



Menu

Wireshark Tutorial: Decrypting HTTPS Traffic

16,057 people reacted

51

7 min. read

SHARE



By Brad Duncan

August 21, 2020 at 6:00 AM

Category: Unit 42

Tags: tutorial, Wireshark, Wireshark Tutorial

Time	Dst	Port	Host	Server Name	Info
2018-08-03 19:06:20	192.0.79.32	80	college.usatoday.com		GET /2017/03/01/
2018-08-03 19:06:20	192.0.78.19	443		r-login.wordpress.com	Client Hello
2018-08-03 19:06:20	192.0.78.19	443		r-login.wordpress.com	Client Hello
2018-08-03 19:06:20	192.0.77.32	443		s2.wp.com	Client Hello
2018-08-03 19:06:20	192.0.77.32	443		s2.wp.com	Client Hello
2018-08-03 19:06:20	192.0.77.32	443		s2.wp.com	Client Hello
2018-08-03 19:06:20	192.0.77.32	443		s2.wp.com	Client Hello
2018-08-03 19:06:20	192.0.77.32	443		s1.wp.com	Client Hello
2018-08-03 19:06:20	192.0.77.32	443		s1.wp.com	Client Hello
2018-08-03 19:06:20	216.58.218...	443		fonts.googleapis.com	Client Hello
2018-08-03 19:06:20	216.58.218...	443		fonts.googleapis.com	Client Hello
2018-08-03 19:06:20	52.84.125...	80	d15krst4gi8g86.clou...		GET /css/usatoda
2018-08-03 19:06:20	52.84.125...	80	d15krst4gi8g86.clou...		GET /js/script.j
2018-08-03 19:06:20	57.94.125...	80	d15krst4gi8g86.clou...		GET /free/mastod...

This post is also available in: [日本語 \(Japanese\)](#)

Executive Summary

This tutorial is designed for security professionals who investigate suspicious network activity and review packet captures (pcaps) of the traffic. The instructions assume you are familiar with [Wireshark](#), and it focuses on Wireshark version 3.x.

When reviewing suspicious network activity, we often run across encrypted traffic. Why? Because most websites use the Hypertext Transfer Protocol Secure (HTTPS) protocol. But like most websites, various types of malware also use HTTPS. When reviewing pcaps from malware activity, it's very helpful to know what's contained within post-infection traffic.

This Wireshark tutorial describes how to decrypt HTTPS traffic from a pcap in Wireshark. Decryption is possible with a text-based log containing encryption key data captured when the pcap was originally recorded. With this key log file, we can decrypt HTTPS activity in a pcap and review its contents.

Today, we will examine HTTPS activity from a [Dridex](#) malware infection.

Note: Our instructions assume you have customized your Wireshark column display as previously described in ["Customizing Wireshark – Changing Your Column Display."](#).

Here is a Github repository with a ZIP archive containing the pcap and a key log file used for this tutorial.

Warning: The pcap used for this tutorial contains Windows-based malware. There is a risk of infection if using a Windows computer. We recommend you review this pcap in a non-Windows environment like BSD, Linux or macOS if at all possible.

The Context Behind Encrypted Traffic

In the mid- to late-1990s, the most common protocol used by websites was Hypertext Transfer Protocol (HTTP), which generated unencrypted web traffic. However, as security became an increasing concern, websites started switching to HTTPS, and now we rarely see HTTP traffic from web browsing.

HTTPS is essentially an encrypted communications tunnel containing HTTP traffic. These tunnels first used Secure Sockets Layer (SSL) as an encryption protocol. Today most HTTPS traffic uses Transport Layer Security (TLS).

HTTPS Web Traffic

HTTPS traffic often reveals a domain name. For example, when viewing <https://www.wireshark.org> in a web browser, a pcap would show www.wireshark.org as

the server name for this traffic when viewed in a **customized Wireshark column display**. Unfortunately, we don't know other details like the actual URL or data returned from the server. Following the Transmission Control Protocol (TCP) stream from a pcap will not reveal the content of this traffic because it is encrypted.

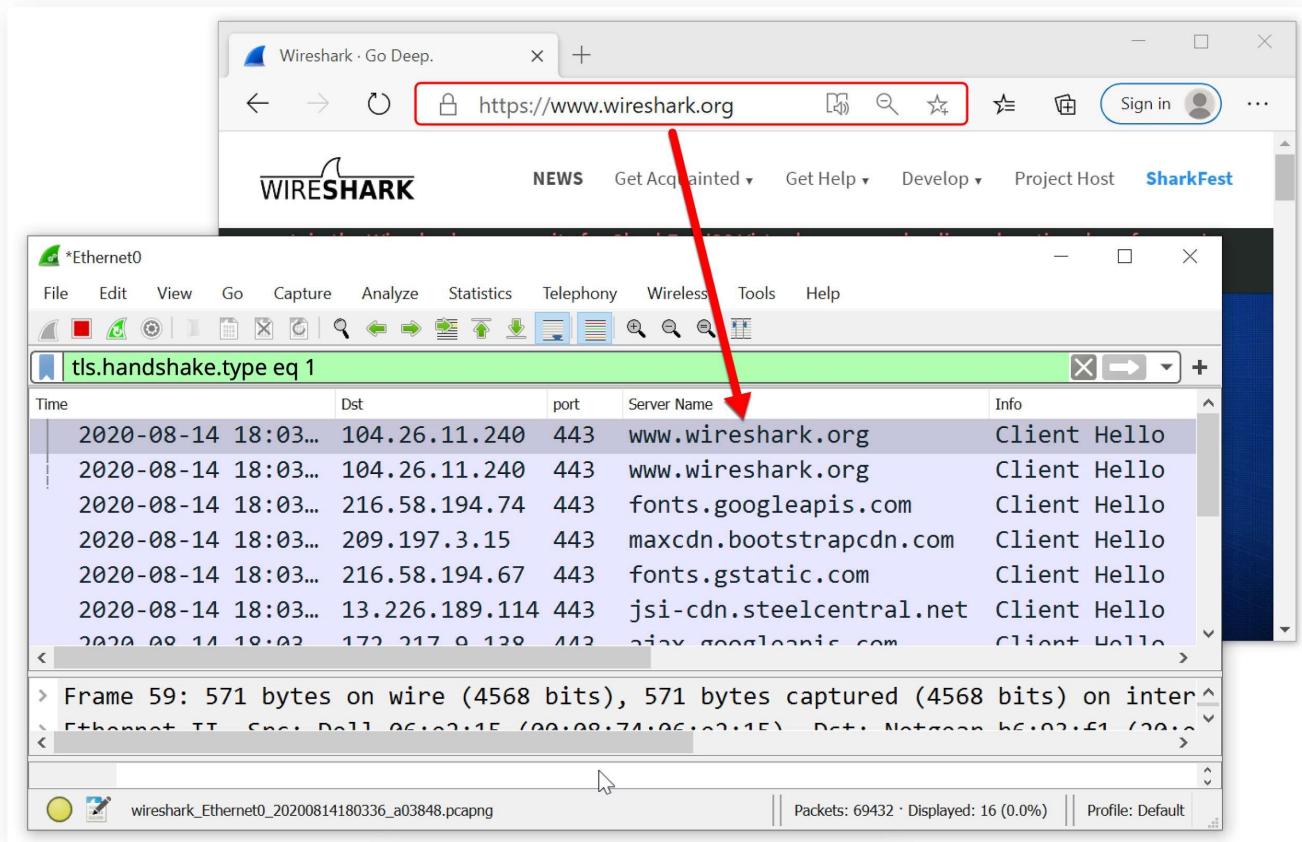


Figure 1. Traffic from HTTPS traffic to www.wireshark.org.

