# An Observatory for the SSLiverse

Peter Eckersley, Jesse Burns

Defcon 18, Las Vegas, USA

July, 2010

# Quick overview

- Electronic Frontier Foundation, funded by NL Net
  - – with volunteer help from iSEC Partners
- Collected x.509 Certificates used for HTTPS on the internet
- Looked for odd behavior, checking up on CAs
- Identified “trusted” intermediaries – foreign, security agencies, companies
- Weird, wonderful and suspicious certificates found
- Noted interesting behaviors of servers & clients
- Will be opening data for further review

# Agenda

- Why we need an HTTPS Observatory
- Data Collection Technique
- Results Summary
- Interesting Questions
- Vulnerabilities
- Conclusions
- Future work

# Why We Need an Observatory

# Why We Need an HTTPS Observatory

HTTPS is a rather important protocol!

“Certificate **Authority**”

The words cry out for accountability & transparency

Several recent exploits based on CA mistakes

- Trust model: 1 of *N* CAs (*N* is large)
- Just how large is *N*, exactly?
- Who are these CAs we trust & what’s going on?

# How do we get an HTTPS Observatory

Let's download all the SSL certificates and build a dataset that everyone can study.

(ideally, on an ongoing basis)

# Data Collection Techniques

# Observatory Infrastructure

- Collection:
  - Three low end Linux servers with only 2GB ram
  - Good, shared 100Mbs network connection
  - NMap with poor timings, some python
  - 2-3 months worth of patience
- Analysis:
  - 1 year old i920 server with a new fast disk and 12G ram
  - 2 little laptops
  - Lots of crazy scripts, OpenSSL and a database
  - OpenSSL
- Currently vaporware:
  - Distribution (coming soon)
  - Some web query forms
  - Full datasets (via BitTorrent)

# 1. Observe the SSLiverse

- NMAP Internet for hosts listening on tcp 443
  - Distribute, resume, chaotically permute
  - Work units of the form 157.*.*.15
  - Remember who replies

- Python Client
  - Connect with custom client, send SSL Hello
  - Collect whole certificate chain from server
  - Drops connection pre-key exchange
  - And the other random garbage they say

# 2. Extract the certificates

Custom client used python and Construct

Based on the RFCs definitions

- Still needed to be tuned with Wireshark & test cases

Only need parts of TLS:

- Handshake type, Protocol Version, HelloRequest, ServerHello, Certificate, ASNCert, Handshake, ContentType, TLSRecord, Random, CompressionMethod, a funny unsigned 24 bit big-endian length

Result: lots of X.509 Certificates

# X.509 Aside

- Designed in 1980s
- By: International Telecommunications Union
- Advantages: extremely flexible & general
- Disadvantages: extremely flexible & general

  extremely ugly

- Also: so many security features that the interactions between them are hard to understand

# 3. Parsing X.509 certificates

How do you parse an X.509 certificate?

- No right way to do it
  - Might want quirky side effects, nulls, charset conversions

Effective, wrong ways are easily identified:

- Parse the output of **openssl x509 -text** prettyprinter
- Yup… gross but it gives you useful data quickly

Other interesting ways

- Use Java's certificate parser
- Use openssl's many obscure parsing facilities
- Custom library

# 4. Analysis

- Stick all the data into MySQL tables
- Build new ones for things like domain <-> cert
- Interesting questions become fancy SQL queries
- Handles the complexity of X.509

Validity

- Crucial concept
- Not easy to measure
- More on this later

# Results Summary

16.2M IPs were listening on port 443

10.8M started an SSL handshake

4.3+M used valid cert chains

1.3+M *distinct* valid leaves

# Crash X.509 Certificate Course

Key usage says your SSL cert != a CAs

Certs need to chain back to trust roots,

- Issuer == Subject
- If AKID or SKID in either cert AKID == SKID
- Valid dates
- Key usage is right
- No 'critical' properties we don't understand

# Valid vs Invalid certs

There is all sorts of crazy stuff in the set of invalid certs

- People pretending to be Microsoft, Google, *, etc…
- Some telcos with wildcard certs for their WAP gateways
- You name it, it's there

Unless otherwise noted, this talk is about the valid certs…

# Interesting Questions

How many CAs are there?

Who are they?

What do they sign?

Server impersonation attacks?

# Number of Trusted CAs

How many does your browser trust?

Mozilla: 124 trust root s (~60 organizations)

Microsoft: lists only 19 trust roots in Windows 7

- Silent on-demand updating!
- Can make this 300+ certs
- 100+ from controlling organisations

# Number of trusted certificate signers?

We observed:

