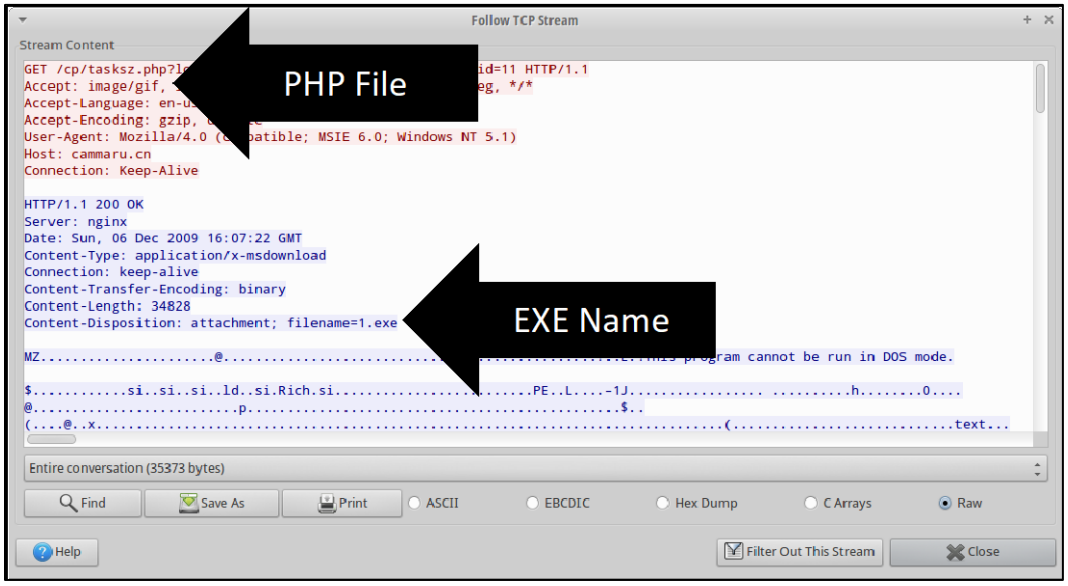
Part 2 – Analysis of 192.168.1.111

This section provides answers and a method of solving each of the lab questions. Note that there may be more than one way to arrive at the correct answer. You may discover a different solution to the same problem.

1. For the first several questions, start by filtering on the victim IP address in Wireshark to only show traffic going to and coming from it (i.e., ip.addr == 192.168.1.111 display filter). One of the first TCP packets shows the victim asking for a PHP file called tasksz.php from 89.248.162.164. This packet may indicate the name of the PHP file and the IP address it was downloaded from. More analysis is necessary to confirm this hypothesis.

2. Further analyze this packet using Wireshark's conversation-following capabilities. Right-click on the suspect packet and select **Follow TCP Stream**.

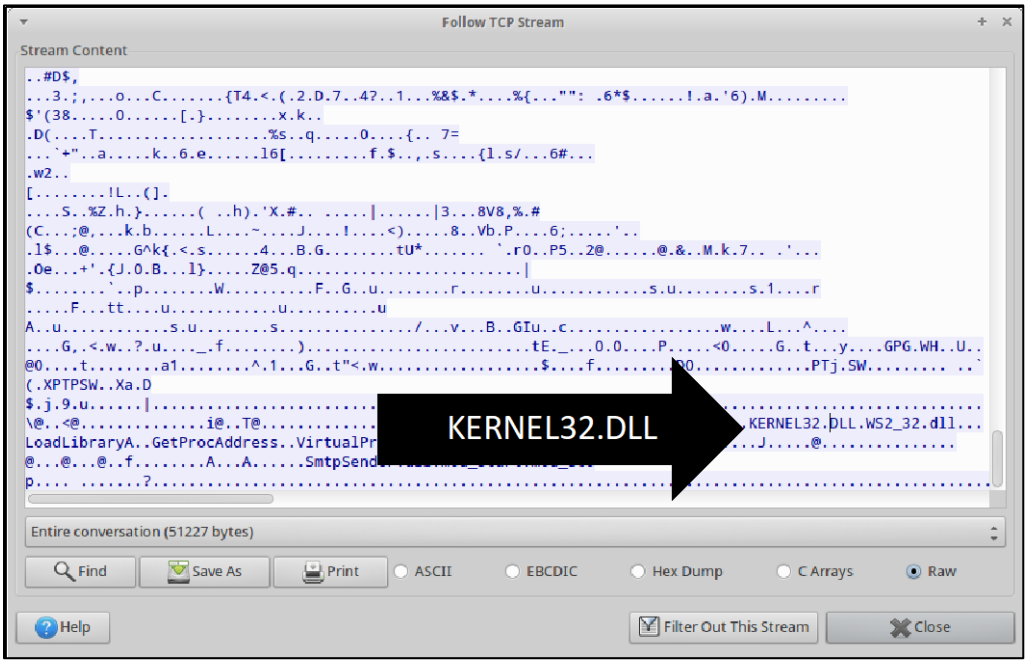


This TCP stream confirms the victim requested *tasksz.php* from *89.248.162.164* (red text). Beneath this request notice that the response includes a stage 2 executable called *1.exe* (blue text).

