TROFI SECURITY®
INTELLIGENT INFORMATION SECURITY

HTTPS://

Forget SSL, is TLS dead, too?!

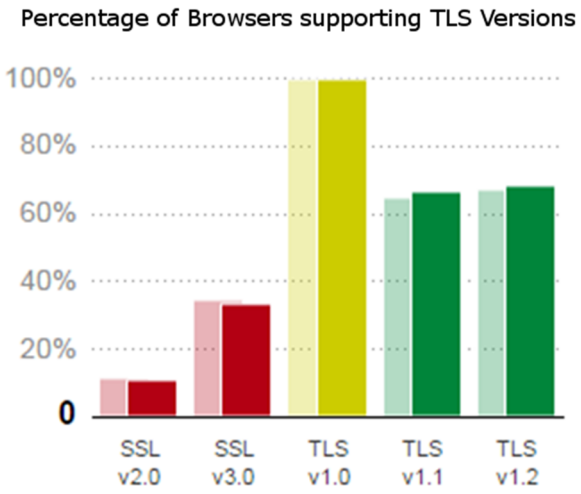
Cybersecurity Challenges | Executive Series

Rod Saunders, CISM | vCISO/Partner at TrofiSecurity
Michael Trofi, CISSP, CISM, CGEIT | vCISO/Founder at TrofiSecurity

A Little History

The **Secure Sockets Layer (SSL)** cryptography protocols, meant to protect the privacy of information exchanged between consumers and websites on the Internet, were originally developed by Netscape during the 1990's, with the first publicly released version, SSLv2.0, being released in December of 1995.[1] SSLv2.0 was quickly followed just 4 months later by SSLv3.0, due to a number of security flaws being discovered in SSLv2.0. SSLv3 was a significant redesign of SSLv2.0 and, arguably, ultimately became one of the technological keystones that helped enable the explosion of e-commerce on the Internet.

Of course, as history has shown us, every technology has its lifespan, and SSLv3.0 is no different. SSLv3.0 was officially supplanted by an updated protocol, **Transport Layer Security (TLS)**, with the release of TLS1.0 in 1999; however, the release of TLS1.0 by no means meant the end-of-life for SSLv3.0. In fact, while vulnerabilities in SSLv3.0 have been known for almost two (2) decades now, and it has been surpassed by two (2) additional revisions to the TLS protocol, TLS1.1 and TLS1.2, the technology industry is only just now beginning to get broad support for ending the use of SSLv3.0.



For example, the **Healthcare Insurance Portability and Accountability Act (HIPAA) Security Rule**, which originally went into full effect in April of 2006, relied on the requirements as specified in **National Institute of Standards and Technology (NIST) Special Publication 800-52**. That publication, originally released in 2005, only more recently deprecated the use of both SSLv3.0 and TLS1.0 as part of its **Revision 1** update, released in April of 2014.

More recently, the payment card industry's **PCI Security Standards Council (PCI SSC)** released PCI DSS v3.1—a "dot" release of PCI DSS 3.0; the version that just went into full effect on January 1, 2015. New PCI DSS versions have historically been released every three (3) years; however, v3.1 was released within only 6 months of the official start of v3.0. While the acceleration of the release process was, in large part, due

[1] "THE SSL PROTOCOL". Netscape Corporation. 2007

to an escalated awareness of a rapidly changing cybersecurity landscape, it was also in direct response to the realization of the breadth of the exposure posed by the continued reliance on SSLv3.0 in the marketplace.

Finally, not to be left behind, the **Internet Engineering Task Force (IETF)** officially deprecated SSLv3.0 in its release of **RFC 7568** in June of this year (2015). Just 9 months shy of its 20th birthday, SSLv3.0 has met its demise, and the virtual world is now a safer place…or, is it?

In reality, that statement couldn't be further from the truth.

First, while regulatory requirements and standards guidance have certainly raised the bar for affected organizations, actual implementations of those requirements and standards are lagging way behind. In fact, according to statistics collected by TrustworthInternet.org, 89.2% of the most popular websites on the Internet today are still vulnerable to the **Browser Exploit Against SSL/TLS (BEAST)** attack[2], which was discovered in 2011 to affect sites supporting TLS1.0. While the vulnerability had been addressed 5 years earlier, with the release of TLS1.1 in 2006, the fact that TLS1.0 remains, even today, as the most prevalently supported TLS version on the Internet, the vast majority of Internet website remain vulnerable to this attack, almost a decade later.

