THE 2016

TLS TELEMETRY REPORT

TRACKING GLOBAL INTERNET ENCRYPTION TRENDS

December 2016

by David Holmes

F5 LABS

TABLE OF CONTENTS

TABLE OF FIGURES

EXECUTIVE SUMMARY

In just four short years, a healthy dose of paranoia about individual privacy as well as emerging support for encryption by browsers, social media sites, webmail, and SaaS applications have pushed encryption estimates from almost non-existent (in the low single digits before 2013) to just over 50% by the end of 2016.

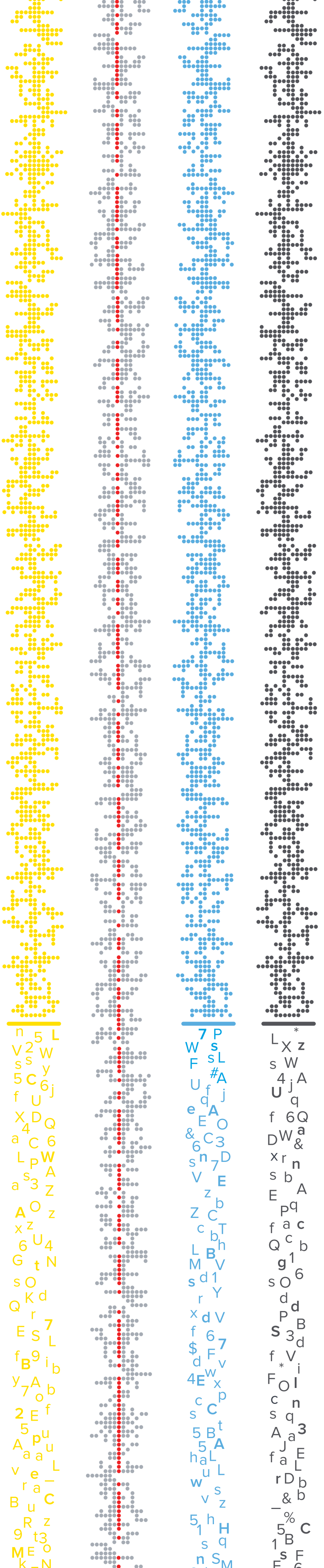
That's quite a victory for data privacy, but just how much of a victory?

F5 Labs explores that question in the first of our annual TLS Telemetry reports. Our goal is not just to report raw data, but to make that data actionable by describing the who, what, when, how, and why of cryptography, and provide guidance on what's next for your organization. This being our initial report, we've taken care to explain our motivations—crazy though they may seem—for scanning the entire TLS Internet, describe our research methodology, and recap the recent history of and summarize the current cryptographic landscape.

Specifically, we look at:

- Usage and preferences for current (and aging) cryptographic protocols such as TLS and SSL
- The implications of self-signed certificates
- Trends that are driving the adoption of Forward Secrecy
- Reasons why HTTP Strict Transport Security adoption is sluggish
- The truth about block and stream ciphers
- The relative security of today's most popular web servers

Finally, we conclude with recommendations for improving your organization's overall cryptographic posture.



THE QUIET YET PROFOUND REVOLUTION

Humanity passed a profound milestone in October 2016 that went largely unsung except for a half dozen retweets of two auto-generated traffic graphs. Both Mozilla and Chrome telemetry showed that the majority of encrypted page requests outnumbered those of unencrypted page requests. By this measure, for the first time in human history, the majority of the Internet was encrypted (at least in transit).

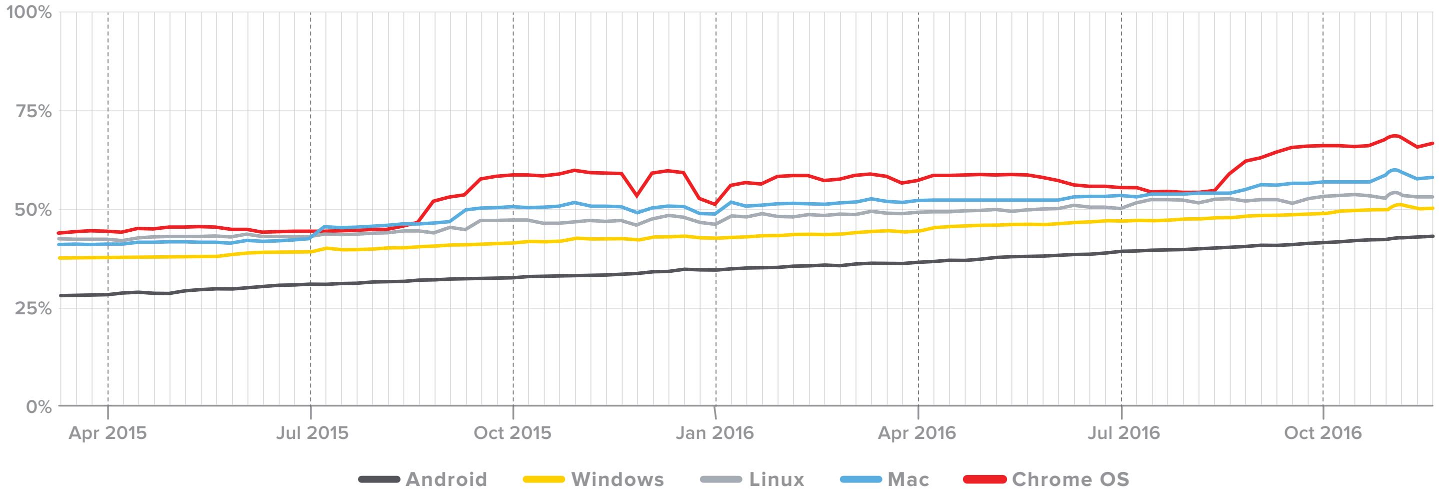


Figure 1: Google Chrome graph showing percentage of pages loaded over HTTPS

We should pause to reflect on the significance of this moment. Long before the digital age, only those in power could use cryptography to defend their interests—or to inflict damage on their enemies. Julius Caesar protected military communications with primitive ciphers. For the next millennia, ambassadors enciphered their communiques to sovereigns back home. Mary, Queen of Scots, used cryptography in a regicide attempt. Axis powers in World War II developed the incredible Enigma machine to achieve 76-bit encryption (with no transistors!). Even more incredibly, the Allied powers broke the Enigma encryption *using pencil and paper.* From then until very recently, cryptography remained in the hands of the agencies that used it in the same way it had been used before: for intelligence purposes toward the protection of the Church or the State.

The global adoption of the World Wide Web has given ordinary people the ability to communicate with their peers directly, though not securely. Nation states, law enforcement, thieves, and even (shudder) lawyers retained the ability to snoop on the digital chatter of the rest of us through wiretaps, broad spectrum mass surveillance, and subpoenas.

October 2016 marks the turning of the tide of the Quiet Revolution when the majority of digital communication pierced the surveillance horizon and became relatively secure.

So while we should most definitely break out the champagne, Mickey big mouths, and edibles in our celebration, afterward, we should pause and ask ourselves, *exactly how much more secure are we as a result of this quiet revolution?*

The researchers at F5 Labs have quantified the answer to that very question.

WHY IS F5 LABS INTERESTED IN SSL/TLS?

We at F5 Labs have an interest in the Encrypted World, and it is not just because we proudly wore propeller-head hats and exchanged code books in school. (Okay, that's part of it.) It's also because a *significant percentage of the Encrypted World is decrypted by F5 devices*. What percentage of commercial TLS traffic terminates on F5 appliances, you might be asking? Well, like your brother's Facebook relationship status, it's complicated; it depends on exactly how that TLS traffic is broken out, but we'll get deeper into the stats in a bit.

