The 2021 TLS Telemetry Report

The good, the bad, and the lingering questions

Authors

David Warburton is Principal Threat Research Evangelist with F5 Labs with over 20 years' experience in network and application security. A regular speaker at industry events and contributor to online and broadcast media, he is the author of F5 Labs publications including the annual 'Phishing and Fraud' and 'TLS Telemetry' reports. Recently he co-authored the SSL/TLS scanning DevOps tool 'Cryptonice' which helps organisations improve their website security posture. He was awarded Masters in Information Security from Royal Holloway University of London where his thesis was on the use of security and cryptography in IoT.

Sander Vinberg is a Threat Research Evangelist for F5 Labs. As the lead researcher on the Application Protection Research Series, he specializes in the evolution of the threat landscape over the long term. He holds a master's degree from the University of Washington in Information Management, as well as bachelor's degrees in History and African and African-American Studies from the University of Chicago.

Table of Contents

YEARS OF PROGRESS

2020

FEB
- Various Microsoft services, including Teams, are taken down due to expired certificate

MAR
- Bug causes Let's Encrypt to fail to properly check DNS CAA records
- Firefox rolls out DNS-over-HTTPS by default for users in the US
- Let's Encrypt announces that it has issued a billion certificates

APR

MAY
- Private key of DigiCert Certificate Transparency log compromised
- Chrome 83 and later will allow using of secure DNS using DNS-over-HTTPS

JUN
- GnuTLS fixes severe vulnerability which allows passive decryption of HTTPS traffic

JUL

AUG
- China's Great Firewall blocks TLS 1.3 and ESNI traffic

SEP
- Raccoon attack exposes timing attack in Diffie-Hellman spec in TLS 1.2 and older
- NCC finds TLS 1.3 MITM vulnerability in WolfSSL
- Amazon now supports TLS 1.3 on their CloudFront CDN platform
- Russia wants to ban ESNI, DoH, DoT and even TLS 1.3

OCT
- Chrome no longer loads mixed content (i.e. HTTP sources will not load when the page is accessed via HTTPS)
- NTS (a secure protocol for NTP) is published as RFC 8915

NOV
- ZeroSSL begins offering free 90-day certificates using the ACME protocol as an alternative to Let's Encrypt

DEC
- Mozilla is pre-loading intermediate certificates in to Firefox to prevent connection errors from badly configured HTTPS servers

2021

JAN
- Kazakhstan attempts again to perform interception of citizen's encrypted TLS sessions by asking them to install the governments root certificate. Mozilla, Apple, Google and Microsoft have blocked this certificate in their browsers
- Proposed standard for compressing certificates ispublished as RFC 8879

FEB
- Firefox distrusts Camerafirma's certificates for HTTPS cerfificates following a discovery of lapses in their security practises
- Apple's OCSP server was down causing delays in opening apps on Mac computers

MAR

APR
- Chrome now uses HTTP-first in addresses typed in to the Omnibox
- OpenSSL 1.1.1k released which fixes 2 high severity vulnerabilities
- IETF formally deprecates TLS 1.0 and TLS 1.1
- EFF announces end of the STARTTLS-Everywhere project

MAY

JUN
- All valid certs on the web have now been issues on or after May 1st 2018 which means that every single cert should be in the SCT logs

JUL
- ALPACA attack leverages wildcard certs protecting multiple services to launch XSS and other attacks
- Attacker compromise the Mongolian certificate authority MonPass. Client software was infected with the Cobalt Strike malware

AUG
- Firefox 91 used HTTPS by default in Private Browsing mode
- Digicert announces 'verified mark certificates' to allow company logos to appeart next to emails

SEP
- Number of certs used on the web exceeds 100 million with Let's Encrypt, Digicert and Sectigo accounting for 94% of the 100M certs
- Research presented at USENIX Security 21 found more than 40 vulnerabilities in STARTTLS implementations that could affect modern email clients

Executive Summary

Creating an encrypted HTTPS website depends on a lot more than simply throwing a digital certificate at it and hoping for the best. As old protocols prove to be insecure and new standards emerge, it has never been more important to keep HTTPS configurations up to date. In fact, Transport Layer Security (TLS) and HTTPS misconfigurations are now so commonplace that in the 2021 OWASP Top 10, Cryptographic Failures now comes in second place.[1]

As this report shows, the issue is not so much the lack of adopting new ciphers and security features but the rate at which old and vulnerable protocols are removed. Attackers know there is a correlation between poor HTTPS configurations and a vulnerable web server. Websites that routinely fail to follow TLS best practices are also found to be running old (and likely vulnerable) web servers.

On top of that is the potential use or abuse of web encryption for malicious purposes. Attackers have learned how to use TLS to their advantage in phishing campaigns, governments worldwide seek to subvert encryption to their benefit, and fingerprinting techniques raise questions about the prevalence of malware servers in the top one million sites on the web.

In order to collect the data for this report, we have continued to develop our own TLS scanning tool, Cryptonice, which is now free and open source. Security teams and website operators can use this to evaluate the cryptographic posture of their own sites and even bake it into their DevSecOps workflows for fully automated HTTPS auditing.

Here are some detailed stats on what's good, what's bad, and what's troubling in the world of TLS:

- TLS 1.3, now just over two years old, has risen to become the preferred protocol for 63 percent of the top one million web servers on the Internet. Support can vary drastically, however. In some countries, such as the United States and Canada, as many as 80 percent of web servers choose it, while in others, such as China and Israel, only 15 percent of servers support it.
- The move to elliptic curve cryptography is slow but steady, with 25 percent of certificates now signed with the Elliptic Curve Digital Signature Algorithm (ECDSA) and over 99 percent of servers choosing non-RSA handshakes when possible.
- Despite widespread TLS 1.3 adoption, old and vulnerable protocols are being left enabled. RSA handshakes are allowed by 52 percent of web servers, SSL v3 is enabled on 2 percent of sites, and 2.5 percent of certificates had expired.

- TLS 1.0 and 1.1 are now officially deprecated due to known security flaws. They have largely disappeared from use across the top one million sites, although a small number of web servers, 0.4 percent, still select one of them during an HTTPS connection.
- Encryption continues to be abused. The proportion of phishing sites using HTTPS and valid certificates has risen to 83 percent, with roughly 80 percent of malicious sites coming from just 3.8 percent of the hosting providers.
- Recent research has found active SSLStrip attacks successfully stealing user logon credentials, indicating the growing need for using HTTP Strict Transport Security (HSTS) headers or completely disabling HTTP services.
- Certificate revocation methods are almost entirely broken, driving a growing desire across the certificate authority (CA) and browser industries to move toward extremely short-term certificates.
- TLS fingerprinting shows that 531 servers in the top one million potentially matched the identity of Trickbot malware servers, and 1,164 matched Dridex servers.

By comparing themselves with the top one million sites, security teams can perform a gap analysis of their own web servers to determine areas of improvement to prioritize. We've also included relevant stories from the past 18 months to illustrate how lapses in TLS can have very real-world consequences.

Introduction

It is now Autumn 2021, which means that eighteen months have passed since F5 Labs last revisited encryption—everyone's favorite dusty corner of Internet infrastructure. Even though encryption can feel like a "solved problem," the devil is still in the details after all these years, and it remains possible to mess this up, solved problem or not. As a result, we analyze how successful the Internet's busiest properties have been at implementing the known best practices around HTTPS and TLS. This report presents those findings and our assessments of devilish encryption details that still need attention in too many places.

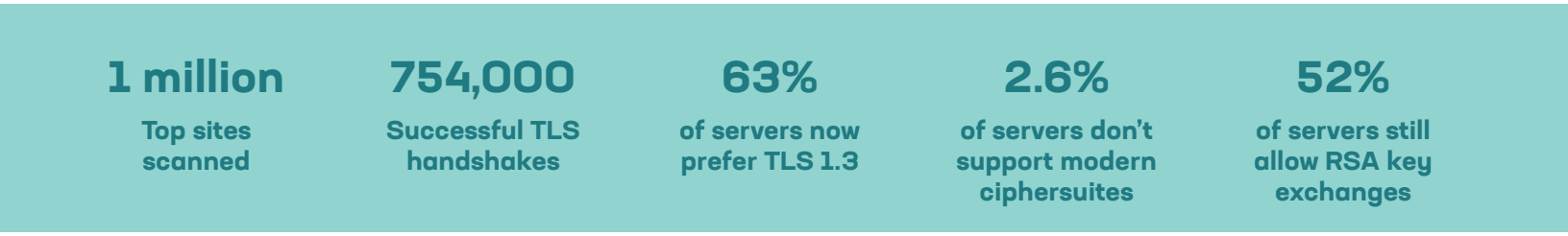


Figure 1
A snapshot of key TLS telemetry findings

Report structure

We start with the good news: progress we've seen toward everyone reaching a minimum level of security. Then we talk about the bad news: stagnation or even regression in encryption practices. Finally, we turn to the ugly side of encryption—how it is being subverted by organized crime and how governments around the world look to weaken or even ban encryption entirely.

Methodology

The majority of data in this study comes from our scans of sites found in the Tranco Top 1 Million list.[2] For the 2019 TLS Telemetry Report[3], we developed the free and open-source tool Cryptonice. Over the past 18 months, we've continued to develop and expand the capabilities of this tool to help us capture even more relevant data from Internet-wide probing. We perform scans of the top 1 million sites once per quarter and average results for any given 12-month period. Rather than scanning every IP address, we focus on the most popular websites on the web, since doing so allows us to perform more accurate scans of web configuration. It also helps us provide insight into differences between various industries. We also look at phishing sites as reported by OpenPhish and use their data to investigate which sites are using encryption and which industries are most targeted. Finally, we supplement our findings with client (browser) data captured by Shape Security to clearly understand the most frequently used browsers and bots.

A further note on methodology and our data: Unfortunately, not every address always resolves, which means that some domains on the list didn't supply any information, and occasionally the scanner was unable to establish a TLS connection. The possible causes for that lack of connection include server timeouts, unavailability of HTTPS, or temporary DNS resolution problems. When Cryptonice targets a domain, it follows redirects as best it can to obtain the HTTPS configuration a user would receive if they visited that same site with a web browser. For example, targeting microsoft.com will take Cryptonice to www.microsoft.com/en-gb. As a result of various connection issues, despite an initial list of 1 million domains, the final number of sites that provided information about TLS configurations was consistently around 754,000 per scan. Figures in this report that present percentages of totals represent proportions of this 754,000 total unless otherwise specified. Broadly speaking, we were able to collect information from roughly 82 to 87% of the top 100,000 sites. Beyond the top 100,000 sites, around 75% of servers responded.

The Good News

The Good News

You're supposed to start with a compliment, right? Let's start with the good news revealed by our research. There's plenty to discuss, from the evolution away from old protocols to more secure certificate management.

