Guide for Analyzing HTTPS/TLS Traffic with Wireshark

Due to the recent policy changes of odtuyayincilik.com.tr, it is not possible to establish an unsecure HTTP connection to the website. As some of you have noticed, Wireshark by default is not able to decrypt the traffic and simply shows the raw TLS data.

Fortunately, by making your browser export the required encryption keys, we can configure the Wireshark to show the decrypted **HTTP/2** (note the version 2 here) traffic. For this to work, we need to first tell our browser to which file it should export the keys.

Note: I have tested this approach with recent versions of Firefox and Google Chrome both on Windows 10 and Linux, so it should work in general. However, I have read that certain old versions of Firefox may not support this. In that case, either temporarily switch to Google Chrome or update your Firefox.

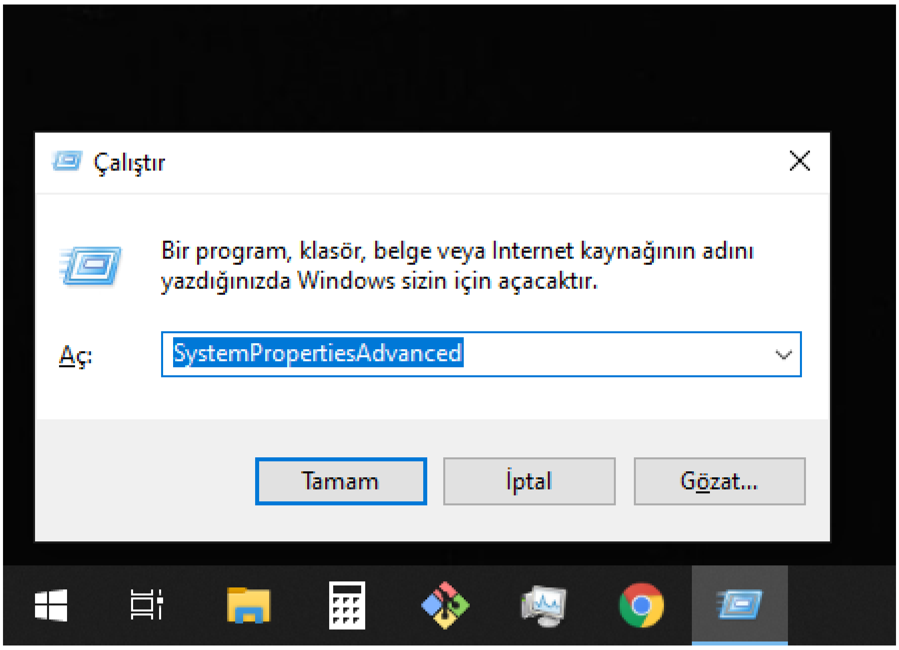
Configuring the Environment Variables

For Ubuntu or similar Linux, do the following:

1. **`$ nano ~/.bashrc`**
2. Append to the end of the file: **`export SSLKEYLOGFILE=~/sslkeylog.pms`**

If you are on Windows (same procedure for 7, 8, 8.1, and 10), do the following:

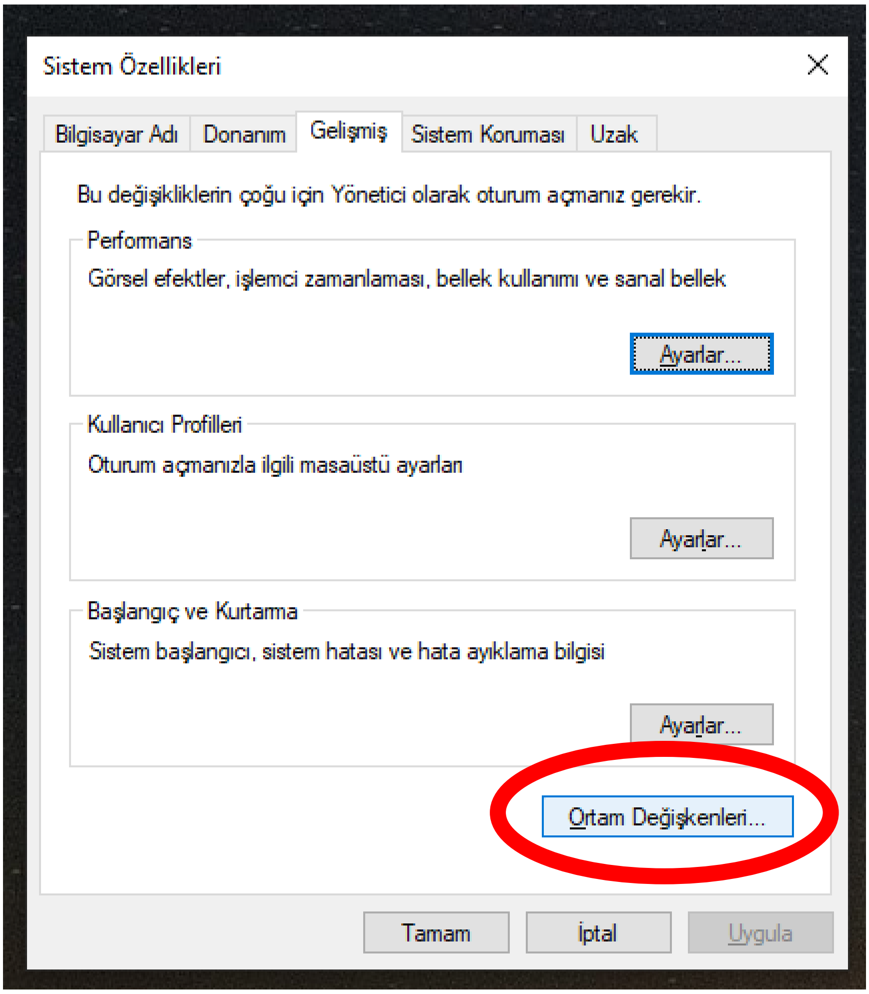
1) Press **Win + R** to open the Run dialog (or search for Run from start menu). Type "**SystemPropertiesAdvanced**" to open the advanced system properties window. (Alternatively, right click My Computer on desktop > Properties > Advances System Settings)



Bir program, klasör, belge veya Internet kaynağının adını yazdığınızda Windows sizin için açacaktır.

Aç: SystemPropertiesAdvanced

2) Click “**Environment Variables**” on the System Properties window.



Sistem Özellikleri

Bu değişikliklerin çoğu için Yönetici olarak oturum açmanız gerekir.