In the summer of 2014, researchers at F5 labs began a project to sample the TLS hosts on the Internet. The goal of the sampling was to collect and aggregate global metrics for TLS protocol selection, cipher selection, and overall cryptographic security posture. For two years the researchers labored day and night, only coming out of their labs for pizza and craft beer. In the summer of 2016, those vigilant researchers had perfected their Internet scanner to the point where instead of just sampling data, it could scan the entire TLS Internet.

This report compares the data from the summer of 2016 back through time to the data from the summer of 2014 to chart the trends of cryptographic data in the Encrypted World.

ARE YOU READY FOR THIS? MAYBE YOU'D BETTER POUR YOURSELF A FRESH CUP OF COFFEE AND SIT DOWN.

OUR SCANNING METHODOLOGY

These are easy days for Internet researchers. In other scientific fields, a researcher has to get grant money to go out and gather sample data: maybe on the other side of the world or maybe by surveying 28,000 Icelandic twins.

But Internet researchers can just spin up a machine with a fast connection and sample the Internet at will. Several research groups do this; most are private research scanners, but some are quite high-profile public scanners. One of the earliest (though now defunct) was the Electronic Frontier Foundation (EFF) SSL Observatory. Another is the Qualys SSL Labs scanner, which generates the SSL Pulse report and probes very deeply into each site. Site owners submit their site IP addresses to the scanner, thereby implying tacit permission for the deep probes. At the time of this writing, the Qualys SSL Labs scanner has a dataset of about 150,000 sites.

Rapid7 is the commercial face of the Metasploit Framework, a penetration toolkit. Its creator, H.D. Moore, enjoys scanning the Internet with rapid-scan tools like zMap. He and his colleagues have teamed up with the University of Michigan to scan *the entire Internet every week*. They don't always post their analysis of the data, but they make the datasets available through a site called Project Sonar[1].

[1] https://sonar.labs.rapid7.com/

F5 Labs pulls lists of known TLS hosts from Project Sonar's SSL/TLS known-hosts lists. Between 2014 and 2016, Project Sonar has been tracking approximately 28 million known TLS hosts on the Internet.

28.2M	656K	226	9
TLS Hosts Scanned	Busiest Sites Scanned	Craft Beers Drunk	Quarterly Scans

Project Sonar's datasets are approximately double that size (60 million lines) because when a host offers multiple certificates from a single IP address, each certificate and IP address pair is counted as a different entity. As an example:

```
23.11.148.142,6ad2b04e2196e48bf685752890e811cd2ed60606
23.11.148.142,4d34ea92764b3a3149119952f41930ca11348361
23.11.148.142,4b7409acde6ab6b6a5f4c8f34eb3994ab1eaaea2
```

The F5 scanner isn't as concerned with the certificates (although it could be, someday), so we simply use the list of 28 million IP addresses from Project Sonar as a starting point. When the F5 TLS scanner first started in 2014, it sampled a subset of the Internet using lightweight probes that collect the TLS characteristics of the individual sites found. The scanner has been sampling approximately one million TLS sites per quarter. In the summer of 2016, we expanded our scan to include a complete census of nearly every single host on the Internet that responds to port 443 (HTTPS).

IP-based Scanning Versus Popular Domains

The IP address-based host data from Project Sonar gives the most complete picture of TLS servers on the Internet. However, a blind TLS request to many IP address servers may fail due to the Server Name Indicator (SNI) problem. That's because SNI multiplexes certificates through a single IP address. For example, suppose that both www.example.com and www.domain.com resolve to the same IP address, 1.2.3.4. When a browser connects to address 1.2.3.4, it can indicate via the SNI extension which site it would like to see—say, www.example.com. The server can then provide the correct certificate for www.example.com and the connection can proceed. If a browser or scanning device connected directly to the address (https://1.2.3.4) without specifying a domain, the result is defined. Some servers will serve a default certificate, but many will refuse the connection. In their recent paper, *Towards a Complete View of the Certificate Ecosystem*[2], J. A. Halderman, et al. document the number of non-responding SNI servers as at least 1.5%.

For this reason, F5 Labs researchers use both IP-based scans and scans via popular domains. The most popular popular-domain list is Amazon's Alexa Top Sites, which lists up to 1 million busiest sites. However, the Alexa Top Sites list is not perfect. It has been rumored to have shortcomings with its ordering and inclusion. Yet, it represents at least a *lingua franca* for popular sites. Since the domains are known for the Alexa Top 1 million sites, the F5 scanner can provide the domain name via SNI and always get a response.

[2] https://jhalderm.com/pub/papers/https-perspectives-imc16.pdf

In the fall of 2016, the number of hosts in the Alexa Top 1 million list that responded to the F5 scanner was 656,161, or approximately 66%.

The Graphs In This Report Reflect Three Different Data Sources:

- Statistics gathered from IP-based scanning (either sampling or the full 2016 scan) are referred to as "Internet-at-large" or just "at-large."
- Statistics gathered from scanning portions or all of the Alexa top 1 million are referred to simply as "Alexa."
- Statistics gathered from devices identified as F5 application delivery controllers (ADCs) are referred to as "F5."

Two key points need to be kept in mind for each topic in this report.

First, for some topics, only one or two data sources were used to gather data. Where there is a "hole," it is typically from the Alexa dataset, which our researchers did not scan every quarter.

Second, in order to keep the probes fairly lightweight (and therefore gentle to the server), only a handful of connections were performed against each server. This means that the F5 scanner largely records the server preference for a particular metric. An example of what we're talking about here is RC4. If a server *purposefully chose* RC4 as the symmetric cipher it was queried, that was recorded. If the server chose Advanced Encryption Standard (AES), we don't know if it supported RC4 or not. The two exceptions are forward secrecy and SSLv3 support.

Therefore the statistics represented here refer typically to *the percentage of TLS hosts that prefer Metric X.* Got it?

What About CDN Distortion?

Content Delivery Networks (CDNs) work by having an enormous number of points-of-presence spread out across the Internet. That's their business model—caching static data around the world to reduce latency and server load. During the summer of 2016, Akamai, for example, had over five million IP address that identified as Akamai TLS hosts. That represented nearly 25% of all TLS hosts included in our summer scan.

Akamai uses the exact same cryptographic settings for each TLS host. Unfortunately, this monoculture of settings has a sharp distorting effect on the IP-based data. For example, the SSLv3 graph (see Figure 3) shows a huge decline in SSLv3 support during 2014 after the POODLE vulnerability. At first glance, it looks like a third of the Internet turned off SSLv3 in a single quarter. But in reality, most of that was a single company, Akamai, turning off SSLv3 for a quarter of the TLS hosts on the Internet.

Clearly, no Internet-wide scan of TLS hosts would be complete without including Akamai TLS hosts. However, the distorting effects of its monoculture of TLS configuration must be acknowledged—for research purposes, if nothing else.

CloudFlare is another CDN whose TLS hosts require clients to submit SNI or they will not respond. Therefore, CloudFlare TLS hosts do not show up in IP scans like the F5 scanner. The F5 Labs research team is considering how to address the lack of CloudFlare visibility in future reports.

Okay, enough about methodology; let's get into the research findings, shall we?

GLOBAL CRYPTOGRAPHIC SECURITY POSTURE

If we were to summarize the overall cryptographic security posture of the Internet today, we'd say, "Meh, not bad!" In general, it appears that about 30% of Internet sites are consistently upgrading their security posture every quarter. Another 30% of sites appear to be stuck in time using yesterday's security settings.

PREFERENCE FOR TLS 1.2 SOARS IN TWO YEARS

Let's have a look at one of the most important indicators of cryptographic posture: the preference for TLS 1.2. Version 1.2 is the most current "official" version of TLS, therefore, any servers that don't prefer version 1.2 are significantly behind the times. This is especially true of millions of Microsoft Internet Information Services (IIS) servers (see Server Smackdown section later).

In our very first scan during the summer of 2014, only about a third (36%) of Internet-at-large hosts preferred TLS 1.2. The vast majority of the rest preferred TLS 1.0. (Apparently, version 1.1 is the least popular of the TLS family, with only a few percent preferring it). By the summer of 2016, nearly *two-thirds* of Internet hosts preferred TLS 1.2.

