The (Decentralized)

USENIX Security 2011

Peter Eckersley
EFF

Jesse Burns
iSEC

SSL/TLS : Earth's most popular cryptographic system

How strong is this infrastructure?

at best, as good as its
ability to authenticate the other party

How does that happen?

With X.509 certificates
signed by Certificate Authorities (CAs)

SSL, TLS, HTTPS, X.509, PKIX

SSL ~= TLS

HTTPS = HTTP + TLS

TLS uses certificates

X.509 = certificates!

PKIX = public X.509

The SSL Observatory

Investigates:

how imperfect are CAs?
what are they signing?
how large is the PKIX attack surface?

The SSL Observatory

Methodology:

Collect all the X.509 certificates
See what's in them

The SSL Observatory

2010:

Scanned all allocated IPv4 space

(port 443)

Built a system for analysing the data

The SSL Observatory

2011:

Decentralized Observatory client

Opt-in feature for HTTPS Everywhere

Uses Tor for anonymization

Launching this afternoon!

Scanning IPv4

3 billion IANA-allocated addresses

Partition into work units

Use `nmap` to SYN scan port 443

Followup to collect certificates

Decentralized Observatory

Interesting phenomena may be localized
Want to see certs from many viewpoints

Observatory Browser Extension

Collects: certificiate chain*, destination domain, approx. timestamp, optional ASN + server IP

Whitelists for scalability

Does not log client IP

Returns: known reasons for mistrust

Early alpha implementation

Those certificates

The CA says:

”This certificate and its key belong to www.eff.org.”

And enforce honestly and reasonableness*

We were afraid of CAs because:

They have a hard job, with odd incentives

2009: 3 vulnerabilities due to CA mistakes

2010: evidence of governments compelling CAs

There seemed to be a lot of them

Also afraid of X.509

Designed in 1980s
By the ITU (!), before HTTP (!!!)

+ extremely flexible & general

- extremely flexible & general
- extremely ugly
- history of implementation vulnerabilities

X.509: Security via digital paperwork

X.509 certs can (and do) contain just about anything

How many kinds of anything?

```
#!/usr/bin/env python

# diversity.py -- estimate the number of different certificate types and
# combinations of fields in them

from dbconnect import dbconnect
db,dbc = dbconnect()
q = ”””
SELECT *,`X509v3 extensions:X509v3 Key Usage`,
       `X509v3 extensions:X509v3 Extended Key Usage`,
       `X509v3 extensions:X509v3 Basic Constraints:CA`,
       `X509v3 extensions:Netscape Cert Type`
FROM all_certs
WHERE certid >= %d and certid < %d
”””

dbc.execute(”SELECT count(certid) from all_certs”)
n = int(dbc.fetchone()[0])
print n, ”rows”

fset = {}
for i in range(n / 1024):
  q1 = q % (i* 1024, (i+1) * 1024)
  dbc.execute(q1)
  batch= dbc.fetchall()
  for row in batch:
    cert, type_fields = row[:-4], row[-4:]
    bits = 0
    for field in cert:
      if field==None:
        bits |= 0x01
      elif type(field) == str and (”critical” in field):
        bits |=0x02
      bits <<= 2
    key = (type_fields, bits)
    fset[bits]=True

print len(fset)
```

By this approximate measure:

10,320 *kinds* of X.509 certs were observed

1,352 kinds were sometimes valid

Not as bad as a million kinds,
still *hard to process automatically*

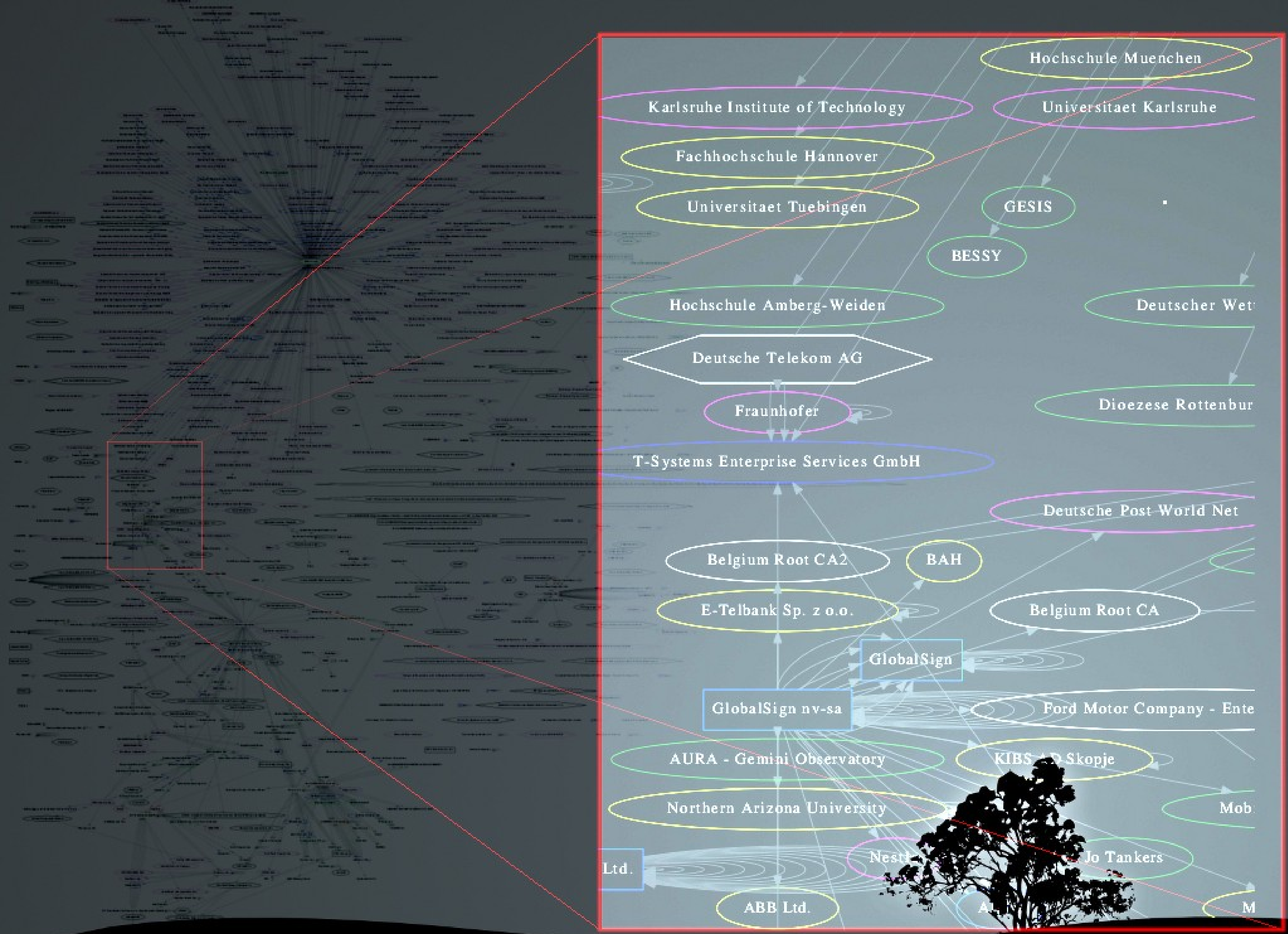
Size of the SSLiverse

16.2M IPs were listening on port 443
11.3M started an SSL handshake
4.3+M used valid cert chains
with only
1.5+M distinct valid leaves

