# Requirement 6.1 to 6.2

# Security Patch Management Installation Policy and Procedures

## 6.1 to 6.2 Overview

In accordance with Payment Card Industry Data Security Standards (PCI DSS) requirements, *[company name]* has established a formal policy and supporting procedures concerning security patch management. This policy is to be implemented immediately. It will be evaluated on a(n) *[annual, semi-annual, quarterly]* basis for ensuring its adequacy and relevancy regarding *[company name]*’s needs and goals.

## 6.1 to 6.2 Policy

Security patch management (patch management) has become a critical security issue due in large part to the exploitation of information technology systems from numerous external and internal sources. Consequently, all system components directly associated with the cardholder data environment must be securely hardened and configured with all necessary and appropriate patches and system updates for preventing the exploitation or disruption of mission-critical services. Similarly, all IT resources not directly associated with the cardholder data environment must also be securely hardened and configured with all necessary and appropriate patches and system updates in order to prevent the exploitation or disruption of mission-critical services.

In accordance with best practices for security patch management, the subsequent three (3) security concerns will be highlighted throughout the Security Patch Management policy. They are as follows (NIST, n.d.):

* **Vulnerabilities**: Software flaws or a misconfiguration that may potentially result in the weakness in the security of a system within the system components directly associated with the cardholder data environment or any other IT resources
* **Remediation:** The three (3) primary methods of remediation are (1) installation of a software patch, (2) adjustment of a configuration setting and (3) removal of affected software.
* **Threats:** Threats are capabilities or methods of attack developed by malicious entities to exploit vulnerabilities and potentially cause harm to a computer system or network. Common examples are scripts, worms, viruses and Trojan horses.

Failure to keep system components and other IT resources patched securely and on a consistent basis can cause unwanted damage to all environments directly associated with the cardholder environment. This includes but is not limited to the following:

* Network devices and all supporting hardware and protocols
* Operating systems within the development and production environments
* Applications within the development and production environments
* Any other mission-critical resources within the cardholder data environment that require patches and security updates for daily operations

Additionally, a Security Patch Management Program (SPMP) is to be implemented, which consists of the following initiatives:

* A formalized Security Patch Management Program employee, complete with his/her roles and responsibilities
* Comprehensive inventory of all system components directly associated with the cardholder environment
* Comprehensive inventory of all other IT resources not directly associated with the cardholder environment
* Subscribing to industry-leading security sources, additional supporting resources for vulnerability announcements and other security patch management alerts and issues
* Procedures for establishing a risk ranking regarding security patch management. This will include but is not limited to (1) the significance of the threat, (2) the existence and overall threat of the exploitation and (3) the risks involved in applying security patch management procedures (its effect on other systems, resources available and resource constraints).
* The creation of a database of remediation activities that needs to be applied
* Test procedures for testing patches regarding remediation
* Procedures for the deployment, distribution and implementation of patches and other related security-hardening procedures
* Procedures for verifying successful implementation of patches and other related security-hardening procedures

## 6.1 to 6.2 Procedure

*[Company name]* has developed and implemented a comprehensive program regarding security patch management, which encompasses the categories and supporting activities listed below. These policy directives will be fully enforced by *[company name]* for ensuring the Security Patch Management Program (SPMP) initiatives are executed in a formal manner and on a consistent basis for all system components within the cardholder data environment and all other IT resources.

### Security Patch Management Program Employee

This individual will be responsible for coordinating, facilitating and undertaking all necessary activities regarding security patch management policies and procedures. Additionally, this individual will have the necessary information technology and security expertise to successfully execute all steps as required. Specifically, this individual will have a strong working knowledge of vulnerability and patch management, as well as system administration, intrusion detection and firewall management.

#### Table 6.1.a

#### Security Patch Management Program Employee

|  |  |  |
| --- | --- | --- |
| **Name** | **Title** | **Contact Information** |
| *Jason Smith* | *Senior Network Engineer* | *smith@company.com* |
| *Mike Larson* | *Backup Network Engineer* | *Mlarson@company.com* |
| *?* | *?* | *?* |
| *?* | *?* | *?* |
| *?* | *?* | *?* |

### Comprehensive Inventory of All System Components Directly Associated with Cardholder Environment

The following table includes all system components that are directly associated with the cardholder environment. These system components are to be listed by network devices, operating systems, applications and any other system components as needed.

#### Table 6.1.b

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **System Components** | **Host Name** | **Physical Location** | **Owner of System Components** | **Primary Use in Cardholder Data Environment** |
| *?* | *?* | *?* | *?* | *?* |
| *?* | *?* | *?* | *?* | *?* |
| *?* | *?* | *?* | *?* | *?* |
| *?* | *?* | *?* | *?* | *?* |
| *?* | *?* | *?* | *?* | *?* |
| *?* | *?* | *?* | *?* | *?* |
| *?* | *?* | *?* | *?* | *?* |
| *?* | *?* | *?* | *?* | *?* |
| *?* | *?* | *?* | *?* | *?* |
| *?* | *?* | *?* | *?* | *?* |
| *?* | *?* | *?* | *?* | *?* |
| *?* | *?* | *?* | *?* | *?* |
| *?* | *?* | *?* | *?* | *?* |
| *?* | *?* | *?* | *?* | *?* |
| *?* | *?* | *?* | *?* | *?* |

### Comprehensive Inventory of all other IT Resources Not Directly Associated with Cardholder Environment

The following table includes all other IT resources not directly associated with the cardholder environment. These IT resources, however, are still considered critical to the daily operations of *[company name]*.

#### Table 6.1.c

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **IT Resources** | **Host Name** | **Physical Location** | **Owner of IT Resources** | **Primary Use within Organization** |
| *?* | *?* | *?* | *?* | *?* |
| *?* | *?* | *?* | *?* | *?* |
| *?* | *?* | *?* | *?* | *?* |
| *?* | *?* | *?* | *?* | *?* |
| *?* | *?* | *?* | *?* | *?* |
| *?* | *?* | *?* | *?* | *?* |
| *?* | *?* | *?* | *?* | *?* |
| *?* | *?* | *?* | *?* | *?* |
| *?* | *?* | *?* | *?* | *?* |
| *?* | *?* | *?* | *?* | *?* |

### Industry-Leading Security Sources and Additional Supporting Resources

Various external security sources and resources are utilized to ensure that *[company name]* maintains awareness of security threats, vulnerabilities and what respective patches, security upgrades and protocols are available.

