Certificate Transparency & Derivatives

Mattijs Jonker

Introduction

- A few years ago (following WIE-KISMET 2019)**:**

  “Why don’t we collect CT, given it’s a rich source of domain names?”

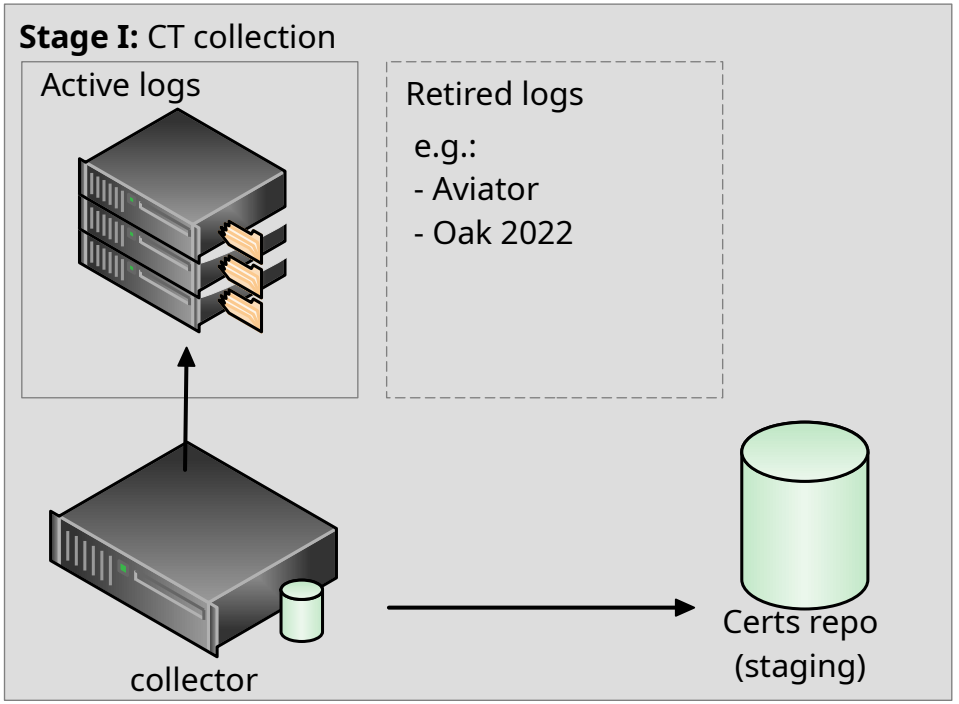
- This prompted scraping efforts:
  - I found a blog about scraping CT that also offers a tool
    - *Axeman* by “CaliDog” (how befitting)
  - Modified Axeman with a bit of help from others (Olivier, Raffaele) to output avro data
  - Built entire pipeline for continual scraping, data warehousing

CT 101

- Public, append-only log for auditing and monitoring purposes
- Use CT Internet standard
- Browser vendors (e.g., Google) drove adoption
  - Certs need to be included in logs to be accepted
  - Policies have changed over years (e.g., Chrome's 1 Google-operated log SCT requirement)
- Logs are nowadays typically *temporally sharded* (e.g., Nimbus 2024, Oak 2024H1)
  - And retired once considered no longer needed