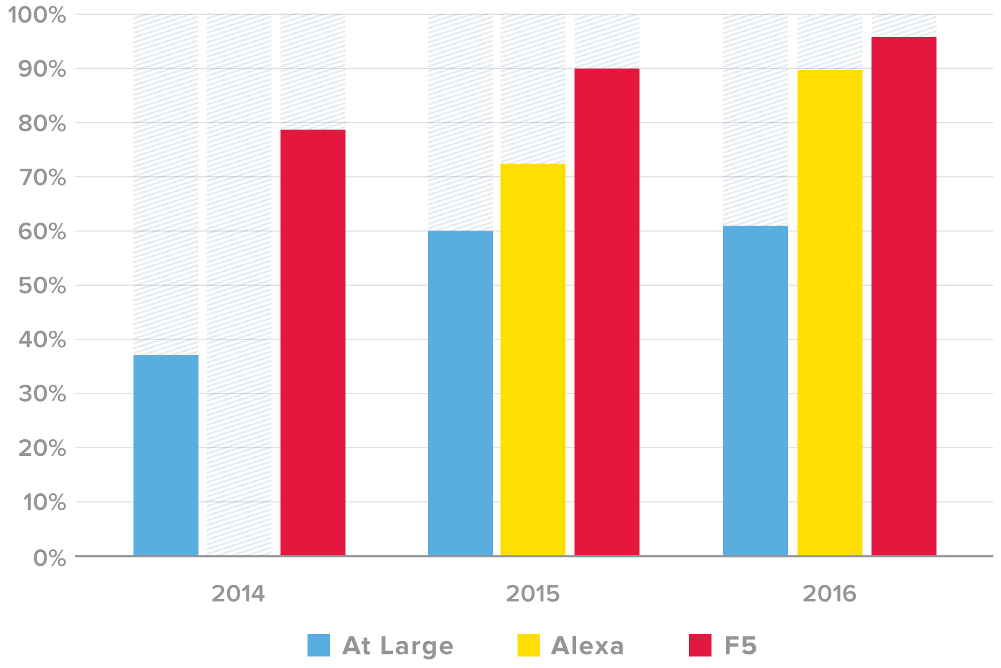


Figure 2: Preference for TLS 1.2 grows in just two years

In the summer of 2014, approximately four out of five F5 devices preferred TLS 1.2. Today, nearly 97% do. Among Alexa hosts (which F5 did not initially differentiate), preference for TLS 1.2 was 73% in 2015 and has now risen to 90% in 2016.

Like the Internet-at-large, the remaining preference is for TLS 1.0.

Since TLS 1.2 is the current standard, the preference for it represents the broadest and most easily understood measure of transport layer security on the Internet today.

SSLV3 DROPPED LIKE IT'S HOT

SSLv3 was actually the very first usable, decently secure transport layer encryption protocol. (SSLv2 had broken cryptographic handshakes and was quickly replaced with SSLv3. And no one ever met SSL version 1 except for a guy at Wing Shack who also says that the moon landing was faked).

SSLv3 had longevity; it survived from the start of the browser age (circa 1990) until the fourth quarter of 2014. In fact, it was so ubiquitous at the start of our project in 2014 that the F5 Labs researchers didn't even bother to measure it. However, Qualys SSL Labs had been tracking its support, and it was 98% in 2014. But, in the spring of 2015, version 3 was mauled by the POODLE cryptographic attack[3]. And although the POODLE attack was never seen in the wild, it was demonstrable enough to cause the Internet to react quickly. Within a single quarter, nearly a third of the Internet disabled support for SSLv3.

The yellow line in Figure 3 represents F5 devices found in the sample. By Q3 of 2016, only about one in eight F5 devices even supported SSLv3. Among Alexa hosts, at the end of 2016, less than 0.01% of hosts supported SSLv3, which is far below the Internet at large (23%).

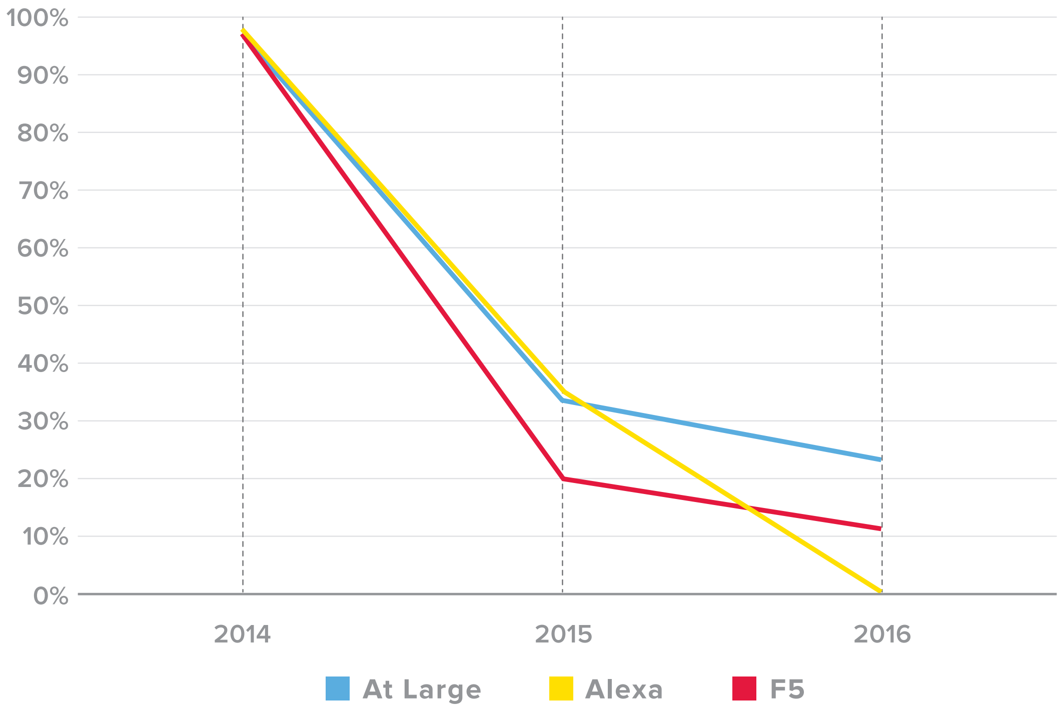


Figure 3: Support for SSLv3 fell sharply as a result of POODLE

[3] https://en.wikipedia.org/wiki/POODLE

SELF-SIGNED CERTIFICATES ARE STILL POLLUTING THE SECURITY WORLD

One of the fundamental principles of TLS—and, in fact, of any asymmetric cryptographic system—is that the public cryptographic key should be signed by a trusted third party. Typically, this is a certificate authority such as Verisign, Comodo, GlobalSign, or Let's Encrypt. Without a trusted signature, a client cannot tell for certain that it is talking to the site it believes it's talking to.

However, an astonishing percentage of Internet hosts that terminate TLS do not, in fact, have a signed certificate. They merely present a certificate wherein the public key has been signed by the associated private key, making verification of the authenticity of the certificate nearly impossible. These are called self-signed certificates, and they're everywhere.

