Datacap Systems Inc. Secure SDLC Development Processes

Are Based on Industry Standards and/or Best Practices: ***(PA-DSS 5.1.a)***

Include Security Throughout and in Accordance with PCI and PA-DSS Requirements: ***(PA-DSS 5.1.b)***

Datacap Systems processes include information security requirements throughout the software development life cycle. Datacap employs an internally generated SPLC is based on a Waterfall model that incorporates the Microsoft Secure Development Lifecycle (SDL) practices. In addition, Datacap Systems follows all PCI-DSS and PA-DSS requirements when developing payment applications. PA-DSS requirement numbers are included in this document to help track relevant controls.

Datacap Systems' SDLC includes the following MS SDL practices:

SDL Practice 1: Training Requirements

All members of a software development team will receive appropriate training to stay informed about security basics and recent trends in security and privacy. Individuals in technical roles (developers, testers, and program managers) that are directly involved with the development of software programs must attend at least one unique security training class each year.

SDL Practice 2: Security Requirements

Security and privacy concerns are addressed "up front" during early planning and design as a fundamental aspect of secure system development.

SDL Practice 3: Quality Gates/Bug Bars

Quality gates and *bug bars* are used to establish minimum acceptable levels of security and privacy quality.

SDL Practice 4: Security and Privacy Risk Assessment

Security risk assessments (SRAs) and privacy risk assessments (PRAs) are mandatory processes employed to identify functional aspects of the software that require deep review.

SDL Practice 5: Design Requirements

All aspects of product software design are considered with regard to security and privacy concerns during the design phase.

SDL Practice 6: Attack Surface Reduction

Attack surface reduction is employed as a means of reducing risk by giving attackers less opportunity to exploit a potential weak spot or vulnerability. Attack surface reduction encompasses shutting off or restricting access to system services, applying the principle of least privilege, and employing layered defenses wherever possible.

SDL Practice 7: Threat Modeling

Threat modeling is used in environments where there is meaningful security risk. Threat modeling will consider of security issues at the component or application level. Threat modeling is a team exercise, encompassing program/project managers, developers, and testers, and represents the primary security analysis task performed during the software design stage.

SDL Practice 8: Use Approved Tools

All development teams will define and publish a list of approved tools and their associated security checks, such as compiler/linker options and warnings.

SDL Practice 9: Deprecate Unsafe Functions

Project teams will analyze all functions and APIs that will be used in conjunction with a software development project and prohibit those that are determined to be unsafe. Once the banned list is determined, project teams will use header files (such as banned.h and strsafe.h), newer compilers, or code scanning tools to check code (including legacy code where appropriate) for the existence of banned functions, and replace those banned functions with safer alternatives.

SDL Practice 10: Static Analysis

Project teams will perform static analysis of source code. Static analysis of source code provides a scalable capability for security code review to ensure that secure coding policies are being followed.

SDL Practice 11: Dynamic Program Analysis

Run-time verification of software programs will be performed to ensure that a program's functionality works as designed. This verification task will specify tools that monitor application behavior for memory corruption, user privilege issues, and other critical security problems.

SDL Practice 12: Fuzz Testing

A fuzz testing strategy will be defined which is derived from the intended use of the application and the functional and design specifications for the application.

SDL Practice 13: Threat Model and Attack Surface Review

E-review threat models and attack surface measurement of a given application when it is code complete. Ensures that any design or implementation changes to the system have been accounted for, and that any new attack vectors created as a result of the changes have been reviewed and mitigated.

SDL Practice 14: Incident Response Plan

An incident response plan will be created for every software release subject to the requirements of the SDL practices.

SDL Practice 15: Final Security Review

A Final Security Review (FSR) will be conducted to examine of all the security activities performed on a software application prior to release. The FSR will be performed by the project security advisor with assistance from the regular development staff and the security and privacy team leads. The FSR should include an examination of threat models, exception requests, tool output, and performance against the previously determined quality gates or bug bars.

