Security Scan Checklist

Security Scan Checklist

IMPORTANT: If you plan on performing a security scan of the IBM TRIRIGA Application Platform, please be sure to do so following the check list ****BEFORE**** doing the security scan:

1. Be sure to apply the latest Mod Release for your Version & Release, and the latest fix pack for that Mod Release (Following IBM's version nomenclature of Version.Release.Mod.FixPack) So if you are testing a base Mod Release like 1.2.3, and 1.3.4 is available, install 1.2.4 first. Then search Fix Central for the most current fix pack available, install that fix pack before continuing. For example if 1.3.4 FP 03 is the latest available GA fix pack, apply that before continuing. **ALL FIXES ARE DONE ON THE LATEST FIX PACK FOR THE LATEST MOD RELEASE FOR A VERSION.RELEASE**
2. For 3.4.2 and older, first, please plan a Platform Upgrade to the latest GA release ASAP.
    1. If you must remain on 3.4.2, Production mode must be set to Y. ProductionMode=Y in TRIRIGAWEB.properties. This will prevent stack traces from being shown to the end user.
3. A web server with SSL should be setup, and https needs to be used with either IHS, Apache or IIS. Some scanning software will flag running with http as a finding.
    1. The SSL layer should also be configured with strong encryption. Some scanning software flag 128 bit cyphers as vulnerable. This is a function of the web server layer, not core to the TRIRIGA Product.
4. A web server with SSO should be setup in front of TRIRIGA, so that company wide password policies can be followed.
    1. Make sure the SSO solution strips headers from the browser that are used in the SSO process (In Apache, research the module mod_headers, and specifically unset header)
5. The default application server ports should be protected by a firewall. Make sure that direct access to the TRIRIGA Application Server running SSO is disabled.
    1. For WebSphere, restrict 9060 and 9080 to only administrators and/or the webserver, no scans should be performed directly on the app server ports, as those ports are to be restricted to only administrators and
6. Make sure any admin web consoles are disabled. (Disable AdminConsole if you have enabled it. For WebLogic make sure /console access is disabled through the web server. If using Jboss, make sure the Web and/or JMX are disabled)
7. Verify the scan results make sense and are verifiable and reproducible. Sanity checks would include things like:
    1. If the results show an Apache Struts Vulnerability, TRIRIGA does not even use Struts, so all the results of the scan should be suspect because it flagged a package not even in use.
    2. If the results show a XSS vulnerability, replicate the vulnerability by hand. If you cannot replicate the result by hand, then it is likely a false positive.
8. Encrypt any plain text passwords.
    1. You should encrypt your database password in the server.xml file. See the usage of securityUtility to set the value of the password to an encrypted value.
    2. If you are still using JBoss, you are no longer supported, as you are not on the latest Mod release for 3.4. You must update to TRIRIGA Platform 3.4.2 or newer and use Liberty.
    3. If you are on 3.3.x and below, your installation is no longer supported (as of April 2018). If you still want to risk using JBoss and the end-of-lifed version of TRIRIGA, database passwords should be encrypted. (https://community.jboss.org/wiki/EncryptingDataSourcePasswords) Also you must restrict port 8009 to only allow connections from the web server and restrict 8001 for only local/internal connections.
9. Set your account lockout options
    1. Depending on your organization's requirements, you can update TRIRIGAWEB.properties set UNSUCCESSFUL_LOGIN_ATTEMPT=X where X is the number of incorrect login attempts allowed before the account is disabled.
10. Missing security based http headers.
    1. Many scans will say missing http headers are a vulnerability. These headers are somewhat arbirtary and the requirements change from scan software to scan software.
    2. TRIRIGA Platform provides a way to add http headers via properties added to the TRIRIGAWEB.properties.
    3. There are some headers that will break some functionality in TRIRIGA. For example, adding a "Content-Security-Policy" or "nosniff" headers will break various scripts and cause issues with the dynamic nature of TRIRIGA. TRIRIGA would need to be proven to be exploitable via these attacks in order to take a support case on them. A support case would be closed if the header is given without the supporting evidence that TRIRIGA is vulnerable to some attack. Most of these headers are just best practices and not official headers required by the http standard.
    4. If your scan states that the HTTP Headers X-My-Secure-Header and x-Frame-Options are not set, it should contain the values "Allow-only-my-value" and "SAMEORIGIN" respectively, these can be added to the TRIRIGAWEB.properties by adding the lines:

