Wireshark Tutorial: Decrypting HTTPS Traffic

unit42.paloaltonetworks.com/wireshark-tutorial-decrypting-https-traffic/

Brad Duncan

August 21, 2020

By Brad Duncan

August 21, 2020 at 6:00 AM

Category: Tutorial, Unit 42

Tags: Wireshark, Wireshark Tutorial

traffic-for-wireshark-column-	setup.pcap					
File Edit View Go Ca	pture Analyze	Stat	tistics Telephony Wirele	ss Tools Help		
	0 9 0 0			. 璽		
http.request or ssl.handshake.	type == 1				Expression	÷
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	1 2.0.77 32	443	г о р	s2. p.com	Client Hello	
2018-08-03 19:06:20	1 2.0.77 31	443	I U N.	s2 p.com	Client Hello	
2018-08-03 19:06: <u>20</u>	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	192.0.77.32	443		s1.wp.com	Client Hello	
2018-08-03 19:06:20	216.58.218	443		<pre>fonts.googleapis.com</pre>	Client Hello	
2018-08-03 19:06:20	216.58.218	443		<pre>fonts.googleapis.com</pre>	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 10.06.20	52 84 125	80	d15knet/ai8a86 clou		GET /ree/usatoda	~
🔵 🎽 traffic-for-wireshark-co	olumn-setup.pcap			Packets: 4448 · Displayed: 1	23 (2.8%) Profile: Defa	ult

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 <u>Wireshark</u>, 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 <u>Dridex</u> malware infection.

Note: Our instructions assume you have customized your Wireshark column display as previously described in "<u>Customizing Wireshark – Changing Your Column Display</u>.".

Here is a Github repository with a ZIP archive containing the <u>pcap and a key log file used for</u> 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 <u>customized Wireshark column display</u>. 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.

		📕 Wiresh	ark · Go Deep.	× +			_	
		$\leftarrow \rightarrow$	🖒 🔒 https	//www.v	vireshark.org 🕼 🔍 🙀	☆ 🖻	Sign in	
		WIRES	1 SHARK	NEWS	Get Acquainted ▾ Get Help ▾ Develop ▾	Project I	lost Shar	kFest
*	Ethernet0					_		×
File	Edit View	Go Capture	Analyze Statistics	Telephony	y Wireless Tools Help			
	I 🙆 🕑 🔳	XC	९ 🗢 🔿 🖀 🛧 👱					
	tls.handshake	type eq 1.						+
Time			Dst	port	Server Name	Info		^
	2020-08-14	18:03	104.26.11.240	443	www.wireshark.org	Client	Hello	
	2020-08-14	18:03	104.26.11.240	443	www.wireshark.org	Client	Hello	
	2020-08-14	18:03	216.58.194.74	443	fonts.googleapis.com	Client	Hello	
	2020-08-14	18:03	209.197.3.15	443	<pre>maxcdn.bootstrapcdn.com</pre>	Client	Hello	
	2020-08-14	18:03	216.58.194.67	443	fonts.gstatic.com	Client	Hello	
	2020-08-14	18:03	13.226.189.114	443	jsi-cdn.steelcentral.net	Client	Hello	
<	2020 00 11	10.05	173 317 0 120	112	aiax googloanis com	Cliont	Hollo 3	
> F	rame 59: 5	71 bytes	on wire (4568	bits)	, 571 bytes captured (4568	bits)	on inte	r^ >``
	wireshark Ftl	nernet() 2020081	4180336 a03848.pcappg	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	Packets: 69432 · Displayed·	16 (0.0%)	Profile: Default	÷

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

Wireshark · Follow TCP Stream (tcp.stream eq 0) · Ethernet0

	^
;YiDX.%+./.,.	
0/.5/.5www.wireshark.org	
•	
zz#h2.http/1.1	
m	
+XRONij.ZI"s.r.Bs+	
······ZZ······	
• • • • • • • • • • • • • • • • • • • •	
$7 \times 8 i 0 E 7 c 2 o \int n$	
Μ Δγ Β "	
·Y. iD. X.%	
+G.{F:cg6dU	
0.nt.=H.n.JoD.FW.15.Bm.S=.o]42VY/F.*%]1H%MY.g.	
39 X\j[
C7(B\$gu	
8.Y.kFtmMB,LbSf#AG;Uk"# +.Q	
%S^ zS]f.	
N E O	¥
Entire conversation (1370 kB) \checkmark Show and save data as ASCII \checkmark Stream 0	▲ ▼
Find: Find Nex	t
Filter Out This Stream Print Save as Back Close Help	

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.

