Yamin Mousselli
CS 8803: Cybersecurity Operations

Snort Write Up

Some Snort Resources I used:

- http://commons.oreilly.com/wiki/index.php/Snort_Cookbook/Rules_and_Signatures
- http://snort.datanerds.net/writing_snort_rules.htm
- http://manual-snort-org.s3-website-us-east-1.amazonaws.com/node27.html

Snort is light-weight and it's used to detect or drop specific byte sequences within a packet. Instead of investigating the content of every packet in Wireshark, IDS signatures are quick to search a ton of packets in a very short period of time. Moreover, IDS signatures are useful for things such as alerting a system when something weird pops up in a packet or dropping the packet(s) all-together if malicious strings are detected within packets. If Snort drops a packet, then it's preventing something from happening and thus an IPS. Otherwise, Snort is detecting and thus an IDS. Snort is technically an IDS/IPS. More specifically, Snort is either an IDS or IPS based on its configuration within an organization's network.

Most Snort lines are one-liners. Snort rules are divided into two logical sections: the rule header and the rule options. The rule header contains the rule's action (alert or drop), protocol (tcp, ip, udp), source and destination IP addresses, and the source and destinations ports. The rule header is everything before the parenthesis. The rule options are not required by any rule, are contained in the parenthesis, and are used to create tighter definitions for the rule to alert, collect, or drop packets.

There is a rules file and within that, there are #includes. This keyword is similar to the one in C where you will include the contents of the files that you have included. Another similarity to programming is that you can define variables and use those variables. Variables are written in all-uppercase similar to when writing constants in Java or C. For example, you can define these variables to be IP addresses.

- You would use the include keyword like this: include: <include file path/name>
- Make sure you put a semi-colon at the end of the signature revision.

Note: For user-defined rules, the signature id (sid) needs to be greater than 1,000,000. The sid is unique identifies the signature. Revisions keep track of signature modifications.

<u>Things to know for VIM</u>

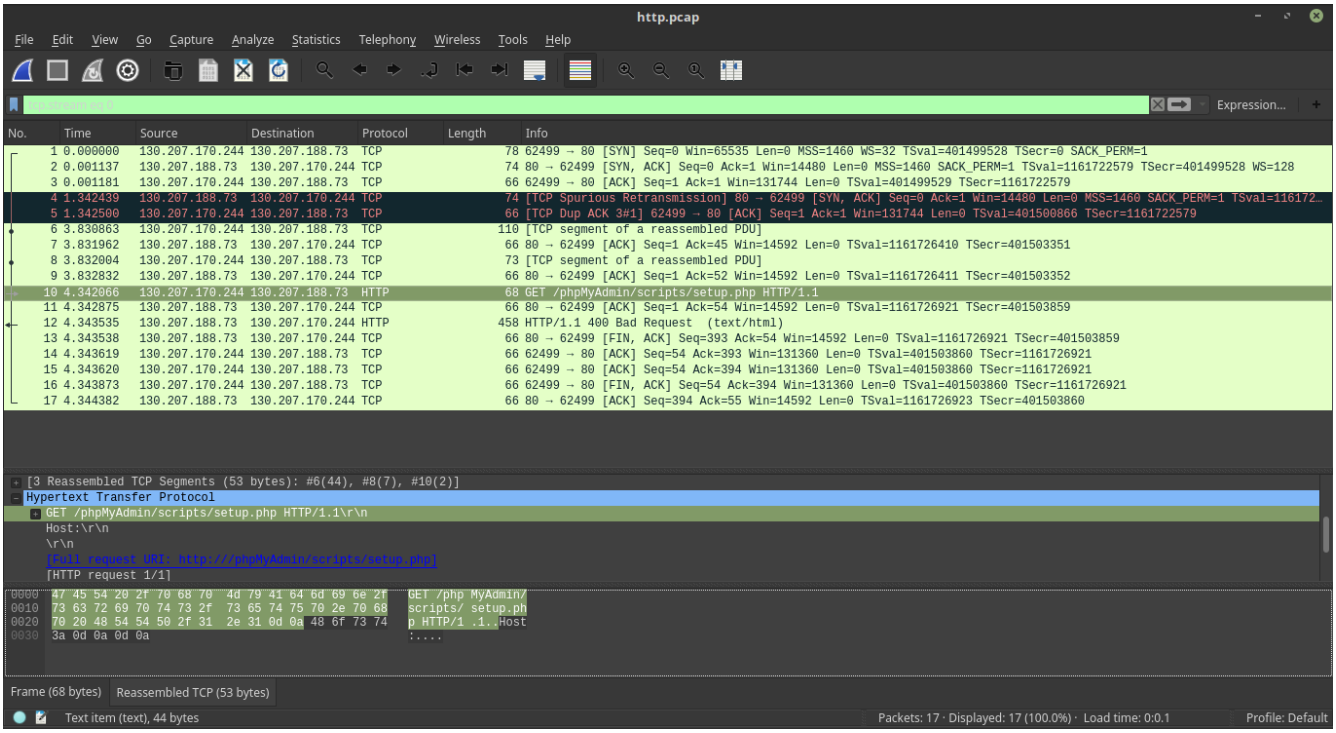
- To go back:
  - Press Escape
  - Press :
  - Type q! (! forces it)
- To insert, press i.
- To move at the beginning or end of file, use Shift ^ or Shift $, respectively.
- To save, use : w.
- You can enter wq! Which would save and quit in that order.

