

MODERN EDUCATION SOCIETY'S WADIA COLLEGE OF ENGINEERING, PUNE

Fourth Year of Computer Engineering (2019 Course)

**410247: Laboratory Practice IV
410244(C): Cyber Security and Digital Forensics**

NAME OF STUDENT:	CLASS: BE
SEMESTER/YEAR: VII	ROLL NO:
DATE OF PERFORMANCE:	DATE OF SUBMISSION:
EXAMINED BY:	EXPERIMENT NO:

TITLE: WIRESHARK

AIM/PROBLEM STATEMENT: Configure and demonstrate use of vulnerability assessment tool like Wireshark or SNORT

OBJECTIVES:

- To understand various vulnerabilities and use of various tools for assessment of vulnerabilities

OUTCOMES:

- Identify various vulnerabilities and demonstrate using various tools.
- To apply the scientific method for security assessment

PRE-REQUISITES:

1. Knowledge of C, C++, python programming
2. Basic knowledge of authentication, access control, intrusion detection and prevention.

THEORY:

Wireshark is an open-source network protocol analysis software program started by Gerald Combs in 1998. A global organization of network specialists and software developers support Wireshark and continue to make updates for new network technologies and encryption methods. Wireshark is absolutely safe to use. Government agencies, corporations, non-profits, and educational institutions use Wireshark for troubleshooting and teaching purposes. There isn't a better way to learn networking than to look at the traffic under the Wireshark microscope.

There are questions about the legality of Wireshark since it is a powerful packet sniffer. The Light side of the Force says that you should only use Wireshark on networks where you have permission to inspect network packets. Using Wireshark to look at packets without permission is a path to the Dark Side.

How does Wireshark work?

Wireshark is a packet sniffer and analysis tool. It captures network traffic on the local network and stores that data for offline analysis. Wireshark captures network traffic from Ethernet, Bluetooth, Wireless (IEEE.802.11), Token Ring, Frame Relay connections, and more.

Ed. Note: A “packet” is a single message from any network protocol (i.e., TCP, DNS, etc.)

Ed. Note 2: LAN traffic is in broadcast mode, meaning a single computer with Wireshark can see traffic between two other computers. If you want to see traffic to an external site, you need to capture the packets on the local computer.

Wireshark allows you to filter the log either before the capture starts or during analysis, so you can narrow down and zero into what you are looking for in the network trace. For example, you can set a filter to see TCP traffic between two IP addresses. You can set it only to show you the packets sent from one computer. The filters in Wireshark are one of the primary reasons it became the standard tool for packet analysis.



How to Download Wireshark

Downloading and installing Wireshark is easy. Step one is to check the official [Wireshark Download page](#) for the operating system you need. The basic version of Wireshark is free.

Wireshark for Windows

Wireshark comes in two flavors for Windows, 32 bit and 64 bit. Pick the correct version for your OS. The current release is 3.0.3 as of this writing. The installation is simple and shouldn't cause any issues.

Wireshark for Mac

[Wireshark is available on](#) Mac as a [Homebrew](#) install.

To install Homebrew, you need to run this command at your Terminal prompt:

```
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

Once you have the Homebrew system in place, you can access several open-source projects for your Mac. To install Wireshark run this command from the Terminal:

```
brew install Wireshark
```

Homebrew will download and install Wireshark and any dependencies so it will run correctly.

Wireshark for Linux

Installing Wireshark on Linux can be a little different depending on the Linux distribution. If you aren't running one of the following distros, please double-check the commands. Ubuntu

From a terminal prompt, run these commands:

1. `sudo apt-get install wireshark`
2. `sudo dpkg-reconfigure wireshark-common`
3. `sudo adduser $USER wireshark`

Those commands download the package, update the package, and add user privileges to run Wireshark.

Red Hat Fedora

From a terminal prompt, run these commands:

1. `sudo dnf install wireshark-qt`
2. `sudo usermod -a -G wireshark username`

The first command installs the GUI and CLI version of Wireshark, and the second adds permissions to use Wireshark.

Kali Linux

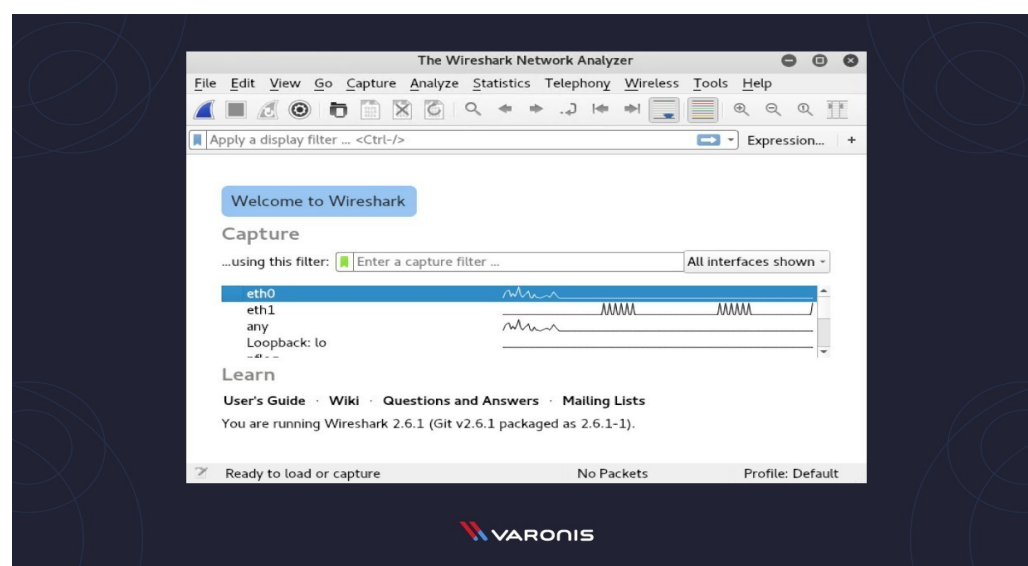
Wireshark is probably already installed! It's part of the basic package. Check your menu to verify. It's under the menu option "Sniffing & Spoofing."

Data Packets on Wireshark

Now that we have Wireshark installed let's go over how to enable the Wireshark packet sniffer and then analyze the network traffic.

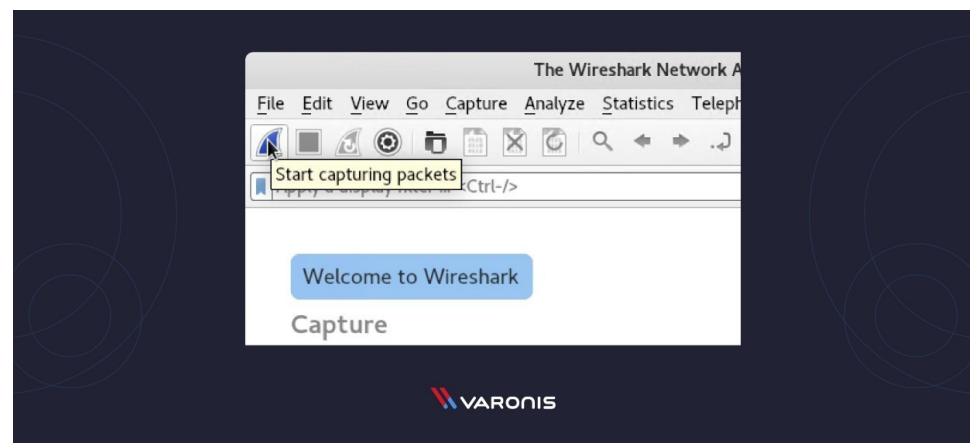
Capturing Data Packets on Wireshark

When you open Wireshark, you see a screen that shows you a list of all of the network connections you can monitor. You also have a capture filter field, so you only capture the network traffic you want to see.

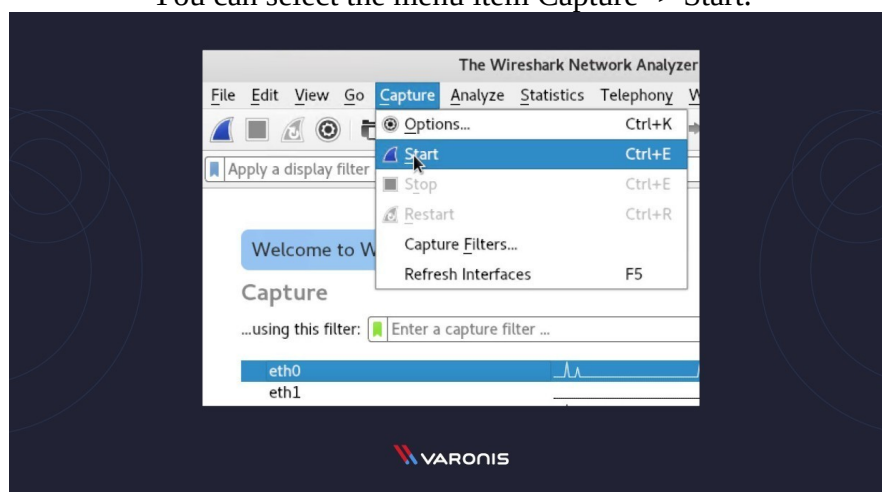


You can select one or more of the network interfaces using “shift left-click.” Once you have the network interface selected, you can start the capture, and there are several ways to do that.

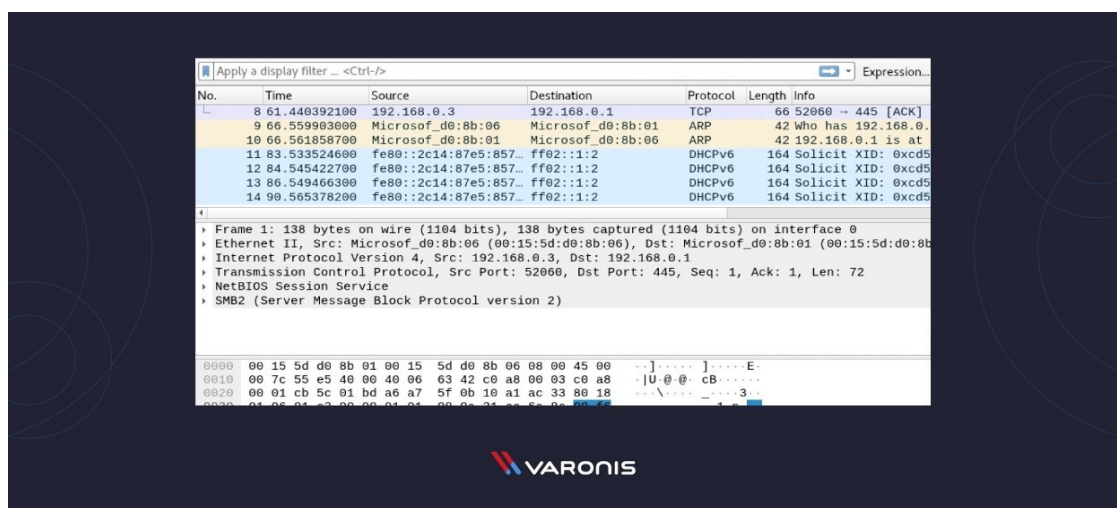
Click the first button on the toolbar, titled “Start Capturing Packets.”