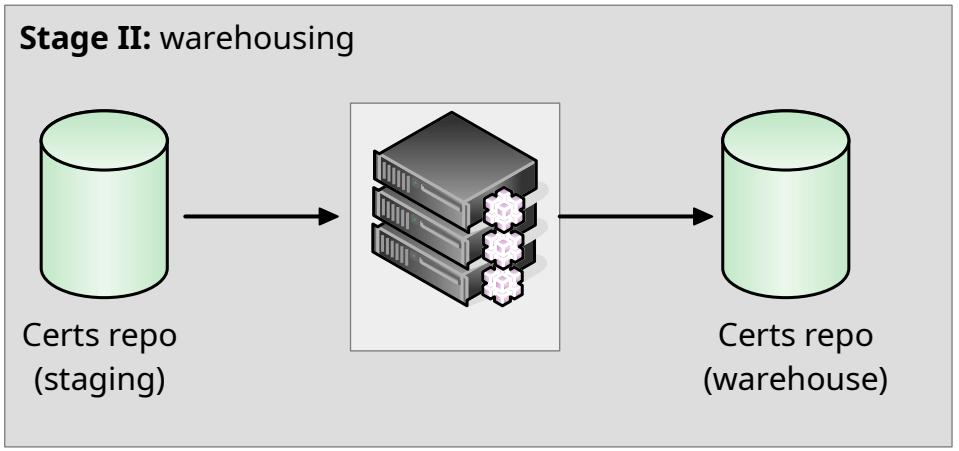
Collection

- Multi-threaded Python scripts that keep scraping state
  - Continually invoke Axeman at given (log, offset)
  - Pre-extraction of x509 values and full DER-encoded leaf
  - Contain full chain of certs to trust anchor also (DER)
- Merge small sets of leaf certs in staging files (row-based, Avro)
  - Stored in S3-interoperable MinIO tenant (on k8s)



Warehousing

- Convert row-based staging data (Avro) to columnar for warehousing (ORC)
- Repo remains the same: MinIO tenant on k8s
- Conversion performed on same k8s cluster (Spark)
- Significant reduction in size (DER deduplication)



