常時SSL/TLS化の実践マニュアル

現場ですぐに役立つ“ハウツー”を集約

digicert® | Symantec™ Website Security

目次

3 第1章 常時SSL/TLS化の基礎知識
3 急増するHTTPSトラフィック
3 常時SSL/TLS化する6つの理由
4 常時SSL/TLS化の進め方
5 第2章 常時SSL/TLS化の実装
5 常時SSL/TLS化実装の流れ
5 改修前の準備
6 コンテンツ、サーバーの改修
7 HTTPSによるアクセスの確認
7 関連システムの更新
7 HTTPSサイトへの一本化（強制転送）
8 HTTPSサイトへの一本化の確認
8 常時SSL/TLS化の強化
9 第3章 FAQ & Tips
12 第4章 SSL/TLSに関する有用なツールとガイドライン

第1章 常時SSL/TLS化の基礎知識

Webサイトを構成するすべてのページをHTTPS化する、いわゆる“常時SSL/TLS化”の波がインターネットの世界に押し寄せている。すでに金融機関やECサイトはもちろん、SNS、検索エンジン、ポータルサイトなどの主要なWebサイトのほとんどが実施済みだ。

急増するHTTPSトラフィック

SSL(Secure Sockets Layer)/TLS(Transport Layer Security)は、インターネット上の通信を暗号化するプロトコルである。サーバーとクライアント間の通信データを暗号化して、悪意のある第三者によるデータの盗聴・改ざん防止を目的に、20年以上にわたって利用されてきた。

SSL/TLSはその安全性の高さから、ID/パスワードを含む個人情報やクレジットカード番号といった決済情報をインターネット経由で安全に通信するために使われている。当初は機密情報を入力するためのフォーム、あるいはフォームを含むページのみをSSL/TLSによって暗号化(HTTPS化)することが多かった。ところが、2013年ごろから事情が大きく変わりつつある。Webサイトを構成する全ページをHTTPS化する常時SSL/TLS化が広まっているのだ。

Googleのクロール結果を基にした統計情報を公開しているHTTP Archiveのデータによると、2011年1月時点でSSL/TLSを利用するHTTPSのトラフィックは全体の2%程度にすぎなかったが、約6年後の2016年12月には36%を占めるまでに増えている。

常時SSL/TLS化する6つの理由

Webサイトの常時SSL/TLS化が進んでいるのには、以下のような明確な理由がある。

①Webサイトのセキュリティ強化

SSL/TLSの適用範囲が、個人情報や決済情報などの機密情報を扱うフォームだけに限定されている場合、その他の通信は暗号化されていないことになる。常時SSL/TLS化によってWebサイト全体の通信を暗号化すれば、通信経路上でのコンテンツ改ざんが防げるようになるほか、HTTPでは無防備なCookieの通信も暗号化されるなど、セキュリティ強化の面で大きなメリットがある。

②Webサイト真正性の証明

インターネットバンキングやショッピングなどのWebサイトを本物そっくりに作り、ログイン情報や決済情報を窃取しようとするフィッシングサイトは相変わらず多い。常時SSL/TLS化を行えば、第三者の認証局(CA=Certification Authority)が発行するEV (Extended Validation) 証明書やOV (Organization Validation) 証明書によって、正規のWebサイトであるという真正性をユーザーに証明することができる。

③検索結果表示順位の向上

常時SSL/TLS化には、SEO対策上の効果が期待できるというメリットもある。例えばGoogleは、2012年3月に自社の検索サイトを常時SSL/TLS化したのち、2014年8月に常時SSL/TLS化したWebサイトを優遇する検索アルゴリズムを実装した。それを機に、一般企業や官公庁で常時SSL化に取り組む動きが加速した。

