Menu

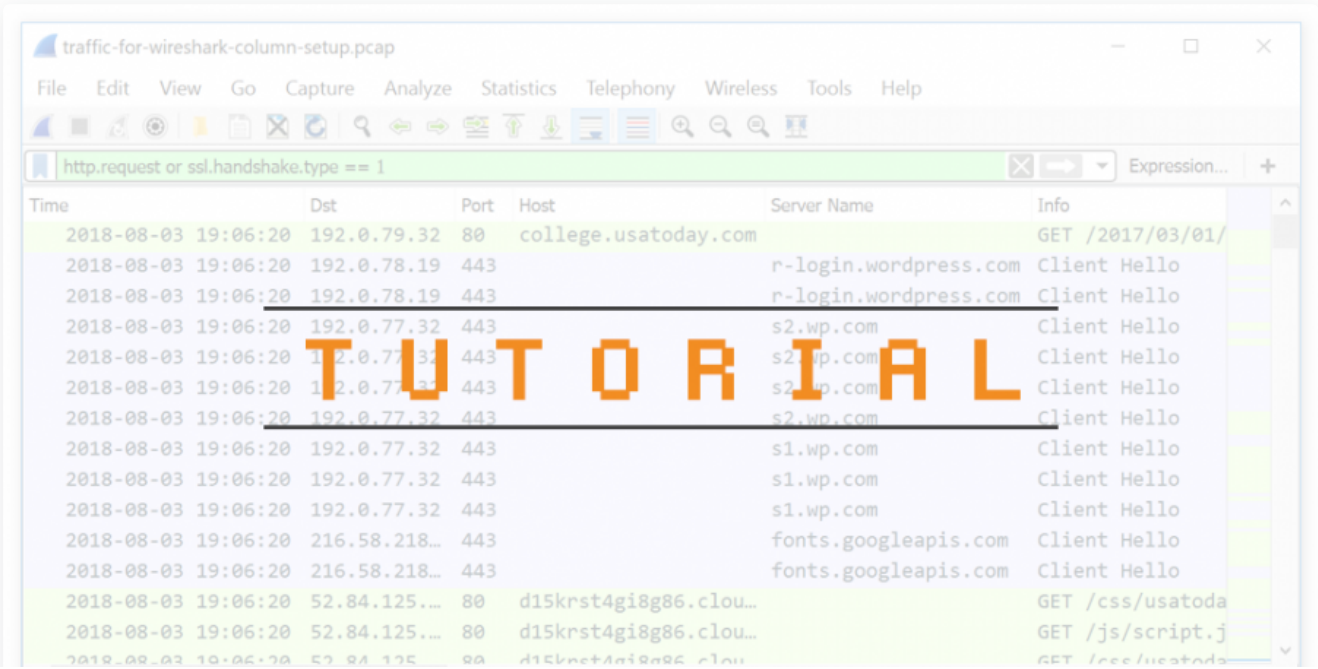
Using Wireshark – Display Filter Expressions

80,930 people reacted

33 9 min. read

SHARE

By Brad Duncan
January 11, 2019 at 6:00 AM
Category: Unit 42
Tags: pcap, tutorial, Wireshark, Wireshark Tutorial



This site uses cookies essential to its operation, for analytics, and for personalized content and ads. By continuing to browse this site, you acknowledge the use of cookies. Privacy statement

Cookie Settings

This post is also available in: 日本語 (Japanese)

As a Threat Intelligence Analyst for Palo Alto Networks Unit 42, I often use Wireshark to review packet captures (pcaps) of network traffic generated by malware samples. To better accomplish this work, I use a customized Wireshark column display as described my previous blog about using Wireshark. Today's post provides more tips for analysts to better use Wireshark. It covers display filter expressions I find useful in reviewing pcaps of malicious network traffic from infected Windows hosts.

Pcaps for this tutorial are available here. Keep in mind you must understand network traffic fundamentals to effectively use Wireshark. And you should also have a basic understanding of how malware infections occur. This is not a comprehensive tutorial on how to analyze malicious network traffic. Instead, it shows some tips and tricks for Wireshark filters. This tutorial covers the following areas:

- Indicators of infection traffic
- The Wireshark display filter
- Saving your filters
- Filters for web-based infection traffic
- Filters for other types of infection traffic

Indicators of Infection Traffic

This tutorial uses examples of Windows infection traffic from commodity malware distributed through mass-distribution methods like malicious spam (malspam) or web traffic. These infections can follow many different paths before the malware, usually a Windows executable file, infects a Windows host.

Indicators consist of information derived from network traffic that relates to the infection. These indicators are often referred to as Indicators of Compromise (IOCs). Security professionals often document indicators related to Windows infection traffic such as URLs, domain names, IP addresses, protocols, and ports. Proper use of the Wireshark display filter can help people quickly find these indicators.

The Wireshark Display Filter

Wireshark's display filter a bar located right above the column display section. This is where you type expressions to filter the frames, IP packets, or TCP segments that Wireshark displays from a pcap.

This site uses cookies essential to its operation, for analytics, and for personalized content and ads. By continuing to browse this site, you acknowledge the use of cookies. Privacy statement

› Cookie Settings

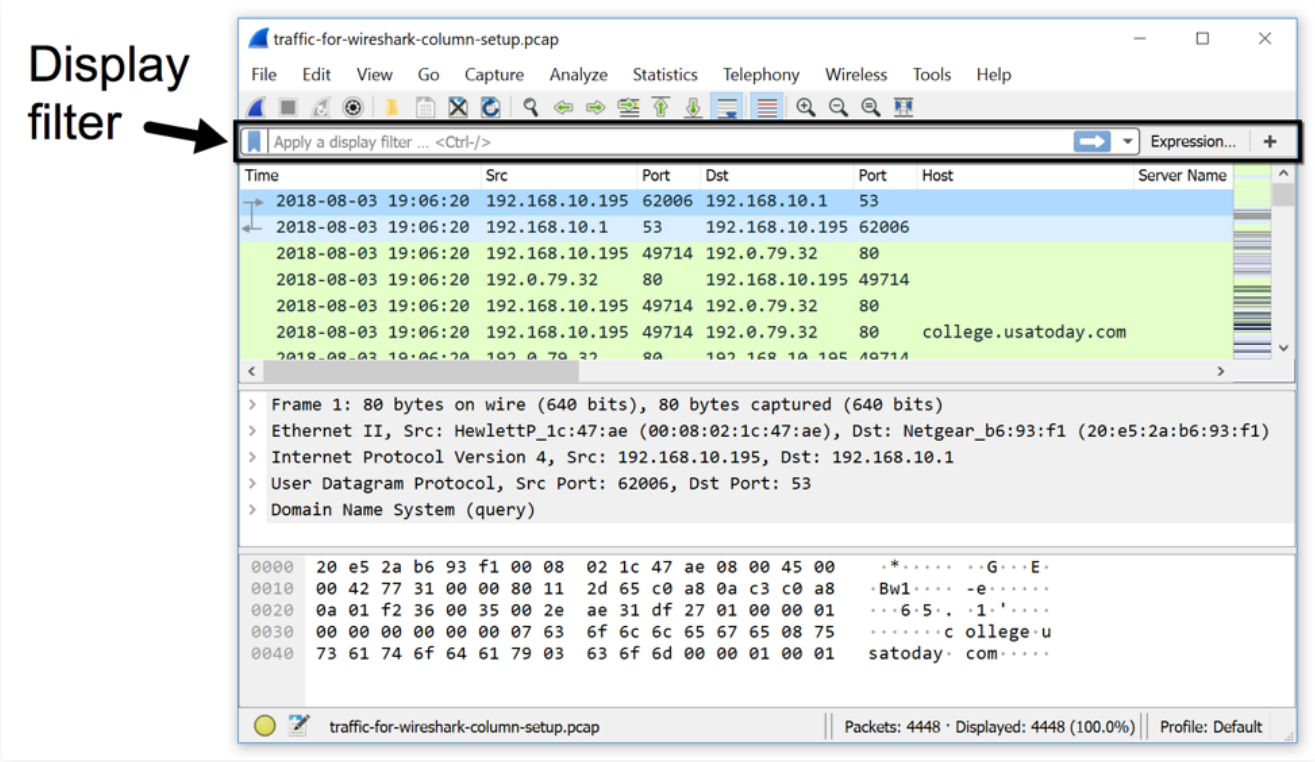


Figure 1. Location of the display filter in Wireshark.

If you type anything in the display filter, Wireshark offers a list of suggestions based on the text you have typed. While the display filter bar remains red, the expression is not yet accepted. If the display filter bar turns green, the expression has been accepted and should work properly. If the display filter bar turns yellow, the expression has been accepted, but it will probably not work as intended.

This site uses cookies essential to its operation, for analytics, and for personalized content and ads. By continuing to browse this site, you acknowledge the use of cookies. Privacy statement

