Let's Encrypt:

Automate all the things

Philip Paeps

SANOG 36 — Nepal

January 2021

FreeBSD

Getting a certificate: common process

1. Generate certificate
2. Build certificate signing request (CSR)
3. Convince certification authority (CA) to sign certificate
4. Repeat annually

Certificate authorities

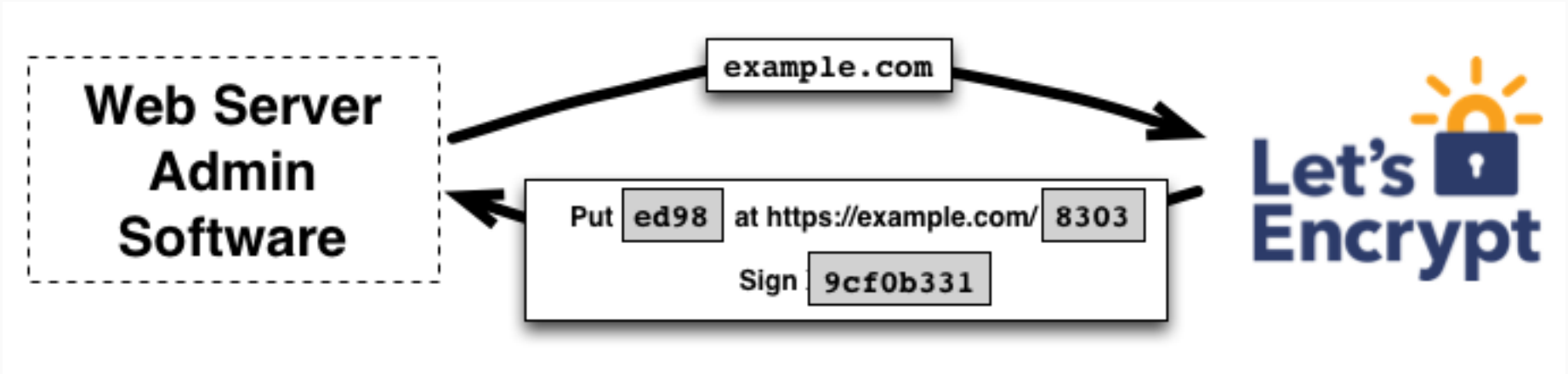
A fundamentally flawed system

- CAs are answerable first to their shareholders. Their primary objective is profit. Improving security on the internet is a secondary concern.
- Anything that happens *annually*, tends to happen *manually*.
  - Error prone
  - Easy to forget

