Capturing Network Traffic With Wireshark 2

A White Paper From

GOLDSTAR
SOFTWARE
www.GoldstarSoftware.com

For more information, see our web site at
http://www.goldstarsoftware.com

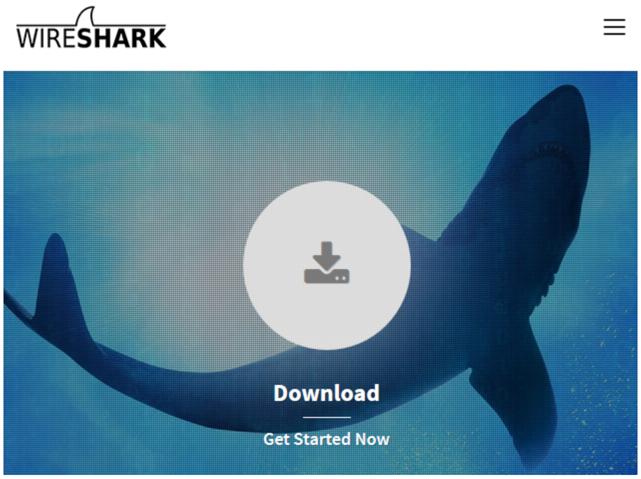
Capturing Network Traffic with Wireshark 2

Last Updated: 08/06/2018

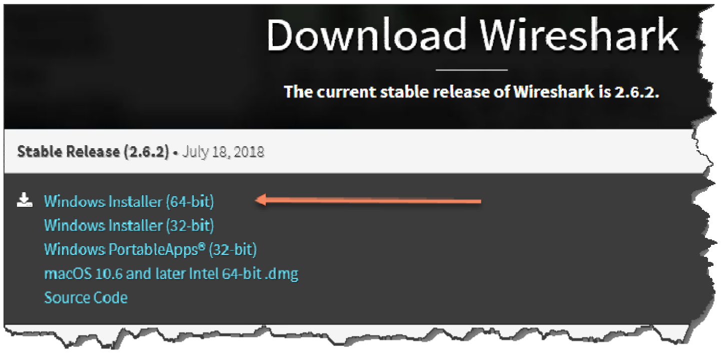
In some cases, the easiest way to identify problems in a network system, such as performance problems or system failures, is to grab a network capture -- a log of all networking packets that enter and leave a workstation. When typical "hit or miss" troubleshooting doesn't seem to be working, we may instruct you to collect a network trace, and these instructions serve as a simple way to perform this task.

Downloading and Installing Wireshark 2

The first step is to download Wireshark. Go to www.wireshark.org and click on the Download button:



Then select the Windows Installer link from the next dialog.



Download Wireshark

The current stable release of Wireshark is 2.6.2.

Stable Release (2.6.2) • July 18, 2018

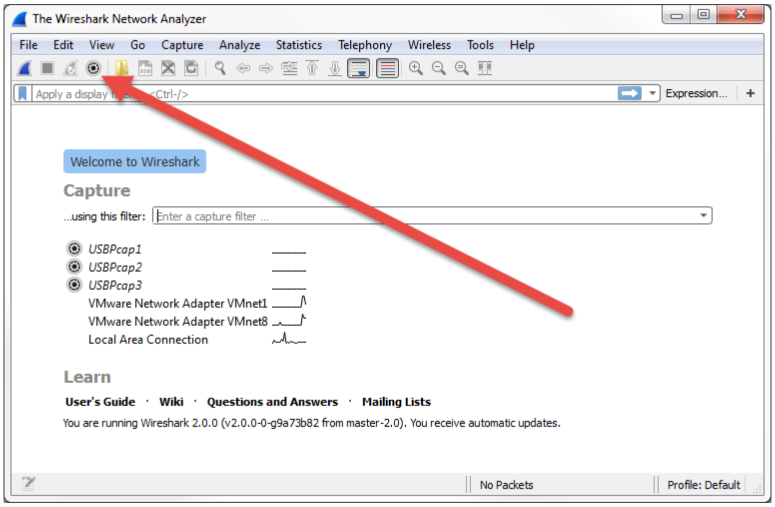
- Windows Installer (64-bit)
- Windows Installer (32-bit)
- Windows PortableApps® (32-bit)
- macOS 10.6 and later Intel 64-bit .dmg
- Source Code

If you are running on a 64-bit operating system, then you want the 64-bit version. Otherwise, download the 32-bit release.