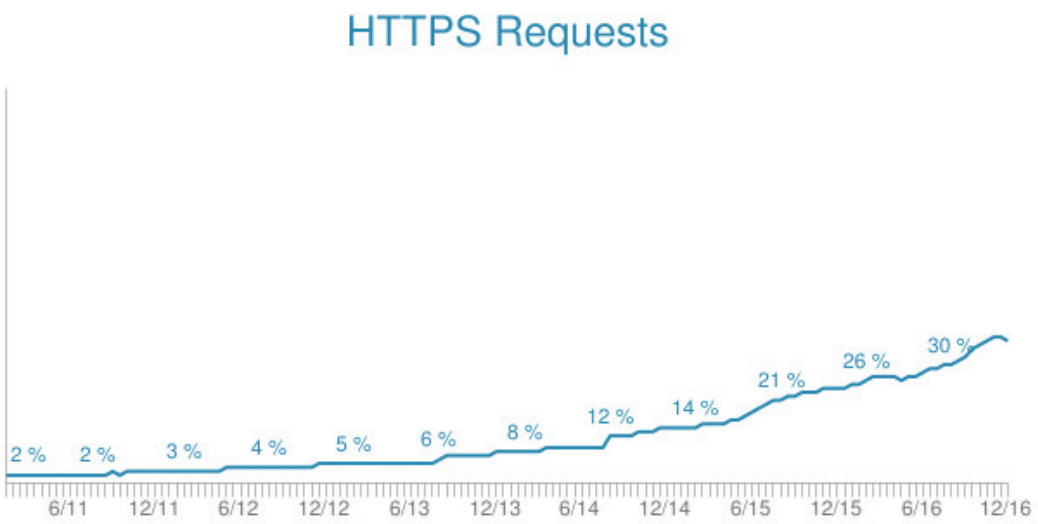


図:HTTP Archiveの「HTTPS Requests」の推移/2016年12月時点

④アクセス解析精度の確保

常時SSL/TLS化は、アクセス解析の精度を確保するためにも必要だ。なぜならHTTPSのWebサイトからHTTPのWebサイトに遷移した場合、リファラ（アクセスログに記録されるリンク元の情報）が送信されないからである。そのため、HTTPSのWebサイトが増加すればするほど、HTTPのWebサイトはアクセス解析の精度を維持するのが難しくなり、常時SSL/TLS化の必要性が高まることになる。

⑤Webサイト高速化の可能性

常時SSL/TLS化の技術的なメリットとして、Webサイトのページ表示が高速化できるという可能性が指摘されている。その鍵を握るのが、2015年2月にIETFが仕様を承認した「HTTP/2」である。Webサイトのレイテンシー改善等に有効とされるHTTP/2では、暗号化なしが定義されているものの、HTTP/2に対応する主要Webブラウザはすべて TLSによる通信を要求する。そのため、WebサイトをHTTP/2対応にした場合、必然的に常時SSL/TLS化が求められる。

⑥開発・運用管理の効率化

Webサイトを部分的にSSL/TLS化した場合、Cookieの受け渡しをHTTPSかHTTPかによって条件分岐する必要があるなど、開発に手間がかかる。それに対し、常時SSL/TLS化を行った場合、全通信が暗号化されるため、Cookieの受け渡しを変えたり、条件分岐を行ったりする必要がなくなる。また、ページ間のリンクのパス指定も相対パスのみに統一することができる。これにより、Webサイトの開発・運用管理がシンプルになり、効率化できる。

常時SSL/TLS化の進め方

常時SSL/TLS化は、以下の段取りで社内調整を進めていく。

①対応Webサイトの決定

最終的には現在稼働中のWebサイト全体を常時SSL/TLS化することになるが、いきなり全ページ対応というのはハードルが高い。最初は、ページ数が少なくビジネスへの影響が小さいサブドメインなど、対応Webサイトを決めてから部分的にHTTPS化しておくことをお勧めする。部分的な稼働テストを実施しながらHTTPS化を進めることが、結果的に常時SSL/TLS化の円滑な導入につながる。

②初期導入費用の試算

すでにSSLサーバ証明書が組み込まれているWebサイトの場合、基本的に初期導入費用はかからない。しかし、多くの企業・組織が利用するレンタルサーバーでは、HTTPSに対応させるのに証明書の取得やインストールに費用がかかるケースがある。BtoC向けのWebアプリケーションを提供しているような大規模サイトの場合、HTTPSに対応したCDN（Contents Delivery Network）の利用も選択肢の一つだ。

③ 証明書の選定

常時SSL/TLS化では、SSLサーバ証明書の選定が最も重要なポイントとなる。一般向けの公開サイトであれば、サイトのHTTPS対応がWebブラウザで視覚的に確認できるEV（Extended Validation） 証明書の導入が望ましい。例えば、シマンテックのEV証明書ならば、シールの認知度が高くコンバージョンレートの向上も期待できる。イントラネット上のサイトはDV（Domain Validated）証明書でも構わないが、DV証明書はセキュリティレベルが低いので注意が必要だ。

④ スケジュールの作成

常時SSL/TLS化を実践するには、サイトのビジネスオーナー、マーケティング部門のSEO担当者、開発部門のコーディング担当者、IT部門のサーバー担当者など多くの関係者がプロジェクトに参加することになる。まずは常時SSL/TLS化の目的を明確にして意思統一を図り、対応サイトの決定、証明書の取得、サーバーの設定、ソースの書き換え、動作確認テストの実行などのスケジュールを作成・共有する。

⑤ プロジェクトの実行

常時SSL/TLS化のスケジュールが確定したら、対応Webサイトへの導入～テストを実行する。この部分的な適用をフェーズ1とし、動作確認テストを入念に実施する。このとき、外部のWebビーコン（識別・追跡ツール）の構成には注意を払い、必要に応じてHTTPS用ビーコンを入手する。フェーズ1で問題がないことを確認できたら、フェーズ2～3と徐々に適用範囲を広げ、最終的にWebサイト全体を常時SSL/TLS化する。

第2章 常時SSL/TLS化の実装

常時SSL/TLS化は、Webサイトのセキュリティ強化や真正性の証明だけでなく、SEO対策や開発・運用管理の効率化といった観点からも、今後避けては通れない道と言える。そこで本章では、Webサイトの常時SSL/TLS化を進める手順について具体的に見ていく。

常時SSL/TLS化実装の流れ

HTTPからHTTPSに切り替える場合、以下の順番で進めていく。下記の各手順にある[新規]、[既存]、[対象]、「◎」「○」「△」「−」は、それぞれ以下を表している。

