Certificates

Public Key Problems

- public-key crypto lets us secure communication
  - confidentiality, integrity, authenticity, non-repudiation
- but it requires that the public keys are authentic
  - you still need an authentic channel for that
- this is a hard practical problem
  - you've never met Google before logging into G-Mail, you just somehow got its key

Alice and Bob

- Alice and Bob both have their own public and private keys
- Alice and Bob have never met
- Alice needs Bob's public key to encrypt
  - she asks Bob over an insecure channel
  - she gets a public Key
  - what can go wrong?

Alice needs a way to validate the key without any bits being exchanged over an authentic channel.

You cannot bootstrap trust. It has to start somewhere.

You cannot bootstrap trust. It has to start somewhere.
With a kernel of trust you can exchange the key.

You cannot bootstrap trust. It has to start somewhere.
With a kernel of trust you can exchange the key.
With the key you can exchange everything else.

Solution 1: trust on first use

Solution 1: trust on first use
This concept is widely used and is called TOFU.

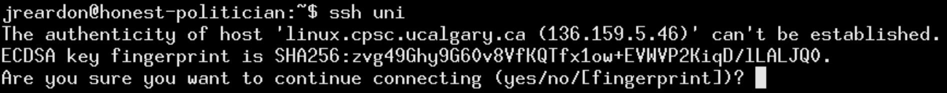
I get something that claims to be Bob's public key.

I get something that claims to be Bob's public key.
I assume it is and (importantly) **only trust that one**

I get something that claims to be Bob's public key.
I assume it is and (importantly) **only trust that one**
I am suspicious if it ever changes

I get something that claims to be Bob's public key.
I assume it is and (importantly) **only trust that one**
I am suspicious if it ever changes
I am safe unless I was being attacked **at that first time**.

I get something that claims to be Bob's public key.
I assume it is and (importantly) **only trust that one**
I am suspicious if it ever changes
I am safe unless I was being attacked **at that first time**.
I can always validate Bob's public key later if I meet Bob.



```
jreardon@honest-politician:~$ ssh uni
The authenticity of host 'linux.cpsc.ucalgary.ca (136.159.5.46)' can't be established.
ECDSA key fingerprint is SHA256:zvg49Ghy9G60v8VfKQTfx1ow+EVWVP2KiqD/lLALJQ0.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

Solution 2: centralized on-demand service

Solution 2: centralized on-demand service
Alice asks Service for Bob’s key (or to validate it)

Solution 2: centralized on-demand service
Alice asks Service for Bob’s key (or to validate it)
would this be practical?

Solution 2: centralized on-demand service
Alice asks Service for Bob's key (or to validate it)
would this be practical?
MITM attacks?

Solution 2: centralized on-demand service

Alice asks Service for Bob's key (or to validate it)

would this be practical?

MITM attacks?

Replay attacks?

Solution 2: centralized on-demand service

Alice asks Service for Bob's key (or to validate it)

would this be practical?

MITM attacks?

Replay attacks?

DoS attacks?

Solution 2: centralized on-demand service

Alice asks Service for Bob's key (or to validate it)

would this be practical?

MITM attacks?

Replay attacks?

DoS attacks?

Corrupt service?

Solution 2: centralized on-demand service

Alice asks Service for Bob's key (or to validate it)

would this be practical?

MITM attacks?

Replay attacks?

DoS attacks?

Corrupt service?

Does this remind you of anything?

Certificates

- a statement about a public key
    - “I certify that KEY XYZ belongs to Bob. Yours sincerely, Trent”
- Bob sends his public key to Trent over an authentic channel
- Trent prepares a document stating Bob owns the key
- Trent signs the document with Trent’s private key
- Trent appends this signature to the document and gives the result to Bob

Bob can show this to anyone without involving Trent!

Bob can show this to anyone without involving Trent!
Alice can verify this without asking Trent,
she only needs Trent's public key!

Bob can show this to anyone without involving Trent!
Alice can verify this without asking Trent,
she only needs Trent's public key!
Does this remind you of anything?

Protocol

- Alice has Trent's public key
- Alice contacts Bob
- Bob gives Alice the certificate signed by Trent
- Alice checks that the signature is valid using Trent's public key
- If Trent is honest, then that is Bob's public key

In practice, the document is called a **digital certificate** or a **cert** for short

In practice, the document is called a **digital certificate**
or a **cert** for short
Trent is called a **certificate authority**
or a **CA** for short

What can go wrong?

What can go wrong?
Alice doesn't have authentic key for Trent
(either bad in the first place, or changed)

What can go wrong?
Alice doesn’t have authentic key for Trent
(either bad in the first place, or changed)
Eve pretended to be Bob and Trent gives her a “Bob” cert

What can go wrong?

Alice doesn't have authentic key for Trent

(either bad in the first place, or changed)

Eve pretended to be Bob and Trent gives her a “Bob” cert

(it's one thing for everyone to know Trent, another for Trent to know everyone)

What can go wrong?
What if Trent is Eve?

What can go wrong?
What if Trent is Eve?
What if Eve breaks into Trent's computers?

How do you know if the whole certificate system works?
Also called Web PKI (public key infrastructure).

https://

That’s the user’s entire interaction with the security.

That’s the user’s entire interaction with the security.
It means encryption is securing the connection
and the website provided a valid certificate.

That's the user's entire interaction with the security.
It means encryption is securing the connection
and the website provided a valid certificate.
What does that mean though?

That's the user's entire interaction with the security.
It means encryption is securing the connection
and the website provided a valid certificate.
What does that mean though?
Did Trent meet Bob face to face?

That's the user's entire interaction with the security.
It means encryption is securing the connection
and the website provided a valid certificate.
What does that mean though?
Did Trent meet Bob face to face?
Who is Trent?

CA's Duties

- Bob claims that BOB.COM is his
- Bob wants to use PK as a public key for it
- What checks are required before issuing cert?

Alice's Duties

- Bob sends Alice: "Bob-signed(Trent-signed(cert))"
    - cert claims "BOB.COM's signing key is PK"
- Alice has to perform some checks on the certificate
    - what checks are needed before going and using PK?

Failing to check that cert is for who you expect!

Failing to check that cert is for who you expect!
Researchers discovered that poorly
designed APIs used in SSL implementations
failed to check the cert matched the sender.

Many critical non-browser software packages such as Amazon's EC2 Java library, Amazon's and PayPal's merchant SDKs, Trillian and AIM instant messaging software, popular integrated shopping cart software packages, Chase mobile banking software, and several Android applications and libraries.

SSL connections from these programs and many others are vulnerable to a man-in-the-middle attack

Failing to check the cert!

```
if ((err = SSLHashSHA1.update(&hashCtx, &clientRandom)) != 0)
        goto fail;
if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)
        goto fail;
if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
        goto fail;
        goto fail;
if ((err = SSLHashSHA1.final(&hashCtx, &hashOut)) != 0)
        goto fail;
err = sslRawVerify(...);