Let’s Encrypt: a not-for-profit CA

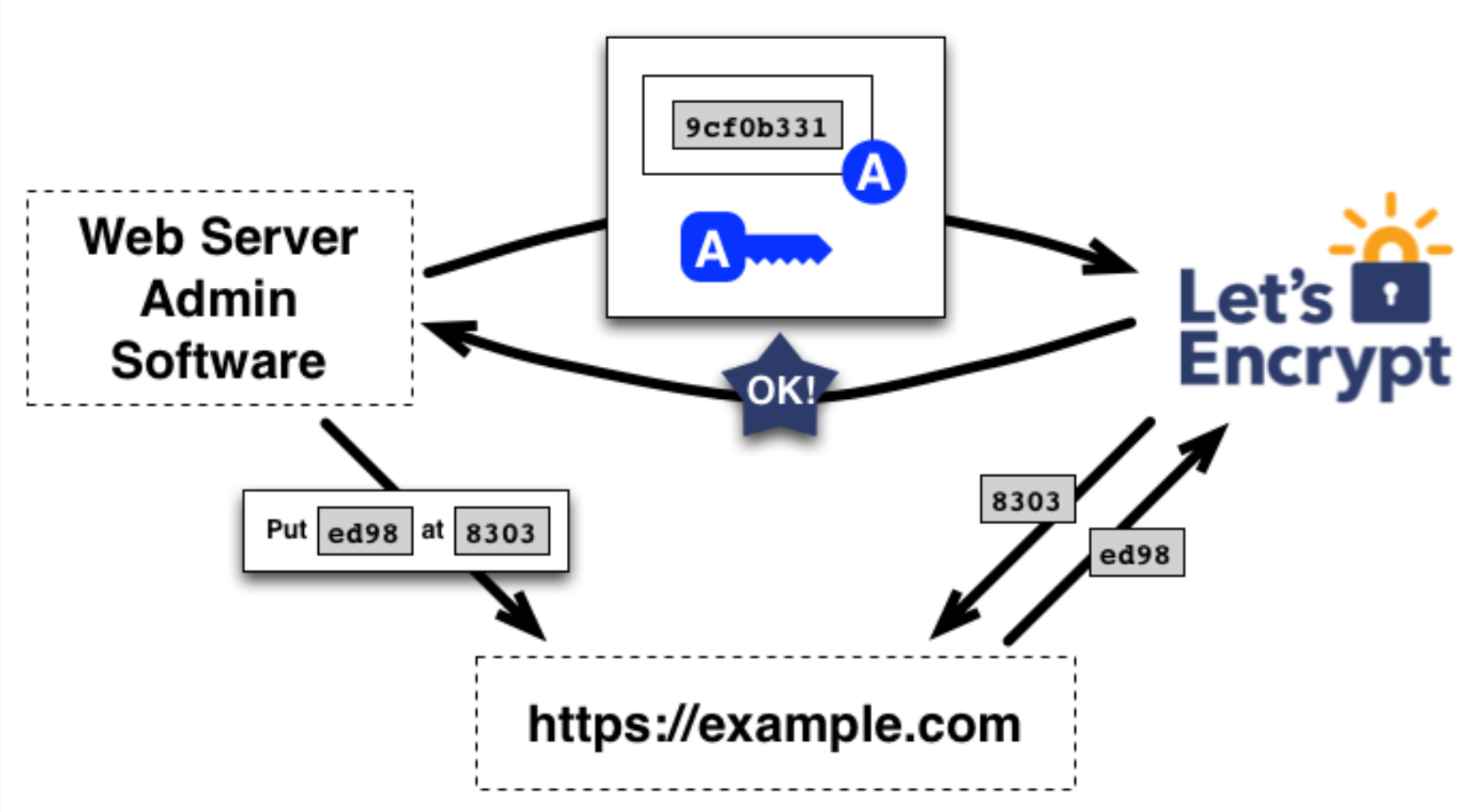
- Free
- Automatic
- Secure
- Transparent
- Open
- Cooperative

How does this work?



FreeBSD

How does this work?



FreeBSD

Under the hood: ACME

- RFC 8555: Automated Certificate Management Environment
- JSON-based protocol over HTTPS
- Several implementations

ACME implementations

certbot

- Reference implementation
- Many dependencies
- Many assumptions

acme.sh

- Tiny shell script
- No awkward dependencies

Many many other clients exist:

https://letsencrypt.org/docs/client-options/

Lists over a hundred (as of November 2020)

acme.sh in a nutshell

Installation

The installer does 3 things:

1. Create and copy acme.sh in $HOME/.acme.sh
2. Alias for acme.sh
3. Daily cronjob to check and renew certificates

```
$ git clone \
https://github.com/acmesh-official/acme.sh.git
$ cd ./acme.sh
$ ./acme.sh --install

OR

$ curl https://get.acme.sh | sh
```

acme.sh in a nutshell

Issue a certificate

```
Single domain:
$ acme.sh --issue -d example.com -w /home/wwwroot/example.com
$ acme.sh --issue -d example.com -w /home/username/public_html
$ acme.sh --issue -d example.com -w /var/www/html

Multiple domains:
$ acme.sh -issue            \
  -d example.com            \
  -d www.example.com        \
  -d cp.example.com         \
  -w /home/wwwroot/example.com
```

Many many options

Opportunities for automation

- Standalone mode
  - No webserver needed
  - Requires privilege to bind to port 80
  - Still requires publicly reachable machine
- DNS mode
  - This is where the real magic is!