1,482 CA Certificates trustable by Windows or Firefox

1,167 distinct issuer strings

651 organizations

but ownerships & jurisdictions overlap

If a CA can sign for one domain, it can sign for any domain

# CAs

Recorded 1,377,067* unique, valid leaf certs

300,224 – signed by one GoDaddy cert

FD:AC:61:32:93:6C:45:D6:E2:EE:85:5F:9A:BA:E7:76:99:68:CC:E7

244,185 – signed by one Equifax cert

48:E6:68:F9:2B:D2:B2:95:D7:47:D8:23:20:10:4F:33:98:90:9F:D4

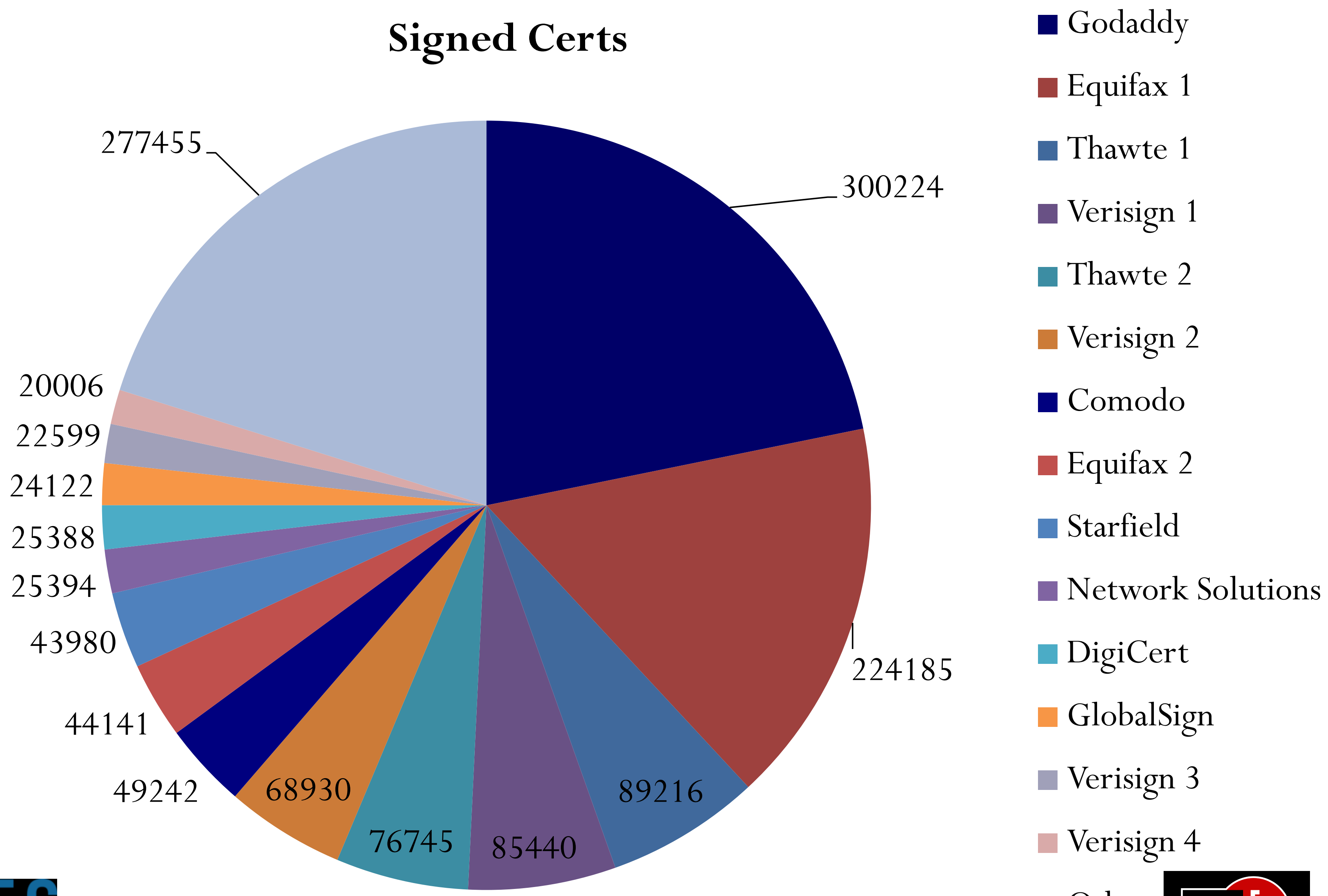
89,216 – signed by Thawte's skid free cert

85,440 – signed USERTRUST's 4 certs w/ skid

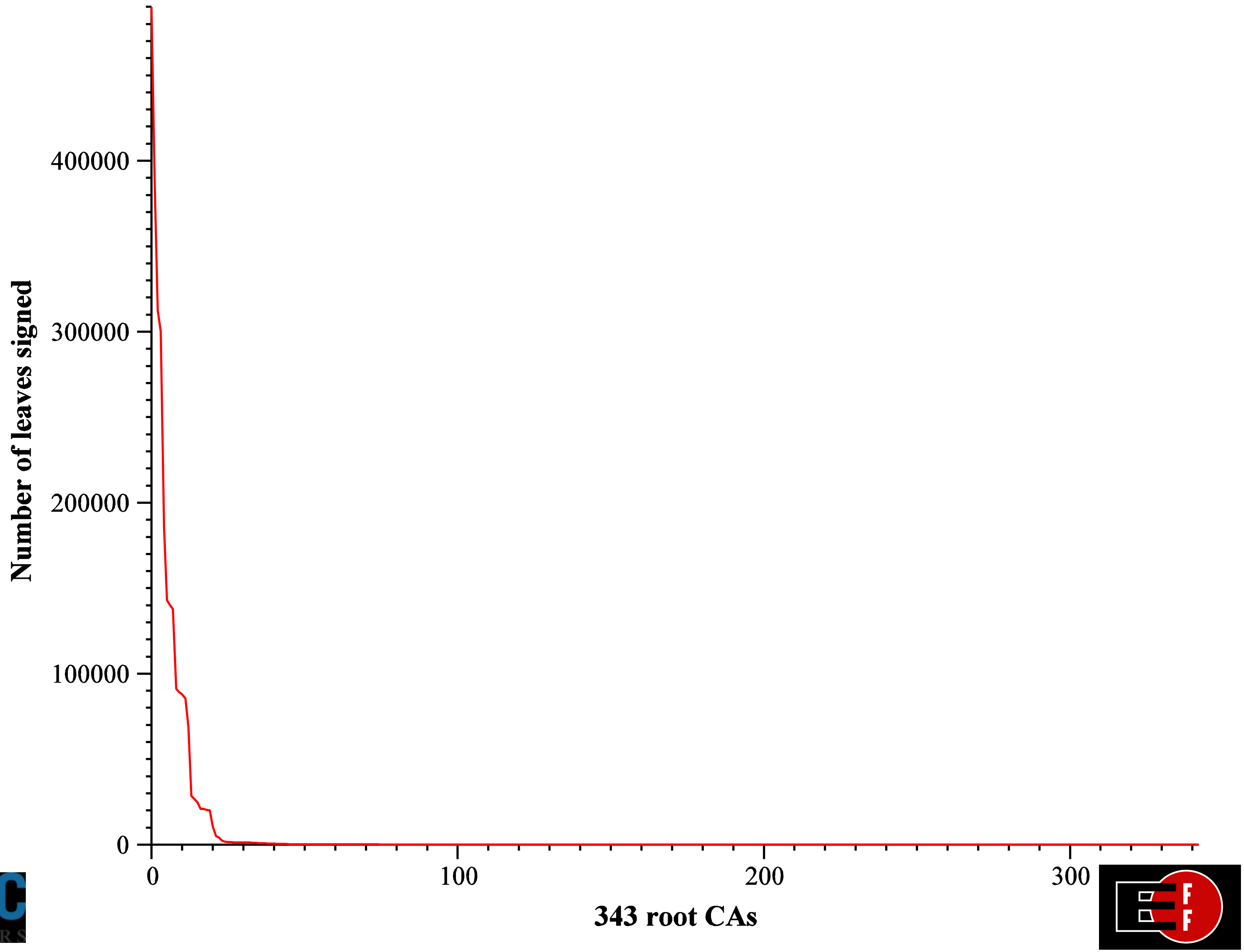
A1:72:5F:26:1B:28:98:43:95:5D:07:37:D5:85:96:9D:4B:D2:C3:45

