ArcGIS Platform SSL/TLS Support and Configuration Briefing

Date: 10/4/18
Version: 2.9

Prepared by:
Esri Software Security & Privacy Team
SoftwareSecurity@esri.com

Introduction

Since POODLE came out in late 2014, marking the death of SSL v3 as a secure protocol, Esri has evolved our support of both SSL & TLS across the ArcGIS Platform to provide you the best security options for your implementations. This article provides general TLS implementation guidance, ArcGIS Platform support, specific ArcGIS product support, and finally standard web server SSL/TLS/Cipher configuration guidance.

General TLS Implementation Guidance

Customers today must balance general accessibility of their applications, compliance demands, utilizing the strongest security TLS version, and choosing the right architecture for TLS support as described below:

Application accessibility

In alignment with current PCI and FedRAMP guidelines, we recommend deprecating usage of TLS 1.0, and since usage of TLS 1.1 is not widespread, most customers should migrate to utilizing only TLS 1.2 as soon as possible.

Architecture for TLS support

In general, application servers should NOT be the front-line for connecting TLS with clients. In other-words, when you are concerned about client communication with ArcGIS Servers, the clients should be establishing their TLS communication with some other web service device in front of ArcGIS (not directly with ArcGIS). This is in-line with the premise of utilizing a DMZ and having web endpoints being from the client to the web server / SSL accelerator located in the DMZ. This architecture configuration decouples TLS client communication concerns from your applications and allows more centralized certificate management and configuration through devices that support SSL acceleration. Yes, the separate web-endpoint could even be utilized with the ArcGIS Web Adaptor to piggyback on the TLS capabilities of your standard web server, instead of dealing with unique application server SSL/TLS restrictions. To be clear, when you jump into the weeds, you will see that in addition to SSL/TLS versions, secure communication with clients is further managed by ciphers which is just the nail in the coffin as to why you should seriously consider NOT terminating your client TLS communication with application servers (such as ArcGIS), but instead standard web servers / load balancers / accelerators.

ArcGIS Platform Support

- SSLv2 is not enabled by default with ArcGIS 10 and later – e.g. Not susceptible to DROWN
- SSLv3 is not enabled by default with ArcGIS 10.3 and later – e.g. Not susceptible to POODLE

Specific ArcGIS Product Support

ArcGIS Online

- Currently supports only TLS 1.0,
- 1.1, and 1.2 (configuration since 2014).
- Starting with the September 2018 release, new organizations will only be able to utilize HTTPS.
- ***Be aware**, with the December 2018 release, TLS 1.0 & 1.1 will be disabled.*
- With the September 2019 release, *ALL* organization only use HTTPS (no HTTP).
- HSTS (HTTP Strict Transport Security) is supported at the organization level in ArcGIS Online. Organizations that allow only HTTPS benefit from HSTS when members are logged into their ArcGIS Online organization. HSTS will be enforces for all ArcGIS Online communications

The following are some clients that we know are unable to use TLS 1.2. Please update your clients to ensure uninterrupted access to the service.

- Android 4.3 and earlier versions
- Firefox version 5.0 and earlier versions
- Internet Explorer 8-10 on Windows 7 and earlier versions
- Internet Explorer 10 on Win Phone 8.0
- Safari 6.0.4/OS X10.8.4 and earlier versions
- ArcPad using Mobile/CE 6.5 and earlier (CE v.7 patch for TLS 1.2 available)

Esri Managed Cloud Services (EMCS) Advanced Plus

- Utilizes only TLS 1.2 by default, but can enable other TLS versions as required by customer.

ArcGIS Enterprise – ArcGIS Server, Portal for ArcGIS, and ArcGIS DataStore

Ideally, by following the architecture for TLS support section of this document above, you are NOT having external clients communicate directly with Esri application servers, therefore the below information is not as critical for the security of your deployment. If you choose otherwise, the below information can be useful for your secure deployment planning efforts.

- SSLv2 – Disabled for ArcGIS 10 and later
- SSLv3 – Disabled for ArcGIS 10.3 and later. Note that ArcGIS Server 10.1SP1 QIP, and 10.2 users can apply the security patch to disable SSLv3
- TLSv1.0 – Enabled for ArcGIS 10 through 10.6 – Starting with 10.6.1, TLS 1.0 will be disabled by default in alignment with PCI and FedRAMP guidelines.
- TLSv1.1 & 1.2 - Enabled for ArcGIS 10.4 and later. Note that users can specify server TLS versions and disable ciphers starting with ArcGIS 10.4
- New installations of ArcGIS Enterprise 10.6.1 disable TLS 1.0 by default. If an existing ArcGIS Enterprise instance is upgraded to version 10.6.1, the previous HTTPS protocol version settings previously configured will persist.
- HSTS (HTTP Strict Transport Security) is supported at the ArcGIS Enterprise tier starting at ArcGIS 10.6.1. For prior versions, HSTS may be implemented by the customer at the web tier. See your web server documentation for instructions for implementing HSTS.

