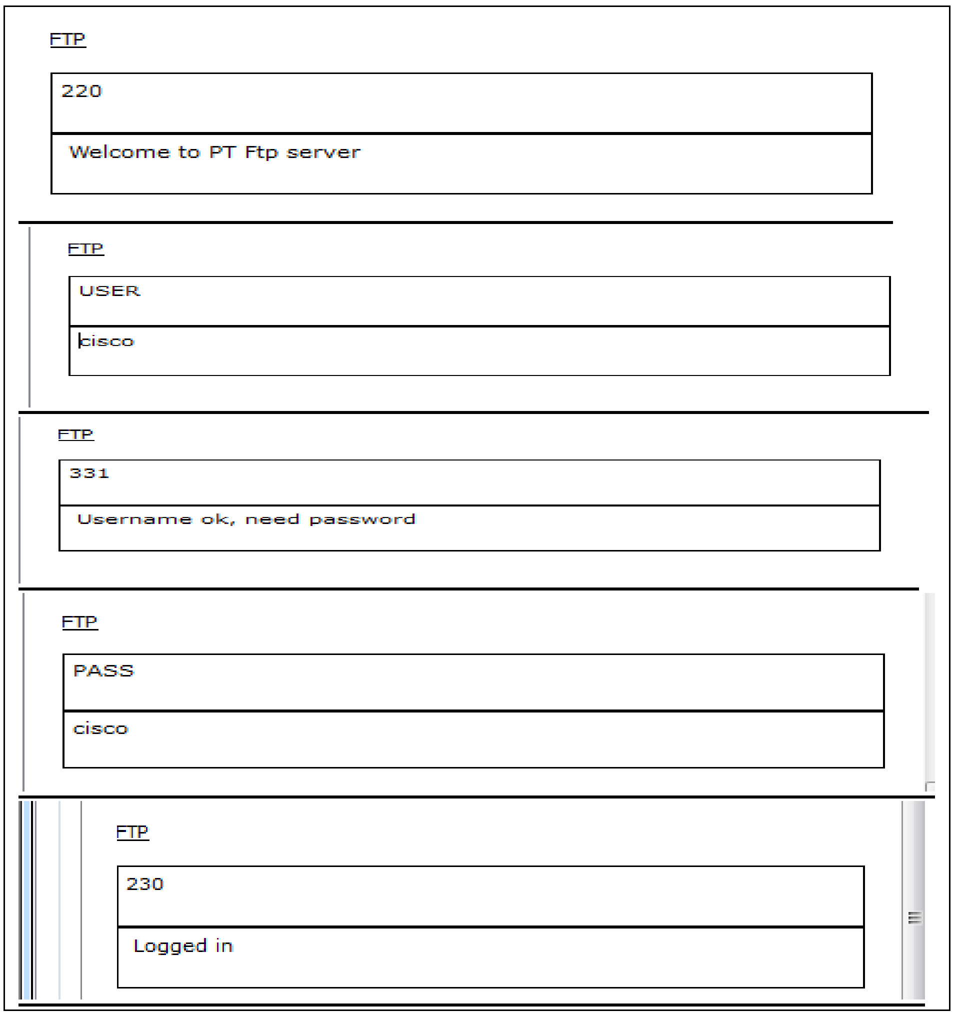
How FTP server resolve the login request.



Simulation Panel

Event List

Vis.	Time(sec)	Last Device	At Device	Type	Info
	6.413	--	PC	FTP	⬜
	6.415	PC	Switch	FTP	⬜
	6.417	Switch	FTP Server	FTP	⬜
	6.417	--	FTP Server	FTP	⬜
	6.419	FTP Server	Switch	FTP	⬜
	6.421	Switch	PC	FTP	⬜
	6.441	--	PC	TCP	🟩
	6.442	PC	Switch	TCP	🟩
	6.444	Switch	FTP Server	TCP	🟩



