Protecting Your Website With Always On SSL

Developing and advocating best practices to mitigate emerging privacy, identity and security threats to online services, government agencies, organizations and consumers, thereby enhancing online trust and confidence and the vitality of the digital economy.

Updated May 1, 2012

Table of Contents

Executive Summary

Trust and consumer confidence are the foundations upon which the Internet has been built. Leading commerce and financial services companies worldwide have long used Secure Socket Layer and Transport Layer Security (SSL/TLS) technologies to secure customer communications and transactions. This security model has been used for more than a decade to ensure trust in Web browsers, mobile devices, e-mail clients, and other Internet applications, and it is still fundamentally sound. Websites and relying parties commonly employ SSL/TLS to protect sensitive information such as passwords and credit card numbers. The number of sites using SSL/TLS has more than doubled since 2005, and today, it is estimated that more than 4.5 million sites are using SSL certificates issued by a Certificate Authority.[1]

But with the rise of Web 2.0 and social networking, people are spending more time online and logged in, and they are communicating much more than just their credit card numbers. Many people use Facebook, Gmail, and Twitter as their primary mode of communication. The threat landscape has also evolved with the proliferation of botnets, malware, data loss, forged email, online fraud, and other security and privacy challenges. Unfortunately, Web security practices have not always kept pace with these changes. Many organizations use the SSL/TLS protocol to encrypt the authentication process when users log in to a website, but do not encrypt subsequent pages during the user's session. This practice is risky because it leaves website visitors vulnerable to malicious online attacks, and can result in millions of users being unknowingly exposed to threats even when visiting a trusted website.

Online Trust Alliance (OTA) is calling on the security, business and interactive advertising communities to work together and protect consumers from harm. It is incumbent on all stakeholders to take reasonable steps to protect trust and consumer confidence by adopting security best practices that are vendor-neutral, easy to implement, and globally accessible. One such practice is *Always On SSL* (AOSSL), the approach of using SSL/TLS across your entire website to protect users with persistent security, from arrival to login to logout. Always On SSL is a proven, practical security measure that should be implemented on all websites where users share or view sensitive information.

This white paper discusses the imperative need for Always On SSL, and the steps you can take to deliver end-to-end protection for your users. It also includes detailed accounts of four organizations—Facebook, Google, PayPal and Twitter—that are leading the way with Always On SSL in a cooperative effort to make the Internet more secure.

OTA wishes to acknowledge input from the OTA Steering Committee and members, including AllClear, DigiCert, Epsilon, IID, Intersections, LashBack, MarkMonitor, Message Systems, Microsoft, PayPal, Pitney Bowes, Publishers Clearing House, Return Path, Secunia, Star Marketing Group, Symantec, TrustSphere and VeriSign, Inc.

Special thanks to input from Alex Rice at Facebook, Adam Langley at Google, Andy Steingruebl at PayPal, John Scarrow at Microsoft, Quentin Lui and Rick Andrews at Symantec, Steve Waite at GlobalSign, Bob Lord at Twitter and Craig Spiezle of OTA for their collaboration in this paper.

Updates of this report will be posted at https://otalliance.org/aossl.htm. To submit comments, please email staff@otalliance.org.

[1] Netcraft February 2012 SSL Survey

The Need for Persistent Protection Online

Users today have access to a large and growing variety of Web 2.0 services that provide them with rich, interactive, personalized experiences as they search, share and shop online. Many services rely on browser cookies to enable these experiences by creating stateful, persistent user sessions. When a user logs into a site, they typically have to submit a username and password to authenticate their identity. The Web server then generates a unique session token ID for the user and transmits it to the Web browser, where it is cached in a cookie. The Web browser sends the cached content of the cookie back to the Web server each time the logged in user interacts with the site, and the cookie remains active until it either expires or is deleted.

HTTP and Insecure Cookies Leave Users Vulnerable to Attack

Many websites use the HTTPS protocol to transmit login information over an encrypted SSL channel, but then downgrade their users to HTTP after setting up the session cookie. This may protect the user's password, but the cookie—including the session ID—is transmitted in plaintext when the Web browser makes subsequent requests to the domain, leaving users vulnerable to session hijacking attacks. It can also give users a false sense of security because they incorrectly assume their entire session is secure, when it is only the login that is encrypted.