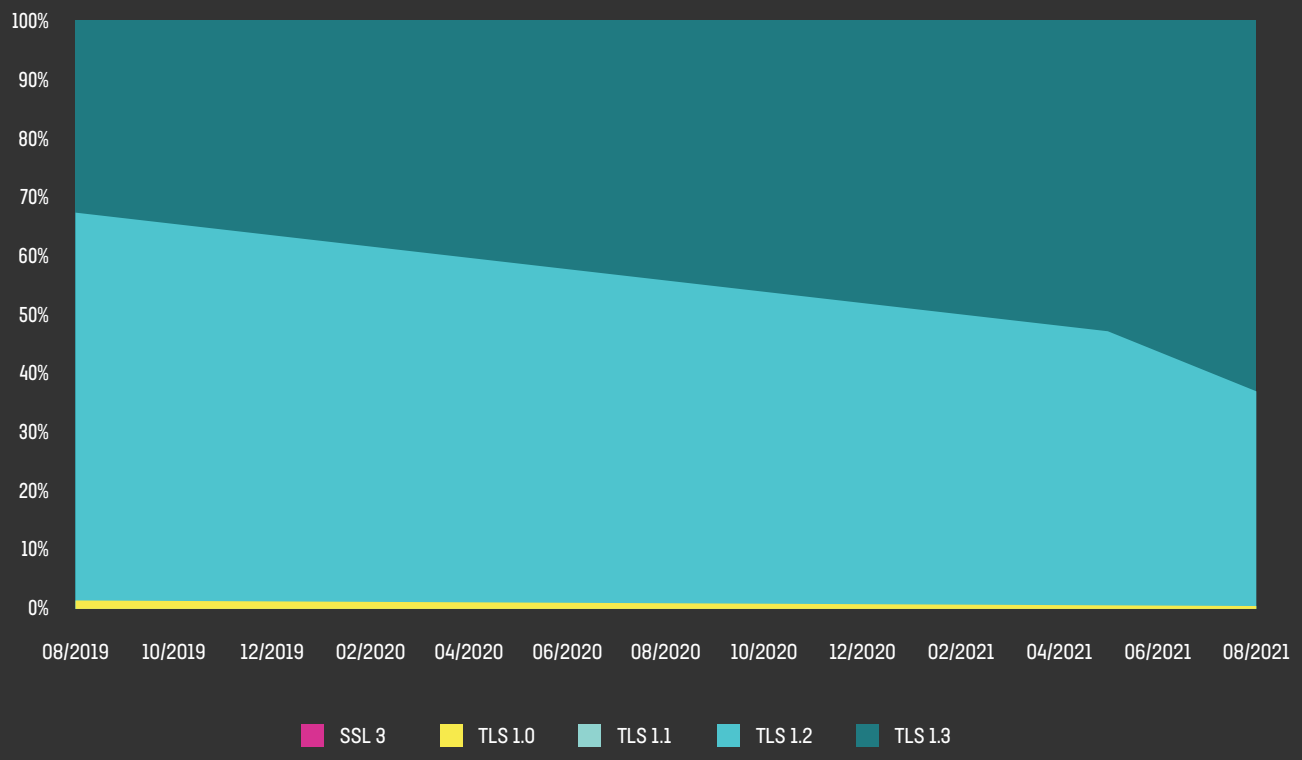
The Shift to TLS 1.3

The very good news is that for the first time, TLS 1.3 is the chosen encryption protocol for the majority of web servers among the top million (Figure 2). While TLS 1.3 has been gradually growing in prevalence, two years ago only 32% of servers defaulted to TLS 1.3, and it only climbed to the number one spot in May 2021. The protocol has seen big jumps in popularity following its adoption by large hosting and CDN providers such as Amazon Cloudfront. Almost 63% of servers prefer TLS 1.3 to other protocols as of August 2021.

Of those sites supporting TLS 1.3, the proportion using the "early data" capability—which allows the server to save time by proactively pushing data to the client—grew from 8.4% in 2019 to 9.2% in 2021.[4]

FIGURE 2: CHOSEN PROTOCOL

Selected SSL and TLS protocols by servers in the top million



The IETF officially deprecated TLS 1.0 and TLS 1.1 in March 2021.[5] Despite this, SSL 3.0 and TLS 1.0 are still the preferred protocols for a small number of sites, as is barely perceptible in Figure 2. TLS 1.0 is preferred by 0.4% of sites, while SSL 3 preference accounts for just 0.002%. On the client side, data from Shape Security show that Chrome is by far the most prevalent browser. At the time of data collection, Chrome 91 was used by almost 34% of connections, with Chrome 90 accounted for 6.5% of connections. Versions of Mobile Safari were in second and third place with a combined total of 23.5%. In total, well over 95% of all browsers in active use support TLS 1.3.

Simply looking at the preferred protocol a server selects for TLS handshakes does not reveal the whole story, however. Support for older, deprecated protocols continues unabated across the entire range of sites (Figure 3). We found no relationship between the amount of traffic a site receives and the protocols it supports. In other words, more popular sites aren't necessarily stricter when it comes to offering TLS protocols. In fact, the top 100 sites were more likely to still support SSL 3, TLS 1.0, and TLS 1.1 than servers with much less traffic.

FIGURE 3: PERCENTAGE OF SITES OFFERING SSL/TLS PROTOCOLS

The availability of SSL and TLS protocols across the top million sites

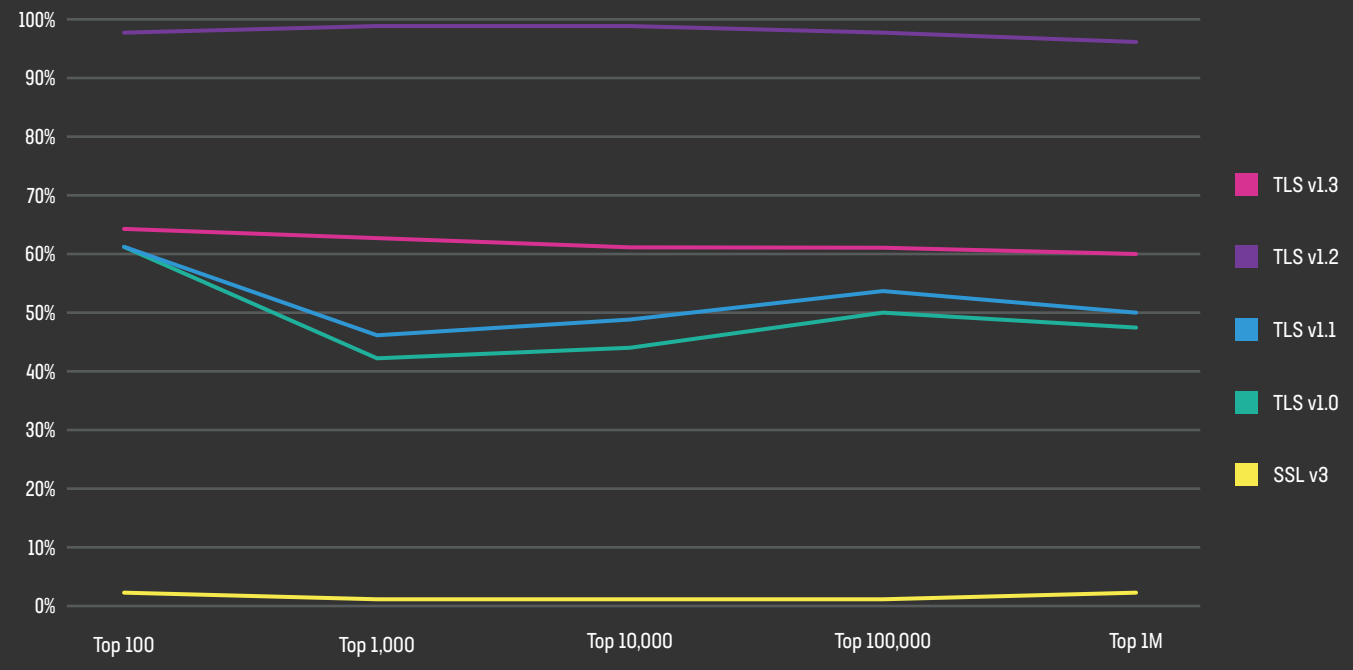
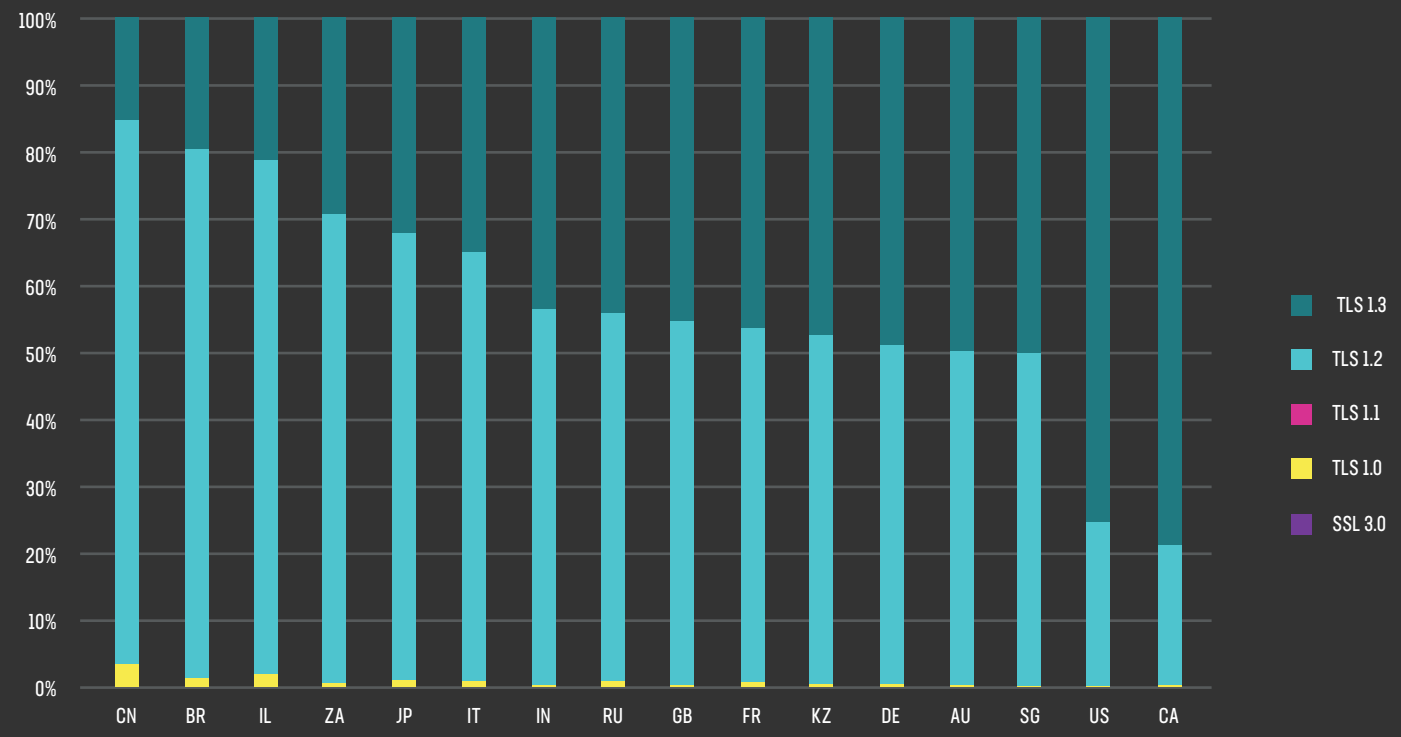


FIGURE 4: DISTRIBUTION OF SSL/TLS PROTOCOLS BY COUNTRY

The distribution of preferred SSL/TLS protocol for selected countries



In the 2019 report, we analyzed servers' preferred TLS protocols by the country in which they were based, using country code top level domains (ccTLDs) such as *.co.uk or *.jp. In 2019, the TLDs with a greater prevalence of TLS 1.3 than the .com TLD were .co for Colombia and .id for Indonesia. At the other end of the spectrum, .cn (China) and .jp (Japan) had the lowest prevalence of TLS 1.3. While Chinese domains were marginally better than Japanese domains in supporting TLS 1.3, they also supported TLS 1.0 at a higher rate.

For this report, we supplanted TLD analysis in favor of geolocation lookups on the servers in question. This had the advantage of allowing us to reveal the true location of the servers, not their putative registration information.[6] Figure 4 shows the distribution of SSL/TLS protocols by country. Canada and the United States are significantly ahead of the pack in TLS 1.3 preference, with Canadian servers in particular preferring TLS 1.3 nearly 80% of the time. At the other end of the spectrum, Chinese servers show little support for TLS 1.3 and also had the highest occurrence of TLS 1.0 protocols.

Simplifying Cipher Suites

When we switched attention from protocols to cipher suites, the scans revealed that sites in the top million offered, on average, nearly 20 separate cipher suites. This illustrates one reason TLS 1.3 is such an important step forward: It simplifies which suites are available. While we found all five TLS 1.3 cipher suites available across the top million sites, our scans found only three being chosen by servers during TLS handshakes: TLS_AES_256_GCM_SHA384, TLS_AES_128_GCM_SHA256, and TLS_CHACHA20_POLY1305_SHA256.

