Section:	Information Security	Effective:	November 2007
Standard:	Web Application Security	Revised:	
Policy Ref:	5.8.29 Web Application Security	Responsibility:	Chief Information Security Officer

Web Application

SECURITY STANDARD

Security Standards are <u>mandatory</u> security rules applicable to the defined scope with respect to the subject.

Overview

Web applications present an opening to a network using standardized protocols. These protocols can be leveraged against the web application. An array of standards will guide the institution towards a more secure web application presence.

Scope

These standards are required for Internet applications and are desired for intranet applications. This document is technical in nature and represents specific programming methods to use in developing, auditing, and maintaining web applications. UTHSCSA web application developers will implement these standards and thus are part of the scope of this document. In addition, web applications that are purchased/leased from third parties must comply with these standards if they are Internet facing. Data owners have the responsibility to ensure their developers follow these standards.

Standard

1. Server Configuration and Access Control
    a. No unused files are to be left on a production web server.
    b. Development should not be performed on the production web server. Test environments must be provided.
    c. File permissions should use the least permission principle.
    d. Minimum privilege should be given to the web server account.
    e. Do not grant write access within the web root to users.
    f. Only grant execute permission on files that require it.
    g. Static HTML pages must not have execute permissions available.
    h. Disable directory viewing on the web server itself.

i. For web applications that require authentication, ensure that all resources recheck credentials.
j. DNS aliases will be used to access applications. The server name is not to be used for user access.

2. Authentication
    a. All authentications must use transmission encryption.
    b. Do not make logon or authentication cookies persistent.
    c. Log each instance of logins at a minimum of 30 days or as required by applicable legal requirements.
    d. Perform data validation on both password and username form fields
    e. Do not hard code the database name, username, or hostname into the application itself.
    f. Where possible, use a directory service for authentication.
    g. For applications that do not use LDAP authentication, user accounts will be locked after 60 days of inactivity.
    h. Back end servers will verify the identity of the requesting web server.
3. Data Validation

    All data input must be validated on the server side. Client side validation is encouraged but not meant to be the sole control for data input validation. The only sustainable strategy for data validation is ACCEPTING KNOWN GOOD DATA. The other options require constant maintenance and thus will not be expounded upon in this standard.
4. Database Interfaces

    a. Store database names, usernames, passwords, and hostnames in a resource outside of the code base.

    b. Publicly classified data may be hosted on a single tier web application. Internal, confidential, and confidential/high risk data must be served on a multi-tier web application.

    c. Use database engines that only allow security permissions down to an individual database object.

d. Web accounts must use a dedicated account that is not a default database account. These accounts must not have administrative privileges on the database.

5. Authenticated Session Management

a. Never accept "chosen" session IDs. Always generate a new session ID for each new login request.

b. Assign a new session ID after successful login for applications that require authentication.

c. Use session ID's that are random and not sequential in nature with a minimum of 10 alphanumeric characters.

d. Use session ID's generated by the engine you are using if applicable.

e. Use encryption to protect the authenticated Session ID in transport from server to client.

f. Session data shall not identify a user or individual.

g. Servers destroy the session related information at logout.

h. Regarding timeouts for browser inactivity:

1. All applications will implement a security timeout feature. Timeouts will be set to the lowest possible level balancing the need to protect sensitive data with the usability of the application. Users will be warned (typically 2-5 minutes) before the session is set to expire and given the opportunity to extend their session.
2. Applications will limit access to sensitive and confidential data to users who need to know and access that data. Where possible, applications will challenge users to validate their credentials beyond the initial application login when accessing confidential data.

3. See the table below for minimum requirements.

Data Classification	Logon Required	Preferred Timeout	Challenge Credentials if possible
Public	No	None	No
Internal	Yes	2 hours	No
Confidential	Yes	1 hour	Yes
Confidential/High Risk	Yes	30 minutes	Yes

i. Applications will preserve user work in draft status before expiring the user's session so data entry is not lost. Alternately, the application may engage a secure screen saver that challenges the user to validate credentials upon re-accessing the application.

Exceptions

Exceptions to this standard must be documented by the developer; reviewed jointly with the Information Security Department and approved by the CISO. Web applications in production prior to this standard have until December 2008 to comply.

References:

Open Web Application Security Project, http://www.owasp.org