› Cookie Settings

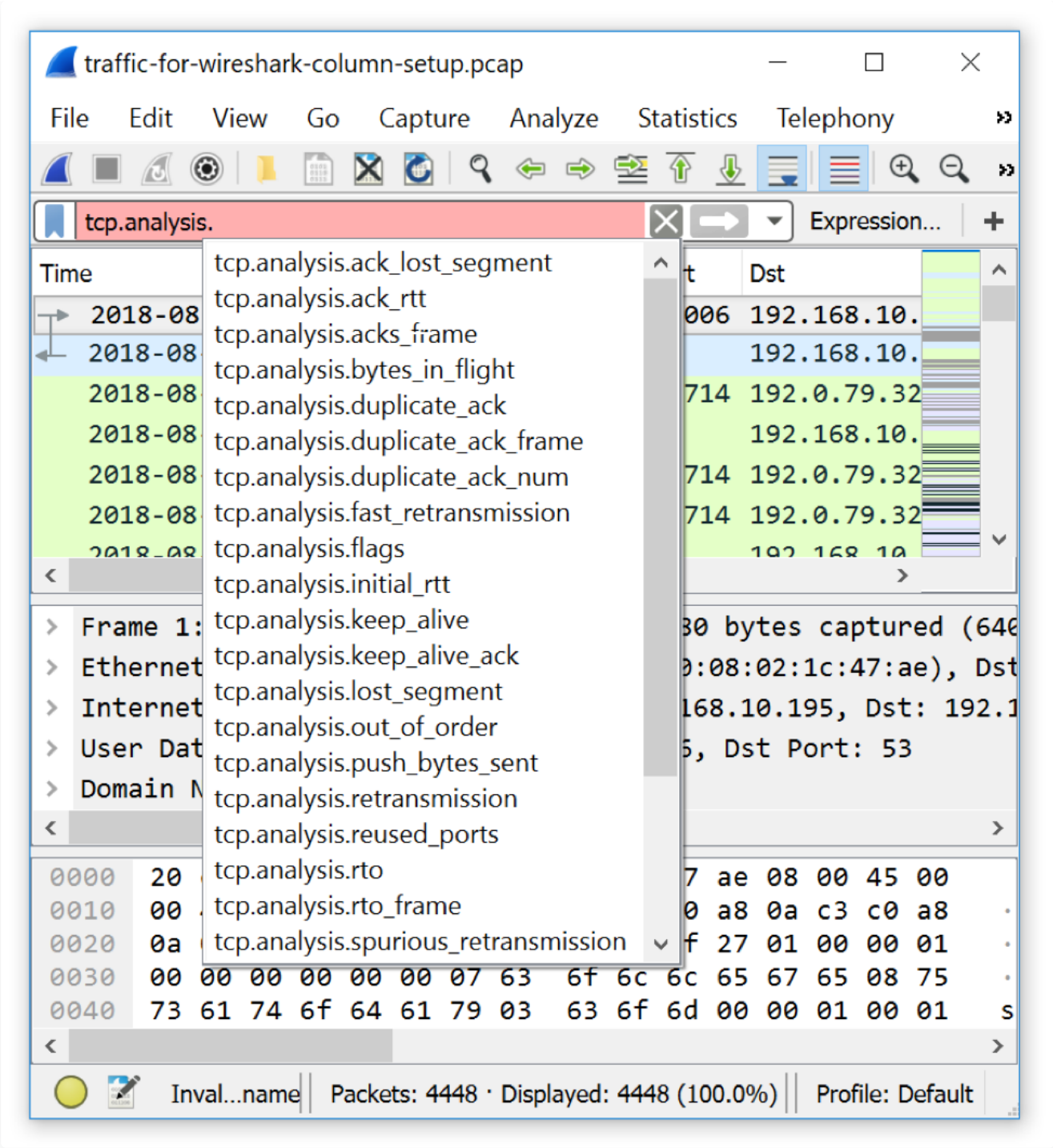


Figure 2. Wireshark's display filter offering suggestions based on what you type.

This site uses cookies essential to its operation, for analytics, and for personalized content and ads. By continuing to browse this site, you acknowledge the use of cookies. Privacy statement

› Cookie Settings

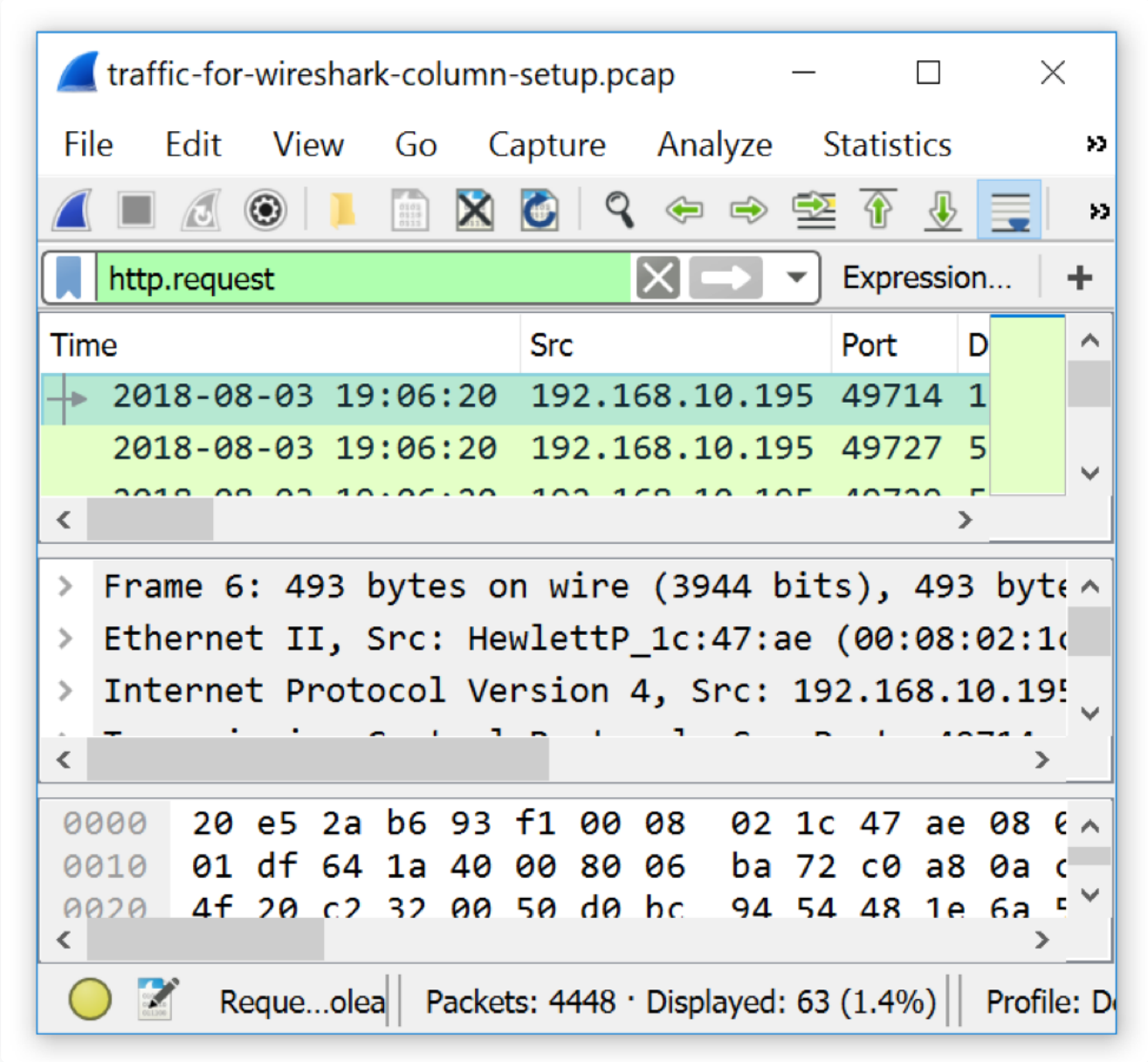


Figure 3. Wireshark's display filter accepts an expression, and it works as intended.

This site uses cookies essential to its operation, for analytics, and for personalized content and ads. By continuing to browse this site, you acknowledge the use of cookies. Privacy statement

› Cookie Settings

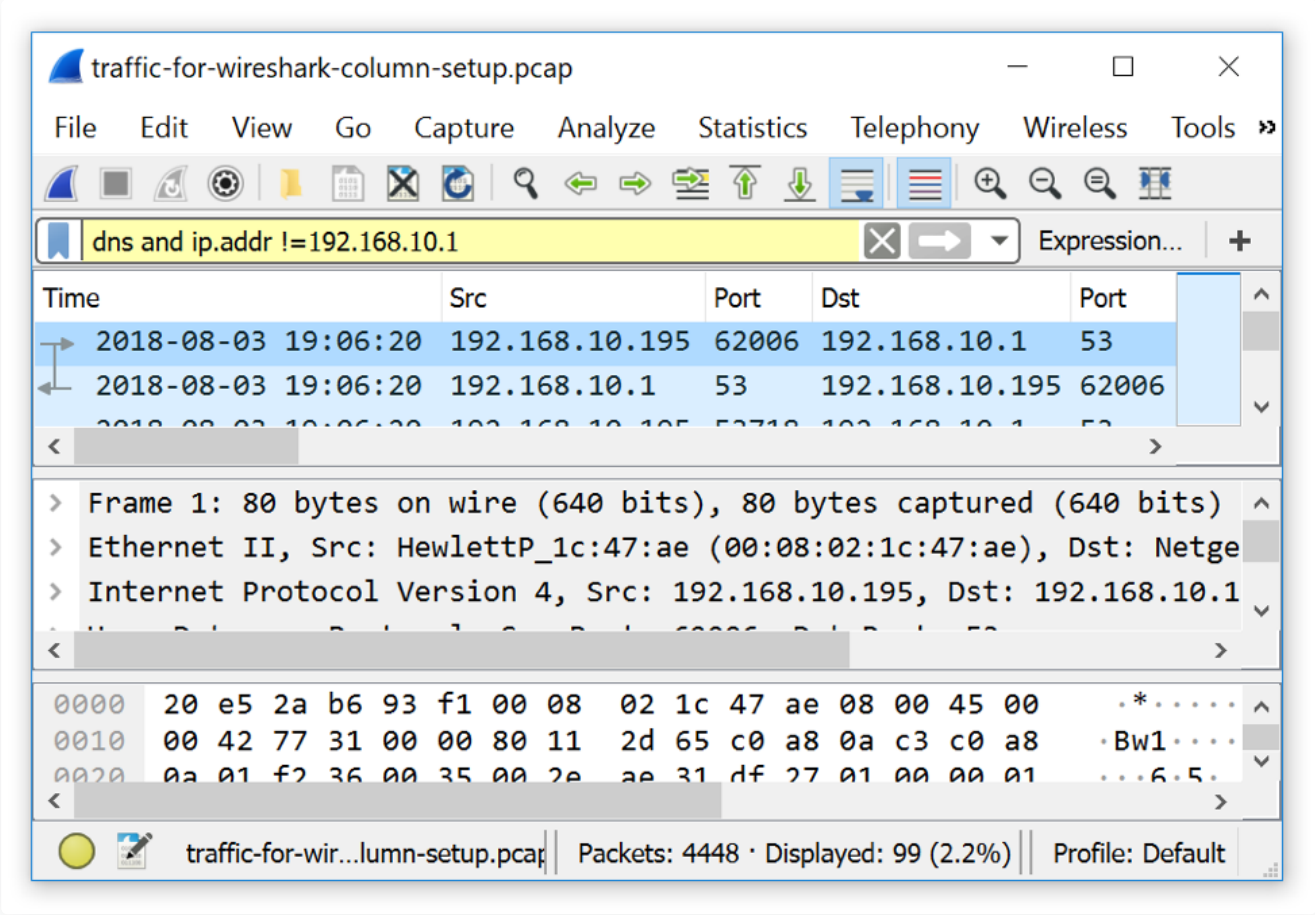


Figure 4. Example of Wireshark's display filter accepting an expression, but it does not work as intended.

Wireshark's display filter uses Boolean expressions, so you can specify values and chain them together. The following expressions are commonly used:

- Equals: == or ***eq***
- And: ***&&*** or ***and***
- Or: || (double pipe) or ***or***

Examples of these filter expressions follow:

- ***ip.addr eq 192.168.10.195 and ip.addr == 192.168.10.1***
- ***http.request && ip.addr == 192.168.10.195***
- ***http.request || http.response***
- ***dns.qry.name contains microsoft or dns.qry.name contains windows***

When specifying a value exclude, do not use ***!=*** in your filter expression. For example, if you want to specify all traffic that does not include IP address 192.168.10.1, use ***!(ip.addr eq 192.168.10.1)*** instead of ***ip.addr != 192.168.10.1*** because that second filter expression will not work properly.