It's common to refer to the cryptography a server and client agree to use as a cipher, although this isn't quite accurate. A cipher is the cryptographic algorithm which, when combined with a secret key, scrambles and protects some data. But to send encrypted data over the insecure Internet we need more than a cipher. The client and server must agree:

- How to safely exchange keys (the key agreement).
- How the keys will be authenticated (signed).
- How to use the cipher (the mode of operation).

Finally, they also need to identify which hashing algorithm to use to ensure encrypted messages have not been tampered with. All of these factors combined represent the suite of agreements known as a cipher suite. (See Figure 5.)

FIGURE 5: WHAT'S IN A CIPHERSUITE

A breakdown of the components that combine to form a cipher suite

TABLE 1: THE MOST POPULAR SELECTED CIPHER SUITES IN THE TOP MILLION SITES

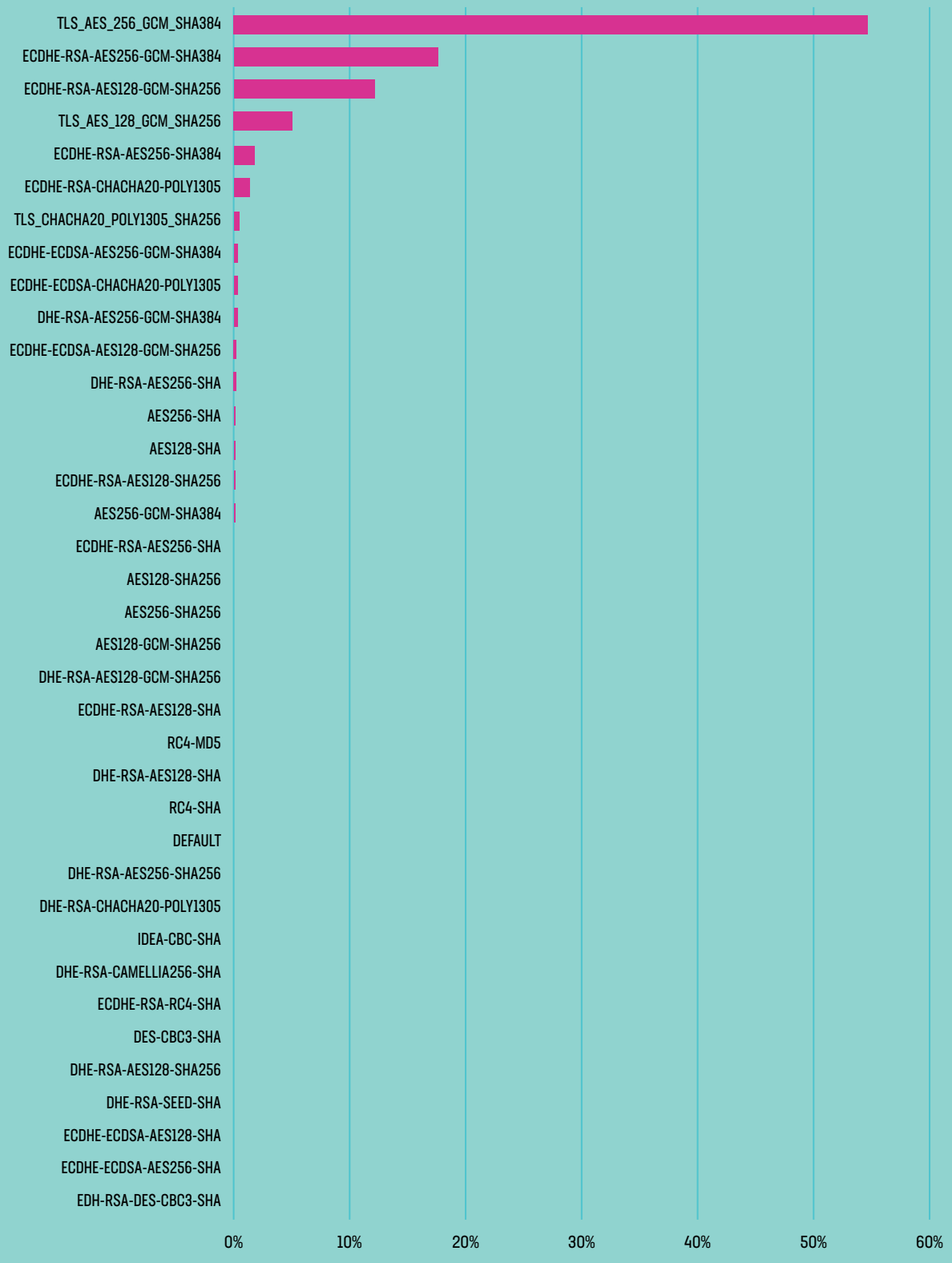
Protocol	Cipher suite chosen by web server	Proportion of top 1M sites
TLS 1.3	TLS_AES_256_GCM_SHA384	56.8%
TLS 1.2	ECDHE-RSA-AES256-GCM-SHA384	18.4%
TLS 1.2	ECDHE-RSA-AES128-GCM-SHA256	12.6%
TLS 1.3	TLS_AES_128_GCM_SHA256	5.4%
TLS 1.2	ECDHE-RSA-AES256-SHA384	1.9%
TLS 1.2	ECDHE-RSA-CHACHA20-POLY1305	1.4%
TLS 1.3	TLS_CHACHA20_POLY1305_SHA256	0.5%
TLS 1.2	ECDHE-ECDSA-AES256-GCM-SHA384	0.4%
TLS 1.2	ECDHE-ECDSA-CHACHA20-POLY1305	0.4%
TLS 1.2	DHE-RSA-AES256-GCM-SHA384	0.4%
TLS 1.2	ECDHE-ECDSA-AES128-GCM-SHA256	0.3%
TLS 1.0	DHE-RSA-AES256-SHA	0.3%

Table 1 shows how just over three-quarters of every TLS connection across the top million sites make use of AES (with a 256-bit key) and the SHA-2 hashing algorithm (with a 384-bit output size). This is true for servers that support TLS 1.3 and those which still use TLS 1.2.

The scan also revealed 58 distinct TLS 1.2 cipher suites available across all servers in the top million. If the rate at which SSL 3 is dying is anything to go by, it will be years before we all benefit from the simplification TLS 1.3 was partially designed to bring. However, as Figure 6 shows, most of those 58 cipher suites were infrequent and made up the long tail, with 84% of servers running TLS 1.2 preferring just one of two cipher suites.

FIGURE 6: CHOSEN CIPHERS FROM TOP 1 MILLION SITES

Preferred cipher suites of servers in the top million



The Shift to Elliptic Curve Crypto

To further improve the security of encrypted communications, most TLS implementations implement forward secrecy, which is achieved by the use of ephemeral (one-time use) keys with elliptic curve cryptography. The final E in the abbreviation DHE and ECDHE indicates the use of ephemeral keys. (See Figure 6).

99.3% **of sites prefer non-RSA key agreements**

TLS 1.3 removes the risk of using RSA key exchange, since it only permits ECDHE key agreements. Between the widespread use of TLS 1.3 and older protocols configured to prefer non-RSA key exchanges, almost every site—99.3% in the top million—chooses not to use RSA to exchange keys during the TLS handshake. This is comforting to see.

The move away from RSA-based certificates to elliptic curve cryptography (ECC) variants has been slower, since RSA is still believed to be a secure way of cryptographically signing data. Nevertheless, ECC certificates using the elliptic curve digital signature algorithm (ECDSA) are increasing. Just over 24% of top sites make use of 256-bit ECDSA certificates, while around 1% use 384-bit ECDSA certificates.

75% **of certificates use traditional RSA signatures**

25% **of certificates use ECDSA elliptical curve signatures**

Decreasing Certificate Lifespans

Regardless of the type of certificate in use, certificate revocation methods are almost entirely broken. That's why desire is growing across the certificate authorities (CAs) and browser industry to move toward extremely short-term certificates. Revoking a stolen certificate becomes much less of an issue if it will expire in just a few weeks.

After repeated failed ballots in the CA/Browser Forum to cap the maximum certificate lifespan at one year, the major browsers instead chose to enforce a 398-day limit. The maximum lifespan of new certificates issued after September 2020 has dropped significantly from three years to only 398 days.[7] The ACME protocol, used to automate requests and issuance of free certificates, defines a maximum certificate age of 90 days. The popularity of these short-term certificates is revealed in the data: The single most common certificate— accounting for 38% of the total—has a lifespan of 90 days. Figure 8 combines 90-day certificates with those accidentally issued for 91 days, bringing the total for 90-91 day certs to 42%.

Short lifespan certificates also help partially solve the issue of changes in domain name ownership. Should the owner of a domain change, administrators must ensure the old owner can't continue using their certificate to perform active attacks and decrypt traffic intended for the new owner. In 2018, security researchers showed pre-existing certificates for 1.5 million domains (0.45% of the Internet), a problem they dubbed BygoneSSL.[8] Some 25% of those pre-existing certificates had not expired.[9]

FIGURE 7: DISTRIBUTION OF CERTIFICATE TYPES AND KEY SIZES

The distribution of certificate types and key sizes for servers in the top million

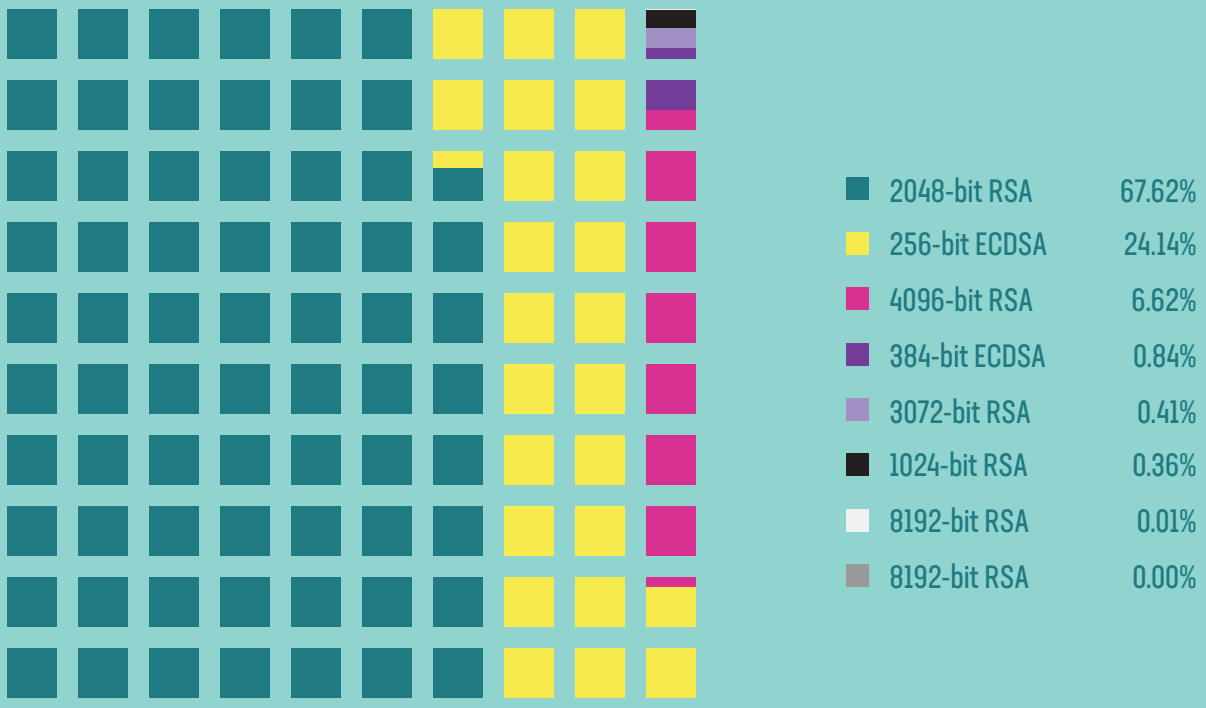
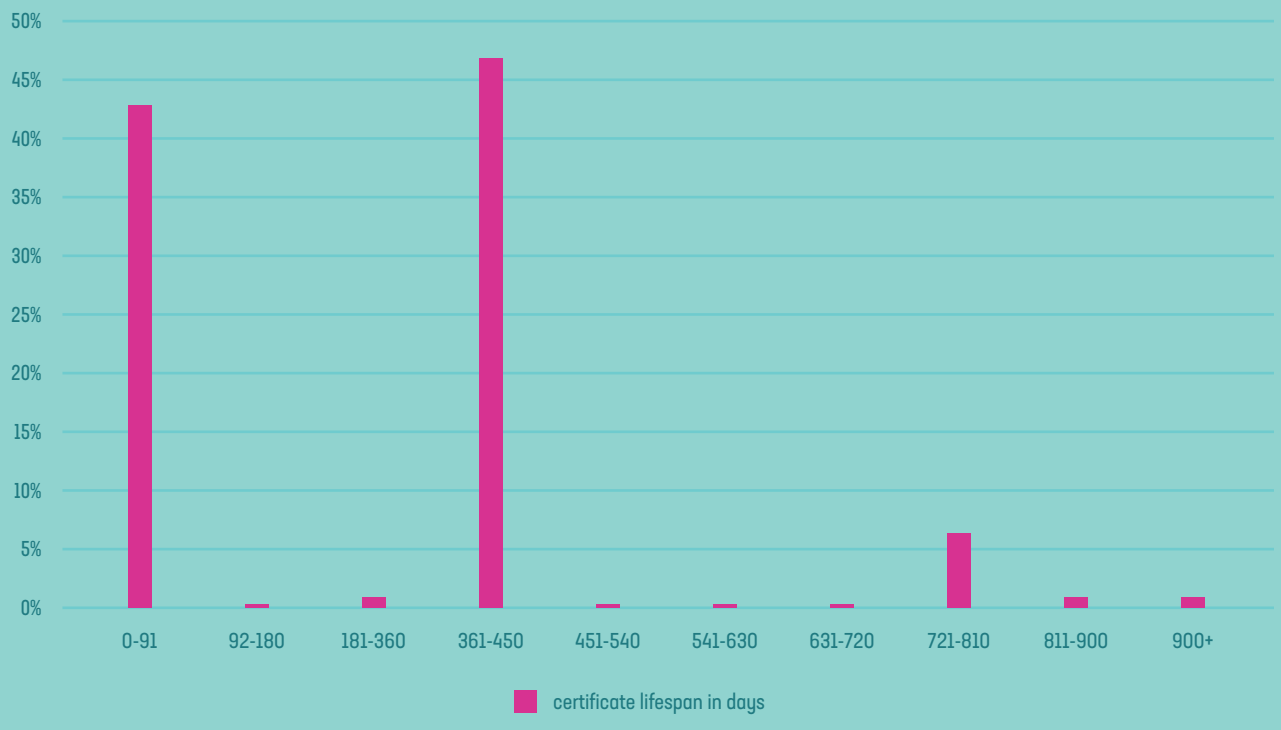