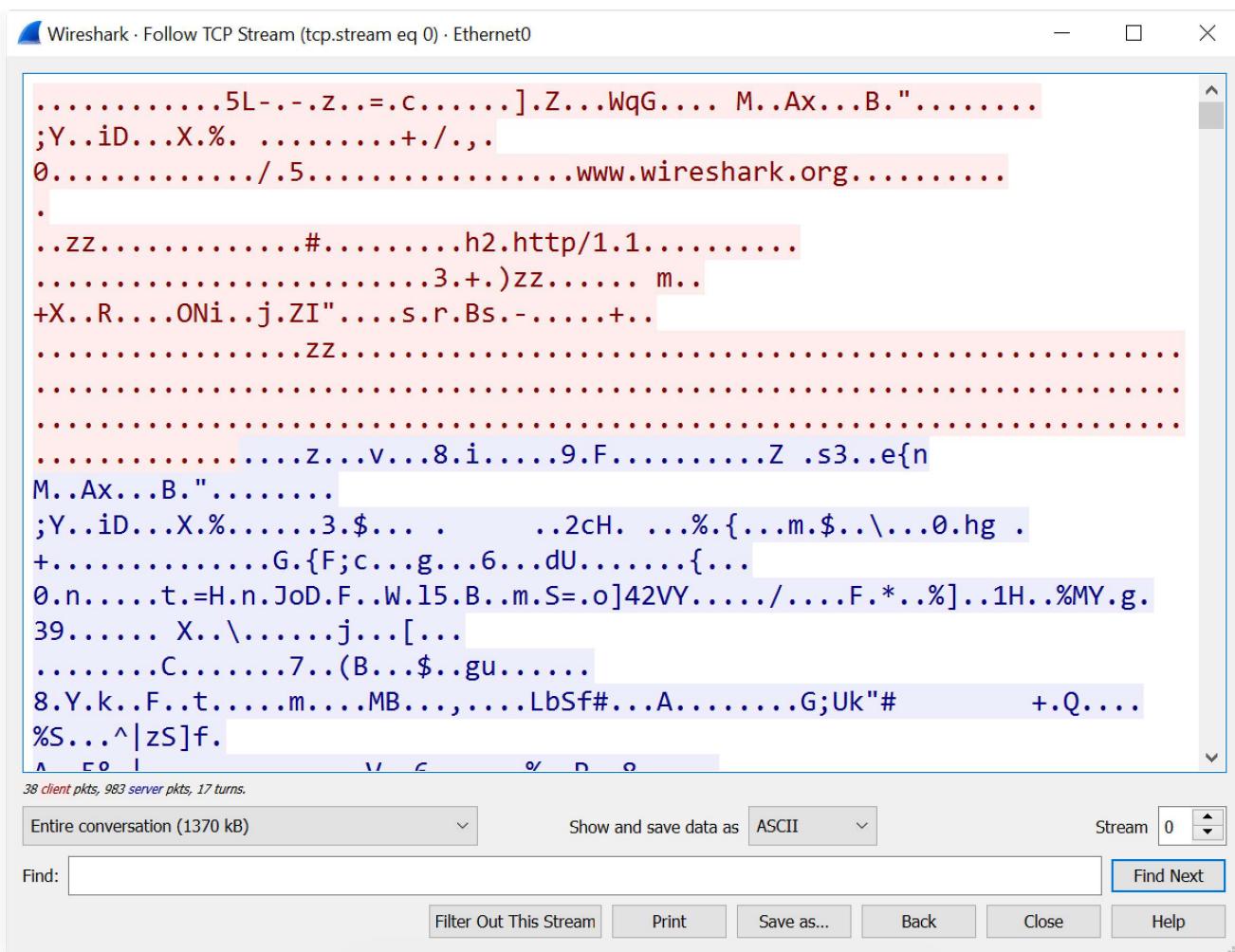


Figure 2. TCP stream of HTTPS traffic to and from server at www.wireshark.org.

Encryption Key Log File

An encryption key log is a text file. An example is shown in Figure 3.

```
Wireshark-tutorial-KeysLogFile.txt - Mousepad
File Edit Search View Document Help
CLIENT_RANDOM 5e85016c2010478311178455b55ee4c6cf4fee8ef941b1a3fa07c3b3f86365c d6b14d25
CLIENT_RANDOM 5e850177170b9e2362c1c72d0dbc4a9133441079bf5ce0cbe2909016d120967a 63c841f6
CLIENT_RANDOM 5e850178c6e06ad9d7eb8ac04fea8d4d4e5f58fe3008dbc795cd46688a85c921 62d5fdb9
CLIENT_RANDOM 5e8501dceb9af936b5e346dc4ac6665f8dc610b37d1a24386fe1a4d67298cbe3 f1b55a36
CLIENT_RANDOM 5e85026cf7611ba445bcdee52a958fe86a531cc3e8e26cb4bcde4d2cdaf79e8 5ffc09c7
CLIENT_RANDOM 5e850391f7b8619e1c8f65c12cfc0731816248133a0c89feb9211cc9e3943747 f0402c77
CLIENT_RANDOM 5e85041398abd602bad132952274a94371868b8dca23e145ceb7ec59562a51b2 937cce1a
CLIENT_RANDOM 5e85049b139b551f1ceba4372df8ed41602b0ca006251dd3283c38ddbf7d2782 a1d744fa
```

Figure 3. The key log file used in this tutorial.

These logs are created using a Man in the Middle (MitM) technique when the pcap is originally recorded. If no such file was created when the pcap was recorded, you cannot decrypt HTTPS traffic in that pcap.

Example of a Pcap With a Key Log File

A password-protected ZIP archive containing the pcap and its key log file is available at [this Github repository](#). Go to the Github page, click on the ZIP archive entry, then download it as shown in Figures 4 and 5. Of note, the pcap contained in this ZIP archive provides access to a Windows-based malware sample when decrypted with the key log. As always, we recommend you exercise caution and follow steps from this tutorial in a non-Windows environment.