When the download finishes, run it with all of the default options (click next, next, etc.) to install the software. When it is done installing, launch Wireshark.

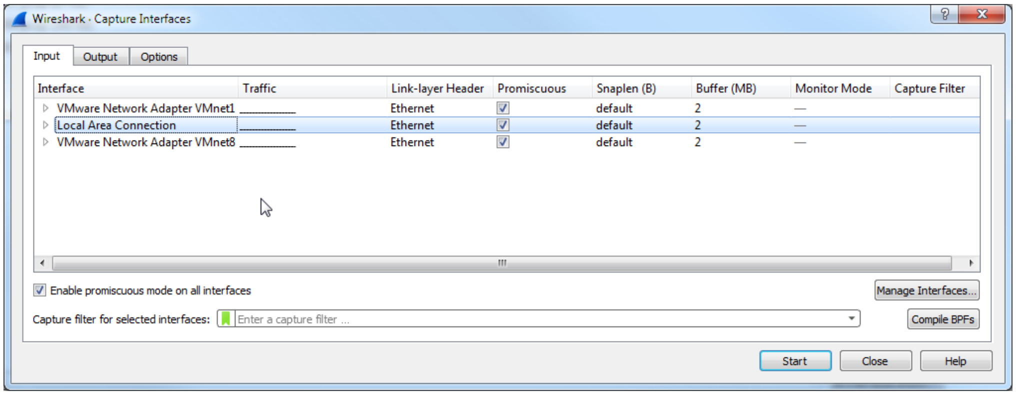
Starting Wireshark and Setting up a Simple Capture

A simple capture is used when you can easily duplicate your network problem, like starting an application. When Wireshark launches, you will see a standard welcome screen, which looks something like this:



You are running Wireshark 2.0.0 (v2.0.0-0-g9a73b82 from master-2.0). You receive automatic updates.

Click on the **Capture Options** button, as indicated above, and you will see the **Capture Options** dialog box.



Interface	Traffic	Link-layer Header	Promiscuous	Snaplen (B)	Buffer (MB)	Monitor Mode	Capture Filter
VMware Network Adapter VMnet1		Ethernet	☑	default	2	—	
Local Area Connection		Ethernet	☑	default	2	—	
VMware Network Adapter VMnet8		Ethernet	☑	default	2	—	

☑ Enable promiscuous mode on all interfaces

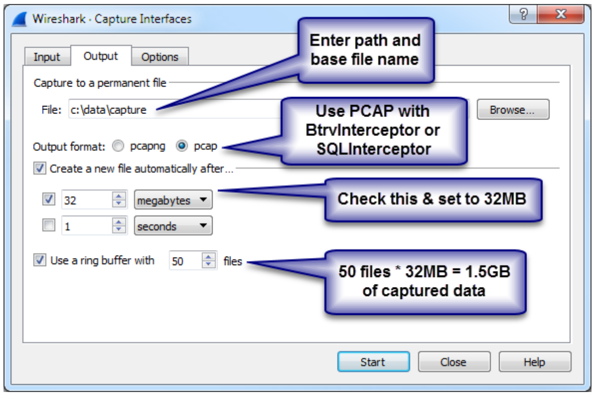
Capture filter for selected interfaces: Enter a capture filter ...

Verify that the correct network card is selected, or your capture will likely be empty, and click Start to start capturing data.

