INTRODUCTION TO NETWORK FORENSICS USING WIRESHARK

by Dauda Sule

Network forensics involves recording, monitoring, capturing and analysis of network traffic in a bid to uncover how incidents occurred (like a breach, attack, abuse or error). Network data is highly volatile and may be easily lost if not captured in real-time; for example, if malicious code is sent to an endpoint, the source or path of the code would be difficult to discover if the traffic data was not captured as it was coming in through the network. There are various tools that can be used to capture and analyze network traffic such as NetworkMiner, tcpdump, snort, windump and Wireshark. This article introduces the use of Wireshark for network analysis.

What you will learn:

- Definition of network forensics
- Basic understanding of network forensics
- Basic network analysis using Wireshark

What you should know:

- Basic understanding of computer networks
- How to operate computer applications and software
- Basic understanding of TCP/IP

Wildpackets (2013) defined network forensics as "the process of capturing, storing, and analyzing network events", data of which can be used to solve network breaches, improve network performance and identify rogue activity. The site further states that network forensics can be used to monitor users and devices, identify sources of data loss and points of security breaches, analyze business transactions and point out the origin of intermittent network issues. Activity monitoring can help identify abnormal traffic, like a change in the network use pattern of a particular endpoint which might signify something is wrong. Network logs from network control mechanisms like routers and firewalls usually provide a good source of digital evidence (Casey, 2004).

Wireshark is an open source network sniffer and protocol analyzer. A packet sniffer is a passive tool used to capture messages being transmitted to and from a system over a network. It is passive because it only monitors and records packets being sent and received on a system, not sending or receiving directly any itself nor interfering with the packets (Kurose and Ross, 2009). Rather what it captures are copies of packets moving within protocols and applications on the system. Wireshark can be used to troubleshoot network problems, examine security problems, debug protocol implementations, and to understand internals of network protocols (Lamping, Sharpe and Warnicke, 2013). Wireshark captures network traffic from both wired and wireless Ethernet networks; however, it does not capture traffic from mobile net-

work dongles on Windows (at least for now). A list of networks that Wireshark can and cannot capture is available here: *http://wiki.wireshark.org/CaptureSetup/NetworkMedia*.

It should be noted that network analysis tools like Wireshark can be used both positively and negatively; network administrators, network security personnel and investigators, and so on use them for troubleshooting, debugging, investigating intrusions and the like, but malicious persons can use it to monitor, spy on and gather reconnaissance data on potential victims. Wireshark is available for free download from the Wireshark website (*http://www.wireshark.org/download.html*).

In an investigation, into a network breach for example, a network sniffer can be used to analyze captured network traffic to discover the path that the intruder followed to get into an organization's network. The network sniffer is able to view the IP address that the intruder used to get in to the network, which can be a starting point for the investigation, even though that may not be a smoking gun. The entry of a malicious code like a network worm can be traced using a network sniffer; it can be used to trace how it got onto the network: could have been downloaded from an endpoint then spread, or could have originated from an endpoint not via download – that could imply infection from a storage device like a thumb drive. Leakage of sensitive data to a competitor could be traced or discovered with a network sniffer by discovering its movement from an IP address in the organization's network to an external IP address. The preceding are just a few basic examples of what network sniffers can be used to uncover whether through analysis of already captured and stored network traffic or live monitoring.

CAPTURING NETWORK TRAFFIC

Following is the use of Wireshark version 1.8.6 on a Windows system to capture and analyze traffic over a network (a wireless network). The wireless access point being a smart phone, and the endpoint a Windows-based Laptop.

Once installed, run Wireshark. The Graphic User Interface as shown in Figure 1 comes up.

The interface is quite user friendly with a variety of options like user guide, help, opening previously captured files, and so on as is visible from Figure 1. Actions can be carried out from the file menu bar, and for some actions shortcuts below the file menu bar and on the interface page (like the starting network capture). For example, clicking on the "Interface List" option under "capture" on the page can be used to view the available network interfaces on the computer whose traffic Wireshark can capture. Once clicked it shows the available network cards and the packets that are sent and captured on them, selecting an available network interface by clicking the checkbox to its immediate left activates the option to start network capture on it (note: there is only one network interface on the system used for this illustration, hence only one card is available in the option). Options for capturing packets can be edited by clicking the options button. Clicking on the details button pops up information about the network interface; like the vendor, the status of the network (connected or disconnected), the throughput and so on, as shown in Figure 2.

A live packet capture can be started by any of the following:

- selecting the required network interface and clicking the start button in the "Interface List" as described above;
- by selecting the desired interface on the main page then clicking "start" above it;
- clicking "capture" on the file menu bar then click "start";

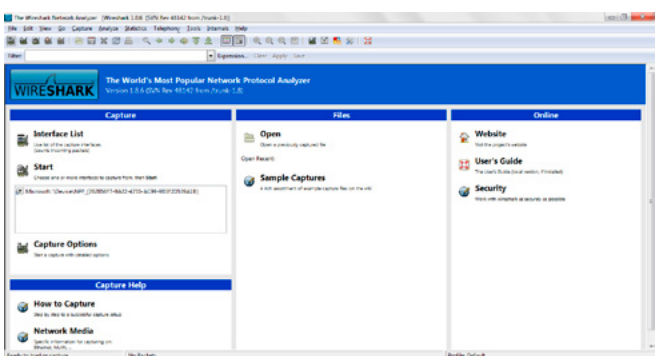


Figure 1. *Initial view when Wireshark is run*

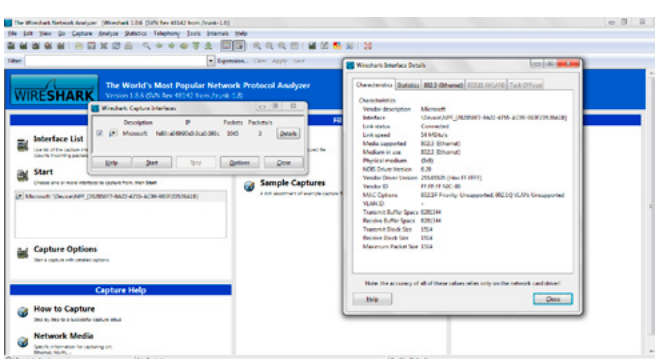


Figure 2. *Interface list showing details of network card*

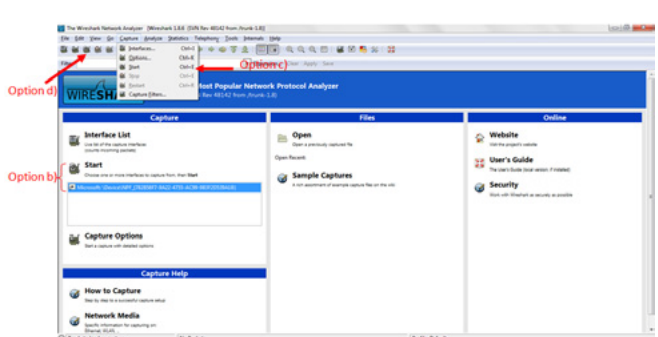


Figure 3. *Capture start options*

- clicking the start shortcut on the bar below the file menu bar;
- using [Ctrl + E] keyboard shortcut.

