AspDotNetStorefront
e-commerce by design

PABP
CISP/PCI Implementation Guidance

Version 1.3

Created On: 11/6/06

Revision History:

Revision 1.3, 2/6/2007 – R. Anderson
Revision 1.2, 11/6/2006 – R. Anderson
Revision 1.1, 10/9/06 – R. Anderson
Revision 1.0, 10/2/06 – R. Anderson

Table Of Contents

1. Applicable Application Version

1.1. Introduction

AspDotNetStorefront v7.0 has been PABP (Payment Application Best Practices) certified, with AspDotNetStorefront working in conjunction with an independent Visa U.S.A. approved auditing firm.

This document supports the following applications:

- AspDotNetStorefront Version 7.0

This document also explains the Cardholder Information Security Program (CISP), Payment Card Industry (PCI) initiative, and the Payment Application Best Practices (PABP) guidelines. The document then provides specific installation, configuration, and ongoing management best practices for using AspDotNetStorefront as a PABP Application operating in a PCI Compliant environment.

1.2. Visa U.S.A. Reference Documents

The following documents detail the Cardholder Information Security Program (CISP) and related materials (e.g. PABP, PCI, etc):

- http://www.visa.com/cisp. This site includes information such as:
  - Industry Letter to Merchants
  - CISP Overview
  - PCI Data Security Standard
  - View all CISP downloads
- A copy of the Payment Card Industry (PCI) Data Security Standard from Visa's security website is available at the following Internet address:

  http.s://www.pcisecuritystandards.org/tech/download_the_pci_dss.htm
- OWASP.ORG/open web app security policies (general policies, etc)
- Additional information for merchants from Visa is available at the following Internet address:

  http://usa.visa.com/business/accepting_visa/ops_risk_management/cisp_merchants.html?it=il|/business/accepting_visa/ops_risk_management/cisp.html|Merchants
- A listing of qualified security assessors from Visa is available at the following Internet address:

  http://usa.visa.com/business/accepting_visa/ops_risk_management/cisp_accessors.html?it=l2|/business/accepting_visa/ops_risk_management/cisp_merchants%2Ehtml|Assessors

We worked with the following Visa U.S.A. approved certification firm on our PABP Certification:

Coalfire Systems, Inc.
Louisville, Colorado:
361 Centennial Parkway,
Suite 150
Louisville, CO 80027
Phone: 303-554-6333
Fax: 303-554-7555

We also work closely with ControlScan.com, which can offer PCI scanning services, which we also use on our own production site (aspdotnetstorefront.com).

In summary, PABP is the standard against which AspDotNetStorefront has been tested and certified.

PCI Compliance is then later obtained yourself (the merchant) on your actual server (or hosting) environment to certify PCI compliance by using various hardware scans, port scans, and configuration evaluation testing methods. This is further defined in the following section. We go into this definition because many users and merchants are not clear as to the differences between "PABP" and "PCI Compliance". While very closely inter-related, they are different.

1.3. Understanding "PABP" vs "PCI Compliance"

As a software vendor, our responsibility is to be "PABP Compliant".

While this is not currently required by Visa U.S.A., as an industry leader in the asp.net shopping cart e-commerce market, we felt it was important to take a leading position and obtain this certification.

We have performed an audit and certification compliance review with our independent auditing firm, to ensure that our platform does conform to industry best practices when handling, managing and storing payment related information.

Note: We want to reiterate that obtaining "PCI Compliance" falls on you (the merchant) and your hosting provider, working together, using PCI compliant server architecture with proper hardware & software configurations and access control procedures. We do outline the PCI ready environment that we performed our testing on at EDTHosting.com. Additionally, EDTHosting.com offers PCI ready hosting plans, suitable for installing AspDotNetStorefront, if you need. Other hosting providers may also offer PCI Compliant hosting options.