SDL Practice 16: Release/Archive

Software release to manufacturing (RTM) or release to Web (RTW) is conditional on completion of the SDL process. The security advisor assigned to the release must certify (using the FSR and other data) that the project team has satisfied security requirements. Similarly, for all products that have at least one component with a Privacy Impact Rating of P1, the project's privacy advisor must certify that the project team has satisfied the privacy requirements before the software can be shipped.

In addition, all pertinent information and data must be archived to allow for post-release servicing of the software. This includes all specifications, source code, binaries, private symbols, threat models, documentation, emergency response plans, license and servicing terms for any third-party software and any other data necessary to perform post-release servicing tasks.

Include Periodic Security Reviews: (PA-DSS 5.1.c)

Datacap Systems performs periodic security reviews intervals defined below throughout the development process and prior to release of the payment application and any updates. The periodic security reviews ensure that security objectives, including PCI DSS and PA-DSS requirements are being met as defined by the SDLC.

Security reviews will be conducted during multiple phases of the SDLC to coincide with PA-DSS guidelines as follows:

- Specification Phase – Functional and architectural specifications will be drafted with consideration to compliance with current PA-DSS requirements.
- Design Phase – Implementation plans for specifications will include review of the current PA-DSS Requirements and Security Assessment procedures.
- Development/Implementation Phase – During coding, the security advisor will conduct on-going informal reviews of implemented code for adherence to PA-DSS requirements.
- Test/Verification Phase – During the final release phase, all code reviews and test procedures will receive a final assessment for the required level of compliance with current PA-DSS requirements.

Development Environment

Use of Live PAN: (PA-DSS 5.1.1)

Datacap Systems does not use live PANs for testing or development. All test transactions are conducted using test cards provided by payment processors or gateways.

Test Data and Accounts: (PA-DSS 5.1.2 & 5.1.3)

Datacap Systems removes all test data and test accounts before the application is released to customers. In addition, all custom payment application accounts, user ID's and passwords are removed before payment application is released to customers.

Code Review: (PA-DSS 5.1.4)

Datacap Systems reviews all payment application code prior to release to customers after any significant change, to identify any potential coding vulnerability.

- Code changes must be reviewed by individuals other than the originating code author, and by individuals who are knowledgeable in code review techniques and secure coding practices.

  There are three parties that have responsibilities in the code review process as follows:

  - The developer – This person that wants his code to be reviewed. He/She is responsible to make sure that his code is reviewed; the issues raised are resolved before it is checked in to the source tree.
  - The code reviewer(s) – This is the individual or group of people other than the developer that work with the developer on a project who are knowledgeable in security and the area the developer is working in.
  - The designated management project security advisor – lead responsible for tracking and approving code reviews on a periodic or on-demand basis.

  Code reviewer(s) have the responsibility to perform their reviews and assign a rating that indicates the compliance with SDL and PA-DSS security coding standards. Dated review reports including date, source files reviewed, reviewer(s) names will be created and filed with the source system. The developer will be advised by the designated project security adviser of any required changes to meet security requirements.

  If the review results in a conclusion that the code reviewed needs remediation to comply with SDL and/ PA-DSS security requirements, the code reviewed will be marked as deficient and returned to the developer with the code review report. The developer must address deficiencies and resubmit the revised code for another code review.

  If the code review is concludes that security is compliant with SDL and PA-DSS requirements, then the review report is filed with the designated management project security advisor for final

approval.

Code reviewer(s) and the designated management project security advisor will have completed appropriate training for conducting security code reviews. Datacap Systems Inc. provides training in secure coding techniques for developers, based on industry best practices and guidance.