FTP

220
Welcome to PT Ftp server

FTP

USER
cisco

FTP

331
Username ok, need password

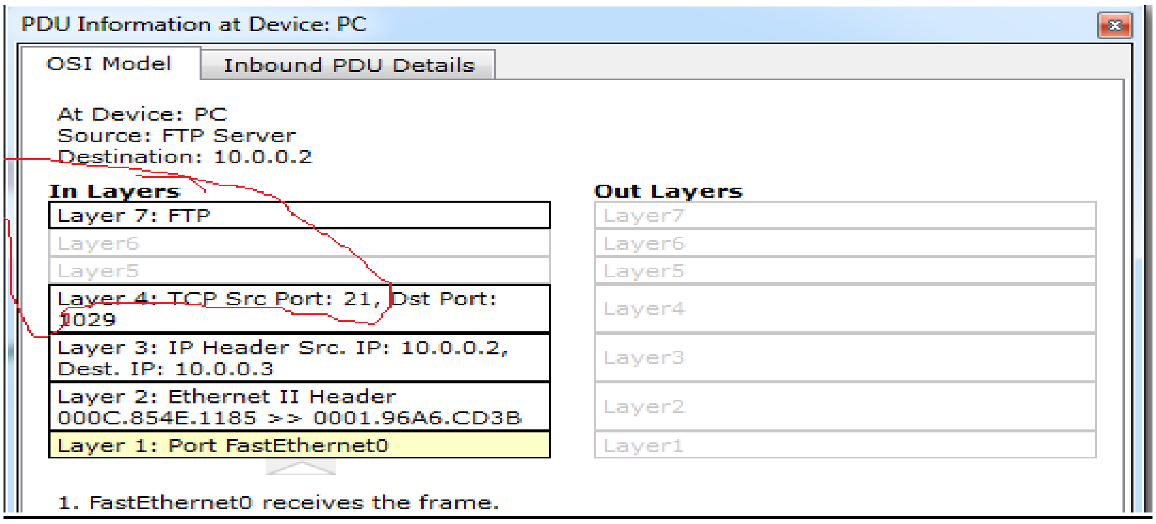
FTP

PASS
cisco

FTP

230
Logged in

Now click on the FTP packet, you can note that the destination port is 21.