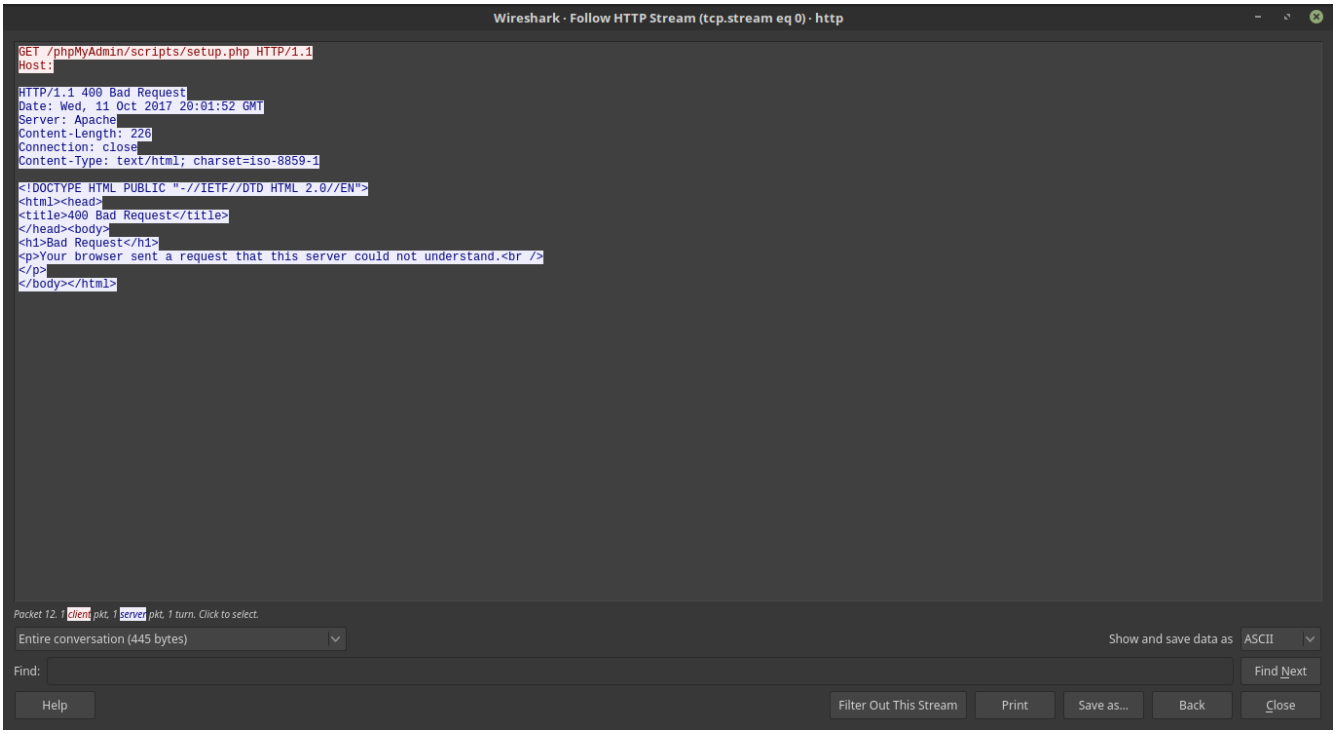
1)
Signature:
alert tcp 10.10.27.1 any → $HOME_NET $HTTP_PORTS (msg:“This is the malicious actor targeting our web services. “; sid:1000001; rev:1;)

2)
Note: You can inspect the user agent within the network tab in the developer tools of a web browser. A user agent will normally be set when you use a web browser to surf the Internet. You can access the Internet by using a script in which the user agent will not be set.

