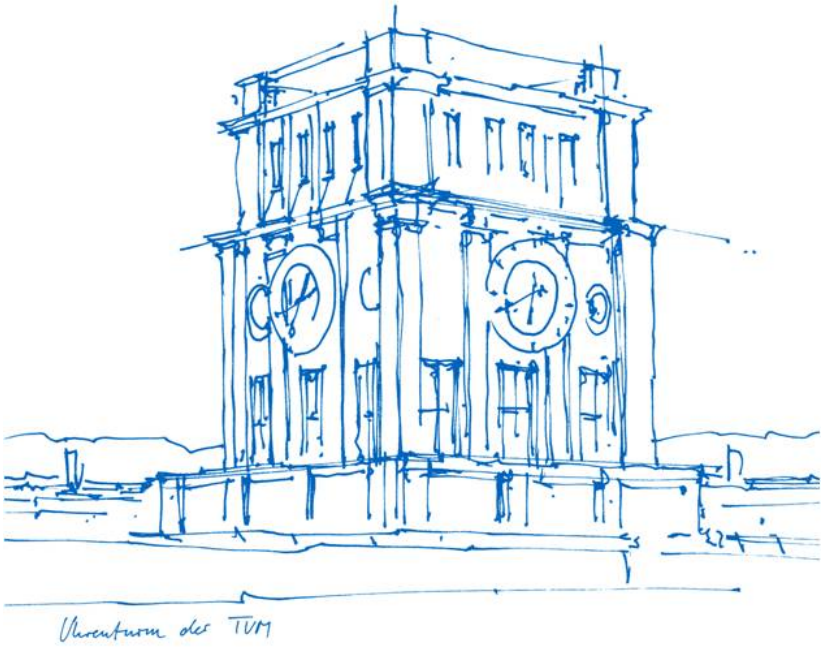
Network Security: Topics in Web Security

Quirin Scheitle

Technische Universität München

30.1.2018



Agenda

- **Evolving the TLS standard**
  - Privacy leaks and other drawbacks in TLS 1.2
  - TLS1.3 standardization and its controversies
- Topics in HTTPS Security
  - Certificate Transparency – transmission modes
  - TLS downgrade protection
  - Host Strict Transport Security
  - Certification Authority Authorization (CAA)

Agenda

- **Evolving the TLS standard**
  - **Privacy leaks and other drawbacks in TLS 1.2**
  - TLS1.3 standardization and its controversies
- Topics in HTTPS Security
  - Certificate Transparency – transmission modes
  - TLS downgrade protection
  - Host Strict Transport Security
  - Certification Authority Authorization (CAA)

A Brief Introduction to TLS

- TLS is a popular encryption protocol in the Internet, with HTTPS one of its main users
- TLS and SSL have a long history, with initial standardization of SSLv3 in 1995 and TLS1.0 in 1999
- TLS applies many of the principles we have developed for our crypto protocols:
  - Perfect Forward Secrecy (though optional)
  - Upgrade Compatibility : Client offers algorithms, server selects
  - If you were to use a cryptographic protocol in something you build, it would probably be TLS

