

Drupal PCI Compliance White Paper

Authors:

Rick Manelius

Greg Knaddison

Ned McClain

v1.2

August 25th, 2014

Updated to PCI-DSS 3.0



Introduction

This document provides a high level overview regarding Payment Card Industry Data Security Standard (PCI DSS) compliance specifically for Drupal eCommerce solutions. The intended audiences include: Drupal developers; companies providing Drupal products, services, and hosting; and businesses evaluating Drupal as part of their eCommerce solution. Goals of this document: to emphasize the importance of PCI compliance; to summarize the options available in becoming and maintaining compliance; to provide clear next steps and additional resources.

Executive Summary

eCommerce volume continues to grow by double digit percentages each year¹ as more and more businesses are supplementing their existing revenue models and/or creating new streams^{2,3}. Simultaneously, Drupal has seen a 21% rise in the number of reported eCommerce installations in 2013 across its very large installation base⁴.

Drupal is an attractive eCommerce platform for many reasons: Its Open Source licensing eliminates one time and recurring costs to own and use, it is extremely modular and flexible, and it has a large, passionate, and world wide community that is continually contributing to and growing the platform. The Drupal community also makes a concerted effort when it comes to ensuring the platform is secure. There is a security team, weekly security advisories, venues to discuss security best practices, a collection of tools to help detect security vulnerabilities, and many other security focused strengths. The result is that Drupal is one of the most secure content management and eCommerce solutions available to merchants today.

However, while it is possible to make a Drupal-based eCommerce environment PCI DSS compliant through proper planning, usage, maintenance, and management, the mere use of Drupal by itself *IS NOT* sufficient to ensure the level of security necessary to become compliant. First, it's possible to make Drupal insecure through the use of inappropriate configurations and site mismanagement. Second, Drupal is only a single component of the cardholder data environment (CDE). Hosting servers, networks, and other factors play a significant role in securing the entire end-to-end process.

How and where the credit card data is processed, transmitted, and stored determines the size, complexity, and risk associated with the cardholder data environment, which directly impacts the time, resources, and expertise required to achieve and maintain compliance. Whenever possible, wholly-outsourced and shared-management implementations can make it easier to complete this process. Merchant-managed

solutions are the most cost prohibitive for all but enterprise level companies, and should be avoided unless the business needs require them.

Regardless of the selected approach, it's always necessary to adhere to the complete PCI DSS standard because a single exploit can undermine a company's PCI compliance, opening the company up to the financial and legal liabilities associated if the exploit were to be breached. Finally, the standard is becoming more stringent with each new version; therefore, what is optional today is likely to become a requirement in the near future.

Definitions

The following terms are used extensively throughout this document and are provided here for convenience.

- **ASV:** Acronym for “Approved Scanning Vendor.” Company approved by the PCI SSC to conduct external vulnerability scanning services.
- **Cardholder Data:** At a minimum, cardholder data consists of the full PAN (primary account number). Cardholder data may also appear in the form of the full PAN plus any of the following: cardholder name, expiration date and/or service code
- **CDE:** Acronym for “cardholder data environment.” The people, processes and technology that store, process or transmit cardholder data or sensitive authentication data, including any connected system components.
- **Credit Cards:** The term "credit cards" is used in this paper for brevity, but the advice also applies to any debit or prepaid "payment cards."
- **DSS:** Acronym for “Data Security Standard” and also referred to as “PCI DSS.”
- **QSA:** Acronym for “Qualified Security Assessor,” company approved by the PCI SSC to conduct PCI DSS on-site assessments.
- **Payment Cards:** For purposes of PCI DSS, any payment card/device that bears the logo of the founding members of PCI SSC (i.e. Visa, Mastercard).
- **PCI:** Acronym for “Payment Card Industry.”
- **ROC:** Acronym for “Report on Compliance,” which contains details documenting an entity’s compliance status with the PCI DSS.
- **SAQ:** Acronym for “Self-Assessment Questionnaire,” a checklist that you must complete and adhere to in order to obtain and maintain PCI compliance.
- **SCC:** Acronym for “Security Standards Council,” it is the governing organization and open forum responsible for the development, management, education, and awareness of PCI Security Standards.

A full list of definitions can be found in the PCI document titled *Glossary of Terms, Abbreviations, and Acronyms*⁵.

Why Does PCI Compliance Matter for Drupal?

PCI compliance is important for the industry as a whole as well as companies specifically using Drupal as part of their eCommerce solution. Here are several key areas to underscore the recommendation and requirement to adhere to PCI compliance.

Trust

Online credit card transactions have become so common and accepted in our society that it's easy to forget what is at stake. When a thief goes after physical credit cards the amount of financial loss to a financial institution or merchant is limited in size and quantity because each card has to be stolen one at a time. However in the case of electronic payments, a single computer hacker can potentially breach multiple websites with the same exploit and steal hundreds, thousands, or even millions of credit cards. The financial loss by a financial institution and the number of people affected in a website compromise can be staggering.

While these scenarios do in fact happen, their frequency is far outweighed by the volume of transactions that are handled securely. This is in part because of the set of security standards set forth by the Payment Card Industry. When properly applied, the risk level for transactions goes down significantly, which leads to more consumer trust in merchants, the process, and the industry as a whole can continue to expand. However, if this trust is broken, consumers will lose this confidence and seek other merchants and means of payment.