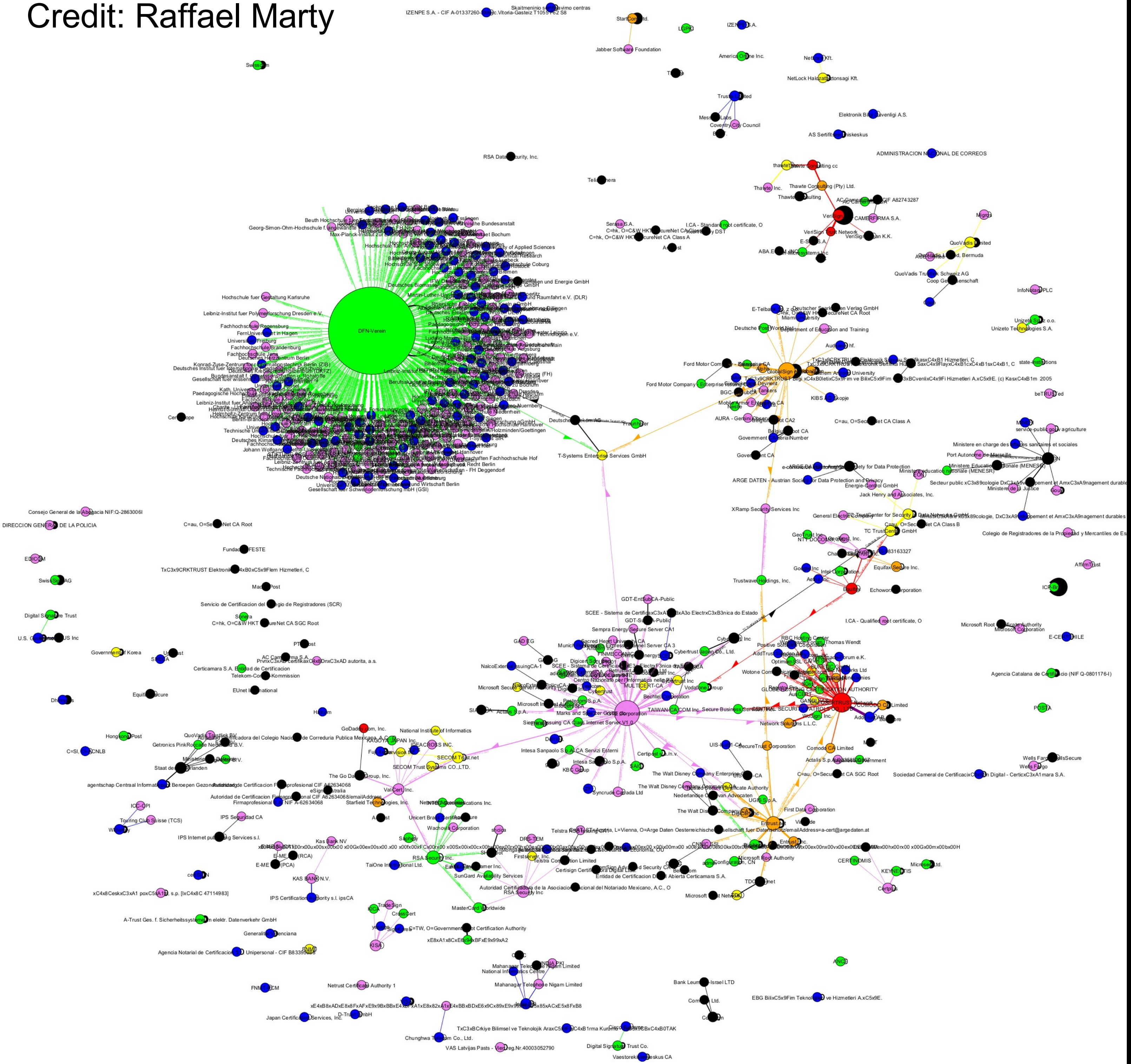
Lots of CAs!

1,482 CAs trustable by Microsoft or Mozilla
1,167 disinct Issuer strings
651 organisations

Mac OS X would add a few more


Hochschule Muenchen
Karlsruhe Institute of Technology
Universitaet Karlsruhe
Fachhochschule Hannover
Universitaet Tuebingen
GESIS
BESSY
Hochschule Amberg-Weiden
Deutscher Wet
Deutsche Telekom AG
Dioezese Rottenbur
Fraunhofer
T-Systems Enterprise Services GmbH
Deutsche Post World Net
Belgium Root CA2
BAH
E-Telbank Sp. z o.o.
Belgium Root CA
GlobalSign
GlobalSign nv-sa
Ford Motor Company - Ente
AURA - Gemini Observatory
KIBS AD Skopje
Northern Arizona University
Mob
Ltd.
Nestl
Jo Tankers
ABB Ltd.
A
M

Credit: Raffael Marty


DFN-Verein
T-Systems Enterprise Services GmbH
Deutsche Telekom AG
IZENPE S.A.
StartCom Ltd.
Jabber Software Foundation
America Online Inc.
NetLock Halozatbiztonsagi Kft.
Elektronik Bilgi Guvenligi A.S.
AS Sertifitseerimiskeskus
ADMINISTRACION NACIONAL DE CORREOS
RSA Data Security, Inc.
Thawte Consulting cc
Thawte Consulting (Pty) Ltd.
VeriSign Trust Network
CAMERFIRMA S.A.
QuoVadis Limited
QuoVadis Trustlink Schweiz AG
Unizeto Technologies S.A.
GlobalSign
Ford Motor Company
Department of Education and Training
Government Root CA
ARGE DATEN - Austrian Society for Data Protection and Privacy
XRamp Security Services Inc
TC TrustCenter GmbH
Equifax Secure Inc.
Trustwave Holdings, Inc.
Colegio de Registradores de la Propiedad y Mercantiles de España
Microsoft Root Certificate Authority
Microsoft Corporation
Agencia Catalana de Certificacio (NIF Q-0801176-I)
Consejo General de la Abogacia NIF:Q-2863006I
DIRECCION GENERAL DE LA POLICIA
C=au, O=SecureNet CA Root
Fundacion FESTE
Servicio de Certificacion del Colegio de Registradores (SCR)
Digital Signature Trust
Government of Korea
Certicamara S.A. Entidad de Certificacion
Telekom-Control-Kommission
EUnet International
GTE Corporation
TAIWAN-CA.COM Inc. Secure Business Services
Entrust.net
Network Solutions L.L.C.
SecureTrust Corporation
Comodo CA Limited
The Walt Disney Company
First Data Corporation
Sociedad Cameral de Certificacion Digital - Certicamara S.A.
Wells Fargo WellsSecure
GoDaddy.com, Inc.
The Go Daddy Group, Inc.
ValiCert, Inc.
Starfield Technologies, Inc.
SECOM Trust Systems CO.,LTD.
National Institute of Informatics
RSA Security Inc.
Wachovia Corporation
SunGard Availability Services
MasterCard Worldwide
TaiOne International Ltd.
Telstra Corporation Limited
Microsoft Trust Network
CERTINOMIS
IPS Certification Authority s.l. ipsCA
IPS Seguridad CA
Touring Club Suisse (TCS)
A-Trust Ges. f. Sicherheitssysteme im elektr. Datenverkehr GmbH
Agencia Notarial de Certificacion SL Unipersonal - CIF B83395988
Netrust Certificate Authority 1
Japan Certification Services, Inc.
Chunghwa Telecom Co., Ltd.
VAS Latvijas Pasts - Vien.Reg.Nr.40003052790
Digital Signature Trust Co.
Vaestorekisterikeskus CA
Mahanagar Telephone Nigam Limited
National Informatics Centre
Bank Leumi le-Israel LTD
EBG Bilisim Teknolojileri ve Hizmetleri A.S.
ANCERT