We have tested and certified to the Visa U.S.A. "Payment Application Best Practices" (PABP) standard, to ensure that when you load AspDotNetStorefront into an environment equivalent to our recommended PCI ready environment, that our application is also following best practices, helping you achieve PCI Compliance easily with respect to how AspDotNetStorefront handles user accounts, passwords, encryption, and other payment data related information.

After installation and initial certification to PCI standards, you should then follow our recommended operational guidelines, defined later in this document, to ensure continued best practices for management of your storefront.

Visa U.S.A. specifies different levels of compliance requirements, driven mostly by the annual transaction volume of your storefront. You should read the documentation provided by Visa (see above) to determine the level of PCI Compliance required for your business.

2. Introduction

2.1. The Payment Card Industry (PCI) Data Security

Since your object as a merchant is to obtain PCI Compliance, we further define the PCI requirements here, to facilitate your compliance process. Remember, AspDotNetStorefront has already been “PABP” certified, so that alleviates the concern that our application could pose security risks that might prevent you from obtaining PCI Compliance. That is why we have obtained the PABP certification.

Systems which process payment transactions necessarily handle sensitive cardholder account information, such as credit card numbers and user passwords.

The Payment Card Industry (PCI) Data Security Standard is a worldwide standard for payment card and consumer financial data protection. It incorporates the requirements of the Visa U.S.A. Cardholder Information Security Program (CISP) and the Visa International Account Information Security (AIS) program, the MasterCard International Site Data Protection (SDP) program, as well as the security requirements of American Express DSS, DiscoverCard DISC and the Japan Credit Bureau (JCB).

The Payment Card Industry (PCI) has developed security standards for handling cardholder information in a published standard called the PCI Data Security Standard (DSS). The security requirements defined in the DSS apply to all members, merchants, and service providers that store, process or transmit cardholder data.

To be in compliance with this standard, all of your company's Internet connections, assigned IP addresses, and all Internet connected servers (Web, email, DNS, etc.) must have no level 3, 4 or 5 severity vulnerabilities in their most recent security audit. Audits must be conducted at least every 90 days. Various firms can assist with your scans, including Coalfire Systems, ControlScan, or other firms.

The PCI DSS requirements apply to all system components within the payment application environment which is defined as any network device, host, or application included in, or connected to, a network segment where cardholder data is stored, processed or transmitted.

The following high level 12 Requirements comprise the core of the PCI DSS:

Build and Maintain a Secure Network

1. Install and maintain a firewall configuration to protect data
2. Do not use vendor-supplied defaults for system passwords and other security parameters

Protect Cardholder Data

3. Protect Stored Data
4. Encrypt transmission of cardholder data and sensitive information across public networks

Maintain a Vulnerability Management Program

5. Use and regularly update anti-virus software
6. Develop and maintain secure systems and applications

Implement Strong Access Control Measures

7. Restrict access to data by business need-to-know
8. Assign a unique ID to each person with computer access

9. Restrict physical access to cardholder data

Regularly Monitor and Test Networks

10. Track and monitor all access to network resources and cardholder data
11. Regularly test security systems and processes

Maintain an Information Security Policy

12. Maintain a policy that addresses information security

The remainder of this document describes the essential guidance for installing, configuring, and managing AspDotNetStorefront in a PCI ready environment.

3. PCI DSS Payment Application Environment Requirements

3.1. Access Control

The PCI DSS requires that access to all systems in the payment processing environment be protected through use of unique users and complex passwords. Unique user accounts indicate that every account used is associated with an individual user and/or process with no use of generic group accounts used by more than one user or process. Additionally any default accounts provided with operating systems, databases and/or devices should be removed/disabled/renamed as possible, or at least should have PCI DSS compliant complex passwords and should not be used. Examples of default administrator accounts include "administrator" (Windows systems), "sa" (SQL/MSDE), and "root" (UNIX/Linux).