- Valid based on OpenSSL 0.9.8k with Firefox **or** all XP i.e. trust roots…

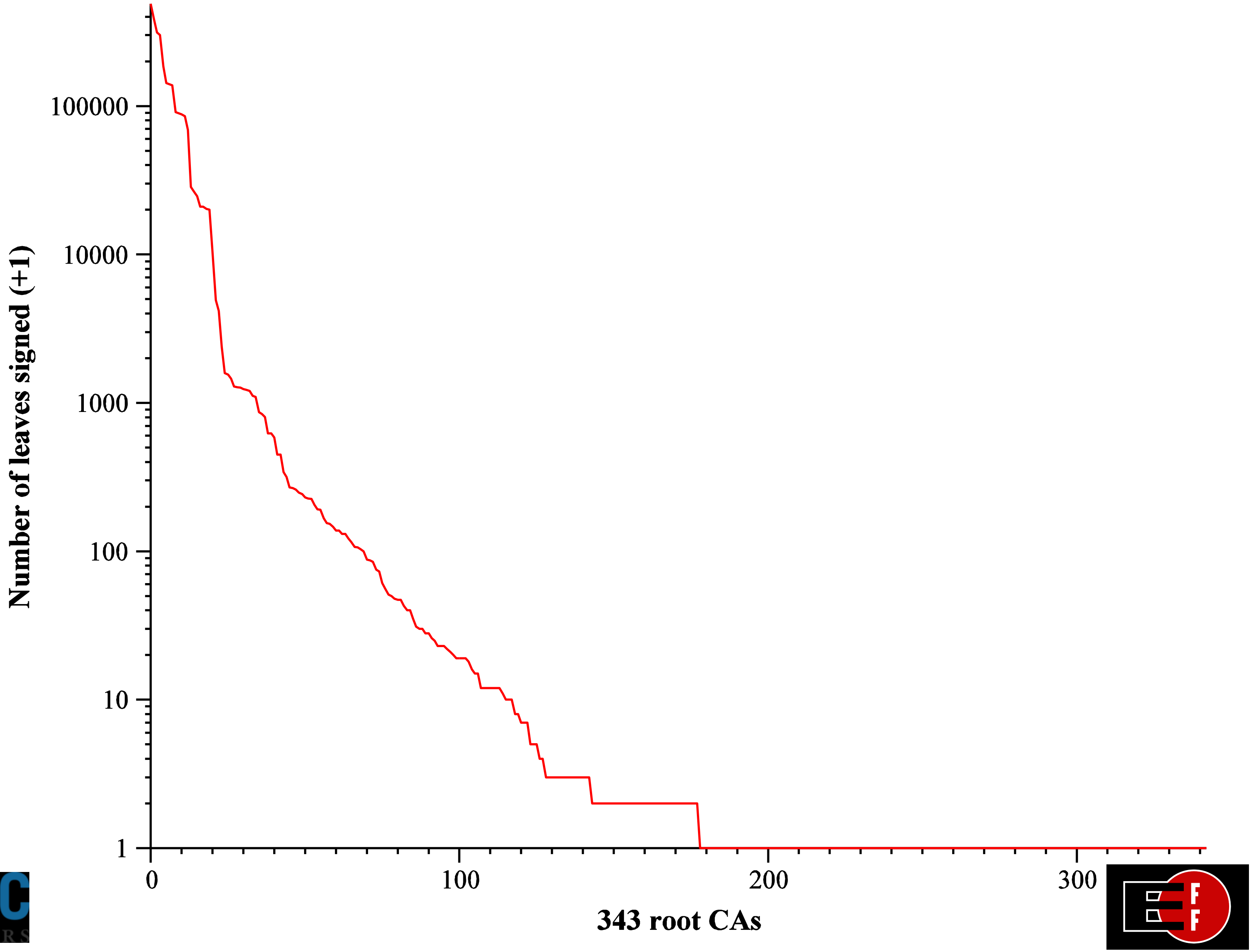
# CA Certificate use frequency



# # Leaves validated per Root CA



# # Leaves validated per Root CA



# CA Usage

When might a root be legitimately unused?

- New, more secure cert being pushed out
  - Needs to be accepted widely before it can be used
  - Obviously legitimate, and improves overall security
- Backup root – maybe if a root needed revoking?!?

