



The sorry state of X.509

**from certification weaknesses
to Man-in-the-middle detection**

Ralph Holz

Network Architectures and Services
Technische Universität München

15-16 November 2012



About the speaker

- PhD student at Technische Universität München, Germany
- PKI background - measurement and analysis of X.509 and OpenPGP
- Also been involved in protocol design and P2P security



Agenda

- The SSL Landscape
- Proposals to enhance or replace X.509
- Crossbear: Detecting and Localising the MitM



The SSL Landscape



SSL/TLS

- The backbone protocols for securing the WWW
- Authentication, confidentiality, integrity
- Public-key cryptography

X.509: Public Key Infrastructure standard

- Certification Authorities (CAs) certify Web sites
- Non-forgable signature on (*identity, public key*)




Browser Panic (www.symantec.de)

SSL Error - Google Chrome

https://www.symantec.de

Conference Rec... Universitätsbiblio...

 **This is probably not the site you are looking for!**

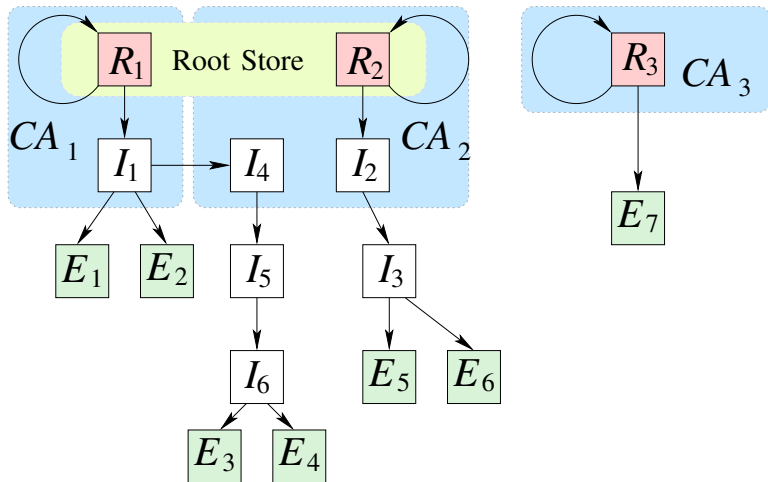
You attempted to reach www.symantec.de, but instead you actually reached a server identifying itself as **symantec.com**. This may be caused by a misconfiguration on the server or by something more serious. An attacker on your network could be trying to get you to visit a fake (and potentially harmful) version of www.symantec.de.

You should not proceed, **especially** if you have never seen this warning before for this site.

▶ [Help me understand](#)

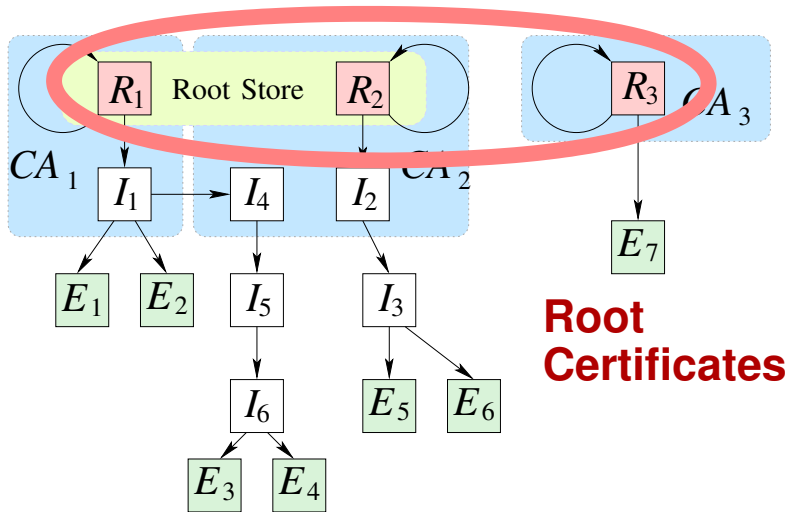


Basic Idea of PKI



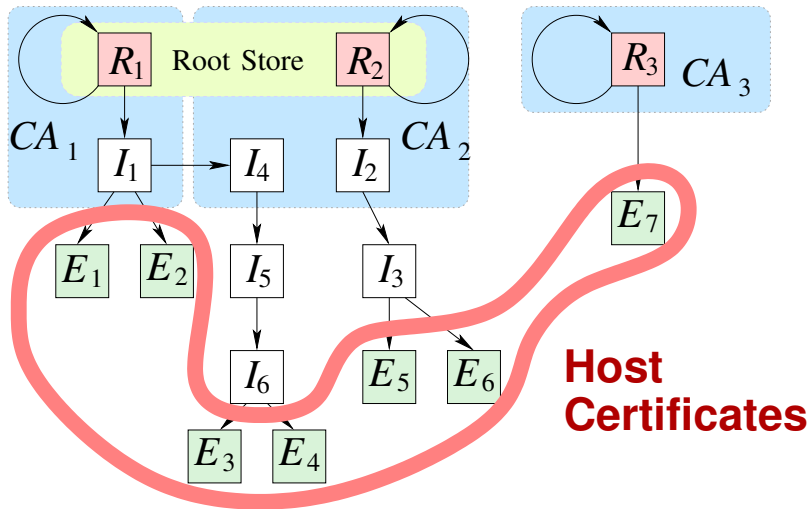


Basic Idea of PKI



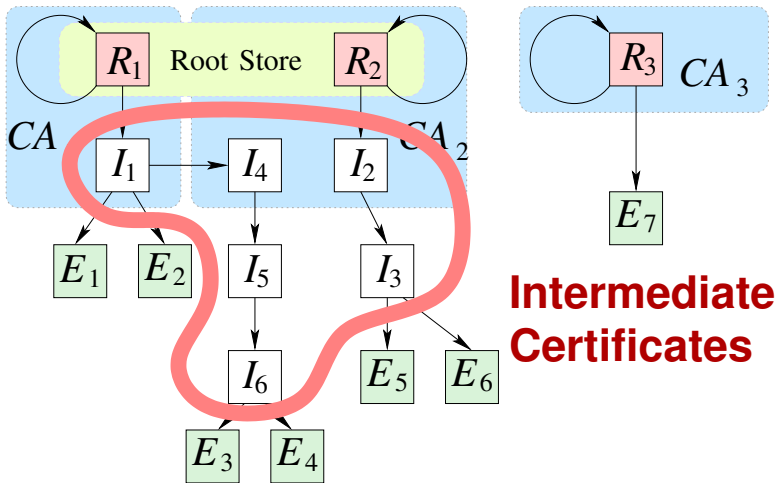


Basic Idea of PKI



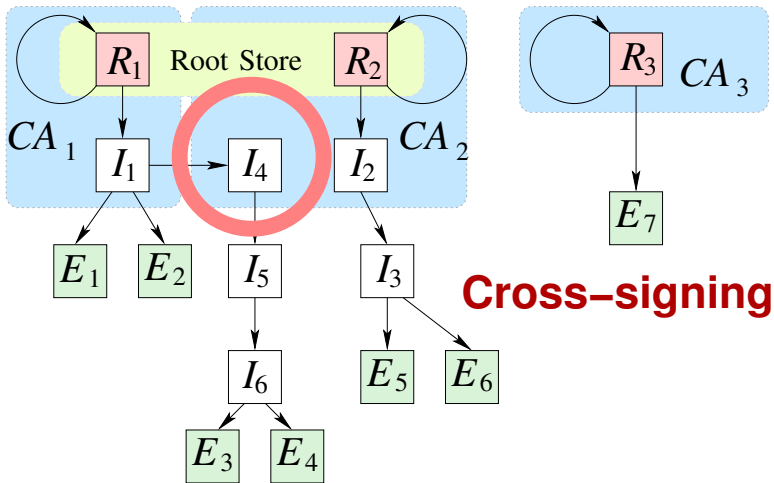


Basic Idea of PKI



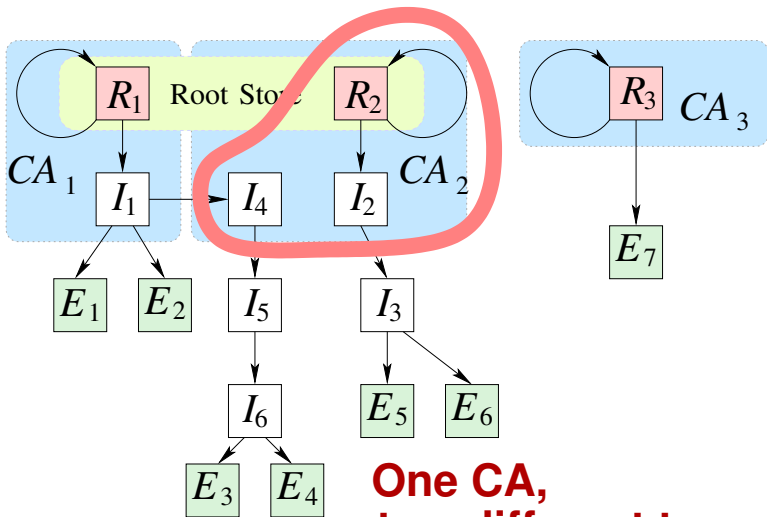


Basic Idea of PKI





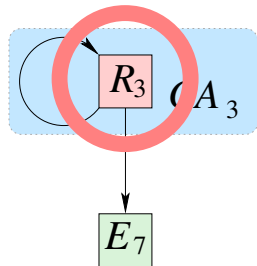
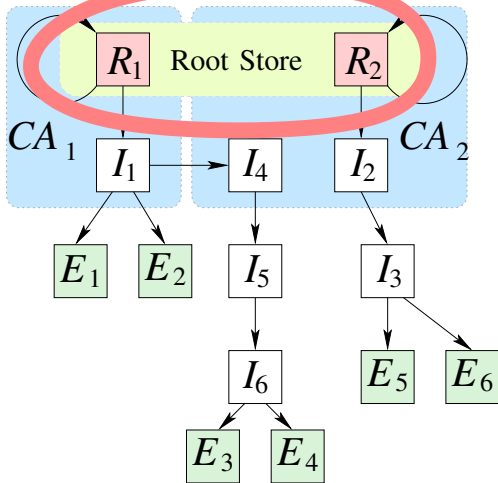
Basic Idea of PKI