FIGURE 8: CERTIFICATE LIFESPANS

The frequency of certificate lifespans in the top million sites



This is the end... (of some protocols)

Over the years, the industry has developed new supporting protocols designed to help close the gaps a basic TLS configuration can leave open. Some protocols are becoming well established standards, such as DNS CAA records and the HTTP Strict Transport Security (HSTS) header. Not all new security protocols survive, however, and while their deaths are rarely to be celebrated, in many instances something much better takes their place.

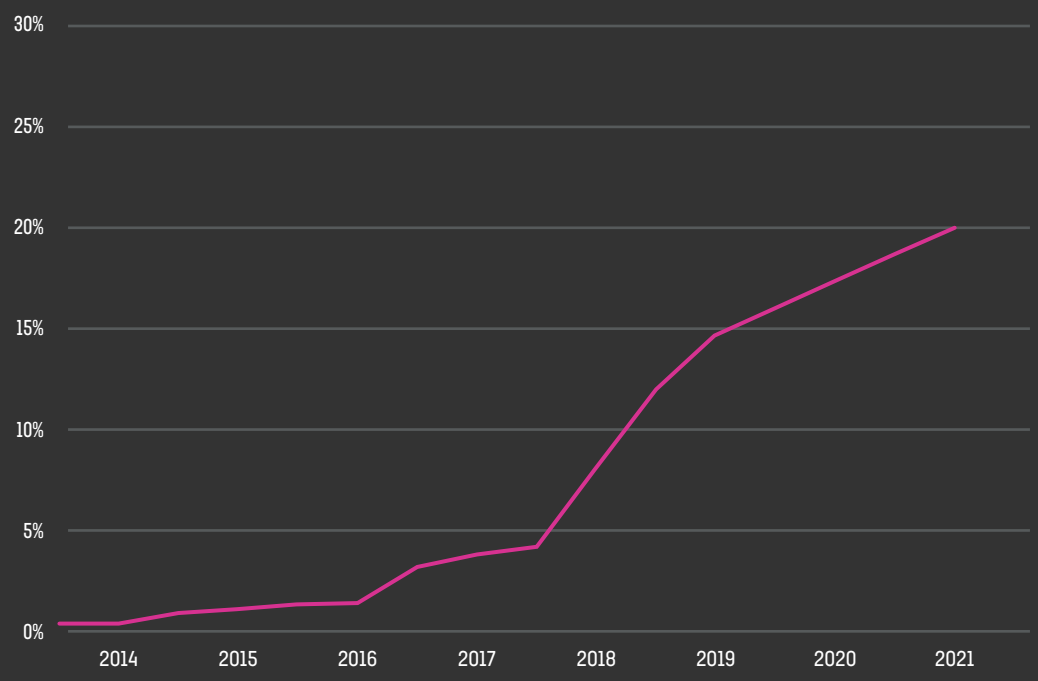
HTTP Strict-Transport-Security

Our scans showed slowed growth of the HSTS response header. At the time of our 2019 report, roughly 15% of sites used it, and in mid-2021, about 20% of scanned sites returned an HSTS header (Figure 9). A number of factors could account for the reduced growth rate. For instance, the majority of website owners who understand HSTS may have already configured it. Perhaps the others simply haven't heard of it or don't believe it's necessary.

Major browsers are beginning to default to HTTPS, however, and web browsers could conceivably disable HTTP altogether within a few years. Such a move to an HTTPS-only web would certainly negate the need for the HSTS header.

FIGURE 9: PREVELENCE OF HSTS HEADER ACROSS TOP 1 MILLION SITES

The use of HSTS across the top million sites



HTTP Public Key Pinning

Our scans also showed that use of HTTP Public Key Pinning (HPKP), the deprecated mechanism in which the server delivers hashes of public keys that must match keys in the certificate chain for future connections, has largely if not completely died out. Only 0.06% of servers in our 2021 scans had HPKP configured, although this is up slightly from our 0.05% finding 18 months ago.

Extended Validation Certificates

Sites can pay to undergo enhanced vetting to ensure the domain, the site, and the organization purporting to run it on that domain all match. The outcome of that additional vetting is an extended validation (EV) certificate. EV certificates don't provide additional technical security; they are intended to help signal to users that a certificate is trustworthy. However, their efficacy is marginal. Most users don't know what they are or how to check for them and don't miss them when they aren't present. Furthermore, research for our 2020 Phishing and Fraud Report found that many phishing sites are hosted on well-known blogging platforms that used EV certs.[10] Cryptonice scans of the top million sites revealed that only 1.8% of web servers use EV certs, down from the 2.2% of servers with EV certs noted in our last report.

Overall, this year's research documented continued, if sometimes slow, progress toward more sophisticated (but higher-performance) security tactics. It also captured evidence that organizations—at least those responsible for the top million sites—are gradually weeding out older, less secure protocols, practices, and certificate management. Older, less secure protocols and cryptography still exist, but mostly in a very small percentage of sites.

Of course, that doesn't make them all right, and our 2021 research also revealed problems that need to be addressed.

The Bad News

The Bad News

It wouldn't be a security report without some bad news, right? The most glaring sign that all is not right within the world of web encryption comes not from our scan data but from OWASP. In the latest OWASP Top 10 ranking of security issues, cryptographic failures (A02:2021) moved up from third position to second position, behind broken access control. This increasingly frequent failure, previously known as sensitive data exposure, was also renamed to emphasize the technical root cause rather than symptoms or outcomes.[11] More than anything else, its growth illustrates why something that can appear as dry and abstruse as cryptography still requires everyone's attention: we simply aren't doing it well enough to ignore it. With that said, Cryptonice also revealed interesting (and troubling) findings that help supply details for understanding why OWASP made the change they did.

To put these findings in context a TLS configuration using less-than-great encryption is not an automatic disaster. It's virtually unheard of for organized cybercriminals to use TLS weaknesses to attack an organization. (Nation states are a different matter.) So it's unlikely that an organization's ongoing support for TLS 1.0, HTTPS misconfigurations, or failing to follow TLS best practices will be responsible for a breach. But such TLS weaknesses may indicate to attackers that the rest of the web server is very likely out of date and therefore vulnerable.

CAs behaving badly

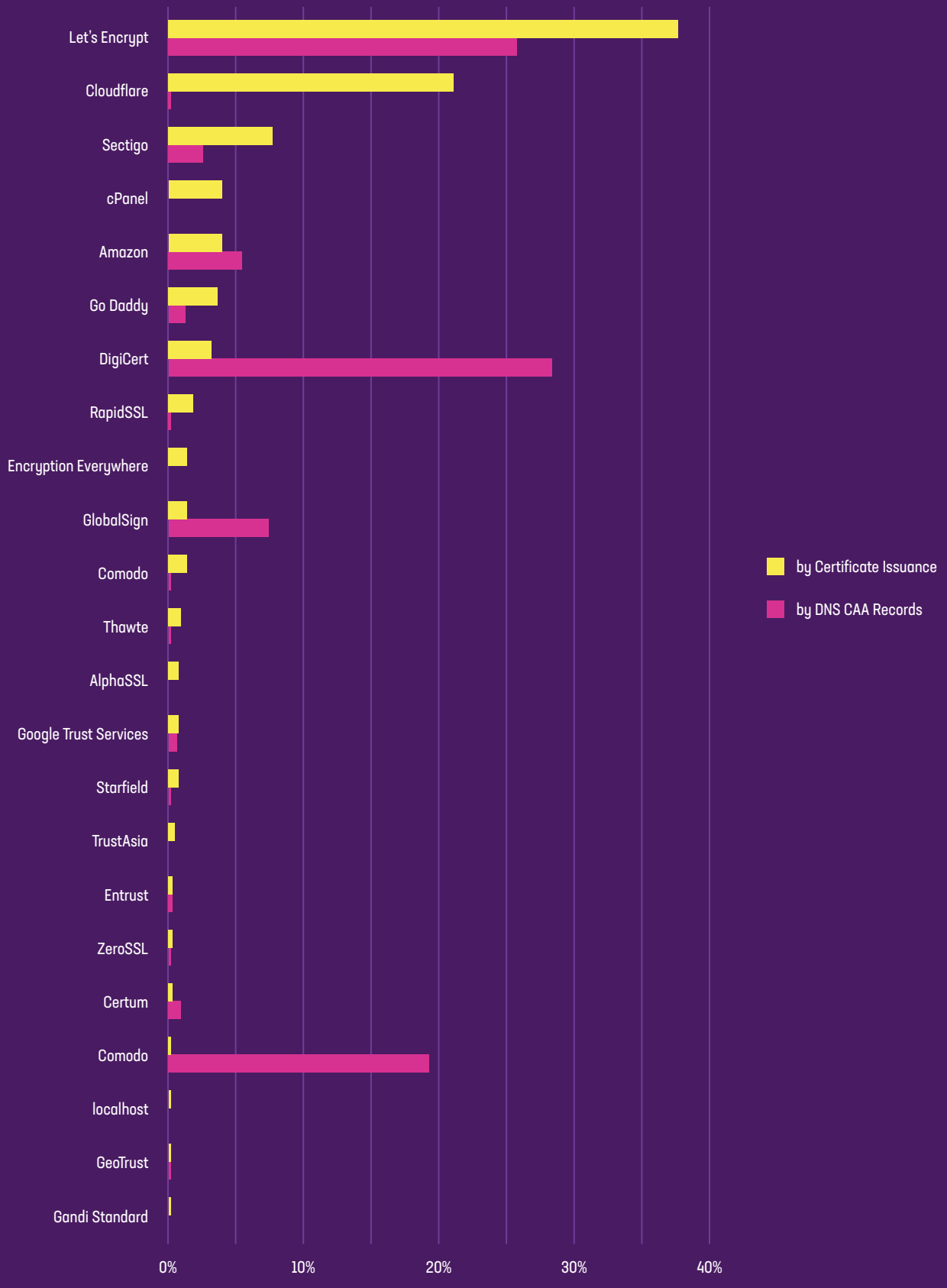
Let's start with the problems that primarily result from malicious or negligent behaviour by certificate authorities. The web depends on a chain of trust, and that chain is anchored with the certificate authorities. There are hundreds of CAs that web browsers inherently trust, despite many of them being completely unknown to everyday users of the web. So it's imperative that the CAs are completely trustworthy and held accountable for their actions.

