HTTPS Everywhere for Internet Explorer

Table of Contents

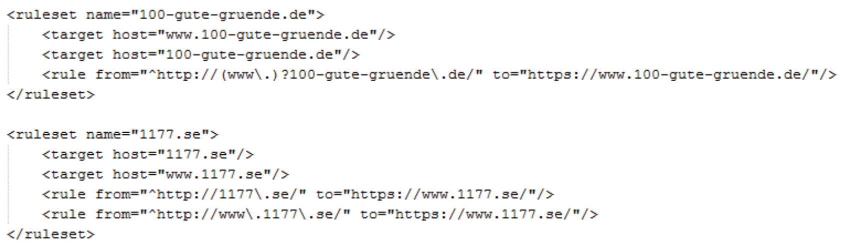
HTTPS Everywhere extension

The Electronic Frontier Foundation (EFF) created the HTTPS Everywhere extension for Firefox and Google Chrome. This browser extension forces the browser to used the HTTPS version of websites when possible. More details are available at https://www.eff.org/https-everywhere.

There are three main features in the extension:

1. Translate HTTP URLs to HTTPS URLs according to a set of rules
2. Secure HTTPS cookies according to a set of rules
3. Add support for STS

Switching from HTTP to HTTPS is not as easy as it. Many websites use different domains with SSL enabled, some pages may not support SSL, etc. The XML rules are required to allow the extension when to substitute HTTPS to HTTP, and how to format the URL.



```
<ruleset name="100-gute-gruende.de">
    <target host="www.100-gute-gruende.de"/>
    <target host="100-gute-gruende.de"/>
    <rule from="^http://(www\.)?100-gute-gruende\.de/" to="https://www.100-gute-gruende.de/"/>
</ruleset>

<ruleset name="1177.se">
    <target host="1177.se"/>
    <target host="www.1177.se"/>
    <rule from="^http://1177\.se/" to="https://www.1177.se/"/>
    <rule from="^http://www\.1177\.se/" to="https://www.1177.se/"/>
</ruleset>
```

Figure 1. Examples of (simple) rules

While HTTPS Everywhere cannot enforce the use of HTTPS on all websites, it goes a long way in ensuring secured connections are used whenever possible.

HTTPS Everywhere for Internet Explorer

Zscaler has worked with the EFF to port the HTTPS Everywhere extension to Internet Explorer. This document describes the features available in the Internet Explorer port, the architecture of the extension, its behavior and limitations.

Requirements

HTTPS Everywhere for Internet Explorer is supported on:

1. Windows XP SP3 to Windows 8 (32-bit and 64-bit)
2. Internet Explorer 6 to 10, 32-bit only
3. Administrator right

Download

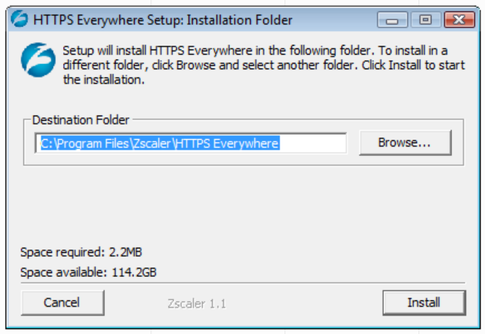
You can download the latest version for HTTPS Everywhere for Internet Explorer at …. The executable is an installer than contains:

- *HTTPSEverywhere.dll*: the extension (Browser Helper Object)

- *rules/default.ruleset*: the default rules from the EFF
- *uninstall.exe*: uninstaller

Installation

Run the installer *https-everywhere.exe*. You will be asked where to store the extensions and the rules. You need to give Administrator access to the installer to register the extension with Internet Explorer.



