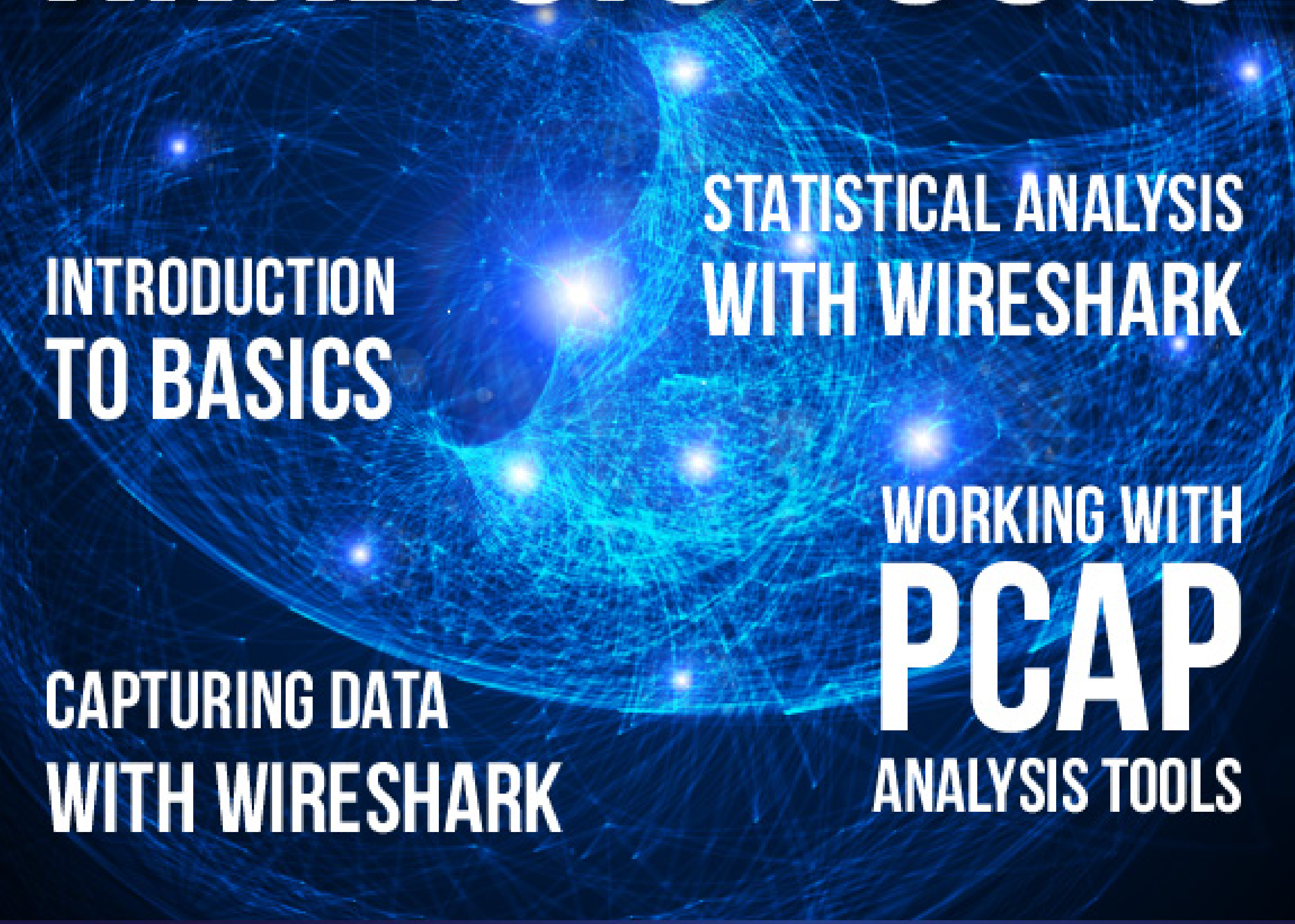
eForensics Magazine

Vol.03 NO. 10

COMPUTER

LEARN PACKET ANALYSIS WITH WIRESHARK AND PCAP ANALYSIS TOOLS



STATISTICAL ANALYSIS WITH WIRESHARK

INTRODUCTION TO BASICS

WORKING WITH PCAP ANALYSIS TOOLS

CAPTURING DATA WITH WIRESHARK

Issue 10/2014 (36) November ISSN 2300-6986 eForensics Magazine TEAM

Editor:
Joanna Kretowicz
joanna.kretowicz@eforensicsmag.com

Betatesters/Proofreaders:
Olivier Caleff, Kishore P.V., JohanScholtz, Mark Dearlove, Massa Danilo, Andrew J. Levandoski, Robert E. Vanaman, Tom Urquhart, M1ndl3ss, Henrik Becker, JAMES FLEIT, Richard C Leitz Jr

Senior Consultant/Publisher:
Paweł Marciniak

CEO: Ewa Dudzic
ewa.dudzic@software.com.pl

Marketing Director: Joanna Kretowicz
jaonna.kretowicz@eforensicsmag.com

Art Director: Ireneusz Pogroszewski
ireneusz.pogroszewski@software.com.pl

DTP: Ireneusz Pogroszewski

Publisher: Hakin9 Media Sp. z o.o. SK
02-676 Warszawa, ul. Postępu 17D
Phone: 1 917 338 3631
www.eforensicsmag.com

DISCLAIMER!

The techniques described in our articles may only be used in private, local networks. The editors hold no responsibility for misuse of the presented techniques or consequent data loss.

TABLE OF CONTENTS

PACKET ANALYSIS

by Eric A. Vanderburg

Almost every computer today is connected. Their communication with others takes the form of packets which can be analyzed to determine the facts of a case. Packet sniffers are also called as network analyzers as it helps in monitoring every activity that is performed over the Internet. The information from packet sniffing can be used to analyze the data packets that uncover the source of problems in the network.

The important feature of packet sniffing is that it captures data that travels through the network, irrespective of the destination. A log file will be generated at the end of every operation performed by the packet sniffer and the log file will contain the information related to the packets. Every packet has a header and body, where the header contains information about the source of the packet and the body contains the actual information about the transfer. There are packet sniffer tools that are available online and many of them are open source tools and hence they are available free of cost. How, when and where should this be performed to collect the best data in a defensible manner? Attend this workshop to find out.

MODULE 1 – BASICS

WHAT IS PACKET ANALYSIS?

Investigations cannot always be contained to a single computer, especially with the way systems are connected these days. Right now, your computer may be connected to dozens of different computers, some to check for software updates, others to gather tweets, email, or RSS feeds. Some connections could be used to authenticate to a domain or access network resources. Now consider an investigation and the potential importance this information could have to it.

Network communication over an Internet Protocol (IP) network can best be understood as a set of packets that form a communication stream. A machine may send and receive thousands of packets per minute and computer networks are used to send these packets to their destination. Packet capture tools can be used to analyze this communication to determine how a computer or user interacted with other devices on the network. Packet analysis can capture these packets so that they can be reviewed to determine what communication took place.

Packet analysis is called as packet sniffing or protocol analysis. A tool that is used for packet analysis is called packet sniffer or packet capture tool. It captures raw data across the wire which helps in analyzing which parties are communicating on the network, what data is flowing, how much data is being transmitted and what network services are in use.

PACKET SNIFFING PROCESS

Packet sniffing can be divided into three steps. The first step is collection when the software gathers all data traversing the network card it is bound to. Next, the data is converted to a form that the program can read and lastly, the program presents the data to be analyzed and can perform pre-programmed analysis techniques on the data.

OSI NETWORK MODEL

Before you can analyze packets, you need to understand how network communication takes place. The OSI network model is a conceptual framework that describes that activities performed to communicate on a network.

TOOLS

There are various packet sniffing tools available on the market. Some popular packet capture tools include Wireshark, Network Miner and NetWitness Investigator, which we will see in detail. All three of these tools are free to download and use and they can be operated in both command line program format and GUI format.

Of the three, Wireshark is the most popular packet sniffer tool that is used worldwide for its ease of installation, ease of use, etc. More importantly, it is an open source tool that is available free of cost. The tool also provides advanced options that will enable forensic investigator or network administrators to delve deep in the packets and capture information. It supports operating systems and numerous protocols, and media types.

There are numerous packet sniffer tools available for network administrators to analyze and understand the traffic flow across the network. It is always difficult to zero down on the best of the lot as almost of them perform required functions seamlessly. Still, there are factors in which they can be ranked and classified as the top packet sniffing tools. The following three tools are identified to be the best in the market, already serving millions of computers from identifying serious threats. Let's get in detail with each of the three packet sniffing tools and understand why they are ranked in such order.

WIRESHARK

Wireshark is a popular open source packet sniffer that performs functions such as network troubleshooting, data analysis, protocol development, etc. The tool uses latest available platforms and forensic investigator or network administrator interface toolkit for serving network administrators. The development version of Wireshark uses Qt while the current releases use GTK+ toolkit. The major advantage of using Wireshark is that it supports multiple platforms, operating systems and protocols. Wireshark comes in both graphical forensic investigator or network administrator interface format and command mode format. Wireshark includes network interface controllers that make it possible for the traffic flowing across the network to be captured via packets. Otherwise, only specified data that is routed to a destination will be captured.

Wireshark supports various protocols and media types. The approximate number of protocols supported by Wireshark is more than 900, and this count goes on increasing as and when an update is released. The primary reason for the increase in count of supported protocols is the open source nature of the tool. Developer has the freedom to develop code for including their new protocol into Wireshark. The Wireshark development team reviews the code that you send and include them in the tool. This makes it possible for protocol to be supported by Wireshark. Also, Wireshark supports major operating systems ranging from Windows; MAC to Linux-based operating systems.

The other major reason for Wireshark to remain on top of a user's list of best packet sniffers is its ease of use. The graphical user interface of the tool is one of the simplest and easiest, available in the online world. The menus are clear with a simple layout, and raw data are represented graphically. This makes it easier for novices to get along with the tool in the early stages of their career. The common problem that users face when using open source software is a lack of proper program support. Wireshark has a highly active forensic investigator or network administrator community that can be ranked as the best among the open source projects. The development team also provides an email subscription of forensic investigator or network administrators on latest updates and FAQs.

Wireshark is very easy to install, and the required system configuration is very minimal as well. Wireshark requires a minimum of 400 MHz of processor speed and 60 MB of free storage. The system should have WinPCap capture driver and a network interface card that supports promiscuous mode and this requires user to have administrator access on the system being used.

Once you are sure that your system has the given configuration, you can install the tool in very short time. Since there will not be data for the first time you open Wireshark, it will not be easier to judge the forensic investigator or network administrator interface.

Installing Wireshark tool is as simple as installing other software in the Windows system. All you need to do is double click the executable file for the installer to open up. Agree to the terms and conditions and select the components you need to be installed along with the packet sniffing tool. Certain components are selected by default, and they are enough for basic operations. Ensure that you select the Install WinPCap option and verify that the WinPCap installation window is displayed some time after Wireshark main installation has started. When the installation is complete, open the tool and select Capture button from the main drop down menu and select interfaces from which you need data to be captured. This will initiate your first data capture using Wireshark, and the main window will then be filled with data that can be used by the user.

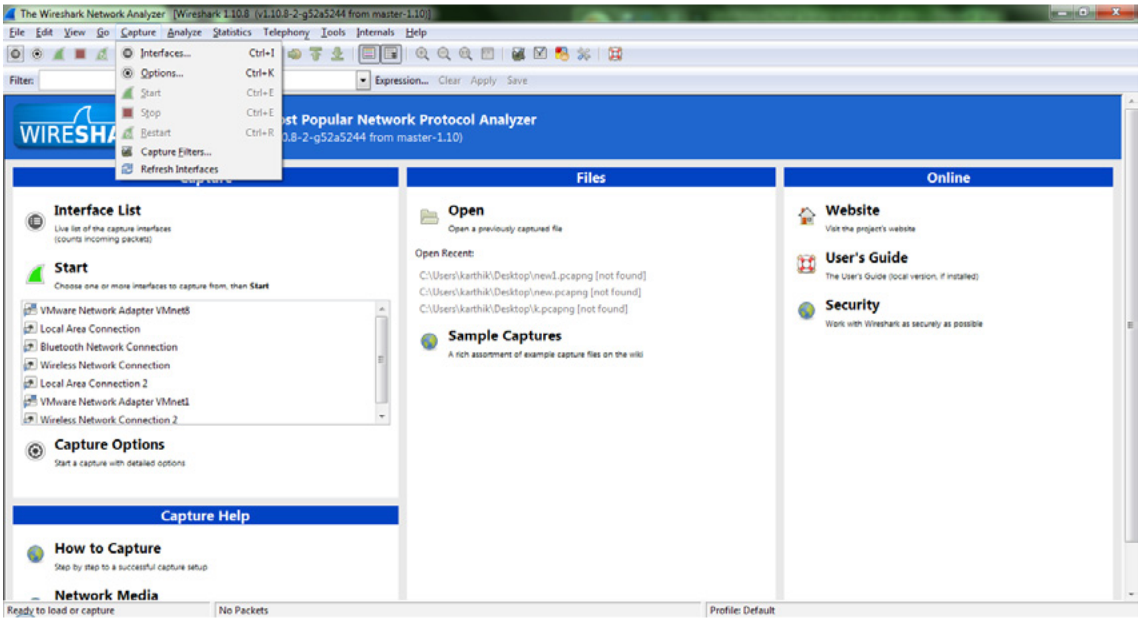
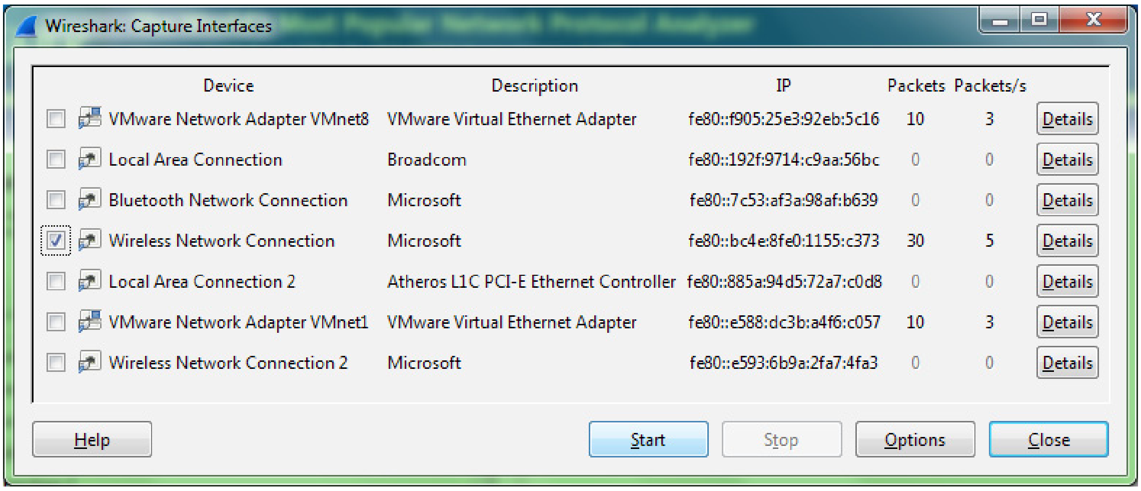

The Wireshark Network Analyzer [Wireshark 1.10.8 (v1.10.8-2-g52a5244 from master-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Interfaces... Ctrl+I
Options... Ctrl+K
Start Ctrl+E
Stop Ctrl+E
Restart Ctrl+R
Capture Filters...
Refresh Interfaces
Filter:
Expression... Clear Apply Save
WIRESHA
st Popular Network Protocol Analyzer
0.8-2-g52a5244 from master-1.10)
Files
Online
Interface List
Live list of the capture interfaces (counts incoming packets)
Start
Choose one or more interfaces to capture from, then Start
VMware Network Adapter VMnet8
Local Area Connection
Bluetooth Network Connection
Wireless Network Connection
Local Area Connection 2
VMware Network Adapter VMnet1
Wireless Network Connection 2
Capture Options
Start a capture with detailed options
Capture Help
How to Capture
Step by step to a successful capture setup
Network Media
Open
Open a previously captured file
Open Recent:
C:\Users\karthik\Desktop\new1.pcapng [not found]
C:\Users\karthik\Desktop\new.pcapng [not found]
C:\Users\karthik\Desktop\k.pcapng [not found]
Sample Captures
A rich assortment of example capture files on the wiki
Website
Visit the project's website
User's Guide
The User's Guide (local version, if installed)
Security
Work with Wireshark as securely as possible
Ready to load or capture
No Packets
Profile: Default

Figure 1. *Home Window of Wireshark*

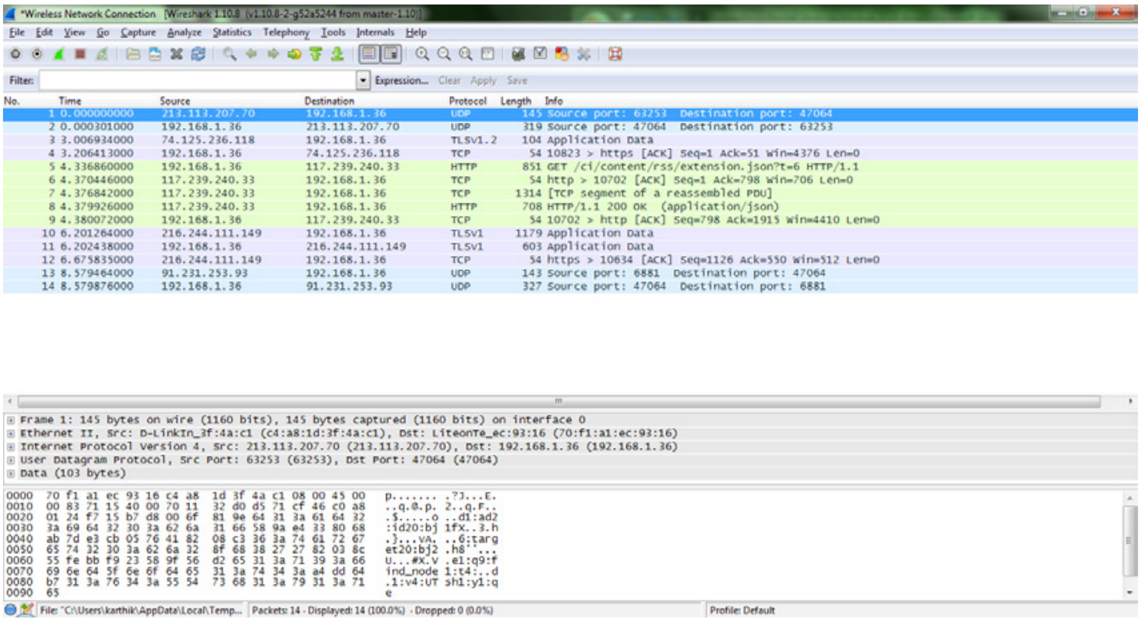


	Device	Description	IP	Packets	Packets/s	
□	VMware Network Adapter VMnet8	VMware Virtual Ethernet Adapter	fe80::f905:25e3:92eb:5c16	10	3	Details
□	Local Area Connection	Broadcom	fe80::192f:9714:c9aa:56bc	0	0	Details
□	Bluetooth Network Connection	Microsoft	fe80::7c53:af3a:98af:b639	0	0	Details
☑	Wireless Network Connection	Microsoft	fe80::bc4e:8fe0:1155:c373	30	5	Details
□	Local Area Connection 2	Atheros L1C PCI-E Ethernet Controller	fe80::885a:94d5:72a7:c0d8	0	0	Details
□	VMware Network Adapter VMnet1	VMware Virtual Ethernet Adapter	fe80::e588:dc3b:a4f6:c057	10	3	Details
□	Wireless Network Connection 2	Microsoft	fe80::e593:6b9a:2fa7:4fa3	0	0	Details

Figure 2. Selecting Interfaces

The main window of Wireshark is where the data that are collected are presented to the forensic investigator or network administrator. Hence, this will be a place where most of the time in the tool will be spent. The main window is broken down into three panes that are interlinked with each other.

