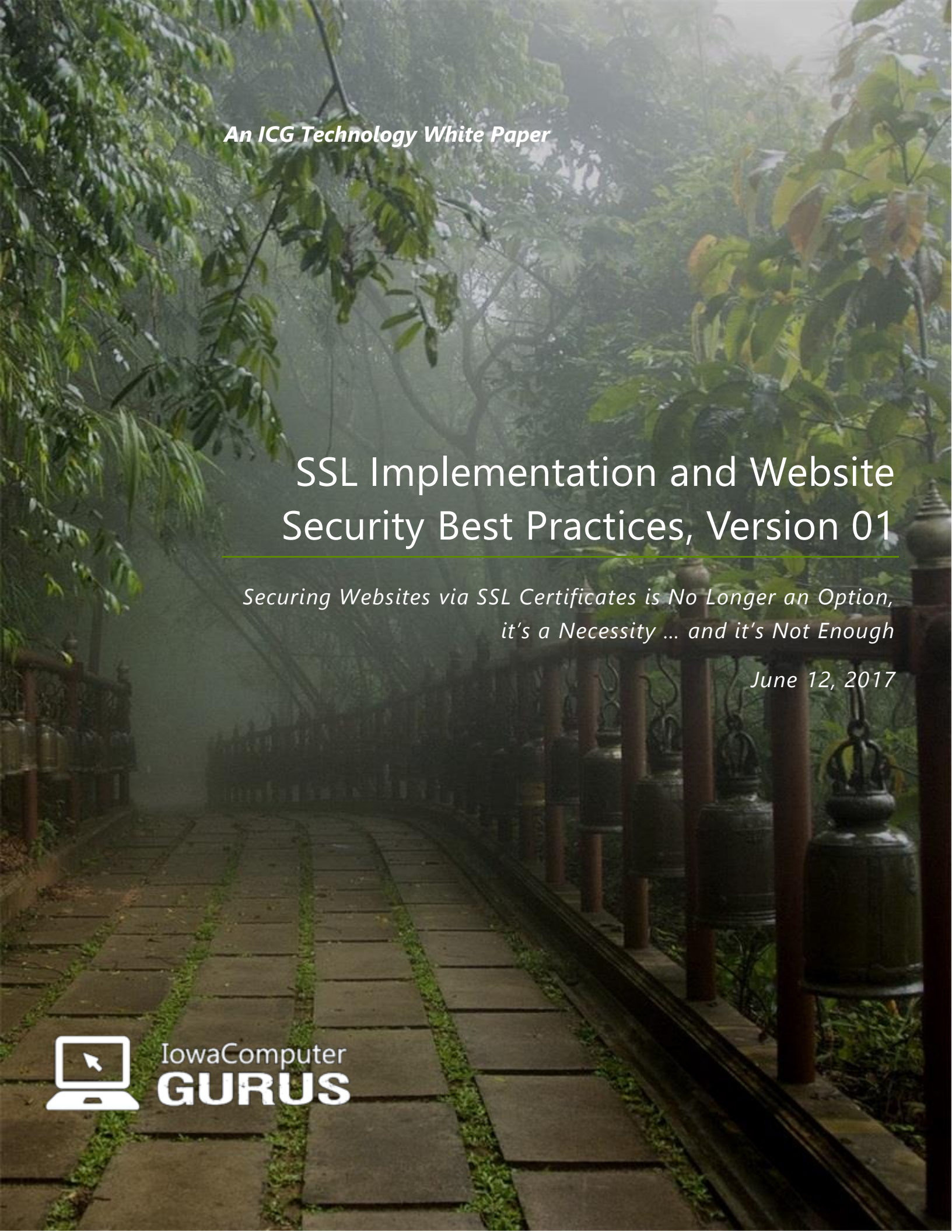


An ICG Technology White Paper

SSL Implementation and Website Security Best Practices, Version 01

Securing Websites via SSL Certificates is No Longer an Option, it's a Necessity … and it's Not Enough

June 12, 2017

IowaComputer GURUS

Contents

The Purpose of This Document

This document has been created to provide a baseline set of information and best practices related to the use and implementation of SSL certificates and other best practices on websites to enhance security. This document is not intended to be an introductory course on website security. Neither is it a comprehensive set of website security guidelines. Rather, it is designed to aid in the application and implementation of SSL protocols and other basic security best practices as fundamental components of secure website policy.

