

Let's Encrypt and DANE

ENOG 11 | Moscow | 8 Jun 2016

Disclaimer:

**I do not work for Let's Encrypt,
but the Internet Society is a sponsor**

**Slides contributed by:
Josh Aas (Let's Encrypt)
Jan Žorž (Internet Society)**

Thanks also to Viktor Dukhovni



A Free, Automated and Open Certificate Authority (CA)

- To encourage and simplify adoption of TLS and provide platform for advancing CA and end user security.
- Certificates are free for anyone owning a domain name
- Software running on a web server can automatically obtain, configure and renew a certificate
- All certificates issued or revoked will be publicly recorded and available for anyone to inspect.
- The automatic issuance and renewal protocol is published as an open IETF standard that others can adopt.
- Let's Encrypt is an effort to benefit the community beyond the control of any one organisation.



Why do we need it?

- The Internet is inherently insecure, unencrypted traffic can be read and modified.
- TLS (formerly SSL) is used to encrypt web connections using HTTPS, but can also be used for e-mail, instant messaging, streaming, VoIP and other applications.
- Around 40% of Firefox page loads are HTTPS, and 62% of emails received by Gmail are over STARTLS.
- Aim is to have 100% of Internet traffic encrypted, and by default.
- CAs are (currently) needed to validate domains and link them with public keys used to establish encrypted connections.



But, authentication via traditional CA is not easy

- CAs issue the X.509 certificates required for TLS on the Web



- Interaction with CAs for certificate issuance and management is complex



In fact, it's too hard...



I can't f'ing figure out how to get a cert ...

God help people that don't know what a CSR is...

I am like 45 minutes in.

Cullen Jennings, Cisco Fellow



The traditional way...

- Figure out you need a certificate
- Work out where to get the certificate from
- What type of certificate you need - DV, OV, EV or maybe you want the Regular, Pro, Gold or Platinum versions? Hmm...
- Figure out how to request a certificate, including making a CSR
- Go through manual verification process, hoping that responses go to the right e-mail addresses (assuming they exist in the first place), and your organisation has a publicly listed phone number and/or DUNS number etc..
- Pay
- Figure out how to install your certificate
- Don't forget to renew it on time!



Let's Encrypt is born...

- Josh Aas and Eric Rescorla decided to address this problem after discussions in IETF HTTP/2 Working Group
- Decided solution was to establish CA offering free certificates globally, but (only) supporting automated processes.
- Established Internet Security Research Group in May 2013, a foundation with IRS 501(c)(3) status
- Mission is to reduce financial, technological and educational barriers to secure communication over the Internet.
- Let's Encrypt announced on Nov 2014, started trials in Sep 2015, public beta in Dec 2015, and entered production service in Apr 2016
- Funded by 30+ sponsors making long-term commitments





What's different?

- Automated Certificate Management Environment (ACME), the key to Let's Encrypt
- Provides API for requesting, validating, revoking and otherwise managing certificates
- Clients **must** request certificates from Let's Encrypt via ACME
- Certbot is recommended client software, but others are available
- ACME is being standardized:





The technical details...

- Let's Encrypt certificates are Domain Validation (DV) – asserts that holder has control over a domain
- Validation is undertaken through challenge verification
 - HTTP: a file on your web server
 - DVSNI: provisioning virtual host at your domain's IP address
 - DNS: provision a DNS TXT record for your domain
- Certificates have 90-day lifetime
 - Encourages automation
 - Limits damage from key compromise or mis-issuance
 - Mitigates the problems of certificate revocation