Locations of options b) to d) are depicted in Figure 3.

Once the packet capture is initiated from any of the above mentioned options, Wireshark starts capturing packets as depicted in Figure 4. The main subdivisions of the interface follow:

- The command menus: located at the topmost, these are made up of the file menu bar and capture menu bar. The file menu bar is a normal file bar, while the capture menu toolbar consists of capture shortcuts which can be gotten from the file menu.
- Packet filter toolbar: this is just below the capture menu bar. It is used to filter the type of packets information displayed in the packet list pane; for example based on protocol, this makes it possible to display only packet data of the selected protocol.

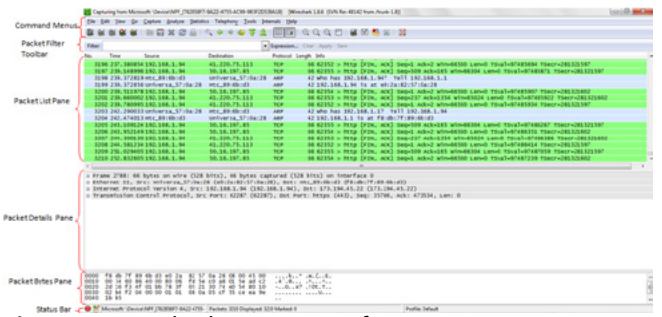


Figure 4. *Wireshark capture interface*

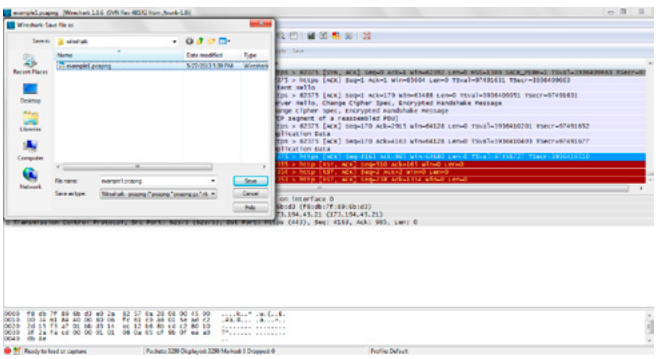


Figure 5. *Saving a packet capture*

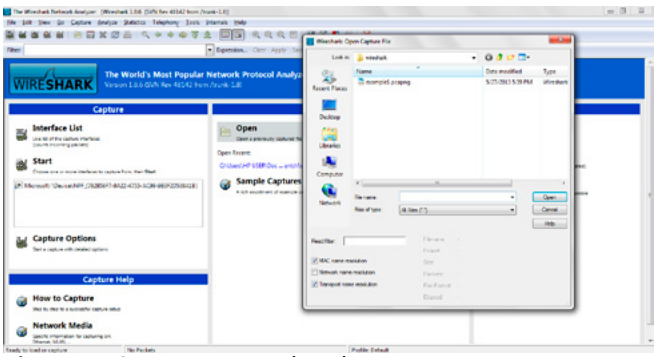


Figure 6. *Opening a stored packet capture*

- Packet list pane: this displays the summary of packets captured each in a row. It shows in each row the Wireshark assigned frame number for each packet, the time the packet was captured, the source and destination addresses, the type of protocol, the length and information pertaining to the protocol type.
- Packet details pane: this shows detailed information on any selected packet in the packet-listing window. Any packet selected by clicking on it in the packet-listing window will have displayed in the packet-header details window details of the Ethernet frame, the Internet Protocol, and other protocol details (like TCP, UDP) depending on the protocol of the selected packet. Each of these can be expanded to show further details.
- The packet bytes pane: this shows all the contents of the captured frame in ACSII and hexadecimal format.
- The status bar: shows some details regarding the state of Wireshark and the packets captured.

The packet capture can be stopped from the file menu bar by clicking capture then stop from the drop-down; the stop button on the capture menu bar (the fourth from the left); or hitting Ctrl + E on the keyboard again. The captured packets can then be analyzed immediately or saved till later. The packet capture is saved just as any normal file is saved (Save, Save As, the floppy disk icon shortcut), as shown in Figure 5. A saved packet capture, or captured network traffic stored in logs, can be retrieved and analyzed by opening the file from the directory in which it is stored. The opening is done like any normal document opening from the file menu, or folder icon shortcut, or the “Open” shortcut in the middle of the initial interface page, to retrieve the captured packet file from the location it is stored (Figure 6).

ANALYSIS OF CAPTURED PACKETS

The time a packet was captured is viewable under the time column in the packet list pane. The time display is set by default to the number of seconds from beginning of a capture, which can be adjusted as required using the view option from the file menu bar. From the view option, move the cursor to “Time Display Format”, which will give a drop down list of options, UTC date and time of the day format is chosen in Figure 7. This enables one to know the time (UTC) and date a specific packet was captured. (Note: the UTC date and time of the day format was chosen just for illustrative purposes, it’s not a requirement). If a packet is of particular interest (especially when analyzing an archived network log), knowing the time it was received/sent on a network can help identify who was responsi-

ble, for example if an endpoint is shared by employees working in shifts. The timing can be very useful in an investigation, the time packets were transferred over a network (whether local or UTC or otherwise) on the suspect endpoint is available on the captured network log, this can be used to verify/nullify a suspect's alibi, even more so if combined with CCTV footage or eye-witness accounts. A suspect in a workplace may try to make it look like an infraction took place at a time when he was off-duty, trying to exonerate himself from the infraction, but the logs can reveal the time such an infraction took place, which when combined with the time the suspect was on or off duty can reveal the truth of the matter.