Setting up a Ring Buffer Capture

For some issues, you may need to capture a LOT of data, or you may not know when the error will occur. Wireshark handles smaller capture files very well, but when your files get TOO large, the system starts to get sluggish. To avoid this, we recommend setting up a "ring buffer" capture, which grabs the network data in a number of smaller, more manageable, capture files, and automatically deletes the oldest data.

Select the **Output** tab.



Wireshark · Capture Interfaces

Capture to a permanent file

File: c:\data\capture

Output format: ○ pcapng ● pcap

☑ Create a new file automatically after...

☑ 32 megabytes

□ 1 seconds

☑ Use a ring buffer with 50 files

Enter path and base file name

Use PCAP with BtrvInterceptor or SQLInterceptor

Check this & set to 32MB

50 files * 32MB = 1.5GB of captured data

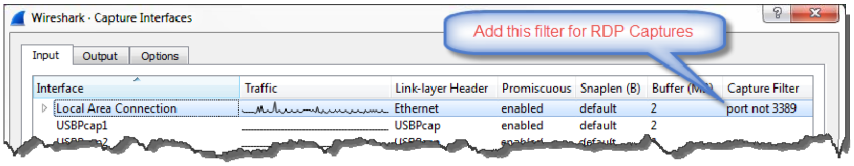
Enter a path and base filename for the capture. This location should have enough disk space for the total size of the capture -- so you will want to verify this FIRST. You can use either output format, but if we will be analyzing database traffic, use PCAP to make the process easier and avoid an extra translation step.

Then, check the **Create a new file automatically after…** option, and set the limiter to 32MB. Also check the **Use a ring buffer with** checkbox and specify the number of files that you want to keep. Note that 50 files of 32MB each = 50*32 = 1.6GB. In many cases, even this is too much, and you might be able to get away with 10 or 20 files. When in doubt, more data is better, and it gives you more time to detect the problem and stop the trace before data is overwritten. This is a good time to re-verify that you have enough disk space in the location provided, because if you run out of disk space, bad things can happen!

Capturing from a Remote Desktop Session

When you are remotely controlling a machine via Remote Desktop, the network capture will see all of your RDP packets, as well as the rest of the data that you want. Obviously, this can change the environment enough such that you don't get a valid capture, so we recommend capturing ONLY from the console itself.

However, if you must use an RDP session, then you want to make one more change to the Capture Options dialog box:



Wireshark · Capture Interfaces

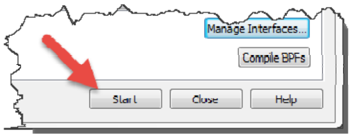
Interface	Traffic	Link-layer Header	Promiscuous	Snaplen (B)	Buffer (MB)	Capture Filter
Local Area Connection		Ethernet	enabled	default	2	port not 3389
USBPcap1		USBPcap	enabled	default	2	

Add this filter for RDP Captures

By adding "port not 3389" to the capture filter, you will exclude the RDP traffic, and get a better handle on exactly what you ARE looking for. Again, if RDP is contributing to the problem, then this filter may actually help disguise the problems you are trying to troubleshoot.

Starting and Stopping the Capture

When you have set up the capture buffer as you need it, click **Start** to begin the capture.



