15-441 Computer Networks
Homework 3

Due: March 29, 2005, 1:30 PM

Lead TA: Maksim Tsvetovat (maksim2042@gmail.com)

Name:
Andrew: ID

March 16, 2005

A Purpose:

This assignment gives you experience with what happens on actual network wires. To achieve this we will use a packet sniffing and network analysis tool called `ethereal` and its terminal mode counterpart `tethereal`. `Ethereal` will allow you to monitor and analyze network packets on the LAN in the CS 441 lab. By capturing data and examining packets you will become more familiar with TCP/IP, and how layered protocols are represented within packets.

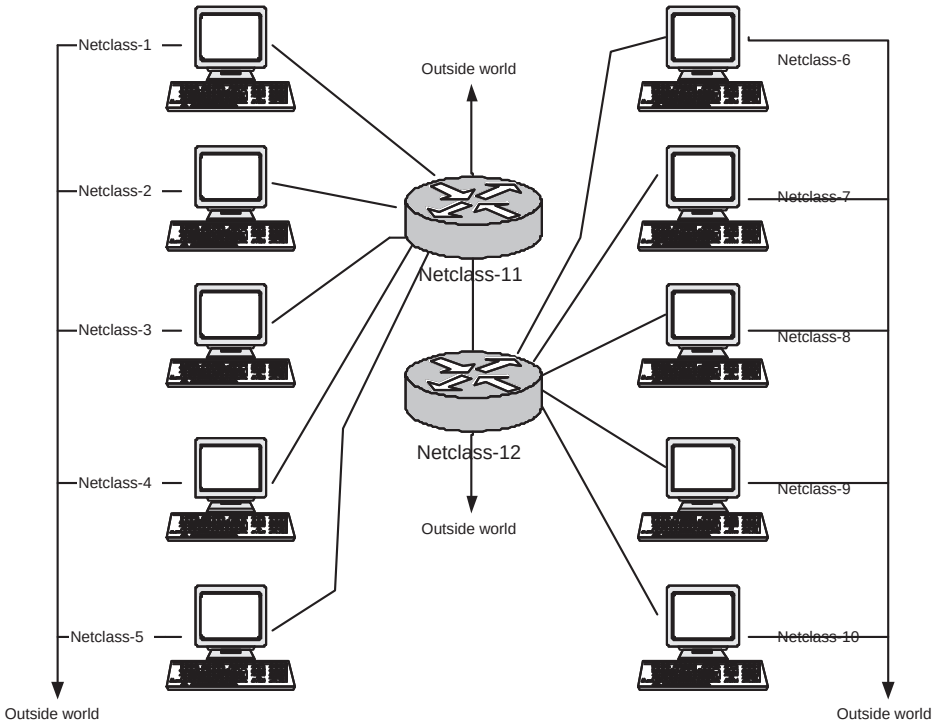
Please do this lab on your own; do not work in groups. Ask the TAs or other students for help, but please do the work yourself.

Preparing for the lab: Follow the instructions below to set-up your account on the netclass machines (machines in the CS441 lab). You won't be able to log in without performing each step.

1. Create a directory in your Andrew home directory called 15-441.
2. Now you need to give access to campusnet, so type the following:
   (a) `>fs sa /afs/andrew.cmu.edu/usr/<your_username> system:campusnet l`
   (b) `>fs sa /afs/andrew.cmu.edu/usr/\<your\_username\>/15-441 system:campusnet rl`
3. Create a file called .klogin inside the 15-441 folder which contains:
   your_username@ANDREW.CMU.EDU
4. You can not use insecure telnet to connect to netclass machines. Therefore, either use Nifty Telnet or use ssh. The username you will use will be your_username@andrew.cmu.edu not just your Andrew-id. Therefore, if you use ssh don't forget to use the "-l" flag with the correct username as otherwise, ssh will try to use just your Andrew-id and therefore you won't be able to login. Also use the "-1" (dash-one) flag to specify SSH version 1.
5. If you have any problems with logging in, send email to Maksim Tsvetovat (`maksim@cs.cmu.edu`)

B Setup for this Lab

The topology of the CS441 lab is shown below: (all end with .intro.cs.cmu.edu)



The lab consists of 12 PCs as shown with two (11 and 12) configured as routers and the rest configured as endpoints. The PCs are connected to form a LAN as well as being connected to the outside world (that's how you can telnet in). This is achieved by the use of 2 interfaces on each PC—one connected to local LAN and one to the outside world. The traffic going on in the LAN is considered private traffic—it is isolated from the outside world. In order to login to these machines use your andrew-id and password and don't log on to 11 and 12.

C Capturing and Viewing packets

Open up a terminal window.Since you don't have write access to the directories other than the local temporary directory, cd to "/tmp". Create a subdirectory with your username You will create your dumpfile (the file that will contain the captured packets) in that subdirectory. You will capture the packets using ``tethereal'' which is in ``/usr/bin''. First run `tethereal -help` to see the command line options. We want to capture 500 packets from the external traffic (hint: there are 2 Ethernet cards on each PC (eth0 and eth1). You need to determine which one is dealing with the LAN traffic and which one is connected outside (which would mean you will see telnet packets when you look at the dump) Save the packets you sniff in a file (so that we can look at them with a graphical interface). Now start up `/usr/bin/ethereal` (make sure your X-Server is running). `Ethereal` will ask for the root password—just choose "Run unprivileged". When `ethereal` is started up it will have three empty panes. Go to File → Open and open up the capture file.

