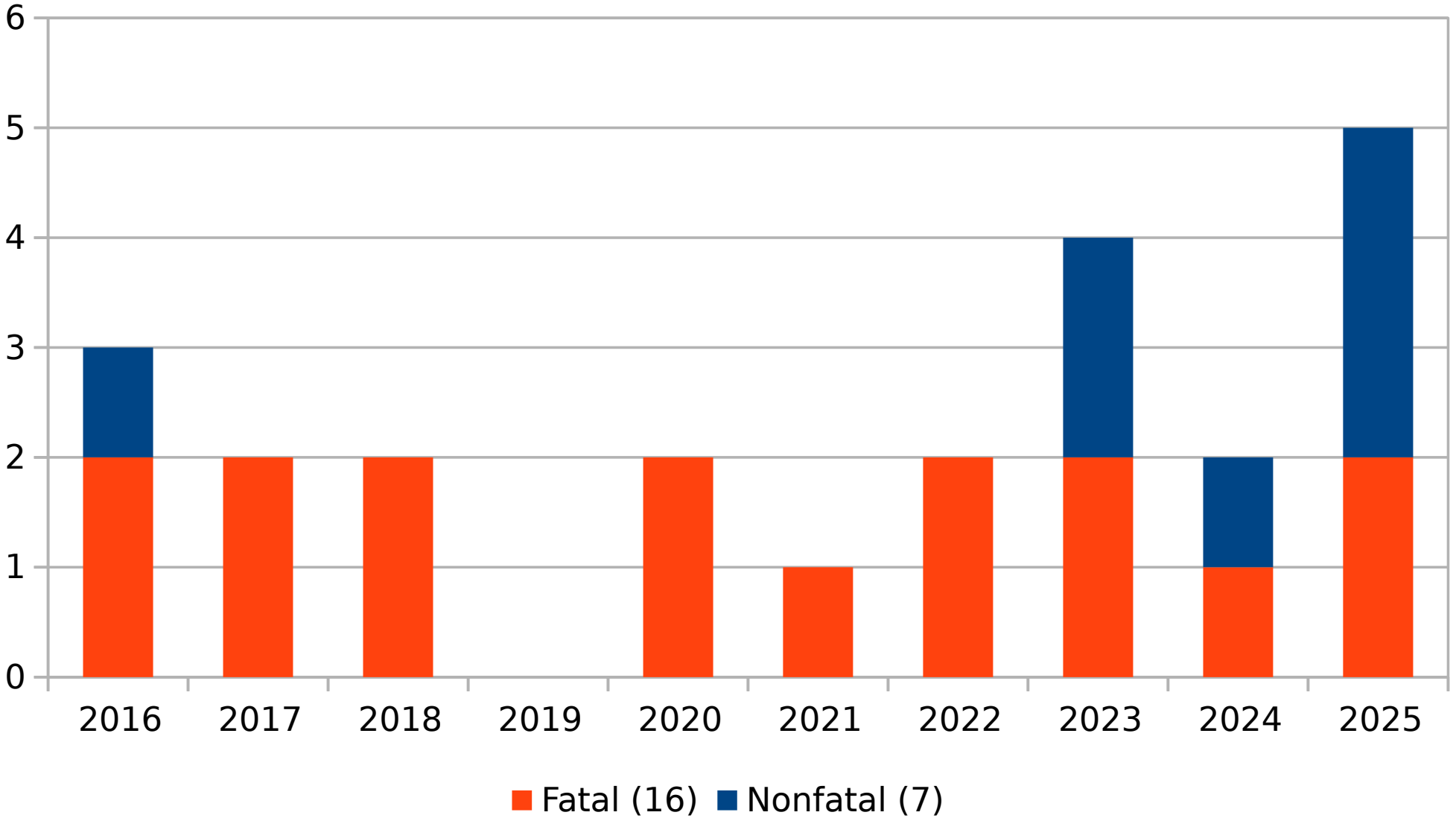
Learning from Certificate Transparency Log Failures

