Logjam: Log4j exploit attempts continue in globally distributed scans, attacks

news.sophos.com/en-us/2021/12/20/logjam-log4j-exploit-attempts-continue-in-globally-distributed-scans-attacks/

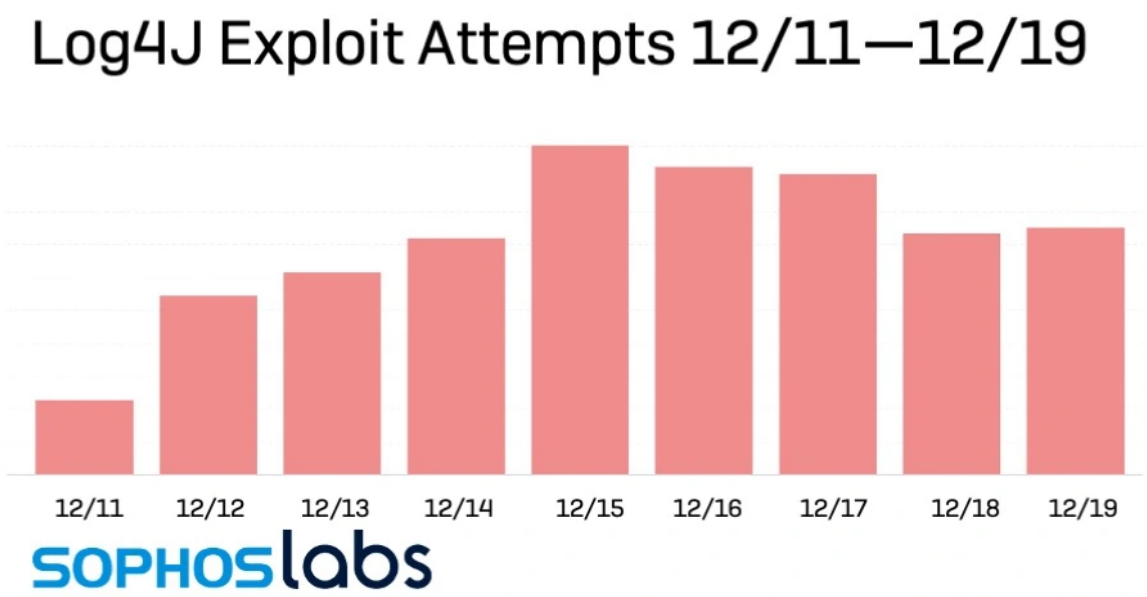
Sean Gallagher December 20, 2021

Since the first vulnerability in the Apache Foundation's Log4j logging tool was revealed on December 10, three sets of fixes to the Java library have been released as additional vulnerabilities were uncovered. This rapid iteration of fixes has left software developers and organizations worldwide scrambling to assess and mitigate their exposure with nearly daily-changing guidance. In the meantime, we've seen attempts to detect or exploit the vulnerability continue non-stop.

As we pass the first week since the exposure of the first vulnerability, SophosLabs has continued to track attempts against our customers' networks to exploit Log4Shell. The traffic we've observed includes benign scans by security researchers and penetration testers as well as malicious activity, and it does not directly reflect the state of criminal and state actor attempts to exploit the vulnerability. But from portions of the data, we can see enough about the requests to gain some insight into the infrastructure involved in these attempts, and in some cases the intent behind them.

What is certain is that we have not seen a significant reduction in exploit attempts since they peaked on December 15, and that these probes and exploits are coming from a globally distributed infrastructure. In some cases, a request comes from an IP address in one

