Sysnet Global Solutions
Ecommerce SAQ Selection: A Guide
November 2015

Ecommerce SAQ Selection

Natasja Bolton, CISSP QSA,
Acquirer Support Manager, Sysnet Global Solutions

1. Introduction

Determining which PCI DSS Self-Assessment Questionnaire (SAQ) is appropriate to a merchant's ecommerce web presence can be difficult.

Often merchants are advised by their web developer or payment service provider to believe a particular SAQ is appropriate to their business but find, when they engage with their acquirer, that in fact a more complex and demanding SAQ assessment is required.

In the following whitepaper we provide a guide to help both merchants and their acquirers reach the same conclusions with regards to the appropriate ecommerce SAQ selection.

2. Available Ecommerce SAQs

There are only three SAQs that can apply to Ecommerce websites:

SAQ D	The merchant website is involved in both the acceptance and processing of the cardholder data, e.g. *Merchant website presents the payment page, accepts/captures the cardholder data and uses a method of direct integration (such as direct API) to submit that cardholder data to the payment processor*
SAQ A-EP	All processing of cardholder data is outsourced to a PCI DSS validated third-party payment processor but the merchant website does have an impact on how cardholder is accepted, e.g. *Merchant website presents the payment page but processing of the cardholder data is handled by a validated PCI DSS compliant Service Provider, such as a website configured for direct post to force submission of the cardholder data direct from the browser to the payment processor*
SAQ A	All payment acceptance and processing are entirely outsourced to PCI DSS validated third-party service providers, e.g. *Wholly outsourced ecommerce entirely provided/operated by a validated PCI DSS compliant Service Provider* OR *Redirect to a hosted payment page (or iFrame of same) provided by a validated PCI DSS compliant service provider)*

3. Selecting the correct Ecommerce SAQ

The applicability and features of each SAQ is discussed in turn.

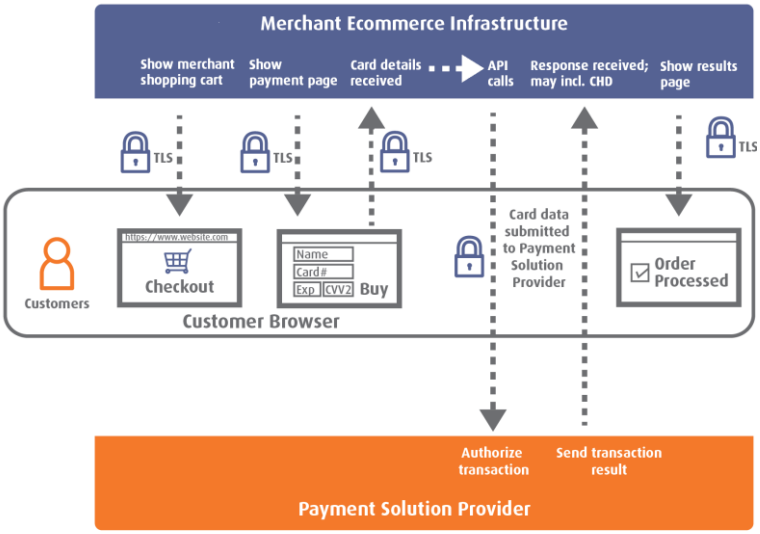
3.1. SAQ D

SAQ D applies to SAQ eligible ecommerce merchants where the merchant website is involved in both the acceptance and processing of the cardholder data

SAQ D applies to SAQ eligible merchants that are unable to meet the eligibility criteria of the other SAQ types.

For ecommerce merchants, SAQ D may be applicable because:

- Customers enter their payment card details into a payment page presented by the merchant's website and that website receives the card data submitted by the customer; and / or,
- The merchant website stores cardholder data; and /or,
- The merchant website receives cardholder data back from the payment processor post-authorisation



The diagram illustrates a typical direct integration website that requires assessment against SAQ D. It can clearly be seen that:

- The page requesting submission of and capturing payment card data is presented by the merchant website (from the merchant domain)
- The card data submitted by the customer is received by the merchant website
- The merchant website controls the submission of the cardholder data to the payment processor

Note: One legacy of PCI DSS v2.0 is the belief amongst many merchants and their ecommerce Service Providers that SAQ D can only possibly be applicable if cardholder data is being stored. This mistaken belief has often led, for implementations like that shown in the diagram above, to the selection of SAQ C for assessment. This is demonstrably incorrect:

- SAQ C specifically states that it "*is not applicable to e-commerce channels*"
- SAQ D states that applicable merchant environments that would use SAQ D include: "*E-commerce merchants who accept cardholder data on their website*"

3.2. SAQ A-EP

SAQ A-EP applies to SAQ eligible ecommerce merchants where all processing of cardholder data is outsourced to a PCI DSS validated third-party payment processor but the merchant website has an impact on how cardholder is accepted

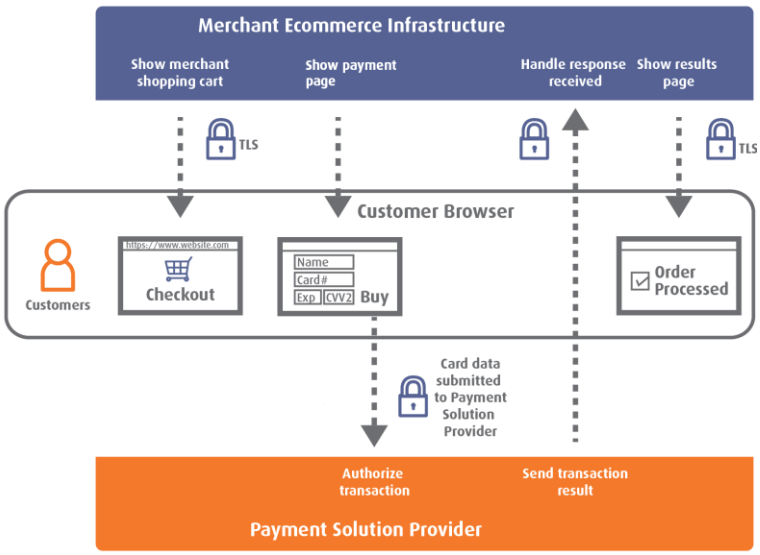
SAQ A-EP applies to SAQ eligible merchants where:

- All processing of cardholder data is outsourced to a PCI DSS validated third-party payment processor,
- The merchant website does have an impact on how cardholder is accepted but,
- The merchant website does not receive, process or transmit cardholder data.

The ecommerce integration methods that help a merchant's website fulfil those criteria include those described as: Silent Order Post, Direct Post, JavaScript created forms.

With these methods the merchant's *"e-commerce website does not receive cardholder data but controls how consumers, or their cardholder data, are redirected to a PCI DSS validated third-party payment processor"*[1]; processing of the cardholder data is handled by a validated PCI DSS compliant service provider.

This is illustrated by the diagram of the Direct Post method below:



In this example, the payment page originates from the merchant website and is configured to force submission of the cardholder data direct from the browser to the payment processor.

The payment page **does not** include the code to instruct the browser to submit the card data back to the merchant website. Typically this is done by excluding the `name` attribute in the form input fields representing the payment card data elements (card number, expiry date, security code, etc).

Sample code:

```
<label>Card Number</label>
<input type="text" data-exmaple-psp="number"></input>
<label>Expiration (MM/YYYY)</label>
<input type="text" name="expiry-month"></input>
<label></label>
<input type="text" name="expiry-year"></input>
<label>CVV2</label>
<input type="text" name="cvv2"></input>
```

name attribute

```
<label>Card Number</label>
<input type="text" data-example-psp="number"></input>
<label>Expiration (MM/YYYY)</label>
<input type="text" data-example-psp="expiry-month"></input>
<label></label>
<input type="text" data-example-psp="expiry-year"></input>
<label>CVV2</label>
<input type="text" data-example-psp="cvv2"></input>
```

no name attribute

With the JavaScript created form method, the payment page configured by the merchant instructs the customer browser to request some JavaScript from the payment processor. Unlike the Direct Post, if you were to inspect the source code of the merchant payment page you would not see form input fields representing the payment card data elements, like those shown in the sample above; however inspection of the web page would reveal that third party JavaScript was being called to create the payment form elements appearing in the customer browser.

The process flows for the Direct Post and JavaScript created form methods are nicely illustrated in the Visa Processing E-Commerce Payments Guide.