**One CA,
two different trees**



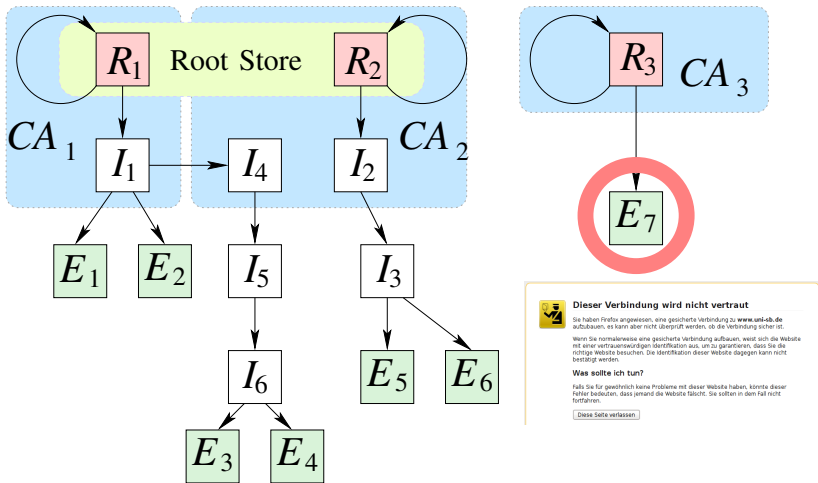
CA in Root Store



CA not in Root Store



Root certificate not in Root Store





An X.509 Certificate

X509v3 Certificate		
Version	Serial no.	Sig. algo.
Issuer		
Validity	Not Before	Not After
Subject		
Subject Public Key Info		
	Algorithm	Public Key
X509 v3 Extensions		
	CA Flag, EV, CRL, etc.	
Signature		



PKI weaknesses since 2008

- Early December 2008:
 - 'Error' in Comodo CA: no identity check
 - Whitehack hacks StartSSL CA
- February 2009
 - 'Easy' attack on MD5: fake CA certificate
- March 2011: Comodo CA hacked
 - Blacklisting of \approx 10 certificates
- July 2011: DigiNotar CA hacked
 - 531 fake certificates *in the wild*



PKI weaknesses since 2008

- Early December 2008:
 - 'Error' in Comodo CA: no identity check
 - Whitehack hacks StartSSL CA
- February 2009
 - 'Easy' attack on MD5: fake CA certificate
- March 2011: Comodo CA hacked
 - Blacklisting of ≈ 10 certificates
- July 2011: DigiNotar CA hacked
 - 531 fake certificates *in the wild*



PKI weaknesses since 2008

- Early December 2008:
 - 'Error' in Comodo CA: no identity check
 - Whitehack hacks StartSSL CA
- February 2009
 - 'Easy' attack on MD5: fake CA certificate
- March 2011: Comodo CA hacked
 - Blacklisting of ≈ 10 certificates
- July 2011: DigiNotar CA hacked
 - 531 fake certificates *in the wild*



PKI weaknesses since 2008

- Early December 2008:
 - 'Error' in Comodo CA: no identity check
 - Whitehack hacks StartSSL CA
- February 2009
 - 'Easy' attack on MD5: fake CA certificate
- March 2011: Comodo CA hacked
 - Blacklisting of \approx 10 certificates
- July 2011: DigiNotar CA hacked
 - 531 fake certificates *in the wild*

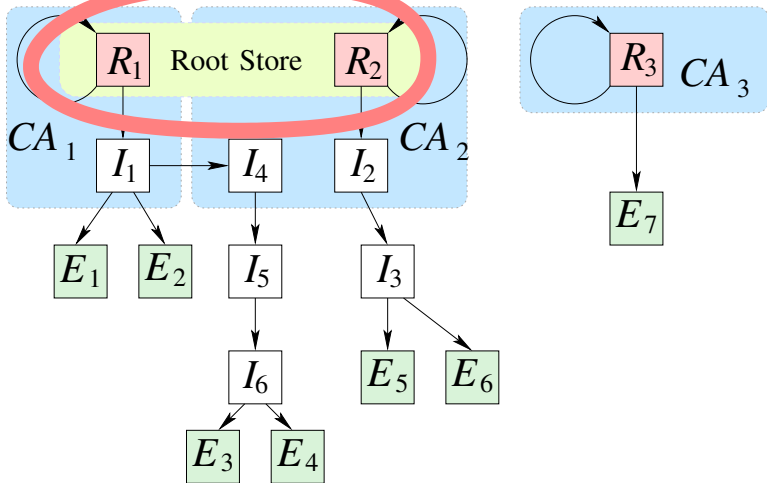


PKI weaknesses since 2008

- Early December 2008:
 - 'Error' in Comodo CA: no identity check
 - Whitehack hacks StartSSL CA
- February 2009
 - 'Easy' attack on MD5: fake CA certificate
- March 2011: Comodo CA hacked
 - Blacklisting of \approx 10 certificates
- July 2011: DigiNotar CA hacked
 - 531 fake certificates *in the wild*



CA's in Root Store





Browser (Client) Root Stores

Your browser chooses the ‘trusted CAs’. Not you.

Any CA may issue a certificate for any domain.

This means the weakest CA determines the strength of the whole PKI.



A good PKI should

- ... allow HTTPs on all WWW hosts
- ... contain only valid certificates
- ... offer good cryptographic security
 - Long keys, only strong hash algorithms, ...
- ... have a sensible setup
 - Short validity periods (1 year)
 - Short certificate chains (but use intermediate certificates)
 - Number of issuers should be reasonable (weakest link!)



Active scans to measure *deployed* PKI

- Scan hosts on Alexa Top 1 million Web sites
- Nov 2009 – Apr 2011: scanned 8 times from Germany
- March 2011: scans from 8 hosts around the globe

Passive monitoring to measure *user-encountered* PKI

- Munich Research Network, monitored all SSL/TLS traffic
- Two 2-week runs in Sep 2010 and Apr 2011

EFF scan of IPv4 space in 2010

- Scan of 2-3 months, no *domain* information



Our Data Sets

Active Scans — Passive Monitoring — EFF IPv4 scan

<i>Location</i>	<i>Time (run)</i>	<i>Type</i>	<i>Certificates</i>
Tuebingen, DE	November 2009	Active scan	833,661
Tuebingen, DE	December 2009	Active scan	819,488
Tuebingen, DE	January 2010	Active scan	816,517
Tuebingen, DE	April 2010	Active scan	816,605
Munich, DE	September 2010	Active scan	829,232
Munich, DE	November 2010	Active scan	827,366
Munich, DE	April 2011	Active scan	829,707
Munich, DE	April 2011	Active scan with SNI	826,098
Shanghai, CN	April 2011	Active scan	798,976
Beijing, CN	April 2011	Active scan	797,046
Melbourne, AU	April 2011	Active scan	833,571
izmir, TR	April 2011	Active scan	825,555
São Paulo, BR	April 2011	Active scan	833,246
Moscow, RU	April 2011	Active scan	830,765
Santa Barbara, US	April 2011	Active scan	834,173
Boston, US	April 2011	Active scan	834,054
Munich, DE	September 2010	Passive monitoring	183,208
Munich, DE	April 2011	Passive monitoring	989,040
EFF servers	March–June 2010	Active IPv4 scan	11,349,678

25 million certificates to evaluate.



Our Data Sets

Active Scans — Passive Monitoring — EFF IPv4 scan

<i>Location</i>	<i>Time (run)</i>	<i>Type</i>	<i>Certificates</i>
Tuebingen, DE	November 2009	Active scan	833,661
Tuebingen, DE	December 2009	Active scan	819,488
Tuebingen, DE	January 2010	Active scan	816,517
Tuebingen, DE	April 2010	Active scan	816,605
Munich, DE	September 2010	Active scan	829,232
Munich, DE	November 2010	Active scan	827,366
Munich, DE	April 2011	Active scan	829,707
Munich, DE	April 2011	Active scan with SNI	826,098
Shanghai, CN	April 2011	Active scan	798,976
Beijing, CN	April 2011	Active scan	797,046
Melbourne, AU	April 2011	Active scan	833,571
İzmir, TR	April 2011	Active scan	825,555
São Paulo, BR	April 2011	Active scan	833,246
Moscow, RU	April 2011	Active scan	830,765
Santa Barbara, US	April 2011	Active scan	834,173
Boston, US	April 2011	Active scan	834,054
Munich, DE	September 2010	Passive monitoring	183,208
Munich, DE	April 2011	Passive monitoring	989,040
EFF servers	March–June 2010	Active IPv4 scan	11,349,678

25 million certificates to evaluate.