Privilege

A merchant's ability to accept credit card payments is a privilege, not a right. The Payment Card Industry has established a system of payment that is convenient for consumers and business owners. It is their system and they get to set the requirements. Anyone accepting online payments must sign a contract and/or accept a terms of use agreement that explicitly states the merchants responsibility to adhere to these requirements (unfortunately this isn't always emphasized in the sign up process, so be sure to read through the fine print). If a merchant fails to adhere to these requirements their ability to accept credit cards as a form of payment can be suspended.

Distributed Components

Every component between a customer's browser window and the payment processor makes up the Cardholder Data Environment (CDE). If one only focuses on securing a single component, the weakest link in the chain can be the source of an exploit. Therefore it's not enough to simply secure Drupal. The PCI standard provides guidance on how to make sure all components are secured.

Author's Note: It's important to note that ALL systems connected to the same network that's behind the same stateful firewall as the eCommerce server are also considered part of the CDE. Therefore, a common strategy is to segment the network such that only the eCommerce relevant components are together in a carefully protected network, thus reducing the overall scope for PCI DSS compliance.

Distributed Companies

It's not common for one company to control the entire end-to-end process of handling a credit card transaction. Therefore, it's important to know exactly how these companies distribute the responsibility as well as liability for ensuring this process is secure.

Financial

If you ever face a security breach and credit card information is stolen, having a proven record of your PCI compliance can protect you from the financial penalties (ranging from \$25 to \$215 per compromised card⁶. Major corporations, such as Heartland Payment Systems, have faced fines as large as \$12.5 million⁷. In addition to fines, your organization and brand can suffer financially due to a breach. Target Brands, Inc. attributed a portion of its 5.3% loss in sales and a 46% drop in profit during the 4th quarter of 2013 to its security breach in November of that year, which resulted in 40 million compromised credit card records⁸. Small companies are equally at risk, accounting for 80% of all instances of unauthorized access⁹. And while a smaller company might have a low quantity of transactions (and therefore a smaller fine), any company that has a reported breach on their record must undergo mandatory Report On Compliance audits, which alone can cost tens of thousands of dollars.

Public Relations

Beyond the financial burden of dealing with a breach, the loss of trust for a company's website, business, and brand can be staggering. The Sony PlayStation breach put more than 77 million credit card numbers at risk¹⁰. The ramifications of alerting a user base and requiring that many users to change their cards and/or put a freeze on their credit cards can quickly lose current and future customers. These transactions can also

impact customer credit scores if they unknowingly overdraft or miss a payment they were not expecting.

Summary

While developers and businesses may see this standard as a nuisance, it's important to realize that these standards allow the industry as a whole to provide the level of security and trust necessary to keep it growing. The PCI DSS should be considered a baseline, minimal level of security for sites handling sensitive information - your site very well may need security controls that exceed this standard. Also adhering to the standards helps ensure protection for the business (legal, PR, and financial), as well as your customers (legal, financial).

The Standard Itself

Note: This paper specifically excludes PA-DSS because Drupal is Open Source Software and (whether right or wrong) falls outside the PA-DSS standard.

The PCI-DSS standard covers 12 requirements across 5 overarching categories¹¹. Rather than restate what is already provided in the materials from the PCI council, the focus here is to specifically identify why each requirement directly affects the security of Drupal within the context of a complete CDE.

Requirement 1. Install and Maintain a Firewall

The integrity of a Drupal code base can only be maintained if access to the server is restricted and protected by a stateful firewall. A lax or nonexistent firewall policy provides more ways for an attacker to find a vulnerability, gain access to the server, and modify the code on the system, which opens the door for harvesting credit card numbers.

Requirement 2. Do Not Use Vendor Supplied Default Passwords

This should be obvious, but any component of your system that has a known default password can be vulnerable if it's not reset immediately. A Drupal specific example is the Commerce Kickstart installation profile. By default, it sets the main administrator username and password to "admin." This should only be used for testing/evaluation purposes and should never be used for a live site.

Requirement 3. Protect Stored Data

Ideally credit cards are not being stored on a Drupal website at all. If they are stored, they must be encrypted (e.g. using a hardware security module or an encryption key

management solution) in such a manner that anything with access to the database, server, or network cannot decode the information in a manner that is not intended. Achieving this is non-trivial and should only be attempted after gaining close familiarity with the PCI SAQ D standard (see the section *Self-Assessment Questionnaire* for definitions of the SAQ validation types).

Requirement 4. Encrypt transmission of cardholder data across open, public networks

The data leaving the Drupal application travels through many routers and networks on the way to the payment processor. Unencrypted data would allow any component along that path to copy cardholder data and other sensitive information. Therefore, encryption with a trusted SSL certificate is a requirement.

Requirement 5. Use and regularly update anti-virus software or programs

If malware is installed on a server running Drupal, it can be difficult or impossible to verify the integrity of the codebase, which opens the door for access to credit card data.

Requirement 6. Develop and maintain secure systems and applications

It's simply not enough to "set it and forget it" when it comes to all components of the CDE. A single discovered vulnerability at the network, server, and/or Drupal app layer has the potential for exploitation. Drupal security advisories are posted every Wednesday and should be watched regularly for any updates affecting Drupal core and/or contributed modules a site is using. Likewise, critical operating system and support software (Apache, PHP, MySQL, and other server software.) patches must be applied within 30 days of release.

Requirement 7. Restrict access to cardholder data by business need-to-know

This requirement impacts not only application design, but also account and server management. For example, the server that houses your eCommerce site should not be used as a place where anyone can connect via FTP using a shared password and upload photos from the company picnic. Any one of those individuals' machines could be used to gain authorized access to the CDE code base and/or card data itself.