You can select the menu item Capture -> Start.



Or you could use the keystroke Control – E.
During the capture, Wireshark will show you the packets that it captures in real-time.



Once you have captured all the packets you need, you use the same buttons or menu options to stop the capture.

Best practice says that you should stop Wireshark packet capture before you do analysis.

Analyzing Data Packets on Wireshark

Wireshark shows you three different panes for inspecting packet data. The Packet List, the top pane, is a list of all the packets in the capture. When you click on a packet, the other two panes change to show you the details about the selected packet. You can also tell if the packet is part of a conversation. Here are some details about each column in the top pane:

- **No.:** This is the number order of the packet that got captured. The bracket indicates that this packet is part of a conversation.
- **Time:** This column shows you how long after you started the capture that this packet got captured. You can change this value in the Settings menu if you need something different displayed.
- **Source:** This is the address of the system that sent the packet.
- **Destination:** This is the address of the destination of that packet.
- **Protocol:** This is the type of packet, for example, TCP, DNS, DHCPv6, or ARP.
- **Length:** This column shows you the length of the packet in bytes.
- **Info:** This column shows you more information about the packet contents, and will vary depending on what kind of packet it is.

Packet Details, the middle pane, shows you as much readable information about the packet as possible, depending on what kind of packet it is. You can right-click and create filters based on the highlighted text in this field. The bottom pane, Packet Bytes, displays the packet exactly as it got captured in hexadecimal. When you are looking at a packet that is part of a conversation, you can right-click the packet and select Follow to see only the packets that are part of that conversation.

