UNIXサーバー構築Ⅱ

第1章　WEBサーバーの設定Ⅲ（SSL）

HTTPS(HTTP over SSL/TLS)

- HTTPS （**HTTP over SSL/TLS**）

HTTPSとは、HTTPにTLS(Transport Layer Security)プロトコルを併用して暗号化通信を実現したものです。SSL/TLSを利用して通信内容を暗号化しています。だたし、SSLには**脆弱性**があるため、現在は**TLSを使用**しています。ポート番号は**443**を使用します。

SSL/TLS（ Secure Socket Layer /Transport Layer Security）

■SSL（**Secure Sockets Layer**）/TLS(**Transport Layer Security**)

データを暗号化して通信するためのプロトコル。公開鍵や秘密鍵を使用して、証明書を発行やデータを暗号化します。公開鍵を使用するため、サイトの正当性を証明するために証明書を利用します。

・**認証局（CA：Certification　Authority）**

デジタル証明書を発行して、公開鍵の正当性を保証する機関のこと。

作成者が作成した鍵に対して、公開鍵証明書を発行して認証局の秘密鍵で署名し、その結びつきを証明します。

SSL/TLS（Apache）

ApacheでSSLを利用するには、**mod_ssl**モジュールを利用します。

設定内容をssl．Confファイルなどに書き込み、httpdの基本設定ファイル（httpd.conf）から参照します。

Apacheで利用するSSL証明書の作成などするコマンドは、**openssl**を使用します。シェルスクリプト**CA**を利用すると、自分のサーバ機自身を認証局（自己認証局）として構築することができます。

SSL証明書の内容

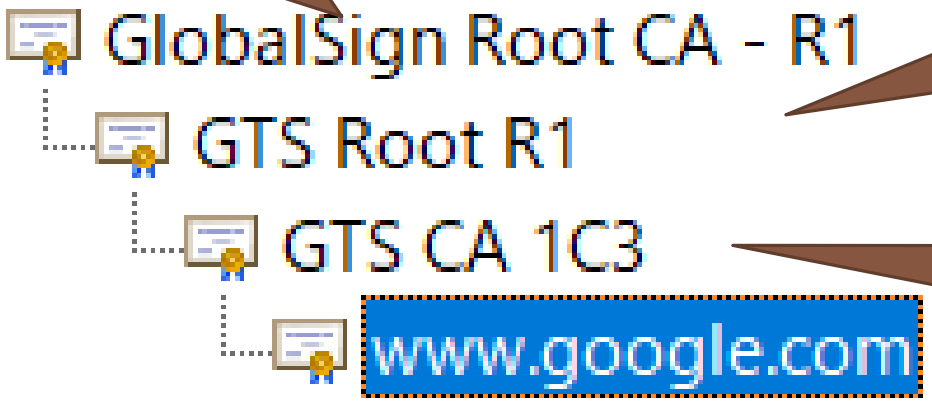

クリック
Google
google.com/?hl=ja
アプリ
この接続は保護されています
お客様がこのサイトに送信した情報（パスワード、クレジッ
ード番号など）が第三者に見られることはありません。詳
クリック
証明書 （有効）
Cookie （26 個が使用中）


何重にも署名され
ていることがわかる
証明書
全般
詳細
証明のパス
証明のパス(P)
GlobalSign Root CA - R1
GTS Root R1
GTS CA 1C3
www.google.com

SSL証明書の仕組み

GTS Rootを認証した機関の証明書（ルート証明書）

ルート証明書はOS購入時にプリインストールされている（Windows Updateなどで更新も可）ため信頼できる



GTS CAを認証した機関の証明書（中間証明書）

googleを認証した機関の証明書（中間証明書）

このサーバーはgoogleで間違いないよという証明と通信に使う公開鍵が入っている

これって誰が証明したの？証明した人もあやしいんじゃ？？

SSL運用手順

① mod_ssl 利用の準備（インストール）

② 公開鍵と暗号鍵（秘密鍵）を作成

③ **公開鍵**を身元を証明する書類とともに**認証局**へ送付
（＝**証明書要求（CSR）**）

④ 認証局は、証明書要求を元に**公開鍵証明書**を発行し、返送。

⑤ 認証局から受け取った 公開鍵証明書と**秘密鍵**をWebサーバにインストール。この証明書を使い、Webサーバは、Webブラウザに対して身元を明確にします。

※証明書は、IPアドレスとドメイン名のペアに対して発行されるため、IPアドレスとドメイン名が変更された場合は、再取得が必要。

SSL対応、Webサーバの構築

自分のサーバ機自身を認証局（自己認証局）とし、自己認証局によって証明書に署名を行う方法（自己署名証明書）で構築します。

※本来はLet ‘s Encryptなどで証明書を発行してもらうことが望ましいです。発行には独自ドメイン名が必要になります。独自ドメインを取るにはグローバルIPアドレスが必要なので、学内ネットワークでは難しくなります。