There is currently no documented way to configure the TLS settings (ciphers or TLS versions) for the ArcGIS Data Store's REST API. It is on Esri's roadmap to provide administrators greater control over their encryption settings in Data Store.

At ArcGIS 10.6.1, ArcGIS Data Store was updated to not allow TLS 1.0 and to use strong ciphers.

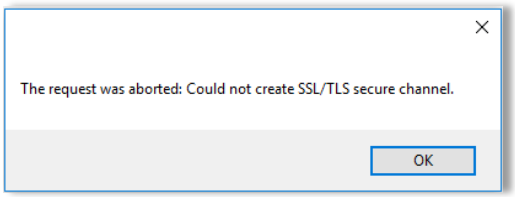
Notes:

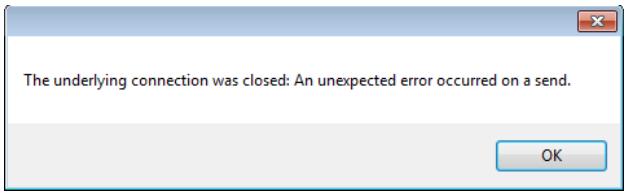
- The Data Store endpoint is NOT intended for end-user connections and the only workflow where a client uses a browser is during the installation and upgrade of the product.
- Esri recommends that port 2443 be firewalled off from users; only the machines running the ArcGIS Data Store and the ArcGIS Server need to be able to access port 2443 for backend communications.
- At ArcGIS 10.5 Esri added logic that causes port 2443 to select the strongest cipher that a client supports - and since the clients are all internal, stronger ciphers and protocols will be used.
- If this guidance is followed, hackers won't be able to use TLS 1.0 vulnerabilities to eavesdrop on internal backend traffic because TLS 1.0 wouldn't be used.

ArcGIS Desktop-based Clients

Using the Add Data button to add data from ArcGIS Online or from Portal for ArcGIS fails with an error by default for versions 10.6 and earlier (works by default with 10.6.1 and later). The Add Data tool contains components built with the Microsoft .Net Framework. Prior to ArcGIS 10.6.1, this tool was built to target the highest TLS version .Net supported at the time the product was released.

The error may read:





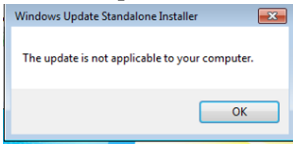
Guidance to address TLS issues with ArcGIS Desktop

Step 1 – OS Patch

If your organization is attempting to run an older version of Desktop on older Windows operating system builds you will need to first install a patch from Microsoft to allow support of TLS 1.2.

- **Windows 7 or Windows 2008 R2 Users - Install this patch**
- **Windows 2012 Users - Install this patch**

If your system already has an appropriate patch or newer .NET version in place, you may receive a prompt indicating the update is not applicable, which is fine, proceed to the next step.



Step 2 – Add Windows Registry Entry

a. ArcGIS Desktop 10.4 – 10.6 users - Click here to download , copy and paste the text into notepad, save the file as ArcMapPost104TLS.reg and then double-click the file to Run and deploy

b. ArcGIS Desktop 10.2 – 10.3.x users – Click here to download , copy and paste the text into notepad, save the file as ArcMapPre104TLS.reg and then double-click the file to Run and deploy

c. Start ArcMap and test.

Notes for Desktop TLS Registry Fix:

1. **Older clients** - ArcGIS Desktop 10.0 – 10.1 can likely follow the same steps as for 10.2-10.3 used, but was not validated
2. **Fallback option** - The registry entries above enable TLS 1.2 as a default for all applications utilizing the relevant .NET version – This should be a desired state for all organizations, but if there are any issues, the user can just remove the registry entries.
3. **Large deployments** - If your organization has a large number of Desktop systems and utilizes Active Directory, the registry entries can be centrally deployed in less than 10 minutes by following the steps here.

Operations Dashboard Windows App

- This app uses TLS 1.0 by default. We recommend migrating to the browser-based version of the app, however if you must continue using the app then use the same strong cryptography resolution (registry entry) as described above for Desktop.