Requirement 8. Assign a unique ID to each person with computer access

Sharing usernames and passwords leaves less detail for actions occurring for each individual user. It also generally leads to other bad practices, such as creating weak passwords and sharing them over insecure channels like email. Never share your "user 1" or root login information between staff.

Requirement 9. Restrict physical access to cardholder data

Similar to Requirement 1, access to the server must be protected at all times. If a person without proper clearance can physically access a server, they can potentially gain root/administrator access and compromise the database and Drupal application layer. If you outsource hosting, ensure the provider is PCI DSS compliant.

Requirement 10. Track and monitor all access to network resources and cardholder data

Audit trails are critical for identifying changes in the system because a single code or configuration alteration can open up a security vulnerability. Having the ability to identify exactly what changed and who changed it is very important for verifying the integrity of a system.

Requirement 11. Regularly test security systems and processes

It's not enough to have a system that works in theory. Periodic vulnerability scans and full penetration tests are necessary to prove that the system responds as expected. These tests have to be run regularly (at least quarterly) and are ideally run whenever there is a configuration or code change that could introduce a new vulnerability. Ensure you are using an officially Authorized Scanning Vendor (ASV) as listed on the PCI site¹².

Requirement 12. Maintain a policy that addresses information security for all personnel

All the hardware testing, security scans, and audits in the world will not help if people are allowed to use insecure passwords or send full credit card numbers via unencrypted channels like email. The human element is often the weakest link in the security chain. A security policy is required to ensure ALL employees know and understand what is acceptable and what is not with respect to maintaining compliance.

Self-Assessment Questionnaire

Knowing how to get started can seem overwhelming at first. Thankfully, the PCI council has provided detailed reference guides and instructions¹³. All of these materials culminate to your Self-Assessment Questionnaire. This form is essentially a checklist that you must complete and adhere to in order to obtain and maintain PCI compliance.

Your credit card processor and acquiring bank are required to ask you for annual SAQ forms - if they haven't asked yet, they will soon!

The key is determining which SAQ you should be filling out because they vary wildly in terms of quantity of responsibilities and the amount of time and effort it takes to complete them. SAQ A has 14 items and usually can be completed in under a day if not within a couple of hours. SAQ D contains between 326 and 347 items and can take months of time and millions of dollars to achieve¹⁴. The following section will help you make sense of these different levels and how you can select the method that balances your business needs with your ability to achieve compliance.

Determining Your Responsibilities

The quantity and difficulty of your PCI responsibilities are a result of your merchant level (which is based on the volume of transactions) and your validation type (which is based on how you're conducting transactions).

Merchant Level

The PCI council has defined 4 levels of transaction volume (Figure 1)

Level / Tier	Merchant Criteria	Validation Requirements
1	Merchants processing over 6 million Visa transactions annually (all channels) or Global merchants identified as Level 1 by any Visa region ²	<ul style="list-style-type: none"> • Annual Report on Compliance ("ROC") by Qualified Security Assessor ("QSA") or internal auditor if signed by officer of the company • Quarterly network scan by Approved Scan Vendor ("ASV") • Attestation of Compliance Form
2	Merchants processing 1 million to 6 million Visa transactions annually (all channels)	<ul style="list-style-type: none"> • Annual Self-Assessment Questionnaire ("SAQ") • Quarterly network scan by ASV • Attestation of Compliance Form
3	Merchants processing 20,000 to 1 million Visa e-commerce transactions annually	<ul style="list-style-type: none"> • Annual SAQ • Quarterly network scan by ASV • Attestation of Compliance Form
4	Merchants processing less than 20,000 Visa e-commerce transactions annually and all other merchants processing up to 1 million Visa transactions annually	<ul style="list-style-type: none"> • Annual SAQ recommended • Quarterly network scan by ASV if applicable • Compliance validation requirements set by merchant bank

Figure 1. The 4 levels of transaction volumes for VISA card brand. Other card brands (e.g. MasterCard, American Express.) have different reporting and validation requirements. Image Source <http://usa.visa.com/merchants/riskmanagement/cispmerchants.html>

There are a few important points to emphasize for the Visa card brand:

- Compliance begins at transaction #1.
- Even though validation is optional at level 4, compliance is still mandatory.
- If a breach occurs and is reported, a company immediately moves to level 1 regardless of its transaction volume.
- Level 1 validation involves more stringent auditing process, including a mandated third-party audit, which adds additional time and money to stay compliant.

Validation Type

The PCI council defines 8 different validation types:

SAQ	Description
A	Card-not-present merchants (e-commerce or mail/telephone-order) that have fully outsourced all cardholder data functions to PCI DSS compliant third-party service providers, with no electronic storage, processing, or transmission of any cardholder data on the merchant's systems or premises. <i>Not applicable to face-to-face channels.</i>
A-EP*	E-commerce merchants who outsource all payment processing to PCI DSS validated third parties, and who have a website(s) that doesn't directly receive cardholder data but that can impact the security of the payment transaction. No electronic storage, processing, or transmission of any cardholder data on the merchant's systems or premises. <i>Applicable only to e-commerce channels.</i>
B	Merchants using only: <ul style="list-style-type: none"> • Imprint machines with no electronic cardholder data storage; and/or • Standalone, dial-out terminals with no electronic cardholder data storage. <i>Not applicable to e-commerce channels.</i>
B-IP*	Merchants using only standalone, PTS-approved payment terminals with an IP connection to the payment processor, with no electronic cardholder data storage. <i>Not applicable to e-commerce channels.</i>
C-VT	Merchants who manually enter a single transaction at a time via a keyboard into an Internet-based virtual terminal solution that is provided and hosted by a PCI DSS validated third-party service provider. No electronic cardholder data storage. <i>Not applicable to e-commerce channels.</i>
C	Merchants with payment application systems connected to the Internet, no electronic cardholder data storage. <i>Not applicable to e-commerce channels.</i>
P2PE-HW	Merchants using only hardware payment terminals that are included in and managed via a validated, PCI SSC-listed P2PE solution, with no electronic cardholder data storage. <i>Not applicable to e-commerce channels.</i>
D	<p>SAQ D for Merchants: All merchants not included in descriptions for the above SAQ types.</p> <p>SAQ D for Service Providers: All service providers defined by a payment brand as eligible to complete a SAQ.</p>