Noteworthy subordinate CAs

U.S. Department of Homeland Security

U.S. Defence Contractors

Oil Companies

CNNIC

Etisalat

Gemini Observatory

A note about CNNIC

Controversy: Mozilla added CNNIC to the trust root in 2009

But: Entrust signed a CNNIC subordinate CA in 2007

SHECA/Unitrust, another Chinese sub-CA appears to date from 2004 in the Microsoft roots

Exposure to *many* jurisdictions

CAs are located in these ~52 countries:

['AE', 'AT', 'AU', 'BE', 'BG', 'BM', 'BR', 'CA', 'CH', 'CL', 'CN',
'CO', 'CZ', 'DE', 'DK', 'EE', 'ES', 'EU', 'FI', 'FR', 'GB', 'HK',
'HU', 'IE', 'IL', 'IN', 'IS', 'IT', 'JP', 'KR', 'LT', 'LV', 'MK', 'MO',
'MX', 'MY', 'NL', 'NO', 'PL', 'PT', 'RO', 'RU', 'SE', 'SG', 'SI',
'SK', 'TN', 'TR', 'TW', 'UK', 'US', 'UY', 'WW', 'ZA']

Vulnerabilities (2010)

~30,000 servers use broken keys

~500 had valid CA signatures, including:

diplomatie.be
yandex.ru
lawwebmail.uchicago.edu

(now fixed/expired)

Vulnerabilities

Certificates that appear ''Valid” but don't identify anyone in particular.

Names like Localhost, Exchange, Mail, and IP addresses

Even private RFC 1918 IP addresses Undermines the idea of CAs

Other whackiness

Certificates that were and were not CA certs

Violations of Extended Validation rules

Certificates with huge lists of names

New CA certificates with keys from expired certificates

Also, we've published the data, so you can do further research on it

The schema for the 2010 datasets was quite baroque

(we may or may not keep using it)

Some simple examples:

```
SELECT RSA_Modulus_Bits, count(*)
FROM valid_certs
GROUP BY RSA_Modulus_Bits
ORDER BY cast(RSA_Modulus_Bits as decimal);
```

```
+-------------------+-----------+
| RSA_Modulus_Bits  | count(*)  |
+-------------------+-----------+
| 511               |         3 |
| 512               |      3977 |
| 730               |         1 |
| 767               |         1 |
| 768               |        34 |
| 1023              |       968 |
| 1024              |    821900 |
| ...               |       ... |
+-------------------+-----------+
```

```
SELECT `Signature Algorithm`, count(*)
FROM valid_certs
WHERE startdate > ”2010”
GROUP BY `Signature Algorithm`;

+---------------------------+----------+
| Signature Algorithm       | count(*) |
+---------------------------+----------+
|  md5WithRSAEncryption     |        3 |
|  sha1WithRSAEncryption    |   455511 |
|  sha256WithRSAEncryption  |       17 |
|  sha512WithRSAEncryption  |        1 |
+---------------------------+----------+
SELECT distinct issuer
FROM valid_certs
WHERE stardate > ”2010” AND
  `Signature Algorithm`= " md5WithRSAEncryption";
```

```
+------------------------------------------------------------------------+
| issuer                                                                 |
+------------------------------------------------------------------------+
|  O=Ministere de la Justice, CN=Autorite de Certification Serveurs      |
|  C=US, O=Anthem Inc, OU=Ecommerce, CN=Anthem Inc Certificate Authority |
+------------------------------------------------------------------------+
```

(fortunately, these CAs don't robo sign)

Validity

”Easy”, just invoke openssl with the Microsoft + Mozilla trust roots

Actually, not that easy...

Firefox and IE cache intermediate CA certificates...

So OpenSSL can't necessarily say whether a cert is valid in these browsers (!!!)

”Transvalidity”

valid, but only if the browser cached the right intermediate CA certs first

→

we catch most transvalid certs

transvalidity.py

First, find invalid certs where a plausible, valid intermediate cert was seen somewhere in the SSLiverse:

```
SELECT  certs1.path, certs1.id, valid_certs.path, certs1.fingerprint,
        certs1.fetchtime
FROM certs1 join valid_certs
ON certs1.issuer = valid_certs.subject and (
        (certs1.`Authority Key Identifier:keyid` is null and
         valid_certs.`Subject Key Identifier` is null)
    or
        certs1.`Authority Key Identifier:keyid` =
        valid_certs.`Subject Key Identifier`
)
WHERE not certs1.valid and
    (locate("unable to get local issuer certificate", certs1.moz_valid) or
     locate("unable to get local issuer certificate", certs1.ms_valid) )
GROUP BY certs1.fingerprint, valid_certs.path
```

Note: some variable names were simplified in this query:
certs1 is an example raw input certs table, Authority Key IDs have longer column names

transvalidity.py (ct'd)

Once we have some missing, valid, possibly determinative CA certs, we re-run OpenSSL:

```
openssl verify -CApath <all roots> -untrusted <rest of chain + query results> cert
```

Results go in the ”transvalid” column

```
select count(*) from valid_certs where transvalid="Yes"
```

→ 97,676 tranvalid certs

More examples of the dataset at work...