Our Data Sets

Active Scans — Passive Monitoring — EFF IPv4 scan

<i>Location</i>	<i>Time (run)</i>	<i>Type</i>	<i>Certificates</i>
Tuebingen, DE	November 2009	Active scan	833,661
Tuebingen, DE	December 2009	Active scan	819,488
Tuebingen, DE	January 2010	Active scan	816,517
Tuebingen, DE	April 2010	Active scan	816,605
Munich, DE	September 2010	Active scan	829,232
Munich, DE	November 2010	Active scan	827,366
Munich, DE	April 2011	Active scan	829,707
Munich, DE	April 2011	Active scan with SNI	826,098
Shanghai, CN	April 2011	Active scan	798,976
Beijing, CN	April 2011	Active scan	797,046
Melbourne, AU	April 2011	Active scan	833,571
İzmir, TR	April 2011	Active scan	825,555
São Paulo, BR	April 2011	Active scan	833,246
Moscow, RU	April 2011	Active scan	830,765
Santa Barbara, US	April 2011	Active scan	834,173
Boston, US	April 2011	Active scan	834,054
Munich, DE	September 2010	Passive monitoring	183,208
Munich, DE	April 2011	Passive monitoring	989,040
EFF servers	March–June 2010	Active IPv4 scan	11,349,678

25 million certificates to evaluate.



Active Scans — Passive Monitoring — EFF IPv4 scan

<i>Location</i>	<i>Time (run)</i>	<i>Type</i>	<i>Certificates</i>
Tuebingen, DE	November 2009	Active scan	833,661
Tuebingen, DE	December 2009	Active scan	819,488
Tuebingen, DE	January 2010	Active scan	816,517
Tuebingen, DE	April 2010	Active scan	816,605
Munich, DE	September 2010	Active scan	829,232
Munich, DE	November 2010	Active scan	827,366
Munich, DE	April 2011	Active scan	829,707
Munich, DE	April 2011	Active scan with SNI	826,098
Shanghai, CN	April 2011	Active scan	798,976
Beijing, CN	April 2011	Active scan	797,046
Melbourne, AU	April 2011	Active scan	833,571
İzmir, TR	April 2011	Active scan	825,555
São Paulo, BR	April 2011	Active scan	833,246
Moscow, RU	April 2011	Active scan	830,765
Santa Barbara, US	April 2011	Active scan	834,173
Boston, US	April 2011	Active scan	834,054
Munich, DE	September 2010	Passive monitoring	183,208
Munich, DE	April 2011	Passive monitoring	989,040
EFF servers	March–June 2010	Active IPv4 scan	11,349,678

25 million certificates to evaluate.



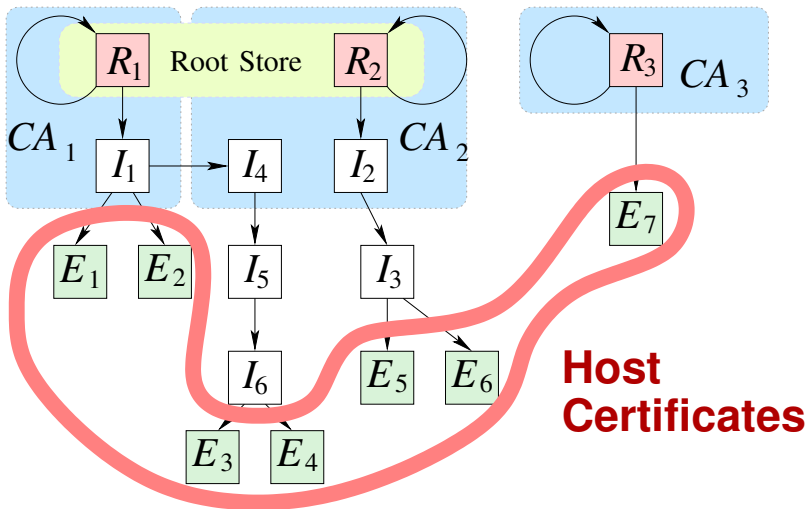
Active Scans — Passive Monitoring — EFF IPv4 scan

<i>Location</i>	<i>Time (run)</i>	<i>Type</i>	<i>Certificates</i>
Tuebingen, DE	November 2009	Active scan	833,661
Tuebingen, DE	December 2009	Active scan	819,488
Tuebingen, DE	January 2010	Active scan	816,517
Tuebingen, DE	April 2010	Active scan	816,605
Munich, DE	September 2010	Active scan	829,232
Munich, DE	November 2010	Active scan	827,366
Munich, DE	April 2011	Active scan	829,707
Munich, DE	April 2011	Active scan with SNI	826,098
Shanghai, CN	April 2011	Active scan	798,976
Beijing, CN	April 2011	Active scan	797,046
Melbourne, AU	April 2011	Active scan	833,571
İzmir, TR	April 2011	Active scan	825,555
São Paulo, BR	April 2011	Active scan	833,246
Moscow, RU	April 2011	Active scan	830,765
Santa Barbara, US	April 2011	Active scan	834,173
Boston, US	April 2011	Active scan	834,054
Munich, DE	September 2010	Passive monitoring	183,208
Munich, DE	April 2011	Passive monitoring	989,040
EFF servers	March–June 2010	Active IPv4 scan	11,349,678

25 million certificates to evaluate.



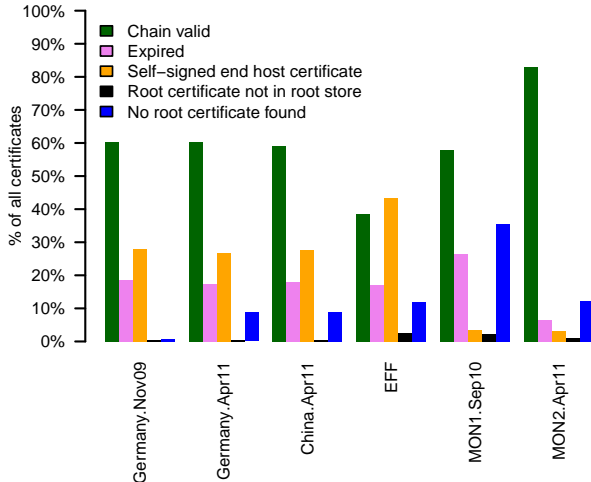
Validity of End-Hosts Certificates





Validation of Certificate Chains

Just check chains, not host names





Now also check host names

- Look in Common Name (CN) and Subject Alternative Name (SAN)
- Munich, April 2011, only valid chains:
 - 12.2% correct CN
 - 5.9% correct SAN

Only **18%** of certificates are fully verifiable

- Positive 'trend': from 14.9% in 2009 to 18% in 2011
- Addendum: recent scans show this is increasing (+ 0.5%)



Active scan

- **2.2%** correct Common Name (CN)
- **0.5%** correct Subject Alternative Name

Top 3 most frequent CNs account for > 50%

- `plesk` or similar in 27.3%
- `localhost` or similar in 25.4% – standard installations?



Many certificates valid for more than one domain

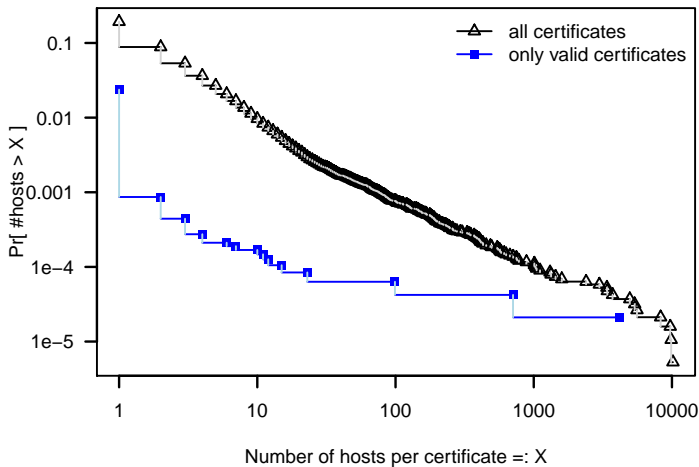
- Domains served by same IP
- Some certificates issued for dozens of domains
- Certificate reuse on multiple machines increases options for attacker

Often found on hosters

- E. g. *.blogger.com, *.wordpress.com

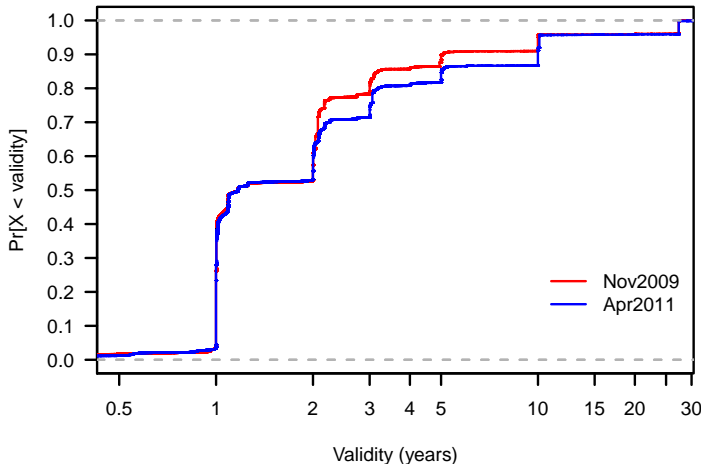


How often does a certificate occur on X hosts?





