Table of Contents

[Revisions 3](#_Toc264270796)

[Objectives 4](#_Toc264270797)

[Responsibilities 4](#_Toc264270798)

[Exceptions to Standard 5](#_Toc264270799)

[Revisions to Standard 5](#_Toc264270800)

[Acknowledgement of Standard 5](#_Toc264270801)

[Security Architecture Requirements 6](#_Toc264270802)

[Authentication Requirements 6](#_Toc264270803)

[Session Management Requirements 7](#_Toc264270804)

[Access Control Requirements 7](#_Toc264270805)

[Input Validation Requirements 8](#_Toc264270806)

[Output Encoding/Escaping Requirements 8](#_Toc264270807)

[Cryptography Requirements 8](#_Toc264270808)

[Error Handling and Logging Requirements 9](#_Toc264270809)

[Data Protection Requirements 10](#_Toc264270810)

[Communication Security Requirements 10](#_Toc264270811)

[HTTP Security Requirements 10](#_Toc264270812)

[Security Configuration Requirements 11](#_Toc264270813)

[Malicious Code Requirements 11](#_Toc264270814)

[Training Requirements 11](#_Toc264270815)

[Secure Code Review Requirements 11](#_Toc264270816)

[Deployment Requirements 11](#_Toc264270817)

[Separation of Duty Requirements 12](#_Toc264270818)

[Incident Reporting Requirements 12](#_Toc264270819)

# Revisions

|  |  |  |  |
| --- | --- | --- | --- |
| Date of Revision | Version | Author | Notes |
| 3/17/2010 | 0.1 | Cody Bertram – Application Security Analyst | Initial rough draft |
| 4/7/2010 | 0.2 | Cody Bertram – Application Security Analyst | Updates based on review conducted by Director of Information Technologies Dyann Bradbury. |
| 5/27/2010 | 0.3 | Cody Bertram – Application Security Analyst | Updates based on review conducted by Director of Information Security Brent Lassi. |
| 6/11/2010 | 0.5 | Cody Bertram – Application Security Analyst | Updates based on review conducted by Vice President of User Experience James Gagliardi, Group Vice President of Technical Development Shane Helget, and Associate Director of globalDirect Development Gary Mullen. |

# Objectives

* Establish responsibility and accountability for Application Security within the organization.
* Protect the organization’s business information and any client or customer information within its custody or safekeeping by safeguarding its confidentiality, integrity, and availability.
* Provide suitable framework for compliance with recognized secure development best practices.

# Responsibilities

* The **Group Vice President of Global Information Technology** is responsible for the overall Application Security posture of Digital River, Inc and all of its global subsidiaries.
* **Technical Development Leadership** is responsible for the ensuring that development staff is aware and capable of meeting the requirements stated within the Digital River Secure Application Development Standard for all development projects.
* The **Director of Information Security** must support the Group Vice President of Global Information Technology, Vice President of Technical Development, and Information Technology Compliance in ensuring that objectives relating to the Digital River Application Development Standard are met.
* **Information Technology** **Compliance** is responsible for reviewing the Digital River Secure Application Development Standard and performing periodic reviews, audits, and reports on the effectiveness of and compliance with the Digital River Secure Application Development Standard and regulatory requirements.
* **Application Security** – Individuals responsible for ensuring that Digital River applications are secure must support all responsibilities stated within the Digital River Secure Application Development Standard.
* **Development –**Individuals responsible for the development of Digital River applications must comply with the Digital River Secure Application Development Standard for all new implementations and work with Application Security to bring identified non-compliant legacy applications into compliance.
* **QA –**Individuals responsible for Quality Assurance of Digital River applications must assist the organization by evaluating that new implementations and legacy code features are compliant with the Digital River Secure Application Development Standard.
* **All Digital River associates and contractors** have responsibility for standard awareness and diligent compliance with and enforcement of Information Security Policies and Standards.

# Exceptions to Standard

All exceptions to this Digital River Secure Application Development Standard must be reviewed, documented and approved by relevant business owners, data stewards, application security personnel, compliance officers, and corporate counsel. Applicable parties will be identified by the Director of Security. Established exceptions must be re-evaluated annually.

# Revisions to Standard

The Digital River Secure Application Development Standard is updated on an as-needed basis with a complete review executed on an annual basis. The annual review is completed and drafted by members of the Information Security and Compliance teams. Proposed modifications will be reviewed by the Group Vice President of Global Information Technology, Vice President of Technical Development, and Director of Information Securityprior to implementation.

# Acknowledgement of Standard

All Digital River employees assisting in the process of development of information systems will receive the Digital River Secure Application Development Standard as part of the new employee packet given to all newly hired staff. The standard document is located on Digital River’s Policy Center. Updates to the standard will be distributed on an annual basis, at the minimum. All personnel must acknowledge acceptance of the standard upon initial release of the standard, initiation of employment, or initiation of a contractual agreement with the organization. Standard acknowledgement will also be required upon any major standard revision, or annually at a minimum.

# Security Architecture Requirements

1. All application components (either individual or groups of source files, libraries, and/or executables) must be identified and documented. The business function for these components must be defined and documented. This includes all custom code, SDKs, and third party-components.
2. A high-level architecture for the application must be defined and documented.
3. Threat modeling information must be maintained by the Digital River Security Team for each application. Application architects must participate in a threat modeling review upon inception of an application’s development lifecycle or upon major version increments or architectural modifications to the application.
4. A list of sensitive data processed by an application must be identified and documented.
5. Security control libraries must be documented, publicized, and standardized for ease of implementation.
6. Legacy code that is no longer being used must be removed from the application. This includes, but is not limited to, all unnecessary functionality, such as scripts, features, subsystems, and comments.

# Authentication Requirements

1. All pages and resources must require authentication except those specifically intended to be public.
2. All password fields must not echo the user’s password when it is entered, and password fields (or the forms that contain them) must have auto complete disabled.
3. If a maximum number of authentication attempts is exceeded, the account must be locked for a period of time long enough to deter brute force attacks as directed in the Digital River Information Security Policy.
4. All authentication controls must be enforced on the server side.
5. All authentication controls (including libraries that call external authentication services) should have a centralized implementation.
6. All authentication controls must fail securely.
7. Users must be able to safely change their credentials using a mechanism that is at least as resistant to attack as the primary authentication mechanism.
8. All authentication events must be logged as indicated by the Digital River Information Security Policy.
9. All passwords must be stored securely using approved encryption algorithm or an approved one-way hash algorithm. One-way hash is preferred. See the Digital River Encryption Roadmap for more information.
10. All authentication credentials for accessing services external to the application must be encrypted and stored in a protected location (not in source code). Encryption must be based on Digital River Information Security Policy approved encryption standards.

# Session Management Requirements

1. The application’s framework must implement a default session management control implementation centralized to the application.
2. Sessions must be invalidated when the user logs out.
3. Sessions must timeout after a specified period of inactivity.
4. Sessions must timeout after an administratively-configurable maximum time period regardless of activity (an absolute timeout).
5. All pages that require authentication must have logout links.
6. The session id must never be disclosed other than in cookie headers; particularly in URLs, error messages, or logs. This includes verifying that the application does not support URL rewriting of session cookies.
7. The session id must be changed on login.
8. The session id must be changed on re-authentication.
9. The session id must be changed or cleared or invalidated on logout or timeout.
10. Only session ids generated by the application framework must be recognized as valid by the application.
11. Authenticated session tokens must be sufficiently long and random to withstand brute force attacks.
12. Cookies which contain authenticated session tokens/ids must have their domain and path set to an appropriately restrictive value for that site.
13. After an administratively-configurable period of time, authentication credentials must expire. Re-authentication should be required before any application-specific sensitive operations are permitted.

# Access Control Requirements

1. Users must only be able to access protected functions for which they possess specific authorization.
2. Users must only be able to access URLs for which they possess specific authorization.
3. Users must only be able to access data files for which they possess specific authorization.
4. Direct object references must be protected, such that only authorized objects are accessible to each user.
5. Users must only access services or functions for which they possess specific authorization.
6. Users must only access data for which they possess specific authorization.
7. Access controls must fail securely.
8. All access controls must be enforced on the server side. Additionally, the access control rules implied by the presentation layer must be enforced on the server side.
9. There must be a centralized mechanism (including libraries that call external authorization services) for protecting access to each type of protected resource.
10. All access control decisions must be capable of being logged and all failed decisions must be logged.

# Input Validation Requirements

1. The runtime environment security controls must prevent buffer overflows.
2. A validation pattern must be defined and applied to all input.
3. All input validation failures must result in input rejection or input sanitization.
4. A character set, such as UTF-8, is specified for all sources of input.
5. All input validation must be performed on the server side.
6. A single input validation control must be used by the application for each type of data that is accepted.
7. Input validation failures should be captured.

# Output Encoding/Escaping Requirements

1. All un-trusted data that are output to HTML (including HTML elements, HTML attributes, JavaScript data values, CSS blocks, and URI attributes) must be properly escaped for the applicable context.
2. All output encoding/escaping controls must be implemented on the server side.
3. Output encoding /escaping controls must encode all characters not known to be safe for the intended interpreter.
4. All un-trusted data that is output to SQL interpreters must use parameterized interfaces, prepared statements, or be escaped properly.
5. All un-trusted data that are output to XML must use parameterized interfaces or be escaped properly.
6. All un-trusted data that are used in LDAP queries must be escaped properly.
7. All un-trusted data that are included in operating system command parameters must be escaped properly.
8. All un-trusted data that are output to any interpreters not specifically listed above must be escaped properly.
9. There should be a single security control implementation for output encoding/escaping for each intended destination type.

# Cryptography Requirements

1. All cryptographic functions used to protect secrets from the application user must be implemented server side.
2. All cryptographic modules must fail securely.
3. Access to any master secret(s) must be protected from unauthorized access (A master secret is an application credential stored as plaintext on disk that is used to protect access to security configuration information).
4. Password hashes must be salted when they are created.
5. Cryptographic module failures must be logged.
6. All random numbers, random file names, random GUIDs, and random strings must be generated using the cryptographic module’s approved random number generator when these random values are intended to be un-guessable by an attacker.
7. Cryptographic modules used by the application must have been validated against FIPS 140-2 or an equivalent standard.
8. Applications must comply with the Digital River Information Security Policy as it relates to key management and cryptography requirements.

# Error Handling and Logging Requirements

1. The application must not output error messages or stack traces containing sensitive data that could assist an attacker, including session id and personal or financial information.
2. All server side errors must be handled on the server.
3. All logging controls must be implemented on the server.
4. Error handling logic in security controls must deny access by default.
5. Security logging controls must provide the ability to log both success and failure events that are identified as security-relevant.
6. Each log event must include:
7. the identity of the user that caused the event (if there is a user associated with the event),
8. the type of event
9. time stamp from a reliable source,
10. whether the event succeeded or failed
11. the source IP address of the request associated with the event,
12. Identity or name of affected data, system component, or resource
13. All events that include un-trusted data must not execute as code in the intended log viewing software.
14. Security logs must be protected from unauthorized access and modification.
15. There must be a single logging implementation that is used by the application.
16. The application must not log application-specific sensitive data that could assist an attacker, including user’s session ids and personal, sensitive, or financial information.
17. A log analysis tool must be made available which allows an analyst to search for log events based on combinations of search criteria across all fields in the log record format supported by this system.
18. Access requests to card holder data must be logged by the application.
19. Security logs must be maintained for at least one year, with a minimum of three months immediately available for analysis.

# Data Protection Requirements

1. All forms containing sensitive information must disable client side caching, including auto complete features.
2. All sensitive data must be sent to the server in the HTTP message body (i.e., URL parameters are never used to send sensitive data) and transmitted over a secure connection.
3. All cached or temporary copies of sensitive data sent to the client must be protected from unauthorized access or purged/invalidated after the authorized user accesses the sensitive data (e.g., the proper no-cache and no-store Cache-Control headers are set).
4. All cached or temporary copies of sensitive data stored on the server must be protected from unauthorized access or purged/invalidated after the authorized user accesses the sensitive data.
5. The application must properly protect shared variables and resources from inappropriate concurrent access.
6. Applications are not allowed to store the full contents of any track from the magnetic stripe (located on the back of a card, contained in a chip, or elsewhere). This data is alternatively called full track, track, track 1, track 2, and magnetic-stripe data.
7. Applications are not allowed to store the card verification code or value (three digit or four-digit number printed on the front or back of a payment card) used to verify card-not present transactions.
8. Mask PAN when displayed (the first six and last four digits are the maximum number of digits to be displayed). Render PAN, at minimum, unreadable anywhere it is stored (including on portable digital media, backup media, in logs).
9. PINs and encrypted PIN blocks are not allowed to be stored by the application.

# Communication Security Requirements

1. All connections to external systems that involve sensitive information or functions must be authenticated.
2. All connections to external systems that involve sensitive information or functions must use an account that has been set up to have the minimum privileges necessary for the application to function properly.
3. Specific character encodings must be defined for all connections (e.g., UTF-8).

# HTTP Security Requirements

1. Redirects must not include un-validated data.
2. The application must accept only a defined set of HTTP request methods, such as GET and POST.
3. Every HTTP response must contain a content type header specifying a safe character set (e.g., UTF-8).
4. The HTTPOnly flag should be used on all cookies that do not specifically require access from JavaScript.
5. The secure flag must be used on all cookies that contain sensitive data, including the session cookie.
6. HTTP headers in both requests and responses must contain only printable ASCII characters.
7. The application must generate a strong random token as part of all links and forms associated with transactions or accessing sensitive data, and the application must verify the presence of this token with the proper value for the current user when processing these requests. This requirement describes the mechanism required to defend against Cross Site Request Forgery (CSRF) attacks.

# Security Configuration Requirements

1. All security-relevant configuration information must be stored in locations that are protected from unauthorized access.

# Malicious Code Requirements

1. It is strictly forbidden to develop or modify code to intentionally introduce malicious behavior into an application.

# Training Requirements

1. All individuals involved in the process of the development of systems code are required to participate in training designed to raise secure development awareness.
2. All individuals involved in quality assurance of developed systems are required to participate in training designed to provide awareness of security requirements.

# Secure Code Review Requirements

1. 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. Code reviews shall ensure code is developed according to secure coding requirements established within this standard.

# Deployment Requirements

1. All code changes (including patches) must be tested before being deployed into production.
2. Test data, accounts, user IDs and/or passwords must be removed before a system goes into production or is released to customers.
3. Web-application firewalls must be deployed in front of public-facing web applications.

# Separation of Duty Requirements

1. Development/test environments must be separate from the production environment, with access control in place to enforce the separation.
2. A separation of duties is required between the creator of a change and those assigned to deploy it into a production environment.

# Incident Reporting Requirements

1. Any suspicion of compromise must be immediately disclosed to the Security Team as required by the Digital River Information Security Policy.