Some organizations use site-wide HTTPS, but neglect to mark session cookies as secure. This, too, is a risk because users often type in partial URLs (e.g., without explicitly typing "https://" beforehand), and their cookies are exposed during that first request, before they are redirected to an HTTPS page. An attacker who is monitoring an open network only needs to capture a single unencrypted HTTP request to steal the victim's cookie and hijack their account.

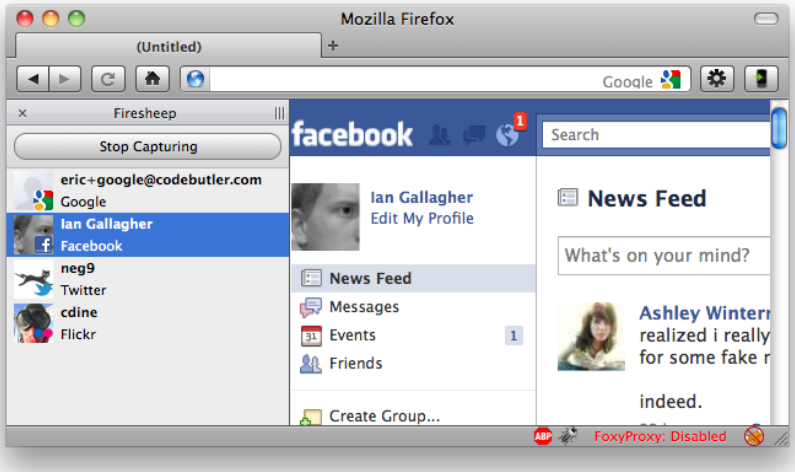
These problems are not new and can affect any website that uses session cookies. Even search engines that echo user query terms are vulnerable to such attacks. Organizations can no longer afford to remain complacent, and user education alone is not enough. The general state of online security throughout the industry has reached a tipping point, and web sites must change in order to preserve end-to-end trust and consumer confidence.

Session Hijacking Has Become Dangerously Easy

Session hijacking is not a new problem, but the recent release of a Firefox browser plug-in called "Firesheep" has increased awareness among both users and attackers about the inherent insecurity of unprotected HTTP connections (and open Wi-Fi networks). Developed by Eric Butler and Ian Gallagher, Firesheep makes it "incredibly easy" for an attacker with no programming skills to "sidejack" someone's user account on a large number of popular websites. Firesheep finds and joins open networks, such as an unencrypted Wi-Fi connection in a coffee shop, library or Internet cafe, and uses a packet sniffer to capture unsecured cookies. As soon as anyone on the network visits an insecure website known to Firesheep, the software grabs and displays their user name and the service(s) to which they are connected to. The attacker can then double-click on the victim's name and instantly gain access to their account.

Firesheep is an innovation in terms of feature integration and ease of use. But as Butler and Gallagher stated, Firesheep exposed the severity and scope of a problem that security experts have been warning about for several years. Software tools such as "Hamster," "Ferret," and “CookieMonster” were introduced years before Firesheep, and also enable attackers to sniff open networks, steal cookies, and hijack HTTP sessions with relative ease.

Figure 1. Screenshot of Firesheep in action



With these tools in hand, an attacker can exploit this vulnerability to gain full or partial access to the victim’s account. Some sites take precautions to avoid full sidejacking (such as asking for the old password when a user attempts to change it), but in many cases, the attacker can completely take over the victim’s account, change their password, and potentially hijack other services that are connected to the victim’s account.

Not Just a Coffee Shop Problem

Some people assume that session hijacking is just a “coffee shop” problem, and that the solution is for users to avoid unencrypted Wi-Fi networks. But this is an unrealistic expectation, and the reality is that tools such as Firesheep can be used to intercept *any* network traffic, wired or wireless, that includes cookies sent over unencrypted HTTP sessions. Another overlooked aspect is the risk to enterprise organizations and government agencies that use social networking sites, webmail, and Web 2.0 applications. If an employee accesses an insecure site and their session is hijacked, the attacker can use the account as a vector to spread malware, or potentially gain access to privileged assets and cause a data security breach. The potential damage that could result from such a breach is something no website operator wants to face.