[1] Reference: https://www.pcisecuritystandards.org/documents/PCI_DSS_v3-1_SAQ_A-EP_rev1-1.pdf

The key difference between SAQ A eligible integration methods (discussed below) and SAQ A-EP eligible methods is:

- SAQ A: the payment page delivered to the customer browser originates in its **entirety** and **directly** from the validated PCI DSS compliant third-party service provider.
- SAQ A-EP: form **elements** on the payment page may be created by HTML loaded from the merchant's website or by JavaScript loaded by the consumer's browser from the validated PCI DSS compliant third-party service provider.

If any element of the payment page originates from a non-compliant service provider, then implementation is not eligible for either SAQ A or SAQ A-EP.

Note: If the merchant using an SAQ A-EP eligible integration methods relies on a third party hosting provider, the merchant must ensure that the third party is a validated PCI DSS compliant service provider (this may need to include PCI DSS Appendix A if the provider is a shared hosting provider).

This is because in a third party hosted scenario many of the PCI DSS requirements in SAQ A-EP are not within the merchant's ability to control, for example the merchant may have no influence over firewall/network management (requirement 1) or of system configuration standards (requirement 2) or vulnerability/patch management (requirement 6), especially if their ecommerce website is an instance on a shared web hosting platform.

3.3. SAQ A

SAQ A applies to SAQ eligible ecommerce merchants where all payment acceptance and processing are entirely outsourced to a PCI DSS validated third-party service provider(s)

This could be where:

- 1. The merchant has **wholly outsourced** their ecommerce website and relies entirely on a validated PCI DSS compliant e-commerce solution provider for all aspects of their website and e-commerce development, maintenance and hosting (where that third-party is validated for all applicable PCI DSS requirements given the scope of the services provided, which may include Appendix A)
- 2. The merchant's website redirects the customer browser to a **hosted payment page** that is delivered in its entirety from a PCI DSS validated third-party service provider
- 3. The merchant's website delivers the hosted payment page, sourced in its entirety from a PCI DSS validated third-party service provider, embedded within an **iFrame** on their web page.

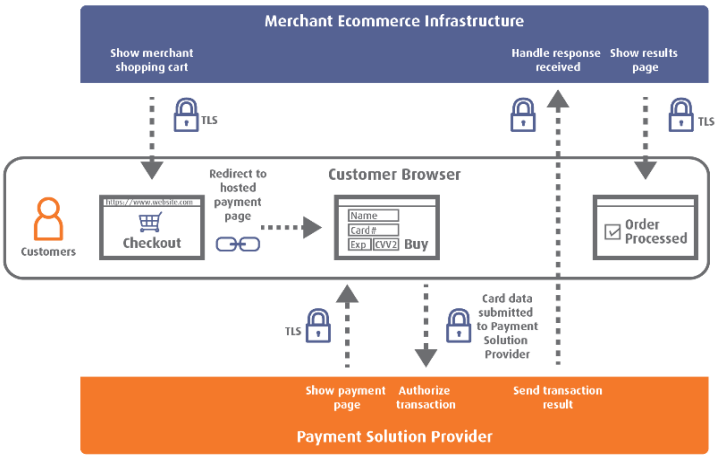
Option 1. above, fulfils the criteria for SAQ A because[2]:

- All processing of cardholder data is entirely outsourced to PCI DSS validated third-party service providers
- All third party(s) handling storage, processing, and/or transmission of cardholder data are PCI DSS compliant

[2] Taken from: https://www.pcisecuritystandards.org/documents/PCI_DSS_v3-1_SAQ_A_rev1-1.pdf

- All elements of the payment page(s) delivered to the consumer's browser originate only and directly from a PCI DSS validated third-party service provider(s)

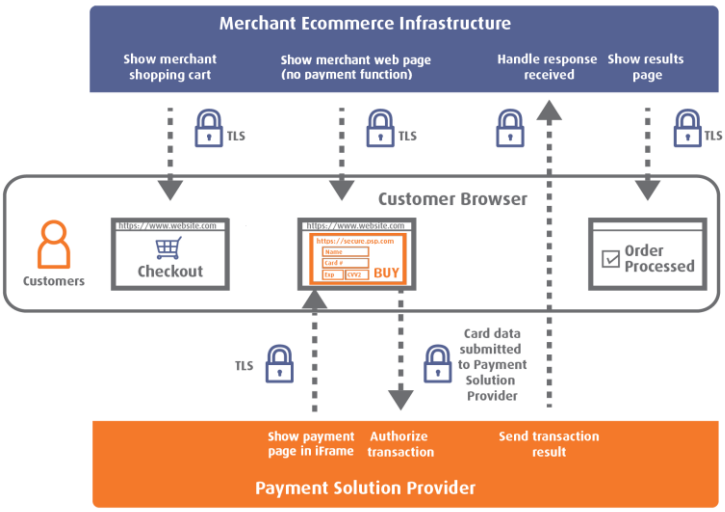
Therefore, if a validated e-commerce solution provider's PCI DSS compliance assessment encompassed all aspects of the hosting, development, maintenance and payment operations of their managed solutions in its scope, then a wholly outsourced merchant ecommerce website may utilise the direct integration method and still be SAQ A eligible.