This site uses cookies essential to its operation, for analytics, and for personalized content and ads. By continuing to browse this site, you acknowledge the use of cookies. Privacy statement

› Cookie Settings

As noted in my previous tutorial on Wireshark, I often use the following filter expression as a way to quickly review web traffic in a pcap:

http.request or ssl.handshake.type == 1.

The value ***http.request*** reveals URLs for HTTP requests, and ***ssl.handshake.type == 1*** reveals domains names used in HTTPS or SSL/TLS traffic.

My previous tutorial contains web traffic generated when a user viewed a URL from ***college.usatoday[.]com*** in August 2018. In the pcap, the user was on a Windows 10 computer using Microsoft's Edge web browser. Filtering on ***http.request or ssl.handshake.type == 1*** outlines the flow of events for this web traffic.

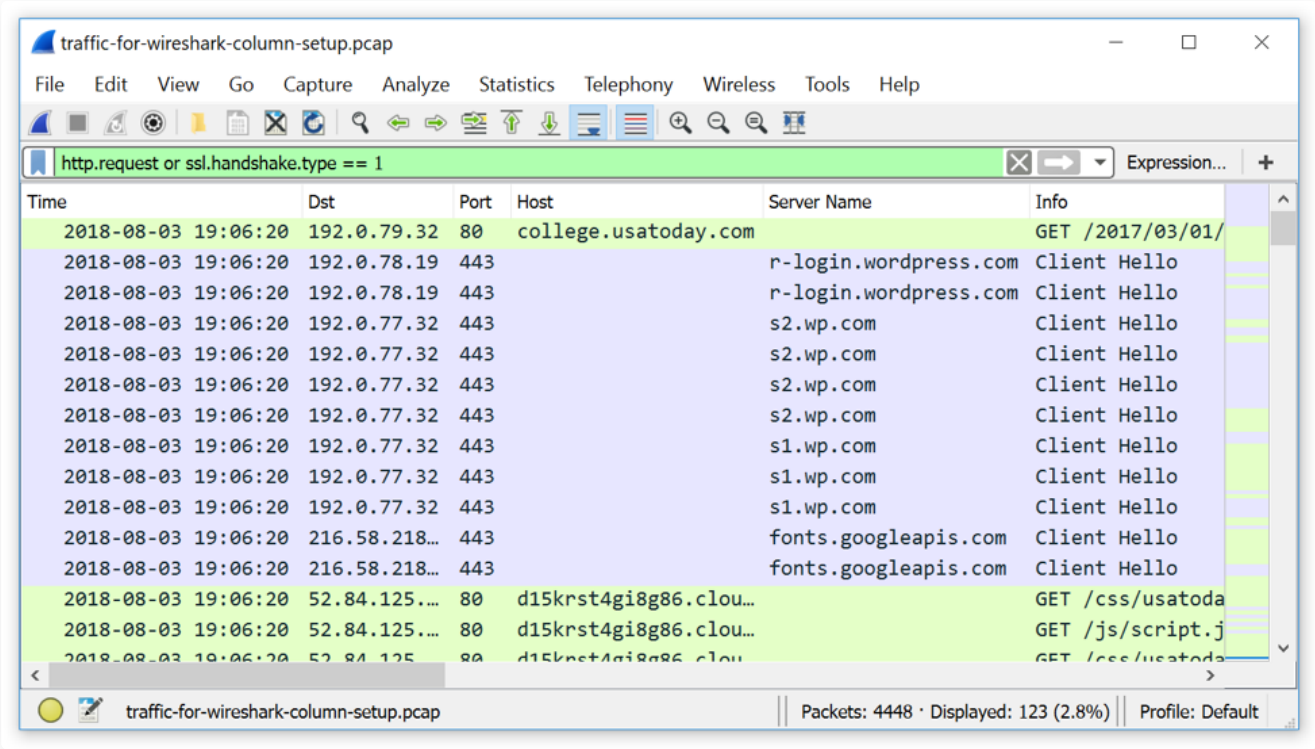


Figure 5. Filtering on web traffic using the previous tutorial's pcap.

However, I also generate pcaps of traffic using Windows 7 hosts, and this traffic includes HTTP requests over UDP port 1900 during normal activity. This HTTP traffic over UDP port 1900 is Simple Service Discovery Protocol (SSDP). SSDP is a protocol used to discover Plug & Play devices, and it is not associated with normal web traffic. Therefore, I filter this out using the following expression:

(http.request or ssl.handshake.type == 1) and !(udp.port eq 1900)

You can also use the following filter and achieve the same result:

This site uses cookies essential to its operation, for analytics, and for personalized content and ads. By continuing to browse this site, you acknowledge the use of cookies. Privacy statement

› Cookie Settings

Filtering out SSDP activity when reviewing a pcap from an infection on a Windows 7 host provides a much clear view of the traffic. Figure 6 shows Emotet activity with IcedID infection traffic from December 3rd, 2018 on a Windows 7 host. It is filtered on web traffic that contains SSDP requests. Figure 7 shows the same pcap filtered on web traffic *excluding* the SSDP requests, which provides a clearer picture of the activity.

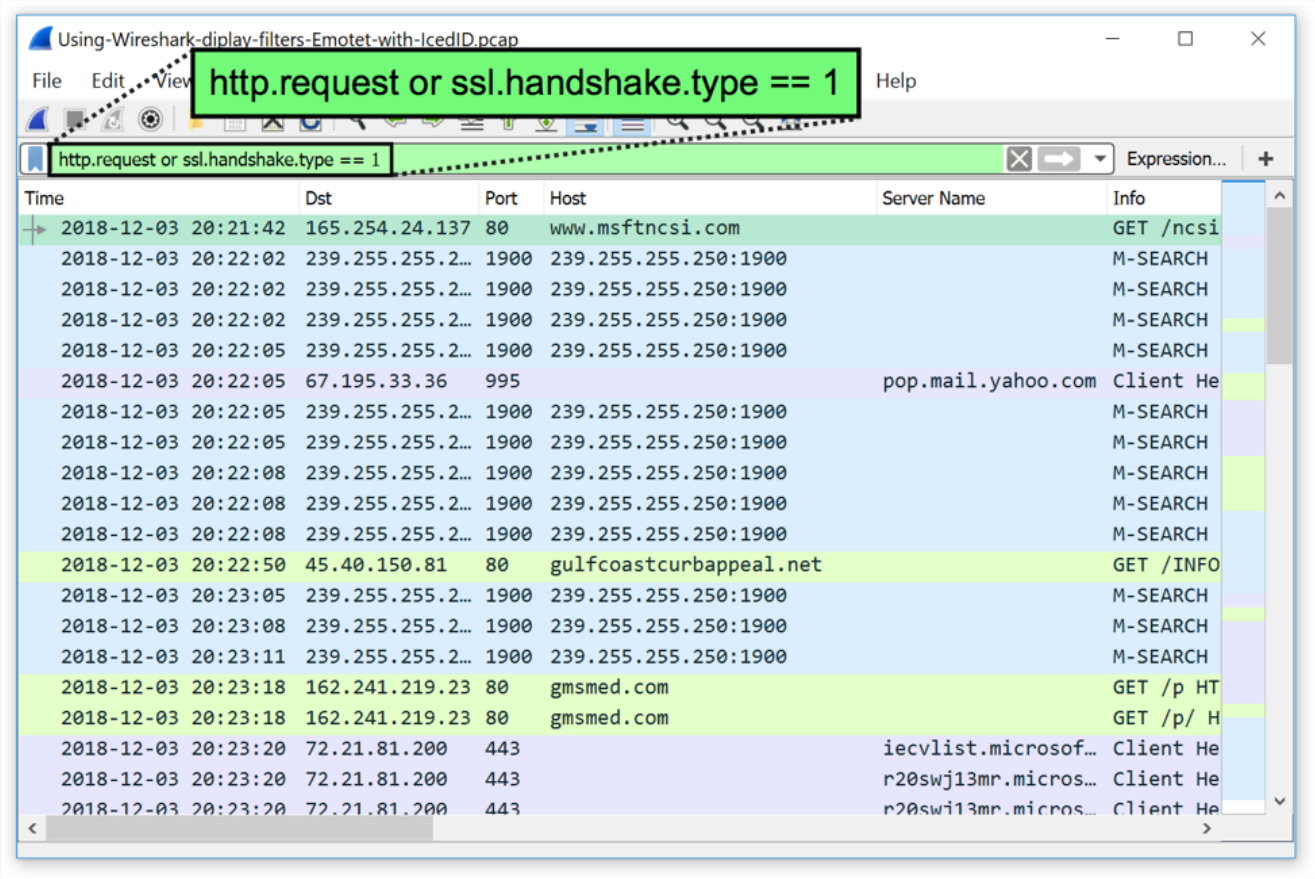


Figure 6. Reviewing web traffic with Emotet and IcedID infection activity in Wireshark without filtering out SSDP traffic.

This site uses cookies essential to its operation, for analytics, and for personalized content and ads. By continuing to browse this site, you acknowledge the use of cookies. Privacy statement