Ripped from the Headlines, Part I

In early 2021, after several years of back and forth with Camerfirma, Mozilla made the decision to stop trusting HTTPS certificates issued by the Portuguese certificate authority. It appears Camerfirma intended to discontinue their work as a certificate authority and had neither remediated a long list of security issues nor complied with Mozilla's requests for transparency on those issues.[12] The implication? Not all certificate authorities are dependable, and some, if not all, treat their work as a line of business, not something akin to a public utility.

FIGURE 10: TOP CERTIFICATE AUTHORITIES

Comparing top CAs by the number of certificates issued to the proportion of defined DNS CAA records



If a web server is still using a certificate signed by a CA whose certificate has been removed from a browser's root store, then TLS handshakes will fail. The most recently distrusted CA, Camerfirma, accounts for four certificates found in the top million sites. Another 114 are still signed by Symantec, WoSign and StartCom, other recently distrusted CAs. All are untrusted and will result in a failed connection.

One of the best ways to protect against misbehaving CAs is to create DNS certificate authority authorization (CAA) records. In such records, site owners maintain information about which CAs are permitted to issue certificates for that domain.

Figure 10 shows the most popular certificate authorities as measured both by the number of certificates they issue and their prevalence within the CAA records. While Let's Encrypt appeared marginally less often in the CAA records than DigiCert, Let's Encrypt was still by far the busiest CA in terms of issuing certificates; nearly 40% of the web's certificates currently come from them. Only Cloudflare, at 21%, even comes close, with Sectigo third at 8%.

DNS CAA records grew in popularity between 2019 and 2021. Of the top million sites, 3.5% are now using CAA, up from 1.8% a few years ago. Despite this positive and steady increase, this finding demonstrates how few sites still actually use them. The average number of CAs defined in CAA records was 5.8, with some sites defining up to 20.

Ripped from the Headlines, Part II

On February 29, 2020, Let's Encrypt identified a business logic issue in the code that runs their certificate authority authorization process.[13] The code in question was intended to allow their customers a grace period after ensuring Let's Encrypt had authorization to issue a certificate to subscribers. In practice, it created the possibility of a certificate mismatch if Let's Encrypt lost authorization during the grace period. Issues like this are a good reminder not to take public key cryptography for granted.

When ALPACAs Attack

Although there have been a number of worrying Man-in-the-Middle (MITM) vulnerabilities discovered in TLS implementations recently (see GnuTLS CVE-2020-13777[14] and WolfSSL CVE-2020-24613[15]) the protocol itself is rarely found to have serious problems.

This is what makes the ALPACA attack so interesting.[16] We are, of course, refer to the Application Layer Protocol Confusion-Analyzing and Mitigating Cracks (ALPACA attack), not the furry llama-like animal, which also would be interesting. This attack takes advantage of wildcard certificates and exposes a vulnerability even when TLS is working correctly. TLS is an application-independent protocol, which is to say that TLS can be used to secure HTTP, SMTP and many other applications. There is nothing within a TLS handshake which defines and restricts the secure session to a specific application, and the ALPACA attack takes advantage of this.

ALPACA is an application layer protocol content confusion attack, exploiting TLS servers implementing different protocols but using compatible certificates, such as multi-domain or wildcard certificates. Attackers can redirect traffic from one subdomain to another, resulting in a valid TLS session. This breaks the authentication of TLS, and cross-protocol attacks may be possible where the behavior of one protocol service may compromise the other at the application layer.

Source 1 Explanation of ALPACA attack from alpaca-attack.com

While the vulnerability is not trivial to exploit, the researchers were keen to point out that they were only able to test some specific scenarios. Others may exist. The US National Security Agency (NSA) deems this issue severe enough to have recently issued guidance around it.[17]

In our scans of the top 1M sites we found over 40% of sites make use of certificates which contain wildcards, e.g., *.example.com. More specifically, we found over 6.5% of web servers use certificates which are also used with mail servers as they contained `mail.` or `smtp.` in the `SubjectAltName` field.

The NSA guidance document suggests that administrators carefully consider the use of wildcard certificates and make use of web application firewalls, even for non-HTTP servers.[18]

Fat fingers and dusty configurations

Misconfigurations and outdated configurations represent a significant source of TLS issues. Since HTTPS is contingent on several components, this section of this report starts with outdated encryption protocols and ciphers, moves on to weak or expired certificates, and finally touches on invalid certificate chains and the transmission of trust.

Legacy protocols and weak ciphers

We've seen plenty of good progress made with the relatively rapid adoption of TLS 1.3 and move to elliptic curve cryptography. But as much as new protocols can improve a website's security, its legacy protocols, frequently left enabled, are a much bigger factor in overall cryptographic strength. Neglecting to turn off SSL 3 leaves web servers vulnerable to the POODLE attack; some TLS 1.0 and TLS 1.1 configurations may also be vulnerable to POODLE v2.[19] Offering RSA key exchanges or allowing export-grade cipher suites can enable attacks on the supposedly secure TLS connection.

While cryptographers believe that using RSA to sign digital certificates is still secure, exchanging keys over the Internet using RSA has proven vulnerable to attack time[20] and time[21] (and time[22]) again. For years cryptographers and the wider security community have called for an end to RSA as a key exchange method recommending instead, the use of Diffie-Hellman (DH) key agreement. While DH, like RSA, may be used with large prime numbers, in current practice DH is more often used with elliptic curves, which results in much smaller keys. These smaller keys have the same security as larger, prime-based crypto and they are computationally far more efficient.

52% **of sites still allow RSA key exchanges**

While more than 99% of servers in the top million prefer the use of Diffie-Hellman or elliptic curve Diffie-Hellman key agreements, 52% still allow the use of RSA key exchanges, should that be all the client supports.

During TLS handshakes, 2,991 web servers (0.4% of scan results) chose a cipher suite F5 Labs considers weak—which is defined as using export-grade key lengths, anonymous authentication, or RC4, DES/3DES, RC2/RC4, MD5, or null encryption. For example, the best cipher suite one website offered to clients was SSL 3 using RC4-SHA, a combination of protocol and cipher suite considered state of the art in the mid-1990s. A larger number of sites (12.4%) made weak cipher suites available but didn't choose them for connections.

On the subject of legacy protocols, we found SSL 3 stubbornly clinging to life in the wild. In 2019, 3% of sites in the top million still allowed this legacy protocol. Our 2021 scan revealed that 2% of sites still have SSL 3 enabled. That represents some progress, but not enough.

Failure to remove older protocols and cipher suites from a web server after the implementation of new ones is sufficient to allow a threat actor to perform downgrade attacks.

POODLE, FREAK, Logjam, and SLOTH are all vulnerabilities in which an active attacker manipulates the TLS handshake between client and server and tricks both into believing either one only supports an older (weaker) connection. While TLS 1.3 does offer downgrade protection against active attackers, those protections should not be relied upon, especially since the implementation of specifications often creates vulnerabilities. Leaving old, vulnerable protocols enabled should only be done in extreme cases when business requirements outweigh the risks.

Weak certificates

Moving on to certificates, we also found more room for improvement. Of the 1 million sites scanned, 0.3% used RSA certificates with 1024-bit keys, which haven't been available from trustworthy CAs since 2013. (Fortunately, only one site in the top 10,000 had a certificate with a 1024-bit key.) Among the RSA certificates alone (that is, leaving aside the elliptic curve based ECDSA certificates), 1024-bit keys made up just under half a percent (Table 2).

Table 2
Keys used with RSA and elliptical curve certificates

RSA	
8192	0.01%
4096	8.83%
3072	0.55%
2048	90.13%
1024	0.48%
512	0.00%
ECC	
384	3.3%
256	96.7%

The size of the public certificate's public key should be chosen carefully. Websites often claim "military grade AES128" encryption, which means very little if that symmetric 128-bit key is initially protected by a certificate whose public key is only 512 bits—akin to fastening a thick metal chain with a plastic cable tie. It's important that the public key (stored in the server's private certificate), the symmetric key size (as defined in the chosen cipher suite), and the message authentication scheme (also defined in the cipher suite) all provide similar levels of security or a weakness in one may undo the entire chain.

The website keylength.com[23] provides a useful look at public key, symmetric key, and hashing algorithms by comparing their security in bits. It provides key length recommendations from various groups around the world, including NIST, NSA, BSI, and ECrypt. Table 3 provides a simplified view of the relative strength of RSA and elliptic curve public-key cryptography.

Table 3
A security comparison of RSA and ECC key sizes

Security level (bits)	RSA key size	ECC key size
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	512

A 256-bit ECC key provides a slightly higher level of security than 2048-bit RSA keys. As shown in Table 1, the majority of connections on the web use AES-256 with a 384-bit hash, but the majority of those are initially protected by a 2048-bit RSA certificate. If a comparable level of security is required across the public key, symmetric, and hashing algorithms, then a 3072-bit or larger RSA key (or a 256- to 384-bit ECC key) should be used for the certificates.

Table 3 also shows how quickly RSA key lengths can get out of hand. Doubling the security from 128 bits to 256 bits requires an RSA key five times larger. This has a huge impact on web server performance, since even a modest doubling of RSA key lengths more than halves performance.

Table 4 lists the results of running the OpenSSL command shown to compare public key operations. Specifically, doubling the RSA key length from 2048 to 4096 bits results in a 72% performance drop for RSA verifications and an 85% drop for RSA signatures. The move to elliptic curve crypto is useful today but will become essential for increasing security in the near future.

```
openssl speed rsa2048 rsa4096 ecdsap256
```

Table 4
Speed comparison of public key operations as performed on a modest Windows 10 desktop

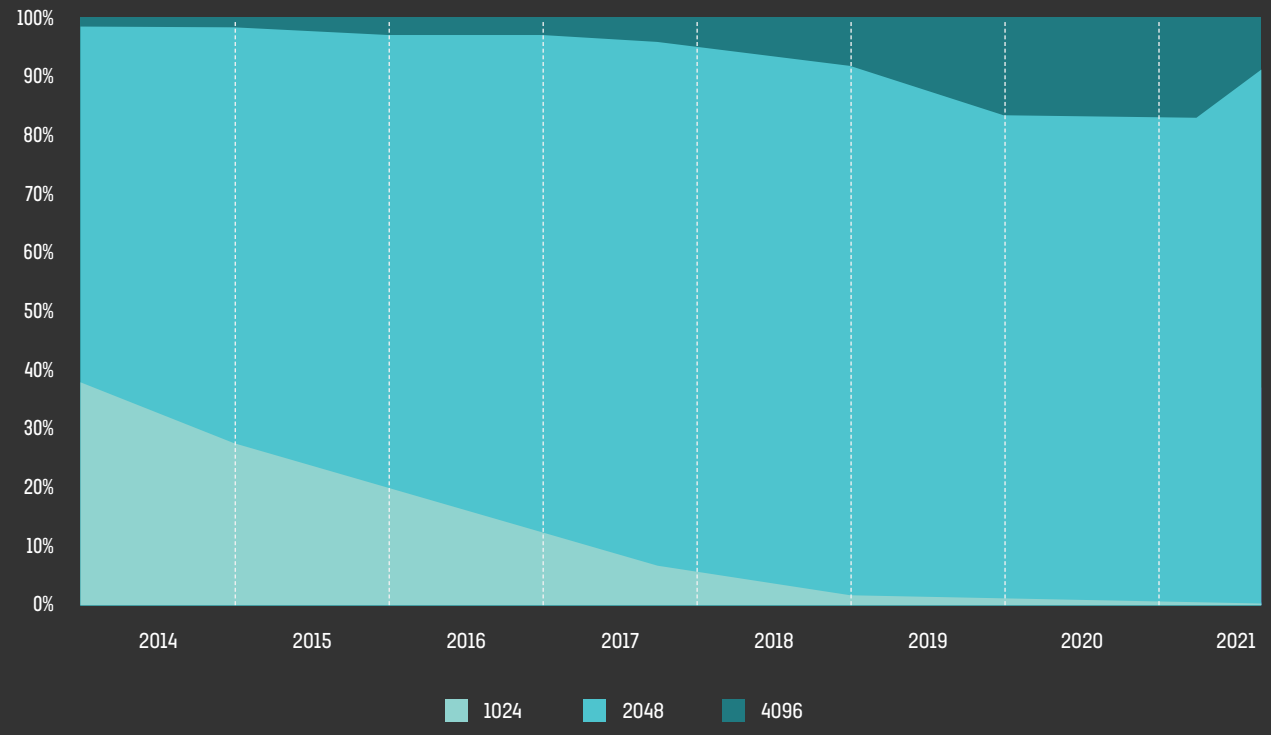
Public-key algorithm	Sign operations per second	Verification operations per second
RSA 2048 bits	1960.5	67381.8
RSA 4096 bits	289.4	18961.0
ECDSA 256 bits	55101.8	17565.9

