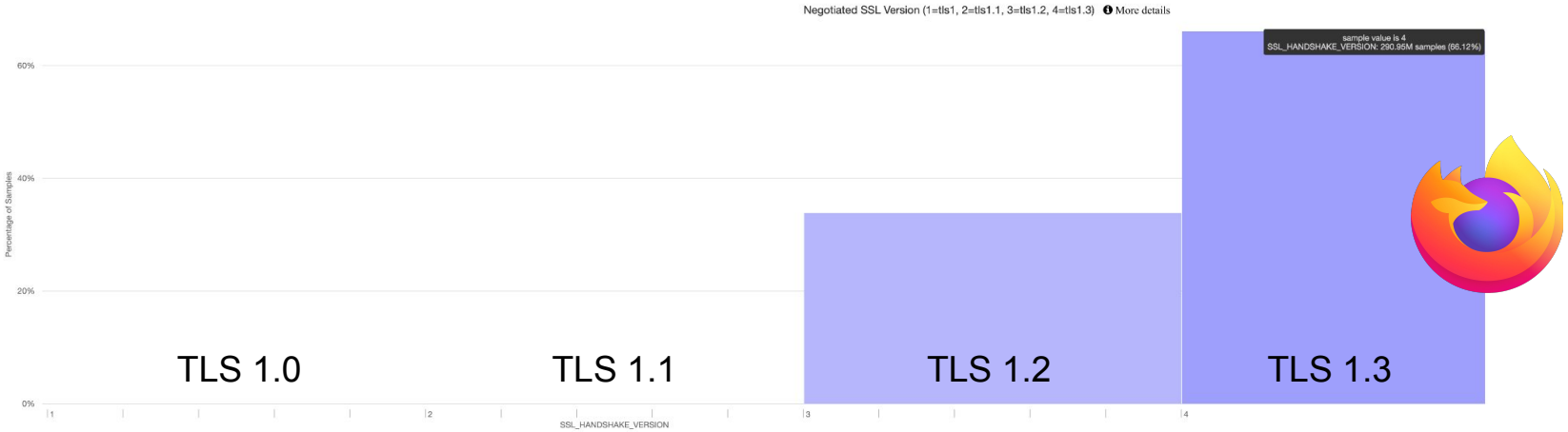
A Client-Side Seat to TLS Deployment

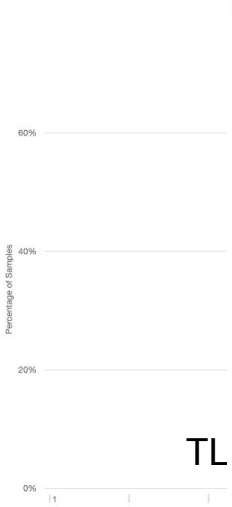
Moritz Birghan
Mozilla Corporation
mbirghan@mozilla.com

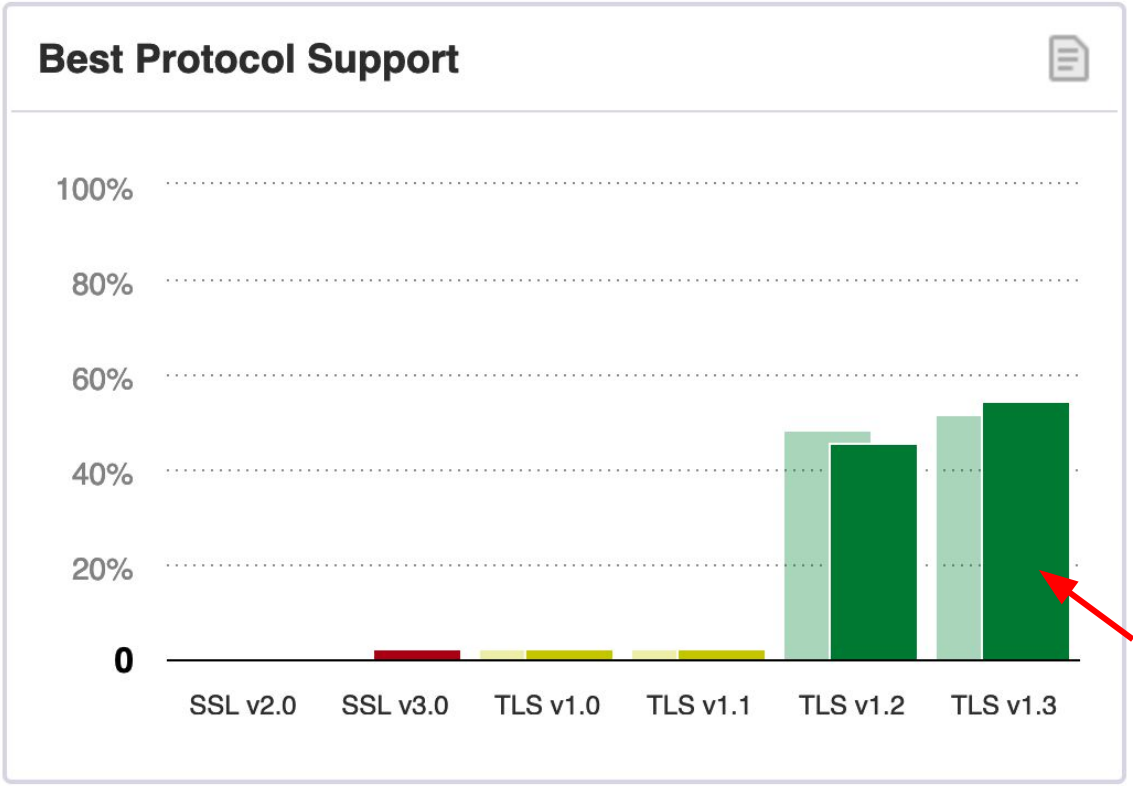
Thyla van der Merwe
ETH Zurich
tvdmerwe@ethz.ch