[新規:] = 新規サイトを常時SSL/TLS化で立ち上げる場合
[既存:] = 既存サイトを常時SSL/TLS化する場合
◎ = 「必ず実施」
○ = 「出来る限り実施」
△ = 「実施が望ましい」
− = 「実施不要」
[対象:] = SSL/TLS化の実施対象

改修前の準備

(1) SSL/TLS対応サーバーおよびCDN(Contents Delivery Network)の情報収集

[新規:◎] [既存:◎] [対象:すべて]

常時SSL/TLS化に乗り出す前に、まずは、その意義を理解しておく。HTTPS対応のWebサーバーに付属するドキュメント(図1)や、CDN(使用している場合)のドキュメントなど、常時SSL/TLS化に関するあらゆる資料に目を通し、対象サイトを定める。

Webサーバーについて不明な点があれば、ホスティングサービス提供事業者に確認するのもよい。

常時SSL/TLS化の意義については、シマンテックのWebページ(第4章「⑦」を参照)やセミナーなどの情報源を参照する。なかでも、SSLサーバ証明書の選定は重要なポイントになるので、以下のWebページを参考に証明書を選定する。

SSLサーバ証明書に関するシマンテックのWebページ

URL: https://www.symantec.com/ja/jp/tls-ssl-certificates/

(2) 現在のWebサイトをクロール、バックアップ

[新規:−] [既存:◎] [対象:すべて]

Webサイトの現在の状態を知り、修正個所を洗い出すためにWebページをクロールする。ページのすべてのリンク、すべてのダウンロードコンテンツを漏れなく巡回し、現在のWebサイトの状況を把握する。また、万一の場合のコンテンツ切り戻しに備え、すべてのコンテンツのバックアップをとっておく。

(3) SSL/TLS対応テストサーバーの用意

[新規:◎] [既存:◎] [対象:すべて]

HTTPSに対応したWebサーバーをセットアップする。従来のWebサイトを常時SSL/TLS化する場合には、テストサーバーに証明書をインストールする。テストサーバー

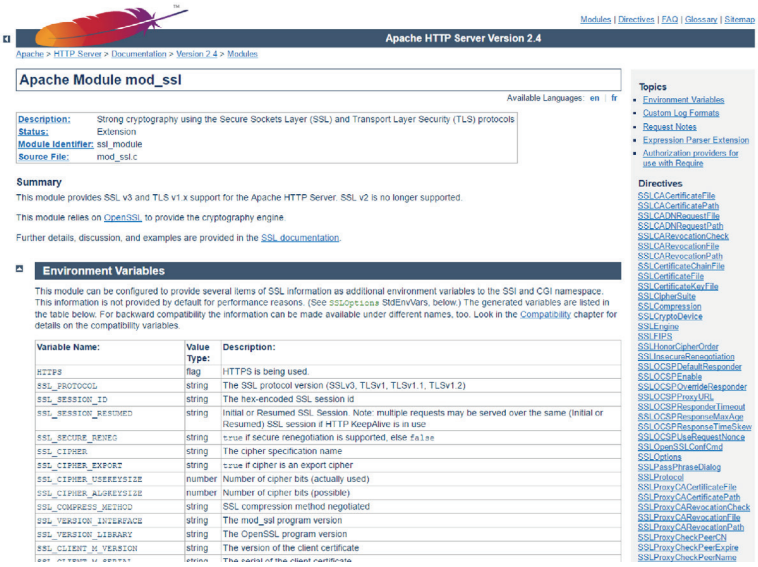


図1:Webサーバーのドキュメントの例(Apache httpd mod_sslモジュールのドキュメントページ)

により事前に動作検証を実施して計画的に移行することがベストプラクティスとなる。

(4) 常時SSL/TLS化前のアクセス状況・検索順位の計測

新規:ー　既存:○　対象:すべて

常時SSL/TLS化後との比較のために、常時SSL/TLS化前のアクセス状況や重要キーワードの検索順位を計測しておく。

コンテンツ、サーバーの改修

(5) 証明書の取得と本番サーバーへのインストール

新規:◎　既存:◎　対象:すべて

まずCSR(Certificate Signing Request=署名要求)を作成し、信頼できるCA(Certification Authority=認証局)からSSLサーバ証明書を取得する。取得した証明書は、Webサーバーにインストールして設置する(図2)。このプロセスは、それぞれのホスティング環境やサーバー設定によって異なるが、証明書を提供するCAが用意する手順書に従えば、容易に行える。

```
SSLEngine on
SSLProtocol all -SSLv2 -SSLv3
SSLCipherSuite ALL:!ADH:!EXPORT:!SSLv2:RC4+RSA:+HIGH:+
MEDIUM:+LOW
SSLCertificateFile /etc/httpd/conf/ssl/server.crt
SSLCertificateKeyFile /etc/httpd/conf/ssl/server.key
```

図2:ssl.confファイルに証明書のパスを追加した例

(6) サーバーの暗号アルゴリズムを調整

新規:○　既存:○　対象:すべて

SSL/TLS化に利用する暗号アルゴリズムは通常、Webサーバーのデフォルト設定で問題はない。ただ、Webサーバーが優先するアルゴリズムが古くなっている、あるいは、セキュリティをより高めたい場合には、Webサイトに合わせてアルゴリズムの優先順位を調整する。また、ECC(Elliptic Curve Cryptography)に対応したSSLサーバ証明書は、RSA版に比べてCPUに対する負荷が少なく、その分、レスポンス向上が期待できる。なお、ベストプラクティスとされている暗号アルゴリズムについては、IPA、CRYPTEC、Mozillaなどのガイドラインを参照するとよい。