PDU Information at Device: PC

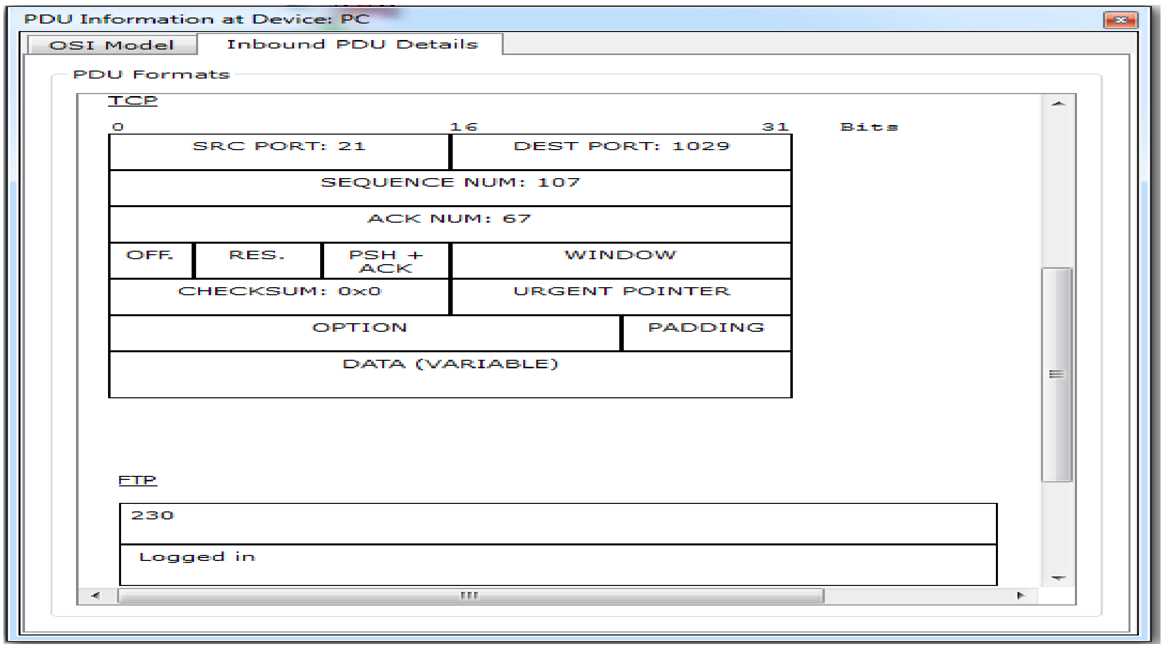
OSI Model | Inbound PDU Details

At Device: PC
Source: FTP Server
Destination: 10.0.0.2

In Layers	Out Layers
Layer 7: FTP	Layer7
Layer6	Layer6
Layer5	Layer5
Layer 4: TCP Src Port: 21, Dst Port: 1029	Layer4
Layer 3: IP Header Src. IP: 10.0.0.2, Dest. IP: 10.0.0.3	Layer3
Layer 2: Ethernet II Header 000C.854E.1185 >> 0001.96A6.CD3B	Layer2
Layer 1: Port FastEthernet0	Layer1

1. FastEthernet0 receives the frame.

Now scroll the Outbound PDU Details, you can see the FTP PDU.



PDU Information at Device: PC

OSI Model | Inbound PDU Details

PDU Formats

TCP

```
0                 16                31    Bits
| SRC PORT: 21    | DEST PORT: 1029 |
| SEQUENCE NUM: 107                 |
| ACK NUM: 67                       |
| OFF. | RES. | PSH + ACK | WINDOW  |
| CHECKSUM: 0x0   | URGENT POINTER  |
| OPTION                  | PADDING |
| DATA (VARIABLE)                   |
```

FTP

```
| 230       |
| Logged in |
```

Wireshark Lab: HTTP

we're now ready to use Wireshark to investigate protocols in operation. In this lab, we'll explore several aspects of the HTTP protocol: the basic GET/response interaction, HTTP message formats, retrieving large HTML files, retrieving HTML files with embedded objects, and HTTP authentication and security.

The Basic HTTP GET/response interaction

Let's begin our exploration of HTTP by downloading a very simple HTML file - one that is very short, and contains no embedded objects. Do the following:

1. Start up your web browser.

2. Start up the Wireshark packet sniffer, as described in the introductory lab (but don't yet begin packet capture). Enter "http" (just the letters, not the quotation marks) in the display-filter-specification window, so that only captured HTTP messages will be displayed later in the packet-listing window. (We're only interested in the HTTP protocol here, and don't want to see the clutter of all captured packets).

