Redmineを常時SSL化する。

本記事で実施すること

- インストールしたばかりのRedmineにSSLを設定する。
- 常時SSLで接続できるようにする。

想定している読者

- RedmineにSSLを設定したい。
- SSL暗号化強度を見直したい。

前提

ここでの環境は以下の通りです。

- Ubuntu 20.04系 / Ubuntu 22.04系（Redmine 5.x)
- Apache2.4
- ドメインでRedmineにアクセスできるようになっている。
- 既に有効なSSL証明書と秘密鍵を持っている

特記事項

以下のようなSSL証明書の作成は別項で記載します。

- ローカルネットワークでも証明書を設定したい
- Let's Encryptで証明書を取得したい

手順

さっくりとした手順

1. 証明書を適切なディレクトリに配置します。
2. 常時SSLに対応できるようにモジュールをインストールします。
3. Apacheの設定ファイルを常時SSL化に対応させます。
4. Redmineの設定を変更します。
5. [オプション]外部に公開しているRedmineの暗号化強度を確認します。

証明書の配置

※SSL証明書は、最長でも一年程度と更新サイクルが短くなっています。(Let's Encryptに至っては3ヶ月)

そこで、証明書更新の際にApacheの設定ファイルを修正することなく行えるように、証明書/秘密鍵ファイルにシンボリックリンクを張ります。

ディレクトリを作成します。

```
sudo　mkd　/etc/certs
#　証明書を格納するディレクトリです

sudo　mkd　/etc/private
#　秘密鍵を格納するディレクトリです
```

ディレクトリに証明書と秘密鍵を格納します。

- SCPやSFTPでアップロードして対象ディレクトリに配置する
- Let's Encryptなどで作成したファイルをそれぞれ対象ディレクトリにコピー/移動する

など、適当な方法を用います。

ここではLet's Encryptで2023年1月に作成した「hoge.sample.com (/etc/letsencrypt/live/ドメイン名/に格納されています)」を

- hoge.sample.com.crt.202301 (証明書:/etc/certsに格納)
- hoge.sample.com.key.202301 (秘密鍵:/etc/privateに格納)

と定義します。

※中間証明書と分ける場合は

hoge.sample.com.crt.CA.202301

などと作成するとよいでしょう。

証明書のシンボリックファイルを作成します。

```
cd  /etc/certs && pwd
#  /etc/certsにいることを確認

sudo  ln -sf  hoge.sample.com.crt.202301  hoge.sample.com.crt

ls -l  hoge.sample.com.crt
#  リンクの向き先がhoge.sample.com.crt.202301であることを確認します
```

- ★中間証明書のシンボリック

```
sudo  ln -sf  hoge.sample.com.crt.CA.202301  hoge.sample.com.CA.crt
```

このケースは、Global Sign, Sectigo, GeoTrustのように中間証明書が発行元から提供されている場合です。(Let's Encryptの場合はchain.pemに相当)

秘密鍵のシンボリックファイルを作成します。

```
cd  /etc/private && pwd
#  /etc/privateにいることを確認

sudo  ln -sf  hoge.sample.com.key.202301  hoge.sample.com.key

ls -l  hoge.sample.com.key
#  リンクの向き先がhoge.sample.com.crt.202301であることを確認します
```

証明書の整合性を確認します。

```
openssl  x509 -noout -dates -subject -in  /etc/certs/hoge.sample.com.crt
```

- notBefore=
- notAfter=

の期限が有効期間内であれば、適用できる証明書です。また、

- subject=CN =

が適用するドメインと一致していることを確認します。（例ではhoge.sample.com 。ワイルドカード証明書であれば *.sample.com

- ★証明書と秘密鍵のハッシュ値を確認

```
openssl  x509 -pubkey -in  /etc/certs/hoge.example.com.crt -noout  |  openssl  md5
(stdin)=  ハッシュ値
#  SSL証明書ファイル

openssl  pkey -pubout -in  /etc/private/hoge.example.com.key  |  openssl  md5
(stdin)=  ハッシュ値
#  秘密鍵ファイル

###  2つのハッシュ値が合っていれば証明書と秘密鍵の整合性は取れています
```

証明書と秘密鍵から取り出した公開鍵のハッシュ値を確認します。2つのハッシュ値が合致していれば、適切な証明書と秘密鍵の組み合わせです。

- ★中間証明書の整合性を確認 (Let's Encryptのように中間証明書が結合されている場合)

```
openssl x509 -issuer_hash -noout -in /etc/certs/hoge.example.com.crt

sed -n -e'1d' -e'/BEGIN/,$p' /etc/certs/hoge.example.com.crt | openssl x509 -subject_hash -noout
```

- ★中間証明書の整合性を確認 (別に中間証明書が発行元から提供されている場合)

```
openssl x509 -issuer_hash -noout -in /etc/certs/hoge.example.com.crt

openssl x509 -subject_hash -noout -in /etc/certs/hoge.sample.com.CA.crt
```