Currently, *[company name]* subscribes to the following types of security sources and resources (NIST, n.d.):

* Vendor websites and email alerts
* Vendor mailing lists, newsletters and additional support channels for patches and security
* Third-party websites and email alerts
* Third-party mailing lists
* Online forums and discussion panels
* Conferences, seminars and trade shows

Listed below are the specific security resources and sources to which *[company name]* subscribes for patch management, alerts, security and support as applicable:

#### Table 6.1.d

#### Online Resources for Patch Management, Alerts, Security and Support, As Applicable

|  |  |  |
| --- | --- | --- |
| **Vendor/Provider and Type of System** | **Website** | **Other** |
| *CISCO* | [**http://www.cisco.com/warp/public/707/cisco-sa-20030717-blocked.shtml**](http://www.cisco.com/warp/public/707/cisco-sa-20030717-blocked.shtml) | *Security Advisory Alert Board* |
| *IBM AIX* | [**http://www-03.ibm.com/systems/power/software/aix/service.html**](http://www-03.ibm.com/systems/power/software/aix/service.html) | *AIX support and alert website* |
| *Microsoft* | [**http://technet.microsoft.com/en-us/wsus/default.aspx**](http://technet.microsoft.com/en-us/wsus/default.aspx) | *Windows Server Update Services (WSUS)* |
| *Oracle* | [**http://www.oracle.com/technology/deploy/security/alerts.htm**](http://www.oracle.com/technology/deploy/security/alerts.htm) | *Critical Patch Updates and Security Alerts* |
| *Apache* | [**http://www.apache.org/dist/httpd/patches**](http://www.apache.org/dist/httpd/patches) | *Official Patches for Apache* |
| *?* | *?* | *?* |
| *?* | *?* | *?* |
| *?* | *?* | *?* |
| *?* | *?* | *?* |

*Please note: This is just a sample used to illustrate how this section should be completed. For an in-depth listing of all vendors, providers, their products and respective websites, please view Appendix D from the following URL:* <http://csrc.nist.gov/publications/nistpubs/800-40-Ver2/SP800-40v2.pdf>*. Additionally, please add any other vendors that you use.*

### Risk Ranking for Security Patch Management

A Risk Ranking matrix will be established regarding security patch management. Specifically, system components and other associated IT resources will be given a risk ranking pertaining to the importance of security patch management activities to be undertaken.

In accordance with NIST SP 800-30, *[company name]* will adhere to the following definitions regarding risks that are related to all system components within the cardholder environment and any other IT resources.

* **High:** The threat source is highly motivated and sufficiently capable; controls to prevent the vulnerability from being exercised are ineffective.
* **Medium:** The threat source is motivated and capable, but controls are in place that may impede successful exercise of the vulnerability.
* **Low:** The threat source lacks motivation or capability, or controls are in place to prevent, or at least significantly impede, the vulnerability from being exercised.

#### Table 6.1.E

#### Risk Ranking Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Critical Security Threats** | **Response Mechanisms to Initiate** | **Priority Level 1 (High)** | **Priority Level 2 (Medium)** | **Priority Level 3 (Low)** |
| *Vendor Patches and security updates defined as "high,” "critical" or "urgent" for all system components and other IT resources affected by threat* | *Please discuss your response mechanisms for these types of security threats.* | *X* |  |  |
| *Vendor Patches and security updates defined as "medium,” "moderate" or "important" for all system components and other IT resources affected by threat* | *Please discuss your response mechanisms for these types of security threats.* |  | *X* |  |
| *Vendor Patches and security updates defined as "low," "non-essential" or "non-urgent" for all system components and other IT resources affected by threat* | *Please discuss your response mechanisms for these types of security threats.* |  |  | *X* |
| *Security alerts from SANS, CERT, NIST, CIS and all other industry-leading associations* | *Assign risk accordingly based on each individual threat.* |  |  |  |
| *Recommendations from all other industry-leading security sources (online forums, email subscriptions to security forums, etc.) regarding threats* | *Assign risk accordingly based on each individual threat.* |  |  |  |

Additionally, the Security Patch Management Program employee will also be responsible for the following critical activities:

* Being aware of all known threats or vulnerabilities that could significantly impact system components within the cardholder data environment and any other IT resources. This requires consistent oversight and management of all online resources used for security patch management as previously described.
* Having a strong technical and business understanding of all critical systems within the organization’s IT infrastructure, as well as knowing which systems are essential for day-to-day operations
* Having response mechanisms and procedures in place to immediately report the scope of the exploitation (systems affected), the impact to the IT infrastructure as a whole and which remediation activities and plan of action initiatives are already available to the management in the event of network exploitation.

### Database of Remediation Activities that Need to be Applied

The database for remediation activities will consist of listing the relevant Uniform Resource Locators (URL) for each patch and specific advice and any other comments deemed critical to the patch itself. Additionally, the Security Patch Management Program employee will be responsible for keeping the database accurate and relevant.

#### Table 6.1.f

|  |  |  |
| --- | --- | --- |
| **System Components within Cardholder Data Environment and other IT Resources** | **Uniform Resource Locator (URL) for Patch** | **Notes/Comments** |
| *Oracle* | <http://www.oracle.com/technology/deploy/security/alerts.htm#CriticalPatchUpdates> | *Online board and listing for Oracle products and their respective patches* |
| *Microsoft* | <http://www.microsoft.com/security/updates/bulletins/default.aspx> | *Online board and listing for Microsoft products and their respective patches* |
| *?* | *?* | *?* |
| *?* | *?* | *?* |
| *?* | *?* | *?* |

### Test Procedures for Testing Patches Regarding Remediation

Security patch management testing procedures must be observed to ensure the authenticity of the patch or any other security upgrades before they are released to day-to-day operations.

The following testing procedures are to be adhered to (NIST, n.d.):

* An acceptable test environment (non-production systems) will be determined and utilized, if necessary, for each and every patch and security upgrade implemented by the SPMP employee.
* For vendors providing patches, the authenticity of the downloaded patch will need to be verified. This verification process will be determined as needed for patches and security upgrades.
* A virus scan is to be run on all patches before installation.
* Determine *patch dependency* or any other issues that may result in the installation of the patch. Would the installation of the new patch disable another? Are other patches uninstalled when the new patch is installed?