        httpheader.X-My-Secure-Header=Allow-only-my-value

        httpheader.X-Frame-Options=SAMEORIGIN

11. Some scans will complain that the jsessionid cookie does not have some flags set. We support and have tested two flags, the secure flag and httpOnly flag. The httpOnly flag is set by default for Liberty, and secure flag can be added if you are using https only.
    1. Secure Flag: When running in a SSL environment, you must setup the jsessionid cookie with the Secure flag. This is an application server dependent activity, and the TRIRIGA installer will not configure this. Please consult the Traditional Websphere, Websphere Liberty, or Oracle Weblogic documentation on how to set this flag.
    2. HttpOnly Flag: TRIRIGA supports the use of the HttpOnly flag, and when using WebSphere Liberty, this flag is on by default. For Traditional Websphere and Oracle Weblogic, this flag must be set in the Application Server configuration. Please refer to the documentation for those application servers on how to set this flag.

Submitting a ticket

If all of these pre-requisites are followed and a vulnerability still appears in the scan, support tickets can be entered with the IBM support team.

Note, a generic PMR opened to review a vulnerability scan will not be reviewed. The triage must happen before the PMR is opened, and a actual use case proviing the exploit needs to be opened.

Scan tools are intended as a starting point to look further into potential problems; not all items in a scan report are necessarily a vulnerability in the application.

A penetration test should be conducted based off the results, and each item should be proven to be exploitable within the context of the application before submitting a ticket.

The following link provides an example of the kind of information expected when submitting a vulnerability to support (do not use this page to actually submit the issue--if you are entitled to TRIRIGA support please use the same process you normally use for any support ticket): https://www.ibm.com/scripts/contact/contact/us/en/security_vulnerabilities/.

Vulnerability Scanning Methodologies

Static Source Scans – These scans take binary code and analyze it for potential coding issues that may result in an exploit. These tools analyze code statically, often without understanding the full context of the application stack and the variables involved. In cases like TRIRIGA Application Platform, a large and very dynamic web application platform, it's difficult to get reliable results from a static code scan. You may find false positives such as SQL Injection, Cross-Site Scripting, Insufficient Input Validation, Cross-Site Forgery and other issues flagged due to the code scanning tool not having a holistic understanding of the application stack. For this reason, **we do not recommend relying on static scans**.

Third Party Libraries - When dealing with vulnerabilities that arise because of third party libraries, it is necessary to prove that the TRIRIGA platform has a specific exploit by way of the methods explicitly used by such third party library.

Virus Scanning - TRIRIGA supports ICAP virus scanning, and relies upon 3rd party Anti Virus server frameworks to scan uploads for malicious content. Malicious text file contents , such as EICAR test files will not be detected automatically by TRIRIGA without having it configured to work with the ICAP service. Make sure to set the properties VIRUS_SCAN_ENABLED, VIRUS_SCANNER_IP_ADDRESS, and VIRUS_SCANNER_IP_PORT in order to point to 3rd party Anti Virus software to remediate.

Recommended: Dynamic Web Scans – These tools will scan the actual application running in context of a web container. Most are passive scans which take a starting URL point, and then spider through the application trying to manipulate and exploit various endpoints. There are also active scanning tools which will run scans while using the application. The latter is useful for true penetration testing to prove an exploit from a passive scan, because app scans can also result in a number of false positives (however it will be much more honed than a static source scan).

To search for known vulnerabilities, please reference the IBM PSIRT Blog: https://www.ibm.com/blogs/psirt/

For more information on IBM's procedures for code scans and app scans as a part of our development lifecycle, please refer to the IBM SEF redbook: http://www.ibm.com/developerworks/library/se-framework/