The filter toolbar can be used to select packets based on type of field or protocol. For example, TCP, HTTP, DNS can be criteria for filtering, which will display packets with such criteria in the packet list pane. This is achieved by typing in the criteria in the filter toolbar and clicking on apply. Wireshark is case sensitive and requires that the characters for the filtering criteria be entered in lower case. Figure 8 shows the packet list pane showing filtered results for DNS. This allows the analyst to view and analyze DNS related packets.

To view details and analyze a network packet, the packet is clicked on in the packet list pane, making it highlighted. The time the packet was sent from one IP address to another can be seen under the "Time" column. The IP address from which it was sent, and the one which received it, are visible under the "Source" and "Destination" columns respectively. The protocol type is visible under "Protocol", length shows the packet size in bytes, and information gives a general description of the packet. In Figure 9, the packet selected has a Wireshark frame number of 916 in the capture; it was captured at 20:19 (8:19 PM) on 28th May 2013, sent from IP address 192.168.1.94 to IP address 192.168.1.1, is a ninety-one bytes long DNS packet, and was a standard query.

The selected packet can be further scrutinized in the packet details pane. For this particular packet, you can view details of the Frame, Ethernet II, Internet Protocol version, the transfer protocol (UDP in this case) and DNS; each of them is expandable for full details. For example, expanding the Frame gives further details pertaining to the frame, like the actual time the frame was captured, the frame number, the packet length (in bytes and bits), the protocols in the packet, and so on. The status bar indicates what each detail represents if the detail is clicked on. Figure 10 shows the expanded Frame details in the packet details frame. Ethernet II shows the source port and the destination represented as "Src" and "Dst" respectively. In this example, Ethernet II indicates as below:

```
Src: Universa_57:0a:28 (e0:2a:82:57:0a:28),
Dst: Htc_89:6b:d3 (fb8:d3:7f:89:6b:d3)
```

That means the source of the packet is `Universa_57:0a:28` (which is the endpoints network card), and the destination is `Htc_89:6b:d3` (the destination, in this case a smart phone wireless hotspot). The hexadecimal figures in brackets after both the source and destination represent their MAC (Medium Access Control) addresses in 48-bit – that is the network card's 48-bit MAC address is `e0:2a:82:57:0a:28`, while that of the wireless hotspot is fb8:d3:7f:89:6b:d3. (Note: the 48-bit MAC addresses are in hexadecimal format, the first six digits identify the vendor – called the Organizationally Uniqiue Identifier, OUI – the last six digits represent the MAC's serial number). Once Ethernet II is expanded, it is divided into the destination and the source which are also expandable. Selecting any component of the expanded destination or source highlights the

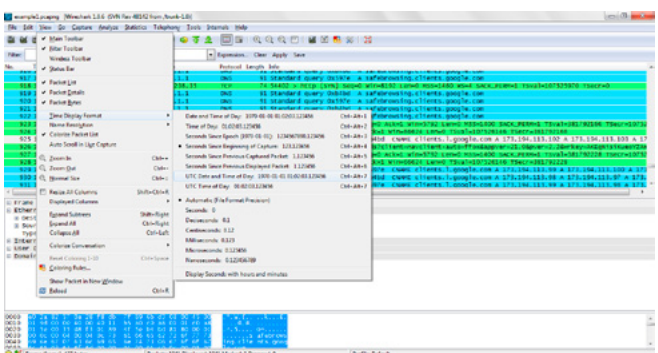


Figure 7. *Changing time display format*

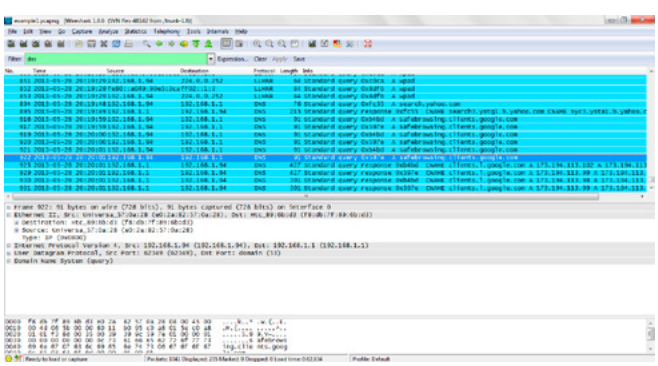


Figure 8. *Filtered DNS results*

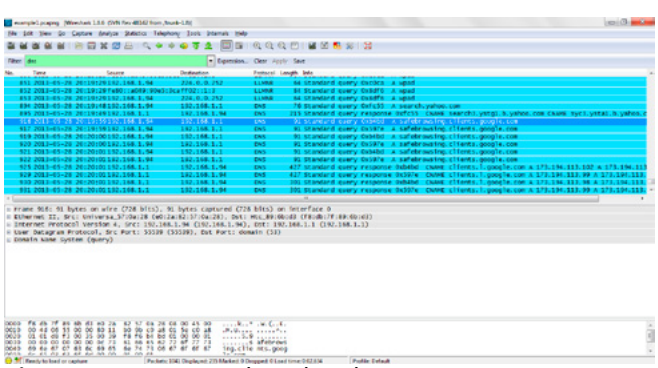


Figure 9. *Viewing a selected packet*

bytes representing such in the packet bytes pane as depicted in Figure 11.

In the Internet Protocol field under Ethernet II in the packet details pane, the version of Internet Protocol, source IP and destination IP address are visible. Upon expansion, it is further broken down into “Differentiated Services Field”, “Flags”, and “Header Checksum”, each giving further information and expandable. It can be observed from Figure 12 that the packet has version 4 of Internet Protocol and the header length is 20bytes. The source and destination IP addresses are also visible as was seen in the packet list pane: 192.168.1.94 and 192.168.1.1 respectively. The User Datagram Protocol field shows the source and destination port numbers, once expanded checksums can be viewed if available. Figure 13 shows the source port as 55539 and the destination port as domain or 53 (port 53 is the default port for Domain Name Server – DNS – protocol), and checksum unavailable.