Second, in many cases, there are financial incentives to ***not upgrade*** to the latest regulatory requirements, standards, and security best practices. According to NetMarketShare.com, up to 36% of Internet users are still using browsers, which are not capable of supporting the latest version of TLS (TLS 1.2) out-of-the-box[3]. This leaves many organizations in the situation of having to weigh the pros and cons of upgrading. Do they upgrade, and potentially lose the ability to service what could be a significant percentage of their online client base? Or, do they delay the upgrades in hopes that their client base will update their browsers sooner than hackers decide to take advantage of the situation, or the regulatory agencies, with their limited ability to audit requirements much less enforce them, levy fines?

Either way, adoption has been slow, at best. But, at least it's happening, right? At least organizations are finally at the point of having to make those decisions, and even if they delay a while, once they bring their technology implementations into compliance, ***then*** the virtual world will finally be a safer place.

Or, not. Yes, time for more reality, and a reminder of that oft forgotten, universal truth that "compliance" is by no means the same as "security".

Forget for the moment that technological and financial hurdles to ending the use of SSLv3.0, and even TLS1.0, exist. Assume your every client has a brand new computer and operating system, a fancy new browser, and no limitations with supporting the current regulatory standards. Would ***that*** be enough to make the virtual world a safer place?

[2] https://www.trustworthyinternet.org/ssl-pulse/
[3] https://www.netmarketshare.com/browser-market-share.aspx

The answer is, well…***potentially***, in a perfect world of collaboration and agreement.
More accurately, it's only ***theoretically***, and probably for not much longer, unless new protocols, standards and, most importantly, *disciplines* are adopted.

Using the current NIST SP800-52r1 guidelines, the minimum standard for "secure" communication protocols is TLS1.1, with a strong recommendation for TLS1.2, and a requirement to develop migration plans to TLS1.2 by January of 2015. After 19+ years of SSLv3.0, why the rush to deprecate all but the most current version of the TLS protocol, TLS1.2, in just a couple years?

The answer is as you might expect…***breaches***. Or, at least the ***possibility*** of breaches.

TLS, like SSL, has a useful lifespan determined by its usability and integrity in the marketplace. Like SSL, TLS has been found to suffer from a number of vulnerabilities. In February of 2015, the IETF published a list of known attacks on the TLS protocol in RFC 7457***, which included attacks on TLS1.2***.[4] In fact, on December 8, 2014, it was announced that TLS 1.2 was found to be vulnerable to the very same **Padding Oracle On Downgrade Legacy Encryption (POODLE)** attack that helped hastened the push to end support for SSLv3.0.[5]

While some of the vulnerabilities published by the IETF related directly to vulnerabilities in the underlying cipher suites and modes used to implement the protocols, others related to errors in the various vendor implementations of the protocol, while still others related to the usage of the protocols by applications and systems. Regardless of the root cause, the fact remains that even under "ideal" circumstances, achieving "secure communications" with the current state of existing technologies would seem all but impossible.

So, where does that leave organizations?

If TLS1.2 has been compromised, and TLS1.3 is still only an IETF Draft[6], is there any hope for organizations to ensure information privacy and integrity for consumers and business partners? While not a simple answer, there is, actually, a glimmer of hope; however, it requires organizations, consumers, and technology providers alike, to each do their part; not by any means an easy task, but it is doable, and there is rapidly growing demand in the industry to make this collaboration a reality as quickly as possible.

[4] https://tools.ietf.org/html/rfc7457
[5] https://community.qualys.com/blogs/securitylabs/2014/12/08/poodle-bites-tls
[6] https://tools.ietf.org/html/draft-ietf-tls-tls13-09

A Little Context

The solution lies, first and foremost, in understanding a little more about how the TLS protocols work, and what it really means to say they've been "compromised".

Transport Layer Security (TLS) defines a set of cryptographic **protocols**. This is not to be confused with cryptographic **algorithms**. A protocol defines the requirements and limitations on the methods of secure connection negotiation, communication, and termination. An algorithm, on the other hand, is the actual mathematical computation responsible for encrypting and/or hashing messages and data.

Each TLS protocol supports a number of what are called **cipher suites**, one of which will be negotiated between the Client and Server, depending on the capabilities of each. A cipher suite is a preset combination of cryptographic algorithms used for cryptographic key exchange, bulk encryption, message authentication, and random number generation. Based on the requirements and limitations of the particular TLS protocol, only certain cipher suites meet the specifications to be supported by a given TLS version. Therefore, the security of a given TLS version is dependent upon a combination of the communication methods stipulated by the protocol version, and any inherent limitations in each of the algorithms that comprise each cipher suite.

