SECURITY STANDARD FOR APPLICATION AND WEB DEVELOPMENT AND DEPLOYMENT

Architecture, Standards and Planning Branch
Office of the CIO ● Province of BC
People ● *Collaboration* ● *Innovation*

Document Version 1.3
Published: April 2015

Replaces: Version 1.2

Table of Contents

DOCUMENT CONTROL

Date	Author	Version	Change Reference
October 16, 2012	Clive Brown	V 1.0	First release
November 1, 2012	Henry Lee	V 1.1	Revised to accommodate the comments from ASRB members
November 15, 2012	Henry Lee	V 1.2	Revised to accommodate the comments from NR sector, DataBC and MoH.
April 9, 2015	Clive Brown	V 1.3	Add: Effective date info, Appendix D - Assessment Guidelines, and Document Header / Footer details

INTRODUCTION

This document contains the standard for secure development and deployment of government applications. This is a standard of the Government of British Columbia, approved by the Chief Information Officer (CIO) and forms part of the government IM/IT Standards Manual.

This standard is subject to change in response to changes in application and web vulnerabilities and the ongoing evolution of secure coding techniques.

APPLICABILITY

This standard applies to application software used for a B.C. government service. It establishes the baseline technical controls for secure government applications. See Appendix C for details of compliance schedule for this standard.

This standard identifies a minimum set of application and web security controls. Security Threat and Risk Assessments may identify additional application security requirements.

NOTES TO USERS

This standard identifies the technical security requirements of government applications. Ministries and business units responsible for government applications will apply processes, including assigning roles and responsibilities, as appropriate for their organization to demonstrate compliance with this standard.

For example, where a custom code review is required, some organizations may use internal staff to perform code reviews, others may use a contracted resource to perform code reviews, while others may choose to instruct application developers to perform code reviews and report results to government. Ministries and business units are to assess the risks and apply the appropriate level of due-diligence to their processes .

A System Security Plan records information about, and decisions regarding, the development and deployment of information systems. The System Security Plan is a requirement of the Information Security Policy, section 8.1.1.

The System Security Plan is used to record a summary of risks identified in the Security Threat and Risk Assessment, approvals to deploy information systems, roles and responsibilities for information system security management, procedures and standards used to mitigate risks, and procedures used to monitor the information system and communicate security-relevant events and incidents.

The term MUST is defined as an absolute requirement of the specification. SHOULD means that valid reasons for using alternate methods may exist in particular circumstances, but the full implications must be understood and carefully weighed before choosing a different course.

1. SECURE SOFTWARE MAINTENANCE	**Effective Date: November 15, 2012**
1.1. Security patches	**Reviewed: February 5, 2015**

Purpose

This section specifies the government standard for applying security patches. Applying the latest security patches protects software applications from known vulnerabilities.

The aim of this standard is to ensure that security patches are applied in a timely manner.

Context

Security vulnerabilities are used to gain unauthorized access to information systems. Many of these vulnerabilities are fixed by security patches which must be installed by those who manage the information systems.

Secure and trustworthy systems need the most recently released and appropriate security patches to protect against exploitation and compromise of sensitive information and critical services by malicious individuals and malicious software.

Standard

A risk-based approach to prioritize the installation of security patches SHOULD be used.

The latest security patches for system components and software SHOULD be applied.

When the application of a security patch is delayed more than 5 weeks from the release of the patch, a Security Threat and Risk Assessment SHOULD be conducted and the planned timeline and reasons documented.

All the patch management activities MUST be logged in the System Security Plan for the associated information system.

Additional Guidance

- Standard for a risk-based approach to prioritizing patches is covered in Section 1.2.
- All non-essential services should be disabled to help reduce the need to apply security patches.
- Patch management activities include (but are not limited to):
  - Ensuring patches are from authorized sources
  - Assessing the business impact of implementing (or not implementing) patches
  - Adequate testing of patches
  - Identification of appropriate timing and method of applying patches
  - Reporting on patch management activities
  - Contingency plans for failures during patch management activities

References

- OCIO – Information Security Policy 8.5.3 Restrictions on changes to software packages
- OCIO – Information Security Policy 8.6.1 Control of technical vulnerabilities
- The Standard for Information Security Threat and Risk Assessment Methodology, Process and Assessment Tool is covered in the OCIO IM/IT Architecture & Standards Manual, Section 6.11

1. SECURE SOFTWARE MAINTENANCE	**Effective Date: November 15, 2012**
1.2. Security vulnerabilities management	**Reviewed: February 5, 2015**

Purpose

This section specifies the government standard for identifying and prioritizing newly discovered security vulnerabilities.