Figure 2. Breakdown of PCI-DSS version 3.0 SAQ types as a function of merchant activities.

For the purposes of a typical Drupal eCommerce site, the SAQ A, A-EP, and D are the most relevant types^{A,B}. It is also extremely important to understand which type the system falls into because there is a large difference in time, effort, risk, and expense in achieving compliance.

Type	PCI Controls to Meet	Estimated Time	Estimated Cost
A	14	Hours	\$100-\$1,000
A-EP	139	Weeks to Months	\$10,000-\$100,000
D	326-347	Months	\$100,000-\$1,000,000+

Figure 3. Approximate PCI compliance costs per SAQ type. Factors include audits (\$30,000-\$100,000), time spent meeting each requirement, and more.

For many eCommerce stores, selecting a payment method that places them in type C or D can be cost prohibitive. However, achieving SAQ A or A-EP is not always possible given the available payment gateway options that can satisfy a business's feature requirements, which drives the expansion or reduction of the CDE and ultimately determines the validation type.

Drupal Specific Examples

The shopping cart selected (Drupal Commerce, Ubercart, Pay, Stripe, etc) and the payment method within that shopping cart (hosted payment page, direct post, iframe, onsite.) are usually the most significant factors in expanding or reducing the CDE, which directly impacts the SAQ type. Here are some situations that may force one into SAQ D:

- A client that wants complete control over the checkout process may shy away from hosted payment pages, which may be the only viable shared-management option for a particular payment gateway^C.
- A client sometimes must use a particular payment gateway because of an existing contract/business relationship, and that gateway may offer no shared-management options.

Overview of Payment Methods

The PCI council defines 3 types of payment methods: Merchant-managed, Shared-management, and Wholly-outsourced.

Merchant-managed

A general guideline is that if the company's servers store, transmit, or handle cardholder data, then it's merchant managed. An example would be customer submitting a

payment directly on a Drupal site using Ubercart connected to Authorize.Net because a customer's payment information passes through the merchant's webserver on the way to being sent to Authorize.Net. It does not matter if the merchant is storing the data on their servers, since a hacker could gain access if they were able to compromise the server.

Onsite payments posted directly back to Drupal (i.e. an HTTP POST request containing the cardholder data is submitted to the Drupal application, which results in a bootstrap and the passing of the cardholder data through the form API) immediately fall into SAQ D.

Merchant-managed examples:

- Authorize.Net Automated Recurring Billing (ARB)
- Authorize.Net Customer Information Manager (CIM)
- PayPal Payments Standard (PPS)

Shared-management

In a shared-management approach, the credit card information never touches the server that is running the Drupal application. This is generally accomplished by one of 3 approaches:

- Hosted Payment Page (HPP)
- Direct Post
- Inline Frame (iframe)

A HPP approach is where a user is redirected from a Drupal site to a third-party site in order to enter their payment details. If the transaction is successful, the customer is redirected back to the Drupal site with the payment authorization details. Depending on the specific payment gateway's security requirements, these payment authorization details may get sent back to the payment gateway's API to further validate the payment.

Example HPP solutions:

- Authorize.Net Server Integration Method (SIM)
- PayPal Payments Standard (PPS)
- PayPal Express Checkout (EC)

A direct post approach is where the customer remains on the website, but their card data is submitted directly from the customer's browser to the payment gateway and a response is sent directly back to the customer with a one time authentication code or

"token" that is then used by Drupal to immediately communicate with the payment gateway's API to validate the payment. There are two flavors of direct post: setting an HTML form's action attribute to a 3rd party API endpoint or achieving the same with Javascript.

Example direct post solutions:

- Authorize.Net Direct Post Method (DPM)
- Braintree Payments
- Stripe

An iframe approach is where the payment details portion of the checkout form is loaded through an iframe directly from the payment processor. Similar to a direct post strategy, an iframe gives the customer the impression they are always on site while the credit card details are sent directly to the payment gateway.

Example iframe solutions:

- Hosted PCI
- Authorize.Net Hosted Customer Information Manager (CIM)
- PayPal Payments Advanced (PPA)
- PayPal Payflow Link (PFL)

The common denominator in all shared-management configurations is that the checkout process begins on Drupal website (a component of the CDE) managed by the merchant, but the customer is technically sending their credit card credentials directly to the payment processor's CDE. This occurs by redirection (HPP), loading a payment form from their servers (iframe), or posting the form directly through JavaScript or the action attribute on an HTML form (direct post).