Who Should Use This Document

This document is intended to be used by website developers with at least an intermediate understanding of general website design and development principles. It may also be of value to general website owners seeking a basic understanding of the principles of SSL and website security best practices.

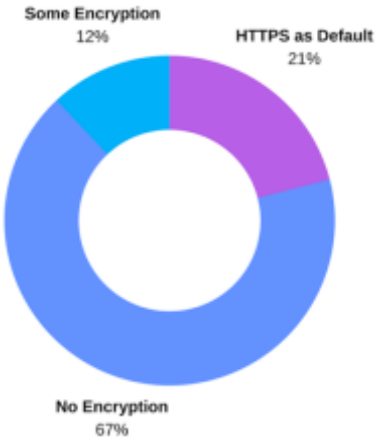
Document Revision History

Rev. Date	Notes
6/12/2017	First public release, Version 01.

Website Security and SSL – Background

Website security has become the fundamental concern of the internet. According to recent statistics, there are well-over 1 billion websites,[1] most of them relying on out-of-date platforms and utilizing lax security. In fact, in late 2016, Google issued a transparency report[2] that detailed 79% of the top websites did not deploy HTTPS by default – one of the most basic encryption security protocols – and 67% used outdated encryption or none at all.

And the level of security threats is growing every year. Computer security market leaders McAfee[3] and Symantec[4] have both recently issued threat assessment reports that suggest that all forms of internet security are under persistent and growing levels of attack. Further, these attacks can be addressed through the vigilant application of best practices and coordination.



This combination of lax security and heightened threat environment has prompted industry leaders to take what are – by tech industry standards – some rather drastic actions.

- In 2016 – recognizing that vast numbers of websites that were not employing SSL/TLS security – the Internet Security Research Group (ISRG) adopted a mission to reduce the "financial, technological, and educational barriers to secure communication over the internet." Their first major initiative is Let's Encrypt – a free, automated, and open SSL certificate authority (CA) being run for the general public benefit.[5] This initiative is broadly supported by major organizations.
- Google declared that by January 2017 websites not protected by basic SSL encryption would be penalized within Google Search by identifying them as non-secure.[6] Since Google Search accounts for 75% of search traffic, this move has put additional pressure on website owners to implement SSL.
- In addition – suspecting that some older Symantec-issued SSL certificates were either not current or did not meet existing standards – Google threatened to cease recognition of those certificates in their Chrome browser. With a commanding market share of more

[1] Most recent Netcraft News report as of this writing, May 26, 2017.
[2] "HTTPS on Top Sites" – a dynamic grid viewed for this document on June 8, 2017: https://www.google.com/transparencyreport/https/grid/
[3] McAfee Labs 2017 Threats Predictions: https://www.mcafee.com/us/resources/reports/rp-threats-predictions-2017.pdf
[4] Symantec Internet Security Threat Report 2017: https://www.symantec.com/content/dam/symantec/docs/reports/istr-22-2017-en.pdf
[5] You can read our overview of Let's Encrypt in our article dated June 5, 2017: https://www.iowacomputergurus.com/Resources/Blog/Post/398/Introduction-to-Let-s-Encrypt-Website-Security-Free-TLS-SSL-Certificates-for-Everyone
[6] Google Security Blog dated September 8, 2016, and viewed June 8, 2017: https://security.googleblog.com/2016/09/moving-towards-more-secure-web.html

than 60%,[7] this action brought Symantec to the table where they have agreed to update their related practices and procedures.