Structure

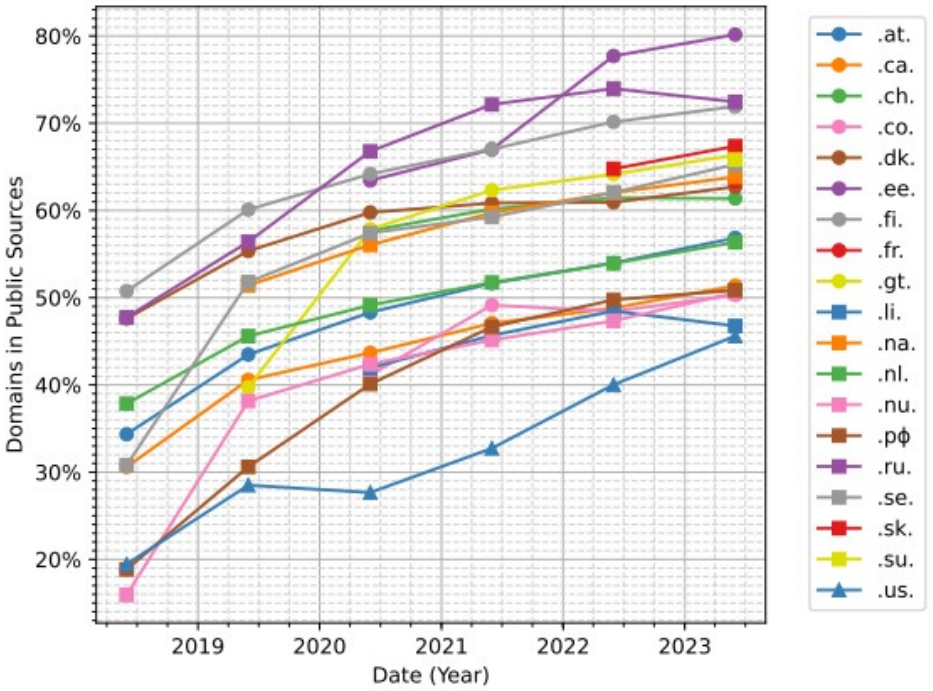
```
root
 |-- cert_index: integer (nullable = true)
 |-- cert_link: string (nullable = true)
 |-- chain: array (nullable = true)
 |    |-- element: struct (containsNull = true)
 |    |    |-- as_der: string (nullable = true)
 |    |    |-- extensions: struct (nullable = true)
 |    |    |    |-- authorityInfoAccess: string (nullable = true)
 |    |    |    |-- authorityKeyIdentifier: string (nullable = true)
 |    |    |    |-- basicConstraints: string (nullable = true)
 |    |    |    |-- certificatePolicies: string (nullable = true)
 |    |    |    |-- crlDistributionPoints: string (nullable = true)
 |    |    |    |-- keyUsage: string (nullable = true)
 |    |    |    |-- subjectKeyIdentifier: string (nullable = true)
 |    |    |-- fingerprint: string (nullable = true)
 |    |    |-- not_after: float (nullable = true)
 |    |    |-- not_before: float (nullable = true)
 |    |    |-- serial_number: string (nullable = true)
 |    |    |-- subject: struct (nullable = true)
 |    |    |    |-- C: string (nullable = true)
 |    |    |    |-- CN: string (nullable = true)
 |    |    |    |-- L: string (nullable = true)
 |    |    |    |-- O: string (nullable = true)
 |    |    |    |-- OU: string (nullable = true)
 |    |    |    |-- ST: string (nullable = true)
 |    |    |    |-- aggregated: string (nullable = true)
 |-- leaf_cert: struct (nullable = true)
 |    |-- as_der: string (nullable = true)
 |    |-- extensions: struct (nullable = true)
 |    |    |-- authorityInfoAccess: string (nullable = true)
 |    |    |-- authorityKeyIdentifier: string (nullable = true)
 |    |    |-- basicConstraints: string (nullable = true)
 |    |    |-- certificatePolicies: string (nullable = true)
 |    |    |-- crlDistributionPoints: string (nullable = true)
 |    |    |-- keyUsage: string (nullable = true)
 |    |    |-- subjectKeyIdentifier: string (nullable = true)
 |    |-- fingerprint: string (nullable = true)
 |    |-- not_after: float (nullable = true)
 |    |-- not_before: float (nullable = true)
 |    |-- serial_number: string (nullable = true)
 |    |-- subject: struct (nullable = true)
 |    |    |-- C: string (nullable = true)
 |    |    |-- CN: string (nullable = true)
 |    |    |-- L: string (nullable = true)
 |    |    |-- O: string (nullable = true)
 |    |    |-- OU: string (nullable = true)
 |    |    |-- ST: string (nullable = true)
 |    |    |-- aggregated: string (nullable = true)
 |    |-- issuer: struct (nullable = true)
 |    |    |-- C: string (nullable = true)
 |    |    |-- CN: string (nullable = true)
 |    |    |-- L: string (nullable = true)
 |    |    |-- O: string (nullable = true)
 |    |    |-- OU: string (nullable = true)
 |    |    |-- ST: string (nullable = true)
 |-- seen: float (nullable = true)
 |-- source: struct (nullable = true)
 |    |-- name: string (nullable = true)
 |    |-- url: string (nullable = true)
 |-- update_type: string (nullable = true)
 |-- name: string (nullable = true)
 |-- all_domains: array (nullable = true)
 |    |-- element: string (containsNull = true)
 |-- batch-group: integer (nullable = true)
```

Some numbers

- Currently have data from 70 logs
  - 35 billion (G) certificates (non-unique)
  - Roughly 1/5th of that in unique certs
- Most and all larger logs listed at *https://sslmate.com/resources/certspotter_stats*

