Heartland Payment Systems, Inc.
SecureSubmit
Technical Solution Assessment White Paper

January 20, 2016

Prepared for: Heartland PAYMENT SYSTEMS™ — The Highest Standards | The Most Trusted Transactions

Company Contacts

Joseph Wysocki – Executive Director, e-commerce
Mark Hagan – Merchant Integration Support

Coalfire Contacts

Blair McBride, Director Application Validation Services, QSA
Nick Trenc, Senior Consultant, QSA, PA-QSA, CISSP, CISA

PA-QSA Company Information:
Coalfire Systems, Inc. – 11000 Westmoor Circle, Suite 450, Westminster, CO

This page intentionally left blank

Contents

Executive Summary

Overview

Heartland Payment Systems, Inc. (Heartland), one of the nation's largest payment processors, has developed a merchant data protection solution to protect sensitive consumer cardholder data entered via a consumer's browser. With SecureSubmit, e-commerce merchants now have an alternate approach for protecting sensitive data submitted during the e-commerce checkout process, which can minimize the burden of merchant PCI DSS compliance.

Heartland engaged Coalfire Systems, Inc. (Coalfire), to conduct an independent technical assessment of their SecureSubmit hosted e-commerce solution to achieve the following:

1. Briefly describe how SecureSubmit is implemented in a typical customer environment and how cardholder data is protected during a typical transaction;
2. How SecureSubmit compares to other technologies (e.g. iFrame) to meet PCI DSS requirements;
3. Identify those requirements in the PCI DSS Self-Assessment Questionnaires that could be removed from customer management responsibility and marked Not Applicable when SecureSubmit is correctly implemented.

SecureSubmit provides a solution that provides relief for merchants from storing cardholder data by using tokenization. Upon entry in the consumer's browser, cardholder data is securely transmitted via HTTPS/TLS v1.2 to Heartland's servers via Heartland's Portico PCI DSS compliant gateway. This transmission takes place directly between the consumer and Heartland. The data is then processed and assigned a token. This tokenized data is then sent to the merchant's server(s) for storage and future reference. The secure token is the only data that is transmitted between the merchant's servers and Heartland. As such, no cardholder data ever traverses or is stored on the merchant's servers.

During this engagement, Coalfire assessed SecureSubmit by hosting a simple checkout page in Coalfire's lab environment representing the merchant's e-commerce site accessing the SecureSubmit JavaScript that in turn creates the iFrame controls for each of the cardholder data input fields. Coalfire's assessment procedures included technical and functionality testing of the web page hosting the iFrame and network traffic analysis to ensure that all cardholder data submission was directly between the browser and the Heartland backend and not passing through the test web application hosted inn the Coalfire lab. It should be noted that Heartland provides instructions, SDKs, and sample code for implementation across a wide variety of languages and integration (in the form of a plugin) with various e-commerce and Content Management System platforms. Languages supported include PHP, .Net, Java, Ruby, and Python. Some of the e-commerce and CMS platforms supported include Magento, Marketpress, X-Cart, OpenCart, ZenCart, and WordPress among others.

In this paper, Coalfire will describe how SecureSubmit, when implemented properly, can help merchants remove areas of responsibility and applicability from their associated PCI DSS SAQs. It will also describe a typical implementation and how cardholder data is protected as well as removed from the merchant's environment.

In order for a merchant to leverage the SecureSubmit solution to reduce and minimize the burden of PCI DSS compliance, the following assumptions are made:

1. The merchant is following all guidance from Heartland in implementing the SecureSubmit solution. While Heartland's SecureSubmit has made it very easy to ensure cardholder data is encrypted properly for the merchant, the merchant must use the jQuery plugin properly.

2. The merchant must protect the e-commerce server hosting the iFrame used to leverage Heartland's SecureSubmit by following all relevant PCI requirements and security best practices. The merchant has these specific responsibilities:

    a. Managing website and servers, including applicable PCI DSS requirements

    b. Having written agreements with any third-parties and ensuring they protect cardholder data on behalf of the merchant, in accordance with PCI DSS

    c. Securing the web page(s) containing the iFrame

Audience

Merchants wishing to reduce the applicable PCI DSS controls and ease their compliance efforts will benefit from understanding the contents of this paper.