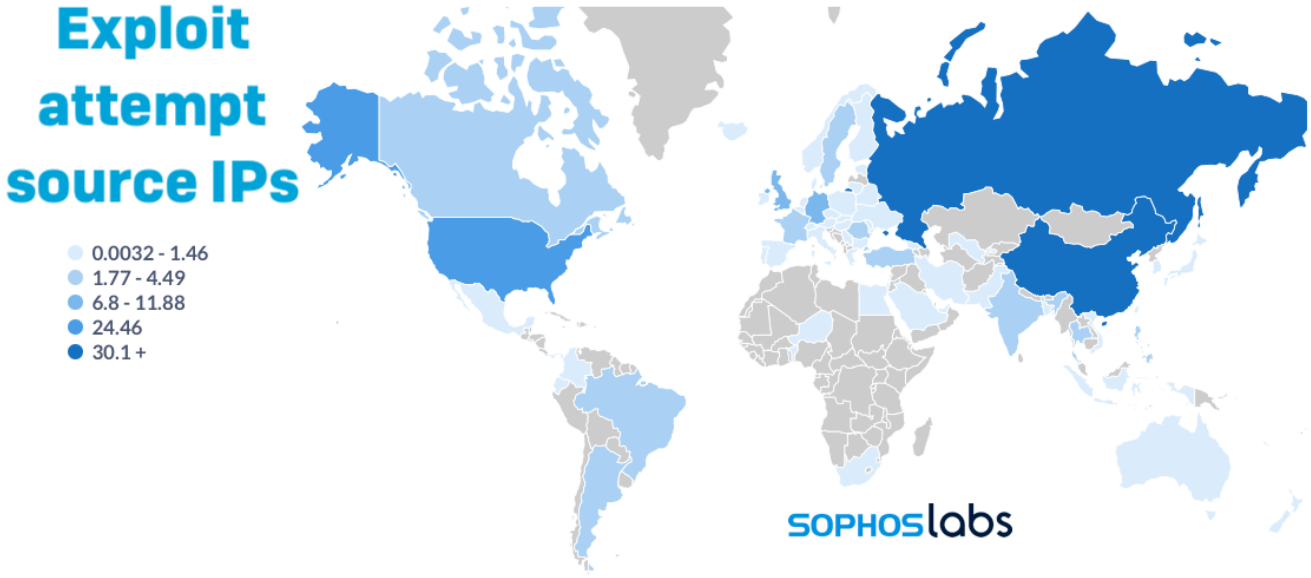
geographic region, with embedded URLs for Log4j that connect to servers elsewhere—sometimes multiple different servers. We have seen millions of incoming attempts to exploit Log4j in customer telemetry.



Who's doing this?

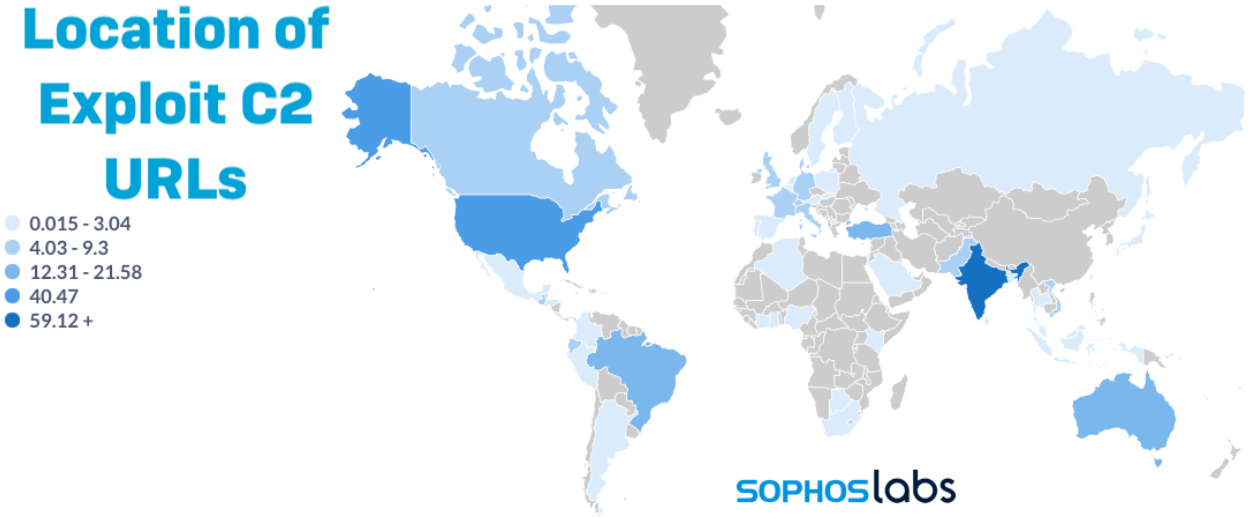
While we cannot distinguish the intent of every request, the segment of our telemetry that provided traffic details provides a snapshot of the infrastructure involved in Log4j abuse. Looking at the source of attempted abusive packets thus far, the vast majority come from IP addresses in Russia and China. This does not include traffic that conceals its source by use of virtual private networks; a statistically significant amount of traffic was routed through NordVPN's exit point in Panama, for example.

Of the traffic we could identify a source for, 11% came from a single IP address in Russia: 195[.]54[.]160[.]149. This IP address has been associated with the Kinsing cryptocoin-mining botnet.