-					W	ireshark	-tutoria	l-KeysLo	gFile.t	kt - Mou	isepad					- + ×
File	Edit	Search	View	Docum	ent	Help			_		-					
CLIE	NT_R	ANDOM	5e850	016c20	1047	783111	78455b	55ee4o	:6cfd4	fee8e	f941b	la3fa	07c3b	3f8636	5c	d6b14d25
CLIE	NT R	ANDOM	5e850	917717	0b9e	2362c	1c72d0	dbc4a9	913344	1079b	f5ce0	cbe29	09016	d12096	7a	63c841f6
CLIE	NTR	ANDOM	5e850	9178c6	e06a	ad9d7e	b8ac04	fea8d4	ld4e5f	58fe3	008db	c <mark>795</mark> c	d4668	8a85c9	21	62d5fdb9
CLIE	NTR	ANDOM	5e850	01dceb	9af9	36b5e	346dc4	ac6665	of8dc6	10b37	d1a243	386fe	1a4d6	7298cb	e3	f1b55a36
CLIE	NTR	ANDOM	5e85(026cfd	7611	Lba445	bcdee5	2a9581	fe86a5	31cc3	e8e26	cb4bc	de4d2	cdaf79	e8	5ffc09c7
CLIE	NTR	ANDOM	5e850	0391f7	b861	l9e1c8	f65c12	cfc073	318162	48133	a0c89 ⁻	feb92	11cc9	e39437	47	f0402c77
CLIE	NT^{R}	ANDOM	5e850	941398	abde	602bad	132952	274a94	137186	8b8dca	a23e14	45ceb	7ec59	562a51	.b2	937cce1a
CLIE	NT^{R}	ANDOM	5e850	049b13	9b55	51f1ce	ba4372	df8ed4	1602b	0ca00	6251do	d3283	c38dd	bf7d27	82	ald744fa
	_															

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.

 \times

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 <u>this</u> <u>Github repository</u>. 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.

wireshark-tutorial-decrypting-HTTPS-traffic/Wireshark-tutorial-on-decrypting-HTTPS-SSL-TLS-tr	raffic.zip at master · pan-ur	nit42/wireshark - +
wireshark-tutorial-di × +		
	PS-traffic/blot 🚥 🗵 🏠	
Why GitHub? $ imes$ Team Enterprise Explore $ imes$ Marketplace Pricing $ imes$	Search	Sign in Sign up
₽ pan-unit42/wireshark-tutorial-decrypting-HTTPS-traffic	Watch 6 Star Star	0 양 Fork 0
<> Code () Issues 1 Pull requests () Actions () Projects () Security () Insights		
GitHub is home to over 50 million developers working together to host and revie manage projects, and build software together.	ew code,	Dismiss
Sign up		
generation wireshark-tutorial-decrypting-HTTPS-traffic / Wireshark-tutorial-on-decrypting-HTTPS-SSL	-TLS-traffic.zip	Go to file
brad-duncan Added zip archive with pcap and KeysLog file for tutorial.	Latest commit 2d8d620 3 ho	ours ago 🕲 H' tory
At 1 contributor		
532 KB		Download
View raw		

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.

Open *Wireshark-tutorial-on-decrypting-HTTPS-SSL-TLS-traffic.pcap* in Wireshark. Use a basic web filter as described in this previous <u>tutorial about Wireshark filters</u>. 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.

