CSUSB Web Application Security Standard

CSUSB, Information Security & Emerging Technologies Office

Last Revised: **05/20/2015**

Final

REVISION CONTROL

Document Title: CSUSB Web Application Security Standard

Author: Javier Torner

File Reference:

Date	By	Action	Pages
02/10/2015	J Torner/L Carrizales	Created Standard	All
02/16/2015	L Carrizales	Updated Standard	All
03/10/2015	L Carrizales	Updated Standard with IT Subcommittee Recommendations	All
03/17/2015	L Carrizales	Updated Standard with Web Accessibility Coordinator Recommendations	All
04/22/2015	L Carrizales	Standard approved by ISET Subcommittee	All
05/20/2015	L Carrizales	Standard approved by IT Governance	All

Review/Approval History

Date	By	Action	Pages

1.0 CSUSB Web Application Security Standard

Introduction:

In order to help safeguard university web applications from unauthorized changes, and to ensure that they are consistent with user and management expectations, owners of web applications must create and implement an approval process for initial development and on-going change requests. Where possible, this approval process should enforce separation of duties for those individuals who are involved in one or more of the following processes: developing/modifying web applications, approving changes, and/or authorizing deployment into production environments. In those organizations where separation of duties is not possible due to staffing limitations or availability, the organization must employ proper management oversight and approval.

The Software Development Life Cycle (SDLC) is defined as a period that begins with conception of a new development project and ends with retirement or removal of the developed software from all active use. A SDLC typically includes five phases: 1-Initiation, 2-Development/Acquisition, 3-Implementation, 4-Operation Maintenance, and 5-Disposal. This standard incorporates these various sections within each section listed below.

1.0 Web Application Standard
2.0 Web Application Approval Process
3.0 Web Application Version Control
4.0 Web Application Development
5.0 Security Vulnerabilities
6.0 Software Testing
7.0 Change Control

Scope:

This standard applies to all web development projects that departments implement, develop and maintain which are locally developed and configured.

Required:

All CSUSB web development projects should have:

- ✓ A legitimate purpose,
- ✓ An identified web administrator, a person who is the MPP/Administrator Level for the appropriate division/college/department or their designee,
- ✓ A technical administrator,
- ✓ Reasonable security measures in place.

Web administrators, technical administrators and web developers should be clearly identified by department heads and updated as staffing or functional roles are changed.

When a department requests a new website or changes to an existing website, the web development process should include documentation of the implementation process, change control and the appropriate process of approvals.

This process must include implementing a documented trail of approval that can be provided to auditors on demand.

The records of the approved changes must be retained for three (3) years.
Enforce separation of duties for approval and deployment by implementing one of the following options:

a) Deployment to production must be approved by an individual/group who has proper authority and who was not involved in developing or making changes to the application.

b) If a department cannot separate these responsibilities due to staffing limitations or availability, the department must ensure that proper management oversight is in place to monitor, document, and approve all changes.

2.0 Web Application Approval Process

This standard is intended to provide developers and administrators of campus web applications with an understanding of the approval process required for initial development and change requests.

Recommended:

The manner in which web application development and change requests are documented, approved, and retained is left largely up to the discretion of the individual departments. For example support.csusb.edu, or Kayako tickets, e-mail, blogs, wikis, or a workflow can be used for tracking changes and approval.

The department should define the roles and responsibilities that form a chain of approval from the client who requested the web application, the functional group who supports and/or "owns" the data, the developers of the web application, the tech support group who will be supporting/maintaining it, etc.

The approval process should document at a minimum:

- A description of the modification to be approved
- Any important details (e.g. deployment details, modification details)
- Any deviations from the normal process (e.g. deployment details, chain of approval changes)
- The name and position of the approver
- The name and position of the individual/group who made the changes
- The name and position of the individual/group who is responsible for moving the changes to production
- The name and position of the individual/group who authorized the changes to production
- The date and time of approvals

3.0 Web Application Version Control

To ensure the tracking and documentation of changes, and the integrity and retention of source code, developers of all campus web applications are required to use a version control system.

Version control systems allow tracking and documentation of changes to software, management of concurrent access when multiple people must work on the same files, comparison of the differences between versions of source code, and simplified recovery to an earlier version in case of errors.