User Education Alone Is Not Enough

In an effort to increase user awareness and combat the danger of sidejacking, the Electronic Frontier Foundation (EFF) created their own Firefox extension, HTTPS-Everywhere, which forces Firefox to use only HTTPS connections for certain websites. In addition, the EFF and Access, a non-profit Internet rights organization, have created HTTPS Now, an international education campaign to improve awareness and adoption of Always On SSL as the "minimum level of security" for Web browsing. The campaign includes a website where users can go to search for and contribute information about how websites use HTTPS.[2] HTTPS-Everywhere is regarded as a tool that works well on a defined list of websites, but it won't protect users if they visit sites it doesn't support. Additionally, HTTPS-Everywhere is not supported on Chrome, Safari or Internet Explorer, and its value will be limited until all major browsers support or include this functionality by default.

The work that the EFF has done is commendable, but user education and client-side tools alone will not eliminate website session management vulnerabilities, and it is highly unlikely that people will stop using open networks at cafes, libraries, airports and other public places for Internet access. Website operators should safeguard their users' data privacy regardless of the browser or the types of network people use. By taking protective measures before a sidejacking attack succeeds, companies can avoid losing customers and incurring crippling costs from litigation and redressing negative publicity.

Protecting the Entire User Experience With Always On SSL

Always On SSL is a fundamental, cost-effective security measure that provides end-to-end protection for website visitors. It is not a product, service, or replacement for your existing SSL certificates, but rather an approach to security that recognizes the need to protect the entirety of a user's session, not just the login screen. Always On SSL starts with the site-wide use of HTTPS, but it also means setting the *secure* flag for all session cookies to prevent their contents from being sent over unencrypted HTTP connections. Additional measures, such as Extended Validation (EV) SSL certificates and HSTS, can further strengthen your infrastructure against man-in-the-middle attacks.

Everyone Else is Doing It, and So Should You

As online attacks become more frequent and easier to execute, organizations around the world are under increasing scrutiny to ensure that all online transactions involving confidential data are secure. Government officials and privacy groups are pushing for companies to provide Always On SSL.

[2] https://www.eff.org/press/archives/2011/04/19-0

In January 2011, in response to reports of SSL hacks, Sen. Charles Schumer (D-NY) sent a letter to Yahoo!, Twitter and Amazon[3] to pull up the "welcome mat for would-be hackers" that HTTP presents, and instead asked them to expedite the implementation of Always On SSL.

Today, some of the world's largest and most trusted websites, including Facebook, Google, PayPal and Twitter, have embraced Always On SSL, and are implementing HTTPS to encrypt all communications to and from their websites, including promotional and non-confidential data. These organizations recognize the growing importance of persistent protection, and are working hard to provide a secure experience for online users across the Web.

Facebook

As the most-visited site on the Web,[4] Facebook had 845 million monthly active users at the end of December 2011, with a daily average of 483 million active users during that same month.[5] Facebook is dedicated to protecting people's information, and their security teams have developed sophisticated systems to keep the site safe from spam, phishing, malware and other security threats.[6] In January 2011, as part of a major effort to make its platform more secure for users, Facebook began implementing Always On SSL by introducing the ability for users to browse the site over HTTPS. User response to the change was overwhelming, with over 19 percent of active Facebook users choosing to enable secure browsing.

Orchestrating the Million App Migration

Facebook faced a more daunting challenge in migrating its ecosystem of more than one million developers to HTTPS and OAuth 2.0[7], an open standard co-authored with Yahoo, Twitter, Google, and others. This was a critical endeavor, because many third-party apps would be blocked when users browsed securely if they did not support HTTPS and caused mixed-content security warnings in the browser. To help developers make the transition, Facebook outlined a six-month plan in their Developer Roadmap for sites and apps to migrate to HTTPS.[8]

The transition went quickly for developers that had written their apps with support for secure connections, but for some larger developers with a multitude of applications, it took more time and resources to find and rewrite the necessary code, and to make the necessary infrastructure improvements.

For Facebook, the transition to an Always On SSL approach has been well worth the effort. The company now has a single standard for authentication and apps served through HTTPS, which allows them to provide a simpler, more secure and reliable platform. With this first step complete, Facebook is now focusing efforts on expanding international infrastructure to bring latency down to tolerable levels. This is a critical component for Facebook, since approximately 80 percent of their active users are located outside of North America.