3. Wait a bit more than one minute (we'll see why shortly), and then begin Wireshark packet capture.

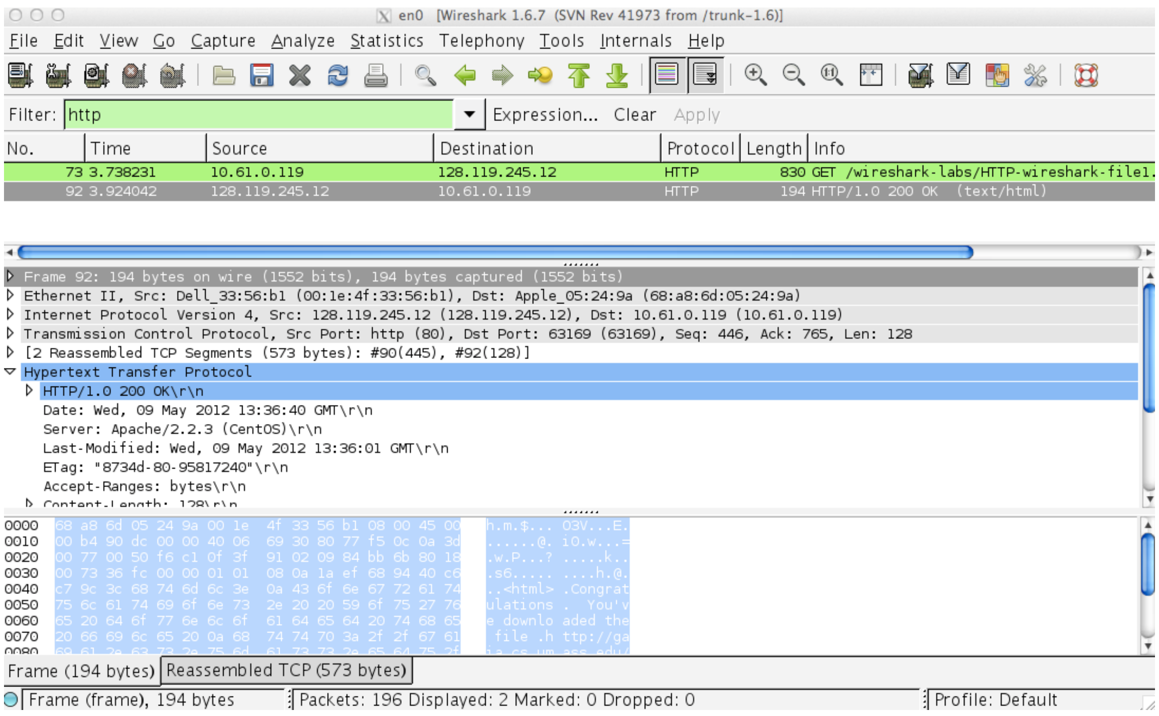
4. Enter the following to your browser

http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html

Your browser should display the very simple, one-line HTML file.

5. Stop Wireshark packet capture.

Your Wireshark window should look similar to the window shown in Figure 1. If you are unable to run Wireshark on a live network connection, you can download a packet trace that was created when the steps above were followed.


en0 [Wireshark 1.6.7 (SVN Rev 41973 from /trunk-1.6)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: http
Expression... Clear Apply

```
No.  Time       Source          Destination     Protocol Length Info
  73 3.738231   10.61.0.119     128.119.245.12  HTTP     830 GET /wireshark-labs/HTTP-wireshark-file1.
  92 3.924042   128.119.245.12  10.61.0.119     HTTP     194 HTTP/1.0 200 OK  (text/html)
```

```
▷ Frame 92: 194 bytes on wire (1552 bits), 194 bytes captured (1552 bits)
▷ Ethernet II, Src: Dell_33:56:b1 (00:1e:4f:33:56:b1), Dst: Apple_05:24:9a (68:a8:6d:05:24:9a)
▷ Internet Protocol Version 4, Src: 128.119.245.12 (128.119.245.12), Dst: 10.61.0.119 (10.61.0.119)
▷ Transmission Control Protocol, Src Port: http (80), Dst Port: 63169 (63169), Seq: 446, Ack: 765, Len: 128
▷ [2 Reassembled TCP Segments (573 bytes): #90(445), #92(128)]
▽ Hypertext Transfer Protocol
  ▷ HTTP/1.0 200 OK\r\n
    Date: Wed, 09 May 2012 13:36:40 GMT\r\n
    Server: Apache/2.2.3 (CentOS)\r\n
    Last-Modified: Wed, 09 May 2012 13:36:01 GMT\r\n
    ETag: "8734d-80-95817240"\r\n
    Accept-Ranges: bytes\r\n
  ▷ Content-Length: 128\r\n
```

```
0000  68 a8 6d 05 24 9a 00 1e  4f 33 56 b1 08 00 45 00   h.m.$... O3V...E.
0010  00 b4 90 dc 00 00 40 06  69 30 80 77 f5 0c 0a 3d   ......@. i0.w...=
0020  00 77 00 50 f6 c1 0f 3f  91 02 09 84 bb 6b 80 18   .w.P...? .....k..
0030  00 73 36 fc 00 00 01 01  08 0a 1a ef 68 94 40 c6   .s6..... ....h.@.
0040  c7 9c 3c 68 74 6d 6c 3e  0a 43 6f 6e 67 72 61 74   ..<html> .Congrat
0050  75 6c 61 74 69 6f 6e 73  2e 20 20 59 6f 75 27 76   ulations .  You'v
0060  65 20 64 6f 77 6e 6c 6f  61 64 65 64 20 74 68 65   e downlo aded the
0070  20 66 69 6c 65 20 0a 68  74 74 70 3a 2f 2f 67 61    file .h ttp://ga
0080  69 61 2e 63 73 2e 75 6d  61 73 73 2e 65 64 75 2f   ia.cs.um ass.edu/
```

Frame (194 bytes)
Reassembled TCP (573 bytes)
Frame (frame), 194 bytes
Packets: 196 Displayed: 2 Marked: 0 Dropped: 0
Profile: Default

Figure 1: Wireshark Display after http://gaia.cs.umass.edu/wireshark-labs/ HTTP-wireshark-file1.html has been retrieved by your browser.

