Finding a Needle in a Haystack - Deciphering Certificate Transparency Logs and Proposing a Better Infrastructure for Querying Certificate Transparency Logs: CertSight

Mohammed Adain*
madain3@gatech.edu
Georgia Institute of Technology
Atlanta, Georgia, USA

Swebert Correa*
scorrea34@gatech.edu
Georgia Institute of Technology
Atlanta, Georgia, USA

Abstract

Certificate Transparency (CT) is a framework designed to provide public visibility into the issuance of digital certificates, aiming to detect misissued or fraudulent certificates and enhance the security of the internet's Public Key Infrastructure (PKI). The CT logs, which store signed records of all issued certificates, serve as a critical tool for detecting potential certificate-related threats. However, querying and analyzing these logs at scale remains challenging due to their growing size, limited query efficiency, and lack of standardization across different log providers.

This report explores the current state of querying CT logs, identifying key limitations such as slow query times, inconsistent data formats, and difficulties in retrieving relevant information across multiple log servers. We propose a new infrastructure for querying Certificate Transparency logs, focusing on improving query speed, scalability, and the ease of integration with existing systems. The proposed infrastructure leverages distributed databases, optimized indexing strategies, and advanced filtering techniques to enable efficient and real-time querying of CT logs. Additionally, we discuss the implementation of a unified query interface that simplifies cross-log querying while maintaining the integrity and security of the CT data.

Through a detailed evaluation and comparison with existing systems, we demonstrate the potential of our proposed infrastructure to significantly enhance the performance of CT log querying, providing a more reliable, faster, and user-friendly approach to monitoring certificate issuance. This solution is expected to contribute to more effective detection of certificate misissuances and bolster the overall security of internet communications.

ACM Reference Format:
Mohammed Adain and Swebert Correa. 2024. Finding a Needle in a Haystack - Deciphering Certificate Transparency Logs and Proposing a Better Infrastructure for Querying Certificate Transparency Logs: CertSight. In . ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Certificate Transparency (CT) logs are publicly accessible, append-only databases designed to record the issuance of SSL/TLS certificates by Certificate Authorities (CAs). These logs provide a transparent, tamper-proof record of all publicly trusted certificates, playing a crucial role in enhancing the security and oversight of the digital certificate ecosystem. By ensuring that all issued certificates are logged in a publicly verifiable manner, Certificate Transparency aims to detect misissued or fraudulent certificates, thereby preventing potential security threats, such as man-in-the-middle attacks or the exploitation of compromised certificates.

The information contained within CT logs offers significant insights into the certificate issuance process, making it possible to track certificates issued to domains and identify potential risks. For example, one can easily verify whether a malicious actor was granted a certificate for a legitimate domain by querying CT logs. This visibility enables domain owners, security experts, and organizations to monitor the health and security of their digital assets in real time.

In this project, we focus on exploring the potential of CT logs to derive actionable insights regarding SSL/TLS certificate issuance. Specifically, we aim to develop a tool that allows domain owners to efficiently query CT logs for unusual certificate activities, such as unauthorized certificate issuance or changes in the certificate landscape. The goal is to create an automated notification system that alerts domain owners when suspicious events are detected, thereby enhancing the security and integrity of their domain certificates. Through this tool, we aim to empower domain owners with the capability to quickly respond to potential threats, improving overall cybersecurity and trust in the digital certificate ecosystem.

1.1 Research goals

The goal of this research is to extract valuable insights from the Certificate Transparency (CT) logs and identify potential attack vectors associated with TLS certificates. By analyzing certificate issuance and renewal patterns, we aim to detect anomalies such as unauthorized certificates or unusual patterns of certificate activity that could indicate malicious behavior. These insights will help in identifying vulnerabilities in the certificate ecosystem, providing an opportunity to proactively address security risks.

1.2 Contributions

This work makes the following key contributions:

1. Novel System for Efficient Querying: We propose CertSight, a scalable system for faster querying of Certificate Transparency (CT)

*Both authors contributed equally to this research.