The three panes are packet list pane, packet details pane and packet bytes pane. The packet list displays the packets that are available for forensic investigator or network administrator analysis. On selecting packet, the corresponding packet details are displayed in the packet details pane. The corresponding size of the packets will be displayed in the packet bytes pane. The packet list pane displays the packet number and the time at which the packet was captured by the tool. It also displays the source and destination of the packet and other information related to the packet such as packet protocol, etc. The packet bytes pane displays the raw data in the same form as it was originally captured and cannot be of more use. More information about Wireshark can be found at *https://www.wireshark.org/*. The tool can also be downloaded from the site.


*Wireless Network Connection [Wireshark 1.10.8 (v1.10.8-2-g52a5244 from master-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save

```
No.   Time          Source             Destination        Protocol  Length  Info
    1 0.000000000   213.113.207.70     192.168.1.36       UDP        145 Source port: 63253  Destination port: 47064
    2 0.000301000   192.168.1.36       213.113.207.70     UDP        319 Source port: 47064  Destination port: 63253
    3 3.006934000   74.125.236.118     192.168.1.36       TLSv1.2    104 Application Data
    4 3.206413000   192.168.1.36       74.125.236.118     TCP         54 10823 > https [ACK] Seq=1 Ack=51 Win=4376 Len=0
    5 4.336860000   192.168.1.36       117.239.240.33     HTTP       851 GET /ci/content/rss/extension.json?t=6 HTTP/1.1
    6 4.370446000   117.239.240.33     192.168.1.36       TCP         54 http > 10702 [ACK] Seq=1 Ack=798 Win=706 Len=0
    7 4.376842000   117.239.240.33     192.168.1.36       TCP       1314 [TCP segment of a reassembled PDU]
    8 4.379926000   117.239.240.33     192.168.1.36       HTTP       708 HTTP/1.1 200 OK  (application/json)
    9 4.380072000   192.168.1.36       117.239.240.33     TCP         54 10702 > http [ACK] Seq=798 Ack=1915 Win=4410 Len=0
   10 6.201264000   216.244.111.149    192.168.1.36       TLSv1     1179 Application Data
   11 6.202438000   192.168.1.36       216.244.111.149    TLSv1      603 Application Data
   12 6.675835000   216.244.111.149    192.168.1.36       TCP         54 https > 10634 [ACK] Seq=1126 Ack=550 Win=512 Len=0
   13 8.579464000   91.231.253.93      192.168.1.36       UDP        143 Source port: 6881  Destination port: 47064
   14 8.579876000   192.168.1.36       91.231.253.93      UDP        327 Source port: 47064  Destination port: 6881
```

```
⊞ Frame 1: 145 bytes on wire (1160 bits), 145 bytes captured (1160 bits) on interface 0
⊞ Ethernet II, Src: D-LinkIn_3f:4a:c1 (c4:a8:1d:3f:4a:c1), Dst: LiteonTe_ec:93:16 (70:f1:a1:ec:93:16)
⊞ Internet Protocol Version 4, Src: 213.113.207.70 (213.113.207.70), Dst: 192.168.1.36 (192.168.1.36)
⊞ User Datagram Protocol, Src Port: 63253 (63253), Dst Port: 47064 (47064)
⊞ Data (103 bytes)
```

```
0000  70 f1 a1 ec 93 16 c4 a8  1d 3f 4a c1 08 00 45 00   p....... .?J...E.
0010  00 83 71 15 40 00 70 11  32 d0 d5 71 cf 46 c0 a8   ..q.@.p. 2..q.F..
0020  01 24 f7 15 b7 d8 00 6f  81 9e 64 31 3a 61 64 32   .$.....o ..d1:ad2
0030  3a 69 64 32 30 3a 62 6a  31 66 58 9a e4 33 80 68   :id20:bj 1fX..3.h
0040  ab 7d e3 cb 05 76 41 82  08 c3 36 3a 74 61 72 67   .}...vA. ..6:targ
0050  65 74 32 30 3a 62 6a 32  8f 68 38 27 27 82 03 8c   et20:bj2 .h8''...
0060  55 fe bb f9 23 58 9f 56  d2 65 31 3a 71 39 3a 66   U...#X.V .e1:q9:f
0070  69 6e 64 5f 6e 6f 64 65  31 3a 74 34 3a a4 dd 64   ind_node 1:t4:..d
0080  b7 31 3a 76 34 3a 55 54  73 68 31 3a 79 31 3a 71   .1:v4:UT sh1:y1:q
0090  65                                                 e
```

File: "C:\Users\karthik\AppData\Local\Temp...
Packets: 14 · Displayed: 14 (100.0%) · Dropped: 0 (0.0%)
Profile: Default

Figure 3. *Main Window*

NETWORK MINER

Network Miner is a packet analysis tool that also includes the ability to perform packet sniffing. It is available for Windows, Linux and MAC OS. It is passive packet capturing tool that detects operating systems, traffic, and network ports. On the contrary, Wireshark is an active packet capturing tool.

The difference between an active and passive packet sniffing tool is that in active sniffing, the sniffing tool sends the request over the network and uses the response to capture packets while passive sniffing does not send request for receiving a response. It simply scans the traffic without getting noticed in the network.

The places where passive sniffing comes in handy are radar systems, telecommunication and medical equipment, and many others. Another difference between active and passive sniffing technique is that the latter uses host-centric approach which means it uses hosts for sorting out data while active sniffing uses packets. Similar to Wireshark, network miner also comes with easy to use interface, simple installation and ease of use.

NETWITNESS INVESTIGATOR

The NetWitness Investigator is the packet sniffing tool that is a result of 10 years of research and development has been used in most complex threat environments. The Netwitness investigator has been used only with critical environments for so long, but the company has released the free version of the software, making it available for the public as well. The investigator captures live packets from both wireless and wired network interfaces. It supports most major packet capture systems. The free version of the tool allows 25 simultaneous users to capture data up to a maximum of 1 GB.

The tool has other interesting features such as effectively analyzing the data in layers of networking, from users email addresses, files, IPv6 support, full content searching, exporting the information collected in PCAP format, and others. As the number of users using the internet has grown over the years, it was important for the Internet Engineering Task Force to come up with unique IP addresses that can be used for new devices. IPv6 will replace the current generation IPv4 protocol. The introduction of IPv6 allows increased number of IP addresses which helps more users to communicate over the internet. This is because, IPv4 addresses are only 32 bits long that supports 4.3 billion addresses whereas IPv6 addresses are 128 bits long and supports over hundred trillion and trillion addresses. With new set of protocols used for communication, it is important for the forensic tools to provide support for the protocols for seamless operation. NetWitness Investigator thus provides support for IPv6 which will be the future of all internet communication. Every new release of the tool comes in which many new features that may not be available in other packet sniffing tool. Netwitness investigator requires certain minimum configuration support for installation. The tool can be installed in windows operating system, with at least 1 GB RAM, 1 Ethernet port, a large amount of data storage, etc. The free version of the tool supports only the Windows operating system while the commercial version provides support for Linux as well. One important feature of investigator is that it does not alert forensic investigator or network administrators for problems in network based on known threats. Instead, it catches packets in real time and analyzes the network for differences in behavior and reports the same to the forensic investigator or network administrator immediately. The commercial version of the software brings in more benefits when compared to the free version. Some of the features that are present only in enterprise version are support for Linux platform, remote network monitoring, informer, decoder and automated reporting engine.

HOW PACKET ANALYZERS WORK

Packet analyzers intercept network traffic that travel through the wired and wireless network interfaces that they have access to. The structure of the network along with how network switches and other tools are configured decides what information can be captured. In a switched wired network, the sniffer can capture data from only the switch port it is attached to unless port mirroring is implemented on the switch.

However, with wireless, the packet sniffing tool can capture data from only one channel, unless there are multiple interfaces that allow data to be captured from more than one channel. RFC 5474 is a framework that is used for selection of packets and reporting them. It uses the PSAMP framework which selects packets in statistical methods and exports the packets to the collectors. RFC 5475 describes the various techniques of packet selection that are supported by PSAMP. These frameworks help users perform the processes seamlessly.

The data that is received initially will be in raw format that only the software can understand. It needs to be converted to human readable form for the forensic investigator or network administrator to interpret. The tool performs this operation in the process called conversion. The data can then be analyzed, and necessary information can be obtained. Thus, the place where the fault is present can be identified, and necessary actions can be taken. Normally, there are three basic types of packet sniffing, and they are ARP sniffing, IP sniffing and MAC sniffing.

In ARP sniffing, the information is transferred to the ARP cache of the hosts. The network traffic is directed towards the administrator. In IP sniffing, the information corresponding to an IP address filter is captured. MAC sniffing is similar to IP sniffing except for device sniffing information packets of a particular MAC address.

COMPONENTS OF PACKET SNIFFER

Before delving in detail on how packet sniffers work, it is important to understand the components that are part of the sniffer. The four major parts of a sniffer are hardware, driver, buffer and packet analysis. Most packet sniffers work with common adapters, but some require multi adapters, wireless adapters and others. Before installing the sniffer in the system, diagnose whether the system contains the necessary adapter for the sniffer. Next important component for a sniffer to work is the drive program. Without the driver, the sniffer cannot be installed in the system. Once the sniffer is installed, it requires a buffer that is the storage device for capturing data from the network.

There are two types in which data can be stored in the buffer. In the first method, the data can be stored in the buffer until the storage space runs out. This prevents new data from being stored as there is no storage space. The other method is to replace the old data with new data as and when the buffer overflows. The forensic investigator or network administrator has the option to select buffer storage method. Also, the size of the buffer depends on the EMS memory (Expanded memory specification) of the computer. When the EMS memory of the computer is higher, more data can be stored in the buffer.

The packet analysis is the most essential and core part of the sniffing process as it captures and analyses the data from the packets. Many advanced snipping tools have been introduced of late which allows users to replay the stored contents so that they can be edited and retransmitted based on requirements.

WORKING PRINCIPLE

The working principle of a sniffing tool is very simple. The network interfaces present in the segment will usually have a hardware address, and they can see the data that is transmitted over the physical medium. The hardware address of one network interface is designed to be unique so it should be different from the address of another network interface. Hence, packet that is transmitted over the network will pass through the host machines, but will be ignored by the machines except for the one to which the packet is destined to. However, in practice, this is not always the case because hardware addresses can be changed in software and virtualization technologies are frequently used to generate hardware addresses for virtual machines from a set pool.

In IP networks, each network has a subnet mask, network address and broadcast address. An IP address consists of two parts namely network address and host address. The subnet mask helps in separating the IP address into network and host address. The host address is further broken down as subnet address and host address. The subnet mask identifies the IP address of the system by performing AND operation on netmask. It converts the network bits to 1 and host bits to 0. Any network will have two special reserved host addresses, 0 for network address and 255 for host address. Subnetting a network helps in breaking down bigger networks into smaller multiple networks. Network address is an address that identifies a node in a network. The network addresses are unique within a network and there can be more than one network address within any network. A broadcast address is a special address that is used to transmit messages to multiple recipients. Broadcast addresses help network administrators in verifying successful data transmission over the network. Broadcast address is used by various clients and the most important of them are dynamic host configuration protocol and bootstrap protocol that use the address to transmit server requests.

When a network interface card is configured, it will respond to the target network having addresses that exist in the same network as designated by the subnet mask and network address. This is how packet sniffing works and the three basic steps of packet sniffing are collection, conversion and analysis.

COLLECTION

The first step in packet sniffing technique is the collection of raw data from the packets that travel along the network. The sniffer will switch the required network interface to promiscuous mode that will enable data packets from hosts in the system to be captured. When this mode is turned off, only the packets that are addressed to a particular interface will be captured. When this mode is turned on, all packets received on a particular interface will be captured. Packets that are received by the NIC are stored in a buffer and then processed.

It is important for forensic investigator or network administrator to understand where to fit in a packet sniffer for it to capture packets effectively. This is called tapping the wire or getting on the wire in which the packet sniffer is placed in the correct physical location. Placing the sniffer tool at the right position is as tough as analyzing the packets for information. Since there are hardware devices in connecting a network, placing the tool at wrong position will not fetch packets. As seen before, the network interface card should be in promiscuous mode for capturing the data that is flowing across the network. Usually, the operating systems do not allow the forensic investigator or network administrators to turn promiscuous mode on. Individual forensic investigator or network administrator privileges are required to enable this mode, and if that is not possible, packet sniffing cannot be carried out in that particular network. It is much easier to sniff around the packets in a network that has hubs installed because, when traffic is sent over a hub; it traverses to every port that is connected to the hub. Hence, once you connect the packet sniffer to an empty port of the hub, you will receive packets travelling across the hub.

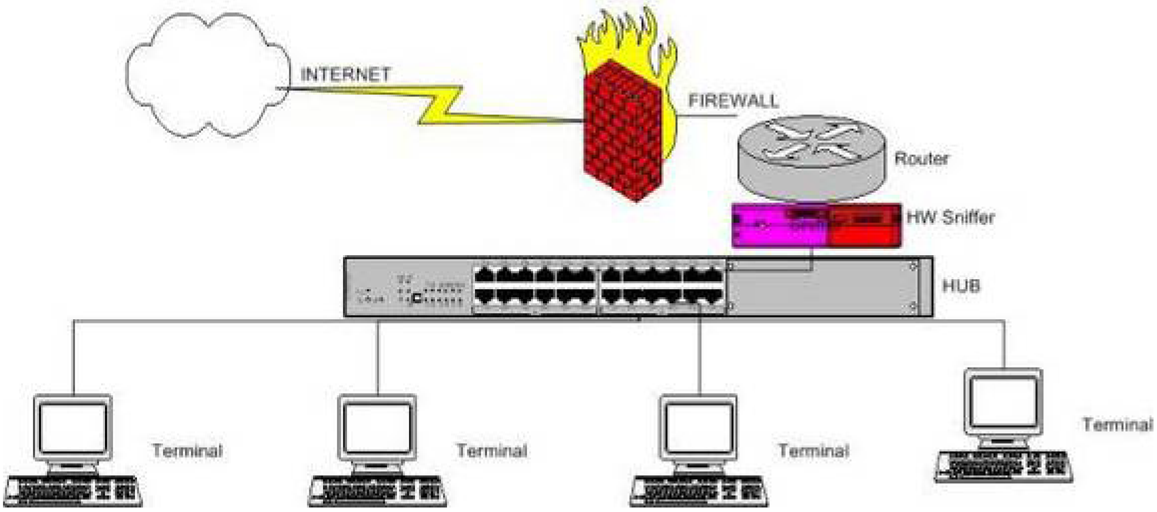


Figure 4. *Example of one location where packet sniffer can be placed. Source: http://www.windowsecurity.com/articles-tutorials/misc_network_security/Network_Analyzers_Part1.html*

The most common type of network is a switched network as it allows broadcast, multicast and unicast traffic. It also supports full duplex communication in which the host system can send and receive packets simultaneously. This increases the complexity of setting up packet sniffing tool in a switched environment. Also, the traffic that is sent to the broadcast address and the host machine can only be captured. Hence the visibility window for the packet sniffer is far lower in a switched environment.

There are three common types of capturing data in a switched network and they are port mirroring, ARP Cache poisoning and hubbing out. Port mirroring is the simplest of the three ways by which packets can be captured. The forensic investigator or network administrator must be able to access the command line interface of the switch for enabling port mirroring. As a forensic investigator or network administrator, you need to do is enter a command in the command line interface which enables the switch to copy traffic from one port to another.

Another method of capturing the data in switched environment is hubbing out, which is a technique in which the target device and the analyzer are localized within a network by connecting them directly to the hub. In hubbing out method, forensic investigator or network administrator needs a hub and some network cables to connect the target to the hub. First, unplug the host from the network, followed by plugging the target and the analyzer to the hub. Then, connect the hub to a network switch which enables the data to be transferred to the nub and simultaneously to the analyzer.

In the seven layer OSI model, the second layer contains MAC addresses while the third layer contains IP addresses, and both these addresses should be used in conjunction for network data transfer. The switches are present in the second layer and hence, the MAC addresses should be converted to IP addresses and vice versa for data transfer. This translation process is called as address resolution protocol. Whenever a computer needs to transfer data to another computer, an ARP request is sent to the switch, which then sends ARP broadcast packet to the systems that are connected to the computer. The target computer which has the equivalent IP address responds to the request by sending out its MAC address. This information is then stored in the cache so that future connections can use this data without sending out new request. This method can easily capture the traffic across the network, and hence ARP cache poisoning is otherwise called as ARP spoofing.

CONVERSION

In this step, the raw data that is captured in the collection step is converted to human readable form. The converted data can only be analyzed for information that can be useful for the network administrator. The work of most of the command prompt packet sniffers stop at this point of time and the remaining work are left over to the end forensic investigator or network administrator.

ANALYSIS

The third and final step of packet sniffing technique is analysis in which the data present in human readable form is analyzed to gather required information. Multiple packets are compared to obtain the behavior of the network. The GUI based packet sniffing tools are handy at this time as they have comparison tools as well.

All these methods ensure that the right packets are captured as part of packet sniffing technique. The network problems can be analyzed, and necessary actions can be taken by the network administrators to prevent further problem in the network. The three packet sniffing tools mentioned above are used widely among the audiences around the globe.

The goal of analyzing data in computer forensics is to identify and explore the digital content for preserving and recovering the original data that is present. There are various instances where computer forensics has come in handy for network administrators. Live analysis is the most effective technique as it ensures that the encrypted file systems can also be captured and analyzed.

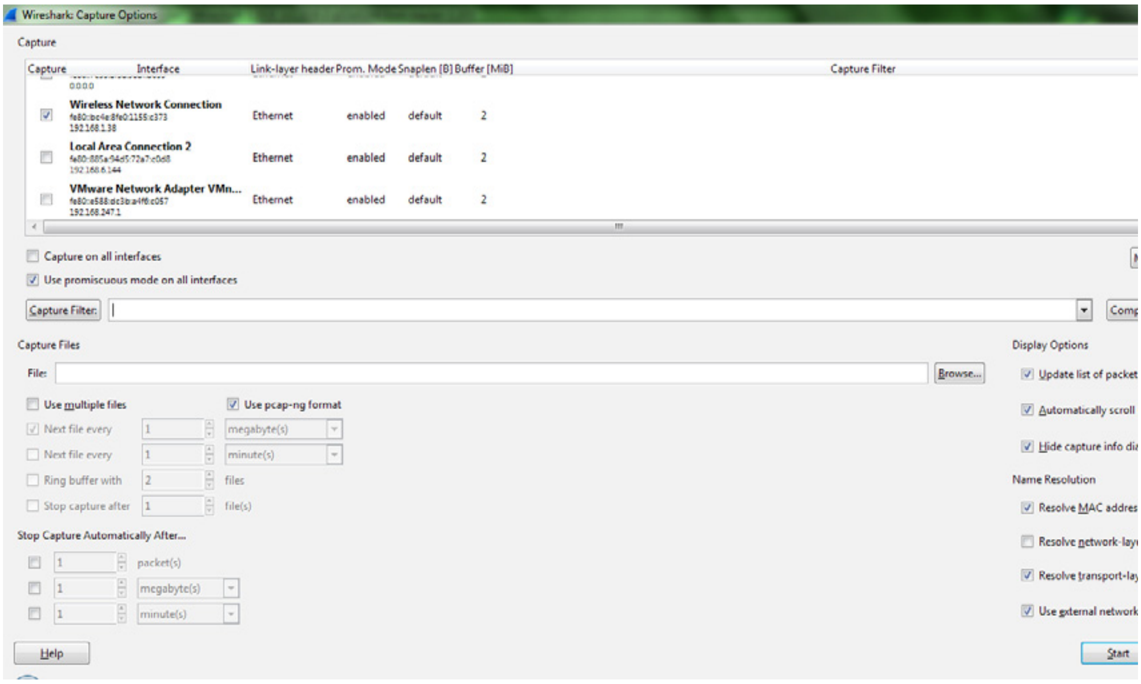
Module 1 of packet analysis introduced you to the basic definitions of what a packet is and how packets form the base for all network communication. The high level information on what packet sniffing technique is and the three major packet sniffer tools available in the online world were also discussed in module 1. As said before, the Wireshark is the most sought after packet sniffer tool among the audiences due to various reasons ranging from easy installation to ease of use. In this module, let's discuss the various processes involved in capturing and analyzing packets using Wireshark. Even though data analysis seems to be an easy task, there are lots more in it when it comes to analyzing packets for identifying problems in the network. Hence it is necessary for every one of you to know all the processes involved in the entire packet sniffing technique. How, when and where should this be performed to collect the best data in a defensible manner? Attend this workshop to find out.

