

UNCLASSIFIED



APPLICATION SECURITY AND DEVELOPMENT (ASD) SECURITY TECHNICAL IMPLEMENTATION GUIDE (STIG) OVERVIEW

Version 4, Release 10

25 October 2019

Developed by DISA for the DoD

UNCLASSIFIED

Trademark Information

Names, products, and services referenced within this document may be the trade names, trademarks, or service marks of their respective owners. References to commercial vendors and their products or services are provided strictly as a convenience to our users, and do not constitute or imply endorsement by DISA of any non-Federal entity, event, product, service, or enterprise.

TABLE OF CONTENTS

	Page
1. INTRODUCTION.....	1
1.1 Executive Summary	1
1.2 Authority	1
1.3 Vulnerability Severity Category Code Definitions	2
1.4 STIG Distribution.....	2
1.5 Document Revisions	2
1.6 Other Considerations.....	3
1.7 Product Approval Disclaimer.....	3
2. ASSESSMENT CONSIDERATIONS.....	4
2.1 Security Assessment Information	4
2.1.1 Documentation.....	4
2.1.2 Functionality	4
3. CONCEPTS AND TERMINOLOGY CONVENTIONS	6
3.1 Architecture	6
4. GENERAL SECURITY REQUIREMENTS	7
4.1 Application Code Scanner.....	7
4.2 Application Scanner	7
4.3 Mobile Code.....	7

LIST OF TABLES

	Page
Table 1-1: Vulnerability Severity Category Code Definitions	2

LIST OF FIGURES

	Page
Figure 3-1: Client Server	6
Figure 3-2: Tiered Applications.....	6

1. INTRODUCTION

1.1 Executive Summary

The Application Security and Development (ASD) Security Technical Implementation Guide (STIG) is published as a tool to improve the security of Department of Defense (DoD) information systems. This document is meant for use in conjunction with the Enclave, Network Infrastructure, Application Server, Database, Browser, and appropriate Operating System (OS) STIGs and relevant technology Security Requirement Guides (SRGs).

The Application Security and Development STIG is designed to be applied to all enterprise applications connected via the network. This includes client applications installed on desktop computers which establish network connections to remote systems, HTML and browser-based applications comprised of numerous web technologies and architectures including Java, JavaScript, .NET, Cloud, RESTful-based, and SOA-oriented web services. This document is a requirement for all DoD- developed, -architected, and -administered enterprise applications and systems connected to DoD networks. An Enterprise Application (EA) is defined as an application or software that is used by the organization to assist in the execution of the organizations missions or meeting organizational goals or tasks. While some EAs may be hosted on a single system with various degrees of redundancy or fault tolerance, many are typically complex in nature, scalable, mission critical, and spread across multiple systems. Management personnel may also choose to designate an application as mission critical and deserving of EA status based upon their own criteria or operational situations. The STIG is not intended to be applied to scripts, administrative or otherwise, firewalls, or other network devices with application management interfaces when a relevant product STIG or technology SRG already exists. These requirements are intended to assist Application Development Program Managers, Application Designers/Developers, Information System Security Managers (ISSMs), Information System Security Officers (ISSOs), and System Administrators (SAs) with configuring and maintaining security controls for their applications.

This guide may be used for both in-house application development and to assist in the evaluation of the security of third-party applications. The guidance provided is not specific to any one platform, programming language, or application type. Some sections of this guide may not apply to all applications. In some cases, specific guidance has been provided based upon platform, programming language, or language types. The presence of specific guidance for a technology does not exempt applications utilizing other technologies from a requirement.

When using this guide to evaluate third-party products, some sections may not be applicable. Answers to some questions will need to be gathered through discussions with vendor representatives or from public information about the company.

1.2 Authority

DoD Instruction (DoDI) 8500.01 requires that “all IT that receives, processes, stores, displays, or transmits DoD information will be [...] configured [...] consistent with applicable DoD cybersecurity policies, standards, and architectures” and tasks that Defense Information Systems

Agency (DISA) “develops and maintains control correlation identifiers (CCIs), security requirements guides (SRGs), security technical implementation guides (STIGs), and mobile code risk categories and usage guides that implement and are consistent with DoD cybersecurity policies, standards, architectures, security controls, and validation procedures, with the support of the NSA/CSS, using input from stakeholders, and using automation whenever possible.” This document is provided under the authority of DoDI 8500.01.