fail:   return err
```

Let's say that Bob's private key is stolen.

Let's say that Bob's private key is stolen.
What's the worst that can happen?

Let’s say that Bob’s private key is stolen.
What’s the worst that can happen?
How can we stop this?

Certification Revocation

- revoke means no longer trust this cert
  - keys can get stolen, or suspected stolen
  - also Bob changes companies
  - Bob wants to use a new key instead
- certs are just a signed statement
- how to remove trust once issued?

Cert Revocation List

- CRL are lists of bad certs.
- periodically given out to parties, e.g., weekly
- can be pushed to parties or posted to specific place

Certs Expire

- gives upper bound on use of stolen key
- keeps cert authorities with customers
- stops **revocation lists** from growing forever.

CRL Deltas

- instead of publishing the whole CRL, give updates (deltas)
- requires active involvement to keep up to date

Online Status Checking

- use an always online party to check if a cert is valid
    - outsource management of CRLs
        - typically to the CA issuing the cert
        - or a delegated provider
    - check done by Alice at the time of use
    - online certificate status protocol (OCSP)
- what does this cost?

What does OCSP provide that CRLs don't?

OCSP-stapling

- periodically get a signature from the online party with a timestamp
  - cert X is still not revoked at time Y
- check now done by Bob
  - OSCP's load substantially reduced for popular sites
- does this remind you of another protocol?

In what ways is OCSP-stapling better than OCSP?

Short-lived Certs

- make all certs only valid for a week
- exposure time is bounded to this low amount
- need to contact the CA to get new certs

Short-lived certs seem equivalent to CRL and OCSP-stapling but they differ in failure conditions. How?

Certs in Practice

- certs are used for TLS
  - transport layer security
  - this is the de facto means to secure web traffic
  - puts the S in HTTPS
    - S is for secure
  - topic of next lecture
- certs deliver a website's public key to a browser
  - authentic delivery of public key for Bob
  - creates authentic channel from Alice to Bob

Alice goes to bob.com and gets a cert for a public key that the owner of bob.com has the private key for.

For web, this is all done in the browser.

For web, this is all done in the browser.
The browser is responsible for checking if a cert is valid
by checking the fields, CRLs, etc.

https://

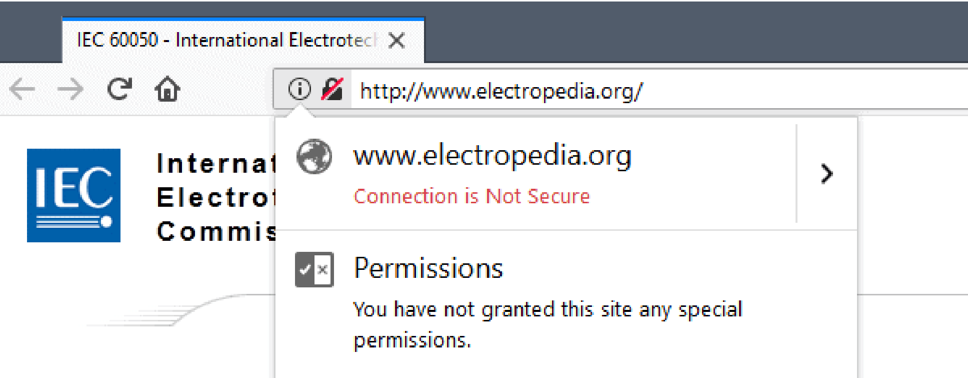
Since 2016, more web traffic is over HTTPS than HTTP

Since 2016, more web traffic is over HTTPS than HTTP

the lock has become more normal

browsers are now warning on insecure pages

Since 2016, more web traffic is over HTTPS than HTTP
the lock has become more normal
browsers are now warning on insecure pages
neutralization of HTTPS, instead of
making the “security” a positive



IEC 60050 - International Electrotec

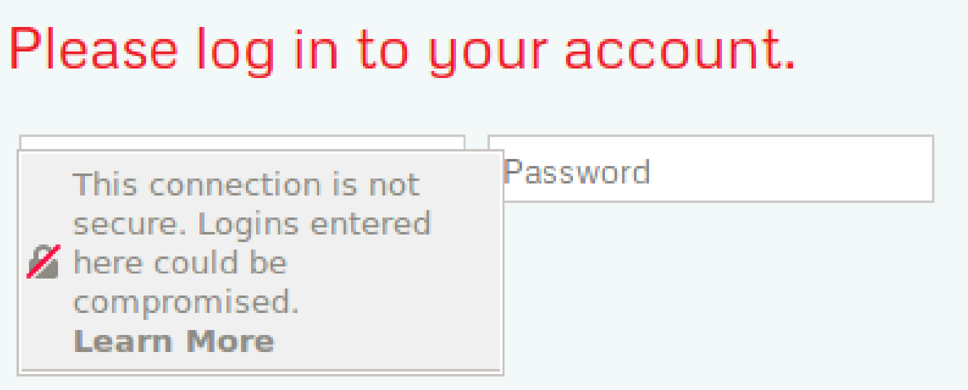
http://www.electropedia.org/

www.electropedia.org

Connection is Not Secure

Permissions

You have not granted this site any special permissions.



Please log in to your account.

This connection is not secure. Logins entered here could be compromised.

p2146r2-1.pdf

File not downloaded: Potential security risk. — open-std.org — 09:38

Three types of validation for certs.

Domain Validated (DV)

- Bob gives a public key to the CA and claims bob.com
- sends an email to admin@bob.com
  - a challenge, e.g., random number to sign with key
- purported owner proves control over the domain by
  - posting DNS TXT records to bob.com
  - putting some random number on bob.com/ca_challenge.html
- no proof that there's anyone named Bob related to it
  - could be a rogue employee with webmaster access
- can be fully automated

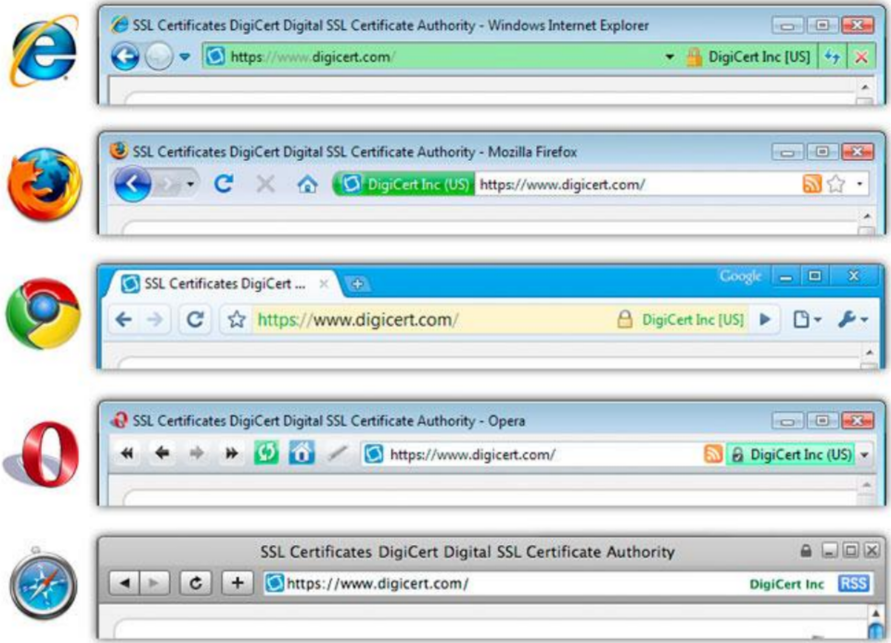
Organization Validated (OV)

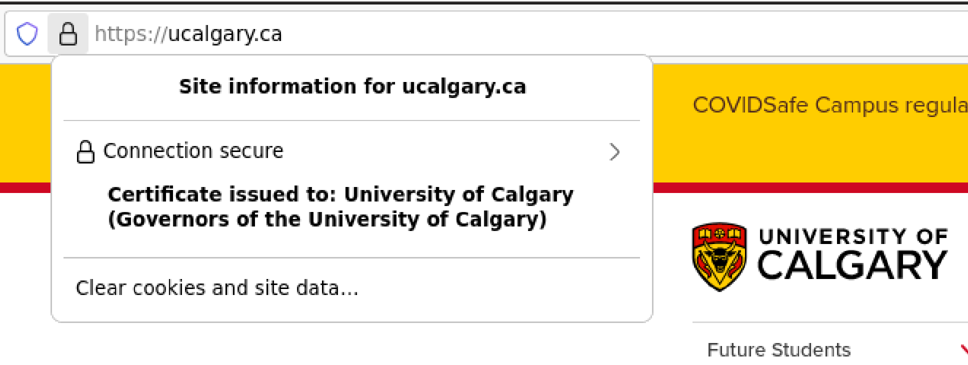
- also checks a business/organization behind the key
- e.g., look up business in a public directory and call them
- exact practice depends on the CA's certificate practice statement
- this extra information then is part of cert
    - but the user still only sees the lock icon

Extended Validation (EV)

- use of government database to confirm existence of legal entities named as Subject
- EV cert issuers are audited, have governance
    - certificate requests must be approved by a human lawyer
- motivated by low confidence DV certs that can be given to phishing websites
    - resulted in same visual experience as a legit site
        - e.g., lock icon, secure browser bar, etc.

Mozilla Corporation (US)
https
ing Started
{O
Cross Reference: /fr...
moz://a


SSL Certificates DigiCert Digital SSL Certificate Authority - Windows Internet Explorer
https://www.digicert.com/
DigiCert Inc [US]
SSL Certificates DigiCert Digital SSL Certificate Authority - Mozilla Firefox
DigiCert Inc (US)
https://www.digicert.com/
SSL Certificates DigiCert ...
Google
https://www.digicert.com/
DigiCert Inc [US]
SSL Certificates DigiCert Digital SSL Certificate Authority - Opera
https://www.digicert.com/
DigiCert Inc (US)
SSL Certificates DigiCert Digital SSL Certificate Authority
https://www.digicert.com/
DigiCert Inc
RSS



https://ucalgary.ca

Site information for ucalgary.ca

Connection secure

Certificate issued to: University of Calgary (Governors of the University of Calgary)

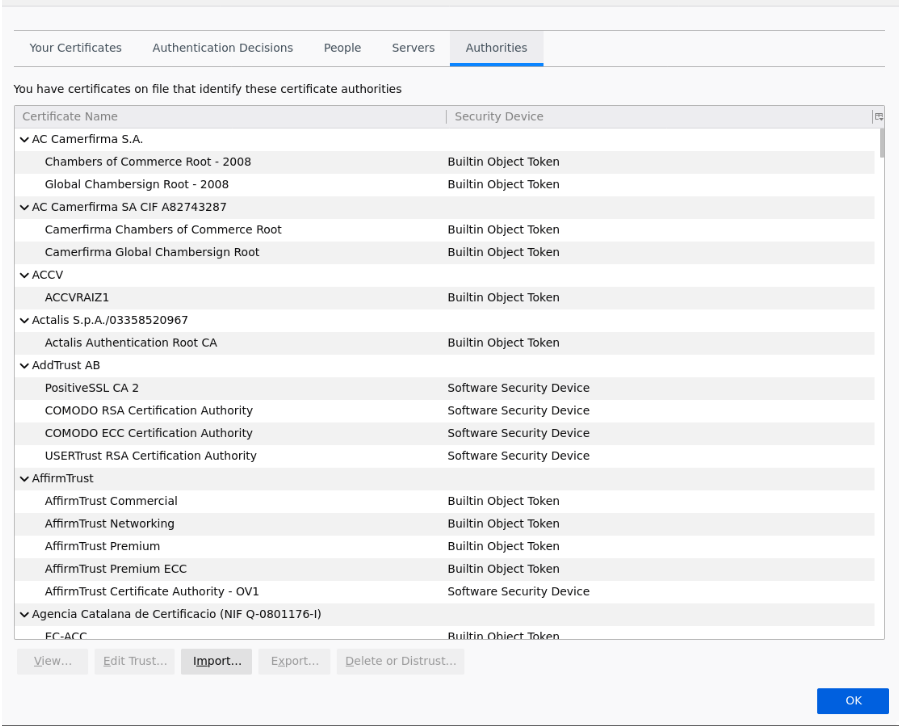
Extended Validation

- this has since stop
  - May 2018, Google removed it from Chrome
  - other browsers soon followed
- this seemed like a good idea, so why did it stop?

Extended Validation Drop Reason

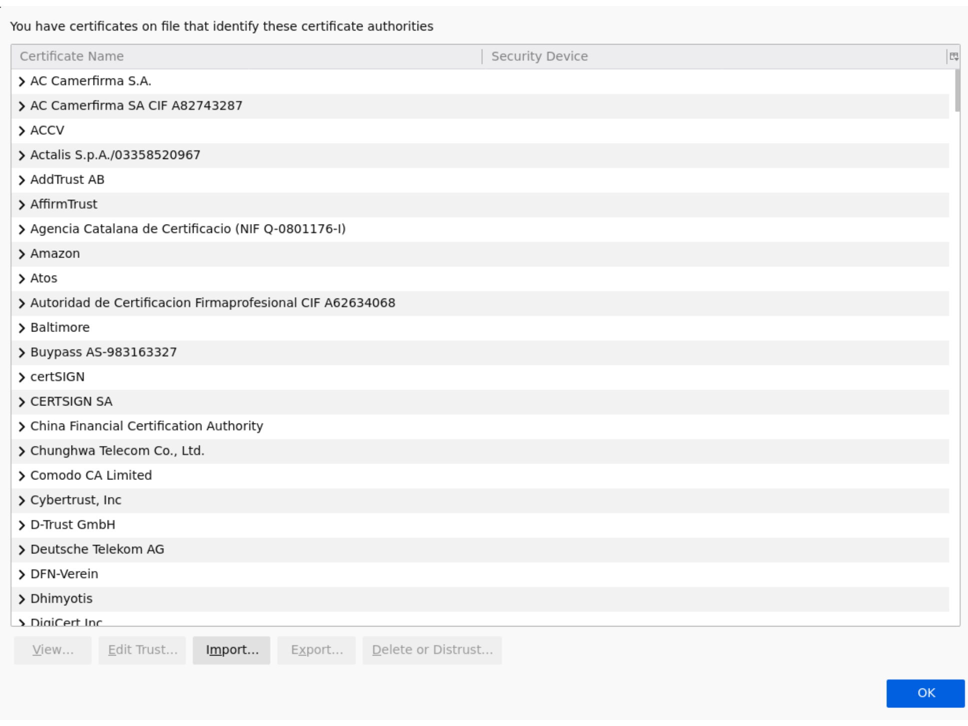
- user studies and A/B testing which showed they were ineffective
    - users do not appear to make secure choices (such as not entering password or credit card information) when the UI is altered or removed
- interfered with the bias towards neutralization of HTTPS
    - secure should be the norm
        - no special indicator
    - insecure is treated as hostile
- could be hacked with similar business names

How do we trust Certificate Authorities?



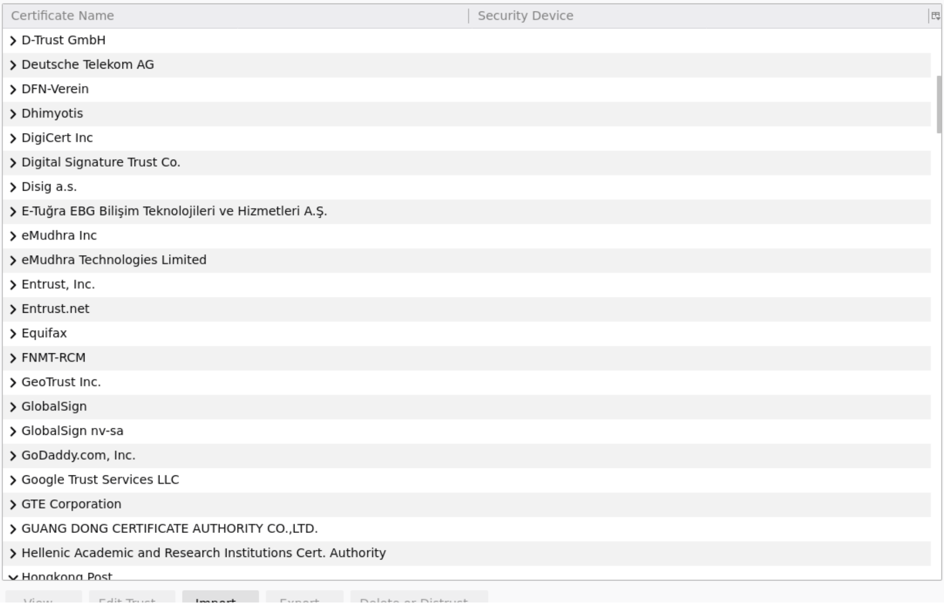
You have certificates on file that identify these certificate authorities

Certificate Name	Security Device
AC Camerfirma S.A.	
Chambers of Commerce Root - 2008	Builtin Object Token
Global Chambersign Root - 2008	Builtin Object Token
AC Camerfirma SA CIF A82743287	
Camerfirma Chambers of Commerce Root	Builtin Object Token
Camerfirma Global Chambersign Root	Builtin Object Token
ACCV	
ACCVRAIZ1	Builtin Object Token
Actalis S.p.A./03358520967	
Actalis Authentication Root CA	Builtin Object Token
AddTrust AB	
PositiveSSL CA 2	Software Security Device
COMODO RSA Certification Authority	Software Security Device
COMODO ECC Certification Authority	Software Security Device
USERTrust RSA Certification Authority	Software Security Device
AffirmTrust	
AffirmTrust Commercial	Builtin Object Token
AffirmTrust Networking	Builtin Object Token
AffirmTrust Premium	Builtin Object Token
AffirmTrust Premium ECC	Builtin Object Token
AffirmTrust Certificate Authority - OV1	Software Security Device
Agencia Catalana de Certificacio (NIF Q-0801176-I)	
EC-ACC	Builtin Object Token



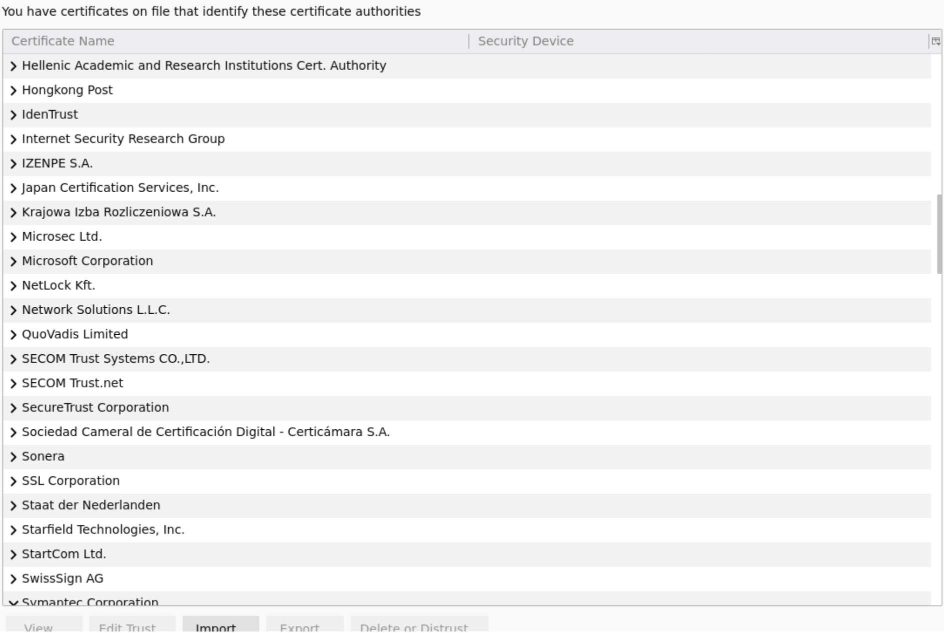
You have certificates on file that identify these certificate authorities

Certificate Name	Security Device
› AC Camerfirma S.A.	
› AC Camerfirma SA CIF A82743287	
› ACCV	
› Actalis S.p.A./03358520967	
› AddTrust AB	
› AffirmTrust	
› Agencia Catalana de Certificacio (NIF Q-0801176-I)	
› Amazon	
› Atos	
› Autoridad de Certificacion Firmaprofesional CIF A62634068	
› Baltimore	
› Buypass AS-983163327	
› certSIGN	