CDF of validity periods, active scans





Key types

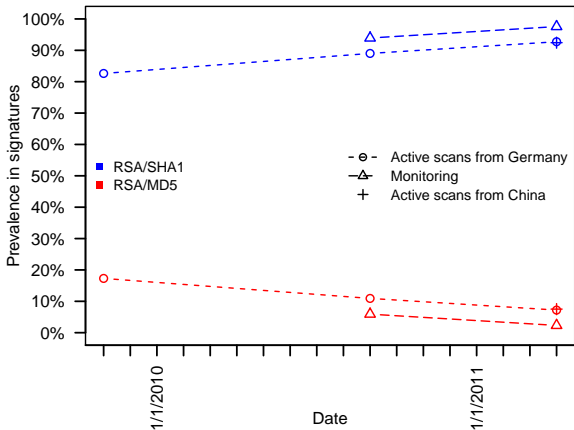
- RSA: 99.98% (rest is DSA)
- About 50% have length 1,024 bit
- About 45% have length 2,048 bit
- Clear trend from 1,024 to 2,048 bit

Weird encounters

- 1,504 distinct certificates that share another certificate's key
- Many traced to a handful of hosting companies
- Nadiah Henninger's work: Embedded devices, poor entropy!
- `www.factorable.net`



MD5 is being phased out



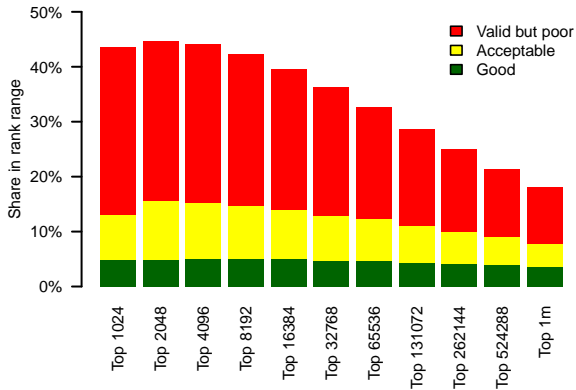


We defined 3 categories

- 'Good':
 - Correct chains, correct host name
 - Chain ≤ 2
 - No MD5, strong key of > 1024 bit
 - Validity ≤ 13 months
- 'Acceptable'
 - Chain ≤ 3 , validity ≤ 25 months
 - Rest as above
- 'Poor': the remainder



Certificate Quality



Validity correlates with rank

- Share of 'poor' certificates higher among high-ranking sites



Proposals to enhance or replace X.509



What to do about these problems?

No silver bullet known

- Part of the problem: SSL meant to protect stuff like credit card numbers
- But state-scale attacks were not in scope back in the 1990s

Several proposals:

- Extended Validation, Base Line Requirements
- Pinning Information
- Keys in DNSSEC (DANE)
- Perspectives/Convergence
- Public Logs: Sovereign Keys, Certificate Transparency



Extended Validation

- CAs to require state-issued documents before certification
- More expensive
- Rarely bought by customers

Base Line Requirements

- CA/Browser forum standard
- Absolute minimum requirements for validation
- Audit-based, rules for audits



Idea

- Browser stores last-seen public key of a site
- Alternatively: store issuing CA
- Recognise again upon next visit

Discussion

- Does not help against attack on first contact
- False alarms when certificates change (not rare!)
- How many certs to store?



Idea

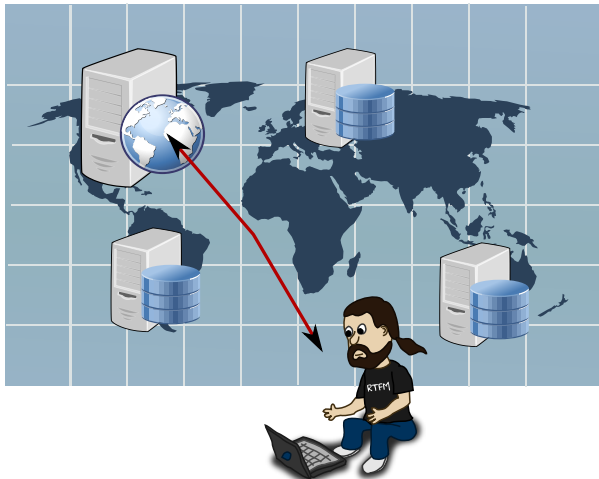
- DNSSEC already a hierarchical **state-level** PKI
- Verification from Root Server down to end-host
- New Resource Record in DNSSEC: public key of site

Discussion

- Straight-forward and strong
- Performance? Caching? DJB says it's poor.
- Countries control their own TLDs. Think `bit.ly`!
- Defence against country-level attack?

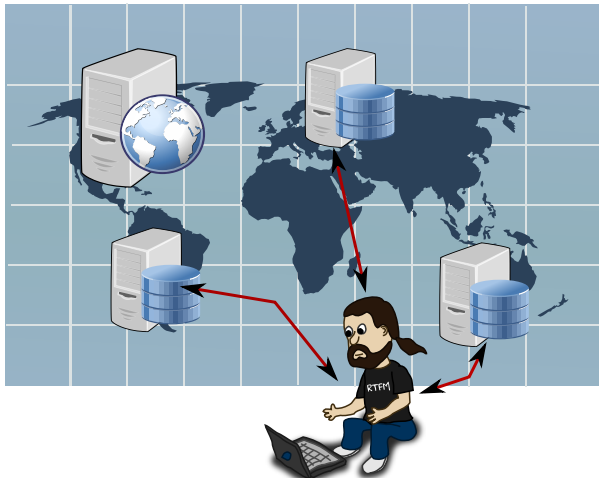


Idea: Notaries





Reconfirm with notaries





Advantages

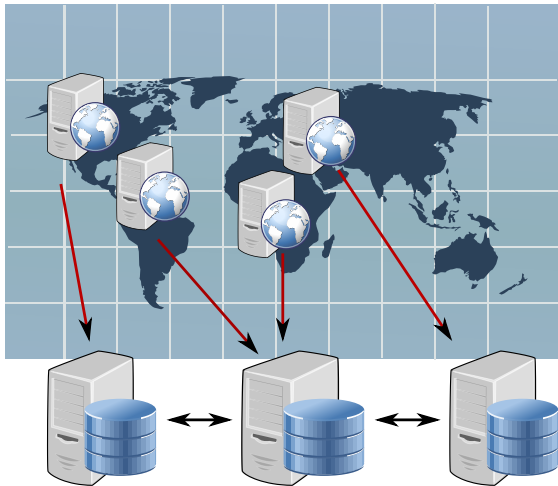
- Works well against MitM in the network
- Reinforcement or replacement of CA system?

Possible problems

- Privacy
- False positives: some sites change certificates frequently
- Content Distribution Networks?

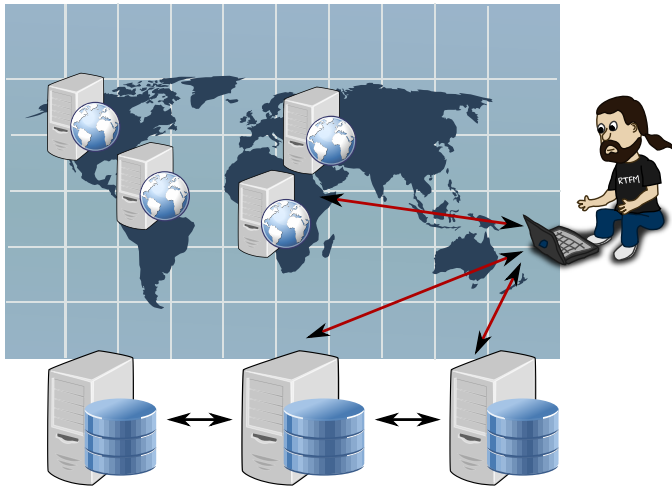


Store information





Retrieve information





Idea: store information publicly and append-only

- Sovereign Keys
 - Sites stores authoritative key to cross-sign its certificates
 - Goal: cross-certification and cross-validation of keys
- Certificate Transparency
 - CAs and others store info about who is certified by whom
 - Goal: detect rogue CA issuing key for a site

Schemes are very new - end of 2011



Sovereign Keys (EFF)

Sites store information on < 30 timeline servers

timestamp	name	key	protocols	evidence
1322736203	A	0x427E8A	https, smtps	$Sig_{CA}(A, \dots)$
1323254603	B	0x7389FB	https:8080	$Sig_B(B, \dots)$
1323657143	C	0x49212A	imaps	$Sig_C(C, \dots)$
1413787143	A	0x427E8A	https, smtps	$Sig_{CA}(A, \dots)$
...

Work-in-progress (alive)

- Timeline is auditable by clients
- Mirrors proposed
- <https://www.eff.org/sovereign-keys>



Pros

- Does not need CA support
- Evidence can be based on DANE DNSSEC, CAs, ...
- Performance and bandwidth?

