# Certbotで無料TLSサーバー

Kazuhiro NISHIYAMA

*Mini Debian Conference Japan 2016*
*2016-12-10*

# 自己紹介

- 西山和広
- id:znz (github, twitter など)
- Ruby コミッター

# Certbot とは?

- https://certbot.eff.org/
- EFF (Electronic Frontier Foundation) による Let's Encrypt クライアント
- 昔は `letsencrypt` (`letsencrypt-auto`) という名前だった
- 他にもクライアントはあります

# Let’s Encrypt とは?

- Let’s Encrypt は、無料で利用できる自動化されていてオープンな認証局（CA）です。公共の利益を図る目的で Internet Security Research Group (ISRG) が運営しています。
  引用元: https://letsencrypt.jp/about/

# 何ができる?

- 無料で使える (商用利用も可能)
- DV (ドメイン認証) 証明書
- 自動発行できる (ACME プロトコル)

# 何ができない?

- EV SSL は無理 (アドレスバーが緑になるもの)
- 証明書に入るのはドメインのみ (組織名などは入らない)
- ワイルドカードも無理

# 普通は困らないけど...

- 短期間に大量発行できない
- https://letsencrypt.org/docs/rate-limits/
- テスト用途には staging サーバーを使いましょう

# 無料証明書といえば...

- notBefore の日付が 2016 年 10 月 21 日より後であり、かつ以下に示す当該ルート証明書にチェインが繋がる証明書への信頼を破棄します。
- 引用元:【翻訳】WoSign と StartCom による今後の証明書は拒否します http://mozsec-jp.hatenablog.jp/entry/2016/10/29/204852

# 無料証明書といえば...

- この変更は Firefox 51 のリリース予定(注2017-01-24) に合わせて反映されます。
- 引用元:【翻訳】WoSign と StartCom による今後の証明書は拒否します http://mozsec-jp.hatenablog.jp/entry/2016/10/29/204852

# 基本的な使い方

- インストール
- 証明書の発行
- 証明書の更新

# インストール (推奨しない)

- `git clone https://github.com/certbot/certbot`
- `./certbot-auto --help` で依存パッケージが **自動で** インストールされる
- 定期実行の仕組みは自分で用意する必要あり (`/etc/cron.daily/local-certbot` を作成するなど)

# インストール (昔の方法)

- 昔は letsencrypt-auto だった (letsencrypt-auto で入れた環境はそのまま使える)
- 今も letsencrypt-auto で入れたサーバーを動かしていますが、問題なく使えています

# インストール (推奨)

- `sudo apt-get install certbot -t jessie-backports`
- apache プラグインを使って apache の設定も自動でするなら python-certbot-apache パッケージ (webroot プラグインしか使ったことがないので説明しません)

# 証明書の発行 (前提条件)

- 単独 Web サーバー (ロードバランサーなどはない)
- ドメイン `www.example.org`
- `DocumentRoot /srv/www/www.example.org/htdocs`

# 証明書の発行 (初回)

```
sudo certbot certonly --webroot \
-w /srv/www/www.example.org/htdocs \
-d www.example.org
```

- メールアドレス入力
- Terms of Service への同意

# メールが送られてくる場合

- 緊急の通知、鍵を紛失したときの復旧、証明書の有効期限が近付いた場合の通知
  引用元: https://letsencrypt.jp/usage/
- 今までメールがきた例: Terms of Service の変更、証明書の有効期限が近づいた時

# 失敗したら

- $DocumentRoot/.well-known/acme-challenge/ファイル名 が外部からちゃんとアクセスできるか確認

# apache に証明書の設定

## 2.4.8 以降なら (jessie はこっち)

```
SSLCertificateKeyFile /etc/letsencrypt/live/www.example.org/privkey.pem
SSLCertificateFile /etc/letsencrypt/live/www.example.org/fullchain.pem
```