Which root CAs created the most subordinate CAs? SubordinateTracking.py

For each root cert:

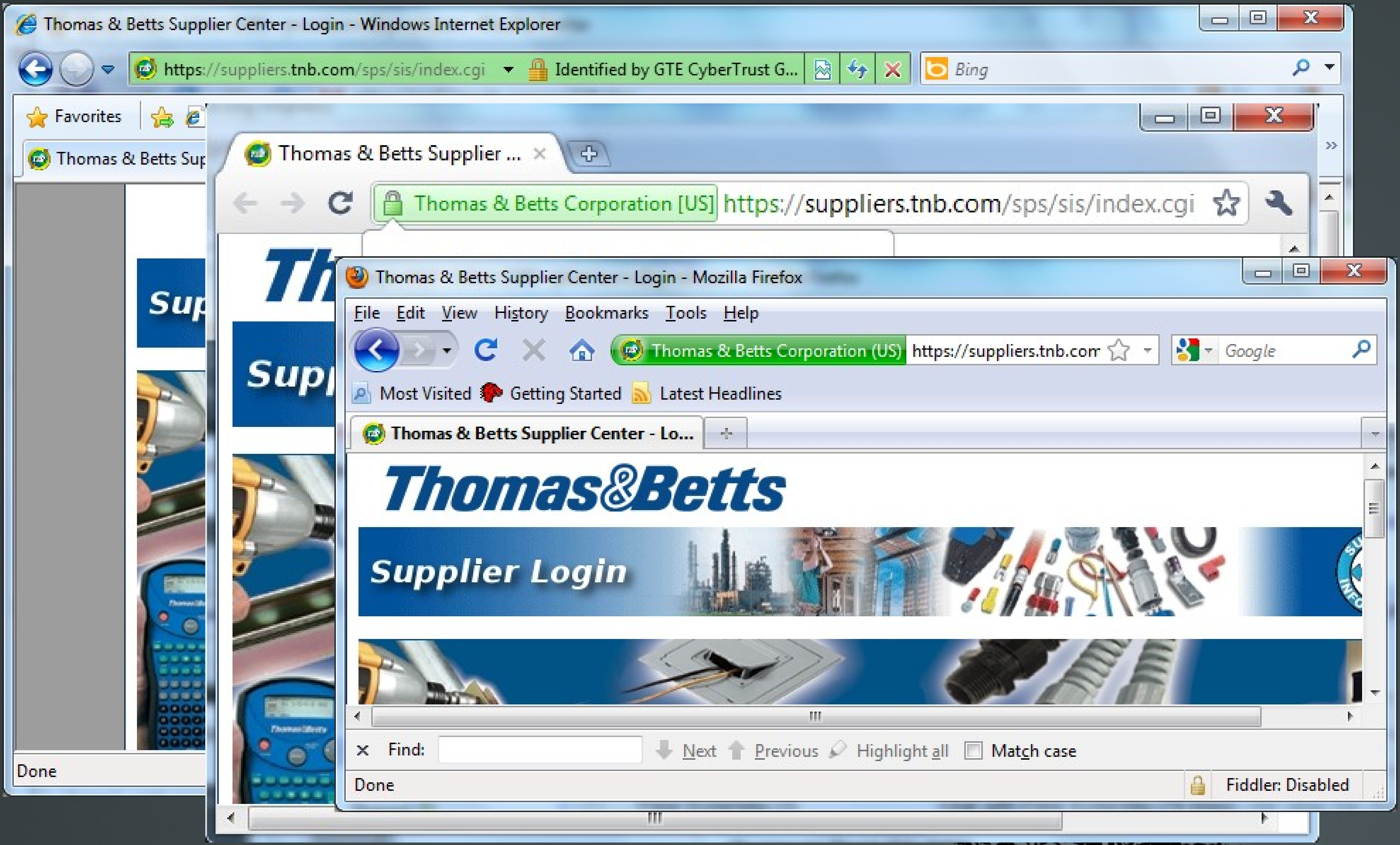
```
SELECT certid, subject, issuer, `Subject Key Idenfier`
FROM valid_certs where issuer = <root CA's subject>
    and locate(”true”, `X509v3 Basic Constraints:CA`)
    and `X509v3 Authority Key Identifier:keyid` = <root CA's SKID>
                                                       (which may be NULL)
```

(and recurse)

Results: top roots by CA proliferation

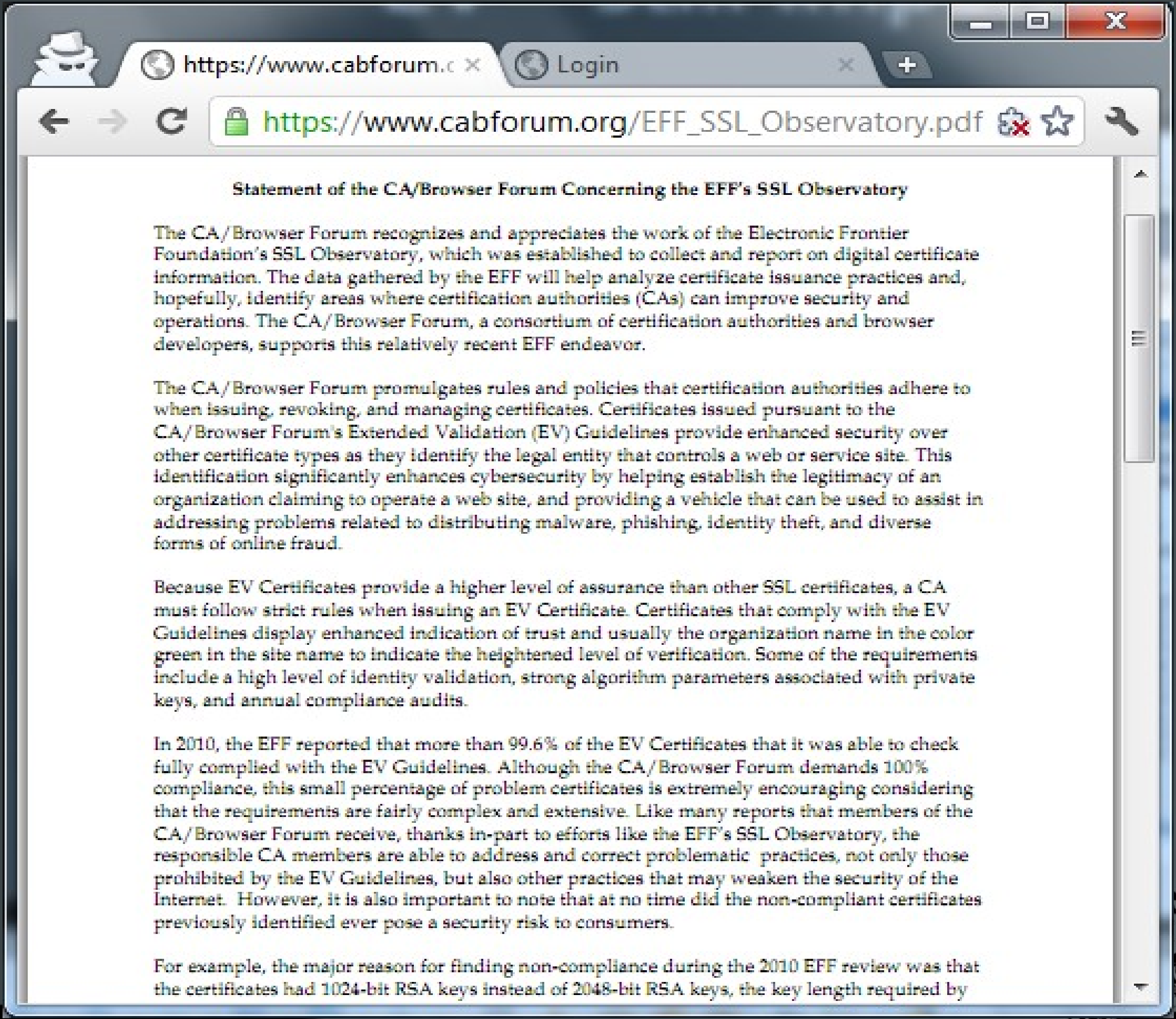
1. C=DE, CN=Deutsche Telekom Root CA 2	252 sub-CAs (4,164 leaves)
2. C=US, CN=GTE CyberTrust Global Root	93 sub-CAs (20,937 leaves)
3. C=SE, CN=AddTrust External CA Root	72 sub-CAs (384,481 leaves)
4. C=BE, CN=GlobalSign Root CA	63 sub-CAs (140,176 leaves)
5. C=US, CN=Entrust.net Secure Server Certification Authority	33 sub-CAs (91,203 leaves)
6. C=FR, O=PM/SGDN, OU=DCSSI, CN=IGC/A...	24 sub-CAs (448 leaves)
7. OU=ValiCert Class 3 Policy Validation Authority	20 sub-CAs (1,273 leaves)
8. O=VeriSign, Inc, OU=Class 3 Public Primary Certification Authority	18 sub-CAs (312,627 leaves)