MODULE 2 – CAPTURING DATA WITH WIRESHARK

CAPTURING PACKETS

It is a known fact that installing Wireshark tool in computers is an easy task as all you need to do is to download and double click the executable file for the installation to begin. It will only take not more than 3 minutes to complete the installation, after which your first dream packet capture can begin. Before getting in detail of the technique of capturing packets, it is important to understand the components and the configuration settings that are essential for proper packet capture. As discussed in module 1, there are various places where the packet sniffer tool can be placed to capture packets. It depends on various factors such as presence of various hardware networking devices, problems in the network etc. Decide where you want your packet sniffer tool to sit and fetch you information that is transmitted across the network. Only then, you can proceed with other steps in analyzing packets. This is common for all packet sniffing tools and not only Wireshark. It is also important to download and install the appropriate Wireshark installation file as there are various formats available for various operating system configurations. There are separate Wireshark package files for 32 bit and 64 bit version of Windows operating system, Linux and MAC operating systems. Download and install the correct package for the packet sniffer tool to function seamlessly. Also, during installation of the file, users will be given the option of selecting the various components that needs to be installed along with the package. It is important to select the required components as well. If unsure, leave it to the default components that are selected by the program itself as it is sufficient for basic level packet sniffing. Once the installation process is completed, it is time to open the tool and start its operation. In Linux environment, the Wireshark entry will be placed in the desktop menu itself. The tool can also be opened by running Wireshark from the root shell of the emulator. Opening the software in Windows is as simple as opening any other executable files. The non GUI version of Wireshark is available for UNIX systems and more information regarding the same can be obtained from various online sources or Wiki community of Wireshark.

On starting the Wireshark tool, click on Capture menu item and select Interfaces option. The Capture Interfaces dialog window opens up where users will be able to select the various interfaces from which the packets should be collected. There are various options for capturing packets and these options can be configured by clicking on the Options button present in Capture Interfaces window. This is the place where the user will have the option to select packets from interfaces which are enabled with promiscuous mode. When this mode is turned OFF, the packets that are corresponding to the particular interface can only be captured, otherwise all packets can be captured. There is also option to select the frame size which will cut down the size of every packet that is received. When this option is not selected, the packets will be captured in full size. Similar to this, there are many other options for the users to customize how packets should be captured. If you are not sure of what the options are and how the results will look like, it is better to leave it with default settings.



Wireshark: Capture Options

Capture

Capture	Interface	Link-layer header	Prom. Mode	Snaplen [B]	Buffer [MiB]	Capture Filter
	0.0.0.0					
☑	**Wireless Network Connection** fe80::bc4e:8fe0:1155:c373 192.168.1.38	Ethernet	enabled	default	2	
□	**Local Area Connection 2** fe80::885a:94d5:72a7:c0d8 192.168.6.144	Ethernet	enabled	default	2	
□	**VMware Network Adapter VMn...** fe80::a588:dc3b:a4f6:c057 192.168.247.1	Ethernet	enabled	default	2	

□ Capture on all interfaces

☑ Use promiscuous mode on all interfaces

Capture Filter: |

Capture Files

File:

□ Use multiple files ☑ Use pcap-ng format

☑ Next file every 1 megabyte(s)

□ Next file every 1 minute(s)

□ Ring buffer with 2 files

□ Stop capture after 1 file(s)

Stop Capture Automatically After...

□ 1 packet(s)

□ 1 megabyte(s)

□ 1 minute(s)

Display Options

☑ Update list of packet

☑ Automatically scroll

☑ Hide capture info di

Name Resolution

☑ Resolve MAC addres

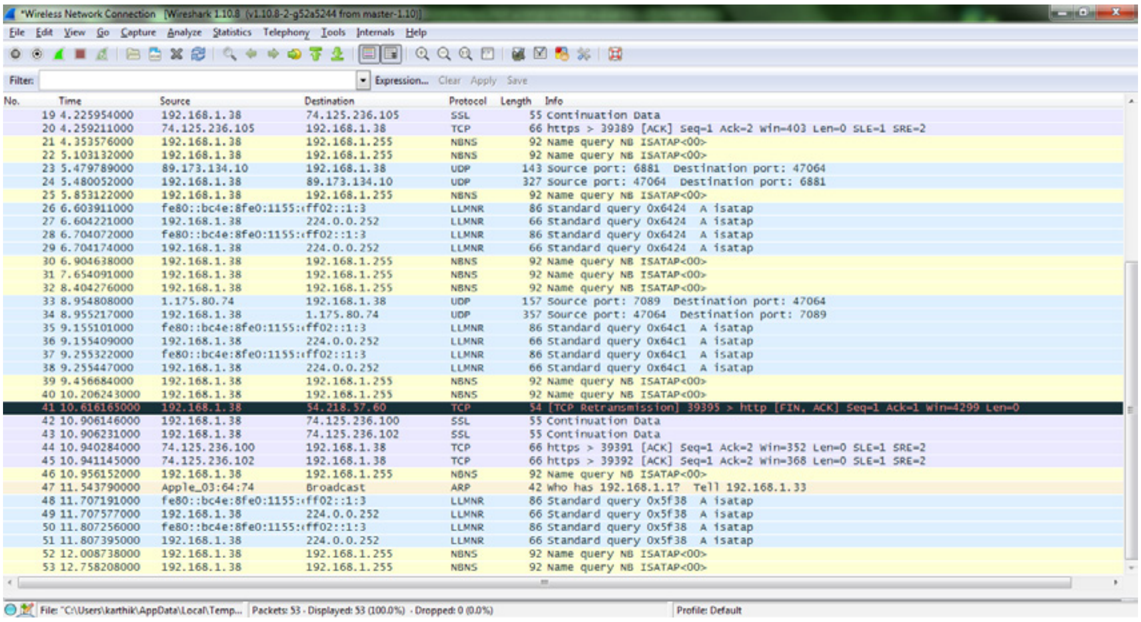
□ Resolve network-lay

☑ Resolve transport-la

☑ Use external network

Figure 1. *Capture Options*

When all the options and settings are configured, it is time to start packet capturing by clicking on Start button. When the packets are captured, Wireshark will show all the packets in the main page. If the destination address of all the packets that are captured ends with 255 or if they are displayed as FFFFFFFF, then the corresponding packets relates to broadcast traffic. Usually, broadcast traffic cannot be used for any analysis and they do not serve any purpose. The packets that are being captured are differentiated with various colors ranging from green, blue and black. The packets captured as part of TCP traffic are denoted in green, DNS packets in dark blue, UDP packets in light blue and TCP packets that have problems in black.



*Wireless Network Connection [Wireshark 1.10.8 (v1.10.8-2-g52a5244 from master-1.10)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save

```
No.     Time         Source                 Destination       Protocol Length Info
     19 4.225954000  192.168.1.38           74.125.236.105    SSL        55 Continuation Data
     20 4.259211000  74.125.236.105         192.168.1.38      TCP        66 https > 39389 [ACK] Seq=1 Ack=2 Win=403 Len=0 SLE=1 SRE=2
     21 4.353576000  192.168.1.38           192.168.1.255     NBNS       92 Name query NB ISATAP<00>
     22 5.103132000  192.168.1.38           192.168.1.255     NBNS       92 Name query NB ISATAP<00>
     23 5.479789000  89.173.134.10          192.168.1.38      UDP       143 Source port: 6881  Destination port: 47064
     24 5.480052000  192.168.1.38           89.173.134.10     UDP       327 Source port: 47064  Destination port: 6881
     25 5.853122000  192.168.1.38           192.168.1.255     NBNS       92 Name query NB ISATAP<00>
     26 6.603911000  fe80::bc4e:8fe0:1155:cff02::1:3          LLMNR      86 Standard query 0x6424  A isatap
     27 6.604221000  192.168.1.38           224.0.0.252       LLMNR      66 Standard query 0x6424  A isatap
     28 6.704072000  fe80::bc4e:8fe0:1155:cff02::1:3          LLMNR      86 Standard query 0x6424  A isatap
     29 6.704174000  192.168.1.38           224.0.0.252       LLMNR      66 Standard query 0x6424  A isatap
     30 6.904638000  192.168.1.38           192.168.1.255     NBNS       92 Name query NB ISATAP<00>
     31 7.654091000  192.168.1.38           192.168.1.255     NBNS       92 Name query NB ISATAP<00>
     32 8.404276000  192.168.1.38           192.168.1.255     NBNS       92 Name query NB ISATAP<00>
     33 8.954808000  1.175.80.74            192.168.1.38      UDP       157 Source port: 7089  Destination port: 47064
     34 8.955217000  192.168.1.38           1.175.80.74       UDP       357 Source port: 47064  Destination port: 7089
     35 9.155101000  fe80::bc4e:8fe0:1155:cff02::1:3          LLMNR      86 Standard query 0x64c1  A isatap
     36 9.155409000  192.168.1.38           224.0.0.252       LLMNR      66 Standard query 0x64c1  A isatap
     37 9.255322000  fe80::bc4e:8fe0:1155:cff02::1:3          LLMNR      86 Standard query 0x64c1  A isatap
     38 9.255447000  192.168.1.38           224.0.0.252       LLMNR      66 Standard query 0x64c1  A isatap
     39 9.456684000  192.168.1.38           192.168.1.255     NBNS       92 Name query NB ISATAP<00>
     40 10.206243000 192.168.1.38           192.168.1.255     NBNS       92 Name query NB ISATAP<00>
     41 10.616165000 192.168.1.38           54.218.57.60      TCP        54 [TCP Retransmission] 39395 > http [FIN, ACK] Seq=1 Ack=1 Win=4299 Len=0
     42 10.906146000 192.168.1.38           74.125.236.100    SSL        55 Continuation Data
     43 10.906231000 192.168.1.38           74.125.236.102    SSL        55 Continuation Data
     44 10.940284000 74.125.236.100         192.168.1.38      TCP        66 https > 39391 [ACK] Seq=1 Ack=2 Win=352 Len=0 SLE=1 SRE=2
     45 10.941145000 74.125.236.102         192.168.1.38      TCP        66 https > 39392 [ACK] Seq=1 Ack=2 Win=368 Len=0 SLE=1 SRE=2
     46 10.956152000 192.168.1.38           192.168.1.255     NBNS       92 Name query NB ISATAP<00>
     47 11.543790000 Apple_03:64:74         Broadcast         ARP        42 Who has 192.168.1.1?  Tell 192.168.1.33
     48 11.707191000 fe80::bc4e:8fe0:1155:cff02::1:3          LLMNR      86 Standard query 0x5f38  A isatap
     49 11.707577000 192.168.1.38           224.0.0.252       LLMNR      66 Standard query 0x5f38  A isatap
     50 11.807256000 fe80::bc4e:8fe0:1155:cff02::1:3          LLMNR      86 Standard query 0x5f38  A isatap
     51 11.807395000 192.168.1.38           224.0.0.252       LLMNR      66 Standard query 0x5f38  A isatap
     52 12.008738000 192.168.1.38           192.168.1.255     NBNS       92 Name query NB ISATAP<00>
     53 12.758208000 192.168.1.38           192.168.1.255     NBNS       92 Name query NB ISATAP<00>
```

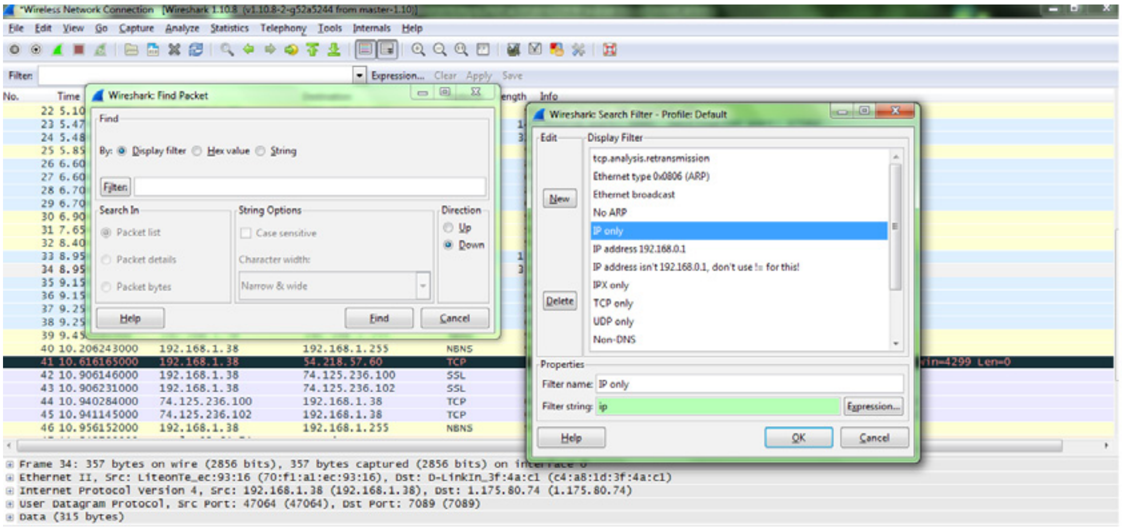
File: "C:\Users\karthik\AppData\Local\Temp... Packets: 53 · Displayed: 53 (100.0%) · Dropped: 0 (0.0%) Profile: Default

Figure 2. *Packets in different colors*

These colors can also be changed based on user preferences in Coloring rules settings page. Select the type of packet for which you need to change the color and click on Edit button the set the color of your choice. Defining colors based on interests will make the task of analyzing packets easier, as there will be hundreds of packets captured at a single time. Once the files are captured, the results can be stored in any format as specified in the tool. The default format in which the files are saved is libpcap format. Now that all the basic concepts related to capturing packets are covered, it is time to get into other concepts such as filtering and marking packets and exporting the PCAP files. The following paragraphs will uncover the basic definitions and processes by which packets can be filtered, marked and exported.

MARKING PACKETS

We have discussed how packets are captured in Wireshark. There are many other interesting options for users to make use of before analyzing the captured packets. These options or settings enable easy access to the packets of particular interest. Since every time the packet sniffing tool is run, very large number of packets will be displayed which will make it really tough to find the packet that will be of most use for the analysis. This concern has been taken into consideration by the Wireshark team after which they have come up with the option of finding and marking packets which would allow the user to mark the packets based on certain criteria. They have also made this process easier for everyone to be able to find and mark packets, even though they have less expertise in using the tool. All the user needs to do is to navigate to Edit menu and click on Find Packet option. The other easy option to open the Find packet dialog is to press CTRL+F from your keyboard, which is the universal shortcut for Find dialog. In the Find packet setting, users are given with three options by which the packets can be searched and found. They are finding the packets by display filter, Hex value and string. The Display filter lets users to select packets based on certain expression they enter in the search field while the Hex value and string options let users to search and find based on hexadecimal value or text. Apart from selecting the search criteria, users can also select the window from which the packets can be found. As seen before, there are three panes on the main window of which the top pane displays the packet list; middle pane displays the packet details while the bottom pane displays the packet bytes. Selecting this option will fetch packets that are found only in the particular pane. User can also select "Case sensitive" option and the direction of searching the packets with either up or down. When all the selections are made, click on Find button to find the first packet that matches the selection criteria. The next and previous packets can be found by clicking CTRL+N or CTRL+B respectively.


*Wireless Network Connection [Wireshark 1.10.8 (v1.10.8-2-g52a5244 from master-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save
No. Time
ength Info
Wireshark: Find Packet
Find
Filter:
Search In
String Options
Direction
Help
Find
Cancel
Wireshark: Search Filter - Profile: Default
Edit
Display Filter
New
Delete
Properties
Expression...
Help
OK
Cancel

Wireshark: Find Packet

By: ● Display filter ○ Hex value ○ String

Search In: ● Packet list ○ Packet details ○ Packet bytes

String Options: □ Case sensitive; Character width: Narrow & wide

Direction: ○ Up ● Down

Wireshark: Search Filter - Profile: Default

Display Filter:
- tcp.analysis.retransmission
- Ethernet type 0x0806 (ARP)
- Ethernet broadcast
- No ARP
- IP only
- IP address 192.168.0.1
- IP address isn't 192.168.0.1, don't use != for this!
- IPX only
- TCP only
- UDP only
- Non-DNS

Filter name: IP only

Filter string: ip

```
22 5.10
23 5.47
24 5.48
25 5.85
26 6.60
27 6.60
28 6.70
29 6.70
30 6.90
31 7.65
32 8.40
33 8.95
34 8.95
35 9.15
36 9.15
37 9.25
38 9.25
39 9.45
40 10.206243000   192.168.1.38      192.168.1.255     NBNS
41 10.616165000   192.168.1.38      54.218.57.60      TCP      win=4299 Len=0
42 10.906146000   192.168.1.38      74.125.236.100    SSL
43 10.906231000   192.168.1.38      74.125.236.102    SSL
44 10.940284000   74.125.236.100    192.168.1.38      TCP
45 10.941145000   74.125.236.102    192.168.1.38      TCP
46 10.956152000   192.168.1.38      192.168.1.255     NBNS
```

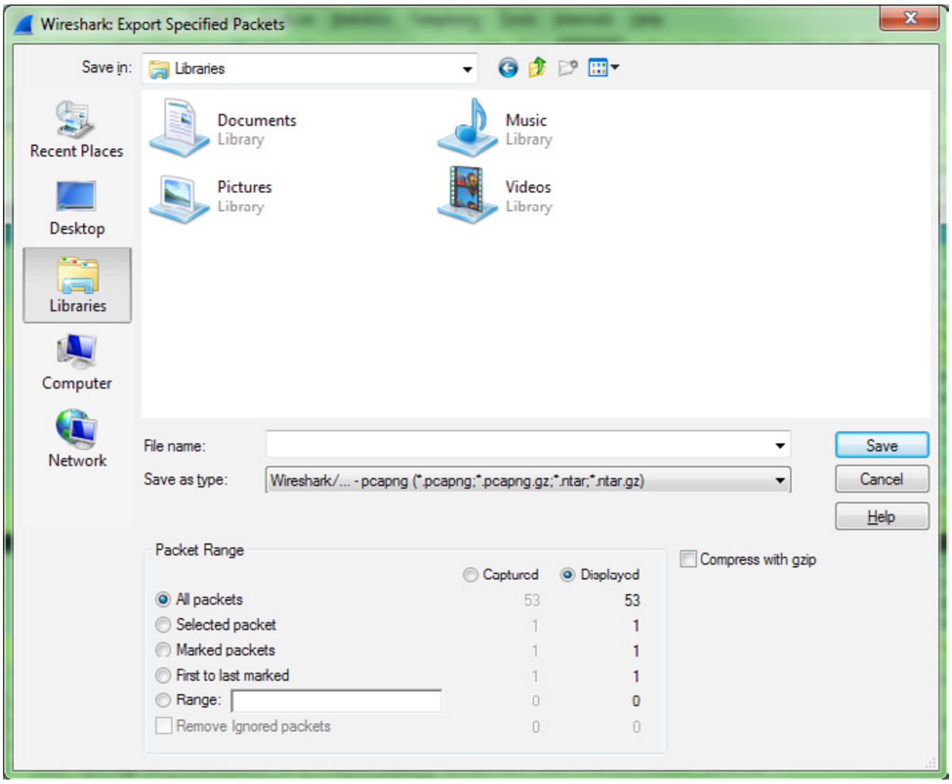
```
⊞ Frame 34: 357 bytes on wire (2856 bits), 357 bytes captured (2856 bits) on interface 0
⊞ Ethernet II, Src: LiteonTe_ec:93:16 (70:f1:a1:ec:93:16), Dst: D-LinkIn_3f:4a:c1 (c4:a8:1d:3f:4a:c1)
⊞ Internet Protocol Version 4, Src: 192.168.1.38 (192.168.1.38), Dst: 1.175.80.74 (1.175.80.74)
⊞ User Datagram Protocol, Src Port: 47064 (47064), Dst Port: 7089 (7089)
⊞ Data (315 bytes)
```

Figure 3. *Find Packets*

Once the interested packets are found based on certain search criteria, users can mark those packets for future reference. Marked packets will be displayed in white text with black background and there is also an option to save only the marked packets. There are so many benefits when it comes to marking packets of particular interest. The first and foremost benefit of marking packets is to find them easily at a later point of time. Otherwise which there will be millions of packets from the packet trace and finding the one that is of interest will take lot of time and efforts. Since there is option to save only the marked packets, the packets with no interest can be discarded. Marking a packet is as simple as making a text bold in word document. When you find the packet that needs to be marked, press CTRL+M. Pressing this when a packet is marked will unmark the packet. Marking or unmarking can be performed by the menu option present in the interface also. The options will be displayed in Edit menu. When you feel that all the packets that are fetched currently needs to be marked, simply press CTRL+A and to unmark all of them, press CTRL+D. There is also an option to jump to the next marked packet in the trace and it is pressing SHIFT+CTRL+N. Navigating to the previous marked packet is SHIFT+CTRL+B. Users are given option to save only the marked packets or the packets that are between the first marked packet and the last marked packets. As you see, all the features provided by Wireshark are easy to use, which is the major reason for its success among the network administrators. As we have seen how to export marked packets, let us now see the various options that are available for exporting the PCAP files.

