# Reproduction: Measuring the Security Harm of TLS Crypto Shortcuts

Jared Crawford
jared13@stanford.edu

## ABSTRACT

Server authenticity, privacy over communication lines, and integrity of transport level data are desirable for many popular applications such as web browsing, email, and Voice over IP. The Transport Layer Security protocol (TLS) provides these security properties between web servers and web browsers at the transport layer. A typical connection over TLS today is a multistep interactive process between server and client which includes an exchange of public key information, a negotiation on which ciphers to use for the resulting communication, and advertisement for support of a variety of TLS extensions. Thus, establishing this secure communication line incurs an initial latency overhead in the form of multiple transport level handshake round trip times. [1] To reduce this overhead for the common case of bursty connections in frequent but short intervals, 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[2]. In this report, I attempt to reproduce these findings by collecting more recent data (May 2018) on TLS cryptography shortcutes.

## 1 INTRODUCTION

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.

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 two week period compared to the ninety day period of the original paper.

The key results of this paper which I reproduce correspond to Figures 2-4 in the paper. 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 DATA SOURCE

A verbose json transcript of each TLS handshake used in the analysis is included in this reproduction's Google Cloud Storage bucket (identifier cs244-jared13-tls-crypto). In total, X TLS handshakes were collected across X distinct domains over a period of 14 days. Only X domains remained in the Top Million for the entire X days. Of these, X ever completed a TLS handshake with a browser-trusted certificate. The distribution of the :, failed connections is as follows: TABLE

### 2.1 Churn

## 3 REPRODUCTIONS

### 3.1 Figure 2

Figure 2

### 3.2 Figures 3-4

Figure 3
Figure 4

## 4 EVALUATION

## 5 CONCLUSION

## REFERENCES

[1] J. Salowey, H. Zhou, P. Eronen, and H. Tschofenig. 2008. *Transport Layer Security (TLS) Session Resumption without Server-Side State.* RFC 5077. RFC Editor. http://www.rfc-editor.org/rfc/rfc5077.txt http://www.rfc-editor.org/rfc/rfc5077.txt.

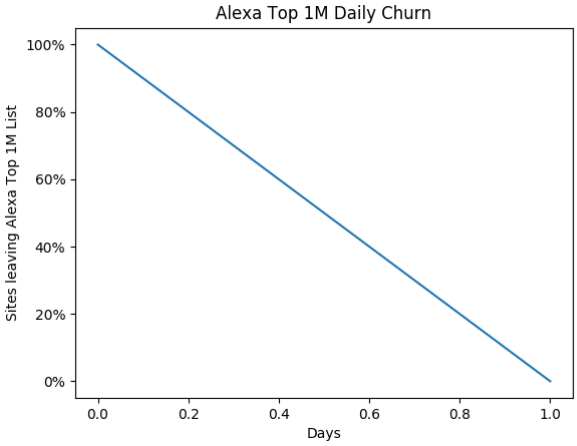


**Figure 1: Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.**

[2] 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

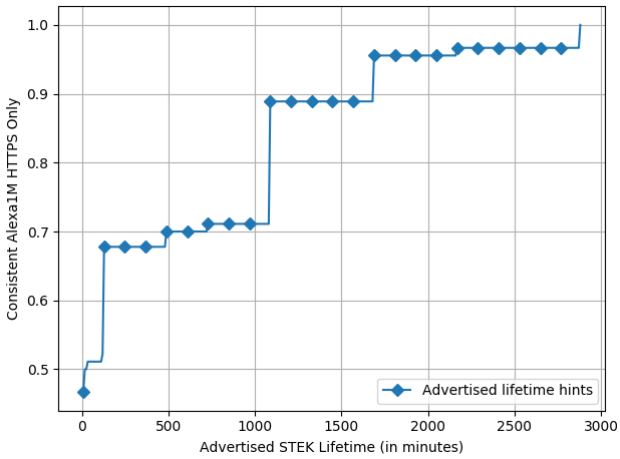


**Figure 2: Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.**

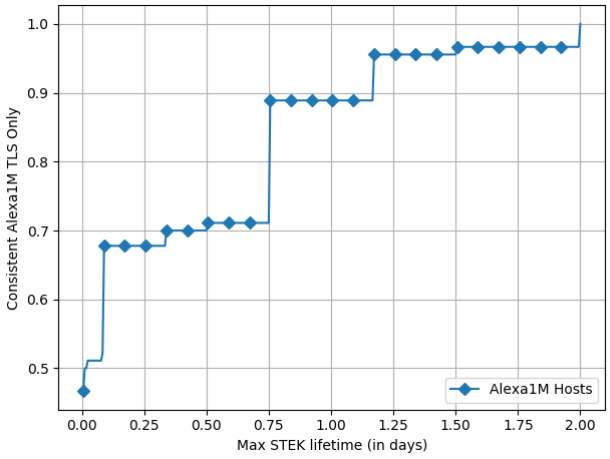


**Figure 3: Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.**

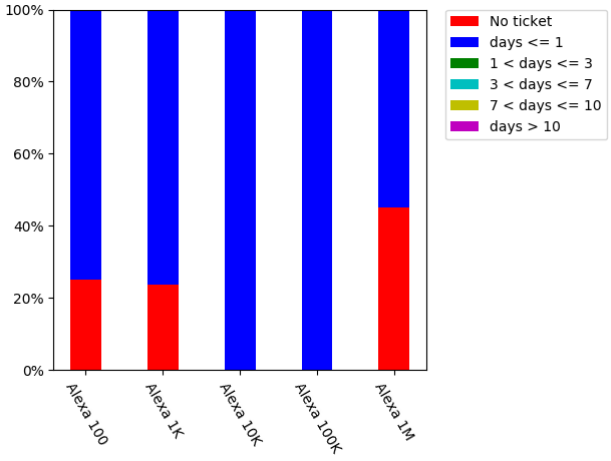


**Figure 4: Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.**