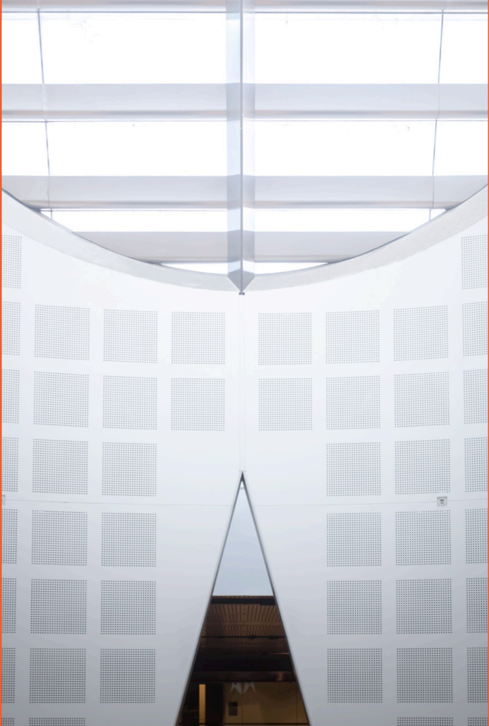
TLS in the wild

Internetscans for security

Presented by

Ralph Holz
School of Information Technologies

THE UNIVERSITY OF SYDNEY



This is joint work

Team TLS

- Johanna Amann (ICSI)
- Olivier Mehani, Dali Kafaar (Data61)
- Matthias Wachs (TUM)

Team BGP

- Johann Schlamp, Georg Carle (TUM)
- Quentin Jacquemart, Ernst Biersack (Eurecom)

About me

Quick CV

- Lecturer at University of Sydney
- Visiting Fellow at UNSW
- Previously Researcher at Data61 (ex-NICTA)
- PhD from Technical University of Munich
- And I do Internet security measurement…
- (Also: blockchains)

About this lecture

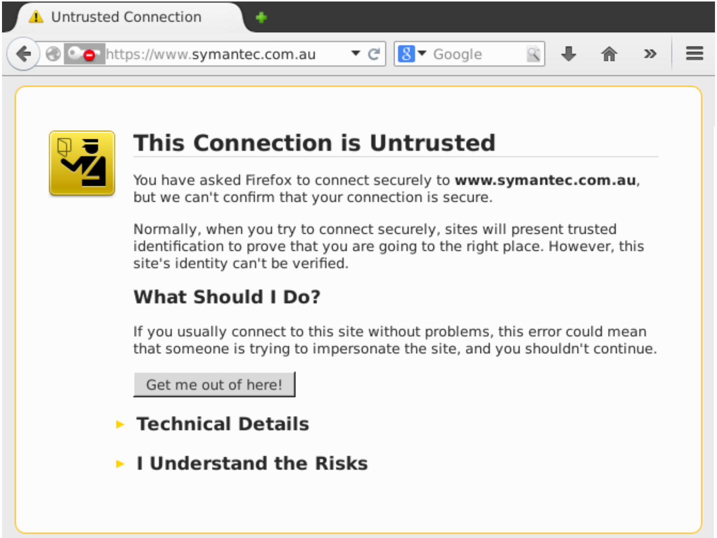
This is a story about

- …how security measurements can identify shortcomings in deployed technology
- …how data from active scans can be reused for further, benign purposes

There are three parts to this story

- From identifying the problem to scanning the Web
- New insights about electronic communication: email and chat
- Reusing data in new contexts. Here: security of Internet routing!

Background: a typical Internet experience



Reason (not a UX fail)

Technical Details

www.symantec.com.au uses an invalid security certificate.

The certificate is only valid for the following names:
symantec.com, norton.com, careers.symantec.com, customercare.symantec.com, jobs.symantec.com, www.account.norton.com, account.norton.com, mynortonaccount.com, www.nortonaccount.com, nortonaccount.com, downloads.guardianedge.com, www.pgp.com, store.pgp.com, na.store.pgp.com, eu.store.pgp.com, uk.store.pgp.com, row.store.pgp.com, nukona.com, www.nukona.com

(Error code: ssl_error_bad_cert_domain)

The X.509 Public Key Infrastructure (PKI)

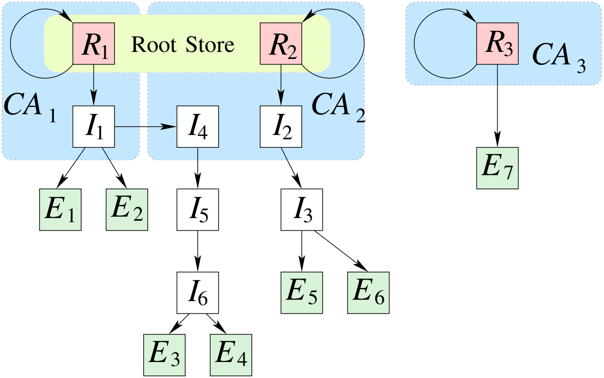
Much of our Internet security is built on X.509

- Every TLS-secured protocol uses X.509
- Further use cases: email, code-signing, …

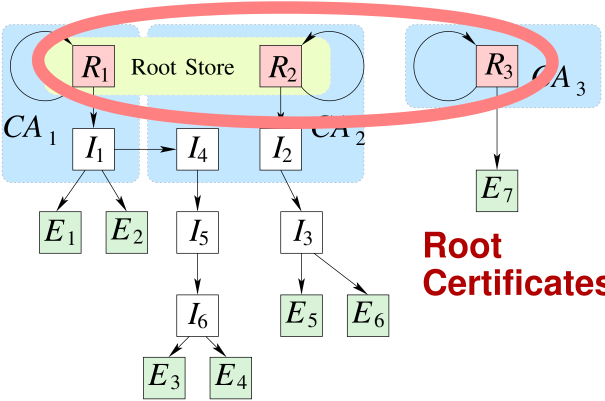
All X.509 PKIs share the same principle

- Certificates bind an entity name to a public key
- Certification Authorities (CAs) act as certificate issuers
- Browsers/OSes preconfigured with CAs' 'root' certificates