The PCI standard requires the following password complexity for compliance (often referred to as using "strong passwords"):

- Passwords must be at least 7 characters
- Passwords must include both numeric and alphabetic characters
- Passwords must be changed at least every 90 days
- New passwords can not be the same as the last 4 passwords

PCI user account requirements beyond uniqueness and password complexity are listed below:

- If an incorrect password is provided 6 times the account should be locked out
- Account lock out duration should be at least 30 min. (or until an administrator resets it)
- Sessions idle for more than 15 minutes should require re-entry of username and password to reactivate the session.

These same account and password criteria must also be applied to any applications or databases included in payment processing to be PCI compliant.

AspDotNetStorefront, as tested to in our PABP audit, meets, or exceeds these requirements.

3.2. Remote Access

The PCI standard requires that if employees, administrators, or vendors are granted remote access to the payment processing environment; access should be authenticated using a two-factor authentication mechanism (username/ password and an additional authentication item such as a token or certificate).

In the case of vendor remote access accounts, in addition to the standard access controls, vendor accounts should only be active while access is required to provide service. Access rights should include only the access rights required for the service rendered, and should be robustly audited.

3.3. Non-Console Administration

Users and hosts within the payment application environment may need to use third-party remote access software such as Remote Desktop (RDP)/Terminal Server, pcAnywhere, etc. to access other hosts within the payment processing environment. However, to be compliant, every such session must be encrypted with at least 128-bit encryption (in addition to satisfying the requirement for two-factor authentication required for users connecting from outside the payment processing environment). For RDP/Terminal Services this means using the high encryption setting on the server, and for pcAnywhere it means using symmetric or public key options for encryption. Additionally, the PCI user account and password requirements will apply to these access methods as well.

3.4. Wireless Access Control

The PCI standard requires the encryption of cardholder data transmitted over wireless connections. The following items identify the PCI standard requirements for wireless connectivity to the payment environment:

- Firewall/port filtering services should be placed between wireless access points and the payment application environment with rules restricting access
- Use of appropriate encryption mechanisms such as VPN, SSL/TPS at 128 bit, WEP at 128 bit, and/or WPA
- If WEP is used the following additional requirements must be met:
    - Another encryption methodology must be used to protect cardholder data
    - If automated WEP key rotation is implemented key change should occur every ten to thirty minutes
    - If automated key change is not used, keys should be manually changed at least quarterly and when key personnel leave the organization
- Vendor supplied defaults (administrator username/password, SSID, and SNMP community values) should be changed
- Access point should restrict access to known authorized devices (using MAC Address filtering)

AspDotNetStorefront does not use wireless transmission means as shipped.

