ADOBE COLDFUSION
SECURITY TECHNICAL IMPLEMENTATION GUIDE
(STIG) OVERVIEW

Version 1, Release 2

01 July 2026

Developed by Adobe and DISA for the DOD

Trademark Information

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

1. INTRODUCTION

1.1 Executive Summary

The Adobe ColdFusion 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 other STIGs, including appropriate operating system STIGs.

This STIG assessment focused on the configuration and security posture of the Adobe ColdFusion application platform deployed within a Microsoft Windows Server environment.

The evaluated system leveraged the following architecture and components:

- **Operating Environment**: Microsoft Windows Server.
- **JVM**: Java Virtual Machine configured with customized startup arguments for memory management, secure cryptographic settings, and application performance tuning.
- **Application Server**: Apache Tomcat (bundled with ColdFusion).
- **Identity Provider (IdP)**: LDAP-based directory services.
- **Security Monitoring**: Integration with an external Security Information and Event Management (SIEM) solution for centralized logging and alerting.

1.2 Authority

Department of Defense Instruction (DODI) 8500.01 requires that "all IT [information technology] that receives, processes, stores, displays, or transmits DOD information will be […] configured […] consistent with applicable DOD cybersecurity policies, standards, and architectures." The instruction tasks that 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 [National Security Agency/Central Security Service], 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 provides an environment that contributes to the security requirements of DOD systems, applicable NIST SP 800-53 cybersecurity controls must 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

Category	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 DOD 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 SRG Compliance Reporting

All technical NIST SP 800-53 requirements were considered while developing this STIG. Requirements that are applicable and configurable will be included in the final STIG. A report marked Controlled Unclassified Information (CUI) will be available for items that did not meet requirements. This report will be available to component authorizing official (AO) personnel for risk assessment purposes by request via email to: disa.stig_spt@mail.mil.

1.6 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.

1.7 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 AO. 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 DOD. While other agencies and organizations are free to use it, care must be given to ensure that all applicable security guidance is applied at both the device hardening level and the architectural level due to the fact that some settings may not be configurable in environments outside the DOD architecture.

1.8 Product Approval Disclaimer

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). Department of Defense 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 in accordance with (IAW) DOD policy and through RMF risk acceptance. Inputs into acquisition or preacquisition product selection include such processes as:

- National Information Assurance Partnership (NIAP) evaluation for National Security Systems (NSS) (https://www.niap-ccevs.org/) IAW CNSSP #11.
- National Institute of Standards and Technology (NIST) Cryptographic Module Validation Program (CMVP) (https://csrc.nist.gov/groups/STM/cmvp/) IAW federal/DOD mandated standards.

2. CONCEPTS AND TERMINOLOGY CONVENTIONS

Understanding the following key terms and concepts is essential for interpreting configuration guidance, assessing security posture, and ensuring compliance with STIG requirements.

- **Adobe ColdFusion**: A rapid application development platform that runs on the Java Virtual Machine (JVM) and uses ColdFusion Markup Language (CFML) to create dynamic, database-driven web applications.
- **CFML (ColdFusion Markup Language)**: A tag-based scripting language used within ColdFusion to build web applications. CFML is processed by the ColdFusion server and rendered as HTML or other output.
- **CFC (ColdFusion Component)**: A modular CFML file that encapsulates related functions and logic. CFCs are used to organize code into reusable components and are foundational to object-oriented development in ColdFusion.
- **CFTHREAD**: A CFML tag used to run code asynchronously in separate threads. This enables performance optimization by allowing tasks to execute in parallel rather than sequentially.
- **RDS (Remote Development Services)**: A ColdFusion feature that allows remote access to server resources for development and debugging purposes. RDS is commonly disabled in production environments due to its security risks.
- **CCS (ColdFusion Component Server)**: A subsystem of ColdFusion that provides service orchestration and component execution. It is often referenced in clustering or distributed deployment configurations.
- **CFSTAT**: A ColdFusion performance monitoring utility that provides real-time statistics on server activity. It outputs metrics such as template requests, memory usage, and thread counts, aiding in performance analysis and troubleshooting.
- **Administrator Console**: A web-based interface used to configure and manage ColdFusion settings, including security, logging, data sources, and performance tuning.
- **JVM Arguments**: Custom startup options passed to the Java Virtual Machine running ColdFusion. These are used to configure memory settings, garbage collection behavior, cryptographic providers, and other low-level performance or security parameters.
- **Server Configuration Files**: XML-based files such as server.xml, neo-security.xml, and neo-logging.xml define key aspects of ColdFusion's operation, including connectors, logging patterns, and admin access.
- **Identity Provider (IdP)**: LDAP was used as the identity provider to authenticate administrative users accessing the ColdFusion Administrator Console.
- **SIEM (Security Information and Event Management)**: An external platform used to aggregate, analyze, and alert on logs and events from the ColdFusion environment and underlying operating system.

2.1 Architecture

From an architectural standpoint, ColdFusion is a Java-based platform that sits between the web server and back-end data sources, interpreting CFML code to deliver dynamic content. Its tightly integrated Admin Console, robust Java foundation, and extensibility make it well-suited for enterprise applications.