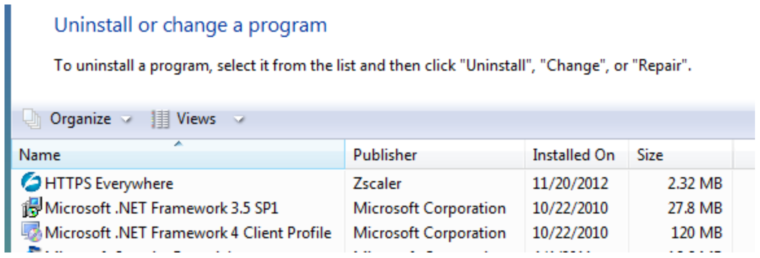
Setup will install HTTPS Everywhere in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.

Destination Folder: C:\Program Files\Zscaler\HTTPS Everywhere

Space required: 2.2MB
Space available: 114.2GB

Figure 2. Installation of HTTPS Everywhere

Restart Internet Explorer.



To uninstall a program, select it from the list and then click "Uninstall", "Change", or "Repair".

Name	Publisher	Installed On	Size
HTTPS Everywhere	Zscaler	11/20/2012	2.32 MB
Microsoft .NET Framework 3.5 SP1	Microsoft Corporation	10/22/2010	27.8 MB
Microsoft .NET Framework 4 Client Profile	Microsoft Corporation	10/22/2010	120 MB

Figure 3. HTTPS Everywhere installed on Windows

You should be prompted by Internet Explorer to allow HTTPS Everywhere (check at the bottom of the screen). Accept.

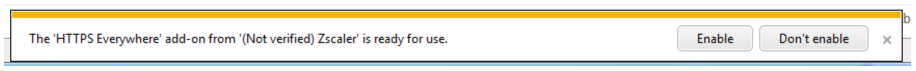
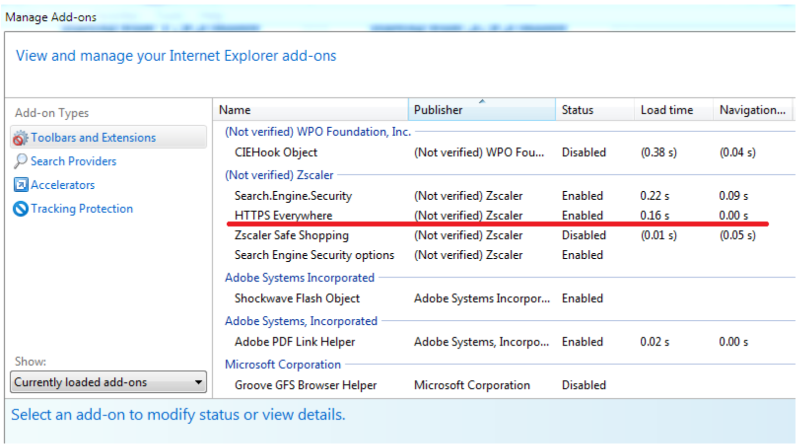

The 'HTTPS Everywhere' add-on from '(Not verified) Zscaler' is ready for use.

Figure 4. Promt to enable HTTPS Everywhere

You can check that the extension has been correctly installed by going to *Tools – Manage Add-ons* and verify that *HTTPS Everywhere* is shown as *Enabled*.



Manage Add-ons

View and manage your Internet Explorer add-ons

Add-on Types
- Toolbars and Extensions
- Search Providers
- Accelerators
- Tracking Protection

Show: Currently loaded add-ons

Name	Publisher	Status	Load time	Navigation...
(Not verified) WPO Foundation, Inc.				
CIEHook Object	(Not verified) WPO Fou...	Disabled	(0.38 s)	(0.04 s)
(Not verified) Zscaler				
Search.Engine.Security	(Not verified) Zscaler	Enabled	0.22 s	0.09 s
HTTPS Everywhere	(Not verified) Zscaler	Enabled	0.16 s	0.00 s
Zscaler Safe Shopping	(Not verified) Zscaler	Disabled	(0.01 s)	(0.05 s)
Search Engine Security options	(Not verified) Zscaler	Enabled		
Adobe Systems Incorporated				
Shockwave Flash Object	Adobe Systems Incorpor...	Enabled		
Adobe Systems, Incorporated				
Adobe PDF Link Helper	Adobe Systems, Incorpo...	Enabled	0.02 s	0.00 s
Microsoft Corporation				
Groove GFS Browser Helper	Microsoft Corporation	Disabled		

Select an add-on to modify status or view details.