Conference'17, July 2017, Washington, DC, USA
© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

logs, using a TLD-based database schema to improve performance and simplify domain-specific analyses.

2. CT Log Analysis: CertSight is leveraged to query and organize CT logs, efficiently processing large datasets and enabling targeted analysis.

3. Identifying Attack Vectors: The system extracts meaningful insights from CT logs, identifying potential security risks such as unusual certificate issuance or renewal patterns, highlighting possible attack vectors.

4. Notification System: We propose a notification system to alert domain owners of suspicious certificate renewal activities, enabling timely responses to potential threats.

These contributions establish a powerful framework for querying, analyzing, and securing domain certificates against potential vulnerabilities.

2 Certificate Transparency Logs

Certificate Transparency (CT) logs are an open framework designed to improve the security and trustworthiness of SSL/TLS certificates by making the issuance of certificates publicly visible. These logs are append-only, tamper-evident databases where Certificate Authorities (CAs) publish details of the certificates they issue. The primary goal of CT logs is to detect and prevent the misuse of certificates, such as unauthorized or fraudulent issuance, which can facilitate phishing attacks or man-in-the-middle exploits. Each entry in the log contains information about the certificate and is cryptographically signed to ensure its integrity.

2.1 Benefits of Certificate Transparency Logs

CT logs bring several critical benefits to the ecosystem of internet security. By providing a publicly accessible record, they allow domain owners, security researchers, and users to monitor the certificates associated with their domains. This transparency ensures that unauthorized certificates can be quickly detected and revoked. Moreover, CT logs enhance the accountability of CAs by requiring them to disclose all issued certificates, thus deterring malpractice or negligence. Modern browsers, such as Chrome, often mandate that certificates be logged in CT before being considered valid, further reinforcing their utility in promoting secure communication.

2.2 Challenges and Future Directions

Despite their benefits, CT logs are not without challenges. The vast number of certificates issued daily can make logs extensive, requiring efficient methods to store and query data. Ensuring the consistency and reliability of logs is also vital, as any log tampering can undermine trust in the system. In addition, domain owners must actively monitor logs to detect unauthorized certificates, which may not always be feasible without automated tools. Looking ahead, advancements in machine learning and better integration with browser and security tools are expected to make CT logs even more effective in safeguarding the Internet's certificate ecosystem.

2.3 Merkle Tree-Based Implementation

CT logs use Merkle trees to ensure the integrity and efficiency of certificate log datasets. A Merkle tree is a binary tree structure where each leaf node represents a cryptographic hash of a certificate entry, and each parent node is the hash of its two child nodes. This hierarchical structure allows CT logs to provide efficient cryptographic proofs, such as *inclusion proofs* (proving a specific certificate is part of the log) and *consistency proofs* (proving that a log has not been tampered with over time). The Merkle tree structure ensures that these proofs can be computed and verified in logarithmic time relative to the size of the log, making the system scalable even for large datasets.

Benefits and Challenges of Merkle Tree Implementation The use of Merkle trees in certificate log datasets provides several benefits. By leveraging cryptographic proofs, CT logs guarantee the immutability of logged data and enable domain owners and browsers to verify the presence or absence of certificates efficiently. This helps in quickly detecting misissuance and enforcing accountability among CAs. However, implementing and maintaining Merkle tree-based systems pose challenges, such as managing the computational overhead of updating the tree as new certificates are logged and ensuring the consistency of logs distributed across multiple servers. Despite these complexities, the integration of Merkle trees is essential for the robustness and trustworthiness of certificate log datasets.

3 Dataset

3.1 Merkle Trees in Certificate Transparency Logs

Certificate Transparency (CT) logs rely on Merkle trees to ensure integrity and efficiency in their operation. A Merkle tree is a cryptographic data structure where each leaf node represents the hash of a certificate entry, and each parent node is the hash of its two child nodes. This hierarchical hashing enables efficient cryptographic proofs, such as *inclusion proofs* (to verify that a certificate exists in the log) and *consistency proofs* (to ensure that the log has not been altered). By using Merkle trees, CT logs can guarantee tamper-evident record-keeping, making it feasible to detect unauthorized certificate issuance while keeping computational overhead manageable for both logging servers and clients.

