



Privacy-Preserving & Incrementally-Deployable Support for Certificate Transparency in Tor

July 15, 2021

Rasmus Dahlberg, Tobias Pulls, Tom Ritter, and Paul Syverson

A flash-back into the past

- June, 2011
- Netherlands, Beverwijk
- DigiNotar



<https://creativecommons.org/licenses/by-sa/3.0/>

What happened?

- DigiNotar issued web certificates
- Did not live up to expectations
- Then tried to cover it up¹



<https://www.bbc.com/news/technology-14989334>

¹ FoxIT. Black Tulip: Report of the investigation into the DigiNotar Certificate Authority breach. Page 3.

What happened?

- DigiNotar issued web certificates
- Did not live up to expectations
- Then tried to cover it up¹



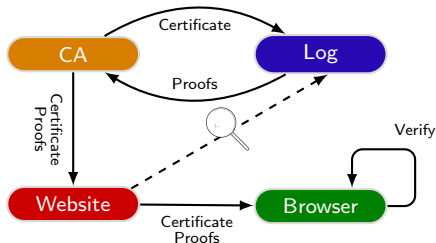
<https://www.bbc.com/news/technology-14989334>

A stealthy attacker might have gotten away with it!

¹ FoxIT. Black Tulip: Report of the investigation into the DigiNotar Certificate Authority breach. Page 3.

Larger problem and solution?

- Digitar was not a one-time incident²
- Many other parties can get it wrong
- Add visibility into issued certificates³



²<https://sslmate.com/certspotter/failures>

³<https://certificate.transparency.dev/>

Certificate Transparency (CT) compliance⁴



“Two logs promised that they will make the certificate public”

⁴ https://github.com/chromium/ct-policy/blob/master/ct_policy.md & <https://support.apple.com/en-us/HT205280>

Problem statement

- Tor Browser does not enforce CT
- Guard against prominent threats
 - ▶ DigiNotar style attacks
 - ▶ Interception to deanonymize
- Aim higher than CT compliance



Problem statement

- Tor Browser does not enforce CT
- Guard against prominent threats
 - ▶ DigiNotar style attacks
 - ▶ Interception to deanonymize
- Aim higher than CT compliance