› CERTSIGN SA	
› China Financial Certification Authority	
› Chunghwa Telecom Co., Ltd.	
› Comodo CA Limited	
› Cybertrust, Inc	
› D-Trust GmbH	
› Deutsche Telekom AG	
› DFN-Verein	
› Dhimyotis	
› DigiCert Inc	



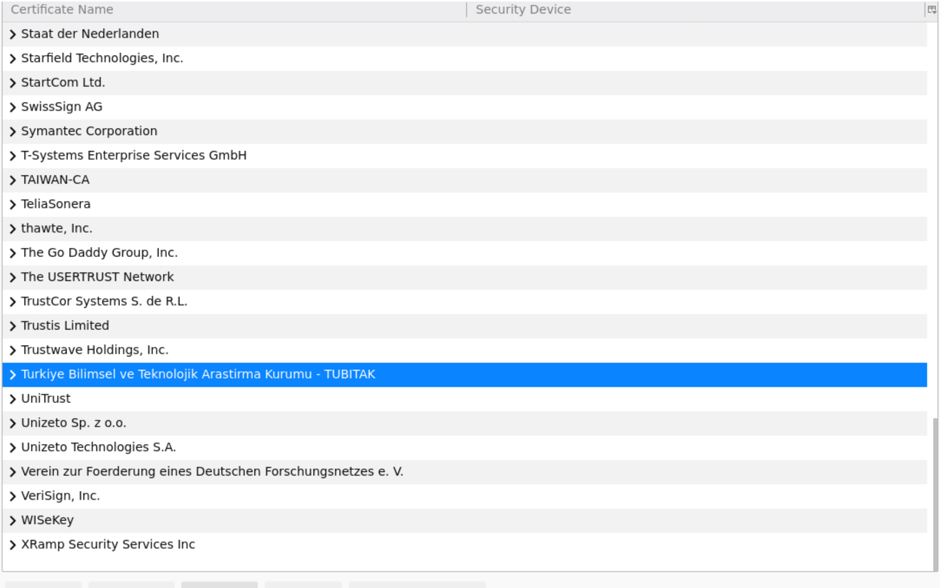
Certificate Name | Security Device

- D-Trust GmbH
- Deutsche Telekom AG
- DFN-Verein
- Dhimyotis
- DigiCert Inc
- Digital Signature Trust Co.
- Disig a.s.
- E-Tuğra EBG Bilişim Teknolojileri ve Hizmetleri A.Ş.
- eMudhra Inc
- eMudhra Technologies Limited
- Entrust, Inc.
- Entrust.net
- Equifax
- FNMT-RCM
- GeoTrust Inc.
- GlobalSign
- GlobalSign nv-sa
- GoDaddy.com, Inc.
- Google Trust Services LLC
- GTE Corporation
- GUANG DONG CERTIFICATE AUTHORITY CO.,LTD.
- Hellenic Academic and Research Institutions Cert. Authority
- Hongkong Post

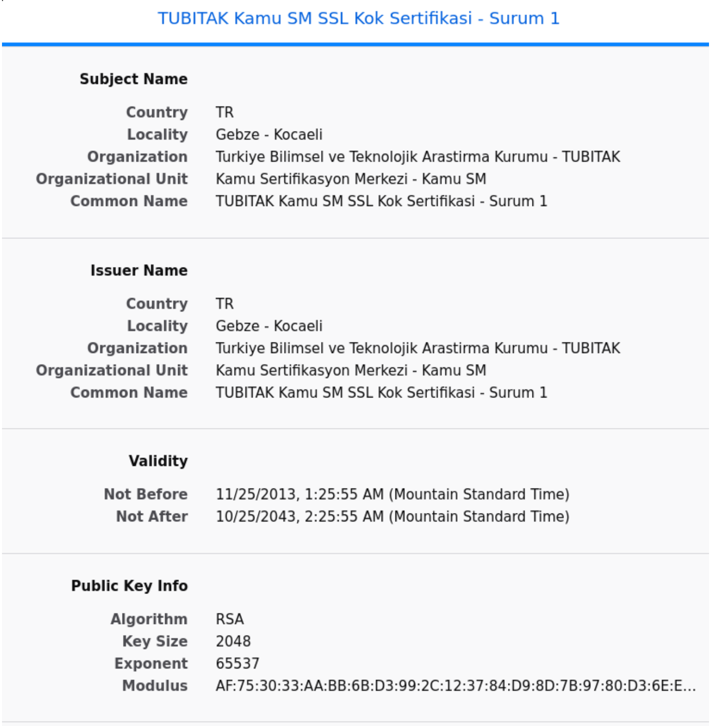


You have certificates on file that identify these certificate authorities

Certificate Name	Security Device
› Hellenic Academic and Research Institutions Cert. Authority	
› Hongkong Post	
› IdenTrust	
› Internet Security Research Group	
› IZENPE S.A.	
› Japan Certification Services, Inc.	
› Krajowa Izba Rozliczeniowa S.A.	
› Microsec Ltd.	
› Microsoft Corporation	
› NetLock Kft.	
› Network Solutions L.L.C.	
› QuoVadis Limited	
› SECOM Trust Systems CO.,LTD.	
› SECOM Trust.net	
› SecureTrust Corporation	
› Sociedad Cameral de Certificación Digital - Certicámara S.A.	
› Sonera	
› SSL Corporation	
› Staat der Nederlanden	
› Starfield Technologies, Inc.	
› StartCom Ltd.	
› SwissSign AG	
⌄ Symantec Corporation	



Certificate Name	Security Device
Staat der Nederlanden	
Starfield Technologies, Inc.	
StartCom Ltd.	
SwissSign AG	
Symantec Corporation	
T-Systems Enterprise Services GmbH	
TAIWAN-CA	
TeliaSonera	
thawte, Inc.	
The Go Daddy Group, Inc.	
The USERTRUST Network	
TrustCor Systems S. de R.L.	
Trustis Limited	
Trustwave Holdings, Inc.	
Turkiye Bilimsel ve Teknolojik Arastirma Kurumu - TUBITAK	
UniTrust	
Unizeto Sp. z o.o.	
Unizeto Technologies S.A.	
Verein zur Foerderung eines Deutschen Forschungsnetzes e. V.	
VeriSign, Inc.	
WISeKey	
XRamp Security Services Inc	



TUBITAK Kamu SM SSL Kok Sertifikasi - Surum 1

Subject Name

Country	TR
Locality	Gebze - Kocaeli
Organization	Turkiye Bilimsel ve Teknolojik Arastirma Kurumu - TUBITAK
Organizational Unit	Kamu Sertifikasyon Merkezi - Kamu SM
Common Name	TUBITAK Kamu SM SSL Kok Sertifikasi - Surum 1

Issuer Name

Country	TR
Locality	Gebze - Kocaeli
Organization	Turkiye Bilimsel ve Teknolojik Arastirma Kurumu - TUBITAK
Organizational Unit	Kamu Sertifikasyon Merkezi - Kamu SM
Common Name	TUBITAK Kamu SM SSL Kok Sertifikasi - Surum 1

Validity

Not Before	11/25/2013, 1:25:55 AM (Mountain Standard Time)
Not After	10/25/2043, 2:25:55 AM (Mountain Standard Time)

Public Key Info

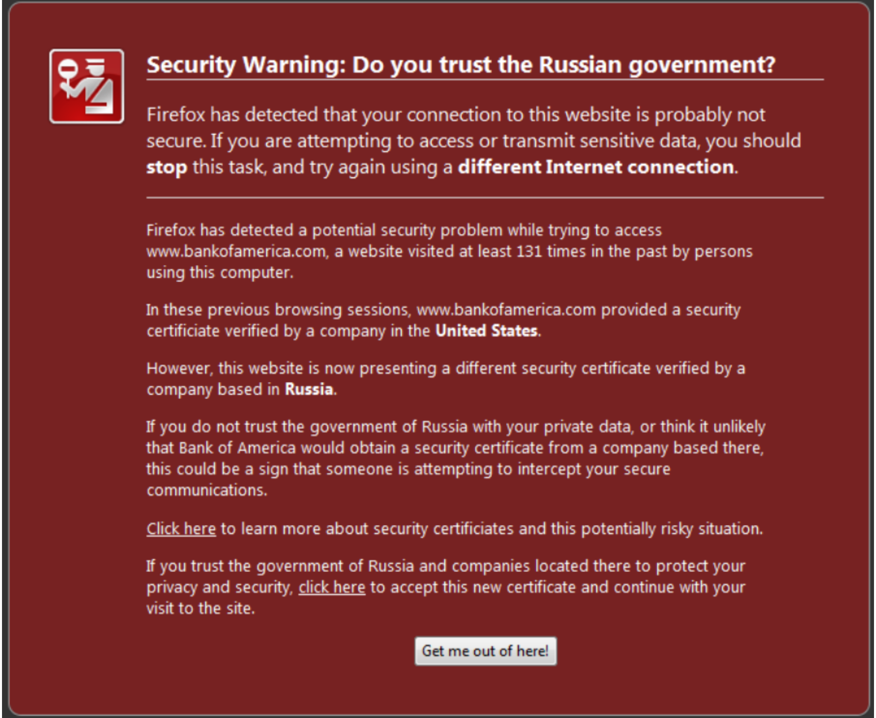
Algorithm	RSA
Key Size	2048
Exponent	65537
Modulus	AF:75:30:33:AA:BB:6B:D3:99:2C:12:37:84:D9:8D:7B:97:80:D3:6E:E...

Any certificate signed by any of these CAs
is accepted as completely valid

Any certificate signed by any of these CAs
is accepted as completely valid
i.e., gets the lock icon and no warnings.

Any certificate signed by any of these CAs
is accepted as completely valid
i.e., gets the lock icon and no warnings.
There's no scale or proportion of trust for these CAs.

Any certificate signed by any of these CAs
is accepted as completely valid
i.e., gets the lock icon and no warnings.
There's no scale or proportion of trust for these CAs.
What can go wrong?



Security Warning: Do you trust the Russian government?

Firefox has detected that your connection to this website is probably not secure. If you are attempting to access or transmit sensitive data, you should **stop** this task, and try again using a **different Internet connection**.

Firefox has detected a potential security problem while trying to access www.bankofamerica.com, a website visited at least 131 times in the past by persons using this computer.

In these previous browsing sessions, www.bankofamerica.com provided a security certificiate verified by a company in the **United States**.

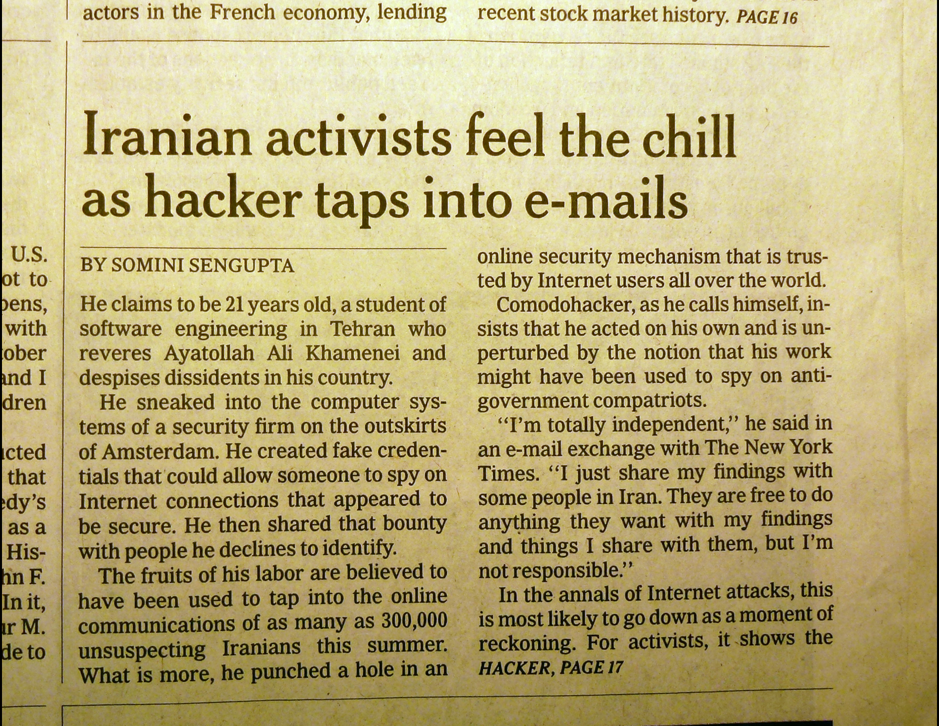
However, this website is now presenting a different security certificate verified by a company based in **Russia**.

If you do not trust the government of Russia with your private data, or think it unlikely that Bank of America would obtain a security certificate from a company based there, this could be a sign that someone is attempting to intercept your secure communications.

Click here to learn more about security certificiates and this potentially risky situation.

If you trust the government of Russia and companies located there to protect your privacy and security, click here to accept this new certificate and continue with your visit to the site.

The attacker who penetrated the Dutch CA DigiNotar last year had complete control of all eight of the company's certificate-issuing servers during the operation and he may also have issued some rogue certificates that have not yet been identified. The final report from a security company commissioned to investigate the DigiNotar attack shows that the compromise of the now-bankrupt certificate authority was much deeper than previously thought.



actors in the French economy, lending

recent stock market history. *PAGE 16*

U.S.
ot to
pens,
with
ober
nd I
dren

cted
that
dy's
as a
His-
hn F.
In it,
r M.
de to

Iranian activists feel the chill as hacker taps into e-mails

BY SOMINI SENGUPTA

He claims to be 21 years old, a student of software engineering in Tehran who reveres Ayatollah Ali Khamenei and despises dissidents in his country.

He sneaked into the computer systems of a security firm on the outskirts of Amsterdam. He created fake credentials that could allow someone to spy on Internet connections that appeared to be secure. He then shared that bounty with people he declines to identify.

The fruits of his labor are believed to have been used to tap into the online communications of as many as 300,000 unsuspecting Iranians this summer. What is more, he punched a hole in an online security mechanism that is trusted by Internet users all over the world.

Comodohacker, as he calls himself, insists that he acted on his own and is unperturbed by the notion that his work might have been used to spy on antigovernment compatriots.

"I'm totally independent," he said in an e-mail exchange with The New York Times. "I just share my findings with some people in Iran. They are free to do anything they want with my findings and things I share with them, but I'm not responsible."

In the annals of Internet attacks, this is most likely to go down as a moment of reckoning. For activists, it shows the

HACKER, PAGE 17

That enormous list of CAs are known as root CAs.

That enormous list of CAs are known as root CAs.
CAs sign certificates for other CAs.

That enormous list of CAs are known as root CAs.
CAs sign certificates for other CAs.
So Turktrust signs for someone you never heard of

That enormous list of CAs are known as root CAs.
CAs sign certificates for other CAs.
So Turktrust signs for someone you never heard of
who signs for someone else

That enormous list of CAs are known as root CAs.
CAs sign certificates for other CAs.
So Turktrust signs for someone you never heard of
who signs for someone else
who signs that some random public key
you've never seen before is Bob's key.

That enormous list of CAs are known as root CAs.
CAs sign certificates for other CAs.
So Turktrust signs for someone you never heard of
who signs for someone else
who signs that some random public key
you've never seen before is Bob's key.
And it gets the lock icon.

Demo