TLS1.2 Handshake Intro

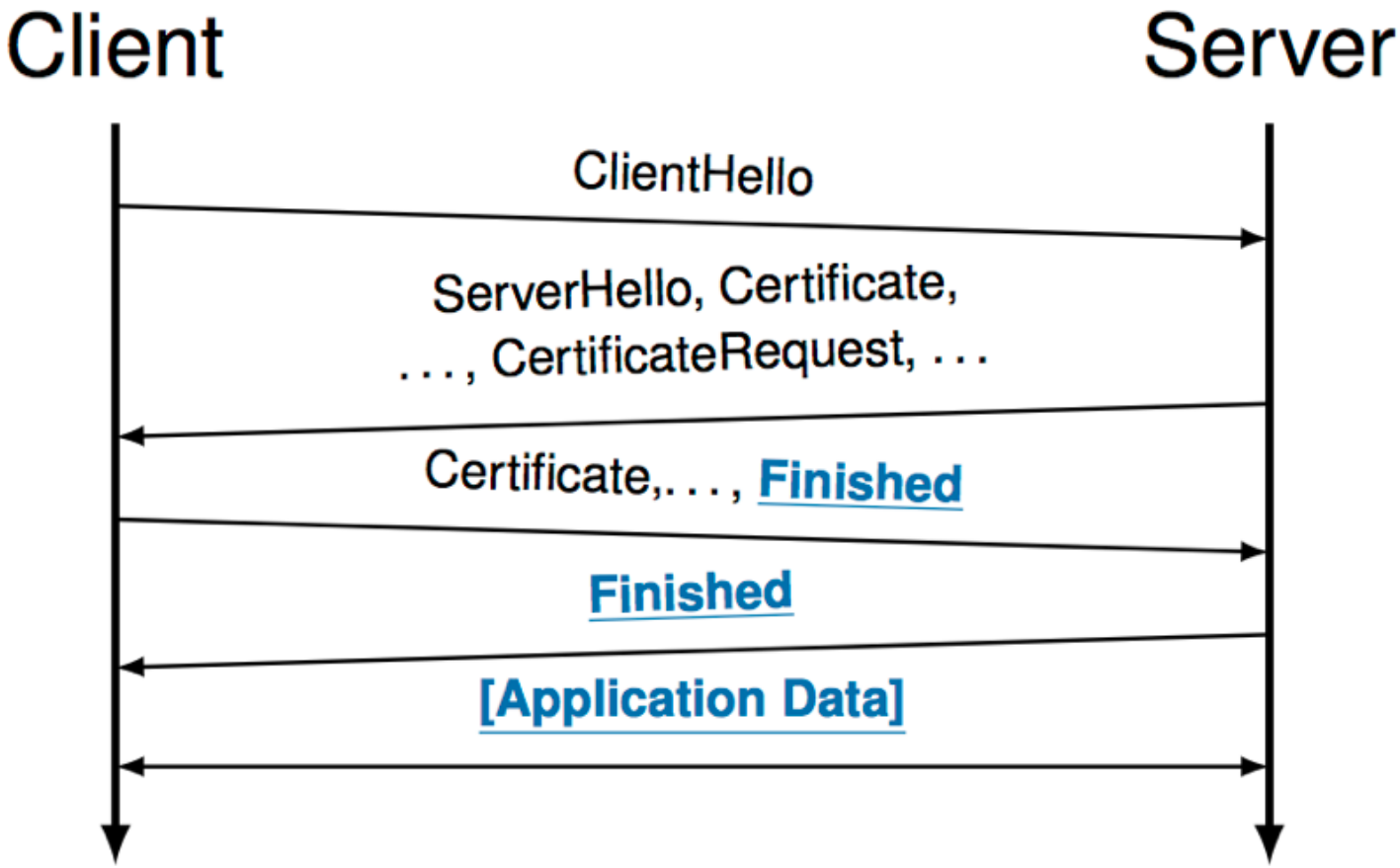


Figure: TLS 1.2 handshake, Unencrypted Data, [Encrypted Data]

TLS 1.2 Handshake Intro

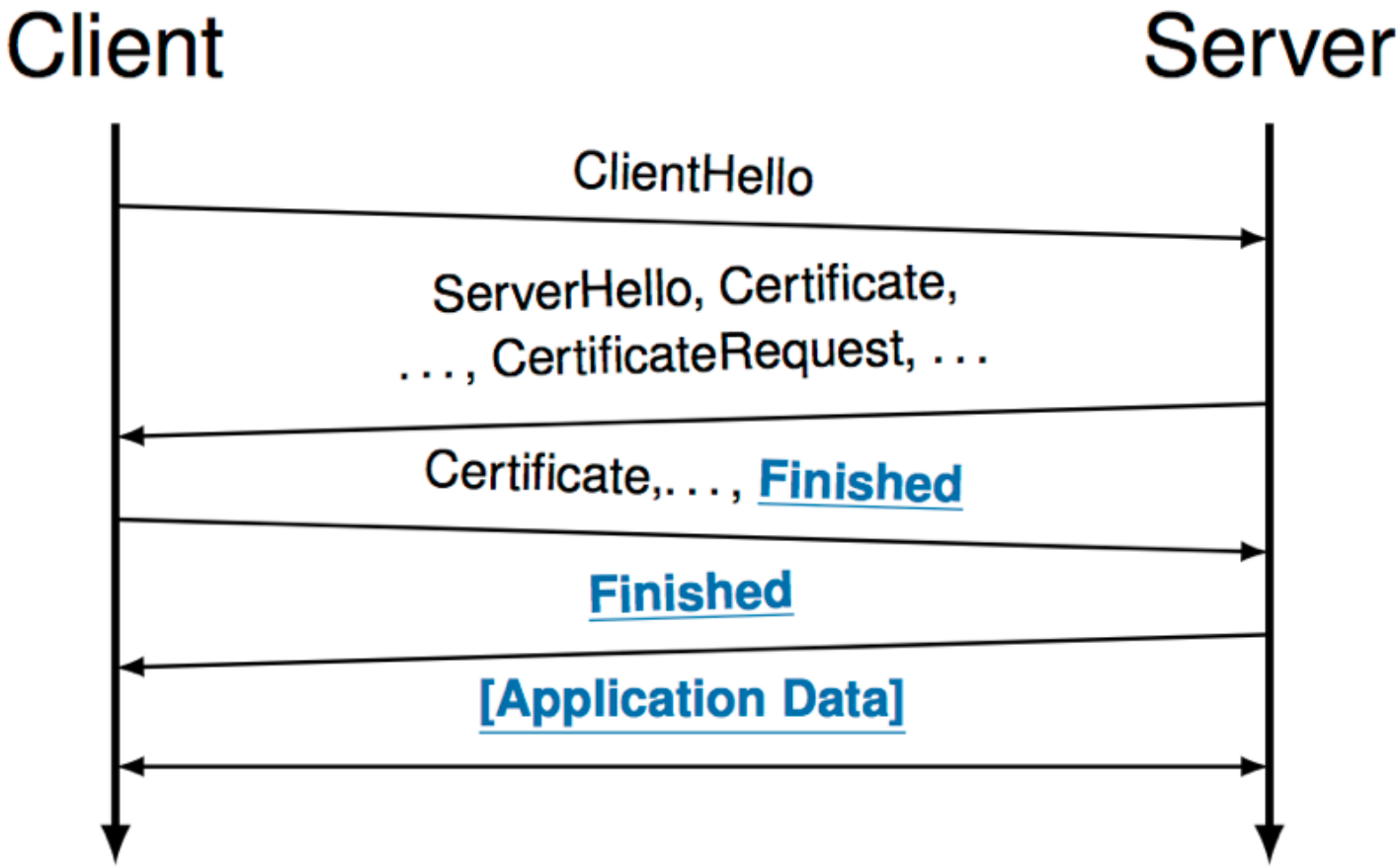


Figure: TLS 1.2 handshake, Unencrypted Data, [Encrypted Data]

Do you see any problems?

