Trust and Transparency

NSF Workshop: Towards Re-architecting Today's Internet for Survivability

Zakir Durumeric, Stanford

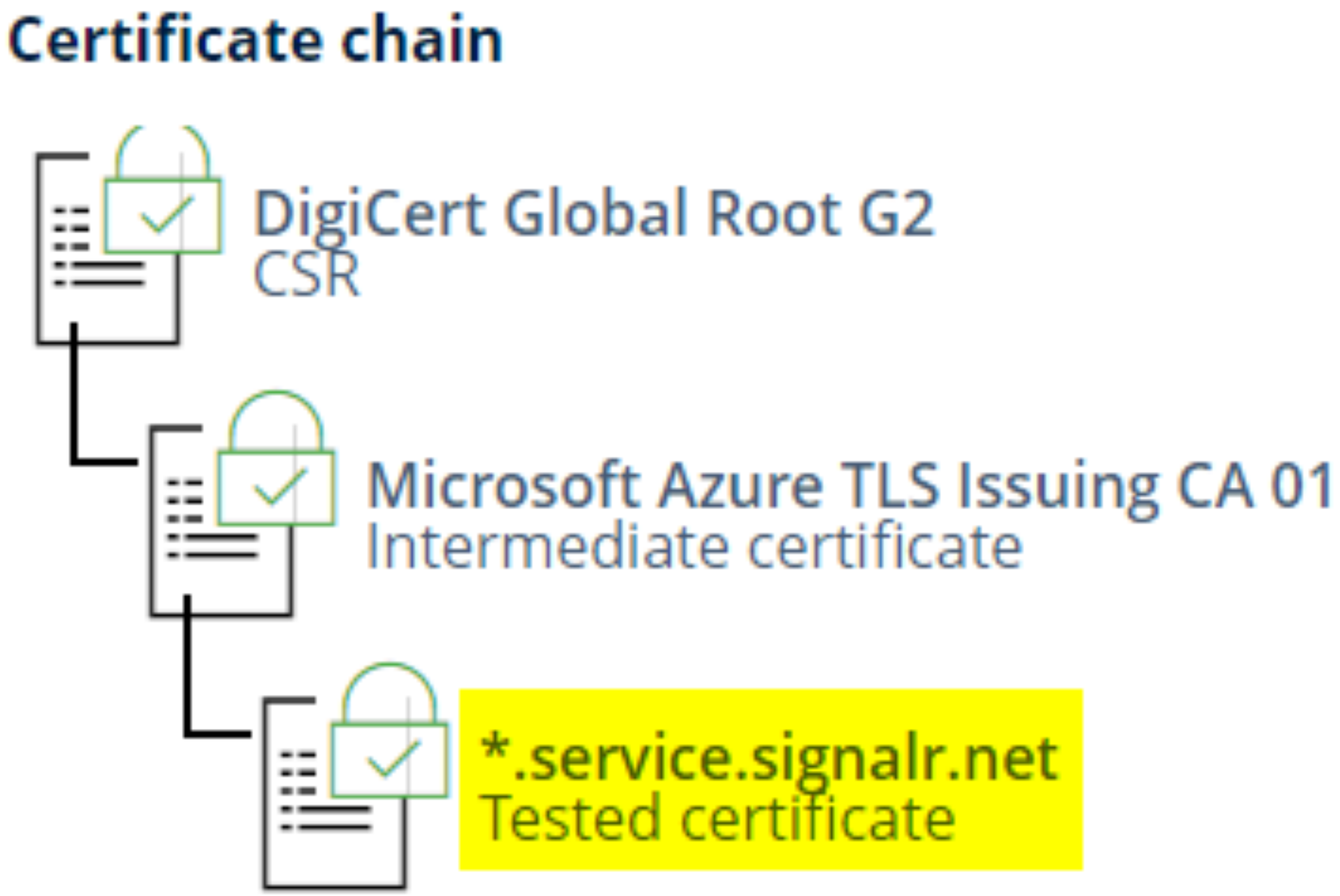
WebPKI — Foundation for Trust on the Web

Authentication on the web is based on validating X.509 certificates signed by Certificate Authorities (CAs)

CAs are organizations trusted to validate the ownership or control of a domain

Browsers delegate authority to CAs, but CAs can delegate authority to other CAs (up to ≈9 layers deep)

There are 1,364 CAs today; any CA can sign a certificate for any domain



Certificate chain

DigiCert Global Root G2
CSR

Microsoft Azure TLS Issuing CA 01
Intermediate certificate

*.service.signalr.net
Tested certificate

Who do we trust?