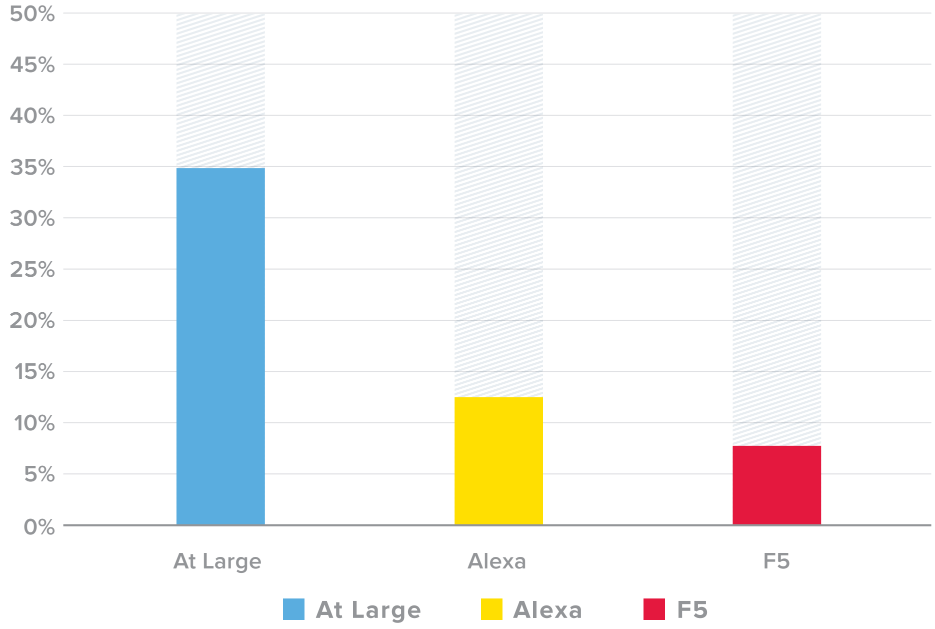


Figure 4: Self-signed certificates as a percentage of population, summer 2016

Prior to 2016, self-signed certificates were a common feature of new devices that had not yet been fully configured. For example, a home router that listens on port 443 for its administrative page (do not do this!) or an internal test site that accidentally got exposed to the Internet. In fact, any website with a self-signed certificate is considered a likely candidate for something that should not be on the Internet in the first place! The problem prior to 2016 was that SSL certificates were not entirely free and were a total nightmare for a non-technical user to configure (assuming they even understood the need for them).

In 2016, the free, open-source certificate authority Let's Encrypt launched with the goal of making TLS ubiquitous across the Internet. Let's Encrypt certificates, in theory, are slightly easier to configure for server administrators, and can be used by hosting providers to provide free certificates for all their websites (WordPress does this, for example).

In our next annual report, we may be able to measure the effect that Let's Encrypt has had on the busiest sites of the world in its full year of operation. In the meantime, F5 Labs researcher David Holmes writes[4] periodically for Security Week magazine about Let's Encrypt and its continuing effect on Internet security.

[4] www.securityweek.com/how-lets-encrypt-will-challenge-ca-industry

FORWARD SECRECY: TRENDING, OR JUST TRENDY?

In 2013, security contractor Edward Snowden starred in his own version of “National Lampoon’s Russian Vacation.” He brought with him some documents from a certain previous employer that turned the world onto the likelihood that nation states were conducting “broad spectrum surveillance” against its citizens.

Snowden had some good news to go with the bad, though. He was confident that good cryptography, when applied correctly, could keep private data from the nation state’s prying eyes. If a server could keep its private key private, then a nation state couldn’t see the data in transit. Except maybe someday they could. Enter the concept of “forward secrecy.”

Suppose that a nation state is interested in Citizen X. Citizen X is communicating with a server in a foreign country, and that server has a 2048-bit RSA key. Such a key would take the nation state hundreds of years to crack, so instead, suppose the nation state records every TLS session between the server and Citizen X. In the future, the nation state is able to send an operative to the server and recover its key (by duplicity, or theft, or a plain old vulnerability like Heartbleed). The nation state could then decrypt all of the saved recordings of Citizen X.

Forward secrecy to the rescue! Sometimes called Perfect Forward Secrecy (PFS), this is an extra encryption step that uses an additional round of key exchange to ensure that only the two parties (in our example, Citizen X and the foreign server) at the time of the transmission can decrypt the session. A third party, even with access to the private key, cannot decrypt the session.

Figure 5 shows that the preference for forward secrecy has increased from one-third to two-thirds of all sampled data in the two years of the F5 Labs scan. That’s quite a jump, though it seems to be leveling off.

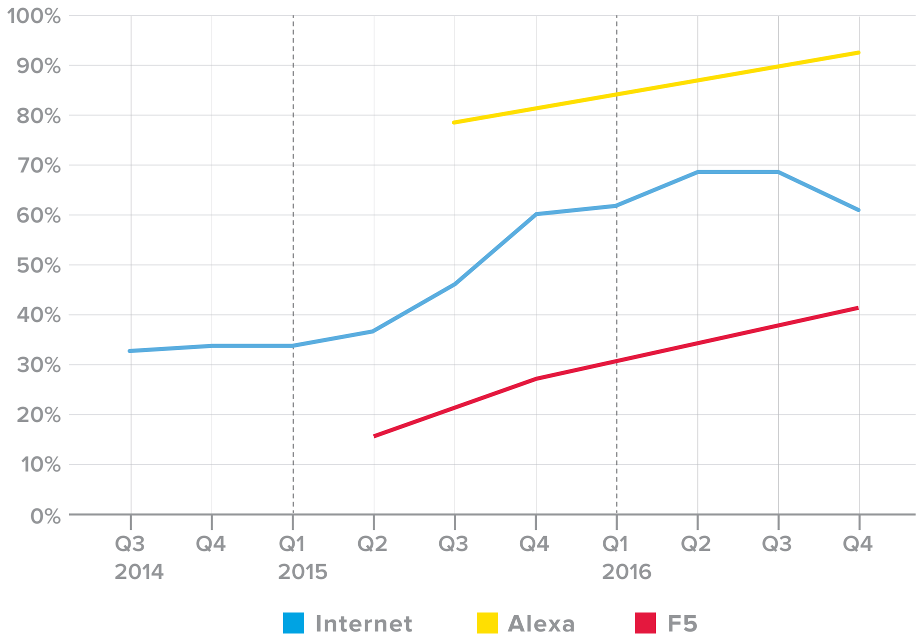


Figure 5: Forward secrecy is on the rise

FLAVORS OF FORWARD SECRECY

There are two different kinds of forward secrecy in use today. One is the ephemeral Diffie-Helman (DHE or sometimes EDH) key exchange algorithm. You can tell if a server is using DHE by examining the server’s cipher string from a browser or other TLS client such as the OpenSSL s_client test tool.

DHE-RSA-AES256-SHA

The other is the elliptic curve version of the same, or the Elliptic Curve Diffie-Helman Ephemeral (ECDHE) key exchange. Elliptic curves require shorter key lengths and therefore use less CPU and memory to achieve the same level of security as RSA keys. From the server, you'd see something like this:

ECDHE-RSA-AES256-SHA

On the Internet at large, in 2014, servers that preferred DHE outnumbered servers that preferred ECDHE by 57% to 43% respectively. However, in 2016, ECDHE is vastly preferred by servers (87% ECDHE versus 13% DHE).

