

27.2.10 Lab - Extract an Executable from a PCAP



This lab has been updated for use on NETLAB+.
www.netdevgroup.com

Objectives

Part 1: Analyze Pre-Captured Logs and Traffic Captures

Part 2: Extract Downloaded Files from PCAP

Background / Scenario

Looking at logs is very important, but it is also important to understand how network transactions happen at the packet level.

In this lab, you will analyze the traffic in a previously captured pcap file and extract an executable from the file.

Instructions

Part 1: Analyze Pre-Captured Logs and Traffic Captures

In Part 2, you will work with the *nimda.download.pcap* file. Captured in a previous lab, *nimda.download.pcap* contains the packets related to the download of the *Nimda* malware. Your version of the file, if you created it in the previous lab and did not reimport your **Workstation** VM, is stored in the */home/analyst* directory. However, a copy of that file is also stored in the **Workstation** VM, under the */home/analyst/lab.support.files/pcaps* directory so that you can complete this lab. For consistency of output, the lab will use the stored version in the *pcaps* directory.

While *tcpdump* can be used to analyze captured files, *Wireshark*'s graphical interface makes the task much easier. It is also important to note that *tcpdump* and *Wireshark* share the same file format for packet captures; therefore, *PCAP* files created by one tool can be opened by the other.

- Launch and log in to the **Workstation** VM (username: **analyst** / password: **cyberops**).
- Launch a terminal and change directory to the **lab.support.files/pcaps** folder, and get a listing of files using the **ls -l** command.

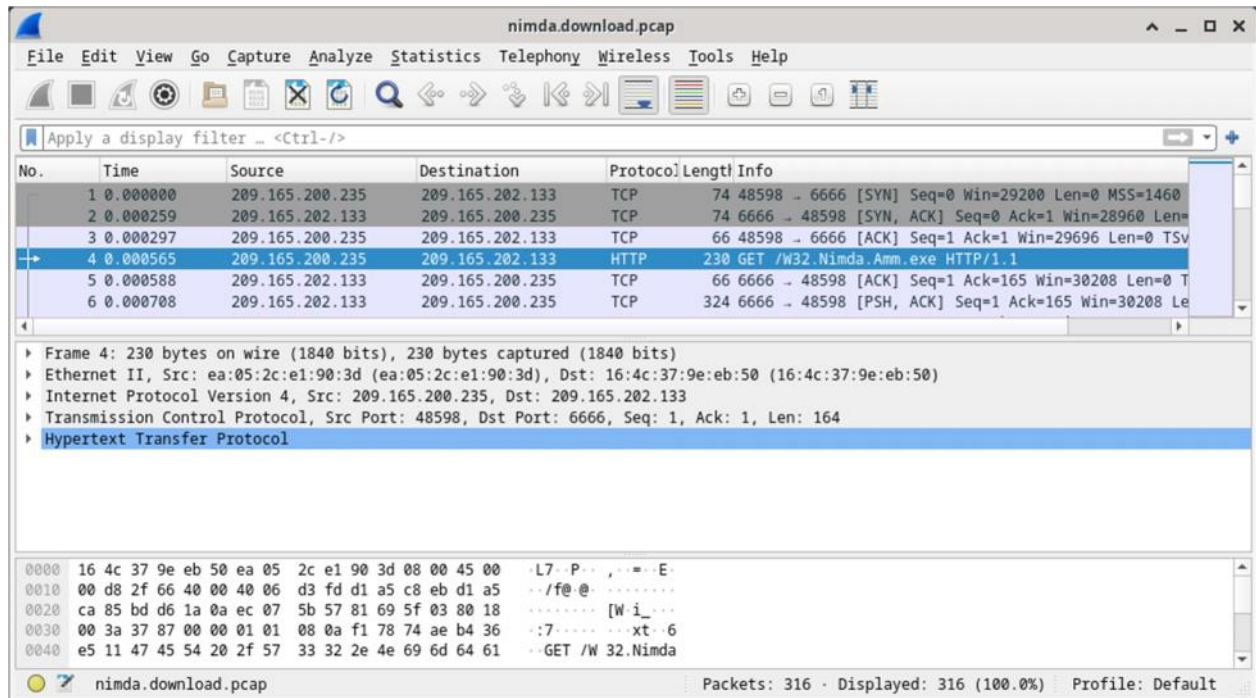
```
[analyst@secOps ~]$ cd lab.support.files/pcaps/
[analyst@secOps pcaps]$ ls -l
total 4028
-rw-r--r-- 1 analyst analyst 371462 Mar 21 2018 nimda.download.pcap
-rw-r--r-- 1 analyst analyst 3750153 Mar 21 2018 wannacry_download_pcap.pcap
```

- Issue the command below to open the **nimda.download.pcap** file in Wireshark.