Basic idea of X.509 PKI

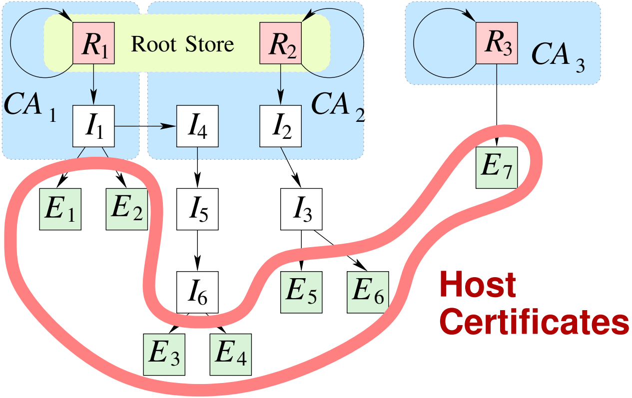


Basic idea of X.509 PKI

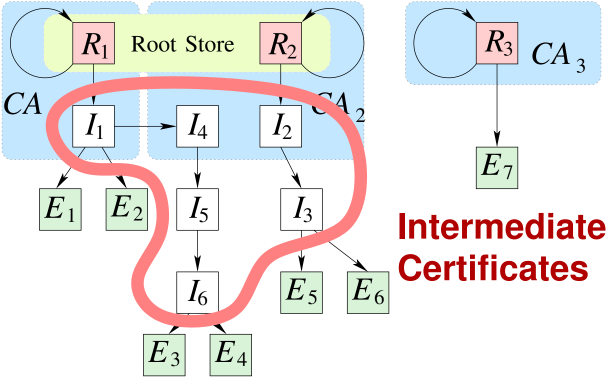


Root Certificates

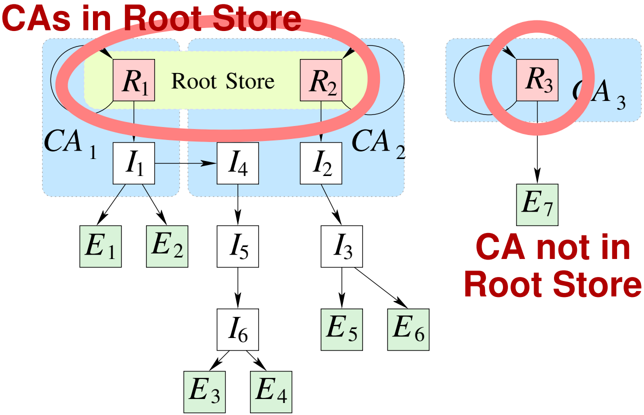
Basic idea of X.509 PKI



Basic idea of X.509 PKI

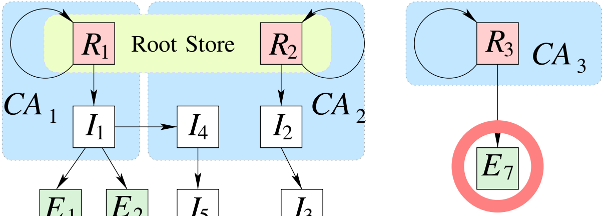


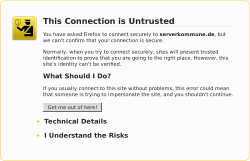
Basic idea of X.509 PKI



Basic idea of X.509 PKI

Root certificate not in Root Store





Best-of attacks on X.509

- Dec 2008:
  - ‘Error’ in Comodo subseller: no identity check
- Mar 2011: Comodo CA hacked
  - Blacklisting of $\approx$ 10 certificates
- Jul 2011: DigiNotar CA melt-down
  - 531 fake certificates *in the wild*
- 2012: Türktrust’s ‘accidental Man-in-the-middle’
- 2012: Trustwave: issued surveillance certs for years
- I stopped tracking it in around that time (PhD was done)

2008–2011: we assess the quality of X.509 for the Web

X.509 should:

- …allow HTTPS on all WWW hosts
- …contain only valid certificates
- …offer good cryptographic security

And there should be:

- Long keys, only strong hash algorithms, …
- Correctly deployed certs

Does it?

Data sets: 25m certificates

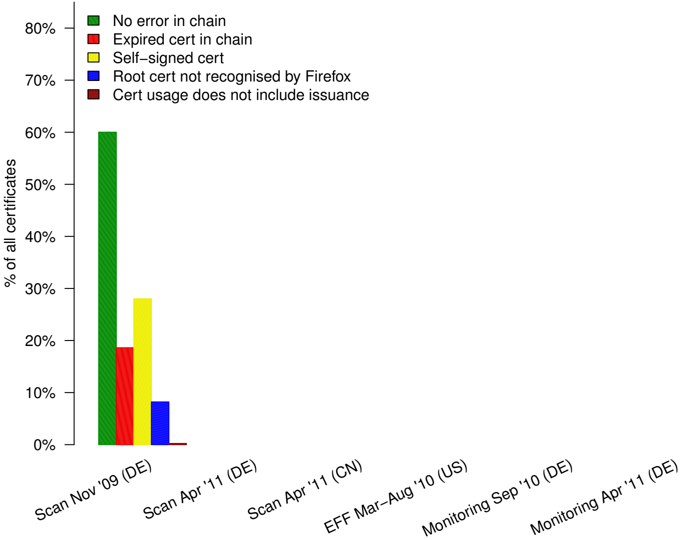
Active scans to measure *deployed* PKI

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

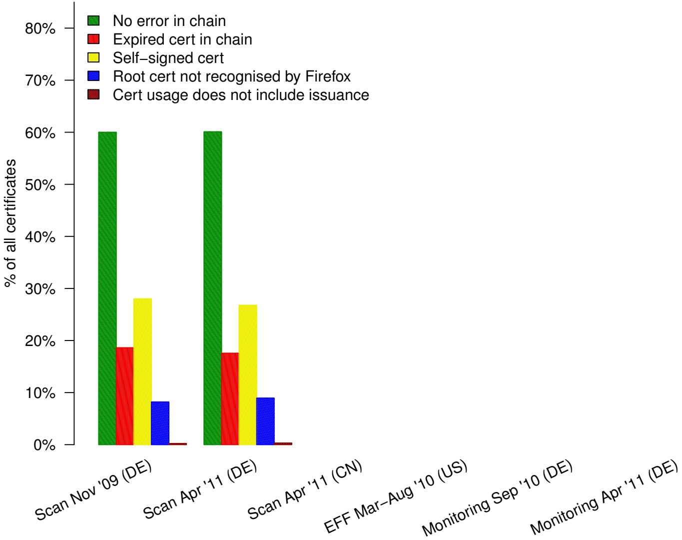
Passive monitoring to measure *user-encountered* PKI