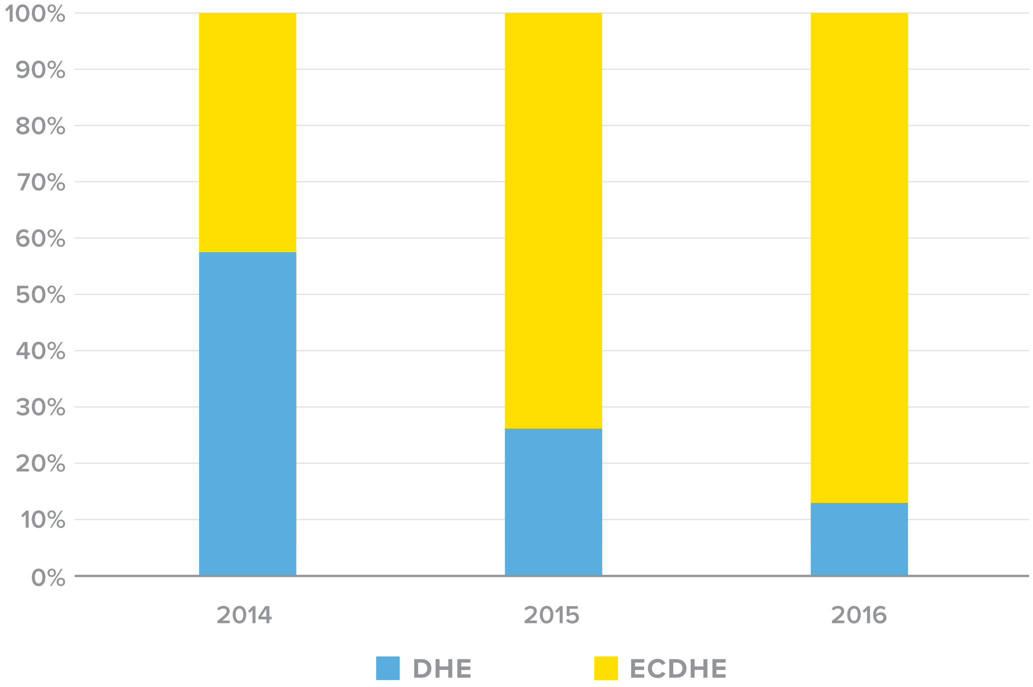


Figure 6: Preference for forward secrecy

LITTLE MOVEMENT ON HTTP STRICT TRANSPORT SECURITY

The absolute easiest way to hijack someone's browser session is to prevent them from using TLS in the first place. The quintessential proof-of-concept tool sslstrip[5] is a man-in-the-middle tool that prevents a browser from following redirects from port 80 to port 443.

Imagine a non-technical user, Joe, who takes his Dell laptop to the corner coffee shop. Little does Joe know that Eve is running a rogue access point at the coffee shop and intercepting his traffic. Joe types mybank.com into his browser address bar. (Who takes the time to type https://? Not Joe.) The default protocol is of course, HTTP, so his browser connects using port 80. Eve intercepts the request and forwards it on to mybank.com. Mybank.com sends back a redirect to https://www.mybank.com/login, but Eve intercepts that, too, using sslstrip and changes the HTTPS to HTTP. Joe fills out the login page over HTTP and Eve intercepts his password. Joe's browser never displayed a warning.

[5] https://moxie.org/software/sslstrip/

In response to sslstrip, the IETF community drafted RFC6797, HTTP Strict Transport Security (HSTS)[6]. HSTS addresses the sslstrip problem with one of the easiest, most elegant little fixes in recent memory: a simple HTTP header in the server response. When a site implements HSTS, the server host redirects the user to the HTTPS version of the site.

Once the secure connection is established, it sends back a header that looks like this:

```
HTTP/1.1 200 OK
Content-Length: 667
Server: Microsoft IIS/7.1
Strict-Transport-Security: max-age=31536000; includeSubdomains
X-Frame-Options: SAMEORIGIN
```

The Strict-Transport-Security header instructs the browser to always use HTTPS instead of HTTP until the max-age time value, even if the user tries to specify the http:// prefix (or uses a bookmark that does the same). This means that as long as Joe has visited the site at least once before and received the HSTS header, he can just type the site name in his browser address bar and the browser will automatically use TLS to protect the session.

Super cool and easy-peasy.

But, if it's so easy, why isn't HSTS being used?

According to the SSL Pulse database, the HSTS adoption rate has been stuck at less than 5% since its inception over three years ago. Our data show even less—globally, less than 2% of sites use HSTS.

There are some legitimate reasons for the low adoption rate of HSTS.

- **Subdomains**. HSTS is most effective when it is applied to all subdomains. But many subdomains contain legacy applications that aren't ready for TLS, and broadly applying HSTS would blackhole those applications.
- **Advertising networks**. Ad networks have their own outdated set of security issues, one of them being that many do not support TLS at all. Ad-supported sites can't afford to drop their ad networks just for HSTS.
- **Hosting networks**. Most websites on the Internet are not running on a single, dedicated server; they're being hosted by a giant hosting provider that support hundreds of sites. Most hosting providers don't include free TLS (though some do). So hosting providers can't blanketly adopt HSTS either, because it would blackhole their mom-and-pop plain vanilla WordPress cupcake store websites.

The Open Web Application Security Project (OWASP) has promoted insecure session management to number two in their OWASP Top Ten project[7]. That's a good start.

[6] https://tools.ietf.org/html/rfc6797

[7] https://www.owasp.org/index.php/Top_10_2013-A2-Broken_Authentication_and_Session_Management

There is some good news, however. The Google Chrome browser now shares HSTS data with all of its users. So, in the example above, Chrome would already have known (from prior users) that mybank.com used HSTS and would never have even tried HTTP. This is the preload feature of HSTS. Other browsers are implementing this as well.

A quick check of the busiest sites in the world shows that four of them have adopted HSTS. Administrators can read more about HSTS at the OWASP HSTS page[8], and can learn how to test their site for HSTS usage at the OWASP HSTS Test page[9]. For those interested in further reading about HSTS, Scott Helme has a great write-up[10] at the securityheaders.io project.

RANK	HOST	OCSP STAPLING	HSTS
1	google.com	No	No
3	facebook.com	No	Yes
4	baidu.com	No	No
5	wikipedia.org	Yes	Yes
6	yahoo.com	Yes	No
8	amazon.com	Yes	No
10	taobao.com	No	No
12	live.com	Yes	Yes
14	twitter.com	No	Yes
17	instagram.com	No	No

Figure 7: Top Alexa Sites and HSTS support

AES CIPHER DOMINATES, BUT STREAM CIPHERS HAVE THE LAST LAUGH

SSL and TLS have two primary encryption modes. The initial handshake of the protocols is done with an asymmetric cryptographic algorithm like RSA, DSA, or an elliptic curve variant. After the client and server complete the handshake, they agree on a symmetric key that will then be used to encrypt and decrypt the records of data that will follow. Asymmetric algorithms are computationally expensive and symmetric algorithms are cheap.

In 2001, the National Institute of Standards and Technology (NIST) chose the Rijndael cipher as the official symmetric cipher to replace the old IBM symmetric cipher, the Data Encryption Standard (DES). At its debut, Rijndael became the Advanced Encryption Standard (AES).

[8] https://www.owasp.org/index.php/HTTP_Strict_Transport_Security

[9] https://www.owasp.org/index.php/Test_HTTP_Strict_Transport_Security_(OTG-CONFIG-007)

[10] https://scotthelme.co.uk/hsts-the-missing-link-in-tls/

Like DES, AES is a block cipher, meaning that the algorithm operates on fixed blocks of data. DES operated on 64-bit (8-byte) data blocks and AES operates on 128 bits (16 bytes). AES claims to be fast in both hardware and software and has no known practical attacks against it, even after 15 years of being the standard.

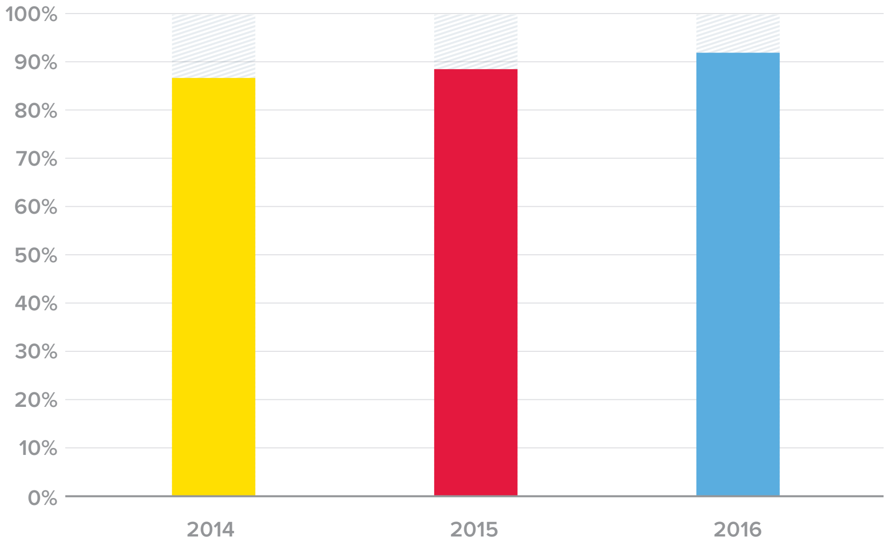


Figure 8: The Dominance of AES as a symmetric cipher

Even at the start of the F5 Labs TLS scanning project in 2014, AES already dominated the Internet's TLS hosts cipher lists. Today over 92% of preferred ciphers are based on AES.