(7) HTMLコードの書き換え

新規:ー　既存:◎　対象:すべて

Webページ内に読み込んでいる画像、CSS、JSファイルなどに「http://」から始まる絶対パスが含まれている場合、ブラウザ側で「安全でない」とエラーが表示される。したがって、「http://」の絶対パスは、「https://」から始まる絶対パスに書き換えるか、相対パス「../」、ルートパス「/」に変更する必要がある。ただ、エラーを避けるためには絶対パスと相対パスを混在させないことが望ましい。また、第三者から提供されているタグやビーコンなどが「https://」による読み込みに対応していない場合は、提供元に問い合わせて対応したものを入手する。

(8) テンプレート内のHTMLコードの書き換え

新規:ー　既存:◎　対象:CMS 利用サイト

CMS、Webアプリケーションのテンプレート内に記載されているHTMLコードの画像、CSS、JSファイルなども忘れずにアップデートする。デプロイ方法によっては、テンプレートのソースをテキストエディタなどに読み込み、検索・置換をかけることでアップデートが行える。こうしてスクリプト、イメージ、リンクなどへの参照を「https://」から始まる絶対パスに書き換えるか、相対パス「../」、ルートパス「/」に変更すればよい。

(9) CookieにSecure属性を追加

新規:◎　既存:◎　対象:Cookie 利用サイト

Cookieを利用するWebサイトでは、Cookieにsecure属性の設定を追加する。これにより、意図しないHTTPアクセスによってcookie情報が漏えいしてしまうリスクが回避できる。

(10) robots.txtファイルの更新

新規:ー　既存:○　対象:robots.txt利用サイト

robots.txt内に記載されているURLをHTTPSのパスに変更する(図3)。また、HTTPSアクセスをブロックする古い記載が残っている場合は、それも削除する。

```
User-agent: *
Allow: /
Sitemap: https://example.com/sitemap.xml
```

図3:robots.txtファイルの記述例

(11) XMLサイトマップの更新

新規:ー　既存:○　対象:XMLサイトマップ利用サイト

XMLサイトマップ内に記載されているURLをHTTPSのパスに変更する。

(12) 正規化(canonical)タグの更新

新規:ー　既存:○　対象:canonical タグ利用サイト

canonicalタグは、同じページを表す複数のURLの中で、どれが正規のURLなのかを宣言して重複を避けるために使用される。このリンク指定をHTTPSに変更する(図4)。

```
<link rel="canonical" href="https://www.example.com/">
```

図4:canonicalタグの例

(13) hreflangタグの更新

新規:ー　既存:○　対象:hreflang タグ利用サイト

hreflangタグは、日本語と英語などの多言語サイトを運用している場合に、検索したユーザーに対して適切な言語のバージョンを表示するために使用される。このリンク指定をHTTPSに変更する。

```
<link rel="alternate" hreflang="ja" href="https://ja.example.com/">
<link rel="alternate" hreflang="en" href="https://en.example.com/">
```

図5:hreflangタグの例

HTTPSによるアクセスの確認

(14) 更新内容の確認

新規:◎　既存:◎　対象:すべて

ここまでに変更した内容を確認するために、Webサイトをクロールしてリンクが更新されていること、壊れていないことをチェックする。Screaming Frog SEO SpiderのようなWebサイト分析ツールを使えば、HTTPSへの変更が漏れているコンテンツを発見することができる。また、任意の外部スクリプトがHTTPSをサポートしているかも確認する。

関連システムの更新

(15) CMSの設定を変更

新規:ー　既存:◎　対象:CMS 利用サイト

CMSを利用している場合、HTTPS化するにあたって設定を変更する必要があるかもしれない。主なCMSシステムでは、HTTPS移行ガイドにその内容が記載されている。

(16) アクセス分析ツールの更新

新規:ー　既存:○　対象:アクセス分析ツール利用サイト

アクセス分析ツールを使って追跡している場合、Webサイトの設定URLをHTTPからHTTPSに変更する。

(17) リンク先URLの更新

新規:ー　既存:○　対象:オンライン広告出稿サイト

バナー広告、リスティング広告、メール広告など、出稿中の広告のリンク先URLをHTTPからHTTPSに変更する。

(18) その他のツールの更新

新規:ー　既存:○　対象:その他のツール利用サイト

その他、レコメンド、サイト内検索、A/Bテスト、最新記事埋め込みツールなど、ページに外部タグを埋め込んでいる場合、必要に応じてツールに設定済みのURLをHTTPからHTTPSに更新する。

(19) 検索エンジンへの登録

新規:○　既存:△　対象:検索エンジンへの反映を急ぐ場合

検索エンジンは通常、申請しなくてもクローラによってWebサイトがHTTPSに切り替わったことを自動認識する。ただ、念のために検索エンジンにHTTPSの新しいURLを申請しておいてもよい。

HTTPSサイトへの一本化 (強制転送)