Performans

Görsel efektler, işlemci zamanlaması, bellek kullanımı ve sanal bellek

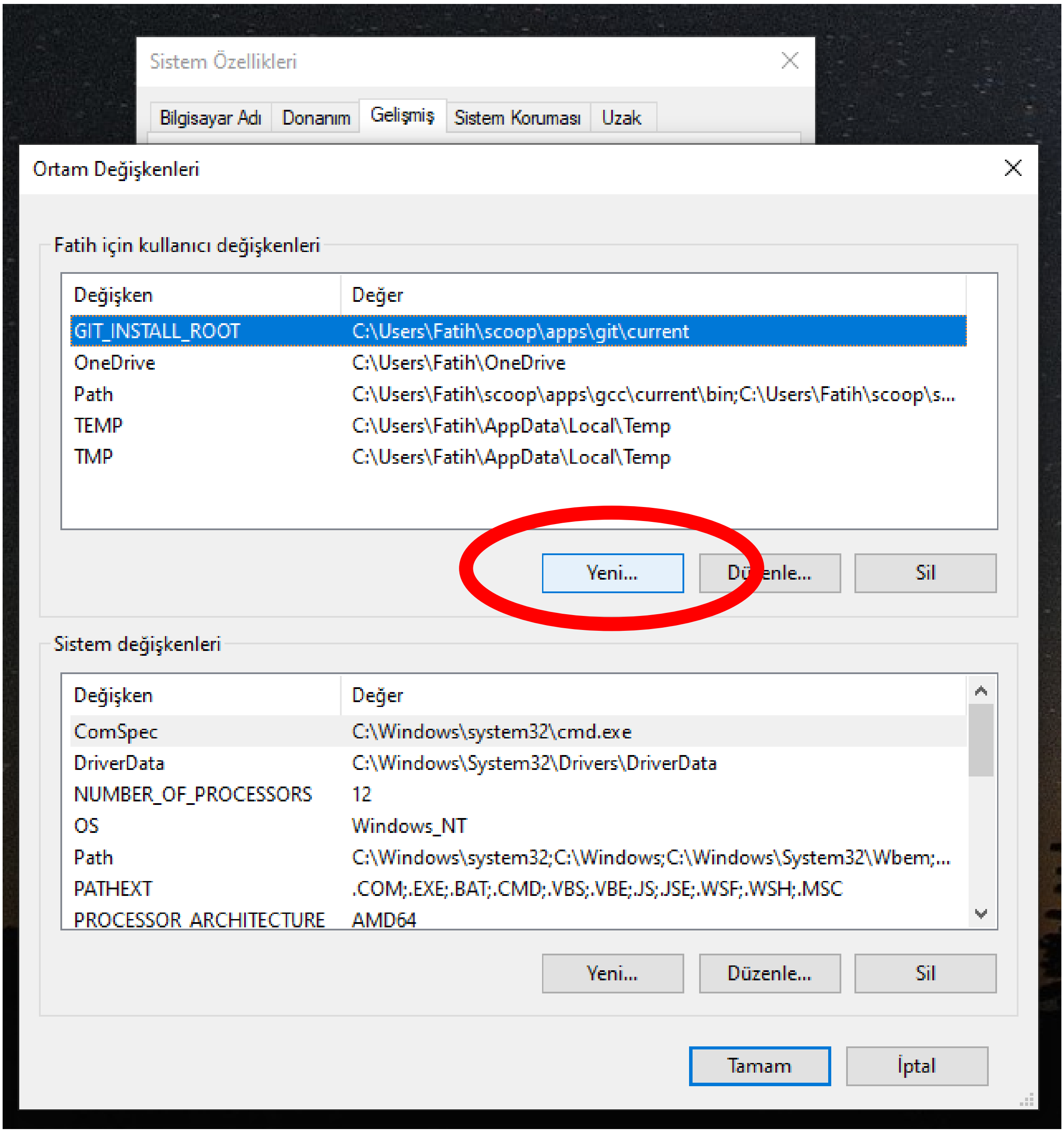
Kullanıcı Profilleri

Oturum açmanızla ilgili masaüstü ayarları

Başlangıç ve Kurtarma

Sistem başlangıcı, sistem hatası ve hata ayıklama bilgisi

3) Click “**New…**” in user environment variables panel



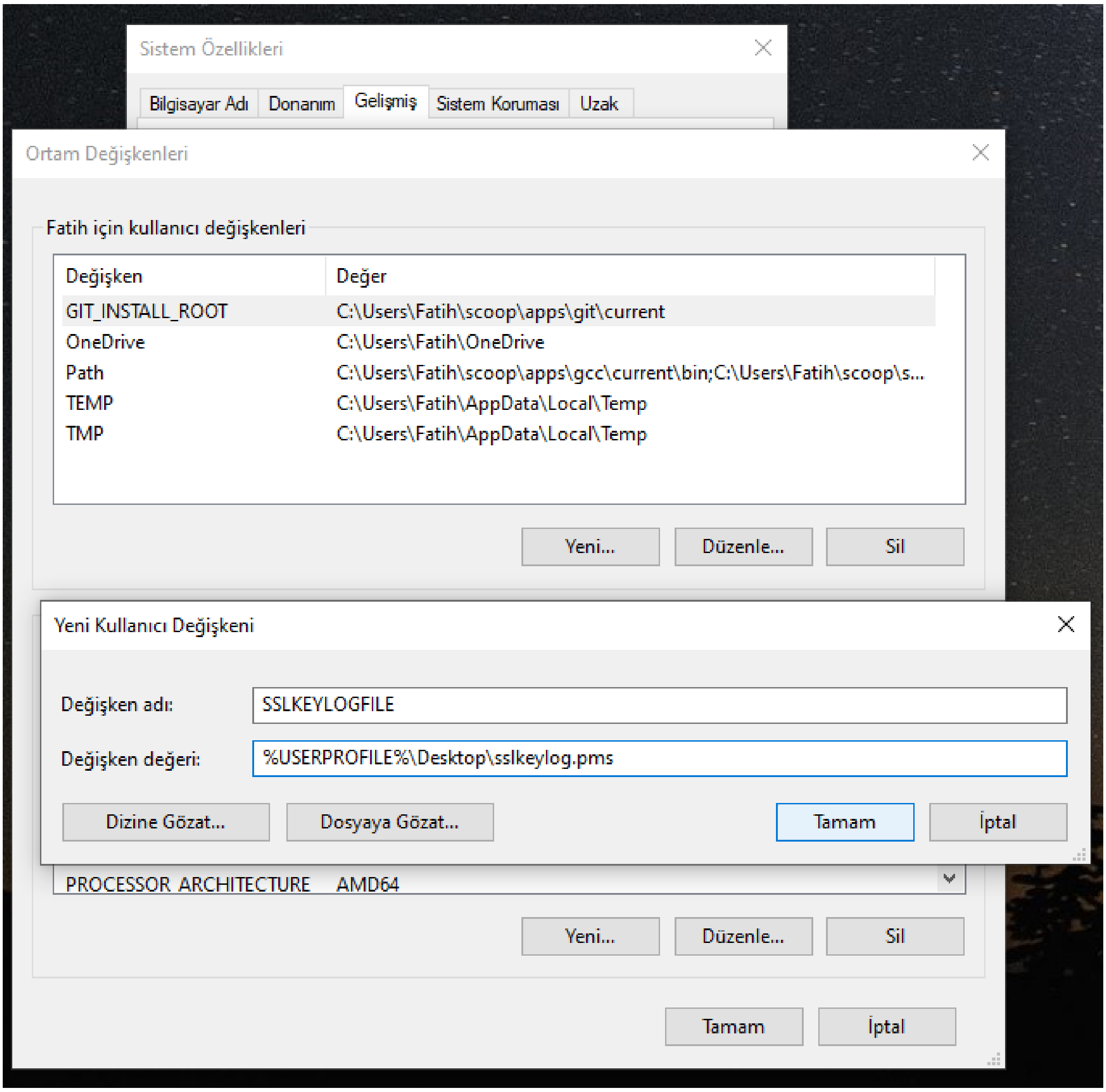
Fatih için kullanıcı değişkenleri

Değişken	Değer
GIT_INSTALL_ROOT	C:\Users\Fatih\scoop\apps\git\current
OneDrive	C:\Users\Fatih\OneDrive
Path	C:\Users\Fatih\scoop\apps\gcc\current\bin;C:\Users\Fatih\scoop\s...
TEMP	C:\Users\Fatih\AppData\Local\Temp
TMP	C:\Users\Fatih\AppData\Local\Temp

Sistem değişkenleri

Değişken	Değer
ComSpec	C:\Windows\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PROCESSORS	12
OS	Windows_NT
Path	C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE	AMD64

4) Fill in the details for the new environment variable:
    a. Variable Name: SSLKEYLOGFILE
    b. Variable Value: %USERPROFILE%\Desktop\sslkeylog.pms
        i. %USERPROFILE% will resolve to something like “C:\Users\<username>”
        ii. This could be any other path you have the permission to write to