Figure 5. HTTPS Everywhere enabled in Internet Explorer

Home pages

The extensions may be loaded too late by Internet Explorer to protect the first URL accessed, the home page(s). To provide full protection, the installer modifies the home page(s) to the secure version. For example, if the home page was http://www.google.com/, it will be changed to https://encrypted.google.com/.

Test

To make sure that the extension is working, type www.google.com in the address bar. It should be changed to https://encrypted.google.com/. If it does not work and you have just installed the extension, restart Internet Explorer.

Updates

You can check if new versions are available at …. You will also get notified when updates are available: a new tab will be open to ….

If you download an update, make sure Internet Explorer is closed before running the installer. If it is still open, you will not be able to override the extension with the new version.

Customize

There are a few settings that can be modified in the registry. These settings can be found under

- Windows XP: *HKEY_CURRENT_USER\Software\Classes\Software\44D1BC7D-F859-4CB3-9E1D-D1ED52181916*
- Windows Vista and newer: *HKEY_CURRENT_USER\Software\AppDataLow\Software\44D1BC7D-F859-4CB3-9E1D-D1ED52181916*

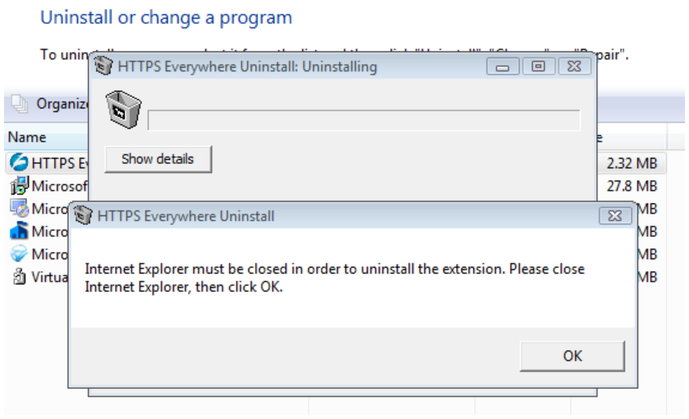
The settings are:

- *LastSoftwareUpdate*: last time a successful check for a new version was done (epoch time)
- *SoftwareUpdateInterval*: how often to check for a new version of HTTPS Everywhere, in seconds

Uninstallation

If you want to disable HTTPS Everywhere for Internet Explorer, open Internet Explorer. Go to *Tools* > *Manage Add-ons*. Select *HTTPS Everywhere* and click on *Disable*.

To remove completely HTTPS Everywhere from your computer, go to the *Control Panel* > *Programs* > *Uninstall a Program*. Select *HTTPS Everywhere* and click on *Uninstall*.


Uninstall or change a program
HTTPS Everywhere Uninstall: Uninstalling
Show details
HTTPS Everywhere Uninstall
OK

Internet Explorer must be closed in order to uninstall the extension. Please close Internet Explorer, then click OK.

Figure 6. Uninstall HTTPS Everywhere

Architecture

The design of HTTPS Everywhere for Internet Explorer is quite different from the Firefox and Chrome version because the Microsoft browser does not offer a powerful extension API like the two other vendors.

Challenges

There were a number of challenges in the port from Firefox and Chrome to Internet Explorer. First, the extension had to be written in C++. An earlier attempt in C# showed that too many features are not available to .Net extensions.

