WEB APPLICATION SECURITY TESTING GUIDELINES

These guidelines were developed to support the Web Application Security Standard. Please refer to this standard for additional information and/or clarification of any of these guidelines.

1. THREAT MODELING GUIDELINES
Development teams should institute threat modeling procedures. Threat modeling sessions occur during development and should include a list of potential security risks considered and a brief description of how each risk will be addressed.

1.1 Threat Modeling Objectives
1.1.1 Proposed Web Application Criticality Rating - Identification of new or a review of the existing application's criticality rating and whether a proposed change will alter the application's criticality rating.
1.1.2 Threat Modeling Plan - Recommendations, appropriate to the application and its architecture, identifying when threat modeling and security testing should be done. The procedure should define a significant change based on application-specific criteria along with potential security risks.
1.1.3 Testing Recommendation - Based on an application and its architecture, identify whether security testing will require automated scanning and/or manual testing.
1.1.4 Documentation Trail for Audit - Capture the results of the items above for future review.

1.2 Threat Modeling Session Recommendations
A threat modeling session includes at least one person from the project development team ("the Developer") and at least one person who is not a member of that project's development team ("the Reviewer"). The Reviewer should have prior development experience with the programming language(s) and other technologies used in the project.

The Reviewer will ask the Developer to consider the application's exposure to various potential security risks. Exposure may be based upon the likelihood of a risk being successfully abused against the application, the harm that could be caused by successful abuse, or other factors the Developer and Reviewer deem relevant.

The Developer and Reviewer should reach agreement on the top risks to the application, ranked by exposure. For each risk identified, the Developer should describe to the Reviewer how the risk will be minimized.

Threat modeling may be accomplished as more than one event. For example, the risks could be defined early in a project, and the techniques for minimizing the risks could be defined later. However, the entire threat modeling process should be completed in advance of security testing and moving the Web application to production.

1.3 Documentation
The results of the threat modeling session should be documented. The document should include:

1.3.1 The development project
1.3.2 The date(s) of the threat modeling session(s)
1.3.3 The names of each person involved, and each person's role (Developer, Reviewer, or other)
1.3.4 Identification of highest exposure risks
1.3.5 How each of the highest exposure identified risks will be minimized by the project.
1.4 External Development
Some applications may not be developed in house, so there is no opportunity for threat modeling. As an alternative to threat modeling, such applications may be scanned instead.

2. SECURITY TESTING GUIDELINES
When creating procedures for Security Testing, consider the following:

2.1 Development Environment
Consider testing all development environments for the department's Web applications including packaged Web applications. Packaged Web applications where the package's architecture inherently protects the application from potential security risks, may have reduced security testing requirements compared to other Web applications. There is no requirement to scan every significant change unless that change includes a new module, extends beyond the native PeopleSoft/PeopleTools security framework, or is a change in authentication technology. For any change other than these, other forms of security testing, as part of the change's QA testing, may be used in place of scanning.

2.2 Critical Migration Processes
When a migration is in response to critical production issues, migration may precede security testing. Post production security testing documentation will include justification, testing results, and a mitigation plan.

2.3 Scanning
The detailed procedure for official scans is defined in the procedure document(s) for the selected scanning tool(s).

3. SCANNING GUIDELINES
Scanning is a form of security testing and may be used during development. Scanning may also be required as part of the Quality Assurance (QA) process depending on the criticality of the Web application. Web applications in production should be periodically scanned according to a schedule defined in the Web Application Security Standard. In all of these cases, consider scanning the application through multiple user roles; for example: logged in, not logged in, guest, trusted user, etc.

3.1 Separation of Duties
To avoid a conflict of interest, the person performing the official scan should not be a member of the specific application's development project team. Developers may perform an unofficial scan of their own work in the development environment. To gain access to the scanning tool, a developer can send an email to infosec@asu.edu.

3.2 Migration Allowance for QA Fixes
If the change moving through QA is only to fix previously identified security issues in production the change may be released to production if the scan shows that the fix is an improvement over the previous production scan results. For example, a fix to a production application containing one High severity issue and one Medium severity issue might be allowed to go to production even if only the High severity issue has been fixed and the Medium severity issue remains. The Medium severity issue should still be addressed within the normal timeline for Medium severity issues.

3.3 Recently Migrated Applications
When an application's code is scanned in a QA environment and is found to have no High or Medium security issues and identical code can be migrated to production, this scan can serve as the appropriate regularly scheduled official scan. A production copy of the application is not required to be scanned until the next scheduled scan.
3.4 Security Issues
A Web application security scan may identify issues at one or more of the following levels:

- 3.4.1 High
- 3.4.2 Medium
- 3.4.3 Low
- 3.4.4 Informational

If the scan finds only Low or Informational issues, these may be addressed at the discretion of the application's development and support team. High and Medium issues should be addressed.

3.5 QA Work Flow
For those centralized applications that are required to be scanned, a member of the project team should contact the Web Scanning Team to request an official scan before initiating a production migration request through a process like the Change Management request (CMT) or a Request for Migration (e.g. Jam migrations). The request should allow enough lead time for the scan to run and any identified vulnerabilities to be fixed before the intended Web application go-live date.