Prior to ~2012, we had zero visibility into the CAs (organizations) we trusted to validate the identity of websites

Through Internet Scanning, we discovered *most* CA certificates

Found that everyone from Coca Cola and Disney to authoritarian regimes to small towns had the ability to sign certificates for any website

CAs had been selling CA certificates to anyone who would pay for one

Except CA Certificates Are Opaque

Symantec / DigiCert operated root c38dcb389593…

```
commonName             = UTN-USERFirst-NetworkApplications
orgUnitName            = http://www.usertrust.com
orgName                = The USERTRUST Network
localityName           = Salt Lake City
stateOrProvinceName = UT
countryName            = US
```

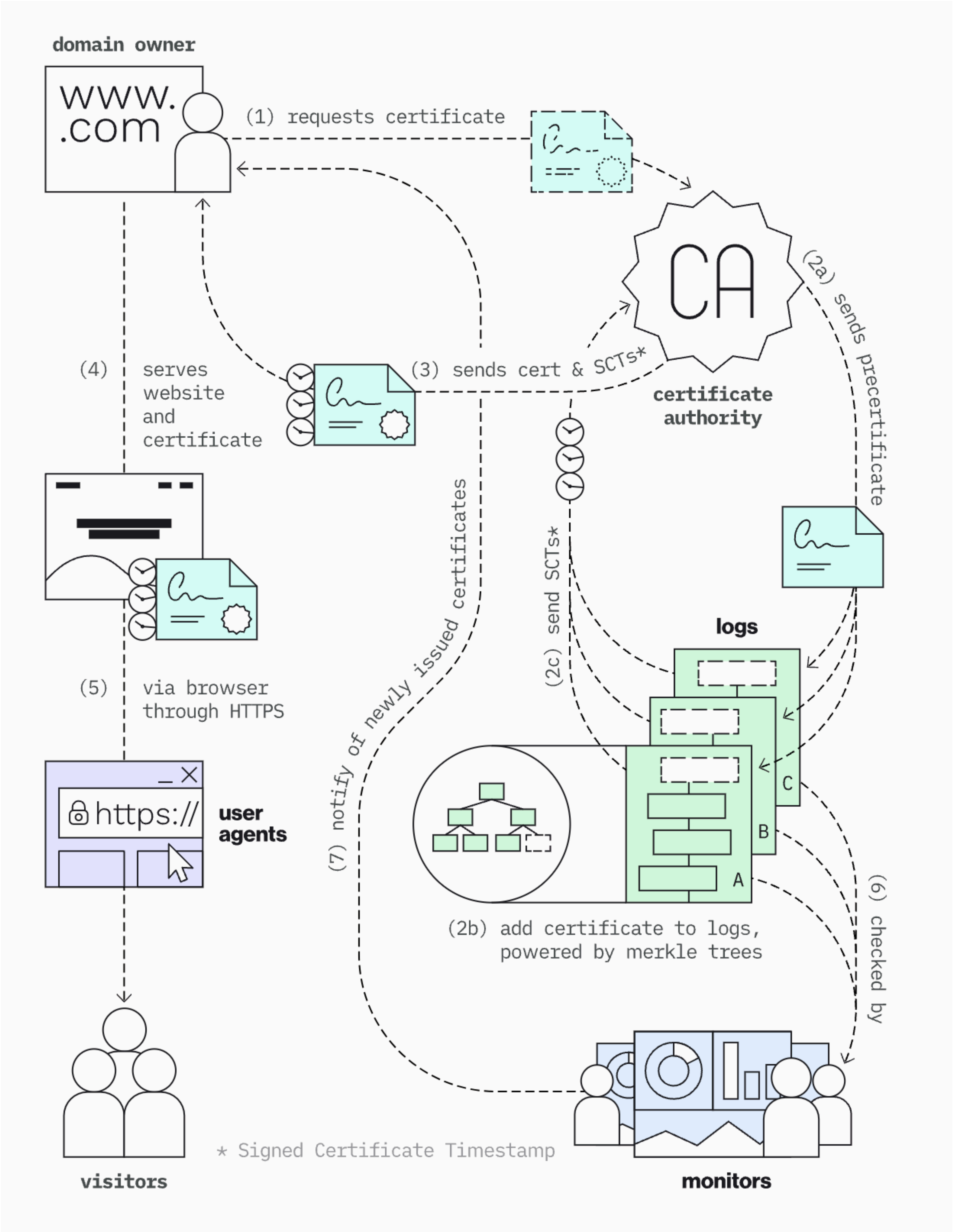
Comodo / Sectigo operated root 43f257412d44…

```
commonName             = UTN-USERFirst-Client Authentication and Email
orgUnitName            = http://www.usertrust.com
orgName                = The USERTRUST Network
localityName           = Salt Lake City
stateOrProvinceName = UT
countryName            = US
```