3.5. Transport Encryption

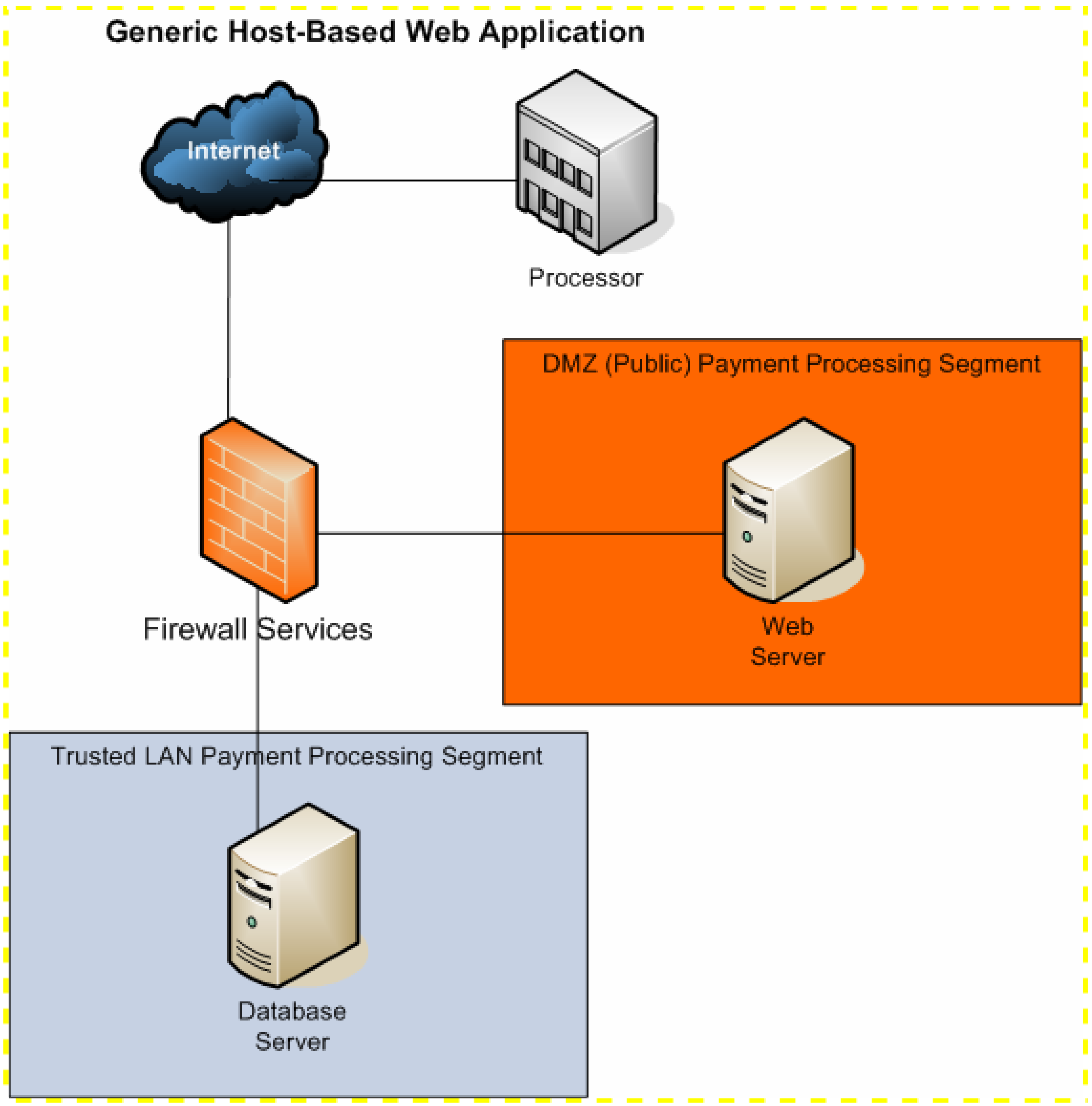
The PCI DSS requires the use of strong cryptography and encryption techniques with at least a 128 bit encryption strength (either at the transport layer with SSL or IPSEC; or at the data layer with algorithms such as RSA or Triple-DES) to safeguard sensitive cardholder data during transmission over public networks (this includes the Internet and Internet accessible DMZ network segments).

Additionally, PCI requires that cardholder information is never sent via email without strong encryption of the data. AspDotNetStorefront does not transmit card information via e-mail. The use of a properly installed 128 bit SSL certificate, available from your hosting provider, meets this requirement. AspDotNetStorefront should then be configured to "go secure" on any page that involves sensitive data (login pages, account pages, cart pages, payment pages, etc).

3.6. Network Segmentation

The PCI DSS requires that firewall services be used (with NAT or PAT) to segment network segments into logical security domains based on the environmental needs for internet access. Traditionally, this corresponds to the creation of at least a DMZ and a trusted network segment where only authorized, business-justified traffic from the DMZ is allowed to connect to the trusted segment. No direct incoming internet traffic to the trusted application environment can be allowed. Additionally, outbound internet access from the trusted segment must be limited to required and justified ports and services.

A simplified high-level diagram of the PCI ready environment that we performed PABP certification on is shown below. This environment was created in full at EDTHosting.com, where AspDotNetStorefront was installed, and audited. You can use this or equivalent architecture to achieve PCI compliance. Check with your preferred hosting provider also to review and analyze if their supported environments conform to PCI DSS requirements..