Wireshark Filters

One of the best features of Wireshark is the Wireshark Capture Filters and Wireshark Display Filters. Filters allow you to view the capture the way you need to see it so you can troubleshoot the issues at hand. Here are several filters to get you started.

Wireshark Capture Filters

Capture filters limit the captured packets by the filter. Meaning if the packets don't match the filter, Wireshark won't save them. Here are some examples of capture filters:

host *IP-address*: this filter limits the capture to traffic to and from the IP address

net 192.168.0.0/24: this filter captures all traffic on the subnet.

dst host *IP-address*: capture packets sent to the specified host.

port 53: capture traffic on port 53 only.

port not 53 and not arp: capture all traffic except DNS and ARP traffic

Wireshark Display Filters

Wireshark Display Filters change the view of the capture during analysis. After you have stopped the packet capture, you use display filters to narrow down the packets in the Packet List so you can troubleshoot your issue.

The most useful (in my experience) display filter is:

`ip.src==IP-address and ip.dst==IP-address`

This filter shows you packets from one computer (`ip.src`) to another (`ip.dst`). You can also use `ip.addr` to show you packets to and from that IP. Here are some others:

`tcp.port eq 25`: This filter will show you all traffic on port 25, which is usually SMTP traffic.

`icmp`: This filter will show you only ICMP traffic in the capture, most likely they are pings.

`ip.addr != IP_address`: This filter shows you all traffic except the traffic to or from the

specified computer.

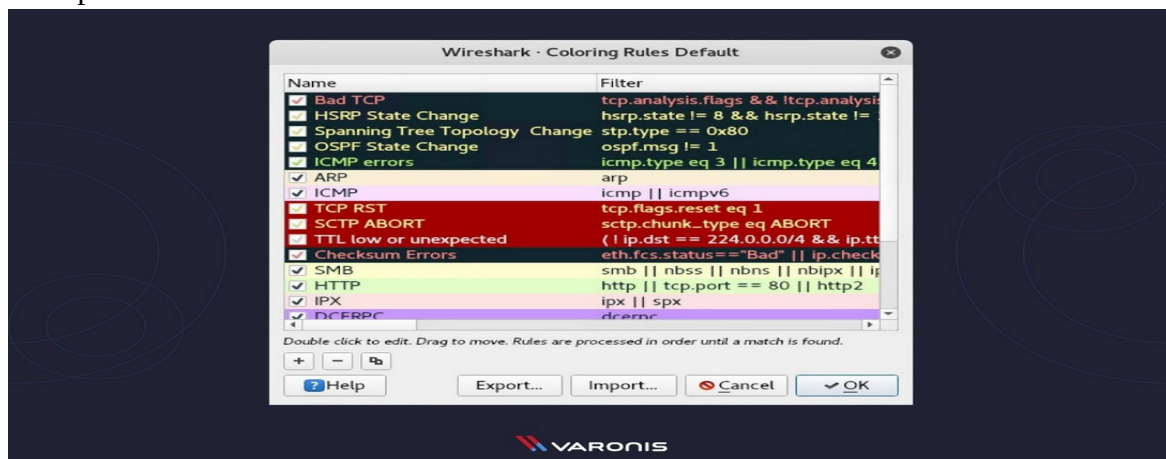
