Associate Professorship of Embedded Systems and Internet of Things
Department of Electrical and Computer Engineering
Technical University of Munich

Evaluating TLS-support of Web Pages and APIs

Research Internship

Jan Lauinger, March 3rd, 2023

Title:	"Evaluating TLS-support of Web Pages and APIs"
Supervisor:	Jan Lauinger
Period:	24 weeks part time or 9 weeks full time

Context

Protocols of TLS-oracles deviate according to different cryptographic properties found in standardized TLS cipher suites [1, 2]. Depending on a TLS version, different cipher-suites are supported on a server or client. For example, TLS 1.3 supports the cipher suite `TLS_AES_128_GCM_SHA256`, which is not found in the specification of TLS 1.2. In order to investigate the popularity of TLS cipher-suites, your task is to build a TLS cipher suite scanner which is able to output cipher suite support per investigated web domain or API.

In the beginning of the research internship, your task is to compile an overview of existing TLS scanners and identify which information about cipher suites can be extracted based on TLS scanner outputs. The overview will be presented to your supervisor such that a decision can be made about the necessity of coding a new TLS scanner or if wrapping an existing TLS scanner is sufficient.

In the second phase, you will implement the TLS scanner. The scanner should be able to read in a list of domains, subdomains, and API endpoints which can be selectively queried. The program should support timeout handling and perform queries concurrently. You should use Golang as your choice of programming language and develop the scanner with as little dependencies as possible. Please make use of the concurrency features provided by the Golang default data structures and randomize requests to prevent blacklisting based on simple metrics such as requests per second. The output of the program should be configurable and should produce either a file or display results directly on the command line interface. To test the applicability of the scanner, use the Alexa Top 1 million list of domains[1] and limit the input according to your testing needs.

Requirements

- High: Independent work ethics.
- Low: Knowledge of the Golang programming language and its concurrency features.

Preliminary Schedule

Task / Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Compile overview of TLS scanners	■	■	■	■	■	■																		
Presentation of Overview & Planning						■	■	■																
Implementation & Evaluation of TLS scanner									■	■	■	■	■	■	■	■	■	■	■	■				
Report writing																		■	■	■	■	■	■	■

References

[1] Fan Zhang, Deepak Maram, Harjasleen Malvai, Steven Goldfeder, and Ari Juels. Deco: Liberating web data using decentralized oracles for tls. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 1919–1938, 2020.

[2] Paul Grubbs, Arasu Arun, Ye Zhang, Joseph Bonneau, and Michael Walfish. Zero-knowledge middleboxes. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 4255–4272, 2022.

[1] https://github.com/mozilla/cipherscan/tree/master/top1m