CASE STUDY - Preparing for a PCI-DSS Audit using Cryptosense Analyzer

v1.0 December 2017

pci-dss@cryptosense.com

Cryptosense

Contents

Cryptosense SA, 231 Rue Saint-Honoré, 75001 Paris France

cryptosense.com

1. Introduction

Payment Card Industry Association Data Security Standard (PCI-DSS) is an information security standard entities must adhere to in order to process cardholder data for the major payment card schemes. Compliance is audited at least every 12 months, with the audit regime depending on the size and function of the entity.

The PCI-DSS Standard, now in version 3.2, contains more than 200 sub-points that address various organizational and technical aspects of how the entity must organize its information security. For all entities, the compliance process is extremely costly, while successfully passing the audit can be business-critical.

In this document, we examine how many of the points in PCI-DSS concern cryptography, and how many of those can be treated by the Cryptosense Analyzer software. By "treated" we mean that the software can both help to ensure the entity is compliant with the requirement, and be used to create evidence which can be presented to an internal or external auditor to show compliance. This allows us to estimate an ROI for deploying Cryptosense Analyzer in this context.

2. Technical and Procedural Requirements

In the 205 sub-points of the PCI-DSS standard, we can identify two types of requirement. One is a requirement on processes, to be verified by interviews with personnel or by watching processes in action. The second type is a technical requirement, to be verified by inspecting technical artifacts such as code or configuration files. Applying these criteria to the 205 sub points we identify 116 procedural requirements and 89 technical requirements.

3. Requirements on Cryptography

Among the 89 technical requirements, about a quarter of the total (22) concern cryptography. Of those, 21 can be treated by Cryptosense Analyzer.

Here are some examples of how the Analyzer can be used to fulfill these compliance obligations.

3.1. Example 1 - Requirement 3.5.1

Section 3 of PCI-DSS concerns the protection of cardholder data: 9 out of 12 requirements in this section concern cryptography, all treated using CS Analyzer. A typical example is requirement 3.5.1.

> *3.5.1 Maintain a documented description of the cryptographic architecture that includes:*
> *Details of all algorithms, protocols, and keys used for the protection of cardholder data, including key strength and expiry date*
> *Description of the key usage for each key*
> *Inventory of any HSMs and other SCDs used for key management*

Cryptosense Analyzer produces a full cartography of the cryptography used by an application including all the operations, protocols, and keylengths. Additionally, Analyzer reports detail key usage and determine whether the keys were used and stored in a secure way. Cryptosense Analyzer for PKCS#11 captures details of the operations of HSMs and can report on the keys stored as well as the security of the HSM's configuration.

3.2. Example 2 - Requirement 4.1

Requirement 4 concerns encryption of cardholder data across open, public networks: 2 of the 3 requirements in this category are technical, both concern cryptography, and both can be treated by Cryptosense Analyzer. For example,

> *4.1 Use strong cryptography and security protocols to safeguard sensitive cardholder data during transmission over open, public networks, including the following:*
> *Only trusted keys and certificates are accepted.*
> *The protocol in use only supports secure versions or configurations.*
> *The encryption strength is appropriate for the encryption methodology in use.*

Cryptosense Analyzer's application security reports show which keys and certificates are used and where they came from. This allows easy verification that only trusted certificates can be used. Protocol versions and configurations are also verified by the Analyzer. All encryption operations are verified for errors in key management, parameter choice, mode choice, use of authentication etc.

3.3. Example 3 - Requirement 6.5.3

Requirement 6 in PCI-DSS covers the development and maintenance of secure systems and applications. Of 16 technical requirements in this section, 7 involve cryptography and all are treated by Cryptosense Analyzer. For example

> *6.5.3 Examine software-development policies and procedures and interview responsible personnel to verify that insecure cryptographic storage is addressed by coding techniques that:*
> *Prevent cryptographic flaws.*
> *Use strong cryptographic algorithms and keys.*

Cryptosense Analyzer verified all cryptographic operations carried out by an application to ensure only strong algorithms are used and cryptographic flaws are avoided.

4. Conclusion

Cryptography plays a major and growing part in PCI-DSS compliance. Cryptosense Analyzer covers almost all the crypto-related requirements. These account for 24% of the total technical requirements that must be satisfied for an audit. They are among the most difficult and time-consuming to verify by hand. Cryptosense Analyzer therefore represents a strong return on investment thanks to its ability to detect non-compliances before an audit and provide reports that help give evidence that technical requirements are met.

See Appendix overleaf.