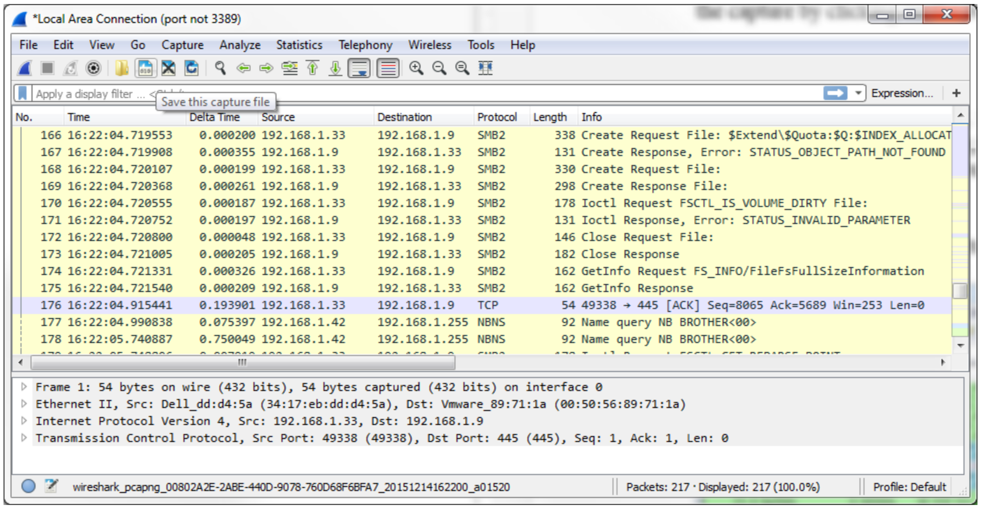
If you do NOT see any packets, then you may have selected the wrong interface. Go back to the **Capture Options** screen and try a different network card.

You should now use your system normally and duplicate the problem that you are trying to troubleshoot. If possible, note the EXACT time that the problem occurs. Then, stop the capture by clicking on the **Stop Capturing Packets** button (🟥) on the toolbar.

Saving the Capture

If you created a circular buffer, then your packet capture is already saved into one or more files in the location you specified. Each filename includes a timestamp of when the file was CREATED, so that you can isolate the file which contains the "interesting" packets.

If you were using a simple capture, then you'll instead see a screen like this:

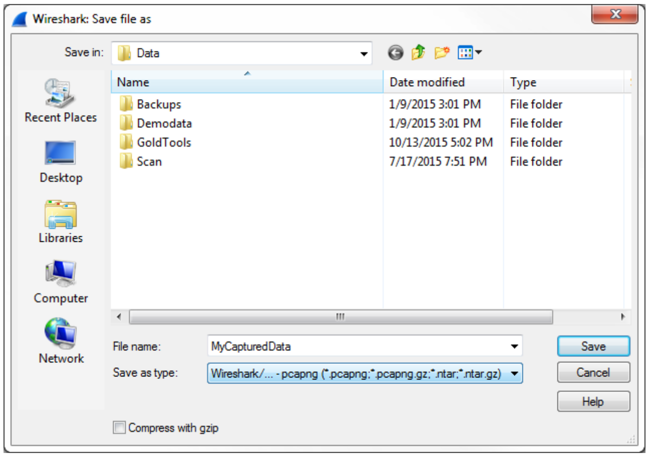


```
No.  Time            Delta Time Source        Destination   Protocol Length Info
 166 16:22:04.719553  0.000200 192.168.1.33  192.168.1.9   SMB2      338 Create Request File: $Extend\$Quota:$Q:$INDEX_ALLOCAT
 167 16:22:04.719908  0.000355 192.168.1.9   192.168.1.33  SMB2      131 Create Response, Error: STATUS_OBJECT_PATH_NOT_FOUND
 168 16:22:04.720107  0.000199 192.168.1.33  192.168.1.9   SMB2      330 Create Request File:
 169 16:22:04.720368  0.000261 192.168.1.9   192.168.1.33  SMB2      298 Create Response File:
 170 16:22:04.720555  0.000187 192.168.1.33  192.168.1.9   SMB2      178 Ioctl Request FSCTL_IS_VOLUME_DIRTY File:
 171 16:22:04.720752  0.000197 192.168.1.9   192.168.1.33  SMB2      131 Ioctl Response, Error: STATUS_INVALID_PARAMETER
 172 16:22:04.720800  0.000048 192.168.1.33  192.168.1.9   SMB2      146 Close Request File:
 173 16:22:04.721005  0.000205 192.168.1.9   192.168.1.33  SMB2      182 Close Response
 174 16:22:04.721331  0.000326 192.168.1.33  192.168.1.9   SMB2      162 GetInfo Request FS_INFO/FileFsFullSizeInformation
 175 16:22:04.721540  0.000209 192.168.1.9   192.168.1.33  SMB2      162 GetInfo Response
 176 16:22:04.915441  0.193901 192.168.1.33  192.168.1.9   TCP        54 49338 → 445 [ACK] Seq=8065 Ack=5689 Win=253 Len=0
 177 16:22:04.990838  0.075397 192.168.1.42  192.168.1.255 NBNS       92 Name query NB BROTHER<00>
 178 16:22:05.740887  0.750049 192.168.1.42  192.168.1.255 NBNS       92 Name query NB BROTHER<00>

▷ Frame 1: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0
▷ Ethernet II, Src: Dell_dd:d4:5a (34:17:eb:dd:d4:5a), Dst: Vmware_89:71:1a (00:50:56:89:71:1a)
▷ Internet Protocol Version 4, Src: 192.168.1.33, Dst: 192.168.1.9
▷ Transmission Control Protocol, Src Port: 49338 (49338), Dst Port: 445 (445), Seq: 1, Ack: 1, Len: 0

wireshark_pcapng_00802A2E-2ABE-440D-9078-760D68F6BFA7_20151214162200_a01520 | Packets: 217 · Displayed: 217 (100.0%) | Profile: Default
```