One might assume that a shared-management approach would qualify them for SAQ A. However, section 3.4.3 in the *PCI DSS eCommerce Guidelines Supplement* document makes it clear that each shared-management method has vulnerabilities¹⁵, which are described in more detail below. Furthermore, *Understanding the SAQs for PCI DSS v3.0* (in addition to the SAQ A and SAQ A-EP v3.0 forms) specifically states that direct post solutions cannot qualify for SAQ A and must use SAQ A-EP¹⁶.

Frustrating as this may be for those wanting to achieve PCI SAQ A, the good news is that using a shared management system still makes it significantly easier and faster to obtain compliance in SAQ C because a significant amount of the responsibility can still be considered outsourced.

Wholly Outsourced

In a wholly outsourced solution, everything regarding the Drupal application is hosted, managed, and under the responsibility (emphasis added) by a third-party vendor. Please note that most hosting services and Drupal vendors do not explicitly take on that responsibility, so be sure to do your due diligence and ensure that they are equally aware of the repercussions of taking on that responsibility. If the third-party vendor does take on that responsibility, you may be eligible for completing SAQ A^D.

Version 3.0 Disclaimer

Version 3.0 eliminated much of the confusion that existed in version 2.0 with respect to selecting the appropriate SAQ form for each shared-management solution. Previously, one could formulate a strong argument for SAQ A, SAQ C, or some arbitrary hybrid of the two. And given the large difference in the implications for each SAQ type (see Figure 2), it was difficult to confidently make a final determination about each of these shared-management solutions. SAQ A-EP not only introduced a middle ground, but the requirements for SAQ A and SAQ A-EP made it clear with respect to which solution was appropriate for each—iframe and hosted payment page methods are compatible with SAQ A while direct post methods are not.

The decision to allow iframe solutions into SAQ A is not without controversy because one can still make the case that a breach of the Drupal application layer can compromise the delivery of the iframe. Therefore, while the final recommendations of this paper are to use iframe and hosted payment page solutions in order to fall within the scope of SAQ A, it is also recommended to always comply with SAQ A-EP (at a minimum) for security reasons and to future proof your Drupal site against the next versions of the PCI-DSS standard.

There are other important changes introduced in the 3.0 standard, such as the requirement to have all components of the CDE documented with an explicit determination of responsibility for each component. Also noteworthy is that SAQ C is no longer applicable for eCommerce channels, which means that all merchant-managed solutions must comply with the more rigorous SAQ D. For a full list of changes, please see the *Version 3.0 Change Highlights* and *Summary of Changes from PCI DSS Version 2.0 to 3.0* documents^{17,18}.

Selecting the Appropriate Method

SAQ A is obviously desirable and recommended because of its lower risk, time, and cost to implement. However, a company's business needs may require a solution that is

more customizable and that may rule out a wholly outsourced or even shared-management solution.

Example: Recurring Payments for Ubercart on Drupal 7

As of July 2014, there is no wholly-outsourced or shared-management solution for a Drupal 7 website using Ubercart with a recurring billing requirement. To store a customer's credit card information directly on Authorize.Net's servers, one must use the included Ubercart Authorize.Net module and enable the customer information manager (CIM) option. Unfortunately, the way Ubercart implements CIM requires the credit card information to pass through the Drupal application. Therefore this is a Merchant-managed solution, which requires one to adhere to every control item in SAQ D.

Author's Note: payment processors like Authorize.Net have the ability to use third-party iframes to integrate with their CIM service and adding this new functionality into the existing Ubercart modules (core or contrib) would make it significantly easier for merchants to achieve compliance. However, requests to add this functionality for Authorize.Net and other payment gateways have been made with no indication that they will be added to a development roadmap¹⁹.

Additional Considerations

One of the easiest ways to use a shared-management approach is to redirect to a hosted payment page. However, website owners are often resistant to doing this because customers are not always keen on being sent to a third-party site. Additionally, third-party HPPs are not always as customizable in terms of look and feel. Finally, HPP solutions are often more difficult to develop against because one needs to be developing on a public facing URL or IP address in order to get the response back appropriately.

With respect to direct post methods using JavaScript (JS), there is always the consideration that a user may have disabled JS by default and therefore be unable to enter a payment at all without prompting them to adjust their browser security settings.

There are considerations with respect to which shopping cart method to use (or continue to use) on top of Drupal. Drupal Commerce is much more popular for Drupal 7 and popularity brings more people to fix bugs and contribute modules. However, there is still a significant user base using Ubercart, and while Ubercart has far fewer shared-management payment gateway modules that are publicly available, there is nothing precluding the community from creating them in order to address the newer and more

stringent PCI-DSS requirements. The possibility of additional costs to become compliant with Ubercart should be a consideration when evaluating the two solutions.

There are also other payment methods on Drupal, such as the stand alone Stripe and Pay modules, which offer a simpler and smaller feature set than Ubercart and Drupal Commerce and may be more appropriate for one time payment solutions.

Recommendations

There is no one-size-fits-all solution. Each company will have to balance the resources available to become compliant with the features necessary for the business. However, there are some general recommendations that apply across the board. The first is to use Drupal Commerce over Ubercart because Drupal Commerce is more developer focused and has a more consistent code base. It also has more shared-management payment solutions, and that trend is likely to continue. Finally, whenever possible, use a shared-management solution (SAQ A-EP compatible) over a merchant managed solution (SAQ D required) in order to significantly reduce the number of potential security exploits as well as the amount of security controls one has to meet in order to achieve compliance. If available, select an iframe solution (SAQ A compatible) because it further reduces the number of requirements necessary for compliance. However, it's still recommended for those using an iframe or hosted payment page solution to adhere to SAQ A-EP because it'll future proof the CDE as well as adhere to many best practices, which should be implemented regardless.