-				Wiresl	har <mark>k-tuto</mark> ri	al-on-d	ecryptin	g-HTTPS-SSL-TLS-traffic.pcap		- + ×
<u>F</u> ile	e <u>E</u> dit	<u>V</u> iew	<u>G</u> o <u>C</u> aptur	e <u>A</u> nalyze	Statistics	Telepho	n <u>y W</u> ire	eless <u>T</u> ools <u>H</u> elp		
	http.re	quest oi	r tls.handsha	ake.type eq	1) and !(ssd	p)		🛛 🗖 🔹 basic	basic+ b	asic+dns
Tim	e			Dst		port	Host	Server Name	Info	
	2020 2020 2020 2020 2020 2020 2020 202	- 04 - 01 - 04 - 01	L 21:02 L 21:02 L 21:02 L 21:04 L 21:06 L 21:11 L 21:13 L 21:16	13.107 40.122 94.103 40.122 20.191 162.255 162.255 162.255	.3.128 .160.14 .84.245 .160.14 .48.196 5.119.25 5.119.25 5.119.25	443 443 443 443 443 443 3 443 3 443 3 443		config.edge.skype.com self.events.data.microsoft.com foodsgoodforliver.com self.events.data.microsoft.com settings-win-ppe.data.microsof 105711.com 105711.com 105711.com	Client Client Client Client Client Client Client	Hello Hello Hello Hello Hello Hello Hello 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.

-	Wiresh	ark-tutorial-on-d	ecryptiı	ng-HTTPS-SSL-TLS-traffic.pcap	- + ×	
<u>F</u> ile	<u>Edit</u> <u>View</u> <u>Go</u> <u>Capture</u> <u>Analyze</u>	Statistics Telepho	ony <u>W</u> ir	eless <u>T</u> ools <u>H</u> elp		
📕 (ht	Сору	•		🛛 🗖 🝷 🔶 basic	basic+ basic+dns	
Time	႖ <u>F</u> ind Packet	Ctrl+F	Host	Server Name	Info	
20	Find Ne <u>x</u> t	Ctrl+N		config.edge.skype.com	Client Hello	
21	Find Pre <u>v</u> ious	Ctrl+B		foodsgoodforliver.com	Client Hello	
20	<u>M</u> ark/Unmark Packet(s)	Ctrl+M		self.events.data.microsoft.com	Client Hello	
20	Mark All Displayed	Ctrl+Shift+M		settings-win-ppe.data.microsof.	Client Hello	
21	Unmark All Displayed	Ctrl+Alt+M		105711.com	Client Hello	
20	Next Mark	Ctrl+Shift+N		105711.com	Client Hello	
	Previous Mark	Ctrl+Shift+B				
	Ignore/Unignore Packet(s)	Ctrl+D				
	Ignore All Displayed	Ctrl+Shift+D				
	Unignore All Displayed	Ctrl+Alt+D				
	Set/Unset Time Reference	Ctrl+T				
•	Unset All Time References	Ctrl+Alt+T	-		P	
	Next Time Reference	Ctrl+Alt+N				
	Previous Time Reference	Ctrl+Alt+B				
	Time Shift	Ctrl+Shift+T				
	Packet Comment	Ctrl+Alt+C				
	Delete All Packet (
	Configuration Pr	Ctrl+Shift+A				
	Preferences	Ctrl+Shift+P				

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.

-	Wireshark · Preferences	×
 Appearance Columns Font and Colo Layout Capture Expert Filter Buttons Name Resolv Protocols 29West 2dparityfec 3GPP2 A11 6LoWPAN 802.11 Radio 802 11 Padiot 	Wireshark - Preferences Protocols Display hidden protocol items splay byte fields with a space character between bytes Look for incomplete dissectors Enable stricter conversation tracking heuristics	x
Help	× <u>C</u> ancel <u>√</u> <u>O</u> K	