EXPORTING PCAP FILES

Since there will be thousands and sometimes millions of packets received as part of packet sniffing technique, it is highly impossible for analyzing all the files at a glance. Hence, the team at Wireshark came up with the option of saving the results in the form of PCAP files that can be opened anytime from Wireshark. All the files and packets can be opened and analyzed whenever the user finds time to do so. The entire files that are captured can be saved to the local computer in as many as 21 formats with PCAP as the default file format. But saving all files, including those that are not useful will result in waste of time and efforts. And this is where finding and marking packets come into picture. As seen in the previous section, the packets can be marked using various methods. Wireshark provides users with the option of saving or exporting only those marked packets so that the user sees only the required packets as and when needed. To avail this option, user has to click on Export specified packets option from main menu. The Export specified packets dialog window displays various options on what packets should be exported. The first option present is exporting all packets, which when selected will export all the packets present in the current capture window. The second option is to export selected packets. This means that, user can select either one packet or more than one packet in the packet list pane. Such selected packets can only be exported when Selected Packets option is checked. The third option is to export marked packets. This is the option which will enable users to export only the marked packets. The fourth option is First to Last marked which will enable users to export all the packets that are present between the first marked packets in the packet list pane to the last marked packet in the pane. The last option is to specify a range in the text box which will export all the packets that are present within the particular range as mentioned in the text box. This option provides more flexibility to users in selecting packets from anywhere within the pane. For example, giving the range as 1, 10-20, 25- will export the first packet followed by the packets from number 10 to 20 inclusive, and packet number 25 till the end of all the packets that are part of current capture.



Wireshark: Export Specified Packets

Save in: Libraries

File name:

Save as type: Wireshark/... - pcapng (*.pcapng;*.pcapng.gz;*.ntar;*.ntar.gz)

□ Compress with gzip

Packet Range

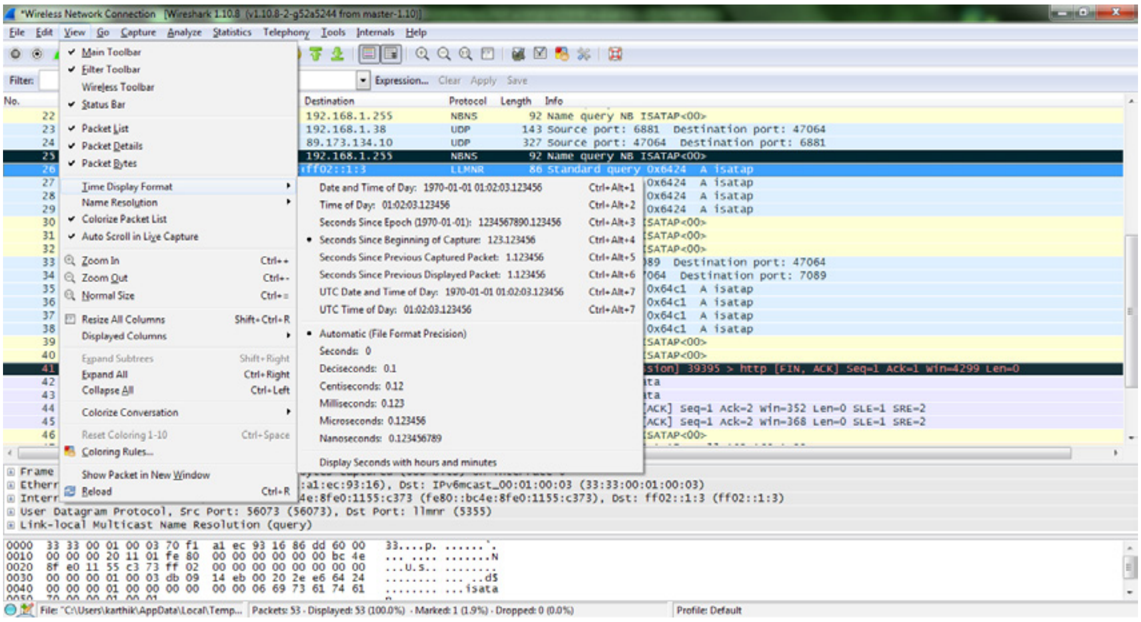
	○ Captured	● Displayed
● All packets	53	53
○ Selected packet	1	1
○ Marked packets	1	1
○ First to last marked	1	1
○ Range:	0	0
□ Remove Ignored packets	0	0

Figure 4. *Exporting packets*

Apart from these options, there is also option to remove the ignored packets from being exported. Normally, the packets that the users consider of no use can be ignored from the packet list pane. To ignore any packet, the user simply needs to select a packet and right click on the mouse. The second option in the list is to ignore packets. Hence when Remove ignored packets check box is selected, the packets that are ignored from the packet list pane will not be exported. The packet range section of Export specified packets dialog window also displays the count of packets that are in specified range as mentioned above. This will allow user to track how many packets are marked or selected and how many of them are being exported currently. When exporting PCAP files, users can also determine if they want the exported file to be in compressed format. The Compress with gzip checkbox lets users to either compress the end result file or to leave the file in native format itself. Apart from exporting the packets that are captured, the packet bytes can also be exported as well. To export the packet bytes, the user needs to select the packet detail and click on File, followed by clicking on Export selected packet bytes. The packet bytes can be exported to raw data format, which can later be retrieved. Wireshark also provides various other ways and formats by which the packets can be saved. The users are given the option to export the packet dissections as plain text file, comma separated value file format, post script file, C arrays, PSML file and PDML file. Exporting as plain text file will turn packets to plain ASCII text file which can be opened with the help of notepad software. Exporting as csv format exports files to CSV, which can then be opened using Excel. The files can also be exported to C arrays which can later be included directly in C programs. PSML and PDML are XML based formats that includes only the packet summary and packet details respectively. With so many options to choose from, the user can easily export the captured packets to the file format of their choice. The saved files can be opened anytime using the Wireshark tool itself for further analysis.

TEMPORAL CONSISTENCY

As seen above, Wireshark has made it very easier for users to capture packets and mark them for future reference. Apart from this basic feature, the tool gives many additional options that can be customized as part of user preference. One important concept that users need to understand well about the tool is the time format that is used in logs and captured packets. Since there are various time formats available and the captured packets display time stamps, it will be confusing if the user is not able to understand from where the time is fetched from. Time will be stamped on all the packets that are captured. The time stamps will also be saved along with the capture file for users to analyze the packets based on time frame. The time displayed in the captured packets is taken from the WinPCap library, which in turn takes the time from the kernel of the operating system. This is true only for live capture of packets. When the data is loaded from a capture file, the time is fetched from the file and not from the operating system. The standard time format as followed by Wireshark consists of date as yyyy—mm—dd followed by time in the form of hh—mm—ss. There are various formats by which Wireshark displays time for the captured packets. The various time presentation formats are available in View menu. The first format is the date and time of day as mentioned above, which is the absolute date and time at which the packet was captured. The next format is displaying only the time of the day at which the packet was captured. Seconds since beginning of capture is the relative time between the first captured packets till the next packet. Seconds since previous captured packet displays the relative time since the previous captured packet while Seconds since previous displayed packet displays the relative time since the previous displayed packet. The Seconds since Epoch displays the relative time since the midnight of January 1, 1970 UTC. The number of decimal places displayed in the time stamp can also be modified. The default setting is to display time in micro seconds while this can be modified to seconds, deciseconds, centiseconds, milliseconds or nanoseconds.


*Wireless Network Connection [Wireshark 1.10.8 (v1.10.8-2-g52a5244 from master-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save
✓ Main Toolbar
✓ Filter Toolbar
Wireless Toolbar
✓ Status Bar
✓ Packet List
✓ Packet Details
✓ Packet Bytes
Time Display Format
Name Resolution
✓ Colorize Packet List
✓ Auto Scroll in Live Capture
Zoom In Ctrl++
Zoom Out Ctrl+-
Normal Size Ctrl+=
Resize All Columns Shift+Ctrl+R
Displayed Columns
Expand Subtrees Shift+Right
Expand All Ctrl+Right
Collapse All Ctrl+Left
Colorize Conversation
Reset Coloring 1-10 Ctrl+Space
Coloring Rules...
Show Packet in New Window
Reload Ctrl+R
Date and Time of Day: 1970-01-01 01:02:03.123456 Ctrl+Alt+1
Time of Day: 01:02:03.123456 Ctrl+Alt+2
Seconds Since Epoch (1970-01-01): 1234567890.123456 Ctrl+Alt+3
● Seconds Since Beginning of Capture: 123.123456 Ctrl+Alt+4
Seconds Since Previous Captured Packet: 1.123456 Ctrl+Alt+5
Seconds Since Previous Displayed Packet: 1.123456 Ctrl+Alt+6
UTC Date and Time of Day: 1970-01-01 01:02:03.123456 Ctrl+Alt+7
UTC Time of Day: 01:02:03.123456 Ctrl+Alt+7
● Automatic (File Format Precision)
Seconds: 0
Deciseconds: 0.1
Centiseconds: 0.12
Milliseconds: 0.123
Microseconds: 0.123456
Nanoseconds: 0.123456789
Display Seconds with hours and minutes
Profile: Default

```
No.   Destination        Protocol  Length  Info
22    192.168.1.255      NBNS          92  Name query NB ISATAP<00>
23    192.168.1.38       UDP          143  Source port: 6881  Destination port: 47064
24    89.173.134.10      UDP          327  Source port: 47064  Destination port: 6881
25    192.168.1.255      NBNS          92  Name query NB ISATAP<00>
26    ff02::1:3          LLMNR         86  Standard query 0x6424  A isatap
27                                         0x6424  A isatap
28                                         0x6424  A isatap
29                                         0x6424  A isatap
30                                         ISATAP<00>
31                                         ISATAP<00>
32                                         ISATAP<00>
33                                         89  Destination port: 47064
34                                         064  Destination port: 7089
35                                         0x64c1  A isatap
36                                         0x64c1  A isatap
37                                         0x64c1  A isatap
38                                         0x64c1  A isatap
39                                         ISATAP<00>
40                                         ISATAP<00>
41                                         ssion] 39395 > http [FIN, ACK] Seq=1 Ack=1 Win=4299 Len=0
42                                         ta
43                                         ta
44                                         [ACK] Seq=1 Ack=2 Win=352 Len=0 SLE=1 SRE=2
45                                         [ACK] Seq=1 Ack=2 Win=368 Len=0 SLE=1 SRE=2
46                                         ISATAP<00>

⊞ Frame
⊞ Etherr :a1:ec:93:16), Dst: IPv6mcast_00:01:00:03 (33:33:00:01:00:03)
⊞ Interr 4e:8fe0:1155:c373 (fe80::bc4e:8fe0:1155:c373), Dst: ff02::1:3 (ff02::1:3)
⊞ User Datagram Protocol, Src Port: 56073 (56073), Dst Port: llmnr (5355)
⊞ Link-local Multicast Name Resolution (query)

0000  33 33 00 01 00 03 70 f1  a1 ec 93 16 86 dd 60 00   33....p. .......`.
0010  00 00 00 20 11 01 fe 80  00 00 00 00 00 00 bc 4e   ... .... .......N
0020  8f e0 11 55 c3 73 ff 02  00 00 00 00 00 00 00 00   ...U.s.. ........
0030  00 00 00 01 00 03 db 09  14 eb 00 20 2e e6 64 24   ........ ... ..d$
0040  00 00 00 01 00 00 00 00  00 00 06 69 73 61 74 61   ........ ...isata

File: "C:\Users\karthik\AppData\Local\Temp...  Packets: 53 · Displayed: 53 (100.0%) · Marked: 1 (1.9%) · Dropped: 0 (0.0%)
```

Figure 5. *Time formats*

Wireshark has also given the option of setting time reference to packets. When a time reference is set to a packet, the corresponding packet will be considered as the starting point for all subsequent packets. The time difference between the referenced packet and the subsequent packets can be easily calculated for analysis. Any number of time references can be configured in a capture file. This packet time referencing can be useful only when the time presentation format is set to Seconds since beginning of capture. When any other time presentation format is selected and packets are marked for reference, Wireshark offers the users to switch the time presentation format to Seconds since beginning of capture. A packet that is time referenced will be marked as *REF* in the time column and the subsequent packets will display the time interval to the referenced packet. To set time reference to packets, right click on the packet and click Set Time Reference (Toggle). Clicking it again will reset the time reference setting. Since Wireshark does not provide any standard time stamp, it is often difficult to identify the accuracy of the time mentioned in the packets. As the time is taken from the kernel of the operating system, the accuracy of the timestamp varies based on the type of tool being used. If the tool is executed from USB network adapter, the accuracy will be really bad as the packets need to travel from USB to the kernel for capturing the time.

FILTERING

With millions of packets getting captured every time a sniffing program is used, it is difficult for users to identify the packet that is of interest. To enable easy identification of packets based on requirements, Wireshark has come up with the option of filtering packets to display what is needed for current analysis. An expression can be used to filter the packets from any number of packets that are captured. It is very easy to create expression for filtering packets. There are two types of filters in Wireshark namely Capture filters and Display filters. The Capture filter filters the packets during the packet capturing process by the help of WinPCap. The Capture packets dialog window includes option for selecting which packets need to captured and which packets should not be captured. Since WinPCap is used for filtering packets during packet capture itself, is will be easier if users are aware of the syntax of WinPCap. Consider, there is a problem in the service that is running on a particular port. When packet capturing is performed it would be difficult for all packets to be analyzed and the particular service be diagnosed. This is where filtering of packets come into picture. Users can filter only the packets from the particular port which makes it little easier to analyze the issues.

It is very simple to filter packets while capturing them. In the Capture Packets dialog window, select Options to open Capture Options dialog window. The expression to filter packets can be given in the Capture Filter text box. The users can also click on the Capture Filter button to open the Capture filter expression builder that helps users to use some default expressions. When the filter expression is given, click on Start button to capture packets based on the filter criteria. On the other hand, the display filter is used to display only the filtered packets after the packets are captured. The Filter option is present above the Packet list pane and the expression can be entered in the filter text box. Hence, you will be capturing all the data but only view the data of interest. This enables the user to view other packets that are not filtered but may come in handy. The most common type of packets that are omitted from view using display filter are the broadcast packets. It is very simple to filter packets in this method as all you need to do is to enter the filter expression in the filter text box located above the packet list pane and click on Apply button. When you do not want the filter to be applied and instead want to view all the packets, simply clear the filter.

Some example syntax for filter expressions:

To capture traffic only from a specific IP address – host 188.12.10
To capture traffic from DNS – port 53
To capture all traffic other than DNS – port not 53

http://wiki.wireshark.org/CaptureFilters gives more examples on syntax for capturing packets.

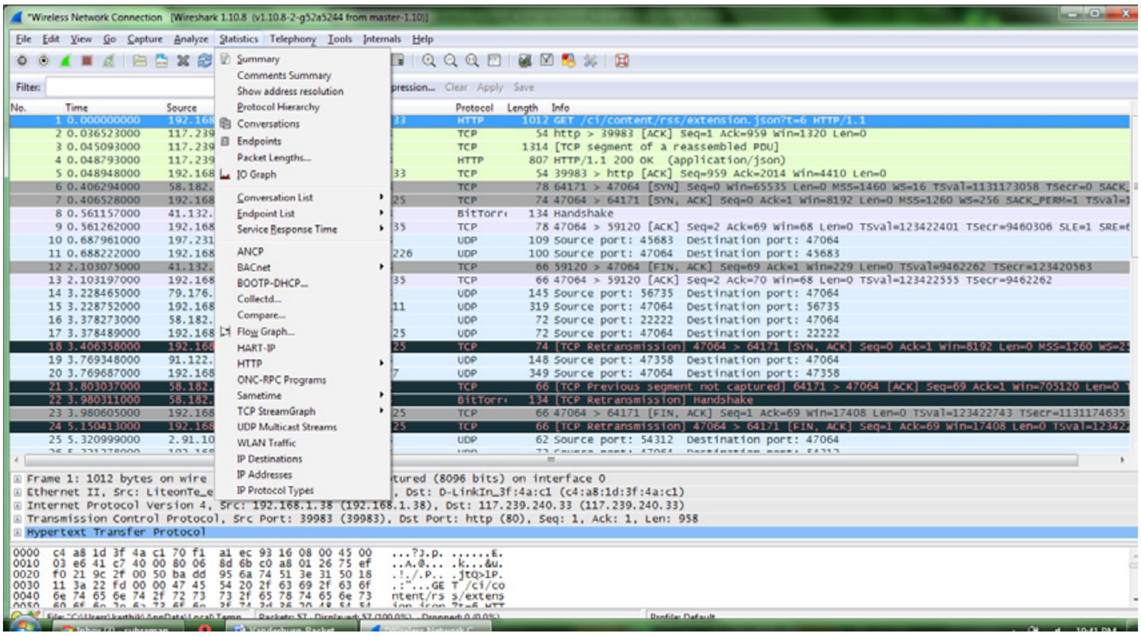
The filter expression dialog is the easy way to filter packets based on some predefined expressions that will help the novice users from using filter option. The filter expression dialog window can be opened by clicking Expression button displayed near the Filter text box above the packet list pane. The filter expression dialog window has three panes with the left pane displaying the default expressions, followed by Relation pane and value pane to the right. An expand icon is displayed adjacent to every expression displayed in the dialog window. On clicking the expand icon, the various criteria for the specific protocol is displayed. Select the criteria and select the relation which should be used for filtering the packets from the relation pane. The value field in the right pane displays a set of predefined values but also comes with a text box that can be used for entering user defined values. On clicking OK, the complete filter expression will be displayed in the Filter text box. The filter expression dialog window provides novice users with the option of selecting filter criteria in the simplest manner. But advanced users would require more expressions that are not present in the default list. Such users can create the filter expression by themselves once they know the syntax. Since, most often, the filter criteria used is the type of packet, it is important to know the name of all the protocols that you need to analyze. For example, if you are troubleshooting a TCP problem in the network, you need to type tcp in the filter text box as the filter criteria. This will fetch only TCP packets and omit packets of other protocols. Similarly, the filter criteria can also be such that one type of packets should not be displayed but all other types should be displayed. For such requirements, there is the ! operator which omits the particular value but displays all other values. Apart from the normal expressions, various comparison and logical operators are available for enhancing the filter expression. The various comparison operators that are available are == (Equal to), != (Not equal to), > (greater than), < (Less than), >= (Greater than or equal to) and <= (Less than or equal to). The comparison operators assist the users in displaying results that are compared to the value displayed in the filter expression. The logical operators assist in combining multiple expressions into a single statement after which filter is applied. The various logical operators used are ‘and’ which will include both conditions in the statement, ‘or’ which will use either of the two conditions, ‘xor’ which searches for one and only one condition to be true and ‘not’ which does not search for any condition given in the expression. The filters can also be saved for future uses. To save a filter, user should navigate to Filter capture dialog window that can be opened from Capture -> Capture Filters. Create a new filter and give a suitable name for the filter. Enter the filter expression and click on Ok button. This will save the filter expression which can be retrieved anytime for filtering packets.