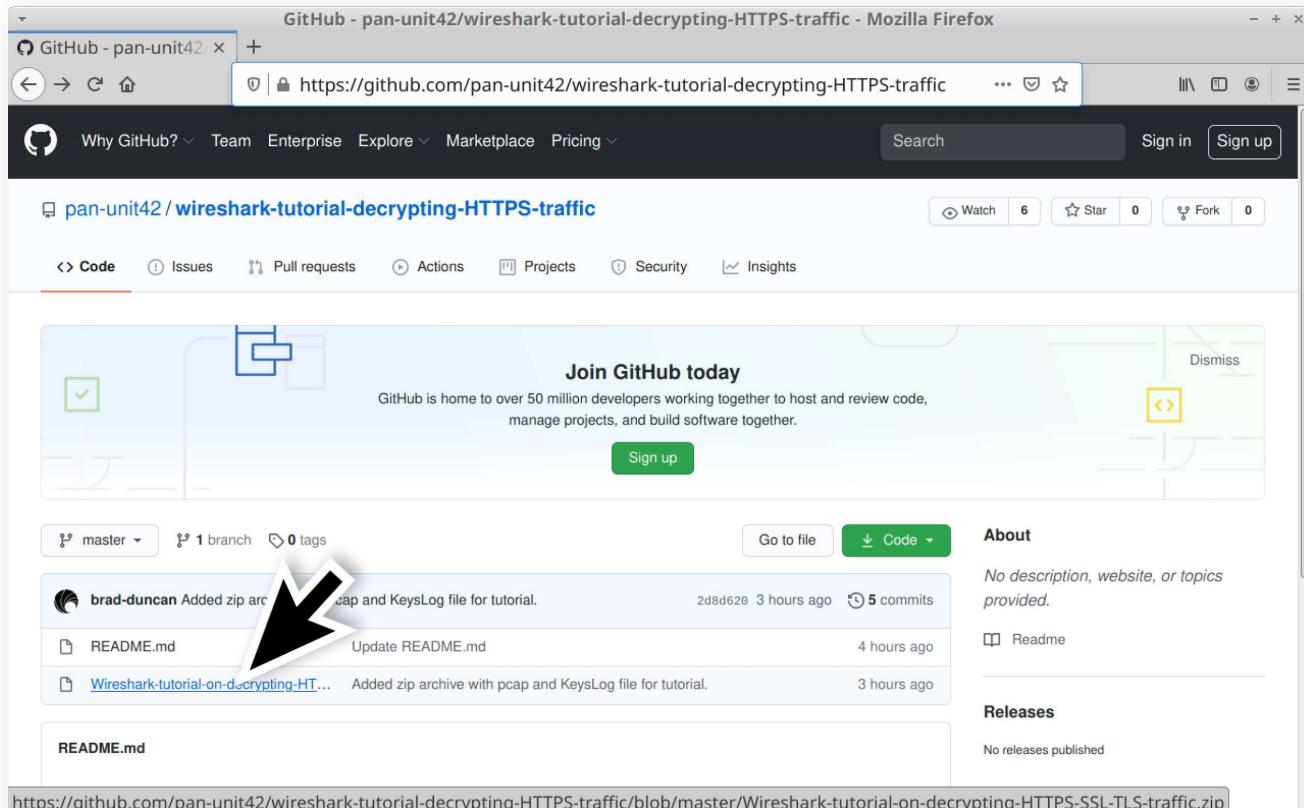


Figure 4. Github repository with link to ZIP archive used for this tutorial.

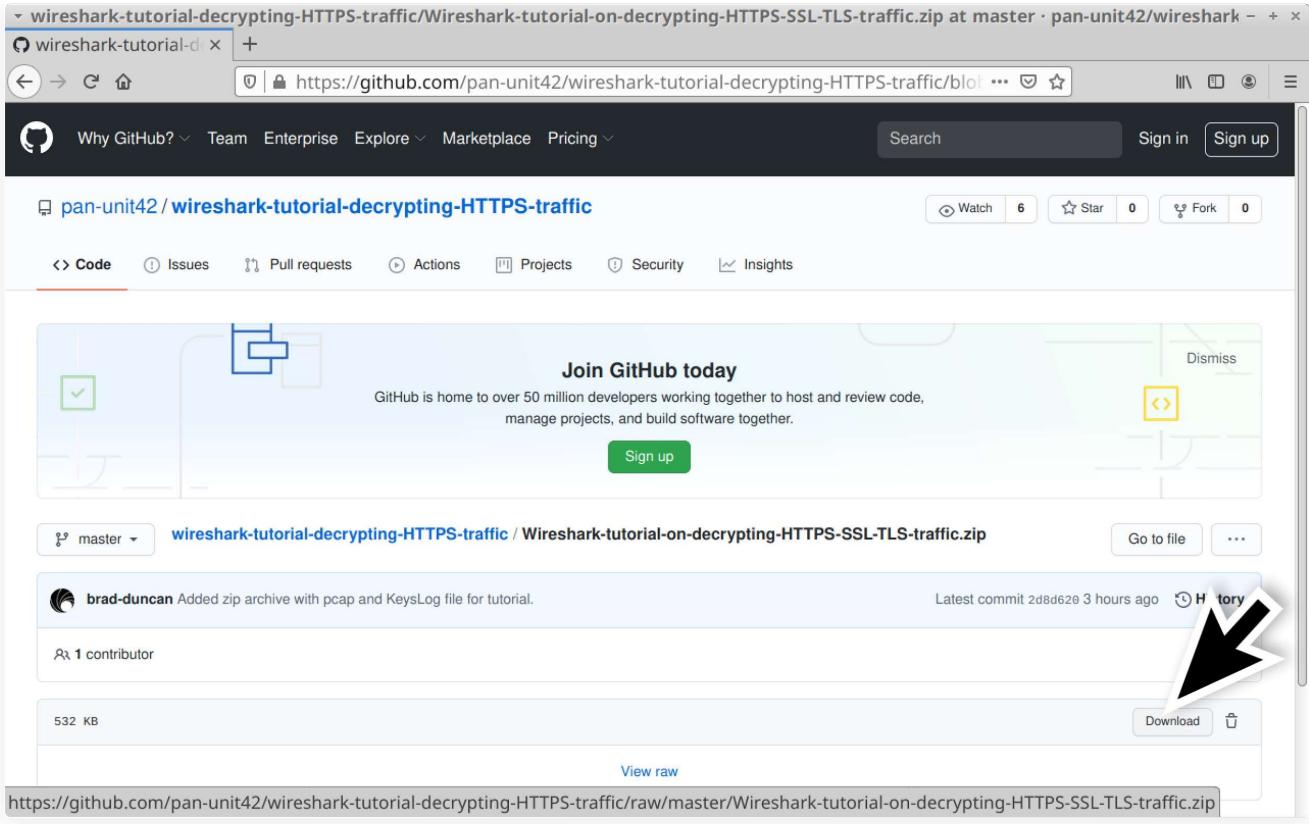


Figure 5. Downloading the ZIP archive for this tutorial.

Use **infected** as the password to extract the pcap and key log file from the ZIP archive. This will provide two files as shown in Figure 6:

- Wireshark-tutorial-KeysLogFile.txt
- Wireshark-tutorial-on-decrypting-HTTPS-SSL-TLS-traffic.pcap