Required:

A version control system must be used to track and retain information about changes.
Appropriate security must be implemented to prevent the users of individual accounts from accessing or modifying another account's data via the version control system or the operating system.
The version control system, along with its data, must be backed up on a regular basis.

Recommended:

At a minimum, the version control system should describe the change, record who made the change, retain the date/time of change; retrieve past versions; and compare versions. Commonly used version control systems are OnBase, Bazaar, CVS, Darcs, Git, Mercurial, Monotone, SVK, and SVN.

Content management systems should be used for implementing version control on static web pages.

4.0 Web Application Development

The purpose of this standard is to assist developers and administrators of campus web applications by providing guidelines and standards for use during the web application development process. For the purposes of these CSUSB Security Standards, a web application is defined as any application that connects to a campus network and/or the Internet and that dynamically accepts user input. This process should incorporate a documented approval process, a documented change management plan, security vulnerability testing, applicable software application testing, and a revision control system.

Departments that develop, maintain, and support web applications must incorporate procedures to ensure these applications are appropriately managed and documented throughout their life-cycle.

These procedures include:

- Formal documentation and approval of a web application throughout its life-cycle from initial proposal through deployment to a production environment
- Formal change management and approval processes that include separation of duties and/or management oversight
- Formal documentation for testing procedures, including:
- Testing for security vulnerabilities
- Formal user acceptance
- Use of a version control system

Required:

<u>Prior to Initial Development:</u>

- Each department that develops a web application must implement a method for documenting the proposal, development, change management, and approval process throughout the life of that web application. For more details on the approval process, refer to Section 2 of this document-Web Application Approval Process.

- A formal (i.e. written) request for a web application must be made by the person or group sponsoring the application.
- Approval of all stakeholders is required prior to moving on to the development phase of the web application.

During the Life-cycle of the Web Application:

- Web application software must utilize a version control system. For more details, refer to Section 3 of this document-Web Application Version Control.
- Web application developers must follow industry best practices to secure the web application (i.e. secure coding guidelines such as the Open Web Application Security Project (OWASP) guidelines.
- Web application developers must use a campus-supported authentication system such as CAS or Shibboleth for web applications that require Cal State San Bernardino user credentials for authentication (i.e. Cal State University San Bernardino username/password). (If your web application will be accessed by users external to Cal State University San Bernardino, each user must have an affiliated user account.
- If the web application requires data from the campus Data Warehouse, the application sponsor must submit an Application Data Request Form for review and approval.

Prior to Deployment to Production Environment:

Web application testing procedures must include the following:

- Software and user acceptance testing as noted in Section 6 of this document - Software Testing.
- Scanning for and repairing security vulnerabilities in accordance with Section 5 of this document-Web Application Security Vulnerabilities.
- Testing for compliance with applicable laws, policies, and industry standards, including but not limited to CSUSB Web Accessibility Standards and Guidelines, confidentiality, privacy, etc.
- Appropriate approval and sign-off must be obtained as defined by the department in accordance with Section 2 of this document-Web Application Approval Process.

Deployment to Production Environment:

- The appropriate chain of approval signoff and separation of duties must be followed in accordance with Section 2 of this document-Web Application Approval Process.

Recommended:

The manner in which a formal (i.e. written) web application development request is made (e-mail, Word document, ticket tracking system, etc.) is left up to the discretion of the department. The following are intended as helpful suggestions. The size and complexity of your web application will determine if any of the following considerations apply.

Prior to Initial Development:

- The formal web application development request may include:
- Functional specifications including the purpose of the web application and its required behavior
- Expected number of concurrent users under normal load
- Expected periods of heavy activity during the year
- Data required from campus repositories such as the Data Warehouse
- Individuals/groups who will have access to the web application (e.g. applicants, students, faculty, staff)

- Expectations for availability
- Expectations for backup and recovery
- The department/resource that will develop the web application may provide an assessment of the following:
- Feasibility, practicality of implementation
- Usefulness
- Impact on and availability of required resources (including other individuals, groups/, departments, or systems)
- Cost-effective utilization of resources
- Commitment of sponsor to thoroughly test the web application
- In-Scope requirements
- Out-of-Scope requirements
- Approval of stakeholders

During the Life-cycle of the Web Application:
Any code changes made to the web application that are outside of the original proposal should be documented, reviewed, and approved according to the department's change management process.