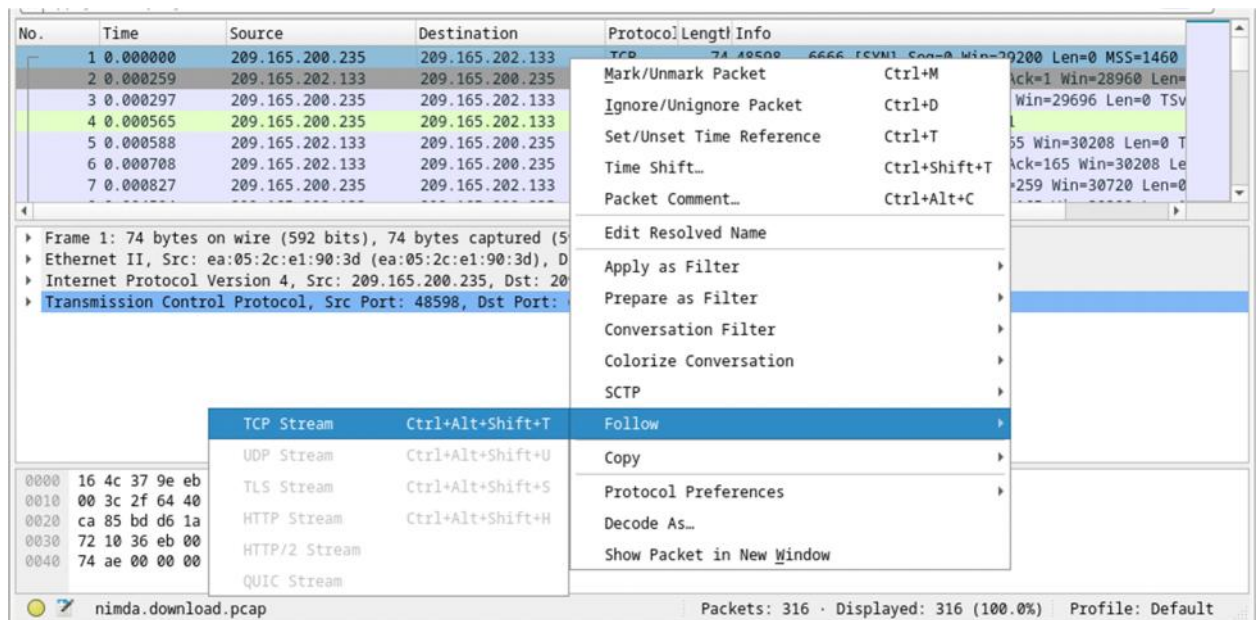
```
[analyst@secOps pcaps]$ wireshark nimda.download.pcap &
```
- The **nimda.download.pcap** file contains the packet capture related to the malware download performed in a previous lab. The **pcap** contains all the packets sent and received while **tcpdump** was running.

27.2.10 Lab - Extract an Executable from a PCAP

Select the fourth packet in the capture and expand the Hypertext Transfer Protocol to display as shown below.