Although the use of the principles and guidelines in these SRGs/STIGs provide an environment that contributes to the security requirements of DoD systems, applicable NIST SP 800-53 cybersecurity controls need to be applied to all systems and architectures based on the Committee on National Security Systems (CNSS) Instruction (CNSSI) 1253.

1.3 Vulnerability Severity Category Code Definitions

Severity Category Codes (referred to as CAT) are a measure of vulnerabilities used to assess a facility or system security posture. Each security policy specified in this document is assigned a Severity Category Code of CAT I, II, or III.

Table 1-1: Vulnerability Severity Category Code Definitions

	DISA Category Code Guidelines
CAT I	Any vulnerability, the exploitation of which will directly and immediately result in loss of Confidentiality, Availability, or Integrity.
CAT II	Any vulnerability, the exploitation of which has a potential to result in loss of Confidentiality, Availability, or Integrity.
CAT III	Any vulnerability, the existence of which degrades measures to protect against loss of Confidentiality, Availability, or Integrity.

1.4 STIG Distribution

Parties within the DoD and Federal Government’s computing environments can obtain the applicable STIG from the Cyber Exchange website at <https://cyber.mil/>. This site contains the latest copies of STIGs, SRGs, and other related security information. Those without a Common Access Card (CAC) that has DoD Certificates can obtain the STIG from <https://public.cyber.mil/>.

1.5 Document Revisions

Comments or proposed revisions to this document should be sent via email to the following address: disa.stig_spt@mail.mil. DISA will coordinate all change requests with the relevant DoD organizations before inclusion in this document. Approved changes will be made in accordance with the DISA maintenance release schedule.

1.6 Other Considerations

DISA accepts no liability for the consequences of applying specific configuration settings made on the basis of the SRGs/STIGs. It must be noted that the configuration settings specified should be evaluated in a local, representative test environment before implementation in a production environment, especially within large user populations. The extensive variety of environments makes it impossible to test these configuration settings for all potential software configurations.

For some production environments, failure to test before implementation may lead to a loss of required functionality. Evaluating the risks and benefits to a system's particular circumstances and requirements is the system owner's responsibility. The evaluated risks resulting from not applying specified configuration settings must be approved by the responsible Authorizing Official. Furthermore, DISA implies no warranty that the application of all specified configurations will make a system 100 percent secure.

Security guidance is provided for the Department of Defense. While other agencies and organizations are free to use it, care must be given to ensure that all applicable security guidance is applied both at the device hardening level as well as the architectural level due to the fact that some of the settings may not be able to be configured in environments outside the DoD architecture.

1.7 Product Approval Disclaimer

The existence of a STIG does not equate to DoD approval for the procurement or use of a product.

STIGs provide configurable operational security guidance for products being used by the DoD. STIGs, along with vendor confidential documentation, also provide a basis for assessing compliance with Cybersecurity controls/control enhancements, which supports system Assessment and Authorization (A&A) under the DoD Risk Management Framework (RMF). DoD Authorizing Officials (AOs) may request available vendor confidential documentation for a product that has a STIG for product evaluation and RMF purposes from disa.stig_spt@mail.mil. This documentation is not published for general access to protect the vendor's proprietary information.

AOs have the purview to determine product use/approval IAW DoD policy and through RMF risk acceptance. Inputs into acquisition or pre-acquisition product selection include such processes as:

- National Information Assurance Partnership (NIAP) evaluation for National Security Systems (NSS) (<http://www.niap-ccevs.org/>) IAW CNSSP #11
- National Institute of Standards and Technology (NIST) Cryptographic Module Validation Program (CMVP) (<http://csrc.nist.gov/groups/STM/cmvp/>) IAW Federal/DoD mandated standards
- DoD Unified Capabilities (UC) Approved Products List (APL) (<http://www.disa.mil/network-services/ucco>) IAW DoDI 8100.04

2. ASSESSMENT CONSIDERATIONS

2.1 Security Assessment Information

2.1.1 Documentation

The term “application documentation” is meant to include product documents provided by the vendor/developer which include details on the application product as well as instructions on how to configure and manage the product. The term “system documentation” is meant to include design and architecture documents that describe how the product is implemented within the enclave. This includes technical details of servers, load balancers, gateways, firewalls, and other components that comprise the application as an overall system. Both of these forms of documentation should include depictions and verbiage that convey how the application is deployed in conjunction with these components and examples of a likely deployment scenario. They will also include details on how to configure the application, thought processes and intentions behind application functionality and best practice recommendations.