The main challenge was the fact that there is no event fired by internet Explorer for all HTTP connections. OnBeforeNavigate is triggered only for the main page, not for its elements (images, scripts AJAX calls, etc.).

The URLs cannot be changed within the regular Browser Helper Object; they have to be changed at a lower level, in WinInet.

WinInet

WinInet is the standard Windows library to handle FTP/HTTP/HTTPS. Internet Explorer uses this library to access the web.

In order to intercept HTTP requests and responses, HTTPS Everywhere for Internet Explorer hooks the WinInet library inside Internet Explorer to replace calls to the WinInet API by its own code.

Three WinInet functions are intercepted by HTTPS Everywhere:

- InternetConnect
- HTTPOpenRequest
- InternetCloseHandle

InternetConnect

This function gives the domain name used for the HTTP/HTTPS request. If there is a rule match for this host name, the connection handle and the host name are saved for later.

HTTPOpenRequest

This function gives the URL path and the scheme (HTTP or HTTPS). The extension checks if the connection handle was saved earlier (meaning we have a host match). If the connection handle was saved, and the URL is using HTTP, a lookup into the HTTPS Everywhere rules is done on the full URL.

If the URL must be transformed, a new HTTPS request is done. For example, if the request was for http://www.google.com/, a new request is made to https://encrypted.google.com/. This new connection is substituted to the original connection. The handle of the new connection is returned.

Because the request is being modified at a lower level than the BHO framework, we need a way to tell Internet Explorer that the URL has been changed. Otherwise, Internet Explorer would display http://ww.google.com/ in the URL bar even though the request is actually made to https://encrypted.google.com/. The new connection is actually not done directly to https://encrypted.google.com/, but rather to an internal server (see more details below) that is going to send a HTTP 302 redirection to the final URL https://encrypted.google.com/. Using the adequate WinInet flag, Internet Explorer sees the 302 redirection, follows the *Location* header, and displays the new URL in the address bar.

One of the problems we faced while testing HTTPS Everywhere for Internet Explorer is that Internet Explorer does not always follow redirections. Instead of following the 302 redirection to retrieve an image, a JavaScript file or style sheet (CSS), Internet Explorer aborts the request. To get around this problem, we use different WinInet flags to hide the redirection. This means Internet Explorer believes it is following the original URL (http://unsecure.com/image.png for example) when it is actually accessing a secure URL (https://secure.com/image.png for example).

In order to know when to notify or not Internet Explorer of the redirection, we have to know whether the main URL (address seen in the address bar) is being requested, or if it is a page element (image, script, CSS, etc.). Fortunately, Internet Explorer sends a different *Accept* header for the main request than the others. The value changes with the different versions of Internet Explorer.

InternetCloseHandle

We do the connections clean up here. The original connection, before the substitution, is closed. We removed any data saved previously about the connection.

Secure cookies

There are rules to secure cookies, that is to add the "*Secure*" attribute to cookies set by the server to make sure they are not sent later in plain text over HTTP.

The cookies are modified in the InternetCloseHandle. If a cookie rule matches the domain, and the cookie is not sent with the secure attribute, it is overwritten by new cookie (same data) with the secure attribute.

Whitelisting

A web server may refuse to serve a secure URL (HTTPS), and may keep redirect the browser to HTTP. To avoid infinite loops, the extension gives up on redirecting to the secure URL is there are more than 5 redirections for the same URL within 5 seconds inside the same tab (values may change in the future).

Internal server

A very simple local HTTP server is created for each tab on different ports. The server only listens to 127.0.0.1. Its only feature is to send a 302 redirection with the Location value set to the value of the X-Redirect header sent in the request. Here is an example of a request and response from the extension to the internal server:

```
GET /redirect HTTP/1.0
X-Redirect: https://encryped.google.com/
User-Agent: HTTPSEVERYWHERE

HTTP/1.0 302 Found
Location: https://encryped.google.com/
```

We call it "HTTP Web server", but it is not capable of accessing the file system or doing anything else that these 2 lines.