- Munich Research Network
- Real SSL/TLS as caused by *users*

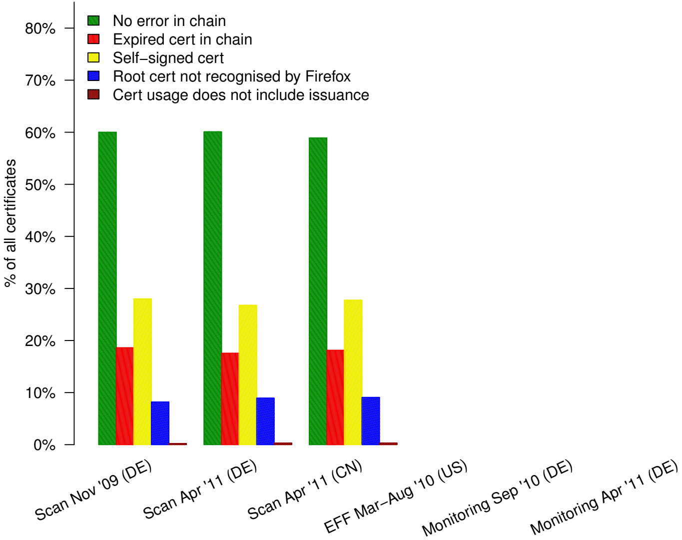
Correctness of certificate chains



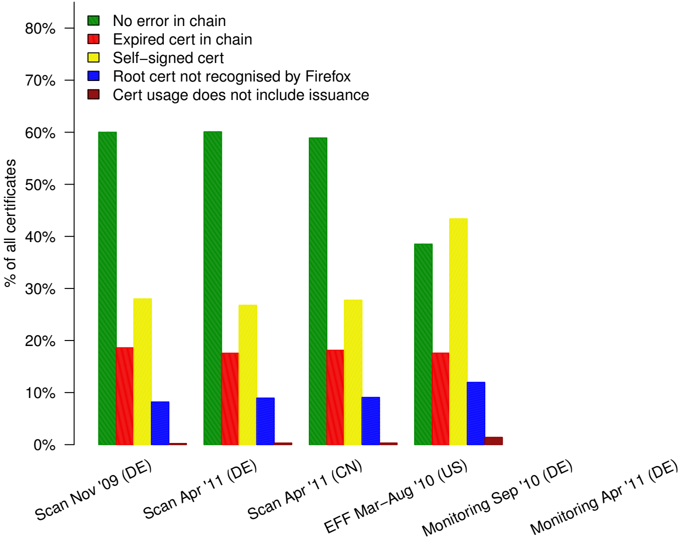
Correctness of certificate chains



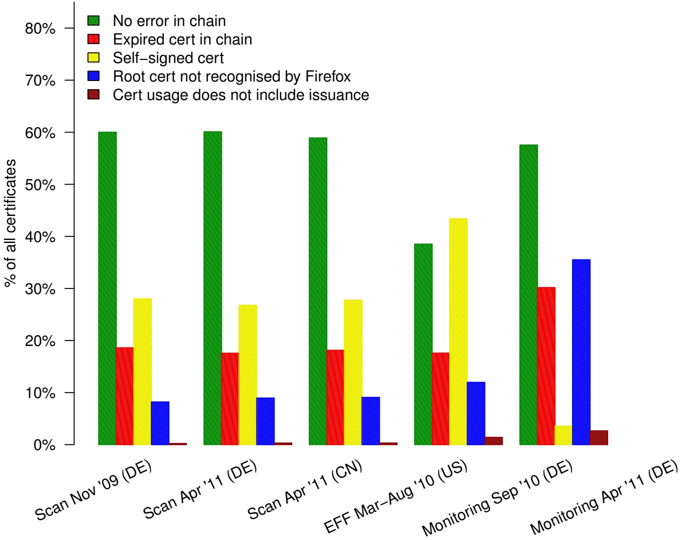
Correctness of certificate chains



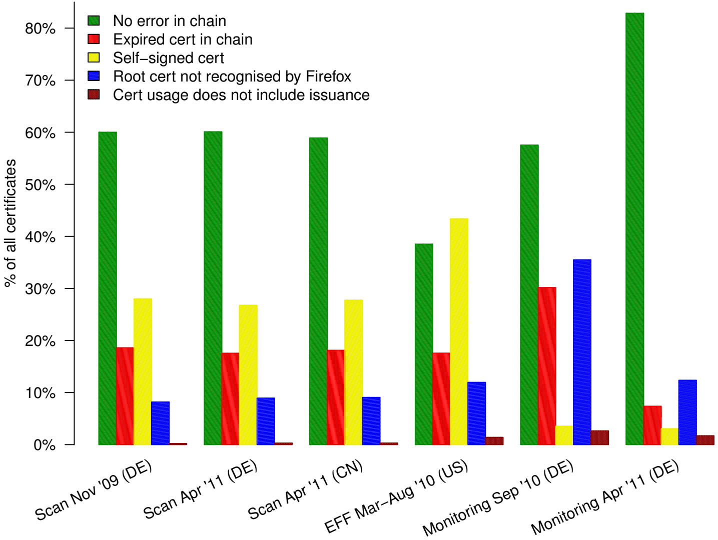
Correctness of certificate chains



Correctness of certificate chains



Correctness of certificate chains



Domain names in certificates

Are certificates issued for the right domain name?

- Tested for scans of Alexa Top 1m
- Compare name in certificate against domain name, incl. wildcard matching
- Only **18%** of certificates are fully verifiable
- **More than 80%** of the deployed certificates show errors

What about…

Email?

- Email: 4.1B accounts in 2014; 5.2B in 2018
- Most prevalent, near-instant form of communication

Chat?

- Once dominant instant-messaging (IRC!)
- Newer: XMPP (also proprietary use)

Research question: how secure are these?

Securing email and chat

SSL/TLS is the common solution

- Responder authenticates with certificate
- Initiator usually uses protocol-specific method
- Direct SSL/TLS vs. STARTTLS in-band upgrade
  - Susceptible to active man-in-the-middle attack

Email protocols

- Email submission: SMTP, SUBMISSION (= SMTP on 587)
- Email retrieval: IMAP, POP3

Investigated properties

In this lecture:

- Deployment numbers
- STARTTLS
- Versions
- Ciphers used/negotiated
- Responder authentication
- Initiator authentication

Focus mostly on email. There is more in the paper.

Data collection (July 2015)

Active scans

- To determine state of *deployment*
- `zmap` in the 'frontend', `openssl`-based 'backend'

Passive monitoring

- To determine *actual use*
- Bro monitor, UCB network

Active scans (July 2015)

Protocol (port)	No. hosts	SSL/TLS	Certs	Interm. (unique)
SMTP†,‡ (25)	**12.5M**	3.8M	**1.4M**	2.2M (1.05%)
SMTPS‡ (465)	**7.2M**	3.4M	**801k**	2.6M (0.4%)
SUBMISSION†,‡ (587)	**7.8M**	3.4M	**754k**	2.6M (0.62%)
IMAP†,‡ (143)	**8M**	4.1M	**1M**	2.4M (0.54%)
IMAPS (993)	**6.3M**	4.1M	**1.1M**	2.8M (0.6%)
POP3†,‡ (110)	**8.9M**	4.1M	**998k**	2.3M (0.44%)
POP3S (995)	**5.2M**	2.8M	**748k**	1.8M (0.44%)
IRC† (6667)	2.6M	3.7k	3k	0.6k (13.17%)
IRCS (6697)	2M	8.6k	6.3k	2.5k (12.35%)
XMPP, C2S†,‡ (5222)	2.2M	54k	39k	5.9k (32.28%)
XMPPS, C2S (5223)	2.2M	70k	39k	33k (8.5%)
XMPP, S2S†,‡ (5269)	2.5M	9.7k	6.2k	5.9k (32.28%)
XMPPS, S2S‡ (5270)	2M	1.7k	1.1k	0.8k (18.77%)
HTTPS (443)	42.7M	27.2M	8.6M	25M (0.93%)

$\dagger$ = STARTTLS, $\ddagger$ = fallback to SSL 3.

Passive observation (July 2015)

Protocol	Port	Connections	Servers
SMTP†	25	**3.9M**	**8.6k**
SMTPS	465	37k	266
SUBMISSION†	587	**7.8M**	**373**
IMAP†	143	26k	239
IMAPS	993	**4.6M**	**1.2k**
POP3†	110	19k	110
POP3S	995	**160k**	**341**
IRC†	6667	50	2
IRCS	6697	18k	15
XMPP, C2S†	5222	14k	229
XMPPS, C2S	5223	**911k**	**2k**
XMPP, S2S†	5269	175	2
XMPPS, S2S	5270	0	0

† = STARTTLS.

