Munich Re Specialty Insurance

Whitepaper:
Log4j/Log4Shell Vulnerability

What is Log4j/Log4Shell?

The Log4Shell vulnerability was discovered on December 9, 2021, by a cloud security team at the e-commerce conglomerate Alibaba. It is found in Log4j, a library commonly used in Java applications for implementing logging. In this context, a library records actions, communications, and user inputs for a particular application.

Because of a Java feature called Java Naming and Directory Interface (JNDI), Java applications can be tricked into making network connections. JNDI normally provides naming and directory functionality for applications written using the Java programming language. Flaws in the vulnerable versions of Log4j allow an attacker to inject a strategically crafted string of code into Log4j, which arbitrarily runs it without verification of its origination. As a result, the data accepted from the end user is not properly sanitized and can allow for code injection, allowing attackers to run the malicious code that they want on the victim's device.

There are two reasons why the Log4Shell vulnerability has caused considerable alarm:

- First, because of Log4j's widespread use and prevalence, the vulnerability exists in a large number of publicly exposed devices; and,
- Second, Log4Shell allows attackers to perform remote code executions relatively easily (potentially opening up a new mass method of gaining and spreading access to victim networks).

According to a recent Microsoft report, attackers have exploited the Log4Shell vulnerability to install malware and cryptominers on compromised systems, steal system credentials, burrow deeper within infected networks, and steal data. Moreover, state-sponsored groups have been found to be exploiting the Log4j vulnerability by using it as a backdoor to deploy ransomware.

[1] *What is Apache Log4J vulnerability and how to prevent it?* PurpleBox RSS. (2021, December 17). Retrieved December 28, 2021, from https://www.prplbx.com/resources/blog/log4j

[2] Nguyen, T. (2019, February 25). *Logging agents vs. logging libraries: Which should you use?*. LogDNA. Retrieved December 28, 2021, from https://www.logdna.com/blog/logging-agents-vs-logging-libraries-which-should-you-use

[3] Ducklin, P. (2021, December 13). *Log4Shell explained – how it works, why you need to know, and how to fix it.* Naked Security by Sophos. Retrieved December 28, 2021, from https://nakedsecurity.sophos.com/2021/12/13/log4shell-explained-how-it-works-why-you-need-to-know-and-how-to-fix-it/

[4] Newman, L. H. (2021, December 14). *The LOG4J vulnerability will haunt the internet for years.* Wired. Retrieved December 28, 2021, from https://www.wired.com/story/log4j-log4shell

[5] Vaughan-Nichols, S. (2021, December 17). *Security firm Blumira discovers major new Log4j attack vector.* ZDNet. Retrieved December 27, 2021, from https://www-zdnet-com.cdn.ampproject.org

What to do about the Log4Shell vulnerability?

While the Log4Shell risk should not be taken lightly, there are actions you can take to mitigate the risk. First and foremost, it is critical to identify all impacted assets and then apply all relevant patches. However, if unable to patch, additional steps can still be taken in order to mitigate the damage from a potential Log4Shell attack:

- Use firewalls to prevent remote calls to servers. Strict firewall rules can break the communication flow back to that attacker and prevent the remote code execution from occurring. Furthermore, it is important to work with your IT teams to disable remote lookups. While these strategies will not completely nullify the risk, they will make it much harder for a threat actor to carry out an attack.
- Monitor assets for any indicators of exploitation or post-exploitation activity. There are efforts that are best done on both the host/device and network fronts simultaneously. However, if an organization is unable to do so, go with what is easier for you, starting with your public-facing assets and working your way inside the perimeter.

Indicators of Exploitation and Compromise

Indicators of exploitation or post-exploitation activity can be observed with the specific exploitation of the Log4j JNDI lookup feature:

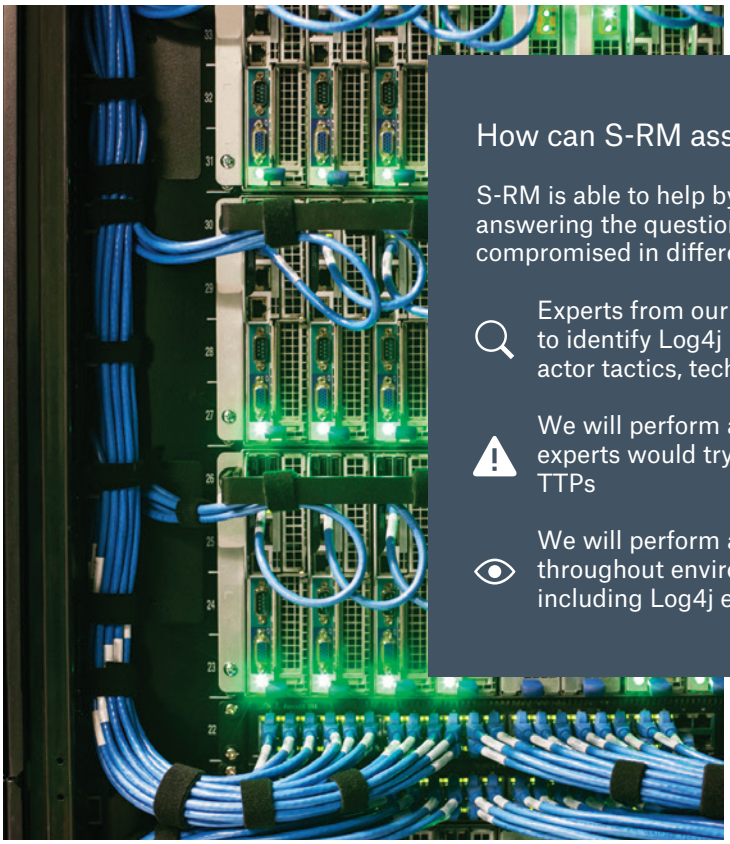
- Java Application Logs containing User-Agent strings that consist of "${jndi:ldap://...}" or a lot of "${...} lookups"
- Any logs containing these User-Agent strings with HTTP status codes of "200" likely indicate successful exploitation.

Common indications of compromise (post-exploitation) include:

- Abnormally high CPU utilization: Potentially indicative of cryptomining malware.
- Cryptomining malware will co-opt the target system's computational resources to mine a cryptocurrency of the attacker's choice.
- New or abnormal running processes, services, and scheduled tasks. These are potentially indicative of Persistent Remote Access Tools or other malware.
- Large spikes in outbound network traffic. This is potentially indicative of Data Exfiltration.
- Recurring outbound connections to unusual external IP addresses. This is potentially indicative of beaconing backdoors providing persistent remote access for a threat actor.
- Abnormal network connections to internal network assets from application servers running Log4j. This is potentially indicative of lateral movement.
- Recent authentications, privileged access, or configuration changes. This is potentially indicative of threat actor actions with the aim of gaining further access to the system or network.

S-RM is a global cyber security consultancy company that helps policyholders manage regulatory, reputational, and operational risks. Munich Re partners with S-RM to provide large policy holders with complimentary risk mitigation workshops. S-RM offers a suite of risk management cyber security consulting services available by request.

[6] Johnston, S. (2021, September 28). *What is Cryptocurrency Mining Malware?* Security Boulevard. Retrieved January 6, 2022, from https://securityboulevard.com/2021/09/what-is-cryptocurrency-mining-malware-2/



How can S-RM assist with recovery?

S-RM is able to help by giving our clients peace of mind. We can assist you in answering the question of whether your environment and systems were compromised in different ways.

- Experts from our incident response team will forensically examine systems to identify Log4j indicators of compromise (IOCs) and associated threat actor tactics, techniques, and procedures (TTPs).
- We will perform a vulnerability assessment, where our offensive security experts would try to identify and exploit vulnerable assets using the Log4j TTPs
- We will perform a compromise assessment by deploying EDR agents throughout environment to help identify ongoing malicious activity, including Log4j exploitation and post-exploitation in real time.

Clients can pick one option, a combination of them, or all three depending on their needs. If any vulnerabilities are found, S-RM is able to produce a report that would outline the mitigation steps needed to take in order to remediate these vulnerabilities. If active or historical exploitation was found, we can assist you in containing and eradicating the threat, as well as mitigate the damage of the compromise.

We believe that taking a proactive approach when it comes to the Log4j vulnerability will not only give our clients the peace of mind that their systems are not vulnerable, but will also save them time, pain, and potentially cost in the long run.

Please contact your insurance broker or S-RM directly:

Joseph Tarraf, Managing Director Cybersecurity
J.Tarraf@s-rminform.com

Andrew Shaughnessy, Senior Associate
A.Shaughnessy@s-rminform.com

Insurance brokers please contact Munich Re Specialty Insurance:

Steve Pacheco, U.S. Head of Cyber and Tech E&O
steve.pacheco@munichre.com

Disclaimer: This Whitepaper is provided for informational purposes only and does not constitute legal advice. It should not be construed as an offer to represent you, nor is it intended to create, nor shall the receipt of such information constitute, an attorney-client relationship. Readers are urged to seek professional or legal advice from appropriate parties on all matters mentioned herein.

© Copyright 2022
Munich Re Specialty Group Insurance Services, Inc.
All rights reserved.

Printed January 2022