The intention of this standard is for information system support staff to keep up-to-date with new vulnerabilities that may impact their environments so that the resulting risk can be assessed and appropriate mitigation controls applied.

Context

Security vulnerabilities can be used to gain unauthorized access to or impact the operation of government information systems. Many of these vulnerabilities are fixed by vendor-provided security patches which must be installed by information system support staff.

By identifying and prioritizing newly discovered security vulnerabilities government can apply a risk based approach to prioritize remediation efforts, including the installation of security patches.

Standard

A record MUST be maintained of security vulnerabilities that are known to impact a software application.

The application owner MUST ensure that security vulnerabilities known to impact a software application used by the information system are prioritized.

The System Security Plan SHOULD identify security vulnerabilities that impact software applications used by the information system.

Additional Guidance

- Criteria for risk ranking security vulnerabilities includes:
  - the Common Vulnerability Scoring System (CVSS)
  - a vendor-supplied patch classification designation
  - an assessment of business risk
- The standard for applying vendor-supplied security patches is covered in Section 1.1.

References

- OCIO – Information Security Policy 8.5.3 Restrictions on changes to software packages
- Common Vulnerability Scoring System (CVSS) – http://www.first.org/cvss/

1. SECURE SOFTWARE MAINTENANCE	Effective Date: November 15, 2012
1.3. Protection of the production environment	Reviewed: February 5, 2015

Purpose

The aim of the standard is to ensure that the information security expectations of government and citizens continue to be met during and after changes to hosted software applications.

Context

Without adequate protection of the production environment, security features could be inadvertently or deliberately omitted or rendered inoperable, processing irregularities could occur, or malicious code could be introduced.

Standard

Production environments MUST be segregated from development and test environments through a combination of physical or logical controls as identified in a Security Threat and Risk Assessment.

Production and non-production environments MUST be protected by separate access controls.

There SHOULD be a separation of duties between personnel assigned to develop software in development environments and those assigned to maintain software in production environments.

If test data contains sensitive data which is classified as MEDIUM or HIGH, test data MUST be removed once testing is complete.

A test account which is dormant MUST either 1) be removed from the system OR 2) be disabled such that the system cannot be logged into from the account. An account which has been, or will be, inactive for greater than 45 days MUST be considered dormant.

Accounts, user IDs, and passwords MUST not be embedded in the source code.

Records of approvals to deploy new software, software modifications and security patches MUST be included in the System Security Plan.

Additional Guidance

References

- OCIO – Information Security Policy 8.5.1 Change control procedures
- OCIO – Information Security Policy 6.1.4 Development and test information systems must be separated from operational information systems

2. SECURE SOFTWARE DEVELOPMENT	Effective Date: November 15, 2012
2.1. Code review requirements	Reviewed: February 5, 2015

Purpose

This section specifies the security requirements for code review once the code is written in order to avoid known vulnerabilities and to enforce secure coding practices.

Context

There are known vulnerabilities that can be exploited by attackers using a variety of malware. A code review identifies vulnerabilities in the code and enables more secure coding.

The use of a comprehensive software development process helps ensure that important system requirements, including security requirements, are met during software development. A variety of software development methodologies are used by software developers. This standard identifies the expected security components of a software development life cycle.

Custom code is code developed by or on behalf of the Province of British Columbia.

Standard

Custom code MUST be reviewed prior to release to production in order to identify any potential coding vulnerability.

Code reviews MUST search for all vulnerabilities referenced in Section 2.2, Secure coding, and address identified vulnerabilities.

All security controls specific to the development and deployment of the software application SHOULD be described in the System Security Plan.

Code reviews SHOULD be performed by individuals other than original code author.

Code reviews SHOULD be performed by individuals who are knowledgeable in code review techniques and secure coding practices.

Additional Guidance

- Either manual or automated source code analysis is acceptable; however the combination of both can produce the best results.
- The standard for protection of the production environment is covered in Section 1.3.
- The standard for secure coding is covered in Section 2.2.
- Development standards for information systems and services are covered in the OCIO IM/IT Architecture & Standards Manual, Section 2.1.

2. SECURE SOFTWARE DEVELOPMENT	**Effective Date: November 15, 2012**
2.2. Secure coding	**Reviewed: February 5, 2015**

Purpose

The aim of this standard is to prevent common coding vulnerabilities from impacting information systems.

Context

The application layer is at risk from both internal and external threats. Without adequate application security controls, an information system can be compromised, eroding trust in government services.

Custom code is code developed by or on behalf of the Province of British Columbia.

Standard

Custom code MUST be tested and remediated for, at a minimum, the common coding vulnerabilities listed in Appendix A.

In addition to the above, custom code used for web applications and application interfaces MUST be tested and remediated for, at a minimum, the web application and application interface vulnerabilities listed in Appendix B.

All applications SHOULD be developed based on secure coding guidelines.

Additional Guidance

- The standard for Code review requirements is covered in Section 2.1.
- Guidance on secure coding is available from: CERT, NIST, OWASP, SANS

References

- OCIO – Information Security Policy 8.2 Correct processing in applications
- CERT – The Software Engineering Institute at Carnegie Mellon University, http://www.cert.org/
- NIST – The National Institute of Standards and Technology, http://csrc.nist.gov/
- OWASP – The Open Web Application Security Project, http://www.owasp.org/
- SANS – Information Security Training and Certification, http://www.sans.org/

3. ATTACK DETECTION AND PREVENTION	Effective Date: November 15, 2012
3.1. Public-facing Web applications	Reviewed: February 5, 2015

Purpose

The aim of this standard is to prevent or reduce application-layer attacks on public facing applications.

Context

Attacks on public-facing web applications are common and often successful, and are made possible by poor coding practices. This standard is intended to reduce the number of compromises on public-facing web applications that result in breaches of sensitive information. Vulnerability assessments of the public-facing web application help reduce the possible attacks. Use of a web-application firewall provides an additional protection measure to mitigate improperly coded or configured applications.

Standard

Owners of public-facing web applications MUST ensure that the applications are protected against known attacks commensurate with the associated Security Threat and Risk Assessment by reviewing public-facing web applications using manual or automated application vulnerability assessment tools or methods at least annually and after significant changes to applications.

Alternatively, or additionally, a web-application firewall configured to provide protection from known attacks can be used.

Additional Guidance

- The standard for secure coding is covered in Section 2.2.

References

- OCIO – Information Security Policy 8.2 Correct processing in applications
- OCIO – Information Security Policy 11.2.2 Technical compliance checking

APPENDIX A: COMMON CODING VULNERABILITIES

The following coding vulnerabilities are based on the *PCI DSS Requirements and Security Assessment Procedures, Version 2.0*. The vulnerabilities and testing procedures are subject to change as secure coding techniques change.

Vulnerability	Testing procedure
Injection flaw	Validate input to verify user data cannot modify meaning of commands and queries, utilize parameterized queries, etc. (for example to prevent: SQL injection, OS Command Injection, LDAP and XPath injection flaws).
Buffer overflow	Validate buffer boundaries and truncate input strings.
Insecure cryptographic storage	Prevent cryptographic flaws. Validate that cryptographic functions are used properly when used to protect stored data.
Insecure communication	Validate that all authenticated and sensitive communications are properly encrypted.
Improper error handling	Validate that sensitive information is not leaked via error messages.

APPENDIX B: WEB APPLICATION AND APPLICATION INTERFACE VULNERABILITIES

The following coding vulnerabilities are based on the *PCI DSS Requirements and Security Assessment Procedures, Version 2.0*. The vulnerabilities and testing procedures are subject to change as secure coding techniques change.

Vulnerability	Testing procedure
Cross-site scripting (XSS)	Validate all parameters before inclusion, utilize context-sensitive escaping, etc.
Improper Access Control	Validate that users are properly authenticated, input is sanitized, and internal object references are not exposed to users.
Cross-site request forgery (CSRF)	Validate that applications do not reply on authorization credentials and tokens automatically submitted by browsers.

APPENDIX C: COMPLIANCE SCHEDULE

For existing systems:
An existing system should be brought into compliance only if it poses an unacceptable security risk or if for some other reason non-compliance raises a tangible issue. Bringing existing systems into compliance simply for the sake of compliance is not advocated.

For new systems:
The standard should be factored in as a requirement for the procurement of new systems.
Where a new system cannot reasonably be made compliant and if that does not pose an unacceptable security risk and does not collide with OCIO strategic objectives then an exception may be obtained through the Office of the Government Chief Information Officer.

APPENDIX D: ASSESSMENT GUIDELINES

In addition to complying with these standards, various assessments should be considered when deploying or significantly changing an application. This appendix provides a list of the assessments, including the tests, scans and reports, which may be necessary and should be considered when developing and deploying applications. Automated scanning tools may assist in the assessment of applications.

The results of a Security Threat and Risk Assessment for an application should be used to identify the appropriate mix of assessments and the timing of assessments on a case-by-case basis.

Network vulnerability scans to identify/report on:

- Assets inappropriately accessible from externally and internally connected network devices

Server vulnerability scans to identify/report on:

- Unauthorized software
- Inappropriately opened server ports, enabled protocols, and enabled services
- Misconfigured or inappropriately enabled high risk services, e.g. ftp and telnet
- Inappropriate stored credentials within batch jobs, scripts, or plain text files
- Inappropriate local accounts that exist with non-expiring passwords
- Inadequate encryption methods and level used
- Ability to gain unauthorized access to encryption keys
- Weak server passwords (such as might be determined via password cracking)
- Missing patches

Application tests to identify/report on:

- Insecure API calls or responses
- Insecure cross application interfaces
- Insecure coding of customized code within COTS products or code that utilizes COTS APIs
- Insecure coding and functionality of sensitive application functions or of any privileged access interfaces such as application administrator screens
- Ability to execute commands or inject code (e.g. OS commands, SQL injection, Cross-site Scripting, LDAP injection)
- Inadequate session management controls
- Ability to perform URL path traversal attacks
- Ability to cause overflow conditions (e.g. parameter overflow and buffer overflow)
- Ability to perform character encoding attacks
- Ability to compromise an application by supplying inappropriate input values (i.e. fuzz testing)
- Security flaws within Web Services (REST-based and SOAP) used by the application

Static code analysis to identify/report on:

- The existence of Logic Bombs or Backdoors

- Enabled debugging features
- Credentials inappropriately stored within code

Middleware scans to identify/report on:

- Inappropriate configuration settings
- Unauthorized directories that can be traversed or displayed, i.e. directory enumeration
- Unauthorized server side application files that are accessible for downloading or inspection by clients (e.g. Viewing php, jsp, or asp file contents)
- Unnecessary product information displayed (e.g. installed modules)
- Unnecessary accounts or features enabled
- Missing patches
- Default passwords

Database scans to identify/report on:

- Inappropriate configuration settings
- Unnecessary accounts or features enabled
- Excessive privileges granted to database objects or to database OS files
- Inappropriate local accounts with non-expiring passwords
- Credentials inappropriately stored within batch jobs or scripts
- Inadequate segregation of duties
- Existence of privileged utilities or enabled debugging features in the production environment
- Inadequacy of encryption method and level used
- Ability to gain unauthorized access to encryption keys
- Weak strength database passwords (e.g. via password cracking)
- Database replication over insecure channels
- Ability to read, modify, copy, or remove configuration data, logs and access control information
- Adequacy of controls for all entrance and exit points of an application
- Missing patches
- Default passwords

Application penetration tests to identify/report on:

- Inappropriate configuration settings
- Unnecessary accounts or features enabled
- Excessive privileges
- Ability to bypass normal application access paths
- Inadequate session management controls
- Inadequate segregation of infrastructure
- Weak application passwords (e.g. via password cracking)
- Inadequate asset segregation by purpose and environment
- Privileged access not via the administrative gateway
- Ability to escalate privileges
- Ability to infiltrate data
- Ability to store malicious content

- Ability to gain unauthorized access to data and to installed product files
- Ability to gain unauthorized access to encryption keys
- Ability to gain unauthorized access to administrative interfaces and tools
- Ability to read, modify, copy, or remove configuration data, logs and access control information
- Privileged utilities or enabled debugging features in the production environment
- Inadequate encryption method and level used
- Ability to gain unauthorized access to encryption keys
- Vulnerability to common attacks such as DDoS, and session replay
- Inappropriate access or application functionality which is not restricted based on accesses granted to user roles
- Adequacy of controls for all entrance and exit points of an application
- Default passwords

Application recovery exercises to identify/report on:

- Ability to perform full recoveries and point-in-time recoveries
- Acceptable levels of business data loss for a point-in-time recovery
- Length and severity of outage
- Alignment of support contracts with recovery objectives

Incident response exercises to identify/report on:

- Logging details and retention requirements as specified in the Province's ARCS/ORCS
- Ability to generate and receive expected notifications and alerts
- Ability to change application and infrastructure privileged access credentials in the event of a breach
- Internal and cross-government response procedures

Audit exercise to identify/report on:

- Adequacy of security assessments
- Missing evidence that is required to pass an audit
- Adequacy of the application documentation