When might a subordinate CA be legitimately unused?

- Hard to imagine hey
  - If you want a more secure one, make it
  - If you get compromised revoke and make a new one
  - Maybe some argument around how long that takes?

# Valid CA Certs Sharing Keys

Many signing certificates share keys!

Identified 80 distinct keys used in multiple CA Certs

Most widely reused, valid Public RSA key:

- Verisign, 2006 2048-bit key
- Certs share subject, lack subject or authority ids
- 4 expire simultaneously in 2021, 1 expires in 2036

# Valid CA Certs Sharing Keys

Some keys are shared between organizations

mergers or acquisitions?

Certificate 1, a 2048-bit RSA, CA signing certificate

American Optimum SSL CA

```
E0:E6:09:81:CF:00:78:0D:13:FE:61:6B:01:DC:0C:A5:17:61:F8:EF
```

Certificate 2

UK Comodo CA, CN=OptimumSSL CA

```
60:87:D7:16:62:34:11:75:62:CE:62:A0:F7:F6:2E:A5:C1:4F:C5:45
```

Simultaneous expiration 2020-05-30 10:48:38

Different start dates, same SKID, AKID & key usage

# Valid CA Certs Sharing Keys

Certificate 1, a 2048-bit RSA, CA signing certificate

UK Comodo CA Limited, CN=PositiveSSL CA

```
DD:C5:8C:53:DF:2E:F2:B2:66:20:BF:1C:A7:D4:15:FF:98:CD:B4:84
```

Certificate 2 Issuer same as 1: US USERTRUST

US Positive Software Corporation, CN=LiteSSL CA

```
93:D7:BC:5C:CC:3A:B6:DB:09:CA:49:6F:25:81:AA:65:7F:16:96:20
```

Certificate 3 – No AKID, Issuer: Swedish AddTrust

US Positive Software Corporation, CN=LiteSSL CA

```
A8:99:38:62:1C:B3:76:17:80:FD:33:7E:E8:85:90:64:2B:37:26:2A
```

1 & 2 expire 2020-05-30 10:48:38, 3 expires 10 months earlier

2&3 share start dates, same SKID key usage

Over 44K certs using this key ID

# CA Certs Sharing Keys to delay expiration

Certificate 1, a 1024-bit RSA, CA signing certificate

Israeli ComSign Ltd.

```
62:E7:8E:92:BF:CA:C0:CD:FA:90:34:1B:F6:27:F7:36:1D:D7:AA:F2
```

Certificate 2, basically identical aside from dates

Israeli ComSign Ltd.

```
29:F4:B6:CC:16:5E:EB:60:CF:DC:95:C9:81:DC:E6:7E:71:28:15:10
```

1 expires 2014-06-14 14:56:31, 2 expire 2020-12-31 21:05:25

41 valid certs with this issuer, **none** expire after 2014-5-29!?

same SKID, AKID, & key usage

Trick adds 2392 days to this key's 6000+ day life.

# CAs signing RFC 1918 (Reserved) IPs

Would the **authentic** 192.168.1.2 please step forward:

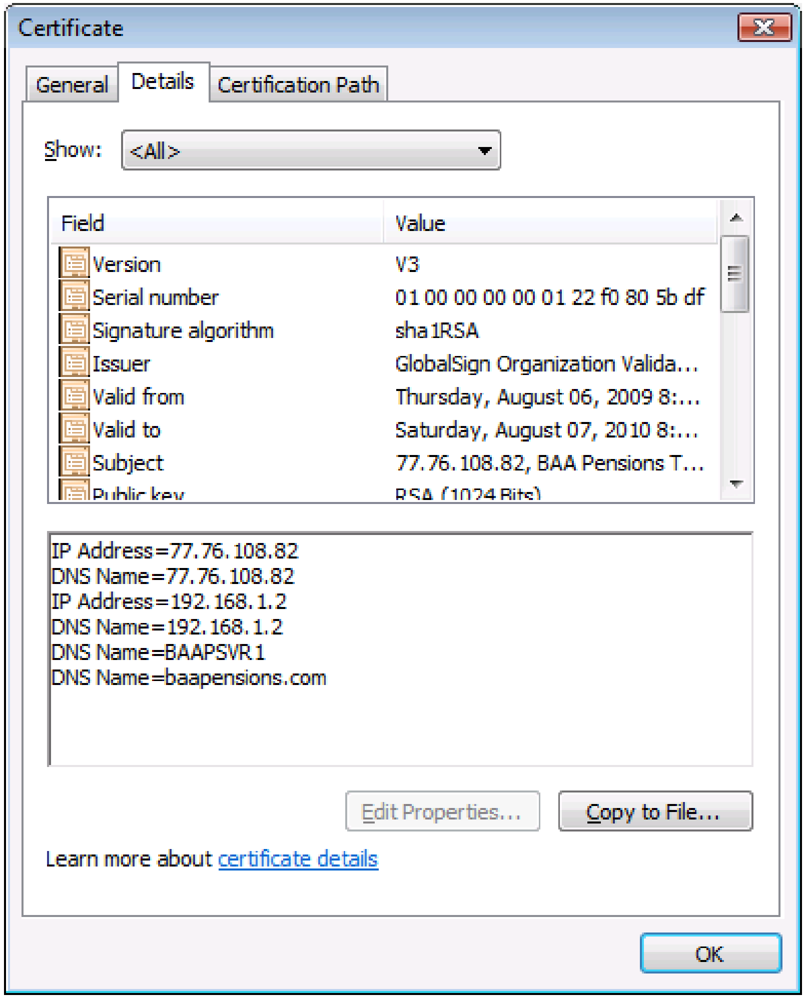
US Equifax asserts it is in Texas

Belgian GlobalSign puts it in:

the US, the UK, Switzerland,

Belgium and cutely also as

77.76.108.82



| Field | Value |
|---|---|
| Version | V3 |
| Serial number | 01 00 00 00 00 01 22 f0 80 5b df |
| Signature algorithm | sha1RSA |
| Issuer | GlobalSign Organization Valida... |
| Valid from | Thursday, August 06, 2009 8:... |
| Valid to | Saturday, August 07, 2010 8:... |
| Subject | 77.76.108.82, BAA Pensions T... |
| Public key | RSA (1024 Bits) |

```
IP Address=77.76.108.82
DNS Name=77.76.108.82
IP Address=192.168.1.2
DNS Name=192.168.1.2
DNS Name=BAAPSVR1
DNS Name=baapensions.com
```

