PCI Compliance
2018

RJB TECHNOLOGY
Smart Custom Software Development

Tokenization is a Key to PCI Compliance

The Payment Card Industry Data Security Standard (PCI-DSS) requires that every system that transmits or stores credit card data is subject to review and must meet certain standards else substantial penalties will apply.

Tokenization is the process of replacing sensitive information, such as credit card numbers, with tokens that are not subject to PCI-DSS. The tokens are "random" values that resemble the sensitive data they replace, but they lack intrinsic value and are therefore useless to hackers.

Removing credit card data from all or part of your environment sounds like a good security measure, and it is. The fact that tokenization can be far cheaper in many cases than alternatives such as encryption, makes this solution an attractive one. Here we will examine tokenization and its standing amongst other compliance strategies.

Contents

Payment Data Security

What is PCI-DSS?

"...while conceptually simple, in practice, implementing PCI-DSS requires a security review of the entire IT infrastructure..."

Companies today use a number of measures to secure data and resources, including, but not limited to: physical security, network security, user authentication/authorization, and resource or device security. Payment Card Industry (PCI) requirements and security "best practices" mandate additional security measures for stored sensitive credit card data.

Stored sensitive credit card data refers to track data, card validation codes, PIN(s), expiry date, or credit card numbers stored on any media e.g. database tables, files (disk or tape), screens, and reports. Large merchants must undergo extensive examinations of their IT security and processes to verify compliance with the Payment Card Industry Data Security Standard (PCI-DSS).

Every system that transmits or stores credit card data is subject to review. Small and mid-sized merchants must go through all the same steps as large merchants except the compliance audit, where they are on the honor system.

The list of DSS requirements is lengthy - a substantial investment of time and money is required to create policies, secure systems, and generate the reports PCI assessors need. While the Council's prescribed security controls are conceptually simple, in practice they demand a security review of the entire IT infrastructure. According to Gartner, a company with 100,000 customer accounts spends on average about $6 per account to roll out encryption appliances.

Solutions for PCI-DSS

"...the typical objective is to lower the cost while remaining compliant..."

Solutions for securing stored sensitive credit card data are discussed briefly below in order of preference i.e. Removal is preferable to Substitution, and Substitution is preferable to Encryption. In practice, all three techniques are used in combination to a greater or lesser degree with the typical objective being to lower the cost while remaining compliant. Hence, the order of preference reflects that cost ranking.

Removal

"...Removal is the only solution that may be used for the card-validation code or value (CVC or CVV)..."

Where possible, credit card numbers are removed from the system. For example, a report containing a credit card number may simply have that data removed from the report, thereby eliminating the security requirement.

- Note 1: Truncation or masking of credit card numbers is a possibility where only a partial credit card number is required. The maximum that can be displayed or stored in the clear are the first 6 and last 4 digits of a credit card. If this limit is not exceeded, the credit card data may be considered as "removed". If there is no NEED to display both the first six and last four digits, then only display the last four.

- Note 2: Removal is the only solution that may be used for sensitive authentication data as described in section 3.2 of PCI-DSS "Appendix: PCI Data Storage Requirements". This includes the card-validation code or value (CVC or CVV - a three or four-digit number printed on the front or back of a payment card used to validate card-not-present transactions).

Tokenization

"...Access to the full credit card number associated with a Token is provided on a "need to know" basis..."

Credit card numbers are substituted with Token references in all programs, database tables, files, and logs, where such substitution is deemed possible.

A token is a number, sometimes similar to a credit card number in appearance and form, but it is not a credit card number. It's designed to resemble a credit card number so that it can easily replace a credit card number in a file or a network transmission i.e. the file or message format doesn't need to change.

If credit card data is replaced with tokens, most of the security checks no longer apply since the tokens themselves are not considered sensitive data. Database tables, files, and logs containing Token references are PCI compliant and remain unencrypted.

Access to the full credit card number associated with a Token is provided on a "need to know" basis for the duration of the retention policy for credit card data, after which such access is limited to the masked credit card number i.e. the full credit card number must be Removed by a "Purge Process" based upon a well-defined retention policy.

Encryption

"...serious thought should be taken on using a crypto hardware solution to store the master key..."

Sensitive credit card data is encrypted using strong encryption technology. Only users with the appropriate credentials are granted authority to decrypt this data. PCI compliant procedures, policies, and processes control all encryption/decryption keys and related authentication/ authorization credentials. Collectively, these are referred to as Key Management processes and procedures.

Encryption solutions are dependent upon the underlying storage technology, and therefore need to be dealt with on a case-by-case basis. Encrypted credit card numbers are subject to the retention policy after which they must be Removed i.e. the full credit card number must be Removed by a purge process for that database table or file.