Our continuing security education activities are comprised of the following:

- Review latest Microsoft Security Development Lifecycle Core Training classes
- Attendance in Coalfire Systems S-SDLC (Secure SDLC) webinars
- Participating in Microsoft E-Learning Security Clinics and Hands-On Labs
- Encourage recommendations for technical library purchases on security subjects
- Regular review of OWASP (Open Web Application Security Project) website (http://www.owasp.org)
- Regular review of US-CERT (United States Computer Emergency Readiness Team) Current Activity (http://www.us-cert.gov/current/)
- Regular review of SecurityTracker 's Weekly Vulnerability Summary Newsletter distributed via email

Datacap develops its payment applications using Microsoft Visual Studio with C++. We subscribe to MSDN and routinely incorporate the latest updates for the development environment provided by Microsoft.

Technical personnel involved in security code reviews are directed to participate in Microsoft, OWASP, US-CERT and other training resources to maintain a high level of security awareness and develop techniques for security design, detection and mitigation in the development and testing phases of our SDL processes.

Secure Source Control: (PA-DSS 5.1.5)

Datacap Systems utilizes Microsoft Source Safe as a control repository for all payment application code. The integrity of the source code is verified to ensure that all changes to the payment application are intended and authorized. Code developers are issued credentials and use Source Safe exclusively for check-in/out of all source code. The designated management project security advisor is responsible for regular backups and integrity checks for active projects.

Application Development

Industry Best Practices Used In Development: (PA-DSS 5.1.6)

Datacap Systems develops payment applications using industry best practices (SDL processes) for secure coding techniques including the following:

- All development is performed with least privilege for the application environment.

  Datacap codes privilege in a manner that does not permit any user inputs to grant access to system resources or data that is not strictly required to perform fundamental payment processing operation. User input access is tightly restricted to only functions that are defined for transaction processing and received from Datacap created client software employing strict authentication.

- All development is performed with fail-safe defaults—i.e., all execution is by default denied unless specified within initial design.

  Datacap's coding standards define all payment requests start with fields set to null. The only input

allowed is a string of data that is strictly validated that NETePay accepts as inputs to payment requests. The application does not allow execution by any outside process.

- Is developed with access-point considerations, including input variances such as multi-channel input to the application.

  Datacap employs a client/server architecture that restricts access only to Datacap created clients that must authenticate using a secure proprietary methodology. Strict application input validation is performed before execution of only valid payment processing commands.

- All payment activity is developed in a manner that secure consideration is given for how PAN and SAD are handled in memory.

  Whenever possible Datacap incorporates support for encrypting (P2PE) card/PIN input devices. For non-encrypting devices, Datacap's coding standards allow for the existence of PAN or SAD data in memory for the shortest time to permit transmission to the payment processor.

Secure Application Development: (PA-DSS 5.1.7)

Datacap Systems provides training in secure coding techniques for developers, based on industry best practices and guidance. This training consists of:

- Review Microsoft Security Development Lifecycle Core Training classes
- Attendance in Coalfire S-SDLC (Secure SDLC) webinars
- Participating in Microsoft E-Learning Security Clinics and Hands-On Labs
- Encourage recommendations for technical library purchases on security subjects
- Regular review of OWASP (Open Web Application Security Project) website (http://www.owasp.org)
- Regular review of US-CERT (United States Computer Emergency Readiness Team) Current Activity (http://www.us-cert.gov/current/)
- Regular review of SecurityTracker 's Weekly Vulnerability Summary Newsletter distributed via email

The following areas are included in the secure coding training:

- Secure application design
- Secure coding techniques to avoid common coding vulnerabilities (Microsoft, OWASP Top 10, CERT Secure Coding)
- Managing sensitive data in memory
- Code reviews
- Security testing (for example, penetration-testing techniques)
- Risk-assessment techniques.
- Address updated training as needed to address new development technologies and methods used.

Secure Application Development: (PA-DSS 5.2)

Datacap Systems develops all payment application code to prevent common coding vulnerabilities. All payment applications (internal and external, and including web administrative access to product) are developed based on secure coding guidelines. Secure coding covers prevention of common coding vulnerabilities in software development processes, to include:

- **5.2.1** Injection flaws, particularly SQL injection (Validate input to verify user data cannot modify meaning of commands and queries, utilize parameterized queries, etc.)