(20) 301転送でHTTPSを強制化

新規:◎　既存:◎　対象:すべて

この手順は、常時SSL/TLS化を実施するプロセスの中で最も重要なポイントだ。.htaccessファイルやサーバーの設定を書き加えることで、HTTPのURLからHTTPSのURLにリダイレクト(転送)させる(図6)。

望ましいのはHTTPのURL(旧URL)とHTTPSのURL(新URL)を1対1で関連づけて転送する設定だが、ページが多い場合はすべてのアクセスについてHTTPを単純にHTTPSに読み替えるよう設定してもよい。

転送については、検索エンジンが「一時的な切り替えではなく、恒久的な切り替えである」と認識できるように301転送(恒久的転送)の設定を行う。

ここで古い転送設定が残っている場合は、見直しが必要になる。とりわけ、古いURLから何度も転送を繰り返す設定や、HTTPSへのアクセスをHTTPに戻す設定が残っていると、ページの表示速度が遅くなったり、HTTPSでアクセスできなかったりと、意図しない挙動になるので注意が必要だ。

```
RewriteEngine On
RewriteCond %{HTTPS} off
RewriteRule ^(.*)$ https://%{HTTP_HOST}%{REQUEST_URI}
[R,L]
```

図6:.htaccessにおける301転送の設定例

(21) HTTPS化の告知

新規:ー | 既存:○ | 対象:すべて

URLをHTTPからHTTPSに切り替えた後には、名刺、広告物、リンク元などにHTTPSから始まる新しいURLを案内する。すぐにHTTPのアクセスがなくなることはないが、新しいHTTPSのURLが広まるにつれてHTTPによるアクセスが減少、代わりにHTTPSによるアクセスが増加することになる。

HTTPSサイトへの一本化の確認

(22) 転送の再確認

新規:ー | 既存:◎ | 対象:すべて

もう一度Webサイト全体をクローリングし、アクセスが正しく行われているかチェックする。また、ブラウザに用意されているデベロッパーモード機能を活用し、ページ内にHTTPの混在コンテンツが残っていないかどうかチェックすることもお勧めする。

常時SSL/TLS化の強化

(23) HSTSの有効化

新規:○ | 既存:○ | 対象:すべて

HSTS(HTTP Strict Transport Security)を有効にして、ブラウザに常にHTTPSを使用するよう指示する(図7)。これにより、ユーザーはHTTPリンクをクリックしても常にHTTPSで接続することになるため、ページ表示速度の向上やMITM攻撃のリスクが軽減できる。なおHTTPSへの切り替え当初は、問題発生の事態に備え、HTTPへの切り戻しも念頭に入れ、max-ageを1日に設定しておき、しばらくしてから1年程度に伸ばすのが安心だ。

```
Header set Strict-Transport-Security "max-age=31536000"
```

図7:.htaccessによるHSTSの有効化例

(24) OCSP Staplingの有効化

新規:△ | 既存:△ | 対象:すべて

OCSP Staplingを有効にすることで、サーバーがブラウザの代わりに証明書の有効／失効を確認するようになる(図8)。これにより、ブラウザがWebサイトにアクセスするたびに証明書の有効性を確認するというプロセスが不要になる。結果、ユーザーにとってはWebサイトの表示速度の向上につながる。

```
SSLUseStapling on
SSLStaplingResponderTimeout 5
SSLStaplingReturnResponderErrors off
```

図8:Apache httpd でのOCSP Staplingの有効化例

(25) HTTP/2への対応

新規:△ | 既存:△ | 対象:HTTP/2に対応させる場合

Apache、NginxなどのWebサーバーをアップグレードし、Forward Secrecyの設定やTLS 1.2の利用といった要件を満せるようにしておけば、HTTP/2によるレイテンシーの改善やプッシュ機能など、新技術の恩恵が享受できるようになる。

第3章 FAQ & Tips

常時SSL/TLS化を進めるに当たり、企業・組織のWebサイト運営者が知っておくべきよくある質問と回答をまとめた。

Q1

HTMLコードにHTTPとHTTPSの混在があると何が問題になるのか?

A1

HTTPSのページにあるHTMLコードの中に「http://」という絶対パスの画像データやCSS、JSファイルなどが混在していると、ブラウザに警告が出たり、ファイルが表示されなかったりする。これは、Cookieなどの盗聴・改ざんを防ぐためのブラウザの仕様だ。

このような事態を回避するためには、すべてのパスが「../」の相対パス、または「/」のルートパスに書き換わっているかどうかを入念にチェックする必要がある。すべてのパスを「https://」から始まる絶対パスに変更することでも対応可能だが、相対パスにしておくほうが、のちの管理が容易になることが多い。また、CSSやJSファイル内の「http://」表記は、対応漏れが起きやすいので細心の注意が必要である。

```
src="../xxxx.html"
src="/xxxx.html"
src="//www.example.com/xxxx.html"
```

混在エラーを回避するHTMLの記述例

Q2

外部ベンダーから提供されているビーコンタグはHTTPS対応か?