Note: Word applications have different arrows and quotation marks than Vim.

When I open Wireshark to inspect http.pcap, I see the GET request with the URI “/php/myMyAdmin/scripts/setup/php” and when I follow the HTTP stream, I do not see a user agent. I also see bad request. Additionally, there isn’t a host for the GET packet with that URI. This tells me something is wrong. If there isn’t a host defined for the packet, then it’s a bad request. The image below reflect my findings.





Additionally, we are going to alert on our web servers so the right hand side of the signature needs HOME_NET on all HTTP ports since we saw the URI was associated with a GET request. We need to alert on the URI and GET requests that don't have a user agent set. Therefore, we need to use the http_uri and http_header.

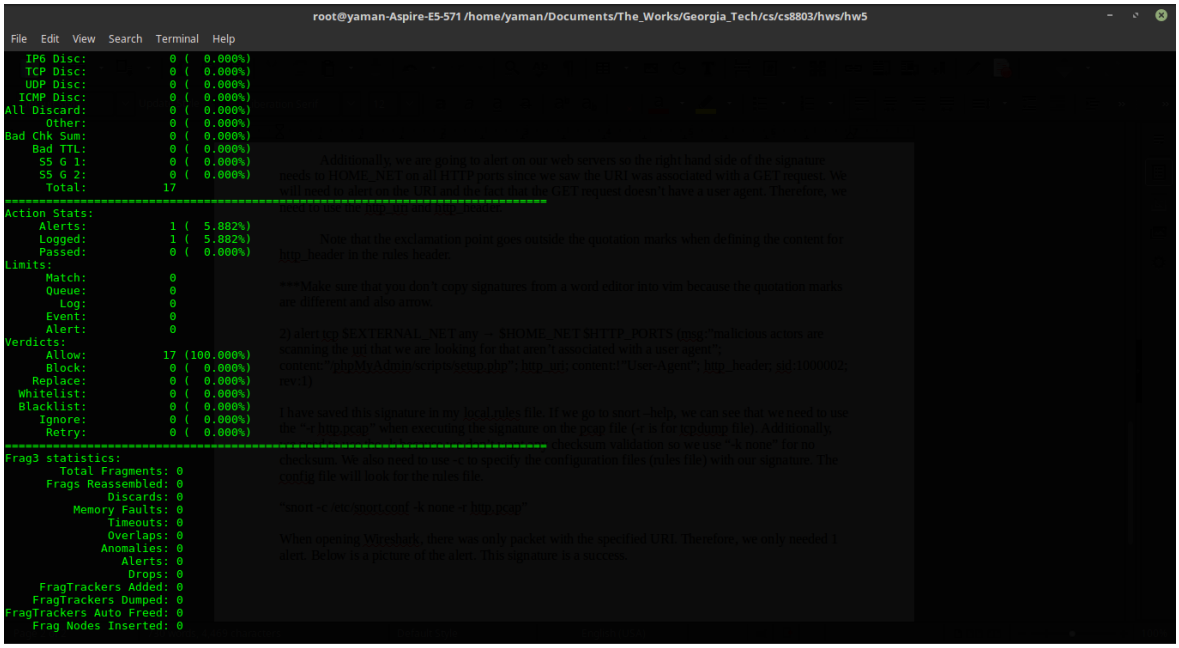
Note that the exclamation point goes outside the quotation marks when defining the content for http_header in the rules header.

Signature:
alert tcp $EXTERNAL_NET any → $HOME_NET $HTTP_PORTS (msg:"malicious actors are scanning the uri that we are looking for that aren't associated with a user agent"; content:"/phpMyAdmin/scripts/setup.php"; http_uri; content:!"User-Agent"; http_header; sid:1000002; rev:1;)

I saved this signature in my local.rules file. If we execute `snort --help` to look at our options, we can see that we need to use "-r http.pcap" when executing the signature on the pcap file (the -r option is for the tcpdump file). Additionally, we need to use the -k flag because we don't want any checksum validation so we use "-k none". We also need to use the -c option to specify the configuration files (rules file) with our signature. The configuration file will look for the rules file.