Cons

- Continuous monitoring of timeline server needed
- Maintain list of timeline servers
- Entries are not space-efficient
- Privacy (suggested remedy: TOR-like proxying)
- Key loss



Store certification proof on public servers

timestamp	name	cert	evidence
1322736203	A	Cert chain by Verisign	$MSig(\text{hashes})$
1323254603	B	Self-signed cert	$MSig(\text{hashes})$
1323657143	C	Cert by CACert	$MSig(\text{hashes})$
...	$MSig(\text{hashes})$

Work-in-progress (alive)

- Timeline consistency can be monitored
- Roles: clients, auditors, monitors (on-behalf)



Certificate Transparency (Google)

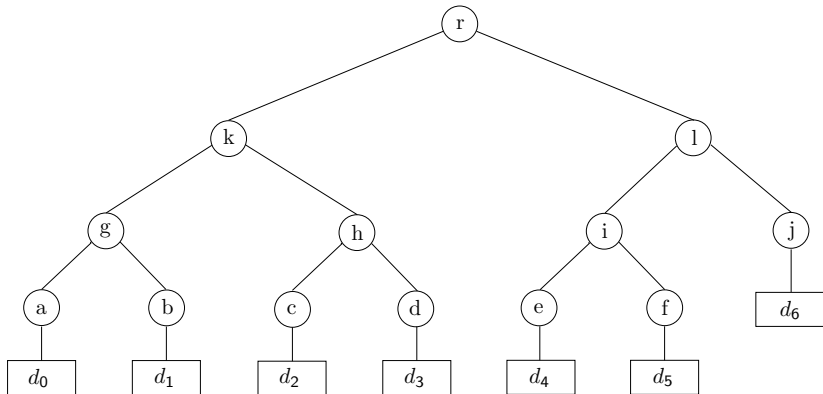


Figure : Log is a Merkle tree, d_i are new certificate chains.



Pros

- Protects against rogue/hacked CAs
- Efficient data structure
- Has Google campaigning for it

Cons

- Requires continuous monitoring of logs
- Monitors need full log at all times, act on behalf of others
- Proofs are $O(\log n)$, but storage is linear



Crossbear: Detecting and Localising the MitM



Case study 1: Syria vs. Facebook?

MAY 5, 2011 | BY PETER ECKERSLEY



A Syrian Man-In-The-Middle Attack against Facebook

UPDATE: If you are in Syria and your browser shows you this certificate warning on Facebook, *it is not safe to login to Facebook*. You may wish to use **Tor** to connect to Facebook, or use proxies outside of Syria.

Certificate:

Data:

Serial Number: c6:4f:50:11:b3:65:dc:b9

Issuer: C=US, ST=California, L=Alto Palo, O=Facebook, Inc.,
OU=Facebook, CN=s.static.ak.facebook.com

Subject: C=US, ST=California, L=Alto Palo, O=Facebook, Inc.,
OU=Facebook, CN=s.static.ak.facebook.com



Case study 2: hotel in in Warsaw?

[...] I spent the night in a hotel in Warsaw, Poland. I bought access to the Internet WiFi. [...]

For the SSL connection to `imap.googlemail.com` (also known as `imap.gmail.com`) at port 993, Thunderbird warned me about a certificate error. The certificate presented by IP address 74.125.115.16 was issued to

Subject: C=US, ST=California, L=Mountain View, O=Google Inc, CN=`imap.googlemail.com`

and it was issued by

Issuer: C=US, ST=California, L=Sunnyvale, O=Fortinet, OU=Certificate Authority, CN=FortiGate CA/emailAddress=`support@fortinet.com`

Actually attaches a traceroute.



Case study 3: DigiNotar vs. Iran?

The screenshot shows a web browser window with a red background. The address bar displays a URL from google.com. A security error message is shown in a white box with a yellow warning icon:

Invalid Server Certificate

You attempted to reach www.google.com, but the server presented an invalid certificate.

[Back](#)

[Help me understand](#)

When you connect to a secure website, the server hosting that site presents your browser with something. This certificate contains identity information, such as the address of the website, which is verified by a third party checking that the address in the certificate matches the address of the website, and not a third party (such as an attacker on your network).

In this case, the server certificate or an intermediate CA certificate presented to your browser is invalid. This malformed, contains invalid fields, or is not supported.

Overlaid on the right is a 'Certificate' dialog box with the following content:

- General Details Certification Path
- Certification path:
 - DigiNotar Root CA
 - DigiNotar Public CA 2025
 - %.google.com
- View Certificate
- Certificate status:
 - This certificate is OK.
- Learn more about [certification paths](#)
- OK



This is *not* a proposal to strengthen X.509.

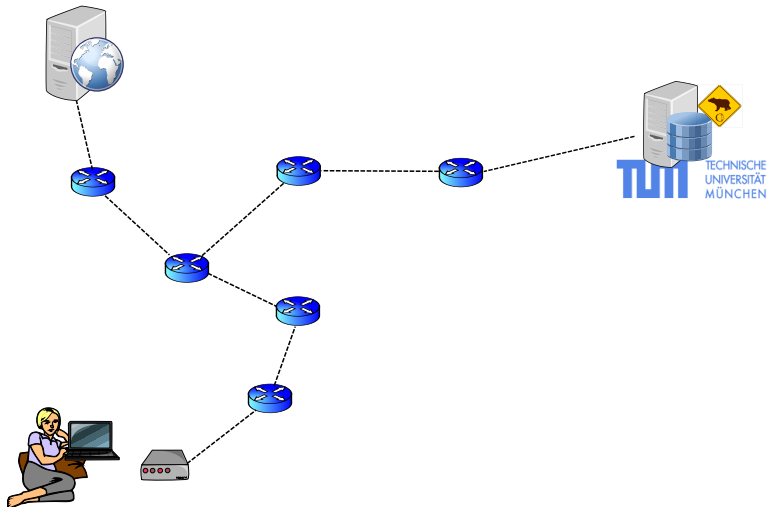
Crossbear: a tool to gather *hard data*.

- Raise reliable data about MitM *in the wild*
- *How often* do MitM occur?
- *Where* are the attackers located?
- *Who* are the attackers?
- Are we jumping at shadows?

Method: combine notary principle, tracing and centralised reporting and analysis.

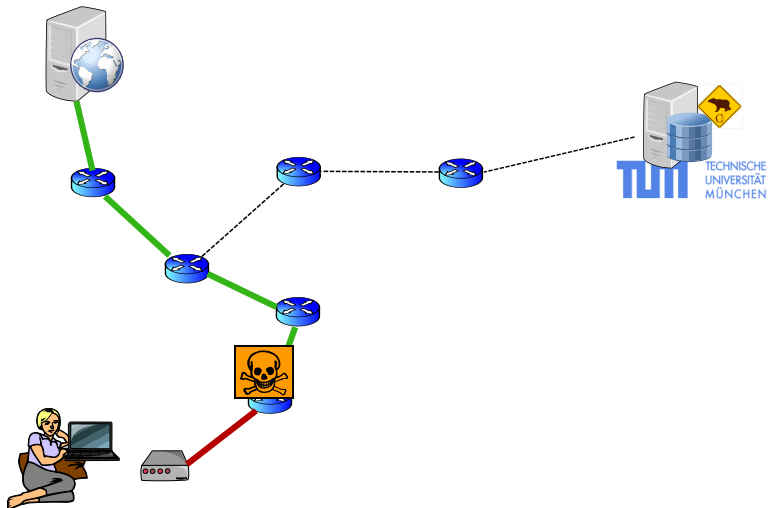


Alice is surfing...