› Cookie Settings

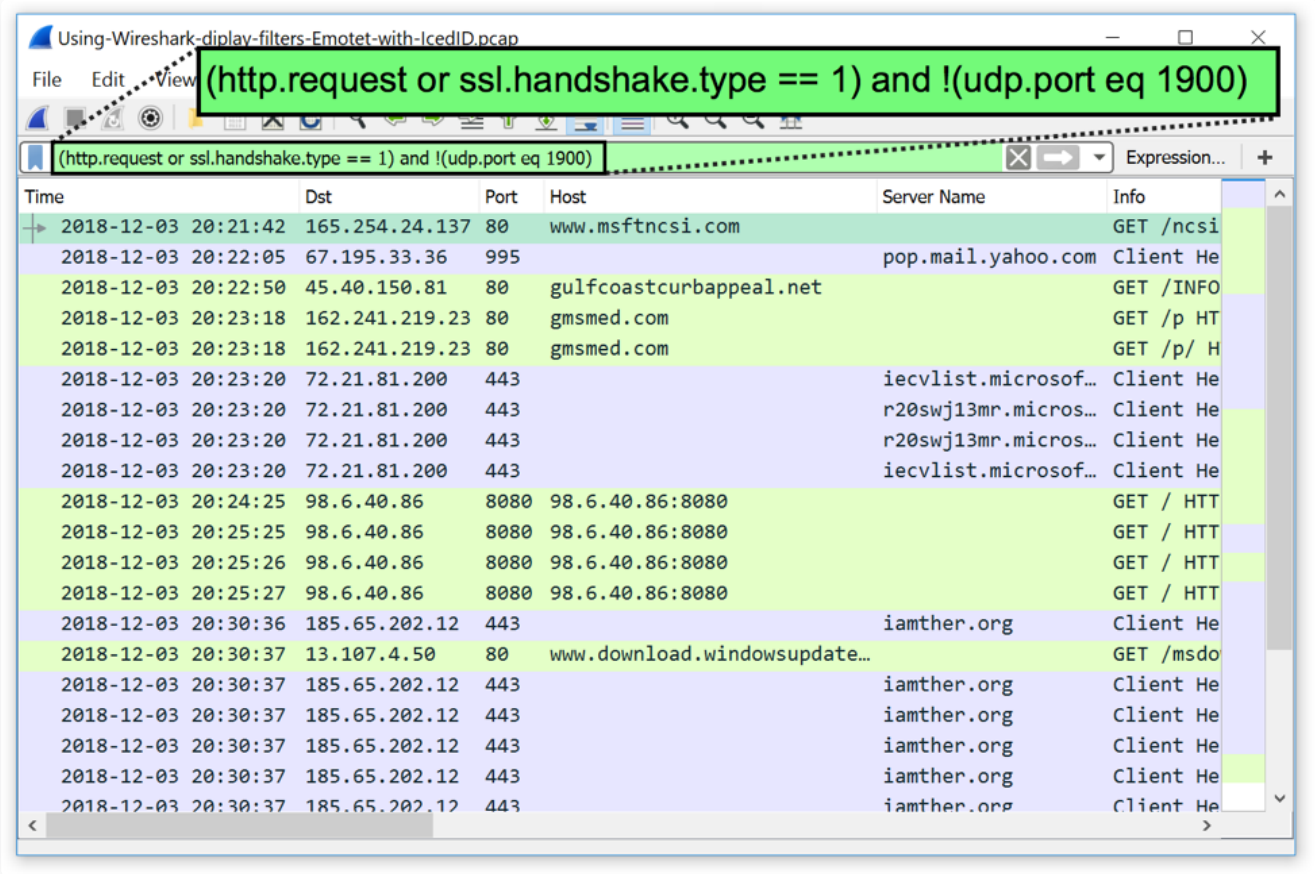


Figure 7. Reviewing web traffic with Emotet and IcedID infection activity in Wireshark while filtering out SSDP traffic.

In Figure 7, we see some indicators of infection traffic, but not every indicator of the infection is revealed. In some cases, an infected host may try to connect with a server that has been taken off-line or is refusing a TCP connection. These attempted connections can be revealed by including TCP SYN segments in your filter by adding ***tcp.flags eq 0x0002***. Try the following filter on the same traffic:

(http.request or ssl.handshake.type == 1 or tcp.flags eq 0x0002) and !(udp.port eq 1900)

Including the TCP SYN segments on your search reveals the infected host also attempted to connect with IP address 217.164.2[.]133 over TCP port 8443 as shown in Figure 8.

This site uses cookies essential to its operation, for analytics, and for personalized content and ads. By continuing to browse this site, you acknowledge the use of cookies. Privacy statement

› Cookie Settings

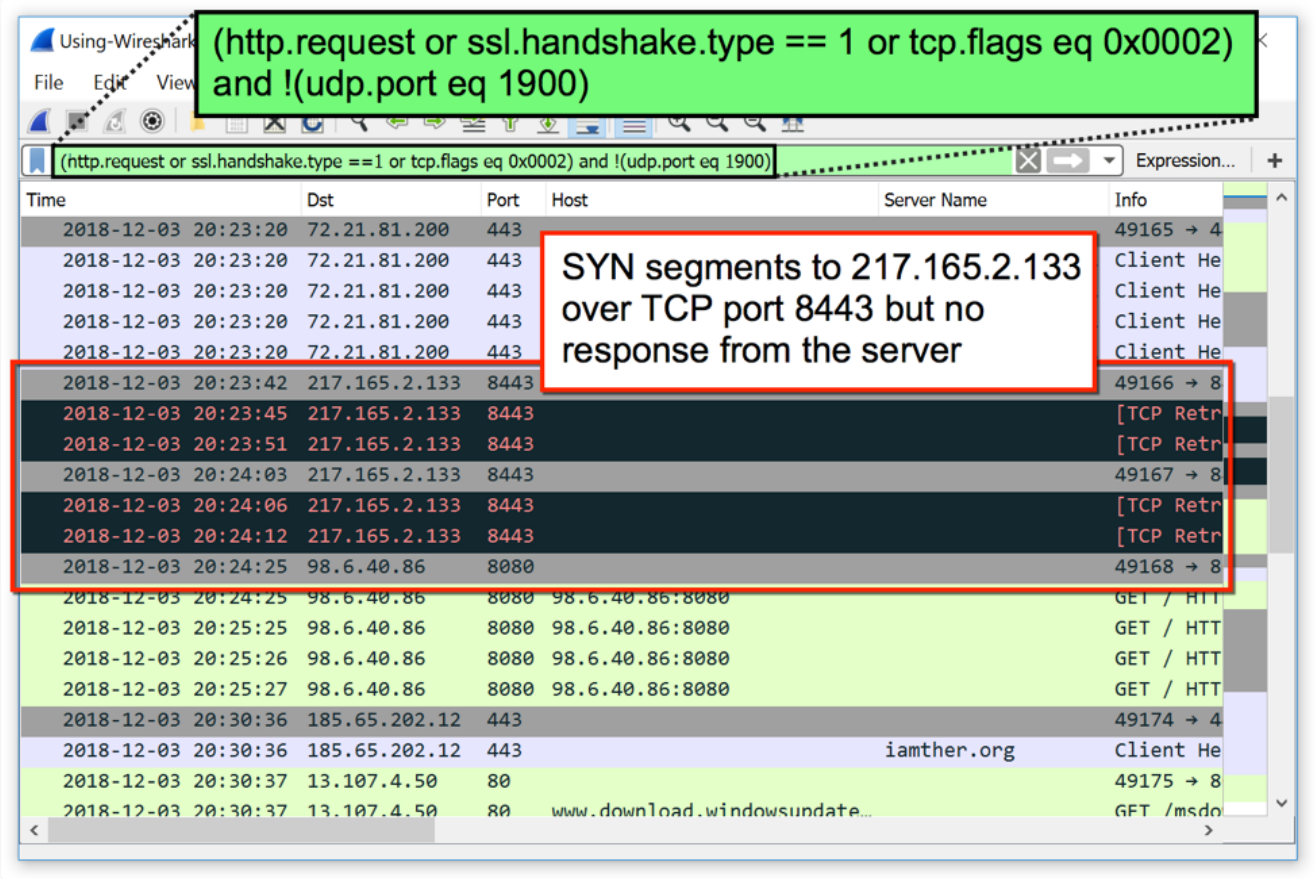


Figure 8. Including TCP SYN segments in your filter can reveal unsuccessful connection attempts by an infected host to other servers.

Filters for Other Types of Infection Traffic

In some cases, post-infection traffic will not be web-based, and an infected host will contact command and control (C2) servers. These servers can be directly hosted on IP addresses, or they can be hosted on servers using domain names. Some post-infection activity, like C2 traffic caused by the Nanocore Remote Access Tool (RAT), is not HTTP or HTTPS/SSL/TLS traffic.

Therefore, I often add DNS activity when reviewing a pcap to see if any of these domains are active in the traffic. This results in the following filter expression:

(http.request or ssl.handshake.type == 1 or tcp.flags eq 0x0002 or dns) and !(udp.port eq 1900)

In Figure 9, I use the above filter expression to review a pcap showing a Nanocore RAT executable file downloaded from ***www.mercedes-club-bg[.]com*** to infect a vulnerable Windows host. The initial download is followed by attempted TCP connections to ***franex.sytes[.]net*** at 185.163.45[.]48 and ***franexserve.duckdns[.]org*** at 95.213.251[.]165. Figure 10 shows the correlation between the DNS queries and the TCP traffic.

This site uses cookies essential to its operation, for analytics, and for personalized content and ads. By continuing to browse this site, you acknowledge the use of cookies. Privacy statement