5. Appendix - Table of PCI-DSS Requirements Involving Cryptography

CID	Control Description	Crypto Involved	Covered by Cryptosense Analyzer
Requirement 2: Do not use vendor-supplied defaults for system passwords and other security parameters			
2.2.3	Implement additional security features for any required services, protocols, or daemons that are considered to be insecure. Note: Where SSL/early TLS is used, the requirements in Appendix A2 must be completed.	Yes	Yes
2.3	Encrypt all non-console administrative access using strong cryptography. Note: Where SSL/early TLS is used, the requirements in Appendix A2 must be completed.	Yes	Yes
Requirement 3: Protect stored cardholder data			
3.4	Render PAN unreadable anywhere it is stored (including on portable digital media, backup media, and in logs) by using any of the following approaches: · One-way hashes based on strong cryptography, (hash must be of the entire PAN) · Truncation (hashing cannot be used to replace the truncated segment of PAN) · Index tokens and pads (pads must be securely stored) · Strong cryptography with associated key-management processes and procedures. Note: It is a relatively trivial effort for a malicious individual to reconstruct original PAN data if they have access to both the truncated and hashed version of a PAN. Where hashed and truncated versions of the same PAN are present in an entity's environment, additional controls must be in place to ensure that the hashed and truncated versions cannot be correlated to reconstruct the original PAN.	Yes	Yes
3.5.1	Additional requirement for service providers only: Maintain a documented description of the cryptographic architecture that includes: · Details of all algorithms, protocols, and keys used for the protection of cardholder data, including key strength and expiry date · Description of the key usage for each key · Inventory of any HSMs and other SCDs used for key management Note: This requirement is a best practice until January 31, 2018, after which it becomes a requirement.	Yes	Yes

Cont.

CID	Control Description	Crypto Involved	Covered by Cryptosense Analyzer
3.5.3	Store secret and private keys used to encrypt/decrypt cardholder data in one (or more) of the following forms at all times: · Encrypted with a key-encrypting key that is at least as strong as the data-encrypting key, and that is stored separately from the data-encrypting key · Within a secure cryptographic device (such as a hardware (host) security module (HSM) or PTS-approved point-of-interaction device) · As at least two full-length key components or key shares, in accordance with an industry- accepted method Note: It is not required that public keys be stored in one of these forms.	Yes	Yes
3.5.4	Store cryptographic keys in the fewest possible locations.	Yes	Yes
3.6.1	Generation of strong cryptographic keys	Yes	Yes
3.6.2	Secure cryptographic key distribution	Yes	Yes
3.6.3	Secure cryptographic key storage	Yes	Yes
3.6.4	Cryptographic key changes for keys that have reached the end of their cryptoperiod (for example, after a defined period of time has passed and/ or after a certain amount of cipher-text has been produced by a given key), as defined by the associated application vendor or key owner, and based on industry best practices and guidelines (for example, NIST Special Publication 800-57).	Yes	Yes
3.6.7	Prevention of unauthorized substitution of cryptographic keys.	Yes	Yes
Requirement 4: Encrypt transmission of cardholder data across open, public networks			
4.1.1	Ensure wireless networks transmitting cardholder data or connected to the cardholder data environment, use industry best practices to implement strong encryption for authentication and transmission.	Yes	No
4.2	Never send unprotected PANs by end- user messaging technologies (for example, e- mail, instant messaging, SMS, chat, etc.).	Yes	Yes
Requirement 6: Develop and maintain secure systems and applications			
6.3.1	Remove development, test and/or custom application accounts, user IDs, and passwords before applications become active or are released to customers.	Partially	Yes

Cont.

CID	Control Description	Crypto Involved	Covered by Cryptosense Analyzer
6.3.2	Review custom code prior to release to production or customers in order to identify any potential coding vulnerability (using either manual or automated processes) to include at least the following: · Code changes are reviewed by individuals other than the originating code author, and by individuals knowledgeable about code-review techniques and secure coding practices. · Code reviews ensure code is developed according to secure coding guidelines · Appropriate corrections are implemented prior to release. · Code-review results are reviewed and approved by management prior to release. Note: This requirement for code reviews applies to all custom code (both internal and public-facing), as part of the system development life cycle. Code reviews can be conducted by knowledgeable internal personnel or third parties. Public-facing web applications are also subject to additional controls, to address ongoing threats and vulnerabilities after implementation, as defined at PCI DSS Requirement 6.6.	Partially	Yes
6.4.5.3	Functionality testing to verify that the change does not adversely impact the security of the system.	Partially	Yes
6.5.3	Insecure cryptographic storage	Yes	Yes
6.5.4	Insecure communications	Yes	Yes
6.5.5	Improper error handling	Partially	Yes
6.5.10	Broken authentication and session management.	Partially	Yes
Requirement 8: Identify and authenticate access to system components			
8.2.1	Using strong cryptography, render all authentication credentials (such as passwords/phrases) unreadable during transmission and storage on all system components.	Yes	Yes
8.2.3	Passwords/passphrases must meet the following: · Require a minimum length of at least seven characters. · Contain both numeric and alphabetic characters. Alternatively, the passwords/ passphrases must have complexity and strength at least equivalent to the parameters specified above.	Partially	Yes

6. References

» PCI-DSS Standard v3.2. Available from https://www.pcisecuritystandards.org/document_library

» Cryptosense Analyzer. More information at https://cryptosense.com/analyzer/