Ports are used to direct different types of network traffic to specific programs that handle them (SYBEX Inc., 1998). Touch et al (2013) indicated that ports are assigned in different ways based on three ranges viz: system ports – 0 to 1023; user ports – 1024 to 49151; and dynamic and/or private ports – 49152 to 65535. Some common default ports are:

- Port 21 for FTP
- Port 23 for Telnet
- Port 25 for SMTP
- Port 53 for DNS
- Port 80 for HTTP, World Wide Web
- Port 110 for POP

Traffic direct to and/or from a particular port can be used to determine what type of traffic was transferred, for example, traffic on port 25 would most likely be e-mail related. Also, when looking for a particular type of traffic, for example Internet traffic, analysis could be narrowed down to port 80. It should be noted, however, that these ports can be changed; that might be used by an intruder as a way of masking his activities. It is also possible for an organization to use different port numbers than the default for protocols, probably for administrative reasons. Hence, one should have it at the back of one’s head that the port number might not have been used in default form when carrying out an investigation.

Casey (2004) mentioned a case in which a disgruntled staff of an organization configured his endpoint with the organization’s CEO’s IP address and used that to send offensive messages – giving the impression that such messages were sent by the CEO. Investigation of network data showed that the CEO’s address was temporarily set with a diffract MAC address from the CEO’s, the MAC address was discovered to be that of the disgruntled staff. Reviewing captured packets on Wireshark reveals both IP address and MAC addresses used to send and receive a packet, closer review can determine if the IP address used is the one that was allocated to a specific endpoint or not by com-

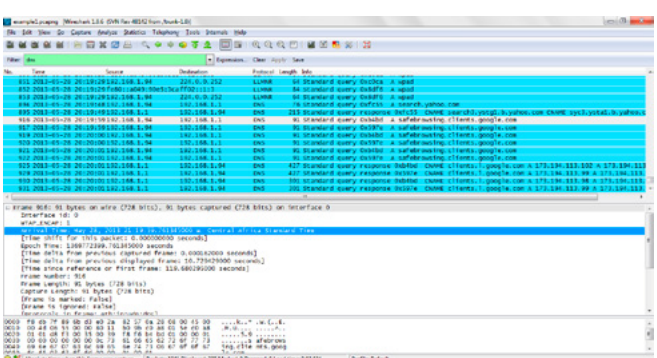


Figure 10. *Expanded frame details in the packet details frame*

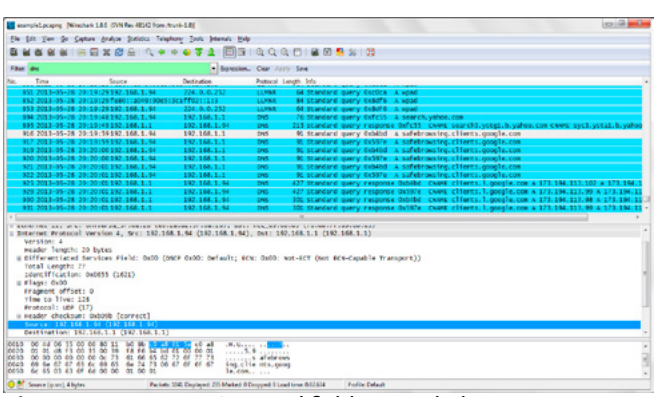


Figure 12. *Internet Protocol field expanded*

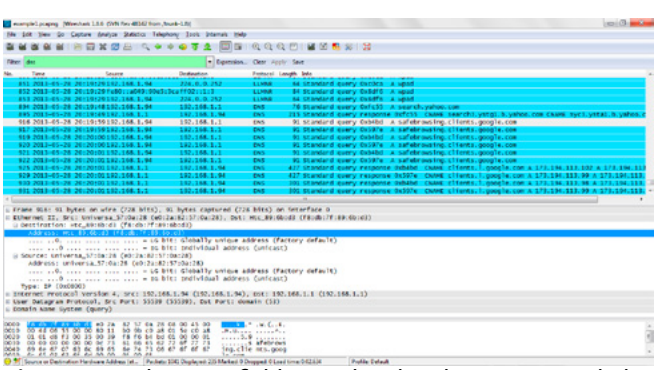


Figure 11. *Ethernet II field in packet details pane expanded*

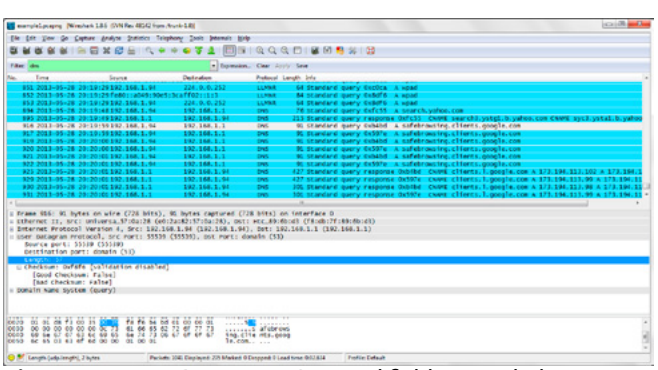


Figure 13. *User Datagram Protocol field expanded*

paring it with the MAC address. That can help to detect an IP spoofing attack.

Under the Domain Name System field, you have the flags and query which are both expandable. The field shows that the packet is a standard query to host address safebrowsing.clients.google.com, with transaction ID 0xb4bd (Figure 14); all which are visible in the information column of the packet list pane. It also shows that the query was responded to in frame number 930. A quick review of items in the packet details field of frame number 930 shows it is a response packet, source and destination from frame 916 are reversed and in the Domain Name Server field, it refers to the request being from frame 916. This confirms that frame 930 is the response to request in frame 916; hence in this case the source is 192.168.1.1, and destination 192.168.1.94.

ANALYZING HTTP PACKETS

Start a Wireshark packet capture and then launch a browser or refresh an already open web page. In this example, an already open Google home page was refreshed. You can stop the packet capture once the web page has loaded. Filter out HTTP packets by entering "http" (in lower case and without quotation marks) into the Filter toolbar and clicking on apply. The first packet after filtering in the packet list as can be observed in Figure 16 shows the packet was captured 14:09 UTC on 5th June, 2013 with frame number 33, the source IP address being 192.168.1.94 and the destination IP address 173.194.41.215. The protocol is of course HTTP, the packet having a length of 571 bytes. The information column describes the packet as GET / HTTP/1.1, meaning it is a request to retrieve HTTP data. The source IP address is known to be the endpoint's IP address, while the destination IP address is for a web site. The destination web site can be figured out in the packet details pane, and using an IP address translator (IP address translators are available online and can be gotten using a search engine). 173.194.41.215 is an IP address for google.com, which is gotten using an IP address translator, and will be seen in the packet details pane.