Page modification

In addition to transforming URLs according to rules at the network level, inside WinInet, URLs are also changed in the HTML source code of the page. This is done mostly to avoid the Mixed content warning (see more details below) from Internet Explorer.

The modification of the URLs (in the *href* and *src* attributes) is done several times in attempt to make the modification as early as possible, but also when the document is complete.

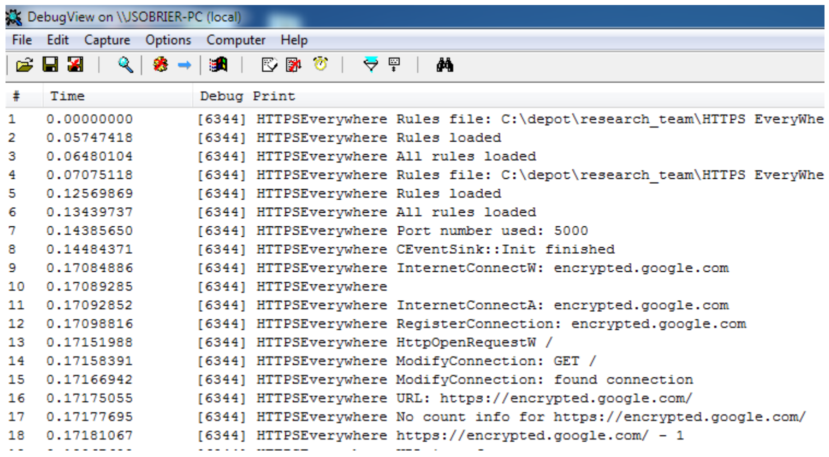
Debugging

You can use DebugView from Microsoft to get the debug messages from HTTPS Everywhere. Make sure you run it as Administrator. There are three levels of messages:

- Debug: informational

- Warning: expected errors
- Error: unexpected errors

All messages are preceded by “HTTPSEverywhere”. The output is currently quite noisy.


DebugView on \\USOBRIER-PC (local)
File Edit Capture Options Computer Help

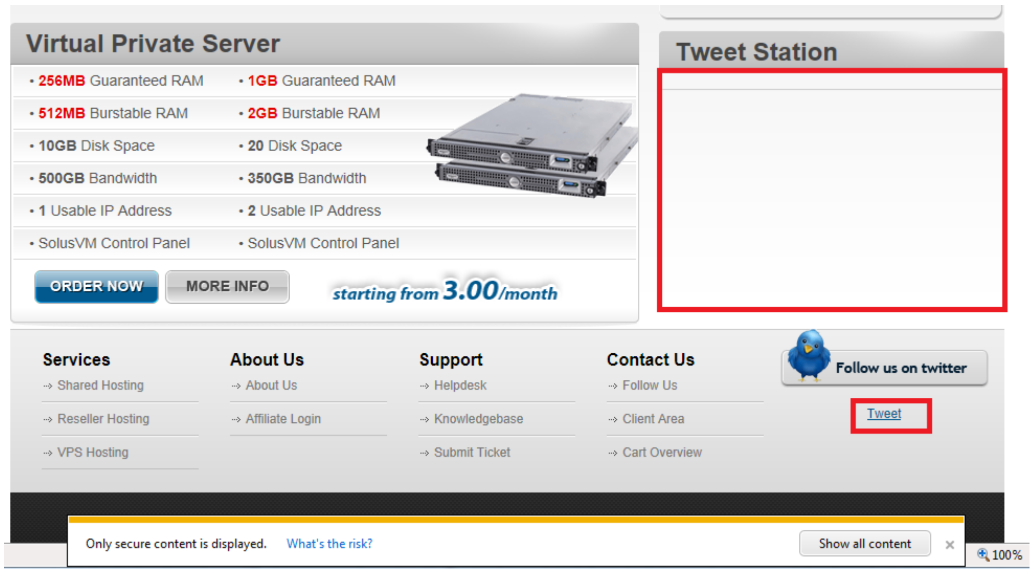
```
#   Time        Debug Print
1   0.00000000  [6344] HTTPSEverywhere Rules file: C:\depot\research_team\HTTPS EveryWhe
2   0.05747418  [6344] HTTPSEverywhere Rules loaded
3   0.06480104  [6344] HTTPSEverywhere All rules loaded
4   0.07075118  [6344] HTTPSEverywhere Rules file: C:\depot\research_team\HTTPS EveryWhe
5   0.12569869  [6344] HTTPSEverywhere Rules loaded
6   0.13439737  [6344] HTTPSEverywhere All rules loaded
7   0.14385650  [6344] HTTPSEverywhere Port number used: 5000
8   0.14484371  [6344] HTTPSEverywhere CEventSink::Init finished
9   0.17084886  [6344] HTTPSEverywhere InternetConnectW: encrypted.google.com
10  0.17089285  [6344] HTTPSEverywhere
11  0.17092852  [6344] HTTPSEverywhere InternetConnectA: encrypted.google.com
12  0.17098816  [6344] HTTPSEverywhere RegisterConnection: encrypted.google.com
13  0.17151988  [6344] HTTPSEverywhere HttpOpenRequestW /
14  0.17158391  [6344] HTTPSEverywhere ModifyConnection: GET /
15  0.17166942  [6344] HTTPSEverywhere ModifyConnection: found connection
16  0.17175055  [6344] HTTPSEverywhere URL: https://encrypted.google.com/
17  0.17177695  [6344] HTTPSEverywhere No count info for https://encrypted.google.com/
18  0.17181067  [6344] HTTPSEverywhere https://encrypted.google.com/ - 1
```

