Let's Encrypt & ACME Overview

Let's Encrypt

Let's Encrypt

- 以下の事項を主義とする認証局
  - Free
  - Automatic
  - Secure
  - Transparent
  - Open
  - Cooperative
- 提供しているのは Internet Security Research Group (ISRG)

Let's Encrypt

- 無料で SSL サーバ 証明書(DV)を入手することが出来る
- 発行された証明書の有効期限は 90 日間
  - 理由: https://letsencrypt.org/2015/11/09/why-90-days.html
  - ツールの充実と共に更に短くする予定らしい
- Domain validation は ACME というプロトコルに従い行われる
  - ACME(Automatic Certificate Management Environment)

ACME (Automatic Certificate Management Environment)

ACME

- Internet draft (- 2016/1/22)
- https://letsencrypt.github.io/acme-spec/
- サーバ/クライアント間での証明書発行の手続きを策定
- 実際に証明書を発行(破棄)するまでに大体次のような手続きが必要
  - Register
  - Authorizations
  - New Cert (Revoke-cert)
- クライアント側の実装は https://github.com/letsencrypt/letsencrypt (Python)
- CA 側の実装は https://github.com/letsencrypt/boulder (Golang)

Directory

- それぞれの手続きに必要な endpoint を directory で提供
- クライアントはまずここを見て endpoint を把握する

```
✗ curl -sSL https://acme-v01.api.letsencrypt.org/directory | jq .
{
  "new-authz": "https://acme-v01.api.letsencrypt.org/acme/new-authz",
  "new-cert": "https://acme-v01.api.letsencrypt.org/acme/new-cert",
  "new-reg": "https://acme-v01.api.letsencrypt.org/acme/new-reg",
  "revoke-cert": "https://acme-v01.api.letsencrypt.org/acme/revoke-cert"
}
```

Register

- まず ACME Server 側にクライアントの登録を行う
- 次のような "contact" フィールドを含んだ JSON を送る
  - JWS(JSON Web Signature) で署名を付ける必要がある

```
{
  "resource": "new-reg",
  "contact": [
    "mailto:cert-admin@example.com",
    "tel:+12025551212"
  ],
}
/* Signed as JWS */
```

- "key" を含んだレスポンスが返ってくるので以降の手続きはそれを使って signature を作る

Authorization

- 証明書発行の認可を行う手続き
- Domain validation をどう行うか等を指定する
- 次のような方法が選べる
  - SimpleHttp
  - DNS
  - DVSNI
  - Proof of possession of a prior key
- 複数の方法を "combinations" の配列で指定することが出来る
  - combination の全てを満たした場合に valid とする

- リクエストの例

```
{
  "status": "pending",

  "identifier": {
    "type": "dns",
    "value": "example.org"
  },

  "challenges": [
    {
      "type": "simpleHttp",
      "uri": "https://example.com/authz/asdf/0",
      "token": "IlirfxKKXAsHtmzK29Pj8A"
    },
    {
      "type": "dns",
      "uri": "https://example.com/authz/asdf/1"
      "token": "DGyRejmCefe7v4NfDGDKfA"
    }
  },

  "combinations": [
    [0, 2],
    [1, 2]
  ]
}
```

Authorization (Challenges SimpleHttp/DNS)

- SimpleHttp
  - HTTP(S) にてアクセスを行いドメイン所有を確認する
  - アクセスを行う先は A レコードもしくは AAAA レコードから決定される
  - アクセス先の `.well-known/acme-challenge/${TOKEN}` を見る
    - 中身には所定の JSON を入れておく
- DNS
  - DNS レコードを用いてドメイン所有を確認する
  - `_acme-challenge` サブドメインの TXT レコードを使用する
  - 値を TOKEN にする
    - ex.) `_acme-challenge.example.com. 300 IN TXT "gfj9Xq...Rg85nM"`

New Cert (Revoke-cert)

- 前述の Authorization の status が valid な場合のみリクエストできる
  - valid じゃないときは 403 とかが返る
- New Cert は `/acme/new-cert` に CSR を送りつける
  - 勿論 JWS で署名する必要がある
  - 色々あった後 DER 形式の証明書を取得できる
- Revoke は `/acme/revoke-cert` に証明書を送りつける
  - CRL/OCSP 等に失効情報が公開される

letsencrypt

letsencrypt

- 前述の ACME の諸々をやってくれるコマンド(python)
  - 基本的に SimpleHttp による Challenge を想定している模様
- 引数ベースでドメイン等のパラメータを設定する
- Apache や Nginx の設定をパースしたり書き換えたりも出来る
  - Nginx は experimental, buggy and not installed by default とのこと
- standalone を指定すると BaseHTTPServer(http.server) を使って Challenge を行う
  - 80 番ポートを LISTEN 出来る必要がある

letsencrypt-auto

- letsencrypt コマンドのラッパー
- 実行するだけで諸々やってくれる
  - 環境構築(yum/apt/brewとか virtualenv とか pip とか)
    - (_gentoo_common.sh もあった)
  - Virtualenv の activate とかもやってくれる
- 公式ドキュメントではこれを使うことになってた
- 毎回 pip install で最新かどうかチェックしたりして若干重い

letsencrypt-auto

- standalone で証明書を取得してみる
- -d を複数指定することで SANs に複数のドメインが書かれる模様
- 色々あったあと /etc/letsencrypt/ 配下に様々なディレクトリが生成される
  - 最新の証明書は /etc/letsencrypt/live/${DOMAIN_NAME} 配下に置かれる
    - /etc/letsencrypt/archive 配下のもののシンボリックリンク

```
./letsencrypt-auto \
  -a standalone \
  -d example.com \
  -d www.example.com \
  --server https://acme-v01.api.letsencrypt.org/directory \
  --agree-dev-preview \
  auth
```

letsencrypt

- `ssl_certificate` には `fullchain.pem` を指定する
- `ssl_certificate_key` には `privkey.pem` を指定する
- 当然だが Postfix/Dovecot でもちゃんと使えている
  - iPhone/Android からもエラー無く接続できている
  - `smtpd_tls_(cert|key)_file` 等に同様に指定するだけ
    - (居ないと思うが) 古ーい dovecot で使う際は結合順に注意が必要

所感

- 無料で DV 証明書が入手できるなんて良い時代
  - 通常の証明書と何ら相違無く利用できている
    - 有効期限が短めなので更新自動化は必須な気がする
- ACME プロトコルの性質上 DNSSEC に対応したりした方が良さそう
  - https://letsencrypt.github.io/acme-spec/#integrity-of-authorizations
  - どうでもいいけど最近の CloudFlare はワンクリックで DNSSEC やってくれる
    - https://blog.cloudflare.com/introducing-universal-dnssec

参考資料

- https://letsencrypt.org
- https://letsencrypt.org/about
- https://letsencrypt.org/howitworks/technology
- https://letsencrypt.org/2015/10/19/lets-encrypt-is-trusted.html
- https://letsencrypt.org/2015/11/09/why-90-days.html
- https://letsencrypt.github.io/acme-spec/
- https://github.com/letsencrypt/boulder
- https://github.com/letsencrypt/letsencrypt
- https://acme-v01.api.letsencrypt.org/directory
- https://letsencrypt.github.io/acme-spec/#integrity-of-authorizations
- http://jxck.hatenablog.com/entry/letsencrypt-acme