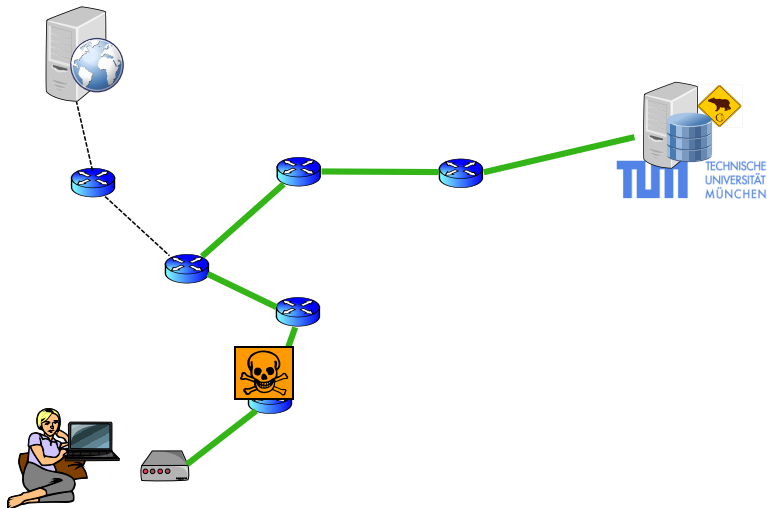


Man-in-the-middle



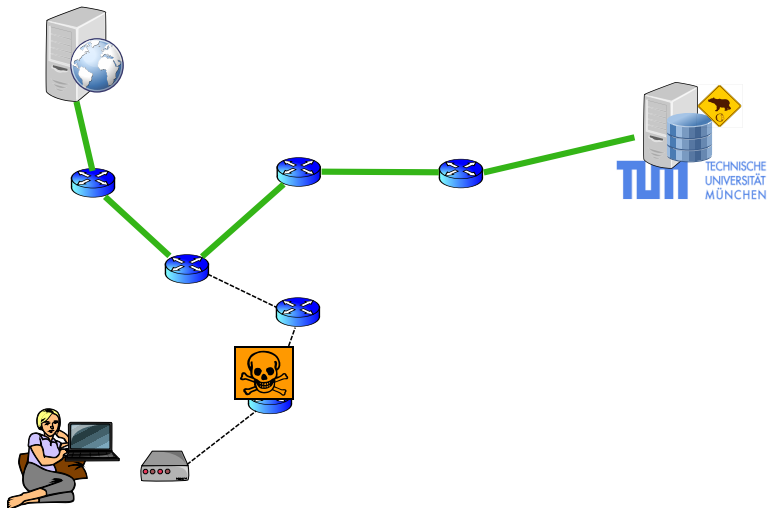


Alice queries Crossbear



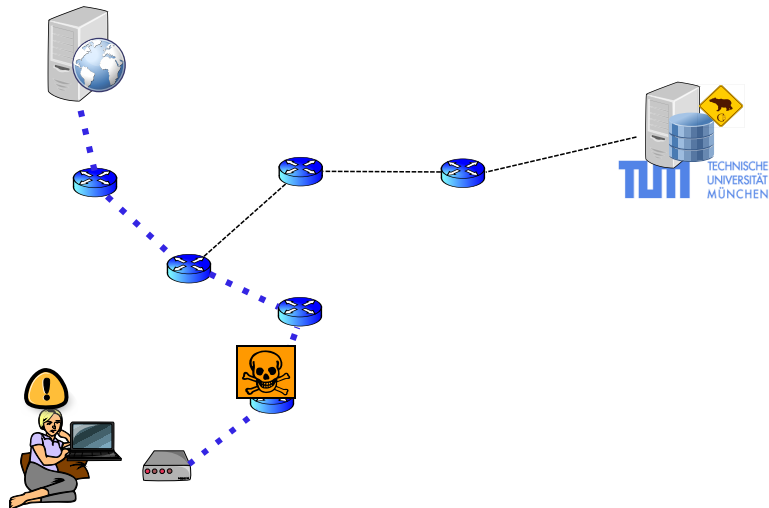


Crossbear checks the server



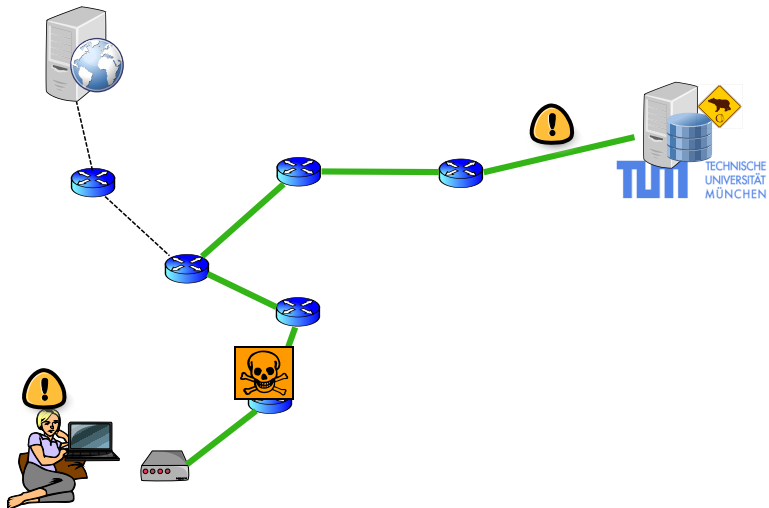


Alice traceroutes to server



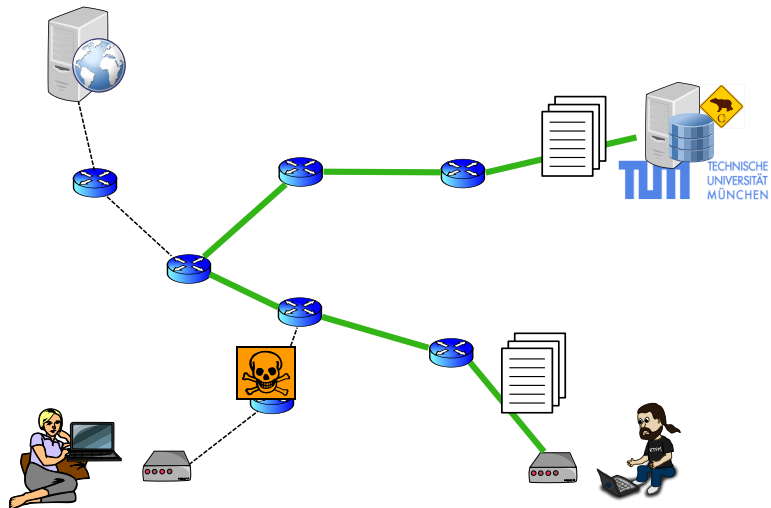


Alice reports to Crossbear



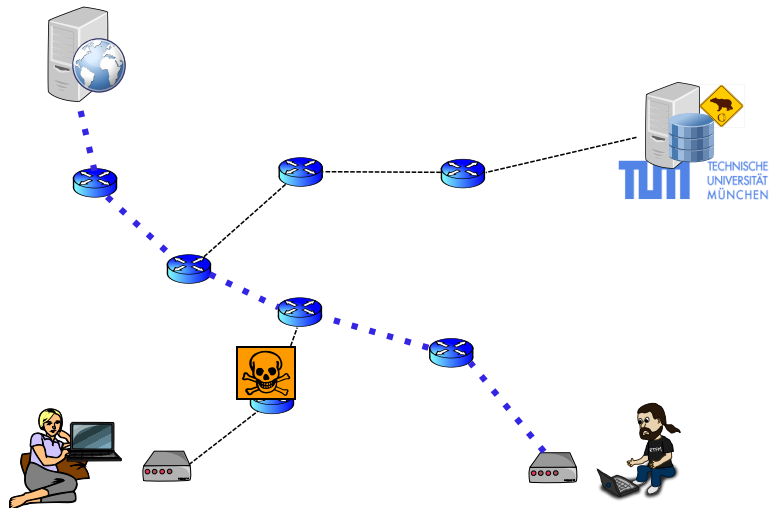


Distribute hunting tasks



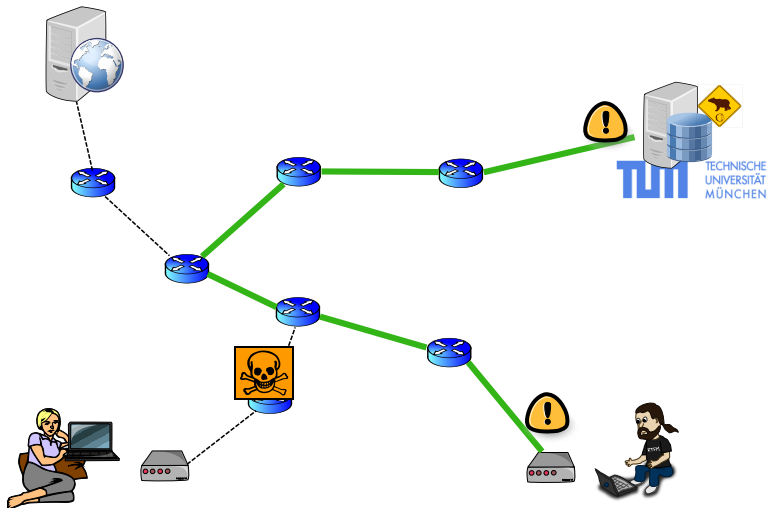


Bob goes hunting



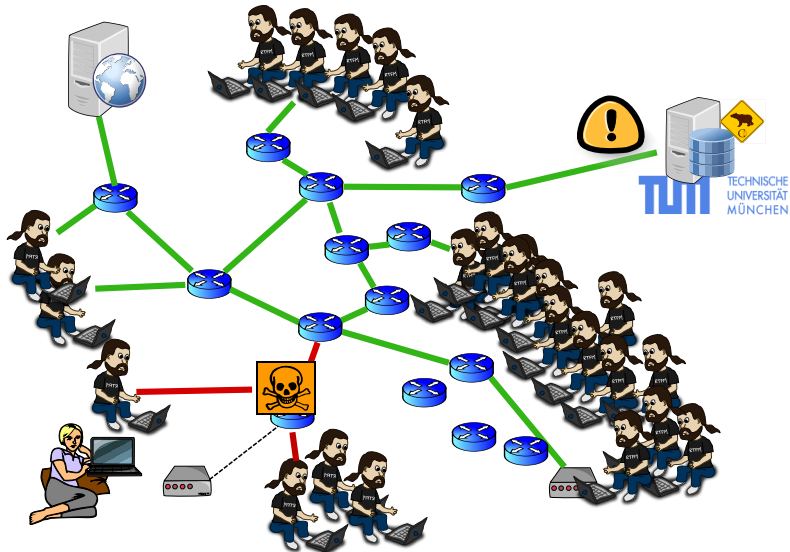


Bob reports



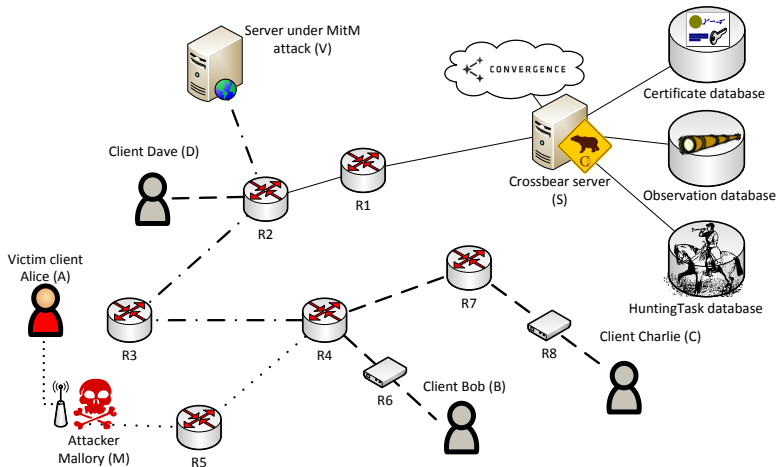


There are many Bobs





Implemented and running





Server: store and analyse

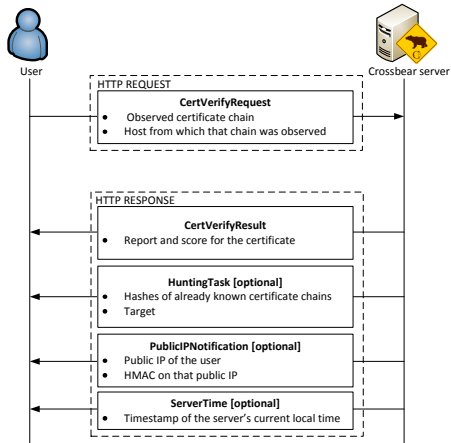
- Crossbear server at TU München, Germany
- Uses Convergence project's notaries for diversity
- Server cert hard-coded into client!