Wireshark comes with all the basic features similar to other packet sniffing programs. But there are many other advanced features that make it the best of the lot. The most important part of any packet sniffing program is to analyze the packets once they are captured. Every tool has their own analysis technique and Wireshark's is one of the best in the business. There are various types of analysis possible with Wireshark; to understand all of them in detail and to explore all the advanced features, attend this workshop.

MODULE 3 – STATISTICAL ANALYSIS WITH WIRESHARK

SUMMARY STATISTICS

The summary statistics window displays the general information about the current capture file. The summary statistics can be obtained by clicking on the Statistics menu item and then clicking Summary option. Once the capturing of files is completed, or is in progress, the summary option is enabled. The various information that is displayed in the summary window is File name, length of bytes, time when the first and last packet are received, capture and display information. The file section in the summary window displays the file name, length of file in bytes, file format and encapsulation format.The time section displays the time when the first packet is received, time when the last packet is received and the time elapsed between the first and last packet. The Capture section displays information about the operating system and the build version of the tool and the capture application format. Display information displays the filter that is used, if any, and the percentage of ignored packets. The table displayed below the display section displays the total number of packets captured, total size of packets captured, average packets captured per second, average packet size, average bytes received per second and average megabytes per second. The table also displays the total packets that are captured, displayed and marked.All the columns in the table can be sorted and displayed. All this information will be useful in analyzing the number of packets captured, size of the packets and the time elapsed during the process.

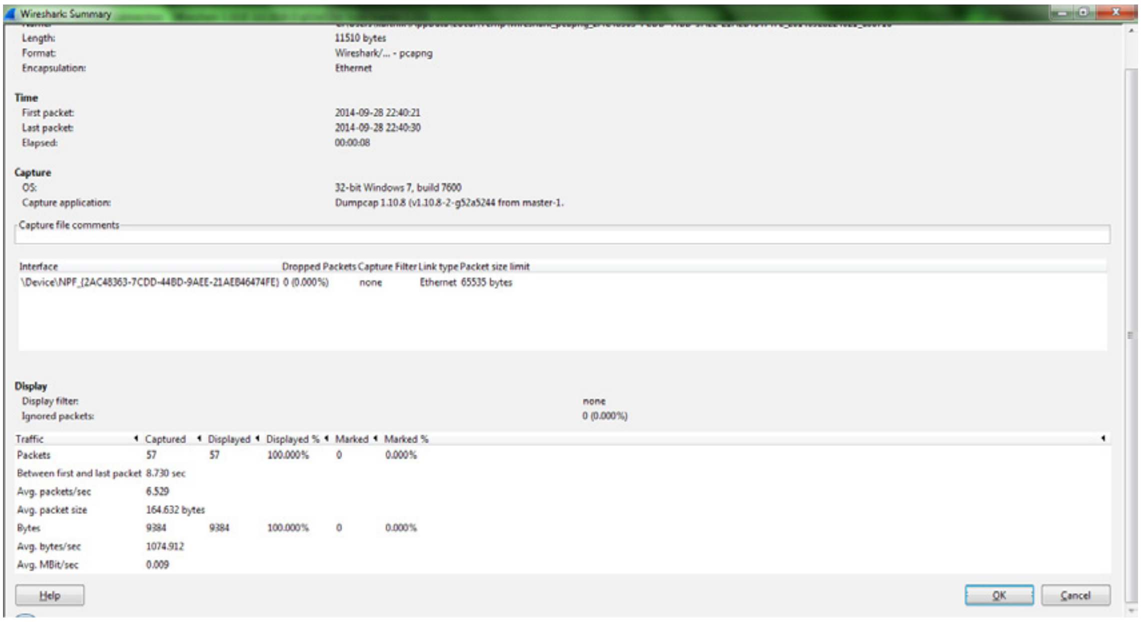


```
No.   Time          Source          Protocol  Length  Info
  1 0.000000000   192.168   33      HTTP      1012 GET /ci/content/rss/extension.json?t=6 HTTP/1.1
  2 0.036523000   117.239           TCP         54 http > 39983 [ACK] Seq=1 Ack=959 Win=1320 Len=0
  3 0.045093000   117.239           TCP       1314 [TCP segment of a reassembled PDU]
  4 0.048793000   117.239           HTTP       807 HTTP/1.1 200 OK  (application/json)
  5 0.048948000   192.168   33      TCP         54 39983 > http [ACK] Seq=959 Ack=2014 Win=4410 Len=0
  6 0.406294000   58.182.           TCP         78 64171 > 47064 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=16 TSval=1131173058 TSecr=0 SACK_
  7 0.406528000   192.168   25      TCP         74 47064 > 64171 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1260 WS=256 SACK_PERM=1 TSval=1
  8 0.561157000   41.132.           BitTorr     134 Handshake
  9 0.561262000   192.168   35      TCP         78 47064 > 59120 [ACK] Seq=2 Ack=69 Win=68 Len=0 TSval=123422401 TSecr=9460306 SLE=1 SRE=
 10 0.687961000   197.231           UDP        109 Source port: 45683  Destination port: 47064
 11 0.688222000   192.168   226     UDP        100 Source port: 47064  Destination port: 45683
 12 2.103075000   41.132.           TCP         66 59120 > 47064 [FIN, ACK] Seq=69 Ack=1 Win=229 Len=0 TSval=9462262 TSecr=123420563
 13 2.103197000   192.168   35      TCP         66 47064 > 59120 [ACK] Seq=2 Ack=70 Win=68 Len=0 TSval=123422555 TSecr=9462262
 14 3.228465000   79.176.           UDP        145 Source port: 56735  Destination port: 47064
 15 3.228752000   192.168   11      UDP        319 Source port: 47064  Destination port: 56735
 16 3.378273000   58.182.           UDP         72 Source port: 22222  Destination port: 47064
 17 3.378489000   192.168   25      UDP         72 Source port: 47064  Destination port: 22222
 18 3.406358000   192.168   25      TCP         74 [TCP Retransmission] 47064 > 64171 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1260 WS=2
 19 3.769348000   91.122.           UDP        148 Source port: 47358  Destination port: 47064
 20 3.769687000   192.168   7       UDP        349 Source port: 47064  Destination port: 47358
 21 3.803037000   58.182.           TCP         66 [TCP Previous segment not captured] 64171 > 47064 [ACK] Seq=69 Ack=1 Win=705120 Len=0 T
 22 3.980311000   58.182.           BitTorr     134 [TCP Retransmission] Handshake
 23 3.980605000   192.168   25      TCP         66 47064 > 64171 [FIN, ACK] Seq=1 Ack=69 Win=17408 Len=0 TSval=123422743 TSecr=1131174635
 24 5.150413000   192.168   25      TCP         66 [TCP Retransmission] 47064 > 64171 [FIN, ACK] Seq=1 Ack=69 Win=17408 Len=0 TSval=123422
 25 5.320999000   2.91.10           UDP         62 Source port: 54312  Destination port: 47064
```

```
⊞ Frame 1: 1012 bytes on wire                tured (8096 bits) on interface 0
⊞ Ethernet II, Src: LiteonTe_e               , Dst: D-LinkIn_3f:4a:c1 (c4:a8:1d:3f:4a:c1)
⊞ Internet Protocol Version 4, Src: 192.168.1.38 (192.168.1.38), Dst: 117.239.240.33 (117.239.240.33)
⊞ Transmission Control Protocol, Src Port: 39983 (39983), Dst Port: http (80), Seq: 1, Ack: 1, Len: 958
⊞ Hypertext Transfer Protocol

0000  c4 a8 1d 3f 4a c1 70 f1  a1 ec 93 16 08 00 45 00   ...?J.p. ......E.
0010  03 e6 41 c7 40 00 80 06  8d 6b c0 a8 01 26 75 ef   ..A.@... .k...&u.
0020  f0 21 9c 2f 00 50 ba dd  95 6a 74 51 3e 31 50 18   .!./.P.. .jtQ>1P.
0030  11 3a 22 fd 00 00 47 45  54 20 2f 63 69 2f 63 6f   .:"...GE T /ci/co
0040  6e 74 65 6e 74 2f 72 73  73 2f 65 78 74 65 6e 73   ntent/rs s/extens
```

Figure 1. *Statistics Menu*



Length: 11510 bytes
Format: Wireshark/... - pcapng
Encapsulation: Ethernet

Time
First packet: 2014-09-28 22:40:21
Last packet: 2014-09-28 22:40:30
Elapsed: 00:00:08

Capture
OS: 32-bit Windows 7, build 7600
Capture application: Dumpcap 1.10.8 (v1.10.8-2-g52a5244 from master-1.

Capture file comments

Interface	Dropped Packets	Capture Filter	Link type	Packet size limit
\Device\NPF_{2AC48363-7CDD-44BD-9AEE-21AEB46474FE}	0 (0.000%)	none	Ethernet	65535 bytes

Display
Display filter: none
Ignored packets: 0 (0.000%)

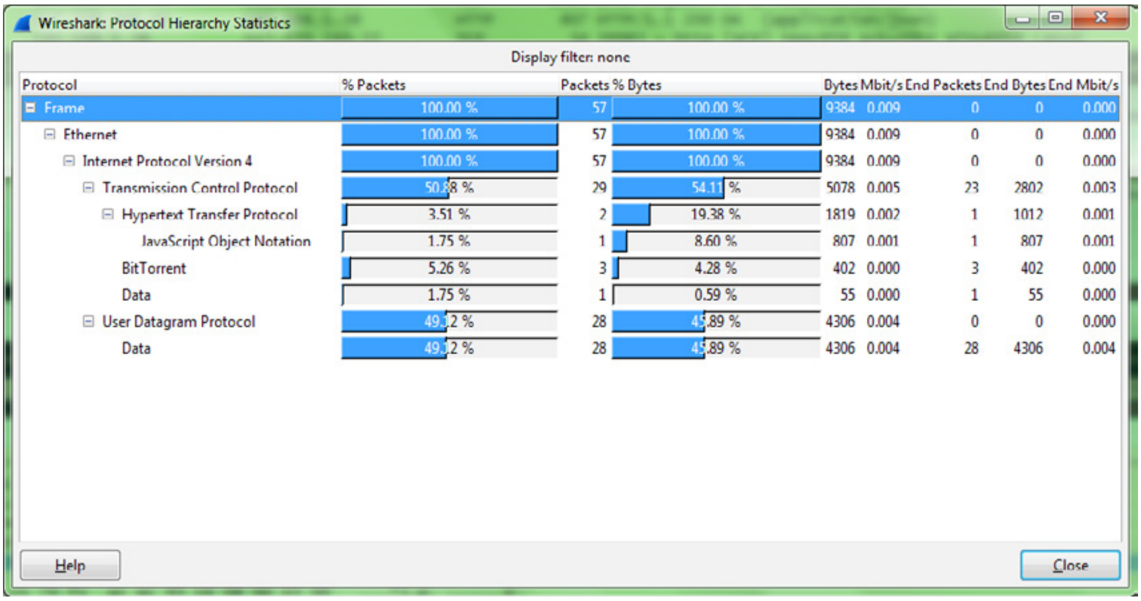
Traffic	Captured	Displayed	Displayed %	Marked	Marked %
Packets	57	57	100.000%	0	0.000%
Between first and last packet	8.730 sec				
Avg. packets/sec	6.529				
Avg. packet size	164.632 bytes				
Bytes	9384	9384	100.000%	0	0.000%
Avg. bytes/sec	1074.912				
Avg. MBit/sec	0.009				

Figure 2. *Summary Statistics*

PROTOCOL HIERARCHY STATISTICS

Usually, there will be more than thousands of packets captured for analysis and it would be better if users are provided with statistics on what type of protocols are received in the file. The protocol hierarchy statistics provides users with information related to what percentage of the captured packets belong to various protocols such as TCP and UDP. This simplifies the process of counting each and every packet to determine how many TCP protocol packets are captured and so on. This will also help users to understand easily if there are any problems in the network as the percentage of all protocols will be displayed. The protocol hierarchy statistics window can be opened by clicking on the Protocol hierarchy option present in the Statistics menu. The percentage value and the count of packets will always be accurate; thereby helping a great deal in analysis.

The various columns that are present in the protocol hierarchy statistics window are protocol, % packets, packets, bytes, megabytes per second, end packets, end bytes, and end megabytes per second. The protocol column displays the name of the protocol; % packets display the percentage of packets that are captured, packets column displays the absolute number of packets that are captured. The bytes column displays the total amount of bytes that are captured. The MBit/s column displays the bandwidth of the captured packet's protocol during the time of capture. There are various methods by which Wireshark recognizes packets. The most common method is to use identifiers, as every protocol has their own identifiers. It also uses the previous traffic information for decoding information. When the protocol does not have any identifier, Wireshark uses the heuristics to recognize the protocols. In some cases, the percentage of packets does not sum up to 100%. This is because; there is no option to display packets that are other than the recognized protocols of Wireshark.It is also important to note that any single packet would contain more than one protocol and this would increase the count of protocol to more than 100%.



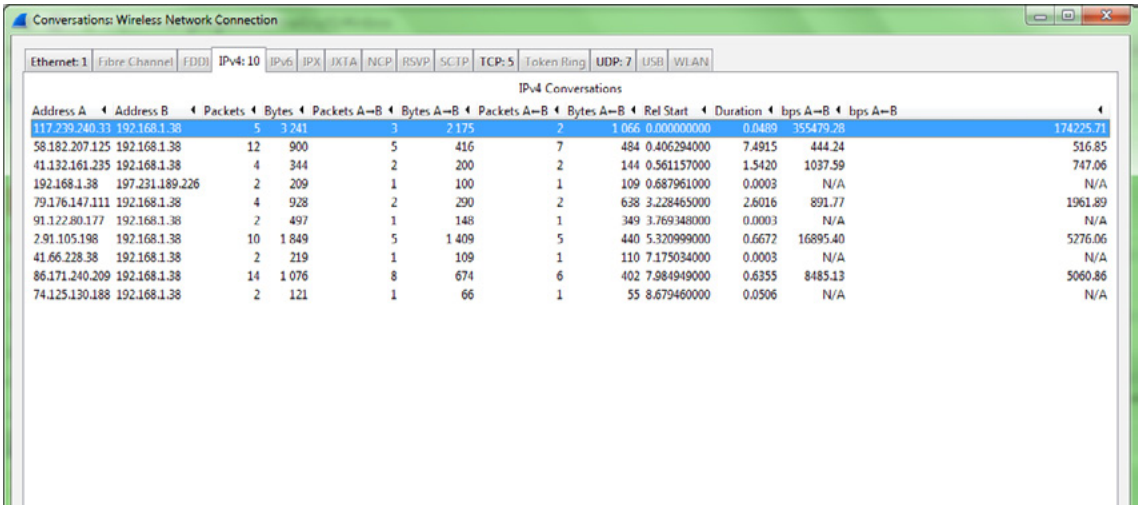
Display filter: none

Protocol	% Packets	Packets	% Bytes	Bytes	Mbit/s	End Packets	End Bytes	End Mbit/s
⊟ Frame	100.00 %	57	100.00 %	9384	0.009	0	0	0.000
⊟ Ethernet	100.00 %	57	100.00 %	9384	0.009	0	0	0.000
⊟ Internet Protocol Version 4	100.00 %	57	100.00 %	9384	0.009	0	0	0.000
⊟ Transmission Control Protocol	50.88 %	29	54.11 %	5078	0.005	23	2802	0.003
⊟ Hypertext Transfer Protocol	3.51 %	2	19.38 %	1819	0.002	1	1012	0.001
JavaScript Object Notation	1.75 %	1	8.60 %	807	0.001	1	807	0.001
BitTorrent	5.26 %	3	4.28 %	402	0.000	3	402	0.000
Data	1.75 %	1	0.59 %	55	0.000	1	55	0.000
⊟ User Datagram Protocol	49.12 %	28	45.89 %	4306	0.004	0	0	0.000
Data	49.12 %	28	45.89 %	4306	0.004	28	4306	0.004

Figure 3. *Protocol Hierarchy Statistics*

CONVERSATION STATISTICS

You are right! Conversation in a network is synonymous to the conversation in normal terms. A conversation in a network is the communication that takes place between two end points of the system. When there are two hosts, the transfer of packets such as SYN, ACK etc. is called conversation. Wireshark has made it easier to analyze the conversation of various packets in the network by the help of the Conversations window that is available in the statistics menu. The Conversations window displays the various protocols at the top of the window. User needs to select the protocol tab which will display all the conversations in the particular protocol. The first two columns are address A and address B which are the source and destination addresses. The next column displays the total number of packets that are transmitted between the hosts followed by the total size of packets in bytes. The next four columns display the total number of packets and bytes that are transferred from address A to B and vice versa. The duration column displays the time of travel from one host to another. All the columns that are displayed in the window can be sorted in both ascending and descending order. The copy button in the dialog window copies the entire content displayed in the column to the clipboard which can later be pasted in the local system. There are more advanced options such as filtering the packets, finding the packets, and preparing for filter etc. in the conversation window. Right click on any conversation to prepare for a filter or actually apply a filter. There is also an option to find packets either from address A to B or from B to A. Users can also color any conversation for easy analysis. Users also have additional options to display only the conversation which matches the current display filter in use. The conversations window will be updated now and then, thereby helping in fetching the latest results when the user opens the window at any time. Before the above said conversation window was introduced, protocol specific conversation windows were only in use. The Wireshark team has decided to keep the old option, as well as with the newer one. The older window can be obtained by clicking the Conversation List option from the Statistics menu. All the other options are present in the old protocol specific window as well.