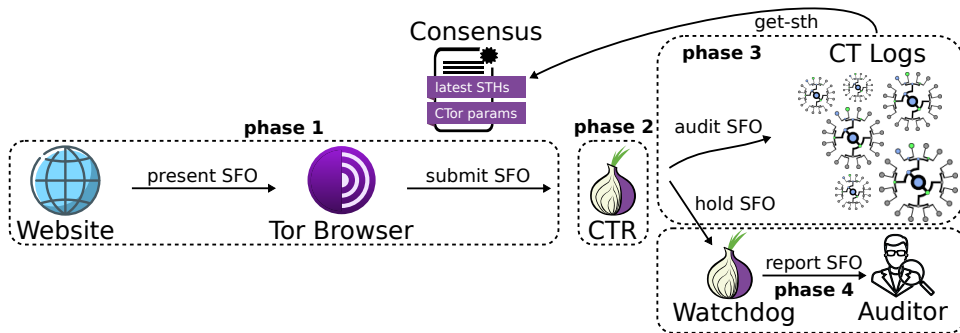


Attacker with browser exploit, CA, CT logs, and usual Tor capabilities

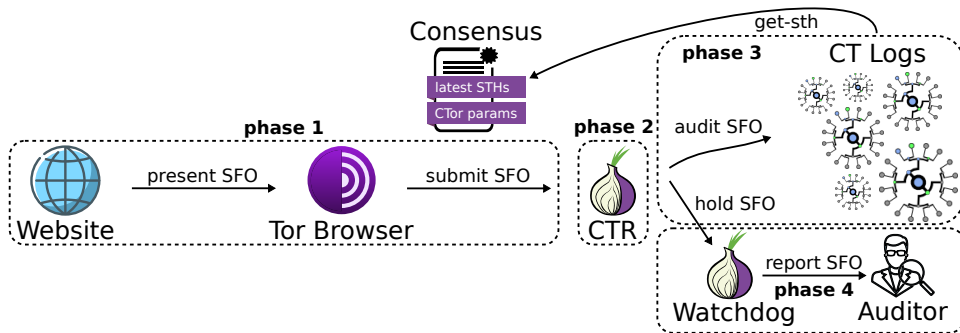
Gradual roll-out plan

- | | |
|---|--|
| 1. Catch up with CT compliant browsers | <i>pairs of logs</i> are trusted blindly |
| 2. Steps towards decentralized verification | <i>some log</i> is trusted blindly |
| 3. Fully decentralized verification | <i>no log</i> is trusted blindly |

Overview of the full design

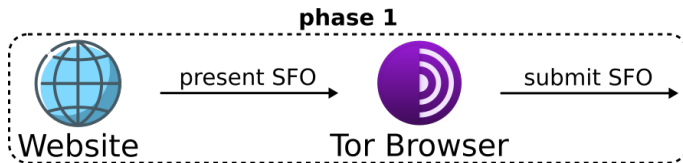


Overview of the full design



Security? Difficult to interfere without detection in any phase

Submission phase



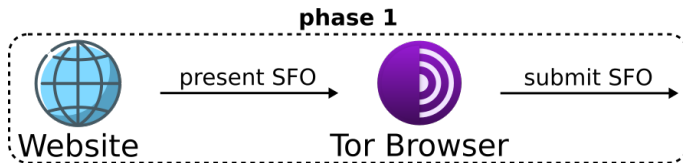
Straw man proposals

- Fetch an inclusion proof
- Rely on a centralized party

What we do instead

- Use Tor relays, “CTRs”
- Probabilistic submit

Submission phase



Straw man proposals

- Fetch an inclusion proof
- Rely on a centralized party

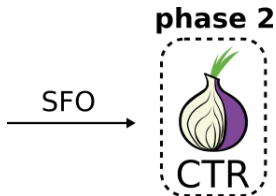
What we do instead

- Use Tor relays, “CTRs”
- Probabilistic submit

It must be difficult to infer which CTR received an SFO

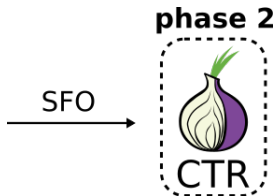
Buffering phase

- Buffer until logging is required
- Add a random delay to leak less
- Cache audited SFOs to leak less



Buffering phase

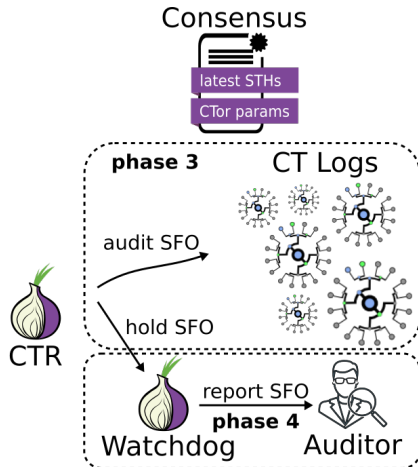
- Buffer until logging is required
- Add a random delay to leak less
- Cache audited SFOs to leak less



The attacker's best bet to interfere is trivially detectable

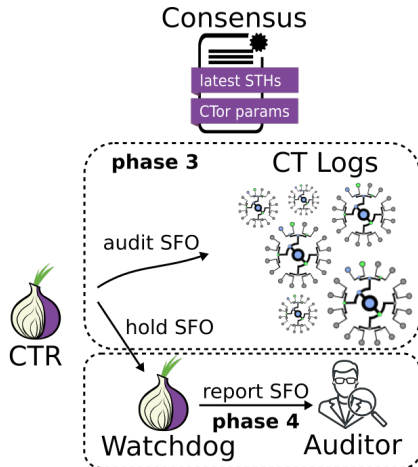
Audit and report phases

- Fetch inclusion proof against a specific STH
- Rely on Tor's consensus to agree on STHs
- Watchdog CTRs do the reporting if needed
 - ▶ Protects against CTR identification



