Community Review of DISA STIG

Introduction

This is a community review of the (USA) Defense Information Systems Agency (DISA) Security Technical Implementation Guide (STIG) dated 2021-12-27 which can be browsed conveniently at this address:

https://www.stigviewer.com/stig/apache_tomcat_application_sever_9/2021-12-27/

Similar documents have been drafted in the past and various members of this community including project maintainers, users, and administrators have expressed frustration regarding many of the items both included and not included in these guides. This document is intended to be a community-sourced review of the latest (at the time of this writing) version of the STIG, including specific recommendations from this community’s members for improvements and modifications to the STIG.

Recommendations and suggestions are welcome from any member of the community. We ask that if your recommended change is discussed and the group consensus is that no change should be recommended, please remove that suggestion from this document and do not bring it back up for discussion without some substantive change to the suggestion or the background/context of that suggestion. We ask this to avoid arguments raging in the history of this document. Feel free to have conversations on the various Tomcat-related mailing lists and post the results of those conversations into this document instead of using the document as a vehicle for communication.

This is a living document and changes can be made at any point, including changes that do not reflect any particular consensus of the Apache Tomcat Project Management Committee (PMC), the Apache Tomcat Security Team, etc. Any official recommendation made to DISA will be made for a specific version of this document which has supported consensus from one of more of the aforementioned groups.

Instructions

For anyone contributing to this review, please read the instructions below and participate in a constructive way. In order to coordinate our recommendations, it will be helpful if everyone follows a few simple rules:

1. To recommend an improvement or modification to the STIG, please file your commentary under a sub-section of this document whose name is the same as the Finding ID, which is how STIGs are generally organized. Please keep these Finding IDs in alphanumeric order for easy identification.
2. In a single short sentence, describe the overall change that is being recommended. For example "This finding is invalid," or "The instructions for identifying this finding are incomplete," etc.
3. Provide appropriate background and context if appropriate. Remember that extraordinary claims require extraordinary evidence.
4. Provide a specific recommendation or recommendations at the end of the section for the finding.

Recommendations

General Recommendations

No general recommendations at this point.

Modifications

V-222931

This finding contains multiple inaccuracies.

This finding claims that Apache Tomcat "currently operates only on JKS, PKCS11, or PKCS12 format keystores." Further, it claims that the finding only applies to JKS keystores. Finally, it claims that Tomcat will use a default password of "changeit" if the keystore was not created with a password.

Many of the above are factually incorrect. Apache Tomcat can use JKS and PKCS12 format keystores, but PKCS11 is another thing entirely which can be accessed in various ways. Apache Tomcat also supports the use of JCEKS keystores and PEM-encoded DER files which can include password-protected content similar to the other keystores.

In all of these cases, Apache Tomcat will use any password provided by the administrator in the appropriate position in the configuration file. Apache Tomcat will indeed use "changeit" as the default password if none is specified in the configuration to load the keys from a keystore. Apache Tomcat does not create keystores, and any administrator would have to explicitly specify "changeit" as the password used to protect a given keystore.

The recommendations should be made to protect the keystore with a password (and explicitly not to use "changeit" which is the commonly-used password in all Java-keystore-related HOWTOs and other guides) and has nothing to do really with Apache Tomcat's behavior at all. Furthermore, all keys should be recommended to be protected, not just those in JKS and PKCS12 key stores. JCEKS and PEM-encoded-DER files should be given the same prescription.

V-222938

The finding ignores valid settings that meet the requirement and the fix text is inaccurate.

Access logging may be configured per Engine, Host or Context. It is sufficient to configure one access log per Engine. It is also possible to meet the requirement to log all requests with various combinations of access logs at different levels.
The fix text states every element needs an access log valve. This is incorrect. The fix text - for the current meaning - should be every Host element needs an access log valve. However, note from the previous paragraph that other approaches are also possible.

A better way of phrasing this could be: For every possible combination of Engine, Host and Context that a request may be mapped to, at least one Container (Engine, Host and Context are all containers) must have a nested access log valve. The recommended approach is one per Host.

**V-222939, V222942**

The findings ignore valid settings that meet the requirement.

The findings require that specific values are included in the pattern for the AccessLogValve but does not take into account that the values "common" and "combined" also include the required values.

The findings need to be edited to accept "common" or "combined". The pattern example included in the finding is the "common" pattern.

**V-222950**

This finding confuses HTTP TRACE and software application stack traces and recommends a course of action that does not provide any security benefit.

This finding claims that "stack tracing provides debugging information ... that could be used to compromise the system." The recommended mitigation is to set the allowTrace attribute of all <Connector> elements to "false". The claim is that by setting this attribute, application stack traces will be suppressed.

The claim is incorrect. While revealing stack traces may be helpful to a potential attacker, setting allowTrace="false" does not disable application stack traces. Instead, it disables the HTTP TRACE verb which provides little to no security benefit. Application stack traces, if they had been possible to generate by an attacker before the configuration change, are still entirely possible for an attacker to cause.

We recommend that V-222950 be completely removed from the STIG as it has no perceivable benefit.

**V-222964**

This finding misses multiple TLS settings for remote JMX.

The finding should also cover:

- `com.sun.management.jmxremote.registry.ssl`
- `com.sun.management.jmxremote.ssl.enabled.protocols`
- `com.sun.management.jmxremote.ssl.enabled.cipher.suites`

**V-222968**

This finding requires that FIPS-validated cipher (suites) are used on secure connectors, but gets confused after that.

It's not clear whether this Finding requires that a FIPS-validated cryptographic module be used in FIPS mode, or that a FIPS-validated cryptographic module must be used (in any mode), or that only cipher suites be FIPS-140-2 validated (whatever that means). It also states that setting FIPSMode="on" on the AprLifecycleListener (a) will enable FIPS and (b) FIPS mode is not possible without it.

Enabling the AprLifecycleListener and using FIPSMode="on" is only applicable if OpenSSL is being used as the underlying cryptographic module, and that module has been built with FIPS support. There are other ways to use FIPS-validated cryptographic modules that do not use this setting.

This recommendation should clarify what is actually required, and be specific about how to ensure that a FIPS-validated module is actually in use.

**V-222981**

The title of this finding describes a requirement that cannot be met.

The LockOutRealm does not distinguish between admin and non-admin users. The lock out rules apply equally to all users.

The title should be edited to remove the text "...for admin users".

**Additions**

**Remote JMX**

The findings cover a number of settings regarding the use of remote JMX but does not cover all risks.

The default remote JMX authentication mechanism does not implement any access logging so it is not possible to audit use, or attempted use, of the interface.

The default remote JMX authentication mechanism does not implement any form of account lock-out and is therefore vulnerable to brute-force attacks.

Remote JMX has a very crude access control model. Access is either "read everything" or "read/write everything". Even read access provides access to potentially sensitive data. Any access via remote JMX should be considered equivalent to "super user" access. There is no way to mitigate this risk short of disabling remote JMX entirely. There is a strong case to be made for accessing JMX via some form of (local to Tomcat) proxy that addresses this risk. It is likely such a proxy would address other risks as well.