- Mozilla quickly followed suit, implementing a visual identifier for websites that were not properly secured with SSL and collected passwords.[8] As the number two browser by market share, this brought the total weight of non-SSL identification pressure in browsers to over 86%.

Clearly, everyone recognizes that website security is THE technology issue of the day. But implementing website security is not as simple as just acquiring an SSL/TLS security certificate. There are several additional adjustments that – if not made properly – can interfere with the performance, overall security, and SEO asset value of your website.

It is in the context of this environment that we have decided to create this website security and SSL Best Practices document.

Choosing an SSL Certificate

All SSL certificates are issued by a Certificate Authority (CA). The CA becomes a trusted third party that issues certificates and then validates/authorizes that certificate when a request is made to a website that has been issued a certificate.

The user clicks a link or types a URL into their browser. A connection is made to the website's server requesting the site (handshake). Simultaneously, certificate code implemented on the site authenticates against a database maintained by the CA. The net result is that the user's browser receives the website data and a validation of encryption from the CA. This allows the website URL to be displayed as "https:" as opposed to "http:" along with other security identifiers such as green coloring in the address bar and/or lock icons.

Types of Certificates

Generally speaking, there are four kinds of SSL certificates:

- Domain Validation (DV)
- Organization Validation (OV)
- Extended Validation (EV)
- Wildcard

[7] Reference ArsTechnica report dated May 5, 2017, and viewed on June 8, 2017: https://arstechnica.com/business/2016/05/firefox-overtakes-microsoft-internet-explorer-edge-browsers-first-time-statcounter/

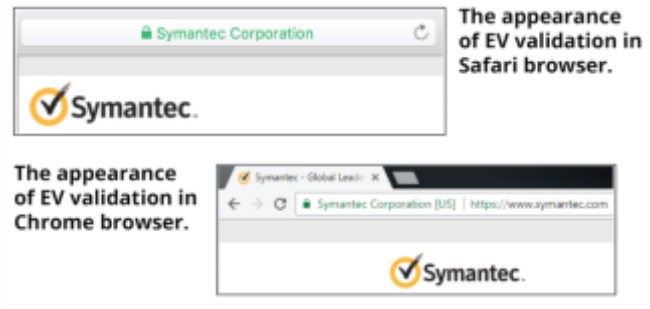
[8] Mozilla Security Blog dated January 20, 2017, and viewed on June 8, 2017: https://blog.mozilla.org/security/2017/01/20/communicating-the-dangers-of-non-secure-http/

Domain Validation (DV) Certificates

DV certificates are the most common and their administration is often automated. In addition to indicating that a website is using basic encryption, this level of validation has a single vetting criteria – that a specific individual has been identified who has the right to administrate the domain.

Organization Validation (OV) Certificates

OV certificates identify that an individual has been identified with the right to administer the domain and that the domain is controlled by a legal entity. This could be a Limited Liability Company (LLC) or some form of incorporation that is officially registered.



Extended Validation (EV) Certificates

EV certificates have the same requirements as OV certificates, but differ in two ways: the administrator and legal entity are validated by a human responsible party at the CA and the legal entity is displayed in the address bar while the user is on the site. This is considered the highest form of SSL verification.

Wildcard Certificates

Wildcard certificates are special versions of DV certificates that are designed to be used across multiple, first-level sub-domains. For example, if your primary domain was "sample.com" a Wildcard certificate would be able to validate the following domains:

First-level domain

- sample.com/

Second-level sub-domains

- support.sample.com
- login.sample.com
- contact.sample.com

But would NOT validate third-level sub-domains or below, such as:

- test.contact.sample.com

Wildcard certificates are generally as a cost-saving maneuver for websites with three or more sub-domains. However, with the recent emergence of extremely low-cost and free SSL providers, this is likely to diminish over time.