Generic Host-Based Web Application
Internet
Processor
DMZ (Public) Payment Processing Segment
Web Server
Firewall Services
Trusted LAN Payment Processing Segment
Database Server

3.7. Information Security Policy/Program

In addition to the preceding security recommendations, a comprehensive approach to assessing and maintaining the security compliance of the payment application environment is necessary to protect the organization and sensitive cardholder data.

The following is a very basic plan every merchant/service provider should adopt in developing and implementing a security policy and program:

- Read the PCI DSS in full and perform a security gap analysis. Identify any gaps between existing practices in your organization and those outlined by the PCI requirements.
- Once the gaps are identified, determine the steps to close the gaps and protect cardholder data. Changes could mean adding new technologies to shore up firewall and perimeter controls, or increasing the logging and archiving procedures associated with transaction data.
- Create an action plan for on-going compliance and assessment.
- Implement, monitor and maintain the plan. Compliance is not a one-time event. Regardless of merchant or service provider level, all entities should complete annual self-assessments using the PCI Self Assessment Questionnaire.
- Call in outside experts as needed. Visa has published a Qualified Security Assessor List of companies that can conduct on-site CISP compliance audits for Level 1 Merchants, and Level 1 and 2 Service Providers. MasterCard has published a Compliant Security Vendor List of SDP-approved scanning vendors as well.

  NOTE: We highly recommend contacting Coalfire Systems, the PABP auditing firm we used for AspDotNetStorefront.

4. AspDotNetStorefront Application Configuration

4.1. Baseline System Configuration

Below are the operating systems and dependent application patch levels and configurations supported and tested for continued PCI DSS compliance:

- Microsoft Windows 2000 Service Pack 4, Windows XP Professional with Service Pack 2. All latest updates and hot-fixes should be tested and applied.
- 256 MB of RAM minimum, 2GB or higher recommended for AspDotNetStorefront
- 200 MB of available hard-disk space
- TCP/IP network connectivity
- SQL Server 2000. All latest updates and hot-fixes should be tested and applied.

In our EDTHosting.com PCI ready environment, we used this setup:

- Microsoft Windows 2003 Server with the latest Service Packs and Updates.
- 4 GB RAM
- 73 GB minimum drive space free.
- TCP/IP network
- SQL Server 2000. All latest updates and hot-fixes applied.

5. AspDotNetStorefront Initial Setup & Configuration

This section documents in detail the configuration options and settings important for installing and configuring AspDotNetStorefront in a PCI ready application environment.

5.1. PCI Environment

Create, or select, a PCI ready hosting or server environment for your site. Many hosting providers offer PCI ready configurations already (e.g. EDTHosting.com).

5.2. Install AspDotNetStorefront

Next, follow our normal installation manual to install AspDotNetStorefront v7.0 or later with all default settings.

Special installation steps to remember, which are also documented in our main manual for performing a "best practices" installation:

- Remember, you should rename your "admin" directory to a new name, which is not easily guessed (e.g. "adm433954forme"). The name can be whatever you choose. You must also then set the

  AppConfig:AdminDir={that same directory name}

- Enable Windows NT Authenticated login for your admin directory. You configure this setting on the server using the IIS Manager. This adds a second level of authentication login protection to the admin control panel for your storefront.
- Set your initial EncryptKey in your /web.config file to a sufficiently long string, that is not easily guessed. Follow similar rules to how a strong password is created (e.g. at least 7 characters long + 1 digit + 1 special character + mix of upper & lower case characters)
- Set your DBConn string so the site can connect to your database. For maximum security, use a windows authenticated login per the instructions documented in the web.config file.
- After initial installation and login to your admin site, CHANGE your administrator e-mail address and password immediately! Do not wait to do this, or you may forget!
- Install into a medium trust (asp.net) environment per standard Microsoft documentation.