1. Select the "+" sign in front of each of the protocol layers to get more detail about each protocol header.
   (a) View the captured broadcast packet data. Select the "Source" column to sort the packets by the source address. What is the most common source address?

   (b) What protocols (ncp, sap, tcp, icmp, arp, udp, etc...) do you see? (HINT: select the "Protocol" column to sort by protocol)

   (c) Which protocol is most common in your captured packets?

   (d) Select a packet in the top frame that is labeled as a TCPpacket. Determine the following values for this packet :

   ```
   Ethernet destination address: ___:___:___:___:___:___

   IP source address: ___:___:___:___

   IPTTL ________

   TCPsource port ___________
   ```

   (e) Select the "header length" field of the IPheader. This should cause a byte in the raw data pane to be highlighted. What value does this byte have and what does it mean?

D TCP Forensics

This question does not require you to use the 441 lab.

You are the TCP specialist at the FBI. One day an FBI agent gives you a packet trace of a TCP connection between two machines on the Internet. The trace is believed to contain important information pertaining to national security.

This packet trace contains 113 packets, most of them IP packets. Each line in the trace is one packet, identified by its packet number (from 0 to 112). The rest of the line in a sequence of bytes, represented as hex numbers. For example, consider the first line of the trace:

```
0       45 00 00 2c f2 7b 00 00 40 06 da 71 80 02 dc 8a 80
        02 d1 4f 6a b0 00 17 43 97 0e d6 00 00 00 00 60 02
        02 00 2b 13 00 00 02 04 05 b4 00 00
```

The first number "0" (packet number 0) implies that this is the first packet received at the trace point.

The first byte of the packet is "45" in hex, which is "69" in decimal. As a hint, the first byte seems to indicate that this is an IPv4 packet, where the IP header contains 20 bytes. The fourth byte of the packet is "2c".

This trace, hw3.trace.txt, is available on the course web page under the Assignments link. To reduce the amount of work you have, we provide a template code tcptrace.c (also available on the course web page under the Assignments link) which parses the trace file into an array of bytes. You are free to use perl or any other tools (and even by hand if you wish) to analyze this trace file.

2. Please answer the following questions (and determine whether this is a threat to national security):

   (a) What is the client's IP address, the client's port number, the server's IP address, and the server's port number of this TCP connection? Based on the IP addresses of the client and server, what are their respective DNS names? You may assume that the computer which initiates the connection is the client, and the other computer is the server.

   ```
   Client IP: __________

   Client Port: ________

   Client DNS Name: ____________________________

   Server IP: __________

   Server Port: ________

   Server DNS Name: ____________________________
   ```

   (b) Which Internet application (i.e. web, FTP, gopher,) is running on top of this TCP connection?

   How do you arrive at this conclusion? (hint: you should not have to guess based on the TCP payload)

(c) Using the TCP connection state diagram in Figure 6 of RFC 793, identify which packets (by their packet number) cause or result from the TCP state transitions of the TCP client? Your answer should be in the form: when a client receives packet X or user request Y, its TCP state is changed from A to B, and packet Z is sent (or some other action takes place). For example, when a client receives user request OPEN, the TCP state is changed from CLOSED to SYN-SENT, and a packet Z is sent.

(d) Using the same format as used in (c), identify the state transitions with respect to the server.

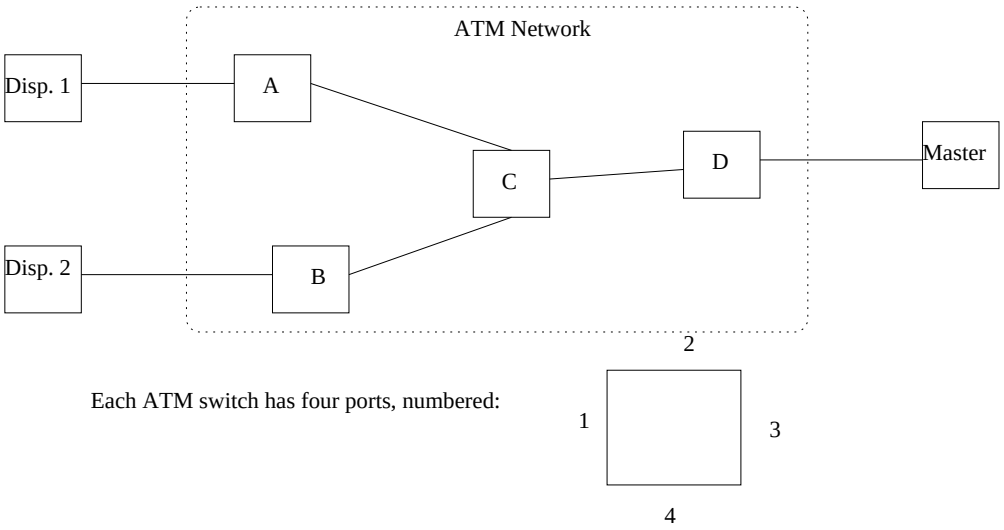
(e) Can you recreate the "crime scene" at the (client's) end user terminal? What did the end user see on the screen? What did the end user type on the keyboard? What is he/she trying to do? We are not looking for a 100% accurate bit-by-bit transcription of the TCP payload, but you should describe in enough detail what the end user has seen on the screen and typed on the keyboard.