› Cookie Settings

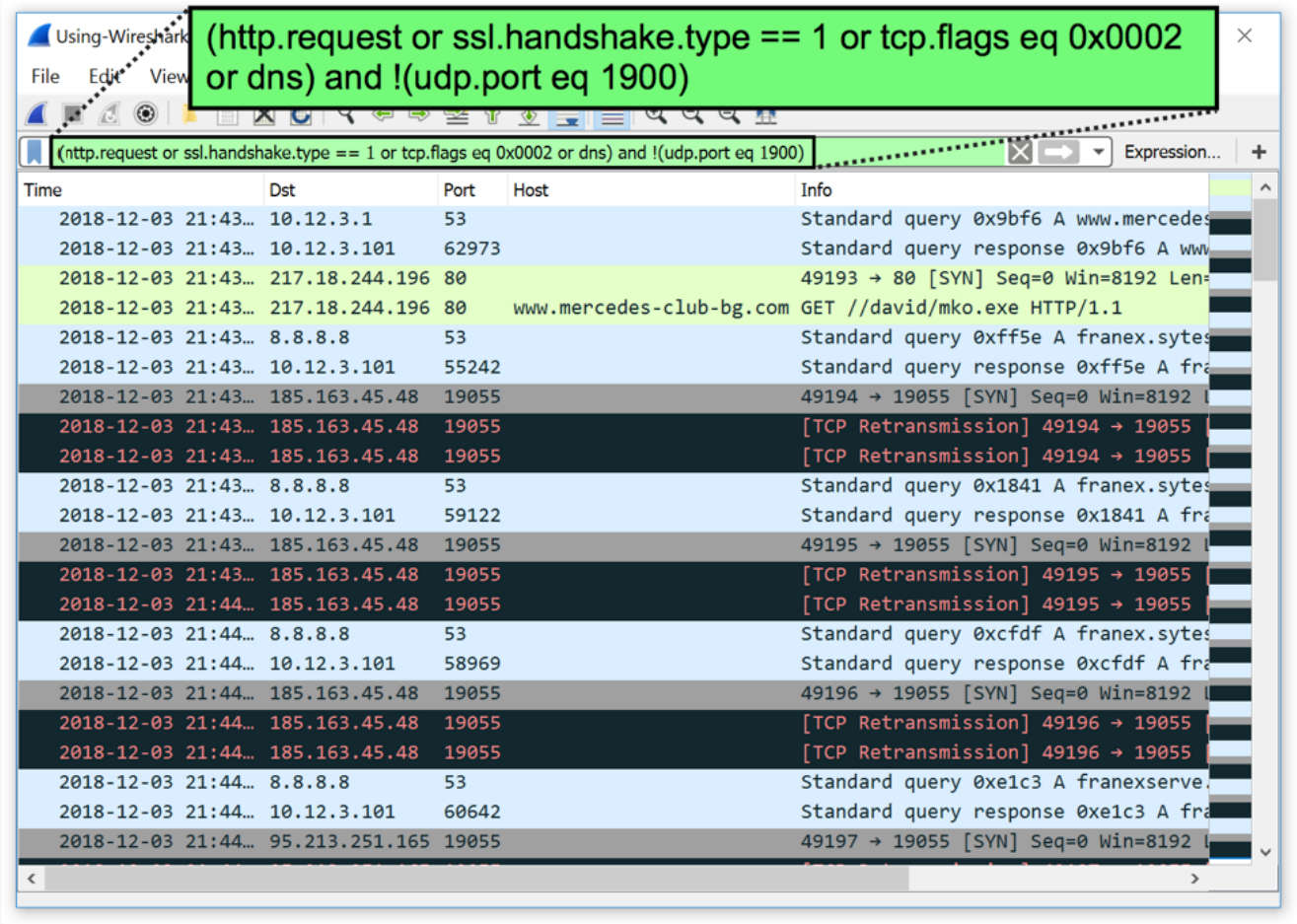
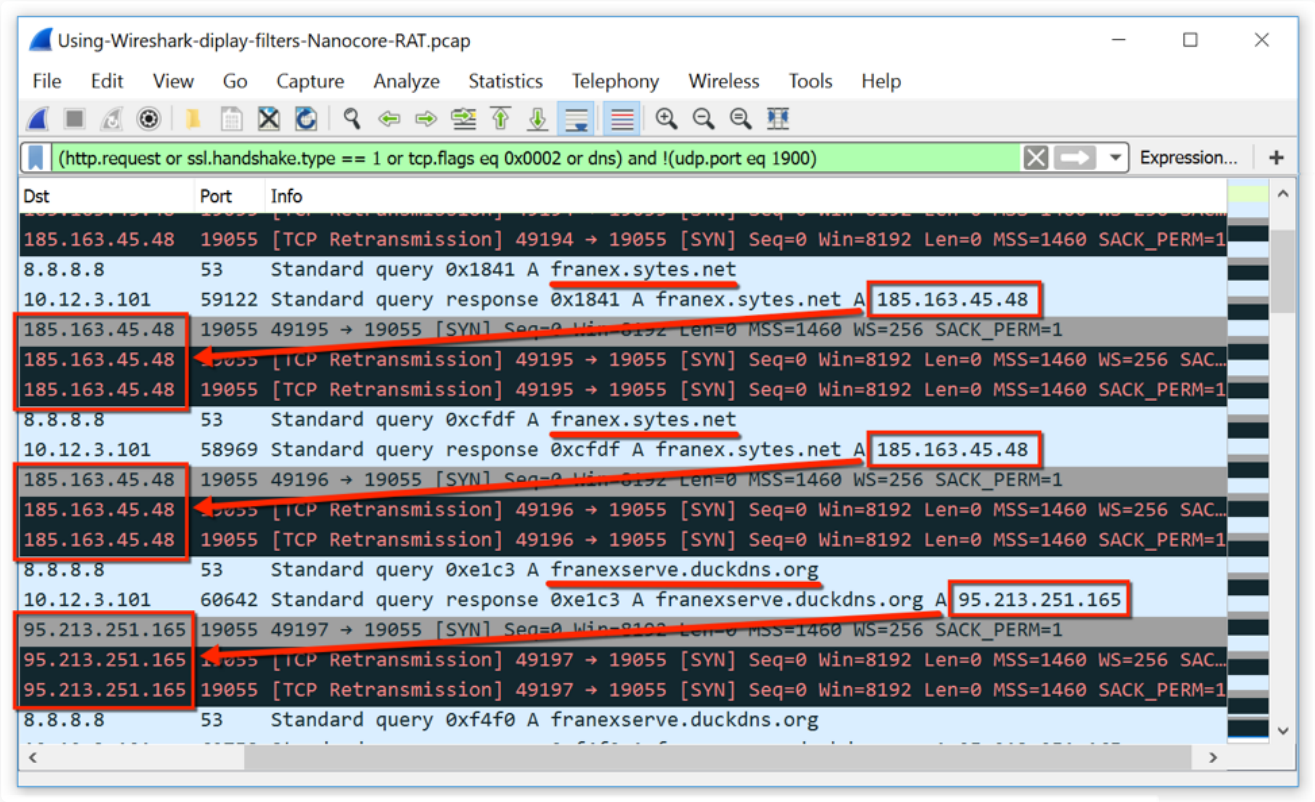


Figure 9. Including DNS queries reveals attempted TCP connections to additional domains.



This site uses cookies essential to its operation, for analytics, and for personalized content and ads. By continuing to browse this site, you acknowledge the use of cookies. Privacy statement

› Cookie Settings

Some infection traffic uses common protocols that can easily be decoded by Wireshark. Figure 11 shows post-infection traffic caused by this malware executable that generates FTP traffic. Using a standard web traffic search that also checks for DNS traffic and TCP SYN flags, we find traffic over TCP port 21 and other TCP ports after a DNS query to ***ftp.totallyanonymous[.]com***.

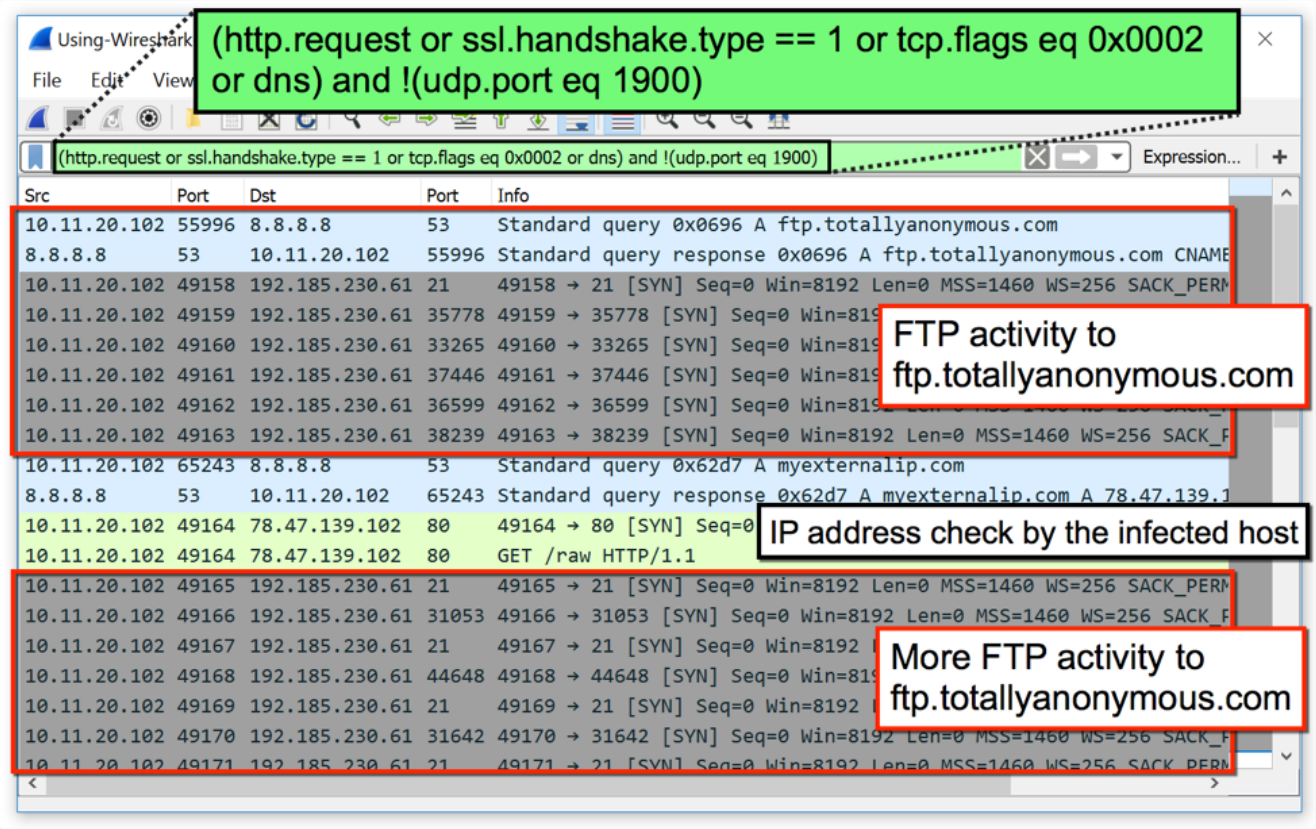


Figure 11. Activity from malware generating FTP traffic.

Realizing this is FTP traffic, you can pivot on ***ftp*** for your display filter as shown in Figure 12. When filtering on ***ftp*** for this pcap, we find the infected Windows host logged into an FTP account at ***totallyanonymous.com*** and retrieved files named ***fc32.exe*** and ***o32.exe***. Scroll down to later FTP traffic as shown in Figure 13, and you will find a file named ***6R7MELYD6*** sent to the FTP server approximately every minute. Further investigation would reveal ***6R7MELYD6*** contains password data stolen from the infected Windows host.

This site uses cookies essential to its operation, for analytics, and for personalized content and ads. By continuing to browse this site, you acknowledge the use of cookies. Privacy statement