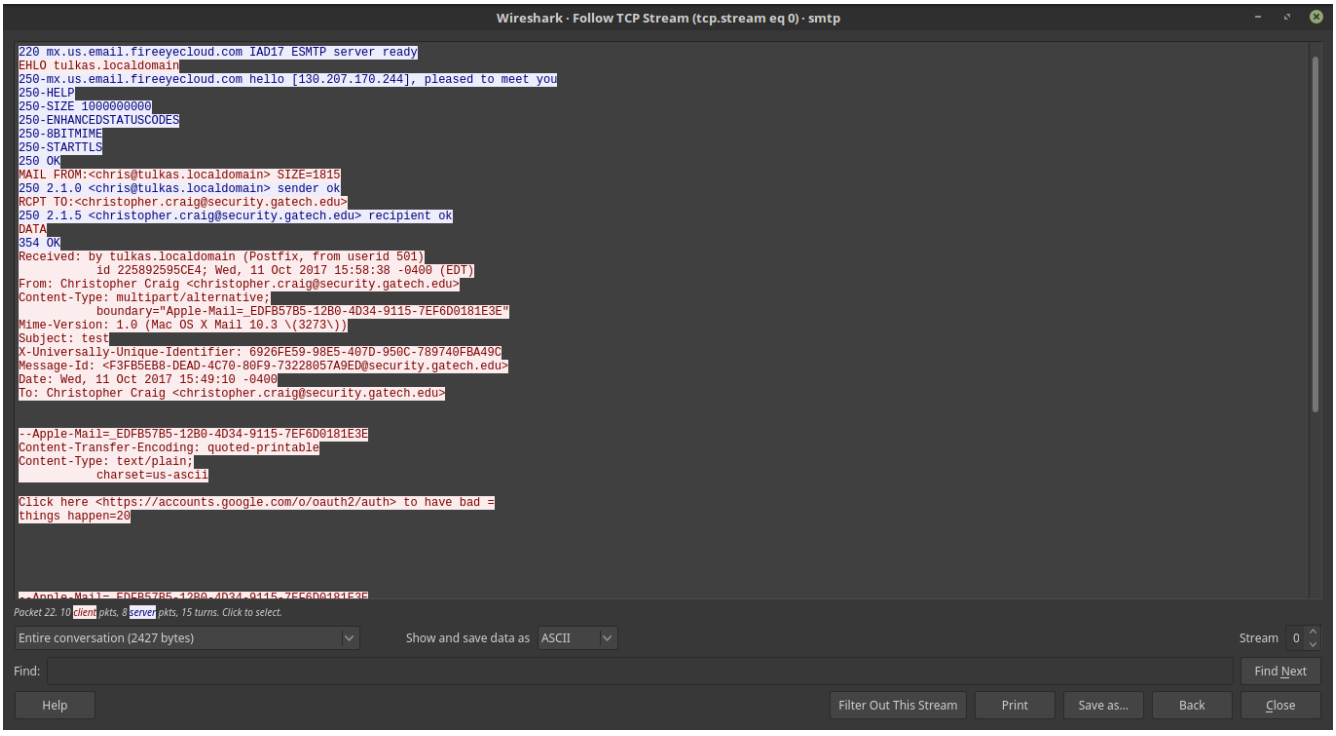
"snort -c /etc/snort.conf -k none -r http.pcap"

When opening Wireshark, there was only packet with the specified URI. Therefore, we only needed 1 alert. Below is a picture of the alert. This signature is successful.