# CAs signing unqualified names

It would be meaningless to assert ownership of such a name

Yet… we saw over 6 thousand unique valid "localhost" certs

From different issuers like:

Comodo , Go Daddy, GlobalSign, Starfield, Equifax, Digicert, Entrust, Cybertrust, Microsoft, and Verisign

Some CAs only signed one "Localhost" name:

Cybertrust, Entrust, Equifax, Microsoft & Verisign

Maybe they have a process to track what they assert?

# Countries use of CAs

Some countries are not using their CAs

Macao – has its own 2048 bit CA in XP

- Isn’t used on the Internet*
- Doesn’t use Chinese or Portuguese CA either
- Signs government websites with commercial certificates from US and UK CAs

* As far as we saw…

# Weak Certs

Two leaf certs

- 508 bit RSA keys – think 512, starting with a 0
- Signed by Equifax and Thawte
- Valid under Mozilla and Microsoft’s trust roots

Fingerprints:

B4:21:9E:89:24:29:41…

7B:BB:1B:CF:FD:6A:1A…

# Vulnerabilities

Yes, a few things pop out when you look.

# Vulnerabilities

Remember the Debian OpenSSL bug?

- Affected keys generated from 2006-2008
- Private keys have only 15-17 bits of entropy
  (i.e. not private)

```
select subject from certs join blacklist on
    sha1(certs.rsa_modulus) = blacklist.hash
```

~ 28K vulnerable certs seen

- Fortunately only 500 are valid
- 12K are private CA certs

# About those vulnerable certificates

530 Validate, 73 of these are revoked

CAs that revoked a lot of vulnerable certs:

- ✓ Starfield (5/5)
- ✓ Comodo (29/30)
- ✓ USERTRUST (24/25)

Some CAs that didn't:

- ✗ Equifax (0/140)
- ✗ ipsca (0/24)
- ✗ Cybertrust (4/125)
- ✗ Thawte (4/35)
- ✗ VeriSign (2/9)
- ✗ Unizeto (0/6)
- ✗ FNMT (0/6)

# Certificates that should not exist

```
select `X509v3 Basic Constraints:CA`, subject
          `X509v3 Key Usage`, from valid_certs
where
 (locate("Certificate Sign", `X509v3 Key Usage`)!=0)
   !=  (locate("TRUE", `X509v3 Basic Constraints:CA`
  ) !=0);
```

**CA:** FALSE

**Key Usage:**Digital Signature, Non Repudiation, Key Encipherment,Data Encipherment, Key Agreement, *Certificate Sign*

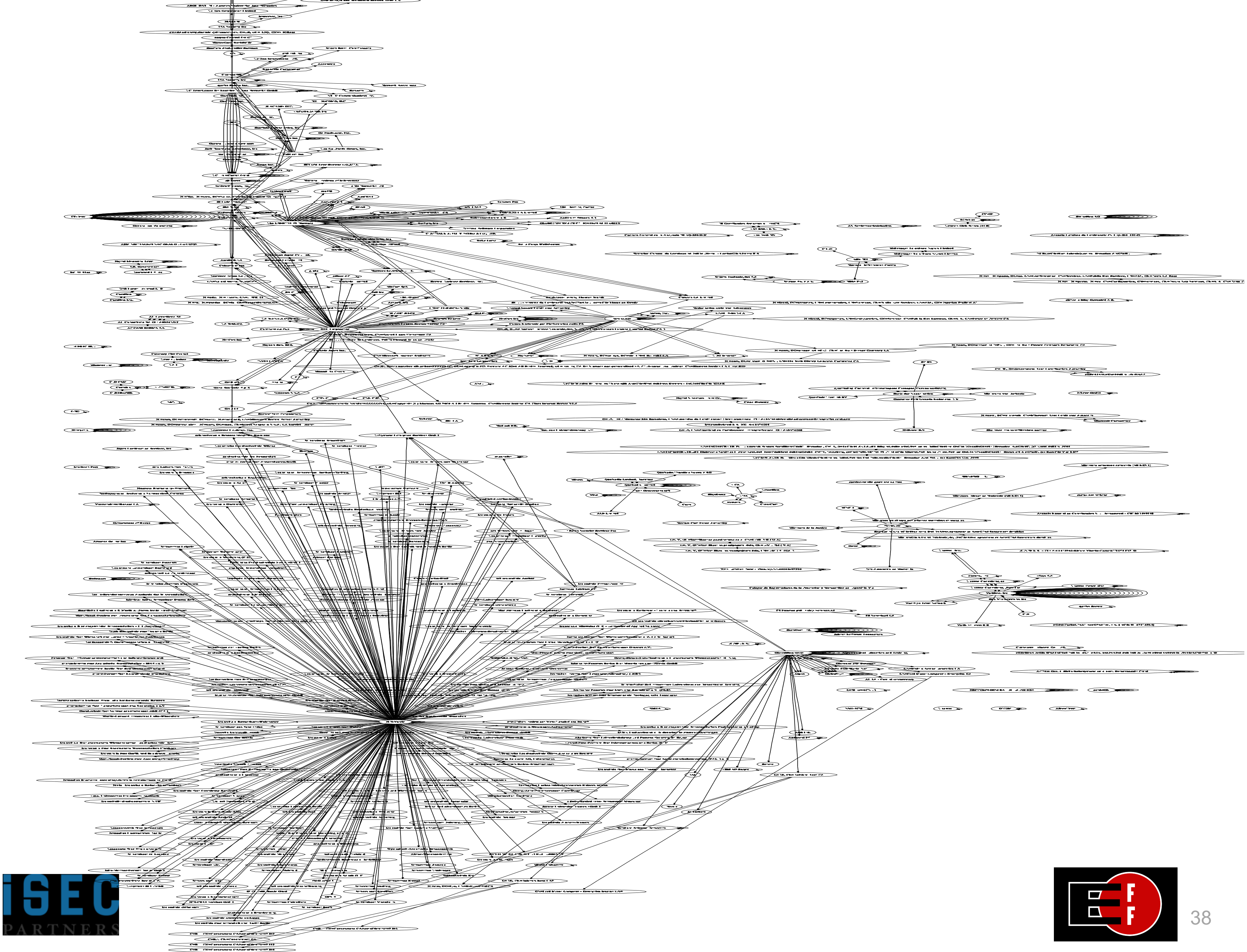
**Issuer:** C=BM, O=QuoVadis Limited,
OU=www.quovadisglobal.com,
CN=QuoVadis Global SSL ICA