3.2 Size and Scale of the Dataset

The dataset of certificate logs has grown to an immense scale due to the increasing number of certificates issued globally. With millions of new SSL/TLS certificates generated daily, CT logs accumulate entries at an extraordinary rate, resulting in datasets that can reach petabyte-scale storage over time. Each log entry includes essential metadata such as certificate fields, issuance timestamps, and signatures. Managing such a large dataset requires efficient storage solutions and querying mechanisms. The size also necessitates the use of distributed systems to handle the volume and ensure reliability, as each log must remain publicly accessible and verifiable for auditing purposes.

3.3 Integration with Google's Xenon Database

Google's Xenon database serves as the backbone for managing CT logs, offering high-throughput ingestion and retrieval of certificate data. Xenon is designed to handle the massive scale of CT operations, providing the speed and resilience needed to store and query millions of entries daily. Through Xenon, CT logs can be indexed and made available to clients via APIs for real-time querying and monitoring. Researchers and security practitioners use Xenon-powered CT log endpoints to audit certificates, track changes in issuance patterns, and identify misissuances. Its robust architecture supports the scalability and reliability required to maintain transparency and trust in the global Public Key Infrastructure.

3.4 Open Source crt.sh

While crt.sh provides an invaluable open-source interface for querying Certificate Transparency (CT) logs, it struggles to handle the massive scale of modern certificate issuance effectively. As the number of SSL/TLS certificates issued daily continues to grow, crt.sh faces challenges in terms of query latency, real-time updates, and handling concurrent users due to its reliance on traditional database architectures. In contrast, Google's proprietary Xenon database is specifically engineered for the high-throughput ingestion and querying of CT logs at scale. Xenon leverages distributed systems and optimized storage to manage the immense size of CT datasets while maintaining low-latency performance. However, unlike crt.sh, Xenon is not open source, limiting accessibility for researchers and developers outside of Google's ecosystem. This trade-off between openness and scalability highlights the need for a scalable, open-source solution that can bridge the gap for public CT log analysis.

4 Methodology

The inspiration for our new system, **CertSight**, was drawn from the hierarchical and distributed design of the Domain Name System (DNS). In DNS, each Top-Level Domain (TLD) has its own resolver, enabling scalability and efficient management of domain name queries. CertSight mirrors this architecture by creating a dedicated database for each TLD, ensuring logical isolation of data and improving query performance. This approach allows CertSight to handle the vast and growing dataset of Certificate Transparency logs efficiently.

Within each TLD-specific database, CertSight organizes entries into tables based on Common Names (CNs) using regex-based mappings, similar to how DNS organizes subdomains under their parent domains. This hierarchical design ensures scalability while optimizing performance for domain-specific queries. By drawing inspiration from the DNS system, CertSight combines a proven, scalable approach with domain-specific flexibility to effectively manage and analyze Certificate Transparency log data.

4.1 ETL pipeline

To query and retrieve data from the Certificate Transparency (CT) log database, we utilized the Python-based utility Axeman, which offers an efficient interface for interacting with public CT logs. Axeman was employed to download CT log entries in batches of 32 entries per query, storing each batch in a separate CSV file. This systematic retrieval allowed for manageable processing of the vast dataset. However, given the size of the dataset, spanning millions of entries, managing and processing the files efficiently became a critical challenge.

```sql
DO $$
DECLARE
  tbl_name TEXT;
  row_count INT;
BEGIN
  FOR tbl_name IN
    SELECT table_name
    FROM information_schema.tables
    WHERE table_schema = 'public'
      AND table_type = 'BASE TABLE'

  LOOP
    EXECUTE format(
        'SELECT COUNT(*)
          FROM (
            WITH duplicate_cns AS (
              SELECT cn
              FROM %I
              GROUP BY cn
              HAVING COUNT(*) > 1
            )
            SELECT t1.*
            FROM %I t1
            JOIN %I t2
              ON t1.cn = t2.cn
            AND t1.not_before <> t2.not_before AND
                 t1.not_after <> t2.not_after
            AND ABS(t1.timestamp - t2.timestamp) >
                 120000
            WHERE t1.cn IN (SELECT cn FROM
                duplicate_cns)
              AND t1.timestamp < (t1.not_before *
                  1000) + ((t1.not_after * 1000) -
                   (t1.not_before * 1000)) / 3
          ) filtered_rows;',
        tbl_name, tbl_name, tbl_name
    ) INTO row_count;

    RAISE NOTICE 'Table: %, Matching Rows: %',
        tbl_name, row_count;
  END LOOP;
END $$;
```

