Summary of HSTS Support in Modern Browsers

Posted May 28, 2015

This a guest blog post by Mazin Ahmed, an external security expert who has helped test and audit ProtonMail. We hope it will educate our readers about web security.

HTTP Strict Transport Security (HSTS) is a web security policy that is made to protect secure HTTPS websites against downgrade attacks that is used to perform Man in the middle attacks. “Once a supported browser receives this header that browser will prevent any communications from being sent over HTTP to the specified domain and will instead send all communications over HTTPS. It also prevents HTTPS click through prompts on browsers”[1].

The downgrade attack can occur when the user request a website in HTTP. When the user requested in HTTP:// URL, the server redirects the user to a secure HTTPS:// URL. Without HSTS, an attacker in the same network is able to stop the user from using the HTTPS version of the site and forcing the user to use the HTTP version, where the attacker is able to control user’s data and hijack the user’s session.

The HTTPS stripping attack that relates to the unuse of HSTS policy can be done via open-source tools, such as SSLStrip by Moxie Marlinspike. The tool has been publicly released on February 2009, and since then many browsers and websites have mitigated the issue.

Preventing the issue should be from two sides: the website, and the client. The website has that is already using HTTPS should apply HSTS policy to prevent the attack. The client has to use an updated version that supports HSTS policy.

The current advisory is tend to inform users that there is still modern browsers that do not support HSTS policy until today.

The following browsers supports HSTS policy (latest versions of browsers):

- Google Chrome
- Mozilla Firefox
- Microsoft Edge
- Safari for OSX
- Opera
- Ghostery
- TOR Browser

The following browsers do not support HSTS policy:

- Internet Explorer (all stable versions do not support HSTS. Only Microsoft Edge and Internet Explorer 10 Technical Preview support it)
- Android Browser (All versions upto 4.4.2 do not support HSTS policy. Newer versions might not be supporting it too)
- Opera Mini (all versions, including Opera Mini 8)
- Maxthon browser
- UC browsers (including UC browser, UC mini, UC browser HD)
- Opera for Android

Testing Browsers for MITM attacks due to Lack of HSTS policy:

You can follow the instructions in the following link to check if the browser supports HSTS policy.

http://www.thoughtcrime.org/software/sslstrip/

Recommendations:

- Always use the latest version of browsers
- If you are concerned about your security, do not use a browser that does not support HSTS policy.
- If you are using a browser that is not listed above, you should test if it supports HSTS policy, and if it does not support it, you should stop using it.

Notes:

- Even if the website is using the maximum SSL encryption possible, if the client's browser does not support HSTS, the client is vulnerable to man in the middle attacks.
- If the client made an initial request to the website in HTTP, an attacker can manipulate the response to uninclude the HSTS header, and would be able to perform MITM attacks too. Therefore, if you are making an initial request to secured site, you should request it in HTTPS to avoid HTTPS stripping that leads to MITM attacks.

The list might be updated for newer information. If there is a mistake on the list, you can contact me to correct it.

References:

http://en.wikipedia.org/wiki/HTTP_Strict_Transport_Security [1]

http://caniuse.com/#feat=stricttransportsecurity

https://www.owasp.org/index.php/HTTP_Strict_Transport_Security

http://blogs.msdn.com/b/ie/archive/2015/02/16/http–strict–transport–security–comes–to–internet–explorer.aspx

https://developer.mozilla.org/en–US/docs/Web/Security/HTTP_strict_transport_security

http://www.chromium.org/hsts

http://www.thoughtcrime.org/software/sslstrip/

About the Author

Mazin Ahmed is an information security specialist with experience in web-application security and mobile application security. Mazin is passionate about information security and has reported vulnerabilities which have been acknowledged by various companies, such as Facebook, Twitter, Linkedin, and Oracle to name a few. Mazin is part of ProtonMail's security group, an independent panel of experts who audit ProtonMail releases on a voluntary basis. You can reach him via Twitter @mazen160, and read more about him by visiting his website.