Fatih için kullanıcı değişkenleri

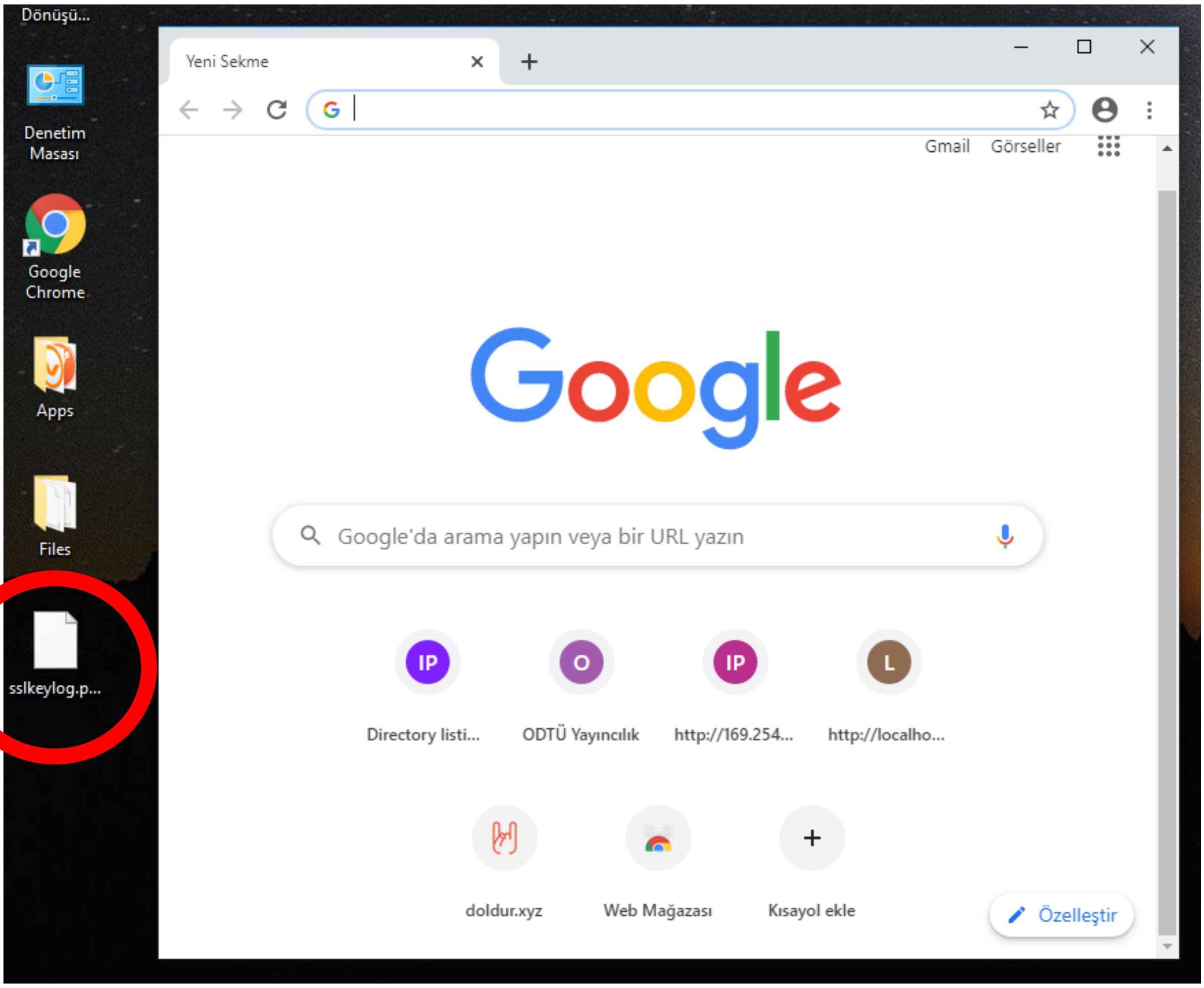
Değişken	Değer
GIT_INSTALL_ROOT	C:\Users\Fatih\scoop\apps\git\current
OneDrive	C:\Users\Fatih\OneDrive
Path	C:\Users\Fatih\scoop\apps\gcc\current\bin;C:\Users\Fatih\scoop\s...
TEMP	C:\Users\Fatih\AppData\Local\Temp
TMP	C:\Users\Fatih\AppData\Local\Temp