Best Practices

Selecting the best certificate type for your application depends on various legal and industry requirements and standards. For example, EV certificates are prevalent on medical and financial websites due to heightened privacy and security regulations governing those industries (HIPAA and SOX respectively). Additionally, the implementation of certain SSL levels is required to pass certain third-party auditing and attestation standards for eCommerce or service organizations, including PCI, SSAE 16, and SOC among others.

As a rule of thumb, DV certificates are sufficient for most small business websites and brochure-type websites. Medical, financial, governmental, and larger eCommerce sites should use only EV certificates. Wildcard certificates should generally be avoided because their application across multiple subdomains also means that they are less secure than other types. OV certificates may not provide enough enhanced value over DV certificates and therefore are not recommended.

If you are still uncertain about which type of SSL certificate is right for you, we are happy to provide a recommendation based upon your use-case.

Acquiring an SSL Certificate

Acquiring an SSL certificate is a relatively straight-forward process. You can acquire your certificate directly from a CA or from one of their authorized resellers. Major CAs include:

- Symantec
- GlobalSign
- VeriSign
- Comodo
- Network Solutions
- DigiCert
- Let's Encrypt

Often, hosting companies and other service providers act as resellers or distributors for these and other CAs.

Best Practice

Generally, the underlying technology, intrinsic value, and implementation of any SSL certificate is substantially the same no matter which CA issues it. Therefore, we see no inherent best practice of choosing one CA or reseller over another. The decision can be made based upon price, convenience, and/or existing business relationship.

Implementing SSL Website Security

The SSL certificate is little more than a simple text file installed on the server and associated directly with your domain. Experienced website engineers and server managers may install the SSL certificate manually on the web or email server. But in the vast majority of cases this is now an automated procedure that involves simply cut-and-pasting the certificate information into web interfaces within control panel or other web management tools (e.g., Plesk, cPanel). If you use one of the major control panels or server management tools, that company will likely have instructions on performing this task within their application. This is a task that can easily be performed by users with basic technical confidence.

This is such a common practice that most major full-service hosting providers have knowledge Base (KB) articles or tutorials specific to their architectures and infrastructures to help with this basic task. For those with little technical experience or confidence, these service providers may even perform the installation on your behalf as a part of their service or for a nominal fee.

SSL certificates can also be installed at the application layer inside many leading email server or content management system interfaces. Instructions for common applications can be found on most CA websites or in the user instructions for the application being secured via SSL. Again, if you are using a full-service hosting or infrastructure provider, they may provide this service on your behalf.

Certificate Installation is Not Enough

Acquiring and installing your SSL certificate is a relatively easy process. But simply applying an SSL certificate to a site does not ensure that the entire site is secure. Applying additional best practices is important because – without them – there can be negative impacts, including:

- Hard external URL links may no longer work properly, harming site traffic and SEO.
- Site visitors might see "mixed content" – both secure and unsecure.
- Users attempting to access sites from more secure and corporate networks may be unable to do so because of firewall security rules.
- SEO asset value within search engine indexing may be damaged or lost.
- If your home page indicates SSL security, but not all pages or page elements are secured, the mismatch can create corporate liability.

Best Practices

Before making any changes to your website, we strongly recommend that a full system backup be run and a roll-back procedure be clearly defined.

Test All Page Content

Once your SSL certificate is installed, it is important to review your website under SSL before making it official. With a certificate, all pages should show the "secure" indicator in the URL navigation bar (lock icon, green text, and/or https displayed). If a page does not display those indicators, it is possible that hard-coded content on that page references an unsecure (non-HTTPS) content item.

If these issues are not resolved, users trying to access you site from networks with strict security settings will simply not be able to do so. And since all modern, major browsers are sensitive to security issues, users accessing your site from less-stringent networks are likely to receive a warning message, such as:

> *"This page contains both secure and nonsecure items. Do you want to display the nonsecure items?"*

An additional positive action of acceptance will be required to proceed to your site, but the real harm is that confidence in your site and content will be diminished and some users will not accept the real or perceived risks of proceeding.