Because of the way Log4j exploits work—by prompting "lookups" to remote servers via LDAP, DNS, and other Java Name and Directory Interface (JNDI) supported protocols—the lookup requests can be directed to a different location than the source of the exploit. For example, a request routed through NordVPN's Panama exit point (94[.]140[.]9[.]194) used a request URL that redirected to a URL in Kenya (41[.]169[.]130[.]19:8443/api/login).

Nearly two-thirds of these requests had URLs for infrastructure in India. And over 40% had URLs directed to infrastructure in the US. Over seven percent of exploit requests were directed to the Interactsh tool's domain—18% of all the traffic to US infrastructure. The numbers in the chart below add up to more than 100% because some exploit attempts used multiple URLs with different destinations.



Note: Percentages add up to more than 100%, as multiple URLs are included in some exploit attempts.

Because Interactsh has been used by both researchers and malicious actors, it's difficult to separate the benign from the bad—just as it is with much of the other traffic we're currently detecting and blocking. But it is clear that malicious exploit attempts remain a majority of this traffic.

Mitigation and protection

When the first patch for Log4j was released, the Apache team offered a number of work-arounds to prevent exploitation. But all of these fixes turned out to be moot as additional vulnerability paths were discovered. The only sure way to protect against exploitation—either to gain remote code execution or to cause denial of service—is to update software to use the current "safe" versions of Log4j (2.17.0 for Java 8, 2.12.3 for Java 7). A list of vulnerable commercial products is being maintained by multiple government computer security agencies, including the US' Cybersecurity and Infrastructure Security Administration (CISA). Organizations should assess their software's vulnerability as soon as possible and deploy updates where possible.

Where fixes are not yet available, network filtering definitions will protect against a large percentage of existing exploit traffic—but do not guarantee protection against emerging threats and highly targeted attacks.

Sophos continues to identify new methods of obfuscation for exploiting traffic, and new payloads that are being deployed via Log4j exploits. The following are the current Signature IDs published to Sophos intrusion protection products (with the latest in bold), by product, as of December 20:

Product	Signature IDs Published
XG	2306426, 2306427, 2306428, 58722, 58723, 58724, 58725, 58726, 58727, 58728, 58729, 58730, 58731, 58732, 58733, 58734, 58735, 58736, 58737, 58738, 58739, 58740, 58741, 58742, 58743, 58744, 58751, 58784, 58785, 58786, 58787, 58788, 58789, 58790, 58795, **58801**, **58802**, **58803**, **58804**, **58805**, **58806**, **58807**, **58808**, **58809**, **58810**, **58811**, **58812**, **58813**
Endpoint IPS	2306426, 2306427, 2306428, 2306438, 2306439, 2306440, 2306441
SG	58722, 58723, 58724, 58725, 58726, 58727, 58728, 58729, 58730, 58731, 58732, 58733, 58734, 58735, 58736, 58737, 58738, 58739, 58740, 58741, 58742, 58743, 58744, 58751, 58784, 58785, 58786, 58787, 58788, 58789, 58790, 58795, **58801**, **58802**, **58803**, **58804**, **58805**, **58806**, **58807**, **58808**, **58809**, **58810**, **58811**, **58812**, **58813**

Note that for SG, the updates are in the next SG sigpack update ,which will be released shortly.

The following is a list as of December 20 of all payloads Sophos has detected as part of Log4j exploit attempts (new payloads in bold) :

- **Linux/Miner-ABU,**Linux/Miner-ADH
- **Linux/Swrort-G (“Mettle”)**
- **Troj/Ransom-GME (TellYouThePass ransomware)**
- **Troj/StealthL-A (Stealth Loader)**
- **App/StlthLdr-A (Stealth Loader installer, PUA)**
- **Mal/ExpJava-AL, Mal/ExpJava-AN, Mal/ExpJava-AO (Khonsari downloaders)**
- **Troj/JavaDl-AAN and Troj/Java-AIN (Khonsari downloaders)**
- **Troj/JavaDl-AAO (N0t4n3xplo1t.class)**
- Troj/Mdrop-JMR, Troj/Mdrop-JMS, Troj/Mdrop-JMP
- Troj/Khonsari-A (new)
- Troj/JavaDl-AAN
- Troj/Java-AIN
- Troj/BatDl-GR
- Mal/JavaKC-B
- XMRig Miner (PUA)
- Troj/Bckdr-RYB
- Troj/PSDl-LR
- Mal/ShellDl-A
- Linux/DDoS-DT, Linux/DDoS-DS
- Linux/Miner-ADG, Linux/Miner-ZS, Linux/Miner-WU
- Linux/Rootkt-M