(1)自己認証局の作成①

認証局を作成（CAスクリプトを使用）

CAスクリプトは**/usr/lib/ssl/misc**にある

コマンド

cd /usr/lib/ssl/misc/

./CA -newca

失敗して、再度実行する前には、作成済みの認証局を削除する
sudo rm -rf /usr/lib/ssl/misc/demoCA/

(1)自己認証局の作成②

以下の内容が聞かれるので、入力する必要がある。以下は例

国コード　　:Country Name　　→　JP

都道府県名　　:State or Province Name→ OSAKA

市区町村名　　:Locality　Name　　→ OSAKA

会社名　　:Organization Name　→ ECCComp

部門名　　:Organization Unit Name→ IT

サーバ名（FQDN）:Common Name →　ie2a99.ecccomp.ac.jp

管理者メールアドレス:Email Address → ie2a99@ecccomp.ac.jp

(2)プライベートキーの作成(秘密鍵)

① SSLプライベートキー保存ディレクトリへ移動

cd /ca/private　　＊ディレクトリ、キー名は任意

② プライベートキーの作成

sudo openssl genrsa -aes256 -out server.key 2048

sudo openssl rsa -in server.key -out server.key

genrsa・・・RSA秘密鍵を生成

aes256・・・暗号化アルゴリズム

server.key・・・秘密鍵のファイル名　＊ファイル名は任意

2048・・・鍵長(2048ビット)

（3）公開鍵の作成

openssl rsa -in server.key -pubout -out public.key

rsa・・・RSA鍵の管理

-in server.key・・・秘密鍵を指定

-pubout・・・公開鍵

-out public.key・・・公開鍵名

公開鍵はサイト証明書発行要求（CSR）にもサーバー証明書（PEM）にも含まれているのでわざわざ作らなくても良い

（4）サイト証明書発行要求（CSR）の作成①

サーバプライベートキーを使用して作成

今回はサーバ名、管理者メールアドレス、は、spamメール対象とならないように実名でなく、一般名「apache」を使用します。

sudo openssl req -new -key server.key -out server.csr　-days 365

req・・・CSRの管理

-key server.key・・・秘密鍵

-out server.csr・・・CSRファイル名

-days 365・・・有効期限（365日）

（4）サイト証明書発行要求（CSR）の作成②

以下の内容が聞かれるので、入力します。以下は設定例

国コード　　：Country Name　　→　JP

都道府県名　　：State or Province Name→ OSAKA

市区町村名　　：Locality　Name　　→ OSAKA

会社名　　：Organization Name　→ ECCComp

部門名　　:Organization Unit Name→ IT

サーバ名（FQDN）：Common Name →　ie2a99.ecccomp.ac.jp

管理者メールアドレス：Email Address → ie2a99@ecccomp.ac.jp

（5）証明書（PEM）の作成①

正式には、（4）で作った証明書発行要求（CSR）と身元を証明する書類を送って認証局に依頼します。

（1）で作った自己認証局で認証します。

＊自己認証局を使用しない場合もあります。

（5）証明書（PEM）の作成②

- 自己認証局がある場合

  sudo openssl ca -out server.pem -infiles server.csr

（エラーが出た場合は、**vi /ca/demoCA/index.txt**

にて内容を削除後、再度実行）

- 自己認証局がない場合

  sudo openssl x509 -days 365 –in server.csr -out server.pem

  –req -signkey server.key

（5）証明書（PEM）の作成③

ca・・・認証局を使用する

-out server.pem・・・証明書

-infile・・・証明書発行要求

-in server.csr・・・証明書発行要求

-days・・・有効期限（日数）

(6)証明書関連ファイルのセキュリティ設定

証明書関連ファイルは所有者以外アクセスできないようにアクセス権を設定します(権限:600)

保存ディレクトリは以下とする(設定例)

sudo chmd 600 /ca

(7)SSLの設定を設定ファイルに登録

sudo vi /etc/apache2/sites-available/default-ssl.conf

DocumentRoot（“/var/www/html”）

ServerAdmin→管理者メールアドレス（ie2a99@ecccomp.ac.jp)

SSLCertificateFile →サーバ証明書のパス（server.pem）

SSLCertificateKeyFile→サーバ秘密鍵のパス (server.key)

■SSLを有効にする

sudo a2ensite default-ssl

sudo a2enmod ssl

(8）Apacheの再起動と動作確認

設定ファイルを変更した場合は、必ずApacheを再起動する

https://IPアドレスで接続すると下記画面が表示される

HTTPからHTTPSへリダイレクト

通常のHTTPでURLを指定した場合でも、HTTPSにリダイレクトするように設定をします。

- 設定ファイル（/etc/apaache2/sites-available/000-default.conf）

  <VirtualHost *:80>のセクションに以下の項目を追記します。

  RewriteEngine On

  RewriteCond %{HTTPS} off

  RewriteRule ^(.*)$ https://%{HTTP_HOST}%{REQUEST_URI}　[R=301,L]

- 設定を有効化、再起動

  a2enmod rewrite

  sudo systemctl restart apache2