Some of the most common forms of missed unsecure content are a CSS, JS, or Image files. Fixing these items is very important as any file that isn't secure might not be served to the user, depending on user security settings.

Manual reviews are more than sufficient for smaller sites. Larger organizations and companies with extensive websites employ automated website scanners to identify all such issues. There are many such tools available. IowaComputerGurus employs several such tools when we are reviewing website security. Some of these are specialty tools designed for specific industries or that emphasize specific kinds of results. However, the two primary tools we use and recommend for standard business website scanning are SemRush and SiteImprove.

If you require more advanced guidance in choosing a security scanning solution, we are happy to help with a recommendation tailored to your use-case.

> *It is important that all site content be fully tested and verified under SSL before proceeding to any other best practice. Failing to do so may result is lost content or the inability to access content, potentially resulting in downtime.*

Force All Pages to Use SSL

To ensure that all the pages and content on your website are using SSL security, you can place a block of code within the File Manager that forces (rewrites) URLs accordingly. If you are using a major Content Management System (CMS) such as WordPress, DNN, Drupal, Joomla, etc., there is either a setting in the website configuration that will perform this task for you or a plugin you can install to force SSL. Consult that application's help files for specific instructions.

If your CMS does not natively support this function or if you are managing a custom or hand-coded website, you will need to manually add a block of code either the beginning of either the .htaccess file (PHP, NginX, and other non-Windows web servers) or added as a rewrite rule within the applications' web.config file (for Windows-based IIS web servers).

> *We strongly recommend that only experienced website administrators attempt to manually edit .htaccess and IIS web.config files.*

The code samples below force all pages to use SSL and set 301 redirects to help ensure that website visitors arriving at a site via hard links will be able to access the content.

Sample Force SSL code for most non-Windows web servers:

```
RewriteEngine On
Rewrite %{SERVER_PORT} 80
Rewrite ^(.*) https://sample.com/$1 [R=301,L]
```

Sample Force SSL code for Windows IIS web servers:

```
<rewrite>
<rules>
<rule name="HTTP/S to HTTPS Redirect" enabled="true" stopProcessing="true">
<match url="(.*)" />
<conditions logicalGrouping="MatchAny">
<add input="{SERVER_PORT_SECURE}" pattern="^0$" />
</conditions>
<action type="Redirect" url="https://{HTTP_HOST}{REQUEST_URI}"
redirectType="Permanent" />
</rule>
</rules>
</rewrite>
```

Set 301 Redirects for All Existing Unsecure Pages

If you have forced all pages to use SSL by employing the methods defined above, then your 301 redirects are likely already in place and no further action is necessary. If you are attempting to set 301 redirects for your SSL implementation without forcing all pages to use SSL (not recommended), then this section is provided for your reference.

Coded redirects are used to send website visitors to a URL that is different from the one that they typed or that was embedded in a link that they clicked. When you install your SSL certificate, your URLs will change from "http:sample.com/" to "https:sample.com/."

Eventually, the major search engines will crawl your site completely and update their indexing. But all the existing SEO value will be lost, and hard-coded cross-links and refers to your pages may not work as expected.

The solution is to apply 301 redirects. Generally speaking, there are eight kinds of 3xx-level redirect status codes, five of which are common. 301 redirects are the only kind you are likely to ever need (there are some rare exceptions). 301 redirects are called "permanent" redirects because they send a signal to the search engines and browsers alike telling them that the associated content and resources have been permanently moved to a new location. The benefits include:

- Site visitors entering the older version of the URL will be seamlessly sent to the new, secure page.
- All or nearly all the existing SEO value will be transferred to the new page.

Implement Strict Transport Security – HSTS

Strict Transport Security (HSTS) makes a website more secure by informing browsers that the website will only use the highest form of security and encryption indicated on the site. This prevents certain types of intrusions and attacks (e.g., protocol downgrade, man-in-the-middle, cookie hijack).