Summary Findings

The following are important highlights of Coalfire's technical evaluation:

- A properly designed and deployed SecureSubmit integrated e-commerce application:
    - Represents an attack surface and threat environment similar to that of a direct-post API.
    - Reduces the risk of consumer cardholder data compromise and removes exposure of plain text cardholder data to the e-commerce merchant by transmitting cardholder data directly from the consumer's browser to the Heartland backend.
    - Can significantly reduce the PCI DSS controls that are applicable similar to merchants implementing a direct-post API solution.
- Implementing a SecureSubmit integrated e-commerce solution should not lower a merchant's level of responsibility for the security of their e-commerce environment. Merchants should implement security best practices on their web server and the supporting network regardless of the impact to any applicable PCI DSS controls.

Application Summary

The SecureSubmit solution was assessed by Coalfire during the timeframe of Januaray 11th – 22nd, 2016.

Coalfire performed testing on the solution by implementing the iFrame-Tokenization.html page supplied by Heartland in Coalfire's lab and focusing on packet captures, data contained in browser requests (GET and POST), and analysis of the same to ensure that the cardholder data communication was strictly between the client's browser and the Heartland backend via iFrame communication and that the only data passed the merchant's e-commerce site was sufficiently encrypted as well as tokenized to meet compliance standards for encrypted or tokenized data in transit. Coalfire found that:

- The SecureSubmit solution utilized HTTPS/TLSv1.2 for all communications to and from the Heartland servers from the consumer's browser.
- At no time was sensitive plain-text data sent from the solution to Heartland's servers.
- Only tokenized data was returned and the token is unique and has no mathematical resemblance to the original data.
- The solution removes the risk of exposure or storage of cardholder data on merchant servers by securely transmitting cardholder data directly from the consumer's browser to Heartland's secure servers and returning only tokenized data to the merchant.

Heartland's SecureSubmit solution allows merchants to seamlessly integrate cardholder data encryption and tokenization into their respective e-commerce websites and checkout pages without redirection or loss of brand identity. All exchanges of sensitive data occur directly between the consumer and Heartland via the Portico Gateway. At no time does cardholder data or sensitive data ever touch the merchant's servers.

For merchants, SecureSubmit consists of the following components:

- A jQuery plugin is required to encrypt sensitive card data, integrated into a merchant's checkout page
- The Portico SDK, available in most popular programming languages, and credentials provided by Heartland
- A Public encryption key, provided via Heartland

The SecureSubmit API library utilizes secret and public keys (obtained directly from Heartland) for submission, validation, and authentication. Upon submission of the form by the consumer, the SecureSubmit plugin will perform the tokenization process described below. Once the data is tokenized, the merchant can then reference the transaction and card using the unique token to perform various actions such as:

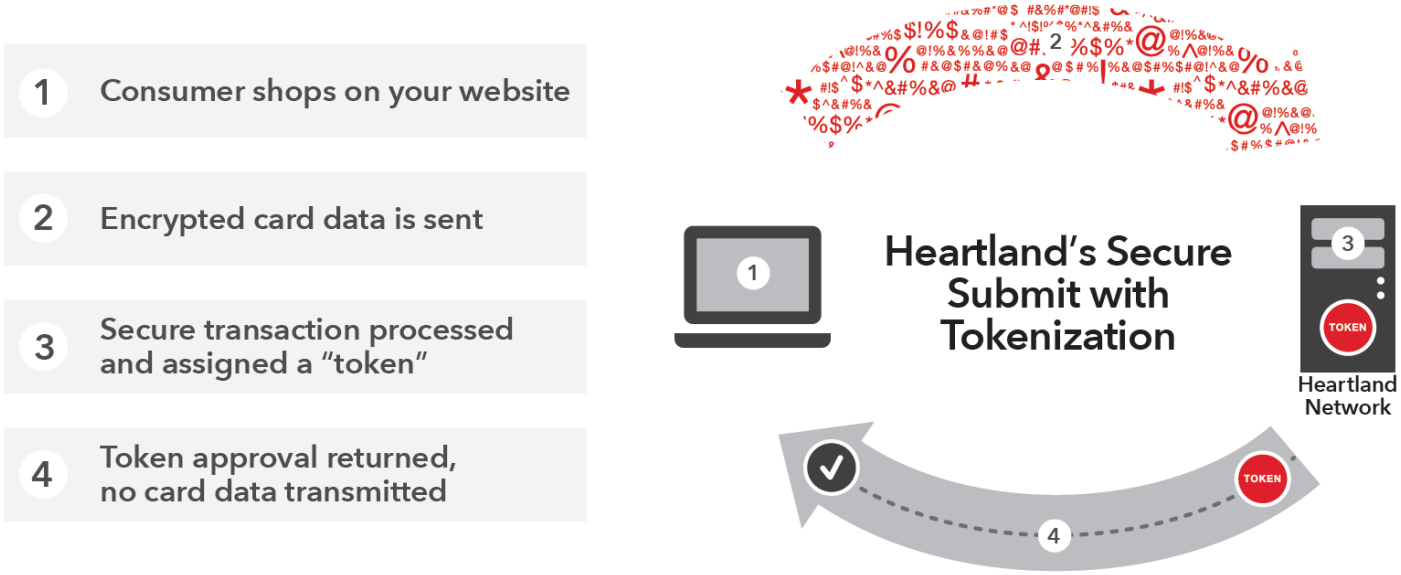
- Returns
- Card verifications
- Refunds
- Reversals