Detection and localisation

- Clients as Firefox add-on (detection and localisation)
- 150 stand-alone hunters on stand-by on PlanetLab (localisation)



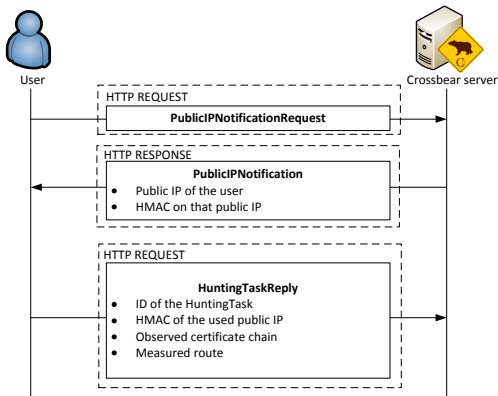
Verification request



NB: SSL-secured connection, server cert hard-coded



Hunter reply



NB: SSL-secured connection, server cert hard-coded



Actually, we also determine on server-side:

- CAs used in certificate chain (→ continuity)
- AS number of hosts in traceroute
(→ frequent reports?)
- Geo data: location of hosts in traceroute
(→ traversed countries)
- WHOIS info

Firefox add-on

- For *savvy users*
- Score-based, several factors
- UI → see code on github



Chosen based on MitM reports we have

Attacker behaviour

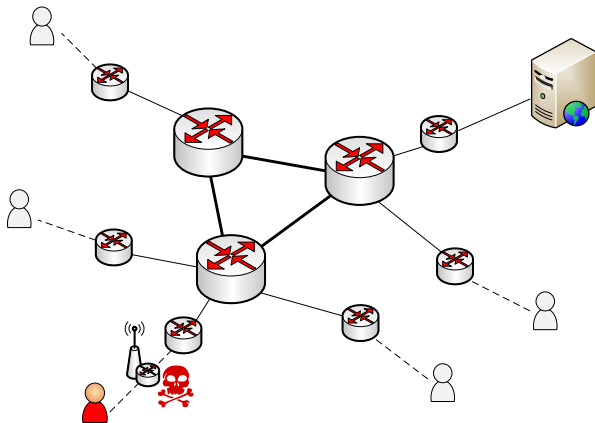
- Non-selective: MitM all attached 'client' systems
- Selective: MitM only some of attached 'client' systems

Attacker position

- Towards periphery, close to victim client
- Towards periphery, close to victim server
- Central location in network (important AS, ...)

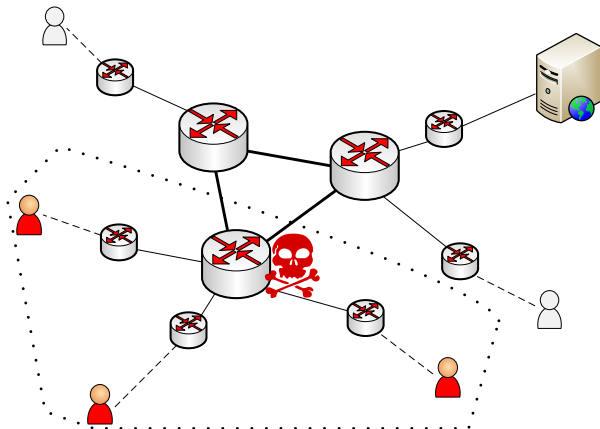


Non-selective, close to victim client





Non-selective, state-level attacker





Detection

- Attack is detected if ≥ 1 reports
- Attacker can only drop connections to Crossbear server

Lends itself well to localisation

- Get ≥ 1 traceroute from victim, ≥ 1 from unpoisoned hunter
- The more, the better
- The closer to intersection point, the better
- Success depends on the number of hunters
- An estimate can be given

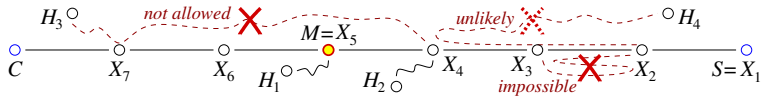


Possible to give a closed-form model, at both router and AS-level.

- Non-selective attacker
- Routing symmetric
(OK if path lengths not much different)
- Routing based only on destination address
(hot-potato routing etc. rare)
- Probability for a node (router, AS) as location for hunter is evenly distributed (*)
- Probability that traffic is forwarded to specific neighbour is evenly distributed (*)



Closed-form model



Ideas

- First, treat path length victim \leftrightarrow server as fixed length
- Probability that randomly placed hunter covers a node is function of node degree
- Probability that one hunter is victim and another just escapes MitM is function of node degree
- Aggregate: sum over all possible path lengths, and all possible locations of attacker on path
- Model only depends on node degrees and path lengths



Router level

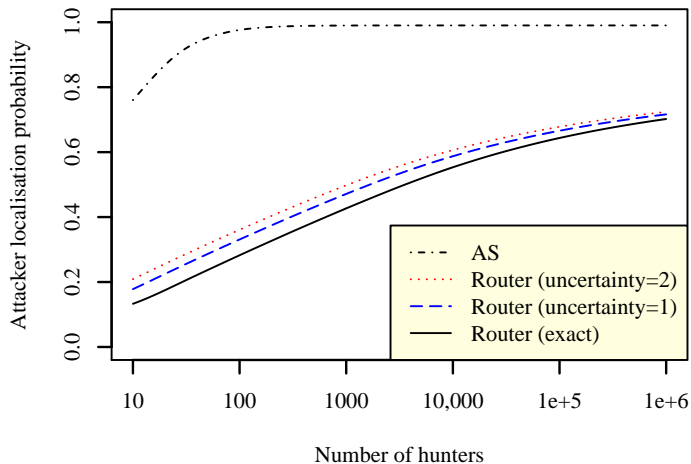
- Node degrees: Rocketfuel
- Path lengths: number of IP hops: traceroutes to 30k random hosts

AS level

- Node degree and path lengths: RouteViews archive

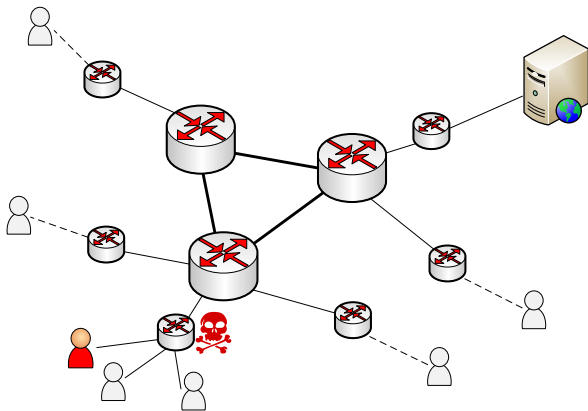


Number of hunters vs. success



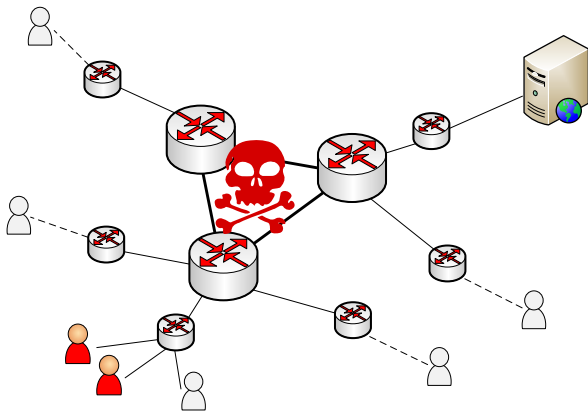


Selective attacker: close to victim



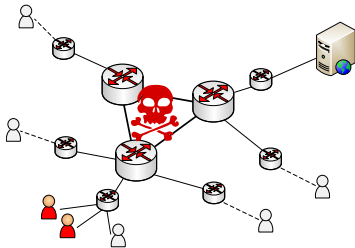


Selective attacker: in core





Selective attackers are a headache



Can be indistinguishable from non-selective attacks

- *Every* attack report to be checked for plausibility
- But attacker should leave some hints – cannot arbitrarily spoof IP addresses



Attack seems to be restricted to few stub AS

- Use BGP data to check traceroutes for plausibility
- Do MitM certificates share properties?
- Which AS in which countries involved? Some 'known' suspect?

MitM reports from just a few companies?

- Check traceroutes for traversed countries and AS
- Might be industrial espionage

All of this is intensive manual work. But only localisation is affected, and it is better than no data all.



Crossbear is an open system

- Malicious injection of data
 - Clients/hunters have no ID, no authentication
 - Attacker can eclipse real hunters in his network, too
 - Should results in clusters of suspicious reports, though
- Denial-of-service attacks
- It is an arms race
- Other detection systems are subject to same attacks



A first step towards gathering better data

- We *do not* advertise Crossbear as a silver bullet
- Best results can be expected against the non-selective attacker
- These are also the attackers we are most interested in

Crossbear is deployed and ready

- 150 hunters on PlanetLab
- 4,000 certificate reports – no MitM



SSH

- We have a PoC for SSH host key verification.
- Patch for OpenSSH
- Want to present (hopefully) as lightning talk at 29C3
- No hunting implemented yet

Open Observatory of Network Interference

- Measurement framework to detect traffic manipulation / censorship
- Won a grant for integrating Crossbear
- Ongoing work



Thank you!



