Nangia Andersen LLP

log4Shell - Advisory

A member firm of ANDERSENGLOBAL

What is a Log4Shell?

A vulnerability was discovered in a popular java library used for logging information called Log4j. The vulnerability if successfully exploited may allow an adversary to execute remote untrusted code on the system running Log4j.

Unpatched version of Log4j will allow an adversary to trigger LDAP queries and other lookup queries via the Java Naming and Directory Interface(JNDI). Besides LDAP, JNDI supports other protocols such RMI, DNS, etc. This enables an adversary to send specially crafted strings to the affected system that contains commands wrapped in specific identifier sequences which causes the vulnerable server to trigger lookups to arbitrary servers which may reside within the network or worse on the internet.

What is affected?

Anything that uses the unpatched Log4j library for logging is affected. Although the list of affected technologies is quite extensive, some of the prominent affected technologies are Apache Struts, Apache Solr, Apache Druid, Elasticsearch, Apache Dubbo, Apple's iCloud, Twitter, Amazon and many more, due to the ubiquitous presence of the Log4j library.

This means that the vulnerability may not only lie in your applications, but also any other underlying infrastructure or third-party libraries, tools and interfaces that enable your applications and infrastructure to log details using Log4j library. It is therefore imperative to find all instances of code in your network and check whether it uses the Log4j library.

How bad is it?

It is very bad. Java and Log4 are prominently used across the internet. Like mentioned above a large number of technologies used by organizations are affected and more are being discovered to be vulnerable, on a daily basis. Even if developers in your organization do not directly use the vulnerable code it is quite possible that the open-source libraries that they depend on may be vulnerable. Vulnerable code has been discovered in technologies by vendors such as Cisco, VMware, IBM, Apple, Cloudflare and many more. The reason the vulnerability is concerning is because of the ease with which it possible to exploit the vulnerability and the high severity impact, allowing threat actors to gain control of remote machines and leaving little traces.

There have been reports that multiple threat groups are leveraging the vulnerability for running their espionage campaigns. Botnets of computers controlled by adversaries are being used to exploit the vulnerability on a large scale across the internet. Additionally, state sponsored attackers have also been exploiting the vulnerability to target private organizations and government entities. However, what is more worrisome is that the Log4j was in active exploitation at least 9 days before the vulnerability was publicly disclosed.

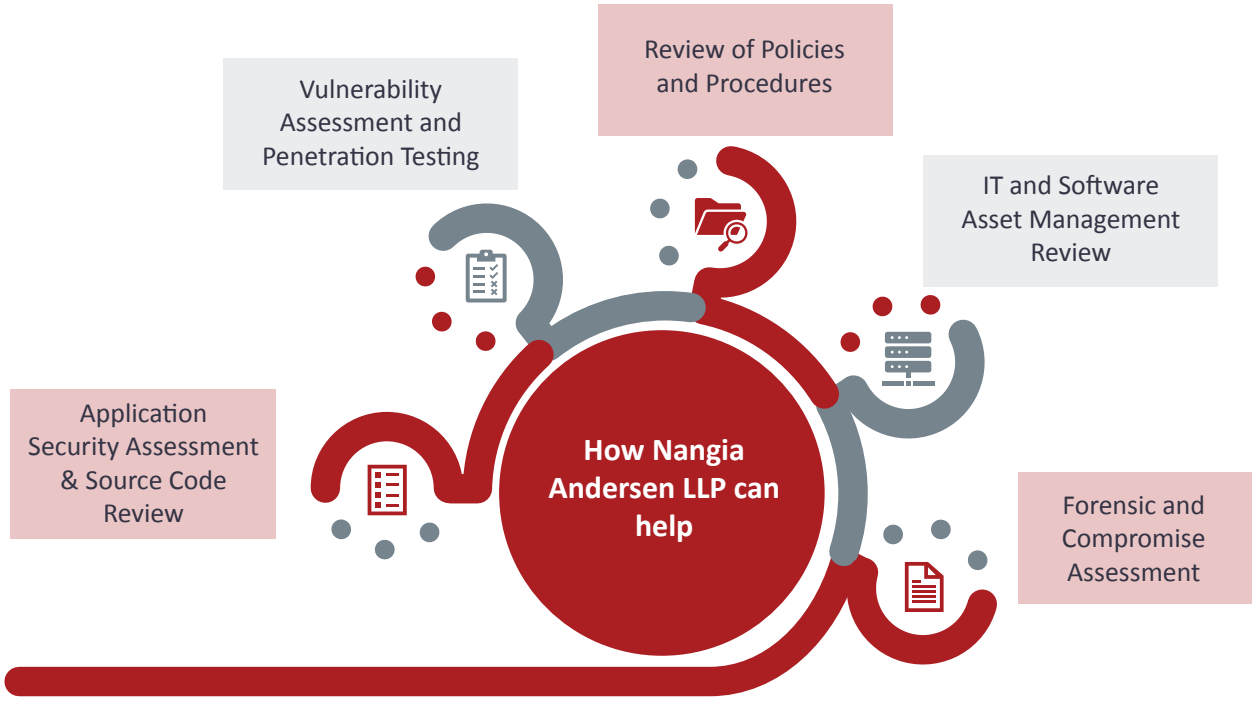
How do you protect yourself?