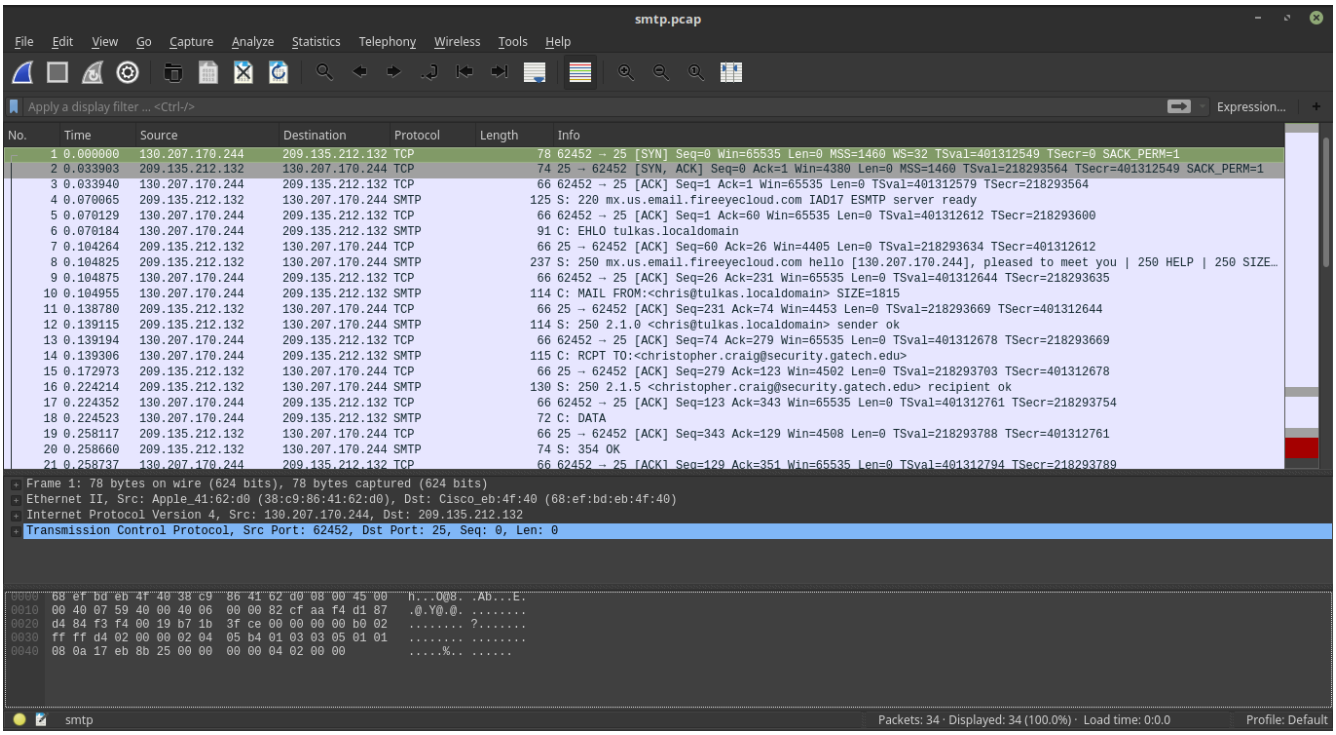
3)

There is only one TCP stream within smtp.pcap because there is only 1 email. When we follow tcp stream, we can see the link that was sent to the victim. The image below shows the email contents.



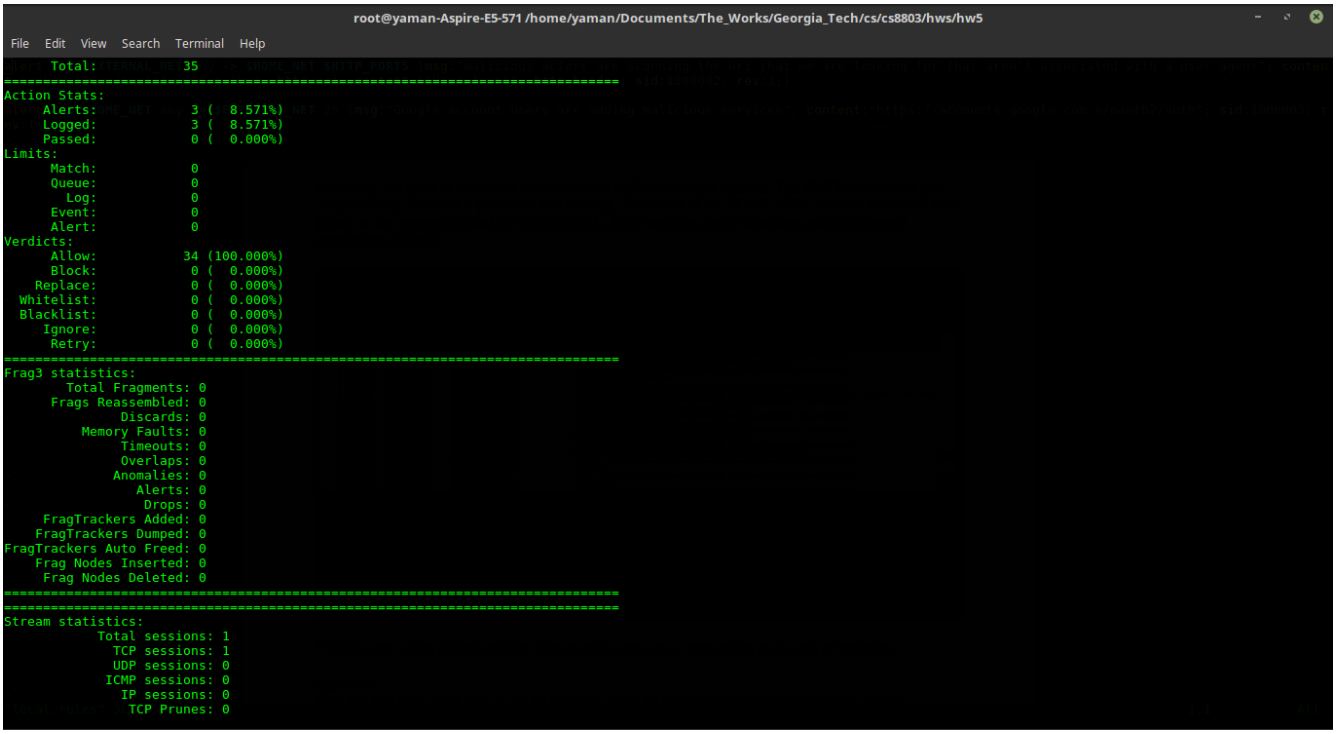
The content we are looking for within the pcap can be found when we follow TCP stream. We only need to put the link in the content. Also SMTP doesn't have a header specific for it. We don't need to specify anything like http_header or http_uri options because we are searching the content of the email.

