National Cyber Security Centre

2112131855-NCSC

A part of the **Department of the Environment, Climate & Communications**

NCSC Alert

Critical vulnerability in Apache Log4j library - CVE-2021-44228

UPDATE

2021-12-13

Status: TLP-WHITE

This document is classified using Traffic Light Protocol. Recipients may share TLP-WHITE *information freely, without restriction. For more information on the Traffic Light Protocol, see https://www.first.org/tlp/. Please treat this document in accordance with the TLP assigned.*

Revision History

Revision	Date	Author(s)	Description
1.0	10 December 2021	CSIRT-IE	Initial Alert created regarding log4j
1.1	13 December 2021	CSIRT-IE	Additional information added regarding description & mitigation steps

Description

A critical vulnerability (CVE-2021-44228) has been identified in Apache Log4j and a patch has been released. Apache Log4j is an open source Java logging library used by many web applications and services.

The vulnerability allows an unauthenticated remote attacker to execute arbitrary code with the privileges of the web server and can be easily exploited by logging a crafted string. Java Naming and Directory Interface (JNDI) triggers a look-up on a server controlled by the attacker and executes the returned code. Proof of Concept exploit code has been published online. Malicious actors have been observed using these exploits to attack webservers. The NCSC advises that organisations assess their web servers for exposure to this risk. This should include services managed and provided by third party service providers.

Several protocols are being abused to gather information and install malware, including

- Lightweight Directory Access Protocol (LDAP)
- Secure LDAP (LDAPS)
- Remote Method Invocation (RMI)
- Domain Name Service (DNS)
- Hypertext Transfer Protocol (HTTP)

Attempts to exploit the vulnerability can be detected in log files for any services using affected log4j versions. The logs will contain user-controlled strings, for example, “Jndi:ldap”.

At the time of publication, the vast majority of observed activity has been scanning, but exploitation and post-exploitation activities have also been observed. Based on the nature of the vulnerability, once the attacker has full access and control of an application, they can perform a myriad of objectives. Microsoft[1] has observed activities including installing coin miners, Cobalt Strike to enable credential theft and lateral movement, and exfiltrating data from compromised systems.

Products Affected

Version of Apache log4j prior to log4j-2.15.0-rc2.
log4j 1.x (potentially impacted as these versions contain a JMS Appender which can use JNDI).

Many services use the logging library and are vulnerable to full server compromise. NCSC-NL maintain a list[2] of impacted services and their current status.

[1] https://www.microsoft.com/security/blog/2021/12/11/guidance-for-preventing-detecting-and-hunting-for-cve-2021-44228-log4j-2-exploitation/
[2] https://github.com/NCSC-NL/log4shell

Impact

Remote Code Execution - system compromise.

Mitigations

The first step an organisation must consider is to determine dependent services and applications (organisation managed and third-party integrated technologies) that leverage the Log4j library. Priority should be placed on external (internet) facing infrastructure. The following repository from the NCSC-NL has compiled a list of security advisories/bulletins linked to Log4Shell (CVE-2021-44228)[3].

The NCSC advises that updates be applied to vulnerable systems in accordance with local change management process. If immediate patching is not possible, administrators should implement the following temporary mitigation steps[4]:

- **For Versions Log4j 2.10.0 – 2.14.1**
  - If running log4j versions 2.10.0 through 2.14.1, The “formatMsgNoLookups” property is available, and the vulnerability can be mitigated by setting the system property "log4j2.formatMsgNoLookups" to “true”.

    ```
    -Dlog4j2.formatMsgNoLookups=true
    Example: java -Dlog4j2.formatMsgNoLookups=true -jar app.jar
    ```

  - Add or modify the configuration file (log4j2.component.properties) within an application folder to include:

    ```
    log4j2.formatMsgNoLookups=true
    ```

- **Log4J 2 Versions 2.0-beta9 – 2.10.0**
  - Per https://logging.apache.org/log4j/2.x/security.html - remove the JndiLookup class from the classpath.

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

- **Log4J 2 Versions 2.7 - 2.10.0**
  - Per https://issues.apache.org/jira/browse/LOG4J2-2109 - within PatternLayout configuration files, disable message pattern lookups by replacing each reference of

- **Log4J 2 Versions < 2.15.0**
  - Per https://logging.apache.org/log4j/2.x/, remove the JndiLookup and JndiManager classes from the log4j-core jar file. **Note:** Removal of the JndiManager will cause the JndiContextSelector and JMSAppender to no longer function.

[3] https://github.com/NCSC-NL/log4shell
[4] https://logging.apache.org/log4j/2.x/security.html

Exploitation attempts can be detected by inspecting log files for the characteristic URL pattern **${jndi:ldap:**//. Organisations should employ network and host based detection capabilities in order to check for exploitation attempts. The following regex will help with obfuscated attempts:

```
\${(\${(.?:|.?:.?:-)('|"|)*(?1)}*|[jndi:lapsrm]('|"|)}*){9,11}
```

A number of IDS signatures have been created in order to detect this activity, organisations should ensure that their Intrusion Detection Systems (IDS) systems are up to date to include these alerts. Emerging threats have open free community detections to alert on current exploit activity in the following SID range: SID range 2034647-2034652[5]. Crowdstrike have released the following Snort rules that may help to detect intrusion attempts[6]:

```
alert tcp any any -> $HOME_NET any (msg: "CrowdStrike CSA-211099 Log4Shell
RCE Attempt (CVE-2021-44228) [CSA-211099]"; flow: from_client, established; content:
"${jndi:ldap://"; classtype:web-application-attack; sid:8001895; rev:20211210;
reference: url,falcon.crowdstrike.com/intelligence/reports/CSA-211099;)

alert tcp any any -> $HOME_NET any (msg: "CrowdStrike CSA-211099 Log4Shell
RCE Attempt (CVE-2021-44228) [CSA-211099]"; flow: from_server, established; content:
"|ca fe ba be 00 00 00|"; content: ""; classtype: trojan-activity; sid:8001896; rev:
20211210; reference:url,falcon.crowdstrike.com/intelligence/reports/CSA-211099;)
```

There are a number of open source Host based detection tools available, including Yara rules and Sigma rules, a link to some of these can be found here: https://github.com/NCSC-NL/log4shell/blob/main/mitigation/README.md

Recommendations

The NCSC recommends that all organisations update to the latest version of Apache log4j or apply the mitigation measures if immediate update is not possible. Organisations should examine logs for attempts to exploit the vulnerability and establish alerts if attempts are made post update.

Organisations should confirm with their service providers that mitigation measures against this vulnerability are in place.

Organisations should notify the NCSC of attempts to exploit this vulnerability at the following email address: certreport@decc.gov.ie.

[5] https://rules.emergingthreatspro.com/open/

[6] https://www.crowdstrike.com/blog/log4j2-vulnerability-analysis-and-mitigation-recommendations/

DISCLAIMER: *This document is provided "as is" without warranty of any kind, expressed or implied, including, but not limited to, the implied warranty of fitness for a particular purpose. NCSC-IE does not endorse any commercial product or service, referenced in this document or otherwise.*

National Cyber Security Centre
29-31 Adelaide Road,
Dublin, D02 X285,
Ireland
Tel: +353 (0)1 6782333
Mail: certreport@decc.gov.ie
Web: ncsc.gov.ie
Twitter: ncsc_gov_ie