Contact

- Twitter: @crossbearteam
- WWW: <https://pki.net.in.tum.de>
- <https://github.com/crossbear/Crossbear>

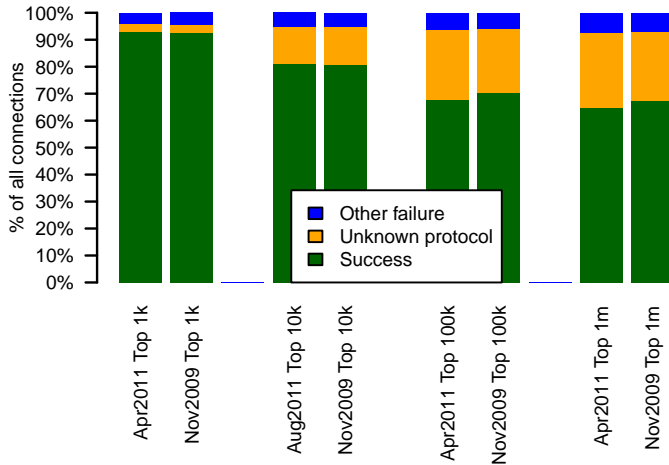


Backup Slides



Errors in TLS Connection Setup

Scans from Germany, Nov 2009 and Apr 2011





UNKNOWN PROTOCOL

- Rescanned those hosts and manual sampling
- Always plain HTTP...
- ... and always an `index.html` with HTML 2 ...
- Hypothesis: old servers, old configurations
- More likely to happen in the lower ranks



CN=plesk or similar

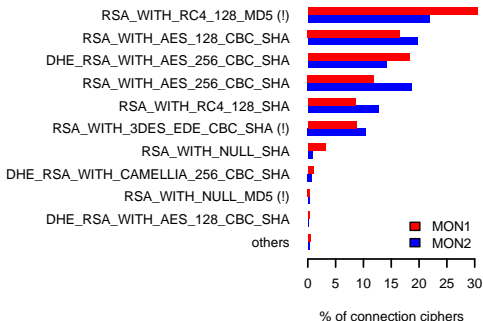
- Found in 7.3% of certificates
- Verified: Plesk/Parallels panels

CN=localhost

- 4.7% of certificates



Results from monitoring

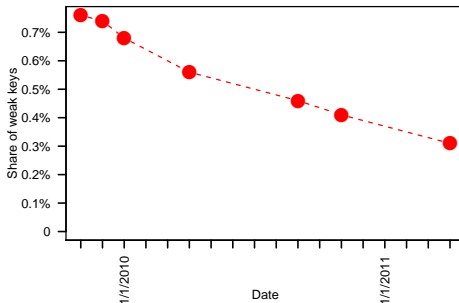


(Mostly) in line with results from 2007 by Lee et al.

- Order of AES and RC4 has shifted, RC4-128 most popular



Weak randomness in key generation – serious bug of 2008

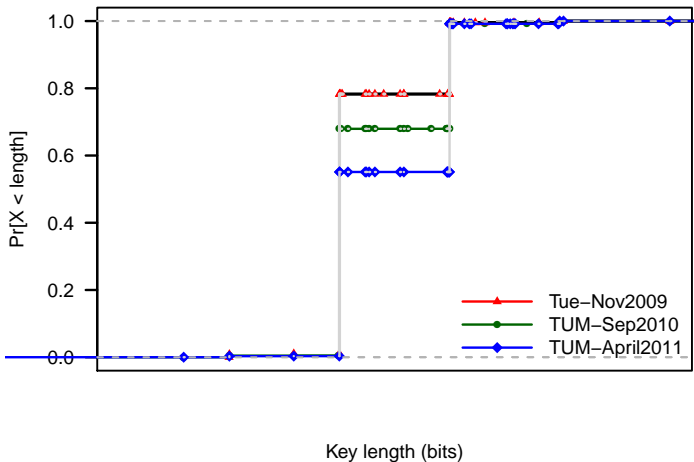


In line with findings of 2009 by Yilek et al.



Public Key Lengths

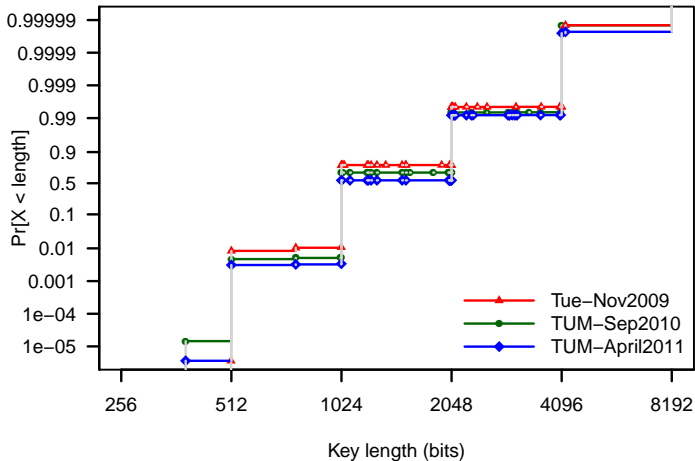
CDF for RSA key lengths – linear Y axis





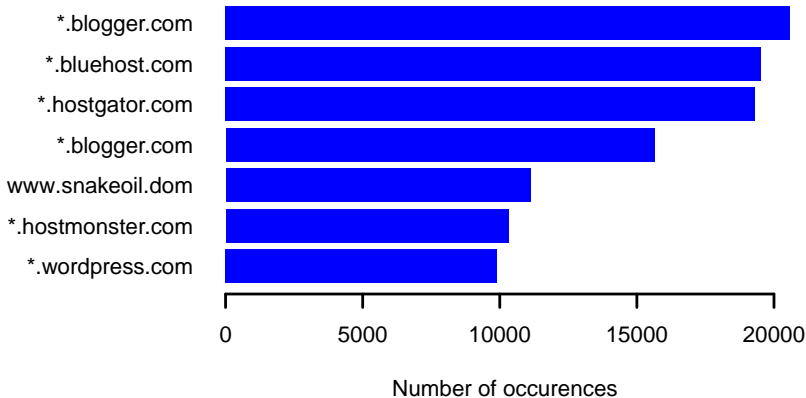
Public Key Lengths

CDF for RSA key lengths – double-log Y axis



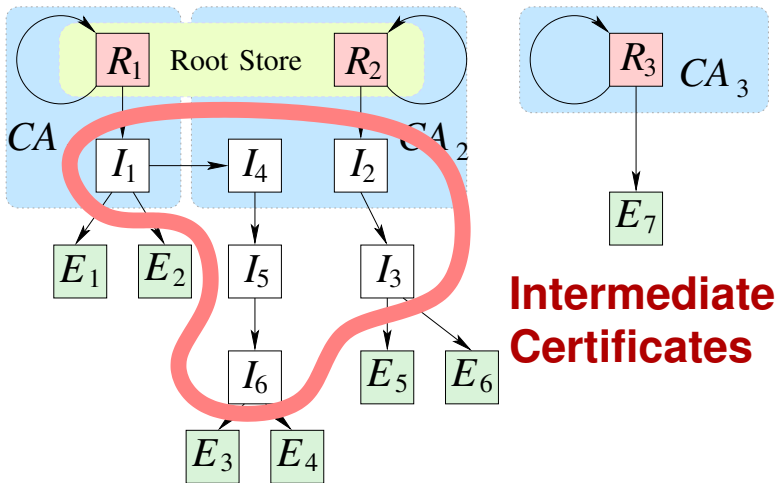


Most frequent Common Name occurrences



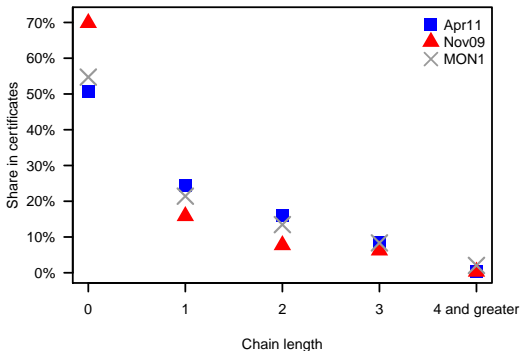


Certificate Chains





Certificate Chain Lengths

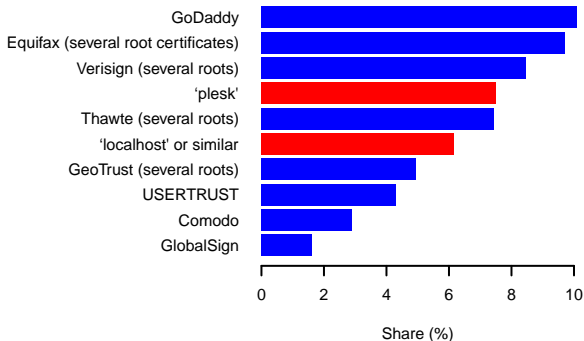


Finding more positive than negative:

- Trend to use intermediate certificates more often
- Allows to keep Root Certificates offline
- But chains still reasonably short



Very few CAs account for $> 50\%$ of certificates



But there are 150+ Root Certificates in Mozilla.