(f) Staple a printout of the source code, script, etc. that you used to answer this problem. This is evidence that you are indeed a qualified FBI TCP expert and did not "hire" someone else to perform this trace analysis.

E ATM

3. The CMU Credit Union is thinking about creating a world-wide network of Cash and Homework Dispensers for CMU students. They've decided to use a network of ATM PVCs (permanent virtual circuits) to connect the dispensers back to their master bank computers, and want you to review their network configuration to make sure that it's set up properly.[1]

   The ATM network is shown in the box in the figure below.



Each ATM switch has four ports, numbered:

   (a) Fill in the missing parts of the configuration table. To save some writing, we'll only worry about traffic flowing TO the master - we'll assume that we configure the outbound links using the same VCIs and swap the input/output parts.

Switch	Input Port	Input VCI	Output Port	Output VCI
A	1	7		
B	1	7		
C	2			
C	4			
D	1			15
D	1			16

[1] Yes. We're using ATM to connect ATMs. Sorry.

(b) The bank comes back to you later and tells you that they're encountering problems when uploading new software images to their dispenser machines. Things are great for handling small customer transactions ("withdraw $50 from account 555"), but these large file transfers fail miserably. You realize that the ATM network provider has never needed to support data before, since their primary business has been voice up until now. Can you explain the problem and how to fix it?

F DNS

The Andrew Linux machines provide the `dig` program that allows you to query Domain Name Service (DNS) servers around the Internet. For more information on how to use dig, consult the man page. When running dig for the purpose of this question, you should use the following format:

```
dig +norecurse @name.of.dns.server record-type domain-name
```

- `name.of.dns.server` is the hostname of the DNS server you wish to query.
- `record-type` is the type of DNS record you wish to retrieve, such as ANY, MX, HINFO, A, or SOA.
- `domain-name` is the name of the host or domain you seek information on.

The DNS is a distributed architecture that uses hierarchical delegation. At the top of the system are the root name servers, who know which DNS server is responsible for each of the top-level domains (such as .com, .net, and .edu). If you send a root server a query for a particular machine, you will receive a reply listing the servers that have been delegated authority for those top-level domains, and you can recursively ask those servers to resolve the name.

1. To discover an actual chain of delegation, run a series of NS queries for `WWW.SCS.CS.NYU.EDU`. You may start with any of the root servers, and you should continue your sequence of queries until you stop getting new delegations (in some domains, this is indicated by a DNS server returning you a delegation pointing to itself, and in other domains this is indicated by a DNS server returning you a SOA record instead).

   Delegation chain for: AOL.COM:

Server queried	NS delegations to
`A.ROOT-SERVERS.NET`	`A.GTLD-SERVERS.NET`, `G.GTLD-SERVERS.NET`
`G.GTLD-SERVERS.NET`	`dns-01.ns.aol.com`, `dns-02.ns.aol.com`, `dns-06.ns.aol.com`, `dns-07.ns.aol.com`
`dns-02.ns.aol.com`	`dns-01.ns.aol.com`, `dns-02.ns.aol.com`, `dns-06.ns.aol.com`, `dns-07.ns.aol.com`

   This was produced by running the following commands:

   ```
   dig +norecurse @a.root-servers.net NS aol.com
   dig +norecurse @G.GTLD-SERVERS.NET NS aol.com
   dig +norecurse @dns-02.ns.aol.com NS aol.com
   ```

   Generate the delegation chain for WWW.SCS.CS.NYU.EDU. Present your results in a table like the one above. Each NS query will typically return two or more answers; choose among them at random. If you query a server and get a timeout, choose an alternate server.

2. The DNS is also used to translate IP addresses into hostnames. Again, the database is distributed in a hierarchical fashion, with a wrinkle. The most specific part of a domain name is on the left (i.e. WWW in WWW.SCS.CS.NYU.EDU), but the reverse is true of IP addresses (i.e., in 128.2.198.101, 128 is toplevel , 128.2 belongs to CMU.EDU, and 128.2.198 belongs to CS.CMU.EDU).

   Thus, address-to-name mapping is handled by reversing the bytes of the IP address and making queries in a special domain. To turn 128.2.198.101 into a hostname, various servers are sent queries seeking PTR records for 101.198.2.128.in-addr.arpa. The first query would be:

   ```
   dig @a.root-servers.net PTR 101.192.3.128.in-addr.arpa.
   ```

   You will know you're done when your query gives you back a PTR record in addition to (or instead of) NS records. Fill in a table like the one above showing a query chain for the IP address 128.2.203.179.