IBM WEBSPHERE LIBERTY SERVER
SECURITY TECHNICAL IMPLEMENTATION GUIDE
(STIG) OVERVIEW

Version 1, Release 2

27 October 2022

Developed by IBM and DISA for the DOD

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 the Defense Information Systems Agency (DISA) of any nonfederal entity, event, product, service, or enterprise.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Page

1. INTRODUCTION

1.1 Executive Summary

The IBM WebSphere Liberty Server 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 such as Application Security and Development and appropriate operating system STIGs.

WebSphere Liberty Server is a Java-based software framework and middleware that is designed to host Java-based web applications. The product provides software libraries that hosted applications can use, as well as an operating environment for web-based applications. IBM currently offers WebSphere in different profiles, including the WebSphere Liberty and the WebSphere Traditional profiles. This STIG was written to be applied to the WebSphere Liberty profile version 21.0.0.1. The scope of the STIG is intended to address the management and security posture of the WebSphere product, not the applications hosted on the application server.

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

	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 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 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.

1.7 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). 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 pre-acquisition 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.
- DOD Unified Capabilities (UC) Approved Products List (APL) (https://www.disa.mil/network-services/ucco) IAW DODI 8100.04.

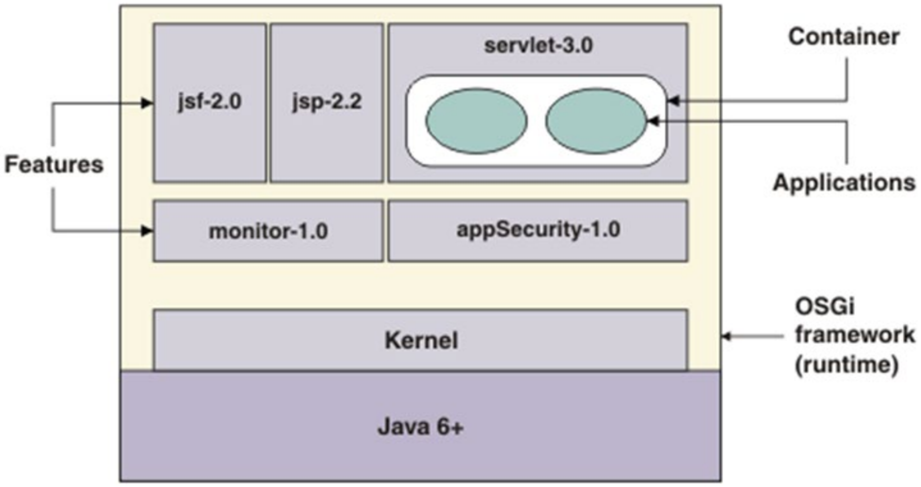
2. ASSESSMENT CONSIDERATIONS

2.1 Security Assessment Information

The WebSphere Liberty Server is a highly composable and dynamic runtime environment. OSGi services are used to manage component lifecycles and the injection of dependencies and configuration. The server process comprises a single JVM, the Liberty kernel, and a number of optional features. The feature code and most of the kernel code run as OSGi bundles within an OSGi framework. Features provide the programming models and services that are required by applications.

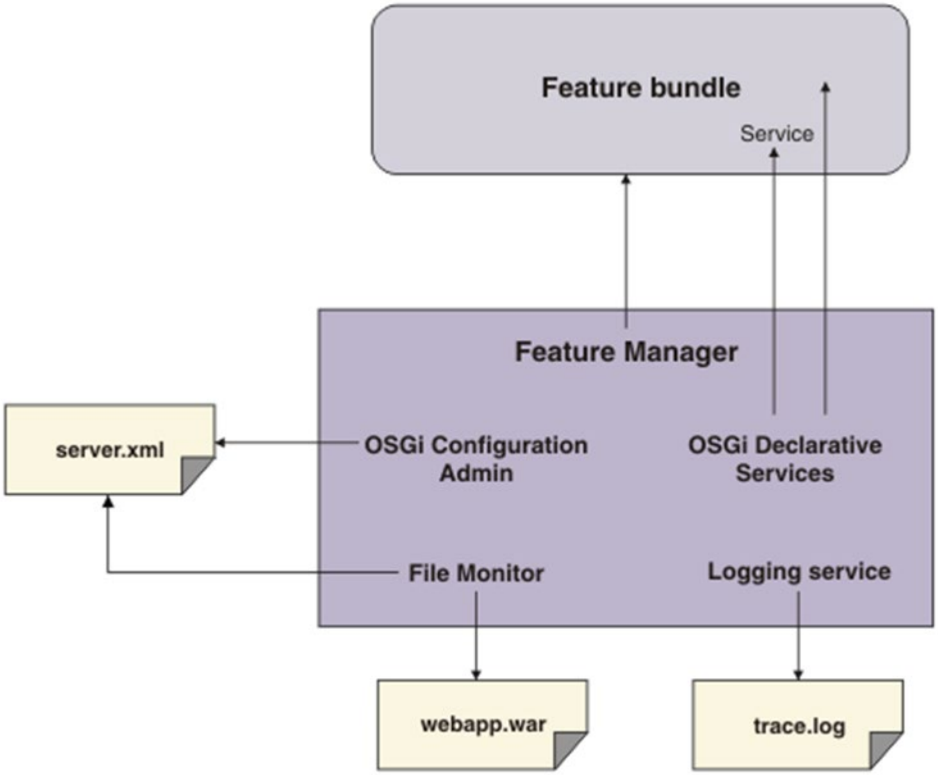
The kernel launcher bootstraps the system and starts the OSGi framework. The configuration is parsed, and then the configured features are loaded by the feature manager. The kernel uses OSGi services extensively to provide a highly dynamic runtime environment. The OSGi Configuration Admin service manages system configuration, and an OSGi Declarative Services component manages the lifecycle of system services. The file monitor service detects application and configuration file changes, and the logging service writes messages and debug information to the local file system.