Python

- All versions of Python included with the ArcGIS Platform since 10.0 support TLS.

Browsers

- SSL & TLS support is determined by a combination of client/server negotiation. As a security precaution beyond our products, we recommend customers disable SSLv3 within their browser settings where it is possible for them to do so.

Operating Systems

- TLS support also varies depending on the operating system and version being utilized. Microsoft has a good summary of their TLS support here. For customers requiring FIPS compliance, please check Appendix B of that document for any special requirements.

API (inbound) Integrations

API Integrations are interfaces or applications–including mobile apps and desktop clients–that are separate from the ArcGIS platform, but use ArcGIS data. If you have any API Integrations, please ensure that TLS 1.2 encryption protocols are enabled in those integrations.

Action Required for API (Inbound) Integrations

If your integrations that use inbound connections to ArcGIS do not have TLS 1.2 enabled after we switch to TLS 1.2 only, **your integrations may experience disruption**. We recommend that you begin planning to support TLS 1.2 as soon as possible.

Please refer to the compatibility guidelines below:

Platform or Library	Compatibility Notes
Java (Oracle) Compatible with the most recent version, regardless of operating system	
Java 8 (1.8) and higher	Compatible with TLS 1.2 by default.
Java 7 (1.7)	Enable TLS 1.2 using the https.protocols Java system property for HttpsURLConnection. To enable TLS 1.2 on non-HttpsURLConnection connections, set the enabled protocols on the created SSLSocket and SSLEngine instances within the application source code. Switching to IBM Java may be an effective workaround if upgrading to a newer Oracle Java version isn't feasible.
Java 6 (1.6) and below (publicly available version)	Not compatible with TLS 1.2 or higher encryption. Switching to IBM Java may be an effective workaround if upgrading to a newer Oracle Java version isn't feasible.
Java (IBM)	
Java 8	Compatible with TLS 1.2 by default. You may need to set com.ibm.jsse2.overrideDefaultTLS=true if your application or a library called it by it uses SSLContext.getinstance("TLS").
Java 7 and higher, Java 6.0.1 service refresh 1 (J9 VM2.6)	Enable TLS 1.2 using the https.protocols Java system property for HttpsURLConnection and the com.ibm.jsse2.overrideDefaultProtocol Java system property for SSLSocket and SSLEngine connections, as recommended by IBM's documentation. You may also need to set com.ibm.jsse2.overrideDefaultTLS=true.

and higher, Java 6 service refresh 10 and higher	
.NET Compatible with the most recent version when running in an operating system that supports TLS 1.2.	
.NET 4.6 and higher	Compatible with TLS 1.2 by default.
.NET 4.5 to 4.5.2	.NET 4.5, 4.5.1, and 4.5.2 do not enable TLS 1.2 by default. Two options exist to enable these, as described below. **Option 1:** .NET applications may directly enable TLS 1.2 in their software code by setting System.Net.ServicePointManager.SecurityProtocol to enable SecurityProtocolType.Tls12. The following C# code is an example: System.Net.ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12 | SecurityProtocolType.Tls; **Option 2:** It may be possible to enable TLS 1.2 by default without modifying the source code by setting the SchUseStrongCrypto DWORD value in the following two registry keys to 1, creating them if they don't exist: "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\.NETFramework\v4.0.30319" and "HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\.NETFramework\v4.0.30319". Although the version number in those registry keys is 4.0.30319, the .NET 4.5, 4.5.1, and 4.5.2 frameworks also use these values. Those registry keys, however, will enable TLS 1.2 by default in all installed .NET 4.0, 4.5, 4.5.1, and 4.5.2 applications on that system. It is thus advisable to test this change before deploying it to your production servers. This is also available as a registry import file. These registry values, however, will not affect .NET applications that set the System.Net.ServicePointManager.SecurityProtocol value.
.NET 4.0	.NET 4.0 does not enable TLS 1.2 by default. To enable TLS 1.2 by default, it is possible to install .NET Framework 4.5, or a newer version, and set the SchUseStrongCrypto DWORD value in the following two registry keys to 1, creating them if they don't exist: "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\.NETFramework\v4.0.30319" and "HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\.NETFramework\v4.0.30319". Those registry keys, however, may enable TLS 1.2 by default in all installed .NET 4.0, 4.5, 4.5.1, and 4.5.2 applications on that system. We recommend testing this change before deploying it to your production servers. This is also available as a registry import file. These registry values, however, will not affect .NET applications that set the System.Net.ServicePointManager.SecurityProtocol value.