Moreover, we need to see which ports to watch both ingress and egress traffic on. Always go to the first packet in Wireshark to see both source and destination ports (the source port will be the client). The SMTP server uses the well-known port 25. The image below reflects my finding. The EXTERNAL_NET variable defines malicious actors because they are the outbound connections and the HOME_NET variable is the server because it is what we are protecting (Think of it as us monitoring inbound connections).



Signature:
alert tcp $EXTERNAL_NET any → $HOME_NET 25 (msg:"google account users are adding malicious services"; content:"https://accounts.google.com/o/oauth2/auth"; sid:1000003; rev:1;)

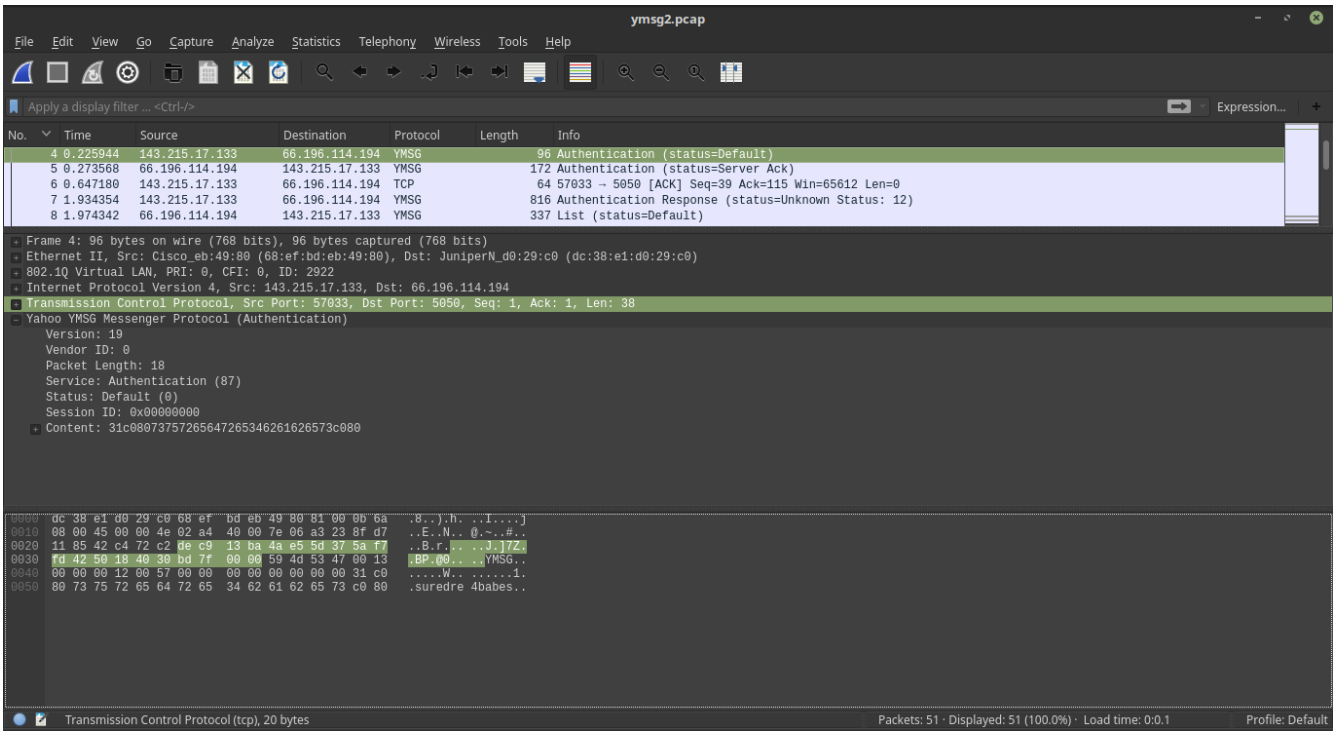
The signature alerted and logged three packets (there's a difference). The picture below reflects my findings.