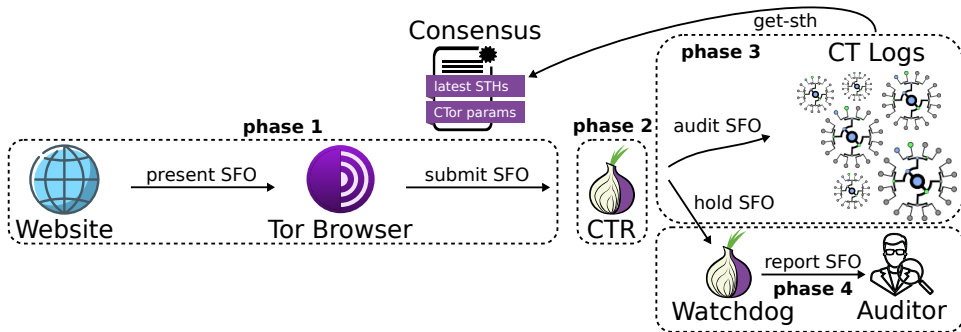
Audit and report phases

- Fetch inclusion proof against a specific STH
- Rely on Tor's consensus to agree on STHs
- Watchdog CTRs do the reporting if needed
 - ▶ Protects against CTR identification

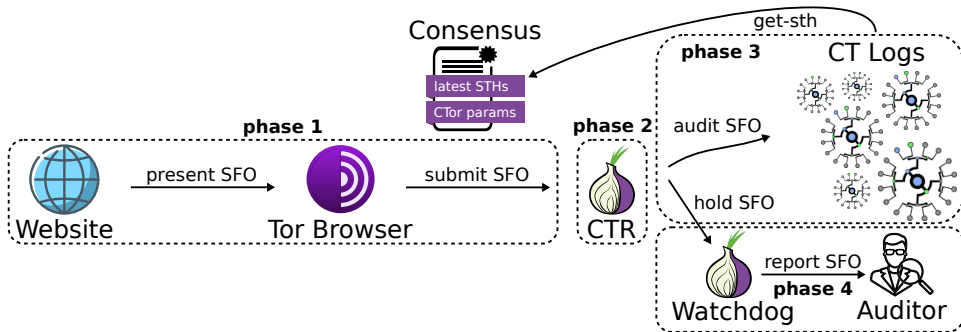


Why not just send to a trusted auditor immediately?

Putting it all together

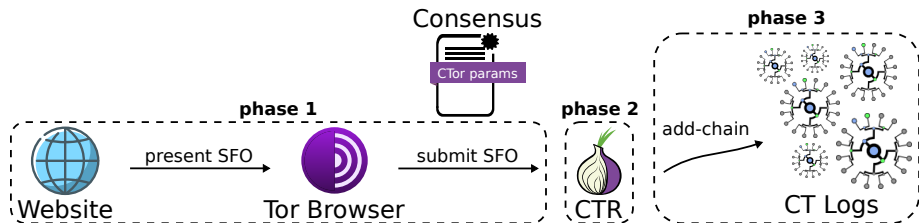


Putting it all together

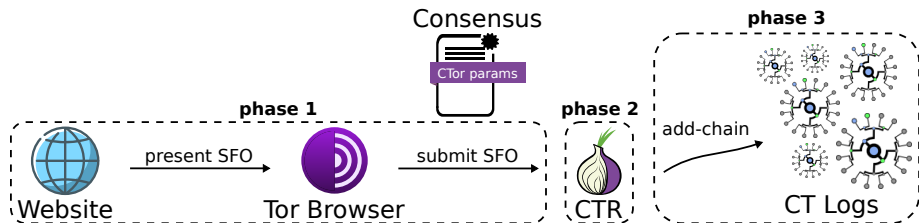


This is quite a leap from CT compliance

Incremental design



Incremental design



Use the log ecosystem against the attacker

Take away

- Tor Browser would benefit from CT
- CT would benefit from more auditing
- Delegated auditing is key in our setting

