Thesis Project Portfolio

s2n-tls Benchnarking and Comparative Analysis

(Technical Report)

Analyzing the Motivations Behind Open Source Software Development

(STS Research Paper)

An Undergraduate Thesis

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Justin Zhang

Spring, 2024

Department of Computer Science

Table of Contents

Sociotechnical Synthesis

Heartbleed was a software vulnerability in 2014 that allowed anyone to get passwords, addresses, encryption keys, and other sensitive information from more than a third of the world's websites (Synopsys, 2020). Kerner (2014) estimated the cost of Heartbleed to be $500 million. Surprisingly, it is actually possible to track down who wrote the single line of code responsible for Heartbleed, since it was a vulnerability in an open-source software library, OpenSSL (Henson, 2014). However unlikely, this person could have intentionally hidden Heartbleed in a way that would get it past code reviews by other contributors. All open-source software, which powers the vast majority of software, is vulnerable to such bad actors.

The STS portion of my thesis analyzes open-source software through the lens of the Social Construction of Technology. I looked at the existing incentives for developers and users to ensure that open-software is "good:" that it has high security, is well-maintained, and has adequate funding. I identify and use the social groups of users, developers, user-developers, large companies, non-profit organizations, and hobby programmers. I analyze the views of each social group in the areas of open-source vs proprietary software, security, and funding. I argue that there is significant closure in open-source software, where social groups perceive any problems that a technology addresses as being solved and innovation decreases. This is evidenced by its already widespread use: there is good reason most software either is open source or relies on open-source software. Using SCOT, we can see that despite potential drawbacks, open-source software is useful to almost all social groups; everyone has a stake in open-source software.

The technical portion of my thesis reflects on my experience as an intern at Amazon Web Services (AWS). My team lacked simple, fast, and easy-to-use comparative benchmarks, so I created a custom benchmarking harness measuring handshake latency, throughput, and memory

usage. I compared the performance between three different TLS protocol implementations: s2n-tls, OpenSSL, and Rustls. All three are security-oriented open-source software libraries, with s2n-tls being created by AWS partly in response to Heartbleed and made open-source for customer trust. I found that the performance of s2n-tls and Rustls was better than OpenSSL, despite OpenSSL still being the most popular. This shows how far open-source technology has matured to have viable competitors in the same application.

Overall, my research was successful. My STS research reassures us that despite any qualms someone might have about open-source software, relevant social groups generally have net positives intentions and effects on open-source software. My technical project was a deep dive into three specific open-source libraries, and it succeeded in filling the gap in s2n-tls benchmarks, being simple, fast, and reliable. I produced useful and actionable results that my team did not have before: s2n-tls generally outperforms Rustls and OpenSSL.

As with all research, there are plenty of avenues for future work. Additional STS research could examine other aspects of good software, such as scalability and usability. Additionally, I could analyze the problem from the perspectives of different STS frameworks, such as Actor-Network Theory or Technical Momentum. For my technical project, the most impactful extensions to it would be to incorporate the benchmarks into testing to catch performance regressions earlier, to test more libraries, and to test more parameters. I could also design a performance dashboard that updates automatically for more readability and ease-of-use, especially for external (non-AWS) customers.

For my STS research, I would like to thank my STS 4500 and 4600 teachers, Travis Elliot and Professor Caitlin Wylie, for helping to frame my whole project and providing valuable feedback throughout the whole process. For my technical project, I would like to thank my

mentor James Mayclin, for thoughtfully and patiently answering every question I had; my manager, Wesley Rosenblum, for always being understanding and flexible; and finally, the whole Transport Libraries team that I had the honor of being a part of. Also, since s2n-tls is open source, we can see all the code for my benchmarks online; it is fulfilling to see my team still building on my project to this day!