The **Internet Assigned Numbers Authority (IANA)**, the organization responsible for the global coordination of the DNS Root Zone and IP Address allocations, currently maintains a TLS cipher registry, which contains 321 supported cipher suites. Of all the combinations of TLS versions and cipher suites available, only TLS1.2, using cipher suites supporting the Authentication Encryption with Associated Data (AEAD) compatible message authentication algorithms (i.e. GCM, CWC, OCB, EAX and CCM), have not yet been compromised (at least as of the time of this writing). So, yes, TLS1.2 has been compromised, but so far only when not configured to limit its cipher suite negotiations to one of the above.

A Bastion of Hope…*Discipline*

It would be a nice silver bullet to simply decide to limit TLS version and cipher negotiation to only those versions that have yet to be compromised, but that is not today's reality. As mentioned before, there is still a significant number of consumers on the Internet, whose browsers simply don't support the TLS1.2 cipher suites. For some organizations, this may not be a problem, if they have control over the client browsers. That's fine for an internal network, but it certainly doesn't work for the broader Internet. So how do organizations find that balance between security and limitation? Between privacy and convenience? The answer is **discipline** and **best practices**.

In the ideal scenario, both organizations and consumers would be cognizant of their own technology limitations and vulnerabilities, and address them accordingly. Obviously, this is as far from reality as one can get. (*How much do your parents know about TLS…much less your grandparents?*) But technologies and standards do exist that help in this situation, and organizations around the globe have the ability to control much of it. In particular, the following recommendations go a very long way to striking the balance organizations seek. But it takes discipline and constant vigilance, and it's not for the faint of heart.

#1 | Put SSL, and TLS 1.0, to Rest… *PLEASE!*

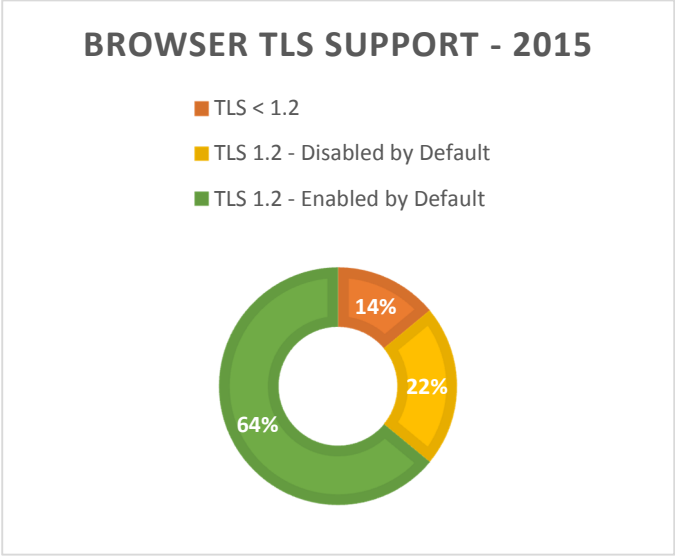
Without a doubt, the long, long overdue first step is to shut SSL down; not just SSLv3, but SSLv2 as well (yes, deprecated in 1996, it's still supported by 10% of websites on the Internet[7]). This goes for TLS1.0 as well, which is currently supported by **99%** of websites today[8]. For many years, the concern was that by shutting down SSLv2/3 and TLS1.0, it would prevent vast numbers of consumers from accessing websites. As of 2015, browsers that support a maximum HTTPS protocol of SSLv3 make up less than 1% of the market; mainly the last straggling IE6 users. Browsers that support a maximum HTTPS protocol of TLS1.0 make up just 7% of the market[9]. While statistically not insignificant, chances are these aren't your primary consumers. At the very least, these are the consumers, who are exposing your organization to far more financial liability than they may be worth.

Why is this important, beyond preventing those folks still using IE6/7 from hurting themselves? **Downgrade attacks.** One of the many ways hackers take advantage of SSLv2/3 or TLS1.0 is by inserting themselves (i.e. Man in the Middle – MITM) between a consumer and organization, and "negotiating" a weaker SSL/TLS version and cipher on a consumer's behalf, in order to ensure they can breach the communication. Even for consumers using the latest and greatest browsers, supporting the latest and greatest TLS version and cipher sites, a downgrade attack may allow a malicious actor to exploit a server's "willingness" to support SSLv2/3 and/or TLS1.0 to compromise the connection.

The bottom line? *At the very least, make SSL (all variants) go away, once and for all! Then take a very hard look as to whether there's really any reasonable justification to keep TLS1.0. Chances are the answer is no.*

#2 | Pick your Minimum TLS version level…*and update it periodically*

If you disable SSLv2/3 and TLS1.0, do you stop there, or keep moving up the TLS chain of versions? This is where the answer gets a little stickier. According to NetMarketShare.com, approximately 14% of the browser market share is made up of browser versions that don't yet support TLS1.2, and an additional 22% of browser versions where TLS1.2 is supported, but disabled by default.[10]



It's a little tougher now, isn't it? Is an organization willing to potentially forgo 14-36% of its online business for the sake of protecting the rest of its users? Again, for

[7] https://www.trustworthyinternet.org/ssl-pulse/
[8] https://www.trustworthyinternet.org/ssl-pulse/
[9] https://www.netmarketshare.com/browser-market-share.aspx
[10] https://www.netmarketshare.com/browser-market-share.aspx

some organizations, who can control the end user's browser, this isn't that big an issue, but for everyone else, this is where the conversation really begins.

This is also not a one-time conversation. The demographics of an organization's web users is changing every day, and the potential vulnerabilities of TLS are changing every day, as well. This is where the discipline really starts to become critical. Any organization concerned with protecting its consumers needs to be on top of this conversation, every single day.

#3 | Disable lower-security cipher suites, order them appropriately…*and update them periodically*

The first suites to disable, if they're supported by your systems, are any that support authentication-only (i.e. no encryption - eNULL) and any that support no-authentication (aNULL). Yes, believe it or not, there are servers configured on the web to allow negotiation to a cipher suite in which none of the data is encrypted. Go figure.

Also, be certain to disable any algorithms using the DES, RC4, or MD5 algorithms. These algorithms were long ago found to provide very minimal security; either due to vulnerabilities with the algorithm itself, or due to the fact that modern processors are now fast enough to brute-force them with relative ease.

The following list is the list of ciphers supported by **TrofiSecurity.com**, and in the order of priority preferred, at the time of this writing. That's not at all to say it should be the list, or order, any other organization should select, but hopefully it opens the discussion. And don't forget to have that discussion on a regular basis. New vulnerabilities surface every day.

Priority	Key Exchange Algorithm	Encryption Algorithm
1	Elliptic Curve Diffie–Hellman (ECDH)	AES in Galois Counter Mode (AESGCM)
2	Diffie–Hellman (DH)	AES in Galois Counter Mode (AESGCM)
3	Elliptic Curve Diffie–Hellman (ECDH)	AES-256 (AES256)
4	Diffie–Hellman (DH)	AES-256 (AES256)
5	Elliptic Curve Diffie–Hellman (ECDH)	AES-128 (AES128)
6	Diffie–Hellman (DH)	128 or 256 bit AES (AES)
7	Elliptic Curve Diffie–Hellman (ECDH)	Triple DES (3DES)
8	Diffie–Hellman (DH)	Triple DES (3DES)
9	RSA	AES in Galois/Counter Mode (AESGCM)
10	RSA	AES
11	RSA	Triple DES (3DES)

#4 | Support the TLS_FALLBACK_SCSV cipher suite

Obviously, any selection of minimum TLS versions and/or specific cipher suites will have the effect of preventing some percentage of users from accessing an organization's systems, since many older browsers will not support the TLS or cipher suite versions. Luckily, this doesn't have to necessarily be an all or nothing situation. In fact, a number of organizations have found that by supporting a little known cipher suite, called **TLS_FALLBACK_SCSV**, they have been able to facilitate an upgrade path for their users, and for their systems.

In reality, TLS_FALLBACK_SCSV isn't actually a cipher suite at all. The SCSV stands for Signaling Cipher Suite Value, and is used by browsers to 'signal' the server not to allow a cipher suite negotiation to happen, if it means the resulting cipher suite chosen would be less secure than the highest security cipher suite supported by the browser. Clearly, this is intended to defend against downgrade attacks.

Great idea, heh? Well, not everyone agrees. In fact, noticeably absent from the list of proponents is Microsoft, who has thus far refused to implement support for it in either IE or IIS, stating that it's a band-aid solution to a much larger issue with how server/product providers handle underlying TLS negotiation.

To be fair, that's a true statement; however, sometimes it's better to have a band-aid than no solution at all. So, for Microsoft shops, skip this step for now. For absolutely everyone else in the world…be sure to upgrade your servers to a version that supports TLS_FALLBACK_SCSV.

The biggest benefit to organizations, is they can ensure that their consumers, who have upgraded their browsers, can be guaranteed the ability to take advantage of more modern and secure cipher suites, while still allowing older browser users an option to access the site, in the most secure manner they can. It doesn't guarantee an older browser user's security, if their browsers only support TLS & cipher suite versions that have been shown to be vulnerable to compromise, but it ensures all users are as secure as they can be when accessing an organization's systems.

#5 | Support HTTP Strict Transport Security (HSTS)

While only more recently supported by the various browser flavors, when a website responds to a client's browser with the Strict Transport Security Header in the response, it notifies supporting browsers that the website will only allow communication over HTTPS, and not to allow any redirects that would attempt to make the browser access the website with unencrypted HTTP.

Simple enough, but it does require that organizations pay close attention to how they've architected their applications, including requiring Secure Cookies, and ensuring that any/all subdomains in use by a site support and require HTTPS as well; ideally using their own TLS Certificates. It also requires end users to do their part in upgrading to more current browsers. Regardless, it's a very worthwhile exercise to not only understand the ins and outs of supporting HSTS, but to understand how the organization's applications integrate (or not) with this capability.

#6 | Do all of the above…*Universally*

One of the largest vulnerabilities organizations often overlook is not implementing security standards consistently across all systems. Specific to SSL/TLS, this happens for the vast majority of organizations when they decide (or fail to consider) their marketing website. More often than not, the justification for not implementing HTTPS on a marketing site is that the site is "static" and/or "doesn't accept private data". This may be true, but how many organizations also put a "login" link on their marketing site, as a convenience for their customers?

Non-HTTPS sites have 2 major drawbacks: the sites owners cannot be validated, and hackers, who manage to establish a Man-in-the-Middle position, can intercept and/or alter website data before it gets to a consumer's browser. That being the case, what guarantees can a non-HTTPS site make about the

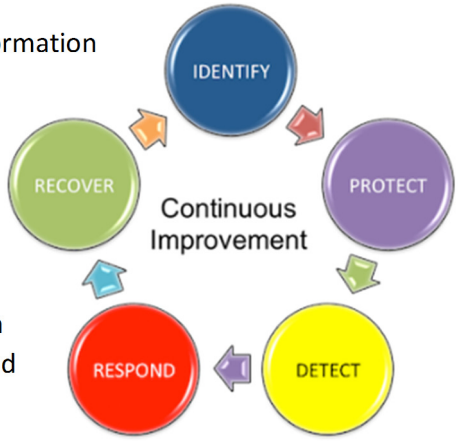
validity of that login link, or the page to which it links, if a hacker may have been able to alter the link? The answer is ***none***.

Organizations need to be cognizant that confidentiality and integrity of online systems depend upon controlling the entire lifecycle of a consumer's interaction with the organization. This requires securing all online points-of-presence, not just the systems that accept confidential data.

#7 | Make Information Security a Critical, Daily Focus

If it's not obvious already, making informed decisions about the above items is not easy. There are any number of considerations to be reviewed and debated for any organization. The only way to be successful at managing this never-ending challenge is to apply specific resources to it, and integrate those resources into the core business strategy of the organization.

Ideally, the topics above would be discussed as part of a broader information security strategy and plan; not one that solely focuses on an organization's website, but one that considers all information security risks to an organization. Don't have a strategy? Well, now is a good time to start working on one, and it should start with an enterprise risk assessment. The sooner an organization is able to understand their risk, and integrate strategies into the organization to address them, the sooner that organization will begin making the steps necessary to ensure the confidentiality, integrity and availability of its systems, and to protect its consumers.



After all, shouldn't that be the goal of all organizations on the Internet?

About Trofi Security

Trofi Security, originally Trofi Systems Solutions, was founded in 1999 to provide IT security advisory and compliance services to client organizations, as well as to serve as a security-community contributor in the development of cross-industry security best practices. Trofi Security's methodologies and expertise have been used successfully on more than 1000 projects, nationally, during its 15 year history.

Trofi Security has grown to become one of the preeminent thought-leaders and advisories in the Information Security industry, across a broad range of client industries. With offices in Colorado, Washington D.C., Rhode Island, and Los Angeles Trofi Security has a national capability to help client organizations evaluate, strategize, develop, and test comprehensive information security programs designed to address their specific risk and regulatory environments.

For more information about Trofi Security visit us online:

trofisecurity.com | linkedin.com/company/trofisecurity | @trofisecurity