Drupal Specific Exploits

To further emphasize the importance of adhering to the PCI standard (for both merchant-managed and shared-management payment solutions), we created a list of the many ways in which a compromised website could be configured in order to steal cardholder data.

- **Direct module manipulation.** The codebase could be altered in such a way to email, log, or display cardholder data upon form submission.
- **Hook alters.** Any module could implement a form alter hook to add in additional validation and/or submission functions to grab the cardholder data.
- **Scripting language.** A Flash or JavaScript keylogger or scraper could be used to harvest information as it is typed in and send it to an external server.
- **HTTPS disabling.** A single configuration change can remove HTTPS and user data would be passed unencrypted.

- **Changing credentials.** Hackers could substitute in their own gateway credentials to collect information into a different account.
- **Modifying a direct post.** Modification of the form action or JavaScript code could send the payment information to an alternative server that harvests cardholder data.
- **Modifying an iframe.** Modification of iframe code could be set to load a payment form from another website acting as a man-in-the-middle attack.
- **Modifying a Hosted Payment Page.** Modification of the hosted payment page URL could redirect a customer to an alternative version of the payment site.

For specific ways that the Drupal application can be compromised, please visit DrupalSecurityReport.org.

Locking Down Drupal for PCI SAQ D

This is a non-exhaustive list of ways in which you can harden security at the Drupal level.

- **PCI ASV Scans.** The results will alert you to areas of the site you may have looked over.
- **Hacked module.** This module will download a copy of each module and run a diff against them to ensure the code matches what was provided from Drupal.org.
- **MD5 Check module.** This module will run a checksum of each module and provide a security alert if anything has changed.
- **Deploy full codebase with each release.** This will ensure that any tampered files get replaced.
- **Storing security configurations in code.** Example: one can force secure pages to always be enabled by adding its system variables into the settings.php.
- **Removing credentials from settings.php.** One can include a settings.local.php file within settings.php to ensure that sensitive information is not distributed.
- **Security Updates.** Always keep up with the Drupal Security Advisories, which are released every Wednesday.
- **Paranoia and Security Review Modules.** These modules provide and review your Drupal configurations to ensure they adhere to best security practices.
- **Security Kit Module.** This module provides Drupal with various security hardening options. This lets you mitigate the risks of exploitation of different web application vulnerabilities.
- **Holistic approach to compliance.** Consider PCI DSS compliance as part of the bigger picture of your organization's compliance requirements. Often organizations will have overlapping compliance requirements from standards such as NIST 800-53, ISO27002, HIPAA, FISMA, and others.

Drupal's Security Team

The Drupal security team is a volunteer group of developers who are passionate about keeping Drupal secure. You can follow them by visiting <http://drupal.org/security>, subscribing to their RSS feeds, joining their group on groups.drupal.org, (see <https://www.drupal.org/security-team> for more information).

Final Message to Drupal Developers

The security of a website largely depends on the quality of your work and your attention to detail. Whenever creating custom code, be sure that it complies with best practices. When using other people's code, be sure to review it to ensure that it is also stable, secure, and community supported. When configuring a site, be sure to leverage modules like Security Review and Coder Review to ensure you are not accidentally opening up a security hole.

While you are not necessarily the one responsible for achieving and maintaining a site's PCI compliance, you are responsible for educating yourself about it as well as notifying your employer and/or client when their decisions will impact the site's overall security posture.

Final Message to Drupal Shops

Your responsibility is to protect your business. It is up to you to be exceedingly clear with respect to who is responsible for PCI compliance before, during, and after a site launch. Any service agreements should also include language regarding each party's PCI-DSS compliance responsibilities as well as links to reference materials that provide clients with a means of understanding the implications of these responsibilities.

This conversation will ultimately lead to a greater focus on security for all delivered websites. This not only can be a valuable upsell to current clients, but it can become a competitive advantage and even attract new business.

Finally, if you are a service provider that offers hosting or managed services for a site that handles credit cards and you agree to take on the PCI DSS responsibilities for your clients, you must comply with all PCI DSS requirements and complete SAQ D as a Service Provider, no matter what shopping cart solutions your clients have implemented.

Final Message to Site Owners

Your decisions about how you handle credit card data affect the livelihood of your business. While achieving PCI compliance may seem cost prohibitive, the reality is that a security breach could potentially bankrupt a small to medium-size business. Beyond that, the decisions you make also affect the credit of your customers because identity theft can cost thousands of dollars and take months to years to reconcile. Finally, there can be legal and PR consequences that occur as a result of a breach, further affecting your ability to sustain and grow your business.

Next Steps

Getting started can seem like a daunting task. However, breaking it down into small steps can make this a very manageable process.

- **Assess current setup.** Questions to ask: Do you have a site already or are you starting from scratch? Are you using Drupal Commerce or Ubercart? Are you willing to migrate between them if necessary? Are you willing to outsource completely if sales and/or budgets cannot justify an alternative?
- **Assess requirements.** Questions to ask: Which is more important: cost or features? Do you need any complex interactions or feature sets during checkout? Or can you suffice with using a PayPal button and calling it a day?
- **Reduce CDE Scope.** Questions to ask: Is it possible to segment your network so that non-related servers are not maintained behind the same stateful firewalls as your eCommerce servers?
- **Assess options.** Now that you know what you have to work with and what the goals are, evaluate the cost/effort/risk of 2-3 scenarios.
- **Decide on a method.** Ultimately the business owner must pick a specific method balancing all the factors.
- **Determine responsibilities.** Once a decision is made, it's required to clearly articulate who is responsible for what at each stage in the development cycle. There should also be clear sign off points established.
- **Complete the relevant SAQ:** The Self-Assessment Questionnaire is your key tool for performing a gap analysis and determining whether you are currently compliant.
- **Prioritize PCI responsibilities.** Becoming compliant can take a long time, but some responsibilities are more critical than others. The PCI industry created a document titled The Prioritized Approach to Pursue PCI DSS Compliance, which gives clear guidance on what should be done first to minimize risk during the entire process.

