NETTITUDE
AN LRQA COMPANY

# CVE-2021-44228

# Log4Shell

Threat Report

Date: 13/12/2021
Version: 1.0

# Contents

# 1 Activity of Interest

## 1.1 Executive Summary

First discovered on 8 December 2021 and made more widely known on 9 December, a major remote code execution vulnerability in Apache's Log4j technology was disclosed which affects software products and applications that use the Apache Log4j library. The vulnerability (known as Log4Shell) allows remote users to compromise the systems running this service by sending a specially crafted HTTP packet to web servers running this library.

Its key characteristics are:

- **Ease of exploitation:** What makes CVE-2021-44228 especially dangerous is how easy it is to exploit. Someone who is an inexperienced hacker can successfully execute this attack simply by sending a single request that forces a log entry to be written. This will then enable the attacker to upload their own code into the application.
- **The prolific use of Apache Web Servers and the Log4j library: This** library is used in millions of websites, applications, and platforms, including a wide range of Apache products. It is also used in gaming platforms such as Minecraft.
- **Proof of concept code exists now and proactive scanning has started already:** The issue was discovered on 9 December and active scanning for this vulnerability started almost immediately. Vulnerability scanning tools have been updated to detect this issue and security products and tools themselves are being patched (where fixes are available).

What should you do?

- **Assess your estate:** Quickly understand your affected products. Review your asset database and vendors' products immediately for use of Apache Log4j. Recognise that vulnerability scanners may not detect or pick up all cases where this library is in use.
- **Test and validate:** Use vulnerability management tools to help scan your environment and follow up with penetration testing activity to validate the exploitability of systems if unsure.
- **Monitor and detect:** It will take time for you to understand where in your environment you are affected, so start to look for evidence of attacks and exploitation as soon as you can. Use a WAF to protect your public-facing systems and block/alert of activity related to this vulnerability and the use of scripts.
- **Apply Fixes/Remediation:** As soon as they become available apply the fixes and remediation actions from the relevant vendors. Upgrade to Log4j version 2.15.0, or apply their appropriate vendor recommended mitigations immediately. You should also update your Java instance to the latest version.

## 1.2 Background and Situation

First discovered on 8 December 2021 and made more widely known on 9 December, a major remote code execution vulnerability in Apache's Log4j technology was disclosed. Log4j is a logging library written in Java and the vulnerability, CVE-2021-44228, also commonly known as Log4Shell, allows a remote actor to send a crafted HTTP packet to servers or other software suite exposed to the internet, running the version below Log4j 2.15.0.

The vulnerable software will store the HTTP request as a legitimate log, which then executes the payload embedded in the log. The vulnerability, therefore, allows an attacker to initiate LDAP traffic to an attacker-controlled node from the Java Naming and Directory Interface" (JNDI). The attacker-controlled node will respond with a malicious Java class file that then begins running on the victim server.

### 1.2.1 How do I know if my systems are vulnerable?

Because of the nature of the issue, vulnerability scanners are not 100% effective at identifying the presence of CVE-2021-44228, even on a negative return there may still be an instance of Log4j in the network. Further advice on identifying vulnerable instances can be found in the Mitigation section.

### 1.2.2 How was this discovered?

Widespread scanning has begun with exploitation initially focussed on coin-mining activity (Kinsing malware), but this is likely due to the initial focus on Minecraft server exploitation. Further exploitation has been associated with the deployment of botnet malware such as Mirai and Tsunami. However, Log4j has a near-ubiquitous presence in almost all major Java-based enterprise apps and servers.

### 1.2.3 What are we likely to see next?

A flurry of activity will likely follow in the next 24-48 hours (Monday, 13 December onwards) as those involved in the Ransomware-As-A-Service (RaaS) supply chain, such as Initial Access brokers, will look to establish persistence before then selling the access on to those skilled in the later stages of kill chain activity.

Sophisticated Nation-State threat actors are likely to have a slightly delayed response, firstly while they assess their own exposure, secondly until they have conducted their planning estimate and identified use cases for the exploit in meeting their collection goals.