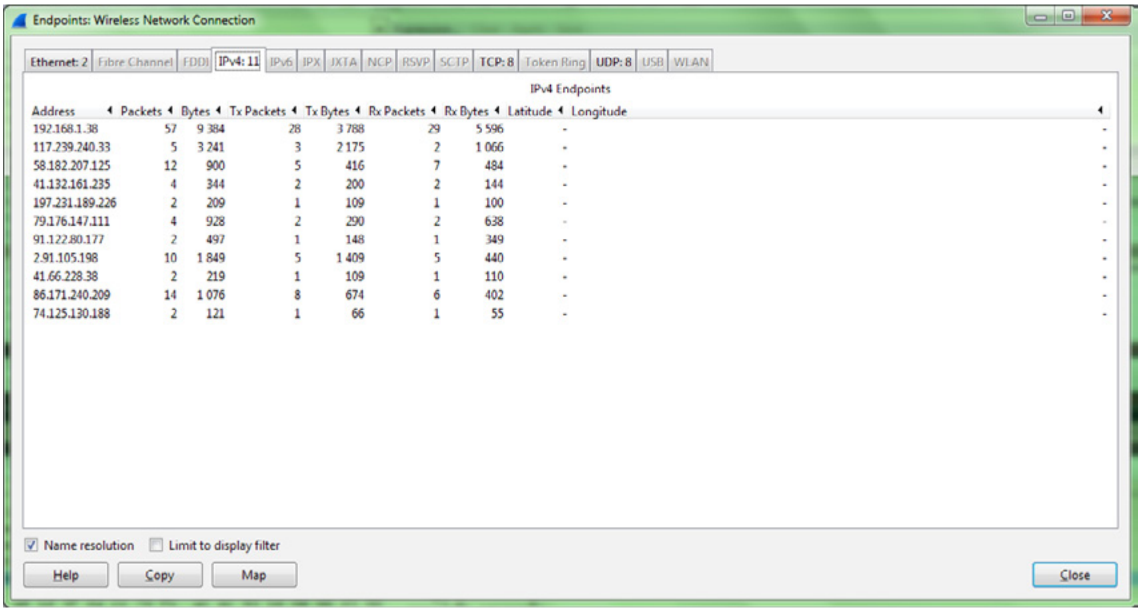
IPv4 Conversations

Address A	Address B	Packets	Bytes	Packets A→B	Bytes A→B	Packets A←B	Bytes A←B	Rel Start	Duration	bps A→B	bps A←B
117.239.240.33	192.168.1.38	5	3 241	3	2 175	2	1 066	0.000000000	0.0489	355479.28	174225.71
58.182.207.125	192.168.1.38	12	900	5	416	7	484	0.406294000	7.4915	444.24	516.85
41.132.161.235	192.168.1.38	4	344	2	200	2	144	0.561157000	1.5420	1037.59	747.06
192.168.1.38	197.231.189.226	2	209	1	100	1	109	0.687961000	0.0003	N/A	N/A
79.176.147.111	192.168.1.38	4	928	2	290	2	638	3.228465000	2.6016	891.77	1961.89
91.122.80.177	192.168.1.38	2	497	1	148	1	349	3.769348000	0.0003	N/A	N/A
2.91.105.198	192.168.1.38	10	1 849	5	1 409	5	440	5.320999000	0.6672	16895.40	5276.06
41.66.228.38	192.168.1.38	2	219	1	109	1	110	7.175034000	0.0003	N/A	N/A
86.171.240.209	192.168.1.38	14	1 076	8	674	6	402	7.984949000	0.6355	8485.13	5060.86
74.125.130.188	192.168.1.38	2	121	1	66	1	55	8.679460000	0.0506	N/A	N/A

Figure 4. *Conversation Statistics*

ENDPOINT STATISTICS

Before going into what details end point statistics provides to its users, it is important to understand what an endpoint is and what its types are. An endpoint, as the name suggests is the logical end of any protocol traffic. The end point varies for every protocol layer and the endpoint statistics window displays the statistics of all the endpoints captured in the file. The various endpoints that are taken into account are the Ethernet endpoint which is the MAC address of the Ethernet, fibre channel endpoint, FDDI endpoint, IPv4, IPX, JXTA, NXT, RSVP, TCP, and token ring endpoint. The endpoint window is similar to the Conversations window in the way the various protocols are displayed in separate tabs. Click on a tab to view the endpoints and the number of packets and bytes transferred. The tab label of every protocol shows the total number of packets captured. This feature helps users to see only the tabs which contain packets. When any protocol does not have any packets captured, the tab label will be greyed out. Even though the label is greyed out, the user can navigate to the page, only to see empty values. The every row present in the tab lists only one endpoint. There is a name resolution and limit to display option as well. The packets can also be filtered and searched from the endpoint statistics window. Coloring can also be applied on the packets for easy analysis. The earlier version of Wireshark had only a protocol specific end point window and not the combined one as available now. But even though there are combined window now, users can view protocol specific windows as well.



IPv4 Endpoints

Address	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes	Latitude	Longitude
192.168.1.38	57	9 384	28	3 788	29	5 596	-	-
117.239.240.33	5	3 241	3	2 175	2	1 066	-	-
58.182.207.125	12	900	5	416	7	484	-	-
41.132.161.235	4	344	2	200	2	144	-	-
197.231.189.226	2	209	1	109	1	100	-	-
79.176.147.111	4	928	2	290	2	638	-	-
91.122.80.177	2	497	1	148	1	349	-	-
2.91.105.198	10	1 849	5	1 409	5	440	-	-
41.66.228.38	2	219	1	109	1	110	-	-
86.171.240.209	14	1 076	8	674	6	402	-	-
74.125.130.188	2	121	1	66	1	55	-	-

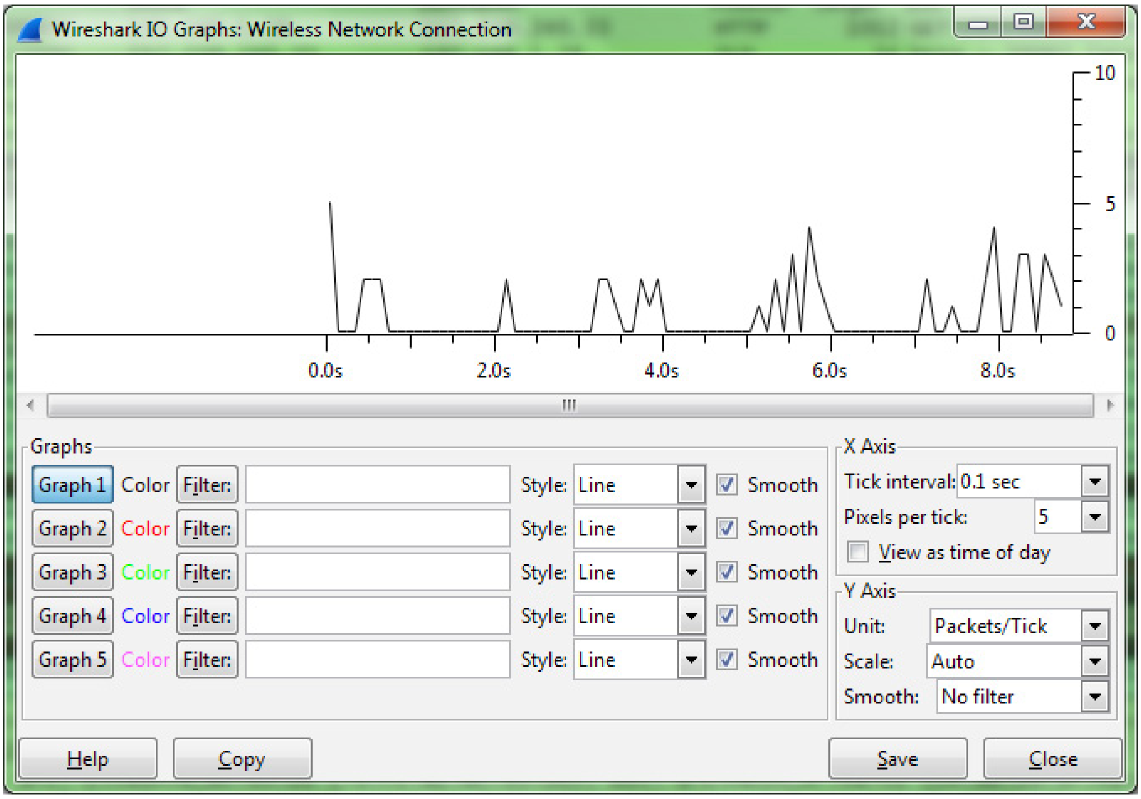
☑ Name resolution □ Limit to display filter

Figure 5. *Endpoint statistics*

I/O GRAPHS

The best way to analyze any statistics is to view them graphically. Graphical representation has always been the choice of technicians who desire to view results and study them. Wireshark has understood this requirement by rendering the I/O graphs window which provides graphical representation of data that is transferred along the network. The sudden spikes and other notifications in the graph would help analyze the flow of traffic across various protocols. Watching the graphical representation of packets in the I/O graphs window is interesting as highs and lows occurs now and then. A sudden high means that more data is downloaded at that point of time. There are 2 axes in the graph with the X axis representing tick interval and the Y axis representing packets transferred per tick interval. Both X and Y axis can be configured to user preference. The default tick interval of X axis is 1 second and the other options available are 0.001 sec, 0.01 sec, 0.1 sec, 10 sec, 1 min, and 10 minutes. The pixels per tick option also have values ranging from 1, 2, 5, and 10. View as Time of day option enables users to view the number of packets that are transferred at the particular time of the day. The time will be displayed along the X axis. Similar to X axis, the values of Y axis can also be altered. The default view of Y axis is by packets/Tick while it can be changed to Bytes/Tick and Bits/Tick. The color of the graph can also be selected by the user along with the graph style. The various styles available are Line, Bar, Impulse, and Dot. The graph can also be filtered to display packets of particular protocol alone. The various filtering options are available when the user clicks on the Filter button that is displayed for every graph in the window. Applying filters come in handy when there are problems in the network that must be diagnosed. Some of the commonly used filters are given below.

The tcp.analysis.lost_segment filter enables users to identity the packets that are lost during transmission. This can be found by the presence of gaps in the sequence numbers in the capture file. The duplicate tcp.analysis.duplicate_ack filter displays the packets that are acknowledged more than once. When the acknowledgement is duplicated, it means there is high latency between the endpoints. The tcp.analysis_retransmission filter displays all the retransmissions occurring in the network. When there are only a few retransmissions, it is not considered a problem. But when there are large numbers of retransmissions, it means the performance of the network is slow; therefore, poor. The tcp.analsysis.window_update notifies the user about the size of TCP window in the transfer. When the size of the window drops down to zero, it means the sender has stopped sending packets and is waiting for acknowledgement from receiver. The tcp.analysis.bytes_in_flight filter provides the number of bytes that are unacknowledged at any point of time. This number is recommended to be close to TCP window size, as any number lower than the window size means problems in the network due to packet loss. The graphical representation can be saved for future reference and analysis. The filter option and the coloring option go hand in hand. When you filter the graph to display ARP and TCP traffic, you give separate colors for both the graphs which will help in differentiating the packets received. The IO graph is an important part of packet analysis as it will be used for various analyses. Hence, it is important to understand the various features that are available in the window.



Graphs

Graph 1 Color Filter: Style: Line ☑ Smooth

Graph 2 Color Filter: Style: Line ☑ Smooth

Graph 3 Color Filter: Style: Line ☑ Smooth

Graph 4 Color Filter: Style: Line ☑ Smooth

Graph 5 Color Filter: Style: Line ☑ Smooth

X Axis

Tick interval: 0.1 sec

Pixels per tick: 5

□ View as time of day

Y Axis

Unit: Packets/Tick

Scale: Auto

Smooth: No filter

Figure 6. *IO Graphs*

RTP STATISTICS

RTP stands for Real Time Transport Protocol and it defines the standard format of packet transfer of audio and video files over IP networks. RTP is always used alongside with RTP Control Protocol which monitors the quality of data transmission over the network. RTP is most often used in real time multimedia sharing applications. It provides end to end transfer of live stream data. Wireshark statistics also includes a window to display the statistics of RTP packets that are transferred across the network. It also displays packets from reverse stream as well. The forward and reverse stream of packets are displayed in separate tabs, named as forward direction and reversed direction. The RTP statistics window displays both basic information such as number of packets and sequence number along with advanced statistics such as arrival time, packet size, and jitter etc. Along with information on individual packets, the lower pane of the window displays the minimum and maximum values. Users can save the file in the form of raw data or in any encoding format. The files can also be exported to Au-File, if there is G. 711 A-law or G.711 mu-law codec.

RESPONSE TIMES

The Service response time is defined as the time difference between the request of a packet and its corresponding response. This information can be obtained for many protocols. Wireshark does not contain a combined window displaying the response time for all the protocols; rather, it has individual windows for selected protocols. The various protocols that are supported can be seen by clicking service response time under the Statistics menu. Click on any of the protocol to display the response time window for that particular protocol. The various columns that are displayed in the window are Index, procedure, calls, minimum service response time, maximum service response time, and average time. Every window has a filter option as well, which will allow user to filter packets based on filter criteria.

All the above statistical tools ensure that packet analysis is performed with ease. All the tools are easy to use and come with many advanced features. Hands on experience on the tool will make you feel comfortable using the above said statistical windows.

With all the focus on Wireshark, one would ideally think if it's the only reliable packet sniffing program in the market. But that's not the case as there are many other programs performing so close to what Wireshark does. A couple of them are NetworkMiner and NetWitness Investigator. These tools provide simple to use options and features making it easier for network administrators to capture and analyze packets. Even though they perform similar function, both these tools also have customized strengths and weaknesses giving options for users to accept or reject the tool. To know more about these tools, read on and also attend the workshop to find out.

MODULE 4 – WORKING WITH PCAP ANALYSIS TOOLS

NETWORKMINER

Network security has always been an area of concern for network administrators. With millions of packets sent and received in the network, it is easier for malicious code to be injected into the system which would compromise on the security of the business application. Active scanning of packets in real time systems becomes an overload as it decreases the performance. Hence, passive scanning of packets by loading the captured files is becoming popular. Passive scanning is also called as network security monitoring and is one of the best ways to analyze the packets transmitted in the network when the network is not active. One of the best tools to analyze security of network is NetworkMiner which is an open source network analysis tool. The software tool can be downloaded free of cost from the internet. Even though the free version comes with loads of features, the professional edition will do many magic before the eyes with various packet analysis windows and options. Some of the features that are available with the professional version include Port independent protocol identification, exporting the results to csv/excel format, configurable file output directory, command line scripting support, host coloring support and Geo IP localization. The command line scripting is provided by NetworkMiner CLI which is the command line interface of the software. The command line version helps in integrating the tool to various scripts such as perl and python. When a pcap is loaded into the tool, the CLI generates CSV files that contain information about sessions, parameters, DNS records, Clear text words, credentials, File Info, messages and hosts. The professional version of the tool costs around $700. The best way to monitor a network is to place the packet capturing tool in the middle of the network or between two endpoints. It is also important to select the capturing tool such that it does not emit any signal to the network which is being monitored. The tool should capture packets and store them in the form of PCAP files that can be processed and analyzed later. Even though NetworkMiner performs live packet sniffing, it is advisable to perform the analysis offline. Storing the packets in the form of PCAP files also makes it an evidence of malicious activity in the network which would compromise the reliability and security of the system. The free version and the professional edition can be obtained from the website: *http://www.netresec.com/?page=NetworkMiner.*

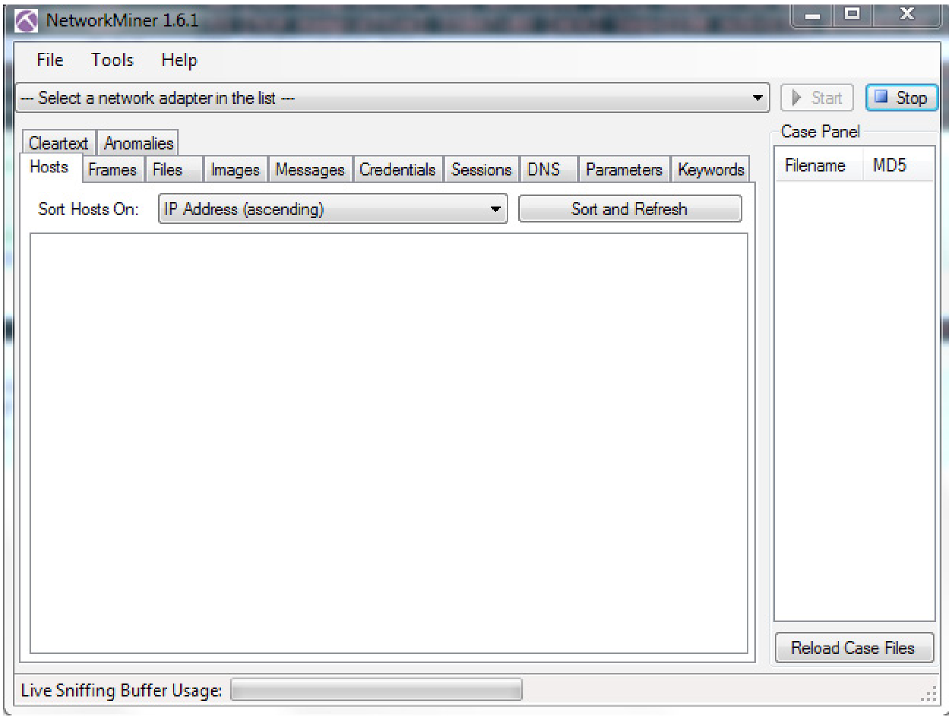
IMPORTING THE PCAP FILE

PCAP file is the base for all packet analysis programs and tools as PCAP is the file format in which the captured packets will be stored in the system. PCAP stands for packet capture and it is an application programming interface that captures traffic from the network. WinPCap is used to capture the packets that can later be used for analysis. Since the coding for PCAP files is written using C language, it is required to include wrapper classes for Java and other programming languages. WinPCap is used by almost the entire packet sniffing programs and network monitors that analyze traffic. NetworkMiner also uses WinPCap to capture packets from the network and the captured packets are stored in files with extension PCAP. These stored files can later be retrieved and opened using NetworkMiner tool. The packets in PCAP can be extracted and analyzed. CapLoader is a part of the NetworkMiner packet sniffing tool that captures and loads large amount of packets. CapLoader is used to index all the PCAP files present in the system, which can then visualize the files in the form of TCP and UDP flows. Users will be seeing all the packets that are captured from the network, and there are filters for tracking only the files of interest, leaving out others. When the files are loaded and filtered, they can be easily sent for analysis to any of the packet analyzing programs including Wireshark. Even though the stand alone data capturing mode of NetworkMiner is sufficient for small to medium file sizes, tools such as CapLoader are essential for capturing, loading and filtering packets and files of size more than gigabytes. Importing PCAP files in NetworkMiner is just a click away as all you need to do is drag the files to the main window of the tool. All the packets will be displayed which can then be selected and filtered.

FILE CARVING

File carving is the technique of searching input from files and other objects using content instead of metadata. File carving comes in handy when the directory files are corrupted. In such case, file carving helps in recovering files and fragments. Similar to file carving, there is memory carving that helps in recovering memory dumps from unknown memory locations. File carving in NetworkMiner and other packet sniffing tools searches for file header and footer and carves out the data present between them. The file carving should always be performed only in the disk image and not in the original file. File carving in NetworkMiner carves packets of any file and saves them in PCAP format. The tool uses techniques from memory and network forensics to extract both sent and received packets from the network including the complete payload. When carving is not performed, the analysis is limited only to the IP address, port numbers and endpoints of communication. When carving is performed, the contents of the communication can be retrieved as well.