Listing 1: SQL Script for Processing Tables in a Schema

To address the issue of handling millions of generated CSV files, we developed a C program to directly interact with the Linux dirent structure. This program efficiently traversed directories to extract only the relevant filename information, significantly reducing the overhead associated with higher-level file management libraries. Using the extracted filenames, the full file paths were constructed dynamically, enabling the ingestion of the CSV files into a PostgreSQL database. The COPY FROM functionality of PostgreSQL, optimized for bulk data loading, was used to import the contents of all CSV files into a single table named ct_logs. This centralized repository provided a unified dataset for further processing and analysis.

A subsequent step involved reorganizing and distributing the data in the ct_logs table across multiple databases. A Python script was developed to read the data from ct_logs and partition it based on the top-level domains (TLDs) of the certificate records. For each unique TLD, a dedicated PostgreSQL database was created, and the corresponding data was inserted into its respective database. This approach improved query performance and scalability by organizing the data in a domain-specific manner, enabling efficient retrieval and analysis for targeted use cases. The end-to-end pipeline, from data retrieval to distribution, ensured the scalable and efficient processing of CT log data while maintaining flexibility for domain-specific analyses.

4.2 Overcoming the precert problem

Precertificates were also logged in Certificate Transparency logs. We had to make sure our query doesn't give us false positives. The pre-certificates are usually logged in close proximity of the original certificate (within a few seconds). To over come this, we used a heuristic approach of ignoring all the entires that are in the proximity of 2 minutes as seen in Listing 1. By setting this threshold, we saw that, most of the false positives of precertificate entries were eliminated.

4.3 Schema used in our new proposed system - CertSight

The proposed system, **CertSight**, introduces a hierarchical and organized database schema to efficiently store and manage Certificate Transparency (CT) log data. The schema leverages a two-tiered structure to ensure scalability and improve query performance by distributing data into specialized databases and tables.

At the first level, a dedicated PostgreSQL database is created for each high-level domain, also known as a Top-Level Domain (TLD). For example, all entries associated with domains ending in .com are stored in the com database, while entries with .org or .net TLDs are stored in the org and net databases, respectively. This division allows for efficient data partitioning and targeted queries based on TLDs, ensuring that CertSight remains scalable as the volume of CT log data grows.

Within each TLD-specific database, tables are created for distinct Common Names (CNs). CNs are grouped into tables based on a predefined regular expression (regex) that maps subdomains and wildcard entries to their parent domain. For instance, entries with CNs such as mail.google.com or *.google.com are stored in a table named google.com within the com database. Similarly, entries with CNs like azure.com, *.azure.com, and *.foo.azure.com are stored in the azure.com table within the same database. This structure ensures that all certificates related to a specific domain are consolidated into a single table, simplifying domain-specific queries and analysis.

The hierarchical schema of CertSight—TLD-based databases and CN-based tables—provides a robust and efficient approach to managing CT log data. By partitioning data in this manner, CertSight optimizes storage and retrieval, minimizes query complexity, and facilitates streamlined monitoring of domain-specific certificate activities.