Another 2010 finding: 512 bit EV cert



Thomas & Betts Supplier Center - Login - Windows Internet Explorer
https://suppliers.tnb.com/sps/sis/index.cgi
Identified by GTE CyberTrust G...
Bing
Favorites
Thomas & Betts Sup
Done
Thomas & Betts Supplier ...
Thomas & Betts Corporation [US]
https://suppliers.tnb.com/sps/sis/index.cgi
Thomas & Betts Supplier Center - Login - Mozilla Firefox
File
Edit
View
History
Bookmarks
Tools
Help
Thomas & Betts Corporation (US)
https://suppliers.tnb.com
Google
Most Visited
Getting Started
Latest Headlines
Thomas & Betts Supplier Center - Lo...
Thomas&Betts
Supplier Login
Find:
Next
Previous
Highlight all
Match case
Done
Fiddler: Disabled

EV & The CA/Browser Forum



Statement of the CA/Browser Forum Concerning the EFF's SSL Observatory

The CA/Browser Forum recognizes and appreciates the work of the Electronic Frontier Foundation's SSL Observatory, which was established to collect and report on digital certificate information. The data gathered by the EFF will help analyze certificate issuance practices and, hopefully, identify areas where certification authorities (CAs) can improve security and operations. The CA/Browser Forum, a consortium of certification authorities and browser developers, supports this relatively recent EFF endeavor.

The CA/Browser Forum promulgates rules and policies that certification authorities adhere to when issuing, revoking, and managing certificates. Certificates issued pursuant to the CA/Browser Forum's Extended Validation (EV) Guidelines provide enhanced security over other certificate types as they identify the legal entity that controls a web or service site. This identification significantly enhances cybersecurity by helping establish the legitimacy of an organization claiming to operate a web site, and providing a vehicle that can be used to assist in addressing problems related to distributing malware, phishing, identity theft, and diverse forms of online fraud.

Because EV Certificates provide a higher level of assurance than other SSL certificates, a CA must follow strict rules when issuing an EV Certificate. Certificates that comply with the EV Guidelines display enhanced indication of trust and usually the organization name in the color green in the site name to indicate the heightened level of verification. Some of the requirements include a high level of identity validation, strong algorithm parameters associated with private keys, and annual compliance audits.

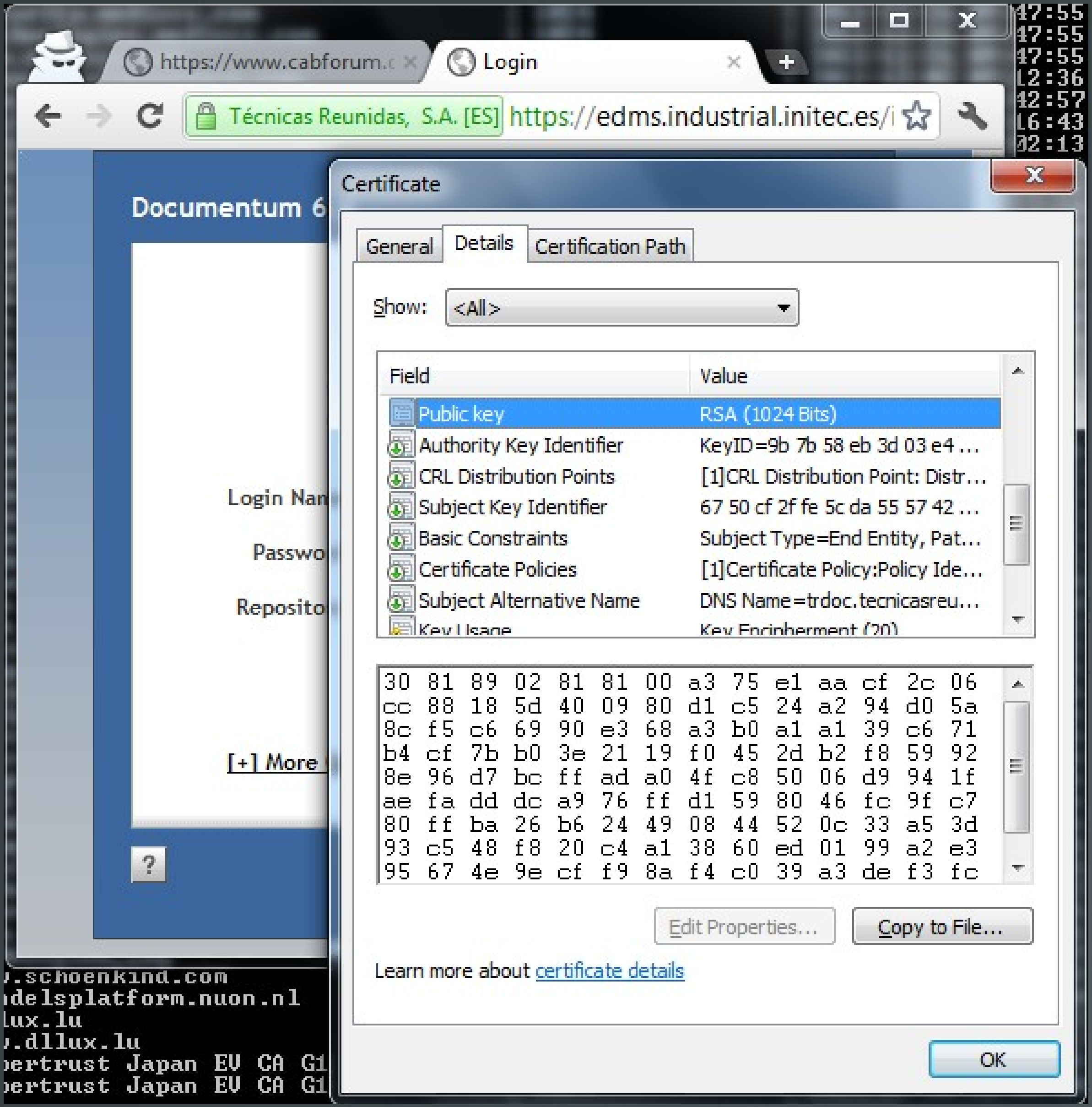
In 2010, the EFF reported that more than 99.6% of the EV Certificates that it was able to check fully complied with the EV Guidelines. Although the CA/Browser Forum demands 100% compliance, this small percentage of problem certificates is extremely encouraging considering that the requirements are fairly complex and extensive. Like many reports that members of the CA/Browser Forum receive, thanks in-part to efforts like the EFF's SSL Observatory, the responsible CA members are able to address and correct problematic practices, not only those prohibited by the EV Guidelines, but also other practices that may weaken the security of the Internet. However, it is also important to note that at no time did the non-compliant certificates previously identified ever pose a security risk to consumers.