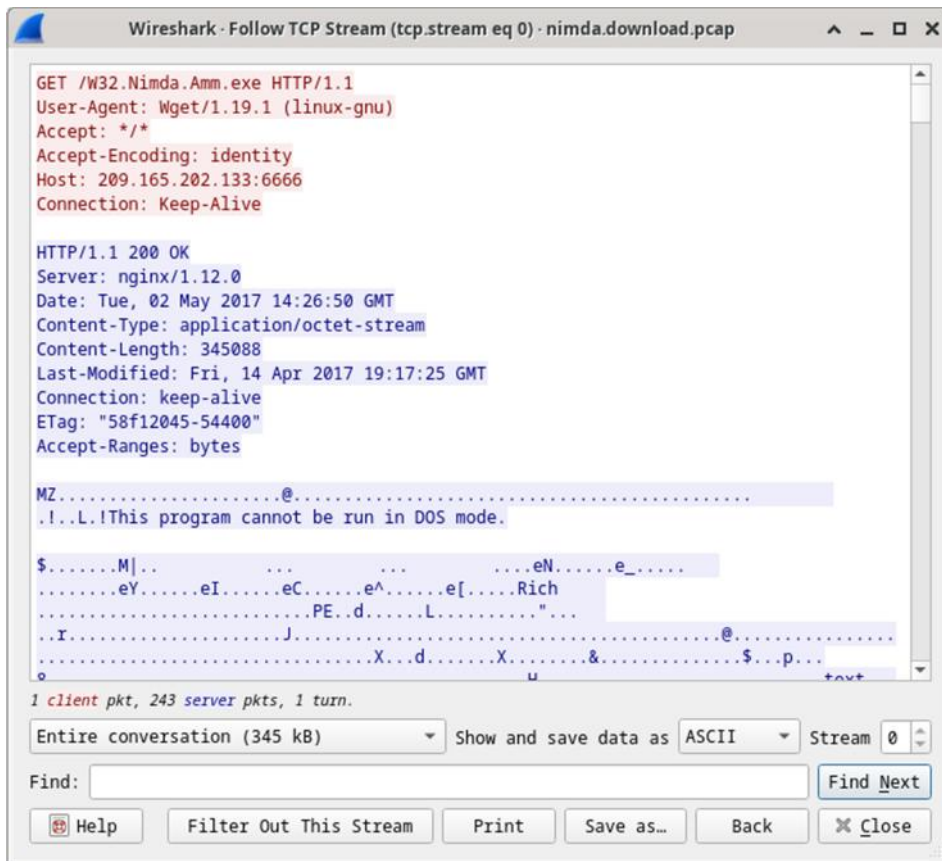


- e. Packets one through three are the *TCP* handshake. The fourth packet shows the request for the malware file. Confirming what was already known, the request was done over *HTTP*, sent as a *GET* request.
- f. Because *HTTP* runs over *TCP*, it is possible to use *Wireshark's Follow TCP Stream* feature to rebuild the *TCP* transaction. Select the first *TCP* packet in the capture, a *SYN* packet. Right-click it and choose **Follow > TCP Stream**.



27.2.10 Lab - Extract an Executable from a PCAP

- g. Wireshark displays another window containing the details for the entire selected TCP flow



What are all those symbols shown in the *Follow TCP Stream* window? Are they connection noise? Data? Explain.

There are a few readable words spread among the symbols. Why are they there?

Challenge Question: Despite the *W32.Nimda.Amm.exe* name, this executable is not the famous worm. For security reasons, this is another executable file that was renamed as *W32.Nimda.Amm.exe*. Using the word fragments displayed by *Wireshark's Follow TCP Stream* window, can you tell what executable this really is?

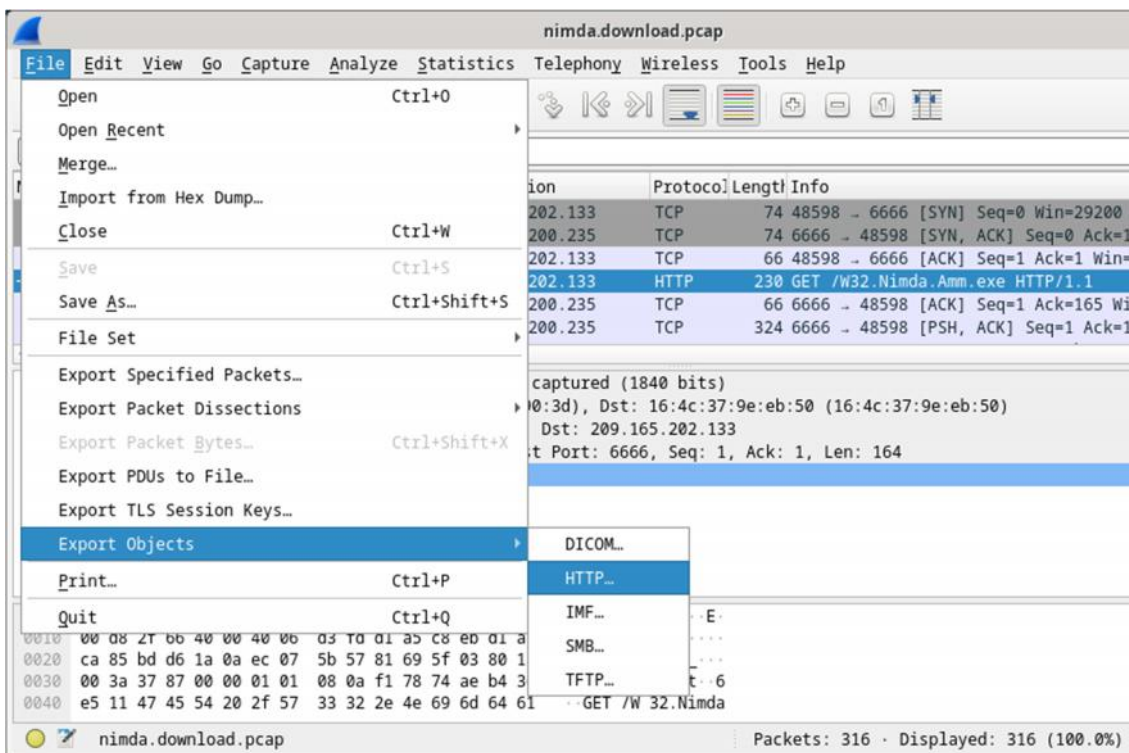
- h. Click **Close** in the Follow TCP Stream window to return to the Wireshark *nimda.download.pcap* file.