Değişken adı: SSLKEYLOGFILE

Değişken değeri: %USERPROFILE%\Desktop\sslkeylog.pms

PROCESSOR_ARCHITECTURE AMD64

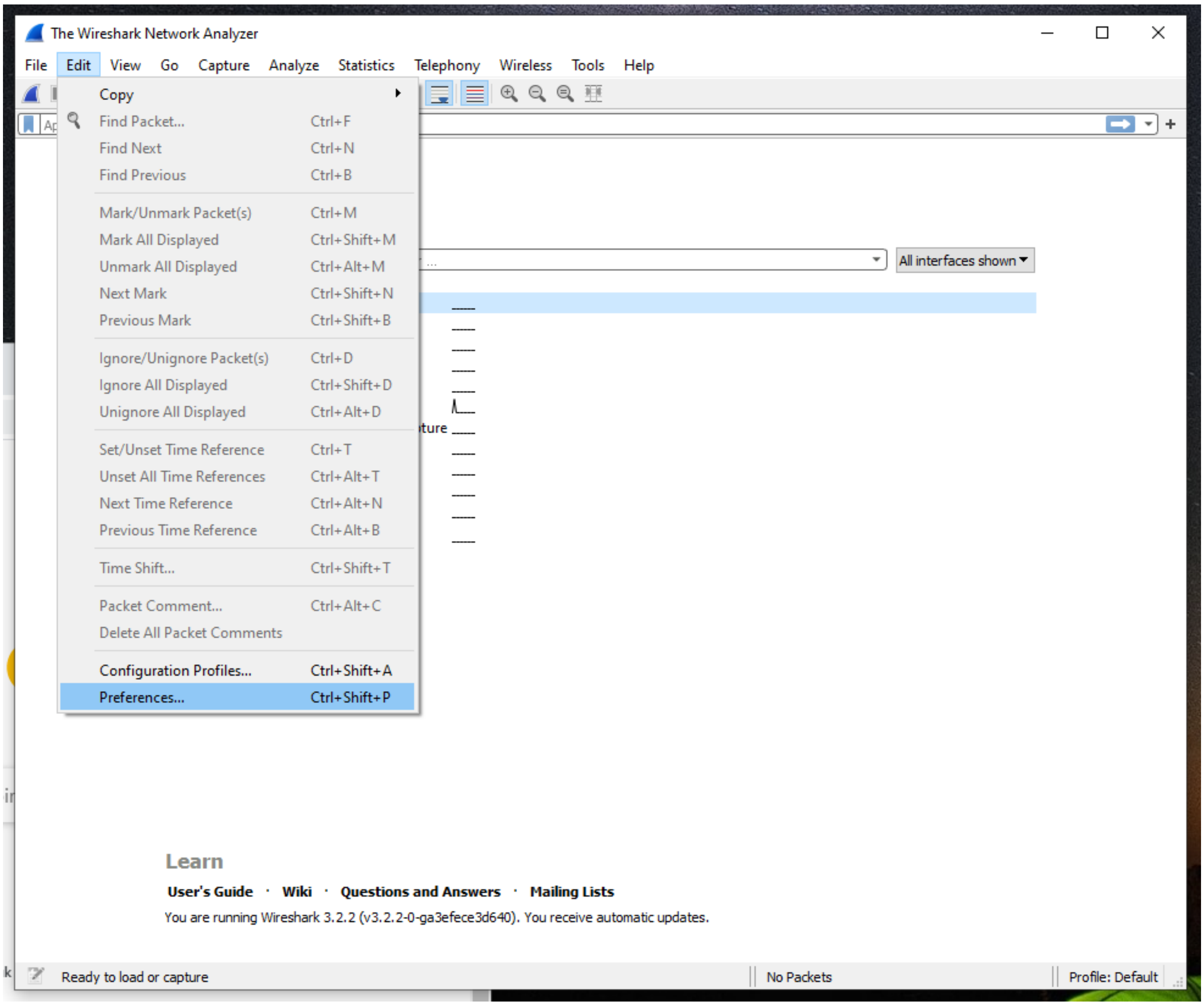
5) Run the browser. Firefox, Google Chrome or any other decent browser (Opera also uses the chromium engine) will respect this environment variable. You should see the “sslkeylog.pms” file created on your desktop (or wherever you set it to).



Checkpoint: Make sure the file is created when the browser is opened. If the file is deleted, it will keep being created each time the browser opens. So when you are done with the homework, you should delete the environment variable created before.

Configuring Wireshark to Use the SSL Key Log

1. Open Wireshark as usual. Click **Edit > Preferences**



The Wireshark Network Analyzer
File
Edit
View
Go
Capture
Analyze
Statistics
Telephony
Wireless
Tools
Help
Copy
Find Packet... Ctrl+F
Find Next Ctrl+N
Find Previous Ctrl+B
Mark/Unmark Packet(s) Ctrl+M
Mark All Displayed Ctrl+Shift+M
Unmark All Displayed Ctrl+Alt+M
Next Mark Ctrl+Shift+N
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
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 Comments
Configuration Profiles... Ctrl+Shift+A
Preferences... Ctrl+Shift+P
All interfaces shown
Learn
User's Guide · Wiki · Questions and Answers · Mailing Lists