STARTTLS support and use

	Active probing	Passive monitoring		
Protocol	**Supported & upgraded**	**Supporting servers**	**Offering connections**	**Upgraded connections**
SMTP	30.82%	59%	97%	94%
SUBMISSION	43.03%	98%	99.9%	97%
IMAP	50.91%	77%	70%	44%
POP3	45.62%	55%	73%	62%

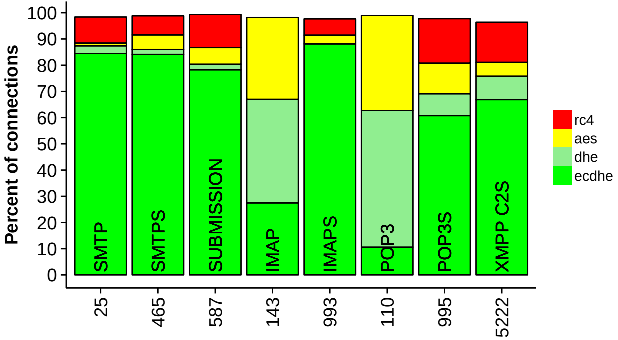
- **Deployment** as scanned: 30-50%—not good
- **Use** as monitored: better, but still not very good
  - SMTP: almost all connections upgrade
  - But not in IMAP/POP3

SSL/TLS versions in use (passive observation)

Version	Active probing Negotiated with server	Passive monitoring Observed connections
SSL 3	0.02%	1.74%
TLS 1.0	39.26%	58.79%
TLS 1.1	0.23%	0.1%
TLS 1.2	60.48%	39.37%

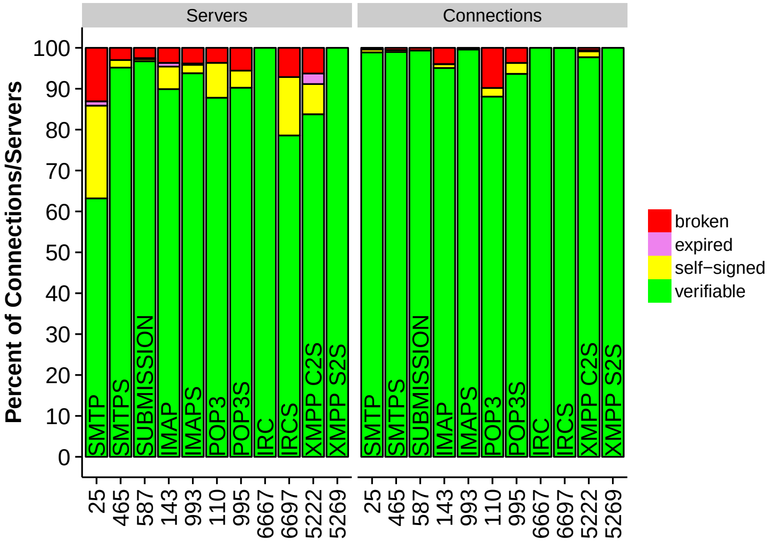
- SSL 3 is almost dead, some use left—are these old clients?
- TLS 1.2 most common in deployments, but not in use (not good)

Ciphers and forward secrecy (from monitoring)

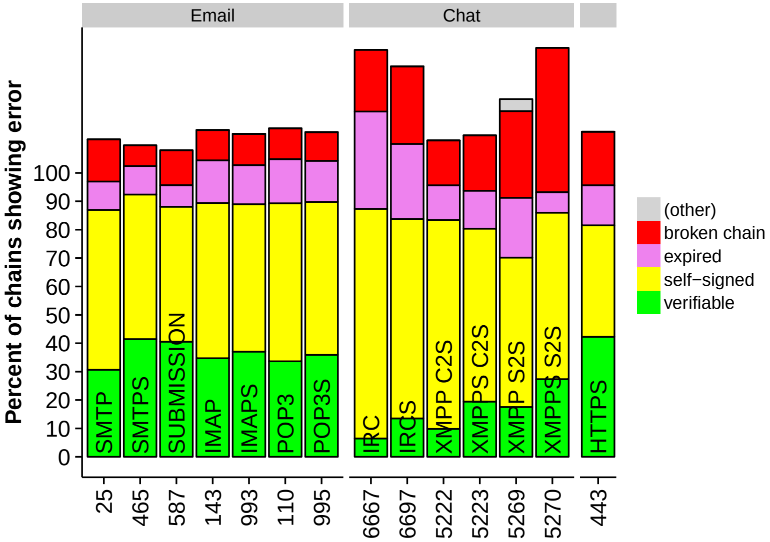


- RC4 has use (up to 17%, not good)
- ECDHE has much use
- DHE: 76% are 1024 bit, 22% 2048 bit, 1.4% are 768 bit

Responder authentication (monitored → use)



Responder authentication (scanned → deployed)



Initiator authentication: SUBMISSION

Combinations offered	Advertised	Servers
PLAIN, LOGIN	**2.1M**	**75.15%**
LOGIN, PLAIN	**224k**	**8.51%**
LOGIN, CRAM-MD5, PLAIN	96k	3.45%
LOGIN, PLAIN, CRAM-MD5	45k	1.63%
DIGEST-MD5, CRAM-MD5, PLAIN, LOGIN	36k	1.30%
CRAM-MD5, PLAIN, LOGIN	29k	1.04%
PLAIN, LOGIN, CRAM-MD5	25k	0.89%
…	…	…

- Plaintext-based methods the vast majority
- Even where CRAM is offered, it's usually not first choice
- No SCRAM

Risks and threats: SSL/TLS-level

STARTTLS

- Less than 50% of servers support upgrade
- But big providers do, have large share of traffic
- MITM vulnerability (reported to be exploited)

Ciphers

- For some protocols, 17% of RC4 traffic (WWW: 10%)
- For some protocols, $\approx$ 30% of connections not forward-secure
- Diffie-Hellman keys $\leq$ 1024 bit in $>$ 60% of connections

Risks and threats: authentication

Responder

- Many self-signed or expired certs, broken chains
- Big providers have correct setups
- Sending mail to ‘small’ domain/provider means risks of MITM
- We know from Foster *et al.* that mail servers do not verify certs in outgoing connections

Initiator

- Plain-text login pervasive
- CRAM not used much (and no implementations for SCRAM?)

Scans are intrusive

Scans are intrusive

Actually, that is so wrong. We do

```
nmap 0.0.0.0/0 | grep | sort -u | wc -l
```

Scans are intrusive

Let's show them what insights **only** scans can give.

Our example will be Internet routing!

The fragility of Internet routing

Connect with us

c|net | Search CNET | Reviews | News | Video | How To | Games | Download | Log In / Join | US Edition

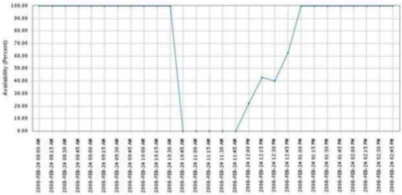
CNET › Tech Culture › How Pakistan knocked YouTube offline (and how to make sure it never happens again)

How Pakistan knocked YouTube offline (and how to make sure it never happens again)

YouTube becoming unreachable isn't the first time that Internet addresses were hijacked. But if it spurs interest in better security, it may be the last.

by Declan McCullagh @declanm / February 25, 2008 2:30 PM PST / Updated: February 25, 2008 4:28 PM PST

0 / / / / / more +



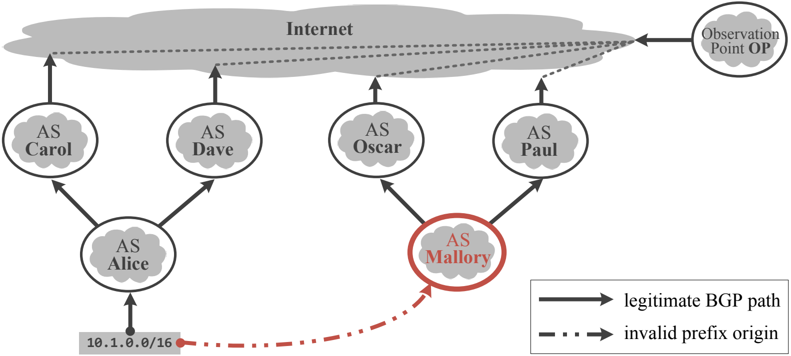
This graph that network-monitoring firm Keynote Systems provided to us shows the worldwide availability of YouTube.com dropping dramatically from 100 percent to 0 percent for over an hour. It didn't recover completely until two hours had elapsed.