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 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											
E	ile	Edit View	Go	Capture	<u>Analyze</u>	Statistics	Telepho	ony <u>W</u> ireless <u>T</u> ools <u>H</u> elp				
	(ht	tp.request	or tls	.handsha	ke.type eq	1) and !(ssd	p)		8	🛛 🖃 🍨 basic 🗆 basic+ 🗆 basic+dns		
Tir	me				Dst		port	Host	Server Name	Info		
	2	920-04-	01 2	21:02	13.107	.3.128	443		config.edge.skype.com	Client Hello		
	2	920-04-	01 2	21:02	13.107	.3.128	443	config.edge.skype.com	1	GET /config/v2/Office/wor		
	2	920-04-	01 2	21:02	40.122	.160.14	443		self.events.data.micr.	. Client Hello		
	2	920-04-	01 2	21:02	40.122	.160.14	443	self.events.data.mi…		POST /OneCollector/1.0/ H		
	2	920-04-	01 2	21:02	40.122	.160.14	443	self.events.data.mi…		POST /OneCollector/1.0/ H		
	2	920-04-	01 2	21:02	94.103	.84.245	443		foodsgoodforliver.com	Client Hello		
	2	920-04-	01 2	21:02	94.103	.84.245	443	foodsgoodforliver.com	1	GET /invest_20.dll HTTP/1		
	2	920-04-	01 2	21:04	40.122	.160.14	443		self.events.data.micr.	. Client Hello		
	2	920-04-	01 2	21:04	40.122	.160.14	443	self.events.data.mi…		POST /OneCollector/1.0/ H		
	2	920-04-	01 2	21:06	20.191	.48.196	443		settings-win-ppe.data.	.Client Hello		
	2	920-04-	01 2	21:06	20.191	.48.196	443	settings-win-ppe.da		GET /settings/v2.0/Storag		
	2	920-04-	01 2	21:11	162.25	5.119.25	3 443		105711.com	Client Hello		
	2	920-04-	01 2	21:11	162.25	5.119.25	3 443	105711.com		POST /docs.php HTTP/1.1		
	2	920-04-	01 2	21:13	162.25	5.119.25	3 443		105711.com	Client Hello		
	2	920-04-	01 2	21:13	162.25	5.119.25	3 443	105711.com		POST /docs.php HTTP/1.1		
	2	920-04-	01 2	21:16	162.25	5.119.25	3 443		105711.com	Client Hello		
	2	920-04-	01 2	21:16	162.25	5.119.25	3 443	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.

 Wireshark · Follow HTTP Stream (tcp.stream eq 2) · Wireshark · Follow · Fo	reshark-tutorial-on-decr	ypting-HT	TPS-SSL-TLS-	traffic.pc	ар – +	×
GET /invest_20.dll HTTP/1.1 Connection: Keep-Alive Accept: */* User-Agent: Mozilla/4.0 (compatible; Win3 Host: foodsgoodforliver.com	2; WinHttp.WinHtt	pReques	t.5)			
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 Connection: keep-alive ETag: "5e84c15c-71400" Accept-Ranges: bytes	GMT					
MZ@@ program cannot be run in DOS mode.				!L.!1	This	
\$/SQ\$k2?wk2?wk2?wwb2?ww.2?w. 2?w.i6vj2?w.i.wj2?w.i=vj2?wRichk2? wPELC^ @	ws2?W.i <vy2?w.i< td=""><td>.:vw2?w.</td><td>i;v{2?wbJ K</td><td>.wf2?wk @.</td><td>(2>w.</td><td>•</td></vy2?w.i<>	.:vw2?w.	i;v{2?wbJ K	.wf2?wk @.	(2>w.	•
1 <mark>client</mark> pkt, 1 server pkt, 1 turn.						
Entire conversation (464 kB) -		Sho	ow and save d	ata as AS	CII	•
Find:					Find <u>N</u> e	ext
elp	Filter Out This Stream	Print	Save as	Back	× <u>C</u> los	e

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.



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

The SHA256 hash of this malware is:

31cf42b2a7c5c558f44cfc67684cc344c17d4946d3a1e0b2cecb8eb58173cb2f

If you <u>search for this hash</u> 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.

Wireshark · Follow HTTP Stream (tcp.stream eq 5) · Wireshark-tutorial-on-decrypting-HTTPS-SSL-TLS-traffic.pcap - + × 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-cacheh0.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> TlsProtocolException('Cannot establish TLS with 162.255.119.253:443 (sni: 105711.com): TlsException("SSL handshake error: SysCallError(104, \ &#v27 · FCONNDESET\ &#v27 ·)&duot ·)&#v27 ·)</n> 1 client pkt, 1 server pkt, 1 turn. Entire conversation (1,134 bytes) * Show and save data as ASCII Find: Find <u>N</u>ext Filter Out This Stream Help Print Save as... Back × Close

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:

```
Get updates from
Palo Alto
Networks!
```

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

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