Migrating to a secure version of TLS

Preparing for the June 2018 deadline

David Morris, PCI Compliance Analyst

Secure Sockets Layer (SSL) has been the most widely used protocol for encrypting communications on the Internet. It is used to establish a secure communications channel between two systems on a network. An SSL connection between two systems can authenticate one or both of the systems and ensure the confidentiality and integrity of the data transmitted between them. SSL was originally developed by Netscape in 1994 and has since then become standardised by the Internet Engineering Task Force (IETF). It has undergone several revisions in order to support new cryptographic algorithms and enhance its security. In 1999 it was renamed by the IEFT as Transport Layer Security (TLS). The differences between TLS version 1.0 and the version which immediately preceded it, SSL version 3, are minor. The recommended versions are now TLS version 1.1 and version 1.2, which were released in 2006 and 2008 respectively. TLS version 1.2 is currently the most secure version and should be selected above all of the preceding versions wherever possible.

In April 2014 the National Institute of Standards and Technology (NIST) classed SSL and TLS version 1.0 as unsafe and recommended migrating to TLS version 1.1 or 1.2. The preceding versions, SSL and early TLS, are no longer considered to be effective security controls by the Payment Card Industry Security Standards Council (PCI SSC). After seeking extensive marketplace feedback the Council has set 30th June 2018 as the deadline for migrating to a secure version of TLS, which in accordance with NIST guidelines is currently TLS version 1.1 or 1.2. The requirement to migrate to a secure version of TLS includes disabling any fall back to SSL or early TLS.

Vulnerabilities

The vulnerabilities which SSL and TLS are subject to fall into three general categories

1. The first category pertains to *cryptographic vulnerabilities in either the SSL/TLS protocol itself or in how it uses cryptographic algorithms*. A prominent example of a vulnerability of this type is the way SSL version 3 uses Cipher Block Chaining (CBC) mode. This vulnerability has been successfully exploited by POODLE, a man-in-the-middle attack that allows the attacker to decrypt data in the SSL channel.
2. The second category pertains to *the implementation of the SSL/TLS protocol*. For example, HEARTBLEED is a bug in the OpenSSL software library; it is not a design flaw in the SSL/TLS protocol itself. An attacker can exploit this bug to read the memory of systems protected by the vulnerable versions of the OpenSSL software, thereby compromising the secret keys used to identify the service providers and to encrypt the transmission of the data
3. The third category pertains to *the configuration of the SSL/TLS protocol*. Examples in this category would include the use of weak cipher suites or key sizes. For instance, the LOGJAM attack exploits systems that support weak export-grade cryptography. A successful LOGJAM attack allows a man-in-the-middle attacker to downgrade vulnerable TLS connections to 512-bit export-grade cryptography. The attacker can then read and modify any data that is transmitted over the connection. While it is possible to implement counter-measures against some of the attacks on the TLS protocol, the only reliable method of protection that is currently available is to migrate to TLS version 1.1 or 1.2. 'NIST Special Publication 800-52 Revision 1' provides useful guidelines on how to configure TLS securely.

The PCI SSC (Security Standards Council) requires that any new implementation within a Cardholder Data Environment not use SSL or early TLS. An implementation is deemed to be new where there is no existing dependency on the use of these vulnerable versions of the protocol. It is important to note that e-commerce implementations must not class consumer web browsers as dependent infrastructure to be supported. By contrast, existing implementations are ones where there is a pre-existing dependency on a vulnerable protocol.

Although it is recommended to remove or disable these vulnerable protocols from a Cardholder Data Environment immediately, they can remain in such an environment until 30th June 2018 where there is a justified business or technical need for a system or application to support them. Any entity that continues to support SSL or early TLS must implement a Risk Mitigation and Migration Plan. This document must include a detailed procedure for migrating to a secure version of TLS by 30th June 2018 at the latest and a description of the controls that have been implemented in order to mitigate the risks associated with any insecure version of the protocol that is being maintained in the interim.