Keynote Systems

THIS WEEK'S MUST READS /

1. How Pakistan knocked YouTube offline (and how to make sure it never happens again)
   Tech Culture
2. Google's Android Wear software will let you leave your phone at home (if there's Wi-Fi)
   Mobile
3. In Adobe's new Lightroom, multiple photos can now meld into one
   Photography
4. Yahoo's new search pact with Microsoft includes opt-out clause
   Internet
5. Microsoft rolling out 34 unscheduled patches for Windows today
   Operating Systems

Origin Relocation Attacks



Monitoring Internet routing

Attack detection systems for BGP exist

- But they mostly address other kinds of attacks
- Or they have enormous false-positive rates

So we built HEAP

- A filter chain to link to attack detection system
- A powerful system to rule out false positives
- The goal is to cut down the number of reported events to a more manageable size

Reason with external data

Idea: rule out benign events, investigate rest

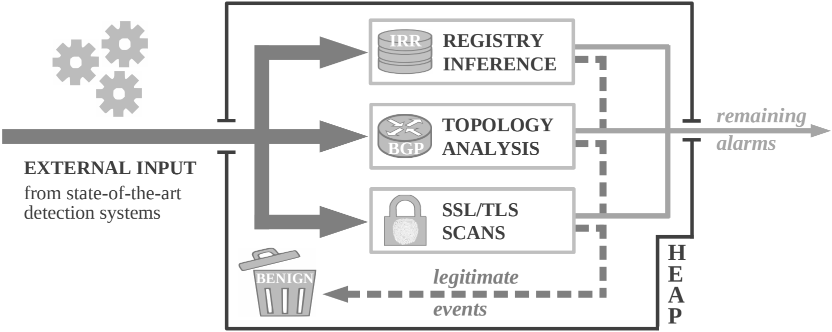


Figure: Hijacking Event Analysis Program (figure courtesy J. Schlamp)

Data source: SSL/TLS scans

IPv4-wide scans

- Create ground truth
  - Identify *beacon hosts* with *unique* keys
  - Filter out all hosts which were in suspicious prefix at scan-time
- With this ground truth:
  - During suspicious event, scan hosts in affected prefixes
  - If key is still the same: not an attack
  - Attacker unlikely to compromise both host(s) and BGP

How to evaluate

Lack of input sources

- Most attack detectors do not focus on subprefix attacks
- Or they are discontinued
- We thus had to build our own, very coarse, ‘detector’
- Essentially, we just counted every subprefix (subMOAS) events as an ‘attack’
- Gross overestimate of real attacks, but it creates a worst case for our evaluation setup
- We discounted events of less than 2 hrs duration

Evaluation results

	total	in %
All subMOAS events	**14,050**	**100.0%**
IRR analysis	5,699	40.56%
topology reasoning	2,328	16.57%
SSL/TLS scans	2,639	18.78%
Legitimate events (cum.)	**7,998**	**56.93%**

I.e. we can rule out more than half of **all** events in our super-coarse detector.

Case study: IP space of Top 1M (Alexa)

Assumption: this is valuable IP space

	total	in %
All subMOAS events	**849**	**100.0%**
IRR analysis	294	34.63%
topology reasoning	146	17.20%
SSL/TLS scans	576	67.85%
Legitimate events (cum.)	**689**	**81.15%**

One conclusion: run a Web server in your prefix, and you increase chances we can monitor your IP space.

Conclusion

A good step forward

- We can rule out 57% of **all** events shorter than 2 hrs
- For important IP spaces, this rises to 80%
- We can show commercial detectors have at least 10% false positives

We offer two conclusions

- IRR data is immensely useful—we wish operators would enter it into the DB more often
- Scans are very useful, too—and ‘opt-in’ to HEAP is as simple as setting up a small Web server with unique key

Summary

Security measurements point out weaknesses in email

- Connections between big providers are already (reasonably) secure
- The risk lies with mail from/to remaining providers
- Authentication mechanisms (initiator) are very poor
- (PS: The Web's security is a mess, too)

Scans can be immensely useful to improve security, too

- Monitor Internet routing and filter alarms

Summary

Security measurements point out weaknesses in email

- Connections between big providers are already (reasonably) secure
- The risk lies with mail from/to remaining providers
- Authentication mechanisms (initiator) are very poor
- (PS: The Web's security is a mess, too)

Scans can be immensely useful to improve security, too

- Monitor Internet routing and filter alarms

Questions?

email: `ralph.holz@sydney.edu.au`

Recommendations

A few things we can do

- Warnings in user agents that mail will be sent in plain
  $\rightarrow$ Google has implemented this now
- Flag-day for encryption (as for XMPP)
- Combine setup with automatic use of, *e.g.*, Let's Encrypt
- Ship safe defaults
- Follow guides, *e.g.*, `bettercrypto.org`
- More in the paper

Recommendations

A few things we can do

- Warnings in user agents that mail will be sent in plain
  $\rightarrow$ Google has implemented this now
- Flag-day for encryption (as for XMPP)
- Combine setup with automatic use of, *e.g.*, Let's Encrypt
- Ship safe defaults
- Follow guides, *e.g.*, `bettercrypto.org`
- More in the paper

Questions?

email: `ralph.holz@sydney.edu.au`

Can X.509 be reinforced?

No ‘silver bullet’ known that would resolve all issues

- Attacker model of SSL/TLS + X.509 $\approx$ protect credit card numbers
- State-scale attacks were not in scope back in the 1990s

New mechanisms

- Pinning: store client-local information about a site
- Store information in the DNS, use DNSSEC
- Notary principle
- Public logs

Attacker models important for assessment

- Weaker attacker:
  - E.g., on WiFi access point, or some local network gateway
  - May control DNS traffic, but cannot interfere with DNSSEC
- Regional attacker:
  - Controls all traffic of a country
  - Control over routing, control over DNS
  - Controls own top-level domain (DNSSEC!)
  - May compromise CA
- Supra-regional attacker. Same as above, plus:
  - ‘Cyber-war’: a state risking ‘digital military confrontation’
  - Attacks on global routing (BGP, possible)
  - Attacks on infrastructure to control DNSSEC

Hardening certification

Vendor efforts

- CA/Browser Forum is a body of browser vendors and CAs
  - Extended Validation standard (2010)
  - Baseline Requirements standard (2012)

Extended Validation (EV)

- Require state-issued documents before certification
- Certificates have OID that browsers evaluate

