Course Individual Project

Information:

Note:

All course related project "PACKET/Wireshark" work is

to be done individually and individual reports (and logs

if needed) are to be uploaded into the student portal for

grading.

First: Create/build a network or get access to a

network that has at least two local (host) machines (any

OS, machines may be real or virtual) and one

switch/router on it. The switch/router maybe real or

virtual (for example a VirtualBox networking

connection). You may use any combination of real/virtual

networks and real/virtual equipment that you have access to the packets on. You may use any Operating Systems

you desire. Your locally created network may be

connected to the Internet, but a maximum of only ONE

Internet uplink

connection is allowed. In addition

to sniffing packets, another learning goal of this task

assignment is for students to gain handson

exposure to

creating their own network rather than just attaching to

the Internet.

10 Point VM BONUS (Can score 100): For having a

Local virtual machine as one or more of the hosts (with

data sender, data receiver, or WireShark analyzer

Installed on it) in your project network.

A learning goal of the bonus is to encourage students

to apply the course concepts in the virtual and cloud

computing worlds (including virtual ports, connections,

virtual cabling, switches, etc.) in addition to applying

course concepts to real world equipment.

Second: Download and install the latest / most current

Wireshark version (wireshark.org).

Third: Using WireShark (or equivalent), perform 3 tasks

chosen from items 0 through 10 from the task list (task

list located after 4th step).

The learning goal of these assignment tasks is to help

students move up from Operating System provided

terminal tools (such as ping, traceroute and tracecert

that confirm basic network connectivity) to more

advanced, profession level

network analysis tools.

Fourth: Write an original 1,000+ word proper

assignment results report. Projects may be rejected

(disqualified) for not meeting this assignment

requirement). Your report must include:

(a) A proper introduction, including:

A logical description and diagram of the

network configuration; a photograph of the

locally created network hardware equipment

arrangement is desirable,

Description of equipment, hosts, cabling and

ports used,

Description of where WireShark was located,

Description of various IP addresses and their

functions/applications.

(b) Individual sections for each task including:

Task description,

Written step by step

Text instructions for the steps taken by the student

High quality plagiarism resistant screen

Captures showing how you performed each

of the tasks you selected.

(c) Screen capture(s) for each step (screen

captures are to include the desktop date and time

clock), and Student's NPU portal. Every closeup

screen capture is to also have a zoom out screen

capture that includes the background of the

desktop (with several desktop icons showing)

and clock showing. Screen captures must be

readable. Screen captures are to be embedded in

your write up report. To the maximum extent

possible students are to configure WireShark to

display date and time of the packets.

(d) A matching copy of your actual WireShark

data log file(s) embedded as native text in your

assignment report appendix.

\*Project Task List (mentioned in Third Step above):

Perform 3 of the following:

0) Note: This task assignment is different from Week

4's, in that you are additionally to use WireShark to

determine the sender's IP address, and possibly

some IPs in the path between the sender and the

receiver. Using the computer language of your

choice, create an email application that sends a

simple text email every few seconds. You can think

of your email application as a malicious "spammer."

Then using WireShark, find and show that the

email(s) was sent and discover the IP address of the

spammer. Hint: Consider using WireShark to look for

Simple Mail Transfer Protocol (SMTP) packets or port

25 or even searching for known text (your NPU

Student ID) in the email. Note: Some third party

Email systems may view your automated email as

Spam and filter it out. Hence, you might not be able

to see both the transmit and the receive sides. Note:

Some students have reported that if they have

strong Google security measures enabled on their

account that they run into difficulties email

themselves in an automated way. The reported

solution is to reduce their account's security settings

temporarily.

1) Discover & sniff the contents of a standard

website textbox in any no encrypted

web site.

Note: If you have a private web server (perhaps by

installing a Linux LAMP or WAMP server system), you

may use it instead of a public web site.

2) Discover/sniff/hack cookies.

Example: Show that a cookie was made.