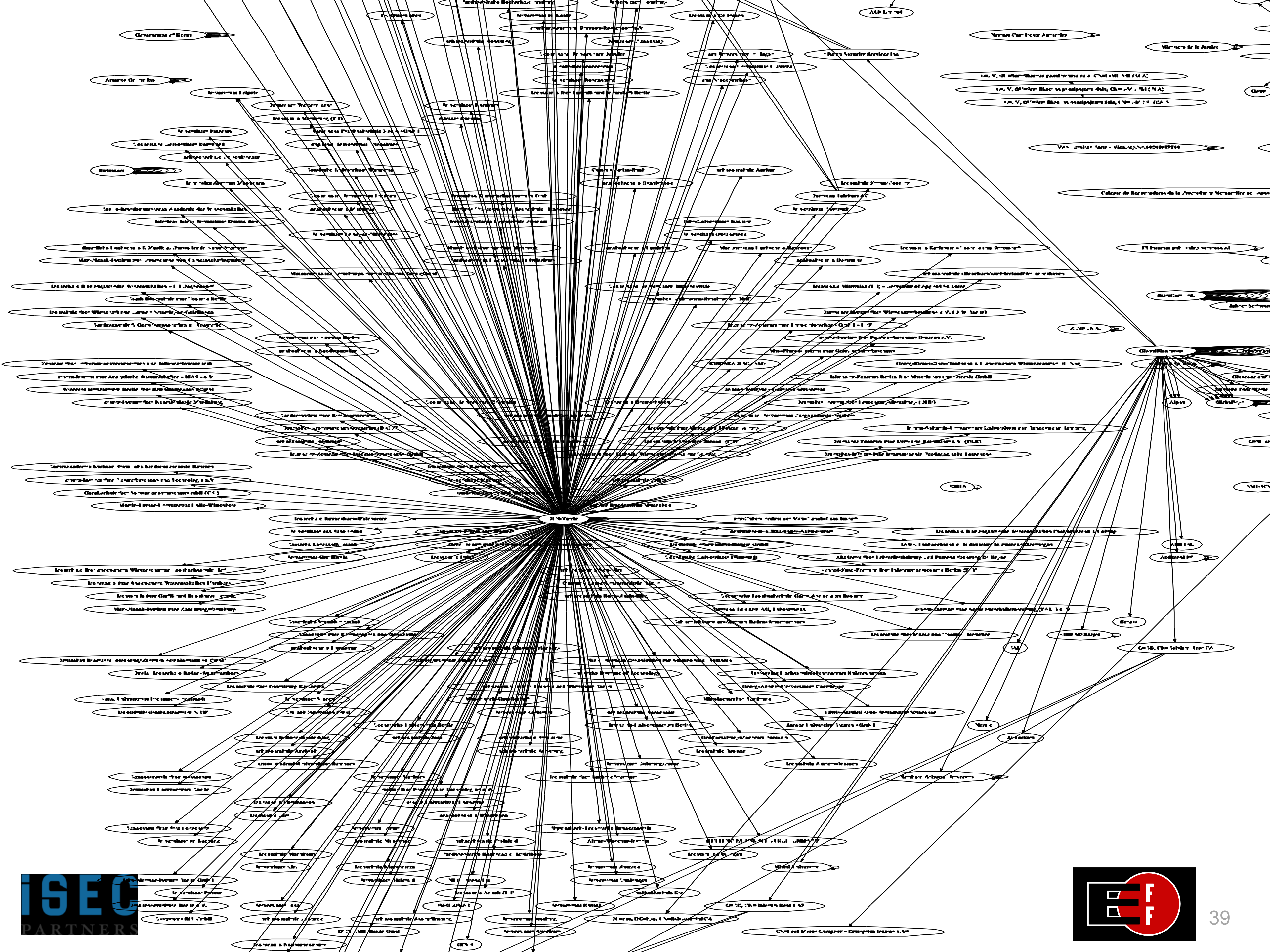
# Pretty Pictures

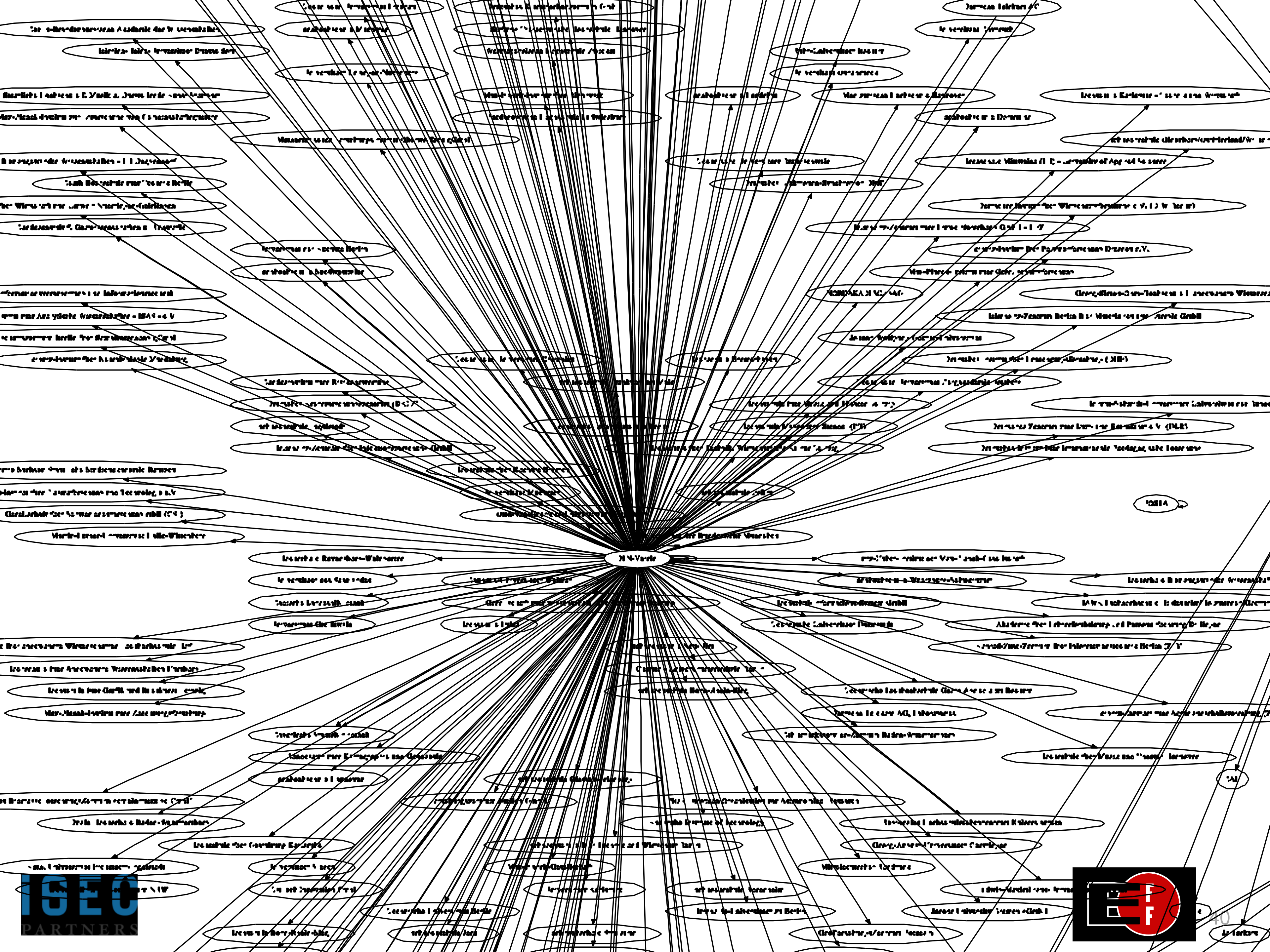
Roots create subordinates

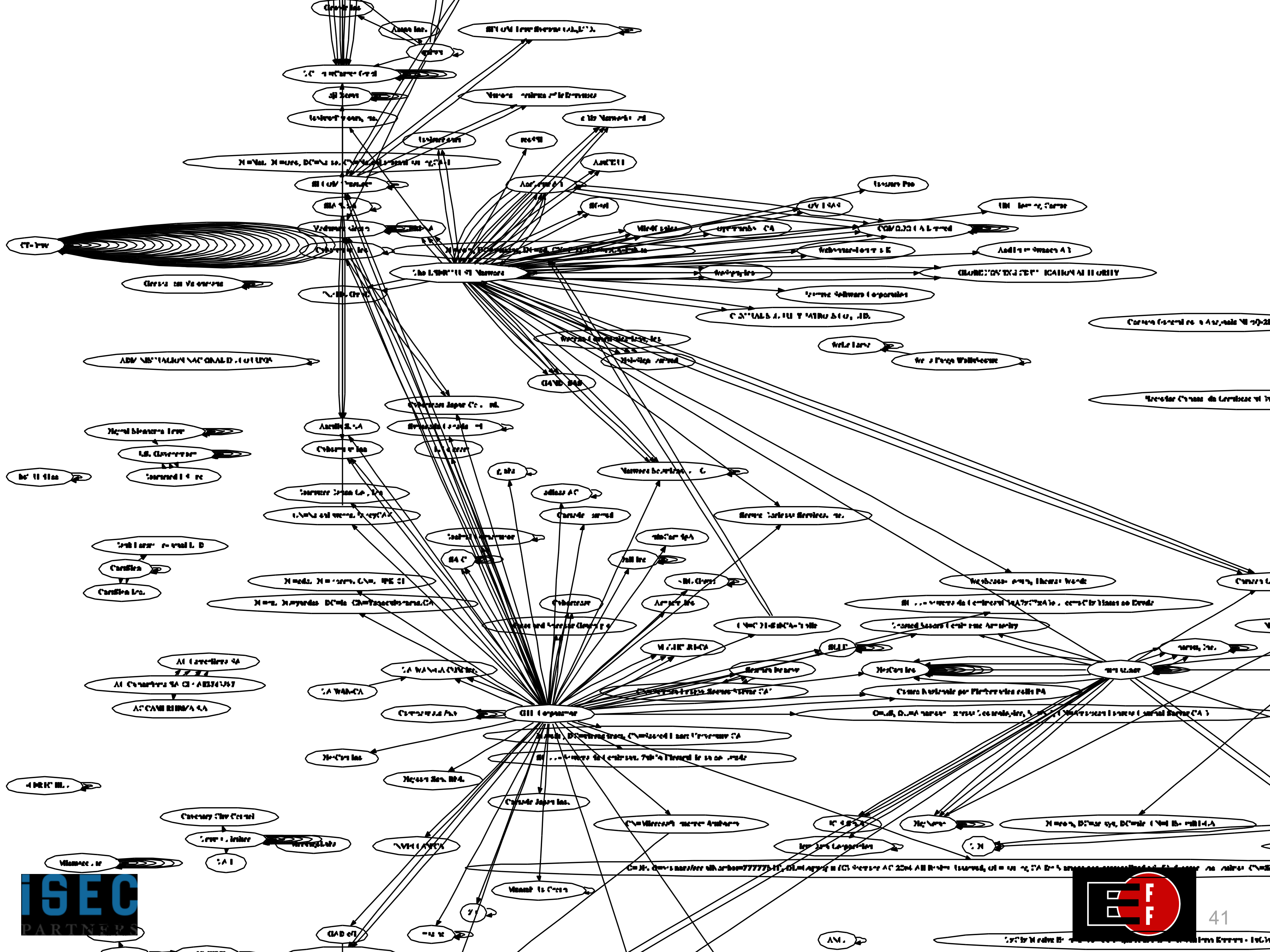
Subordinates create subordinates

A zillion leaves are no good






DFN-Verein
Berlin-Brandenburgische Akademie der Wissenschaften
Max-Planck-Institut zur Erforschung von Gemeinschaftsguetern
Universitaet Stuttgart
Leibniz-Institut fuer Polymerforschung Dresden e.V.
Helmholtz-Zentrum Berlin fuer Materialien und Energie GmbH
Georg-Simon-Ohm-Hochschule fuer angewandte Wissenschaften
Johannes Gutenberg-Universitaet Mainz
Technische Universitaet Chemnitz
Deutsches Zentrum fuer Luft- und Raumfahrt e.V. (DLR)
Martin-Luther-Universitaet Halle-Wittenberg
Max-Planck-Institut fuer Zuechtungsforschung
Akademie fuer Lehrerfortbildung und Personalfuehrung Dillingen
Konrad-Zuse-Zentrum fuer Informationstechnik Berlin (ZIB)
Deutsche Telekom AG, Laboratories
Statistisches Landesamt Baden-Wuerttemberg
Hochschule fuer Musik und Theater Hannover
Karlsruhe Institute of Technology
Georg-August-Universitaet Goettingen
Ludwig-Maximilians-Universitaet Muenchen
Humboldt-Universitaet zu Berlin
Jacobs University Bremen gGmbH