The foolproof way to protect yourself is to patch all the software, servers and applications to the latest version. If you think this is not an arduous task, think again. One needs to update every component of information systems using the Log4j library. This includes, third-party applications, middleware, and any other infrastructure using the vulnerable library.

Protection steps are as follows:

- Patch third-party software as per their published advisory.
- Look for instances of Log4j library in application s developed by the organization or vendors and update the version. For Java 8 and later the recommended version is 2.17.0. For Java 7, upgrade to version 2.12.2
- If you can't patch the vulnerable program, then suppress JNDI lookups by setting the **formatMsgNo Lookups=true.** However, whilst this may be possible to achieve in applications developed by the organization or customized applications provided by vendors, it may not be possible for off-the-shelf programs.
- Replace Context Lookups such as **${ctx:loginId}**or **$${ctx:loginId}** with Thread Context Map patterns or remove Context Lookups if they come from sources which are external to the application such as HTTP headers or user input. Again, this may not be possible to change directly in off-the-shelf programs and organizations may have to rely on vendors to publish relevant patches

How Nangia Andersen LLP can help



Meet our Experts

Srinivasa Rao
Partner & Leader - Risk Advisory Services
srinivasa.rao@nangia-andersen.com

Shrikrishna Dikshit
Partner - Risk Advisory Services
shrikrishna.dikshit@nangia-andersen.com

Kaushal Mehta
Partner - Risk Advisory Services
kaushal.mehta@nangia-andersen.com

Pushpendra Bharambe
Associate Director - Cyber Security Services
pushpendra.bharambe@nangia-andersen.com

Asif Balasinor
Manager - Cyber Security Services
asif.balasinor@nangia-andersen.com

NOIDA
(Delhi NCR - Corporate Office) A-109, Sector - 136, Noida - 201304 | T: +91 120 5123000

DELHI
(Registered Office) B-27, Soami Nagar, New Delhi - 110017
T: +91 0120 5123000

GURUGRAM
812-814, Tower B, Emaar Digital Greens, Sector-61, Gurugram, Haryana - 122102 T: +91 0124 430 1551

MUMBAI
11th Floor, B Wing, Peninsula Business Park, Ganpatrao Kadam Marg, Lower Parel, Mumbai - 400013, India | T: +91 22 61737000

CHENNAI
Prestige Palladium Bayan, Level 5, 129-140, Greams Road, Thousand Lights, Chennai - 600006
T: +91 44 46549201

BENGALURU
Prestige Obelisk, Level 4, No 3 Kasturba Road
Bengaluru - 560 001, Karnataka, India | T: +91 8022280999

PUNE
3rd Floor, Park Plaza, CTS 1085, Ganeshkhind Road, Next to Pune Central Mall, Shivajinagar, Pune - 411005

DEHRADUN
1st Floor, "IDA" 46 E.C. Road, Dehradun - 248001, Uttarakhand
T: +91 135 271 6300

www.nangia-andersen.com | query@nangia-andersen.com