- **5.2.2** Buffer Overflow (Validate buffer boundaries and truncate input strings.)

- **5.2.3** Insecure cryptographic storage (Prevent cryptographic flaws.)

- **5.2.4** Insecure communications (Properly encrypt all authenticated and sensitive communications.)

- **5.2.5** Improper error handling (Do not leak information via error messages)

- **5.2.6** All "High" vulnerabilities as identified in testing the payment application for common vulnerabilities as outlined in PA-DSS requirement 7.1.

Datacap Systems does not develop any current payment applications as web applications or services. If Datacap were to do so in the future, it would include the following items in its secure application development processes:

- **5.2.7** Cross-site scripting (XSS) (Validate all parameters before inclusion, utilize context-sensitive escaping, etc.)

- **5.2.8** Insecure direct object references (Properly authenticate users and sanitize input. Do not expose internal object references to users.)

- **5.2.9** Cross-site request forgery (CSRF) (Do not reply on authorization credentials and tokens automatically submitted by browsers.)

- **5.2.10** Broken authentication and session management

Change Control: *(PA-DSS 5.3)*

Datacap Systems follows change control procedures for all new application code and changes to existing payment application code. The procedures include the following:

- **5.3.1** Documentation of customer impact.

  Creation, review and approval of a Datacap Change Control Form(s) which detail any new functionality and impacts to user installations. Impact information to include backward compatibility (data and application), security issues and conformance to internal specifications.

- **5.3.2** Authorized sign-off by appropriate parties

  Technical personnel at the officer or manager level must approve change Control Form (s). All change forms are included in the secure project Source Safe repository in a change control folder.

- **5.3.3.a** Testing of operational functionality to validate security is not adversely impacted

  At a minimum, testing includes verification to assure that no PAN and Track data is transferred in the clear. In addition, based on the nature of the change, additional security testing (such as fuzz testing, dynamic analysis, and manual test methods) will be performed.

- **5.3.3.b** Testing of vulnerabilities per requirements 5.2.1 through 5.2.6

  If the Change Control Forms indicate that any security issues may be affected, testing for PA-DSS and OWASP vulnerabilities will be performed the treat model analysis will be reviewed and redone as necessary.

- **5.3.4** Back-out or product de-installation procedures

Back-out and/or de-installation procedures will be included on the Datacap Change Control Form(s).

Application Versioning Methodology: (PA-DSS 5.4.1)

Datacap Systems implements wild card versioning and follows a versioning methodology for the application in the format of Major, Minor, and Build:

- **Major** changes include significant changes to the application and would have an impact on PA-DSS requirements.
- **Minor** changes include small changes such as minor enhancements and may or may not have an impact on PA-DSS requirements.
- **Build** changes include bug fixes or rollups and would have no negative impact on PA-DSS requirements and are indicated by the WILDCARD (X).

Based on the above versioning methodology the application version being listed with the PCI SSC is: 5.06

Risk Assessment: (PA-DSS 5.5)

Datacap Systems has incorporated a risk assessment program into application development processes to maintain the security and quality of our applications. This process includes:

- Coverage of all functions of the payment application, including but not limited to, security-impacting features and features that cross trust boundaries.
- Assessment of application decision points, process flows, data flows, data storage, and trust boundaries.
- Identification of all areas within payment applications that interact with PAN/SAD or the cardholder data environment (CDE), as well as any process-oriented outcomes that could lead to the exposure of cardholder data.
- A list of potential threats and vulnerabilities resulting from cardholder data flow analyses, and assign risk ratings (e.g. high, medium, or low priority) to each.
- Implementation of appropriate corrections and countermeasures during the development process.
- Documentation of risk assessment results for management review and approval.