TLS is Everywhere!



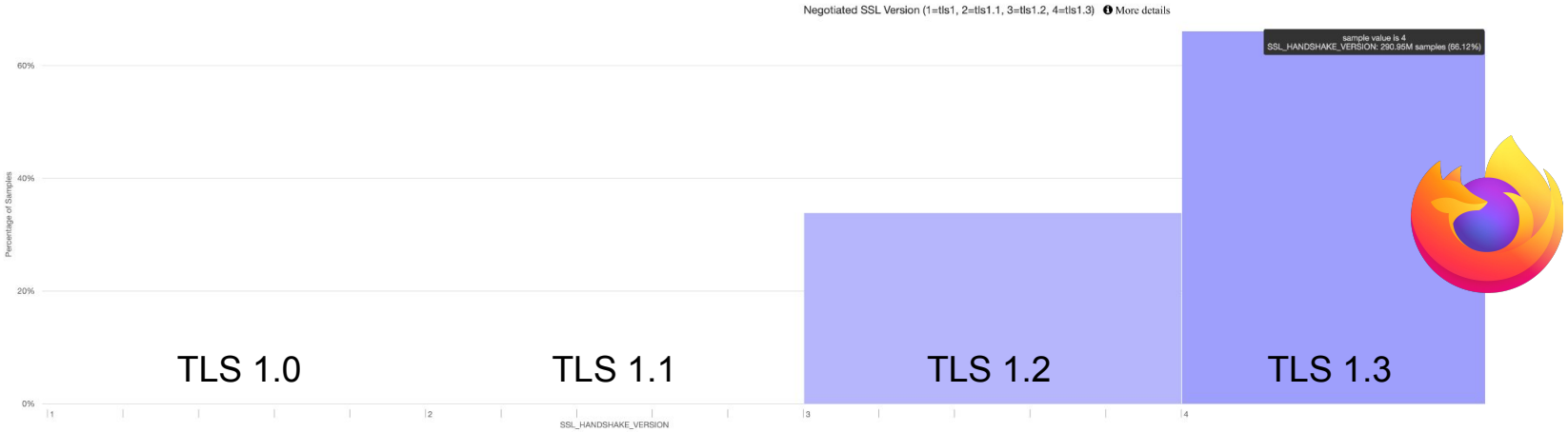
TLS is Everywhere!





Top version supported according to SSL Pulse (https://www.ssllabs.com/ssl-pulse/)

TLS is Everywhere!



TLS 1.0 → TLS 1.1 → TLS 1.2 → TLS 1.3

1999 2006 2008 2018

TLS is Everywhere!