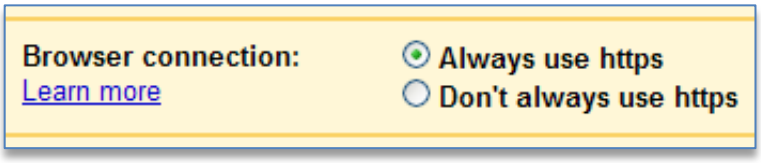
> *"We have a lot more confidence in our ability to deliver a secure, trustworthy service over networks where the privacy of our customers might have otherwise been at risk."*
>
> – Alex Rice, Facebook

[3] http://www.infosecurity-magazine.com/view/16328/senator-schumer-current-internet-security-welcome-mat-for-wouldbe-hackers/
[4] http://www.google.com/adplanner/static/top1000/
[5] http://newsroom.fb.com/content/default.aspx?NewsAreaId=22
[6] https://www.facebook.com/blog/blog.php?post=486790652130
[7] http://oauth.net/2/
[8] https://developers.facebook.com/blog/post/497/

Google

Although Google started out as a search engine that dealt primarily with public information, the company has grown to offer an increasingly customized user experience, and they have long understood the importance of protecting personal information privacy. When Google created Gmail and Google Apps, their intent was to build world-class products so solid they could run their own company on them, so they designed Gmail and Google Apps to support HTTPS from the very beginning. At first, they enforced HTTPS to protect user login information. Then in July 2008, Google rolled out a feature that gave all Gmail and Google Apps users the option to *always use HTTPS*.

Figure 2. Screenshot of HTTPS option in Gmail and Google Apps



In January 2010, Google decided to make HTTPS the default setting for Gmail and Google Apps,[9] making it even easier for users to protect their email between their browsers and Google, all the time. This transition required no additional machines and no special hardware, and the performance impact was negligible. Google's researchers found that SSL/TLS accounts for less than one percent of the CPU load on their production frontend machines—less than 10KB of memory per connection, and less than two percent of network overhead.

Enabling Secure Search

Providing a secure search experience was a more complex undertaking. In May 2010, Google introduced an encrypted search option, giving users the capability to perform searches with better protection against snooping from third parties. More recently, Google began using HTTPS as the default search experience for signed-in users. This change encrypts user search queries and Google's results page.[10] The security benefit of this approach was clear: Users got a more secure and private search experience through the use of end-to-end encryption between their computers and Google. But the ecosystem around search—particularly Web analytics and SEO (search engine optimization)—was more complex.

When users click on a Google search result to visit a website, a *Referrer* flag is supplied by the browser to indicate if the current request was the result of a link from Google. Many SEO practitioners and Web analytics professionals rely on this information to gather usage statistics or to track the impact of their online marketing efforts. However, when users search securely, their query terms are protected. For sites that receive clicks from Google search results, this means that while they will still know that the users came from Google, they won't receive information about each individual query.

However, as Google points out, sites can still receive an aggregated list of the top 1,000 search queries that drove traffic to their site for each of the past 30 days through Google Webmaster Tools. This information helps webmasters keep more accurate statistics about their user traffic, while protecting the confidentiality of individual users who have HTTPS turned on. Furthermore,

[9] http://gmailblog.blogspot.com/2010/01/default-https-access-for-gmail.html
[10] http://googleblog.blogspot.com/2011/10/making-search-more-secure.html

when users choose to click on an ad appearing on Google search results pages, their browsers will continue to send the relevant query over the network so that advertisers can measure the effectiveness of their campaigns and to improve the ads and offers they present.