While 2048-bit certificates are still strong by today's standards, organizations need to keep in mind the expected cover time provided by key lengths. 2048-bit RSA keys are a long way from being factored with any degree of efficiency, but they probably will be easily broken within 10 to 20 years. Businesses should consider whether encrypted data captured today by a passive snooper could cause problems if decrypted in 10 years' time. For shopping transactions on e-commerce stores, that's unlikely. But if the TLS connection is protecting intellectual property or top-secret government communications, cover time should absolutely be a consideration. As previously mentioned, keylength.com is useful for evaluating whether your certificate key length will be adequate in years to come as well as today.

Figure 11 shows that while the number of 1024-bit certificates is declining, the number of 4096-bit certificates also declined in mid-2021 in favor of 2048-bit certificates. This is likely due to many websites moving from RSA to ECDSA certificates, as noted earlier in this report. Of the sites using 1024-bit certificates, nearly 70% are running Apache, with 24% using NGINX. Apache version 2.0, which accounts for 32% of the Apache servers, is the most prominent offender. Since Apache 2.0 was released in 2002 and last patched in 2013, this data strongly suggests that many web servers are configured once and only touched again to renew the certificate.

FIGURE 11: RSA CERTIFICATE KEY LENGTHS

RSA certificate key lengths over time



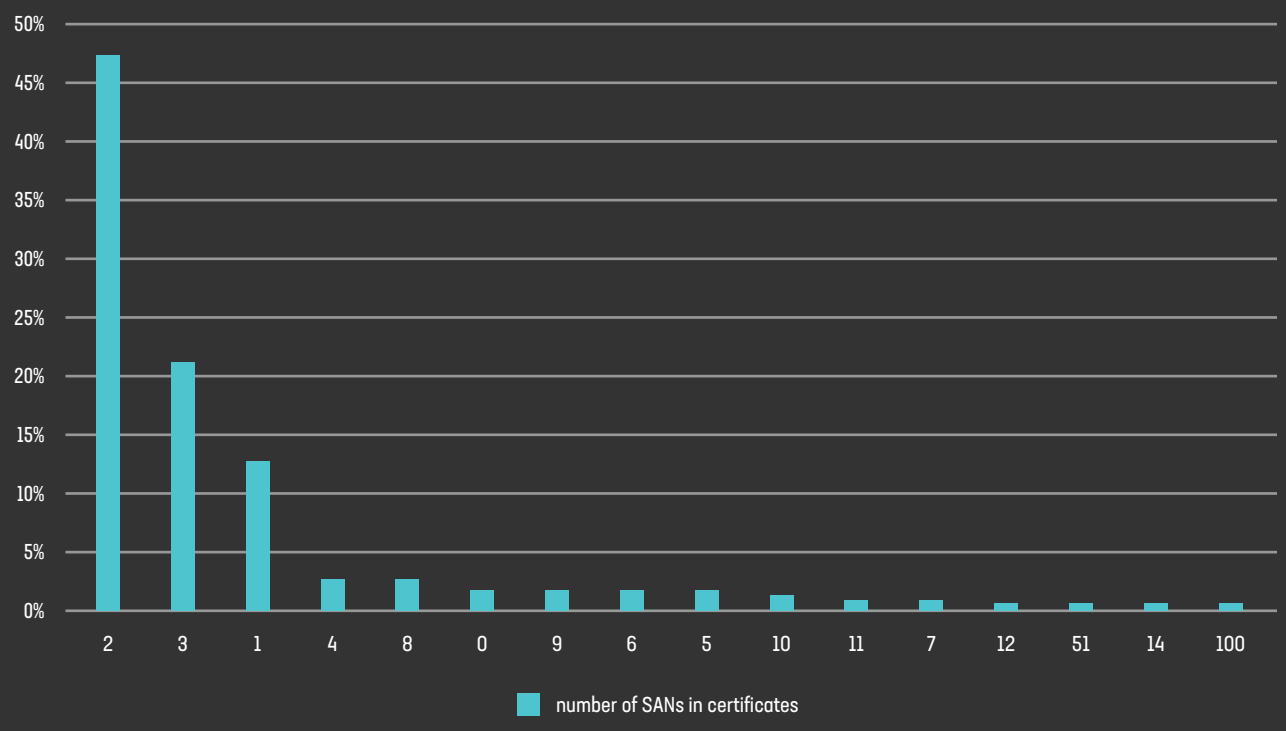
Thankfully, we found only seven sites in the top million using 512-bit certificates. However, we're talking about the most popular, heavily trafficked sites on the Internet—so that's seven too many!

Our scans also revealed the number of subject alternative names (SANs) in the certificates. This information provides a sense of how many discrete entities on the web—whether domains, IP addresses, or common names—fall under the purview of the certificate in question. Nearly half of the certificates included two SANs, and certificates with one, two, or three SANs made up about 80% of the scanned certificates (Figure 12). Fully 90% of the certificates contained between one and nine SANs. However, this distribution had a very long tail indeed: 15 certificates contained 991 SANs, and five certificates included 1,000 SANs.

Certificates with a large number of SANs are increasingly common with CDNs, who provide services to hundreds of websites via a single IP address. But while they might offer some operational or business advantages, certificates with a plenitude of alternative names can slow the TLS handshake and are more disruptive when they are revoked than certificates with a smaller number of domains.

FIGURE 12: SUBJECT ALTERNATIVE NAMES FOUND IN CERTIFICATES

The frequency of certificates with multiple SANs



Expired certificates

Digital certificates expire for the same reason passports, driving licenses, and other physical certificates expire: Claims must be verified regularly to be trustworthy. In the digital world, certificates must expire first and foremost to help with the problems of certificate revocation. Distrusting a stolen or incorrectly issued certificate is unreliable, so currently the best way to protect against those threats is to have the certificate expire as soon as possible.

The X.509 v3 digital certificate specification, used by TLS, defines the use of two timestamps: the notBefore and notAfter dates. Together they specify the earliest a certificate may be used and at what point in the future it should no longer be trusted. An expired certificate prevents users from establishing a connection to a website, mobile app, or API secured with TLS. With fully automated certificate services such as Let's Encrypt, it's surprising to still see services frequently knocked offline due to certificates apparently forgotten about and not renewed in time. Over the past 12 months alone, Google Voice, the Microsoft Exchange portal, Spotify, Github, and SpamCop all suffered outages due to expired certificates. Google Play users also found their American Express credit cards removed from their accounts due to certificate expiration.

Overall, we found that 2.5% (or about 1 in 40) of the certificates in the full set of sites were expired at the time of scan.

2.5% **Certificate currently expired in the top 1 million sites**

Fortunately, less than 1% of sites in the top 1,000 (0.7%) were found to have expired certificates, compared with 3.3% of sites in the top million's bottom 1,000. These rates indicate that better certificate management does somewhat correlate with site popularity. But no one's immune.

Ripped from the Headlines, Part III

In early 2020, Microsoft experienced an outage of its popular messaging service, Teams, when a digital certificate expired. The platform now serves over 250 million monthly active users, and the incident shows how a single certificate error can take out an entire platform affecting millions of customers. It's a reminder that HTTPS misconfigurations such as this can happen to organizations of any size. It also illustrates how the magnitude of the task of managing certificates often rises with the complexity of the organization and its web properties; this problem is difficult to completely master at any scale unless it becomes an absolutely core part of operations.

Invalid certificate chains

A lapsed notAfter date is not the only reason a certificate may no longer be trusted. Web browsers expect to receive not only the leaf (server) certificate for a website, but all certificates responsible for digitally signing that certificate, up to (but not including) the root certificate. The order of the chain is important too, although many web browsers are forgiving and still accept certificate chains in incorrect order. However, certificate chains that entirely miss intermediary certificates will generally not be trusted and result in a failed connection to that HTTPS site.

2% **of sites send an invalid certificate chain**

In our 2021 scan, 2% of sites in the top million sent certificates back in an invalid order—down from 2.5% two years ago.

Ripped from the Headlines, Part IV

In a move that both remediated and illustrated the need for more accurate certificate chains, Mozilla implemented a new capability into Firefox in 2020 that preloads intermediate CA certificates and stores them in the local cache, making it more likely that a TLS handshake will succeed. While this magnanimous act improved the Firefox user experience, it also suggested how commonly servers are configured without specifying the intermediate CA certificates. After all, if the problem wasn't serious, Mozilla wouldn't have bothered to fix it.

Other errors

Our research also found a handful of miscellaneous issues that don't fit the categories above but certainly qualify as bad news:

- 2.8% of sites were vulnerable to denial-of-service attacks via client certificate renegotiation.
- 1% of scanned sites support compression, which can make them vulnerable to the CRIME exploit.[24]
- 0.2% of scanned sites do not support secure renegotiation.

Overall, there's a long list of cryptographic oversights or conscious-but-unwise practices that are contributing to failures that risk data breaches—or could in the future, even if they're secure now. Certificate issues of one sort or another make up the bulk of the list. That's an argument for directing sufficient IT resources toward CA selection and ongoing certificate management.

Abuse and Misuse

In addition to the bad news covered above, our study also explored malicious activity or encryption circumstances that might engender it. For instance, our research showed how encryption helps threat actors make social engineering schemes more believable or harvest credentials from cryptocurrency owners with encryption downgrading attacks. Finally, governments also circumvent encryption for surveillance and espionage.

Threat Hunting with TLS Fingerprinting

Website operators have been using device fingerprinting for many years to help distinguish malicious bots from genuine customers. Since many threat actors intentionally modify their client device's browser headers and other properties, it can be useful to measure a client's hidden signals in an attempt to find its true identity. However, fingerprinting servers, specifically Transport Layer Security (TLS) fingerprinting, is rarely performed. By incorporating the Salesforce JARM TLS fingerprinting technique directly into Cryptonice, we were able to capture server TLS fingerprints for the top one million sites.[25] The results not only revealed a perhaps unsurprising lack of variance, but they also indicated that malicious command-and-control (C&C) servers may be lurking among the world's most popular sites. Before we dive into attacker behavior, however, let's cover what a TLS fingerprint is and what it can tell us.

For each **Client Hello** message in the TLS handshake, a web server may respond with a unique **Server Hello** that will differ based on the operating system, TLS library, the preferred order of cipher suites, and other configuration options. The TLS fingerprinting technique sends specially crafted client parameters to a web server and carefully measures its response to create a unique fingerprint. This can then be compared with other web servers to determine if they are configured the same way. This could be useful for auditing purposes, for example, ensuring that all servers for any given organization are configured the same way. Fingerprinting is also a useful way to identify servers that may deliberately hide HTTP response headers. One caveat: despite its potential advantages, fingerprinting is far from foolproof. Recently, we've seen examples of attackers selling client fingerprints on dark web markets with the specific intention of avoiding fingerprint-based security controls. So, while fingerprinting is not a perfect method for identifying a server, it does highlight areas worthy of future research.