The example in Figure 1 shows in the packet-listing window that two HTTP messages were captured: the GET message (from your browser to the gaia.cs.umass.edu web server) and the response message from the server to your browser. The packet-contents window shows details of the selected message (in this case the HTTP OK message, which is highlighted in the packet-listing window). Recall that since the HTTP message was carried inside a TCP segment, which was carried inside an IP datagram, which was carried within an Ethernet frame, Wireshark displays the Frame, Ethernet, IP, and TCP packet information as well. We want to minimize the amount of non-HTTP data displayed (we're interested in HTTP here, and will be investigating these other protocols is later labs), so make sure the boxes at the far left of the Frame, Ethernet, IP and TCP information have a plus sign or a right-pointing triangle (which means there is hidden, undisplayed information), and the HTTP line has a minus sign or a down-pointing triangle (which means that all information about the HTTP message is displayed).

By looking at the information in the HTTP GET and response messages, answer the following questions. When answering the following questions, you should print out the GET and response messages (see the introductory Wireshark lab for an explanation of how to do this) and indicate where in the message you've found the information that answers the following questions. When you hand in your assignment, annotate the output so that it's clear where in the output you're getting the information for your answer (e.g., for our classes, we ask that students markup paper copies with a pen, or annotate electronic copies with text in a colored font).

1. Is your browser running HTTP version 1.0 or 1.1? What version of HTTP is the server running?

2. What languages (if any) does your browser indicate that it can accept to the server?

3. What is the IP address of your computer? Of the gaia.cs.umass.edu server?

4. What is the status code returned from the server to your browser?

5. When was the HTML file that you are retrieving last modified at the server?

6. How many bytes of content are being returned to your browser?

7. By inspecting the raw data in the packet content window, do you see any headers within the data that are not displayed in the packet-listing window? If so, name one.

In your answer to question 5 above, you might have been surprised to find that the document you just retrieved was last modified within a minute before you downloaded the document. That's because (for this particular file), the gaia.cs.umass.edu server is setting the file's last-modified time to be the current time, and is doing so once per minute. Thus, if you wait a minute between accesses, the file will appear to have been recently modified, and hence your browser will download a "new" copy of the document.

HTTP Authentication

Finally, let's try visiting a web site that is password-protected and examine the sequence of HTTP message exchanged for such a site. The URL

http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wireshark-file5.html is password protected. The username is "wireshark-students" (without the quotes), and the password is "network" (again, without the quotes). So let's access this "secure" password-protected site. Do the following:

• Make sure your browser's cache is cleared, as discussed above, and close down your browser. Then, start up your browser

• Start up the Wireshark packet sniffer

• Enter the following URL into your browser

http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wireshark-file5.html

Type the requested user name and password into the pop up box.

• Stop Wireshark packet capture, and enter "http" in the display-filter-specification window, so that only captured HTTP messages will be displayed later in the packet-listing window.

• Note: If you are unable to run Wireshark on a live network connection, you can use the http-ethereal-trace-5 packet trace to answer the questions below; see footnote 2. This trace file was gathered while performing the steps above on one of the author's computers.)

Now let's examine the Wireshark output. You might want to first read up on HTTP authentication by reviewing the easy-to-read material on "HTTP Access Authentication Framework" at http://frontier.userland.com/stories/storyReader$2159

Answer the following questions:

1. What is the server's response (status code and phrase) in response to the initial HTTP GET message from your browser?
2. When your browser's sends the HTTP GET message for the second time, what new field is included in the HTTP GET message?

The username (wireshark-students) and password (network) that you entered are encoded in the string of characters (d2lyZXNoYXJrLXN0dWRlbnRzOm5ldHdvcms=) following the "Authorization: Basic" header in the client's HTTP GET message. While it may appear that your username and password are encrypted, they are simply encoded in a format known as Base64 format. The username and password are not encrypted! To see this, go to http://www.motobit.com/util/base64-decoder-encoder.asp and enter the base64-encoded string d2lyZXNoYXJrLXN0dWRlbnRz and decode. Voila! You have translated from Base64 encoding to ASCII encoding, and thus should see your username! To view the password, enter the remainder of the string Om5ldHdvcms= and press decode. Since anyone can download a tool like Wireshark and sniff packets (not just their own) passing by their network adaptor, and anyone can translate from Base64 to ASCII (you just did it!), it should be clear to you that simple passwords on WWW sites are not secure unless additional measures are taken.