- **Create plan and execute.** Once you've identified what needs to be done and created a plan, it's time to make consistent progress toward checking off items on the list.

At any stage, you also may wish to hire a professional to help expedite the process and ensure that your plan is sound. Specifically, you would want to locate a QSA (Qualified Security Assessor), which are organizations that have been qualified by the PCI Council to have their employees assess compliance to the PCI DSS standard.

Top 8 Drupal PCI Compliance Myths

Author's Note: This is a trimmed down summary from a longer article²⁰.

Drupal is PCI compliant.

This is incorrect by itself because Drupal is only one piece of the cardholder data environment (CDE). However, when Drupal is up to date with all of its security patches and when it's configured properly to meet its portion of the PCI-DSS requirements, then Drupal is PCI *compatible*. PCI compliance can only be achieved at the CDE level once each component of the CDE has met all the requirements within their area of responsibility.

Ubercart and Drupal Commerce are PCI compliant.

This is also incorrect in and of itself simply because it's a component of a larger system. However, the particular payment method chosen within each eCommerce solution can greatly influence how easy it is to become compliant.

I use HTTPS, therefore my Drupal website is secure.

Securing the transaction from the Drupal application to the payment gateway addresses only one of the 12 sections of the PCI standard. There are a significant number of other vulnerabilities that can exist at the server, network, and application level.

I can store numbers/CCV.

Storing the 3-4 digit security code is never allowed under any circumstances. Storing the full credit card number at the Drupal layer is extremely risky and should not be done without a considerable amount of attention and expertise.

Shared-management Methods are 100% Foolproof.

This is false because modifying code at the Drupal application layer can result in a man-in-the-middle attack, the introduction of a keylogger, and other exploits.

I can achieve PCI compliance using shared hosting.

Shared hosting is simply not secure enough for PCI SAQ A-EP or D because there are simply too many users (both customers and employees of the hosting company) that have access to the server and you simply do not have enough control in locking down the system. Technically an iframe solution (SAQ A compatible) could get by on shared hosting, but we strongly recommend against it.

I can achieve PCI compliance using cloud hosting.

As of July 2014, we are seeing a growing number of reputable hosting providers introducing PCI compliant cloud hosting options. However, you must do your due diligence before immediately accepting their claims. As part of the version 3.0 standard, each party must explicitly agree to the particular sections of the standard that they assume responsibility. Not only should the cloud hosting provider explicitly state that their solution is PCI compliant and they will assume the responsibilities within their jurisdiction, but the cloud hosting provider should also be able to provide documentation (upon request) to validate their claims.

Regardless of if you choose to use a cloud or dedicated hardware solution, you must use a PCI DSS certified service provider if you are outsourcing hosting for eCommerce servers.

I can set it and forget it.

PCI compliance is not a single event that is checked off a list and never revisited. Rather, it's a continually changing state. If a security exploit is discovered and disclosed for Drupal or the OS running the server Drupal is hosted on, then your site is not PCI compliant. Therefore PCI compliance is a continual process that needs to be maintained through vigilance.

Summary

Drupal makes it trivial to get an eCommerce site up in minutes, PCI compliance can take months if not setup correctly. And while PCI compliance has many nuances and complexities, it's a mandatory requirement for the 10's of thousands of reported Drupal eCommerce installations. Small, incremental steps in learning and implementation are key in achieving and maintaining compliance, protecting your business, customers, and development.

Sponsors



AppliedTrust provides IT infrastructure, security, and opensource consulting services. The company serves clients in a variety of industries, including healthcare, financial services, recreation and government. AppliedTrust is a PCI DSS Qualified Security Assessor.

www.appliedtrust.com

Cross(Functional)

CrossFunctional is a Sydney-based online solutions provider with proven expertise in Drupal, WordPress and other open source systems. In operation since 2008, we have grown to provide services at multiple levels and across multiple industries.

crossfunctional.net



CARD.com creates Fair, Fashionable and Fun online payments solutions. CARD.com prepaid Visa cards are your connection to what you love. With FDIC insured accounts, ATM acces, smartphone apps and online tools to manage your account, CARD.com provides full service payment solutions. We are always looking for great brands with large communities looking to extend their connection via branded debit cards or other interesting ideas in the payments industry. Our team is growing, see CARD.com/careers.

www.card.com



Commerce Guys is the software company behind Drupal Commerce, the eCommerce solution that capitalizes on the virtues and power of Drupal, the premier open-source content management system. We focus our knowledge and expertise on providing online merchants with the powerful, responsive, innovative eCommerce solutions they need to thrive.

commerceguys.com



We love web design, whatever you call it nowadays. UI/UX, interface design, you name it. Our clients are not just in Denver, CO, but also all over the USA; many are even overseas. We don't just make websites look pretty, though; we develop complicated websites in-house as well. Yes, we may well be the most established provider of Drupal web development services in Colorado.