Going forward, Google plans to add more support for Always On SSL to their products, and their researchers continue to publish information and advice about SSL/TLS. Google is also strongly advocating broader industry efforts to implement SSL more widely and effectively, with the vision of protecting all legitimate content across the Web to ensure a seamless, secure user experience for everyone.[11]

PayPal

Since 1998, PayPal has been the global leader in online payment solutions, and was an early adopter of SSL/TLS. By 2000, the company had already served all pages behind the user login screen via HTTPS, and had begun using HTTPS to protect the login screen itself by 2006. PayPal was also one of the first organizations to deploy Extended Validation (EV) SSL certificates and began implementing EV SSL across all login pages as early as 2007.[12]

PayPal faced unique challenges in regards to performance because of the transactional nature of their services. Unlike sites where users spend a relatively long period of time in a single session, PayPal serves a larger percentage of short sessions. And since the SSL/TLS handshake is the most time-consuming part of the process, they knew they would have to carefully monitor and manage the potential performance impact on the user experience. However, in some cases, the conversion to HTTPS actually sped up the site because they were able to serve content all from the same servers through connections to the browser.

PayPal's security teams were well aware that their site was an especially attractive target for phishers, hackers, and other cybercriminals, and they began taking extraordinary measures to protect their customers and their reputation. Their objective was to thwart phishing scams and active attack tools such as SSLStrip (as opposed to Firesheep, which is a passive tool that eavesdrops on but does not re-route or modify network packets).[13]

These efforts culminated in the release of the HTTP Strict Transport Security (HSTS) specification, co-authored by Jeff Hodges, a security engineer at PayPal.[14] This specification defines a way for websites to declare themselves accessible only via secure connections, and/or for users to be able to interact with given sites only over secure connections. HSTS today is supported by Google Chrome and Mozilla Firefox, and sites such as PayPal.com that use HSTS will explicitly signal to browsers that they will only deliver and accept encrypted communications, preventing users from accidentally visiting an HTTP page or from being directed to HTTP pages via a phishing or SSLStrip attack.

Twitter

As a real-time information network, Twitter has been at the forefront of many news-making events, especially in areas of the world where freedom of expression is restricted. Although Twitter has been the target of a few news-making security incidents, the company has made rapid strides in making Twitter more secure for users.

---

[11] http://googleblog.blogspot.com/2011/10/making-search-more-secure.html
[12] https://otalliance.org/resources/EV/index.html
[13] http://www.thoughtcrime.org/software/sslstrip/
[14] http://tools.ietf.org/html/draft-hodges-strict-transport-sec-02

Twitter already supported HTTPS for its site, smart phone clients, and mobile website, including features such as the Tweet button, but the company soon moved forward with a more ambitious goal of implementing Always On SSL.[15] In May 2011, Twitter announced that they were providing users the option to always use HTTPS via a user defined setting. The overwhelmingly positive user response prompted Twitter to accelerate deployment to all users in January 2012, making HTTPS the default option for all users.

Twitter overcame some unique challenges in adopting Always On SSL. The company outsources some of its traffic to content delivery networks (CDNs) and it was a priority to ensure that the CDNs had the capacity to handle the increased SSL load. Preserving the ability to track referrals and analytics was also of important Twitter and its partners, and Twitter's engineers rewrote code to work around specific issues related to mixed content.

Lessons Learned

Security experts at Facebook, Google, PayPal and Twitter came away with valuable insights that they have shared with OTA as a public service to help website operators protect themselves and their users from sidejacking and other attacks. These insights are important points to consider before you implement Always On SSL for your website.

SSL Is Not Computationally Expensive

Some organizations have been reluctant to implement Always On SSL because they perceive it will increase website operational overhead and costs. SSL/TLS certificates issued by a certificate authority are not free, but they do have a fixed cost, and there is no requirement to replace your existing SSL certificates. If you need to secure multiple domain names, you can do so by adding them to the subject alternative name (SAN) when you purchase your certificates.

Aside from the cost of the SSL certificate, there is the question of computational requirements and the potential need to purchase additional hardware to handle the extra CPU load.

On large and popular websites, it might seem reasonable to assume that the additional computation required to encrypt and decrypt network packets could lead to a substantial increase in hardware requirements. However, many organizations have not necessarily found this to be the case.