Of additional concern is that there is a high 'worm' potential from Log4Shell, and automated exploitation is likely. Although numerous indicators of activity currently seen have been included in this report, threat actors are also experienced at evading signature driven detections, therefore layered defence strategies will assist with identifying possible second-stage exploitation activity.

## 1.3 What is the issue that has been detected?

Log4j2 is an open-source, Java-based logging framework commonly incorporated into Apache web servers[1]. According to public sources, Chen Zhaojun of Alibaba officially reported a Log4j2 remote code execution (RCE) vulnerability to Apache on 24 November 2021[2,3]. This critical vulnerability subsequently tracked as CVE-2021-44228 (aka "Log4Shell"), impacts all versions of Log4j2 from 2.0-beta9 to 2.14.1[4].

Of note with this vulnerability is the widespread implementation of the technology across so many software vendors. Log4j is included with almost all the enterprise products released by the Apache Software Foundation, such as Apache Struts, Apache Flink, Apache Druid, Apache Flume, Apache Solr, Apache Flink, Apache Kafka, Apache Dubbo, and possibly many more. In addition, other open-source projects like Redis, ElasticSearch, Elastic Logstash, the NSA's Ghidra, and others also use it in some capacity or other. Therefore, all the companies that use any of these products are also indirectly vulnerable to the Log4Shell exploit.

The Apache Log4j library allows for developers to log various data within their applications. In certain circumstances, the data being logged originates from user input. Should this user input contain special characters and be subsequently logged within the context of log4j, the Java method lookup will finally be called to execute the user-defined remote Java class in the LDAP server. This will in turn lead to RCE on the victim server that uses the vulnerable log4j 2 instance[5].

## 1.4 Mitigations - What Can Be Done About It?

Clients should in the first instance update Log4j wherever possible.

According to p0rz9, the Chinese security researcher who first posted the exploit code online, CVE-2021-44228 can only be abused if the `Log4j2.formatMsgNoLookups` option in the library's configuration is set to false. Unfortunately, this option is set to false by default in old releases, meaning that all past Log4j releases since 2.10.0, when this option was added, are vulnerable by default.

The update in the Log4j 2.15.0 release basically sets this option to true in order to block attacks. Log4j users who update to the 2.15.0 version but then set this flag back to false will remain vulnerable to attacks. Similarly, Log4j users who can't update but set the flag to true can block attacks even on older versions.

In terms of identifying potentially vulnerable servers the following commands can be used:

**Windows PowerShell query:**

```
gci 'C:\' -rec -force -include *.jar -ea 0 | foreach {select-string "JndiLookup.class" $_}
| select -exp Path
```

**Linux:**

```
find / 2>/dev/null -regex ".*.jar" -type f | xargs -I{} grep JndiLookup.class "{}"
```

[1] https://logging.apache.org/Log4j2/2.x/
[2] https://logging.apache.org/Log4j2/2.x/security.html
[3] https://bug.cyberkendra.com/2021/12/09/Log4j22-remote-code-execution/
[4] Log4j2 Vulnerability: How to Mitigate CVE-2021-44228 | CrowdStrike
[5] https://unit42.paloaltonetworks.com/apache-log4j-vulnerability-cve-2021-44228/

### 1.4.1 If I can't patch, is there anything else I can do?

Clients can modify the logging config files by setting 'formatMsgNoLookups=true' to stop logging lookups from occurring. The older versions (2.10.0 and below) include '%m {nolookups}' in the configuration files instead of just the '%m' to stop the server from looking up data added to the logs. These are the fixes provided automatically in Version 2.15.0 of Log4j.

If vendors haven't yet issued a patch for internet facing mission-critical systems for your environment, then consider putting a Firewall in front of the appliance and block all outbound traffic from it. However, take care to put in exceptions for any critical 3rd party services e.g. Threat Feeds and also any automatic update services that this may restrict.

