Reproduction: Measuring the Security Harm of TLS Crypto Shortcuts

Jared Crawford
jared13@stanford.edu

ABSTRACT

TLS provides secure communication lines between end users and web servers. However, establishing this communication line requires additional overhead in the form of transport level handshake round trips and computationally expensive cryptographic operations. To reduce this overhead in bursty web traffic, HTTPS supports many mechanisms (such as TLS session tickets and private value reuse in public key exchange) to reuse these secure connection values for a short period of time. These performance optimizations tradeoff with the forward secrecy of user web traffic. In early 2016, a team at the University of Michigan investigated the extent to which the most popular sites were reusing these values in order to make site operators more aware of the risk involved in doing so[1]. In this report, I attempt to reproduce these findings by collecting more recent data (May 2018) on TLS cryptography shortcutes.

1 INTRODUCTION

The authors made TLS 1.2 handshakes over a nine week period with the Alexa Top Million sites and studied the reuse of private ephemeral values in TLS session resumption, TLS session tickets, and Diffie Helmann key exchange. In addition, the researchers observed more general trends such as the churn of the top million sites and protocol usage statistics. In this report, I present a reproduction of these results two years after the papers publication. Due to time constraints, the data collected spans a comparatively smaller one week period compared to the original paper.

The key results of this paper which I reproduce are Figures 1-4 in the paper. TLS supports user advertised lifetimes of ephemeral secrets. For example, a client can specify that their session key should be deleted after 1 hour to maintain a higher degree of forward secrecy. Figures 1-2 demonstrate how compliants sites are with these advertised ephemeral lifetimes. Figures 3-4 show how long (in days) that session ticket encryption keys are reused by sites. Reuse of encryption keys expose users to additional risk because it requires that the server save a copy of these keys for their lifetime. Thus, if a malicious party gains server-side access to a site which reuses keys for one week, client communications during the past week can be retroactively decrypted. Additionally, I present a summary of what TLS features are supported by popular domains (e.g. "of the Majestic Million websites that support HTTPS, 83% support session ID resumption and 76% support session tickets.") This data for this analyisis was collected over a one week day period from May 26 to June 2.

2 REPRODUCTIONS

2.1 Figures 1-2

2.2 Figures 3-4

3 EVALUATION

4 CONCLUSION

5 PROGRESS

I have spent a significant amount of time getting clean installs of ZMap/ZTee/ZGrab as they haven't had a clean build from source in 6 months. Additionally, I had to find a new data source for the top sites since the daily Alexa Top 1M is no longer available for research purposes.

Week of May 7: Setup Go programming environment for building ZMap/ZTee/ZGrab. Lots of reading through documentation for these tools to figure out how to extract TLS fields from a fixed set of sites. Began working on plots with mocked up data; implementing a storage for STEK values.

Week of May 14: Search for new source of "Top sites" list. Alexa Top 1M (and most other top sites list) moved to paid API access instead of free research access. I settled on the Majestic Million list as it was the first free list I found that is updated somewhat regularly. Unlike Alexa Top 1M, the Majestic Million measures in links (more comparable to TrustRank than PageRank of sites). It will be interesting to see if churn / TLS support is significantly different for this list than the Alexa Top 1M.

6 FUTURE WORK

The original CDFs for reuse looked over a 90 day period. Due to time constraints, my new goal is to give a one week sample for figures 3-4. Although this data will be less stratified, STEKs are recommended to be recycled every 24 hours (add citation here), so this should still give a good general picture of STEK reuse.

6.1 Timeline

Week of May 21: Deploy a process to continually scan Majectic Million sites while persisting Session Ticket Encryption Keys to be used in the reproduction of figures 3 and 4.

Week of May 28: Deploy a process to retry Session ID and Session Ticket lifetime data values for the Majestic Million sites every 5 minutes until failure. This data will be used in the reproduction of figures 1 and 2. Analyze the differences between the reproduced results and the findings at the publication of the original paper. Write the final report and blog post summarizing the paper and the reproduced results.

REFERENCES

[1] Drew Springall, Zakir Durumeric, and J. Alex Halderman. 2016. Measuring the Security Harm of TLS Crypto Shortcuts. In *Proceedings of the 2016 Internet Measurement Conference (IMC '16)*. ACM, New York, NY, USA, 33–47. https://doi.org/10.1145/2987443.2987480