5.3. Select Your Payment Gateway

The AspDotNetStorefront platform supports a number of industry standard payment gateways. Contact your merchant account provider for their list of supported and/or recommended gateways. A number of industry standard gateways have already been fully integrated with the AspDotNetStorefront platform such as:

- Authorize.net
- PayPal Website Payments Pro

- Verisign Payflow PRO
- Cybersource
- Paymentech
- etc... (see the supported gateway list on our main site)

Your merchant account or gateway provider will then assign your gateway account settings. You will then follow our normal manual to configure this gateway within the storefront. You must be the admin "super user" to setup any gateway settings. Follow all manual instructions to configure your gateway.

5.4. Obtain and Install 128 bit SSL Certificate

You must then obtain and properly install the SSL certificate for your site. Your selected hosting provider will usually assist you in this step. Once installed, you must tell AspDotNetStorefront to use SSL by doing the following steps:

- Set AppConfig:UseSSL=true
- Set AppConfig:LiveServer=yourdomain.com
- Set AppConfig:RedirectLiveToWW=true

These settings assume the SSL certificate is installed on https://www.yourdomain.com, which is the normal configuration. Our main manual instructions document how to change AppConfig settings. AppConfig settings control the configuration of various storefront settings. Many AppConfig settings are ONLY visible to the Admin Super User. See main manual again for further information.

After setting these SSL settings, you should forcefully restart the web site (by editing web.config file or by using IISReset) to ensure all new settings are indeed active.

5.5. Enable CAPTCHA Prevention on Admin Login Pages

We recommend that you enable the CAPTCHA "completely automated public Turing test data entry requirement on the admin sign in pages. To do this, set:

- AppConfig:SecurityCodeRequiredOnAdminLogin=true

This helps prevent automated BOT (robot) sign-in attacks on your admin site, which of course should never even get that far, as you enabled IIS NT authentication on that admin site, which also no-one should know where it is ☺

5.6. Run Your Store Configuration Wizard

The configuration wizard can assist you with making many initial AppConfig settings for your storefront. To run the configuration wizard, log into your admin site as the super admin user, and then click the "Run Configuration Wizard" link on the home page.

In the configuration wizard, you can select which payment gateway you will be using, whether to use live or test transaction mode, and you can enter the store name, primary e-mail address, etc.

> **NOTE: One of the sections in the Configuration Wizard is to "encrypt your web.config file". Remember to select that option!**

5.7. Conduct Test Transactions

Perform any necessary test mode transactions to ensure that your payment gateway is configured properly. When you are satisfied that transactions are ready to go, and you want your store to go "live", set:

- AppConfig:UseLiveTransactions=true

5.8. On Upgrade to v7.0 (PABP)

All admin users should reset their passwords to strong passwords after the site has been updated from a pre 7.0 release. We are not doing this in the code, as there are complicated issues that we cannot predict how to do it properly for all customer installations, so we are putting this in our release notes here as procedural guidance. Do not omit this step please.

5.9. Choose NOT to Store Credit Cards At ALL!!

Most modern payment gateways (like authorize.net) do NOT require that you store credit cards in the storefront database.

They return a transaction id which AspDotNetStorefront can then later use to do void, capture, and refund transactions against. So our recommendation is to NEVER store credit cards on your site. The only time you have to store credit cards is if you:

- Use Recurring Billing Products
- Process your Transactions "off line" somehow, using the Manual Gateway (not recommended)

Check with your payment gateway provider to ensure they do not require the storage of credit cards. If they do, we might suggest you consider using a more modern payment gateway.

To indicate whether or not to store credit cards (encrypted) in the db, set:

- AppConfig:StoreCCInDB=true or false

If you are not storing credit cards, the only "extra" step for repeat customers is that on checkout of subsequent orders, they must just re-enter their credit card again. This is a very very minor inconvenience, and we think the tradeoff is well worth it to not store the credit cards in the database!

CardExtraCodes (CVV2) values are NEVER kept for longer than is needed on the immediate transaction.