### Procedures for the Distribution, Deployment and Implementation of Patches and other Related Security-Hardening Procedures

All patches and security updates are to be pushed out in a formalized and secure manner, with all critical patches installed within one (1) month of release from a vendor or other approved third party. This includes using the following:

* Enterprise Patch Management software
* Secured email lists sent to authorized personnel
* Secure internal web source for retrieving patches sent out by the SPMP employee

*[Listed above are three common examples of deploying patches. Please modify according to your specific environment.]*

### Procedures for Verifying Successful Implementation of Patches and other Related Security-Hardening Procedures

It is the responsibility of the SPMP employee to verify the successful implementation of all patches and security upgrades to *[company name]*’s IT infrastructure. These activities will consist of, but are not limited to, the following:

* Verifying that the files have been changed as stated in the vendor’s documentation to reflect the updates as needed
* Verifying whether the recommended patches and security updates were installed properly by reviewing patch logs

*[Listed above are two common examples of verifying the successful implementation of patches and security updates. Please modify according to your specific environment.]*

## 6.1 to 6.2 Responsibility for Policy Maintenance

The *[title of responsible party]* is responsible for ensuring that the aforementioned policy is kept current as needed for purposes of compliance with the Payment Card Industry Data Security Standards (PCI DSS) initiatives.

[Return to Table of Contents](#tableofcontents)

# Requirement 6.3

# Software Development Life Cycle Processes

## 6.3 Overview

In accordance with Payment Card Industry Data Security Standards (PCI DSS) requirements, *[company name]* has established a formal policy and supporting procedures concerning *[company name]*’s software development life cycle (SDLC). This policy is to be implemented immediately. It will be evaluated on a(n) *[annual, semi-annual, quarterly]* basis for ensuring its adequacy and relevancy regarding *[company name]*’s needs and goals.

## 6.3 Policy

*[Company name]* has developed and implemented SDLC processes employing industry-accepted standards for purposes of complying with the Payment Card Industry Data Security Standards (PCI DSS) initiatives. The list of industry-leading security standards, benchmarks and frameworks to use includes but is not limited to the following:

* The United States Depart of Justice Systems Development Life Cycle <http://www.usdoj.gov/jmd/irm/lifecycle/table.htm>
* The Open Web Application Security Project (OWASP) <http://www.owasp.org>
* National Institute of Standards Technology (NIST) <http://www.nist.gov>

## 6.3 Procedure

*[Company name]* employs a number of structured software development processes and phases, which include properly authorizing, testing, approving, implementing, documenting and maintaining the specified system/application. All SDLC activities, ranging from the initial requirements stage to the routine maintenance of a system/application, are administered by *[company name]* employees, located at *[location of corporate facility]*.

The SDLC methodology encompasses a number of phases, each concluding with a major milestone. Assessments are conducted after each phase to determine if objectives have been satisfied. *[Company name]* utilizes skilled software engineers throughout all phases, which results in a thorough and uninterrupted process from beginning to end. Next follows a documented methodology process that assists in identifying the approximate time and resources needed for all SDLC activities. *[Company name]*’s SDLC activities for internally-developed systems/applications consist of the following procedures and phases:

### New System/Application and Feature Development

New system/application and feature development is the implementation of a new service or addition of new features and functions to the current product. The same processes are also involved when adding major enhancements to existing functionality.

### Request for New System/Application or Features

The process begins with the request for a new system/application, feature or tool. Authorized personnel will initiate the request. All requests are to be appropriately logged in *[ticketing system and/or some other type of tool]*.

### Feasibility Study

Once a request for a new system/application, feature or tool is received, *[company name]* analyzes it and evaluates its market opportunity and/or operational impact. Once the benefits are identified, *[company name]* conducts a feasibility study with the assistance of the development team. Based on the requirements, if the feature requested can be done in a reasonable fashion, a work estimate to implement the new system, application, feature or tool is prepared. For complete new systems and applications, *[company name]* estimates the market size and develops a business case.

### Estimate and HW/SW Requirements

Along with estimating the effort and time required to implement the new system/application, feature or tool, an estimate of hardware and software required for development and final deployment is conducted. These estimates are passed on to management for final approval.

### Management Decision

After reviewing the business rationale for the new system/application, feature or tool, *[company name]* decides whether the cost/benefits and strategic direction warrant the development to proceed. A review of the business rationale for a completely new project includes studying market opportunity and conducting a competitive analysis. *[Company name]* can opt not to proceed with the development or even to table it for a period of time. As soon as the project receives approval, the process progresses to the development and deployment phases.

### Requirement Analysis

During this phase, a detailed requirements analysis of the new system/application, feature or tool is conducted and documented in the form of a requirements specification. Documents and activities for this phase include obtaining copies of documents used during this phase and interviewing personnel for major activities during this phase

### Design

In this phase, various technical personnel collaborate to develop a detailed design of the various activities involved. The design and development team reviews the design, and the final version is documented in the form of a design specifications document. If the feature or tool is to be a part of an existing system/application or functionality, the existing design document may be modified in lieu of creating a new document. Test plans and procedures for system tests are also developed.

### Implementation

Once the design is finalized, the actual implementation of the system/application, feature or tool begins with a test in a development environment. After all errors found during the testing stage are corrected, the application code is released to a test server.

### Quality Assurance and Testing

Once all the modules are moved to a test server and integrated in the test environment, any necessary test database tables and stored procedures are also created on the test server(s). The test environment is configured as a replica of the production environment or a specific client environment; however, there may be external interfaces which, at times, may not be duplicated, and approximations may be used. Testers then assess the new modules in this test environment. Test cases and scripts are written and documented as required. Any discrepancies are resolved with the development team, and any other additional testing is conducted. Customers and/or third-party users may be involved at different levels in this phase of project cycle, based on a mutual understanding of verification requirements. Test results are documented and reviewed with development personnel and management for final approval.

### Release for Production

Once the system/application, feature or tool is successful in the test environment, *[company name]* approves the release for production. Modules are moved to the production servers where functionality is tested after all modules are updated.