TLS 1.2	TLS 1.3
37 cipher suites	3 mandatory cipher suites
2-RTT	1-RTT and 0-RTT
Client certificate authentication only during initial handshake	Post Handshake Authentication
	More handshake encryption - from early on in the handshake

Measuring TLS in the Wild

- Awareness of evolving protocol support
- Response to updates
- Data driven decision making
- New features not as robust

Measuring TLS in the Wild

Internet Measurement Conference, 2018

Coming of Age: A Longitudinal Study of TLS Deployment

Platon Kotzias
IMDEA Software Institute
Universidad Politécnica de Madrid

Abbas Razaghpanah
Stony Brook University

Johanna Amann
ICSI/Corelight/LBNL

Kenneth G. Paterson
Royal Holloway, University of London

Narseo Vallina-Rodriguez
IMDEA Networks Institute
ICSI

Juan Caballero
IMDEA Software Institue

ABSTRACT

The Transport Layer Security (TLS) protocol is the de-facto standard for encrypted communication on the Internet. However, it has been plagued by a number of different attacks and security issues over the last years. Addressing these attacks requires changes to the protocol, to server- or client-software, or to all of them. In this paper we conduct the first large-scale longitudinal study examining the evolution of the TLS ecosystem over the last six years. We place a special focus on the ecosystem's evolution in response to high-profile attacks.

For our analysis, we use a passive measurement dataset with more than 319.3B connections since February 2012, and an active dataset that contains TLS and SSL scans of the entire IPv4 address space since August 2015. To identify the evolution of specific clients we also create the—to our knowledge—largest TLS client fingerprint database to date, consisting of 1,684 fingerprints.

We observe that the ecosystem has shifted significantly since 2012, with major changes in which cipher suites and TLS extensions are offered by clients and accepted by servers having taken place. Where possible, we correlate these with the timing of specific attacks on TLS. At the same time, our results show that while clients, especially browsers, are quick to adopt new algorithms, they are also slow to drop support for older ones. We also encounter significant amounts of client software that probably unwittingly offer unsafe ciphers. We discuss these findings in the context of long tail effects in the TLS ecosystem.

ACM Reference Format:
Platon Kotzias, Abbas Razaghpanah, Johanna Amann, Kenneth G. Paterson, Narseo Vallina-Rodriguez, and Juan Caballero. 2018. Coming of Age: A Longitudinal Study of TLS Deployment. In *2018 Internet Measurement*

of each new attack and vulnerability that is discovered. Over the last few years various TLS vulnerabilities such as BEAST, Lucky 13, POODLE, Heartbleed, FREAK, Logjam, and multiple attacks against RC4 have been discovered. The Snowden revelations have also highlighted weaknesses in TLS, specifically the reliance on RSA key transport for establishing keying material, a method that can be passively broken by an entity in possession of the server's RSA private key. Addressing these attacks requires changes to the protocol, to server-, or to client-software, or to all of them simultaneously.

Prior work highlights different parts of the TLS ecosystem like specific attacks [6, 9, 10, 17, 41, 44, 44, 63, 74, 82], problems of the PKI [7, 46, 54, 60] or problems of TLS usage in specific areas like on mobile devices [47, 71, 83]. However, to the best of our knowledge, no prior work has examined the specific impact of security issues on protocol deployment.

In this paper, we conduct a large-scale longitudinal study examining the evolution of the TLS ecosystem since 2012 both on the client and on the server side. We analyze trends and evolution of the ecosystem, putting a special focus on changes occurring in response to specific high-profile attacks. For this, we use a combination of passive and active measurement data. Our passive measurements have been running continuously since February 2012 and currently contain protocol information about more than 319.3B TLS connections. The active measurement data provided to us by Censys [42] contains SSL and TLS scans of the entire IPv4 address space starting from August 2015.

To identify the patching behavior and evolution of specific clients we also create the—to our knowledge—largest TLS client fingerprint database to date, consisting of 1,684 fingerprints. These fingerprints

Measuring TLS in the Wild

Coming of Age: A Longitudinal Study of TLS Deployment

Internet Measurement Conference, 2018

Platon Kotzias
IMDEA Software Institute

Abbas Razaghpanah
Stony Brook University

Johanna Amann
ICSI/Corelight/LBNL

The Era of TLS 1.3: Measuring Deployment and Use with Active and Passive Methods

Arxiv, 2019

Ralph Holz[1], Johanna Amann[2], Abbas Razaghpanah[3], and Narseo Vallina-Rodriguez[4]
[1]The University of Sydney, [2]ICSI/Corelight/LBNL, [3]Stony Brook University,
[4]IMDEA Networks Institute/ICSI