From a technical standpoint, implementing HSTS is relatively easy. It requires the addition of a single, short line of text within the website page headers.[9] The header text includes options for:

- Duration – expressed in seconds as "max-age=" and tells the browser how long to "remember" the instruction for that particular domain. In the examples below, 604800 seconds equals one week.
- IncludeSubDomains – if included, all subdomains will follow the same instruction.
- Preload – if included, Google will retain a record of the instruction in it HSTS Preload Service instructing all browsers to only attempt encrypted connections (page load speed improvement).

Sample IIS HSTS header code:

```
Strict-Transport-Security: max-age=604800; IncludeSubDomains; preload
```

Sample Apache HSTS header code:

```
Header always set Strict-Transport-Security "max-age=604800; IncludeSubDomains; preload"
```

Sample NginX HSTS header code:

```
add_header Strict-Transport-Security "max-age=604800; IncludeSubDomains; preload" always;
```

[9] For more information and an overview of HSTS, visit: https://www.owasp.org/index.php/HTTP_Strict_Transport_Security_Cheat_Sheet

If you need assistance with the implementation of HSTS on your site or determining which options are best for your environment, we are happy to make additional recommendations.

Use Google Fetch via Search Console

When you transition from an unsecure website (http:) to a website secured by SSL (https:), Google treat this as a migration to a new site, even if the physical location of the page and content has not changed. Go to Google's Search Console and add the new, secure URL as a "new property" and use the tools to ensure that all pages can be properly indexed.

This process will run a report that lets you know if Google has any difficulty in access your pages and will email you is a problem develops.

Additional Basic Website Security Best Practices

Implement X-Content-Type-Options

In IP technology, MIME types are indicators that tell browsers and servers what kinds of files are being transferred. For most purposes, MIME types are identical to file extensions – the characters that appear to the right of the "dot" in file names. Common examples include Adobe Portable Document Format (.pdf), video files (.mp4), audio files (.mpeg), various image files (.gif, .jpeg/.jpg, .png, etc.), and so on.

"X-Content-Type-Options" – specifically "nosniff" – is a simple implementation of header response instructions that can prevent certain types of MIME sniffing information leaks.[10] The primary function is to ensure that the MIME type agrees with the purpose and function of the file. For example, if content is identified as a ".png" file but the actual content is not an image, then that content would be blocked.

The X-Content-Type-Options instruction was first implemented in Microsoft IE 8, and it is now broadly accepted in all major browsers.

Sample IIS X-Content-Type-Options header code:

```
X-Content-Type-Options: nosniff
```

Sample Apache X-Content-Type-Options header code:

```
Header always set X-Content-Type-Options "nosniff"
```

[10] For more information and an overview of X-Content-Type-Options, visit: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Content-Type-Options

Sample NginX X-Content-Type-Options header code:

```
add_header X-Content-Type-Options “nosniff” always;
```

If you need assistance with the implementation of X-Content-Type-Options on your site, we are happy to make additional recommendations.

Implement X-XSS-Protection

“X-XXS-Protection” is a method a simple implementation of header response instructions recognized by all major browsers to prevent a common vector – cross-site scripting attacks.[11]

Even though the most current networks and browsers employ strong content security policies that obviate the need for X-XSS-Protection, there are literally billions of browser and network installations that do not employ the latest patches and upgrades, so we recommend that this extra precaution be implemented.

Sample IIS X-XSS-Protection header code:

```
X-XSS-Protection: 1; mode=block
```

Sample Apache X-XSS-Protection header code:

```
Header always set X-XSS-Protection “1; mode=block “
```

Sample NginX X-XSS-Protection header code:

```
add_header X-XSS-Protection “1; mode=block” always;
```

There are some significant differences in code syntax when implement X-XSS-Protection across different platforms. If you need assistance with the implementation of X-XSS-Protection on your site, we are happy to make additional recommendations.