*[Listed above are generally accepted industry-standard phases within a software development life cycle. Please modify the above information to reflect your organization’s specific life cycle.]*

## 6.3 Additional Software Development Requirements for PCI DSS

In accordance with Payment Card Industry Data Security Standards (PCI DSS) Requirement 6.5 (*Developing Applications Based on Secure Coding Techniques*), *[company name]* conducts the following activities in relation to the software development processes (Security Standards Council, 2009):

* Changes are tested prior to being implemented.
* Validation of all input (to prevent cross-site scripting; injection flaws and malicious file execution are conducted as needed.)
* Validation of proper error handling is conducted as needed.
* Validation of secure cryptographic storage is conducted as needed.
* Validation of secure communications is conducted as needed.
* Validation of proper Role-Based Access Control (RBAC) is conducted as needed.
* The development/test environments are separated from the production environments, with access control in place to enforce the division.
* There is a separation of duties between personnel assigned to the development/test environments and those assigned to the production environments.
* Production data (live PANs) are not used for testing and development, or are sanitized before use.
* Test data and accounts are removed before a production system becomes active.
* Custom application accounts, user IDs and/or passwords are removed before system goes into production or is released to customers.

## 6.3 Additional Supporting Documentation

**Please Note:** *If your organization would like to comment on and/or list any other supporting documentation that assists with compliance with Requirement 6.3, please do so. For example, you may have developed your own internal documents, checklists and other supporting documentation regarding software development life cycle (SDLC) and related processes. Listed below is a simple example of how best to illustrate this in a table format. You can use the table, modify or simply delete it.*

### Table 6.3

### Additional Supporting Documentation

|  |  |  |  |
| --- | --- | --- | --- |
| **Document Name** | **Description of Document** | **Date Last Updated** | |
| *sdlclifecycle.docx* | *Internally developed Software Development Lifecycle (SDLC) documentation that discusses all major phases and milestones that are undertaken for any software development activities* | *1/1/2010* |
| *Sdlcflow.vsd* | *Internally developed Software Development Lifecycle (SDLC) diagram that illustrates all major phases and milestones that are undertaken for any software development activities* | *1/1/2010* |
| *?* | *?* | *?* |
| *?* | *?* | *?* |
| *?* | *?* | *?* |

## 6.3 Responsibility for Policy Maintenance

The *[title of responsible party]* is responsible for ensuring that the aforementioned policy is kept current as needed for purposes of compliance with the Payment Card Industry Data Security Standards (PCI DSS) initiatives.

[Return to Table of Contents](#tableofcontents)

# Requirement 6.3.2

# Custom Application Code Change Reviews Policy and Procedures

## 6.3.2 Overview

In accordance with Payment Card Industry Data Security Standards (PCI DSS) requirements, *[company name]* has established a formal policy and supporting procedures concerning custom application code change reviews. This policy is to be implemented immediately. It will be evaluated on a(n) *[annual, semi-annual, quarterly]* basis for ensuring its adequacy and relevancy regarding *[company name]*’s needs and goals.

## 6.3.2 Policy

*[Company name]* will ensure that the Custom Application Code Change Reviews policy adheres to the following conditions for purposes of complying with the Payment Card Industry Data Security Standards (PCI DSS) initiatives (Security Standards Council, 2009):

* Code changes are reviewed by individuals other than the originating code author.
* Code changes are reviewed by individuals who are knowledgeable in code review techniques and secure coding practices.
* Code reviews ensure code is developed according to secure coding guidelines, such as the Open Web Security Project Guide (<http://www.owasp.org/index.php/Main_Page>), as stated in *Requirement 6.5,* titled [*Software Development Processes for Secure Coding Guidelines and Techniques*](#_Software_Development_Secure), and for any language-specific platforms for which *[company name]* utilizes to develop internal system and applications.
* Appropriate corrections are implemented prior to release of code.
* Code review results are reviewed and approved by management prior to release.

Furthermore, these activities relating to code changes may be done manually or automatically by *[company name]*.

## 6.3.2 Procedure

The procedures, which ensure that the Custom Application Code Change Reviews policy adheres to the requirements specified by the Payment Card Industry Data Security Standards (PCI DSS) compliance, require observance of the aforementioned policies, the table below to be completed and each of its columns regarding custom application code changes for internal applications to be answered accordingly.

### Table 6.3.7.a

### Custom Application Code Changes for Internally-Developed Applications

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **System or Application** | **Code Changes are Reviewed by Individuals other than the Originating Code Author** | **Code Changes are Reviewed by Experts in Code Review Techniques and Secure Coding Practices** | **Appropriate Corrections are Implemented Prior to Release** | **Code Review Results are Reviewed and Approved by Management Prior to Release** |
| *Internally built Content Management System (CMS). Used by internal employees for facilitating all customer-facing billing, lead generation and inventory management* | *Yes, conducted by code review personnel that are independent of the development team* | *Yes, conducted by code review personnel, which has the expertise and adheres to OWASP guidelines and other industry-leading guidelines for software development processes* | *Yes. This is also done by the code review personnel after communicating with all other parties involved In the software development processes.* | *Yes. This is done by the Chief Technical Officer (CTO) and the Project Managers as needed.* |
| *Customer-Facing Software as a Service (SaaS) platform that accepts credit cards and ACH payments for goods from various third-party value added resellers (VAR)* | *Yes, conducted by code review personnel that are independent of the development team* | *Yes, conducted by code review team, which has the expertise and adheres to OWASP guidelines and other industry-leading guidelines for software development processes* | *Yes. This is also done by the code review personnel after communicating with all other parties involved In the software development processes.* | *Yes. This is done by the Chief Technical Officer (CTO) and the Project Managers as needed.* |
| *?* | *?* | *?* | *?* | *?* |
| *?* | *?* | *?* | *?* | *?* |
| *?* | *?* | *?* | *?* | *?* |
| *?* | *?* | *?* | *?* | *?* |
| *?* | *?* | *?* | *?* | *?* |

Additionally, code reviews ensure code is developed according to secure coding guidelines, such as the Open Web Security Project Guide (<http://www.owasp.org/index.php/Main_Page>), as stated in *Requirement 6.5*, titled *"*[*Software Development Processes for Secure Coding Guidelines and Techniques*](#_Software_Development_Secure)*.", and as noted below:*

* Injection Flaws (SQL, OS and LDAP Injection)
* Cross-site Scripting (XSS)
* Broken Authentication and Session Management
* Insecure Direct Object References
* Cross-site Request Forgery (CSRF)
* Security Misconfiguration
* Failure to Restrict URL Access
* Unvalidated Redirects and Forwards
* Insecure Cryptographic Storage
* Insufficient Transport Layer Protection

Source: [**http://www.owasp.org/images/0/0f/OWASP\_T10\_-\_2010\_rc1.pdf**](http://www.owasp.org/images/0/0f/OWASP_T10_-_2010_rc1.pdf)