1707.12762v2 [cs.CR] 6 Aug 2019

ABSTRACT

TLS 1.3 marks a significant departure from previous versions of the Transport Layer Security protocol (TLS). The new version offers a simplified protocol flow, more secure cryptographic primitives, and new features to improve performance, among other things. In this paper, we conduct the first study of TLS 1.3 deployment and use since its standardization by the IETF. We use active scans to measure deployment across more than 275M domains, including nearly 90M country-code top-level domains. We establish and investigate the critical contribution that hosting services and CDNs make to the fast, initial uptake of the protocol. We use passive monitoring at two positions on the globe to determine the degree to which users profit from the new protocol and establish the usage of its new features. Finally, we exploit data from a widely deployed measurement app in the Android ecosystem to analyze the use of TLS 1.3 in mobile networks and in mobile browsers. Our study shows that TLS 1.3 enjoys enormous support even in its early days, unprecedented for any TLS version. However, this is strongly related to very few global players pushing

It also supports a higher degree of privacy by encrypting as early as possible and improves performance by shortening the handshake.

Development of TLS 1.3 has been driven by major Internet corporations and organizations, in particular Google, Facebook, and Mozilla, who felt a need to address the security needs of their users and make their Web businesses faster to access across all devices. This drive is in line with previous contributions to TLS like Certificate Transparency, the HSTS header for HTTP, and the downgrade protection SCSV. Previous work has found evidence that the control that corporations like Google and Facebook exercise—they control both endpoints of a connection—leads to new security mechanisms being deployed faster [8].

In this paper, we analyze both the use and deployment of TLS 1.3 employing three different data sources: data from large-scale Internet scans, passive traffic observation in the Northern and Southern hemisphere, and data raised by a widely deployed application for the Android OS that can analyze TLS handshakes. Our primary contributions are as follows:

Deployment across DNS zones. We carry out large-

Measuring TLS in the Wild

arXiv:1907.12762v2 [cs.CR] 6 Aug 2019

The Era of TLS 1.3: Measuring Deployment and Use with Active and Passive Methods

Ralph Holz[1], Johanna Amann[2], Abbas Razaghpanah[3], and Narseo Vallina-Rodriguez[4]
[1]The University of Sydney, [2]ICSI/Corelight/LBNL, [3]Stony Brook University,
[4]IMDEA Networks Institute/ICSI

ABSTRACT

TLS 1.3 marks a significant departure from previous versions of the Transport Layer Security protocol (TLS). The new version offers a simplified protocol flow, more secure cryptographic primitives, and new features to improve performance, among other things. In this paper, we conduct the first study of TLS 1.3 deployment and use since its standardization by the IETF. We use active scans to measure deployment across more than 275M domains, including nearly 90M country-code top-level domains. We establish and investigate the critical contribution that hosting services and CDNs make to the fast, initial uptake of the protocol. We use passive monitoring at two positions on the globe to determine the degree to which users profit from the new protocol and establish the usage of its new features. Finally, we exploit data from a widely deployed measurement app in the Android ecosystem to analyze the use of TLS 1.3 in mobile networks and in mobile browsers. Our study shows that TLS 1.3 enjoys enormous support even in its early days, unprecedented for any TLS version. However, this is strongly related to very few global players pushing

It also supports a higher degree of privacy by encrypting as early as possible and improves performance by shortening the handshake.

Development of TLS 1.3 has been driven by major Internet corporations and organizations, in particular Google, Facebook, and Mozilla, who felt a need to address the security needs of their users and make their Web businesses faster to access across all devices. This drive is in line with previous contributions to TLS like Certificate Transparency, the HSTS header for HTTP, and the downgrade protection SCSV. Previous work has found evidence that the control that corporations like Google and Facebook exercise—they control both endpoints of a connection—leads to new security mechanisms being deployed faster [8].

