Network Security and Measurement Assignment 02

HAW Hamburg WS 2020

Prof. Dr. Thomas Schmidt, Raphael Hiesgen, M.Sc. Deadline: November 18, 2020

1. **Scanning TLS Versions**

   TLS 1.3 improves security by (among other things) limiting the available cipher suits. This is only good if servers deprecate older versions of TLS (and SSL) and don't allow falling back to insecure algorithms. Scanning TLS versions can be done by hand or with specialized tools. In this exercise we take a look at `sslyze` and use it to scan TLS versions and supported ciphers.

   Tools: sslyze[1]. *Dataset:* The Alexa top 1M list located in `shared-data` on mobi8.

   (a) Read the documentation and get the basic example running. Explain its workings.

   (b) Extend your script to check for the supported versions of TLS 1.1 and above.

   (c) Run your script over the first 1000 entries of the Alexa top 1M list and report on your findings. (Are any insecure ciphers still in use? You can check the recommendations of the IETF or Mozilla for comparison.)

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 Alexa top 1M list located in `shared-data` on mobi8.

   (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 the top 1000 entries of the Alexa list.

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

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

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 Alexa top 1M list located in `shared-data` on mobi8.

(a) Use `verify-scts` CLI tool to verify `google.com`.

(b) Implement a python script to do the measurement yourself based on ctutlz.

- 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 the Alexa top 1000 entries.

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

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