## 6.3.2 Additional Supporting Documentation

**Please Note:** *This is one of the most critical compliance requirements within PCI DSS. In short, if you develop systems and applications internally, which are involved in the cardholder data environment, you are* REQUIRED *to undertake code reviews on your software platforms on an as-needed basis (i.e., major releases) for PCI compliance. The type and depth of a code review to be undertaken should be commensurate with your environment and the languages you use to build and code your systems and applications. These reviews can be done manually, automatically or a combination of both. Therefore, you will need to list in this section the specific processes, procedures and tools used to conduct code reviews. If you are unsure where to begin, simply Google phrases such as "code review checklist," and you will find numerous helpful templates and documents, many developed for specific languages such as php, Java, .net, C++, etc.*

### Table 6.3.7.A

### Additional Supporting Documentation for Code Reviews

|  |  |  |  |
| --- | --- | --- | --- |
| **Code Review Tool** | **Description of Process** | **System | Application | Platform Performed on** | |
| *codereviewphpchecklist.docx* | *Code review checklist obtained from php forum specifically geared towards php developers. Code review consists of a 125 point review inspection. This process is undertaken before every major release for updates to the applicable systems as noted in the right column.* | *php code reviews are conducted on the internally built Content Management System (CMS), which is used by internal employees for facilitating all customer-facing billing, lead generation and inventory management* |
| *codereviewjavachecklist.docx* | *Code review checklist obtained from Java working group forum specifically geared towards Java developers. This process is undertaken before every major release for updates to the applicable systems as noted in the right column.* | *Java code reviews are conducted on the customer-facing Software as a Service (SaaS). Customer-Facing Software as a Service (SaaS) platform that accepts credit cards and ACH payments for goods from various third-party value added resellers (VAR)* |
| *Autocodereviewer Software* | *Automated open-source code review software used for general code reviews on all applicable languages used* | *All php and Java-based platforms* |
| *?* | *?* | *?* |
| *?* | *?* | *?* |

## 6.3.2 Responsibility for Policy Maintenance

The *[title of responsible party]* is responsible for ensuring that the aforementioned policy is kept current as needed for purposes of compliance with the Payment Card Industry Data Security Standards (PCI DSS) initiatives.

[Return to Table of Contents](#tableofcontents)

# Requirement 6.4

# Change Control Policy and Procedures

## 6.4 Overview

In accordance with Payment Card Industry Data Security Standards (PCI DSS) requirements, *[company name]* has established a formal policy and supporting procedures concerning change control for security patches and software modifications. This policy is to be implemented immediately. It will be evaluated on a(n) *[annual, semi-annual, quarterly]* basis for ensuring its adequacy and relevancy regarding *[company name]*’s needs and goals.

## 6.4 Policy

Change control has become a critical issue due in large part to regulatory compliance purposes and the need to fully document the change control process for accountability and tracking changes. As a result, all system components directly associated with the cardholder data environment and other IT resources that undergo changes must be documented accordingly.

*[Company name]* will ensure that Change Control policy and procedures adhere to the following conditions for purposes of complying with the Payment Card Industry Data Security Standards (PCI DSS) initiatives (PCI DSS Requirements and Security Assessment Procedures, Version 2.0):

* Establishment of change control initiation, implementation and authorization directives
* Establishment of a change control lifecycle
* Establishment of minimum reporting criteria for change control documentation
* Separation of duties, roles and responsibilities exist between the development/test environment(s) and production environment(s), complete with access controls in place.
* Production data with live Primary Account Numbers (PAN) are not to be used for testing or development.
* Test data and all associated accounts are removed before a production system becomes active.
* Documentation of impact is included in the change control documentation.
* Management signoff by appropriate parties, along with approval for all stages of the change control lifecycle, is required for each change.
* Operational functionality testing is performed and must be documented for each change, where applicable.
* For custom code changes made, all updates and releases are tested for compliance with [*Requirement 6.5*](#_Requirement_6.5_to) before being released to production.
* Back-out procedures must be documented for each change.

## 6.4 Procedure

*[Company name]* has developed and implemented a comprehensive program regarding change control, which encompasses the categories and supporting activities listed below. These policy directives will be fully enforced by *[company name]* for ensuring the change control initiatives are executed in a formal manner and on a consistent basis for all system components within the cardholder data environment and all other IT resources.

### Change Control Initiation, Implementation and Authorization Directives

Listed below are the core systems, applications and other supporting directives that form the core platform for *[company name]*’s change control platform.

* **Systems and Applications**

*[List and describe the system(s)/application(s) that undergo routine changes.]*

* **Ticketing System/Software Utility**

*[List and describe the software utility/ticketing system that is used for initiating and documenting all changes. This may be an open-source ticketing system, a vendor-purchased system or an online project management framework application that records and documents the change control process for you.]*

* **Source Code Repository**

*[List and describe the source code repository application you use.]*

* **Meetings and Communication**

*[List and describe meetings/communication that occur(s) internally for change management activities.]*

* **Supporting Information**

*[List and describe all other supporting documents (spreadsheets, memos, authorization forms, etc.) that are used to help facilitate this type of change control relating to internally-developed systems that impact the cardholder data environment.]*

### Change Control Lifecycle

#### Formally Request a Change

All requests for changes will be documented in the *[name of change control ticketing system]* by creating a new change record. The completion of a new request for change will be completed by only authorized personnel and will be documented accordingly with all vital attributes completed in the change control ticket.

#### Categorize and Prioritize the Change

The change will be appropriately categorized and prioritized in accordance with *[company name]*’s change control metrics.

#### Justification and Analysis of the Change

Authorized personnel will work with the change initiator to develop specific justification for the change itself, along with identifying how the change may impact the organization as a whole. Authorized personnel will utilize this information to further research what effect the change may have on the organization’s infrastructure, whether in a technical, operational or business capacity.

#### Approving and Scheduling the Change

Authorized personnel will utilize the *[name of change control ticketing system or utility]* to approve the change as needed and to notify all relevant parties of the approved, pending change.

#### Implementation of the Change

Authorized personnel will develop the technical requirements for the change, review the specific change steps and, finally, execute the change in an efficient manner that will minimize any impact on the infrastructure or end-users who may be affected by the change.