Correction: The arrow in this signature should be bidirectional because you need to check both *inbound* and outbound mail. I had only checked for outbound mail.

4)

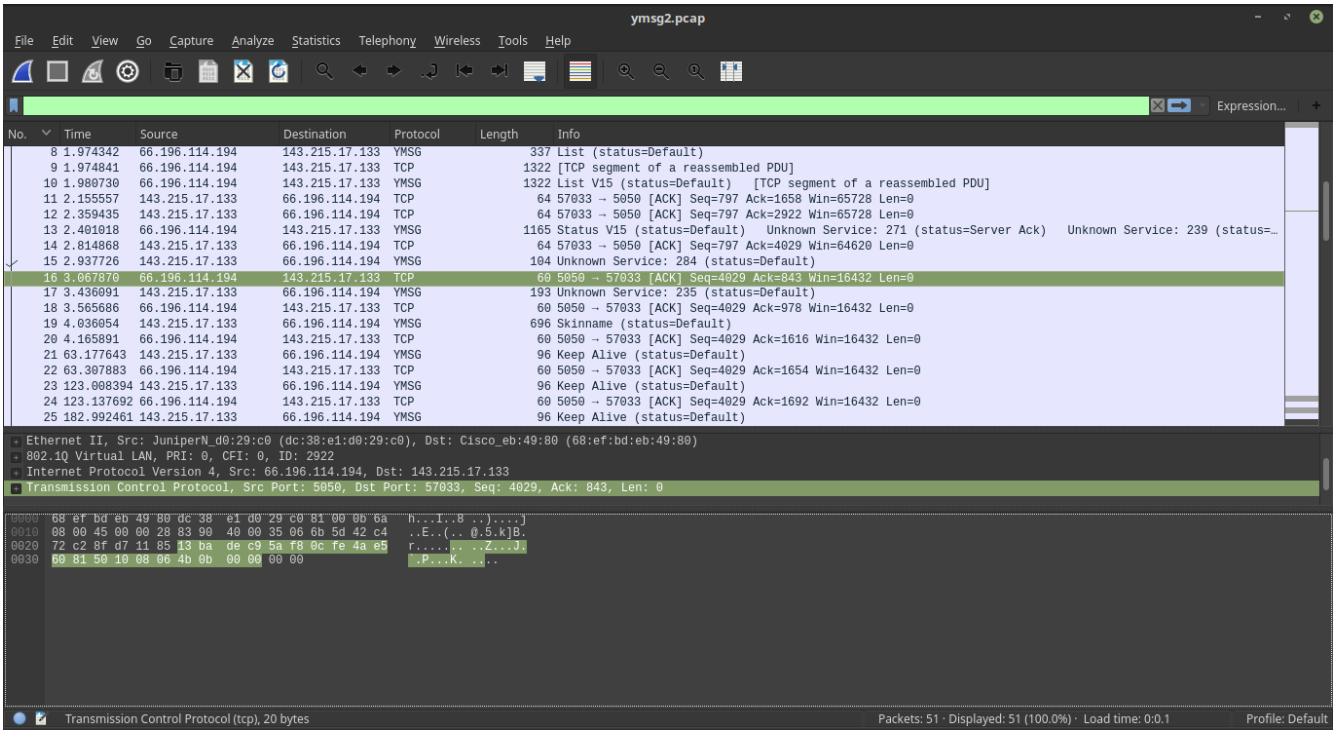
I see that the client IP is 143.215.17.133 and the server IP is 66.196.114.194 within ymsg.pcap in Wireshark. I also see that the destination port is 5050. The image below reflects my findings.