Figure 7. Output from DebugView

Mixed content warning

Some website may contain external elements (images, CSS, scripts) accessible through HTTP even when the main page is requested over HTTPS. This may happen if some URLs do not have rules to be transformed into a secure alternative, or if Internet Explorer has detected the HTTP scheme before the extension had a chance to modify the URL.

For example, on https://123systyems.net/, a warning is shown because the Twitter widget URL is not modified early enough. I have highlighted in red the content not displayed by Internet Explorer:


Virtual Private Server

• 256MB Guaranteed RAM	• 1GB Guaranteed RAM
• 512MB Burstable RAM	• 2GB Burstable RAM
• 10GB Disk Space	• 20 Disk Space
• 500GB Bandwidth	• 350GB Bandwidth
• 1 Usable IP Address	• 2 Usable IP Address
• SolusVM Control Panel	• SolusVM Control Panel

ORDER NOW MORE INFO starting from 3.00/month

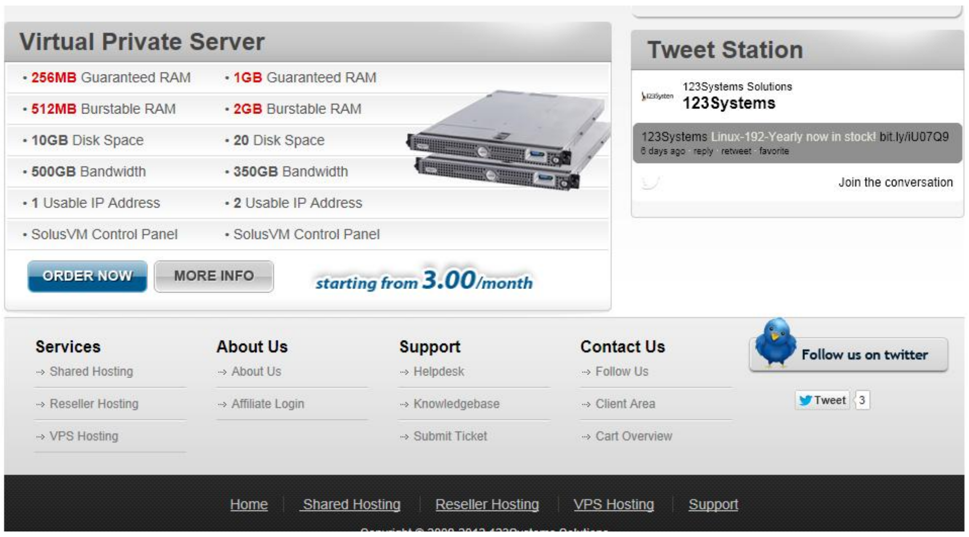
Tweet Station

Services: Shared Hosting, Reseller Hosting, VPS Hosting
About Us: About Us, Affiliate Login
Support: Helpdesk, Knowledgebase, Submit Ticket
Contact Us: Follow Us, Client Area, Cart Overview
Follow us on twitter
Tweet

Only secure content is displayed. What's the risk? Show all content ×

Figure 8. Warning for Internet Explorer and missing content

It is important to remember that URLs are always modified at the network level (WinInet), and that Internet Explorer is not always notified of the change. The user can safely click on “Show all content”. Although Internet Explorer believes it is retrieving the Twitter scripts over HTTP, it is actually accessing Twitter over HTTPS. You can use a proxy like Fiddler to check that no HTTP connection is made to Internet.


Virtual Private Server

• 256MB Guaranteed RAM	• 1GB Guaranteed RAM
• 512MB Burstable RAM	• 2GB Burstable RAM
• 10GB Disk Space	• 20 Disk Space
• 500GB Bandwidth	• 350GB Bandwidth
• 1 Usable IP Address	• 2 Usable IP Address
• SolusVM Control Panel	• SolusVM Control Panel

ORDER NOW MORE INFO starting from 3.00/month

Tweet Station
123Systems Solutions
123Systems
123Systems Linux-192-Yearly now in stock! bit.ly/iU07Q9
