Standard: Web Application Development

Contents

Revision History 5

Executive Summary 5

Introduction and Purpose 6

Scope 6

Standard 6

Developer Training, Lifecycle, and Program Development 6

Usage of DHS "Build Security In" 6

Learn CWE/SANS Top 25 Most Dangerous Software Errors 6

Learn Open Web Application Security Project (OWASP) Top 10 Web Application Vulnerabilities 6

OWASP Top Ten Cheat Sheet 6

Examine History of Acquired Software 6

Web Application Software Security 7

Hardening SQL Servers 7

Session Management using Web Language 7

Error Checking and Input Validation 7

White Listing of User Input 7

Output Sanitization of Errors Displayed 7

Integrating 3rd Party Software Components 7

Web Application Firewall Usage 7

Separation of Production from Nonproduction 7

Remove Samples, Comments, and Debug Code from Production 8

Security Testing of Web Applications 8

Periodic Web Application Scanning Using Automated Tools 8

Manual Web Application Penetration Testing 8

Static Code Analysis 8
Discovered Flaws Create Feedback Loop for Training and Process Improvement 8
Crystal Box Testing 8
Cryptography Used in Web Applications 8
Mandatory Usage of TLS / SSL for Sensitive Applications 9
No Self-Signed Certificates 9
Current and Standard Digital Certificate Required 9
Digital Certificate Validity Period 9
3rd Party Web Application Development 9
Third-Party Software Development 9
Software Vendors Must Perform Security Tests 9
General Technical Assurance 9
Specifications for Web Applications Developed In-House 9
Security in the Systems Development Life Cycle (SDLC) 10
Purchasing Information Security Solutions 10
Use of Evaluated Products 10
No Secret Credentials in Imbedded in Web Application Systems 10
Privacy Requirements for All New Web Application Development Specifications 10
Processing of Input Data in Web Applications 10
Input Data Validation and Rejected Item Handling 10
Control of Internal Processing 10
Modification of Production Business Information 11
Software Failure to Properly Operate 11
Software Feedback to User 11
Announcing System Unavailability to Users 11

Tracing Errors and Security Problems to Developers 11
Changes to Sensitive, Critical, or Valuable Information 11
Concealing Customer Account Numbers 11
Credit Card Number Usage 11
Health Information 11
Errors and Record Manipulations 12
Temporary Files and Storage 12
Message Integrity 12
Production System Input Transaction Authorization 12
Rejected or Suspended Input Validation 12
Output Data Validation 12
Output Data Controls 12
References 12

Revision History

Standard	Effective Date	Email	Version	Contact	Phone
OIT-WADS		strevena@csustan.edu	1.0	Stan Trevena	209.667.3137

Executive Summary

The Web Application Development Standard defines the requirements and guidelines for protecting web applications as they are developed for all Stanislaus State University computer and communication system information, with the goal of safeguarding the confidentiality, integrity, and availability of information stored, processed, and transmitted by Stanislaus State. Web applications, in many cases, process sensitive information from backend relational database. This standard of due care will help ensure that web applications are developed with proactive security in mind for handling sensitive university information. This software security lifecycle must be managed for all Web-based technology in-house developed, purchased, and acquired software in order to help prevent, detect, and correct security flaws. 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.

Introduction and Purpose

This standard defines the requirements for protecting web applications as they are developed for all Stanislaus State computer and communication system information, with the goal of safeguarding the confidentiality, integrity, and availability of information stored, processed, and transmitted by Stanislaus State. Web applications are often exploited as backdoors onto secure networks, and in many cases process sensitive information from backend relational databases. This standard of due care will help ensure that web applications are developed with proactive security in mind for handling sensitive university information.

Scope

This standard applies to all Stanislaus State, Self-Funded, and Auxiliary ("campus") computer systems and facilities, with a target audience of Stanislaus State Information Technology employees and partners.

Standard

Developer Training, Lifecycle, and Program Development

Developers must design security into web applications during the development process rather than after software is released into production. Developers shall be trained and updated on secure coding practices. This software security lifecycle must be managed for all Web-based technology including in-house developed, purchased, and acquired software in order to help prevent, detect, and correct security flaws.

Usage of DHS "Build Security In"

Developers must become familiar with the DHS "Build Security In" program for software assurance: https://buildsecurityin.us-cert.gov/

Learn CWE/SANS Top 25 Most Dangerous Software Errors

Developers shall study and learn the CWE (Common Weakness & Exposures) and SANS "Top 25 Most Dangerous Software Errors" http://cwe.mitre.org/top25/

Learn Open Web Application Security Project (OWASP) Top 10 Web Application Vulnerabilities

Campus developers will study and learn the OWASP Top 10 web application weaknesses in order to help prevent flaws: https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

OWASP Top Ten Cheat Sheet

Campus developers will use OWASP Top Ten Cheat Sheet as a reference while developing campus in-house web applications: https://www.owasp.org/index.php/OWASP_Top_Ten_Cheat_Sheet

Examine History of Acquired Software

For acquired application software, examine the product security process of the vendor (history of vulnerabilities, customer notification, patching/remediation) as part of the overall enterprise risk management process.