For example, the major reason for finding non-compliance during the 2010 EFF review was that the certificates had 1024-bit RSA keys instead of 2048-bit RSA keys, the key length required by

“The CA/Browser Forum has also taken action, requiring that the CAs responsible for the non-compliant EV Certificates examine their other EV certificates for similar problems. The CA/Browser Forum expects all EV certificate issuers to adopt procedures that prevent these types of mistakes.

The issuing CAs reported that the non-compliant certificates have now been revoked and are no longer functional on the web”

There are still some 1024 bit EV certs out there!



Show: <All>

Field	Value
Public key	RSA (1024 Bits)
Authority Key Identifier	KeyID=9b 7b 58 eb 3d 03 e4 ...
CRL Distribution Points	[1]CRL Distribution Point: Distr...
Subject Key Identifier	67 50 cf 2f fe 5c da 55 57 42 ...
Basic Constraints	Subject Type=End Entity, Pat...
Certificate Policies	[1]Certificate Policy:Policy Ide...
Subject Alternative Name	DNS Name=trdoc.tecnicasreu...
Key Usage	Key Encipherment (20)

```
30 81 89 02 81 81 00 a3 75 e1 aa cf 2c 06
cc 88 18 5d 40 09 80 d1 c5 24 a2 94 d0 5a
8c f5 c6 69 90 e3 68 a3 b0 a1 a1 39 c6 71
b4 cf 7b b0 3e 21 19 f0 45 2d b2 f8 59 92
8e 96 d7 bc ff ad a0 4f c8 50 06 d9 94 1f
ae fa dd dc a9 76 ff d1 59 80 46 fc 9f c7
80 ff ba 26 b6 24 49 08 44 52 0c 33 a5 3d
93 c5 48 f8 20 c4 a1 38 60 ed 01 99 a2 e3
95 67 4e 9e cf f9 8a f4 c0 39 a3 de f3 fc
```

Learn more about certificate details

```
17:55
17:55
17:55
12:36
12:57
16:43
02:13
```

```
.schoenkind.com
delsplatform.nuon.nl
lux.lu
.dllux.lu
ertrust Japan EV CA G1
ertrust Japan EV CA G1
```

Observed: 8/11/2011

Revocation!

Revocation is important and problematic

... also quite informative

Using the Observatory to study revocations in the real world

```
SELECT DISTINCT `X509v3 extensions:X509v3 CRL Distribution Points`
FROM valid_certs;
```

-> extract URLs
-> fetch CRLs
-> make a revoked table

(questions/all_crls in the source)

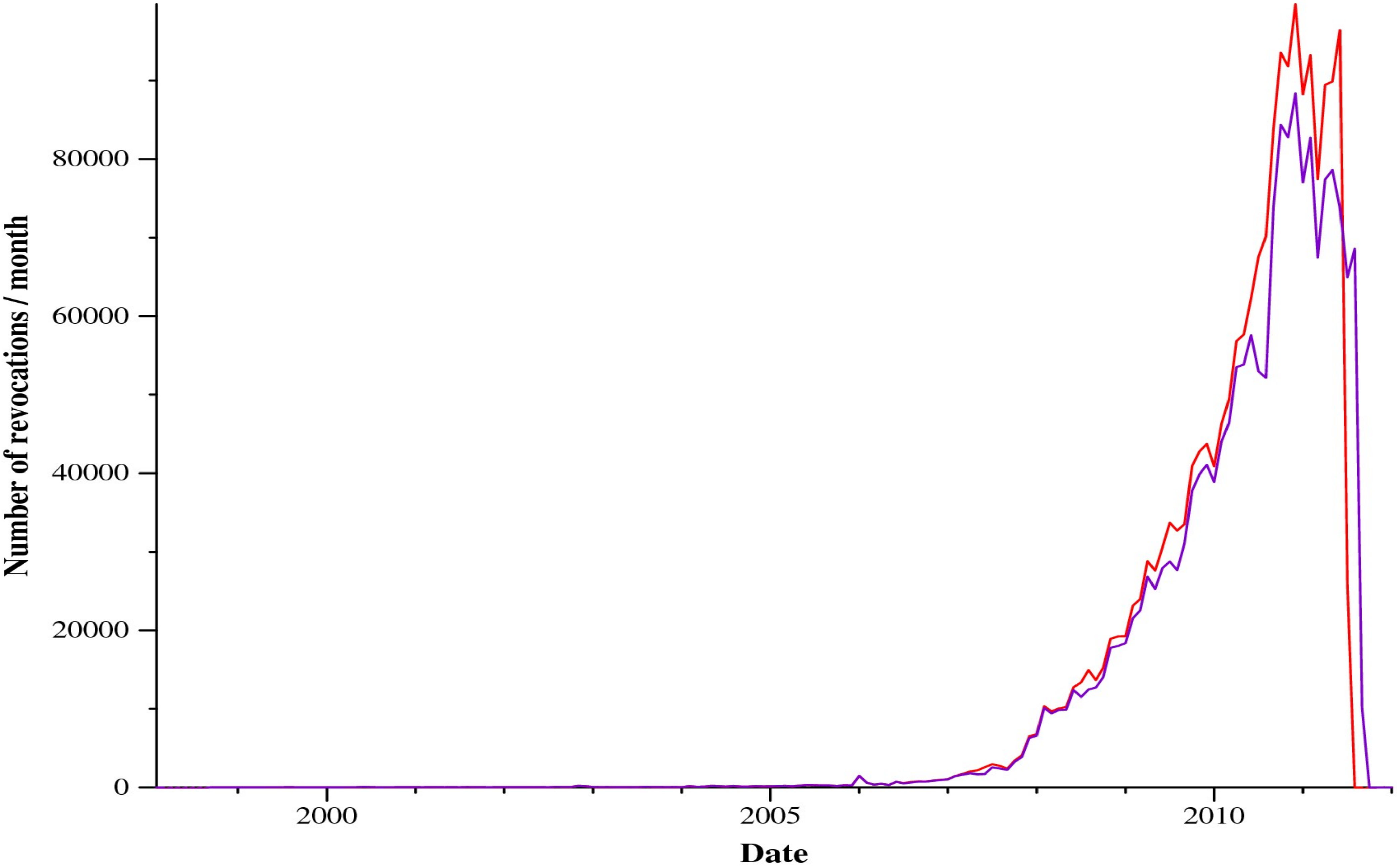
Revocations!