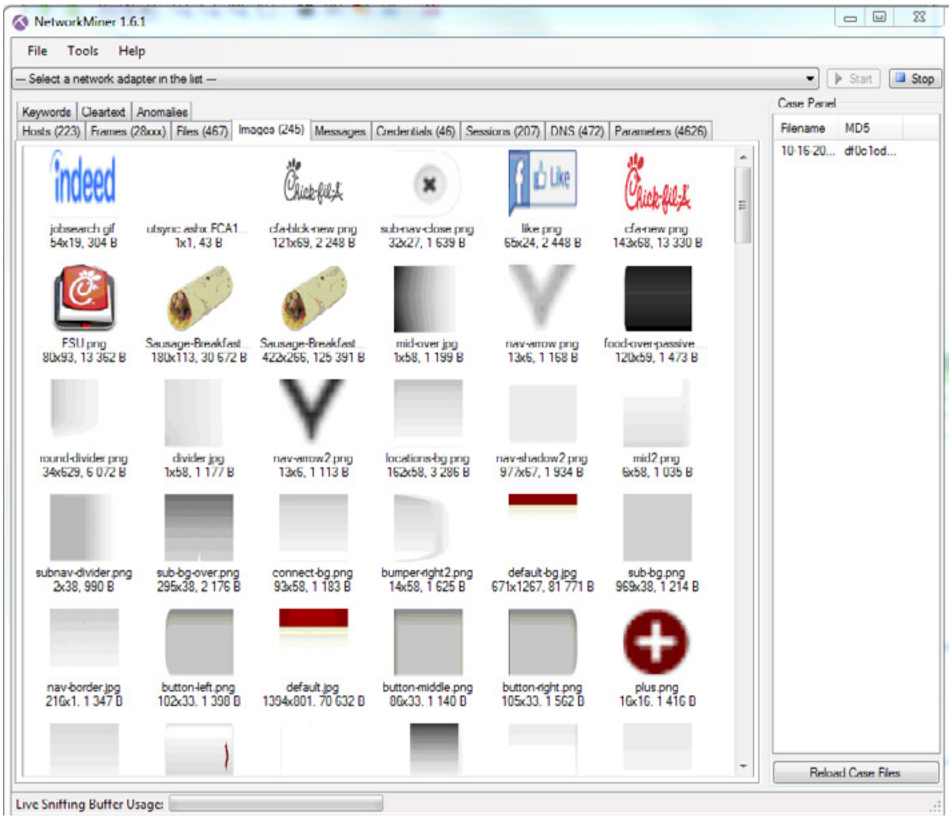
Importing a PCAP: 1. Open Network Miner. Go to File à Open and select the pcap file you saved from Wireshark



Wait for Network Miner to analyze the file.

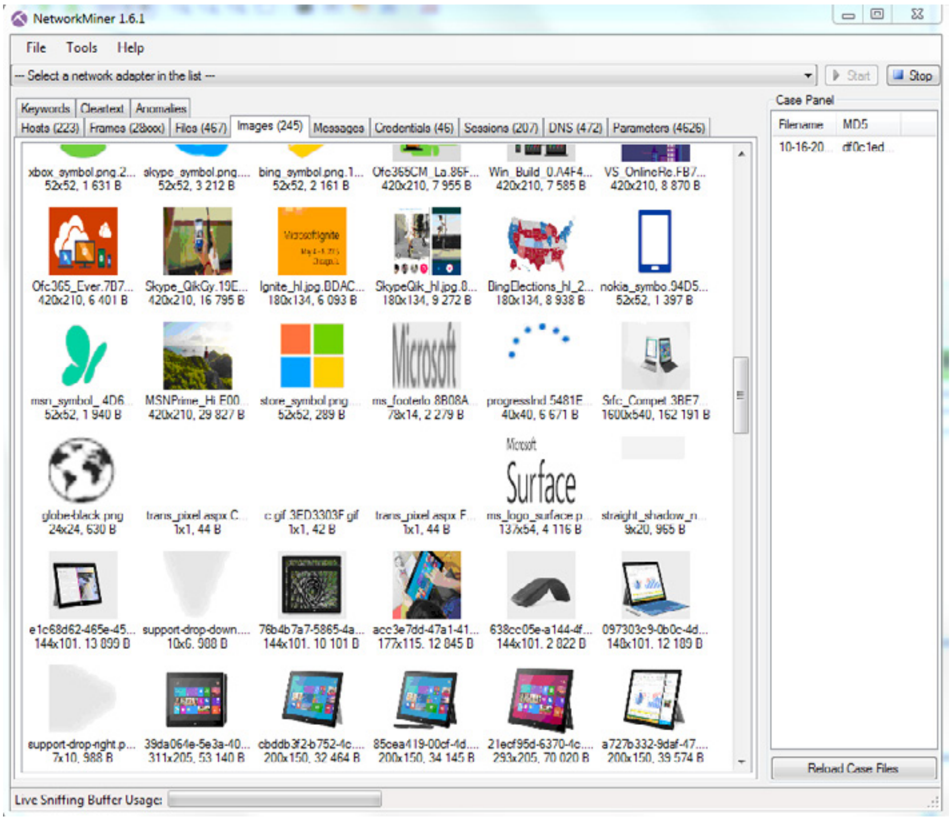
IMAGE CARVING

Image carving is a similar variant of the file carving technique in which images are carved and displayed in place of files. The images that are transferred in the communication can be picked up for analysis. For long time, it was really tough to capture and analyze images that are transferred across the network. Most of the sniffing programs available today come with the concept of image carving. Similar to file carving, image carving looks for content present in the transfer instead of metadata. Using this technique, the images that are lost in communication can also be retrieved and rendered. To carve image files, user needs to enter the type of image they need to retrieve and carve the memory for fetching any images that are present in the system. Image carving can also be performed in mobile forensics to track and monitor images that are stored in the mobile phone. Even though image carver available in market today supports few image formats, new formats will be added in very short time. The images tab in NetworkMiner displays all the captured images. The tab header displays the number of images that are captured in total. The images tab can be found next to files tab and previous to messages tab.



jobsearch gif 54x19, 304 B	utsync ashx FCA1... 1x1, 43 B	cfa-blck-new png 121x69, 2 248 B	sub-nav-close png 32x27, 1 639 B	like png 65x24, 2 448 B	cfa-new png 143x68, 13 330 B
FSU png 80x93, 13 362 B	Sausage-Breakfast... 180x113, 30 672 B	Sausage-Breakfast... 422x266, 125 391 B	mid-over jpg 1x58, 1 199 B	nav-arrow png 13x6, 1 168 B	food-over-passive... 120x59, 1 473 B
round-divider png 34x629, 6 072 B	divider jpg 1x58, 1 177 B	nav-arrow2 png 13x6, 1 113 B	locations-bg png 162x58, 3 286 B	nav-shadow2 png 977x67, 1 934 B	mid2 png 6x58, 1 035 B
subnav-divider.png 2x38, 990 B	sub-bg-over.png 295x38, 2 176 B	connect-bg.png 93x58, 1 183 B	bumper-right2.png 14x58, 1 625 B	default-bg.jpg 671x1267, 81 771 B	sub-bg.png 969x38, 1 214 B
nav-border.jpg 216x1. 1 347 D	button-left.png 102x33. 1 398 D	default.jpg 1394x801. 70 632 D	button-middle.png 86x33. 1 140 D	button-right.png 105x33. 1 562 D	plus.png 16x16. 1 416 D

Filename	MD5
10 16 20...	df0c1od...



xbox_symbol.png.2... 52x52, 1 631 B	skype_symbol.png.... 52x52, 3 212 B	bing_symbol.png.1... 52x52, 2 161 B	Ofc365CM_La.86F... 420x210, 7 955 B	Win_Build_0.A4F4... 420x210, 7 585 B	VS_OnlineRc.FB7... 420x210, 8 870 B
Ofc365_Ever.7D7... 420x210, 6 401 B	Skype_QikGy.19E... 420x210, 16 795 B	Ignite_hl.jpg.BDAC... 180x134, 6 093 B	SkypeQik_hl.jpg.8... 180x134, 9 272 B	BingElections_hl_2... 180x134, 8 938 B	nokia_symbo.94D5... 52x52, 1 397 B
msn_symbol_4D6... 52x52, 1 940 B	MSNPrime_Hi.E00... 420x210, 29 827 B	store_symbol.png.... 52x52, 289 B	ms_footerlo.8B08A... 78x14, 2 279 B	progressInd.5481E... 40x40, 6 671 B	Sfc_Compet.3BE7... 1600x540, 162 191 B
globe-black.png 24x24, 630 B	trans_pixel.aspx.C... 1x1, 44 B	c.gif.3ED3303F.gif 1x1, 42 B	trans_pixel.aspx.F... 1x1, 44 B	ms_logo_surface.p... 137x54, 4 116 B	straight_shadow_n... 9x20, 965 B
e1c68d62-465e-45... 144x101. 13 899 D	support-drop-down.... 10x6. 908 D	76b4b7a7-5865-4a... 144x101. 10 101 D	acc3e7dd-47a1-41... 177x115. 12 845 D	638cc05e-a144-4f... 144x101. 2 822 D	097303c9-0b0c-4d... 148x101. 12 189 D
support-drop-right.p... 7x10, 988 B	39da064e-5e3a-40... 311x205, 53 140 B	cbddb3f2-b752-4c.... 200x150, 32 464 B	85cea419-00cf-4d.... 200x150, 34 145 B	21ecf95d-6370-4c.... 293x205, 70 020 B	a727b332-9daf-47.... 200x150, 39 574 B

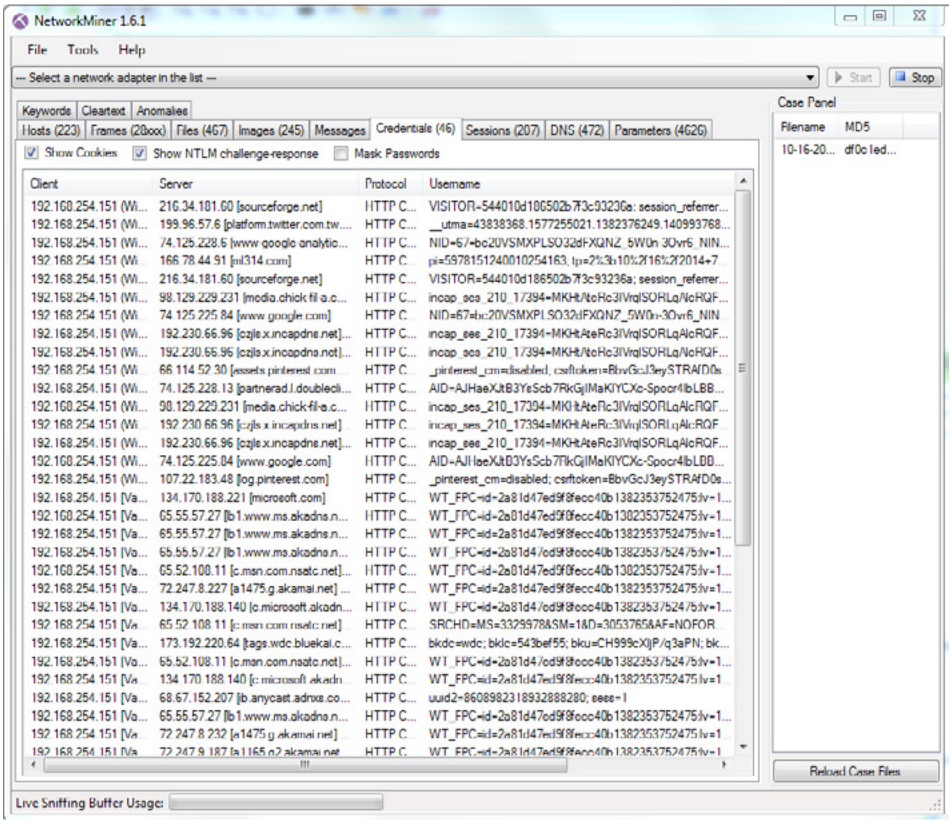
Filename	MD5
10-16-20	df0c1ed

MESSAGE CARVING

Consider your organizations' network where individuals use their own mobile phones to transfer text messages across the channel. These text messages do not come into the corporate network as most of them are sent through various channels. Only some of the messages are sent using the company's texting app. Hence, network administrators were able to track down messages that were sent only through company's texting app and not the other messaging channels. But advancements in technology and introduction of forensics tools have made it possible to retrieve any number of messages from mobile device. The tools are highly advanced that even very old messages that are deleted from the mobile can be traced down. This technique of capturing messages from mobile devices is called message carving. Message carving has various advantages as they help in retrieving lost and deleted messages from the communication. They also help in tracking down messages that are not transferred through company mobile app. NetworkMiner comes with the option of message carving and all text messages that are transferred through the supported protocols can be captured using NetworkMiner. The extracted messages are stored in messages tab that can be found next to images tab. The tab header also displays the number of messages that are captured and displayed.

CREDENTIALS

The main window of NetworkMiner tool has various tabs of which one of the tabs is Credentials. The Credentials tab displays the user credentials such as username and password for all the supported protocols that are transmitted without encryption. All the credentials are extracted by the tool and are displayed in the tab which allows user to identify the protocol that does not encrypt the password. Using this technique, the username of popular online services such as Gmail can be identified. It is always recommended that all the protocols should encrypt and transfer user credentials. Navigate to Credentials tab of NetworkMiner to identify all the credentials that are extracted and displayed. The tab header displays the count of number of credentials that are displayed by the tool.


NetworkMiner 1.6.1
File Tools Help
--- Select a network adapter in the list ---
Start
Stop
Keywords | Cleartext | Anomalies
Hosts (223) | Frames (28xxx) | Files (467) | Images (245) | Messages | Credentials (46) | Sessions (207) | DNS (472) | Parameters (4626)
Case Panel
Reload Case Files
Live Sniffing Buffer Usage:

☑ Show Cookies ☑ Show NTLM challenge-response □ Mask Passwords

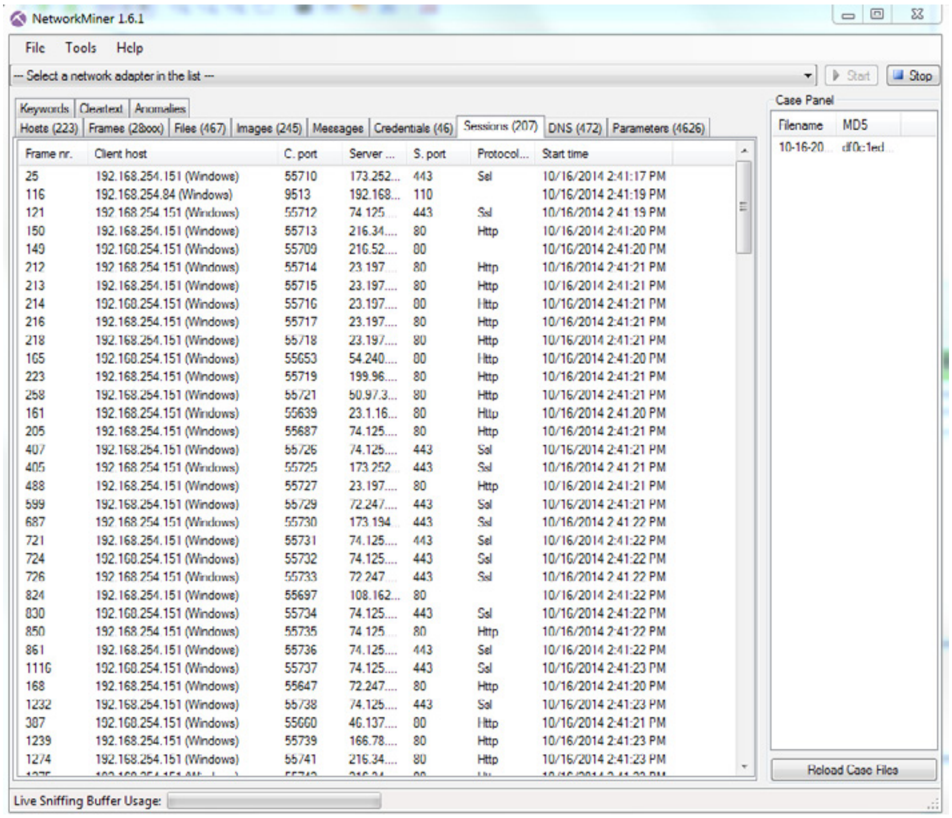
Client	Server	Protocol	Username
192.168.254.151 (Wi...	216.34.181.60 [sourceforge.net]	HTTP C...	VISITOR=544010d186502b7f3c93236a; session_referrer...
192.168.254.151 (Wi...	199.96.57.6 [platform.twitter.com.tw....	HTTP C...	__utma=43838368.1577255021.1382376249.140993768...
192.168.254.151 (Wi...	74.125.228.6 [www.google-analytic...	HTTP C...	NID=67=bc20VSMXPLSO32dFXQNZ_5W0n-3Ovr6_NIN...
192.168.254.151 (Wi...	166.78.44.91 [ml314.com]	HTTP C...	pi=5978151240010254163; tp=2%3b10%2f16%2f2014+7...
192.168.254.151 (Wi...	216.34.181.60 [sourceforge.net]	HTTP C...	VISITOR=544010d186502b7f3c93236a; session_referrer...
192.168.254.151 (Wi...	98.129.229.231 [media.chick-fil-a.c...	HTTP C...	incap_ses_210_17394=MKHtAteRc3IVrqISORLqAlcRQF...
192.168.254.151 (Wi...	74.125.225.84 [www.google.com]	HTTP C...	NID=67=bc20VSMXPLSO32dFXQNZ_5W0n-3Ovr6_NIN...
192.168.254.151 (Wi...	192.230.66.96 [czjls.x.incapdns.net]...	HTTP C...	incap_ses_210_17394=MKHtAteRc3IVrqISORLqAlcRQF...
192.168.254.151 (Wi...	192.230.66.96 [czjls.x.incapdns.net]...	HTTP C...	incap_ses_210_17394=MKHtAteRc3IVrqISORLqAlcRQF...
192.168.254.151 (Wi...	66.114.52.30 [assets.pinterest.com....	HTTP C...	_pinterest_cm=disabled; csrftoken=BbvGcJ3eySTRAfD0s...
192.168.254.151 (Wi...	74.125.228.13 [partnerad.l.doublecli...	HTTP C...	AID=AJHaeXJtB3YsScb7RkGjIMaKIYCXc-Spocr4lbLBB...
192.168.254.151 (Wi...	98.129.229.231 [media.chick-fil-a.c...	HTTP C...	incap_ses_210_17394=MKHtAteRc3IVrqISORLqAlcRQF...
192.168.254.151 (Wi...	192.230.66.96 [czjls.x.incapdns.net]...	HTTP C...	incap_ses_210_17394=MKHtAteRc3IVrqISORLqAlcRQF...
192.168.254.151 (Wi...	192.230.66.96 [czjls.x.incapdns.net]...	HTTP C...	incap_ses_210_17394=MKHtAteRc3IVrqISORLqAlcRQF...
192.168.254.151 (Wi...	74.125.225.84 [www.google.com]	HTTP C...	AID=AJHaeXJtB3YsScb7RkGjIMaKIYCXc-Spocr4lbLBB...
192.168.254.151 (Wi...	107.22.183.48 [log.pinterest.com]	HTTP C...	_pinterest_cm=disabled; csrftoken=BbvGcJ3eySTRAfD0s...
192.168.254.151 [Va...	134.170.188.221 [microsoft.com]	HTTP C...	WT_FPC=id=2a81d47ed9f8fecc40b1382353752475:lv=1...
192.168.254.151 [Va...	65.55.57.27 [lb1.www.ms.akadns.n...	HTTP C...	WT_FPC=id=2a81d47ed9f8fecc40b1382353752475:lv=1...
192.168.254.151 [Va...	65.55.57.27 [lb1.www.ms.akadns.n...	HTTP C...	WT_FPC=id=2a81d47ed9f8fecc40b1382353752475:lv=1...
192.168.254.151 [Va...	65.55.57.27 [lb1.www.ms.akadns.n...	HTTP C...	WT_FPC=id=2a81d47ed9f8fecc40b1382353752475:lv=1...
192.168.254.151 [Va...	65.52.108.11 [c.msn.com.nsatc.net]...	HTTP C...	WT_FPC=id=2a81d47ed9f8fecc40b1382353752475:lv=1...
192.168.254.151 [Va...	72.247.8.227 [a1475.g.akamai.net] ...	HTTP C...	WT_FPC=id=2a81d47ed9f8fecc40b1382353752475:lv=1...
192.168.254.151 [Va...	134.170.188.140 [c.microsoft.akadn...	HTTP C...	WT_FPC=id=2a81d47ed9f8fecc40b1382353752475:lv=1...
192.168.254.151 [Va...	65.52.108.11 [c.msn.com.nsatc.net]...	HTTP C...	SRCHD=MS=3329978&SM=1&D=3053765&AF=NOFOR...
192.168.254.151 [Va...	173.192.220.64 [tags.wdc.bluekai.c...	HTTP C...	bkdc=wdc; bklc=543bef55; bku=CH999cXIjP/q3aPN; bk...
192.168.254.151 [Va...	65.52.108.11 [c.msn.com.nsatc.net]...	HTTP C...	WT_FPC=id=2a81d47ed9f8fecc40b1382353752475:lv=1...
192.168.254.151 [Va...	134.170.188.140 [c.microsoft.akadn...	HTTP C...	WT_FPC=id=2a81d47ed9f8fecc40b1382353752475:lv=1...
192.168.254.151 [Va...	68.67.152.207 [ib.anycast.adnxs.co...	HTTP C...	uuid2=8608982318932888280; sess=1
192.168.254.151 [Va...	65.55.57.27 [lb1.www.ms.akadns.n...	HTTP C...	WT_FPC=id=2a81d47ed9f8fecc40b1382353752475:lv=1...
192.168.254.151 [Va...	72.247.8.232 [a1475.g.akamai.net] ...	HTTP C...	WT_FPC=id=2a81d47ed9f8fecc40b1382353752475:lv=1...
192.168.254.151 [Va	72.247.9.187 [a1165.g2.akamai.net	HTTP C	WT_FPC=id=2a81d47ed9f8fecc40b1382353752475:lv=1

Filename	MD5
10-16-20...	df0c1ed...