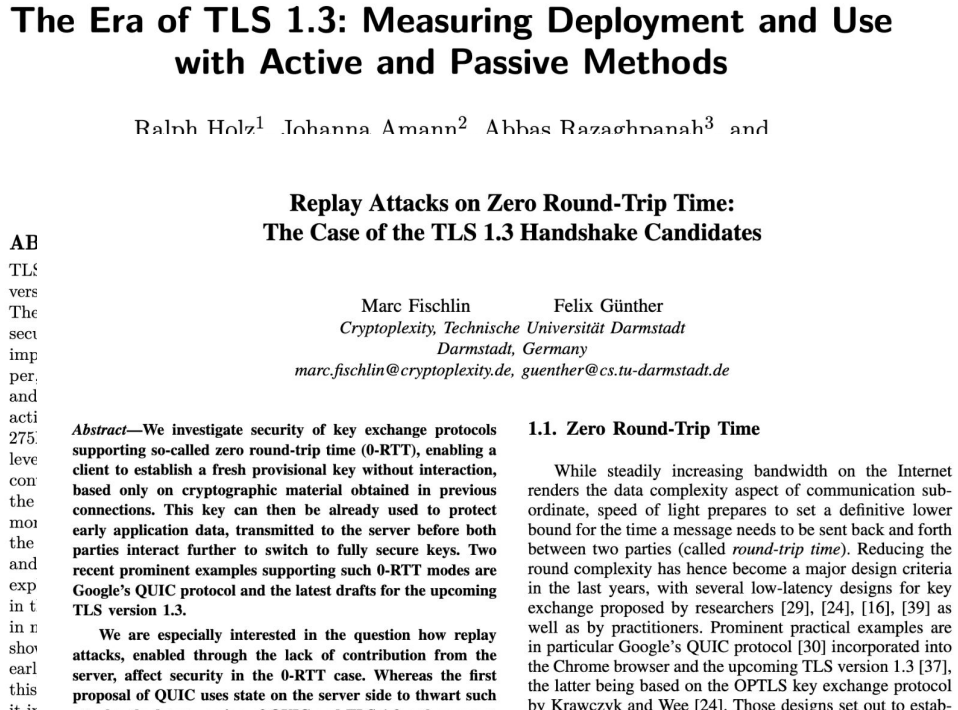
In this paper, we analyze both the use and deployment of TLS 1.3 employing three different data sources: data from large-scale Internet scans, passive traffic observation in the Northern and Southern hemisphere, and data raised by a widely deployed application for the Android OS that can analyze TLS handshakes. Our primary contributions are as follows:

Deployment across DNS zones We carry out large-

- 0-RTT acceptance was not measured
- PHA was not measured

- Interesting due to 0-RTT vulnerabilities
- Possible PHA vulnerability

Measuring TLS in the Wild



The Era of TLS 1.3: Measuring Deployment and Use with Active and Passive Methods

Ralph Holz[1] Johanna Amann[2] Abbas Razaghpanah[3] and

Replay Attacks on Zero Round-Trip Time: The Case of the TLS 1.3 Handshake Candidates

Marc Fischlin Felix Günther
Cryptoplexity, Technische Universität Darmstadt
Darmstadt, Germany
marc.fischlin@cryptoplexity.de, guenther@cs.tu-darmstadt.de

***Abstract*—We investigate security of key exchange protocols supporting so-called zero round-trip time (0-RTT), enabling a client to establish a fresh provisional key without interaction, based only on cryptographic material obtained in previous connections. This key can then be already used to protect early application data, transmitted to the server before both parties interact further to switch to fully secure keys. Two recent prominent examples supporting such 0-RTT modes are Google's QUIC protocol and the latest drafts for the upcoming TLS version 1.3.**

We are especially interested in the question how replay attacks, enabled through the lack of contribution from the server, affect security in the 0-RTT case. Whereas the first proposal of QUIC uses state on the server side to thwart such

1.1. Zero Round-Trip Time

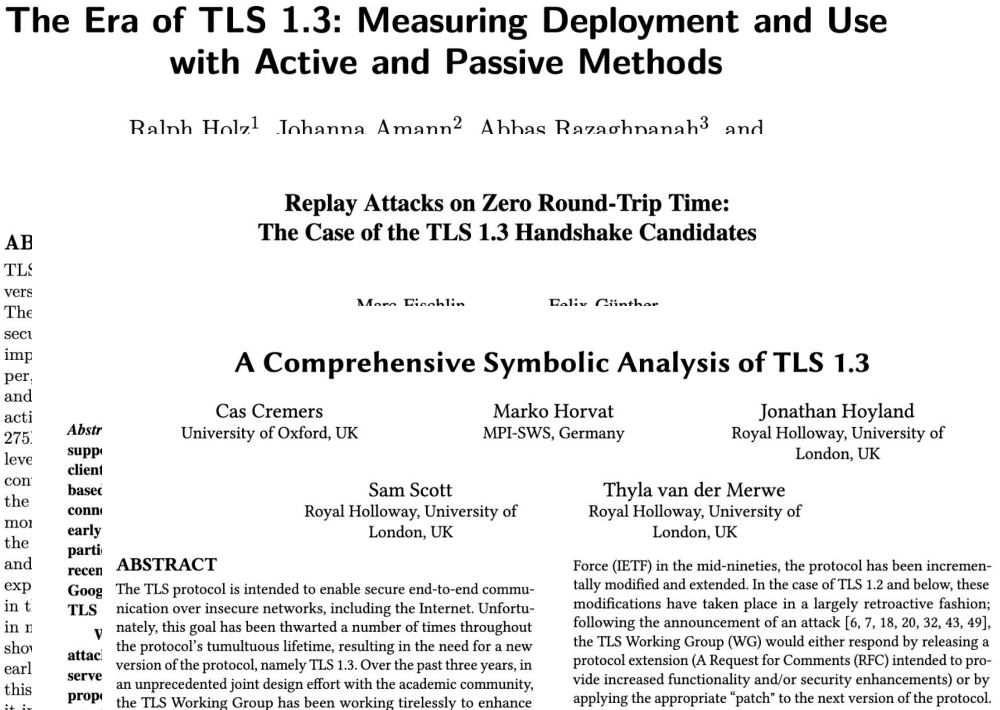
While steadily increasing bandwidth on the Internet renders the data complexity aspect of communication subordinate, speed of light prepares to set a definitive lower bound for the time a message needs to be sent back and forth between two parties (called *round-trip time*). Reducing the round complexity has hence become a major design criteria in the last years, with several low-latency designs for key exchange proposed by researchers [29], [24], [16], [39] as well as by practitioners. Prominent practical examples are in particular Google's QUIC protocol [30] incorporated into the Chrome browser and the upcoming TLS version 1.3 [37], the latter being based on the OPTLS key exchange protocol by Krawczyk and Wee [24]. Those designs set out to estab-

