Transport Layer Security (TLS/SSL) 1.0 & 1.1 and Cipher Suite Deprecation

What is happening?

Effective *June 27, 2018* Surescripts will no longer support all versions prior to TLS 1.2 (including TLS 1.0 and TLS 1.1) over HTTPS to and from Surescripts related domains. Any older browsers or API clients that do not support TLS 1.2 will no longer work. This change is recommended by various standards bodies and to best protect participant/patient data and affects all Participants connecting to and from Surescripts via TLS (HTTPS).

Participants must use TLS 1.2 and associated strong cipher suites by *June 27, 2018* to continue exchanging transactions with Surescripts.

Failure to migrate to TLS v1.2 with strong ciphers will result in Participant transactions failing.

What is TLS?

Transport Layer Security (TLS) is a protocol that ensures privacy between communicating applications and their users on the Internet. When a server and client communicate, TLS ensures that no third-party may eavesdrop or tamper with any message. Transport Layer Security (TLS) is the successor to Secure Sockets Layer (SSL) and is the protocol that enables secure "https://" connections to websites. Websites use TLS to secure all communications between their servers and clients. TLS 1.2 was defined in RFC 5246 in August 2008 and is the most secure version of SSL/TLS protocol at this time. Surescripts currently supports TLS 1.2 in Production and Non-Production environments and has done so since 2012.

Why is Surescripts Requiring TLS 1.2 Only?

The revision and deprecation of security related protocols like TLS 1.0 and 1.1 are to be expected as encryption techniques evolve/improve and processing speeds increase over time. With increasing attacks, it is recommended by various standards bodies that secure connections be migrated to TLS 1.2. One of the benefit is that TLS 1.2 expands support for authenticated encryption ciphers with AES-GCM cipher suites that are not prone to these attacks. This deprecation notice is for the security of both Surescripts and Participants. Participants who have kept up with the latest developments are already very secure, but Participants who have not kept up to date could end up using a less secure methods.

What are cipher suites?

Cipher suite is a named combination of authentication, encryption, message authentication code (MAC) and key exchange algorithms used to negotiate the security settings for a network connection using the Transport Layer Security (TLS) network protocol.

X.509 (SSL) Root CA Certificate Requirement for TLS 1.2

X.509 server and client certificates issued by approved 3rd Party Public CA (other than Surescripts Public CA) must have a Certification Path/Trust Chain to Root CA Certificate that was signed with a signature algorithm of SHA-1 or preferably SHA-2/256. Any Root CA that was signed with MD5 may not function or may cause interoperability issues and should not be used.

(Most Public CA will issue new certificate for same FQDN/CN at no cost for the remaining validity if the change is related to signature algorithm.)

(Server, Intermediate and root CA certificates should be signed SHA-2/256.)

How do I know this change affects my organization?

Testing outbound to Surescripts

You can connect via browser or application to the following test pages in Surescripts Non-Production to see if you are able to connect outbound using TLS v1.2 with strong cipher suite.

https://staging.surescripts.net/checktls

https://switch-cert01.surescripts.net/checktls (or https://switch-cert01.rxhub.net/checktls)

https://messaging.surescripts.net/checktls

https://switch.surescripts.net/checktls (or https://switch.rxhub.net/checktls)

https://admin.surescripts.net/checktls

When using a browser that does a HTTP GET or when using an application that does HTTP POST, you will get a result in the browser or in the response payload show TLS version and negotiated cipher.

Example:

```
You are connecting from Source IP 107.1.199.132 using TLSv1.2 and Cipher
name=ECDHE-RSA-AES256-GCM-SHA384:256
```

(If you not using TLS v1.2 the actual output may say TLS v1.0 or v1.1 and any number of ciphers)

To test inbound to participant:

OpenSSL (https://www.openssl.org/)

An example to check TLS 1.2 using OpenSSL:

```
openssl s_client -connect www.yourdomain.com:443 -tls1_2
```

You should shake and get a certificate chain.   This will at least show if TLS 1.2 inbound to your system is supported. Further review of ciphers should be done to see what is supported and preference. If you received and error like "handshake error" your system either not support TLS 1.2 or is not configured correctly for TLS 1.2

https://www.owasp.org/index.php/Testing_for_SSL-TLS_(OWASP-CM-001)

Participants with third-party or custom written applications who rely on older libraries may be affected by this TLS upgrade - see below. (Check your vendor's website or contact your vendor for assistance in verifying that you will not be impacted by this upgrade.)

Participants that utilize older browser versions for UI may also be affected by the TLS upgrade - see below.

What Cipher suites are supported and not by Surescripts?

Supported TLS 1.2 Cipher Suites Inbound to Surescripts.

Cipher Suite (hex value)	Bits	Protocols	Key Exchange	Authentication	Cipher	MAC
ECDHE-RSA-AES256-GCM-SHA384 (0xc030)	256	TLS1.2	ECDHE	RSA	AES-GCM	SHA384
ECDHE-RSA-AES256-SHA384 (0xc028)	256	TLS1.2	ECDHE	RSA	AES	SHA384
ECDHE-RSA-AES256-CBC-SHA (0xc014)	256	TLS1.2	ECDHE	RSA	AES	SHA
AES256-GCM-SHA384 (0x9d)	256	TLS1.2	RSA	RSA	AES-GCM	SHA384
AES256-SHA256 (0x3d)	256	TLS1.2	RSA	RSA	AES	SHA256
AES256-SHA (0x35)	256	TLS1.2	RSA	RSA	AES	SHA
ECDHE-RSA-AES128-GCM-SHA256 (0xc02f)	128	TLS1.2	ECDHE	RSA	AES-GCM	SHA256
ECDHE-RSA-AES128-SHA256 (0xc027)	128	TLS1.2	ECDHE	RSA	AES	SHA256
ECDHE-RSA-AES128-CBC-SHA (0xc013)	128	TLS1.2	ECDHE	RSA	AES	SHA

AES128-GCM-SHA256 (0x9c)	128	TLS1.2	RSA	RSA	AES-GCM	SHA256
AES128-SHA256 (0x3c)	128	TLS1.2	RSA	RSA	AES	SHA256
AES128-SHA (0x2f)	128	TLS1.2	RSA	RSA	AES	SHA

(The listed ciphers could be removed and additional strong Cipher Suites added in the future)

Although the TLS 1.2 standards supports more cipher suites than listed above, Surescripts will be only be supporting more secure TLS cipher suites, and disabling support for other cipher suites that are not necessary for interoperability. For example, Surescripts will not support weaker, “Export-Grade” cryptography. (See examples of not supported below)

Not Supported Inbound to Surescripts or Outbound from Surescripts

Cipher Suite (hex value)	Bits	Protocols	Key Exchange	Authentication	Cipher	MAC
DHE-RSA-AES256-GCM-SHA384 (0x9f)	256	TLS1.2	EDH	RSA	AES-GCM	SHA384*
DHE-RSA-AES256-SHA256 (0x6b)	256	TLS1.2	EDH	RSA	AES	SHA256
DHE-RSA-AES256-SHA (0x39)	256	TLS1.2	EDH	RSA	AES	SHA
DHE-RSA-DES-CBC3-SHA (0x16)	168	TLS1.2	EDH	RSA	DES	SHA
DHE-RSA-AES128-GCM-SHA256 (0x9e)	128	TLS1.2	EDH	RSA	AES-GCM	SHA256*
DHE-RSA-AES128-SHA256 (0x67)	128	TLS1.2	EDH	RSA	AES	SHA256
DHE-RSA-AES128-SHA (0x33)	128	TLS1.2	EDH	RSA	AES	SHA
ADH-AES128-GCM-SHA256 (0xa6)	128	TLS1.2	ADH	None	AES-GCM	SHA256
ADH-AES128-SHA (0x34)	128	TLS1.2	ADH	None	AES	SHA
DHE-RSA-DES-CBC-SHA (0x15)	64	TLS1.2	EDH	RSA	DES	SHA
DHE-RSA-CAMELLIA256-SHA (0x88)	256	TLS1.2	EDH	RSA	CAMELLIA	SHA
DHE-RSA-CAMELLIA128-SHA (0x45)	128	TLS1.2	EDH	RSA	CAMELLIA	SHA
ECDHE-RSA-DES-CBC3-SHA (0xc012)	168	TLS1.2	ECDHE	RSA	DES	SHA
DHE-RSA-DES-CBC3-SHA (0x16)	168	TLS1.2	EDH	RSA	DES	SHA
ECDH-RSA-DES-CBC3-SHA (0xc00d)	168	TLS1.2	ECDH	RSA	DES	SHA

DES-CBC3-SHA (0xa)	168	TLS1.2	RSA	RSA	DES	SHA
DHE-RSA-DES-CBC-SHA (0x15)	64	TLS1.2	EDH	RSA	DES	SHA
RC4-SHA (0x5)	128	TLS1.2	RSA	RSA	RC4	SHA

(more ciphers could be added to this not supported list and removed from supported list as existing suites found to be insecure)

Supported TLS 1.2 Cipher Suites Outbound to Participants

Cipher Suite (hex value)	Bits	Protocols	Key Exchange	Authentication	Cipher	MAC
ECDHE-RSA-AES256-SHA384 (0xc028)	256	TLS1.2	ECDHE	RSA	AES	SHA384
ECDHE-RSA-AES256-CBC-SHA (0xc014)	256	TLS1.2	ECDHE	RSA	AES	SHA
AES256-GCM-SHA384 (0x9d)	256	TLS1.2	RSA	RSA	AES-GCM	SHA384
AES256-SHA256 (0x3d)	256	TLS1.2	RSA	RSA	AES	SHA256
AES256-SHA (0x35)	256	TLS1.2	RSA	RSA	AES	SHA
ECDHE-RSA-AES128-SHA256 (0xc027)	128	TLS1.2	ECDHE	RSA	AES	SHA256
ECDHE-RSA-AES128-CBC-SHA (0xc013)	128	TLS1.2	ECDHE	RSA	AES	SHA
AES128-GCM-SHA256 (0x9c)	128	TLS1.2	RSA	RSA	AES-GCM	SHA256
AES128-SHA256 (0x3c)	128	TLS1.2	RSA	RSA	AES	SHA256
AES128-SHA (0x2f)	128	TLS1.2	RSA	RSA	AES	SHA
ECDHE-ECDSA- AES256-GCM-SHA384	256	TLS1.2	ECDHE	ECDSA	AES-GCM	SHA384
ECDHE-ECDSA- AES128-GCM-SHA256	128	TLS1.2	ECDHE	ECDSA	AES-GCM	SHA256
ECDHE-ECDSA- AES256-SHA384	256	TLS1.2	ECDHE	ECDSA	AES	SHA384
ECDHE-ECDSA- AES256-SHA	256	TLS1.2	ECDHE	ECDSA	AES	SHA
ECDHE-ECDSA- AES128-SHA256	128	TLS1.2	ECDHE	ECDSA	AES	SHA256
ECDHE-ECDSA- AES128-SHA	128	TLS1.2	ECDHE	ECDSA	AES	SHA

(listed ciphers could be removed and additional strong Cipher Suites added in the future)

<u>Cipher suites preference</u>

Participants should prioritize cipher suite preference from strongest to weakest – both when acting as the Client (Participant to Surescripts) and the Server (Surescripts to Participant).

Surescripts prioritized cipher suite preference from strongest to weakest – both when acting as the Client and the Server.

If no preference is defined, weaker ciphers which may be faster but less secure could end up being negotiated despite stronger ciphers being available between the client and server.

Future Support for TLS 1.3 (Draft)

At this time Surescripts DOES NOT support TLS v1.3

Any current implementations of TLS v1.3 are based on different draft versions may not /do not interoperate with each other.

In the future Surescripts will look to test and implement support for TLS v1.3 (in addition to current TLS v1.2) once the draft is finalized and mainstream supported by major industry vendors. Please be aware that use of improper use/configuration of TLS 1.3 (client or server) may cause connectivity failures.

Library Support

If you have code that connects with Surescripts, you must ensure that it will continue to work after June 28*, 2018*. Each language and library is different, but we've identified the popular ones that may be of concern.

These library/languages may require significant changes/upgrades in order to support TLS 1.2:

Java before 6u45 / 7u45

.NET before 4.5 (does not support TLS 1.2)

.NET 4.5 (must be have setting changed to explicitly enable TLS 1.2)

OpenSSL before 0.9.8

Most dynamic languages rely on the underlying operating system/distributions' supplied OpenSSL package. Openssl version 1.0.1 or greater is required.

Browser support

Most current browsers have supported TLS 1.2 for several years. However, the following browsers DO NOT support TLS 1.2:

Internet Explorer 10 (by default TLS 1.2 is not enabled, but can be enable via GPO or manually)

Google Chrome 29

Firefox 26

Safari 8

Operating system support

Most current operating systems have supported TLS 1.2 for several years. However, the following operating systems DO NOT support TLS 1.2:

Windows XP & Windows Server 2003 (and prior versions)

Windows Vista & Windows Server 2008** (and prior versions)

[**Windows Server 2008 **R2** does support TLS 1.2**]**

References

https://www.ietf.org/rfc/rfc5246.txt

http://www.nist.gov/itl/csd/tls-043014.cfm

Enabling/Adding TLS 1.2 on browsers

Internet Explorer:

Open Internet Explorer

Click Alt T and select “Internet Options”.

Select the “Advanced” tab.

Scroll down to the “Security” section.

Locate and check “Use TLS 1.2”.

Then, press the “OK” button.

Google Chrome:

Open Google Chrome

Click Alt F and select “Settings”.

Scroll down and select “Show advanced settings…”

Scroll down to the Network section and click on “Change proxy settings…”

Select the “Advanced” tab.

Scroll down to the “Security” section.

Locate and check “Use TLS 1.2”.

Then, press the “OK” button.

FireFox:

Open FireFox

Type in “about:config” in the URL bar and press Enter

Scroll down to “security.tls.version.max” and press enter

Set the value to 3

Then, press the “OK” button.

Opera:

Open Opera

Click Ctrl+F12

Click on “Security”

Click on “Security Protocols…”

Check on “Enable TLS 1.2”

Press the “OK” button.

Then, press the “OK” button.

Safari:

There are no options for enabling SSL protocols. If you are using Safari version 7 or greater, TLS 1.2 is automatically enabled.

(Note that these changes may not be available without local administrative privileges or may be controlled by corporate policies that your system administrators would change.)