NCSC
National Cyber Security Centre

2112200905-NCSC

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

NCSC Alert

Critical vulnerabilities in Apache Log4j library (CVE-2021-44228, CVE-2021-45046 & CVE-2021-45105)

UPDATE 3

2021-12-20

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
1.2	15 December 2021	CSIRT-IE	Updated Products Impacted & Mitigation Steps
1.3	20 December 2021	CSIRT-IE	Added CVE-2021-45046 & CVE-2021-45105 to advisory, added separate Detection segment

Description

CVE-2021-44228

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.

CVE-2021-45046

The Log4j vulnerability CVE-2021-45046 has been upgraded to a base CVSS score of 9.0[2].

The fix to address CVE-2021-44228 in Apache Log4j 2.15.0 was incomplete in certain non-default configurations. When the logging configuration uses a non-default Pattern Layout with a Context Lookup (for example, $$ctx:loginId), attackers with control over Thread Context Map (MDC) input data can craft malicious input data using a JNDI Lookup pattern, resulting in an information leak and remote code execution in some environments and local code execution in all environments; remote code execution has been demonstrated on macOS but no other tested environments.

[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://logging.apache.org/log4j/2.x/security.html

CVE-2021-45105
Apache Log4j2 does not always protect from infinite recursion in lookup evaluation which may lead to a Denial of Service.

Products Affected

CVE-2021-44228

- All versions from 2.0-beta9 to 2.14.1
  Applications using Log4j 1.x are only vulnerable to this attack when they use JNDI in their configuration.

CVE-2021-45046

- All versions from 2.0-beta9 to 2.15.0, excluding 2.12.2

CVE-2021-45105

- All versions from 2.0-beta9 to 2.16.0

Many services use the logging library and are vulnerable to full server compromise. NCSC-NL maintains a list[3] of impacted services and their current status.

Impact

- Remote Code Execution - system compromise
- Denial of Service
- Local Code Execution

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. NCSC-NL has compiled a list of security advisories/bulletins linked to Log4Shell (CVE-2021-44228)[4].

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[5]:

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

CVE-2021-44228

- **Log4j 2.x mitigation**: Implement one of the mitigation techniques below.
  - Java 8 (or later) users should upgrade to release 2.16.0.
  - Users requiring Java 7 should upgrade to release 2.12.2.
  - Otherwise, in any release other than 2.16.0, you may remove the JndiLookup class from the classpath: zip -q -d log4j-core-*.jar org/apache/logging/log4j/core/lookup/JndiLookup.class
- Note that only the log4j-core JAR file is impacted by this vulnerability. Applications using only the log4j-api JAR file without the log4j-core JAR file are not impacted by this vulnerability.
- Also note that Apache Log4j is the only Logging Services subproject affected by this vulnerability. Other projects like Log4net and Log4cxx are not impacted by this.

CVE-2021-45046

- **Log4j 2.x mitigation**: Implement one of the mitigation techniques below.
  - Java 8 (or later) users should upgrade to release 2.16.0.
  - Users requiring Java 7 should upgrade to release 2.12.2.
  - Otherwise, in any release other than 2.16.0, you may remove the JndiLookup class from the classpath: zip -q -d log4j-core-*.jar org/apache/logging/log4j/core/lookup/JndiLookup.class
- Users are advised not to enable JNDI in Log4j 2.16.0. If the JMS Appender is required, use Log4j 2.12.2.
- Note that only the log4j-core JAR file is impacted by this vulnerability. Applications using only the log4j-api JAR file without the log4j-core JAR file are not impacted by this vulnerability.
- Also note that Apache Log4j is the only Logging Services subproject affected by this vulnerability. Other projects like Log4net and Log4cxx are not impacted by this.

CVE-2021-45105

- **Log4j 2.x mitigation**: Implement one of the mitigation techniques below.
  - Java 8 (or later) users should upgrade to release 2.17.0.
  - Alternatively, this can be mitigated in configuration:
    - In PatternLayout in the logging configuration, replace Context Lookups like $ctx:loginId or $$ctx:loginId with Thread Context Map patterns (%X, %mdc, or %MDC).
    - Otherwise, in the configuration, remove references to Context Lookups like $ctx:loginId or $$ctx:loginId where they originate from sources external to the application such as HTTP headers or user input.
- Note that only the log4j-core JAR file is impacted by this vulnerability. Applications using only the log4j-api JAR file without the log4j-core JAR file are not impacted by this vulnerability.
- Also note that Apache Log4j is the only Logging Services subproject affected by this vulnerability. Other projects like Log4net and Log4cxx are not impacted by this.

Detection

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[6]. Crowdstrike have released the following Snort rules that may help to detect intrusion attempts[7]:

```
 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 (log4j 2.17.0 for Java 8 (or later) and log4j 2.12.2 for Java 7) 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.

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

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

[7] 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