NOTE FOR MANUAL GATEWAY USERS: This means that if you are using the MANUAL gateway to process your credit cards "offline" do not ask us to store the CVV2 code for you. We will not do it. You must process those transactions in an offline manner without using the CVV2 code. It is just a slightly higher per transaction cost from your merchant account. Again, do not ask us to modify the code to store the CVV2. We get this request repeatedly, and we will not do any such code modification.

5.10. When Storing Credit Cards, Perform Proper Maintenance!!

If you do need to store credit cards in the database, we will fully encrypt + salt the values. Your EncryptKey in /web.config is used for this.

We then also recommend that you periodically:

- Update your EncryptKey, using the Admin -> Misc -> Change EncryptKey feature in your storefront to cycle to a new EncryptKey every 90 days. This page is only available to the admin super user.
- We recommend that you FULLY REMOVE them after 30 (or at most 60) days after the orders are placed to accommodate normal industry standard return policies. It is almost never needed to retain credit cards for orders that are older than 30/60 days. To perform this maintenance, select the Admin -> Monthly Maintenance menu option in your admin site. This page is only available to the admin super user also.

Related Topic: Note that all user passwords are also fully salted + hashed in the database. This means that users cannot ever recover a lost password, only request a new "one time use" only password. See our main manual for more information.

5.11. Properly Train and Monitor Admin Personnel

It is your responsibility to institute proper personal management techniques for allowing admin user access to credit cards, site data, etc. You can control whether each individual admin user can see credit cards (or only last 4).

In most systems, security breach is the result of unethical personnel. So pay special attention to whom you trust into your admin site and who you allow to view full decrypted payment information.

You can set on each admin user profile, in the admin site, whether they can view full credit card information, or not. In nearly all cases, the viewing of full credit card information is rarely required. The admin site void, capture, and refund screens seldom require this information.

5.12. Super Admin Key Management Roles & Responsibilities

As part of normal PABP operations, you will also need to train and ensure that the Super Admin personnel understand, accept and perform duties in their role as Encrypt key custodians. As a custodian, they are responsible for the safeguarding of these chosen keys, and for performing routine updates to the keys as a normal course of business, defined periodically herein.

As part of this role, there are specific responsibilities of supplying and securing the EncryptKey and to properly use the option of encrypting the web.config file using tools provided in the storefront administration site. The Super Admin must also maintain guardianship over the salt keys chosen for order record encryption, and for following recommended periodic key changes using the supplied Super Admin pages in the store administration site.

Super Admin users must also be required to sign an acknowledgement that they know, understand, and have been properly prepared to perform these responsibilities. Additionally, Super Admin users/Key Custodians should fully familiarize themselves with PCI compliance guidelines, as those will contain the most relevant and accurate security procedures in addition to AspDotNetStorefront specific guidelines in this document.

6. PABP Operational Considerations

Once installed, and operational, please be sure you understand how various admin, login, and other actions are handled by the storefront. You will also have to train your customer service personnel to understand how to properly advise customers. In some cases, increased security comes at slightly decreased customer "friendliness". For example, if a customer loses their password, they can no longer recover it.

The customer must request a new one-time-use password, and then change that again after next successful login. Many store administrators will view this type of annoyance for their customers with great frustration.

Similarly, an admin user can no-longer see a customer's password. So they cannot offer any assistance to the remembering of a customer's password.

We recommend that you follow our default settings for the storefront however, as the decrease in customer "friendliness" is very minor.

Various PABP related operational features & considerations are documented below.

6.1. Customer Lock Out

Each customer database record now has a locked out flag. The admin super user can set this on any customer record to lock that user out of the site. You can do this on the admin site.

6.2. Admin Requires Strong Passwords

For admin users, unique & complex passwords must be required and enforced. Complex password is defined as 7 chars: uppercase + lowercase + number + symbol. For normal store users, it should be AppConfig controlled about whether complex passwords are required for normal store user accounts. The default setting is to just require strong passwords on admin users.