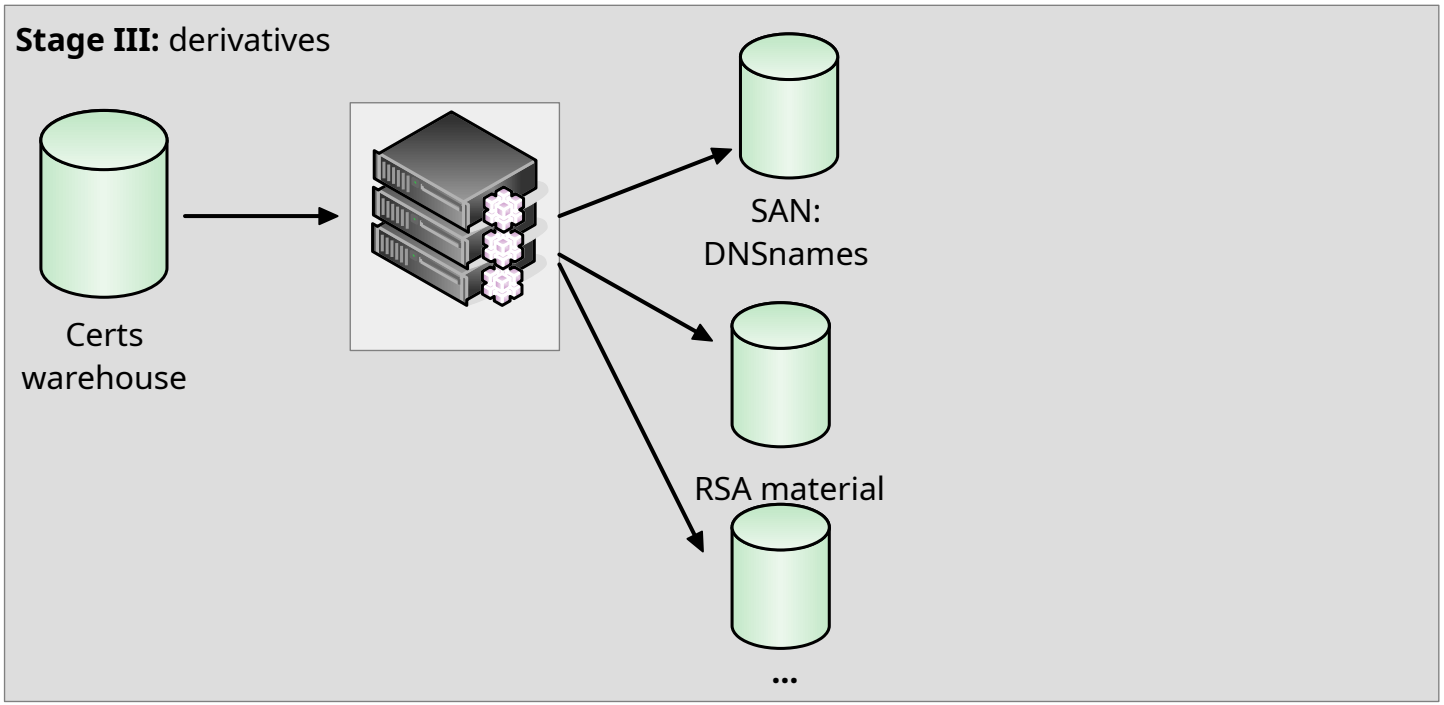
Driving stakes into the ground

- Studied degree of ccTLD coverage in public sources (CT+CC) [accepted ACM CCR]
  - Used the 19 ccTLDs that we have zone files for as ground truth
- High-level findings:
  - 43% – 80% names (registered) covered
  - Substantial coverage of names intended for Web
  - Increases over time (5y period)
  - More than half of domains appear on the same day w.r.t. delegation



Derivatives

- Extract derivatives:
  - Subject Alternative Names (RFC 5280) – notably, *DNSnames*
  - RSA material (to look for shared primes)
  - …



Plans for data sharing / value add

- Been measuring for a few months:
  - registered country-code domain names amassed from CT
  - Daily upcert and pruning (NXDOMAIN window)
- Will extend fDNS measurement to FQDNs in future:
  - OpenINTEL fDNS measurement is primarily zonefile seeded
  - Perhaps sensitive labels first (citrix, vpn, secure, ldap, ad, …)
- Expose via S3 (so Alfred can integrate it ;-])
  - Note: our current “opencc” data consists only of public zone seeded fDNS measurement (.ch, .li, .se, .nu, .ee, .sk, .fr)

Questions?