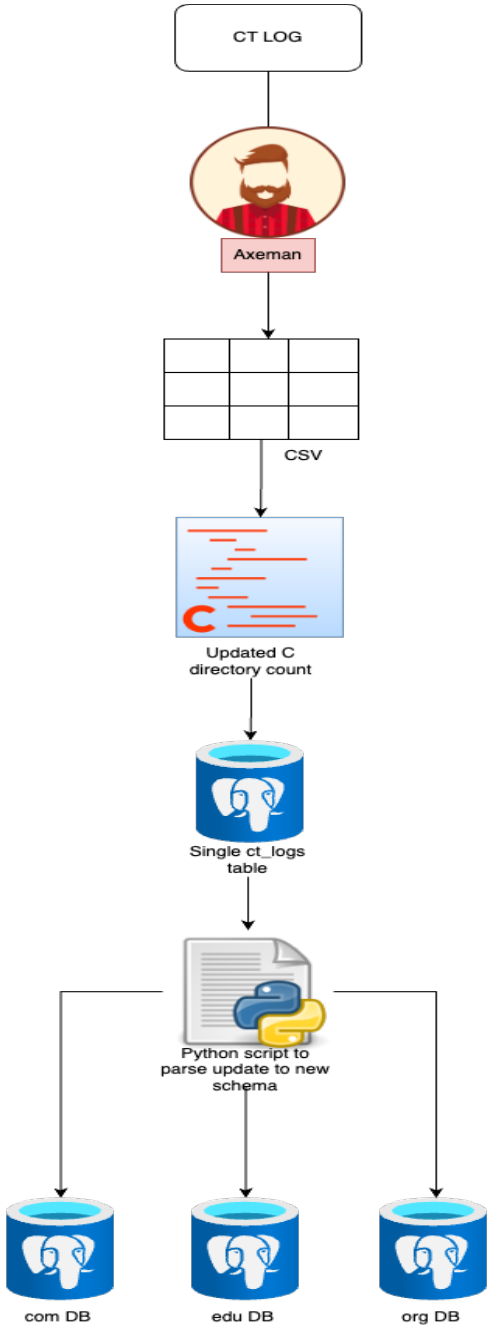

CT LOG
Axeman
CSV
Updated C directory count
Single ct_logs table
Python script to parse update to new schema
com DB
edu DB
org DB

Figure 1: ETL flow of our data pipeline.

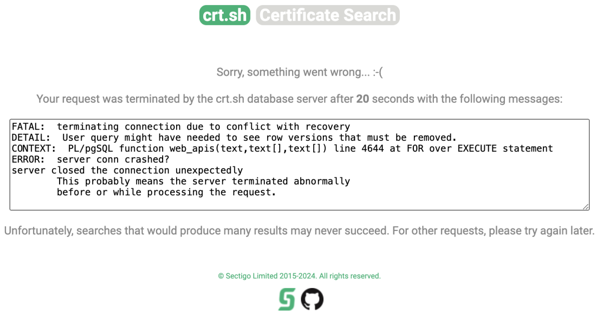
4.4 Notification System

Notification System for Anomalous Certificate Activity: As part of our system, we propose a notification mechanism to alert domain owners about potential anomalies in certificate renewal patterns. When a Common Name (CN) is identified in a newly queried Certificate Transparency (CT) log, the system checks its last recorded entry in CertSight. The lookup is really quick becasue of the way CertSight's schema is designed. If the newly issued certificate is found to be renewed before one-third of its expiration time, the system flags it as suspicious. To notify the domain owners, a WHOIS lookup is performed to retrieve their contact information, and an email alert is sent to inform them of the irregular activity. This proactive approach empowers domain owners to investigate and address potential security concerns promptly.

5 Results

5.1 Query Performance improvements over other CT Log monitors

We benchmarked our system CertSight against well known CT Log monitors like crt.sh. Firstly, we noticed crt.sh was very unreliable in terms of performance and the quries seemed to be timing out after 20 seconds most of the times.



crt.sh Certificate Search

Sorry, something went wrong... :-(

Your request was terminated by the crt.sh database server after **20** seconds with the following messages:

```
FATAL:  terminating connection due to conflict with recovery
DETAIL:  User query might have needed to see row versions that must be removed.
CONTEXT:  PL/pgSQL function web_apis(text,text[],text[]) line 4644 at FOR over EXECUTE statement
ERROR:  server conn crashed?
server closed the connection unexpectedly
        This probably means the server terminated abnormally
        before or while processing the request.
```

Unfortunately, searches that would produce many results may never succeed. For other requests, please try again later.