#### Post-Implementation Review for any Changes

Authorized personnel will conduct a post-implementation review process for ensuring that the change has achieved its stated goals for the organization. This review process may include any of the following:

* Accepting the final change
* Rejecting the change
* Modifying the change
* Discussing end-user satisfaction regarding the change
* Final notes and correspondence conducted in the *[name of change control ticketing system or utility]* regarding the change

### Minimum Reporting Criteria for Change Control Documentation

Change control requests are to be fully documented to ensure that all aspects of the change itself are thoroughly understood, from the original request to the subsequent migration with the associated release to the production environment. With that said, the following attributes constitute the minimum reporting criteria for change control documentation within the ticketing system or other associated tool that records and documents all change control activities:

* An assigned ID or change tracking number
* Representation of all critical dates relating to the requested change itself, such as when the change was originally submitted and approved, as well as when it was migrated to various stages for testing and final deployment to production
* Default fields for categorizing (i.e., normal change or emergency change, etc.) and prioritizing (i.e., critical to routine maintenance) the requested change itself
* Documented notation, communication and correspondence throughout the life of the requested change itself is to include, but is not limited to, the following:
  + Documentation of impact
  + Management signoff
  + Operational functionality
  + Back-out procedures

### Separation of Duties between Different Environments

Separation of duties is a critical component of *[company name]*’s software development platform, for which the change control process is a significant component. As such, all roles and responsibilities within the given environments (i.e., development, testing, production) at *[company name]* are to be logically and physically separated at all times with compensating controls utilized where applicable. No single person is to have complete control over or assignment to all stages, beginning from initial development to the subsequent migration of a given release to the production environment.

### Production Data and Test Data Requirements

Production data, which for the scope of PCI DSS compliance concerns live Primary Account Numbers (PAN), are never to be used for testing or development, and test data and all associated accounts are to be removed before a production system becomes active. Developing systems with security as the primary concern will help to ensure that these directives regarding production data and test data are adhered to at all times.

## 6.4 Additional Supporting Documentation

**Please Note:** *If your organization would like to comment on and/or list any other supporting documentation that assists with compliance with Requirement 6.4, please do so. For example, you may have developed your own internal documents, checklists and other supporting documentation regarding software change control and related processes. Listed below is a simple example of how to best illustrate this in a table format. You can use the table, modify or simply delete it.*

### Table 6.4.A

### Additional Supporting Documentation

|  |  |  |  |
| --- | --- | --- | --- |
| **Document Name** | **Description of Document** | **Date Last Updated** | |
| *changecontrol101.docx* | *Internally developed change control documentation that discusses all major phases and milestones that are undertaken for any software development activities* | *1/1/2010* |
| *changecontrolflow.vsd* | *Internally developed change control diagram that illustrates all major phases and milestones that are undertaken for the entire change control lifecycle* | *1/1/2010* |
| *Changecontrolticket.xlx* | *Internally developed change control ticket that is to be used when making an original request for a change. This ticket is submitted manually and filed accordingly.* | *1/1/2010* |
| *Redmine.org* | *Open-source project management framework that is used for documenting the entire change control lifecycle* | *N/A* |
| *?* | *?* | *?* |
| *?* | *?* | *?* |
| *?* | *?* | *?* |

## 6.4 Responsibility for Policy Maintenance

The *[title of responsible party]* is responsible for ensuring that the aforementioned policy is kept current as needed for purposes of compliance with the Payment Card Industry Data Security Standards (PCI DSS) initiatives.

[Return to Table of Contents](#tableofcontents)

# Requirement 6.5 to 6.59

# Software Development Secure Coding Guidelines and Training Policy and Procedures

## 6.5 to 6.5.9 Overview

In accordance with Payment Card Industry Data Security Standards (PCI DSS) requirements, *[company name]* has established a formal policy and supporting procedures concerning software development and secure coding guidelines and training. This policy is to be implemented immediately. It will be evaluated on a(n) *[annual, semi-annual, quarterly]* basis for ensuring its adequacy and relevancy regarding *[company name]*’s needs and goals.

## 6.5 to 6.5.9 Policy

*[Company name]* will ensure that the software development and secure coding guidelines and training policy and procedures adhere to the following conditions for purposes of complying with the Payment Card Industry Data Security Standards (PCI DSS) initiatives (PCI DSS Requirements and Security Assessment Procedures, Version 2.0):

* Software developers and all other relevant personnel involved in the development of software for *[company name]* are required to undergo annual training in secure coding techniques for the software platforms(s) with which they work.
* Software developers and all other relevant personnel involved in the development of software for *[company name]* are required to submit their Secure Coding Training checklist on an annual basis as evidence that they are knowledgeable in secure coding techniques.
* Software developers involved in the software development process will adhere to professional guidelines, such as the Open Web Application Security Project (OWASP) Code of Ethics and CWE/SANS.
* *[Company name]*’s software development lifecycle includes policies, processes and procedures to ensure that internally-developed applications are not vulnerable to the following threats:
* Injection Flaws (SQL, OS and LDAP Injection)
* Cross-site Scripting (XSS)
* Broken Authentication and Session Management
* Insecure Direct Object References
* Cross-site Request Forgery (CSRF)
* Security Misconfiguration
* Failure to Restrict URL Access
* Unvalidated Redirects and Forwards
* Insecure Cryptographic Storage
* Insufficient Transport Layer Protection
* All "High" vulnerabilities and threats as identified in the [*Risk Ranking Table*](#_Table_6.1.E) found in the [*Security Patch Management Installation Policy and Procedures*](#_Requirement_6.1_to)

Source: [**http://www.owasp.org/images/0/0f/OWASP\_T10\_-\_2010\_rc1.pdf**](http://www.owasp.org/images/0/0f/OWASP_T10_-_2010_rc1.pdf%20)

## 6.5 to 6.5.9 Procedure

*[Company name]* has developed and implemented a comprehensive program regarding software development and secure coding guidelines and training, which encompasses the categories and supporting activities listed below. These policy directives will be fully enforced by *[company name]* to ensure that the software development and secure coding guidelines and training initiatives are executed in a formal manner and on a consistent basis for all system components within the cardholder data environment and all other IT resources.