Note: You can manually delete it before going to

the web page. If you wish and if you have your own

web server, you may create your own cookie writer.

Please take a screen capture before, during

(Wireshark will log) and after cookie creation.

3) Create a specific site or IP address or Machine

filter within WireShark and show that it is

functioning.

4) Create a HTTP, TCP or UDP protocol or port filter

and show that it is functioning (before and after).

5) Track a DHCP handshake and show its progress.

6) Track an Encryption/SSL handshake setup.

Note:

You do not have read (break) the encrypted text,

just show that you found it and/or captured a copy

of it.

7) Use WireShark to show traffic inbetween

two

Machines. Note: If you are using virtual machines

via Virtual Box, then you probably will have to adjust

Virtual Box’s network interface settings for the

specific virtual machine's configuration.

8) Configure FTP between two machines and then

use Wire Shark to watch as a file is transferred or the

FTP session is set up.

9) Use Wires hark to troubleshoot a network.

Please document the test trouble *you* created and

show how you found it through traffic analysis.

Example 1: You have a 3 machine Y topology

network where you caused a link failure by pulling

the plug. Example 2: You changed a firewall setting

to block a specific type of traffic.

10) Custom. You may request written permission and

preapproval

from the Professor to investigate

another capability/feature of the WireShark package.

\* Foot Note: While the actual Wireshark packet work for

this course project is to be done individually; with written

Professor preapproval,

Network hardware equipment use,

Sharing, and configuration may be done collaboratively by

students, in the same classroom course section only. With

written Professor preapproval,

students may share

hardware and cabling to build their networks. Each student

must state in their individual assignment report which

hardware (real or virtual) elements they supplied, which

they shared and the other students they shared with.

Students are NOT allowed in any case to share

Wires hark assignments/work, data, data logs, or

screen captures.

.

Report

Introduction

Wirehark is network packet analyzer. The use of a network packet analyser is to capture network packets and display that packet data as detailed as possible.  It is a traffic analyzer, that helps you learn how networking works, diagnose problems and much more. A network packet analyses is a measuring device used to examine what's going on inside a network cable, just like a voltmeter is to examine what's going on inside an electric cable. In the past, packet capturing tools were e very expensive or it is proprietary with the arrival of Wireshark , all that has changed and it's is one of the best open source packet analysers available today.

Although the range of Internet applications are many and varied, the protocol suites associated with the different application/network combinations have a common structure. The different types of network operate in a variety of modes circuit-switched or packet-switched, connection-oriented or connectionless and hence each type of network has a different set of protocols for interfacing to it. Above the network-layer protocol, however, all protocol suites comprise one or more application protocols and a number of what are called application-support protocols. In order to mask the application protocols from the services provided by the different types of network protocols, all protocol suites have one or more transport protocols. These provide the application protocols with network-independent information interchange service and, in the case of the TCP/IP suite, they are the Transmission Control Protocol (TCP) and the user datagram protocol (UDP). TCP provides a connection-oriented (reliable) service and UDP a connectionless (best-effort) service. Both protocols are present in the suite and the choice of protocol used is determined by the requirements of the application. There is also a version of TCP for use with wireless networks.

**IP address**: is the IP Address of the server/appliance with the private key  
**Port:** is usually 443 for SSL/TLS  
**Protocol**: is usually HTTP  
**Key FIle**: is the location and file name of the private key. This is the key used in the certificate key pair of SSL virtual server for which you are trying to decrypt the traffic. All the SSL key and certificates are saved on Netscape appliance in config /ssl directory. To use the key to decrypt the traffic it should be saved to the local disk and this path should be specified while decrypting the traffic

HTTP

The Hyper Text Transport Protocol is a text-based request-response client-server protocol. A HTTP client performs a HTTP request to a HTTP server, which in return will issue a HTTP response. The HTTP protocol header is text-based, where headers are written in text lines.