07.12762v2 [cs.CR] 6 Aug 2019

- 0-RTT acceptance was not measured
- PHA was not measured

- Interesting due to 0-RTT vulnerabilities
- Possible PHA vulnerability

Measuring TLS in the Wild



The Era of TLS 1.3: Measuring Deployment and Use with Active and Passive Methods

Ralph Holz[1] Johanna Amann[2] Abbas Razaghpanah[3] and

Replay Attacks on Zero Round-Trip Time: The Case of the TLS 1.3 Handshake Candidates

Marc Fischlin Felix Günther

A Comprehensive Symbolic Analysis of TLS 1.3

Cas Cremers
University of Oxford, UK

Marko Horvat
MPI-SWS, Germany

Jonathan Hoyland
Royal Holloway, University of London, UK

Sam Scott
Royal Holloway, University of London, UK

Thyla van der Merwe
Royal Holloway, University of London, UK

ABSTRACT

The TLS protocol is intended to enable secure end-to-end communication over insecure networks, including the Internet. Unfortunately, this goal has been thwarted a number of times throughout the protocol's tumultuous lifetime, resulting in the need for a new version of the protocol, namely TLS 1.3. Over the past three years, in an unprecedented joint design effort with the academic community, the TLS Working Group has been working tirelessly to enhance

Force (IETF) in the mid-nineties, the protocol has been incrementally modified and extended. In the case of TLS 1.2 and below, these modifications have taken place in a largely retroactive fashion; following the announcement of an attack [6, 7, 18, 20, 32, 43, 49], the TLS Working Group (WG) would either respond by releasing a protocol extension (A Request for Comments (RFC) intended to provide increased functionality and/or security enhancements) or by applying the appropriate "patch" to the next version of the protocol.

- 0-RTT acceptance was not measured
- PHA was not measured

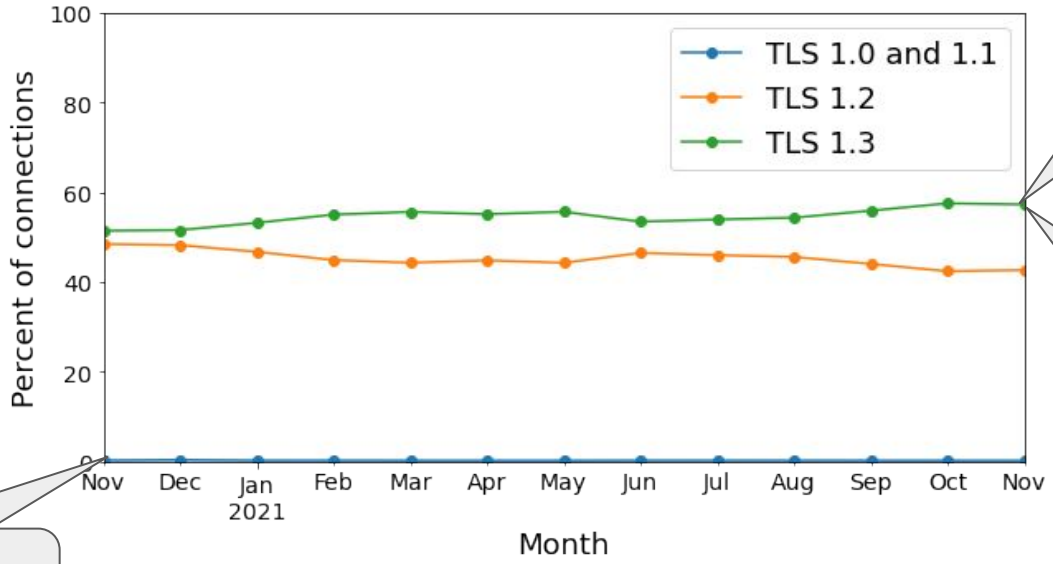
- Interesting due to 0-RTT vulnerabilities
- Possible PHA vulnerability