## 2.4.7 以前なら

```
SSLCertificateFile /etc/letsencrypt/live/www.example.org/cert.pem
SSLCertificateKeyFile /etc/letsencrypt/live/www.example.org/privkey.pem
SSLCertificateChainFile /etc/letsencrypt/live/www.example.org/chain.pem
```

# 証明書のパス

- (たぶん) 互換性のため certbot に名前が変わっても /etc/letsencrypt のままでした
- /etc/letsencrypt/live 以下に /etc/letsencrypt/archive 以下へのシンボリックリンク
- 更新するごとに archive に連番で証明書や鍵などが溜まっていく

# ログのパス

- /var/log/letsencrypt/letsencrypt.log* にログがある

# 証明書の発行 (2個目以降)

- `sudo certbot certonly --webroot -w /srv/www/hoge.example.org/htdocs -d hoge.example.org`
- メールアドレス入力と Terms of Service への同意はなし (`/etc/letsencrypt/accounts` 以下に保存されたアカウント情報が再利用される)

# 証明書の更新

- `certbot renew` で有効な期間が 30 日未満の証明書があれば更新される
- `/etc/cron.d/certbot` または `/lib/systemd/system/certbot.timer` で 0:00,12:00 に自動実行

# 自動実行のばらつき

- cron.d は perl -e 'sleep int (rand(3600))' で
- certbot.timer は RandomizedDelaySec=3600 で
- jessie-backports の certbot 0.9.3-1~bpo8+1 で確認

# jessie だと

- RandomizedDelaySec は jessie の systemd だと対応していないので `systemctl status certbot.timer -l` で確認すると `[/lib/systemd/system/certbot.timer:6] Unknown lvalue 'RandomizedDelaySec' in section 'Timer'` と出ている
- #843607 で報告

# サーバーに更新リクエストが集中する可能性あり

- パッケージが悪いということで何もしない?
- 更新がある時だけ、と頻度も低いのでそんなに大きな問題にはならない?
- 設定で対処?

# 遅延設定例

/etc/systemd/system/certbot.service.d/delay.conf を以下の内容で作成

```
[Service]
ExecStart=
ExecStart=/bin/bash -c 'sleep $((RANDOM%3600))'
ExecStart=/usr/bin/certbot -q renew
```

- パッケージのバージョンアップなどで対処されたら忘れずに削除すること

# 証明書更新の反映 (1/5)

- renew-hook を使う
  - 更新成功時のみ呼ばれる
- post-hook でも良い
  - 更新試行後に呼ばれる
  - post-hook は pre-hook と組み合わせて standalone プラグインの時に Web サーバーを止めるのに向いている

# 証明書更新の反映 (2/5)

- renew-hook は初回の証明書作成時には呼ばれなかった
- 初回も呼んでほしいなら post-hook の方が良いかもしれない

# 証明書更新の反映 (3/5)

- `/etc/letsencrypt/cli.ini` を以下の内容で作成

```
renew-hook = apachectl graceful
```

- nginx なら `service nginx reload`

# 証明書更新の反映 (4/5)

- ドメインに応じて Web サーバー以外のデーモンを再起動するスクリプトを作ってフルパスで指定するのもあり

```
renew-hook = /etc/letsencrypt/renew-hook
```

# 証明書更新の反映 (5/5)

/etc/letsencrypt/renew-hook

```
apachectl graceful
for domain in $RENEWED_DOMAINS; do
  case "$domain" in
    mx*)
      service postfix reload >/dev/null
      service dovecot reload
      ;;
  esac
done
```

# 基本的な使い方のまとめ

- certbot webroot プラグインで既存の Web サーバーの設定を全くいじらずに証明書発行が可能
- standalone プラグインと違って、ダウンタイムも発生しない

# 様々なトピック

その他の雑多なトピックを紹介

# staging サーバー (1/3)

- rate limit を気にせず試せる
- `/etc/letsencrypt/accounts` 以下に保存されたアカウント情報はサーバーごとなので別アカウントになる
  (本番に切り替えて使っていても staging の有効期限切れが近づくとメールがきた)