Click the **Save** button to bring up the **Save File As** dialog box.



Wireshark: Save file as

Save in: Data

Name	Date modified	Type
Backups	1/9/2015 3:01 PM	File folder
Demodata	1/9/2015 3:01 PM	File folder
GoldTools	10/13/2015 5:02 PM	File folder
Scan	7/17/2015 7:51 PM	File folder

File name: MyCapturedData

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

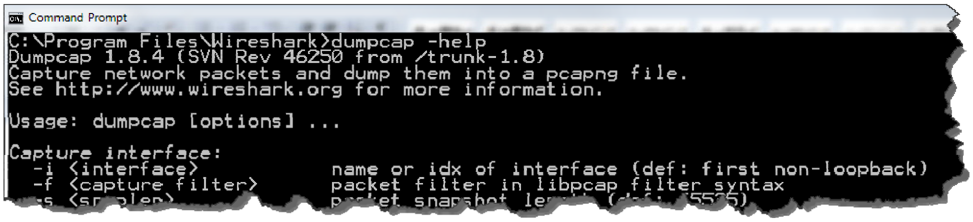
□ Compress with gzip

Enter a filename without an extension and click **Save**. Note that if you are going to use a trace file with SQLInterceptor or BtrvInterceptor, you should also change the **Save As Type** field to PCAP instead of PCAPNG.

You can then submit the saved trace data to Goldstar Software via Email or (for larger data sets) via FTP.

Visual C++ Crashes

In some Wireshark versions, memory allocation seems to be an issue, and it has been causing pop-up dialog boxes that indicate the Visual C++ Library is crashing. If you see this problem, then you can avoid the user interface and use the command-line tool *dumpcap* instead. Use the option “-help” to get information about the command line options:



Command Prompt

```
C:\Program Files\Wireshark>dumpcap -help
Dumpcap 1.8.4 (SVN Rev 46250 from /trunk-1.8)
Capture network packets and dump them into a pcapng file.
See http://www.wireshark.org for more information.

Usage: dumpcap [options] ...

Capture interface:
  -i <interface>           name or idx of interface (def: first non-loopback)
  -f <capture filter>      packet filter in libpcap filter syntax
  -s <snaplen>             packet snapshot length (def: 65535)
```

A command like this should work for most users:

```
dumpcap -i 1 -b filesize:32768 -b files:100 -f "port not 3389" -w C:\Cap.pcap
```

If you still can't get it to work, contact Goldstar Software and let us work with you to help! Please note that this may be a billable support call if you have already used up your free support time.