Base Line Requirements

- Minimum requirements for validation, forbid less secure practices

Discussion of these standards

- Sanctions for standard violations unclear
  - What justifies removal from root store?
- CAs have repeatedly violated the standards agreed upon:
  - Certificates without revocation information
  - Certificates with keys that are too short
  - Certificates with expiry periods that are too long
- These standards address operational practices, but are hard to enforce
- The standards do not address stronger attackers, e.g., a compromised CA like DigiNotar

Pinning

Defence against rogue CAs issuing malicious certs

- Idea: client stores information about a host/Web site on first contact
- *E.g.*, store the public key of a site
- Use this information to reidentify a site later
- *E.g.*, if public key is suddenly different on next connect: warn user

Pinning assumes a secure first connection

- Thus also known as 'trust-on-first-use'
- Inherent bootstrapping problem

Two pinning variants

Static pinning

- Preloaded pins for important sites:
  - Implemented in Google Chrome and Mozilla Firefox
- User-driven pinning:
  - add-ons for browsers that allow users to store and compare public keys of sites

Dynamic pinning

- Idea: communicate helpful information to aid clients with pinning

Issues with pinning

- For certain users, secure first contact may not be possible
  - *E.g.*, dissidents in authoritarian countries
- Life-cycle problem
  - Servers may (legitimately) update/upgrade their keys—synchronise pinning information
- Scalability
  - Browsers cannot come preloaded with pins of all sites

HSTS: HTTP Strict Transport Security

- Dynamic ‘pinning’: tell clients that this site supports HTTPS
- Example:
  `Strict-Transport-Security: max-age=31536000; includeSubDomains`
- Instructs browser:
  - To expect HTTPS for next 12 months, including subdomains
  - To redirect on port 443
  - To disallow user override of certificate warnings
- Simple, powerful
- Very little danger for server operators to misconfigure (and lose customers)

HPKP: HTTP Public Key Pinning

- Dynamic pinning
- Servers communicate life-time and hash value of their X.509 public key in the HTTP header
  - `Public-Key-Pins: pin-sha256="cUPcT..."; max-age=5184000; report-uri="https://www.example.net/hpkp-report"`
- Addresses short-comings of simple pinning:
  - Life-cycle management for key upgrade/compromise: ‘backup pins’ communicated in addition to the primary ones
- Easy to deploy, no problems for clients that are not aware of the pinning
- Features reporting function: report key mismatches to a URL!

Asssessment of pinning

- Extremely strong if assumption of secure first connection holds
- Attacker can only attack client or server, but there is no other Trusted Third Party to compromise
- Practical usefulness first demonstrated by Google:
  - Google pins all Google sites in their browser (static pinning)
  - This was how the DigiNotar incident was detected!
- Concept can hold up to any attacker who cannot compromise either client or server
  - Hence, addresses the stronger forms of attacker

Cross-validation with public logs

- **Idea:** log information about certificate issuance with a number of distinct parties
- Logs store information *publicly* and *append-only*: audit trail
  - No way to delete previous entries
  - Sign and timestamp new entries
- **Certificate Transparency (CT):**
  - Make **transparent** who issued certificates to whom and when
  - **Anyone** can verify logs' content and their correct operation
  - Enables detecting rogue CA issuing certificates for a domain
  - Proposed: 30+ logs around the globe, run by different parties
  - Note: goal is detection, not a direct defence for clients!

Public log: a Merkle Hash Tree

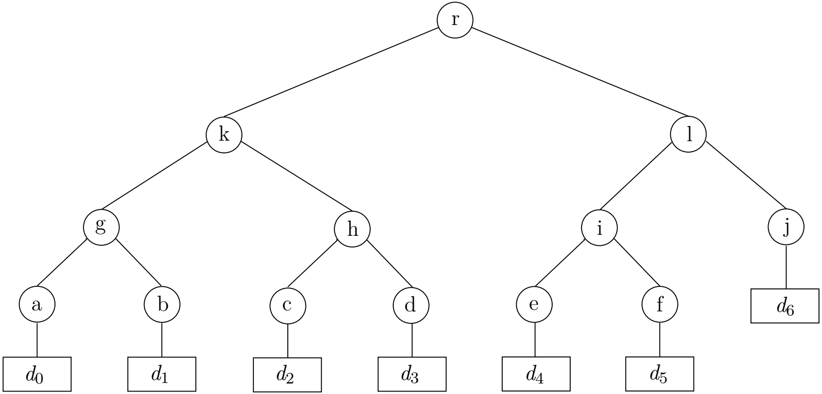


Figure: Log is a Merkle tree, d_i are new certificate chains.

Proving properties of Merkle Hash Trees

The tree structure is beneficial for proving certain conditions are met

- Proofs do not require full copies of the tree—a subset, logarithmic in size, is enough
- Algorithms to determine the subsets, and how to carry out the proofs, are described in RFC 6962
- Logs must allow to retrieve the necessary subset for any given certificate in the tree
- So-called monitors and auditors are entities that continously watch the operation of logs and use these proofs to determine the logs are well-behaving
- **Cross-validation**: watching the watchers

Proofs

Consistency

- Prove the append-only property
- Prove that no certificate was removed from the tree, or some certificate injected in the wrong position
- Works by obtaining subset of nodes needed to prove that tree from a certain moment t_0 on always adhered to the append-only property
- In other words: the logs cannot fake the logged history once they have started logging

Inclusion (audit path proof)

- Prove that a certificate has been included in the tree

Watchers: monitors

Computationally powerful entities tracking the operation of several logs

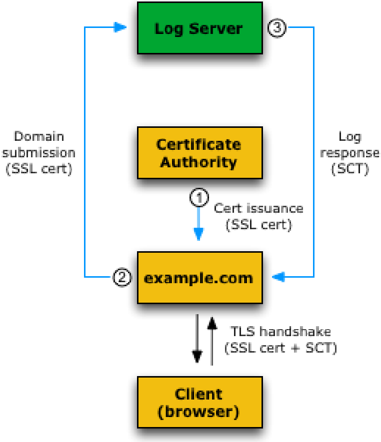
- Primary function: continously verify the *append-only* property (consistency checks)
- Act on behalf of less powerful entities, e.g. browsers or domain owners
- Possible parties fulfilling this role: ISPs, CAs. But anyone is free to set up a monitor.
- Secondarily, they may also keep copies of logs
- This enables them to search for violating certificate issuances:
  - E.g. they have a list of domains to 'protect'
  - They may watch continously if a second certificate for a domain appears, which the domain owner never authorised

Watchers: auditors

Auditors are computationally less powerful entities

- Typically, they do not keep copies of the logs
- Typical parties fulfilling this role: browsers
- Auditors may check either consistency (like monitors, but without having copies of the logs)
- They may also do inclusion checks