A system security plan, application configuration guide, security classification guide, as well as coding standards, application vulnerability scan reports, and automated code review results are all part of the suite of system documentation that is expected to be available for review when conducting a security assessment of an application. While the STIG may reference application documentation, system documentation or both, it is possible that either sets of documents or other documentation will need to be reviewed in order to obtain configuration details.

The purpose of the system security plan is to provide an overview of the security requirements of the system and describe the controls in place or planned for meeting those requirements. The system security plan also delineates responsibilities and expected behavior of all individuals who access the system. Information on developing a system security plan is defined in NIST Special Publication 800-18 and is available at the NIST web site at <http://csrc.nist.gov/publications/nistpubs/800-18-Rev1/sp800-18-Rev1-final.pdf>.

General guidance on classification guides and marking data can be found in the following DoD policy documents:

- DoD Manual 5200.01 “Volume 1: DoD Information Security program Overview, Classification and Declassification” – 24 Feb 2012
- DoD Manual 5200.45 “Instructions for Developing Security Classification Guides” – 2 April 2013

2.1.2 Functionality

When reviewing an application, aspects of application functionality must be evaluated to ensure the appropriate controls exist to protect the application and the application data. Items to consider include the type of data processed by the application such as classified, unclassified, and publicly releasable or Personally Identifiable Information (PII) data. The application’s network connections, network access controls, data entry/egress points, application

authentication mechanisms, application access controls, and application auditing mechanisms. These items will vary based upon application architecture, design, and data protection requirements.

3. CONCEPTS AND TERMINOLOGY CONVENTIONS

3.1 Architecture

Networked applications are considered to be any application that communicates over a network. The application can vary in complexity and may include a local client with an Open Database Connectivity (ODBC) connection to a backend database, a single web server with a backend database, or a fully redundant cluster of web and application servers. Application architecture examples are provided here. These are not approved or proposed architectures or an all-inclusive list of architectures where the STIG applies. These are merely high level application architecture examples meant to convey what is meant by the term “networked application”.

Figure 3-1: Client Server

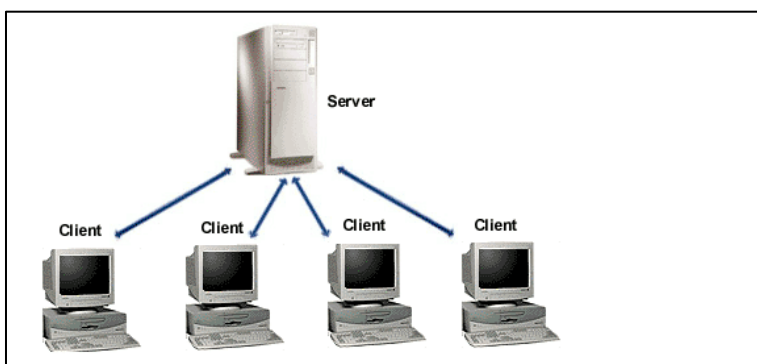
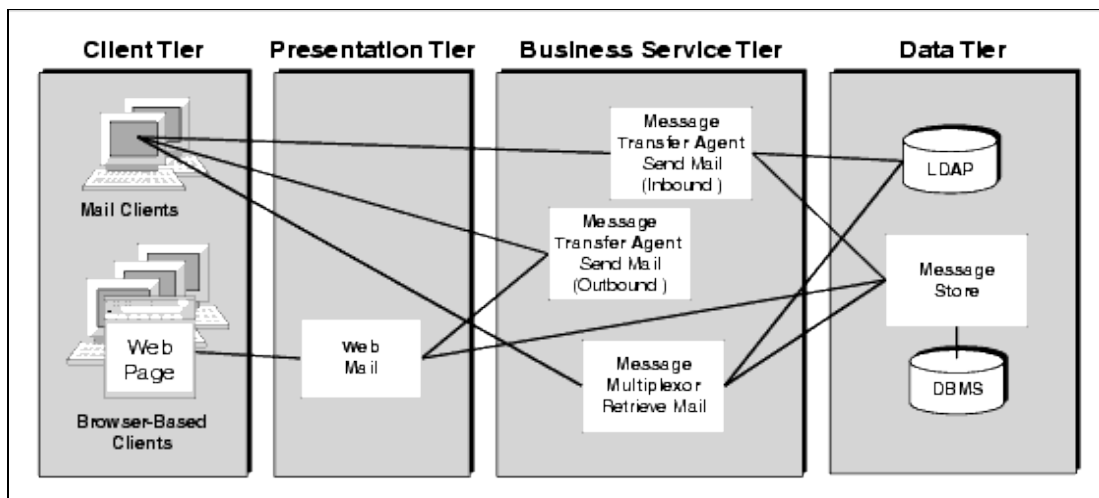