- Voiding transactions
- Editing transactions
- Transaction reporting

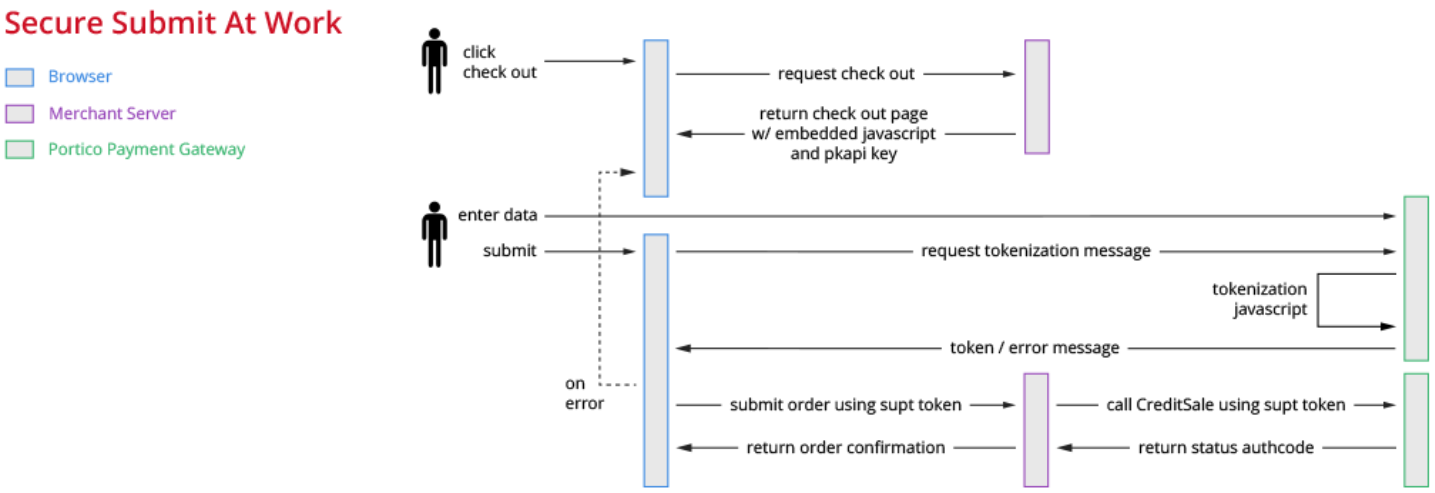
In a typical deployment scenario, the merchant hosts the SecureSubmit integrated e-commerce application and all back-end components are hosted by Heartland. The system components work together in a SecureSubmit integrated e-commerce solution as follows:

1. Consumer initiates the checkout process with the SecureSubmit integrated checkout page which includes two JavaScript files hosted by Heartland's Portico gateway. Portico is Heartland's PCI DSS compliant payment gateway that runs web services using SOAP.
2. The checkout page on the merchant site creates 4 iFrames representing the 3 data entry fields for the card number, card expiration data and card CVV as well as the submit button to create a complete payment page.
3. Customer inputs card data into iFrames and submits the request to the gateway, at which time the SecureSubmit Javascript takes control over submission such that Javascript sends a message from the submit button child window (iFrame page from Heartland) to the card number child window (iFrame page from Heartland) to start tokenization.
4. Card number child window sends messages to the other child windows to request the data held within their fields.
5. Once all data is received, card number child window accumulates the card data into a tokenization request to our service.
6. On success, card number child window sends a message to the parent window with the single-use token.
7. Form in parent window is allowed to submit to backend services with single-use token and generic data (expiration date, last four, etc.). Card number and security code remain in child windows.
8. The authorization response (which contains tokenized data) is returned to the merchant e-commerce system and presented to the consumer's browser.

The diagrams below show a typical SecureSubmit request/response sequence:



The following diagrams show browser headers from a typical SecureSubmit transaction:



The network traffic captured below shows SecureSubmit securely sending the cardholder data (via HTTPS/TLSv1.2) to Heartland's servers to be tokenized.

```
72 8.21121800 10.51.100.46    23.235.44.133   TLSv1.2   220 Client Key Exchange, Change Cipher Spec, Encryp
73 8.21863900 10.51.100.46    23.235.44.133   TLSv1.2   220 Client Key Exchange, Change Cipher Spec, Encryp
74 8.22442900 10.51.100.46    23.235.44.133   TLSv1.2   220 Client Key Exchange, Change Cipher Spec, Encryp
75 8.23665100 23.235.44.133   10.51.100.46    TLSv1.2   145 Change Cipher Spec, Encrypted Handshake Message
76 8.23669200 10.51.100.46    23.235.44.133   TCP        54 51972→443 [ACK] Seq=379 Ack=3697 Win=64768 Len=
77 8.24019800 65.118.49.55    10.51.100.46    TCP        60 443→51969 [ACK] Seq=1 Ack=225 Win=128256 Len=0
78 8.24063100 65.118.49.55    10.51.100.46    TLSv1.2  1514 Server Hello
79 8.24067000 10.51.100.46    65.118.49.55    TCP        54 51969→443 [ACK] Seq=225 Ack=1461 Win=65536 Len=
80 8.24069800 65.118.49.55    10.51.100.46    TCP      1514 [TCP segment of a reassembled PDU]
81 8.24071700 10.51.100.46    65.118.49.55    TCP        54 51969→443 [ACK] Seq=225 Ack=2921 Win=65536 Len=
82 8.24102900 65.118.49.55    10.51.100.46    TCP      1230 [TCP segment of a reassembled PDU]
83 8.24105300 10.51.100.46    65.118.49.55    TCP        54 51969→443 [ACK] Seq=225 Ack=4097 Win=64512 Len=
84 8.24107400 65.118.49.55    10.51.100.46    TLSv1.2   338 Certificate
85 8.24109100 10.51.100.46    65.118.49.55    TCP        54 51969→443 [ACK] Seq=225 Ack=4381 Win=65536 Len=
86 8.24196100 10.51.100.46    65.118.49.55    TLSv1.2   412 Client Key Exchange, Change Cipher Spec, Encryp
87 8.24335600 23.235.44.133   10.51.100.46    TLSv1.2   145 Change Cipher Spec, Encrypted Handshake Message
88 8.24338300 10.51.100.46    23.235.44.133   TCP        54 51971→443 [ACK] Seq=379 Ack=3697 Win=64768 Len=
89 8.24901500 23.235.44.133   10.51.100.46    TLSv1.2   145 Change Cipher Spec, Encrypted Handshake Message
90 8.24904500 10.51.100.46    23.235.44.133   TCP        54 51970→443 [ACK] Seq=379 Ack=3697 Win=64768 Len=
91 8.30274700 65.118.49.55    10.51.100.46    TCP        60 443→51969 [ACK] Seq=4381 Ack=583 Win=127898 Len
92 8.31227800 65.118.49.55    10.51.100.46    TLSv1.2   145 Change Cipher Spec, Encrypted Handshake Message
93 8.31237700 10.51.100.46    65.118.49.55    TCP        54 51969→443 [ACK] Seq=583 Ack=4472 Win=65536 Len=
94 8.32227800 10.51.100.46    65.118.49.55    TLSv1.2   555 Application Data
```