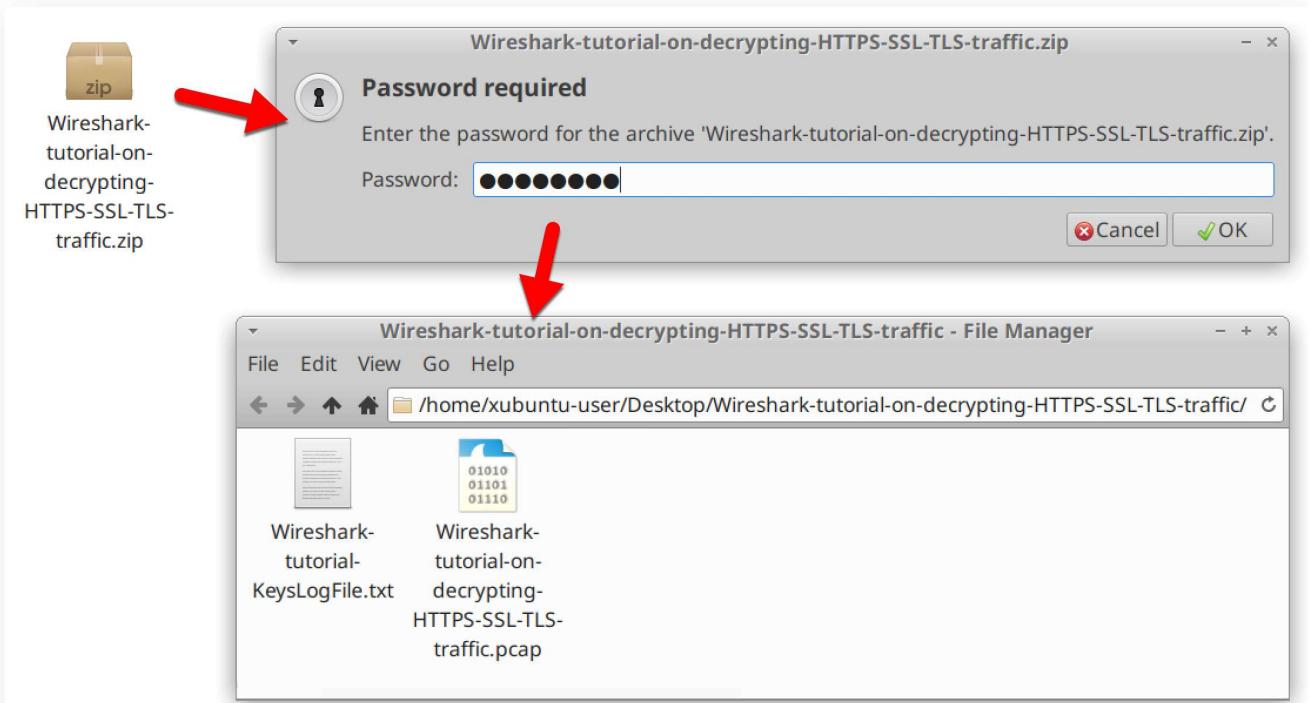


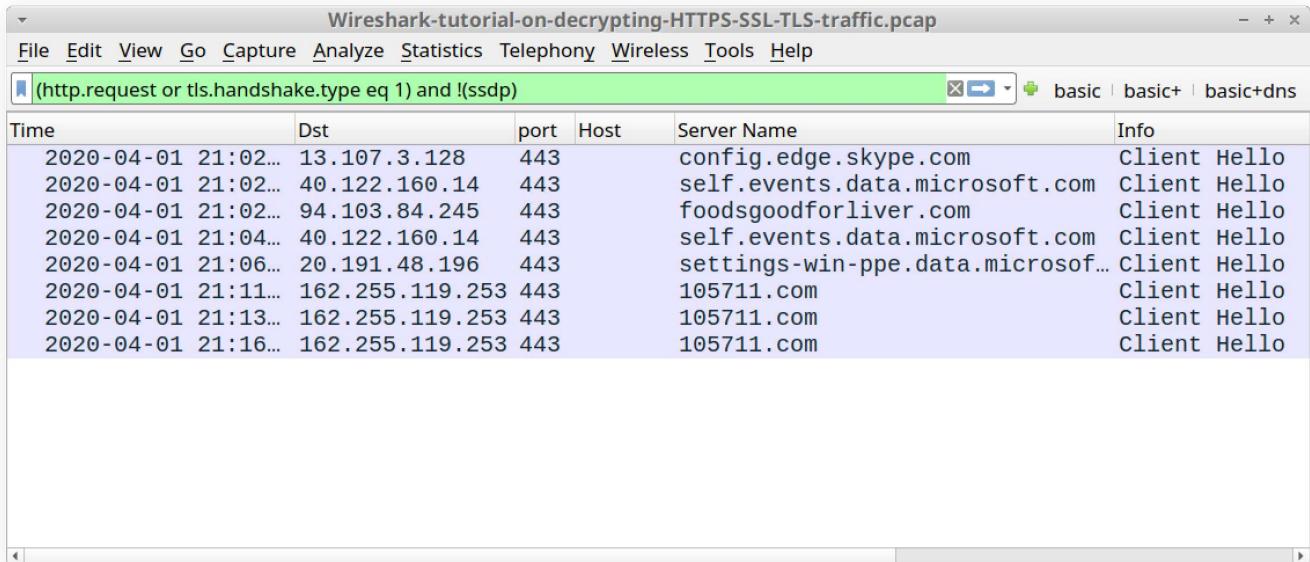
Figure 6. Key log file and pcap for this tutorial.

HTTPS Traffic Without the Key Log File

Open [Wireshark-tutorial-on-decrypting-HTTPS-SSL-TLS-traffic.pcap](#) in Wireshark. Use a basic web filter as described in this previous [tutorial about Wireshark filters](#). Our basic filter for Wireshark 3.x is:

```
(http.request or tls.handshake.type eq 1) and !(ssdp)
```

This pcap is from a Dridex malware infection on a Windows 10 host. All web traffic, including the infection activity, is HTTPS. Without the key log file, we cannot see any details of the traffic, just the IP addresses, TCP ports and domain names, as shown in Figure 7.



Time	Dst	port	Host	Server Name	Info
2020-04-01 21:02...	13.107.3.128	443		config.edge.skype.com	Client Hello
2020-04-01 21:02...	40.122.160.14	443		self.events.data.microsoft.com	Client Hello
2020-04-01 21:02...	94.103.84.245	443		foodsgoodforliver.com	Client Hello
2020-04-01 21:04...	40.122.160.14	443		self.events.data.microsoft.com	Client Hello
2020-04-01 21:06...	20.191.48.196	443		settings-win-ppe.data.microsoft...	Client Hello
2020-04-01 21:11...	162.255.119.253	443		105711.com	Client Hello
2020-04-01 21:13...	162.255.119.253	443		105711.com	Client Hello
2020-04-01 21:16...	162.255.119.253	443		105711.com	Client Hello

Figure 7. Viewing the pcap in Wireshark using the basic web filter without any decryption.

Loading the Key Log File

Open [Wireshark-tutorial-on-decrypting-HTTPS-SSL-TLS-traffic.pcap](#) in Wireshark. Then use the menu path **Edit -> Preferences** to bring up the Preferences Menu, as shown in Figure 8.

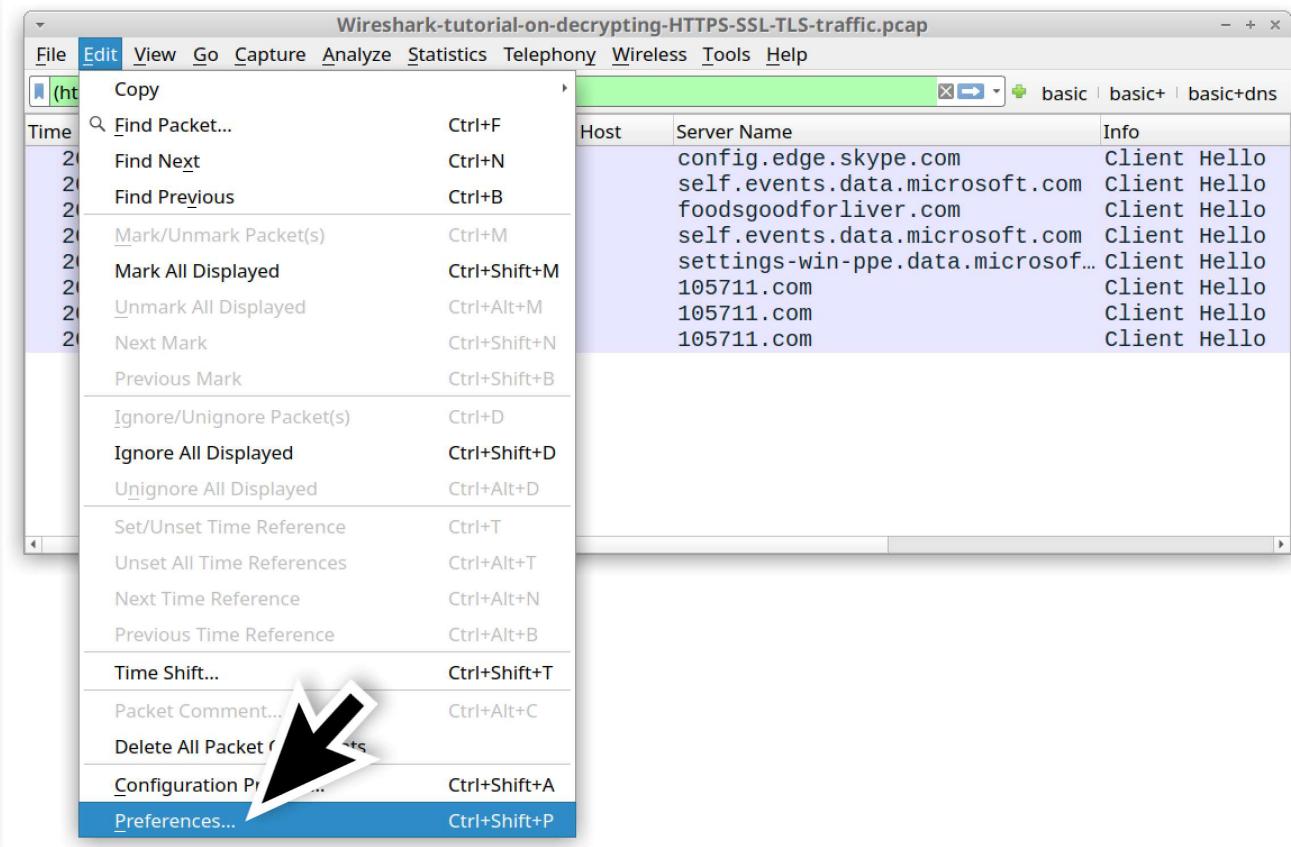


Figure 8. Getting to the Preferences Menu in Wireshark.