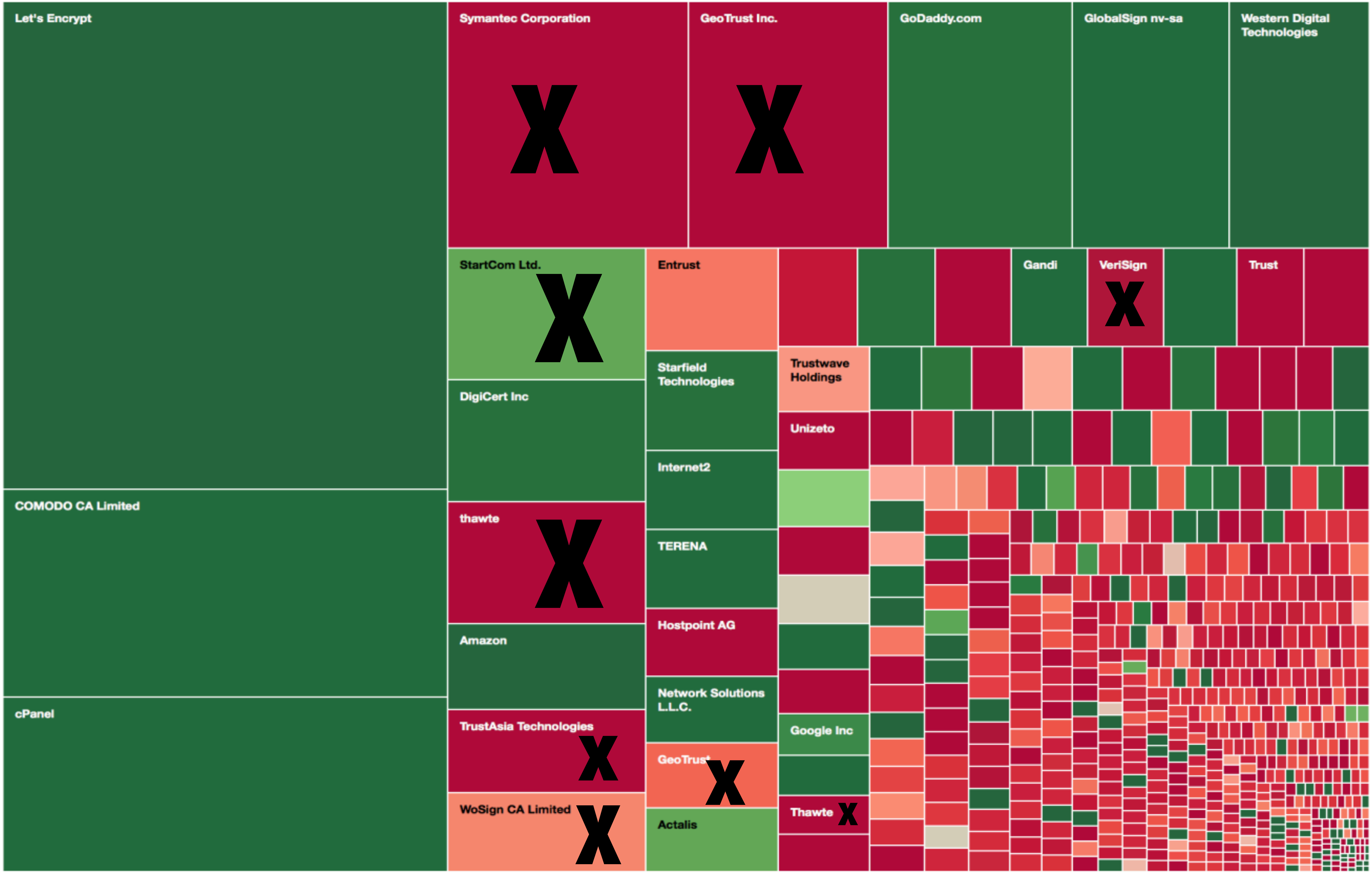
Certificate Transparency

Problem: We only discovered "*mis-issued*" certificates through luck — Google Crawls, ZMap Scans, or User Reports

Solution: In 2017 Google Chrome requires all certificates to be logged in public Certificate Transparency (CT) Logs (Merkle Tree + Standardized API)


domain owner
www.
.com
(1) requests certificate
CA
certificate authority
(2a) sends precertificate
(3) sends cert & SCTs*
(4) serves website and certificate
(2c) send SCTs*
(7) notify of newly issued certificates
logs
C
B
A
(2b) add certificate to logs, powered by merkle trees
(6) checked by
(5) via browser through HTTPS
https://
user agents
visitors
monitors

* Signed Certificate Timestamp


Let's Encrypt
Symantec Corporation
GeoTrust Inc.
GoDaddy.com
GlobalSign nv-sa
Western Digital Technologies
StartCom Ltd.
Entrust
Gandi
VeriSign
Trust
Starfield Technologies
Trustwave Holdings
DigiCert Inc
Unizeto
Internet2
COMODO CA Limited
thawte
TERENA
Amazon
Hostpoint AG
Network Solutions L.L.C.
Google Inc
TrustAsia Technologies
GeoTrust
cPanel
WoSign CA Limited
Thawte
Actalis

Good:

- Transparency is an *incredible* security primitive → distributed trust with the ability to monitor and verify
    - Where else can we use this in the Internet? DNS? Routing?
- Google and Mozilla are actively managing their root programs and have *dramatically* improved the ecosystem
    - New Google Chrome Root Store is a sane “Fresh Start”

Bad:

- Our transparency systems are untenably brittle — nobody can successfully run a CT log (including Google and Cloudflare)
    - CT = CAP Theorem + Cryptographic Verification = Disaster
- Control is based on a small number of parties
- Massive amount of data produced — not necessarily analyzed
- Who pays to run free CAs, CT Logs, and queryable databases?
- eIDAS regulation potentially massively complicates trust on the web by limiting what security controls that browsers can require