› Cookie Settings

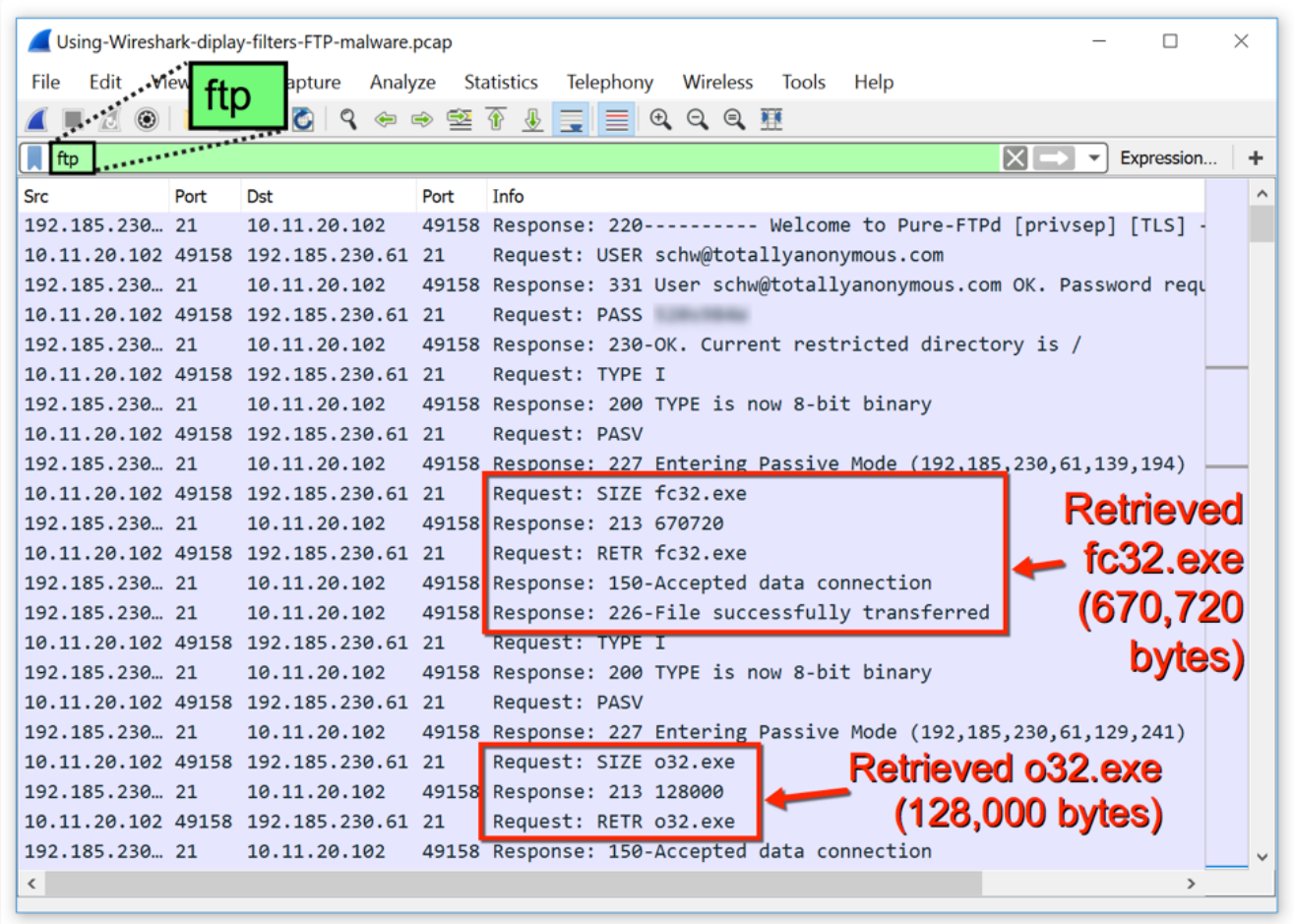


Figure 12. Using ftp as a filter and finding the name of files retrieved by the infected host when viewing the FTP control channel over TCP port 21.

This site uses cookies essential to its operation, for analytics, and for personalized content and ads. By continuing to browse this site, you acknowledge the use of cookies. Privacy statement

› Cookie Settings

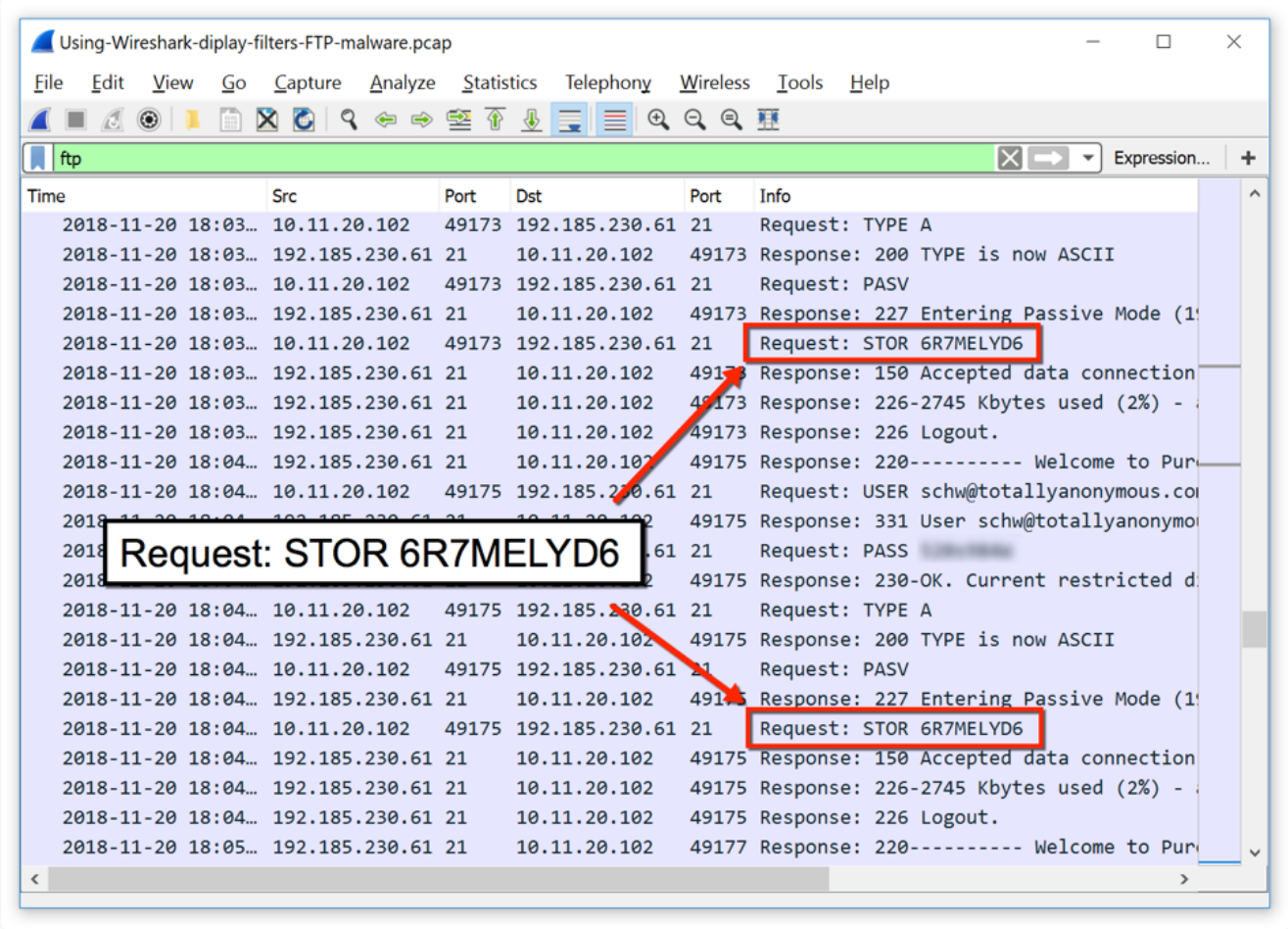


Figure 13. The FTP control channel over TCP port 21 also shows information stored to the FTP server as a file named 6R7MELYD6.

In addition to FTP, malware can use other common protocols for malicious traffic. Spambot malware can turn an infected host into a spambot designed to send dozens to hundreds of email messages every minute. This is characterized by several DNS requests to various mail servers followed by SMTP traffic on TCP ports 25, 465, 587, or other TCP ports associated with email traffic.

Let's try this filter expression again:

(http.request or ssl.handshake.type == 1 or tcp.flags eq 0x0002 or dns) and !(udp.port eq 1900)

When viewing spambot traffic, you'll find DNS queries to mail servers and TCP traffic to SMTP-related ports as previously described.

This site uses cookies essential to its operation, for analytics, and for personalized content and ads. By continuing to browse this site, you acknowledge the use of cookies. Privacy statement

› Cookie Settings

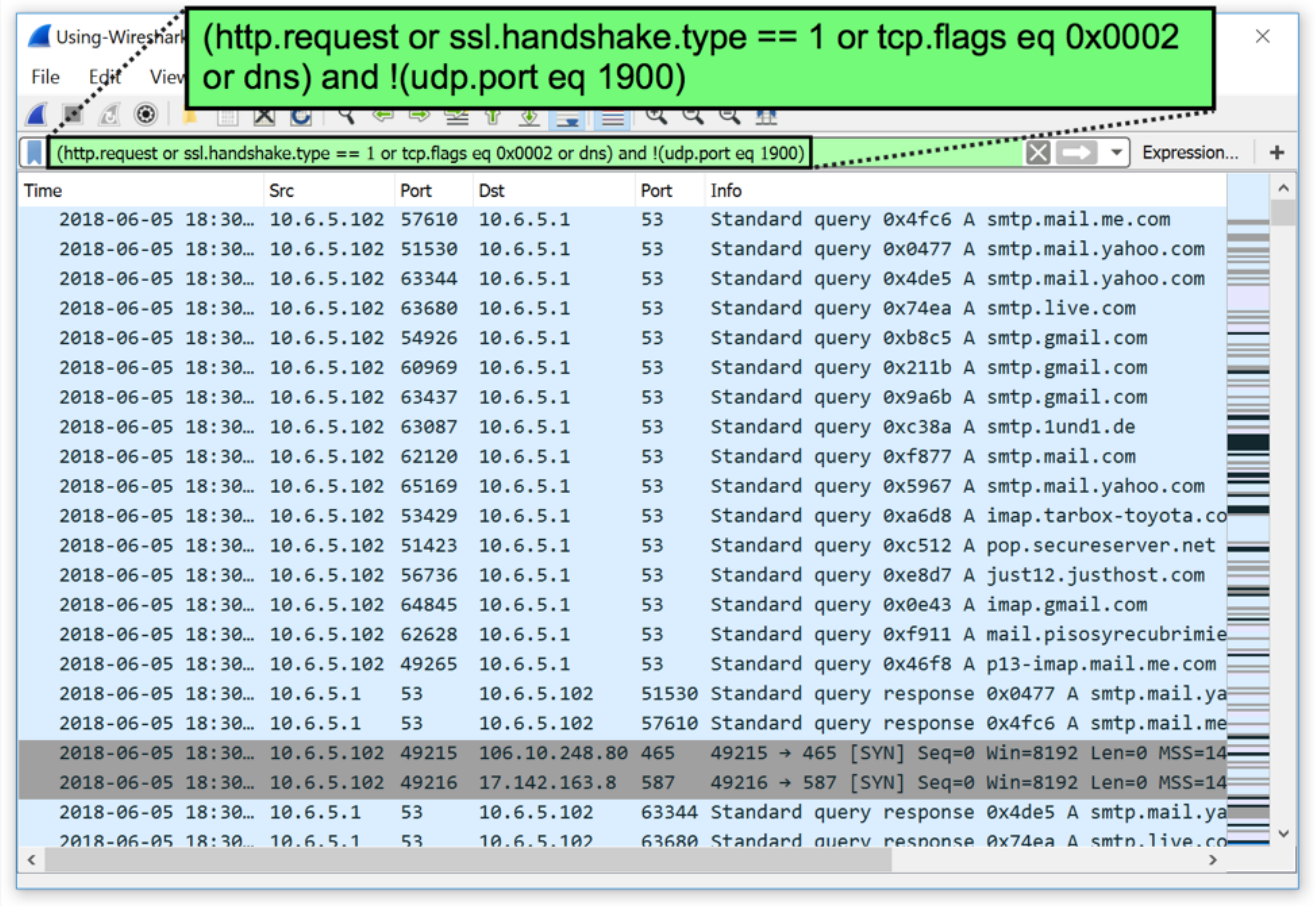
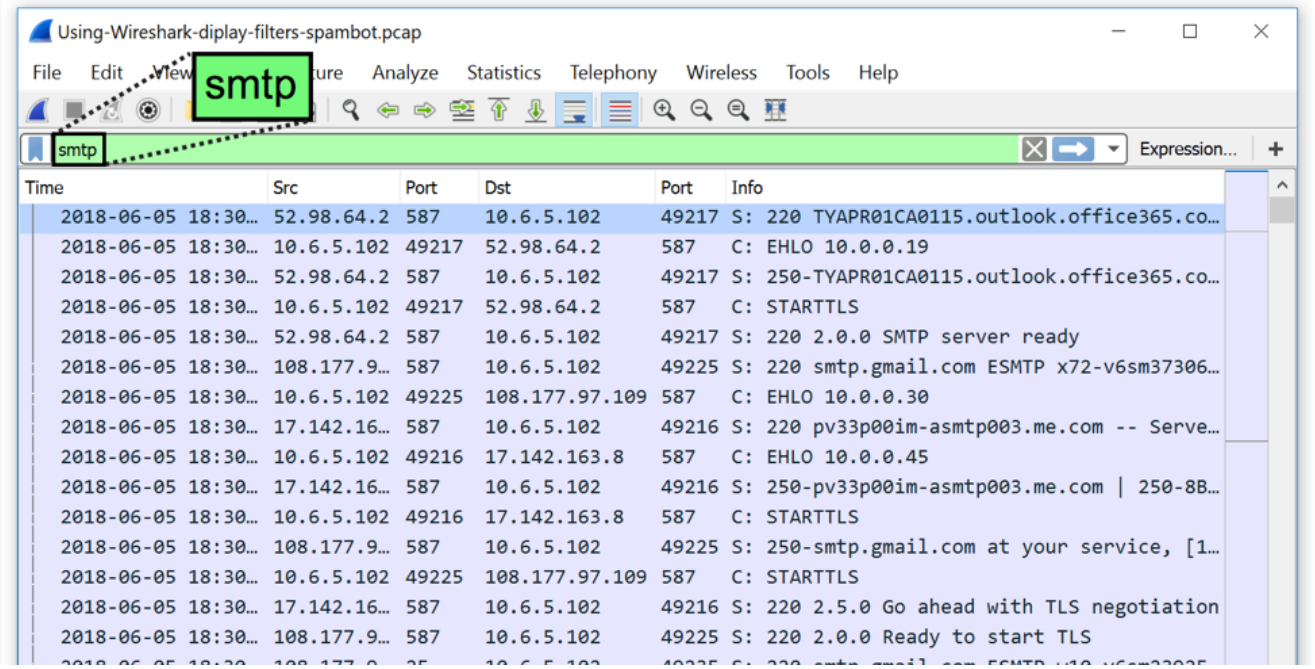