Once the cardholder data has been converted to a token, Heartland's servers send the following response:

```
▼ jsonp_callback_1562({object: "token", token_value: "supt_BgZr7e6oPtuPNQbtVUJ…
  ▶ card: {number: "**********5373"}
    object: "token"
    token_expire: "2016-01-22T20:27:47.8086422"
    token_type: "supt"
    token_value: "supt_BgZr7e6oPtuPNQbtVUJ1qRii"
```

This response contains the token (**`supt_BgZr7e6oPtuPNQbtVUJ1qRii`**) and properly masked cardholder data (**************5373**). These data elements can be safely stored on a merchant's servers without any impact to PCI DSS requirements as they are no longer considered cardholder data or sensitive authentication data (SAD) as defined by the PCI Security Standards Council.

At this point, a token has been issued and the card is ready to be charged by the merchant using the token provided.

Relevant Findings

The following findings are relevant highlights from this assessment:

- SecureSubmit provides a unique solution to address the protection of consumer cardholder data entered via browser-based e-commerce applications. SecureSubmit immediately tokenizes the consumer's sensitive cardholder data and this unique token is then used for all transactional processing.
- A Heartland SecureSubmit e-commerce solution:
    - Allows an e-commerce merchant to seamlessly integrate browser-context submission of cardholder data into their existing e-commerce application.
    - Reduces the risk of sensitive data compromise and removes exposure of plain text cardholder data to the e-commerce merchant by transmitting the encrypted data through the merchant network directly from the consumer's browser.

- Represents a limited attack surface and threat environment similar to that of Direct-post API.
- Can minimize applicable PCI DSS controls and validation requirements similar to merchants implementing an iFrames solution.

- Merchants using the SecureSubmit solution, along with their acquiring banks and QSA, can make a risk based determination to reduce the number of applicable controls the merchant must validate according to the SAQs. This not only reduces the cost of validating PCI DSS compliance but also significantly reduces the risk of cardholder data compromise in the environment.

SecureSubmit leverages an inline frames (iFrames) approach. In iFrames solutions such as Heartland's SecureSubmit, the merchant hosts an e-commerce web site that uses licensed APIs to redirect the payment information directly from the consumer's browser to the payment processor. This solution allows merchants to utilize client-side code to capture sensitive information from a consumer's browser and securely transmit it directly to a payment processor without having to pass any sensitive data back to or through the merchant's systems or servers. Heartland's SecureSubmit further uses tokenization to identify specific transactions. That tokenized data (which is no longer considered cardholder data or sensitive data) is sent back to the merchant for finalize the transaction. Merchants can then use that unique identifier to perform further actions (refunds, returns, reversals, etc.).

This solution is comparable to other e-commerce solutions such as third-party hosted payment pages in that there is some shared management between the merchant and the vendor that codes/maintains the APIs that are responsible for capturing and transmitting sensitive data captured from the consumer's browser. While the API vendor is responsible for the proper maintenance and security of the code itself, the merchant is responsible for properly implementing the code and for the underlying security of the systems on which the code is installed.

The SecureSubmit solution's allows the customer to build out their payment page in such a way that "all elements are delivered to the consumer's browser originate only and directly from a PCI DSS validated third-party service provider" making the merchant eligible for SAQ-A.

Use of an iFrame solutions provides high assurance that cardholder data is not stored or transmitted by the eCommerce solution and therefore offers the ability for merchants to reduce the number of applicable PCI DSS controls and potentially mark certain items in the PCI DSS SAQ A as "Not Applicable". In evaluating Heartland's SecureSubmit solution, Coalfire has identified 9 items in the PCI DSS SAQ A-EP that could potentially be marked as "Not Applicable", thereby reducing time and costs associated with a PCI DSS assessment. However, the overall security of the merchant's environment is important and should be taken into consideration before eliminating any security controls.