Figure 3-2: Tiered Applications



4. GENERAL SECURITY REQUIREMENTS

4.1 Application Code Scanner

An application code scanner is an automated tool that analyzes application source code for security flaws, malicious code, and back doors. Some Integrated Development Environments (IDE) includes rudimentary code scanner capabilities. These tools can often help developers identify potential flaws in the program logic allowing them to correct the issue prior to application release. Source code is not always required in order to perform code security tests. Some application code scanners will operate on binary or compiled byte code allowing system administrators to perform code scanning tests on application code without having access to the actual source code itself. While code remediation might not be possible, mitigations (e.g., IDS or network access restrictions) can be put into place to assist in minimizing the identified security risk. Application code scanners should be utilized whenever possible. Particularly in the development environment where code that has been identified as requiring remediation can be addressed prior to release.

4.2 Application Scanner

An application scanner, sometimes referred to as an active vulnerability testing tool, is a tool that is able to communicate with the application and test the application for known security vulnerabilities. An application scanner can be used to test development or production application systems for security vulnerabilities resulting from either application code errors or application system misconfigurations. These vulnerabilities include SQL Injection, Code Injection, Cross Site Scripting (XSS), file disclosures, permissions, and other security vulnerabilities that can be found in network accessible applications. Application vulnerability scanners can identify security weaknesses that are related to the underlying system configuration enabling administrators to reconfigure systems in order to eliminate identified vulnerabilities. Application vulnerability scans must be utilized and should be conducted on a regular basis, such as after any product updates or major reconfigurations and prior to activating new applications in their production environment.

4.3 Mobile Code

Mobile code is a broad term encompassing code obtained from a remote system that is downloaded across a network and executed on a local machine without the user's explicit initiation or knowledge. The primary risk associated with mobile code is that code developed by unknown sources to carry out unknown functions can automatically download and execute on users' workstations and servers, typically without the users' knowledge. Mobile code can be embedded or invoked from a Web page, email body or attachment, or Word document (e.g., www.somewebsite.com/maliciousmacroinside.doc), it downloads the document and automatically invokes Microsoft Word to open it. If the Word document contains a VBA macro, Word may automatically execute it.

The advent of HTML5 with its expanded features and functionality has resulted in the reduction in the use of some mobile code technologies. Web based applications can now leverage HTML5 functionality to perform many of the tasks previously relegated to mobile code. Mobile code will

continue to be used in some respects, however a contraction of that use is expected as web based technologies continue to evolve and capabilities improve.

Mobile code has different capabilities and varying risk factors are involved with utilizing mobile code. The following is the table of mobile code categories and the associated risk factor assignments:

1. CATEGORIES

- a. Category 1 mobile code technologies exhibit a broad functionality, allowing unmediated access to workstation, server and remote system services and resources. Category 1 mobile code technologies can pose a significant threat to DoD information systems. There are two subgroups of Category 1 mobile code technologies.
- b. Category 1A - technologies can differentiate between signed and unsigned mobile code and can be configured to allow the execution of signed mobile code while simultaneously blocking the execution of unsigned mobile code. The following mobile code technologies are assigned to Category 1A:
 - (1) ActiveX controls
 - (2) Mobile code scripts that execute in Windows Scripting Host (WSH) (e.g., JavaScript, VBScript downloaded via URL file reference or email attachments)
 - (3) The Microsoft Internet Explorer browser ActiveX runtime implementation when configured to implement usage restrictions.
- c. Category 1X consists of those mobile code technologies and implementations that are prohibited from being used in DoD information systems because they cannot implement the required Category 1 usage restrictions. They cannot differentiate between signed and unsigned mobile code or cannot be configured to block the execution of unsigned mobile code while enabling the execution of signed mobile code. The following mobile code technologies are assigned to Category 1X:
 - (1) HTML Applications (e.g., hta files) that download as mobile code
 - (2) Scrap objects (e.g., .shs and .shb files)
 - (3) Microsoft Disk Operating System (MS-DOS) batch scripts
 - (4) Unix shell scripts
 - (5) Binary executables (e.g., .exe files) that download as mobile code
 - (6) Shockwave movies (e.g., .dcr, .dxr, .dir files), including Xtras, that execute in the Shockwave for Director Plugin
 - (7) The Mozilla ActiveX Plugin a.k.a. the Netscape ActiveX Plugin runtime implementation
 - (8) The Netscape 8.0 and Netscape 8.1 browser ActiveX runtime implementations, unless the Netscape internal Internet Explorer rendering engine (filename NPTrident.dll) is uninstalled/deleted from the browsers.
- d. Category 2 mobile code technologies have full functionality, allowing mediated or controlled access to workstation, server, and remote system services and resources. Category 2 mobile code technologies can pose a moderate threat to DoD information systems. The following mobile code technologies are assigned to Category 2:

- (1) Java applets and other Java mobile code
- (2) Visual Basic for Applications (VBA) (e.g., Microsoft Office macros, also used by Corel Office)
- (3) LotusScript (e.g., Lotus Notes scripts)
- (4) PerfectScript (e.g., Corel Office macros)
- (5) Postscript
- (6) Mobile code executing in .NET Common Language Runtime2
- (7) Portable Document Format (PDF)
- (8) Flash animations (e.g., .swf and .spl files) that execute in the Shockwave Flash Plugin
- (9) Rich Internet Applications (e.g. Adobe Air, Microsoft Silverlight, and Java FX)
- e. Category 3 mobile code technologies support limited functionality, with no capability for unmediated access to workstation, server, and remote system services and resources. Category 3 mobile code technologies pose limited risk to DoD information systems. The following mobile code technologies are assigned to Category 3 and may be used when executing in the browser:
 - (1) JavaScript, including Jscript and ECMAScript variants, when executing in the browser
 - (2) VBScript, when executing in the browser
- f. Emerging mobile code technologies refer to all mobile code technologies, systems, platforms, or languages whose capabilities and threat level have not yet undergone a risk assessment and been assigned to one of the three risk categories described above. Because of the uncertain risk, the use of emerging mobile code technologies in DoD information systems is prohibited. The following are Emerging Technologies:
 - (1) All mobile agent systems and platforms

2. EXCLUSIONS

- a. Software Exclusions. The following software does not meet the DoD definition of mobile code:
 - (1) Software items that are preinstalled on the client workstation or host
 - (2) Software downloads, patches, and updates that are explicitly, intentionally, and manually initiated by the user
- b. Technology Exclusions. The following technologies are not presently designated as mobile code:
 - (1) XML, when used as a data description format for non-executable content. However, future technologies that rely on XML formatting may be considered mobile code. For example, if code is embedded within XML data, it is considered mobile code
 - (2) SMIL
 - (3) Quicktime

- (4) VRML (exclusive of any associated Java applets or JavaScript scripts. Applets or scripts associated with VRML worlds are considered mobile code)
- c. Application Exclusions. The following technology application areas are not considered Mobile Code:
 - (1) Scripts and applets that execute in the context of the Web server. Examples of technologies in this application area include Java servlets, Java Server Pages, CGI, Active Server Pages (ASP) and ASP.NET Pages, Cold Fusion Markup Language CFML, PHP, SSI, server-side JavaScript, and server-side LotusScript
 - (2) Local programs, plugins, libraries and command scripts that were preinstalled on the workstation. Examples of technologies in this application area may include binary executables, ActiveX controls, UNIX shell scripts, MS-DOS batch scripts, scripts that execute within Windows Scripting Host (WSH), and Perl scripts
 - (3) Distributed object-oriented programming systems that do not convey executable objects. Examples of technologies in this area include: SOAP, CORBA, and DCOM. Java RMI and Java Jini technologies are excluded when no executable objects are downloaded into the client; otherwise, they are Category 2. Similarly, if code is embedded within a SOAP message, it is considered mobile code
 - (4) Software patches, updates, including self-extracting updates - software updates that must be explicitly and intentionally invoked by the user are outside the scope of the mobile code designation. Examples of technologies in this area include: Netscape SmartUpdate, Microsoft Windows Update, Symantec AntiVirus LiveUpdate, Netscape Web browser plug-ins, and Linux RPM packages
- d. Single Enclave Exclusion. Mobile code that originates from and travels exclusively within a single enclave boundary is exempt from the mobile code requirements, including scripts that are run upon logon/startup that are distributed within a network (e.g., a system boots up and in doing so pulls a script from a network server that the system runs) within the same enclave. However, if an enclave consists of geographically dispersed computing environments that are connected by the NIPRNet, SIPRNet, Internet, or a public network, the aforementioned exclusion is no longer valid