### 1.4.2 Can I detect someone trying to exploit this?

Analysts at Recorded Future observed scanning beginning within hours of the vulnerability disclosure, nearly all from the TOR network. GreyNoise and BadPackets observed similar traffic. Users can detect traffic attempting to exploit the software by monitoring packet captures or web server logs by looking at the string '{jndi:ldap://}' which is required to initiate the exploit. Most commonly this is in the User-Agent field, but may be found elsewhere if other protocols are used to initiate the traffic. This has been validated by POC code published to GitHub.

Note that this is only one potential exploitation of the vulnerability which uses JNDI lookup. There are other schemes which allow lookup as well, including:

- `{web:http://…}` which you can use for SSRF
- `{env:…}` which you can use to extract environment variables out of the server, .e.g AWS keys.

### 1.4.3 How have Nettitude responded to this?

The Nettitude SOC have been conducting activity associated with the vulnerability over the weekend. Known IOC's have been uploaded into our technology stack, providing both log and EDR based detections, with further threat-hunting activity already conducted on clients flagged as potentially vulnerable. Alarms have also been established using regex to pick up on exploitation, and these are being continually tuned in line with the latest developments.

Nettitude have also updated our Penetration Testing methodology to specifically test for this vulnerability. As of this morning, we have helped a number of our clients mitigate against this critical risk issue on their external and internal estate.

## 1.5 Assessment

With a score of 10/10 on the CVSSv3 severity scale, Log4Shell is as serious as it gets in terms of security flaws, being both remotely exploitable and requiring little technical skill to execute.

Industry commentary have compared it to potentially having the same level of impact as the Heartbleed and ShellShock vulnerabilities. Initial noise associated to Log4j is likely to be mass-scanning by security researchers and tentative reconnaissance from threat actors who are both looking to assess the scale and impact of the vulnerability in the operating landscape.

Whilst there is emerging evidence of exploits being used for coin-mining activity this is not widespread at the time of writing. Exploitation from more sophisticated threat actors is likely to take place in the next 24-48 hour window (as of Monday 13 December 2021), whilst processes are established for who, and how to target entities that are on advanced threat actor intelligence collection plans.

Remediation activity focussed on Log4j is likely to take a few weeks as vendors that have built capability on the underlying technology look to push out patches in the coming weeks. Couple this with the usual seasonal pressures on IT departments and threat actors will look to take advantage of those that do not prioritise appropriate patching. In terms of ransomware activity, given the relatively low levels of sophistication to target this vulnerability, we would expect to see a spike in activity from initial access brokers over the coming weeks as they look to establish persistence on targets.

This will then lead to a likely spike in selling activity to RaaS operators, who will conduct the more sophisticated actions in the kill chain. Cobalt Strike BEACON deployment is a likely next step that Microsoft security personnel have reportedly already detected[6].

# 2 Indicators

## 2.1 Botnet C2

log.exposedbotnets.ru # Tsunami botnet C2
194.59.165.21:8080 # Tsunami botnet C2
195.133.40.15:25565 # Mirai botnet C2
185.154.53.140:80 # Kinsing botnet C2
138.197.206.223:80 # Kinsing payload delivery server
18.228.7.109:80 # Kinsing payload delivery server
82.118.18.201:80 # Kinsing payload delivery server
92.242.40.21:80 # Kinsing payload delivery server
185.191.32.198:80 # Kinsing payload delivery server
80.71.158.12:80 # Kinsing payload delivery server
185.191.32.198:80 # Kinsing payload delivery server

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

45.137.155.55:80 # Kinsing payload delivery server
185.191.32.198:80 # Kinsing payload delivery server
45.137.155.55:80 # Kinsing payload delivery server
62.210.130.250:80 # Mirai payload delivery server
http://210.141.105.67/wp-content/themes/twentythirteen/m8 # Kinsing payload URL
http://159.89.182.117/wp-content/themes/twentyseventeen/ldm # Kinsing payload URL

## 2.2 Additional Initial Indicators

Scanning and exploitation activity has been seen from the following IP addresses:

1.14.17.89
1.116.59.211
1.209.249.188
5.157.38.50
5.182.210.216
5.199.143.202
18.27.197.252
20.71.156.146
20.205.104.227
23.120.182.121
23.129.64.131
23.129.64.135
23.129.64.139
23.129.64.141
23.129.64.145
23.129.64.146
23.129.64.148
23.154.177.2
23.154.177.7
34.247.50.189
35.170.71.122
37.19.212.104
37.123.163.58
45.12.134.108
45.13.104.179
45.33.120.240
45.83.193.150
45.130.229.168
45.137.21.9
45.146.164.160
45.153.160.2
45.153.160.130
45.153.160.131
45.153.160.133
45.153.160.134
45.153.160.135
45.153.160.136
45.153.160.138
45.154.255.147
45.155.205.233
46.105.95.220
46.166.139.111
46.182.21.248
51.15.43.205
51.15.76.60
51.77.52.216
51.255.106.85
54.173.99.121
60.31.180.149
61.19.25.207
62.76.41.46
62.102.148.68
62.102.148.69
62.171.142.3
64.113.32.29
66.220.242.222
67.205.191.102
68.79.17.59
68.183.44.143
68.183.198.247
72.223.168.73
79.172.214.11
81.17.18.60
81.17.18.61
81.17.18.62
81.30.157.43
82.221.131.71
88.80.20.86
89.163.154.91
89.163.252.230
89.249.63.3
91.203.5.146
91.219.237.21
91.245.81.65
92.242.40.21
93.189.42.8
94.142.241.194
94.230.208.147
101.35.154.34
103.90.239.209
103.103.0.141
103.103.0.142
103.214.5.13
104.244.72.7
104.244.72.115
104.244.72.129
104.244.73.43
104.244.74.57
104.244.74.211
104.244.75.74
104.244.76.13
104.244.76.170
104.244.76.173
104.244.77.235
104.244.78.213
104.244.79.6
107.189.1.160
107.189.1.178
107.189.8.65

107.189.10.137
107.189.11.153
107.189.12.135
107.189.14.76
107.189.14.98
107.189.14.182
107.189.29.41
107.189.29.107
107.189.31.241
109.70.100.26
109.70.100.27
109.70.100.28
109.70.100.34
109.70.100.36
109.237.96.124
116.24.67.213
121.4.56.143
121.5.219.20
122.161.50.23
122.161.53.44
133.18.201.195
134.56.204.191
134.122.34.28
135.148.43.32
137.184.28.58
137.184.96.216
137.184.98.176
137.184.99.8
137.184.102.82
137.184.104.73
137.184.106.119
137.184.111.180
138.68.167.19
139.59.8.39
139.59.175.247
140.246.171.141
142.93.34.250
142.93.36.237
142.93.151.166
143.110.221.204
143.198.32.72
143.198.45.117
145.220.24.19
146.56.131.161
147.182.131.229
147.182.150.124
147.182.154.100
147.182.167.165
147.182.169.254
147.182.199.94
147.182.213.12
147.182.219.9
150.158.189.96
151.80.148.159
151.115.60.113
157.230.32.67
157.245.109.75
159.65.3.102
159.65.58.66
159.65.146.60
159.65.155.208
159.65.175.123
159.65.194.103
159.89.113.255
159.89.180.119
159.223.9.17
161.35.119.60
162.247.74.202
163.172.157.143
164.90.199.216
167.71.13.196
167.99.164.160
167.99.164.201
167.99.172.58
167.99.172.213
170.210.45.163
171.25.193.20
171.25.193.25
171.25.193.77
171.25.193.78
172.106.17.218
175.6.210.66
176.10.99.200
176.10.104.240
178.17.171.102
178.17.171.150
178.20.55.16
178.62.79.49
178.176.202.121
178.176.203.190
179.43.187.138
181.214.39.2
185.4.132.183
185.10.68.168
185.14.97.147
185.38.175.130
185.38.175.131
185.38.175.132
185.56.80.65
185.83.214.69
185.100.86.128
185.100.87.41
185.100.87.202
185.107.47.171
185.107.47.215
185.107.70.56
185.129.61.1
185.129.61.4
185.130.44.108
185.154.53.140
185.165.168.77
185.220.100.240
185.220.100.241
185.220.100.242
185.220.100.243
185.220.100.244
185.220.100.245
185.220.100.246
185.220.100.247
185.220.100.248
185.220.100.249
185.220.100.250
185.220.100.251
185.220.100.252
185.220.100.253
185.220.100.254
185.220.100.255
185.220.101.32
185.220.101.33
185.220.101.34
185.220.101.35
185.220.101.36

