

Wireshark HTTPS Decryption

[Originalartikel](#)

[Backup](#)

<html> <p>If you've done any network programming or hacking, you've probably used Wireshark. If you haven't, then you certainly should. Wireshark lets you capture and analyze data flowing over a network; think of it as an oscilloscope for network traffic. However, by design, HTTPS traffic doesn't give up its contents. Sure, you can see the packets, but you can't read them; that's one of the purposes of HTTPS is to prevent people snooping on your traffic from reading your data. But what if you are debugging your own code? You know what is supposed to be in the packet, but things aren't working for some reason. Can you decrypt your own HTTPS traffic? The answer is yes and [rl1987] shows you how.</p><p>Don't worry, though. This doesn't let you snoop on anyone's information. You need to share a key between the target browser or application and Wireshark. The method depends on the target applications like a browser writing out information about its keys. Chrome, Firefox, and other software that uses NSS/OpenSSL libraries will recognize an SSLKEYLOGFILE environment variable that will cause them to produce the correct output to a file you specify.</p><p>How you set this depends on your operating system, and that's the bulk of the post is describing how to get the environment variable set on different operating systems. Wireshark understands the file created, so if you point it to the same file you are in business.</p><p>Of course, this also lets you creep on data the browser and plugins are sending which could be a good thing if you want to know what Google, Apple, or whoever is sending back to their home base using encrypted traffic.</p><p>Wireshark and helpers can do lots of things, even Bluetooth. If you just need to replay network data and not necessarily analyze it, you can do that, too.</p> </html>

From:
<https://schnipsl.qgelm.de/> - **Qgelm**

Permanent link:
<https://schnipsl.qgelm.de/doku.php?id=wallabag:wb2wireshark-https-decryption>

Last update: **2025/06/27 11:17**