One in a Million ... Or Not, As It Turns Out

Across the one million sites on the Internet, we found only 8,851 different TLS fingerprints, and of those only 4,035 were completely unique. Why so few? Why does every server not have a completely unique fingerprint? One possible explanation is that many servers are configured exactly the same way, that is, using defaults. A standard install of Ubuntu 20.04 with NGINX 1.21.3 using default TLS configurations will, in all likelihood, result in the same fingerprint. However, the real reason for such low variations in fingerprints is the sheer number of content delivery network (CDN), DDoS-mitigation, and cloud proxies being used.

One single TLS fingerprint, belonging to Cloudflare, accounted for almost 20 percent of the top one million sites. In aggregate, Cloudflare is responsible for 262 unique fingerprints, which constitute 25 percent of the fingerprints found.

The second most common fingerprint, at 2.4 percent, is found with NGINX. This is significantly less than Cloudflare in the top spot. However, if we combine all fingerprints associated with NGINX, then it rises to the top, claiming almost 28 percent of all fingerprints in the top one million.

The most common unique Apache fingerprint was just behind NGINX, at 2 percent, and combined Apache fingerprints account for exactly 27 percent of the total, placing it in third place for unique fingerprints and second place for aggregate fingerprints. See Table 5 for a summary of fingerprints by the top three website server platforms. In all, 80 percent of the top one million sites produced just 203 unique fingerprints,

Table 5
The most commonly found TLS fingerprints across the top 1 million sites

Web Server	Most Common Single Fingerprint	Most Common Single Fingerprint Host	Total Fingerprints for This Server	Number of Different Fingerprints for This Server
Cloudflare	18.9%	CLOUDFLARENET	24.5%	262
NGINX	2.4%	Google	27.9%	2,340
Apache	2.0	Amazon-02	27.0%	2,935

The results of F5 Lab's fingerprinting of the top one million sites leads to the question: is lack of variance in TLS configurations a concern from a security perspective? If TLS fingerprinting is a reliable way to identify vulnerable servers, then clearly yes. The JARM fingerprinting method has been included in services such as Shodan.[26] This means that not only can website owners perform lookups for their own matching fingerprints but so too can threat actors, as shown in Figure 13.

Figure 13
Example Shodan query to search for TLS fingerprints

On the whole, however, cloud providers often benefit from having teams of dedicated engineers who understand how to correctly secure HTTPS deployments. They can continuously maintain TLS configurations and ensure that all customers benefit from the ever-changing best practices.

TABLE 6: TLS FINGERPRINTS FOR KNOWN MALWARE SERVERS

Malicious Server C&C or Red Team Tool	JARM Fingerprint	Overlap with Top One Million Sites
AsyncRAT	1dd40d40d00040d1dc1dd40d1dd40d3df2d6a0c2caaa0dc59908f0d3602943	0
Cobalt Strike (1)	07d14d16d21d21d07c42d41d00041d24a458a375eef0c576d23a7bab9a9fb1	4
Cobalt Strike (2)	07d14d16d21d21d07c42d41d00041d24a458a375eef0c576d23a7bab9a9fb1	4
Deimos	00000000000000000041d00000041d9535d5979f591ae8e547c5e5743e5b64	0
Dridex (1)	21d19d00000000021c21d19d21d21db2e1191a3715fa469c667680e6cfab7f	9
Dridex (2)	05d10d20d21d20d05c05d10d05d20d74fcf6501ae7a92319e575bfafd2a827	1153
EvilGinx2	20d14d20d21d20d20c20d14d20d20daddf8a68a1444c74b6dbe09910a511e6	6
MacC2 (1)	2ad2ad0002ad2ad22c42d42d000000faabb8fd156aa8b4d8a37853e1063261	3
MacC2 (2)	2ad2ad0002ad2ad00042d42d000000ad9bf51cc3f5a1e29eecb81d0c7b06eb	14
MacShellSwift	2ad00000000000000000000000000eeebf944d0b023a00f510f06a29b4f46	0
Merlin	29d21b20d29d29d21c41d21b21b41d494e0df9532e75299f15ba73156cee38	21
Merlin C2	29d21b20d29d29d21c41d21b21b41d494e0df9532e75299f15ba73156cee38	21
Metasploit	07d14d16d21d21d00042d43d000000aa99ce74e2c6d013c745aa52b5cc042d	1
Metasploit SSL listener (1)	07d14d16d21d21d00042d43d000000aa99ce74e2c6d013c745aa52b5cc042d	1
Metasploit SSL listener (2)	07d14d16d21d21d07c42d43d000000f50d155305214cf247147c43c0f1a823	11
Mythic	2ad2ad0002ad2ad00042d42d000000ad9bf51cc3f5a1e29eecb81d0c7b06eb	14
QakBot	04d02d00004d04d04c04d02d04d04d9674c6b4e623ae36cc2d998e99e2262e	62
Sliver	2ad2ad0002ad2ad00041d2ad2ad41da5207249a18099be84ef3c8811adc883	9
Trickbot	2ad2ad0002ad2ad22c2ad2ad2ad2adce7a321e4956e8298ba917e9f2c22849	531

Malicious Servers

Here is where it gets interesting from a cybercrime standpoint: since phishing sites and C&C servers will intentionally attempt to disguise their configuration, fingerprinting techniques can be a useful way to spot the true identity of web servers. Table 6 shows JARM fingerprints for known malware servers or red team tools and the number of times that fingerprint was found in our scans of the top one million sites.

As we would hope, the number of (potentially) malicious servers is very low, though it is far from zero. Trickbot and Dridex (2), in particular, show relatively high counts for their associated fingerprints. This does not necessarily mean that all servers are infected with those malware families, simply that the web servers we scanned have identical TLS fingerprints to those malware strains. These could be false positives, or they could indicate that a small percentage of the web's most popular sites are being controlled, either knowingly or unknowingly, by attackers.

This is particularly significant in light of the growth of ransomware in 2020 and 2021. The 2021 Application Protection Report noted that Trickbot and Cobalt Strike were two of the top three most frequently observed malware variants for delivering ransomware, along with Emotet. The implication is that some of the web's most popular sites are also delivery vehicles for some of the most devastating attack trends in the last five years.

Phishing in the Murky Depths

As reported in our 2020 Phishing and Fraud Report, 70% of phishing sites used HTTPS with valid certificates to appear more legitimate to victims. Data from OpenPhish indicates that this figure is now almost 83%, with only 17% of phishing sites using insecure HTTP-only connections. We also found that the majority of malicious sites—just under 80%—come from just 3.8% of the hosting providers.

In our 2019 TLS Telemetry Report, we indicated that the web hosting control panel cPanel— through its AutoSSL capability arising from its integration with Sectigo—was the preferred method of obtaining and installing free digital certificates on phishers' websites. Today, Let's Encrypt has taken the lead, providing 28% of certificates for phishing sites. Phishers are either finding alternative ways to deploy their sites or perhaps using the optional Let's Encrypt plug-in for cPanel.

For service providers, phishers tended to prefer Fastly, though several other providers—namely Unified Layer, Cloudflare, and Namecheap—hosted similar proportions of phishing networks (Figure 14).

When it comes to phishing bait (after the use of generic or highly targeted spear phishing attacks), the brands most commonly targeted in phishing attacks were Facebook and Microsoft Outlook/Office 365. This reflects the value of stolen credentials from these sites, in part because so many other accounts rely on these as identity providers (IdPs) or for password resets.

Figure 15 also contains another finding hiding in plain sight: If we combine the various webmail targets, such as Outlook, Outlook365, and other webmail providers, webmail accounts for 10.4% of impersonated web functions—just under Facebook's percentage. In other words, the diversity of webmail platforms shouldn't obfuscate the fact that phishing victims are almost equally likely to experience a phish against their webmail accounts as against their Facebook accounts.

FIGURE 14: HOSTS USED FOR PHISHING SITES

Most popular hosting platforms for phishing sites

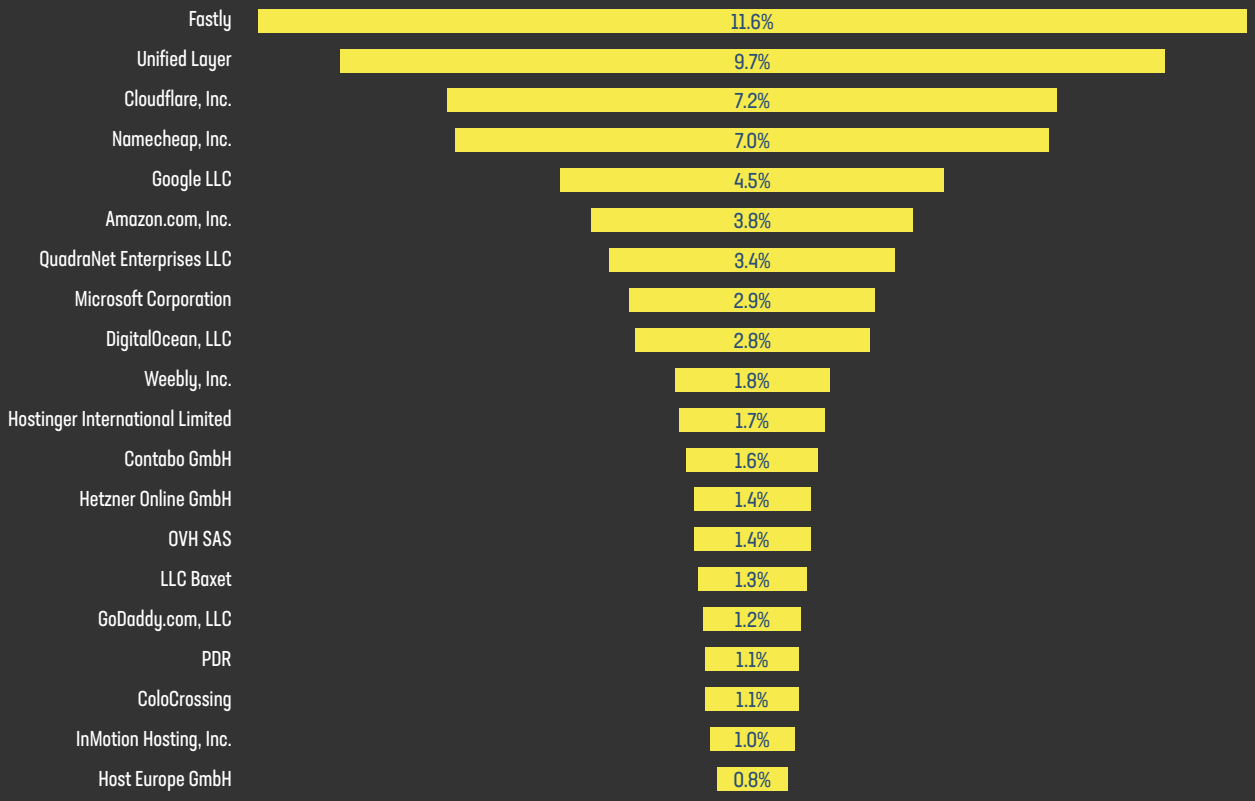
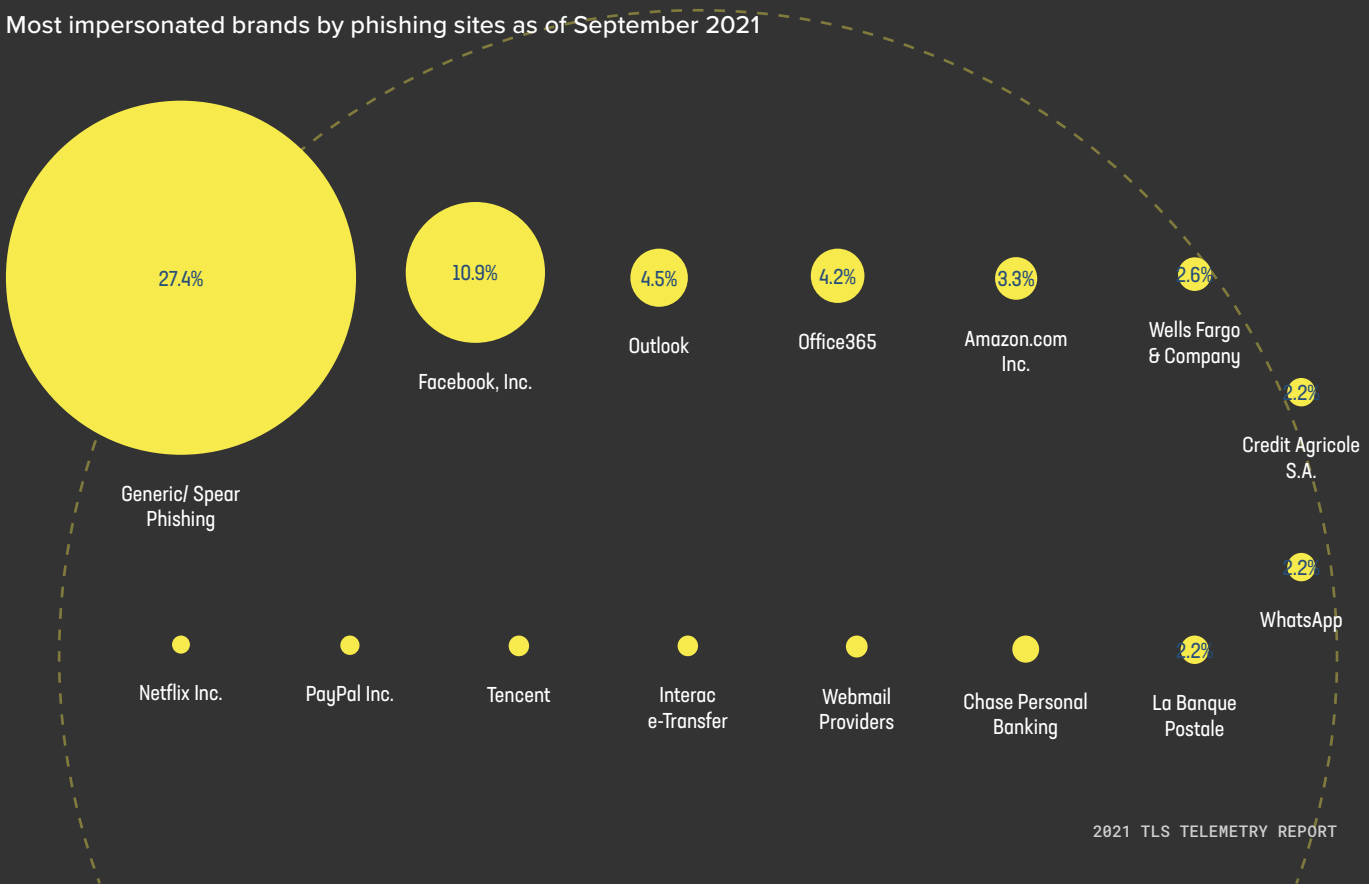