A quick look at the packet details pane shows the frame number is 33 and that the packet contains 571 bytes of data; Internet Protocol was Version 4; and TCP source port was 60829 and destination port 80 (which is the default HTTP port). An expansion of the Hypertext transfer Protocol field reveals the language for the packet is US English (en-US); the packet is compatible with Mozilla 5.0, Microsoft Internet Explorer 10.0; and the website *www.google.com.ng*. The next packet in the example is frame number 67. It is quite similar to the previous frame number 33, save it is a GET image request. Hence, its description in the packet list pane is GET /images/srpr/logo4w.png HTTP/1.1 in the Hypertext Transfer Protocol field, details show that it accepts images in PNG.

In the packet that follows (frame number 68), in the Line-based text data under the Hypertext

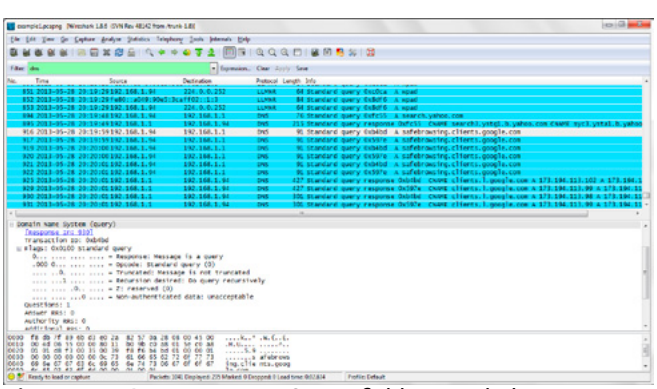


Figure 14. *Domain Name Server field expanded*

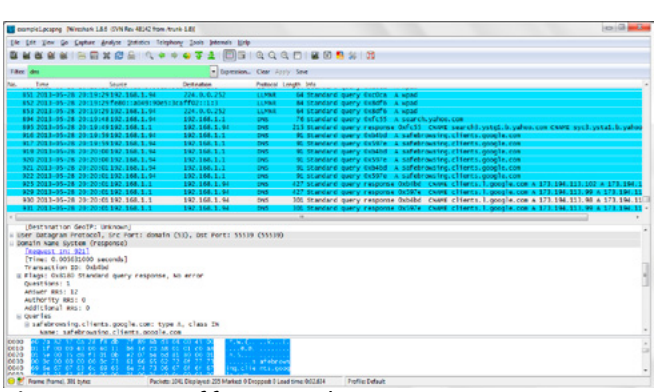


Figure 15. *View of frame number 930*

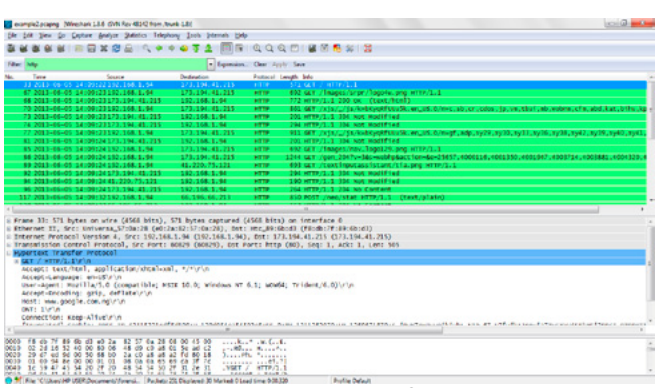


Figure 16. *Packet capture with HTTP filtered*

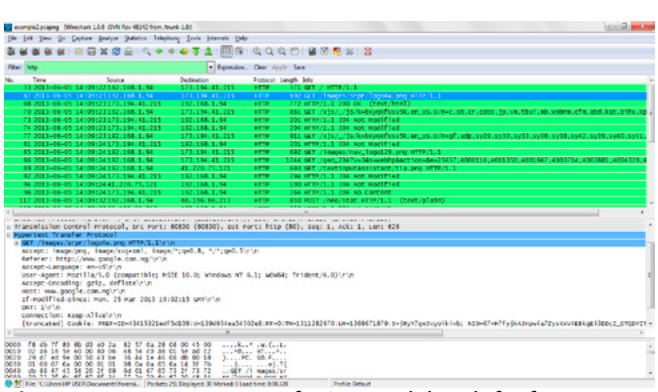


Figure 17. *Hypertext Transfer Protocol details for frame number 67*

Transfer Protocol field, the HTML script for the packet is displayed as shown in figure 18. The script can be used to reconstruct the web page.

Knowing the HTTP data can help identify which websites were visited and what was downloaded; this can help in tracing the source of a problem like malware or a slow network. Identifying pornographic websites or free download/torrent sites on the network can show the problem resulted from visiting such websites, which can further be traced to an endpoint. In the event a user accesses an unauthorized website using a browser in private browsing mode (in a bid to cover his/her tracks), Wireshark can be used to analyze network logs to identify the breech – the unauthorized website's HTTP data will be on the network logs, and the IP address of the endpoint that accessed such a site can be revealed. That is possible because the use of private browsing mode only removes traces from the browser, but cannot affect the network log. The packet sniffer can be used to reveal what sites were visited by a suspect which could be smoking gun evidence; it could, for example, provide breakthrough evidence in a child pornography case where it is discovered that a suspect visited site related to child pornography.