Figure 2-1: WebSphere Liberty Server



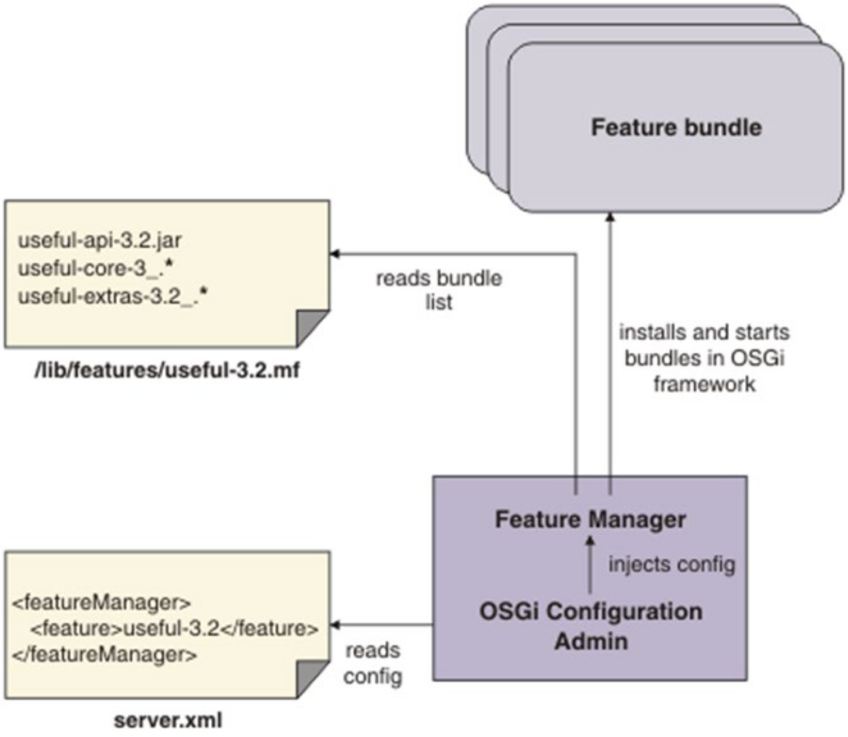
Features are specified in the system configuration files, which are the server.xml file and any other configuration files specified in the server.xml. The server configuration files populate the OSGi Configuration Admin service, which injects the feature configuration into the feature manager service. The feature manager maps each feature name to a list of bundles that provide the feature. The bundles are installed into the OSGi framework and started. The feature manager responds to configuration changes by dynamically adding and removing features while the server is running.

Figure 2-2: WebSphere Liberty Features



Runtime services provide configuration default settings so the configuration specified is kept to a minimum. Specify the features needed, along with any additions or overrides to the system default settings, in a server.xml file. Choose to structure the configuration into a number of separate files that are linked to the parent server.xml file by using an “include” syntax. At server startup or when the user configuration files are changed, the kernel configuration management parses the configuration and applies it over the system default settings. The set of configuration properties that belongs to each service is injected into the service each time the configuration is updated.

Figure 2-3: WebSphere Liberty Features During Runtime



The OSGi Declarative Services component is used so that function can be decomposed into discrete services, which are activated only when needed. This behavior helps to keep the runtime environment footprint small and the startup fast. Declared services are added to the OSGi service registry, and dependencies between services can be resolved without loading implementation classes. Service activation can be delayed until a service is used—when the service reference is resolved. Configuration for each service is injected as the service is activated and is reinjected if the configuration is later modified.