```
openssl s_client -showcerts -connect my.ucalgary.ca:443
```

TURKTRUST, a certificate authority in Mozilla’s root program, mis-issued two intermediate certificates to customers. TURKTRUST has scanned their certificate database and log files and confirmed that the mistake was made for only two certificates.

TURKTRUST, a certificate authority in Mozilla's root program, mis-issued two intermediate certificates to customers. TURKTRUST has scanned their certificate database and log files and confirmed that the mistake was made for only two certificates.

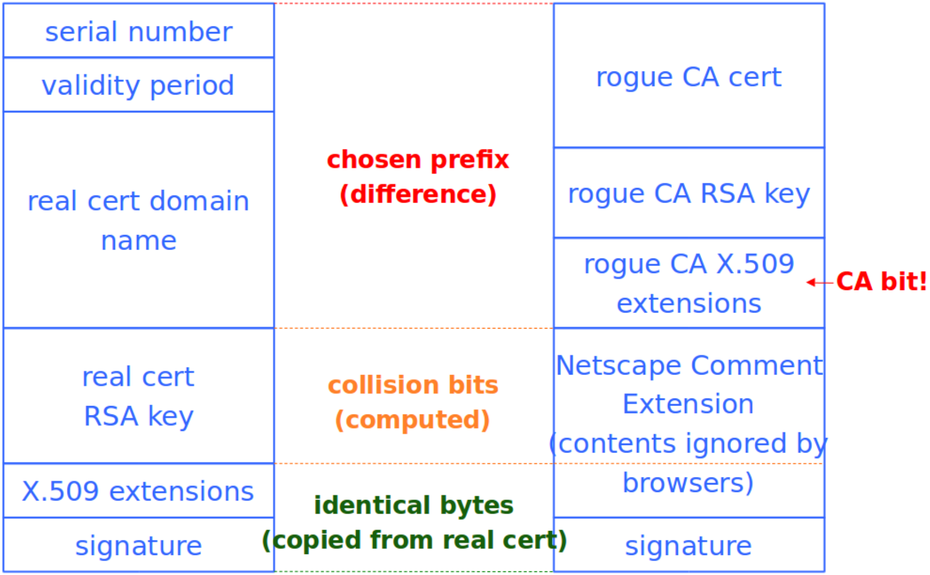
Mozilla is actively revoking trust for the two mis-issued certificates which will be released to all supported versions of Firefox in the next update.

TURKTRUST accidentally issued **intermediary** CA certs.

TURKTRUST accidentally issued **intermediary** CA certs.
Those are the ones in the middle,
and are **just as good** as the root.

TURKTRUST accidentally issued **intermediary** CA certs.
Those are the ones in the middle,
and are **just as good** as the root.
A CA can, with a signature,
turn anyone into a CA as well.

Also, the MD5 collision issue can be used to create intermediary CA certs!



serial number

validity period

real cert domain name

real cert RSA key

X.509 extensions

signature

chosen prefix (difference)

collision bits (computed)

identical bytes (copied from real cert)

rogue CA cert

rogue CA RSA key

rogue CA X.509 extensions

CA bit!

Netscape Comment Extension (contents ignored by browsers)

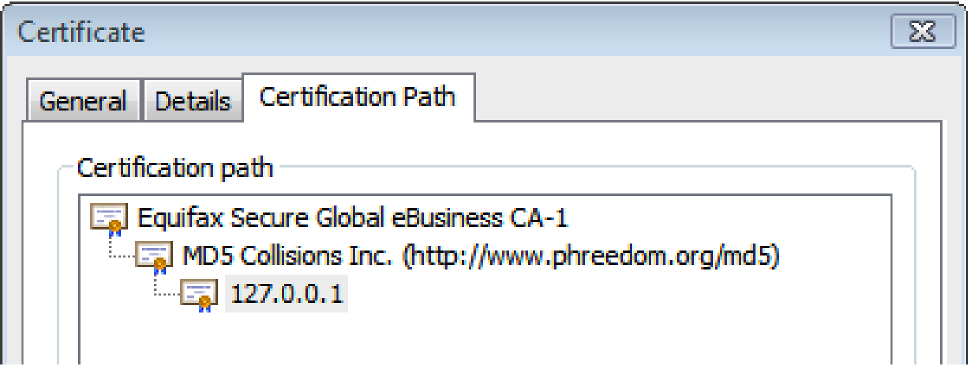
signature

This means that it is a master key!

This means that it is a master key!
A network attacker to easily forge fake
certificates for any website!

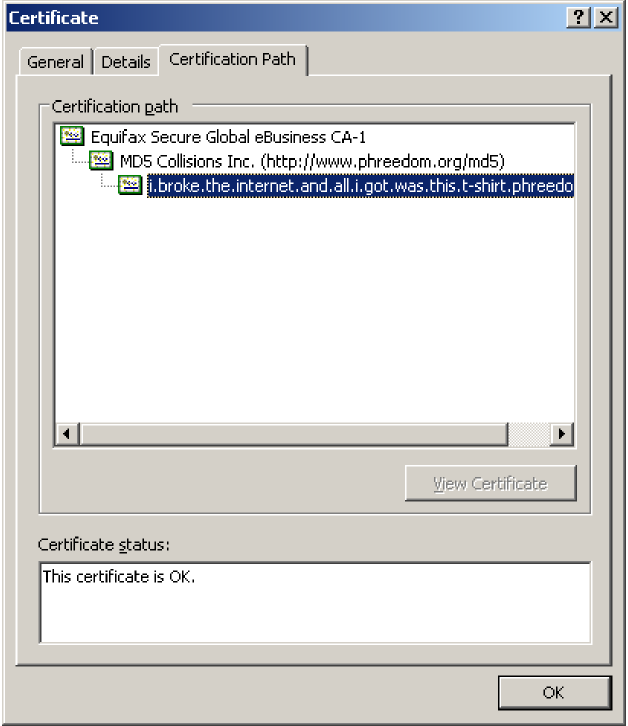
This means that it is a master key!
A network attacker to easily forge fake certificates for any website!
Users will get wrong public key and not have any indication something is wrong.

https://



Certification path

- Equifax Secure Global eBusiness CA-1
  - MD5 Collisions Inc. (http://www.phreedom.org/md5)
    - 127.0.0.1



Certification path

Equifax Secure Global eBusiness CA-1

MD5 Collisions Inc. (http://www.phreedom.org/md5)

i.broke.the.internet.and.all.i.got.was.this.t-shirt.phreedo

Certificate status:

This certificate is OK.

The security of HTTPS is only as strong as the practices of the least trustworthy/competent CA.

The security of HTTPS is only as strong as the practices of the least trustworthy/competent CA.

WEAKEST LINK

Fake certs is probably the most practical
way to break Internet security but...

Fake certs is probably the most practical
way to break Internet security but...
it is clear if the attack gets done.

Fake certs is probably the most practical
way to break Internet security but...
it is clear if the attack gets done.
Public key signatures provided non-repudiability
so if I sign a bad cert I can't undo it.

Fake certs is probably the most practical
way to break Internet security but...
it is clear if the attack gets done.
Public key signatures provided non-repudiability
so if I sign a bad cert I can't undo it.
If I'm the kind of CA that gives out bad
certs then I'll stop being in the CA club.

Certificate Transparency (CT)

After DigiNotar, Google employees wanted to create an open source framework for detecting mis-issued certificates.

After DigiNotar, Google employees wanted to create an open source framework for detecting mis-issued certificates.

idea: log all new certificates from a CA

System was voluntary at first.

System was voluntary at first.
In 2015, Chrome required CT
logging for all new EV certs

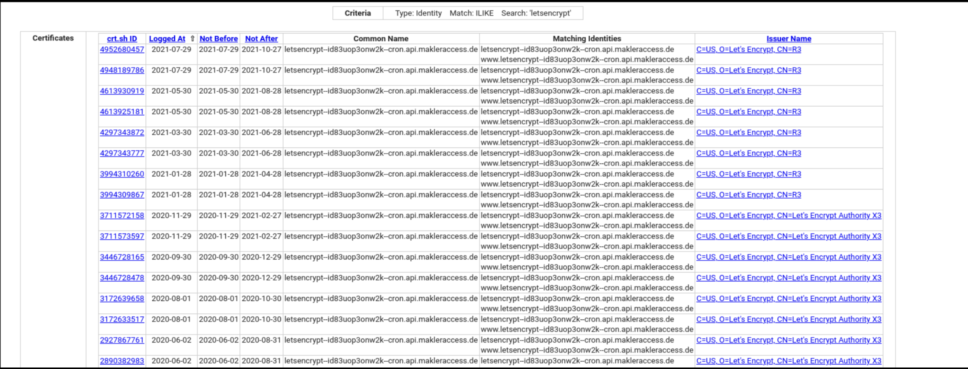
System was voluntary at first.
In 2015, Chrome required CT
logging for all new EV certs
i.e., would reject cert if
it did not appear in logs.

System was voluntary at first.
In 2015, Chrome required CT
logging for all new EV certs
i.e., would reject cert if
it did not appear in logs.
In 2016, required CT for all
certs from Symantec (Norton)

System was voluntary at first.
In 2015, Chrome required CT
logging for all new EV certs
i.e., would reject cert if
it did not appear in logs.
In 2016, required CT for all
certs from Symantec (Norton)
(they had issued 187 certificates
without the domain owner's knowlege)

System was voluntary at first.
In 2015, Chrome required CT
logging for all new EV certs
i.e., would reject cert if
it did not appear in logs.
In 2016, required CT for all
certs from Symantec (Norton)
(they had issued 187 certificates
without the domain owner's knowlege)
In 2018, all certs.

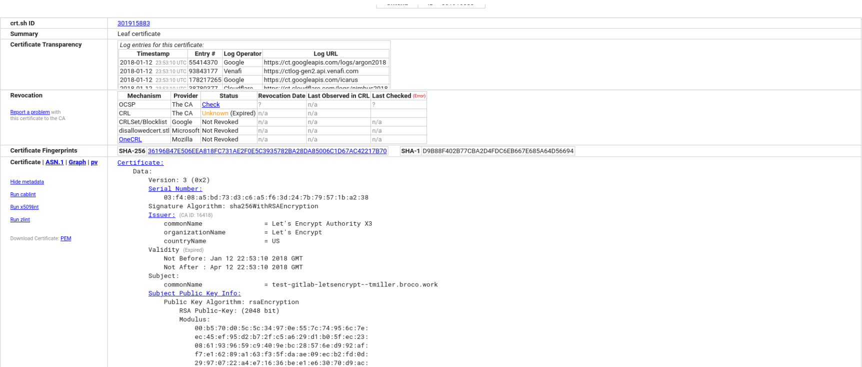
View the logs: https://crt.sh/



Criteria	Type: Identity Match: ILIKE Search: 'letsencrypt'

Certificates

crt.sh ID	Logged At ⇧	Not Before	Not After	Common Name	Matching Identities	Issuer Name
4952680457	2021-07-29	2021-07-29	2021-10-27	letsencrypt--id83uop3onw2k--cron.api.makleraccess.de	letsencrypt--id83uop3onw2k--cron.api.makleraccess.de www.letsencrypt--id83uop3onw2k--cron.api.makleraccess.de	C=US, O=Let's Encrypt, CN=R3
4948189786	2021-07-29	2021-07-29	2021-10-27	letsencrypt--id83uop3onw2k--cron.api.makleraccess.de	letsencrypt--id83uop3onw2k--cron.api.makleraccess.de www.letsencrypt--id83uop3onw2k--cron.api.makleraccess.de	C=US, O=Let's Encrypt, CN=R3
4613930919	2021-05-30	2021-05-30	2021-08-28	letsencrypt--id83uop3onw2k--cron.api.makleraccess.de	letsencrypt--id83uop3onw2k--cron.api.makleraccess.de www.letsencrypt--id83uop3onw2k--cron.api.makleraccess.de	C=US, O=Let's Encrypt, CN=R3
4613925181	2021-05-30	2021-05-30	2021-08-28	letsencrypt--id83uop3onw2k--cron.api.makleraccess.de	letsencrypt--id83uop3onw2k--cron.api.makleraccess.de www.letsencrypt--id83uop3onw2k--cron.api.makleraccess.de	C=US, O=Let's Encrypt, CN=R3
4297343872	2021-03-30	2021-03-30	2021-06-28	letsencrypt--id83uop3onw2k--cron.api.makleraccess.de	letsencrypt--id83uop3onw2k--cron.api.makleraccess.de www.letsencrypt--id83uop3onw2k--cron.api.makleraccess.de	C=US, O=Let's Encrypt, CN=R3
4297343777	2021-03-30	2021-03-30	2021-06-28	letsencrypt--id83uop3onw2k--cron.api.makleraccess.de	letsencrypt--id83uop3onw2k--cron.api.makleraccess.de www.letsencrypt--id83uop3onw2k--cron.api.makleraccess.de	C=US, O=Let's Encrypt, CN=R3
3994310260	2021-01-28	2021-01-28	2021-04-28	letsencrypt--id83uop3onw2k--cron.api.makleraccess.de	letsencrypt--id83uop3onw2k--cron.api.makleraccess.de www.letsencrypt--id83uop3onw2k--cron.api.makleraccess.de	C=US, O=Let's Encrypt, CN=R3
3994309867	2021-01-28	2021-01-28	2021-04-28	letsencrypt--id83uop3onw2k--cron.api.makleraccess.de	letsencrypt--id83uop3onw2k--cron.api.makleraccess.de www.letsencrypt--id83uop3onw2k--cron.api.makleraccess.de	C=US, O=Let's Encrypt, CN=R3
3711572158	2020-11-29	2020-11-29	2021-02-27	letsencrypt--id83uop3onw2k--cron.api.makleraccess.de	letsencrypt--id83uop3onw2k--cron.api.makleraccess.de www.letsencrypt--id83uop3onw2k--cron.api.makleraccess.de	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
3711573597	2020-11-29	2020-11-29	2021-02-27	letsencrypt--id83uop3onw2k--cron.api.makleraccess.de	letsencrypt--id83uop3onw2k--cron.api.makleraccess.de www.letsencrypt--id83uop3onw2k--cron.api.makleraccess.de	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
3446728165	2020-09-30	2020-09-30	2020-12-29	letsencrypt--id83uop3onw2k--cron.api.makleraccess.de	letsencrypt--id83uop3onw2k--cron.api.makleraccess.de www.letsencrypt--id83uop3onw2k--cron.api.makleraccess.de	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
3446728478	2020-09-30	2020-09-30	2020-12-29	letsencrypt--id83uop3onw2k--cron.api.makleraccess.de	letsencrypt--id83uop3onw2k--cron.api.makleraccess.de www.letsencrypt--id83uop3onw2k--cron.api.makleraccess.de	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
3172639658	2020-08-01	2020-08-01	2020-10-30	letsencrypt--id83uop3onw2k--cron.api.makleraccess.de	letsencrypt--id83uop3onw2k--cron.api.makleraccess.de www.letsencrypt--id83uop3onw2k--cron.api.makleraccess.de	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
3172633517	2020-08-01	2020-08-01	2020-10-30	letsencrypt--id83uop3onw2k--cron.api.makleraccess.de	letsencrypt--id83uop3onw2k--cron.api.makleraccess.de www.letsencrypt--id83uop3onw2k--cron.api.makleraccess.de	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
2927867761	2020-06-02	2020-06-02	2020-08-31	letsencrypt--id83uop3onw2k--cron.api.makleraccess.de	letsencrypt--id83uop3onw2k--cron.api.makleraccess.de www.letsencrypt--id83uop3onw2k--cron.api.makleraccess.de	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
2890382983	2020-06-02	2020-06-02	2020-08-31	letsencrypt--id83uop3onw2k--cron.api.makleraccess.de	letsencrypt--id83uop3onw2k--cron.api.makleraccess.de	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3



crt.sh ID 301915883

Summary Leaf certificate

Certificate Transparency

Log entries for this certificate:

Timestamp	Entry #	Log Operator	Log URL
2018-01-12 23:53:10 UTC	55414370	Google	https://ct.googleapis.com/logs/argon2018
2018-01-12 23:53:10 UTC	93843177	Venafi	https://ctlog-gen2.api.venafi.com
2018-01-12 23:53:10 UTC	178217265	Google	https://ct.googleapis.com/icarus
2018-01-12 23:53:10 UTC	38780377	Cloudflare	https://ct.cloudflare.com/logs/nimbus2018

Revocation

Report a problem with this certificate to the CA

Mechanism	Provider	Status	Revocation Date	Last Observed in CRL	Last Checked (Error)
OCSP	The CA	Check	?	n/a	?
CRL	The CA	Unknown (Expired)	n/a	n/a	
CRLSet/Blocklist	Google	Not Revoked	n/a	n/a	n/a
disallowedcert.stl	Microsoft	Not Revoked	n/a	n/a	n/a
OneCRL	Mozilla	Not Revoked	n/a	n/a	n/a

Certificate Fingerprints

SHA-256	36196B47E506EEA818FC731AE2F0E5C3935782BA28DA85006C1D67AC42217B70	SHA-1	D9B88F402B77CBA2D4FDC6EB667E685A64D56694

Certificate | **ASN.1** | **Graph** | **pv**

Hide metadata

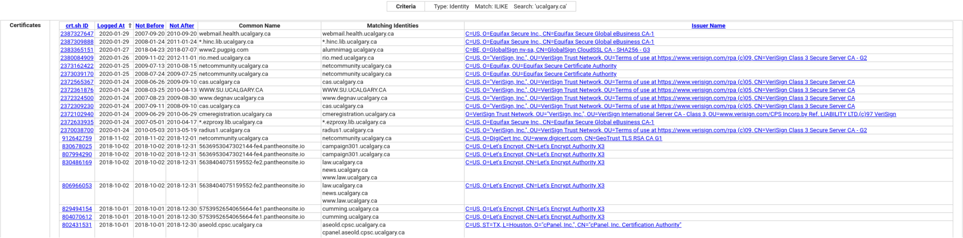
Run cablint

Run x509lint

Run zlint

Download Certificate: PEM

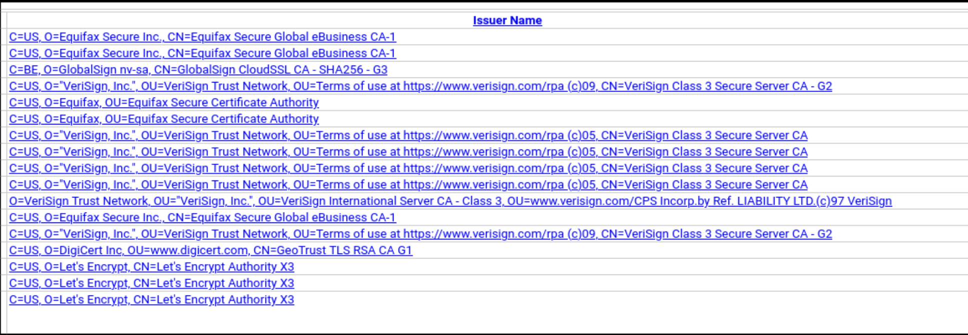
```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            03:f4:08:a5:bd:73:d3:c6:a5:f6:3d:24:7b:79:57:1b:a2:38
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: (CA ID: 16418)
            commonName                = Let's Encrypt Authority X3
            organizationName          = Let's Encrypt
            countryName               = US
        Validity (Expired)
            Not Before: Jan 12 22:53:10 2018 GMT
            Not After : Apr 12 22:53:10 2018 GMT
        Subject:
            commonName                = test-gitlab-letsencrypt--tmiller.broco.work
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                RSA Public-Key: (2048 bit)
                Modulus:
                    00:b5:70:d0:5c:5c:34:97:0e:55:7c:74:95:6c:7e:
                    ec:45:ef:95:d2:b7:2f:c5:a6:29:d1:b0:5f:ec:23:
                    08:61:93:96:59:c9:40:9e:bc:28:57:6e:d9:92:af:
                    f7:e1:62:89:a1:63:f3:5f:da:ae:09:ec:b2:fd:0d:
                    29:97:07:22:a4:e7:16:36:be:e1:e6:30:70:d9:ac:
```



Criteria	Type: Identity Match: ILIKE Search: 'ucalgary.ca'

Certificates

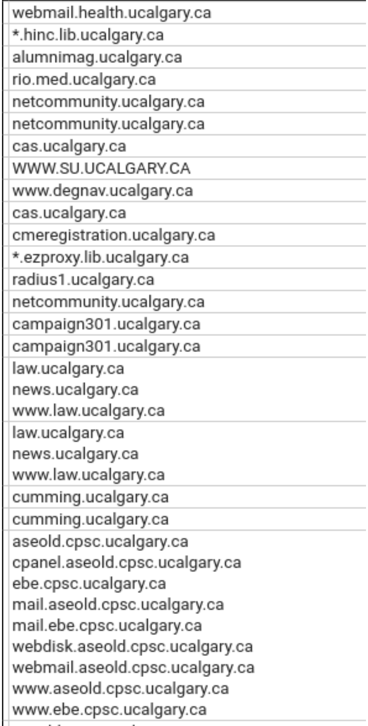