The Risk Mitigation and Migration Plan does not need to adhere to a prescribed form. The PCI SSC (Council) has published guidelines for devising this plan in the 'Information Supplement: Migrating from SSL and Early TLS'. The plan should include a description of how a vulnerable protocol is being used; detailing the type of environment in which the protocol is being used, the type of data being transmitted and the types of system components involved. It is important to detail the ways in which the risks associated with a vulnerable protocol have been evaluated and the controls that have been implemented to mitigate these risks.

Since attacks always get better, never worse, it is imperative to monitor an insecure protocol closely and re-evaluate the risk it presents in light of any new information. In March 2016, researchers published the DROWN attack. This attack exploits a vulnerability in SSL version 2. Servers that do not support SSL version 2 were also shown to be vulnerable to DROWN, if they reuse RSA keys or certificates used by other servers that do support SSL version 2. The research indicated that 33% of all HTTPS servers were vulnerable to DROWN at the time of its disclosure. The plan should also document the change control processes that have been implemented to ensure that no new implementations introduce an insecure protocol to the Cardholder Data Environment. It is also important to identify which systems are being migrated to a secure form of TLS and a timeline for the migration process with a target date of 30th June 2018 at the latest; it is essential that the migration method includes steps to ensure that there will be no fall back to any of the insecure versions of the SSL/TLS protocol after the migration target date.

The National Vulnerability Database

NIST maintains the National Vulnerability Database (NVD) in order to identify and track vulnerabilities and provide remediation information. The Common Vulnerability Scoring System (CVSS) is coordinated with the NVD. A CVSS score is used to gauge the severity of a vulnerability by determining the difficulty of exploiting the vulnerability and the impact of that exploitation. Any vulnerability detected by an ASV scan that receives a CVSS score of 4 or greater will be classed as a failing vulnerability and must be addressed by the Scan Customer. Some SSL vulnerabilities have been assigned a CVSS score of 4.3 and will cause an ASV scan to fail.

If a vulnerability of this type relates to a new implementation, the configuration or component that caused the vulnerability must be disabled or removed from the Cardholder Data environment. If, on the other hand, the vulnerability relates to an existing implementation where there is a justified business or technical dependency, the Scan Customer can consult with the Approved Scanning Vendor to have the vulnerability documented as an exception under 'Exceptions, False Positives, or Compensating Controls' in the ASV Scan Report.

This measure would require the Scan Customer to provide the Approved Scanning Vendor with documented confirmation that they have successfully implemented a Risk Mitigation and Migration Plan for the environment where the failing vulnerability had been detected. If a vulnerability of this type is detected after 30th June 2018, the Scan Customer must remove or disable the vulnerable system or follow the Addressing Vulnerabilities with Compensating Controls process to verify that the system in question is not affected by the vulnerability. Some examples of a this type of scenario would be where a vulnerable SSL or TLS connection through an encrypted channel or where the data being transmitted across the connection is itself encrypted. In both of these examples the SSL or vulnerable TLS connection is not acting as a security control and the confidentiality and integrity of the data is being provided by an extraneous configuration or system.

Although compensating controls can be effective in certain circumstances and it is currently possible to implement counter-measures against some of the attacks on TLS, migration to TLS version 1.1 or 1.2 should be undertaken as soon as possible as this is the only reliable way to protect against the known vulnerabilities. Web-based technologies and e-commerce facilities are most at risk. Environments that include web browsers, Javascript and security-related session cookies should be monitored closely and have proper change control processes in place to ensure that changes to the environment do not introduce weaknesses or vulnerabilities.

Businesses providing e-commerce facilities should also be mindful of how restrictions relating to insecure versions of SSL and TLS are being implemented by web browsers. If migration away from SSL and early TLS is not undertaken effectively, web browsers might restrict a client's connection to the server of the e-commerce facility, thereby preventing or jeopardising potential custom.