A2

動画ファイル、SNSボタン、アフィリエイトタグ、アクセス解析タグ、JSファイルなどのビーコンタグを通じて、外部ベンダーから提供されているビーコンは、必ずHTTPS対応のものを利用する。HTTPS対応のビーコンタグが設置できていないと、外部サイトの動画やアフィリエイト広告などが正しく表示されないといった不具合が起きる。このような場合、外部ベンダーにHTTPS対応のビーコンの提供を依頼する必要がある。

```
<script src="https://www.google.com/jsapi" type="text/javascript"></script>
```

HTTPS対応ビーコンタグ(JSファイル)の設置例

Q3

暗号化/復号化によってページの表示速度は落ちないか?

A3

原理上、SSL/TLSでは暗号化/復号化処理の際にネゴシエーションの負荷がかかるため、ページの表示速度が低下すると考えられている。しかし今日では、サーバやデバイスのCPU性能が向上し、かつ、一度使用したコネクションを継続して利用する「Keep-Alive」設定を「on」にし、ネゴシエーション回数を最小化することが一般的になっている。そのため、SSL/TLSの暗号化・復号化処理による性能低下はほとんど気にならない。また、サーバをHTTP/2対応版へとアップグレードすることで、レイテンシーが改善され、むしろHTTPよりもHTTPSのほうが、ページ表示速度が速くなる可能性もある。

Q4

証明書の有効期限とはなんでしょうか?

A4

証明内容が保証されている期間のこと。SSLサーバ証明書は、実世界の証明書と同様に認証内容に有効期限が設定されている。そのため、定期的(通常では1年おき)に認証を受けて、証明書を更新しなければならない。主要なブラウザでは、デベロッパーモードで有効期間が確認できる。また、有効期限が切れるとアクセスした際にブラウザが警告を表示させる。

Q5

ブラウザが必ずHTTPSへアクセスするようにできないか?

A5

Webサーバ管理者がHSTS(HTTP Strict Transport

Security)を設定することで、訪問者のブラウザにWebサイトがHTTPS対応であることを覚えさせることができる。HSTS対応ブラウザの利用者は、次回の接続からHTTPではなくHTTPSで接続することになる。

Q6

HTTPSの負荷を軽減できないか?

A6

一度使用したコネクションを継続して利用する「Keep-Alive」の設定を「on」にすることで、HTTPSの負荷が軽減できる。また、RSA方式に比べて暗号効率の高いECC(Elliptic Curve Cryptography)に対応したSSLサーバ証明書を利用すれば、CPUへの負荷をさらに低減することが可能だ。

Q7

Cookieの漏えいリスクを回避するには?

A7

Webサーバ管理者がCookieのsecure属性を有効にすることで、CookieはHTTPSによる暗号接続時にのみ通信経路上を流れるようになる。これにより、セッション情報や個人を特定する情報を含むCookieが通信経路上で漏えいしてしまうリスクが回避できる。その意味でも、Cookieを利用するWebサイトはHTTPではなく、HTTPSで構築することが望ましい。

Q8

常時SSL/TLS化したWebサイトへのHTTP接続を回避するには?

A8

常時SSL/TLS化したはずなのに、HTTPによる訪問が発生している場合、以下の原因が考えられる。これらをなくすことでHTTP接続を回避することができる。

- リンク元がHTTPのままになっている
- HTTPからHTTPSへ301転送(恒久的な転送)をしていない
- .htaccessファイルにHTTPSをHTTPに転送する設定が残っている
- HTMLのURLをHTTPS、または相対パスに書き換えていない

なお、Web サーバ側でHSTS 設定を行うことで、Web ブラウザに次回以降はHTTPS で接続するよう強制化することが可能である。

Q9

HTTP接続のリクエストをHTTPS接続に転送するには?

A9

HTTP接続のリクエストをHTTPS接続に転送するには、Webサーバに設置する.htaccessファイルを使って301転送を設定する(設定例は第2章(20)を参照)。

Q10

検索エンジンにHTTPSサイトであると認識させるには?

A10

Googleなどの検索エンジンにHTTPSサイトであると認識させるには、.htaccessでSSL/TLS接続をブロックしないこと、robots.txtやメタタグのnoindexなどで検索エンジンのクロールを排除しないことが必要になる。

Q11

サイト内検索がうまく機能しないのはなぜか?

A11

サイト内検索が機能しない要因としては、サイト内検索ツールの設定でHTTPSを対象にするという調整が漏れていることが考えられる。サイト内検索ツールの設定を見直し、HTTPSを対象にするだけで機能するようになる。

Q12

アクセス数が激減した原因は?

A12

Webサイトへのアクセス数が激減したように見えるのは、ログ解析ツールの設定でHTTPSを対象にする調整が

漏れている場合が多い。こちらももう一度確認しておくこと。

Q13

WordPressをHTTPS化するには?

A13

WordPressにHTTPSを組み込むには、管理画面から「設定」→「一般」を開き、「WordPressアドレス(URL)」「サイトアドレス(URL)」の「http」を「https」に変更する。証明書のインストールやHTTPSの有効化は、Really Simple SSLプラグインをインストールしてアクティブ化するだけで完了する。

Q14

HTTPSサイトを公開する際の注意点は?