Analysts even build filters to detect specific attacks, like this filter to detect the [Sasser worm](#):
ls_ads.opnum==0x09

Additional Wireshark Features

Beyond the capture and filtering, there are several other features in Wireshark that can make your life better.

Wireshark Colorization Options

You can setup Wireshark so it colors your packets in the Packet List according to the display filter, which allows you to emphasize the packets you want to highlight. Check out some examples here.



Wireshark Promiscuous Mode

By default, Wireshark only captures packets going to and from the computer where it runs. By checking the box to run Wireshark in Promiscuous Mode in the Capture Settings, you can capture most of the traffic on the LAN.

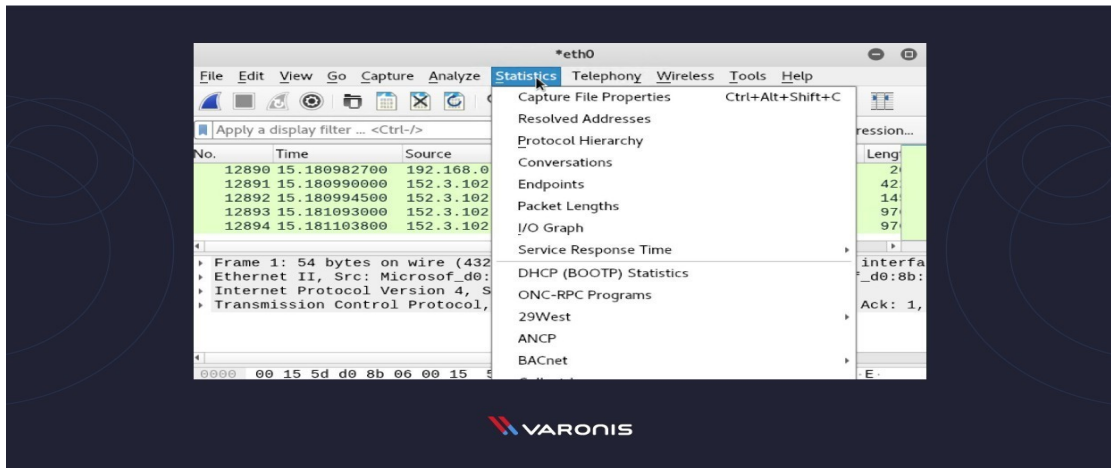
Wireshark Command Line

Wireshark does provide a [Command Line Interface \(CLI\)](#) if you operate a system without a GUI. Best practice would be to use the CLI to capture and save a log so you can review the log with the GUI. Wireshark Commands

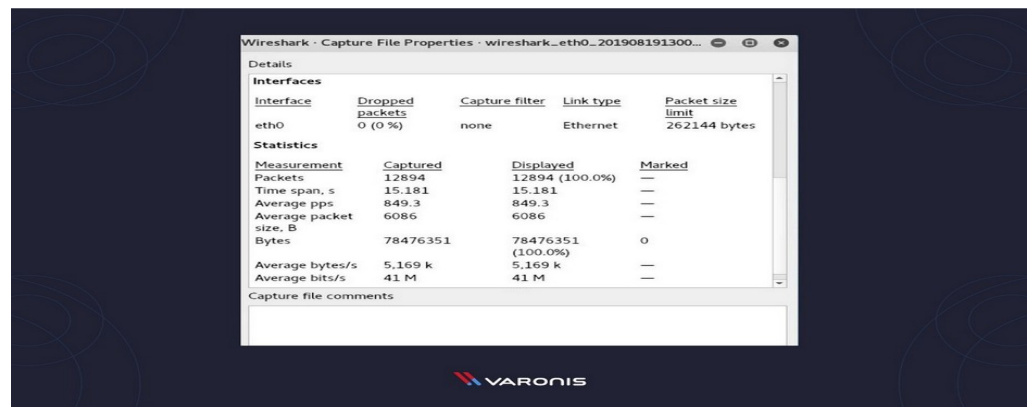
- wireshark : run Wireshark in GUI mode
- wireshark -h : show available command line parameters for Wireshark
- wireshark -a duration:300 -i eth1 -w wireshark. : capture traffic on the Ethernet interface 1 for 5 minutes. -a means automatically stop the capture, -i specifics which interface to capture

Metrics and Statistics

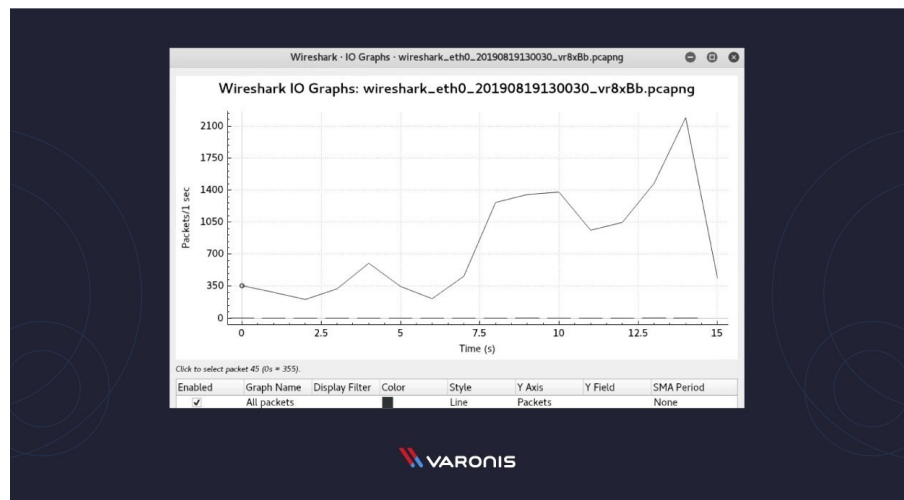
Under the Statistics menu item, you will find a plethora of options to show details about your capture.



Capture File Properties:



Wireshark I/O Graph:



WIRESHARK/SNORT LINKS FOR CODE

<https://github.com/sujay-mahadik/CL7/blob/master/ICS/Assignment4/README.md>

CONCLUSION: Thus, we have implemented wireshark successfully.

QUESTIONS:

1. What should I look for in Wireshark capture?
2. How do you analyze packet captures?