Researchers at Google, for example, have performed extensive research on the computational load associated with Always On SSL, and found that it required no additional hardware to implement in their IT environment. Most evidence suggests that technology advancements have minimized the computational impact of SSL/TLS, but it is a good idea to profile the performance of your Web server to see what the performance penalty is for your environment.[16]

> *"If you stop reading now you only need to remember one thing: SSL/TLS is not computationally expensive any more. Ten years ago it might have been true, but it's just not the case any more. You too can afford to enable HTTPS for your users."*
>
> *– Adam Langley, Google*

[15] https://dev.twitter.com/docs/tweet-button/faq
[16] http://www.imperialviolet.org/2010/06/25/overclocking-ssl.html

Network Latency Presents Performance Challenges

The use of end-to-end HTTPS does incur some network latency, largely because of additional round trips required between the client and server to complete the SSL/TLS handshake.[17] This can be a particularly thorny issue over long distances, especially for international users in areas where network bandwidth is limited, or for sites where users initiate a high volume of relatively short SSL/TLS sessions. Latency is not a trivial problem to solve. However, the performance penalty can be managed with proper planning, and financial services companies have already shown they provide rich, low-latency browsing experiences while still implementing strong encryption by default.[18] In addition, Google researchers are experimenting with new technologies such as "False Start," which has been shown to reduce the latency associated with a SSL/TLS handshake by 30 percent.[19]

Secure Web Development Makes the Transition Easier

Secure development practices, if followed from the earliest stages, can result in secure websites and Web applications that cost about the same to develop but are far more cost effective in the long run.[20] Locating and rewriting code can be a costly and time-consuming process, especially for larger organizations with many products. All sites built today should use HTTPS by default and always redirect HTTP connections immediately to HTTPS, especially for Web forms. There are many other aspects to take into consideration; resources such as MozillaWiki and groups such as the Open Web Application Security Project (OWASP) provide comprehensive guidelines that you can follow to build secure Web applications and services.[21]

Mixed Content Creates Complexity

Mixed content is a complicated issue that arises from the hyperlinked nature of the Internet. Most websites display content from multiple sources, often from third parties. If any of this content is linked to with an HTTP link, it could compromise the security of an otherwise secure site by enabling an active attacker to exploit the loading of a cascading style sheet or JavaScript code.[22] Similarly, when an HTTPS page loads an image, iframe, or font over HTTP, a man-in-the-middle attacker can intercept the HTTP resource. Additionally, sites such as Facebook are beginning to require the use of SSL/TLS to avoid mixed content, and will block apps and content that doesn't use HTTPS.[23] To avoid these problems, websites must avoid calling files via HTTP in their code. This includes, but is not limited to, the following elements:

- Image files and links to them in the <img> tag
- External CSS (.css) files
- JavaScript (js.) files
- Embedded media and iframe content (Flash, etc.)
- URLs in your DOCTYPE or <html> tags
- Calls to external APIs and SDKs (e.g., the Facebok SDK)

[17] http://www.semicomplete.com/blog/geekery/ssl-latency.html
[18] http://www.wired.com/images_blogs/threatlevel/2009/06/google-letter-final2.pdf
[19] http://googleonlinesecurity.blogspot.com/2010/05/extending-ssl-to-google-search.html
[20] https://www.owasp.org/index.php/OWASP_Guide_Project
[21] https://wiki.mozilla.org/WebAppSec/Secure_Coding_Guidelines
[22] https://www.eff.org/https-everywhere/deploying-https
[23] http://googleonlinesecurity.blogspot.com/2011/06/trying-to-end-mixed-scripting.html

Relative links are one way to avoid the issue of mixing secure and non-secure content because they don't specify either HTTP or HTTPS. However, relative links can potentially be exploited for search engine spamming or "302 hijacking" attacks, so your must consider your needs carefully when deciding how and where to use relative links. [24]

Implementing Always On SSL for Your Website

Companies who are serious about protecting their customers and their business reputation long term will implement Always On SSL. OTA has outlined the steps you can take to implement Always On SSL and protect your users. The level of protection and assurance you can provide depends on the security features you choose to implement, as summarized in Table 1.

Table 1. Summary of Always On SSL Security Measures

Security Feature	Good	Better	Best
Persistent HTTPS	✓	✓	✓
Cookies Set With *Secure* Flag	✓	✓	✓
Persistent HTTPS with Extended Validation		✓	✓
HTTP Strict Transport Security (HSTS)			✓

Enforce Persistent HTTPS on Every Web Page

First and foremost, Always On SSL is about making it as easy as possible for visitors to *always use HTTPS* when they are on your website, no matter what page they are on. The HTTPS protocol is the same text based protocol as HTTP, except that it runs over an encrypted SSL/TLS session. Here are a few the steps you must take to enforce HTTPS:

- Install an SSL/TLS certificate from a third-party certificate authority
- Switch from Port 80 to Port 443 for all connections to the Web server
- Specify the encryption strength (e.g., 128-bit).