ANALYZING ETHERNET AND ARP DATA

The Address Resolution Protocol (ARP) is used to get the MAC address of a specific IP address. For example, when an endpoint is sending a packet to a destination host, it only has the destination IP address; hence as it sends the packet, it asks which

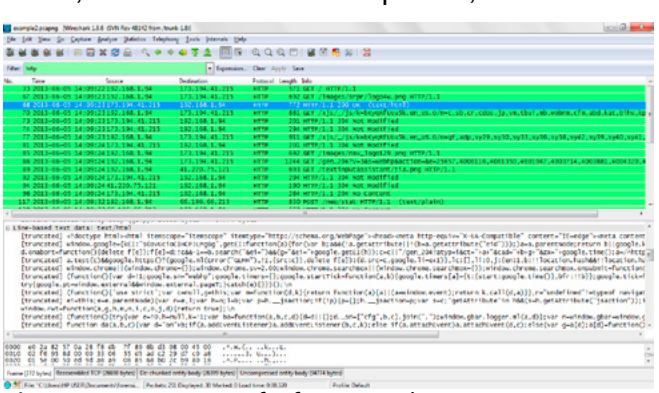


Figure 18. *HTML script for frame number 68*

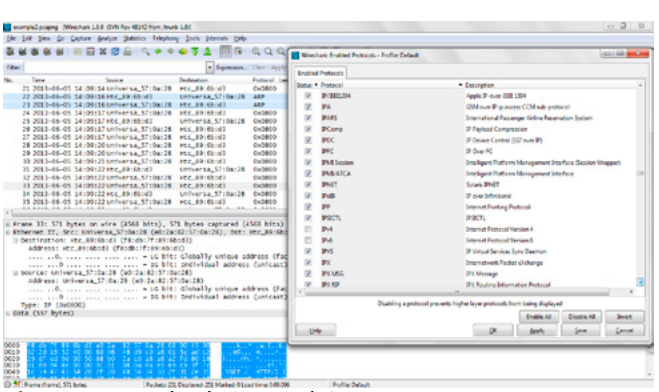


Figure 19. *Uncheck IPv4 and IPv6*

host has the IP address, and the response form the destination states the MAC address. *Dynamic Host Configuration Protocol* (DHCP), on the other hand, uses MAC addresses to assign IP addresses to endpoints that are authorized on a network. Where DHCP is used, an unauthorized MAC address will not get an IP address assigned to it automatically (Casey, 2004). DHCP logs can be used to retrieve a MAC address that was assigned a specific IP address within a particular time frame, this can determine which endpoint was used to carry out a specific action based on packets captured.

In order to analyze Ethernet and ARP data, IP protocols view may be disabled. This is done by clicking Analyze in the File command bar, then clicking on Enabled Protocols and unchecking IP Version 4 and 6 (shown in Figure 19) which results in the look of the interface changing.

We can refer back to packet with frame number 67. In this interface, source and destination addresses are not stated in IP (as IP protocols have been disabled for the view); rather they are indicated by name and hexadecimal – that is source: `Universa_57:0a:28`, and destination: `Htc_89:6b:d3` – in the packet list pane. Within the packet details pane, under the Frame field the date and time of packet arrival is stated based on the time zone of the endpoint (June 5 2013, 15:09 West/Central African Time). The frame number and length are also available under the Frame field. Figure 20 depicts the Frame filed.

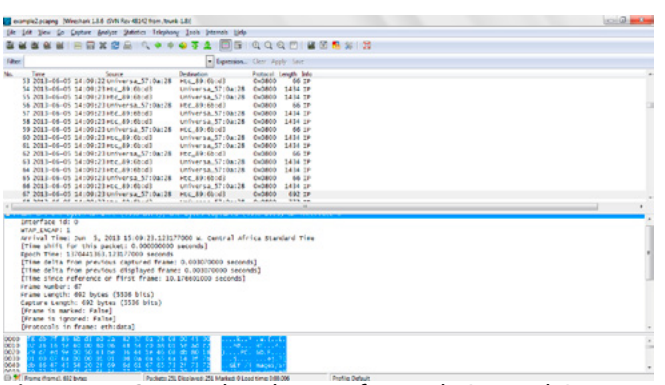


Figure 20. *Captured packet interface with IPv4 and IPv6 disabled and packet frame 67 selected*