Web Application Software Security

Campus web application developers should adhere to secure coding practices and other controls for helping to secure web applications against common vulnerabilities, threats, and attacks such as SQL Injection and Cross-Site Scripting.

Hardening SQL Servers

For web applications that connect to a backend relational DB, hardening templates should be utilized to secure the Database server(s).

Session Management using Web Language

Session management with tokens, cookies, and alternatives should use a commonly accepted and supported web language. Developers must not invent their own session management algorithm or implementation that is used in production web application code.

Error Checking and Input Validation

For in-house developed web applications, developers must ensure that explicit error checking is performed and documented for all input, including for size, data type, and acceptable ranges or formats.

White Listing of User Input

For each functional web application, developers should rely on white listing on input forms of allowing known good versus filtering out known bad (black listing) based on signatures wherever possible.

Output Sanitization of Errors Displayed

Web application developers must not display system error messages to end users.

Integrating 3rd Party Software Components

For any acquired web application software components that are acquired or otherwise downloaded and integrated into the custom web application developed by the campus developers: the version running must still be supported by the vendor, and the version must be updated to the latest.

Web Application Firewall Usage

Web Application Firewalls should be used, where necessary, to protect campus web applications against common attacks, including SQL injection and command injection. If the traffic is encrypted, the WAF device must sit behind the encryption or be capable of decrypting in order to inspect the traffic.

Separation of Production from Nonproduction

The Office of Information Technology (OIT) must ensure that separate environments for production and nonproduction web applications are maintained. Developers should not typically have unmonitored access to production environments. Web servers should be physically segmented into DMZ subnets. Relational database back-ends should not reside on subnets with servers which are accessible directly from the internet.

Remove Samples, Comments, and Debug Code from Production

For in-house developed web applications, developers must ensure that development artifacts such as sample data and scripts, unused libraries, components, debug code, or comments are not included in the deployed production web application.

Security Testing of Web Applications

Approved web application security assessments and penetration testing, using remote scanners as well as manual methods are required prior to releasing production web applications. The lessons learned from discovered weaknesses should serve as a feedback loop to developer training, to improve awareness as well as the training program content.

Periodic Web Application Scanning Using Automated Tools

For any web application processing sensitive level 1 or level 2 information for University users: in-house developed web applications should be tested using automated remote web application scanners prior to deployment, whenever updates are made to the application, and on a periodic basis. It is advised that departments perform scanning in non-production instances wherever possible to avoid data integrity issues in production.

Manual Web Application Penetration Testing

For any web application processing sensitive level 1 or level 2 information for University users: the selected web applications should undergo a web application penetration test on a yearly basis. Security testing will include manual testing of vulnerabilities related to business logic flaws, which cannot be identified by an automated scanner.

Static Code Analysis

In-house developed web applications, where applicable, must be tested prior to deployment using automated static code analysis tools, looking for input validation and output encoding routines of web application code.

Discovered Flaws Create Feedback Loop for Training and Process Improvement

Any discovered web application weaknesses during security testing should help create a feedback loop into a process to improve security training, Systems Development Life Cycle (SDLC), and developer awareness for secure coding practices. Any discovered weaknesses should be documented to be used for process and developer training improvements.

Crystal Box Testing

For any web applications that process sensitive data: Security testing must include testing of two sample credentials at each different authorization/permission level of the web application.

Cryptography Used in Web Applications

Proper industry encryption ciphers and algorithms standard must be used in web applications in order to safe-guard sensitive data.

Mandatory Usage of TLS / SSL for Sensitive Applications

Web applications developed in order to process Level 1 or Level 2 information must use the most secure version of TLS/SSL available.

No Self-Signed Certificates

Self-signed certificates must not be used in any externally facing production web application handling sensitive level 1 or level 2 information including usernames and passwords.

Current and Standard Digital Certificate Required

A current digital certificate is required for every web application handling Stanislaus State business to which customers, prospects, and others may connect. Digital Certificate Validity Period The validity period for digital certificates issued by Stanislaus State must never be longer than three years.

Digital Certificate Validity Period

The validity period for digital certificates issued by Stanislaus State must never be longer than three years.

3rd Party Web Application Development

Outsourced web application development should be approved, supervised, and monitored by the university. Third party web developers producing code for Stanislaus State, must follow the procedures and requirements as if developed in-house.

Third-Party Software Development

All third parties who develop custom web applications on behalf of Stanislaus State must be bound by a contract approved by the Information Security Officer. This contract, at a minimum, must include a clear and explicit definition of property rights, licensing arrangements, functional requirements, security measures, escrow arrangements, auditing rights, and testing processes.