© Sectigo Limited 2015-2024. All rights reserved.

Figure 2: Query erroring out with ambiguous message in crt.sh

Here are a few queries which were ran on both the systems and we compare the time taken by each:

Table 1: Time taken by crt.sh and CertSight in seconds

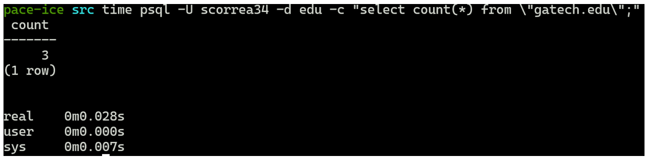
Domain	crt.sh	CertSight
gatech.edu	17.756	0.28
umich.edu	11.32	0.021
uw.edu	25.6	0.249

We see that our system performs 6241.42% faster than crt.sh.

```
$ time curl -X GET https://crt.sh/?q=gatech.edu >
    /dev/null
  % Total    % Received % Xferd  Average Speed
      Time    Time     Time  Current
                                 Dload  Upload
                                     Total   Spent
                                         Left
                                     Speed
100 1778k    0 1778k    0     0   116k      0 --
    :--:--  0:00:15 --:--:--  544k

real    0m17.756s
user    0m0.045s
sys     0m0.031s
```

Listing 2: Time of curl request made to crt.sh



```
pace-ice src time psql -U scorrea34 -d edu -c "select count(*) from \"gatech.edu\";"
 count
-------
     3
(1 row)


real    0m0.028s
user    0m0.000s
sys     0m0.007s
```

Figure 3: Query Time of our Model

5.2 Statistics Captured

The Certificate Transparency (CT) logs dataset is immense, containing billions of certificate entries that serve as a public ledger for SSL/TLS issuance. To test our approach, we downloaded a subset of 114 million entries from this dataset. This sample provides a manageable yet representative view of the broader CT ecosystem, allowing for meaningful analysis of certificate issuance patterns and potential security threats. Within this dataset, we focused on certificates associated with the 1,514 currently recognized Top-Level Domains (TLDs), providing insights into the distribution of certificates across diverse domain categories.

In our analysis of this sample, we identified 442 '.edu' domains with a total of 1,537 certificates. Among these, we detected 20 potential anomalies that could signify malicious activity, such as unauthorized certificate issuance or misconfigured certificate details. These anomalies highlight the utility of CT logs in identifying threats and improving certificate ecosystem security. While this sample is only a fraction of the complete dataset, it demonstrates the effectiveness of targeted analysis in uncovering issues within domain-specific subsets of the CT logs.