newmediadenver.com



Townsend Security creates encryption and key management solutions that help organizations meet compliance requirements and mitigate the risk of data breaches. Over 3,000 companies worldwide trust Townsend Security's NIST and FIPS 140-2 validated solutions to meet the requirements in PCI DSS, HIPAA, FISMA, and other regulations. Learn more about Key Connection for Drupal or join our Drupal Developer program.

townsendsecurity.com



Copperly educates businesses in the Drupal community and beyond about cost-effective credit card processing options. We work with businesses to identify the pricing model and implementation best suited to their needs.

www.copperly.com



HostedPCI eliminates the risk associated with handling credit cards by delivering state of the art transaction processing technology that achieves 100% Continuous PCI Compliance, quickly and painlessly. HostedPCI allows merchants of any size to guarantee total protection against credit card theft for their online checkout, call center and mobile transactions.. It's fast & easy to implement, and extremely cost-effective compared to traditional methods.

www.hostedpci.com

References

1. US Census Bureau News CB13-78 - <http://goo.gl/s6e7F>
2. Global eCommerce sales will top \$1.25 trillion by 2013 - <http://goo.gl/DIMff>
3. Introduction to the E-commerce & Internet Business - <http://goo.gl/2uAoo>
4. Drupal Usage trends - <http://goo.gl/IEbTH>
5. PCI DSS - Glossary of Terms, Abbreviations, and Acronyms - <http://goo.gl/IE0Fb>
6. 2010 Annual Study: U.S. Cost of a Data Breach - <http://goo.gl/EoEp6>
7. Heartland Data Breach: MasterCard, Visa Impose Hefty Fines - <http://goo.gl/t0lQT>
8. Data-breach costs take toll on Target profit - <http://goo.gl/aKjIOI>
9. In Data Leaks, Culprits Often Are Mom, Pop - <http://goo.gl/aGYbq>
10. Sony PlayStation suffers massive data breach - <http://goo.gl/Um5cE>
11. Official PCI Security Standards Council Site - <http://goo.gl/Z1lUr>
12. Approved Scanning Vendors - <http://goo.gl/FzzH6>
13. Navigation PCI DSS - <http://goo.gl/H5jOK>
14. New! More! A First Look at the PCI DSS 3.0 SAQs - <http://goo.gl/YnjkJZ>
15. Information Supplement: PCI DSS E-commerce Guidelines - <http://goo.gl/R21rw>
16. Understanding the SAQs for PCI DSS v3.0 - <http://goo.gl/V8ZrVF>
17. Version 3.0 Change Highlights - <http://goo.gl/XDBzky>
18. Summary of Changes from PCI DSS Version 2.0 to 3.0 - <http://goo.gl/7nR1Gt>
19. Integrate authorize.net (Hosted) CIM - <http://goo.gl/rvOfz>
20. Top 12 Drupal PCI Compliance Myths - <http://goo.gl/phZcg1>

Footnotes

- A. If you qualify as a service provider (SP), you must be SAQ D-SP compliant and you must undergo a full ROC audit by a qualified QSA. Basically, services providers are treated the same as being Level 1, no matter how small of a hosting shop you are.
- B. Be advised that a simple matter of handing transactions via fax, terminal, and email in addition to your Drupal site can and often will increase your CDE scope.
- C. It should be noted that the stigma against hosted payment pages continues to fade away as it becomes a more common and accepted form of transaction, particularly in the EU.
- D. Visit <http://www.visa.com/splisting> to confirm whether or not a vendor is a valid Service Provider.

Reviewers

The authors of this paper would like to thank the following individuals for reviewing the paper and providing feedback to improve its accuracy and utility. In alphabetical order: Ryan Cross (rcross), Robert Douglass (robertdouglass), Trent Hein (thein), Michael Hess (mlhess), Dave Long (longwave), Alex Knoll (arknoll), Ryan Szrama (rszrama), Peter Wolanin (pwolanin).

Appendix

Additional Resources

The following is a list of documents that can be found directly from the Official PCI Security Standards Council Site:

- PCI Standards - All Documents - <http://goo.gl/sH1Jx>
- Navigation PCI DSS - <http://goo.gl/lpnno>
- PCI DSS Quick Reference Guide - <http://goo.gl/Z3UZ7>
- The Prioritized Approach to Pursue PCI DSS Compliance - <http://goo.gl/SFZtL>
- Ten Common Myths of PCI SSC - <http://goo.gl/TVtxS>
- PCI DSS Glossary of Terms, Abbreviations, and Acronyms - <http://goo.gl/ab348>
- PCI Security Standards Council - Overview - <http://goo.gl/rQbVs>
- PCI DSS Self-Assessment Questionnaire (SAQ) Instructions and Guidelines - <http://goo.gl/RX0qZ>
- PCI DSS SAQ C V2.0 - <http://goo.gl/HCvY1>

Disclaimer

The authors are not lawyers and the contents of this document do not constitute legal advice. The authors are not responsible or liable for any loss or damages you and/or your business may incur as a result of reading this document. Everyone's PCI compliance needs will be unique to their specific configurations and business needs. It is up to the reader to do their own due diligence and keep up with the latest information found at the PCI Security Standards Council Site.

License

This document is Copyright 2013-2014 Rick Manelius, Greg Knaddison, and Ned McClain Creative Commons Attribution-No Derivative Works 3.0 Unported <http://creativecommons.org/licenses/by-nd/3.0> <http://drupalpcicompliance.org/>. You may share and re-post the PDF on other sites without modification as long as you clearly link to <http://drupalpcicompliance.org/>.