Initially, you may choose to enable this as an optional feature for your users. But in the long run, the recommended practice is to make HTTPS the default, and to give users the option to disable it if needed. By enforcing site-wide use of HTTPS, you can provide the minimum level of security required to make meaningful security or privacy guarantees to your users.

Ensure Correct Implementation of Your SSL Certificates

To enable HTTPS, you should use a valid SSL/TLS certificate from a third-party certificate authority (CA). While a self-signed certificate will encrypt communications between the user and website, only a certificate issued by a CA tells your customers that the domain's identity has been verified by a trusted source. If the connection uses a self-signed certificate, the Web browser may identify it as a potential risk and display an error message warning the user that the site may not be safe to visit. Selecting the right certificate authority is very important.

[24] http://www.dummies.com/how-to/content/prevent-someone-from-hijacking-your-web-sites-sear.html

Make sure the CA you work with maintains strict security practices and has a robust infrastructure with good customer support. Additional considerations include the CA's authentication and certificate issuance practices, the reliability and speed of its certificate validation systems, an annual audit report from a well-known, independent auditing firm, and a warranty that the CA will live up to its representations and commitments.

In addition, make sure that your SSL certificate includes all intermediate certificates in the chain of trust. Problems with your certificate can cause many Web browsers to block users from accessing your site, or to display a security warning message when your site is accessed. Sites like Facebook that enforce strict validation may also block your content from their users if your certificate chain has problems. There are several third-party SSL analysis tools available that you can use to check your SSL/TLS implementation and fix any errors or warnings.

Set the Secure Flag for All Session Cookies

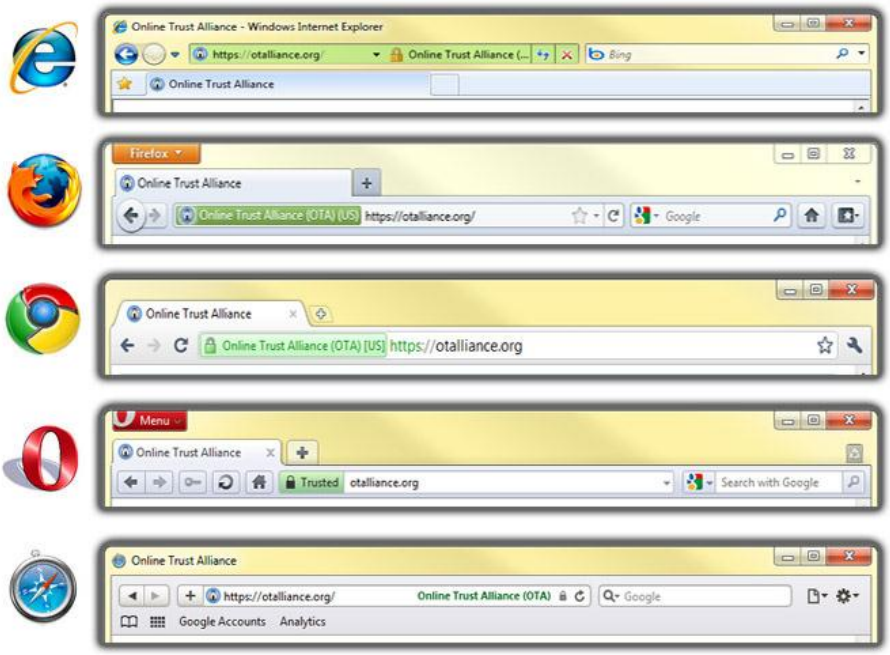
A session cookie can be set with an optional "secure" flag, which tells the browser to contact the origin server using only HTTPS whenever it sends back this cookie. The Secure attribute should be considered security advice from the server to the user agent, indicating that it is in the session's interest to protect the cookie contents. This measure helps prevent cookies from being sent over HTTP, even if the user accidentally makes (or is tricked into making) a browser request to the Web server via HTTP.