### Initiatives for Secure Coding Techniques

Secure coding is much more than just reviewing code via manually or with automated tools. Rather, it is a fundamental component of the entire software development lifecycle and related processes for *[company name].* As such, software developers and all other relevant personnel involved in the development of software for *[company name]* are required to undergo annual training in secure coding techniques for the software platforms(s) they work with and to sign and submit a Secure Coding Training checklist on an annual basis for compliance with [*Requirements* *6.5.a* and *6.5.b*](#_Requirement_6.5.a_and) of the Payment Card Industry Data Security Standards (PCI DSS) Provisions. *[Please note that the* [*Secure Coding Training checklist*](#_Secure_Coding_Training) *is provided within this document].*

### Developing Secure Applications to Thwart Common Threats

As part of developing software based on secure coding techniques, there is a plethora of malicious vulnerabilities and threats that pose significant dangers to *[company name]'s* internally developed software platforms upon which our customers rely. These threats are continually sought and identified on an annual basis by the Open Web Application Security Project (OWASP), and as such, developers and all other relevant personnel in the development of software are to have a comprehensive understanding and in-depth of knowledge of these vulnerabilities. Additionally, while many of the vulnerabilities can be eliminated with secure coding techniques, other critical processes and procedures must also be initiated by network engineers and other IT staff for ensuring the security of *[Company name]*’sinternally developed software platforms.

Provided within the Secure Coding Training checklist is a section that is to be acknowledged by developers and all other relevant personnel in the development of software, which confirms their understanding of these vulnerabilities. This section is to be revisited on an annual basis to remain in compliance with [*Requirement 6.5.1 to 6.5.9*](#_Requirement_6.5_to)of the Payment Card Industry Data Security Standards (PCI DSS) provisions.

## 6.5 to 6.5.9 Additional Supporting Documentation

**Please Note:** *If your organization would like to comment on and/or list any other supporting documentation that assists with compliance with Requirement 6.5 to 6.5.9, please do so. For example, you may have developed your own internal documents, checklists and other supporting documentation regarding secure coding training. Listed below is a simple example of how best to illustrate this in a table format. You can use the table, modify or simply delete it.*

### Table 6.5 to 6.5.9 (A)

### Additional Supporting Documentation

|  |  |  |  |
| --- | --- | --- | --- |
| **Document Name** | **Description of Document** | **Date Last Updated** | |
| *codetraining2010.docx* | *Internally developed general code training document that lists all relevant sites to visit and the information that must be read along with requiring a signature and submitting document on an annual basis* | *1/1/2010* |
| *codetrainingphp2010.docx* | *Internally developed code training document for php developers that lists all relevant sites to visit and the information that must be read along with requiring a signature and submitting document on an annual basis* | *1/1/2010* |
| *codetrainingjava2010.docx* | *Internally developed code training document for Java developers that lists all relevant sites to visit and the information that must be read along with requiring a signature and submitting document on an annual basis* | *1/1/2010* |

## 6.5 to 6.5.9 Responsibility for Policy Maintenance

The *[title of responsible party]* is responsible for ensuring that the aforementioned policy is kept current as needed for purposes of compliance with the Payment Card Industry Data Security Standards (PCI DSS) initiatives.

[Return to Table of Contents](#tableofcontents)

# Requirement 6.5.a and 6.5.b

# Secure Coding Training Checklist

|  |  |
| --- | --- |
| **Name and Title of Developer or Appropriate IT Personnel** | *Jason Smith, Senior Software Developer* |
| **Roles and Responsibilities for Developing Software** | *Internal Software as a Service (SaaS) customer billing platform* |
| **Platform(s) in which Developer or Appropriate IT Personnel is Proficient** | *PHP and Java* |
| **Date of Submittal of Most Recent Security Coding Training Checklist Document Confirming Compliance** | *1/1/2009* |

### Required Training Procedures for Secure Coding for OWASP-Recognized Vulnerabilities

The following vulnerabilities have been actively sought and identified on an annual basis by the Open Web Application Security Project (OWASP), and as such, developers and all other relevant personnel in the development of software are to have a comprehensive understanding and in-depth knowledge of these vulnerabilities.

|  |  |  |  |
| --- | --- | --- | --- |
| **Security Threat and Vulnerability** | **Initiatives undertaken for having a comprehensive understanding and in-depth of knowledge of these vulnerabilities** | **Has a complete understanding of each vulnerability been established, and have procedures ensuring a secure software development process been implemented?** | **Date of Completion** |
| Injection Flaws (SQL, OS and LDAP injection) | *Read all OWASP-referenced material; adherence to all OWASP guidelines for this specific threat* | *Yes* | *1/1/2010* |
| Cross-site Scripting (XSS) | *Read all OWASP referenced material and Adherence to all OWASP guidelines for this specific threat* | *Yes* | *1/1/2010* |
| Broken Authentication and Session Management | *Read all OWASP referenced material and Adherence to all OWASP guidelines for this specific threat* | *Yes* | *1/1/2010* |
| Insecure Direct Object References | *?* | *?* | *1/1/2010* |
| Cross-site Request Forgery (CSRF) | *?* | *?* | *1/1/2010* |
| Security Misconfiguraiton | *?* | *?* | *1/1/2010* |
| Failure to Restrict URL Access | *?* | *?* | *1/1/2010* |
| Unvalidated Redirects and Forwards | *?* | *?* | *1/1/2010* |
| Insecure Cryptographic Storage | *?* | *?* | *1/1/2010* |
| Insufficient Transport Layer Protection | *?* | *?* | *1/1/2010* |

**Source:** [**http://www.owasp.org/index.php/Main\_Page**](http://www.owasp.org/index.php/Main_Page)

### Adherence to OWASP-Recognized Vulnerabilities

Software developers involved in the software development process are also to adhere to the following professional guidelines as stated below and in accordance with the Open Web Application Security Project (OWASP) Code of Ethics**.**

|  |  |  |
| --- | --- | --- |
| **OWASP Code** | **Have the thorough understanding and observance of each code been achieved?** | **Date of Acknowledgment** |
| Perform all professional activities and duties in observance of all applicable laws and the highest ethical principles. | *Yes* | *1/1/2010* |
| Promote the implementation and observance of standards, procedures and controls for application security. | *Yes* | *1/1/2010* |
| Maintain appropriate confidentiality of proprietary or otherwise sensitive information encountered in the course of professional activities. | *?* | *?* |
| Execute professional responsibilities with diligence and honesty. | *?* | *?* |
| Refrain from any activities which might constitute a conflict of interest or otherwise damage the reputation of employers, the information security profession or [company name]. | *?* | *?* |
| Do not intentionally injure or denounce the professional reputation or practice of colleagues, clients or employers. | *?* | *?* |