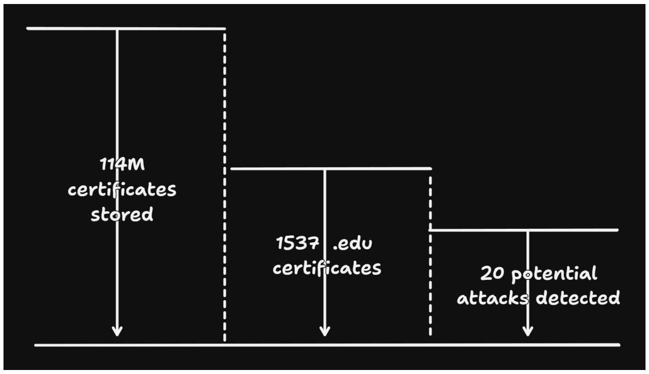


Figure 4: Waterfall diagram showing the CT log numbers for the TLD - edu

6 Discussion

The research on Certificate Transparency (CT) logs reveals critical insights into the challenges of managing and analyzing the rapidly growing ecosystem of digital certificates. Our proposed system, CertSight, addresses several key limitations in existing CT log querying infrastructure, demonstrating significant improvements in performance and analysis capabilities. The performance benchmarking against crt.sh highlights the substantial bottlenecks in current open-source CT log monitoring tools. With query times reduced by over 6,000%, CertSight proves that innovative architectural approaches can dramatically enhance the efficiency of certificate data retrieval. This is particularly crucial given the exponential growth of SSL/TLS certificates, which now number in the millions daily. Our hierarchical approach, inspired by the Domain Name System (DNS), introduces a novel method of organizing certificate data. By creating TLD-specific databases and CN-based tables, we've developed a scalable solution that not only improves query performance but also simplifies domain-specific analyses. This approach addresses the fundamental challenge of managing petabyte-scale datasets while maintaining query efficiency. The analysis of our 114-million-entry dataset yielded particularly interesting findings. The identification of 20 potential anomalies within just 442 .edu domains underscores the critical importance of systematic CT log monitoring. These anomalies could represent unauthorized certificate issuances, potential security breaches, or misconfigurations that might otherwise go undetected. Challenges and Future Work Despite the promising results, several areas remain for future research and improvement:

Developing more sophisticated anomaly detection algorithms Implementing machine learning techniques for predictive threat analysis Creating more comprehensive automated notification systems Exploring methods to further optimize distributed database performance

Additionally, as part of this project, we also made an **opensource contribution to the Axeman project**

https://github.com/CaliDog/Axeman/pull/25.

7 Conclusion

Certificate Transparency logs represent a fundamental mechanism for enhancing internet security, providing unprecedented visibility into the certificate issuance ecosystem. Our research contributes a significant advancement in CT log analysis through the CertSight system, addressing critical challenges of scale, performance, and actionable insight generation. The key contributions of this work include:

A novel, scalable system for efficient CT log querying A hierarchical database approach that dramatically improves query performance A methodology for identifying potential security anomalies in certificate issuance Empirical evidence of the system's effectiveness in processing large-scale certificate datasets

As digital communication continues to grow in complexity and scale, tools like CertSight become increasingly vital. By providing domain owners, security researchers, and organizations with a powerful mechanism to monitor and analyze certificate activities, we take a significant step toward a more transparent and secure internet ecosystem. The future of Certificate Transparency lies in continuous innovation—developing more intelligent, responsive, and comprehensive monitoring systems that can keep pace with the evolving landscape of digital certificates and cybersecurity threats.

8 Appendix

All the code written as part of this project is available in this GitHub repo: $https : //github.com/MohammedAdain/sii_project$.

References

[1] Certificate Transparency Google Group. 2024. Certificate Transparency Discussion: "I74Wp-KdWHc". https://groups.google.com/g/certificate-transparency/c/I74Wp-KdWHc. Accessed: 2024-12-10.

[2] Security Stack Exchange. 2024. Certificate Transparency Logs: Why are so many operated by same entities, and how? https://security.stackexchange.com/questions/237868/certificate-transparency-logs-why-are-so-many-operated-by-same-entities-and-how Accessed: 2024-12-10.

[3] Laurence Lundblade, Adam Langley, and Ben Laurie. 2013. Certificate Transparency. https://www.rfc-editor.org/rfc/rfc6962#section-5.3 Accessed: 2024-12-10.

[4] Ryan Sears. 2017. Parsing Certificate Transparency Lists Like a Boss. *Medium* (2017). https://medium.com/cali-dog-security/parsing-certificate-transparency-lists-like-a-boss-981716dc506

[5] Google Certificate Transparency. 2024. CT Log List (v3). https://www.gstatic.com/ct/log_list/v3/log_list.json Accessed: 2024-12-10.

[6] Google Certificate Transparency. 2024. Fetch Logs Documentation. https://github.com/google/certificate-transparency-community-site/blob/master/docs/google/fetch-logs.md Accessed: 2024-12-10.