Threat modeling is employed to assess security issues at the component or application level. Threat modeling is conducted as a team exercise, encompassing program/project managers, developers, and testers, and represents the primary security analysis task performed during the software design stage.

Datacap Systems employs the **Microsoft Threat Assessment Tool** as a primary means of identifying and assessing application vulnerabilities. Additional threat assessments are performed utilizing manual review and analysis of application constructs and data flows. Datacap's SDLC requires that all potential threats identified be mitigated or assessed as not applicable.

Final Release: (PA-DSS 5.6)

Datacap Systems development processes require all releases of the payment application be formally authorized and approved by either the Vice President of Engineering (Leon Morsillo) or the Vice President of Business Development (Gale Peters).

Vulnerabilities and Application Updates

Vulnerability Identification and Remediation: (PA-DSS 7.1)

Datacap Systems has established a process to identify and assign a risk ranking to newly discovered security vulnerabilities and to test our payment applications for vulnerabilities. *Any underlying software or systems that are provided with or required by the payment are included in this process.*

- **7.1.1** Datacap Systems monitors for security vulnerability information from the following reputable sources:
  - Microsoft Advisories
  - SANS newsletters
  - US Cert Vulnerability summaries
  - http://cve.mitre.org/compatible/vulnerability_alerting.html#VulnerabilityNotificationService (Vulnerability Notification Services)

- **7.1.2** Datacap Systems assigns risk ratings to identified vulnerabilities in the following manner:
  - NVD - http://nvd.nist.gov/cvss.cfm
    - Vulnerabilities are labeled "Low" severity if they have a CVSS base score of 0.0-3.9.
    - Vulnerabilities will be labeled "Medium" severity if they have a CVSS base score of 4.0-6.9.
    - Vulnerabilities will be labeled "High" severity if they have a CVSS base score of 7.0-10.0.

- **7.1.3** Datacap Systems tests payment applications for new vulnerabilities in the following manner:
  - Datacap employs manual testing methods (code review and functional testing) and purpose-built test tools developed internally to verify susceptibility to new vulnerabilities.

Security Patches and Software Upgrade: (PA-DSS 7.2)

Datacap Systems uses the following process for timely development and deployment of security patches and upgrades.

- **7.2** Timely development and deployment of patches to customers:

  Once Datacap identifies a relevant vulnerability, we work to develop and test an updated NETePay 5 application that helps protect NETePay 5 against the specific, new vulnerability. We attempt to publish an updated application within 10 days of the identification of the vulnerability. We will then contact vendors and dealers to encourage them to install the updated application. Typically, merchants are expected to respond quickly to and install available updated applications within 30 days.

- **7.2.1** Delivery of patches and updates in a secure manner with a known chain-of-trust:

  Datacap does not deliver software and/or updates via remote access to customer networks. We deliver a complete updated NETePay 5 application which is made available via download from a Datacap supplied URL in an email notification. Downloaded software is code signed with a VeriSign certificate to assure integrity.

- **7.2.2.a** Delivery of patches and updates in a manner that maintains the integrity of the deliverable:

Downloads of NETePay applications and all associated components will be provided from a secure Datacap Systems Inc. website employing SSL supported by an SSL certificate. Downloaded software is code signed with a VeriSign certificate to assure integrity.

- **7.2.2.b** Integrity testing of the patch or update by the target system prior to installation:

  Downloads are code signed with a VeriSign certificate to assure integrity. Datacap recommends that users enable Microsoft UAC (User Access Control) to verify that downloaded applications are verifiable. Self-extracting installers are delivered which are created with InstallShield to verify the integrity of the code through a checksum automatically and will not install if corrupt.

Application Release Notes: (PA-DSS 7.3)

Datacap Systems provides release notes for all releases of our payment application including updates and new releases. These release notes are included in the Application Release Approval Form and describe the impact of the update and how the version number was changed to reflect the updated release.