However, F5 Labs researcher David Holmes has always been sweet on stream ciphers[11], specifically the old RC4 cipher, for its speed and elegance. While AES could claim to be fast, nothing could beat RC4 for performance. Stream ciphers like RC4 would generate a random-ish stream of data, which would then be XORed with the plaintext. Decryption used the exact same stream and the exact same operation: XOR. XOR happens to be among the fastest of all opcodes on any CPU in the last 25 years. Unfortunately, RC4 was found to have several biases that became manifest if keys were not rotated quickly enough.

Eventually, it rightly fell out of favor, and less than 5% of servers prefer it now.

Stream ciphers, though, may be having the last laugh with the rapid adoption of the Galois/Counter Mode (GCM)[12] of operation. Counter modes effectively make block ciphers into stream ciphers, resulting in the best of both worlds. In the winter of 2016, nearly 48% of TLS servers preferred some form of AES-GCM. This is up from only 20% in 2014.

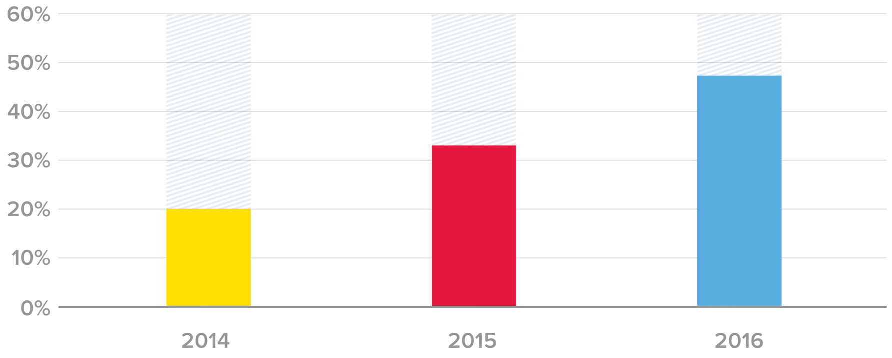


Figure 9: Stream ciphers (via GCM) are gaining momentum

[11] http://www.securityweek.com/memoriam-goodbye-rc4-old-crypto-favorite

[12] https://en.wikipedia.org/wiki/Galois/Counter_Mode

SERVER SMACKDOWN

Mirror, mirror, on the data center wall, which is the most secure web server of them all? By noting the "server" string header among server responses, we can assign some relative scoring. Let's focus on the three most popular web servers which, according to our scanning data, are:

- Apache variants
- NGINX variants
- Microsoft Internet Information Server (IIS)

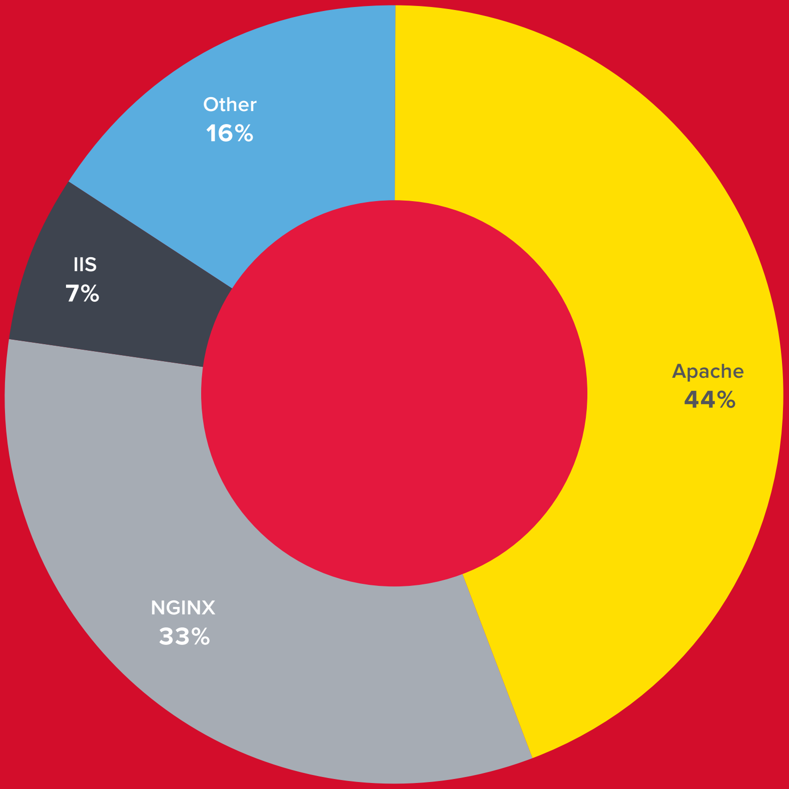


Figure 10:
Chart of server variants –
Alexa Top 1 million sites

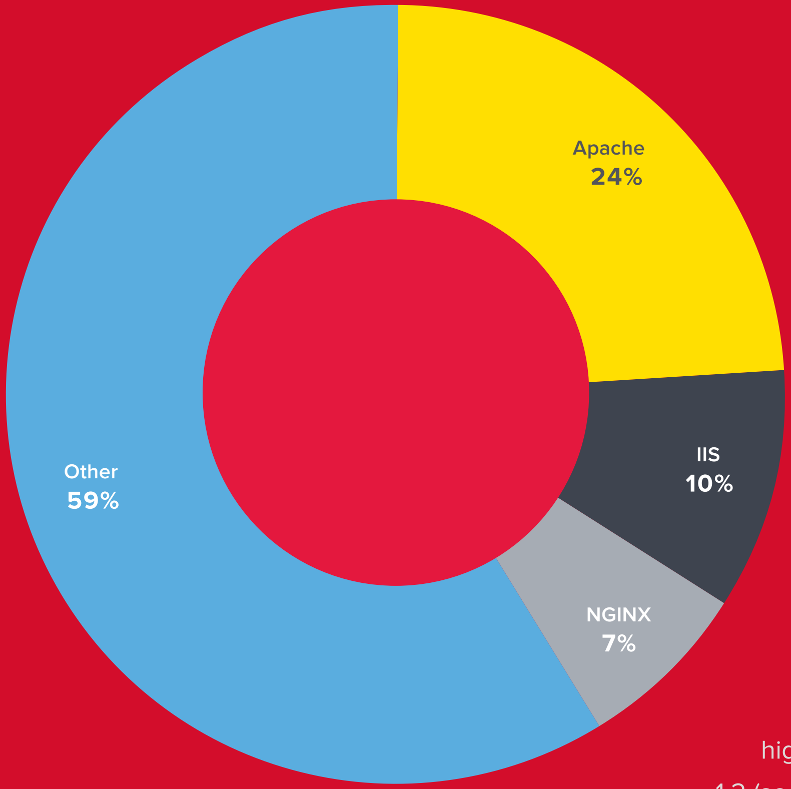


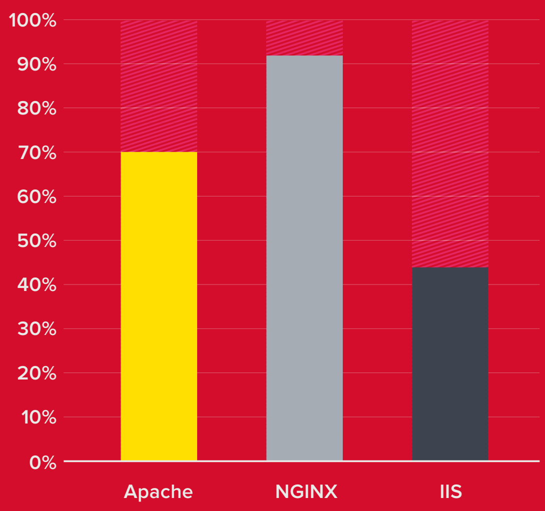
Figure 11: Chart of server variants — Internet at large