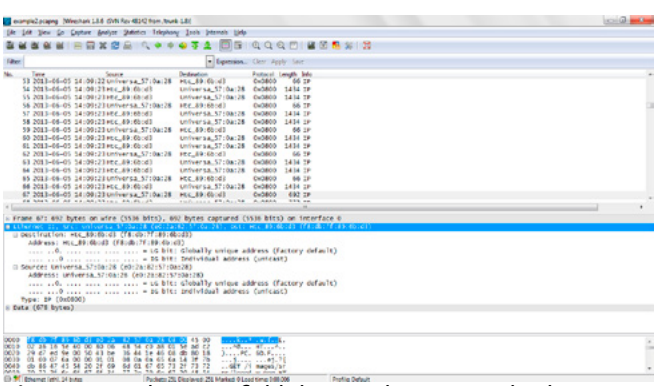


Figure 21. *Ethernet II field showing bytes in packet bytes pane*

Listing 1. *Print out of packet frame 7*

```
No.   Time              Source      Destination Protocol Length Info
67 2013-06-05 14:09:23.123177000 Universa_57:0a:28 Htc_89:6b:d3 0x0800 692       IP
Frame 67: 692 bytes on wire (5536 bits), 692 bytes captured (5536 bits) on interface 0
Ethernet II, Src: Universa_57:0a:28 (e0:2a:82:57:0a:28), Dst: Htc_89:6b:d3 (f8:db:7f:89:6b:d3)
Destination: Htc_89:6b:d3 (f8:db:7f:89:6b:d3)
Address: Htc_89:6b:d3 (f8:db:7f:89:6b:d3)
.... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
.... ...0 .... .... .... .... = IG bit: Individual address (unicast)
Source: Universa_57:0a:28 (e0:2a:82:57:0a:28)
Address: Universa_57:0a:28 (e0:2a:82:57:0a:28)
.... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
.... ...0 .... .... .... .... = IG bit: Individual address (unicast)
Type: IP (0x0800)
Data (678 bytes)
0000 45 00 02 a6 16 5e 40 00 80 06 48 54 c0 a8 01 5e E....^@...HT...^
0010 ad c2 29 d7 ed 9e 00 50 43 be 36 44 1e 46 08 db ..)....PC.6D.F..
0020 80 18 01 00 07 6a 00 00 01 01 08 0a 0a 65 6a 14 .....j.......ej.
0030 3f 7b db 86 47 45 54 20 2f 69 6d 61 67 65 73 2f ?{..GET /images/
0040 73 72 70 72 2f 6c 6f 67 6f 34 77 2e 70 6e 67 20 srpr/logo4w.png
0050 48 54 54 50 2f 31 2e 31 0d 0a 41 63 63 65 70 74 HTTP/1.1..Accept
0060 3a 20 69 6d 61 67 65 2f 70 6e 67 2c 20 69 6d 61 : image/png, ima
0070 67 65 2f 73 76 67 2b 78 6d 6c 2c 20 69 6d 61 67 ge/svg+xml, imag
0080 65 2f 2a 3b 71 3d 30 2e 38 2c 20 2a 2f 2a 3b 71 e/*;q=0.8, */*;q
0090 3d 30 2e 35 0d 0a 52 65 66 65 72 65 72 3a 20 68 =0.5..Referer: h
00a0 74 74 70 3a 2f 2f 77 77 77 2e 67 6f 6f 67 6c 65 ttp://www.google
00b0 2e 63 6f 6d 2e 6e 67 2f 0d 0a 41 63 63 65 70 74 .com.ng/..Accept
00c0 2d 4c 61 6e 67 75 61 67 65 3a 20 65 6e 2d 55 53 -Language: en-US
00d0 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a 20 4d 6f ..User-Agent: Mo
00e0 7a 69 6c 6c 61 2f 35 2e 30 20 28 63 6f 6d 70 61 zilla/5.0 (compa
00f0 74 69 62 6c 65 3b 20 4d 53 49 45 20 31 30 2e 30 tible; MSIE 10.0
0100 3b 20 57 69 6e 64 6f 77 73 20 4e 54 20 36 2e 31 ; Windows NT 6.1
0110 3b 20 57 4f 57 36 34 3b 20 54 72 69 64 65 6e 74 ; WOW64; Trident
0120 2f 36 2e 30 29 0d 0a 41 63 63 65 70 74 2d 45 6e /6.0)..Accept-En
0130 63 6f 64 69 6e 67 3a 20 67 7a 69 70 2c 20 64 65 coding: gzip, de
0140 66 6c 61 74 65 0d 0a 48 6f 73 74 3a 20 77 77 77 flate..Host: www
0150 2e 67 6f 6f 67 6c 65 2e 63 6f 6d 2e 6e 67 0d 0a .google.com.ng..
0160 49 66 2d 4d 6f 64 69 66 69 65 64 2d 53 69 6e 63 If-Modified-Sinc
0170 65 3a 20 4d 6f 6e 2c 20 32 35 20 4d 61 72 20 32 e: Mon, 25 Mar 2
0180 30 31 33 20 31 39 3a 30 32 3a 31 35 20 47 4d 54 013 19:02:15 GMT
0190 0d 0a 44 4e 54 3a 20 31 0d 0a 43 6f 6e 6e 65 63 ..DNT: 1..Connec
01a0 74 69 6f 6e 3a 20 4b 65 65 70 2d 41 6c 69 76 65 tion: Keep-Alive
01b0 0d 0a 43 6f 6f 6b 69 65 3a 20 50 52 45 46 3d 49 ..Cookie: PREF=I
01c0 44 3d 34 33 34 31 35 33 32 31 65 64 66 35 64 31 D=43415321edf5d1
01d0 30 39 3a 55 3d 31 33 39 64 39 35 34 65 61 35 34 09:U=139d954ea54
01e0 35 30 32 65 38 3a 46 46 3d 30 3a 54 4d 3d 31 33 502e8:FF=0:TM=13
01f0 31 31 32 38 32 39 37 30 3a 4c 4d 3d 31 33 36 39 11282970:LM=1369
0200 36 37 31 38 37 39 3a 53 3d 6a 52 79 59 37 67 77 671879:S=jRyY7gw
0210 4a 76 79 6f 69 6b 69 76 62 3b 20 4e 49 44 3d 36 Jvyoikivb; NID=6
0220 37 3d 48 37 66 79 6a 6b 41 4a 6e 70 77 69 61 37 7=H7fyjkAJnpwia7
0230 5a 79 73 58 78 56 34 45 42 6b 67 45 69 33 44 44 ZysXxV4EBkgEi3DD
0240 63 49 5f 4f 54 51 44 59 49 54 69 65 51 48 79 34 cI_OTQDYITieQHy4
0250 4d 7a 53 71 43 57 35 57 47 74 68 67 58 71 6e 53 MzSqCW5WGthgXqnS
0260 38 69 6b 65 41 64 70 70 7a 33 53 77 47 39 34 43 8ikeAdppz3SwG94C
0270 73 6b 6d 51 66 47 6c 47 68 35 76 78 4a 79 53 58 skmQfGlGh5vxJySX
0280 34 63 6f 55 70 72 57 45 70 6d 2d 51 61 35 37 2d 4coUprWEpm-Qa57-
0290 35 54 69 64 39 74 73 74 64 78 48 41 48 59 4d 4d 5Tid9tstdxHAHYMM
02a0 70 78 0d 0a 0d 0a px....
```

In the Ethernet II field, the destination and source addresses are stated with the full 48-bit Ethernet addresses in brackets. The hexadecimal figures for both are visible in the packet bytes pane when the field is selected (Figure 21), and individually when either destination or source is selected.

Details of the packet frame 67 can be printed as shown in Listing 1, summarizing the captured packet. Clicking File and then print brings the dialog box in Figure 22; checking "Selected packet only" radio button ensures only the packet is printed.

Packets involving the *Address Resolution Protocol* (ARP) can be filtered out using the Filter bar. In the example, two such packets are found – ARP packets are normally a request and a reply as is observed in the example. Figure 23 shows the ARP request packet frame number 22, under the ARP field the packet type is in brackets as "request". The packet list pane shows that the source of the packet is the wireless access point with MAC address `Htc_89:6b:d3`, and the destination `Universa_57:0a:28;` and has a length of 42 bytes. The "Info" column has a question "who has 192.168.1.94? Tell 192.168.1.1". That is the ARP trying to resolve the end point's address 192.168.1.94 for the wireless access point 192.168.1.1. In the packet details pane, the target MAC address is stated as `00:00:00_00:00:00` as the address is not yet resolved; hence stated as unknown.

Frame 23 which is the ARP reply packet shows the response to the request in frame 24. The packet list pane shows the source and destination as the reverse of frame 24 as this packet is a response from the end point to the wireless access point; the length is the same 42 bytes. The "Info" column answers the question posed in the previous packet stating: `192.168.1.94 is at e0:2a:82:57:0a:28` – that is 192.168.1.94 belongs to the MAC address of the endpoint (in hexadecimal notation). Hence; under the Address Resolution Protocol field in the packet details pane, the source MAC address of the previous request is now identified and stated as the sender (`Univers_57:0a:28`), along with IP address details. A print out of the packet can also be done as was done for frame number 67 to yield a similar output.