crt.sh ID	Logged At ⇧	Not Before	Not After	Common Name	Matching Identities	Issuer Name
2387327647	2020-01-29	2007-09-20	2010-09-20	webmail.health.ucalgary.ca	webmail.health.ucalgary.ca	C=US, O=Equifax Secure Inc., CN=Equifax Secure Global eBusiness CA-1
2387309888	2020-01-29	2008-01-24	2011-01-24	*.hinc.lib.ucalgary.ca	*.hinc.lib.ucalgary.ca	C=US, O=Equifax Secure Inc., CN=Equifax Secure Global eBusiness CA-1
2383365151	2020-01-27	2018-04-23	2018-07-07	www2.pugpig.com	alumnimag.ucalgary.ca	C=BE, O=GlobalSign nv-sa, CN=GlobalSign CloudSSL CA - SHA256 - G3
2380084909	2020-01-26	2009-11-02	2012-11-01	rio.med.ucalgary.ca	rio.med.ucalgary.ca	C=US, O="VeriSign, Inc.", OU=VeriSign Trust Network, OU=Terms of use at https://www.verisign.com/rpa (c)09, CN=VeriSign Class 3 Secure Server CA - G2
2373162422	2020-01-25	2009-07-13	2010-08-15	netcommunity.ucalgary.ca	netcommunity.ucalgary.ca	C=US, O=Equifax, OU=Equifax Secure Certificate Authority
2373039170	2020-01-25	2008-07-24	2009-07-25	netcommunity.ucalgary.ca	netcommunity.ucalgary.ca	C=US, O=Equifax, OU=Equifax Secure Certificate Authority
2372565367	2020-01-24	2008-06-26	2009-09-10	cas.ucalgary.ca	cas.ucalgary.ca	C=US, O="VeriSign, Inc.", OU=VeriSign Trust Network, OU=Terms of use at https://www.verisign.com/rpa (c)05, CN=VeriSign Class 3 Secure Server CA
2372361876	2020-01-24	2008-03-25	2010-04-13	WWW.SU.UCALGARY.CA	WWW.SU.UCALGARY.CA	C=US, O="VeriSign, Inc.", OU=VeriSign Trust Network, OU=Terms of use at https://www.verisign.com/rpa (c)05, CN=VeriSign Class 3 Secure Server CA
2372324500	2020-01-24	2007-08-23	2009-08-30	www.degnav.ucalgary.ca	www.degnav.ucalgary.ca	C=US, O="VeriSign, Inc.", OU=VeriSign Trust Network, OU=Terms of use at https://www.verisign.com/rpa (c)05, CN=VeriSign Class 3 Secure Server CA
2372309230	2020-01-24	2007-09-11	2008-09-10	cas.ucalgary.ca	cas.ucalgary.ca	C=US, O="VeriSign, Inc.", OU=VeriSign Trust Network, OU=Terms of use at https://www.verisign.com/rpa (c)05, CN=VeriSign Class 3 Secure Server CA
2372102940	2020-01-24	2009-06-29	2010-06-29	cmeregistration.ucalgary.ca	cmeregistration.ucalgary.ca	O=VeriSign Trust Network, OU="VeriSign, Inc.", OU=VeriSign International Server CA - Class 3, OU=www.verisign.com/CPS Incorp.by Ref. LIABILITY LTD.(c)97 VeriSign
2372633935	2020-01-24	2007-05-01	2010-04-17	*.ezproxy.lib.ucalgary.ca	*.ezproxy.lib.ucalgary.ca	C=US, O=Equifax Secure Inc., CN=Equifax Secure Global eBusiness CA-1
2370038700	2020-01-24	2010-05-03	2013-05-19	radius1.ucalgary.ca	radius1.ucalgary.ca	C=US, O="VeriSign, Inc.", OU=VeriSign Trust Network, OU=Terms of use at https://www.verisign.com/rpa (c)09, CN=VeriSign Class 3 Secure Server CA - G2
912642759	2018-11-02	2018-11-02	2018-12-01	netcommunity.ucalgary.ca	netcommunity.ucalgary.ca	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=GeoTrust TLS RSA CA G1
830678025	2018-10-02	2018-10-02	2018-12-31	5636953047302144-fe4.pantheonsite.io	campaign301.ucalgary.ca	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
807994290	2018-10-02	2018-10-02	2018-12-31	5636953047302144-fe4.pantheonsite.io	campaign301.ucalgary.ca	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
830486169	2018-10-02	2018-10-02	2018-12-31	5638404075159552-fe2.pantheonsite.io	law.ucalgary.ca news.ucalgary.ca www.law.ucalgary.ca	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
806966053	2018-10-02	2018-10-02	2018-12-31	5638404075159552-fe2.pantheonsite.io	law.ucalgary.ca news.ucalgary.ca www.law.ucalgary.ca	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
829494154	2018-10-01	2018-10-01	2018-12-30	5753952654065664-fe1.pantheonsite.io	cumming.ucalgary.ca	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
804070612	2018-10-01	2018-10-01	2018-12-30	5753952654065664-fe1.pantheonsite.io	cumming.ucalgary.ca	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
802431531	2018-10-01	2018-10-01	2018-12-30	aseold.cpsc.ucalgary.ca	aseold.cpsc.ucalgary.ca cpanel.aseold.cpsc.ucalgary.ca	C=US, ST=TX, L=Houston, O="cPanel, Inc.", CN="cPanel, Inc. Certification Authority"



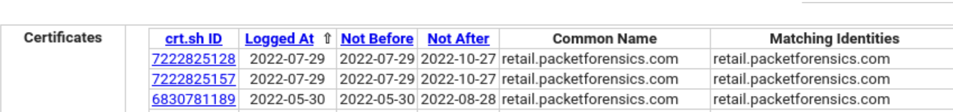
Issuer Name
C=US, O=Equifax Secure Inc., CN=Equifax Secure Global eBusiness CA-1
C=US, O=Equifax Secure Inc., CN=Equifax Secure Global eBusiness CA-1
C=BE, O=GlobalSign nv-sa, CN=GlobalSign CloudSSL CA - SHA256 - G3
C=US, O="VeriSign, Inc.", OU=VeriSign Trust Network, OU=Terms of use at https://www.verisign.com/rpa (c)09, CN=VeriSign Class 3 Secure Server CA - G2
C=US, O=Equifax, OU=Equifax Secure Certificate Authority
C=US, O=Equifax, OU=Equifax Secure Certificate Authority
C=US, O="VeriSign, Inc.", OU=VeriSign Trust Network, OU=Terms of use at https://www.verisign.com/rpa (c)05, CN=VeriSign Class 3 Secure Server CA
C=US, O="VeriSign, Inc.", OU=VeriSign Trust Network, OU=Terms of use at https://www.verisign.com/rpa (c)05, CN=VeriSign Class 3 Secure Server CA
C=US, O="VeriSign, Inc.", OU=VeriSign Trust Network, OU=Terms of use at https://www.verisign.com/rpa (c)05, CN=VeriSign Class 3 Secure Server CA
C=US, O="VeriSign, Inc.", OU=VeriSign Trust Network, OU=Terms of use at https://www.verisign.com/rpa (c)05, CN=VeriSign Class 3 Secure Server CA
O=VeriSign Trust Network, OU="VeriSign, Inc.", OU=VeriSign International Server CA - Class 3, OU=www.verisign.com/CPS Incorp.by Ref. LIABILITY LTD.(c)97 VeriSign
C=US, O=Equifax Secure Inc., CN=Equifax Secure Global eBusiness CA-1
C=US, O="VeriSign, Inc.", OU=VeriSign Trust Network, OU=Terms of use at https://www.verisign.com/rpa (c)09, CN=VeriSign Class 3 Secure Server CA - G2
C=US, O=DigiCert Inc, OU=www.digicert.com, CN=GeoTrust TLS RSA CA G1
C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3

Baseline	
Path	**Ratio**
/	97.79%
favicon.ico	28.55%
robots.txt	4.92%
login.php	0.84%
wp-login.php	0.69%
.git/HEAD	0.47%

- **Malicious CT Bots:** Our results show that when one creates a website, they must ensure that all security best practices are applied prior to creating TLS certificates. Once a domain appears on CT logs, admins should expect to receive numerous requests to their sites within minutes of certificate creation, from potentially malicious web bots. This is especially true for `Sensitive` domains that indicate the site could be a vulnerable web application, which are likely to receive tens of probes ranging from fingerprinting attempts to unsolicited POST requests. In total, we observe 105 malicious web-request campaigns targeting our measurement nodes. Furthermore, we find hundreds of unique IP addresses that extend their probes beyond web servers, attempting to authenticate with exposed network services such as SSH.

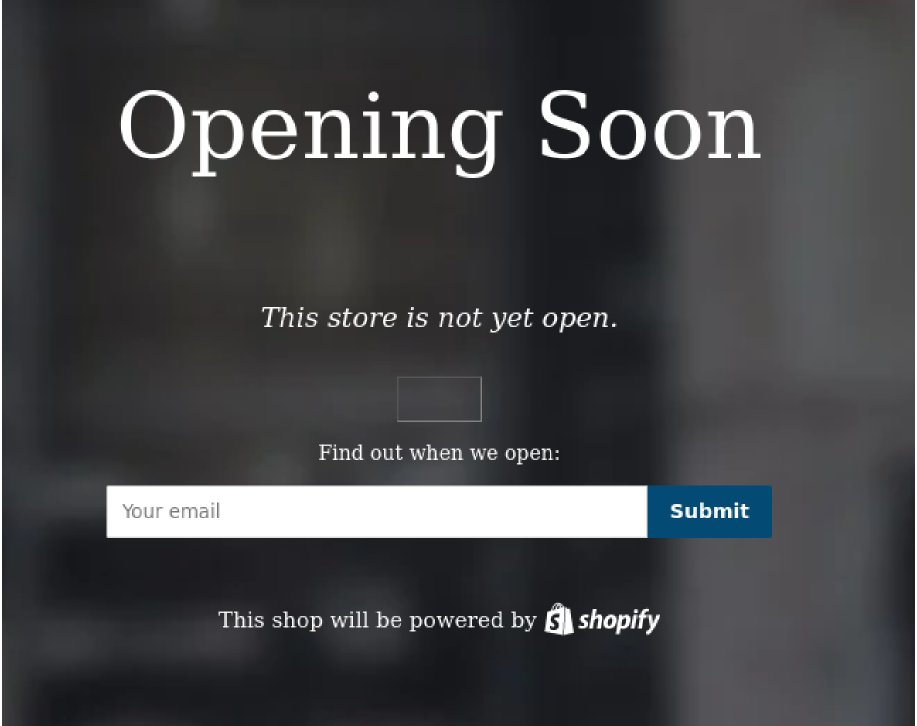


webmail.health.ucalgary.ca
*.hinc.lib.ucalgary.ca
alumnimag.ucalgary.ca
rio.med.ucalgary.ca
netcommunity.ucalgary.ca
netcommunity.ucalgary.ca
cas.ucalgary.ca
WWW.SU.UCALGARY.CA
www.degnav.ucalgary.ca
cas.ucalgary.ca
cmeregistration.ucalgary.ca
*.ezproxy.lib.ucalgary.ca
radius1.ucalgary.ca
netcommunity.ucalgary.ca
campaign301.ucalgary.ca
campaign301.ucalgary.ca
law.ucalgary.ca news.ucalgary.ca www.law.ucalgary.ca
law.ucalgary.ca news.ucalgary.ca www.law.ucalgary.ca
cumming.ucalgary.ca
cumming.ucalgary.ca
aseold.cpsc.ucalgary.ca cpanel.aseold.cpsc.ucalgary.ca ebe.cpsc.ucalgary.ca mail.aseold.cpsc.ucalgary.ca mail.ebe.cpsc.ucalgary.ca webdisk.aseold.cpsc.ucalgary.ca webmail.aseold.cpsc.ucalgary.ca www.aseold.cpsc.ucalgary.ca www.ebe.cpsc.ucalgary.ca



Certificates

crt.sh ID	Logged At ⇧	Not Before	Not After	Common Name	Matching Identities
7222825128	2022-07-29	2022-07-29	2022-10-27	retail.packetforensics.com	retail.packetforensics.com
7222825157	2022-07-29	2022-07-29	2022-10-27	retail.packetforensics.com	retail.packetforensics.com
6830781189	2022-05-30	2022-05-30	2022-08-28	retail.packetforensics.com	retail.packetforensics.com



Opening Soon

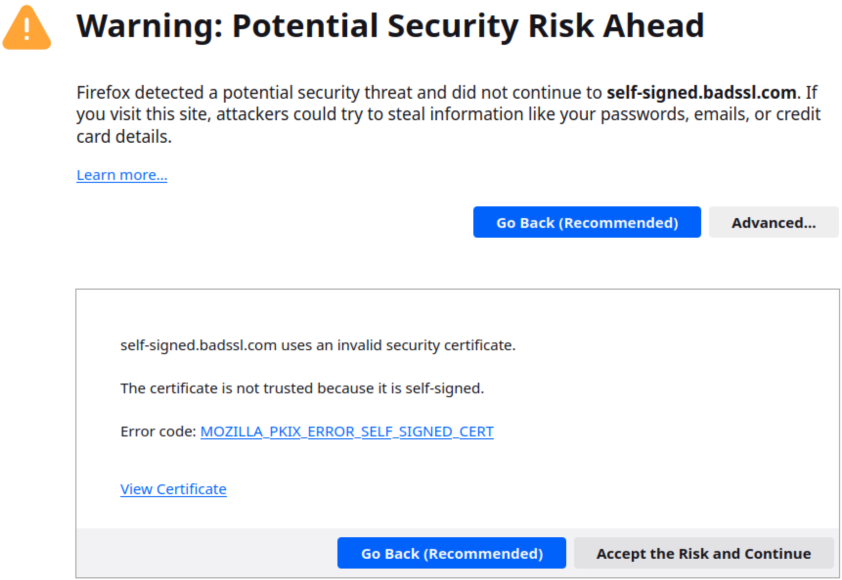
This store is not yet open.

Find out when we open:

This shop will be powered by shopify

How do we get certs?

- pay for one of the trusted authorities to give you one.
- use a **self-signed cert**
  - "joel's public key is XXX signed by XXX"
  - only for backwards compatibility
  - you sign your key with your own key
  - still not an authentic channel but what attacks it stop?



Warning: Potential Security Risk Ahead

Firefox detected a potential security threat and did not continue to **self-signed.badssl.com**. If you visit this site, attackers could try to steal information like your passwords, emails, or credit card details.

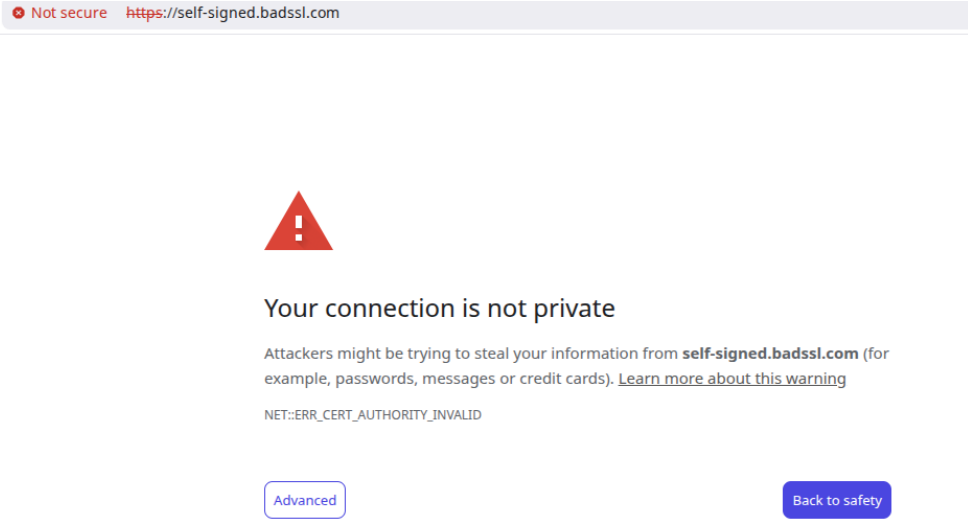
Learn more…

self-signed.badssl.com uses an invalid security certificate.

The certificate is not trusted because it is self-signed.

Error code: MOZILLA_PKIX_ERROR_SELF_SIGNED_CERT

View Certificate



Your connection is not private

Attackers might be trying to steal your information from **self-signed.badssl.com** (for example, passwords, messages or credit cards). Learn more about this warning

NET::ERR_CERT_AUTHORITY_INVALID

What is the trust model being used for the "proceed anyways" or "confirm security excpetion"

This alarm bell design is good, but it incentives **not using security** because not using security generally had no alarm bells! (Think about threat model.)

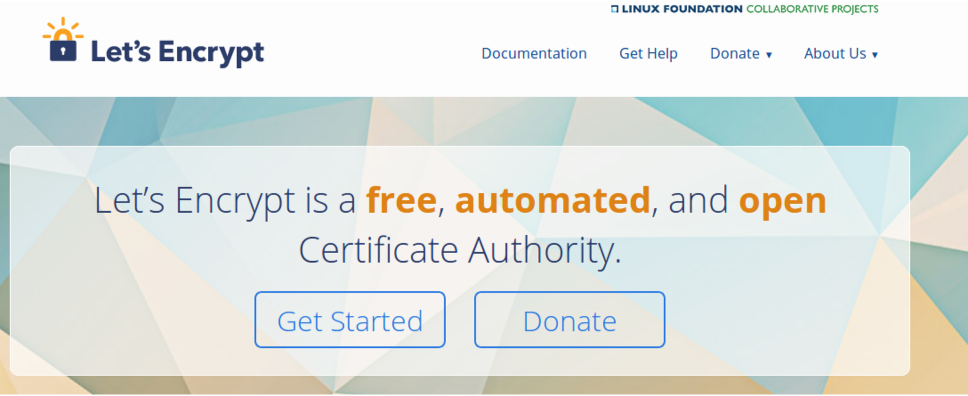
It should be as hard or worse to use insecure sites.

It should be as hard or worse to use insecure sites.
Best case of a self-signed cert: it’s the real cert.

It should be as hard or worse to use insecure sites.
Best case of a self-signed cert: it’s the real cert.
Worst case of a self-signed cert: not using security.

It should be as hard or worse to use insecure sites.
Best case of a self-signed cert: it's the real cert.
Worst case of a self-signed cert: not using security.
If you get a not real cert then you are
being actively man-in-the-middled when
you confirm the security exception (TOFU)

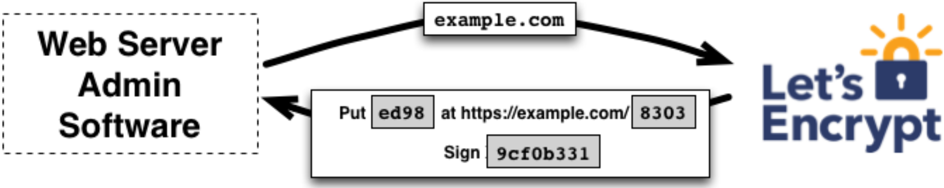
It should be as hard or worse to use insecure sites.
Best case of a self-signed cert: it's the real cert.
Worst case of a self-signed cert: not using security.
If you get a not real cert then you are
being actively man-in-the-middled when
you confirm the security exception (TOFU)
No cert means any **passive** attacker can read your
traffic as well as actively modify, now and later.



Let's Encrypt is a **free**, **automated**, and **open** Certificate Authority.

Let’s Encrypt

- run by ISRG (Internet Security Research Group)
- free automated open cert signing
  - only does DV, not OV or EV
  - supported by donations and volunteers
- allows anyone with just a webpage to have a nice signed cert
  - browsers trust the letsencrypt cert
  - avoids the warning alarms for self signed certs
  - avoids not using encryption



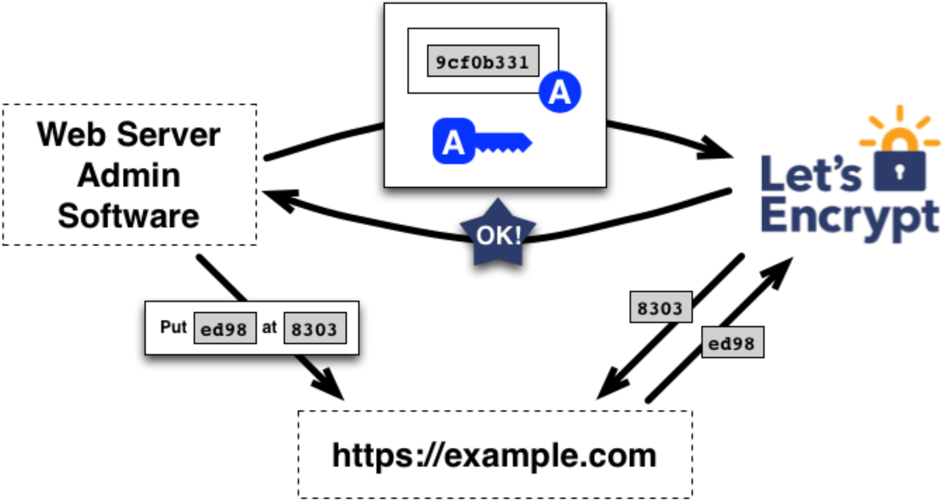
Web Server Admin Software

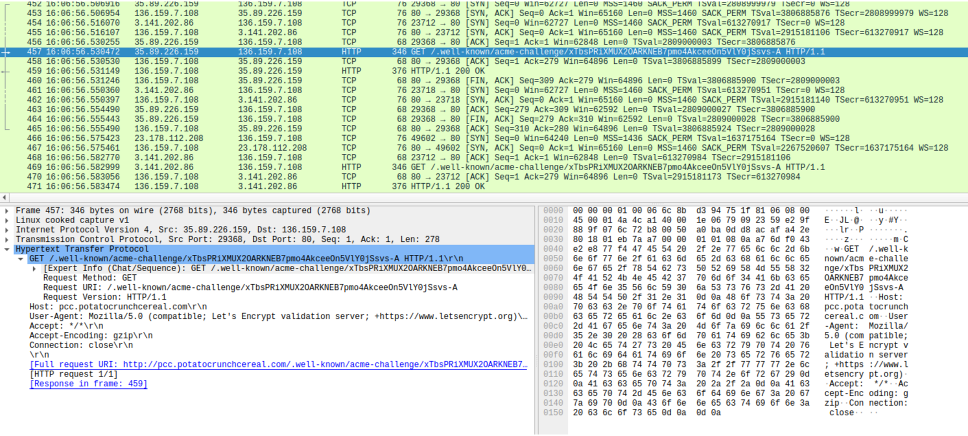
example.com

Put ed98 at https://example.com/8303

Sign 9cf0b331

Let's Encrypt

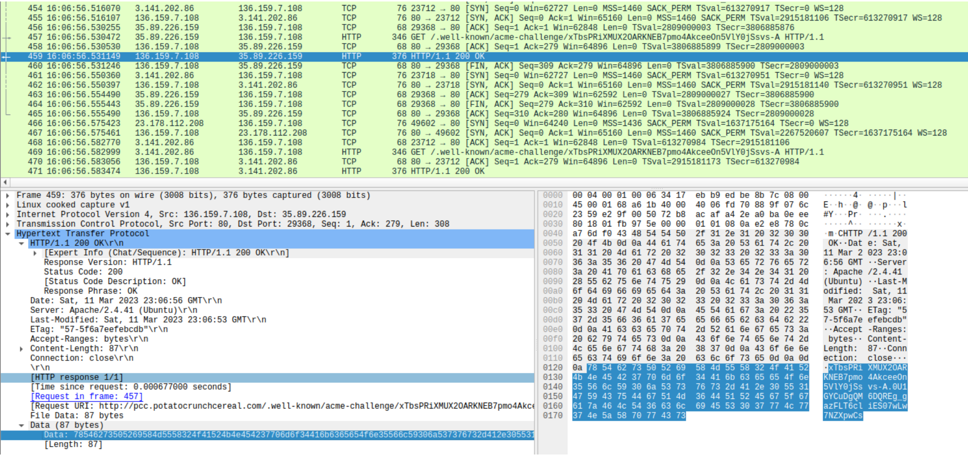

9cf0b331
A
A
Web Server
Admin
Software
Let's Encrypt
OK!
Put ed98 at 8303
8303
ed98
https://example.com