The diagram illustrates the **hosted payment page method**, showing how this method also fulfils the criteria for SAQ A.

The payment page (the page requesting and capturing the payment card data) is delivered in its **entirety** from the Payment Solution Provider (PSP).

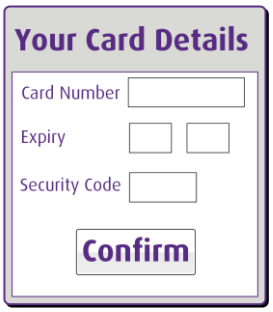
All elements of the payment page are sourced/delivered directly from the validated PCI DSS compliant third-party PSP.



This second diagram below shows the difference between the hosted payment page method and the iFrame.

In this case the hosted payment page is embedded within an **iFrame** on the merchants webpage but it is still the case that the **entirety** (all elements of) the payment page being delivered to the customer browser **originates only and directly** from the validated PCI DSS compliant PSP.

It is not always easy to tell whether the payment page that pops up is a hosted payment page. Often the URL is not immediately visible in the pop-up payment window that appears, such as this mock-up:



However, use of the web browser's developer tools may help to determine that the pop-up window is, as an example, a modal iFrame with a source URL (`src`) of the validated PCI DSS compliant PSP. In other words, the **entirety** of the payment page is still being delivered from the validated PCI DSS compliant PSP.

We have also seen instances where, on first inspection it looks like the ecommerce website uses an SAQ A-EP rather than an SAQ A eligible method: it appears that

JavaScript is being used to create the payment page or that the merchant is "*embedding a payment form in a <DIV>*[3]". On further investigation it may become apparent that the merchant page calls JavaScript to instantiate an iFrame with a source of the PSP's hosted payment page: e.g. `iframe src="https://PSP_hosted_payment_page"`

4. Conclusion

Sysnet's QSAs are finding that the payment solution providers are developing payment integration methods that stretch or go right up to the bounds of the guidance and definitions provided in the SAQs and PCI SSC FAQs SAQ A and A-EP. This creativity can make determination of the appropriate SAQ incredibly difficult.

This guide has illustrated some of the common methods used, the distinguishing features of each and shown how they relate to the criteria and definitions of the ecommerce SAQs. In many cases however a detailed investigation of the code and method may be required to make a determination of the correct SAQ selection. A familiarity with your web browser's developer tools is most useful in undertaking any such investigation.

Further reading:

Ecommerce integration options and their PCI DSS compliance implications are explained in:

- Visa Europe e-commerce guide to security and PCI DSS requirements – diagrams and articulates PCI DSS scoping and assessment requirements applicable to different ecommerce integration types:
- PCI SSC FAQs:
  - Why is SAQ A-EP used for direct post while SAQ A is used for iFrame or URL redirect?
  - Why is there a different approach for direct post implementations than for iFrame and URL redirect?
  - If a merchant's ecommerce implementation meets the criteria that all elements of the payment page originates from a PCI DSS compliant service provider is the merchant eligible to complete SAQ A or SAQ A-EP
  - PCI SSC Ecommerce Guidelines, based on PCI DSS v2.0 but still good guidance
- MasterCard PCI Whitepaper
- Visa Data Security Alert (hosted payment pages)
- Web Browser Developer Tools:
  - Mozilla Firefox: https://developer.mozilla.org/en-US/docs/Tools/Page_Inspector
  - Internet Explorer 11: https://msdn.microsoft.com/en-us/library/bg182326(v=vs.85).aspx
  - Google Chrome: https://developer.chrome.com/devtools

[3] Reference: http://pcissc.force.com/faq/articles/Frequently_Asked_Question/Why-is-there-a-different-approach-for-Direct-Post-implementations-than-for-iFrame-and-URL-redirect-what-are-the-technical-differences-and-how-do-they-impact-the-security-of-e-commerce-transactions