Impact on Selected Standards

The following table shows which PCI DSS SAQ items could potentially be marked as "Not Applicable" (NA) when Heartland's SecureSubmit solution is properly deployed in an otherwise PCI DSS compliant environment.

PCI DSS Requirement		Comments
9.5	Are all media physically secured (including but not limited to computers, removable electronic media, paper receipts, paper reports, and faxes)? *For purposes of Requirement 9, "media" refers to all paper and electronic media containing cardholder data.*	By default, the merchant has no access to full card numbers. As such, there is no SecureSubmit output that contains full card numbers. However, merchants can request access to this data and that data is served via another Heartland application. In all situations where full card numbers are printed, care should take to ensure that all media is physically secure in accordance with this requirement.
9.6	(a) Is strict control maintained over the internal or external distribution of any kind of media? (b) Does this include the following:	No cardholder data is stored on the merchant's servers. As such, any backups of the merchant's environment will not contain any sensitive data.
9.6.1	Is media classified so the sensitivity of the data can be determined?	No cardholder data is stored on the merchant's servers. As such, any backups of the merchant's environment will not contain any sensitive data.
9.6.2	Is media sent by secured courier or other delivery method that can be accurately tracked?	No cardholder data is stored on the merchant's servers. As such, any backups of the merchant's environment will not contain any sensitive data.
9.6.3	Is management approval obtained prior to moving the media (especially when media is distributed to individuals)?	No cardholder data is stored on the merchant's servers. As such, any backups of the merchant's environment will not contain any sensitive data.
9.7	Is strict control maintained over the storage and accessibility of media?	No cardholder data is stored on the merchant's servers. As such, any backups of

		the merchant's environment will not contain any sensitive data.
9.8	(a) Is all media destroyed when it is no longer needed for business or legal reasons?	No cardholder data is stored on the merchant's servers. As such, any backups of the merchant's environment will not contain any sensitive data.
	(c) Is media destruction performed as follows:	
9.8.1	a) Are hardcopy materials cross-cut shredded, incinerated, or pulped so that cardholder data cannot be reconstructed?	By default, the merchant has no access to full card numbers. As such, there is no SecureSubmit output that contains full card numbers. However, merchants can request access to this data and that data is served via another Heartland application. In all situations where full card numbers are printed, care should take to ensure that all media is physically secure in accordance with this requirement.
	(b) Are storage containers used for materials that contain information to be destroyed secured to prevent access to the contents?	By default, the merchant has no access to full card numbers. As such, there is no SecureSubmit output that contains full card numbers. However, merchants can request access to this data and that data is served via another Heartland application. In all situations where full card numbers are printed, care should take to ensure that all media is physically secure in accordance with this requirement.

PCI DSS Compliance Control Impact Overview

The PCI Security Standards Council has developed the PCI DSS to mitigate the risk of compromise to a specific data set (i.e. payment card data). The standard is focused on the system components that are applicable within the PCI DSS. For all in-scope system components, all PCI DSS controls apply. The PCI DSS is based on industry security best practices but is not focused on the overall information security of merchants. To reduce applicable PCI DSS controls, you must reduce the potential security risk and access to payment card data specifically.

As most of the PCI DSS controls are designed to manage risk to cardholder data from specific threat scenarios, it is therefore possible to reduce their applicability by securing the card data in the merchant environment so that the threat scenarios are no longer a viable risk. PCI DSS compliance scope reduction does not remove a merchant from the requirement to be PCI compliant nor does it eliminate a merchant's responsibility to validate compliance to their Acquirer. Rather, PCI DSS compliance scope reduction is only focused on addressing the applicability of specific controls to a merchant's environment based on "isolation" of data, systems and networks from security risks to payment card data.

PCI DSS compliance scope control reduction's biggest payoff for merchants is the opportunity to eliminate the cost of PCI DSS control deployment for the sole purpose of meeting compliance obligations. The second major benefit is the reduction of cost and effort to validate PCI DSS compliance of the merchant environment. Many merchants have sensitive data assets, other than payment card data, in their environment that have a risk of compromise. Reducing PCI DSS compliance scope for payment card data does not mean the PCI DSS controls are not justified to protect the merchants' other information assets.