On the left side of the Preferences Menu, click on Protocols, as shown in Figure 9.

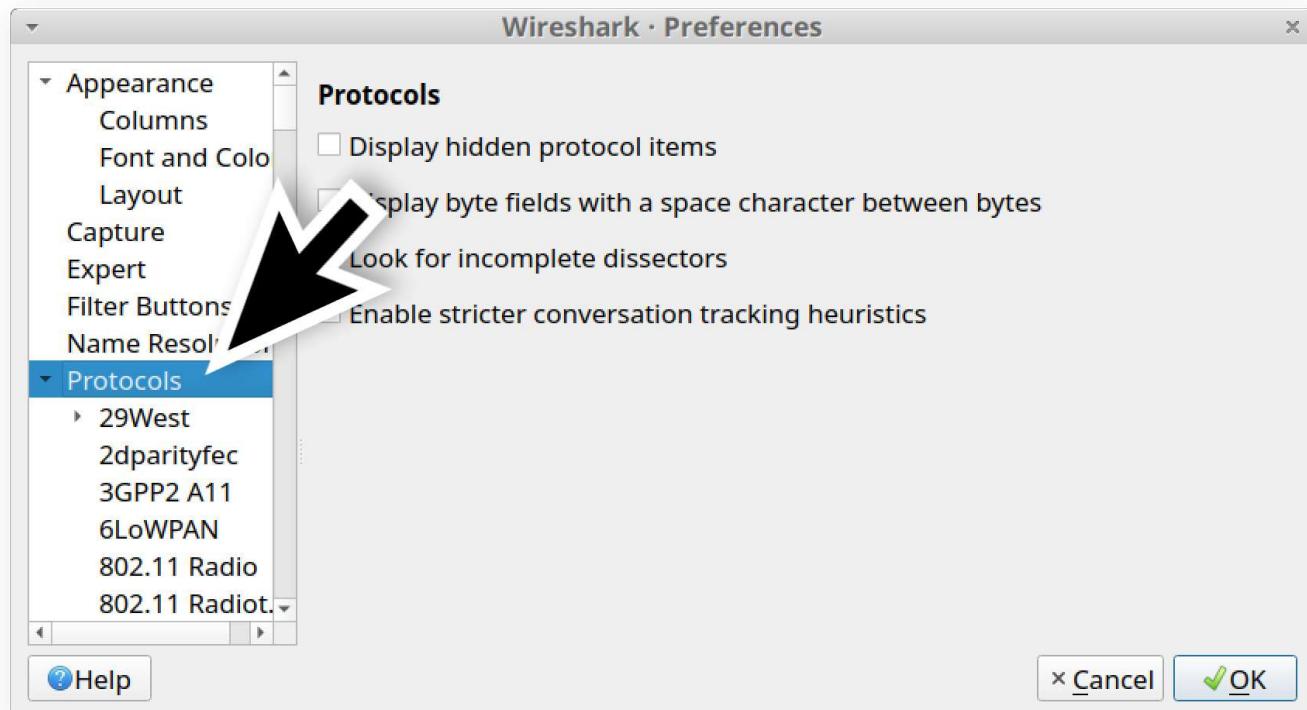


Figure 9. Selecting Protocols in the Preferences Menu.

If you are using Wireshark version 2.x, scroll down until you find **SSL** and select it. If you are using Wireshark version 3.x, scroll down to **TLS** and select it. Once you have selected SSL or TLS, you

should see a line for **(Pre)-Master-Secret log filename**. Click on the “Browse” button and select our key log file named **Wireshark-tutorial-KeysLogFile.txt**, as shown in Figures 10, 11 and 12.

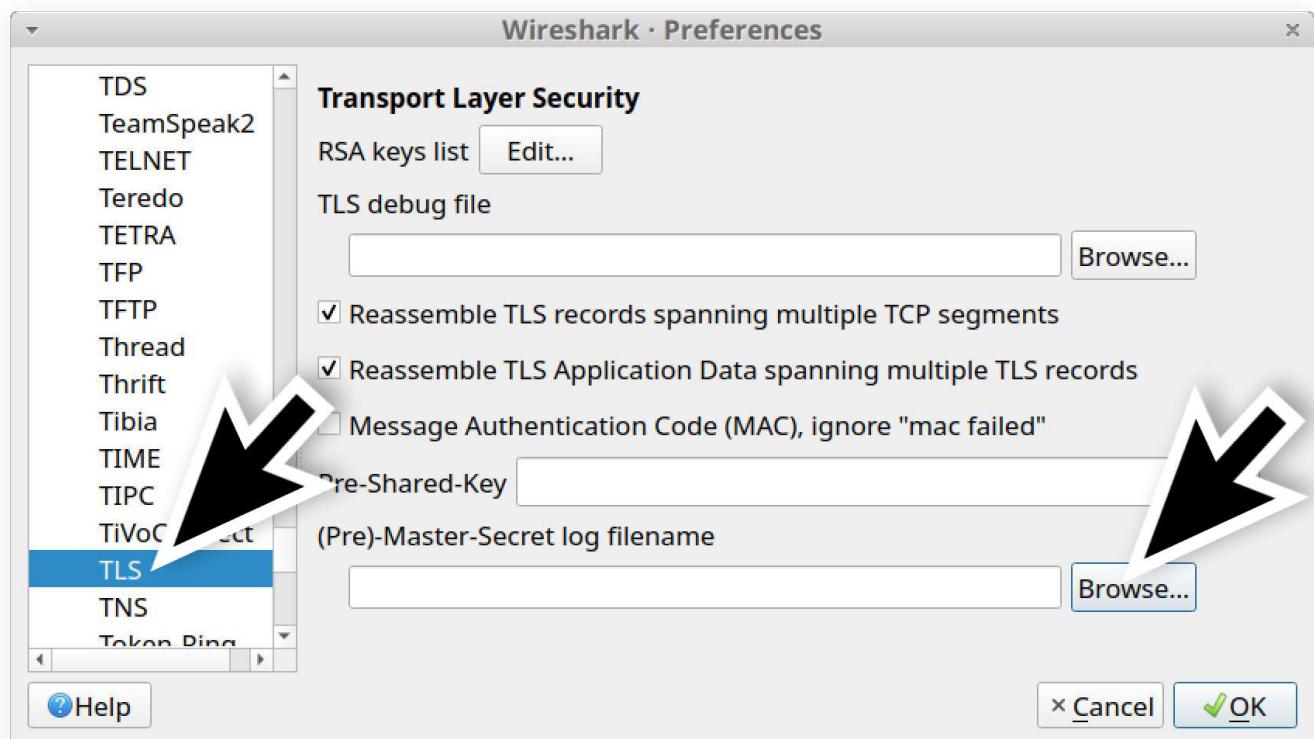


Figure 10. Finding the (Pre)-Master-Secret log filename field under TLS in Wireshark 3.x.