The Web Scanning Team will schedule, configure and run a security scan against the application in the QA environment. If High or Medium issues are found, they should be addressed before the application goes live. Choices for addressing issues are detailed in Section 3.9. When the change has no remaining High or Medium issues and all other project requirements have been met, the application may be migrated to production.

The procedure for emergency changes is the same although lead times may be compressed.

3.6 Production Work Flow
Production official scans should be scheduled according to the schedule outline in the Web Application Security Standard.

3.7 Remediation Documentation and Tracking
When a scheduled official scan finds High or Medium security issues in a production Web application, an issue tracking system case should be opened to track the progress of addressing the issues. The official scanning team and or the Information Security Team tracks the remediation process.

3.8 Remediation Timelines
The following timelines should be followed when remediating issues found via an official scan:

- 3.8.1 High severity - Four weeks from the date the problem is reported to the responsible party
- 3.8.2 Medium severity - Six weeks from the date the problem is reported to the responsible party

3.9 Acceptable Remediation Measures
The application's development and/or support team has several options for addressing an issue. These are described further in subsequent sections. The team may:

- 3.9.1 Fix the issue
- 3.9.2 Determine that the issue is a false positive
- 3.9.3 Determine that the issue is not a concern
- 3.9.4 Obtain a waiver or extension

When all High and Medium severity issues have been addressed by one of the above methods, the case should be closed (for a production Web application) or the change may go live (for a QA scan).
3.10 Fixing the Issue
The application's development and/or support team may make changes intended to correct the issue. The new version of the application should be re-scanned, preferably in a QA environment. If the scan shows that the fix is an improvement over the previous scan results, the fix may be released to production. Note that a fix to a production application containing one High severity issue and one Medium severity issue might be allowed to go to production even if only the High severity issue has been fixed and the Medium severity issue remains. The Medium severity issue should still be addressed within the normal timeline from when it was originally reported. The attempted fix and re-scan results should be briefly documented in the issue tracking system.

3.11 Verification
For an issue to be considered fixed, a minimum of two official scans should be done: a "before" scan showing the issue and an "after" scan showing that the issue is no longer found. Both of these scans should target the same Web server host.

3.12 False Positive
In some cases the scanner will flag a possible issue that, upon examination, is found to not actually be a security issue. For example, if a Web page contains a string of digits that match the pattern of a credit card number, the scanner may raise an alert on that pattern. However, the digits might not actually be a credit card number. This is called a False Positive. The Web Scanning Team may remove obvious High and Medium severity false positives from the scanning report, but some may still remain.

If an issue is determined to be a False Positive, the finding, in addition to supporting rationale, should be documented in the issue tracking system.

3.13 When the Issue is Not a Concern
In some cases the scanner will flag a possible issue that, upon examination, is found to be a correctly identified issue, but for business or other reasons the concern is not relevant in the specific situation. For example, the scanner may find a pattern matching an email address on a Web page and flag it as a risk. Upon review, it is seen that the pattern actually is an email address. The scanner is correct. This is not a false positive. However, this may not be a pertinent concern to those responsible for the Web application. In situations like this, the security posture being evaluated by the scanner does not match the security posture desired by ASU. As such, the security posture desired by ASU takes precedence over the security posture of the scanner.

If it is determined that an issue, while correctly identified, is not a concern in this situation, the finding, with supporting rationale, should be documented in the issue tracking system.

3.14 Obtaining a Waiver or Extension
The waiver or extension option exists for security issues that are a valid concern but cannot or will not be fixed within the timelines outlined above for business or other reasons. A situation that would require an extension might include the critical processing freeze.

3.14.1 Extension
An extension may be requested to defer addressing the issue until some specific date in the future. The request should explain why the recommended timeline will not be met. If the initial extension's target date might not be met, another extension may be requested. Approval of the initial extension does not guarantee approval of a subsequent extension request.

3.14.2 Waiver
A waiver may be requested if there is no intent to correct the issue now or in the foreseeable future. Possible reasons include the following:

3.14.2.1 The vulnerability has been mitigated with compensating controls that reduce or eliminate the exposure.

3.14.2.2 It is not possible to correct the issue due to technical or budget constraints.

3.14.2.3 The application's sponsor agrees to accept the risk associated with the vulnerability, including any costs that may arise
from successful exploitation.

All reasons for a waiver should be detailed in the waiver request.

3.14.3 Waiver/Extension Request Process
The waiver or extension request should be submitted by adding a note to the issue where the original problem was recorded describing the rational for the waiver or extension and then reassigning the case to the Information Security Team. The Information Security Team will present the waiver to the Chief Information Security Officer who will evaluate the request and either grant or decline the waiver or extension. When the final decision is made, the Information Security Team will update the waiver or extension status in the issue tracking system in addition to notifying the requester of the status. In the event of an extension, the extension expiration date should be documented in the issue tracking system. The issue tracking system case should be kept open until every issue has been resolved through something other than an extension request.