Drawbacks in TLS1.2 – Handshake recall:

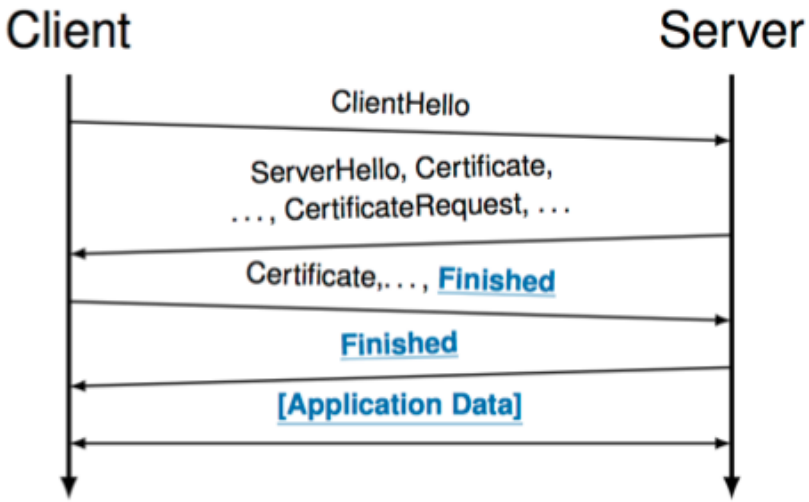


Figure: TLS 1.2 handshake, Unencrypted Data, [Encrypted Data]

- SNI unencrypted (part of ClientHello)
- Server Certificate unencrypted
- Client Certificate unencrypted
- ***Problematic?***

Drawbacks in TLS1.2 – Handshake recall:

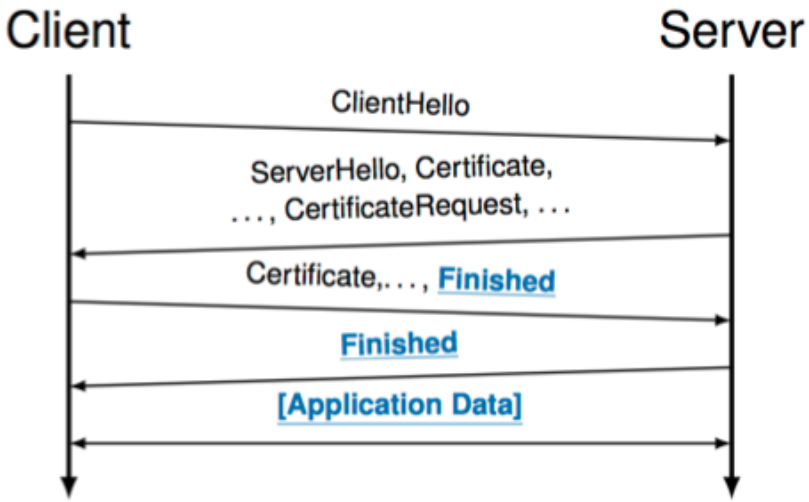


Figure: TLS 1.2 handshake, Unencrypted Data, [Encrypted Data]

- SNI unencrypted (part of ClientHello)
- Server Certificate unencrypted
- Client Certificate unencrypted
- ***Problematic? → Severe privacy and censorship implications***

TLS 1.2 Client Certificate Authentication (CCA)

Where is CCA used?

- **Network authentication**: 802.1x EAP
- **VPN**: OpenVPN, F5 EdgeConnect, . . .
- **Web**: HTTPS
- **IoT**: MQTT
- **Remote device management,** for example MobileIron
- **Apple Push Notification Service (APNs)**

Apple Statistics:

- 1 billion active devices (2016)
- 800 million iTunes accounts (2014)

Push Notification Services

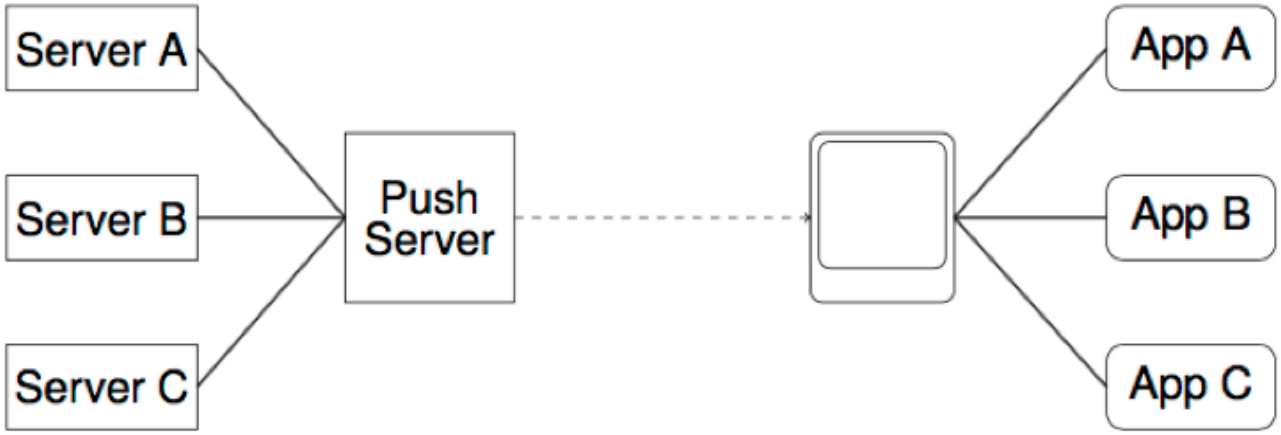


Figure: Push Service Architecture: Messages brokered to Apps through the Push Notification Service.

Resource efficient notification of (mobile) applications:

- **Apple's APNs:** OS, MacOs, iTunes
- **Google's FCM:** Android, Chrome
- **Microsoft's WNS:** Windows, Windows Phone

Paradigms:

- Tightly integrated with operating system
- Always connected to backend

Problem of combining TLS-CCA with APNs

- APNs is “always-on”: One of the first things your phone does when joining a new network is logging into the APNs service
- Upon login, a cryptographically unique client certificate is transmitted over the network in plain text
- This permits precise tracking of individual devices* through all intermediate parties
  - * For mobile devices such as phones and laptops, the correlation of device and user movement is very strong

Responsible Disclosure

We informed Apple's product security team before publication:

- Contact with OpenPGP secured mail
- Very quick response
- Several phone calls, continuous contact
- Several engineers in calls and working on resolution

Impact:

- MacOS & iOS fixed with January 2017 security patches
- APNs Backend patched
- iTunes on Windows patched a bit later (SChannel is complicated . . .)

