Network Security and Measurement Assignment 02

HAW Hamburg SS 2026

Prof. Dr. Thomas Schmidt, Timo Salomon, M.Sc., Yue Xin, M. Sc. Deadline: May 6, 2026

1. **TLS Early Data**

   TLS 1.3 comes with a lot of improvements for security and performance. A core performance feature is the support of "early data", also known as 0-RTT data. This allows TLS to forgo the extra round trip when re-establishing connections. Scanning TLS can be done by hand or with specialized tools. In this exercise we take a look at `sslyze` and use it to scan TLS 1.3 support for "early data".

   Tools: sslyze[1]. *Dataset:* The Tranco top list located in `shared-data`.

   (a) Read the documentation and get the *Full Example* running. Explain its workings.

   (b) Modify your script to check for *TLS 1.3* and *TLS 1.3 Early Data* support for a given list of domains.

   (c) Collect data for a sample of 1000 top list entries and present your findings.

2. **Securing MX Records with DANE**

   While DANE has the potential to improve security of all TLS interactions it sees more use with mail servers than for general web browsing. In this exercise we will compare the deployment of DANE (estimated through the existence of TLSA records) for mail servers and web servers.

   Tools: dig, dnspython, pydig, ldns. *Dataset:* The Tranco top list located in `shared-data`.

   (a) Implement the lookup for the `TLSA` records for the web and mail server of a given domain. (*Remember that the name of the `TLSA` records includes the port and transport protocol in addition to the domain.*)

   (b) Collect a dataset for 1000 sampled top list entries.

   (c) Visualize your findings. Check your mail provider for comparison.

3. **CT Log Verification and Usage**

   In 2018 Google made the use of certificate transparency mandatory for new certificates, at least for website that want to be considered trusted by the Chrome browser. CT requires publication of certificates in at least two different logs.

   Tools: ctutlz[2]. *Dataset:* The Tranco top list located in `shared-data`.

   Specific Python env: Since ctutlz is no longer actively maintained it is not compatible to other libraries in our base env (e.g., sslyze). Use a venv on your machine for this task with ctutlz and its requirements (e.g., pip install pyOpenSSL==23.3.0 ctutlz). We have a secondary Jupyter kernel with ctutlz installed, which has stille some issues... To activate this env in a terminal run
   source /opt/tljh/user/bin/activate && conda activate venv_ctutlz

[1] `https://nabla-c0d3.github.io/sslyze/documentation/`

[2] https://pypi.org/project/ctutlz/

(a) Use `verify-scts` CLI tool (in the ctutlz env) to verify `google.com`.

(b) Implement a python script to do the measurement yourself based on ctutlz. You can also use other libraries if you want.

- The REPL code in the README is missing a step, check the code of `verify_scts.py` for the ctlog setup.
- Use the cert verification method per default.
- Collect the number of logs that verify a domain, potential failures, as well as the related log names.
- Handle exceptions and record errors.

(c) Collect data for 1000 sampled top list entries.

(d) Visualize statistics for logs per certificate and log usage.