Figure: logs and TLS/X.509



Source: certificate-transparency.org

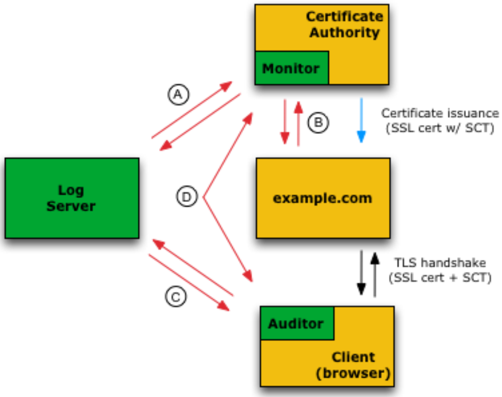
Interactions: logs and other parties

Certification: CAs and logs

- When issuing a cert, CA must send it to at least two logs for incorporation
- Log returns a Signed Certificate Timestamp (SCT) proving it has accepted the cert
- SCT must be forwarded to actual domain operator
- Client learns about SCT
  - with OCSP request (stapling, current status), or
  - can retrieve as DNS record,
  - or as TLS extension or,
  - CA may directly add to X.509 cert
- SCT is sent to any TLS client connecting to the domain: client knows which logs track this cert

Browsers can be auditors, CAs can be monitors

This is an example configuration—anyone can audit or monitor



Source: certificate-transparency.org

Gossiping

Problem: split-horizon attacks

- Monitors and auditors cannot prevent logs from keeping ‘alternate histories’, where one history is the real one, shown some parties, and the other is a fake one, shown to other parties
- With considerable effort, such split-horizon attacks can be used by attackers to bypass the cross-validation system and trick clients
- Thus: gossiping between auditors, monitors, TLS clients
- Gossiping is not yet specified, but here are the main ideas:
  - Clients, auditors and monitors should notify domains which tree head they see
  - This means that logs showing alternate histories to some clients will be ultimately detected

Discussion of Certificate Transparency

Advantages

- Adds transparency to X.509 in the hope of detecting malicious behaviour early
- Google pushed this into the market
- Has already proven its worth on several occasions
- CT is strong reinforcement ofr X.509, thwarting even state-level attackers

Problems

- Expensive!
- No direct, immediate help for clients
- Very complex setup

Certificate Authority Authorization (CAA)

Store which CA is responsible for a domain in the DNS

- *E.g.*, Google may add value `symantec.com` to resource records for `google.com`
  - Try it: `dig +short -t TYPE257 google.com`
- The value is a unique identifier for a CA
- Before issuing a certificate for a domain, a CA must query the CAA record
- Also define a URL where one can report violations, e.g. if you find a certificate that is not from the CA defined in the CAA record
- Problem: DNS itself is not secure

Discussion of CAA

Advantages

- Very simple, cheap
- CAs can quickly query if the domain owner **wants** them to be responsible
- Avoiding DNSSEC reduces complexity
- The URL for reporting is very valuable addition

Issues

- No DNSSEC means well-positioned attacker can interfere with DNS query (even weakest attacker we discussed)
- No direct protection for **clients**
- No defence at all when a CA is compromised

DANE: DNS-based authentication of named entities

TLSA record: additional trust anchor

- Very flexible ways to store trust anchor in DNS entry
- *Selector* field:
  - Store full certificate or just public key
- *Matching Type* field:
  - Exact value provided or hash value (SHA256 or SHA512)
- *Certificate Usage* field specifies:
  - Cert (or public key) of issuing CA
  - Cert (or public key) of end-host certificate if CA-issued
  - Cert (or public key) of self-signed certificate
- DANE-TLSA mandates use of DNSSEC
  - Mandates to abort connection on mismatch between DNS entry and TLS cert

Discussion of DANE-TLSA

Advantages

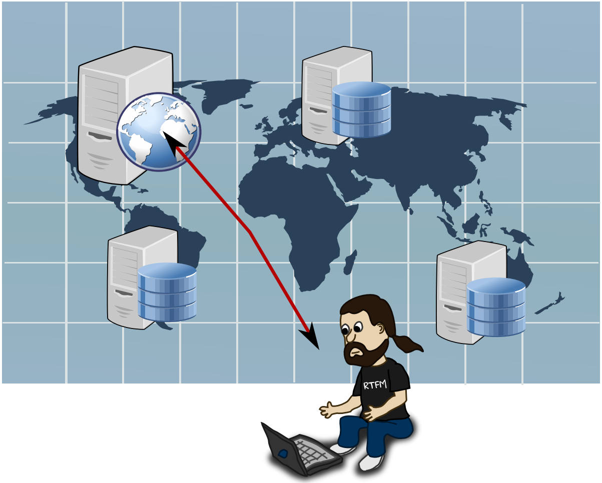
- Out-of-band mechanism with strong reassurance on certificate validity
- Protects completely against our weaker, local attacker

Potential issues

- DNS operators need to become PKI operators—requires extra care and training
- Mandated 'hard fail' is *disincentive* for operators—same
- Countries are often in control of their TLDs—think of `bit.ly`. This enables state-level attacks:
  - Regional attacker: can modify TLSA records of his zone
  - Global attacker: may be able to modify some other zones

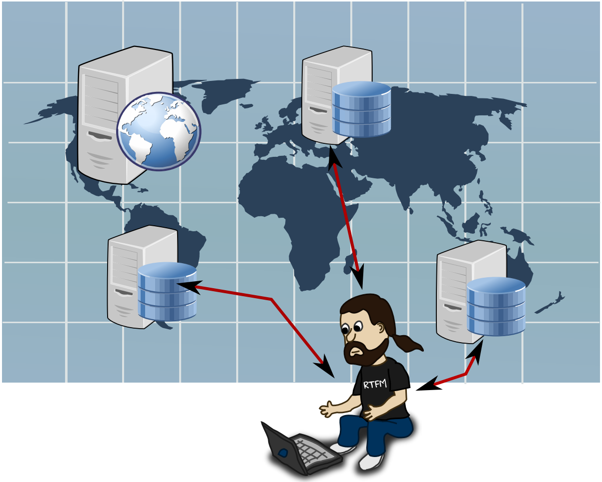
Notary-based systems

When connecting to a host and receiving the TLS certificate…



Notary-based systems

…connect to some special notaries elsewhere and double-check



Perspectives: a notary system

- Assumption: no attacker can control all paths through the Internet
- A number of notary systems are distributed around the globe, run by independent operators
- Notaries **scan** a list of domains regularly. Store and sign which certificates they see, at which time.
- Each notary also **shadows** a number of other notaries:
  - Downloads their observations and signs and stores them, too
  - Checks for inconsistencies: no contradicting entries
  - Defence against misbehaving or compromised notaries
- When clients connect to a domain, they receive a certificate. They double-check with 1-2 notaries **and** their shadows.

Discussion: Perspectives

- Security depends on:
  - Attacker's capability to compromise notaries and his position in network
  - Attacker being able to predict which notaries and shadows a client will use
  - Many notaries necessary—else attacker can compromise 'just enough' of them
  - Attacker sitting on 'last hop' to server can trick all notaries
- Huge problem: which notaries should a user trust?
  - Most users do not have background for such a decision
  - Preconfigure it? Then everyone uses the same notaries
- In practice, notary systems have so far *failed* because they are not acceptable to the typical user base.