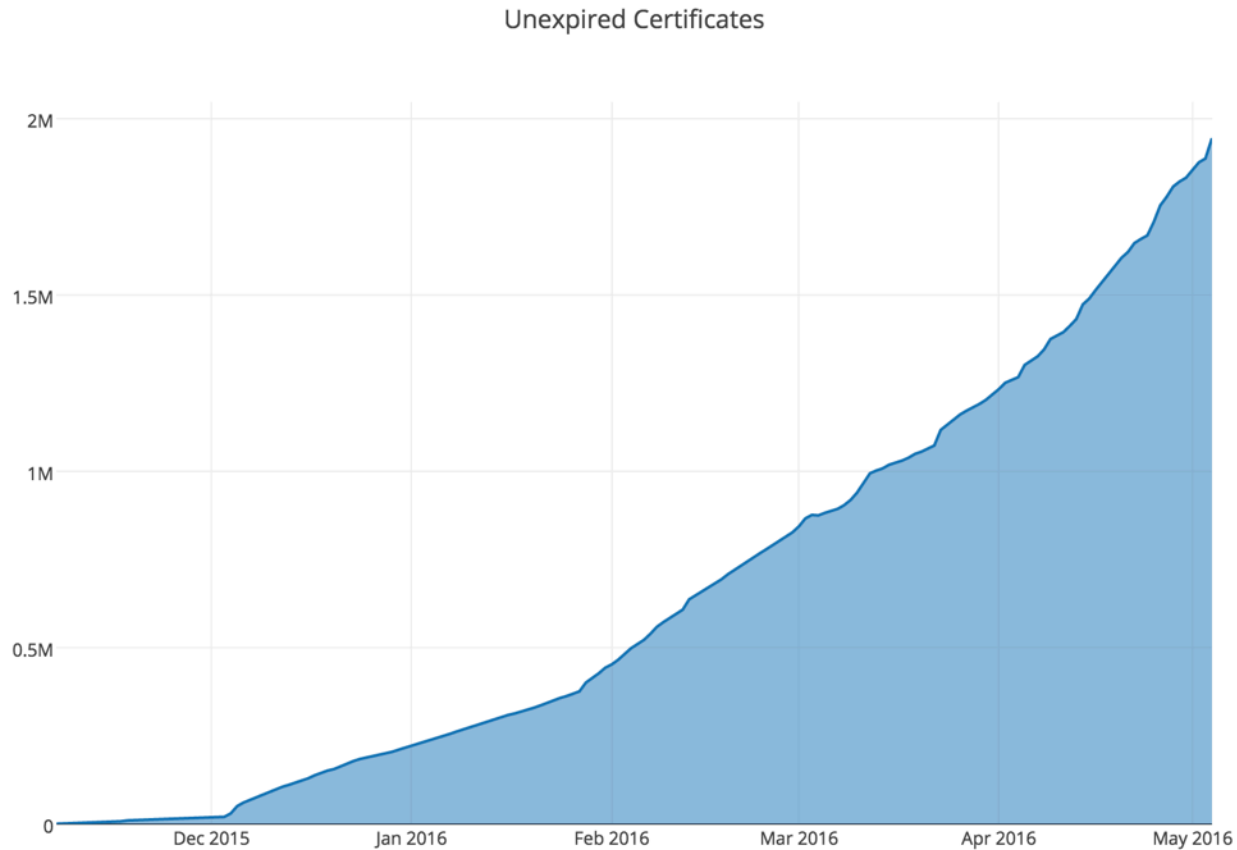


Trust anchor...

- End entity certificates are validated through a chain-of-trust originating from a root certificate, known as the trust anchor
- Root certificate trust is established through distribution of root certificates in operating systems or browsers
 - Main certification programs are Microsoft, Apple and Mozilla
- Let's Encrypt issues certificates from intermediate CA called Let's Encrypt Authority X3, signed by ISRG Root X1
- ISRG Root X1 is not yet trusted in all OSs and browsers so cross-signed by IdenTrust DST Root CA X3



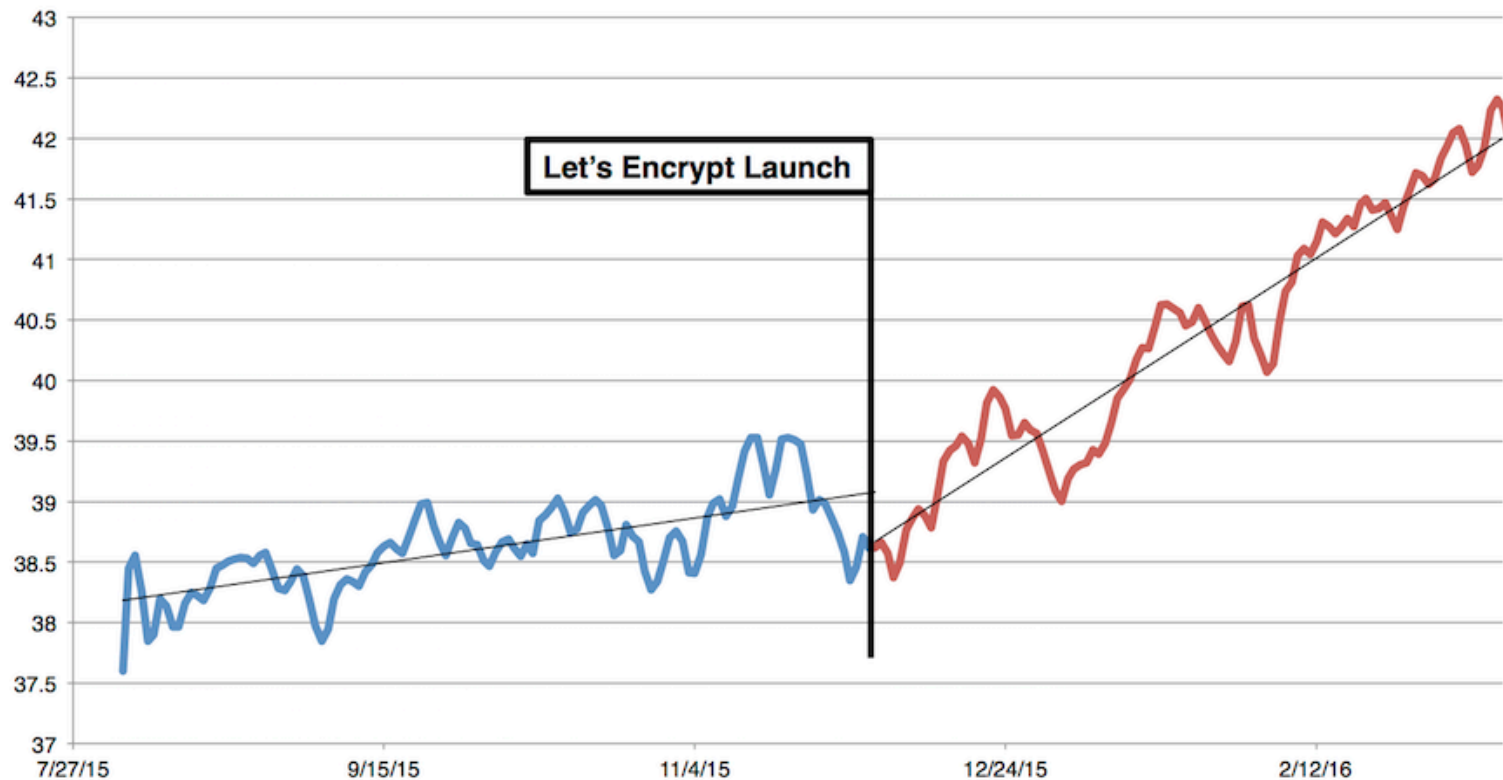
How successful is it?