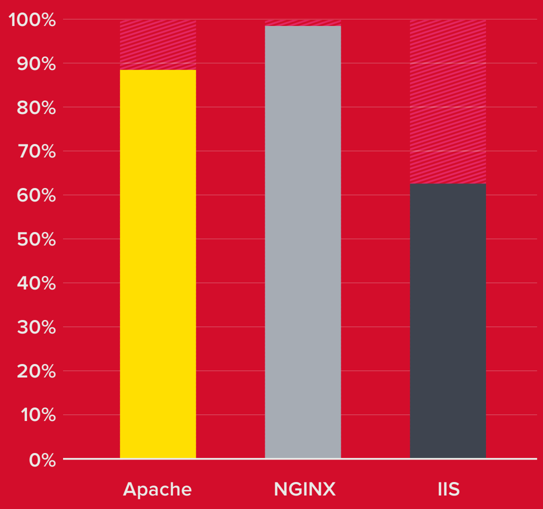
Of course, many of these web servers are front-ended by a high-security device—a content delivery network (CDN) or an application delivery controller—which may or may not be terminating TLS on their behalf. However, the F5 Labs scanner can detect these conditions and remove them from the sample data, so we can be relatively confident that we are looking mostly at web servers that are terminating TLS directly.

Of these most popular servers, NGINX tends to score highest among the cryptographic posture, preferring TLS 1.2 (see Figure 12), supporting forward secrecy for nearly every instance (see Figure 13), and having the highest (though still low) support for HTTP Strict Transport Security (HSTS) (not shown).

(Left) Figure 12:
Server preference for TLS 1.2

(Right) Figure 13:
Server preference for
forward secrecy





Apache scores in the middle, with relatively high preference for TLS 1.2 (see Figure 12) and support for forward secrecy (see Figure 13), and with a majority of its servers disabling SSLv3 (see Figure 14).

At the very bottom in all categories is Microsoft's Internet Information Server (IIS). Only 44% of IIS servers can even speak TLS v1.2 today—far below NGINX (92%) and Apache (69%). Of IIS servers, nearly 60% prefer TLS v1.0 and still support SSLv3 (bad). Perhaps the biggest threat to IIS servers (actually to the server administrators) is this odd threat vector: the Payment Card Industry's Data Security Standard (PCI-DSS) 3.2 compliance. In the summer of 2018, in order to be compliant with the PCI-DSS requirements, payment systems will have to support only TLS v1.2, and nearly 60% of IIS servers today simply aren't there.

The most likely explanation of the Microsoft IIS security posture is that most Windows applications and servers use the underlying Windows OS crypto stack, also known as schannel. For Windows XP and Windows Server 2008[13], schannel supported only version of SSLv3 and TLS v1.0.

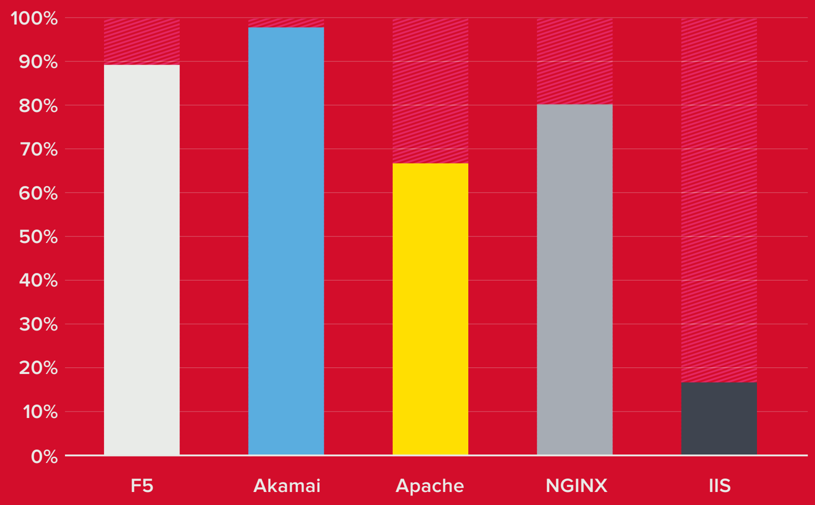


Figure 14: Percentage of servers disabling SSLv3

[13] https://msdn.microsoft.com/en-us/library/windows/desktop/aa380512(v=vs.85).aspx

TLS 1.3 IS SIGNALING A RADICAL CHANGE

At the time of this writing, the most current and official version of the TLS protocol is version 1.2. The IETF committee for TLS has been working on the draft of the next version for several years. This version is likely to be called version 1.3, though it is such a radical departure from the existing 1.2 standard, the final versioning could be different.

TLS 1.3 will include several significant features:

- Only ciphers that support Forward Secrecy will be allowed.
- Only elliptic curve key exchanges will be allowed (deprecating RSA-based key exchanges, though RSA signatures will still be allowed for certificates).
- A zero-round-trip (0RTT) feature whereby TLS clients can send data with the very first packet of the handshake. This feature existed for a time in 2011 as a Google projected call False Start.
- A one-round-trip (1RTT) feature whereby TLS client and servers can resume sessions after only a single exchange of packets.
- Removal of server-based session caches.
- Substantive internal protocol clean up.
- Ciphersuite negotiation replaced with new protocol negotiation mechanism.

Because of the radical nature of these features, adoption of the new protocol may happen more slowly than usual. Hardware manufacturers who create firmware to process TLS will likely wait for the official Internet draft of 1.3 to be completed before finalizing chipset designs. It will be months or even years before those designs to make it into the dense decryption devices that handle much of the traffic for the busiest sites on the Internet.

The F5 Labs research team will begin sampling sites for TLS 1.3 compatibility in the next few scans, and is expecting to report on TLS 1.3 adoption in its next annual report.

RECOMMENDATIONS

Given all the data that we've covered in this report, are there recommendations that can be made for organizations to improve their overall Internet cryptographic posture? Absolutely. Let's look at where the Internet has been falling short and see what can be done to improve things.

- **STOP USING SELF-SIGNED CERTIFICATES!**
- **SUPPORT TLS 1.2 AND GET READY FOR TLS 1.3**
- **CONTRIBUTE TO CERTIFICATE TRANSPARENCY**
- **GET ABOARD THE STRICT TRANSPORT SECURITY TRAIN**
- **ENCOURAGE ADVERTISING NETWORKS TO GET THEIR ACTS TOGETHER**
- **TURN ON OCSP STAPLING**
- **GET GOOD GRADES ON YOUR SITE!**

STOP USING SELF-SIGNED CERTIFICATES!

As stated in the earlier section about self-signed certificates, nearly one in three Internet TLS hosts cannot be verified by a client. There is no good reason for this. Best practices for each self-signed host would be to either remove it from external Internet access, or replace its self-signed certificate replaced with a real, signed certificate. Adoption of the free Let's Encrypt certificate authority removes the cost barrier but replaces it with an automation barrier.

Device manufacturers can build in support for Let's Encrypt automation so that consumers of new devices should be able to retrieve a key and certificate for their device should they consciously decide to put it on the Internet. However, it is very likely that consumers themselves have no idea that their devices are externally addressable.

SUPPORT TLS 1.2 AND GET READY FOR TLS 1.3

Until TLS 1.3 achieves official Internet Draft status, version 1.2 is the best that we can get. Yet nearly half of Internet hosts cannot support it. In an ideal world, SSLv3 support would be miniscule, but because there are so many Windows XP clients still in the wild, sites are tempted to still support it—in fact, 29% in the most recent scan data.