```
452 16:06:56.506916  35.89.226.159   136.159.7.108   TCP    76 29368 → 80 [SYN] Seq=0 Win=62727 Len=0 MSS=1460 SACK_PERM TSval=2808999979 TSecr=0 WS=128
453 16:06:56.506954  136.159.7.108   35.89.226.159   TCP    76 80 → 29368 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=3806885876 TSecr=2808999979 WS=128
454 16:06:56.516070  3.141.202.86    136.159.7.108   TCP    76 23712 → 80 [SYN] Seq=0 Win=62727 Len=0 MSS=1460 SACK_PERM TSval=613270917 TSecr=0 WS=128
455 16:06:56.516107  136.159.7.108   3.141.202.86    TCP    76 80 → 23712 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=2915181106 TSecr=613270917 WS=128
456 16:06:56.530255  35.89.226.159   136.159.7.108   TCP    68 29368 → 80 [ACK] Seq=1 Ack=1 Win=62848 Len=0 TSval=2809000003 TSecr=3806885876
457 16:06:56.530472  35.89.226.159   136.159.7.108   HTTP  346 GET /.well-known/acme-challenge/xTbsPRiXMUX2OARKNEB7pmo4AkceeOn5VlY0jSsvs-A HTTP/1.1
458 16:06:56.530530  136.159.7.108   35.89.226.159   TCP    68 80 → 29368 [ACK] Seq=1 Ack=279 Win=64896 Len=0 TSval=3806885899 TSecr=2809000003
459 16:06:56.531149  136.159.7.108   35.89.226.159   HTTP  376 HTTP/1.1 200 OK
460 16:06:56.531246  136.159.7.108   35.89.226.159   TCP    68 80 → 29368 [FIN, ACK] Seq=309 Ack=279 Win=64896 Len=0 TSval=3806885900 TSecr=2809000003
461 16:06:56.550360  3.141.202.86    136.159.7.108   TCP    76 23718 → 80 [SYN] Seq=0 Win=62727 Len=0 MSS=1460 SACK_PERM TSval=613270951 TSecr=0 WS=128
462 16:06:56.550397  136.159.7.108   3.141.202.86    TCP    76 80 → 23718 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=2915181140 TSecr=613270951 WS=128
463 16:06:56.554490  35.89.226.159   136.159.7.108   TCP    68 29368 → 80 [ACK] Seq=279 Ack=309 Win=62592 Len=0 TSval=2809000027 TSecr=3806885900
464 16:06:56.555443  35.89.226.159   136.159.7.108   TCP    68 29368 → 80 [FIN, ACK] Seq=279 Ack=310 Win=62592 Len=0 TSval=2809000028 TSecr=3806885900
465 16:06:56.555490  136.159.7.108   35.89.226.159   TCP    68 80 → 29368 [ACK] Seq=310 Ack=280 Win=64896 Len=0 TSval=3806885924 TSecr=2809000028
466 16:06:56.575423  23.178.112.208  136.159.7.108   TCP    76 49602 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1436 SACK_PERM TSval=1637175164 TSecr=0 WS=128
467 16:06:56.575461  136.159.7.108   23.178.112.208  TCP    76 80 → 49602 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=2267520607 TSecr=1637175164 WS=128
468 16:06:56.582770  3.141.202.86    136.159.7.108   TCP    68 23712 → 80 [ACK] Seq=1 Ack=1 Win=62848 Len=0 TSval=613270984 TSecr=2915181106
469 16:06:56.582999  3.141.202.86    136.159.7.108   HTTP  346 GET /.well-known/acme-challenge/xTbsPRiXMUX2OARKNEB7pmo4AkceeOn5VlY0jSsvs-A HTTP/1.1
470 16:06:56.583056  136.159.7.108   3.141.202.86    TCP    68 80 → 23712 [ACK] Seq=1 Ack=279 Win=64896 Len=0 TSval=2915181173 TSecr=613270984
471 16:06:56.583474  136.159.7.108   3.141.202.86    HTTP  376 HTTP/1.1 200 OK
```