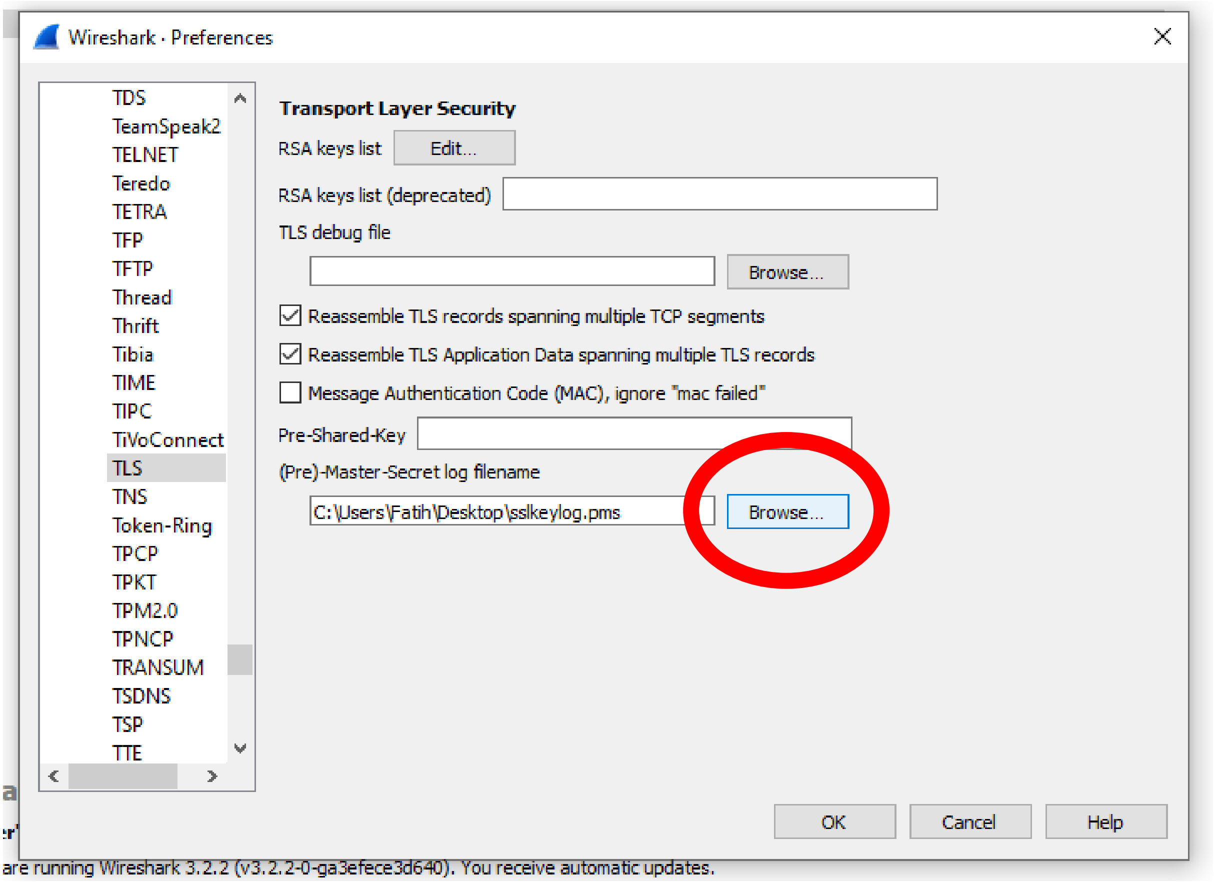
You are running Wireshark 3.2.2 (v3.2.2-0-ga3efece3d640). You receive automatic updates.

Ready to load or capture

No Packets

Profile: Default

2. On the left panel, expand the **Protocols** node. Lots of protocols will be shown. Press **T** and find the **TLS** protocol. Click the bottom-most **Browse** button and select the newly created file. Click **OK** to save. Note that this file will automatically grow each time the browser generates a new key and Wireshark will see the updates. So you only need to do this once.



Transport Layer Security

RSA keys list [Edit...]

RSA keys list (deprecated)

TLS debug file [Browse...]

☑ Reassemble TLS records spanning multiple TCP segments

☑ Reassemble TLS Application Data spanning multiple TLS records

☐ Message Authentication Code (MAC), ignore "mac failed"

Pre-Shared-Key

(Pre)-Master-Secret log filename

C:\Users\Fatih\Desktop\sslkeylog.pms [Browse...]

are running Wireshark 3.2.2 (v3.2.2-0-ga3efece3d640). You receive automatic updates.

And that's it. Try to observe the traffic by navigating to odtuyayincilik.com.tr and see that the traffic is decrypted.

<u>Hint:</u> To isolate the HTTP traffic, try applying the following filters:

1. **`http2`**
2. **`http2.headers`**
3. **`http2.headers.set_cookie`**

Notice the **HTTP version 2** here. Before the change, the website used HTTP/1.1, which means the appropriate filter would be **`http.set_cookie`**

The last one should narrow the traffic down sufficiently to answer the question in homework.