185.220.101.37
185.220.101.38
185.220.101.39
185.220.101.41
185.220.101.42
185.220.101.43
185.220.101.44
185.220.101.45
185.220.101.46
185.220.101.48
185.220.101.49
185.220.101.50
185.220.101.51
185.220.101.52
185.220.101.53
185.220.101.54
185.220.101.55
185.220.101.56
185.220.101.57
185.220.101.58
185.220.101.60
185.220.101.61
185.220.101.62
185.220.101.63
185.220.101.129
185.220.101.131
185.220.101.132
185.220.101.133
185.220.101.134
185.220.101.135
185.220.101.138
185.220.101.139
185.220.101.140
185.220.101.141
185.220.101.142
185.220.101.143
185.220.101.144
185.220.101.145
185.220.101.147
185.220.101.148
185.220.101.149
185.220.101.150
185.220.101.151
185.220.101.152
185.220.101.153
185.220.101.154
185.220.101.155
185.220.101.156
185.220.101.157
185.220.101.158
185.220.101.159
185.220.101.160
185.220.101.161
185.220.101.162
185.220.101.163
185.220.101.164
185.220.101.165
185.220.101.167
185.220.101.168
185.220.101.169
185.220.101.170
185.220.101.171
185.220.101.172
185.220.101.173
185.220.101.174
185.220.101.175
185.220.101.176
185.220.101.177
185.220.101.178
185.220.101.179
185.220.101.180
185.220.101.181
185.220.101.182
185.220.101.183
185.220.101.184
185.220.101.185
185.220.101.186
185.220.101.187
185.220.101.188
185.220.101.189
185.220.101.190
185.220.101.191
185.220.102.7
185.220.102.8
185.220.102.241
185.220.102.242
185.220.102.246
185.220.102.249
185.220.102.252
185.220.102.253
185.220.102.254
185.220.103.4
185.220.103.7
185.220.103.119
185.232.23.46
185.250.148.157
188.166.48.55
188.166.74.97
188.166.92.228
188.166.122.43
188.166.223.38
191.232.38.25
193.3.19.159
193.31.24.154
193.110.95.34
193.189.100.195
193.189.100.201
193.189.100.203
193.218.118.183
193.218.118.231
194.48.199.78
194.59.165.21
194.135.33.152
194.163.133.36
195.19.192.26
195.54.160.149
195.123.247.209
195.133.40.15
195.176.3.19
195.176.3.24
195.206.105.217
195.251.41.139
195.254.135.76
197.246.171.83
197.246.171.111
198.98.51.189
198.98.60.19
199.195.250.77
204.8.156.142
205.185.115.217
205.185.117.149
206.189.20.141

| | | |
|---|---|---|
| 209.127.17.234 | 209.141.45.227 | 212.193.57.225 |
| 209.127.17.242 | 211.154.194.21 | 213.202.216.189 |
| 209.141.41.103 | 212.192.246.95 | 221.199.187.100 |
| 209.141.45.189 | 212.193.30.142 | |

# 3 Revision History

| Version | Issue Date | Issued by | Comments |
|---|---|---|---|
| 1.0 | 13 Dec 2021 | Nettitude Threat Intelligence | Initial Release |


NETTITUDE
AN LRQA COMPANY

**UK Head Office**
Jephson Court, Tancred Close, Leamington Spa, CV31 3RZ

**Americas**
50 Broad Street, Suite 403, New York, NY 10004

**Asia Pacific**
1 Fusionopolis Place, #09-01, Singapore, 138522

**Europe**
Leof. Siggrou 348 Kallithea, Athens, 176 74
+30 210 300 4935

**Follow Us**

solutions@nettitude.com
www.nettitude.com