Software Vendors Must Perform Security Tests

Stanislaus State does not purchase software from vendors who have not passed a series of security checks defined by the Information Security Officer regardless of procurement mechanism (state procurement, ProCard, Auxiliary procurement).

General Technical Assurance

Specifications for Web Applications Developed In-House

All software developed by in-house staff, and intended to process sensitive, valuable, or critical information, must have a written formal specification. This specification must be part of an agreement between the involved Information Owner(s) and the system developer(s). This statement must be drafted and approved prior to the time when programming efforts begin.

Security in the Systems Development Life Cycle (SDLC)

For all business web application systems, systems designers and developers must consider security from the beginning of the systems design process through conversion to a production system. Any code changes made to the web application that are outside of the original proposal ought to be documented, reviewed, and approved.

Purchasing Information Security Solutions

Stanislaus State must purchase commercially-available information security solutions rather than build the solutions in-house, unless the cost-effectiveness of an in-house solution has been clearly analyzed, documented, and approved by the Information Security Officer.

Use of Evaluated Products

If all essential functional requirements can otherwise be met, an information systems security product which has been evaluated by a government agency is preferred and must be used rather than a product which has not been evaluated.

No Secret Credentials in Imbedded in Web Application Systems

Campus developers must not embed any hardcoded authentication credentials (secret usernames and passwords) in production web applications.

Privacy Requirements for All New Web Application Development Specifications

All new campus web applications developed internally or via third parties must include data privacy specifications within the formal requirements definition.

Processing of Input Data in Web Applications

Data input to web applications should be validated and sanitized to ensure that this data is correct and appropriate. User input should be considered unsafe, and the application should sanitize user supplied input before passing the data to the next layer of business logic. Additional controls may be required for systems that process, or have an impact on, sensitive, valuable or critical information. Such controls should be determined on the basis of security requirements and risk assessment.

Input Data Validation and Rejected Item Handling

All transactions to be input to a multi-user production computer system must be subjected to reasonableness checks, edit checks, or validation checks, and transactions that fail such checks must either be rejected with a notification of the rejection sent to the submitter, corrected and resubmitted, or suspended pending further investigation.

Control of Internal Processing

Control Validation checks should be incorporated into web applications to detect any corruption of information through processing errors or deliberate acts.

Modification of Production Business Information

System privileges must be established and maintained so that all system users are prevented from modifying production data in an unrestricted manner.

Software Failure to Properly Operate

Whenever web applications developed in-house fails to produce the expected results, it must always provide either an error message or some other indication of failure, one or both of which must be presented to the user.

Software Feedback to User

Whenever web applications developed in-house receives input from a user, feedback must be provided indicating whether the request was performed.

Announcing System Unavailability to Users

If a web application is unavailable but still running, it must announce this fact to users before a login process begins.

Tracing Errors and Security Problems to Developers

All complaints about software errors, omissions, and security problems that are attributable to web application software developed in-house must be traced back to the designers, programmers, and other development staff involved.

Changes to Sensitive, Critical, or Valuable Information

Transactions affecting sensitive, critical, or valuable information must be processed only if the originating individual or system is authorized to submit such transactions.

Concealing Customer Account Numbers

The account numbers appearing on computer-generated receipts provided to customers must be partially-concealed or truncated wherever possible.

Credit Card Number Usage

Stanislaus State or auxiliary employees must not develop web applications which collect, transmit, or store full 16-digit credit card numbers, expiration dates or security codes. Credit card numbers must not be used for customer identification or any other purpose.

Health Information

Stanislaus State or auxiliary employees must not develop web applications which collect, transmit, or store medical records.

Errors and Record Manipulations

Stanislaus State production web applications must be built so that no single person can make an error or manipulate the records without such events being detected by some other person during the routine execution of that other person's duties.

Temporary Files and Storage

Temporary files, and temporary storage locations within the memory of general-purpose computers, must be overwritten when the programmed process that created them completes its work.

Message Integrity

Requirements for ensuring authenticity and protecting message integrity in applications should be identified, and appropriate controls identified and implemented.

Production System Input Transaction Authorization

Methods must be in place to ensure that all input to production web applications that has been submitted for processing has been properly authorized.

Rejected or Suspended Input Validation

Input transactions that are corrected for resubmission, or that are suspended and later approved for resubmission, must be subjected to the same validation procedures that original input transactions receive.

Output Data Validation

Data output from a web application should be validated to ensure that the processing of stored information is correct and appropriate to the circumstances.

Output Data Controls

Automated and manual controls must be established to validate the correctness and accuracy of all sensitive and critical information which has been processed by Stanislaus State production application systems.

References

1. U.S. Department of Homeland Security (DHS): Build Security In Common Weakness Enumeration (CWE): "CWE and SANS Top 25 Most Dangerous Software Errors"

2. Open Web Application Security Project (OWASP): "OWASP Top Ten Project

3. Open Web Application Security Project (OWASP): "OWASP Top Ten Cheat Sheet