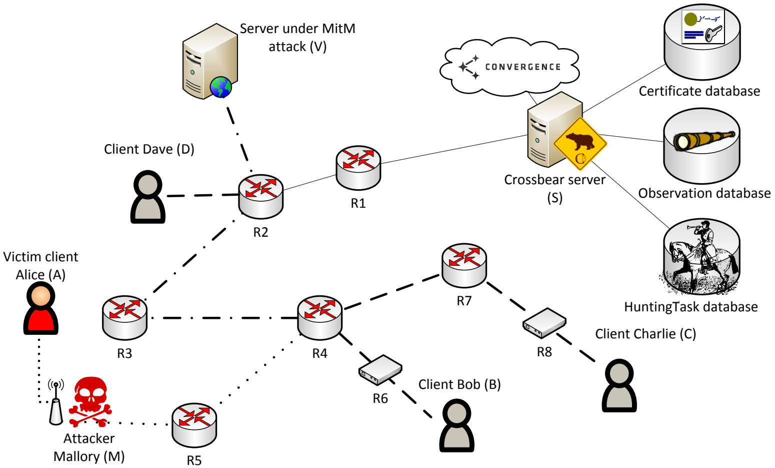
Notary-based systems

Examples

- Perspectives (CMU, 2009): browser plug-in
  - In operation
  - But shadow concept never implemented
  - Few notaries—project cannot guarantee their benign intentions
- Convergence (Marlinspike, 2011): browser plug-in, discontinued
- Crossbear (Holz, 2011):
  - Different goal: detect attacks by finding mismatches between notaries and clients
  - Interpret a mismatch as potential attack, try to determine position of attacker

Crossbear

Goal: *detection and localisation*



Current status and gazing into crystal ball

- Certificate Transparency is supported and deployed for EV certs
  - Has detected misbehaving logs and CAs
- HSTS seems to have some traction among important sites
  - But HPKP has little deployment: risk to operator
- DANE-TLSA has little deployment so far (as does DNSSEC)
- Notary concepts have no deployment to mention

On XMPP

Majority of certs for XMPP are self-signed.

- Inspection of Common Names shows: proprietary use
  - Content Distribution Network (`incapsula.com`)
  - Apple Push
  - Samsung Push
  - Unified Communication solutions

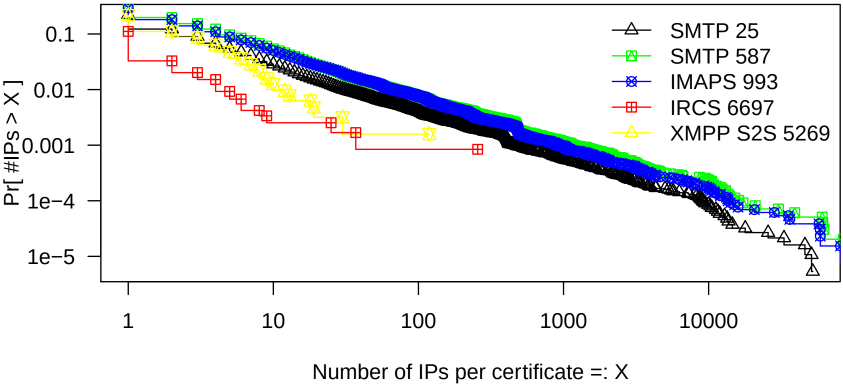
Oddity of scans

The Internet has background noise.

- Independent of port you scan, about 0.07-0.1% of IPs reply with SYN/ACK, but do not carry out a handshake
- Confirmed with authors of `zmap`
- Important to keep in mind when investigating protocols with smaller deployments, where SSL/TLS does not seem to succeed very often

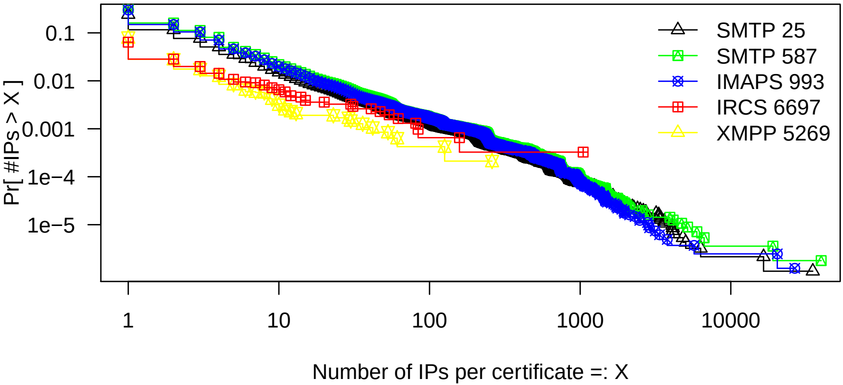
Certificate reuse—valid certs

Much reuse, even among valid certs

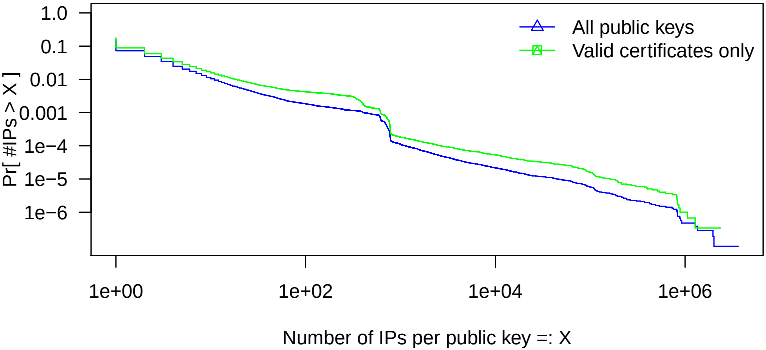


Certificate reuse—self-signed

Many default certs from default configurations



Key reuse across *all* protocols



Oddity in IMAPS...

Common name	Occurrences
*.securesites.com	88k
*.sslcert35.com	31k
localhost/emailAddress=webaster@localhost	27k
localhost/emailAddress=webaster@localhost	21k
*.he.net	19k
www.update.microsoft.com	19k
*.securesites.net	11k
*.cbeyondhosting2.com	11k
*.hostingterra.com	11k
plesk/emailAddress=info@plesk.com	6k

Table: Selected Common Names in IMAPS certificates.

Oddity in IMAPS…

Common name	Occurrences
*.securesites.com	88k
*.sslcert35.com	31k
localhost/emailAddress=webaster@localhost	27k
localhost/emailAddress=webaster@localhost	21k
*.he.net	19k
www.update.microsoft.com	19k
*.securesites.net	11k
*.cbeyondhosting2.com	11k
*.hostingterra.com	11k
plesk/emailAddress=info@plesk.com	6k

Table: Selected Common Names in IMAPS certificates.

Mapping to ASes

AS number	Registration information	CIRCL rank
3257	TINET-BACKBONE Tinet SpA, DE	9532
3731	AFNCA-ASN - AFNCA Inc., US	4804
4250	ALENT-ASN-1 - Alentus Corporation, US	9180
4436	AS-GTT-4436 - nLayer Communications, Inc., US	10,730
6762	SEABONE-NET TELECOM ITALIA SPARKLE S.p.A., IT	11,887
11346	CIAS - Critical Issue Inc., US	557
13030	INIT7 Init7 (Switzerland) Ltd., CH	6255
14618	Amazon.com Inc., US	4139
16509	Amazon.com Inc., US	3143
18779	EGIHOSTING - EGIHosting, US	4712
21321	ARETI-AS Areti Internet Ltd.,GB	2828
23352	SERVERCENTRAL - Server Central Network, US	11,135
26642	AFAS - AnchorFree Inc., US	–
41095	IPTP IPTP LTD, NL	6330
54500	18779 - EGIHosting, US	–