A14

第2章を基に、HTTPSサイトを公開する際の必須確認事項をチェックリスト形式で紹介する。他の事項については、第2章を参照されたい。

確認項目	内容
HTMLコードの確認	Webページに読み込む画像、CSS、JSファイルなどに「http://」から始める絶対パスが残されていないかどうかを確認する ※第2章(7)(14)参照
テンプレートHTMLコードの確認	CMSやWebアプリケーションを利用している場合には、それらのテンプレートに記載されているHTMLコードの内容を確認し、画像、CSS、JSファイルなどに「http://」の絶対パスが含まれていないかどうかを点検する ※第2章(8)(14)参照
CookieのSecure属性の確認	CookieにSecure属性が追加されているかどうかを確認し、通信経路上でのCookieの漏えいを回避できるようにしておく ※第2章(9)参照
301転送の確認	Webサイト全体をクローリングし、HTTPのURLからHTTPSのURLへの301転送(301リダイレクト)が正しく行われているかどうかを確認する ※第2章(20)・(22)参照
CMS設定の確認	CMSを利用している場合には、HTTPS化に当たって必要な設定が正しく行われているかどうかを確認する ※第2章(15)参照

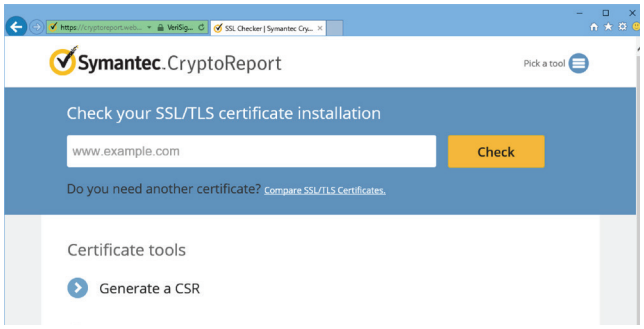
第4章 SSL/TLSに関する有用なツールとガイドライン

常時SSL/TLS化に取り組むにあたっては、数多くの有用なツールやガイドラインがインターネット上に存在している。それらの中から、役に立つ情報をいくつか紹介する。

① Symantec Crypto Report

https://cryptoreport.websecurity.symantec.com/

Symantecが運営する「Symantec Crypto Report」では、Webサイトにインストールされているの証明書の内容を確認することができる。「Check your SSL/TLS certificate installation」の欄にWebサイトのURL（「https://」以降のFQDN）を入力し、「Check」をクリックすると、証明書のチェーンとともにサーバー情報が表示される。結果からは証明書の情報やSSL/TLSの設定に脆弱性が残っていないか、などが分かる。



② SSL/TLS暗号設定ガイドライン（IPA）

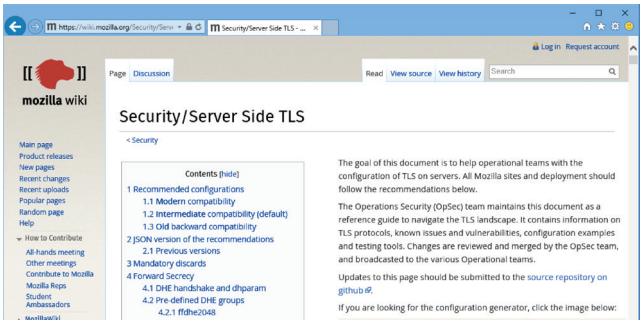
https://www.ipa.go.jp/security/vuln/ssl_crypt_config.html

独立行政法人情報処理推進機構（IPA）の「SSL/TLS暗号設定ガイドライン」は、SSL/TLSサーバーを構築・運営する企業・組織が適切なセキュリティを考慮した暗号設定ができるようにするための実用的なガイドライン。実際に設定すべき「要求設定項目」として3つの設定基準（「高セキュリティ型」「推奨セキュリティ型」「セキュリティ例外型」）を提示している。

③ Security/Server Side TLS - MozillaWiki

https://wiki.mozilla.org/Security/Server_Side_TLS

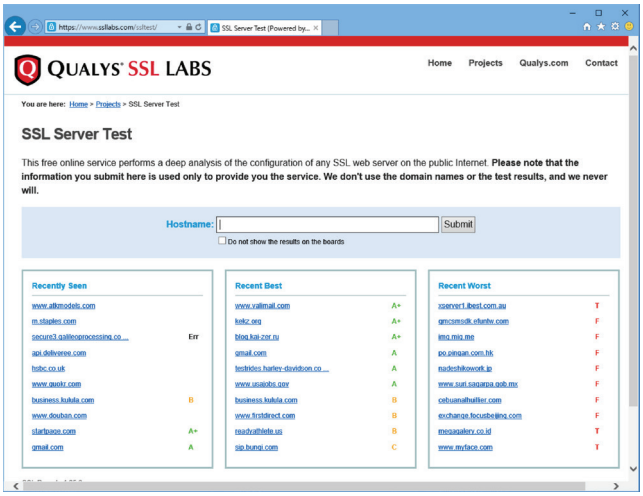
Webブラウザ「Firefox」を開発しているMozilla FoundationのOperations Security（OpSec）チームが運営する「Security/Server Side TLS - MozillaWiki」には、Mozillaが推奨するSSL/TLSに関しての推奨設定をはじめ、既知の問題点や脆弱性、テストツールなどに関する情報が詳細に記載されている（英文のみ）。