証明書と中間証明書のハッシュ値を確認します。2つのハッシュ値が合致していれば、適切な認証局によって署名された証明書となります。

常時SSLに必要なモジュールをインストールします。

モジュールの確認

```
cat /etc/apache2/mods-available/rewrite.load
cat /etc/apache2/mods-available/ssl.load
cat /etc/apache2/mods-available/headers.load
# 内容が表示されていればインストールされています
```

モジュール有効化

```
sudo a2enmod rewrite

sudo a2enmod ssl

sudo a2enmod headers
```

モジュール反映

```
sudo systemctl restart apache2
```

Apache設定ファイルの作成

http接続のみの設定ファイル無効化とコンフィグファイル退避

```
sudo mkd -p /etc/apache2/old
# ファイルのバックアップを格納するディレクトリです

cd /etc/apache2/sites-available

sudo a2dissite redmine.conf
sudo systemctl restart apache2.service
# http接続のみの設定ファイルを無効化し、設定を反映します

sudo mvredmine.conf ../old/redmine.con$(date +%Y%m%)l
# 上記、http接続のみの設定ファイルをバックアップ用ディレクトリに移動します
```

常時SSL化の設定ファイル作成

- 自分の環境に合わせます。（【】でくくっている部分の直下にコメントで詳細を書いています
- cat 〜 _EOF_の部分をコピーして別のエディタに貼り付け → 【】内を編集(このとき、【】も取り除きます) → コマンド実行という流れがやりやすいです。

```
cat <<- __EOF__ | sudo tee -a /etc/apache2/sites-available/redmine.conf
<VirtualHost *:80>
        servername 【hoge.example.com】
        # ドメイン名を指定します
        RewriteEngine On
                RewriteCond %{HTTPS} off
                Rewri$ https://%{HTTP_HOST}%{REQUEST_URI} [R=301,L]
# HTTPアクセスを強制的にHTTPSにリダイレクトします
</VirtualHost>
```

```
<VirtualHost *:443>
        ServerName 【hoge.example.com】
        # ドメイン名を指定します
        DocumentRoot 【/home/www-data/redmine/public】
        # 自身の環境に合わせます
        <Directory 【/home/www-data/redmine/public】>
        # 自身の環境に合わせます
                Options -MultiViews
                AllowOverride All
                Require all granted
        </Directory>

#SSL設定
    SSLEngine on
        Protocols h2 http/1.1
    # SSLを有効化します

SSLCertificateFile 【/etc/certs/hoge.example.com.crt】
# SSL証明書を指定します
SSLCertificateKeyFile 【/etc/private/hoge.example.com.key】
# 秘密鍵を指定します

# SSLCACertificateFile 【/etc/certs/hoge.example.com.CA.crt】
# 中間証明書が発行元から別ファイルで提供されている場合は、この直上をコメントアウトして中間証明書を指定します

#セキュリティヘッダー付与

        Header always set Strict-Transport-Security "max-age=63072000"
        Header set X-Content-Type-Options "nosniff"
        Header always append X-Frame-Options "SAMEORIGIN"
        Header set X-XSS-Protection "1; mode=block"

</VirtualHost>

SSLProtocol                             all -SSLv3 -TLSv1 -TLSv1.1 -TLSv1.2
SSLCipherSuite                          ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-
384:EC6-GCM-SHA384
SSLHonorCipherOrder             off
SSLSessionTickets               off

SSLUseStapling On
SSLStaplingCache "shmcb:logs/ssl_stapling(32768)"
# これらのセクションはSSL暗号化強度を高めるための記述です
# </VirtualHost>の外側に書くことにご注意ください
__EOF__
```

設定を反映します

```
sudo a2ensite redmine
# 再設定したredmineの設定を有効化します

sudo apache2ctl configtest
# Syntax OKを確認します。エラーがある場合はモジュールのインストールミスやファイルの指定ミスが多いです。

sudo systemctl restart apache2.service
```

設定反映後、ブラウザで指定したredmineのドメインにアクセスします。

- https接続になっている
- 証明書が設定されている
- 設定した証明書が有効期限である

を確認します。

Redmineの設定を変更します。

1. Redmineに管理者でログインします
2. 管理 → 設定に移動します
3. 「ホスト名とパス」を例に従って記入します(hoge.example.com)
4. 「プロトコル」をHTTP→HTTPSにします

これで、Redmineの常時SSL化が有効になりました。

（オプション）RedmineサイトのSSL暗号化強度を測定します。

以下の条件を満たしていれば、このチェックが可能です。

- Redmineがインターネット環境に公開されていること。
- Let's Encryptやその他市販の正規の証明書を導入していること。

SSL強度チェックサイトにアクセスします。

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

情報を入力してチェックを行います。

1. Hostname:RedmineのURL
2. 「Do not show the results on the boards」にチェック(サーバにログを残しません)
3. 「Submit」をクリックします。

2023年1月時点で、上記のApache設定ファイル通りであれば「A+」と、ある程度の暗号強度の通信制が担保できています。

ファイル

0_Redmine基本アイコン.jpg	258 KB	2024/01/05	手動人形