Figure 14. Wireshark filtered on spambot traffic to show DNS queries for various mail servers and TCP SYN packets to TCP ports 465 and 587 related to SMTP traffic.

If you use ***smtp*** as a filter expression, you'll find several results. In cases where you find STARTTLS, this will likely be encrypted SMTP traffic, and you will not be able to see the email data.



This site uses cookies essential to its operation, for analytics, and for personalized content and ads. By continuing to browse this site, you acknowledge the use of cookies. Privacy statement

› Cookie Settings

Figure 15. Filtering on SMTP traffic in Wireshark when viewing spambot traffic.

In recent years, email traffic from spambots is most likely encrypted SMTP. However, you might find unencrypted SMTP traffic by searching for strings in common email header lines like:

- ***smtp contains "From: "***
- ***smtp contains "Message-ID: "***
- ***smtp contains "Subject: "***

Keep in mind the Wireshark display filter is case-sensitive. When searching spambot traffic for unencrypted SMTP communications, I often use ***smtp contains "From: "*** as my filter expression as shown in Figure 16.

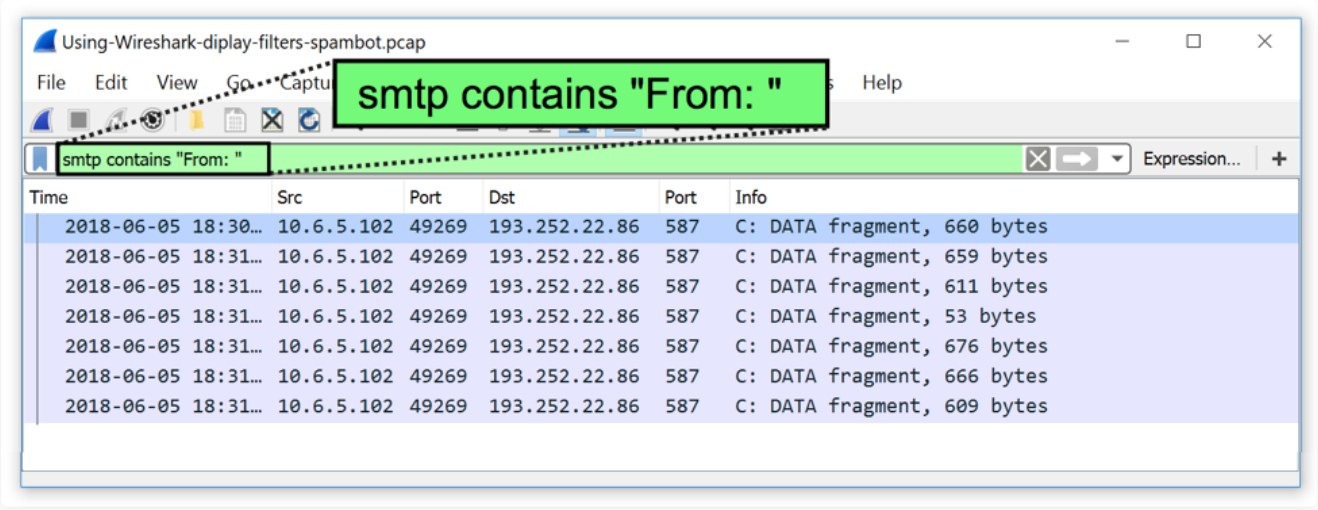


Figure 16. Filtering in Wireshark to find email header lines for unencrypted SMTP traffic.

After filtering for SMTP traffic as show in Figure 16, you can follow TCP stream for any of the displayed frames, and you'll find one of the emails sent from the spambot.

This site uses cookies essential to its operation, for analytics, and for personalized content and ads. By continuing to browse this site, you acknowledge the use of cookies. Privacy statement

› Cookie Settings

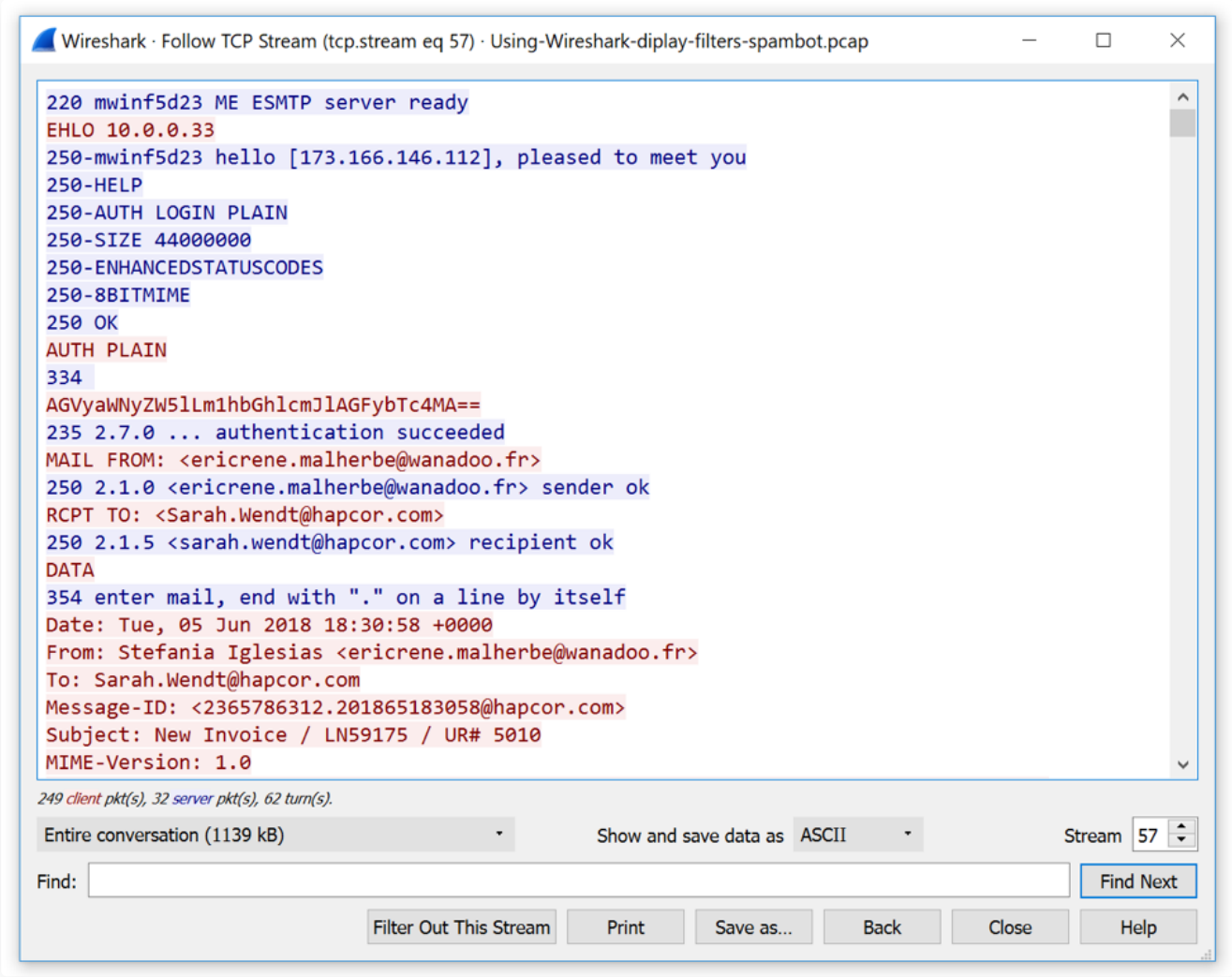
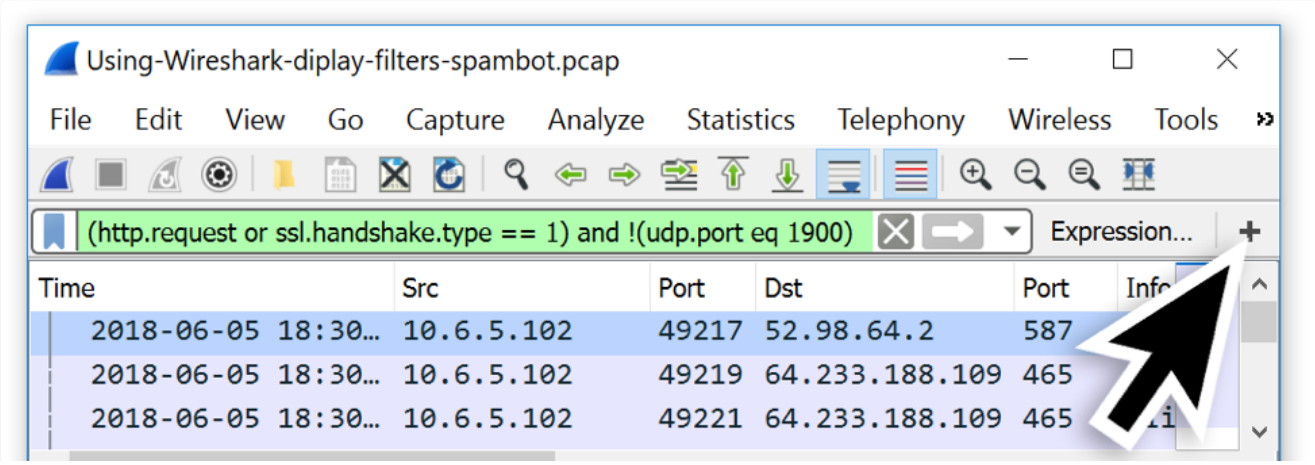


Figure 17. TCP stream showing unencrypted SMTP traffic from a spambot-infected host.