HTTP/1.1 allows for client-server connections to be pipelined, whereby multiple requests can be sent without waiting for a response from the server. The only restriction is the server MUST return the responses in the same order as they were received. This enables greater efficiency, especially on revalidation.

An encrypted variant named HTTPS is also available. This is often used where privacy of data is necessary when using online banking. The HTTPS protocol is in fact two protocols running on top of each other. The first protocol is a security protocol like SS. The second protocol, which runs on top of this security protocol, is HTTP.

Cookies

Cookies are a way for a HTTP server to store information on the client, which will be presented back to the server in following requests. The purpose is to have a way of maintaining information between the client and the server to simulate a session http by itself has no notion of sessions; it's just a way to exchange object.

When I'm browsing a web page, there should be some cookies exchange between my machine and the remote web server. I capture the packets when browsing web pages, is it possible for me to extract cookies from the captured packet. If the website was not using https, then you are able to see the cookies. Just expand the HTTP tree in the packet details pane of Wireshark. Whether the cookie-values are encrypted depends on the implementation of the web-application. When the website did use https, you can decrypt the traffic, but only if you have possession of the private key of the server. Which you do not have when just browsing webpage unless you are the site administrator .

TCP/IP protocol suite

It will be helpful to illustrate the position of each protocol relative to the others in the TCP/IP suite. The IP protocols and network dependent protocols below them are all part of the operating system kernel with the various application protocols implemented as separate programs or processes. The two transport protocols, TCP and UDP, are then implemented to run within the operating system kernel.

Screenshot: TCP/IP protocol suite interlayer communication

Capture filters are not to be confused with display filters. The former are much more limited and are used to reduce the size of a raw packet capture. The latter are used to hide some packets from the packet list.

Capture filters are set before starting a packet capture and cannot be modified during the capture. Display filters on the other hand do not have this limitation and you can change them on the fly.

In the main window, one can find the capture filter just above the interfaces list and in the interfaces dialog

The Transmission Control Protocol (TCP)

The transmission control protocol (TCP) provides two communicating peer application protocols in a client computer and the other in a server computer - with a two-way, reliable data interchange service. This is transparent to the two communicating peer TCP protocol entities which treat all the data submitted by each local application entity as a stream of bytes. The stream of bytes flowing in each direction is then transferred from one TCP entity to the other in a reliable way; that is, to a high probability, each byte in the stream flowing in.

Screenshot: TCP packet format

# User Datagram Protocol (UDP)

The UDP layer provides datagram based connectionless transport layer (layer 4) functionality in the Internet protocol family. UDP is only a thin layer, and provides not much more than the described UDP port multiplexing.  Just like IP, UDP doesn't provide any mechanism to detect, packet loss, duplicate Packets, and the like.

## Protecting the system:

Network administrators use Wireshark for troubleshooting the network problems. Protocol examination is a procedure used to notice in a real time. The raw data sent across the network interface is helpful for network arrangement and troubleshooting. Wireshark is used to monitoring distributed application and that monitored data can be used for detecting errors so performance will be improved. It is mainly used for examining the security problems and debugging protocol implementations. Easy to access and learn TCP\IP protocols, MAC frame, IP datagram.

Conclusion:

In this project, I will examine common UDP and TCP traffic with wireshark and This report explains the operation of Wireshark - Network Analyser with clear demonstration. Initially report describes the overview and outstanding features of Wireshark like TCP Steam, Promiscuous Mode, TethereaI, Plugins platform working etc. In mechanism illustrated the internal function blocks, Interfaces and Packages of Wireshark. Next in demonstration part capturing procedure steps, configuration options and filters are described with graphical representation. This report mainly focuses on how Wireshark grabs data packets from the network and why it is the best among all the sniffers. Lastly some of the limitations/weaknesses those are present in Wireshark. In particular, I would like to state that the assignment helped a lot to learn about all the options in Wireshark. Finally I thank Sir Jim corner for giving me this chance to explore my knowledge.