# staging サーバー (2/3)

- `--test-cert` オプションまたは `--staging` オプション

```
certbot certonly --test-cert --webroot \
-w /srv/www/www.example.jp/htdocs \
-d www.example.jp
```

# staging サーバー (3/3)

- `/etc/letsencrypt/archive` 以下のファイルの連番は本番と staging で共通
- staging で 1,2 と試した後、本番で発行すると 3 からになる

# rsa-key-size の変更

/etc/letsencrypt/cli.ini で

```
rsa-key-size = 4096
```

(デフォルトは 2048)

# 更新時の通知設定例

moreutils を入れて /etc/systemd/system/certbot.service.d/diffmail.conf を以下の内容で作成 (ExecStopPost の行は実際は 1 行)

```
[Service]
ExecStopPost=/bin/bash -c
"diff -u
<(cut -d: -f4- /var/log/letsencrypt/letsencrypt.log.1 | egrep -v '^DEBUG')
<(cut -d: -f4- /var/log/letsencrypt/letsencrypt.log | egrep -v '^DEBUG') |
ifne mail -s 'Change certbot log' root"
```

# 更新時の通知設定例 (解説)

- timer の実行後に実行する hook はなさそうだった
- service に直接 ExecStopPost を設定するのが良さそうにみえた
- 毎回変わる日時などが混ざる部分を除外して diff
- moreutils の ifne で diff があるときだけメール送信

# メールサーバーの例

- Web を用意できるなら webroot で更新
- メールサーバーに証明書を設定して renew-hook で reload

# 単独 Web サーバーではない環境例 (1/4)

- ロードバランサーで $DocumentRoot/.well-known/acme-challenge を certbot 実行サーバーに固定
- やったことないので詳しいことは不明

# 単独 Web サーバーではない環境例 (2/4)

- DNS で認証したい
- ACME プロトコルには DNS による認証もある
- `certbot` は向かない (途中で止まって手動で DNS 設定するようになっているようだった)

# 単独 Web サーバーではない環境例 (3/4)

- dehydrated (旧 letsencrypt.sh)
- Debian には letsencrypt.sh パッケージがある (2016-12-09 現在 0.3.0-1)
- sid には dehydrated パッケージがある (2016-12-09 現在 0.3.1-1)

# 単独 Web サーバーではない環境例 (4/4)

- https://github.com/lukas2511/dehydrated (旧 letsencrypt.sh)
- Wiki に nsupdate で Bind と連携する例などがある
- 他の API 対応 DNS サーバーの例もある

# 試してないけどメールアドレス変更

certbot --help all の情報によると

```
certbot register --update-registration --email EMAIL
```

で登録しているメールアドレスを変更できそうです (未確認)

# letsencrypt-auto (certbot-auto) からの移行 (1/2)

- `/etc/letsencrypt` 以下は共通
- `certbot` パッケージを入れて、運用を切り替えるだけでいけそう
- 試してないけど

# letsencrypt-auto (certbot-auto) からの移行 (2/2)

- git clone したディレクトリとか
- `~/.local/share/letsencrypt` 以下とか
- 不要になったパッケージはうまく消す必要あり

# letsencrypt パッケージ時代に入れた環境からの移行 (1/2)

- `sudo apt-get install certbot -t jessie-backports` で入るはず
- どこかのバージョンアップのタイミングで `sudo apt-get install` 引っかかったパッケージ `-t jessie-backports` という作業が必要だったことも

# letsencrypt パッケージ時代に入れた環境からの移行 (2/2)

- そのまま purge すると `/etc/letsencrypt` `/var/log/letsencrypt` が消されてしまうので、`sudoedit /var/lib/dpkg/info/letsencrypt.postrm` でコメントアウトしてから purge すると良い

# まとめ

- certbot で無料で SSL/TLS サーバー証明書
- 自動更新で運用の手間いらず
- メールサーバーなど Web 以外でも使える
- 複数サーバー環境などでは dehydrated など他のクライアントの方が良いかも