Measuring TLS in the Wild

- Counting usage numbers
- Internal data collection policy
- No personally identifiable data
- Opt-out option
- Data review
- All of our probes have expired

Probe Dictionary

Our Results: Version Usage



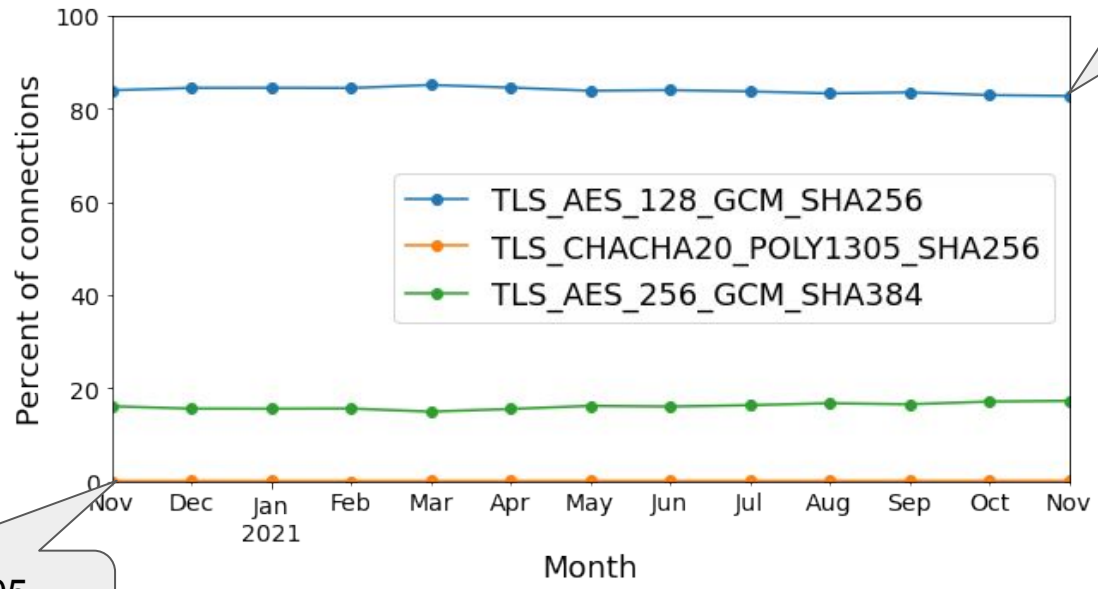
TLS 1.3 is the most used with 57.23%

Confirms insights of Holz et. al. with regards to TLS 1.3 growth

TLS 1.0 and 1.1 see less than 0.1% usage

TLS version usage in Firefox

Our Results: Cipher Suite Usage



AES 128 is the most used with 82.69%

ChaCha20 Poly1305 usage is less than 0.1%

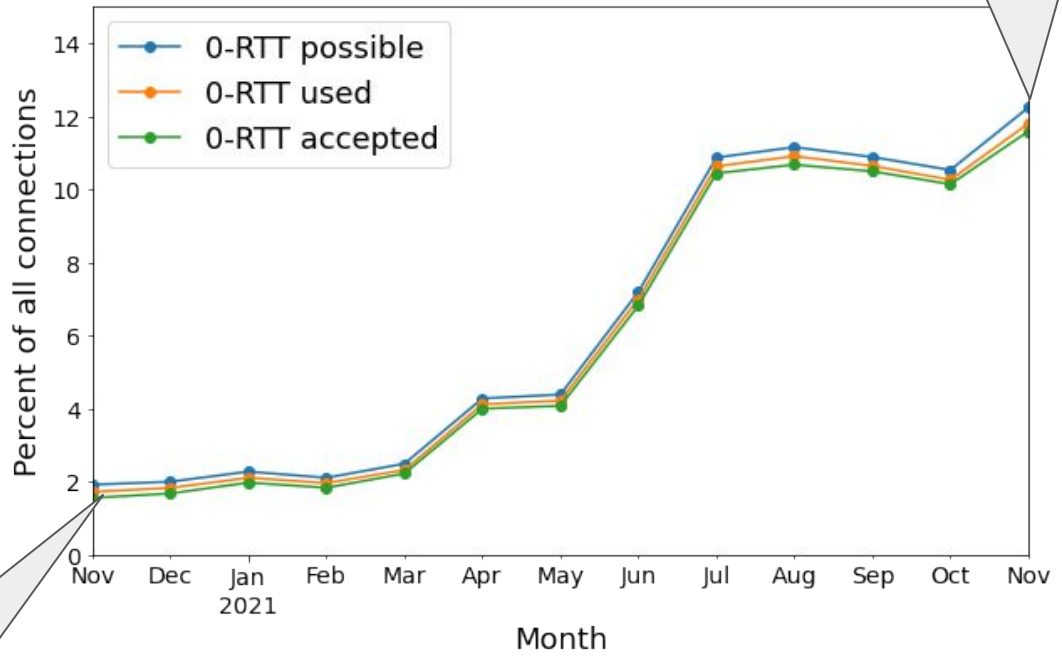
TLS 1.3 cipher suite usage in Firefox

Our Results: 0-RTT