We currently see ~1.96 million revocations
(the number fluctuates)

The BuyPass CA issued 4 revocations in the future
(Nov 2011)

The Certum CA issued 5 revocations at the epoch
(1970)

Two scans of revocations as a function of time

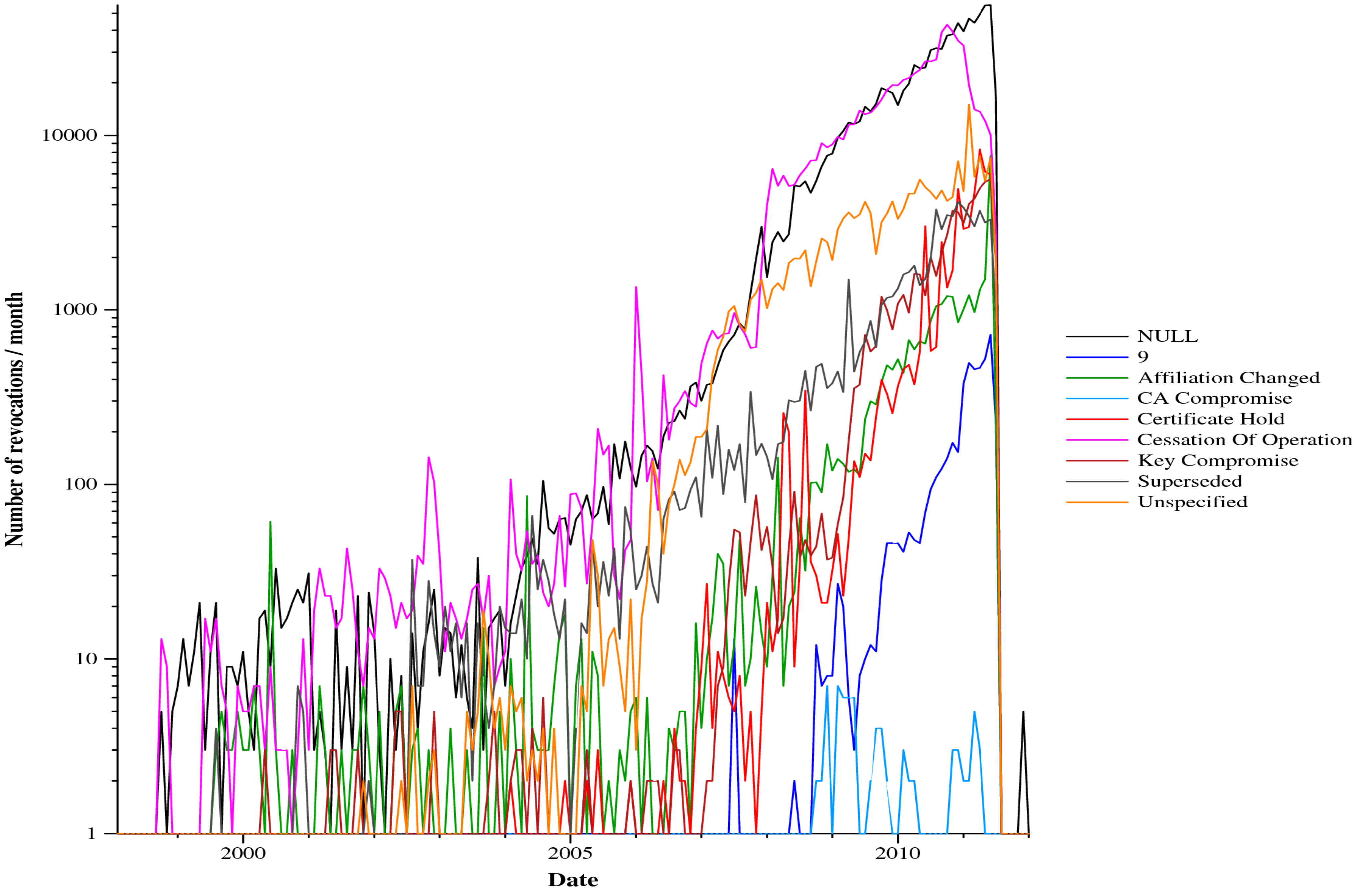


Listed reasons for revocation in CRLs

```
SELECT reason, count(*)
FROM revoked
GROUP BY reason;
+------------------------+----------+
| reason                 | count(*) |
+------------------------+----------+
| NULL                   |   876049 |
| 9                      |     4589 |  -- Privilege Withdrawn
| Affiliation Changed    |    27089 |
| CA Compromise          |       55 |
| Certificate Hold       |    52786 |
| Cessation Of Operation |   700770 |
| Key Compromise         |    59527 |
| Superseded             |    66415 |
| Unspecified            |   174444 |
+------------------------+----------+

-- Thanks for the honesty of those CAs who admitted CA compromise rather
-- than burying it!
```

Listed revocation reasons over time



Revocations are somewhat reassuring...

what about *revocability*?

Revocation Support

Of 1.3+ Million unique valid leaves

683 lack revocation info

all but 1,977 have CRL Distribution Points

and 153,966 have no OCSP information

Revocation Support

Some CAs offer unrevokable certificates

e.g. at https://www.akd.nl/

```
 Truncated issuer name                                   # non-rev leaves
+--------------------------------------------------------------+-------+
|  C=IT, O=I.T. Telecom, OU=Servizi di certificazione, CN=I.T. |   275 |
|  C=US, O=Anthem Inc, OU=Ecommerce, CN=Anthem Inc Certificate |   152 |
|  C=NL, O=DigiNotar, CN=DigiNotar Services 1024 CA/emailAddre |   135 |
|  O=VeriSign Trust Network, OU=VeriSign, Inc., OU=VeriSign In |    88 |
|  C=US, OU=American Express Technologies, ST=NY, CN=American  |     6 |
|  C=NL, O=DigiNotar, CN=DigiNotar Cyber CA/emailAddress=info@ |     5 |
|  C=IT, O=Centro Nazionale per l'Informatica nella PA, OU=Ser |     5 |
|  C=JP, O=Japan Certification Services, Inc., CN=SecureSign P |     5 |
|  O=VeriSign, Inc., OU=VeriSign Trust Network, OU=Terms of us |     3 |
|  C=MY, O=Digicert Sdn. Bhd., OU=457608-K, CN=Digisign Server |     2 |
|  C=MY, O=Digicert Sdn. Bhd., OU=457608-K, CN=Digisign Server |     2 |
|  CN=ACEDICOM Servidores, OU=PKI, O=EDICOM, C=ES              |     2 |
|  C=FR, O=service-public gouv agriculture, OU=0002 110070018, |     1 |
|  C=NL, O=DigiNotar, CN=DigiNotar Services CA/emailAddress=in |     1 |
|  C=US, O=Apple Inc., OU=Apple IST Certification Authority, C |     1 |
+--------------------------------------------------------------+-------+
```

Disused unrevocable CAs

Cybertrust sub CA valid from 2001-09-06.