What now?

Push TLS 1.3 standardization which encrypts certificates

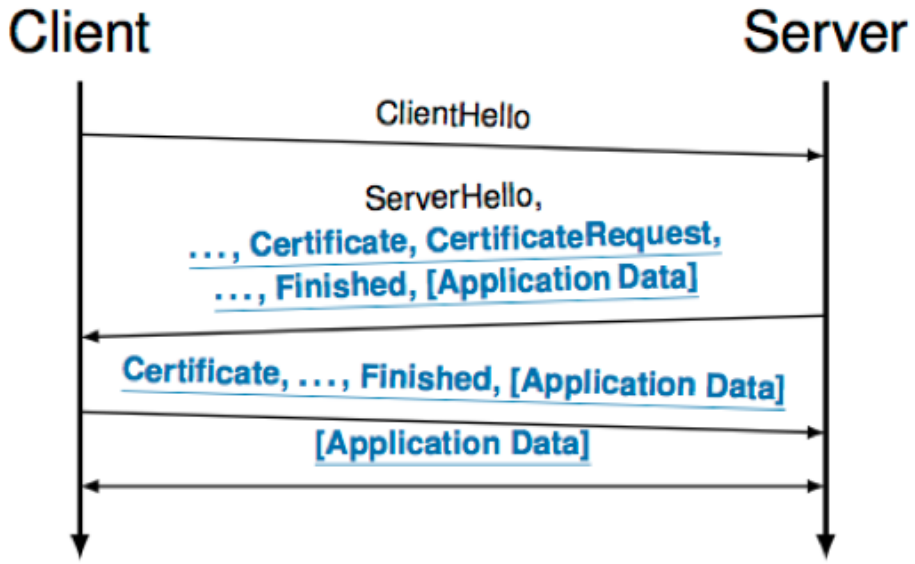


Figure: TLS 1.3 handshake, Unencrypted Data, [Encrypted Data]

But: ClientHello Extensions still unencrypted:

- Server Name Indication (SNI)
- Application-specific data

TLS 1.2 privacy leaks summary

- Non-encrypted SNI, Client and Server Certificates are severe privacy leaks that permit for user monitoring, tracking, and censorship

- Background reading:

Matthias Wachs, Quirin Scheitle, Georg Carle, “Push Away Your Privacy: Precise User Tracking Based on TLS Client Certificate Authentication,”in Network Traffic Measurement and Analysis Conference (TMA), Jun. 2017

Agenda

- **Evolving the TLS standard**
  - Privacy leaks and other drawbacks in TLS 1.2
  - **TLS1.3 standardization and its controversies**
- Topics in HTTPS Security
  - Certificate Transparency – transmission modes
  - TLS downgrade protection
  - Host Strict Transport Security
  - Certification Authority Authorization (CAA)

TLS 1.3 Standardization

- IETF RFC process, currently in draft 23:
  - https://tools.ietf.org/html/draft-ietf-tls-tls13-23
  - At 154 pages, quite a long RFC
  - Good discussion insights from mailing list:
    - https://www.ietf.org/mail-archive/web/tls/current/maillist.html

TLS 1.3 Standardization – Key Points

- “0-RTT”
  - TLS1.2 requires another RTT on top of TCP setup before sending data – this negatively affects user experience
  - TLS1.3 offers 0-RTT connections for known hosts

TLS 1.3 Standardization – Key Points

- “0-RTT”
  - TLS1.2 requires another RTT on top of TCP setup before sending data – this negatively affects user experience
  - TLS1.3 offers 0-RTT connections for known hosts
- Removal of static key use
  - TLS1.3 aimed to remove static key use to enforce perfect forward secrecy (PFS)
  - Pushback from network monitoring community, highly controversial

TLS 1.3 Standardization – Key Points

- "0-RTT"
  - TLS1.2 requires another RTT on top of TCP setup before sending data – this negatively affects user experience
  - TLS1.3 offers 0-RTT connections for known hosts
- Removal of static key use
  - TLS1.3 aimed to remove static key use to enforce PFS
  - Pushback from network monitoring community, highly controversial
- Deployment Problems
  - "Security" applications monitor/intercept traffic ("middle boxes")
  - The middle boxes may only permit "known" traffic, leading to "ossification" of the Internet – deployment of new protocols has become difficult
  - Experiments by Chrome and Firefox to enable TLS1.3 in early 2017 resulted in some middle boxes blocking TLS1.3 connections or even crashing
  - Proposed solutions:
    - Make TLS1.3 look similar to TLS1.2
    - "Greasing": Make sure few fields in a protocol are static so that middle boxes can not "rust" on them (door hinge analogy)

Agenda

- Evolving the TLS standard
  - Privacy leaks and other drawbacks in TLS 1.2
  - TLS1.3 standardization and its controversies
- **Topics in HTTPS Security**
  - Certificate Transparency – transmission modes
  - TLS downgrade protection
  - Host Strict Transport Security
  - Certification Authority Authorization (CAA)

Background reading for “Topics in HTTPS Security”:

Johanna Amann, Oliver Gasser, Quirin Scheitle, Lexi Brent, Georg Carle, Ralph Holz, “Mission Accomplished? HTTPS Security after DigiNotar,” IMC’17

Topics in HTTP Security – Background

- The selection of the following topics is based on current research done at our chair
- The methodology of these studies is heavily focused on *measuring* security at a large scale
- This provides answers to what security mechanisms are being used at scale, how they are being used, and what future security mechanisms should look like

- Typical methodology of large-scale scans:
  - Gather ~300M domains (.de ~18M, .com ~132M)
  - Connect to every domain and evaluate responses
  - Fast, but stable scanning is challenging
  - Sometimes combine with passive measurements for additional insights (for example, which features are actually being used by real clients?)