FIGURE 15: MOST IMPERSONATED BRANDS OR WEBSITES

Most impersonated brands by phishing sites as of September 2021



When is Encryption Not Encryption?

The web is well on its way to becoming 100% encrypted, but we're currently in an awkward transition phase. Although the majority of websites prefer encrypted HTTPS connections, their operators hesitate to disable HTTP altogether, and many sites simultaneously allow clients to connect over both secure and insecure protocols. Some site operators fear inadvertently blocking access to customers using old browsers, while others may not understand the risks of leaving the unencrypted HTTP protocol enabled.

Once a secure TLS session has been established, the client and server can be sure communications are private and trustworthy. If, however, an active attacker (let's call her Mallory) can intercept the very first connection between the two, she can trick the client into thinking the website only supports HTTP connections.

Figure 16
An attacker fooling a client with SSLstrip deception

This kind of attack, dubbed SSLstrip by its creator, Moxie Marlinspike, is extremely potent and can be used to capture sign-in credentials, personal information, and payment card details from any website. However, this attack can't be performed remotely. An SSLstrip attack requires the attacker to be on the same network as the victim, since they need to capture and modify all network traffic. An ideal place for Mallory to hang out, therefore, would be a coffee shop that offers free Wi-Fi and where multiple victims all use the same open network.

Ripped from the Headlines, Part V

Beginning in January 2020, administrators for The Onion Router (Tor) noticed a significant proportion of Tor exit relays, ranging from 20 to 27% of all exit relays, were modifying a small number of connections between clients and a very specific set of hosts: cryptocurrency exchanges. The relays were preventing HTTPS redirects, resulting in users defaulting to the HTTP versions of the cryptocurrency exchange sites, making the information they passed along to those sites potentially vulnerable. This kind of attack, known as SSLstrip or SSLstripping, was still ongoing in May 2021[27] despite ongoing efforts by the Tor Project to remove malicious relays.

A few tactics can prevent this kind of attack:[28]

- Completely disable HTTP connections and force all clients to use HTTPS.
- Use HSTS to instruct the web browser to only ever attempt connections to the website over HTTPS.
- Have users configure their browsers to only use HTTPS connections for all websites. (Users can search browser settings for “https” to do so.)

More broadly, this kind of attack illustrates the degree to which certain assumptions about the Internet and its infrastructure no longer hold on the dark web. Much of the care necessarily exercised by site administrators and product owners on the web is offloaded to users on the dark web, and cryptography is no different.

Future Encryption vs Government Interception

Governments all over the world continue to propose and roll out new laws that affect the strength and types of encryption permitted within their borders. Law enforcement agencies say that limiting encryption or providing lawful interception is essential for bringing criminals to justice. Privacy advocates (and many opposition political parties) argue that governments only wish to perform mass surveillance of their citizens, and that weakening encryption threatens everyone's security. Despite these concerns, nations around the world are prohibiting or limiting the use of encryption.

Their motives may be mixed. In the wake of the July 2021 revelations about the scope and scale of NSO Group's use of their Pegasus spyware, for instance, information about its application by Kazakhstan went under the radar for most of the world. However, since the F5 Labs reported in 2019 on the Kazakhstani government's attempts to decrypt its citizens' traffic, we also noticed the nation's government has used Pegasus to track not only domestic dissidents but also the country's current and former prime ministers. While this new surveillance is not TLS-related, it stands as a proof that the Kazakhstani government isn't finished spying on its own people,[29] and it's unlikely to be the only one.

More recently, newer encryption standards such as TLS 1.3, encrypted DNS such as DNS over HTTPS (DoH) or DNS over TLS (DoT), and eSNI (encrypted server name indicator) have come into the crosshairs of restrictive governments. In August 2020, security researchers found that the Great Firewall of China was attempting to block TLS connections that used eSNI values in the TLS handshake.[30] The Russian government has also proposed an amendment to an existing law to prohibit the use of any technology that enables "hid[ing] the name (identifier) of a web page or site."[31] If implemented, this law will outlaw the use of eSNI and encrypted DNS.

Major browsers such as Chrome and Firefox allow manual configuration of DoH (although in some regions, such as the United States, DoH is now enabled by default). But few users are aware of the value or even the need for such manual configurations, leaving them at the mercy not only of intrusive governments but potential attackers and negligent site operators whose cryptography practices lag behind current standards.

Is Quantum Cryptography Here Yet?

No.

And that's not even a real question, so stop asking.[32]

Conclusion

The desire to intercept, weaken, and circumvent encryption has never been greater. Nation-states and cybercriminals alike are attempting to work around the problems caused by strong encryption. While this rarely results in direct attacks against cryptographic algorithms or protocols, it often leads attackers to instead think of creative ways to intercept or capture information before or after it has been encrypted. With these risks ever-present, it has never been more important to focus on strong and up-to-date HTTPS configurations, particularly when digital certificates are shared across different services.

As with many areas of information security, weaknesses come not from the latest and greatest features that we struggle to adopt, but the old ones we are reluctant to disable.

Here at F5 we have a passion for all things crypto. The F5 Labs research team frequently reports on updates to our Cryptonice HTTPS scanner, in addition to providing everything from educational series on the basics of crypto to deep dives on currently recommended best practices. Subscribe to the F5 Labs newsletter to ensure you are always kept up to date.

Sources

1 https://owasp.org/www-project-top-ten/

2 https://tranco-list.eu/

3 https://www.f5.com/labs/articles/threat-intelligence/2019-tls-telemetry-report-summary

4 For more information about Early Data, see https://httpwg.org/specs/rfc8470.html

5 https://datatracker.ietf.org/doc/rfc8996/

6 Often, country code top level domains (ccTLD) whose letter combinations have other meanings become widely used for other purposes. The most obvious examples of this are the use of the ccTLD .io, which is supposed to be allocated to the British Indian Ocean Territory, for tech startups due to the assocation of IO with input/output, as well as the use of the Macedonian domain .me for personal mail services and the like. This practice is the reason why using geophysical lookups at the time of scan is the best way to determine the location of a host, rather than using ccTLD.

7 https://thehackernews.com/2020/09/ssl-tls-certificate-validity-398.html

8 https://insecure.design/

9 https://insecure.design/

10 https://www.f5.com/labs/articles/threat-intelligence/2020-phishing-and-fraud-report

11 https://owasp.org/Top10/A02_2021-Cryptographic_Failures/

12 For Mozilla's reasoning behind their decision, see https://groups.google.com/g/mozilla.dev.security.policy/c/PnAAWnxyosM?pli=1. For the list of security issues, see https://wiki.mozilla.org/CA:Camerfirma_Issues.

13 https://community.letsencrypt.org/t/2020-02-29-caa-rechecking-bug/114591

14 https://gitlab.com/gnutls/gnutls/-/issues/1011

15 https://research.nccgroup.com/2020/08/24/technical-advisory-wolfssl-tls-1-3-client-man-in-the-middle-attack/

16 https://alpaca-attack.com/

17 https://us-cert.cisa.gov/ncas/current-activity/2021/10/08/nsa-releases-guidance-avoiding-dangers-wild-card-tls-certificates

18 https://media.defense.gov/2021/Oct/07/2002869955/-1/-1/0/CSI_AVOID%20DANGERS%20OF%20WILDCARD%20TLS%20CERTIFICATES%20AND%20THE%20ALPACA%20TECHNIQUE_211007.PDF

19 https://www.globalsign.com/en/blog/poodle-vulnerability-expands-beyond-sslv3-to-tls

20 http://archiv.infsec.ethz.ch/education/fs08/secsem/bleichenbacher98.pdf

21 https://robotattack.org/

22 https://drownattack.com/

23 https://www.keylength.com/

24 https://threatpost.com/crime-attack-uses-compression-ratio-tls-requests-side-channel-hijack-secure-sessions-091312/77006/

25 https://github.com/salesforce/jarm

26 https://www.shodan.io/

27 https://nusenu.medium.com/how-malicious-tor-relays-are-exploiting-users-in-2020-part-i-1097575c0cac; https://therecord.media/thousands-of-tor-exit-nodes-attacked-cryptocurrency-users-over-the-past-year/

28 https://blog.torproject.org/bad-exit-relays-may-june-2020

29 https://eurasianet.org/kazakhstan-activists-tracked-by-pegasus-angered-but-not-surprised

30 https://gfw.report/blog/gfw_esni_blocking/en/

31 https://regulation.gov.ru/projects#npa=108513

32 Quantum computing isn't really involved with cryptography, as such. We already have quantum key distribution, and quantum computers might (one day) drastically speed up the cracking of current encryption. But there's no such thing as 'quantum cryptography'. At least not yet.

APPLICATION THREAT INTELLIGENCE

US Headquarters: 401 Elliott Ave W, Seattle, WA 98119 | 888-882-4447 // Americas: info@f5.com // Asia-Pacific: apacinfo@f5.com // Europe/Middle East/Africa: emeainfo@f5.com // Japan: f5j-info@f5.com