6.3. Data Encryption

Version 7.0+ of AspDotNetStorefront uses a completely new encryption method. Hash codes are using 256 bit keys. Salt keys are now chosen by the web storefront admin for the address and orders tables, so every single record can have a separate salt value.

6.4. Require Periodic EncryptKey changes.

The admin site now reminds you every 90 days to change your EncryptKey. When you do this, ALL encrypted data will be unencrypted and re-encrypted using the new key. Existing encryption salt keys will still be used.

6.5. Encrypted Data Now Salted

The store super user can now select a salt key from various address table record fields, and order table record fields, so theoretically, each store and database record can have a different salt key in effect. This strengthens the encryption. Should any single order record encryption be compromised, it will not lead to compromise of any other record (same for address table).

6.6. Web.config Encryption

This is now fully supported, and performed in the "Configuration Wizard" in the admin site, where the admin super user can specify to encrypt their web.config file. The default encryption is the Windows Data Protection API. The site owner can specify to RSA encryption be used if key portability is needed in a server farm environment. The encryption used is separate and different from the credit card encryption.

6.7. Admin User Disallow Recent Passwords

AspDotNetStorefront now saves and disallows the last 4 passwords for admin users so they cannot use them again right away. We also force changing of admin passwords on AppConfig defined interval. Disallow any of prior last 4 password's to be used on change. Default interval should be set to 30 days. Store admin can change this policy if they want.

6.8. Admin Users Can Lock Themselves Out!!

The sign-in pages on the admin site and store site execute logic to detect the number of invalid login attempts > threshold (AppConfig controlled) and then locks that user out from the site for an AppConfig controlled period of time (defaulted to 30 mins).

Note that even the admin super users can LOCK THEMSELVES OUT OF THEIR OWN SITE for while!!!

6.9. Password Storage

In v7.0+, we changed the password storage from an encrypted method to a (random!) salt + hash method for compliance. Passwords can no longer be recovered. If a password is lost, the user must request a new one time use (strong) password which will be automatically generated and their record is marked as "must change password" on next login.

6.10. Session Expiration

The central customer & session management logic now sets the expiration for forms authentication to be 15 minutes.

6.11. Admin Users Cannot Say "Remember Me"

The "remember me" checkbox has been removed from the admin site. Additionally, a CAPTCHA "completely automated public Turing test" field can be enabled on both public facing store login pages and admin login pages to prevent the use of BOT attacks.

6.12. All Password Changes Logged

The site now logs all admin password changes in the Security Log table. This table is a one year write only table. There is no update, delete, or alter facilities provided on it. The admin super user ONLY can view this log. It is automatically aged at one year duration.

6.13. Admin User Segregation

1.1 Admin users cannot see each other's passwords, and additionally admin users now cannot even view other admin user profiles or accounts, period. Only the super admin can do this.

6.14. Downloading Orders Into QuickBooks

We support and allow the downloading of orders, including credit card information into QuickBooks. We will decrypt and securely download (via https) into QuickBooks, using your authenticated connection from T-HUB (our QuickBooks export utility). You are fully responsible for management of that downloaded, after it is pulled from the storefront database, and should follow all CISP/PCI best practices methods for managing that data.

Your PC then becomes an extension of your CISP/PCI environment, and is subject to all of the same requirements!

6.15. Security Log

AspDotNetStorefront now logs the following to a write only, encrypted, one-year automatically aging Security Log:

- Password changes
- “super admin” marked AppConfig changes
- All credit card clear text retrieval events
- Any access to unencrypted credit card information is also logged along with which admin user viewed the information.
- Any attempt to view credit card numbers which fail (this should never occur in practice)
- All admin logins, logouts, failed logins, and password changes
- Any time admin user rights are added, or removed from any user account.

The Security Log is encrypted, and not viewable except by the admin super user. AspDotNetStorefront now uses a 2-Pass Data database field "delete" of sensitive data. First, we do a write over the data with 111111 (just for example) and then we perform a NULL overwrite over that. This can prevent SQL paging from keeping that old data on the disk.