Through the link that was provided to us, I was able to figure out that the protocol number for the YAHOO_SERVICE_LIST is 0x55. The number encoding in Wireshark within the byte sequence is 00 55. We need to specify 2 bytes in our signature because the service number for the Yahoo messenger protocol is 2 bytes.

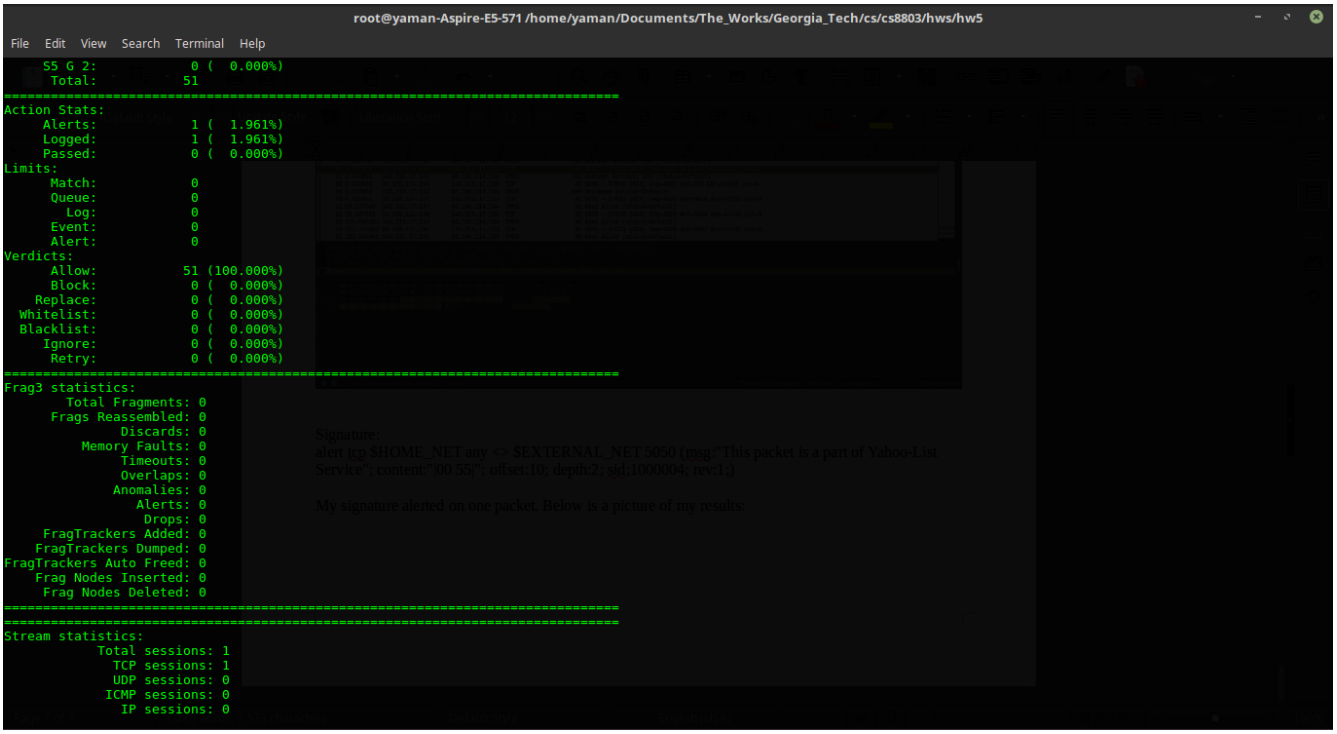
The offset is 10 bytes because in both the protocol and Wireshark, you can see the service number starts on the 11th byte. Therefore, you must start on the 11th byte. Offset always starts the beginning of the sequence. Depth specifies how far in the packet you should look. Depth should only be 2 bytes because the sequence number for the Yahoo messenger protocol is only 2 bytes long.

I also see that the server has an ACK which is weird. Usually, it's the client that sends the SYN but the server is asking the client for something which tells me the Yahoo Messenger Protocol is weird. This led me to have the bi-directional arrow in my signature because I need it to alert on both ends. The image below reflect my findings:



Signature:
alert tcp $HOME_NET any <> $EXTERNAL_NET 5050 (msg:"This packet is a part of Yahoo-List Service"; content:"|00 55|"; offset:10; depth:2; sid;1000004; rev:1;)

My signature alerted on one packet. Below is a picture of my results:



Correction: While the signature works, ports should be any b/c yahoo messenger could use other ports. Therefore, include content:"YMSG" within the signature

Below is a picture of all my signatures: