WHITE PAPER

AUTHORIZE.NET

SAQ ELIGIBILITY WHITE PAPER

NICK TRENC | CISSP, CISA, QSA, PA-QSA

COALFIRE

North America | Europe
877.224.8077 | info@coalfire.com | coalfire.com

TABLE OF CONTENTS

Executive Summary **3**
Audience 3
PCI Self-Assessment Questionairres 4
Methodology 4
Summary Findings 5
Conclusion 5
Assessor Comments 5
Tools and Techniques 6
References 6
SAQ Eligibility **7**
Accept Hosted 7
Accept Customer 8
Accept.js 9
Accept Mobile 10
Accept UI 11

EXECUTIVE SUMMARY

Authorize.Net, a leading provider of payment gateway services and wholly owned subsidiary of Visa (NYSE: V), engaged Coalfire Systems Inc. (Coalfire), a respected Payment Card Industry (PCI) Qualified Security Assessor (QSA) and Payment Application – Qualified Security Assessor (PA-QSA) company, to conduct an independent technical assessment of their Accept e-commerce technical solutions for Self-Assessment Questionnaire (SAQ) scope and eligibility. Coalfire conducted assessment activities including technical testing, architectural assessment, and documentation review.

In this white paper, Coalfire will describe the varying levels of SAQ eligibility for each of the current Accept products described within this paper. Those products include all of the following:

Accept Hosted is a hosted payment form, with a mobile-optimized design, which can be embedded into a checkout page.

Accept Customer provides a hosted solution specifically tailored to integrate seamlessy with the Authorize.Net Customer Profiles (tokenization) APIs.

Accept.js allows developers to have complete control over the user experience without card data ever touching the backend services.

Accept Mobile consists of iOS and Android SDKs which can be easily included in any mobile application that requires payment functionality.

Accept UI combines the pre-built Accept UX experience with the flexibility of a JavaScript library to provide complete control over the transaction itself.

AUDIENCE

This assessment white paper has three target audiences:

1. **Developers:** This audience may be evaluating any of the included Authorize.Net products for use within their e-commerce or mobile payment solutions.
2. **Merchant and Service Provider Organizations:** This audience may be evaluating Authorize.Net products for deployment in their cardholder data environment and determining what benefits could be achieved from the utilization of those solutions.
3. **QSA and Internal Audit Community:** This audience may be evaluating Authorize.Net during the evaluation of solutions and products that are installed within a merchant or service provider environment for PCI DSS audit purposes.

PCI SELF-ASSESSMENT QUESTIONAIRRES

Payment Card Industry (PCI) Data Security Standard (DSS) Self-Assessment Questionnaires (SAQs) are tools for self-assessing PCI DSS compliance for merchants that are not required by their acquirers or the payment card brands to complete a PCI DSS Report on Compliance (ROC).

Per PCI SSC definitions, the varying levels of SAQ depend on the type of functionality present for processing cardholder data. The level of compliance and complexity follows a scale from A to D, with A being the simplest. SAQ A currently has approximately 20 applicable controls, while SAQ D has hundreds. The SAQ levels pertinent to this white paper are as follows:

SAQ A applies to card-not-present merchants (e-commerce or mail/telephone order) who have completely outsourced to third-parties for all cardholder data processing functions and have no electronic storage, processing, or transmitting of cardholder data. These third-parties must be validated as PCI compliant.

Entirely outsourcing all cardholder data functions does not mean a merchant can ignore PCI compliance, but the requirements that the merchant must comply with are nominal.

Completing and maintaining SAQ A is therefore a fairly easy task for merchants and the requirements that remain in place only focus on two key areas. First, any paper copy of cardholder data (such as receipts or reports) must be physically protected or destroyed. Secondly, a list of service providers (such as those providing the above-mentioned payment processing) must be kept and their compliance status monitored.

This type of SAQ is not applicable for any card-present payment channels.

SAQ A-EP - This is a new SAQ type and has been solely designed for e-commerce merchants who only partially outsource payment processing to PCI DSS compliant third-party service providers. The merchant will have a website that redirects consumer users to a payment processor at point of payment, but the web server itself does not electronically store, process, or transmit card data. The key difference between SAQ A and SAQ A-EP is the way in which cardholder data is redirected to the payment processor and where the payment page components are generated. This will dictate whether a merchant will be SAQ A or A-EP eligible. Many e-commerce merchants who previously were able to utilize SAQ A must now use SAQ A-EP for validation.

If all elements of the payment form originate from the payment processor (redirect or iFrame), then a SAQ A can be used. For other methods, such as direct post (browser API/silent order post) or JavaScript created forms, SAQ A-EP should be used.

Both of these SAQ types are only applicable to e-commerce channels.

An entire list of the SAQ levels can be found on PCI's website at www.pcisecuritystandards.org.

METHODOLOGY

Coalfire completed a multi-faceted technical assessment during the course of this project using industry and audit best practices. Coalfire conducted technical lab testing from our Colorado lab from April 24, 2017 to April 28, 2017.

At a high level, testing consisted of the following tasks:

1. Technical review of the architecture of the full solution and its components within Authorize.Net's development sandbox.
2. Implementation of the various Authorize.Net API solutions in the cloud.
3. Technical review of all API/SDK reference guides and implementations.

SUMMARY FINDINGS

The following findings are relevant highlights from this assessment. Coalfire found that, with proper implementation utilizing guidance provided by Authorize.Net[1], the Authorize.Net solutions are eligible for the SAQs indicated below (assuming no other methodology of accepting cardholder data is present):

Accept Hosted – SAQ A (utilizes iFrame for redirect or direct post of cardholder data)

Accept Customer – SAQ A (utilizes iFrame for redirect or direct post of cardholder data)

Accept.js – SAQ A-EP (utilizes embedded JavaScript to transmit cardholder data)

Accept Mobile – SAQ A (Based on cardholder enrollment being outside of the merchant's payment processing)

Accept UI – SAQ A (utilizes iFrame for redirect or direct post of cardholder data)

CONCLUSION

Authorize.Net provides an innovative approach to allowing merchants to integrate e-commerce and mobile payments to help secure cardholder data. In the case of Accept Mobile, the direct interaction between Authorize.Net and the cardholder removes the need for the cardholder to send credit card data to the merchant. This reduces payment security risks within a merchant's backend processing platform. For the other four solutions, Authorize.Net provides flexible and robust solutions for meeting a variety of types of e-commerce transactions whether the merchant wants full control of their payment UX or just wants someone else to do it for them. These solutions provide, especially in the case of the mobile solution, an innovative approach for tackling e-commerce and mobile integration whilst remaining inline with the PCI SSC's guidance on securing e-commerce and mobile transactions.

While the above mentioned methodology removes compliance scope from the mobile solution, Coalfire, the largest assessor of PA-DSS applications, recommends that mobile application developers who integrate with Authorize.Net's Accept Mobile solution consider using PA-DSS as a framework for basic application security to protect their customer's data.

ASSESSOR COMMENTS

Our assessment put a significant focus on validating the SAQ scope/applicability for each of the Authorize.Net solutions covered by this white paper. Merchants using Accept Hosted, Accept Customer, or Accept UI, when properly implemented following guidance from Authorize.Net[1] and when no other input of cardholder data is present, can utilize a SAQ A. Merchants using Accept.js or Accept Mobile, when properly implemented following guidance from Authorize.Net[1] and when no other input of cardholder data is present, can utilize a SAQ A-EP. However, as most e-commerce environments and configurations vary drastically, it is important to note that use of this product does not guarantee security or compliance. A defense-in-depth strategy that provides multiple layers of protection should always be

[1] Final approval of the appropriate SAQ type to be utilized for any merchant rests with that merchant's acquiring bank and/or the payment brands

followed as a best practice. Please consult with Authorize.Net for policy, implementation, and configuration questions and best practices.

Using a mobile application for the initial enrollment of a cardholder doesn't confer PCI DSS scope; however, the FAQ[2] does suggest using the PA-DSS as a framework to ensure the application's security is robust.

It should also not be construed that the use of Authorize.Net solutions guarantee full PCI DSS compliance. Disregarding PCI requirements and security best practice controls for systems and networks inside or outside of PCI DSS scope can introduce many other security or business continuity risks to the merchant. Security and business risk mitigation should be any merchant's goal and focus for selecting appropriate security controls.

TOOLS AND TECHNIQUES

For this review, Coalfire utilized the following tools:

TOOL NAME	DESCRIPTION
Wireshark	Wireshark Ethernet port sniffer used to observe the traffic coming in and out of the system.
Additional tools	Google Chrome developer tools

REFERENCES

PCI Security Standards
www.pcisecuritystandards.org

SAQ Instructions
https://www.pcisecuritystandards.org/documents/SAQ-InstrGuidelines-v3_2.pdf?agreement=true&time=1494272996008

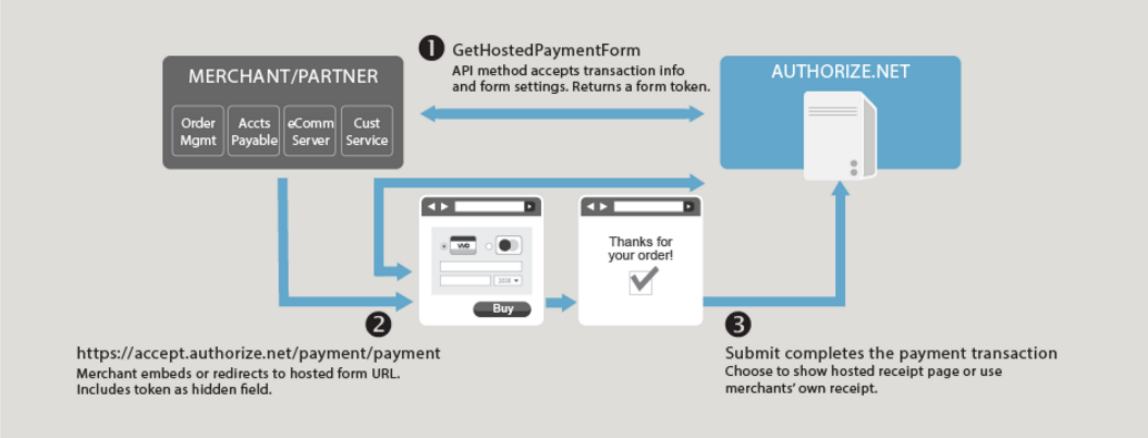
PCI E-commerce Guidelines
https://www.pcisecuritystandards.org/pdfs/best_practices_securing_ecommerce.pdf?agreement=true&time=1494273010428

Authorize.Net Developer Center
Developer.authorize.net

[2] See PCI FAQ # 1283

SAQ ELIGIBILITY

ACCEPT HOSTED



Step 1. Using our Authorize.Net API, call the API method getHostedPaymentPageRequest. The API response contains a form-validation token.

Step 2. Using the form-validation token returned in step 1, embed the payment form or redirect the customer to the payment form by sending an HTML form post to our URL - https://accept.authorize.net/payment/payment.

Step 3. The customer fills in the payment form. The transaction is processed when the user submits the form. The customer returns to the merchant's site, and the merchant displays a result page based on the url followed or the response information sent.

Figure 1 - Hosted Dataflow

As seen in the dataflow diagram provided, while utilizing an API method within an iFrame, a merchant's website would directly post or redirect cardholder data transactions to Authorize.Net, thereby allowing merchants to wholly offload payment processing of cardholder data to Authorize.Net. Authorize.Net retains full control of the UX and the payment transaction. Based on these findings, the Authorize.Net Accept Hosted solution could be eligible for a SAQ A[3]. See PCI's Best Practices For Securing eCommerce document for further details.

```
▶<div id="addShipDiv" style="margin-left: 5%; display: none;">…</div>
▼<div id="iframe_holder" class="center-block" style="width:90%;max-width: 1000px">
  ▶<iframe id="load_payment" class="embed-responsive-item" name="load_payment" width="100%" height="650px" fra
  ▶<iframe id="load_profile" class="embed-responsive-item" name="load_profile" width="100%" height="1150px" fr
  ▶<iframe id="add_payment" class="embed-responsive-item panel" name="add_payment" width="100%" frameborder="0
  ▶<iframe id="edit_payment" class="embed-responsive-item panel" name="edit_payment" width="100%" frameborder=
  ▶<form id="send_token" action="https://test.authorize.net/customer/addShipping" method="post" target="add_sh
  ▶<form id="send_hptoken" action="https://test.authorize.net/payment/payment" method="post" target="load_paym
  </div>
```

Figure 2 - Hosted iFrame Example

[3] Final approval of which SAQ type to be utilized for any merchant rests with that merchant's acquiring bank and/or the payment brands.

ACCEPT CUSTOMER

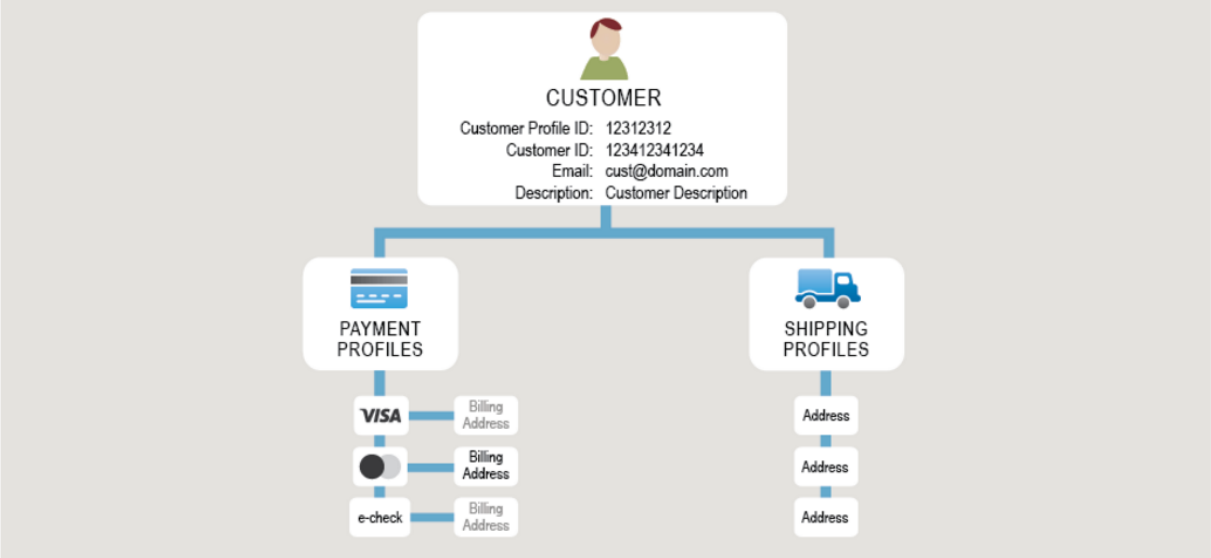


Figure 3 - Customer Dataflow

Accept Customer allows e-commerce merchants to create recurring payment data via customer profiles. Cardholder data saved during the createCustomerProfileRequest is only saved at Authorize.Net and only the customer's profile ID is returned in order to reference. The createCustomerProfileRequest process is completed solely via iFrame, thereby allowing merchants to wholly offload payment processing of card data to Authorize.Net. Authorize.Net retains full control of the UX and the payment profile process in these transactions. Based on these findings, the Authorize.Net Accept Customer solution could be eligible for a SAQ A[4]. See PCI's Best Practices For Securing E-commerce document for further details.

```
<div id="iframe_holder" class="center-block" style="width:90%;max-width: 1000px">
  <iframe id="load_payment" class="embed-responsive-item" name="load_payment" width="100%" height="65
  <iframe id="load_profile" class="embed-responsive-item" name="load_profile" width="100%" height="11
  <iframe id="add_payment" class="embed-responsive-item panel" name="add_payment" width="100%" frameb
  <iframe id="edit_payment" class="embed-responsive-item panel" name="edit_payment" width="100%" fram
  <form id="send_token" action="https://test.authorize.net/customer/editPayment" method="post" target
  <form id="send_hptoken" action="https://test.authorize.net/payment/payment" method="post" target="l
  </div>
<div class="tab-content panel-group">…</div>
```

Figure 4 - Customer iFrame example

[4] Final approval of which SAQ type to be utilized for any merchant rests with that merchant's acquiring bank and/or the payment brands.

ACCEPT.JS

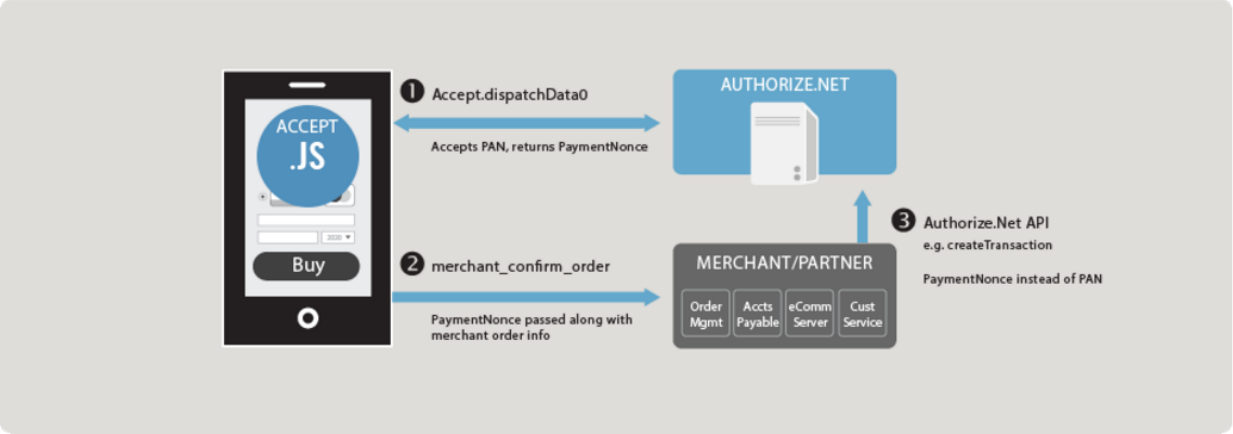
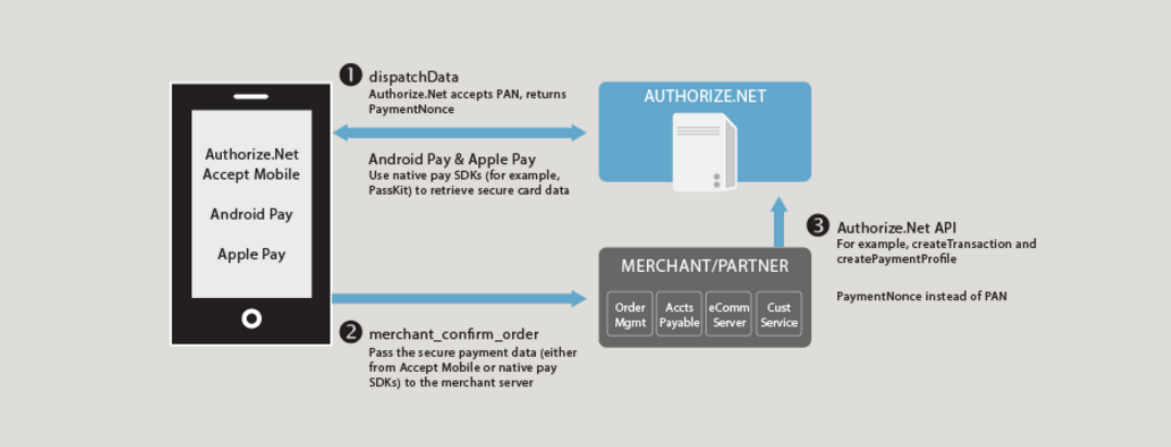


Figure 5 - JS Dataflow

Accept.js allows e-commerce merchants to embed a JavaScript control within their application while still allowing merchants full control over design and form of their website (unlike Hosted and Customer solutions). The JavaScript call sends payment information directly to Authorize.Net and returns a payment nonce (one-time use token). The merchant website can then pass the payment nonce during the customer order to Authorize.Net for order processing. Based on these findings, the Authorize.Net Accept.js solution could be eligible for a SAQ A-EP[5]. See PCI's Best Practices For Securing eCommerce document for further details.

[5] Final approval of which SAQ type to be utilized for any merchant rests with that merchant's acquiring bank and/or the payment brands.

ACCEPT MOBILE



The three payment options work similarly, using a three-step process. Only step 1 differs, depending on which in-app solution you implement.

Step 1.

The **Accept Mobile SDK** sends payment and authentication information to Authorize.Net, which returns a payment nonce.

Apple Pay and **Android Pay**: Send a request using Apple or Google's native SDK to retrieve an encrypted BLOB (Binary Large Object), which identifies the customer's payment information.

Step 2. For all in-app options, you send the encrypted payment data obtained in Step 1 to your server.

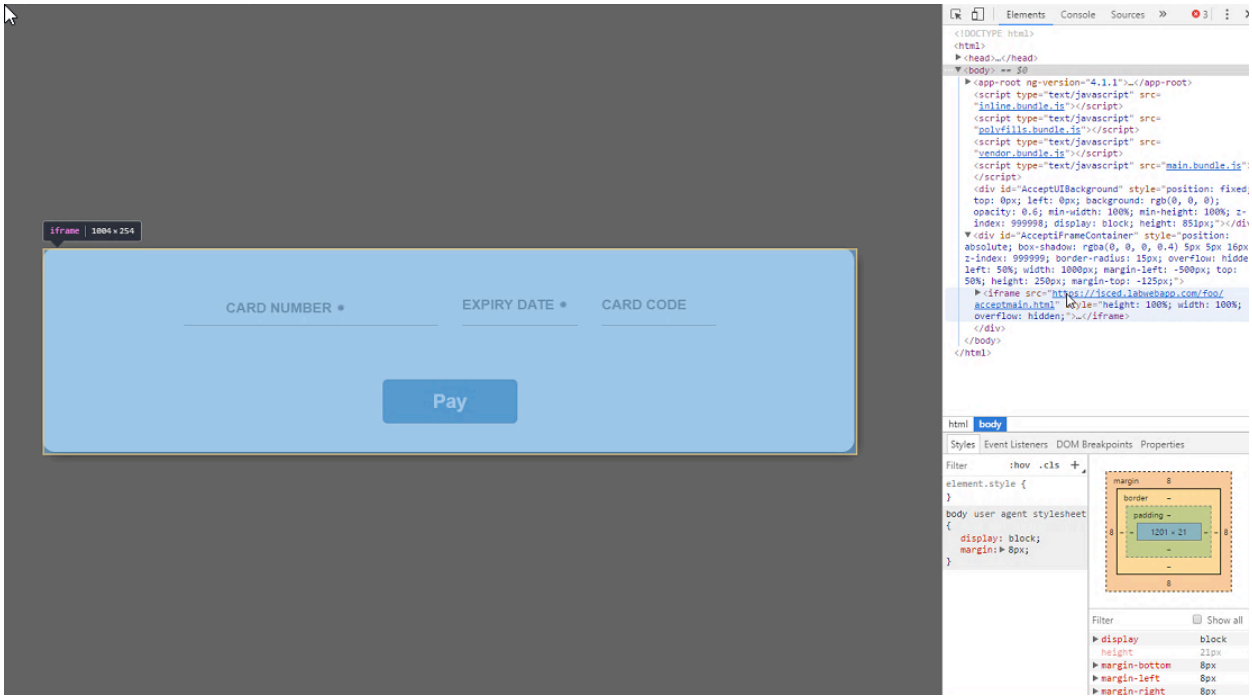
Step 3. Your server constructs a transaction request using the Authorize.Net API, placing the encrypted payment information that it received in Step 2 in the `opaqueData` element.

Figure 6 - Mobile Dataflow

Accept Mobile allows merchant application developers to embed an Software Development Kit (SDK), which permits the cardholder to enroll their payment card with Authorize.Net. In return for enrollment, Authorize.Net replaces the cardholder's card number with a token that can be used with that merchant's application for payment transactions. As such, the card relationship is between the cardholder and Authorize.Net while the payment transaction between the merchant and Authorize.Net never contains cardholder data. Because of this, the merchant has wholly outsourced all cardholder data functionality to Authorize.Net. Based on these findings, the Authorize.Net Accept Mobile solution could be eligible for SAQ-A. See PCI's FAQ #1283 and the Best Practices For Securing eCommerce document for further details.

ACCEPT UI

Accept.UI has the same UX of Accept Hosted, with the flexibility of Accept.js, all in a compliance-friendly iFrame. This solution is intended to support business models without a specific checkout page, but instead accept payment inline as part of the web page flow with said iFrame. Based on these findings, the Authorize.Net Accept UI solution could be eligible for a SAQ A[6]. See PCI's Best Practices For Securing eCommerce document for further details.



[6] Final approval of which SAQ type to be utilized for any merchant rests with that merchant's acquiring bank and/or the payment brands.

ABOUT THE AUTHOR

Nick Trenc |Principal - Practice Lead, Application Security

Nick Trenc (ntrenc@coalfire.com) is a Practice Director and Application Security Specialist with Coalfire Systems. Nick has several years of experience working as a QSA and PA-QSA helping clients develop systems and software for use in PCI DSS environments and has authored and spoken on multiple security topics including mobile security, application security, virtualization, cyber risk management, secure software development, and PCI DSS and PA-DSS compliance. He holds a CISSP, CISA, QSA, and PA-QSA.

Published 09/2017.

ABOUT COALFIRE

As a trusted advisor and leader in cybersecurity, Coalfire has more than 15 years in IT security services. We empower organizations to reduce risk and simplify compliance, while minimizing business disruptions. Our professionals are renowned for their technical expertise and unbiased assessments and advice. We recommend solutions to meet each client's specific challenges and build long-term strategies that can help them identify, prevent, respond, and recover from security breaches and data theft. Coalfire has offices throughout the United States and Europe. www.coalfire.com