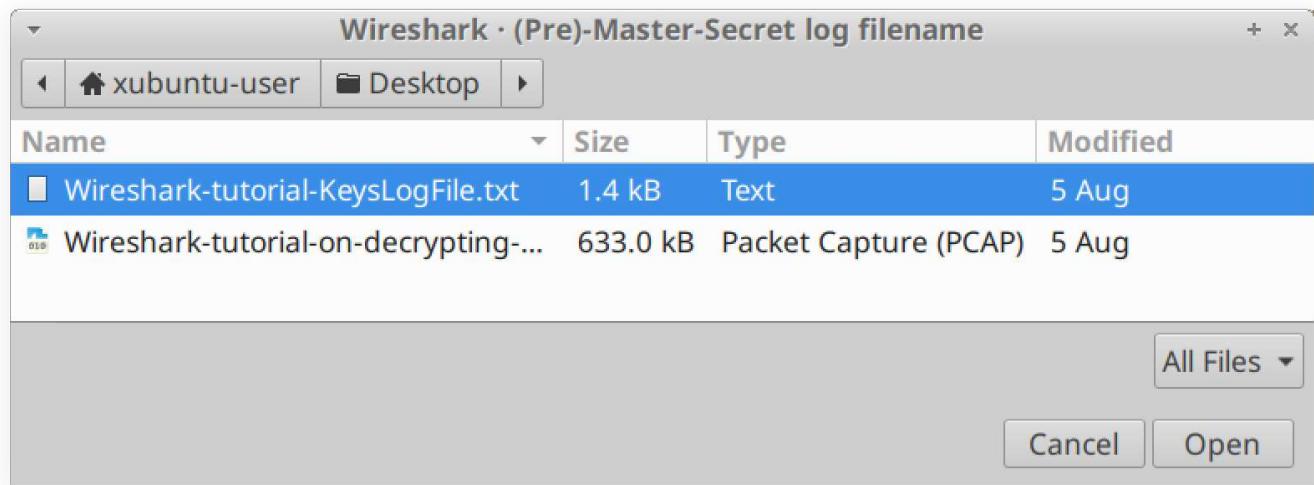


Figure 11. Selecting our key log file for this tutorial.

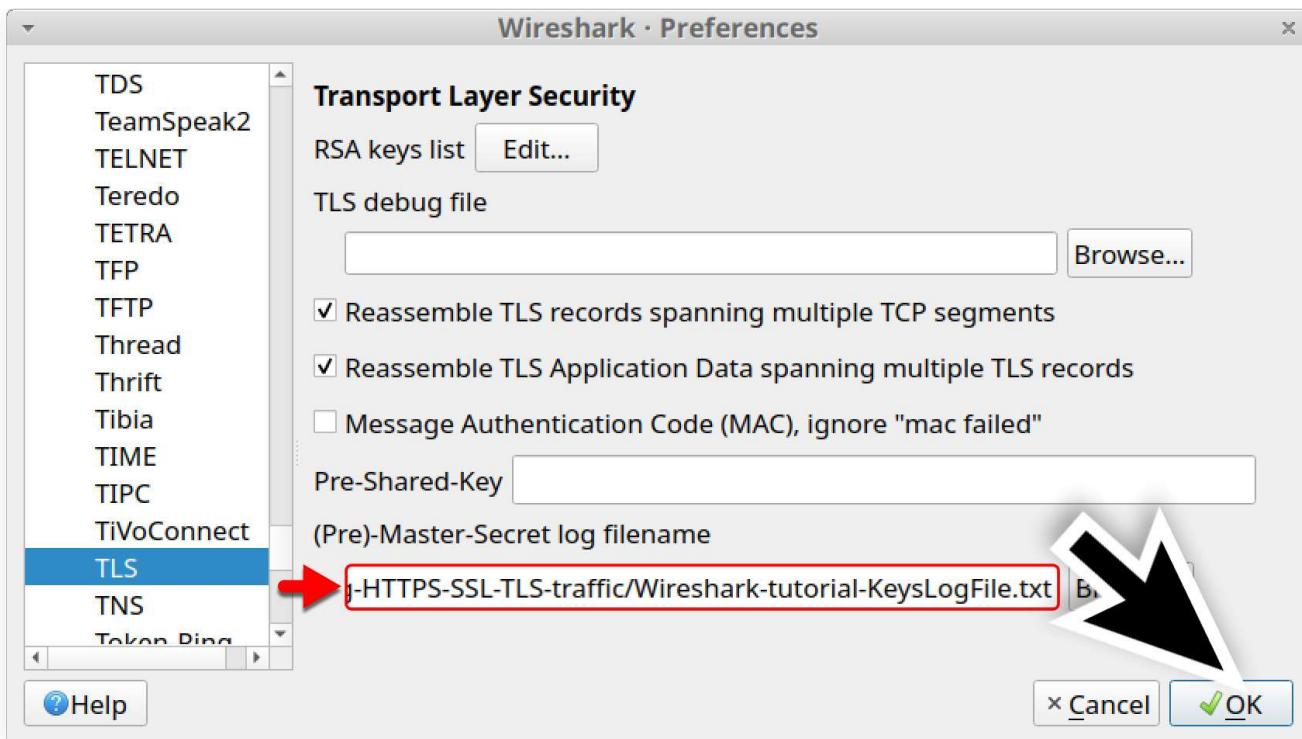


Figure 12. Once the file has been selected as the (Pre)-Master-Secret log filename, click “OK.”

HTTPS Traffic With the Key Log File

Once you have clicked “OK,” when using the basic filter, your Wireshark column display will list the decrypted HTTP requests under each of the HTTPS lines, as shown in Figure 13.

Wireshark-tutorial-on-decrypting-HTTPS-SSL-TLS-traffic.pcap					
Time	Dst	port	Host	Server Name	Info
2020-04-01 21:02...	13.107.3.128	443		config.edge.skype.com	Client Hello
2020-04-01 21:02...	13.107.3.128	443	config.edge.skype.com		GET /config/v2/Office/wor...
2020-04-01 21:02...	40.122.160.14	443		self.events.data.mic...	Client Hello
2020-04-01 21:02...	40.122.160.14	443	self.events.data.mi...		POST /OneCollector/1.0/...
2020-04-01 21:02...	40.122.160.14	443	self.events.data.mi...		POST /OneCollector/1.0/...
2020-04-01 21:02...	94.103.84.245	443		foodsgoodforliver.com	Client Hello
2020-04-01 21:02...	94.103.84.245	443	foodsgoodforliver.com		GET /invest_20.dll HTTP/1...
2020-04-01 21:04...	40.122.160.14	443		self.events.data.mic...	Client Hello
2020-04-01 21:04...	40.122.160.14	443	self.events.data.mi...		POST /OneCollector/1.0/...
2020-04-01 21:06...	20.191.48.196	443		settings-win-ppe.data...	Client Hello
2020-04-01 21:06...	20.191.48.196	443	settings-win-ppe.da...		GET /settings/v2.0/Stora...
2020-04-01 21:11...	162.255.119.253	443		105711.com	Client Hello
2020-04-01 21:11...	162.255.119.253	443	105711.com		POST /docs.php HTTP/1.1
2020-04-01 21:13...	162.255.119.253	443	105711.com		Client Hello
2020-04-01 21:13...	162.255.119.253	443	105711.com		POST /docs.php HTTP/1.1
2020-04-01 21:16...	162.255.119.253	443	105711.com		Client Hello
2020-04-01 21:16...	162.255.119.253	443	105711.com		POST /docs.php HTTP/1.1

Figure 13. HTTPS decryption in Wireshark after using the key log file.

In this pcap, we now see HTTP requests to `microsoft.com` and `skype.com` domains previously hidden in the HTTPS traffic. We also find the following traffic caused by the Dridex infection:

- `foodsgoodforliver[.]com` – GET /invest_20.dll

- 105711[.]com - POST /docs.php

The GET request to `foodsgoodforliver[.]com` returned a DLL file for Dridex. The POST requests to `105711[.]com` are command and control (C2) traffic from the Dridex-infected Windows host.

We can review the traffic by following HTTP streams. Right-click on the line to select it, then left-click to bring up a menu to follow the HTTP stream. Figures 14 and 15 show following the HTTP stream for the HTTP GET request to `foodsgoodforliver[.]com`.