5.0 Security Vulnerabilities

The purpose of this standard is to provide guidelines and documentation for reviewing web applications for security vulnerabilities prior to deployment.

Web applications are susceptible to attacks that may result in exposure or modification of sensitive data, or impact on availability of services to authorized users. Application testing is conducted to identify security flaws introduced in the design, implementation or deployment of an application. Developers and application administrators must identify functions that are critical to security, and test those functions to verify correct operation.

Required:

Web applications must be reviewed and tested for security vulnerabilities. Applications that store, process or provide access to Level 1 or Level 2 information must be tested to an appropriate level of detail based on assessed risk.

Vulnerability assessment must be coordinated with and approved by authorized individuals.
All security flaws must be entered into a defect tracking system, clearly identified as a security defect, and categorized according to severity. This information must be protected appropriately, prioritized, and fixed before the application is released.

Flaws discovered in applications that are already released must be assessed to determine whether there is a low/medium/high level of exposure due to the following factors:

- The likelihood that the security flaw would be exposed
- The impact on information security, integrity and application availability
- The level of access that would be required to exploit the security flaw

- Emergency procedures for addressing security flaws must be defined and documented prior to production deployment.

Recommended:

- Web software applications should be developed per secure coding guidelines such as the Open Web Application Security Project (OWASP) guidelines.
- Peer-review code with at least one other technically trained individual.
- Validate all data received via the HTTP Request. Not validating data can result in attacks such as Cross Site Scripting, SQL Injection, HTTP Response Splitting, Log Injection, and Directory Traversal.
- Validate the data on the server-side. All data (even hidden fields and data from pull down lists) are subject to being modified by a malicious user and should be validated server-side.
- Pass session IDs and cookies via SSL (HTTPS). Hackers can intercept unprotected session IDs and cookies and use them to compromise the user's session (session hijacking), and the security of your system.
- Vulnerability scans should be performed before moving application to production or whenever there are changes to the application and must include the top 10 vulnerabilities from OWASP.
- Review the OWASP guidelines. Identify those potential vulnerabilities that may apply to your web application. Review your code and test your application to ensure that your application is not vulnerable.

Static Analysis Tools

Static code analysis is the analysis of computer software that is performed without actually executing programs built from that software.

Static Analysis Tools should:

- Support the programming languages required
- Scan and report vulnerabilities with a minimum of false positives and false negatives
- Support a centralized security policy management so all scans use established policies
- Scan for malicious code detection
- Support the use of an underlying DBMS to collect, report, export and analyze scan results
- Provide remediation for vulnerabilities found
- Provide measurement metrics for long term trending of applications
- Enable collaboration between security teams and development and QA
- Provide customization capabilities to accommodate unique coding styles
- Correlate dynamic testing to assist in the prioritization of static results

Web Application Vulnerability Scanners

Web application scanners allow testers and application developers the ability to scan web applications in a fully operational environment and check for many known security vulnerabilities. Web application scanners parse URLs from the target website to find vulnerabilities. These scanners check web applications for common security problems such as SQL injection, cross-site scripting, command injection, buffer overflow, session management, and other vulnerabilities. These tools can be used to satisfy code review requirements based on the security checks provided by the tool. Web application scanners should be used on each web application release prior to deployment to a production environment.

Campus Provided Tools:

Acunetix
Nessus

Periodic Vulnerability Assessment

The Information Security and Emerging Technologies Office will conduct periodic vulnerability assessments of all web applications. Departments are required to remediate and resolve all vulnerabilities within the timeframe specified on the CSUSB Vulnerability Management Standard.

If it is technically infeasible to correct the vulnerabilities, the department must follow the procedures detailed within section 8: Non-Compliance and Exceptions of this document.

6.0 Software Testing

This standard addresses the quality assurance review and web application source code testing for software development projects. Comprehensive testing of web applications is important in mitigating problems during production processing and is critical in protecting sensitive data and minimizing risks to our university.

Software testing exists to ensure that consistent and thorough processes are followed during the release of new software by the developer(s) to the campus community.

Required:

Web applications must be tested in a pre-production environment prior to production implementation. The programmer/developer responsible for code modifications must document the enhancements or bug fixes that are to be introduced, the general timeline for production release, and the users affected by the change(s).

Web application modifications must be reviewed and approved by the functional users (refer to Section 2 of this document-Web Application Approval Process).

