

Facebook Muscles Up Security... by Default

by [G.S. Hans](#) [1], [Joseph Lorenzo Hall](#) [2]

November 28, 2012

Thanks to some fancy math, it just got a lot harder for someone to snoop on your Facebook conversations. And that's all thanks to [Facebook's decision](#) [3] to automatically scramble the communications stream from your keyboard to the actual site itself.

The move actually entails changing the method your communications take before they land on a Facebook page. What used to be a wide-open, snoop-able, stream of communications emanating from your keyboard to your latest status update is now protected by a secure method called "HTTPS."

In January 2011, [Facebook allowed users to opt into HTTPS](#) [4], and alluded to a future default setting. The project of scaling HTTPS for all Facebook users while preserving the site's performance presented a technical challenge, but Facebook says that it's addressed those concerns. This is a very welcome move from CDT's perspective as HTTPS provides a secure connection between users and websites. While users can opt out of the switch, Facebook's move to HTTPS by default within North America (and to the rest of the world early next year) will provide users with heightened security as they use the world's most popular social network service.

[CDT has previously discussed](#) [5] how easy ([and possibly illegal](#) [6]) it is to eavesdrop on non-encrypted HTTP — normal web surfing, while describing the benefits of the secure HTTPS version. Here's how it works: the HTTPS protocol encrypts the communication link from the user to the website, preventing eavesdroppers — for example, on open Wi-Fi networks — from snooping on your web surfing activity. Technically, what happens here is that the user's browser and the destination server do some fancy math to exchange a secret key. With that secret key, the browser can then wrap communications sent to the server in a "sheath" or "tunnel" of encryption that is virtually impossible to break open. Many institutions, such as banks and payment processors, have long used HTTPS by default in order to protect their users' security and ensure that sensitive data remains private.

As more and more websites collect and maintain important and possibly sensitive user data, HTTPS can prevent the unwanted collection of your private information. While Facebook facilitates sharing information among individuals, users constantly pick and choose which of their Facebook friends can see specific posts or information. The move to HTTPS ensures that strangers don't get access to that data that Facebook users believe they have safeguarded.

Of course, Facebook's move to HTTPS does not inoculate users against security breaches or other kinds of malware. A secure connection to an unintended party could still be a security risk, and phishing schemes — where a seemingly innocuous link actually points to a malicious website — carried out via email could use HTTPS in the URL (such as <https://faceb00k.com> [7]) and still collect user data. Yet Facebook's decision to move to HTTPS by default, despite the logistical hurdles and some slowness in the time a page will take to load, signals both the importance of protecting user data, and the importance of popular websites in demonstrating strong security safeguards.

Other companies that handle sensitive user data or communications should follow Facebook's decision and enable HTTPS by default as well.

-
- [privacy](#)

- [HTTPS](#)

Copyright © 2013 by Center for Democracy & Technology. CDT can be freely copied and used as long as you make no substantive changes and clearly give us credit. [Details](#).

Source URL: <https://cdt.org/blogs/cdt/2811facebook-muscles-security-default>

Links:

[1] <https://cdt.org/personnel/gs-hans>

[2] <https://cdt.org/personnel/joseph-lorenzo-hall>

[3]

<https://developers.facebook.com/blog/post/2012/11/14/platform-updates--operation-developer-love/>

[4] <https://blog.facebook.com/blog.php?post=486790652130>

[5] <https://www.cdt.org/blogs/aaron-brauer-rieke/dont-get-hijacked-net-firesheep-and-https>

[6] http://www.computerworld.com/s/article/9194159/Is_it_legal_to_use_Firesheep_at_Starbucks_

[7] <https://faceb00k.com>