It's recommended that organizations support TLS 1.2+ and drop support for SSLv3.

CONTRIBUTE TO CERTIFICATE TRANSPARENCY

The Certificate Authority (CA) industry is complicated. There are hundreds of CAs, and any of them can issue certificates for any DNS entry on the Internet. Several "rogue certificates" from mismanaged CAs have caused scandals in since 2011, leading to the dissolution of at least CA, DigiNotar.

Three technologies[14] have been proposed to address this challenge. Convergence and TACK were proposed by famed security researcher/cypherpunk Moxie Marlinspike, but neither were broadly adopted by the security community.

[14] http://www.securityweek.com/convergence-replacement-throwdown-dane-vs-tack-vs-ct

The Certificate Transparency (CT) project is the latest effort. CT is sponsored by Google and integrated into the Chrome browser. The CT project's goal:

> *"Broadly speaking, our goal is to provide an open auditing and monitoring system that lets any domain owner or certificate authority (CA) determine whether their certificates have been mistakenly issued or maliciously used."*[15]

CT logs allow browsers to detect rogue certificates and can warn legitimate certificate owners of imposters and fraudsters. CertSpotter[16] is a service that allows for domain monitoring or one-time checking for rogue certificates by accessing the CT logs. The sites crt.sh[17] and censys.io[18] are also good resources. CT logs also allow researchers (like the F5 labs research team) to cross check and verify data for additional TLS research projects.

While it is possible to submit your own certificate information to the CT project, for most enterprise and retail sites it is simply easier to let the certificate authority itself do this for you. Most major CAs today automatically contribute their logs to the CT project and will issue you a certificate with the CT information embedded in an X.509 extension.

GET ABOARD THE STRICT TRANSPORT SECURITY TRAIN

As noted in the previous section on HTTP Strict Transport Security (HSTS), adoption rates are pathetically low. Yes, there are challenges to implementation, but without HSTS, a client can be trivially redirected away from transport layer security provided by TLS. That is, it doesn't really matter how cryptographically secure your website is if your users get diverted before they can even see your certificate.

It's recommended that all new sites employ HSTS from the beginning, and that an annual gap analysis of non-HSTS be performed. Examine all subdomains to make sure they're ready for HTTPS. For those subdomains that aren't, consider deploying an TLS proxy from an ADC. This can be a quick fix that requires few or no back-end changes.

For those that are ready to make the jump, test HSTS for a few days in test and production. You don't want to blackhole your site for six months if you aren't really ready.

Use the Google HSTS portal[19] to check your website for HSTS readiness.

[15] https://www.certificate-transparency.org/faq

[16] https://sslmate.com/certspotter/

[17] https://crt.sh/

[18] https://censys.io/

[19] https://hstspreload.org/

ENCOURAGE ADVERTISING NETWORKS TO GET THEIR ACTS TOGETHER

Interestingly, one of the last holdouts against an encrypted Internet are the advertising networks. High-profile sites work from an advertising revenue model; that is, creating content for eyeballs but getting paid by Internet ads displayed. The advertising networks have been dragging their feet about supporting TLS and until they do, ad-based sites will not be able to turn on HSTS and go to an all-encrypted model.

Google has been encouraging the world to switch to encrypted ad networks with their AdSense network. Business pressure remains, however, because even with AdSense, non-HTTPS ads get removed from the ad auction, resulting in reduced auction pressure, and lower revenues for the site. The path forward for the ad network industry is not entirely clear here; further pressure on the ad industry may be needed.

TURN ON OCSP STAPLING

TLS is the cornerstone of the global public key infrastructure (PKI). If PKI has an Achilles heel, it's certificate revocation. The Online Certificate Status Protocol (OCSP) was meant to be the bandage for PKI's Achilles heel, but it failed in practice for many reasons. But, the advent of OCSP "stapling" makes OCSP relevant again. With OCSP stapling, a TLS termination device can insert a signed copy of the certificate status into the TLS handshake itself, giving the client reassurance that the certificate hasn't been revoked without requiring a separate connection to a questionable OCSP server. It's a win-win for security and user experience.

Turn on OCSP stapling for sites that aren't using short-lived certificates (that is, every site that isn't using Let's Encrypt).

SSLMate has a handy guide[20] to enabling OCSP stapling for Apache and NGINX servers.

GET GOOD GRADES ON YOUR SITE!

The Qualys SSL Labs server test remains one of the best and easiest ways to check your site's cryptographic security posture. The test parameters that determine your site's letter grade change over time. Qualys SSL Labs has recently released the grading changes for 2017[21]. Significant changes to the grading system include:

- Penalty for using 3DES (C)
- Penalty for not using Forward Secrecy (B)
- AEAD suites are required to get a A+

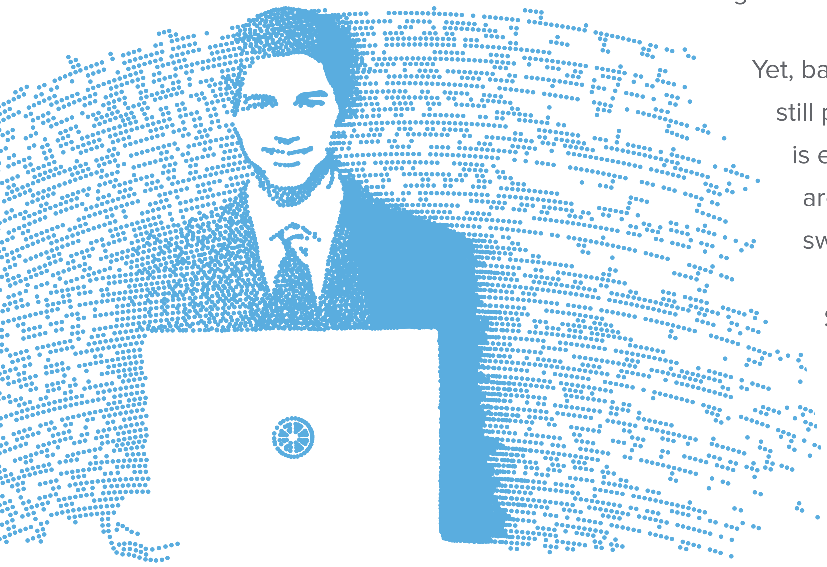
And there are other changes as well. See the Qualys site to prepare your site configuration for the coming year.

[20] https://sslmate.com/blog/post/ocsp_stapling_in_apache_and_nginx

[21] https://blog.qualys.com/ssllabs/2016/11/16/announcing-ssl-labs-grading-changes-for-2017

CONCLUSION

Our analysis from the last two years of scanning data show some good news: broad changes across the Internet in terms of protocol selection, preference for forward secrecy, and rapid deprecation of insecure protocols such as DES and RC4. The incredible adoption of the open source certificate authority Let's Encrypt is spurring encryption for millions of sites and driving the industry forward.



Yet, bastions of insecure sites stubbornly exist. Millions of hosts still present self-signed certificates. Strict transport security is enabled on less than 2% of the other hosts. Ad networks are dragging the most popular sites on Internet back from switching to an all-encrypted world.

Security administrators and DevSecOps teams make the Internet safer with each push of TLS configuration changes. By following the recommendations laid out in this report, they can take the next steps toward a more secure, global encrypted Internet. As that happens, F5 labs researchers hope to measure and trumpet these changes in our upcoming reports.

ABOUT F5 LABS

F5 Labs combines the threat intelligence data we collect with the expertise of our security researchers to provide actionable, global intelligence on current cyber threats—and to identify future trends. We look at everything from threat actors, to the nature and source of attacks, to post-attack analysis of significant incidents to create a comprehensive view of the threat landscape. From the newest malware variants to zero-day exploits and attack trends, F5 Labs is where you'll find the latest insights from F5's threat intelligence team.

For more information, visit: F5Labs.com

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