Key Management processes and procedures are related to the encryption solution. For example, tamper proof crypto hardware solutions, also known as hardware security modules (HSM), are generally considered to be the most secure encryption environment. This is primarily because crypto hardware can safely store the most critical encryption key - the Key Encryption Key (KEK), also referred to as a master key.

Other encryption technologies that are not based on crypto hardware, either: do not use a KEK (not secure), or store the KEK on disk (less secure), or require that an operator enter the KEK during system startup (can present operational difficulties). There are many crypto hardware solutions available in the marketplace, and serious thought should taken on using such a solution to store the KEK.

What is a Token Service?

"...approach ensures that when a credit card is used it will always be assigned the same Token..."

A Token Service is a secure application program and database that provides an Application Programming Interface (API) for secure storage of credit card numbers, and the issuance of "substitute" PCI-DSS compliant tokens in their place.

A hash algorithm is typically used to provide a permanent 1-1 relationship between a Token and a credit card number. This approach ensures that when a credit card is used it will always be assigned the same Token, even if the credit card number had been previously purged from the system, for example due to a retention policy.

The SHA3-256 algorithm is used to provide a hash value for a given credit card that will be used as an index to the Token table. SHA3-256 is a federal government currently recommended standard hash algorithm. Refer to Secure Hashing for further information on government recommendations for hash algorithms.

The input to the SHA3-256 algorithm is a combination of credit card number plus "salt" value. The salt is a randomizing agent that further obscures the method by which the hash value was created. Its purpose is to make it much more difficult for a hacker to pre-compute the SHA3-256 hash values for a list of valid credit cards, since he/she would not have the salt value that was used for the calculation. For indexing purposes the salt must be predictably generated to ensure that the same input always produces the same output. In this way we can guarantee that the result can be reproducible for future index lookup. Note: The requirement for reproducibility differentiates this approach somewhat from the random salt approach used in a Message Authentication Code (MAC).

Salts commonly used for index lookup include:

- Characters pulled from an MD5 sum of the card number
- Characters generated from digits of the card number
- Proprietary generation based on a combination of factors including XOR, Base64, or simple math
- Characters pulled from an MD5 sum that is salted with a proprietary set of bits that is treated like a secret encryption key and securely stored

Creating a salt from the message input (i.e. credit card number) to the hash algorithm is sometimes referred to as the Initialization Vector (IV) Message Dependent Approach.

For further information about this topic refer to Collision-Resistant Message Preprocessing. As stated in the article the IV Message Dependent Approach "is an extremely simple way of thwarting the known collision attacks". This approach is ideal for situations where data streaming is not involved during the hash calculation, as is the case for hashing a credit card number.

How It Works

In a correctly implemented tokenization system the merchant never sees customer credit card information. They only see tokens, which are essentially strings of information. Here is what happens to credit cards from the point of swipe until the payment process is completed.

1. A credit card is swiped in a POS machine or entered into an e-commerce site.
2. The POS machine (or e-commerce site) passes the PAN to the credit card tokenization system.

3. The tokenization system generates a string of 16 random characters to replace the PAN, or retrieves the associated token (if it has already been created) and records the correlation in the data vault.
4. The tokenization system returns the token to the POS terminal (or e-commerce site) and it is used to represent the customer's credit card in the system.
5. If the business is using a payment processor's tokenization solution, the token is sent to the payment processor, who, using the same tokenization technology, can de-tokenize and view the original credit card number and process payment. If the organization is using a third party tokenization solution, the token is sent to the third party, who then de-tokenizes it and sends it along to the payment processor for credit card processing.

What Does a Token Look Like?

"Most organizations use format preserving tokens to avoid causing validation issues with existing applications and business processes."

There are two types of token formats: format preserving and non-format preserving.

Format preserving tokens maintain the look and feel of original 16-digit payment card data. For example:

Payment Card Number: 4111 1111 1111 1111
Format Preserving Token: 4111 8765 2345 1111

Non-format preserving tokens don't resemble the original data and could include both alpha and numeric characters. For example:

Payment Card Number: 4111 1111 1111 1111
Non-format Preserving Token: 25c92e17-80f6-415f-9d65-7395a32u0223

Most organizations use format preserving tokens to avoid causing validation issues with existing applications and business processes.

How Tokenization Reduces Cost for PCI-DSS Compliance

Tokenization reduces cost for PCI-DSS compliance by reducing the scope of the audit process associated with compliance. Essentially, the Cardholder Data Environment (CDE) is reduced in size, since many systems will now contain secure tokens rather than a primary account number (PAN).

A properly implemented tokenization solution reduces or eliminates the need for a merchant to retain PAN in their environment once the initial transaction has been processed. With adequate segmentation and process controls, a tokenization solution minimizes the number of merchant system components that need to be protected according to PCI DSS. For more information on how tokenization may be used to reduce costs, visit PCI DSS Tokenization Guidelines