Part 2: Extract Downloaded Files from PCAP

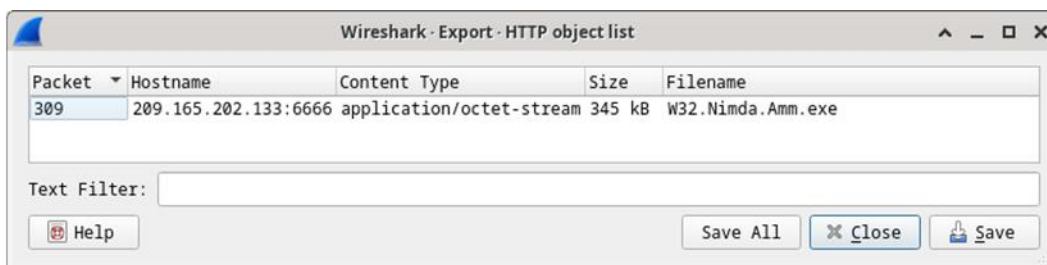
Because capture files contain all packets related to traffic, a *PCAP* of a download can be used to retrieve a previously downloaded file. Follow the steps below to use *Wireshark* to retrieve the *Nimda* malware.

27.2.10 Lab - Extract an Executable from a PCAP

- In that fourth packet in the *nimda.download.pcap* file, notice that the *HTTP GET* request was generated from 209.165.200.235 to 209.165.202.133. The Info column also shows this is in fact the *GET* request for the file.
- With the *GET* request packet selected, navigate to **File > Export Objects > HTTP**, from **Wireshark's** menu.



- Wireshark will display all *HTTP* objects present in the *TCP* flow that contains the *GET* request. In this case, only the *W32.Nimda.Amm.exe* file is present in the capture. It will take a few seconds before the file is displayed.



Why is *W32.Nimda.Amm.exe* the only file in the capture?

- In the **HTTP object list** window, select the **W32.Nimda.Amm.exe** file and click **Save** at the bottom of the screen.
- Click the left arrow until you see the **Home** button. Click **Home** and then click the **analyst** folder (not the analyst tab). Save the file there.

- f. Return to your terminal window and ensure the file was saved. Change directory to the **/home/analyst** folder and list the files in the folder using the **ls -l** command.

```
[analyst@secOps pcaps]$ cd /home/analyst
[analyst@secOps ~]$ ls -l
total 364
drwxr-xr-x 2 analyst analyst 4096 Sep 26 2014 Desktop
drwx----- 3 analyst analyst 4096 May 25 11:16 Downloads
drwxr-xr-x 2 analyst analyst 4096 May 22 08:39 extra
drwxr-xr-x 8 analyst analyst 4096 Jun 22 11:38 lab.support.files
drwxr-xr-x 2 analyst analyst 4096 Mar 3 15:56 second_drive
-rw-r--r-- 1 analyst analyst 345088 Jun 22 15:12 W32.Nimda.Amm.exe
[analyst@secOps ~]$
```

Was the file saved?

- g. The **file** command gives information on the file type. Use the file command to learn a little more about the malware, as show below:

```
[analyst@secOps ~]$ file W32.Nimda.Amm.exe
W32.Nimda.Amm.exe: PE32+ executable (console) x86-64, for MS Windows
[analyst@secOps ~]$
```

As seen above, **W32.Nimda.Amm.exe** is indeed a Windows executable file.

In the malware analysis process, what would be a probable next step for a security analyst?