Agenda

- Evolving the TLS standard
  - Privacy leaks and other drawbacks in TLS 1.2
  - TLS1.3 standardization and its controversies
- Topics in HTTPS Security
  - **Certificate Transparency – transmission modes**
  - TLS downgrade protection
  - Host Strict Transport Security
  - Certification Authority Authorization (CAA)

Background reading for “Topics in HTTPS Security”:

Johanna Amann, Oliver Gasser, Quirin Scheitle, Lexi Brent, Georg Carle, Ralph Holz, “Mission Accomplished? HTTPS Security after DigiNotar,” IMC’17

Certificate Transparency – Recap

- Public logs that certificates can be logged to
- Upon submission of a certificate, the log responds with a signed certificate timestamp (SCT) – a promise to include a certificate into the log in the near future
- A client can use SCTs to validate that a certificate has been logged – Chrome will require all new (i.e. notValidBefore > March 31) certificates to be logged [1] starting April 1st
- But: How to provide SCTs to browsers?

[1] https://groups.google.com/a/chromium.org/forum/#!msg/ct-policy/sz_3W_xKBNY/6jq2ghJXBAAJ

CT – 3 distribution mechanisms

1. Embed SCT in x509 certificate: Very little can go wrong, users do not have to do anything
2. Deliver SCT via a TLS extension – lot of user effort, easy to get wrong
3. Deliver SCT via OCSP stapling – OCSP stapling is little used, relies on responsive OCSP servers at CA, frequent queries to CA OCSP server

Method 1 stands out as easy to use and resilient.

But... Would including the SCT in the certificate not change the certificate and break the signature?
Yes!
Solution: CAs submit a *precertificate* to CT logs. These precertificates are a promise of the CA to issue an exact copy of the percertificate, but with SCTs included.

Example: https://crt.sh/?id=245604874
Task: Find the according full certificate – is it the same?

CT – 3 distribution mechanisms

1. Embed SCT in x509 certificate: Very little can go wrong, users do not have to do anything
2. Deliver SCT via a TLS extension – lot of user effort, easy to get wrong
3. Deliver SCT via OCSP stapling – OCSP stapling is little used , relies on responsive OCSP servers at CA, frequent queries to CA OCSP server

Measurement results for distribution methods (~April ,17), across all certificates with SCT, show very ckear preference of x509 embedding:

with SCT	868.5k
via X.509	867.6k
via TLS	885
via OCSP	49

Johanna Amann, Oliver Gasser, Quirin Scheitle, Lexi Brent, Georg Carle,Ralph Holz, “Mission Accomplished? HTTPS Security after DigiNotar,” IMC‘17

Agenda

- Evolving the TLS standard
  - Privacy leaks and other drawbacks in TLS 1.2
  - TLS1.3 standardization and its controversies
- Topics in HTTPS Security
  - Certificate Transparency – transmission modes
  - **TLS downgrade protection**
  - Host Strict Transport Security
  - Certification Authority Authorization (CAA)

TLS downgrade attacks and their protection

- Assume an attacker has the capability to break TLS1.1
- But the victim-to-be prefers TLS1.2
- Attacker plan: Interfer with TLS1.2 handshakes – victim will downgrade to TLS1.1

TLS downgrade attacks and their protection

- Assume an attacker has the capability to break TLS1.1
- But the victim-to-be prefers TLS1.2
- Attacker plan: Interfer with TLS1.2 handshakes – victim will downgrade to TLS1.1

- How to protect? RFC 7507:
- If a client has tried a higher TLS version previously, it appends an Signaling Cipher Suite Value (SCSV) to the TLS1.1 handshake:

  TLS_FALLBACK_SCSV {0x56, 0x00}
- This SCSV tells the server that the client supports a higher TLS version
- If the server supports a higher TLS version, it must cancel the connection with an *inappropriate_fallback* alert

TLS downgrade attacks and their protection

- Is this RFC7507 downgrade protection SCSV a good idea?
  - It is easily applicable (no user input, simple TLS library update)
  - It comes with little risk

Of 51M domains we have scanned in April‘17, 96% correctly deployed SCSV → widespread deployment

Agenda

- Evolving the TLS standard
  - Privacy leaks and other drawbacks in TLS 1.2
  - TLS1.3 standardization and its controversies
- Topics in HTTPS Security
  - Certificate Transparency – transmission modes
  - TLS downgrade protection
  - **Host Strict Transport Security**
  - Certification Authority Authorization (CAA)

Host Strict Transport Security (HSTS)

- Problem: If a user visits http://www.tum.de, they will be redirected using a HTTP redirect to the secure https://www.tum.de
- From this point on, all communication will be secure – but that initial redirect is insecure, an attacker can redirect to other pages using MITM techniques

- Solution idea: Instruct the browser that www.tum.de is to be used strictly with secure transports (HTTPS) for a certain amount of time. Exemplary header:

```
Strict-Transport-Security: max-age="31536000“
```

This header instructs the browser to visit the current website only using HTTPS for the next `31536000` seconds – which corresponds to 1 year.

Is this good?
+ Easy to deploy
+ Little risk
- Requires website operator action

Background: RFC 6797

HSTS Refinements

- Using the includeSubDomains parameter, a domain operator can enforce HSTS for all subDomains of the current domain. If tum.de would set:

  `Strict-Transport-Security: max-age=31536000; includeSubDomains`

  … all subdomains would only be accessible through HTTPS, even `insecuresite.subdomain.net.in.tum.de`

- HSTS still relies on Trust-on-First-Use (TOFU) – the initial insecure redirect, before setting the header, would still be susceptible to attack.

- How to solve this problem?