Implement X-Frame-Options

Website security is not only about protecting your site, it’s also about protecting your website visitors. Implementing X-Frame-Options is designed to do just that.

[11] For more information and an overview of X-XSS-Protection, visit: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-XSS-Protection

There is a malicious technique called "clickjacking" that allows one set of images, content, and links to "hide in plain sight" by appearing as a transparent iframe over your content. In this case, what you see in not what you get as users might be "seeing" your website but interacting with another.

X-Frame-Options was created by Microsoft in 2009, but many sites still do not protect themselves using this simple header instruction. It works by specifying the websites that are allowed to place iframes on your website – or none at all. Code values include:

- DENY – no iframes will be allowed on your website.
- SAMEORIGIN – if you want to use iframes, this value means that only iframes originating from your domain will be allowed.
- ALLOW-FROM – if you want to allow iframes from specific external sites the ALLOW-FROM value lets you identify the domains that you trust to do so. In this case, the ALLOW-FROM value would be followed by the trusted domain's URL.

Sample IIS X-Frame-Option header code:

```
X-XSS-Protection: SAMEORIGIN
```

Sample Apache X-Frame-Option header code:

```
Header always set X-Frame-Options "SAMEORIGIN"
```

Sample NginX X-Frame-Option header code:

```
add_header X-Frame-Options "SAMEORIGIN" always;
```

Summary

While the security best practices provided in this document are not all-inclusive, they form the core of what we consider to be a relatively-easy to implement website security policies that protect against most common attacks. Perhaps surprisingly, most active websites do not employ even the most basic of these techniques.

Website security will continue to be a moving target as technology evolves and malicious attacks continue to evolve with it. IowaComputerGurus is committed to remaining at the forefront of complete website security to protect our customers and the larger internet community.

If you want to ensure that you are viewing the latest version of this document or want advice and recommendations specific to your environment and use case, please let us know. We are always happy to help.

About IowaComputerGurus Inc.

IowaComputerGurus Inc., a Microsoft Certified Partner organization specializes in developing custom solutions using the Microsoft .NET development stack and often using the DotNetNuke application framework for web application development. Based in Des Moines, Iowa, they provide services to customers all over the world and base their business on providing quality, affordable technology solutions with the best customer service. The company is led by Mitchel Sellers, a Microsoft MVP, AP.Net Insider, Microsoft Certified Professional, DNN MVP, and published author.

Contacting IowaComputerGurus

IowaComputerGurus, Inc.
5550 Wild Rose Lane, Suite 400
West Des Moines, Iowa 50266

Phone: (515) 270-7063
Fax: (515) 866-591-3679
Email: webmaster@iowacomputergurus.com
Website: http://www.iowacomputergurus.com

Feedback

IowaComputerGurus is committed to quality and appreciates constructive feedback on our Best Practices documents, white papers, and other technical guides. If you discover any errors or have suggestions of additional items for inclusion or any modifications to existing content, please use one of the above listed contact methods to let us know.

Disclaimer

This document reflects the opinions and experience of the authors as a non-compensated service to the community. It is provided "as-is," and no warrantee is expressed or implied as to the serviceability or applicability of any information or recommendation for any particular purpose, application, or environment. It is the reader's responsibility to conduct their own research, consult experts, and determine whether this information is right for their website, application, and / or environment. Additionally, the reader understands use of this documentation constitutes agreement to the current terms of use as published on the IowaComputerGurus.com website, which are subject to change.

Copyright and Trademark Notices

to anyone in its entirety, however, it must remain intact and unchanged with all copyright notices and disclaimers clearly visible.

"DNN" is a registered trademark of DNN Software Corporation. "WordPress" is a trademark of Automatic, Inc. Drupal is a trademark of Dries Buytaert. Joomla is a trademark of Open Source Matters. Microsoft, Windows, ASP.NET, SQL, SQL Server, and other related product names are the trademarks of Microsoft Corporation.

All other trademarks remain the property of their respective owners.