Recommended Protocols

This page is aimed at giving a list of recommended protocols and also the one that we want to avoid.

Introduction

All the communications between the applications should be encrypted, which is a part of the requirement in the CII badging itself. In ONAP we have multiple applications talking to one another. We will try and address all the scenarios. This is a work in progress, so if you think we are missing any scenario then please let us know.

Browser(Webapp)/Rest client

Generally, in the past, we were using SSL as the to go protocol for HTTPS encryption, but because of the discovery of poodle attack all versions of SSL and TLS v.0 are no longer recommended and it is required that we follow TLS

Scenario:

In case of using a rest client or a browser to access a client. Most of the applications will have a rest based API when communicating with an external client.

Recommendations:

Protocols

There are altogether 5 protocols available for SSL and TLS combined.

Protocol	Status
TLS v1.2	Recommended. They support the latest cryptographic algorithms
TLS v1.1	Ok to use as long as the backward compatibility has been turned off.
TLS v1.0	Insecure do not use
SSL v2	Insecure do not use
SSL v1	Insecure do not use

Cipher Suites

For the encryption and decryption to happen in a secure way we have to define what cipher suite we want to cover. The strength of the TLS is directly dependent on the cipher suite that we decide to use. The servers should be configured to disable all week ciphers.

While choosing a cipher for the encryption choose one that supports PFS, A cheat sheet for ciphers can be found in https://www.owasp.org/index.php/TLS_Cipher_String_Cheat_Sheet. In the list, you can use any ciphers on the A+, A or B category.

Additional requirements

Having TLS enabled only on certain pages does not accomplish what was intended. For optimum results, these guidelines[1] should also be followed

Description	WebApp	Rest Client
All pages must be served over HTTPS. This includes CSS, scripts, images, AJAX requests, POST data and third party includes. Failure to do so creates a vector for man-in-the-middle attacks	Yes	NA
Just protecting authenticated pages with HTTPS, is not enough. Once there is one request in HTTP, man-in-the-middle attacks are possible, with the attackers being able to prevent users from reaching the secured pages	Yes	NA
The HTTP Strict Transport Security Header must be used and pre-loaded into browsers. This will instruct compatible browsers to only use HTTPS, even if requested to use HTTP	Yes	Yes
Cookies must be marked as Secure	Yes	NA

Reference

1. https://www.owasp.org/index.php/Transport_Layer_Protection_Cheat_Sheet
2. https://github.com/ssllabs/research/wiki/SSL-and-TLS-Deployment-Best-Practices
3. https://www.owasp.org/index.php/TLS_Cipher_String_Cheat_Sheet