HSTS Refinements

- Using the includeSubDomains parameter, a domain operator can enforce HSTS for all subDomains of the current domain. If tum.de would set:

  `Strict-Transport-Security: max-age=31536000; includeSubDomains`

  … all subdomains would only be accessible through HTTPS, even `insecuresite.subdomain.net.in.tum.de`

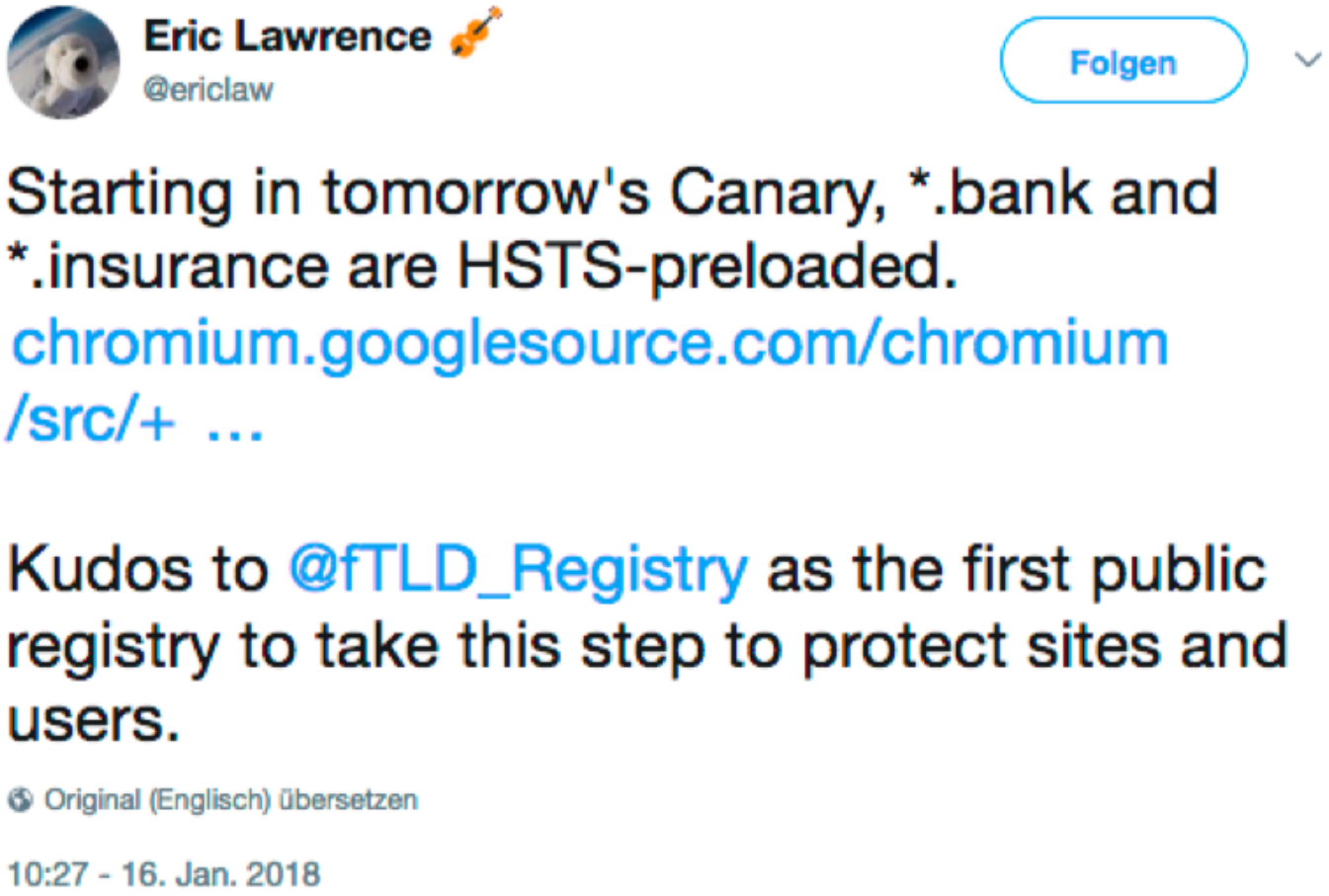
- HSTS still relies on Trust-on-First-Use (TOFU) – the initial insecure redirect, before setting the header, would still be susceptible to attack.

- How to solve this problem? → HSTS preloading:
- Domains can be included in a browser's preload list, i.e. a browser knows before first contact that that domain must only be used through HTTPS.

https://hstspreload.org/
https://hg.mozilla.org/mozilla-central/file/tip/security/manager/ssl/nsSTSPreloadList.inc
https://chromium.googlesource.com/chromium/src/net/+/master/http/transport_security_state_static.json

HSTS -- TLDs

- Entire TLDs can be preloaded:



Agenda

- Evolving the TLS standard
  - Privacy leaks and other drawbacks in TLS 1.2
  - TLS1.3 standardization and its controversies
- Topics in HTTPS Security
  - Certificate Transparency – transmission modes
  - TLS downgrade protection
  - Host Strict Transport Security
  - **Certification Authority Authorization (CAA)**

Certification Authority Authorization (CAA)

- A very recent addition (Sep 2017) to the HTTPS ecosystem, CAA gives the user the possibility to restrict issuance of certificate to a certain Certification Authority (CA)
- Broad concept: Your browsers trusts dozens to hundreds of root CAs
- An attacker must only convince one of these to mis-issue for your domain to get a valid certificate
- But you as the domain owner might only want to ever obtain certificates from 1 certain CA, so why permit all these hundreds of CAs to issue for your domain?