```
Frame 457: 346 bytes on wire (2768 bits), 346 bytes captured (2768 bits)
Linux cooked capture v1
Internet Protocol Version 4, Src: 35.89.226.159, Dst: 136.159.7.108
Transmission Control Protocol, Src Port: 29368, Dst Port: 80, Seq: 1, Ack: 1, Len: 278
Hypertext Transfer Protocol
  GET /.well-known/acme-challenge/xTbsPRiXMUX2OARKNEB7pmo4AkceeOn5VlY0jSsvs-A HTTP/1.1\r\n
    [Expert Info (Chat/Sequence): GET /.well-known/acme-challenge/xTbsPRiXMUX2OARKNEB7pmo4AkceeOn5VlY0…
    Request Method: GET
    Request URI: /.well-known/acme-challenge/xTbsPRiXMUX2OARKNEB7pmo4AkceeOn5VlY0jSsvs-A
    Request Version: HTTP/1.1
  Host: pcc.potatocrunchcereal.com\r\n
  User-Agent: Mozilla/5.0 (compatible; Let's Encrypt validation server; +https://www.letsencrypt.org)\…
  Accept: */*\r\n
  Accept-Encoding: gzip\r\n
  Connection: close\r\n
  \r\n
  [Full request URI: http://pcc.potatocrunchcereal.com/.well-known/acme-challenge/xTbsPRiXMUX2OARKNEB7…
  [HTTP request 1/1]
  [Response in frame: 459]
```

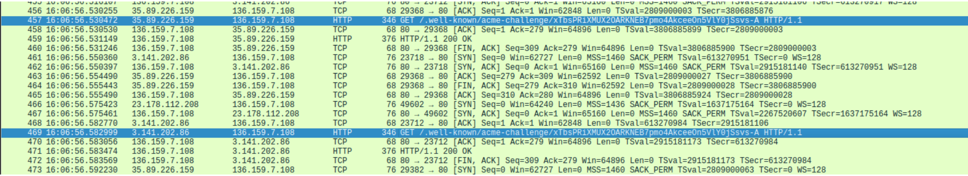
```
0000  00 00 00 01 00 06 6c 8b  d3 94 75 1f 81 06 08 00   ······l· ··u·····
0010  45 00 01 4a 4c a1 40 00  1e 06 79 09 23 59 e2 9f   E··JL·@· ··y·#Y··
0020  88 9f 07 6c 72 b8 00 50  a0 ba 0d d8 ac af a4 2e   ···lr··P ·······.
0030  80 18 01 eb 7a a7 00 00  01 01 08 0a a7 6d f0 43   ····z··· ·····m·C
0040  e2 e8 77 f4 47 45 54 20  2f 2e 77 65 6c 6c 2d 6b   ··w·GET  /.well-k
0050  6e 6f 77 6e 2f 61 63 6d  65 2d 63 68 61 6c 6c 65   nown/acm e-challe
0060  6e 67 65 2f 78 54 62 73  50 52 69 58 4d 55 58 32   nge/xTbs PRiXMUX2
0070  4f 41 52 4b 4e 45 42 37  70 6d 6f 34 41 6b 63 65   OARKNEB7 pmo4Akce
0080  65 4f 6e 35 56 6c 59 30  6a 53 73 76 73 2d 41 20   eOn5VlY0 jSsvs-A 
0090  48 54 54 50 2f 31 2e 31  0d 0a 48 6f 73 74 3a 20   HTTP/1.1 ··Host: 
00a0  70 63 63 2e 70 6f 74 61  74 6f 63 72 75 6e 63 68   pcc.pota tocrunch
00b0  63 65 72 65 61 6c 2e 63  6f 6d 0d 0a 55 73 65 72   cereal.c om··User
00c0  2d 41 67 65 6e 74 3a 20  4d 6f 7a 69 6c 6c 61 2f   -Agent:  Mozilla/
00d0  35 2e 30 20 28 63 6f 6d  70 61 74 69 62 6c 65 3b   5.0 (com patible;
00e0  20 4c 65 74 27 73 20 45  6e 63 72 79 70 74 20 76    Let's E ncrypt v
00f0  61 6c 69 64 61 74 69 6f  6e 20 73 65 72 76 65 72   alidatio n server
0100  3b 20 2b 68 74 74 70 73  3a 2f 2f 77 77 77 2e 6c   ; +https ://www.l
0110  65 74 73 65 6e 63 72 79  70 74 2e 6f 72 67 29 0d   etsencry pt.org)·
0120  0a 41 63 63 65 70 74 3a  20 2a 2f 2a 0d 0a 41 63   ·Accept:  */*··Ac
0130  63 65 70 74 2d 45 6e 63  6f 64 69 6e 67 3a 20 67   cept-Enc oding: g
0140  7a 69 70 0d 0a 43 6f 6e  6e 65 63 74 69 6f 6e 3a   zip··Con nection:
0150  20 63 6c 6f 73 65 0d 0a  0d 0a                      close·· ··
```



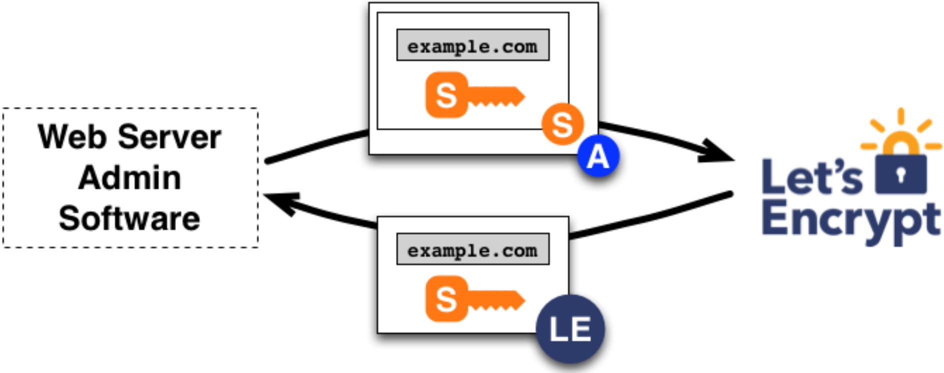
```
454 16:06:56.516070  3.141.202.86     136.159.7.108    TCP    76 23712 → 80 [SYN] Seq=0 Win=62727 Len=0 MSS=1460 SACK_PERM TSval=613270917 TSecr=0 WS=128
455 16:06:56.516107  136.159.7.108    3.141.202.86     TCP    76 80 → 23712 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=2915181106 TSecr=613270917 WS=128
456 16:06:56.530255  35.89.226.159    136.159.7.108    TCP    68 29368 → 80 [ACK] Seq=1 Ack=1 Win=62848 Len=0 TSval=2809000003 TSecr=3806885876
457 16:06:56.530472  35.89.226.159    136.159.7.108    HTTP  346 GET /.well-known/acme-challenge/xTbsPRiXMUX2OARKNEB7pmo4AkceeOn5VlY0jSsvs-A HTTP/1.1
458 16:06:56.530530  136.159.7.108    35.89.226.159    TCP    68 80 → 29368 [ACK] Seq=1 Ack=279 Win=64896 Len=0 TSval=3806885899 TSecr=2809000003
459 16:06:56.531149  136.159.7.108    35.89.226.159    HTTP  376 HTTP/1.1 200 OK
460 16:06:56.531246  136.159.7.108    35.89.226.159    TCP    68 80 → 29368 [FIN, ACK] Seq=309 Ack=279 Win=64896 Len=0 TSval=3806885900 TSecr=2809000003
461 16:06:56.550360  3.141.202.86     136.159.7.108    TCP    76 23718 → 80 [SYN] Seq=0 Win=62727 Len=0 MSS=1460 SACK_PERM TSval=613270951 TSecr=0 WS=128
462 16:06:56.550397  136.159.7.108    3.141.202.86     TCP    76 80 → 23718 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=2915181140 TSecr=613270951 WS=128
463 16:06:56.554490  35.89.226.159    136.159.7.108    TCP    68 29368 → 80 [ACK] Seq=279 Ack=309 Win=62592 Len=0 TSval=2809000027 TSecr=3806885900
464 16:06:56.555443  35.89.226.159    136.159.7.108    TCP    68 29368 → 80 [FIN, ACK] Seq=279 Ack=310 Win=62592 Len=0 TSval=2809000028 TSecr=3806885900
465 16:06:56.555490  136.159.7.108    35.89.226.159    TCP    68 80 → 29368 [ACK] Seq=310 Ack=280 Win=64896 Len=0 TSval=3806885924 TSecr=2809000028
466 16:06:56.575423  23.178.112.208   136.159.7.108    TCP    76 49602 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1436 SACK_PERM TSval=1637175164 TSecr=0 WS=128
467 16:06:56.575461  136.159.7.108    23.178.112.208   TCP    76 80 → 49602 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=2267520607 TSecr=1637175164 WS=128
468 16:06:56.582770  3.141.202.86     136.159.7.108    TCP    68 23712 → 80 [ACK] Seq=1 Ack=1 Win=62848 Len=0 TSval=613270984 TSecr=2915181106
469 16:06:56.582999  3.141.202.86     136.159.7.108    HTTP  346 GET /.well-known/acme-challenge/xTbsPRiXMUX2OARKNEB7pmo4AkceeOn5VlY0jSsvs-A HTTP/1.1
470 16:06:56.583056  136.159.7.108    3.141.202.86     TCP    68 80 → 23712 [ACK] Seq=1 Ack=279 Win=64896 Len=0 TSval=2915181173 TSecr=613270984
471 16:06:56.583474  136.159.7.108    3.141.202.86     HTTP  376 HTTP/1.1 200 OK
```

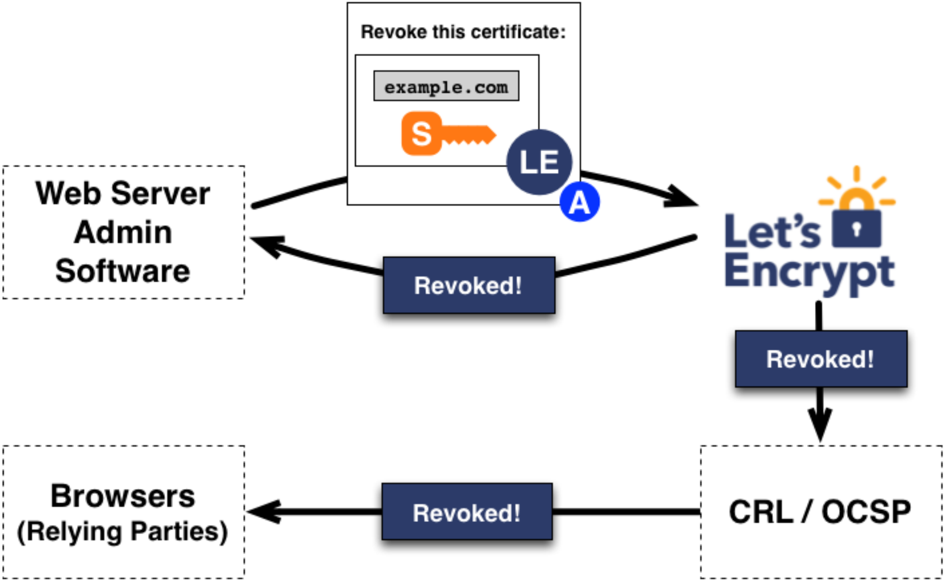
```
Frame 459: 376 bytes on wire (3008 bits), 376 bytes captured (3008 bits)
Linux cooked capture v1
Internet Protocol Version 4, Src: 136.159.7.108, Dst: 35.89.226.159
Transmission Control Protocol, Src Port: 80, Dst Port: 29368, Seq: 1, Ack: 279, Len: 308
Hypertext Transfer Protocol
  HTTP/1.1 200 OK\r\n
    [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
    Response Version: HTTP/1.1
    Status Code: 200
    [Status Code Description: OK]
    Response Phrase: OK
  Date: Sat, 11 Mar 2023 23:06:56 GMT\r\n
  Server: Apache/2.4.41 (Ubuntu)\r\n
  Last-Modified: Sat, 11 Mar 2023 23:06:53 GMT\r\n
  ETag: "57-5f6a7eefebcdb"\r\n
  Accept-Ranges: bytes\r\n
  Content-Length: 87\r\n
  Connection: close\r\n
  \r\n
  [HTTP response 1/1]
  [Time since request: 0.000677000 seconds]
  [Request in frame: 457]
  [Request URI: http://pcc.potatocrunchcereal.com/.well-known/acme-challenge/xTbsPRiXMUX2OARKNEB7pmo4Akce
  File Data: 87 bytes
  Data (87 bytes)
    Data: 785462735052695844d5558324f41524b4e454237706d6f34416b6365654f6e35566c59306a537376732d412e305531
    [Length: 87]
```

