|  |  |
| --- | --- |
| Name | M.Malathi M.E |
| College | National Engineering College |
| Project Title | Intelligent Threat Detection And Response: Ai Integration In Cybersecurity Frameworks |
| Contact Number | 9500468498 |

**Report**

Stage:2

1. **Nessus Plugins & Threat Detections And Response**

**Vulnerability Name: Apache Log4j Installed (Linux / Unix)**

**plugin : 44228**

**Severity : High**

**Description :**  One or more instances of Apache Log4j, a logging API, are installed on the remote Linux / Unix Host.  
  
The plugin timeout can be set to a custom value other than the plugin's default of 45 minutes via the 'timeout.156000' scanner setting in Nessus 8.15.1 or later.

Solution: the Apache Log4j vulnerability (CVE-2021-44228 and CVE-2021-45046) had been widely discussed. This vulnerability allowed remote code execution and was found in the Log4j 2 library, a popular Java-based logging utility used in many applications.

To address the Apache Log4j vulnerability, consider the following steps:

Identify Affected Systems:

Use vulnerability scanning tools to identify systems running affected versions of Log4j.

Apply Patches:

Update Log4j to a patched version that addresses the vulnerability. Check the Apache Log4j website or your Linux distribution's security advisories for information on the latest patched version.

Check Applications and Dependencies:

Identify and update any applications or systems that use Log4j as a library. Some applications may embed their own version of Log4j.

Restart Affected Services:

After applying the patch, restart any services or applications that use Log4j to ensure the changes take effect.

Monitor and Test:

Monitor system logs and network traffic for signs of exploitation.

Conduct thorough testing to ensure that the patch did not introduce any issues or conflicts with your applications.

Review and Update Security Policies:

Update security policies and practices to include regular checks for vulnerabilities and timely application of patches.

Consider Workarounds:

If immediate patching is not possible, consider implementing workarounds or mitigations recommended by security authorities.

Update Incident Response Plans:

Update incident response plans to include procedures for addressing vulnerabilities and responding to security incidents.

1. **Vulnerability Name: Apache Log4j Message Lookup Substitution RCE**

**plugin :**  155998

**Severity : High**

**Description :** A remote code execution vulnerability exists in Apache Log4j < 2.15.0 due to insufficient protections on message lookup substitutions when dealing with user controlled input. A remote, unauthenticated attacker can explolit this, via a web request to execute arbitrary code with the permission level of the running Java process.  
  
The plugin relies on callbacks from the target being scanned and hence any firewall rules or interaction with other security devices will affect the efficacy of the plugin. The plugin will also not yield results on Tenable.io and customers are encouraged to use plugin IDs 155999, 156000, 156001, and 156002 instead when scanning with Tenable.io. We continue to explore options for additional detection.  
  
This plugin will have the scanner listen for the callback on a random port in the 50000 to 60000 range.

Solution: Upgrade to Apache Log4j version 2.15.0 or later, or apply the vendor mitigation.  
  
Upgrading to the latest versions for Apache Log4j is highly recommended as intermediate versions / patches have known high severity vulnerabilities and the vendor is updating their advisories often as new research and knowledge about the impact of Log4j is discovered. Refer to https://logging.apache.org/log4j/2.x/security.html for the latest versions.

**Vulnerability Name: Debian DSA-5020-1 : apache-log4j2 - security update**

**plugin :**  156015

**Severity : High**

**Description :** The remote Debian 10 / 11 host has a package installed that is affected by multiple vulnerabilities as referenced in the dsa-5020 advisory.  
  
- Improper validation of certificate with host mismatch in Apache Log4j SMTP appender. This could allow an SMTPS connection to be intercepted by a man-in-the-middle attack which could leak any log messages sent through that appender. (CVE-2020-9488)  
  
- Apache Log4j2 <=2.14.1 JNDI features used in configuration, log messages, and parameters do not protect against attacker controlled LDAP and other JNDI related endpoints. An attacker who can control log messages or log message parameters can execute arbitrary code loaded from LDAP servers when message lookup substitution is enabled. From log4j 2.15.0, this behavior has been disabled by default. In previous releases (>2.10) this behavior can be mitigated by setting system property log4j2.formatMsgNoLookups to true or by removing the JndiLookup class from the classpath (example: zip -q -d log4j-core-\*.jar org/apache/logging/log4j/core/lookup/JndiLookup.class). Java 8u121 (see https://www.oracle.com/java/technologies/javase/8u121-relnotes.html) protects against remote code execution by defaulting com.sun.jndi.rmi.object.trustURLCodebase and com.sun.jndi.cosnaming.object.trustURLCodebase to false. (CVE-2021-44228)

SOLUTION: Upgrade the apache-log4j2 packages.

**Vulnerability Name: Ubuntu 18.04 LTS / 20.04 LTS : Apache Log4j 2 vulnerability (USN-5192-1)**

**plugin :**  156054

**Severity : High**

**Description :** The The remote Ubuntu 18.04 LTS / 20.04 LTS host has a package installed that is affected by a vulnerability as referenced in the USN-5192-1 advisory.  
  
- Apache Log4j2 <=2.14.1 JNDI features used in configuration, log messages, and parameters do not protect against attacker controlled LDAP and other JNDI related endpoints. An attacker who can control log messages or log message parameters can execute arbitrary code loaded from LDAP servers when message lookup substitution is enabled. From log4j 2.15.0, this behavior has been disabled by default. In previous releases (>2.10) this behavior can be mitigated by setting system property log4j2.formatMsgNoLookups to true or it can be mitigated in prior releases (<2.10) by removing the JndiLookup class from the classpath (example: zip -q -d log4j-core-\*.jar org/apache/logging/log4j/core/lookup/JndiLookup.class).  
(CVE-2021-44228)