A Free Certificate Authority to Encrypt the Entire Web

Seth Schoen
Senior Staff Technologist, EFF

Acronyms

- SSL (Secure Sockets Layer) – the old name for the main security layer for TCP
- TLS (Transport Layer Security) – the modern name for SSL
- HTTPS (HTTP Secure) – HTTP plus TLS
- X.509 – the format used by TLS certs
- PKI (Public Key Infrastructure) – an infrastructure for distributing crypto keys
- CA (Certificate Authority) – an entity that issues digital certificates in a PKI

Importance of TLS

- Still occasionally dealing with the idea that it's only needed for *financial data*
- … more often these days, the idea that it's only needed for *logins*

- We need to articulate a stronger vision that *networks are untrustworthy and communications need to be protected*
- Networks are routinely attacking us and plain HTTP offers no defense

Just a few examples

- Sidejacking and location tracking
- Integrity of software downloads
- Reader privacy
- Content-based censorship prevention
- Protection against ad injection, tracking-header-injection and even malware injection at ISP

Barriers to adoption

- Perception that TLS is slow (especially for session establishment) or is very computationally intensive
- Difficulty integrating into some server and data center designs (like load balancing)
- *Cost and effort of obtaining and managing PKI certificates*
- Even a skilled person who understands PKI conceptually may take ~1 hour to get and deploy a cert … and then it may expire, or omit some vhosts

Let's Encrypt

- Initially, a collaboration among EFF, University of Michigan, and Mozilla
  - to create a fully automated CA to issue certificates to any site for any purpose, quickly and at no charge
  - Aiming to be cheaper, easier, *and* more secure than existing CAs
- Thanks to partners including Akamai, Cisco, and IdenTrust, we'll have *publicly trusted* certificates accepted by browsers

Cross-signing

- Root CAs can and do delegate their authority to intermediate CAs — currently hundreds of named entities
- Browsers then *automatically trust* these intermediates; end-entity certs are almost always signed by intermediates, not roots
- Our CA will be cross-signed by IdenTrust; mainstream browsers will trust us *immediately*; browser users won't have to install our CA's root certificate

Let's Encrypt concept

- Lowest level of validation for PKIX certificates is DV (Domain Validation) — verification by the CA that the applicant *controls the domain name* (or a server that the domain name is pointed at)
- Explicitly doesn't confirm the identity of the applicant
- We can *replace the certificate authority with a very small shell script*

Let's Encrypt concept

- OK, there's actually a lot of engineering work plus work to comply with industry standards (Baseline Requirements and WebTrust audit) and that “shell script” may grow in size

- But DV certificate issuance can be fully automated in the common case, and that's what we're going to do!

Let's Encrypt concept

- Client (user's web server) connects to server (Let's Encrypt CA) using a client application (that may be bundled with the OS or offered by a hosting or platform provider)

- We're developing a protocol called ACME (Automated Certificate Management Environment) to handle conversations about cert issuance

Let's Encrypt concept

- Client claims to control a particular name or names, and asks for a cert for them
- Server issues one or more *challenges* to ask the client to prove its control (and/or possibly prove control of other cryptographic keys)
- Client satisfies these challenges and server verifies this automatically, then issues cert and sends it over

ACME

- A JSON-based protocol for talking about certificate issuance and revocation, primarily invented by Richard Barnes
- Handling each step in our DV issuance process

Request	Response
challengeRequest	challenge
authorizationRequest	authorization
certificateRequest	certificate
revocationRequest	revocation
statusRequest	(any)

Let's Encrypt status

- We're incorporated as the Internet Security Research Group (ISRG) to pursue the Let's Encrypt CA and other initiatives to improve Internet security
  - U.S. nonprofit status is pending
- We're preparing for WebTrust audits and build-out and expect to have public issuance in summer 2015
- Right now we have a tech preview and welcome testing and collaboration

DVSNI

- One validation method we've developed that's stronger than existing manual DV challenges used by some CAs today
- Basically, the verifier asks the applicant to put up a self-signed cert containing certain server-provided information
- Then the verifier connects and negotiates a TLS session and checks that the cert does contain that information
- Proves *control of the web server itself*

Convenience

- We anticipate people who administer their own web servers will run something like

```
sudo apt-get install lets-encrypt

sudo lets-encrypt
```

  and the lets-encrypt client will not only *obtain*, but also *deploy*, the new cert in less than one minute
- We're working on a client that can parse and write Apache (and other) configs

Safety

- We care a lot about avoiding misissuance and plan to adopt technologies to stop it
- One possibility is publishing all certs in Google's Certificate Transparency system
- We may have a policy preventing issuance for a domain that already has a valid cert unless the applicant can prove control of its subject key
- We can also have mechanisms for domains to ask us never to issue for them

Wider integration

- We'd like to be integrated on every server OS or web server and every hosting and application platform
- The ACME protocol is likely to be submitted to standardization at IETF and will be an open standard
- You can use the protocol to request certs from us without using our client software
- Contractual relationship isn't required, though we welcome new sponsors

Thanks!

You can contact me with any questions:

Seth Schoen <schoen@eff.org>

FD9A 6AA2 8193 A9F0 3D4B F4AD C11B 36DC 9C7D D150

https://letsencrypt.org/

https://github.com/letsencrypt

Thanks to our colleagues with whom we're developing Let's Encrypt and ACME, including Josh Aas (Mozilla), Richard Barnes (Mozilla), Peter Eckersley (EFF), Alex Halderman (UMich), James Kasten (UMich), Eric Rescorla (Mozilla)