The USERTRUST Network
GTE Corporation
COMODO CA Limited
Cybertrust Japan Co., Ltd.
Cybertrust Inc
Aardlu S.p.A
Network Solutions L.L.C.
Wells Fargo
Wells Fargo WellsSecure
GlobalSign
Canon Japan Inc.
CN=Microsoft Internet Authority
Washington Post
Cybertrust
Certifica
Certifica Inc.
U.S. Government
AC Camerfirma SA
AC CAMERFIRMA SA
Camerfirma SA CIF A82743287
Cybertrust Japan Co., Ltd.
Cybercom Inc
Amazon Inc
Harbour Inc
NetCom Inc
Centro Nazionale per l'Informatica nella PA
Entrust Software Corporation
AddTrust AB
Deutsche Telekom AG
Digital Signature Trust
Secure Business Services, Inc.
CT-Trust
ENTRUST
Network Solutions
GANDI SAS
GlobalSign Interactive Communications, Inc
AOL

# Subordinate CAs

Interesting Subordinate CAs:

- Department of Homeland Security
- CNNIC from 2007, removing that root helps you how?
- Etisalat
- Booz Allen Hamilton
- Gemini Observatory – Can I have a CA?
- Companies: Dell, Ford, Google, Marks and Spencer, Vodaphone…
- Hundreds more….

# Subordinate CAs

Countries with valid CAs: 46

- USA, South Africa, The UK, Belgium, Japan, Germany, The Netherlands and Israel

lots more

Countries without CAs but with Subordinate CAs:

- United Arab Emirates, Iceland, Luxembourg, Macedonia, Malaysia, Russian Federation,

* 64 roots didn’t include a country – probably US based

# Unwashed Self-Signed Masses

Argument for persistence of key, TOFU, or ssh model.

- Trusted introducer is nice, but some want to skip it
- Reduced complexity, cost
- More security if CAs sign – say random subordinates

X509 isn't simple however:

- What name is a self-signed cert valid for?

IE, Firefox and Chrome track the site a self-signed cert is for

Firefox lets you track permanent assumptions about these.

Substituting trust-chained different cert allowed

# Conclusions & Discussion

Is the CA model fundamentally broken?

Can we do any better?

Are we observing middleperson / server impersonation attacks?

# Future Work

Release our data

Detecting private attacks and non-public addresses

Consider an analysis of CA importance

# Special thanks to sponsor NL Net

## Thanks to:

Chris Palmer, Christopher Soghoian, Seth Schoen, Jennifer Granick, Andy Steingruebl, Jeff Hodges, Jacob Appelbaum, Len Sassaman for suggestions, advice and support.

## Other iSEC Contributors:

- Pavan Walvekar – raw name analysis
- Eray Ozkan – provided MS roots

iSEC PARTNERS