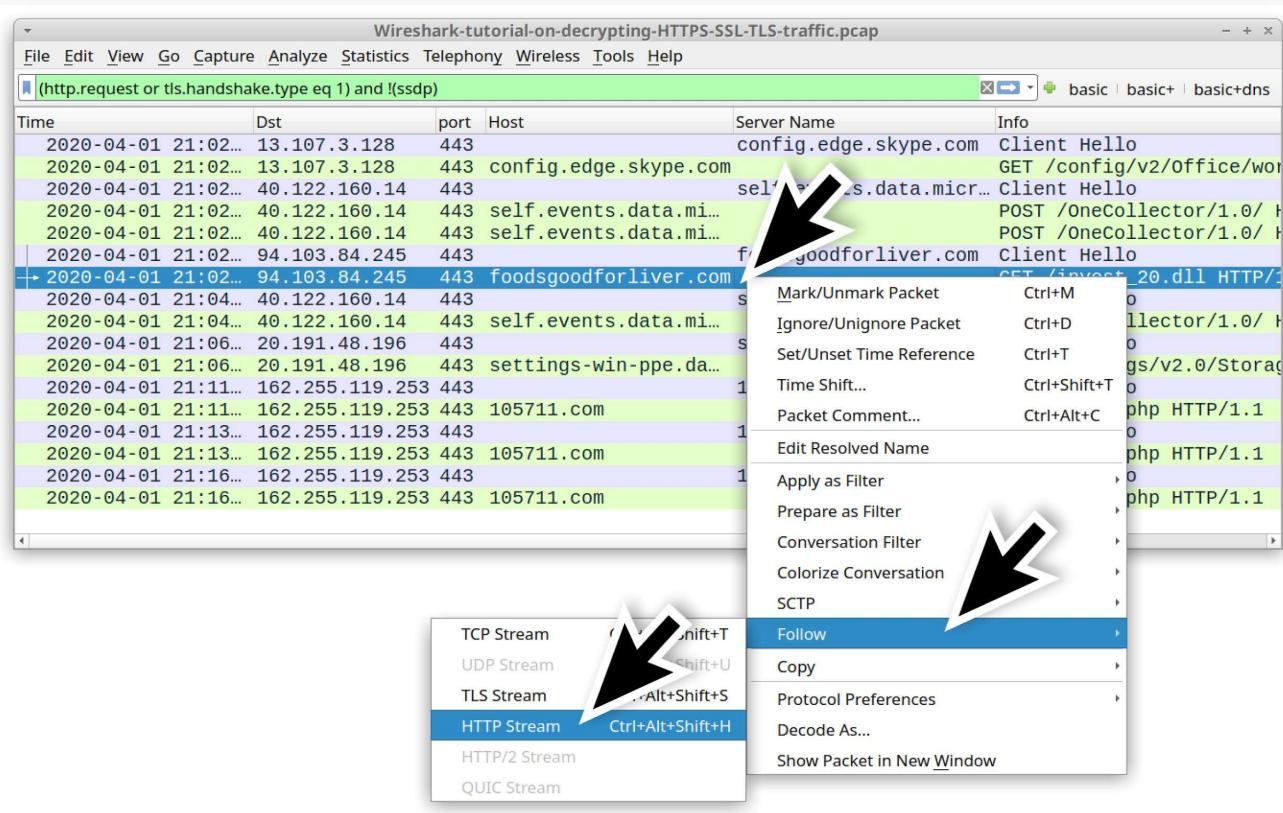
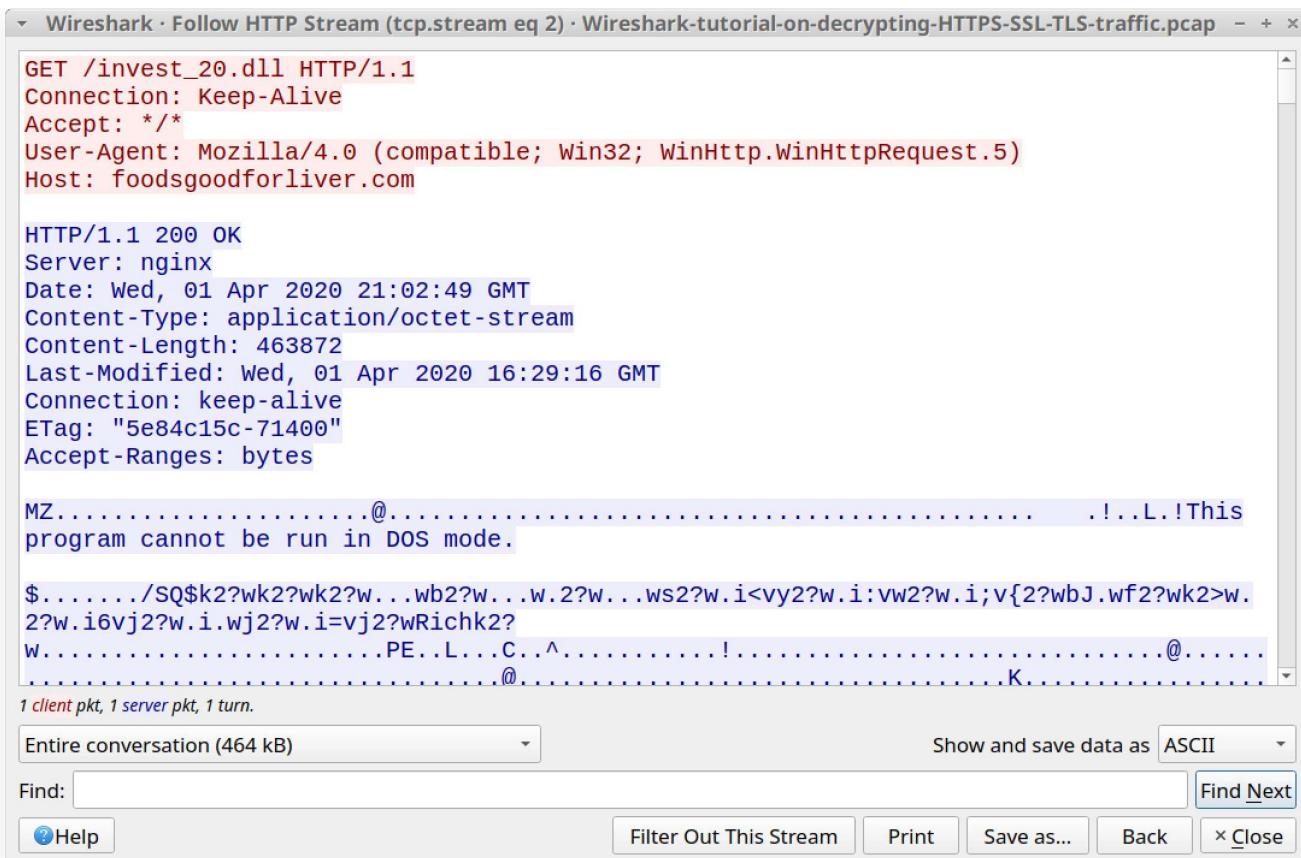


Figure 14. Following HTTP stream for the GET request to `foodsgoodforliver[.]com`.



```

GET /invest_20.dll HTTP/1.1
Connection: Keep-Alive
Accept: */*
User-Agent: Mozilla/4.0 (compatible; Win32; WinHttp.WinHttpRequest.5)
Host: foodsgoodforliver.com

HTTP/1.1 200 OK
Server: nginx
Date: Wed, 01 Apr 2020 21:02:49 GMT
Content-Type: application/octet-stream
Content-Length: 463872
Last-Modified: Wed, 01 Apr 2020 16:29:16 GMT
Connection: keep-alive
ETag: "5e84c15c-71400"
Accept-Ranges: bytes

MZ.....@..... !..L.!This
program cannot be run in DOS mode.

$...../SQ$k2?wk2?wk2?w...wb2?w...w.2?w...ws2?w.i<vy2?w.i:vw2?w.i;v{2?wbJ.wf2?wk2>w.
2?w.i6vj2?w.i.wj2?w.i=vj2?wRichk2?
W.....PE..L..C..^.....!.....@.....K.
.....@.....K.

1 client pkt, 1 server pkt, 1 turn.

Entire conversation (464 kB) Show and save data as ASCII
Find: Find Next
Help Filter Out This Stream Print Save as... Back Close

```

Figure 15. HTTP stream indicates an EXE or DLL returned from the server.