3. CONCEPTS AND TERMINOLOGY CONVENTIONS

3.1 WebSphere Design and Structure

WebSphere consists of the following concepts and elements:

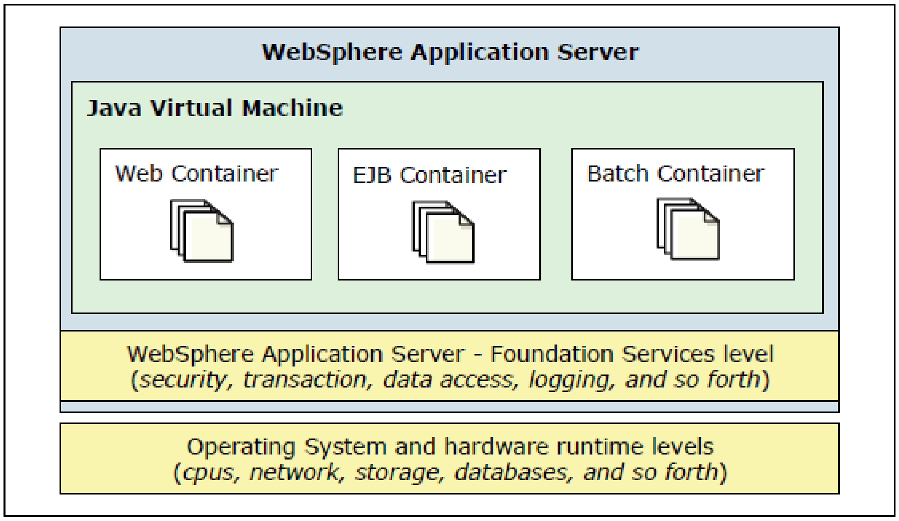
- Applications.
- Containers.
- Application server.

3.1.1 Applications

At the heart of WebSphere is the ability to run applications, including the following:

- Java Platform, Enterprise Edition (EE) applications.
- Portlet applications.
- Session Initiation Protocol (SIP) applications.
- OSGi applications.
- Batch applications.
- Business-level applications.

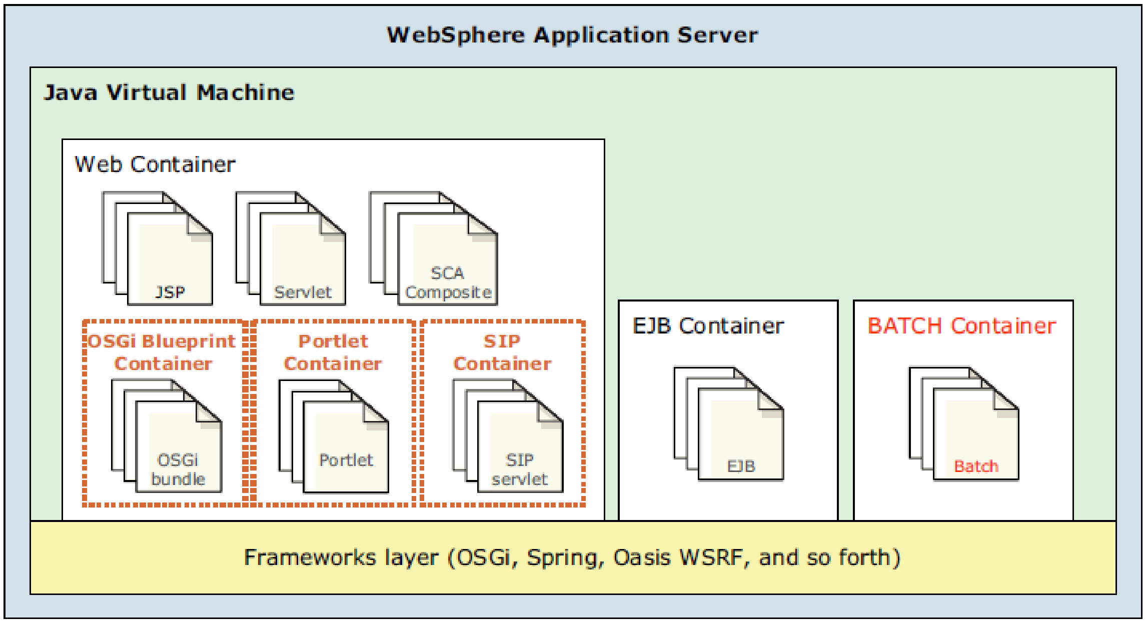
Figure 3-1: WebSphere Applications


WebSphere Application Server
Java Virtual Machine
Web Container
EJB Container
Batch Container
WebSphere Application Server - Foundation Services level
(security, transaction, data access, logging, and so forth)
Operating System and hardware runtime levels
(cpus, network, storage, databases, and so forth)

3.1.2 Containers

Containers provide runtime support for applications. They are specialized code in the application server that run specific types of applications. Containers can interact with other containers by sharing session management, security, and other attributes.

Figure 3-2: WebSphere Containers



3.1.3 Application Server

WebSphere Liberty Server is a standalone application server.