TOSHIBA

TGCS Response to Log4Shell Vulnerability Version December 23, 2021

What is Log4Shell Vulnerability?
On December 10, 2021, critical zero-day security vulnerability CVE-2021-44228 (https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-44228) in log4j was published along with the proof of concept (an example of working exploit). This vulnerability, also called Log4Shell, received a CVE base score of 10 which means it is critical vulnerability with extremely high probability of exploitation. An attacker can perform a so-called Remote Code Execution, i.e., run a custom code on remote server by sending specific strings within server requests. As a result of the attack, an attacker can gain full access to the remote server.

According to multiple sources including Apache Foundation, log4j versions 2 and above are vulnerable. Apache Foundation released the mitigation instructions and log4j version 2.17 which fixes the issue: https://logging.apache.org/log4j/2.x/security.html

Are TGCS products vulnerable?
TGCS performed thorough reviews of its core software products. Our preliminary findings show that our TGCS products,

- TCx Elevate Server Platform (all versions),
- Retail Enterprise Management Service (REMS, all versions under support 1.5, 1.4, and 1.3.1-1),
- PAS (all versions as they use REMS)
- WebPOS (specific customer versions)
  contain some versions of log4j2 which can be vulnerable to log4Shell exploits if access to your local network is compromised.

- CHEC EBOSS version 7.1 uses IBM WebSphere product, which is exposed to Log4Shell according to IBM security bulletin.

- BOSS and EBOSS 7.2 levels < 7567 (BOSS 7.2.3) contain log4j-core package and therefore are exposed to log4shell vulnerability.

We will continue to work with specific clients that have customization built on the core to determine if vulnerabilities exist.

There is a security update immediately available for WebPOS; it has been communicated to affected customers. TGCS engineers are currently developing security updates for other products. We will continue working on code reviews, internal and external scans, and mitigation measures in addressing the vulnerability affecting Elevate Server Platforms and REMS.

TOSHIBA

Before the patch is available, how to mitigate the log4Shell vulnerability of TCx Elevate Server Platform?

Most current exploits are coming through the Internet, so immediately verify that your controllers running TCx Elevate Server Platform are protected from external calls.

Additionally, follow the instructions published by Apache in this article: https://logging.apache.org/log4j/2.x/security.html, where Apache recommends removing the JndiLookup class from the classpath. Note that Elevate does not contain log4j-core jar, which is the only jar that is vulnerable to log4shell according to Apache. However, as a precautionary measure, delete JndiLookup class as recommended by Apache:

zip -q -d **pax-logging-log4j*.jar** org/apache/logging/log4j/core/lookup/JndiLookup.class

Before the patch is available, how to mitigate the log4Shell vulnerability of REMS?

Most current exploits are coming through the Internet, so immediately verify that your controllers running Retail Enterprise Management Service (REMS) are protected from external calls.

Additionally, follow the instructions published by Apache in this article: https://logging.apache.org/log4j/2.x/security.html, where Apache recommends removing the JndiLookup class from the classpath. Note that REMS does not contain log4j-core jar, which is the only jar that is vulnerable to log4shell according to Apache. However, as a precautionary measure, delete JndiLookup class as recommended by Apache:

1. Stop REMS service.
2. Delete the directory and files:

   <REMS Installation path>\data

   <REMS Installation path>\karaf.pid

   <REMS Installation path>\lock

3. Use the tool of your preference to edit the jar file named:

   C:\ REMS Installation path>\system\org\ops4j\pax\logging\pax-logging-log4j2\1.11.4\ pax-logging-log4j2-1.11.4.jar

And delete the file named: org/apache/logging/log4j/core/lookup/JndiLookup.class

4. Start REMS Service.
5. Wait until all the services are Active.

TOSHIBA

How to mitigate log4shell vulnerability in IBM WebSphere?
Please follow instructions provided by IBM in security bulletin:
https://www.ibm.com/support/pages/node/6525706

What if you cannot update IBM WebSphere?
If you cannot upgrade IBM WebSphere for any reason as recommended by IBM security bulletin referenced above, you can perform the following mitigation steps (which are also described in the same IBM security bulletin):

Set the JVM custom property log4j2.formatMsgNoLookups to the value true

- For information on setting custom JVM custom properties in WebSphere Application Server, see https://www.ibm.com/docs/en/was-nd/9.0.5?topic=jvm-java-virtual-machine-custom-properties
- After setting the JVM custom property, restart the application server.

Before the patch is available, how to mitigate the log4Shell vulnerability of BOSS 7.2 levels < 7567 (BOSS 7.2.3)?
Most current exploits are coming through the Internet, so immediately verify that your servers are protected from external calls.

Additionally, follow the instructions published by Apache in this article: https://logging.apache.org/log4j/2.x/security.html, where Apache recommends removing the JndiLookup class from the classpath.

Stop TomEE service before proceeding with the following steps, and once completed, the service has to be re-started.

For each of the following .jar files, please remove this class:
org/apache/logging/log4j/core/lookup/JndiLookup.class

Files to remove class from:
C:\Program Files\Toshiba\ScsBoss\TomEE\temp\1-DataReplicator\WEB-INF\lib\log4j-core-2.10.0.jar
C:\Program Files\Toshiba\ScsBoss\TomEE\temp\2-EventReceiver\WEB-INF\lib\log4j-core-2.10.0.jar
C:\Program Files\Toshiba\ScsBoss\TomEE\webappslane\DataReplicator\WEB-INF\lib\log4j-core-2.10.0.jar
C:\Program Files\Toshiba\ScsBoss\TomEE\webappslane\EventReceiver\WEB-INF\lib\log4j-core-2.10.0.jar

If you are unsure how to remove a class from a .jar file, one option is to use 7zip.

1. Download and install 7zip.
2. Add a path to the 7-Zip directory in your environment PATH variable.

3. Open a CMD window as admin
4. For each .jar file listed above,
    a. Navigate to the directory
    b. Copy the log4j-core-2.10.0.jar file to a file name with a different extension (log4j-core-2.10.0.old, for instance)
    c. Run the following command, confirming that it indicates a file was removed:
    7z d log4j-core-2.10.0.jar org/apache/logging/log4j/core/lookup/JndiLookup.class
    d. If you want to confirm the class has been removed, you can run "7zfm .", which will start a 7zip GUI and navigate to where the file was to ensure it is no longer in the .jar.

Who should you contact if you have questions?
Please contact us through the same channels as you typically use for other issues and questions, such as technical support or your TGCS account representative.

Changes in Version December 15
- Added information about IBM WebSphere vulnerability and its effect on CHEC EBOSS

Changes in Version December 16
- Added more information about mitigation steps for IBM WebSphere

Changes in Version December 20
- Updated the latest version of the Apache fix from 2.16 to 2.17.
- Added clarification regarding CHEC EBOSS 7.2.

Changes in Version December 23
- Added information about BOSS and EBOSS 7.2 exposure.
- Updated the mitigation instructions for REMS and Elevate based on most recent Apache recommendations.

changes in the product(s) and/or program(s) described in this publication at any time without notice.