Is it encouraging use of TLS?

Percentage of Firefox Pageloads using HTTPS (15-day moving average)



The Internet is not just the Web though!

ISOC / Go6Lab tested SMTP connections to top 1 million domain names

- 70% of attempted sessions were encrypted with TLS
- 60% of TLS sessions were established with trusted certificate
- Other sessions used untrusted certificate or opportunistic TLS
- None of the top 10 mail servers and their domains were DNSSEC signed (as of Jan 2016)
- Only 128 sites could be verified with TLSA by DANE!
- How can this be improved?

DNS-Authenticated Name Entities (DANE)

DANE allows certificates to be cryptographically bound to DNS names

- Domain administrators can certify the keys used for establishing TLS sessions by storing them in the DNS
- Aims to improve inherent situation whereby any CA can issue certificate for any name
- Requires DNS records to be signed with DNSSEC
- Clients need to be DNSSEC and DANE aware

DNS TLSA records associate certificates or public keys with domains

- *Usage*, *Selector* and *Matching Types* fields determine which CAs, certificates and hashes can be accepted
- Let's Encrypt recommends 2 1 1 and 3 1 1 records

When do things fail in DANE?

When do things fail in DANE?

- TLSA record is published in non-DNSSEC zone (therefore can't trust the data)
- Domain where looked up records (e.g. MX) reside are not DNSSEC signed (therefore no verification)
- Wrong certificate hash in TLSA record

Let's Encrypt, DANE and email

Validity of Let's Encrypt certificate is 90 days

- By default the underlying key is changed is changed when renewing
- So also is hash, so work needed if planning to publish 3 1 1 TLSA
- Using 2 1 1 TLSA means lack of DST Root CA X3 in certificate chain
- So need to fetch DST Root CA X3 certificate and add it to *fullchain.pem* file
- However, how do we know if root certificate has changed?
- Trick is to use same CSR when Let's Encrypt certificate is renewed, and generate 2048 bit RSA private key
- This means same underlying key can be used for 3 1 1 TLSA records, so hash doesn't change
- Underlying key can be rotated on different schedule

Moral of the story...

70% of e-mail can already be encrypted in some way – so enable TLS on your server!

Let's Encrypt offers free certificates and automated ordering/installation, so no excuse not to make encryption the default on the Internet

Using DANE is not that difficult and can be automated

But...

We need more DNSSEC signed domains!

We need even more DANE/TLSA verified servers!

Further Information

Deploy360 - DNSSEC

<http://www.isoc.org/deploy360/dnssec/>

Let's Encrypt

<https://letsencrypt.org/>

HOWTO

<http://www.internetsociety.org/deploy360/blog/2016/01/lets-encrypt-certificates-for-mail-servers-and-dane-part-1-of-2/>

<http://www.internetsociety.org/deploy360/blog/2016/03/lets-encrypt-certificates-for-mail-servers-and-dane-part-2-of-2/>

https://dane.sys4.de/common_mistakes

Thank You!

Kevin Meynell

Deploy360@isoc.org