CPS:

`http://www.us-hosting.baltimore.com/CPS/OmniRoot.html`

(but that is gone!)

No CRL, OCSP, or even country!

The Subject of this cert is:

C=ww, O=global, OU=pki, CN=rootca

Disused unrevocable CAs

That CA signed an sub-CA too – valid from 2002-03-12

Subject is:

C=ww, O=global, OU=pki, CN=issuingca

Again dead CPS: http://www.cwsecurity.net/

4 expired certs observed below this CA

So, how do we fix this mess?

Some proposed mitigations

Consensus measurement
(Perspectives & Convergence.io)

More vigilant auditing
(Decentralized Observatory)

DNSSEC + DANE

Certificate Pinning via HTTPS headers

PKIX attack surface

Compromise | Malice | Compulsion

at

~600 CAs | target site | DNS

or anywhere on the network in between :(

PKIX -> DNSSEC?

Compromise | Malice | Compulsion

at

~600 CAs | target site | DNS

or anywhere on the network in between :(

PKIX -> DNSSEC?

Compromise | Malice | Compulsion

at

~~~600 CAs~~ | target site | DNS

or anywhere on the network in between :(
~~~

PKIX -> DNSSEC?

Compromise | Malice | Compulsion

at

~~~600 CAs~~ | target site | DNS

or anywhere on the network in between :(?
~~~

The biggest win from DNSSEC *could* be simplified TLS deployment

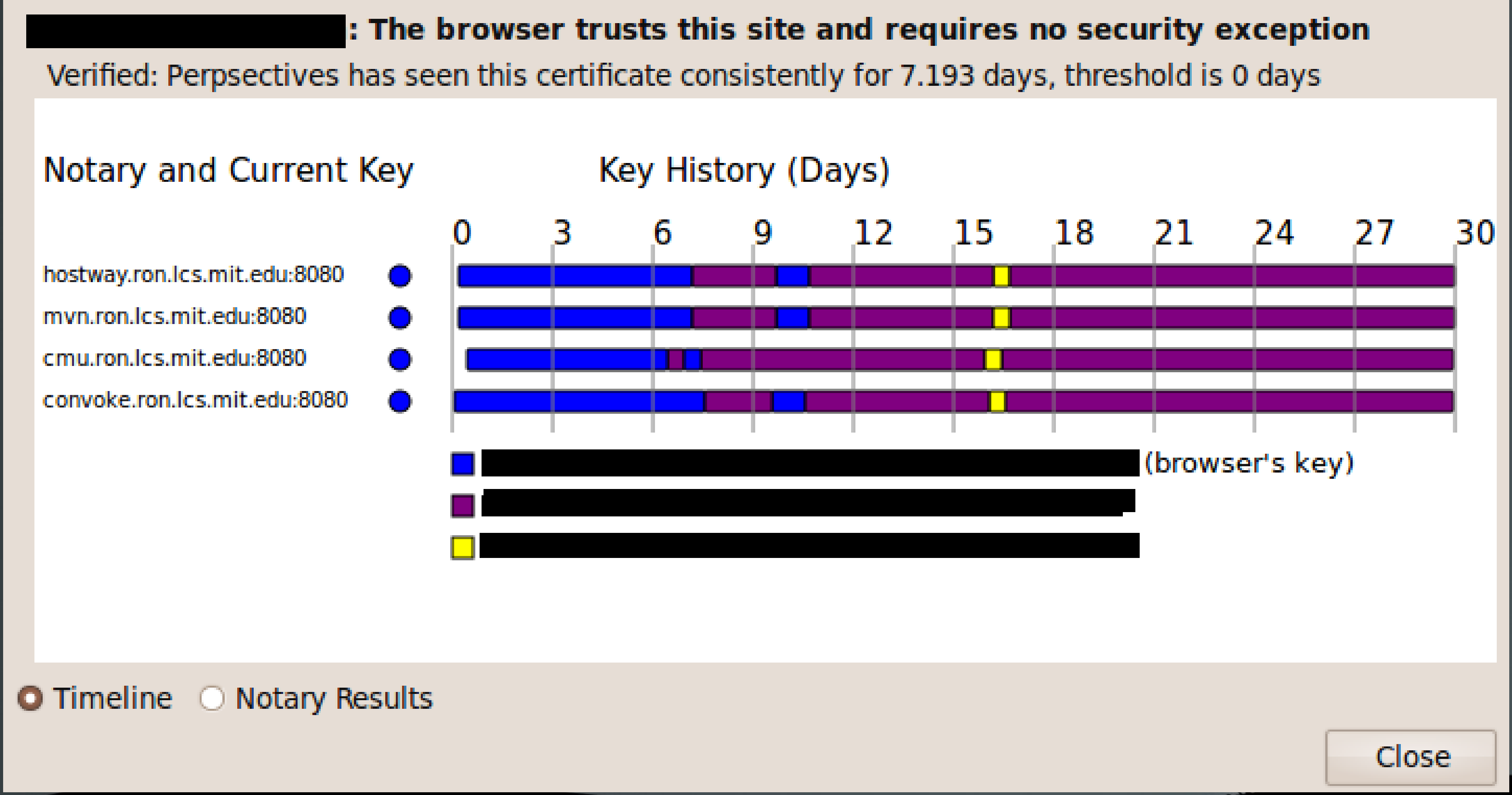
What to do about bit.ly or google.ae?

We would probably need a DNSSEC Observatory!

Persectives, Convergence.io and other ”consensus” approaches

A nice idea but...

The problem with consensus is false positive warnings



[redacted]: The browser trusts this site and requires no security exception

Verified: Perpsectives has seen this certificate consistently for 7.193 days, threshold is 0 days

Notary and Current Key

Key History (Days)

hostway.ron.lcs.mit.edu:8080 ●

mvn.ron.lcs.mit.edu:8080 ●

cmu.ron.lcs.mit.edu:8080 ●

convoke.ron.lcs.mit.edu:8080 ●

🟦 [redacted] (browser's key)

🟪 [redacted]

🟨 [redacted]

● Timeline ○ Notary Results

Identity pinning

Idea: whoever used to be domain.com
should stay domain.com

Much simpler than DNSSEC,
bigger security win,
if it is implemented correctly

The right way to pin

Create a ”private CA” just for this domain

Use that *in parallel* to PKIX

Unfortunately

Ironically, cross-signing of leaves not supported by X.509!

(X.509 assumes one Issuer per leaf cert)

will require something new...

Cross-signing for pinning

Possible solutions:

A second leaf cert signed by the pinned ”private CA” key

A magic X.509 extension with a cross signature (possibly in a randomly appended cert in the chain)

PKIX -> Pinning?

Compromise | Malice | Compulsion

at

~600 CAs | target site | DNS

or anywhere on the network in between :(

PKIX -> Pinning?

Only on first connection

Compromise | Malice | Compulsion

at

~600 CAs | target site | DNS

or anywhere on the network in between :(

FIN