Saving Your Filters

Some filter expressions are very tedious to type out each time, but you can save them as filter buttons. On the right side of the Wireshark filter bar is a plus sign to add a filter button.



This site uses cookies essential to its operation, for analytics, and for personalized content and ads. By continuing to browse this site, you acknowledge the use of cookies. Privacy statement

Cookie Settings

Figure 18. Saving a filter expression as a filter expression button in Wireshark.

Click on that plus sign to save your expression as a filter button. You have the following fields:

- Label
- Filter
- Comment

The comment is optional, and the filter defaults to whatever is currently typed in the Wireshark filter bar. Once you have typed your label, then click the OK button as shown in Figure 19.

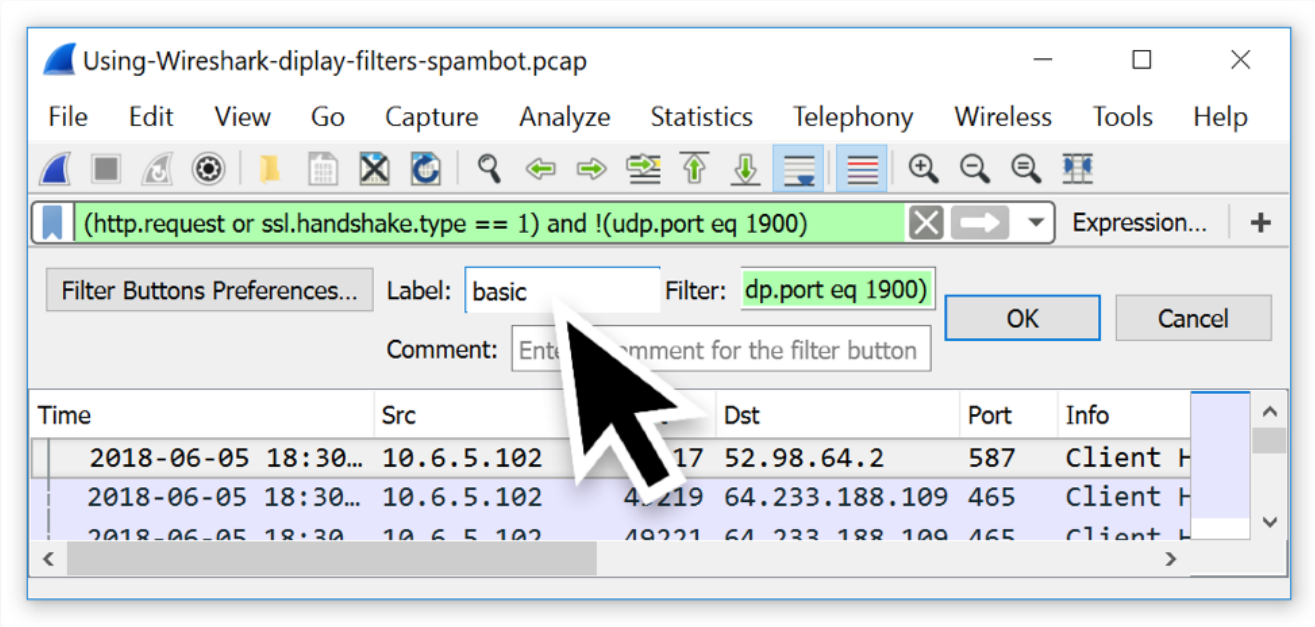
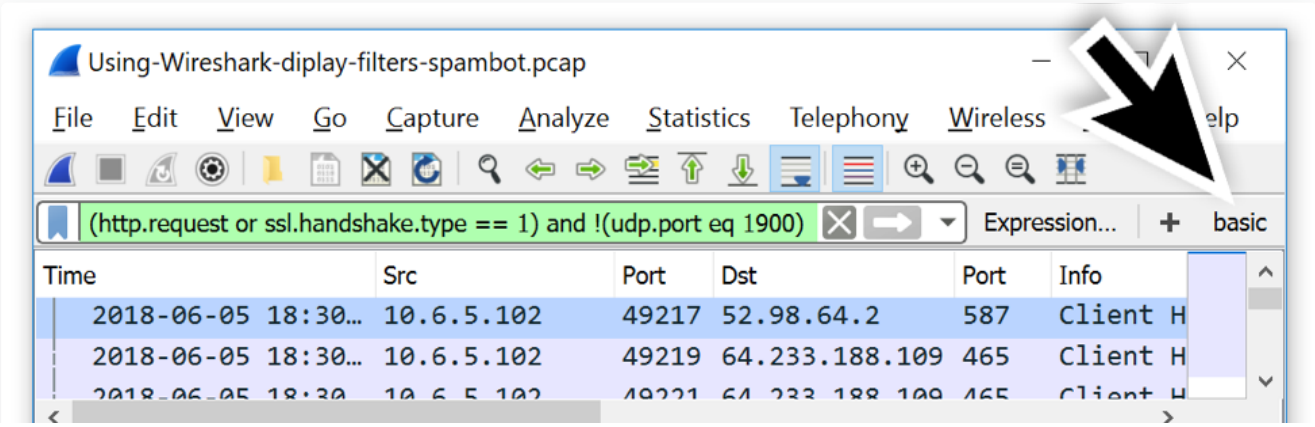


Figure 19. After typing a filter label, click the OK button.

In the above image, I typed "basic" for the filter ***(http.request or ssl.handshake.type == 1) and !(udp.port eq 1900)*** to save as my basic filter. Figure 20 show that filter button labeled "basic" to the right of the plus sign.



This site uses cookies essential to its operation, for analytics, and for personalized content and ads. By continuing to browse this site, you acknowledge the use of cookies. Privacy statement

Cookie Settings

For my normal filter setup in Wireshark, I create the following filter buttons:

- basic *(http.request or ssl.handshake.type == 1) and !(udp.port eq 1900)*
- basic+ *(http.request or ssl.handshake.type == 1 or tcp.flags eq 0x0002) and !(udp.port eq 1900)*
- basic+DNS *(http.request or ssl.handshake.type == 1 or tcp.flags eq 0x0002 or dns) and !(udp.port eq 1900)*

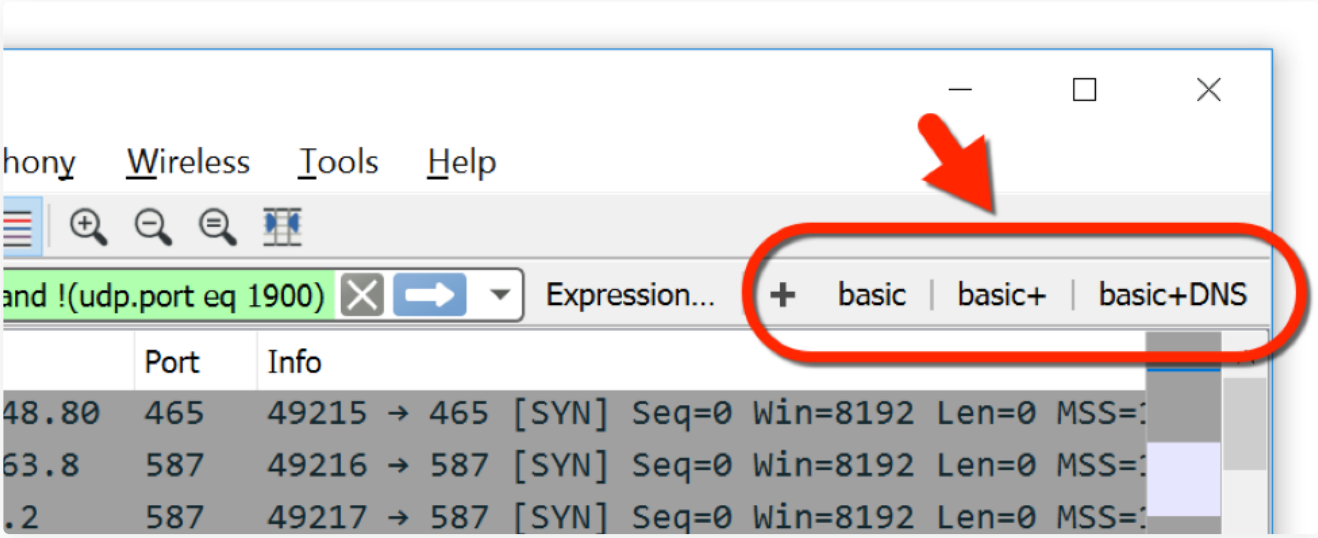


Figure 21. Filter buttons I routinely use on Wireshark.

Summary

This tutorial covered the following areas:

- Indicators of infection traffic
- The Wireshark display filter
- Filters for web-based infection traffic
- Filters for other types of infection traffic
- Saving your filters

Proper use of Wireshark display filters can help security professionals more efficiently investigate suspicious network traffic. Pcaps used in this tutorial can be found here.

Get updates from

This site uses cookies essential to its operation, for analytics, and for personalized content and ads. By continuing to browse this site, you acknowledge the use of cookies. Privacy statement

› Cookie Settings

Sign up to receive the latest news, cyber threat intelligence and research from us

Email address

Subscribe

I'm not a robot

reCAPTCHA
Privacy - Terms

By submitting this form, you agree to our Terms of Use and acknowledge our Privacy Statement.



Popular Resources

Resource Center

Blog

Communities

Tech Docs

Unit 42

Sitemap

Legal Notices

Privacy

Terms of Use

Documents

Account

Manage Subscriptions

Report a Vulnerability

This site uses cookies essential to its operation, for analytics, and for personalized content and ads. By continuing to browse this site, you acknowledge the use of cookies. Privacy statement

› Cookie Settings