.NET 3.5 and below	Not compatible with TLS 1.2
Python Compatible with the most recent version when running on an operating system that supports TLS 1.2.	
Python 2.7.9 and higher	Compatible with TLS 1.2 by default.
Python 2.7.8 and below	Not compatible with TLS 1.2
Ruby Compatible with the most recent version when linked to OpenSSL 1.0.1 or higher.	
Ruby 2.0.0	TLS 1.2 is enabled by default when used with OpenSSL 1.0.1 or higher. Using the :TLSv1_2 symbols with an SSLContext's ssl_version helps ensure that TLS 1.0 or earlier is disabled.
Ruby 1.9.3 and below	The :TLSv1_2 symbol does not exist in 1.9.3 and below, but it is possible to patch Ruby to add that symbol and compile Ruby with OpenSSL 1.0.1 or higher.
Microsoft WinINet Compatible with the most recent version.	
Windows Server 2012 R2 and higher Windows 8.1 and higher	Compatible with TLS 1.2 by default.
Windows Server 2008 R2 to 2012 Windows 7 and 8	Compatible by default if Internet Explorer 11 is installed. If Internet Explorer 8, 9, or 10 is installed, then TLS 1.2 will need to get enabled by the user or an administrator for compatibility. Review the Enabling TLS 1.2 in Internet Explorer article to enable TLS 1.2.
Windows Server 2008 and below	Not compatible with TLS 1.2.

Windows Vista and below	
Microsoft Secure Channel (Schannel) Compatible with the most recent version.	
Windows Server 2012 R2 and higher Windows 8.1 and higher	Compatible with TLS 1.2 by default.
Windows Server 2012 Windows 8	TLS 1.2 disabled by default, but is available if enabled by an application. TLS 1.2 can be enabled by default within the registry. Those registry settings are also available as a registry import file.
Windows Server 2008 R2 Windows 7	Compatible by default in client mode when Internet Explorer 11 is installed. If Internet Explorer 11 is not installed or if Salesforce needs to connect to a service running on this type of system, then TLS 1.2 can be enabled by default within the registry. Those registry settings are also available as a registry import file.
Windows Server 2008 and below Windows Vista and below	Not compatible with TLS 1.2.
Microsoft WinHTTP and Webio	
Windows Server 2012 R2 and higher Windows 8.1 and higher	Compatible with TLS 1.2 by default
Windows Server 2008 R2	With KB3140245 applied, Webio is compatible by default, and WinHTTP can be configured via registry settings to enable TLS 1.2.

SP1 and 2012 Windows 7 SP1	
Windows Server 2008 and below Windows Vista and below	Not compatible with TLS 1.2
OpenSSL Compatible with the most recent version, regardless of operating system.	
OpenSSL 1.0.1 and higher	Compatible with TLS 1.2
OpenSSL 1.0.0 and below	Not compatible with TLS 1.2
Mozilla NSS Compatible with the most recent version, regardless of operating system.	
3.15.1 and higher	Compatible with TLS 1.2
3.15 and below	Not compatible with TLS 1.2.

Standard Web Server SSL/TLS/Cipher Configuration Guides:

- Microsoft IIS: https://technet.microsoft.com/en-us/library/security/3009008.aspx
  - If you don't want to get into IIS weeds with the above approach, check out the free IIS Crypto tool instead
- Tomcat Web Server: http://blog.facilelogin.com/2014/10/poodle-attack-and-disabling-ssl-v3-in.html
- Apache Web Server: https://www.digicert.com/ssl-support/apache-disabling-ssl-v3.htm
- IBM WebSphere Application Server: http://www-01.ibm.com/support/docview.wss?uid=swg21687173

Other References

- ArcGIS Server: Restrict SSL protocols and cipher suites
- Portal for ArcGIS: Restrict SSL protocols and cipher suites
- OWASP: Transport Layer Protection Cheat Sheet
- SSLLabs: SSL and TLS deployment best practices
- https://docs.microsoft.com/en-us/dotnet/framework/network-programming/tls
- https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-R2-and-2012/dn786418(v=ws.11)#bkmk_schanneltr_tls12
- https://support.microsoft.com/en-us/help/3154518/support-for-tls-system-default-versions-included-in-the-net-framework
- https://support.microsoft.com/en-us/help/3154519/support-for-tls-system-default-versions-included-in-the-net-framework
- http://desktop.arcgis.com/en/system-requirements/latest/arcgis-desktop-system-requirements.htm

Feedback

We welcome your feedback concerning the information provided within this briefing and any suggestions you may have. Feel free to contact the Software Security & Privacy Team @ SoftwareSecurity@Esri.com