Certification Authority Authorization (CAA)

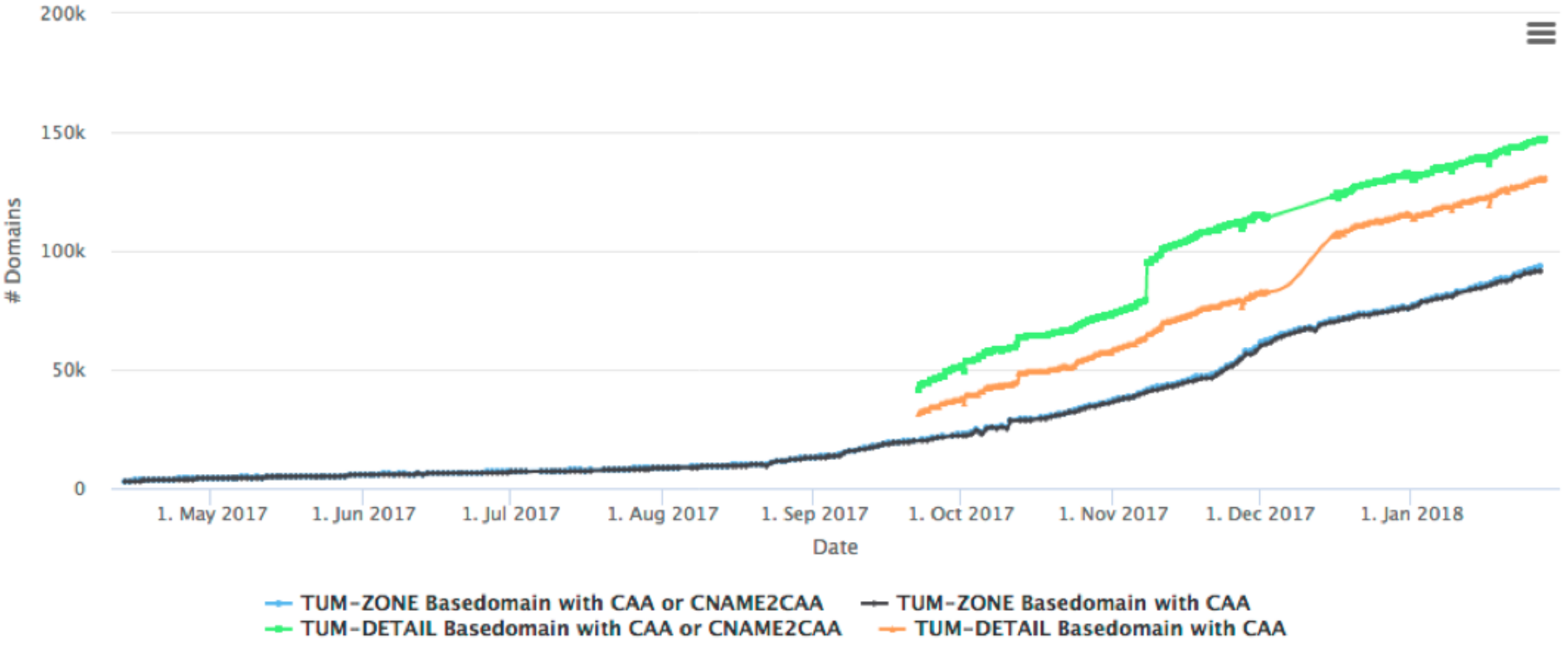
- CAA uses a newly defined DNS record type, in which a domain owner can specify a list of CAs permitted to issue certificates for the domain
- Since September 8, 2017, CAs must validate all domain names that are part of a Certificate Signing Request (CSR) [CSRs are the requests sent by domain owners to CAs, asking for signature and creating a certificate]
- Exemplary CAA records:

Domain	Type	Flags	Tag	Value
tum.de	CAA	0	issue	"letsencrypt.org"
tum.de	CAA	0	issue	"pki.dfn.de"
tum.de	CAA	0	issuewild	";"

- For simplicity, we will ignore the “flags” parameter for this lecture
- Let’s look at the meaning of this record set:
  - Lets’Encrypt and DFN-PKI are both permitted to issue standard certificates for tum.de
  - No CA (“;”) is permitted to issue wildcard certificates for tum.de
    - A wildcard certificate is any certificate with an asterisk (*) in it

Certification Authority Authorization (CAA)

- Measurement study by our chair, including background reading: https://caastudy.github.io/
- Encouraging growth from ~3k domains in April 17 to ~150k domains in January 18



Certification Authority Authorization (CAA)

- Exercise: Assume you are a CA with the CAA value „netsec.top“. You receive a CSR for the following DNS domain names:

  `tum.de, www.tum.de, *.tum.de`

  For which DNS domains do you conduct CAA lookups?

Certification Authority Authorization (CAA)

- Exercise: Assume you are a CA with the CAA value „netsec.top“. You receive a CSR for the following DNS domain names:

  `tum.de, www.tum.de, *.tum.de`

  For which DNS domain names do you conduct CAA lookups?

  `tum.de → tum.de`
  `www.tum.de → www.tum.de`
  `*.tum.de → tum.de`

Certification Authority Authorization (CAA)

- Exercise: Assume you are a CA with the CAA value „`netsec.top`“. You receive a CSR for the following DNS domain names:

  `tum.de, www.tum.de, *.tum.de`

  For which DNS domain names do you conduct CAA lookups?

  `tum.de` → `tum.de`
  `www.tum.de` → `www.tum.de`
  `*.tum.de` → `tum.de`

Certification Authority Authorization (CAA)

- Exercise: Assume you are a CA with the CAA value „`netsec.top`“. You receive a CSR for the following DNS domain names:

  `tum.de, www.tum.de, *.tum.de`

  For which DNS domain names do you conduct CAA lookups?

  `tum.de` → `tum.de`
  `www.tum.de` → `www.tum.de`
  `*.tum.de` → `tum.de`

  For both www.tum.de and tum.de, you obtain the following CAA records set:

  `tum.de issue “;”`

  Are you permitted to issue?