Since we have the key log file for this traffic, we can now export this malware from the pcap. Use the menu path **File** -> **Export Objects** -> **HTTP** to export this file from the pcap, as shown in Figure 16.

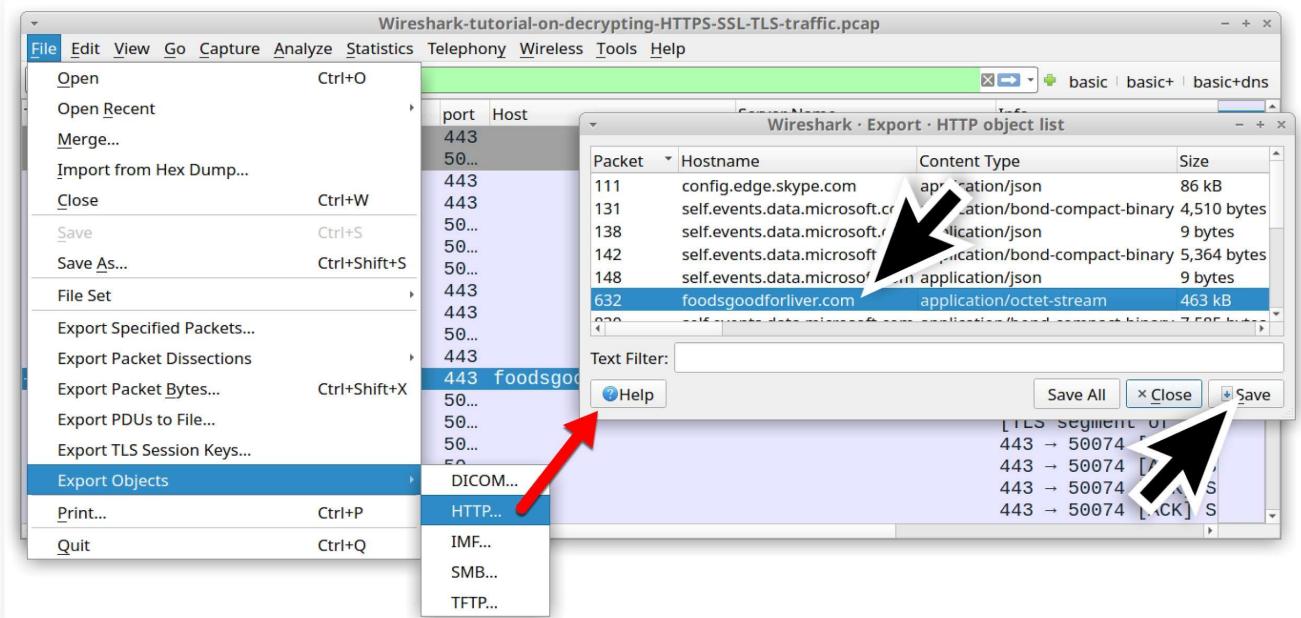
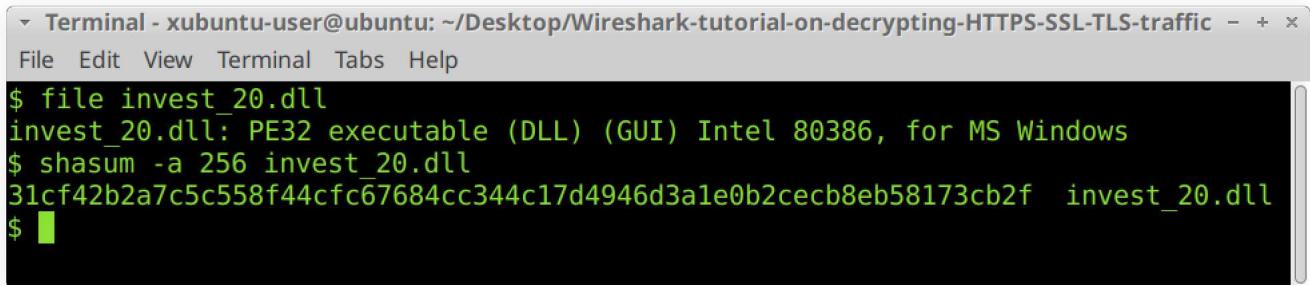


Figure 16. Exporting the malware binary returned from foodsgoodforliver[.]com.

If you are in a BSD, Linux or macOS environment, open a terminal window and use the `file` command to confirm this is a DLL file. Then use `shasum -a 256` to get the SHA256 hash of the

file, as shown in Figure 17.



```
Terminal - xubuntu-user@ubuntu: ~/Desktop/Wireshark-tutorial-on-decrypting-HTTPS-SSL-TLS-traffic - + x
File Edit View Terminal Tabs Help
$ file invest_20.dll
invest_20.dll: PE32 executable (DLL) (GUI) Intel 80386, for MS Windows
$ shasum -a 256 invest_20.dll
31cf42b2a7c5c558f44cf67684cc344c17d4946d3a1e0b2cecb8eb58173cb2f  invest_20.dll
$
```

Figure 17. Getting the SHA256 hash of this malware in a Linux environment.

The SHA256 hash of this malware is:

31cf42b2a7c5c558f44cf67684cc344c17d4946d3a1e0b2cecb8eb58173cb2f

If you search for this hash online, you should find results from at least two publicly available online sandbox environments.

Finally, we can review C2 traffic from this Dridex infection. Use your basic web filter, then follow an HTTP stream from one of the POST requests to 105711 [.] com. An example from one of the HTTP streams is shown in Figure 18.

POST /docs.php HTTP/1.1
 Accept: */*
 User-Agent: Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.88 Safari/537.36
 Host: 105711.com
 Content-Length: 369
 Connection: Close
 Cache-Control: no-cache
C.....n...qoD.M
h0.w.-o
 ~=...EU.....` .i"6.
 ..I|...2..... _x...*..b..7.. \s....U\$...aK...v.K...1.R.....,.....
 +'...Z@...xx.....KG...2.{.;vU&W.w.....m.....Ta..j.V.....xZ.}...yz....c./.bY.
 1X.8..h\$.....v...\$.0.BTk....k..UL..
 j.....)....B..g0....HL._@..j]{.m..M...9`("..x.P.E7.tn.....-.....\J.9..U:
 `5..w}.E..j..8.k...}.=A7C....@...\$.I...X.HTTP/1.1 502 Bad Gateway
 Server: mitmproxy 6.0.0.dev
 Connection: close
 Content-Length: 393
 Content-Type: text/html

 <html>
 <head>
 <title>502 Bad Gateway</title>
 </head>
 <body>
 <h1>502 Bad Gateway</h1>
 <p>TlsProtocolException('Cannot establish TLS with 162.255.119.253:443
 (sni: 105711.com): TlsException("SSL handshake error: SysCallError(104,
 \x27;ECONNRESET\x27;)\x27;)\x27;)</p>
 1 client pkt, 1 server pkt, 1 turn.

Figure 18. HTTP stream from one of the Dridex C2 POST requests.

Conclusion

This tutorial reviewed how to decrypt HTTPS traffic in a pcap with Wireshark using a key log text file. Without a key log file created when the pcap was originally recorded, you cannot decrypt HTTPS traffic from that pcap in Wireshark.

For more help with Wireshark, see our previous tutorials:

- [Customizing Wireshark – Changing Your Column Display](#)
- [Using Wireshark – Display Filter Expressions](#)
- [Using Wireshark: Identifying Hosts and Users](#)
- [Using Wireshark: Exporting Objects from a Pcap](#)
- [Wireshark Tutorial: Examining Trickbot Infections](#)
- [Wireshark Tutorial: Examining Ursnif Infections](#)
- [Wireshark Tutorial: Examining Qakbot Infections](#)