④ SSL Server Test

https://www.ssllabs.com/ssltest/

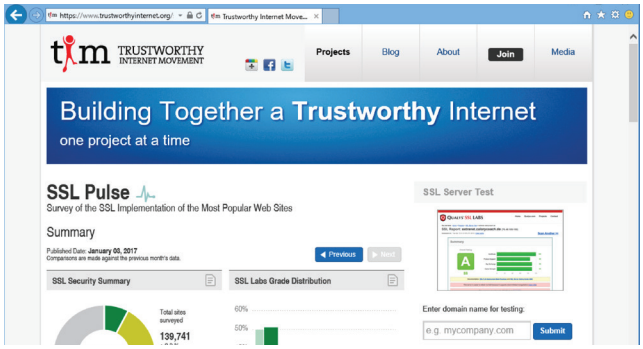
Qualysが提供する「SSL Server Test（Powered by Qualys SSL Labs）」は、SSLサーバ証明書の設定状況の確認や安全性診断が無料で行えるサイト。診断した結果は、A～Fのランクとして公表される。企業・組織が運営するWebサイトのSSL/TLS設定状況を評価したり、利用しているWebサイトの安全性を確認したりするのに有効なツールだ。



⑤ SSL Pulse

https://www.trustworthyinternet.org/ssl-pulse/

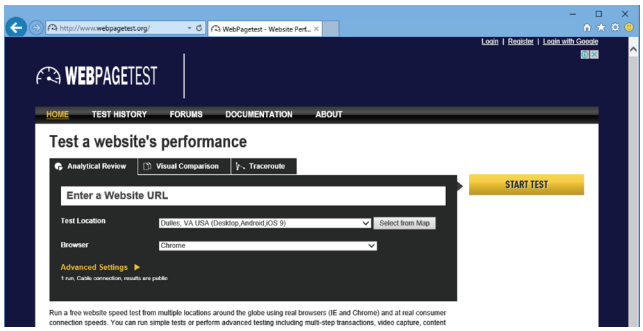
QualysのCEOが設立した非営利組織 Trustworthy Internet Movementの「SSL Pulse」は、Amazon.comのWebサイトのアクセス数を調査する米国Alexaによる世界のアクセストップ100万サイトを対象に、SSL/TLS関連の統計情報をダッシュボードとして公開しているサイト。常時SSL/TLS化が急速に進む状況を確認できる。



⑥ WebPagetest

http://www.webpagetest.org/

Googleのサポートにより運営されているオープンソースプロジェクトの「WebPageTest」は、Webサイトの表示速度を測定するサイト。測定した結果は、A～Fのランクで表示される。企業・組織は自社のWebサイトの速度を測定することにより、Webサイトのパフォーマンスに関する問題の原因を究明することができる。



⑦ 常時SSL/TLSに関する情報サイト(シマンテック・ウェブサイトセキュリティ)

https://www.symantec.com/ja/jp/page.jsp?id=always-on-ssl

シマンテック・ウェブサイトセキュリティが運営する常時SSL/TLSに関する情報サイト。EV SSL証明書を使った「緑色のアドレスバー」が確認できるほか、常時SSL/TLS化を実践するのに役立つ多種多様な情報が掲載されている。

⑧ BadSSL

https://badssl.com/

「BadSSL」は、SSL/TLSのさまざまな状況を再現しながら証明書をテストし、接続の安全性に問題のあるWebサイトを評価するサイト。例えば、自己署名証明書が置かれたページなども用意されており、ブラウザが接続しないようになっているかといったテストが行える。

© 2018 DigiCert, Inc. All rights reserved. DigiCert および DigiCert のロゴは DigiCert, Inc の商標または登録商標です。 シマンテック (Symantec)、ノートン (Norton)、およびチェックマークロゴの商標は Symantec Corporation の ライセンスにもとづき使用されています。
その他の名称もそれぞれの所有者による商標である可能性があります。

デジサート・ジャパン合同会社は、本書の情報の正確さと完全性を保つべく努力を行っています。
ただし、デジサート・ジャパン合同会社は本書に含まれる情報に関して、(明示、黙示、または法律によるものを問わず）いかなる種類の保証も行いません。デジサート・ジャパン合同会社は、本書に含まれる誤り、省略、または記述によって引き起こされたいかなる（直接または間接の）損失または損害についても責任を負わないものとします。さらに、デジサート・ジャパン合同会社は、本書に記述されている製品またはサービスの適用または使用から生じたいかなる責任も負わず、特に本書に記述されている製品またはサービスが既存または将来の知的所有権を侵害しないという保証を否認します。本書は、本書の読者に対し、本書の内容に従って作成された機器または製品の作成、使用、または販売を行うライセンスを与えるものではありません。最後に、本書に記述されているすべての知的所有権に関連するすべての権利と特権は、特許、商標、またはサービス・マークの所有者に属するものであり、それ以外の者は、特許、商標、またはサービス・マークの所有者による明示的な許可、承認、またはライセンスなしにはそのような権利を行使することができません。
デジサート・ジャパン合同会社は、本書に含まれるすべての情報を事前の通知なく変更する権利を持ちます。

デジサート・ジャパン合同会社
https://www.websecurity.symantec.com/ja/jp
03-6893-2727 | websales_jp@digicert.com