**Source:** [**http://www.owasp.org/index.php/About\_The\_Open\_Web\_Application\_Security\_Project**](http://www.owasp.org/index.php/About_The_Open_Web_Application_Security_Project)

### Required Training Procedures for Secure Coding for CWE/SANS Top 25 Software Errors

The following twenty-five (25) software errors have been actively sought and identified by CWE/SANS, and as such, developers and all other relevant personnel in the development of software are to have a comprehensive understanding and in-depth of knowledge of these vulnerabilities.

|  |  |  |  |
| --- | --- | --- | --- |
| **Security Threat and Vulnerability** | **Initiatives undertaken for having a comprehensive understanding and in-depth of knowledge of these vulnerabilities** | **Has a complete understanding of each vulnerability been established, and have procedures ensuring a secure software development process been implemented?** | **Date of Completion** |
| *CWE-79:* [*Failure to Preserve Web Page Structure ('Cross-site Scripting')*](http://cwe.mitre.org/top25/#CWE-79) | *Read all CWE/SANS top 25 referenced material; adherence to all of the guidelines for this specific threat* | *Yes* | *1/1/2010* |
| *CWE-89:* [*Failure to Preserve SQL Query Structure (aka 'SQL Injection')*](http://cwe.mitre.org/top25/#CWE-89) | *Read all OWASP referenced material and Adherence to all OWASP guidelines for this specific threat* | *Yes* | *1/1/2010* |
| *CWE-352:* [*Cross-site Request Forgery (CSRF)*](http://cwe.mitre.org/top25/#CWE-352) | *Read all CWE/SANS Top 25 referenced material and adherence to all of the guidelines for this specific threat* | *Yes* | *1/1/2010* |
| *CWE-434:* [*Unrestricted Upload of File with Dangerous Type*](http://cwe.mitre.org/data/definitions/434.html) | *?* | *?* | *?* |
| *CWE-78:* [*Failure to Preserve OS Command Structure (aka 'OS Command Injection')*](http://cwe.mitre.org/top25/#CWE-78) | *?* | *?* | *?* |
| *CWE-209:* [*Information Exposure through an Error Message*](http://cwe.mitre.org/top25/#CWE-209) | *?* | *?* | *?* |
| *CWE-601:* [*URL Redirection to Untrusted Site ('Open Redirect')*](http://cwe.mitre.org/top25/) | *?* | *?* | *?* |
| *CWE-362:* [*Race Condition*](http://cwe.mitre.org/top25/index.html#CWE-362) | *?* | *?* | *?* |
| *CWE-120:* [*Buffer Copy without Checking Size of Input ('Classic Buffer Overflow')*](http://cwe.mitre.org/data/definitions/120.html) | *?* | *?* | *?* |
| *CWE-22:* [*Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')*](http://cwe.mitre.org/data/definitions/22.html) | *?* | *?* | *?* |
| *CWE-98:* [*Improper Control of Filename for Include/Require Statement in PHP Program ('PHP File Inclusion')*](http://cwe.mitre.org/data/definitions/98.html) | *?* | *?* | *?* |
| *CWE-805:* [*Buffer Access with Incorrect Length Value*](http://cwe.mitre.org/top25/) | *?* | *?* | *?* |
| *CWE-754:* [*Improper Check for Unusual or Exceptional Conditions*](http://cwe.mitre.org/top25/) | *?* | *?* | *?* |
| *CWE-129:* [*Improper Validation of Array Index*](http://cwe.mitre.org/data/definitions/129.html) | *?* | *?* | *?* |
| *CWE-190:* [*Integer Overflow or Wraparound*](http://cwe.mitre.org/data/definitions/190.html) | *?* | *?* | *?* |
| *CWE-131:* [*Incorrect Calculation of Buffer Size*](http://cwe.mitre.org/data/definitions/131.html) | *?* | *?* | *?* |
| *CWE-494:* [*Download of Code Without Integrity Check*](http://cwe.mitre.org/data/definitions/494.html) | *?* | *?* | *?* |
| *CWE-770:* [*Allocation of Resources Without Limits or Throttling*](http://cwe.mitre.org/top25/) | *?* | *?* | *?* |
| *CWE-285:* [*Improper Access Control (Authorization)*](http://cwe.mitre.org/top25/#CWE-285) | *?* | *?* | *?* |
| *CWE-807:* [*Reliance on Untrusted Inputs in a Security Decision*](http://cwe.mitre.org/top25/) | *?* | *?* | *?* |
| *CWE-311:* [*Missing Encryption of Sensitive Data*](http://cwe.mitre.org/top25/) | *?* | *?* | *?* |
| *CWE-798:* [*Use of Hard-coded Credentials*](http://cwe.mitre.org/top25/) | *?* | *?* | *?* |
| *CWE-306:* [*Missing Authentication for Critical Function*](http://cwe.mitre.org/top25/) | *?* | *?* | *?* |
| *CWE-732:* [*Incorrect Permission Assignment for Critical Resource*](http://cwe.mitre.org/data/definitions/732.html) | *?* | *?* | *?* |
| *CWE-327:* [*Use of a Broken or Risky Cryptographic Algorithm*](http://cwe.mitre.org/top25/#CWE-327) | *?* | *?* | *?* |

**Source:** [**http://www.sans.org/top25-software-errors/**](http://www.sans.org/top25-software-errors/)

### Secure Code Training Procedures for Specific Languages

*[Name of employee]* has also undertaken the following secure coding training initiatives in *[name of platform, such as php, .net, Java, C++, etc.]* for further establishing and ultimately maintaining the necessary skills needed for the respective coding language(s).

|  |  |  |
| --- | --- | --- |
| **Security Coding Training Initiatives Undertaken** | **Subject Matter Covered** | **Date of Activity** |
| *PHP Online Training Course* | *Discussed recent security threats for php and what measures to take during the software development lifecycle processes to ensure a safe and secure coding environment* | *1/1/2010* |
| *PHP Security Convention* | *Three-day convention in Washington, D.C. regarding current php coding issues for developers* | *?* |
| *?* | *?* | *?* |
| *?* | *?* | *?* |
| *?* | *?* | *?* |
| *?* | *?* | *?* |

**ACKNOWLEDGMENT**: Developing systems and applications related to the cardholder data environment requires an in-depth knowledge of secure coding for all applicable platforms, for which I acknowledge I have undertaken on an annual basis. I understand that failure to perform annual training in secure coding may result in loss of software development rights and access rights to system components and other essential IT system resources within*[company name].*