6 days ago · reply · retweet · favorite
Join the conversation

Services: Shared Hosting, Reseller Hosting, VPS Hosting
About Us: About Us, Affiliate Login
Support: Helpdesk, Knowledgebase, Submit Ticket
Contact Us: Follow Us, Client Area, Cart Overview
Follow us on twitter
Tweet 3

Home | Shared Hosting | Reseller Hosting | VPS Hosting | Support

Figure 9. Twitter content retrieved securely

Coming next

Here is a list, in random order, of features and improvements we are working on:

- Faster loading time

- Support for Internet Explorer 64-bit
- Support for HSTS
- Use rules from HTTPS Everywhere 3.0
- Option to use custom rules
- UI to select enable/disable rules
- UI to show active rules on the page
- UI to disable extension on specific domains/URLs
- Automatic update of default rules
- Option to not modify the home pages during the installation
- No Administrator right required
- One Click deployment
- Option to disable debug messages
- Option to choose the port range for the internal servers

The source code will be available on the EFF website.

Notes

Here are a couple of notes and comments that did not fit in the previous sections.

The installer is created with NSIS. The installation does not look as good sexy as InnoSetup, but the resulting executable is a lot smaller.

The best way to test a browser extension on multiple OS and version of Internet Explorer is to use Microsoft Virtual PC and the free Internet Explorer images. Despite what the websites says, VPC works fine on Windows 7 Home edition. However, I had to use VirtualBox to try Windows 8.

The performance numbers (load time and navigation time) given by Internet Explorer seem to be very unreliable and changed often on my set up.

Internet Explorer 10 Metro (not the Desktop version) on Windows 8 does not load any plugin (some exceptions for Flash) or browser extension. You have to use the Desktop version to load HTTPS Everywhere.

Version history

0.0.0.1: first version released to the public

Author

Julien Sobrier is a senior Security Researcher at Zscaler. He initiated the port of HTTPS Everywhere to Internet Explorer in 2012.