Recommended:

Immediately following the completion of source code modification(s) and prior to deploying software changes to production:

- Test case scenarios should be documented and completed for both unit/module testing and integrated testing. Scenarios may include:
- What will be tested
- What are the expected results of the testing
- What are the actual results of the testing

The deployment of new applications may include pre-production peer review of the development source code. The complexity of the application may indicate whether or not peer review is required Resources required and testing considerations to be identified may include:

Hardware

- Are there any considerations for testing the software in the existing available hardware environment? E.g. does the test/pre-production environment sufficiently replicate the production environment?
- Performance/usability impacts to this system and/or other applications?
- System administration support for backups of test cases, snapshots, installation of system software patches, etc.?

Software

- What test tools will be used during testing?
- What software interfaces should be identified?
- Are there potential compatibility problems with other application software?

Staff

- Functional users responsible for testing
- Developer responsibilities: peer review, unit testing, integrated testing
- Administrative signoff on test cases

A proposed schedule should be created and published to the development team and/or appropriate functional users. Items on the schedule may include dates and team assignments for:

- Unit Testing
- Integrated Testing
- Regression Testing
- Functional User Testing
- Production Rollout
- Post-Production Testing
- Web Accessibility Testing

7.0 Recommended Practices

It is the responsibility of unit managers to follow CSUSB Web Application Security Standard and development guidelines. This standard is intended to complement the patch management, server management and change management guidelines that must be followed.

For the purpose of this standard, sensitive data is defined as information that is not intended to be public, including data classified by the California State University (CSU) as Levels 1 and 2.

Encryption

- Valid Transport Layer Security/Secure Sockets Layer (TLS/SSL) certificates must be used for all sensitive information in transit between the client, server and other servers
- Production services that use TLS/SSL certificates must obtain them from a recognized Certificate Authority (CA)
- Applications using cryptography must use industry standard algorithms and implementations

Authentication and Authorization

- Shibboleth should be used to authenticate users from CSUSB and other InCommon Federation members
- If Shibboleth cannot be used to authenticate users from CSUSB, then CSUSB Active Directory (LDAP) must be used
- Web applications that process sensitive data must verify authorization for each request

Data Validation

- Web applications must validate all data for expected values
- Web applications must use server-side validation
- Web applications that use data from another source must take steps to ensure the external data is trustworthy
- Web forms and interactive elements must use a secure token to verify the user intentionally initiated the request
- Web applications must validate all data that is passed to interpreters, including Web browsers, database systems and command shells
- Web applications must only send data and code to the browser that the user is authorized to see or use

Session Management

- Web applications must set the 'secure' flag for cookies that contain sensitive data to ensure they are only sent over secure connections
- Web applications must keep session times to the minimum duration necessary for operation
- Web applications must have server-based disconnects
- Web applications must use a secure session key/token to avoid sending 'hidden data' to the browser

8.0 Non-Compliance and Exceptions

During the annual risk assessment process or in response to an audit, departments may be required to produce documentation describing their approval processes and procedures that are associated with this standard.

Applications may be scanned or physically examined for compliance with this standard at any time. If a web application is found to be non-compliant and the problem is not resolved in the timeframe determined in consultation with the Information Security & Emerging Technologies Office, the host device may be removed from the Cal State University San Bernardino network until it does comply. If it is technically infeasible for an information asset to meet this standard, departments must submit a request for exception to their Vice President and forward it to the Information Security & Emerging Technologies Office for review and approval.

Developers may be required to produce documentation or other evidence verifying compliance with this standard.

9.0 Appendix

Definitions:

Separation Of Duties - The designated approver must not have access to deploy the code him/herself. The developer(s) who modifies a web application should not have access to deploy those changes to production.

Web Application - For the purposes of these standards, a web application and all web development projects are defined as any application that connects to a campus network and/or the Internet and that may dynamically accept user input.

Web Administrator – A person who is the MPP/Administrator Level for the appropriate division/college/department or their designee.

Related Policy:

CSU Information Security Policy 8000 – 8000: Information Security Policy

CSU Information Security Policy - 8055: Change Control

CSU Information Security Policy - 8060: Access Control

CSU Information Security Policy - 8070: Information Systems Acquisition, Development and Maintenance

CSUSB Safeguarding Confidential Information

CSUSB Information Technology Accessibility Policy