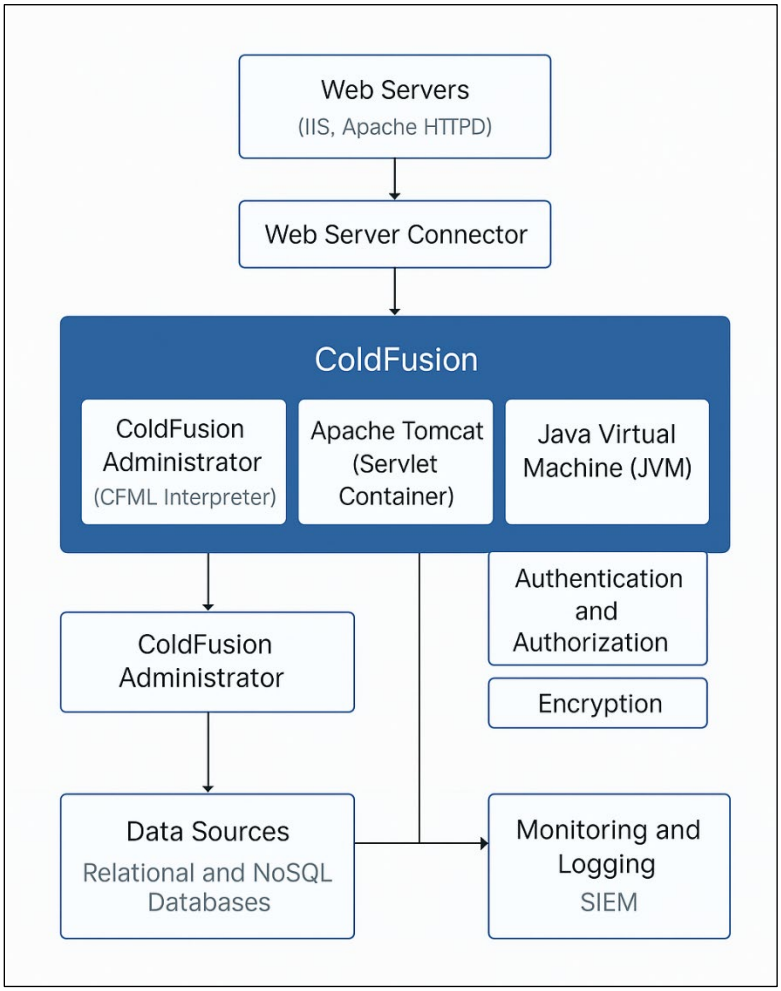
1. Application Server Layer:
   - Apache Tomcat: ColdFusion runs atop Apache Tomcat, which serves as its underlying Java servlet container. Tomcat handles HTTP(s) requests, servlet execution, and connection management.
   - ColdFusion Engine (CFML Interpreter): The CFML engine interprets and executes ColdFusion Markup Language (CFML) code. CFML is a tag-based scripting language used to create dynamic content.
   - Java Virtual Machine (JVM): ColdFusion runs within a JVM, allowing it to leverage Java libraries and benefit from Java's performance and security features. JVM arguments can be customized for memory management, cryptographic settings, and garbage collection.
2. Web Server Connector:
   - ColdFusion supports integration with external web servers (e.g., IIS, Apache HTTPD) through a web server connector, which forwards HTTP requests to the Tomcat-based ColdFusion engine.
   - This separation enables flexible deployment options and supports load balancing and clustering.
3. ColdFusion Administrator:
   - A browser-accessible web interface for configuring and managing ColdFusion settings, including:
     - Data source definitions.
     - Logging and debugging.
     - Security and user management.
     - Scheduling and performance tuning.
4. Data Access Layer:
   - ColdFusion provides built-in support for relational databases via JDBC.
   - Developers define data sources through the Admin Console, enabling CFML code to easily interact with back-end databases using CFQuery tags or ORM.
   - NoSQL databases are also supported via configurable drivers.
5. Security Components:
   - Authentication and Authorization: ColdFusion LDAP/Active Directory integration and external SAML or OAuth providers. This STIG was tested using LDAP.
   - Access Controls: Role-based restrictions can be applied at the component, method, or application level.
   - Encryption: ColdFusion provides support for FIPS-compliant algorithms, TLS configuration, and encrypted session/client variables.
6. Asynchronous and Parallel Processing:
   - CFTHREAD: Used to run background or parallel tasks within a single request life cycle.
   - Scheduled Tasks: Built-in job scheduler for triggering scripts at defined intervals.
7. Monitoring and Logging:
   - ColdFusion can be integrated with external SIEM tools for centralized event collection.
   - Built-in tools include:

- CFSTAT: Outputs real-time server statistics.
- Server Monitor: Graphical tool for tracking performance, memory usage, and thread activity.

- Logging is configurable through neo-logging.xml, and log files are written to the defined directory.

8. Development and Deployment Features:
   - CFCs (ColdFusion Components): Modular, object-oriented units of reusable CFML code.
   - Custom Tags and Functions: Extend CFML using custom libraries or Java classes.
   - RDS (Remote Development Services) (disabled in production): Allows developers to remotely browse files, execute code, and access databases.

Figure 3-1: Architecture



3. GENERAL SECURITY REQUIREMENTS

3.1 Summary

ColdFusion deployments must be secured through a layered approach that incorporates both platform-level and application-level controls. Underlying technologies must be hardened using appropriate STIGs. Systems must enforce strong authentication mechanisms such as LDAP to restrict administrative access. Unused or high-risk features such as Remote Development Services (RDS) must be disabled in production environments. ColdFusion servers must run with the minimum necessary privileges, and configuration files must be protected using appropriate file system permissions. All communications must be encrypted using TLS, and the system must log all administrative actions, errors, and access events to an external SIEM. ColdFusion must also employ secure JVM arguments and be regularly updated to mitigate vulnerabilities in the platform and underlying Java components.