Andrew Ayer

www.agwa.name

SSLMate / Cert Spotter

Log Failures By Year



Log Failure Modes

6 Unincorporated SCT (fatal)

5 MMD violation (nonfatal)

5 Multi-day outage (3 fatal, 2 nonfatal)

4 Inconsistent STH (fatal)

1 Data loss (fatal)

1 Inconsistent entry (fatal)

1 Private key compromise (fatal)

Root causes

6 Asynchronous merging

6 Wrong storage model

4 Unknown (likely asynchronous merging)

2 Configuration fragility

2 Cosmic ray

1 Manual administration

1 Insufficient operator resources

1 Security compromise

Root causes

6 Asynchronous merging

6 Wrong storage model

4 Unknown (likely asynchronous merging)

2 Configuration fragility

2 Cosmic ray

1 Manual administration

1 Insufficient operator resources

1 Security compromise

Asynchronous Merging is Risky

- Logs accept certificates faster than they can merge them
  - MMD violations: 2016, 2024, 2025, 2025, 2025

Asynchronous Merging is Risky

- Logs accept certificates faster than they can merge them
  - MMD violations: 2016, 2024, 2025, 2025, 2025
- Logs forget about certificates
  - Unincorporated SCTs: 2017, 2018, 2018, 2023, 2025

Asynchronous Merging is Risky

- Logs accept certificates faster than they can merge them
  - MMD violations: 2016, 2024, 2025, 2025, 2025
- Logs forget about certificates
  - Unincorporated SCTs: 2017, 2018, 2018, 2023, 2025
- Recommendation: logs should not return an SCT until a new tree head is signed and stored
  - Natural back pressure
  - Fails safely
  - Simpler

CT Doesn’t Need Databases

- UPDATE and DELETE not needed

CT Doesn’t Need Databases

- UPDATE and DELETE not needed
- Indexes not needed for basic read operations

RFC 6962	static-ct-api
Duplicate prevention	Duplicate prevention
get-proof-by-hash	

CT Doesn’t Need Databases

- UPDATE and DELETE not needed
- Indexes not needed for basic read operations

RFC 6962	static-ct-api
Duplicate prevention	Duplicate prevention
get-proof-by-hash	

- Data can be kept in immutable, numbered tiles
  - Filesystem
  - Object storage (e.g. S3)

Problems with Databases

- More failure modes, harder recovery
  - Botched recovery: 2017 inconsistent STH
  - Slow recovery: 2022, 2023 multi-day outage
  - Very slow recovery: 2023 multi-week outage (3 logs)
  - Corruption: 2024 unincorporated SCT

Problems with Databases

- More failure modes, harder recovery
  - Botched recovery: 2017 inconsistent STH
  - Slow recovery: 2022, 2023 multi-day outage
  - Very slow recovery: 2023 multi-week outage (3 logs)
  - Corruption: 2024 unincorporated SCT
- Hard to scale reads
  - 2025 multi-day outage (2 logs)
  - 3 operators now use tile-based caches in front of DB

Problems with Databases

- More failure modes, harder recovery
  - Botched recovery: 2017 inconsistent STH
  - Slow recovery: 2022, 2023 multi-day outage
  - Very slow recovery: 2023 multi-week outage (3 logs)
  - Corruption: 2024 unincorporated SCT
- Hard to scale reads
  - 2025 multi-day outage (2 logs)
  - 3 operators now use tile-based caches in front of DB
- Low limits
  - RDS: 16TB
  - PostgreSQL: 32TB
  - ext4: 1EB
  - zfs: 2^128 bytes
  - S3: unlimited

Configuration Fragility

- 2016 Inconsistent STH
  - Operator reused key for test log

Configuration Fragility

- 2016 Inconsistent STH
  - Operator reused key for test log
- 2020 Inconsistent STH (2 logs)
  - Operator pointed test log at production etcd

Configuration Fragility

- 2016 Inconsistent STH
  - Operator reused key for test log
- 2020 Inconsistent STH (2 logs)
  - Operator pointed test log at production etcd
- Mitigation: Sunlight's Global Lock Backend
  - Detects reused keys
  - Guards against multiple writers

Recommendations

- Don’t return SCTs until tree head is stored
  - Don’t disable TesseraCT’s publication awaiter
- Store data in tiles on filesystem or in object storage
- Log software should be resilient to config mistakes
  - Sunlight’s Global Lock Backend should be table stakes

Would have prevented 12-18 out of 23 failures