possible:
0-RTT can be used for connection

used:
0-RTT is used for connection

accepted:
0-RTT was accepted by server

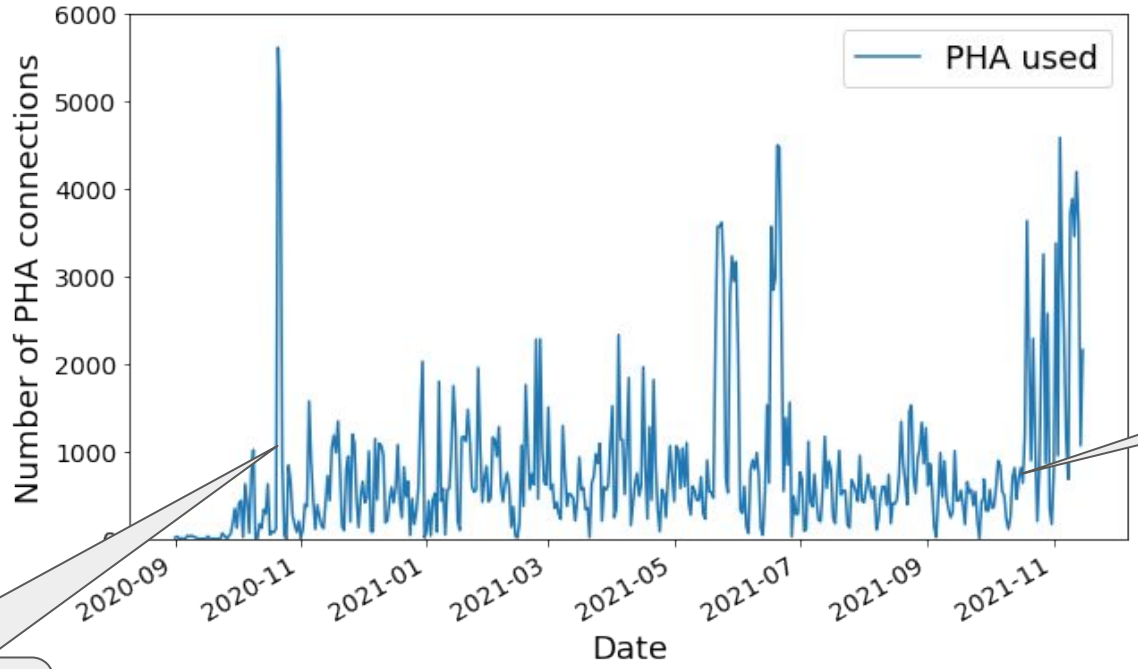


Big rise in 0-RTT possible to 11.82%

High acceptance rate of 98.20%

0-RTT usage in Firefox

Our Results: PHA



High variance between days

One big spike then lower usage

PHA usage in Firefox

Takeaways

- Built on previous studies to confirm previous insights
- 0-RTT usage is low but climbing
- PHA usage is very low
- New features not as robust so it is good to know their usage
- Maybe some of the less robust features are not needed going forward
- Future work: multi-vendor study would be interesting