Certification Authority Authorization (CAA)

- Exercise: Assume you are a CA with the CAA value „`netsec.top`“. You receive a CSR for the following DNS domain names:

  `tum.de, www.tum.de, *.tum.de`

  For which DNS domain names do you conduct CAA lookups?

  `tum.de` → `tum.de`
  `www.tum.de` → `www.tum.de`
  `*.tum.de` → `tum.de`

  For both www.tum.de and tum.de, you obtain the following CAA records set:

  `tum.de issue “;”`

  Are you permitted to issue?

  No, „issue ;“ defines that no CA is permitted to issue

Certification Authority Authorization (CAA)

- Exercise: Assume you are a CA with the CAA value „netsec.top“. You receive a CSR for the following DNS domain names:

  `tum.de, www.tum.de, *.tum.de`

  For which DNS domain names do you conduct CAA lookups?

  `tum.de` → `tum.de`
  `www.tum.de` → `www.tum.de`
  `*.tum.de` → `tum.de`

  For both www.tum.de and tum.de, no CAA record set exists.
  Are you permitted to issue?

Certification Authority Authorization (CAA)

- Exercise: Assume you are a CA with the CAA value „netsec.top“. You receive a CSR for the following DNS domain names:

  `tum.de, www.tum.de, *.tum.de`

  For which DNS domain names do you conduct CAA lookups?

  `tum.de → tum.de`
  `www.tum.de → www.tum.de`
  `*.tum.de → tum.de`

  For both www.tum.de and tum.de, no CAA record set exists.
  Are you permitted to issue?

  Yes, absence of CAA records means any CA is permitted to issue.

Certification Authority Authorization (CAA)

- Exercise: Assume you are a CA with the CAA value „netsec.top“. You receive a CSR for the following DNS domain names:

  `tum.de, www.tum.de, *.tum.de`

  For which DNS domain names do you conduct CAA lookups?

  `tum.de → tum.de`
  `www.tum.de → www.tum.de`
  `*.tum.de → tum.de`

  For both www.tum.de and tum.de, you obtain the following CAA record set:

  `tum.de issuewild “;”`

  Are you permitted to issue?

Certification Authority Authorization (CAA)

- Exercise: Assume you are a CA with the CAA value „netsec.top“. You receive a CSR for the following DNS domain names:

  `tum.de, www.tum.de, *.tum.de`

  For which DNS domain names do you conduct CAA lookups?

  `tum.de` → `tum.de`
  `www.tum.de` → `www.tum.de`
  `*.tum.de` → `tum.de`

  For both www.tum.de and tum.de, you obtain the following CAA record set:

  `tum.de issuewild ";"`

  Are you permitted to issue?

  No: In the absence of „issue“, any CA is permitted to issue for tum.de and www.tum.de, **but** for „*.tum.de“, no CA is permitted to issue. As all DNS domain names in the CSR must be permitted, you must refuse to issue the certificate.

Certification Authority Authorization (CAA)

- Exercise: Assume you are a CA with the CAA value „netsec.top“. You receive a CSR for the following DNS domain names:

  `tum.de, www.tum.de, *.tum.de`

  For which DNS domain names do you conduct CAA lookups?

  `tum.de → tum.de`
  `www.tum.de → www.tum.de`
  `*.tum.de → tum.de`

  For both www.tum.de and tum.de, you obtain the following CAA records set:

  `tum.de issue “;”`
  `tum.de issuewild “netsec.top”`

  Are you permitted to issue?

Certification Authority Authorization (CAA)

- Exercise: Assume you are a CA with the CAA value „netsec.top“. You receive a CSR for the following DNS domain names:

  `tum.de, www.tum.de, *.tum.de`

  For which DNS domain names do you conduct CAA lookups?

  `tum.de → tum.de`
  `www.tum.de → www.tum.de`
  `*.tum.de → tum.de`

  For both www.tum.de and tum.de, you obtain the following CAA records set:

  ```
  tum.de issue “;”
  tum.de issuewild “netsec.top”
  ```

  Are you permitted to issue?

  No, „issue ;“ defines that no CA is permitted to issue. Even though our CA is permitted to issue wildcard certificates, we are not permitted to issue for tum.de and www.tum.de and must hence refuse to issue the certificate.

Certification Authority Authorization (CAA)

- Exercise: Assume you are a CA with the CAA value „`netsec.top`“. You receive a CSR for the following DNS domain names:

  `tum.de, www.tum.de, *.tum.de`

  For which DNS domain names do you conduct CAA lookups?

  `tum.de` → `tum.de`
  `www.tum.de` → `www.tum.de`
  `*.tum.de` → `tum.de`

  For both www.tum.de and tum.de, you obtain the following CAA records set:

  `tum.de issue “netsec.top”`

  Are you permitted to issue?

Certification Authority Authorization (CAA)

- Exercise: Assume you are a CA with the CAA value „`netsec.top`“. You receive a CSR for the following DNS domain names:

  `tum.de, www.tum.de, *.tum.de`

  For which DNS domain names do you conduct CAA lookups?

  `tum.de` → `tum.de`
  `www.tum.de` → `www.tum.de`
  `*.tum.de` → `tum.de`

  For both www.tum.de and tum.de, you obtain the following CAA records set:

  `tum.de issue “netsec.top”`

  Are you permitted to issue?

  Yes, exclusively you are permitted to issue certificates for tum.de.
  You may also issue wildcard certificates (lack of issuewild means any CA can issue)

Certification Authority Authorization (CAA)

- A simple goal, but mighty complexity!
- See our tracker and an extensive study on what went wrong in practice under

  `https://caastudy.github.io`

Agenda

- Evolving the TLS standard
  - Privacy leaks and other drawbacks in TLS 1.2
  - TLS1.3 standardization and its controversies
- Topics in HTTPS Security
  - Certificate Transparency – transmission modes
  - TLS downgrade protection
  - Host Strict Transport Security
  - Certification Authority Authorization (CAA)

The End – Questions?