```
0000  00 04 00 01 00 06 34 17  eb b9 ed be 8b 7c 08 00   ······4· ·····|··
0010  45 00 01 68 a6 1b 40 00  40 06 fd 70 88 9f 07 6c   E··h··@· @··p···l
0020  23 59 e2 9f 00 50 72 b8  ac af a4 2e a0 ba 0e ee   #Y···Pr· ···.····
0030  80 18 01 fb 97 5e 00 00  01 01 08 0a e2 e8 78 0c   ·····^·· ······x·
0040  a7 6d f0 43 48 54 54 50  2f 31 2e 31 20 32 30 30   ·m·CHTTP /1.1 200
0050  20 4f 4b 0d 0a 44 61 74  65 3a 20 53 61 74 2c 20    OK··Dat e: Sat,
0060  31 31 20 4d 61 72 20 32  30 32 33 20 32 33 3a 30   11 Mar 2 023 23:0
0070  36 3a 35 36 20 47 4d 54  0d 0a 53 65 72 76 65 72   6:56 GMT ··Server
0080  3a 20 41 70 61 63 68 65  2f 32 2e 34 2e 34 31 20   : Apache /2.4.41
0090  28 55 62 75 6e 74 75 29  0d 0a 4c 61 73 74 2d 4d   (Ubuntu) ··Last-M
00a0  6f 64 69 66 69 65 64 3a  20 53 61 74 2c 20 31 31   odified:  Sat, 11
00b0  20 4d 61 72 20 32 30 32  33 20 32 33 3a 30 36 3a    Mar 202 3 23:06:
00c0  35 33 20 47 4d 54 0d 0a  45 54 61 67 3a 20 22 35   53 GMT·· ETag: "5
00d0  37 2d 35 66 36 61 37 65  65 66 65 62 63 64 62 22   7-5f6a7e efebcdb"
00e0  0d 0a 41 63 63 65 70 74  2d 52 61 6e 67 65 73 3a   ··Accept -Ranges:
00f0  20 62 79 74 65 73 0d 0a  43 6f 6e 74 65 6e 74 2d    bytes·· Content-
0100  4c 65 6e 67 74 68 3a 20  38 37 0d 0a 43 6f 6e 6e   Length:  87··Conn
0110  65 63 74 69 6f 6e 3a 20  63 6c 6f 73 65 0d 0a 0d   ection:  close···
0120  0a 78 54 62 73 50 52 69  58 4d 55 58 32 4f 41 52   ·xTbsPRi XMUX2OAR
0130  4b 4e 45 42 37 70 6d 6f  34 41 6b 63 65 65 4f 6e   KNEB7pmo 4AkceeOn
0140  35 56 6c 59 30 6a 53 73  76 73 2d 41 2e 30 55 31   5VlY0jSs vs-A.0U1
0150  47 59 43 75 44 67 51 4d  36 44 51 52 45 67 5f 67   GYCuDgQM 6DQREg_g
0160  61 7a 46 4c 54 36 63 6c  69 45 53 30 37 77 4c 77   azFLT6cl iES07wLw
0170  37 4e 5a 58 70 77 43 73                            7NZXpwCs
```



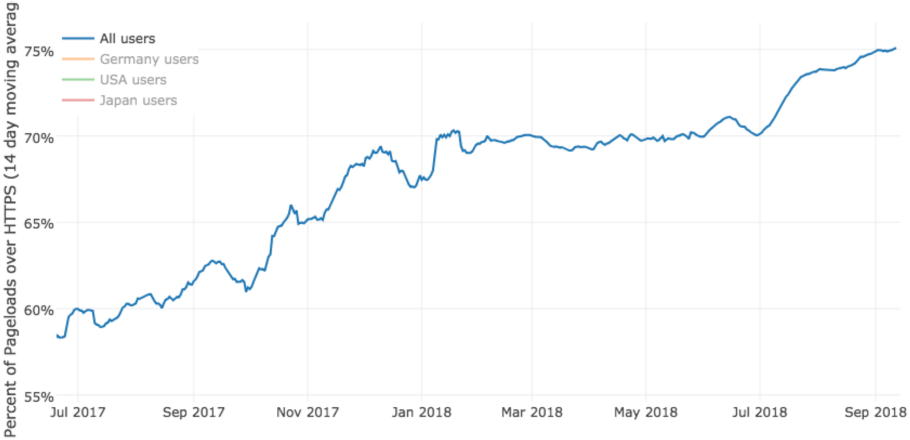
```
    455 16:06:56.516107  136.159.7.108    3.141.202.86     TCP        76 80 → 23712 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=2915181106 TSecr=613270917 WS=128
    456 16:06:56.530255  35.89.226.159    136.159.7.108    TCP        68 29368 → 80 [ACK] Seq=1 Ack=1 Win=62848 Len=0 TSval=2809000003 TSecr=3806885876
    457 16:06:56.530472  35.89.226.159    136.159.7.108    HTTP      346 GET /.well-known/acme-challenge/xTbsPRiXMUX2OARKNEB7pmo4AkceeOn5VlY0jSsvs-A HTTP/1.1
    458 16:06:56.530530  136.159.7.108    35.89.226.159    TCP        68 80 → 29368 [ACK] Seq=1 Ack=279 Win=64896 Len=0 TSval=3806885899 TSecr=2809000003
    459 16:06:56.531149  136.159.7.108    35.89.226.159    HTTP      376 HTTP/1.1 200 OK
    460 16:06:56.531246  136.159.7.108    35.89.226.159    TCP        68 80 → 29368 [FIN, ACK] Seq=309 Ack=279 Win=64896 Len=0 TSval=3806885900 TSecr=2809000003
    461 16:06:56.550360  3.141.202.86     136.159.7.108    TCP        76 23718 → 80 [SYN] Seq=0 Win=62727 Len=0 MSS=1460 SACK_PERM TSval=613270951 TSecr=0 WS=128
    462 16:06:56.550397  136.159.7.108    3.141.202.86     TCP        76 80 → 23718 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=2915181140 TSecr=613270951 WS=128
    463 16:06:56.554490  35.89.226.159    136.159.7.108    TCP        68 29368 → 80 [ACK] Seq=279 Ack=309 Win=62592 Len=0 TSval=2809000027 TSecr=3806885900
    464 16:06:56.555443  35.89.226.159    136.159.7.108    TCP        68 29368 → 80 [FIN, ACK] Seq=279 Ack=310 Win=62592 Len=0 TSval=2809000028 TSecr=3806885900
    465 16:06:56.555490  136.159.7.108    35.89.226.159    TCP        68 80 → 29368 [ACK] Seq=310 Ack=280 Win=64896 Len=0 TSval=3806885924 TSecr=2809000028
    466 16:06:56.575423  23.178.112.208   136.159.7.108    TCP        76 49602 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1436 SACK_PERM TSval=1637175164 TSecr=0 WS=128
    467 16:06:56.575461  136.159.7.108    23.178.112.208   TCP        76 80 → 49602 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=2267520607 TSecr=1637175164 WS=128
    468 16:06:56.582770  3.141.202.86     136.159.7.108    TCP        68 23712 → 80 [ACK] Seq=1 Ack=1 Win=62848 Len=0 TSval=613270984 TSecr=2915181106
    469 16:06:56.582999  3.141.202.86     136.159.7.108    HTTP      346 GET /.well-known/acme-challenge/xTbsPRiXMUX2OARKNEB7pmo4AkceeOn5VlY0jSsvs-A HTTP/1.1
    470 16:06:56.583056  136.159.7.108    3.141.202.86     TCP        68 80 → 23712 [ACK] Seq=1 Ack=279 Win=64896 Len=0 TSval=2915181173 TSecr=613270984
    471 16:06:56.583474  136.159.7.108    3.141.202.86     HTTP      376 HTTP/1.1 200 OK
    472 16:06:56.583569  136.159.7.108    3.141.202.86     TCP        68 80 → 23712 [FIN, ACK] Seq=309 Ack=279 Win=64896 Len=0 TSval=2915181173 TSecr=613270984
    473 16:06:56.592230  35.89.226.159    136.159.7.108    TCP        76 29382 → 80 [SYN] Seq=0 Win=62727 Len=0 MSS=1460 SACK_PERM TSval=2809000063 TSecr=0 WS=128
```


Web Server Admin Software
example.com
S
S
A
Let's Encrypt
example.com
S
LE


Revoke this certificate:
example.com
S
LE
A
Web Server Admin Software
Let's Encrypt
Revoked!
Revoked!
Browsers (Relying Parties)
Revoked!
CRL / OCSP

Let's Encrypt

- started in 2014 by EFF and backed by Akamai, Google, Facebook, Mozilla, and more
- more than 600 million active certificates (2025)
  - largest certificate issuer in the world
  - issuing around 7 million certificates a day
- 83% of all firefox traffic in 2021 is HTTPS (secured)
  - it was 67% in 2017
  - it was 25% in 2013
  - steady since 2021
- it used to be hard and expensive to get a cert


Percent of Pageloads over HTTPS (14 day moving averag
All users
Germany users
USA users
Japan users
75%
70%
65%
60%
55%
Jul 2017
Sep 2017
Nov 2017
Jan 2018
Mar 2018
May 2018
Jul 2018
Sep 2018