3. To determine if the bot attempted to change the kernel, start by searching for the Windows XP kernel filename (i.e., "KERNEL32.DLL") in the packet list. Choose **Edit** > **Find Packet** and search for this string. Be sure to use uppercase letters when searching since this feature is case sensitive by default.

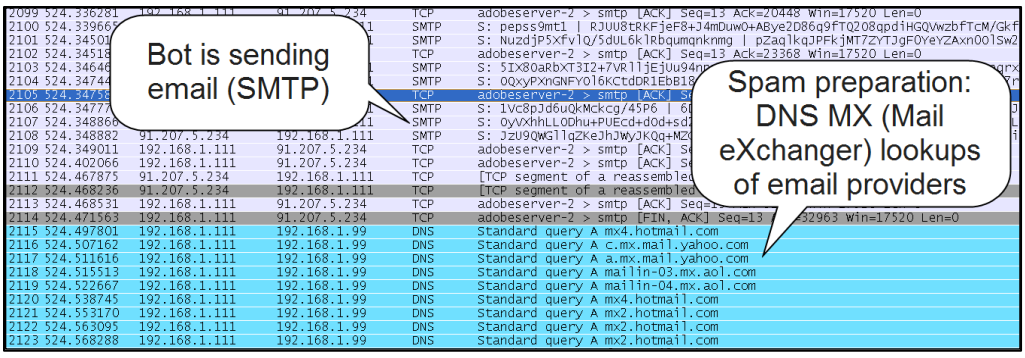
   One of the first results may indicate a malicious call referencing "KERNEL32.DLL." This reference may be an attempt to tamper with the kernel. More analysis is necessary to confirm this hypothesis.

4. Further analyze this packet using Wireshark's conversation-following capabilities. Right-click on the suspect packet and select **Follow TCP Stream**. Skimming through the stream to find "KERNEL32.DLL" may take some time. Instead, press **Find** and search for the filename to immediately jump to it.

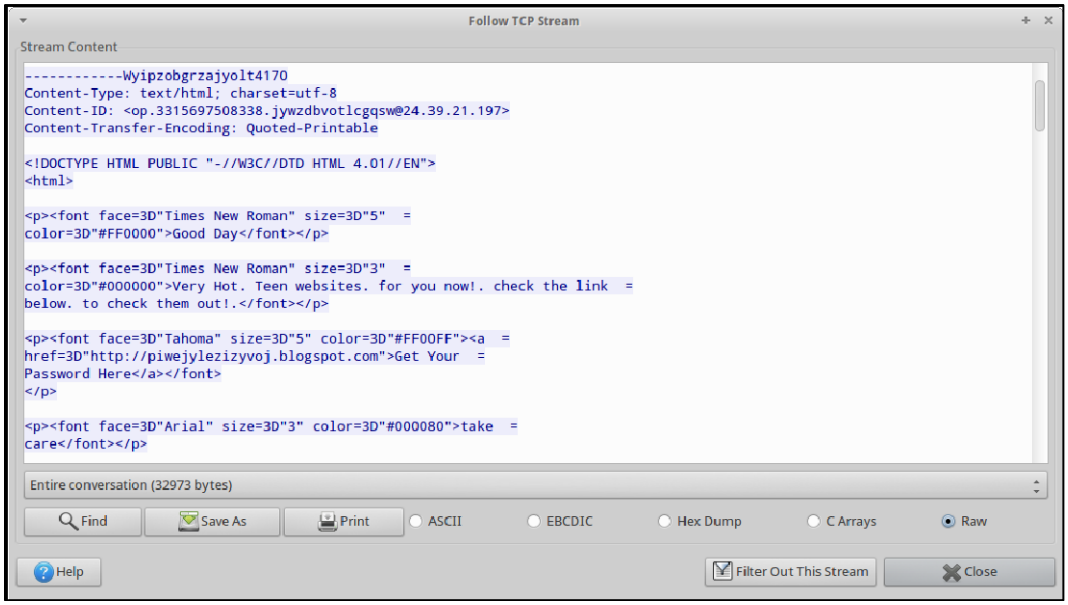


As noted in the hints, this reference may be an attempt to change the kernel. It could also be a program calling out to import advertised kernel library functions it needs to run, which is very common in Windows and is not malicious. With the information discovered so far, it is not possible to determine 1.exe's intent (i.e., *unsure*). Reverse engineering could help but is beyond the scope of this course.

5. For the remaining question, start by scrolling to the end of the packet list. Notice the SMTP and DNS activity. This traffic may be a bot looking up email provider DNS MX records and sending spam messages. More analysis is necessary to confirm this hypothesis.



6. Further analyze this packet using Wireshark's conversation-following capabilities. Right-click on one of the packets (e.g., 2066) and select **Follow TCP Stream**.



This stream confirms the start of an HTML *spamming* email message temping users to click a link (i.e., "Good Day … Check the link below. …") using the *SMTP* protocol.

Answers

- What is the name of the PHP file? **tasksz.php**
- What is the name of the Stage 2 executable? **1.exe**
- What is the PHP/EXE download IP address? **89.248.162.164**
- Did the bot attempt to change the kernel? **Unsure**
- What protocol was used for the bot activity? **SMTP**
- What is this activity called? **Spamming**