SESSIONS

Apart from analyzing packets that are captured and the files, images and messages that are carved, NetworkMiner also has many other features that are of importance. Consider a high end security network in which it is not recommended to allow all incoming and outgoing communications. In such cases, it would be better to prevent all of the incoming and outgoing traffic with the exception of certain sessions.

The sessions that are considered for exception can be allowed to make communications across the network. To identify which sessions are present in the network, network administrator should know all the session transactions across the network. NetworkMiner comes handy at such times as it provides a list of all enabled sessions in the network. The sessions tab in the NetworkMiner window displays all the sessions that are available. The tab header displays the total count of sessions that are active at the current time. The network administrator can determine which sessions are important and which are not, thereby preventing unwanted sessions to undergo communication.



Frame nr.	Client host	C. port	Server ...	S. port	Protocol...	Start time
25	192.168.254.151 (Windows)	55710	173.252...	443	Ssl	10/16/2014 2:41:17 PM
116	192.168.254.84 (Windows)	9513	192.168...	110		10/16/2014 2:41:19 PM
121	192.168.254.151 (Windows)	55712	74.125....	443	Ssl	10/16/2014 2:41:19 PM
150	192.168.254.151 (Windows)	55713	216.34....	80	Http	10/16/2014 2:41:20 PM
149	192.168.254.151 (Windows)	55709	216.52....	80		10/16/2014 2:41:20 PM
212	192.168.254.151 (Windows)	55714	23.197....	80	Http	10/16/2014 2:41:21 PM
213	192.168.254.151 (Windows)	55715	23.197....	80	Http	10/16/2014 2:41:21 PM
214	192.168.254.151 (Windows)	55716	23.197....	80	Http	10/16/2014 2:41:21 PM
216	192.168.254.151 (Windows)	55717	23.197....	80	Http	10/16/2014 2:41:21 PM
218	192.168.254.151 (Windows)	55718	23.197....	80	Http	10/16/2014 2:41:21 PM
165	192.168.254.151 (Windows)	55653	54.240....	80	Http	10/16/2014 2:41:20 PM
223	192.168.254.151 (Windows)	55719	199.96....	80	Http	10/16/2014 2:41:21 PM
258	192.168.254.151 (Windows)	55721	50.97.3....	80	Http	10/16/2014 2:41:21 PM
161	192.168.254.151 (Windows)	55639	23.1.16....	80	Http	10/16/2014 2:41:20 PM
205	192.168.254.151 (Windows)	55687	74.125....	80	Http	10/16/2014 2:41:21 PM
407	192.168.254.151 (Windows)	55726	74.125....	443	Ssl	10/16/2014 2:41:21 PM
405	192.168.254.151 (Windows)	55725	173.252....	443	Ssl	10/16/2014 2:41:21 PM
488	192.168.254.151 (Windows)	55727	23.197....	80	Http	10/16/2014 2:41:21 PM
599	192.168.254.151 (Windows)	55729	72.247....	443	Ssl	10/16/2014 2:41:21 PM
687	192.168.254.151 (Windows)	55730	173.194....	443	Ssl	10/16/2014 2:41:22 PM
721	192.168.254.151 (Windows)	55731	74.125....	443	Ssl	10/16/2014 2:41:22 PM
724	192.168.254.151 (Windows)	55732	74.125....	443	Ssl	10/16/2014 2:41:22 PM
726	192.168.254.151 (Windows)	55733	72.247....	443	Ssl	10/16/2014 2:41:22 PM
824	192.168.254.151 (Windows)	55697	108.162....	80		10/16/2014 2:41:22 PM
830	192.168.254.151 (Windows)	55734	74.125....	443	Ssl	10/16/2014 2:41:22 PM
850	192.168.254.151 (Windows)	55735	74.125....	80	Http	10/16/2014 2:41:22 PM
861	192.168.254.151 (Windows)	55736	74.125....	443	Ssl	10/16/2014 2:41:22 PM
1116	192.168.254.151 (Windows)	55737	74.125....	443	Ssl	10/16/2014 2:41:23 PM
168	192.168.254.151 (Windows)	55647	72.247....	80	Http	10/16/2014 2:41:20 PM
1232	192.168.254.151 (Windows)	55738	74.125....	443	Ssl	10/16/2014 2:41:23 PM
387	192.168.254.151 (Windows)	55660	46.137....	80	Http	10/16/2014 2:41:21 PM
1239	192.168.254.151 (Windows)	55739	166.78....	80	Http	10/16/2014 2:41:23 PM
1274	192.168.254.151 (Windows)	55741	216.34....	80	Http	10/16/2014 2:41:23 PM

Filename	MD5
10-16-20...	df0c:1ed...

KEYWORD SEARCHING

Another important feature of NetworkMiner tool is to search the captured protocols using keywords. Since there will be millions of packets in a PCAP file, it would be difficult to search for a particular protocol. The keyword searching functionality enables users to search for specific keywords. The tool will retrieve all packets that have the keyword specified irrespective of the protocol.

With so many interesting and useful features, NetworkMiner is serving many network administrators with all they need to analyze traffic using captured files. Even though NetworkMiner captures live traffic, the tool was first designed to load and analyze packets that are captured using other programs.

NETWITNESS INVESTIGATOR

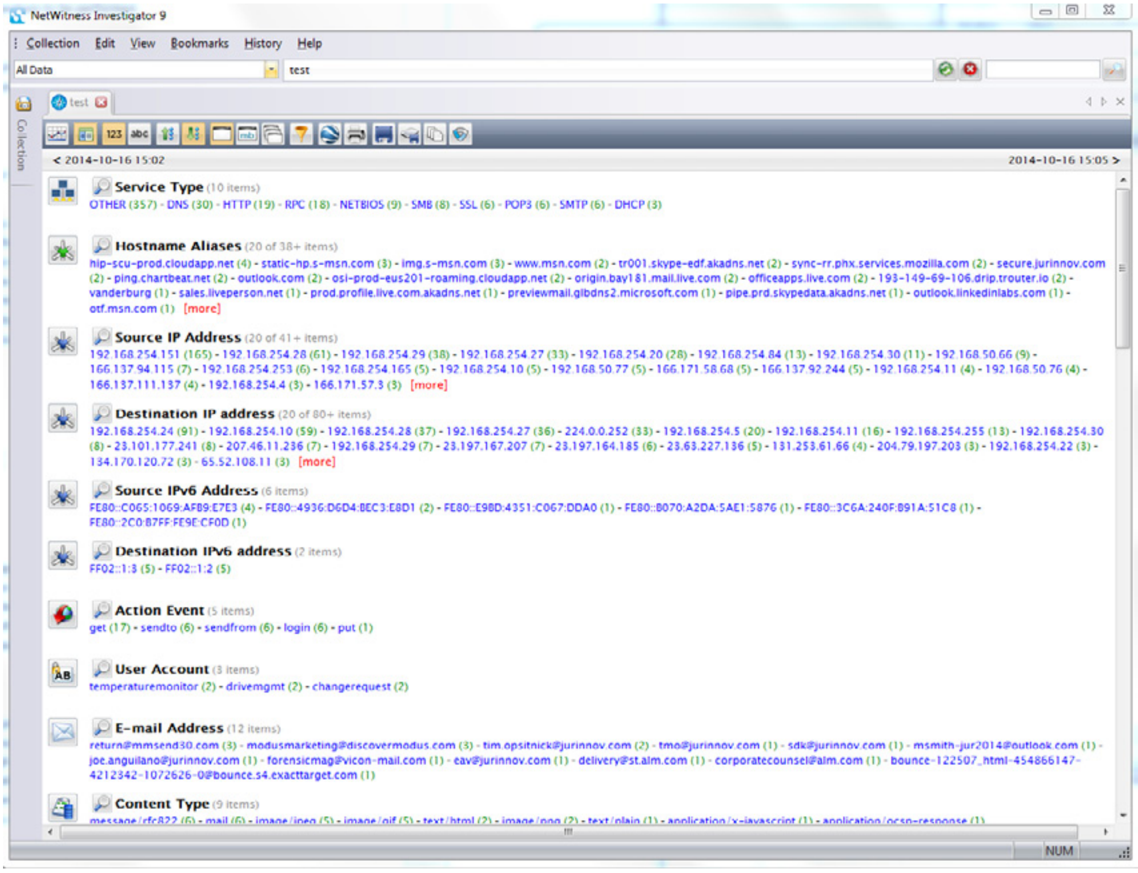
It is important to know everything that is happening in the network and this requires high skilled network administrator and the support of high end tool that drills into the network to capture data. NetWitness Investigator is one such tool that performs extensive threat analysis on all components of the network model. It is an award winning tool that analyses raw data for all information required to investigate fraud and forensics. Capturing mountains of data will help in fetching the required information to study the network. The NetWitness Investigator comes with various tools that help in analyzing packets and identify any potential threats to the network. The tool uses WinPCap driver for capturing packets and to analyze raw data present in various formats such as TCP, PCAP, GZ files etc.

The application can analyze both live packet data and input file data and reports can be generated in various formats for the study of informational risk, events and users in the network, hostname aliases, source and destination of all packets etc. Every session of data capture is saved as individual collection which will help in easy navigation of packets and data. The saved sessions can be later retrieved and analyzed with ease. The tool also comes with hosts of customized options for users to personalize the packets they need for analysis. There are many other advanced features such as bookmarking the packets collected which can be easily indexed and searched, history section that would provide the history of when the packets were collected. If Google Earth package is installed in the system, the tool will also locate the place from where the packets were captured and analyzed. The latest version of the application supports 802.11 interface, right click custom actions and support for Windows latest version. The free version of the software allows for 25 simultaneous packet capturing sessions with maximum capacity of 1 GB per capture. The professional edition has many advanced features when compared to the free version, but the free version is more than enough for all basic analysis to be performed. The software was lately acquired by EMC/ RSA in 2011 but the software is still available in free version as well. The freeware of the software can be downloaded from the website: *http://www.emc.com/security/security-analytics/security-analytics.htm#!freeware.*

IMPORTING THE PCAP FILE

As with other packet capturing programs such as Wireshark and NetworkMiner seen above, importing PCAP files for packet analysis is an easy task. All the user needs to do is create a new collection and import the file to be analyzed.

- Create a new collection
- Select the collection
- Click on the Import Packets icon and browse to the pcap file you want to import. Select it to import.
- Wait for the progress bar to complete



< 2014-10-16 15:02 2014-10-16 15:05 >

Service Type (10 items)
OTHER (357) - DNS (30) - HTTP (19) - RPC (18) - NETBIOS (9) - SMB (8) - SSL (6) - POP3 (6) - SMTP (6) - DHCP (3)

Hostname Aliases (20 of 38+ items)
hip-scu-prod.cloudapp.net (4) - static-hp.s-msn.com (3) - img.s-msn.com (3) - www.msn.com (2) - tr001.skype-edf.akadns.net (2) - sync-rr.phx.services.mozilla.com (2) - secure.jurinnov.com (2) - ping.chartbeat.net (2) - outlook.com (2) - osi-prod-eus201-roaming.cloudapp.net (2) - origin.bay181.mail.live.com (2) - officeapps.live.com (2) - 193-149-69-106.drip.trouter.io (2) - vanderburg (1) - sales.liveperson.net (1) - prod.profile.live.com.akadns.net (1) - previewmail.glbdns2.microsoft.com (1) - pipe.prd.skypedata.akadns.net (1) - outlook.linkedinlabs.com (1) - otf.msn.com (1) [more]

Source IP Address (20 of 41+ items)
192.168.254.151 (165) - 192.168.254.28 (61) - 192.168.254.29 (38) - 192.168.254.27 (33) - 192.168.254.20 (28) - 192.168.254.84 (13) - 192.168.254.30 (11) - 192.168.50.66 (9) - 166.137.94.115 (7) - 192.168.254.253 (6) - 192.168.254.165 (5) - 192.168.254.10 (5) - 192.168.50.77 (5) - 166.171.58.68 (5) - 166.137.92.244 (5) - 192.168.254.11 (4) - 192.168.50.76 (4) - 166.137.111.137 (4) - 192.168.254.4 (3) - 166.171.57.3 (3) [more]

Destination IP address (20 of 80+ items)
192.168.254.24 (91) - 192.168.254.10 (59) - 192.168.254.28 (37) - 192.168.254.27 (36) - 224.0.0.252 (33) - 192.168.254.5 (20) - 192.168.254.11 (16) - 192.168.254.255 (13) - 192.168.254.30 (8) - 23.101.177.241 (8) - 207.46.11.236 (7) - 192.168.254.29 (7) - 23.197.167.207 (7) - 23.197.164.185 (6) - 23.63.227.136 (5) - 131.253.61.66 (4) - 204.79.197.203 (3) - 192.168.254.22 (3) - 134.170.120.72 (3) - 65.52.108.11 (3) [more]

Source IPv6 Address (6 items)
FE80::C065:1069:AFB9:E7E3 (4) - FE80::4936:D6D4:8EC3:E8D1 (2) - FE80::E98D:4351:C067:DDA0 (1) - FE80::8070:A2DA:5AE1:5876 (1) - FE80::3C6A:240F:891A:51C8 (1) - FE80::2C0:87FF:FE9E:CF0D (1)

Destination IPv6 address (2 items)
FF02::1:3 (5) - FF02::1:2 (5)

Action Event (5 items)
get (17) - sendto (6) - sendfrom (6) - login (6) - put (1)

User Account (3 items)
temperaturemonitor (2) - drivemgmt (2) - changerequest (2)

E-mail Address (12 items)
return@mmsend30.com (3) - modusmarketing@discovermodus.com (3) - tim.opsitnick@jurinnov.com (2) - tmo@jurinnov.com (1) - sdk@jurinnov.com (1) - msmith-jur2014@outlook.com (1) - joe.anguilano@jurinnov.com (1) - forensicmag@vicon-mail.com (1) - eav@jurinnov.com (1) - delivery@st.alm.com (1) - corporatecounsel@alm.com (1) - bounce-122507_html-454866147-4212342-1072626-0@bounce.s4.exacttarget.com (1)

Content Type (9 items)
message/rfc822 (6) - mail (6) - image/jpeg (5) - image/gif (5) - text/html (2) - image/png (2) - text/plain (1) - application/x-javascript (1) - application/ocsp-response (1)

The default collection in Investigator is called the demo collection. User needs to create a new collection using the option present in Menu or by using CTRL+L shortcut key. Once the new collection is created, double click on the collection name to see the status change to Ready. Right click on the collection name to import packets and select the PCAP file which is installed in the system. The import process will begin right away and depending on the size of the packet. As discussed in previous paragraph, the free version of the tool allows for 25 simultaneous 1 GB capture data. Hence depending on the size of the packet you are importing, it would take a little while for all data to be captured. Once the application completed importing PCAP Files, the status would change to Ready. It is time to open the captured data and explore the world of packets. The source, destination, aliases, services and ports are given in the upper part of the report while the countries and cities are given in the lower part. A number is displayed adjacent to every row and it displays the number of packets that are present. Clicking on that number will display the detailed summary of the particular packet. From here on, it is all up to you to analyze whatever information you want from the packets imported.

DETECTING DATA BREACHES

Forensic analysis usually detects all the data breaches present in the system with the help of malicious activity present in the audit logs. It would have been more useful if the network administrator or the security team analyzed these breaches in real time. Even though it is difficult to analyze and understand the potential threats, the packet sniffing programs come in handy. When real time information of packets and traffic in the network is identified, the data breaches can also be identified. It will be possible to spot differences in network behavior and user can go for in depth analysis on the packets that behave differently.

CCFP℠

Certified Cyber Forensics Professional

THE STANDARD in CYBER FORENSICS

Cyber forensic knowledge requirements have expanded and evolved just as the nature of digital information has, requiring cyber forensics professionals to understand far more than just hard drive and intrusion analysis.

The Certified Cyber Forensics Professional (CCFP℠) indicates a standard of expertise in forensics techniques and procedures, standards of practice, and legal and ethical principles to assure accurate, complete and reliable digital evidence admissible to a court of law.

Do you have what it takes to become a CCFP?

DOWNLOAD the CCFP Snapshot >

CCFP™ Certified Cyber Forensics Professional

THE STANDARD IN CYBER FORENSICS

CCFP SNAPSHOT

CCFP℠ is the definitive credential for experienced and knowledgeable cyber forensics professionals in a variety of environments. Whether you're a law enforcement officer supporting criminal investigations, a digital forensics specialist consulting clients, or an eDiscovery specialist working on litigation, CCFP demonstrates your ability to gather, analyze, and deliver digital evidence that is accurate, complete, and reliable. The certification covers a range of skills necessary to support these environments from intrusion analysis to incident response, and newer challenges, such as mobile forensics and cloud forensics.

PERSONAL PROFILE: WHO ARE YOU?

What's required:

CCFP candidates must possess a four-year college degree and a minimum of three years of cumulative paid full-time professional work experience in digital forensics or IT security in three of the six domains of the (ISC)²® CCFP CBK®. Those who do not hold a four-year college degree can obtain the CCFP with six years of cumulative paid full-time digital forensics or IT security experience in three out of the six domains of the CBK.

What job title do you have?

CCFPs hold a diverse range of titles including digital forensics examiner, forensics investigator, computer forensics engineer, eDiscovery consultant and cyber intelligence analyst. Professionals in law enforcement, government entities, corporate environments and firms that provide forensics investigations, all benefit from the certification.

What advice would you give to a young cyber forensics professional (or aspiring professional) in terms of how to get ahead and succeed in their career?

"You do not learn forensics by learning a tool. You have to understand file systems and memory, and once you understand these two things very well, any tool will work. Working in this industry is far more than just finding the evidence; report writing is just as important. Our job as cyber forensics professionals is to interpret and describe our findings."

– James Phillips, CCFP

(ISC)²®

(ISC)²®

What you will learn:

- What role disaster recovery plays in your organization
- How to constructing a disaster recovery plan
- The roles projects play when planning your disaster recovery environment and plan
- Assembling an incident response team and process
- How incident response and disaster recovery are processes that work together

What you should know:

- Familiarity with basics of a information technology organization
- A understanding of risk management
- Basic concept of computer networks