Enhance Security and Trust with Extended Validation Certificates

For stronger protection against exploits such as SSLStrip, the OTA recommends that websites consider deploying Extended Validation (EV) SSL certificates. EV secured sites undergo a rigorous verification process established by the CA Browser Forum, a collaboration of more than 30 leading certification authorities and browser software vendors.

This verification process confirms the identity and existence of website operators using reliable third party sources. Users visiting a website secured with EV SLL certificates will see a green bar and the organization's name in the URL bar, providing visual reassurance of the website operator's identity.

Figure 3. Web browsers displaying the use of EV SSL certificates



The OTA recommends that responsible organizations deploy an EV certificate on any site requiring a secure connection. IT departments should assist management and end users in understanding that EV certificates will protect the security of users and reduce the organization's vulnerability to attacks. All users should upgrade to browsers that support EV certificates and every website conducting online transactions should evaluate EV certificates as part of their security and brand protection strategy.

Implement HSTS to Prevent Active Attacks

HTTPS connections are often initiated when visitors are redirected from an HTTP page or when they click on a link (such as a login button) that directs them to an HTTPS site. However, it is possible to launch a man-in-the-middle attack during this transition from an unsecured connection to a secure one, either passively or by tricking a victim into clicking an HTTP link to a legitimate website (via a phishing email, for example).

The strongest defense against these types of attacks is to implement HTTP Strict Transport Security (HSTS) for your website. This specification defines a way for websites to declare themselves accessible only via secure connections, and/or for users to be able to interact with given sites only over secure connections. HSTS is supported by Google Chrome and Mozilla Firefox, and sites such as PayPal.com that use HSTS will explicitly signal to browsers that they will only deliver and accept encrypted communications.[25] Using HSTS helps to prevent attackers from stealing session cookies when users are redirected from HTTP to HTTPS, and is currently the strongest defense against phishing and man-in-the-middle attacks.

[25] http://hacks.mozilla.org/2010/08/firefox-4-http-strict-transport-security-force-https/

Conclusion

In the past, many experts have advised website developers and operators to use SSL/TLS to protect user authentication, financial transactions, and other key activities, but many organizations were hesitant to encrypt their entire sites because of concerns about cost, performance and other issues. However, the Internet has reached a tipping point where it is clear that selective use of HTTPS is no longer adequate to protect today's mobile, always-online users. SSL/TLS itself is still fundamentally sound, but Firesheep was a clarion call for website operators to protect the entire user experience, not just the login page or the shopping cart. Simply put, SSL is like a safety belt in an automobile: It should always be on in transit.

Always On SSL is not a "silver bullet" for stopping hijackers, and must be implemented as part of an overall security strategy for protecting users when they interact with your website. Nevertheless, it is a proven approach to stopping sidejacking and other man-in-the-middle attacks, and one that is no longer computationally expensive for the vast majority of organizations. As Facebook, Google, PayPal, Twitter and others have demonstrated, it is possible for even the largest and most complex websites to deliver a rich user experience over HTTPS. Issues such as latency and mixed content can present challenges, but the guidelines and best practices outlined in this white paper will help you manage these issues and optimize performance for your users.

More importantly, Always On SSL can help you protect the trust that users have in your website. Protecting trust and consumer confidence is a very difficult problem that cannot be solved through purely technical means. At some level, users simply have to trust in the system, and as Ken Thompson, one of the principal authors of UNIX, once wrote, "perhaps it is more important to trust the people who wrote the software."[26] Taking an Always On SSL approach to security can help you give users the assurance of knowing you take their security and privacy seriously, and that you are taking reasonable steps to protect them.

About The Online Trust Alliance (OTA)

OTA is an independent non-profit with a mission to develop and advocate best practices and public policies which mitigate emerging privacy, identity and security threats to online services, organizations and consumers, thereby enhancing online trust and confidence. By facilitating an open dialog with industry, business and governmental agencies to work collaboratively, OTA is making progress to address various forms of online abuse, threats and practices that threaten to undermine online trust and increase the demand for regulations.

https://www.otalliance.org/

[26] http://www.ece.cmu.edu/~ganger/712.fall02/papers/p761-thompson.pdf