ARP data can identify the endpoints and network interfaces that interacted over a network; this can be used to identify an unauthorized connection within the network. For example, a war-driver on a wireless network can be identified from ARP traffic analysis.

Figure 25 shows a DHCP (DHCPv6) packet filtered out. The Internet Protocol used is version 6, hence the IP addresses are shown in hexadecimal format (Internet Protocol field). The UDP field shows the source port to be 546, which is the DHCPv6 client port; and the destination port 547, the DHCPv6 server port. Under the DHCPv6 field, we can see the client identifier.

SUMMARY

Network forensics is a very important field in the information age. It can be used to monitor users and devices and to track network breaches, troubleshoot and improve network security and performance. It can also be used to track and indict of-

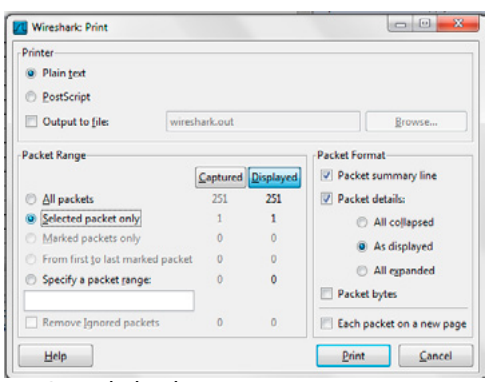
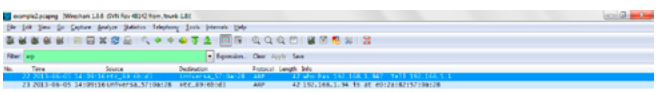


Figure 22. *Print dialog box*



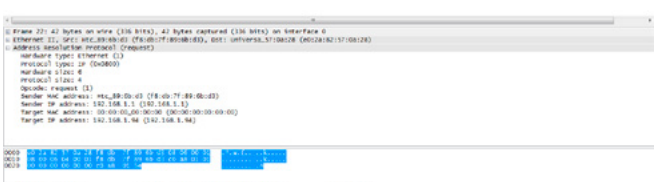


Figure 23. *ARP request packet*

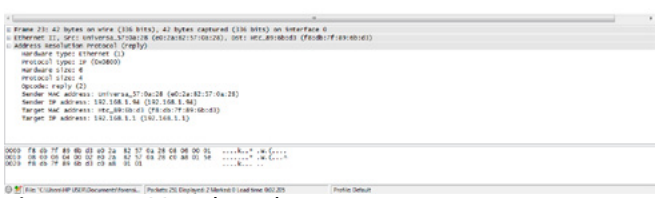
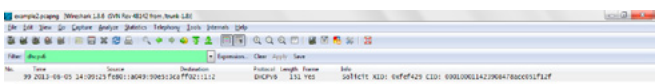


Figure 24. *ARP reply packet*



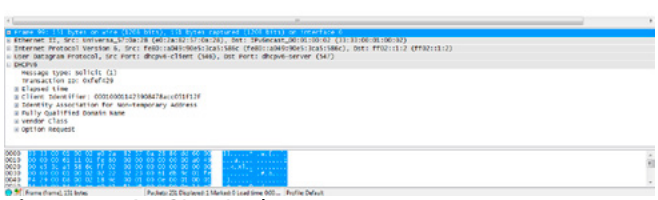


Figure 25. *DHCPv6 Packet*

REFERENCES

- Casey, E. (2004) Digital Evidence and Computer Crime: Forensic Science, Computers and the Internet. 2nd ed. Elsevier Academic press.
- Kurose, J.F. and Ross, K.W.. (2009) Wireshark Lab: Getting Started [Online]. Available from: *http://wps.aw.com/wps/media/objects/7134/7305312/WireShark_Labs/Wireshark_INTRO_Sept_15_2009.pdf* (Downloaded: 16 March 2010).
- Kurose, J.F. and Ross, K.W.. (2009) Wireshark Lab: HTTP [Online]. Available from: *http://wps.aw.com/wps/media/objects/7134/7305312/WireShark_Labs/Wireshark_HTTP_Sept_15_2009.pdf* (Downloaded: 16 March 2010).
- Kurose, J.F. and Ross, K.W.. (2009) Wireshark Lab: Ethernet and ARP [Online]. Available from: *http://wps.aw.com/wps/media/objects/7134/7305312/WireShark_Labs/Wireshark_Ethernet_ARP_Sept_15_2009.pdf* (Downloaded: 13 April 2010).
- Lamping, U., Sharpe, R. and Warnicke, E. (2013) Wireshark User's Guide for Wireshark 1.11 [Online]. Available from: *http://www.wireshark.org/download/docs/user-guide-a4.pdf* (Downloaded: 17 May 2013).
- SYBEX Inc. (1998) Using Port Numbers and Protocols [Online]. Available from: *http://msdn.microsoft.com/en-us/library/aa227632(v=vs.60).aspx* (Accessed: 17 June 2013).
- Touch, J. et al (2013) Service Name and Transport Port Number Registry [Online]. Available from: *http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xml* (Accessed: 17 June 2013).
- Wildpackets (2013) Four Ways Network Forensics Can Help You [Online]. Available from: *http://blog.wildpackets.com/2013/06/06/four-ways-network-forensics-can-help-you.html?goback=.gde_80784_member_247550610* (Accessed: 6 June 2013).

fenders. Wireshark is one of the tools for network forensics which is freely available.

Wireshark has a wide range of uses and interfaces which were not covered in this article, the article merely introduced the basics. Wireshark among other things can also be used to monitor and analyze mobile traffic and VoIP traffic. Packet sniffers come in very handy when analysis of network based evidence is required.

About the Author

Dauda Sule, CISA. He is currently the Marketing Manager of Audit Associates Limited which is a consultancy firm that specializes in designing and organizing training programs pertaining to auditing, fraud detection and prevention, information security and assurance, and anti-money laundering. He is a CISA and has a M.Sc. in Computer Security from the University of Liverpool. Dauda also has a first degree black belt in Taekwondo. He has previous experience of over five years in the Nigerian Banking industry, and also did some time in Gtech Computers (a computer and allied services company) as a systems security and assurance supervisor.