DNS mode

Demo

```
$ acme.sh --issue -k 4096 --dns dns_azure --challenge-alias certbot.trouble.is -d pacnog27.trouble.is
[Mon Nov 30 07:45:43 UTC 2020] Creating domain key
[Mon Nov 30 07:45:45 UTC 2020] The domain key is here: /home/certbot/.acme.sh/pacnog27.trouble.is/pacnog27.trouble.is.key
[Mon Nov 30 07:45:45 UTC 2020] Single domain='pacnog27.trouble.is'
[Mon Nov 30 07:45:45 UTC 2020] Getting domain auth token for each domain
[Mon Nov 30 07:45:47 UTC 2020] Getting webroot for domain='pacnog27.trouble.is'
[Mon Nov 30 07:45:47 UTC 2020] Adding txt value: QweUWz4_efyxJM9EfTlBQr9D11Kj0WJ_7DBpx19jH7g for domain:  _acme-challenge.certbot.trouble.is
[Mon Nov 30 07:45:52 UTC 2020] validation value added
[Mon Nov 30 07:45:52 UTC 2020] The txt record is added: Success.
[Mon Nov 30 07:45:52 UTC 2020] Let's check each dns records now. Sleep 20 seconds first.
[Mon Nov 30 07:46:14 UTC 2020] Checking pacnog27.trouble.is for _acme-challenge.certbot.trouble.is
[Mon Nov 30 07:46:14 UTC 2020] Domain pacnog27.trouble.is '_acme-challenge.certbot.trouble.is' success.
[Mon Nov 30 07:46:14 UTC 2020] All success, let's return
[Mon Nov 30 07:46:14 UTC 2020] Verifying: pacnog27.trouble.is
[Mon Nov 30 07:46:18 UTC 2020] Success
[Mon Nov 30 07:46:18 UTC 2020] Removing DNS records.
[Mon Nov 30 07:46:18 UTC 2020] Removing txt: QweUWz4_efyxJM9EfTlBQr9D11Kj0WJ_7DBpx19jH7g for domain: _acme-challenge.certbot.trouble.is
[Mon Nov 30 07:46:22 UTC 2020] validation record removed
[Mon Nov 30 07:46:22 UTC 2020] Removed: Success
[Mon Nov 30 07:46:22 UTC 2020] Verify finished, start to sign.
[Mon Nov 30 07:46:22 UTC 2020] Lets finalize the order, Le_OrderFinalize: https://acme-v02.api.letsencrypt.org/acme/finalize/67909898/6485905412
[Mon Nov 30 07:46:23 UTC 2020] Download cert, Le_LinkCert: https://acme-v02.api.letsencrypt.org/acme/cert/03896bb36534a837d3d47949ea2485987399
[Mon Nov 30 07:46:24 UTC 2020] Cert success.
-----BEGIN CERTIFICATE-----
[…]
-----END CERTIFICATE-----
[Mon Nov 30 07:46:24 UTC 2020] Your cert is in  /home/certbot/.acme.sh/pacnog27.trouble.is/pacnog27.trouble.is.cer
[Mon Nov 30 07:46:24 UTC 2020] Your cert key is in  /home/certbot/.acme.sh/pacnog27.trouble.is/pacnog27.trouble.is.key
[Mon Nov 30 07:46:24 UTC 2020] The intermediate CA cert is in  /home/certbot/.acme.sh/pacnog27.trouble.is/ca.cer
[Mon Nov 30 07:46:24 UTC 2020] And the full chain certs is there:  /home/certbot/.acme.sh/pacnog27.trouble.is/fullchain.cer
```

DNS mode

How does this work

- Verify ownership of domain by checking for TXT record
- Indirection: CNAME verification domain to Azure
  - For automation: Azure API is easy to use
  - For security: don’t expose API access to domain’s DNS
  - Could use ns-update to self-host, but Azure is easier
  - Could use any number of other DNS providers
- acme.sh will renew the certificate regularly (cron)

DNS mode

How does this work

```
trouble.is DNS:

; certbot keeps LetsEncrypt.org certificates fresh.
;   acme.sh on rincewind.trouble.is manages LetsEncrypt.org challenges
;   using the Azure API.  Domains we manage need a CNAME pointing
;   _acme-challenge.DOMAIN. to _acme-challenge.certbot.trouble.is.
certbot          NS      ns1-04.azure-dns.com.
                 NS      ns2-04.azure-dns.net.
                 NS      ns3-04.azure-dns.org.
                 NS      ns4-04.azure-dns.info.

[...]

; PacNOG 27 Let's Encrypt talk
pacnog27      A      127.0.0.1
              AAAA   ::1
              MX 0   .
_acme-challenge.pacnog27   CNAME _acme-challenge.certbot
```

The end

- Pointers to more resources:
  https://letsencrypt.org
  https://github.com/acmesh-official/acme.sh/wiki

- Contact me: philip@trouble.is