LOGPOINT

Tackling Log4Shell requires a defence-in-depth strategy

There’s no one cyber tool that can protect your enterprise against Log4Shell. A combination of tools and a defense-in-depth mindset will give organizations the ability to detect post-compromise activity and stop the attack.

Dec 14, 2021 13:09 CET

Tackling Log4Shell requires a defence-in-depth strategy

The Log4Shell vulnerability is serious – it’s difficult to detect, used in lots and lots of software, and is the perfect vehicle to get malware into your network. There’s no one cyber tool that can protect your enterprise against Log4Shell. A combination of tools and a defense-in-depth mindset will give organizations the ability to detect post-compromise activity and stop the attack.

By Associate Security Analyst Engineer Bhabesh Raj, and Senior Cyber Analyst Kennet Harpsøe, LogPoint

COPENHAGEN – December 14, 2021– On Dec. 9, 2021, Apache disclosed a critical remote code execution vulnerability (CVE-2021-44228), also known as Log4Shell, which affects Apache Log4j versions 2.0-2.14.1. Log4j is a popular logging library in Java and is used in several enterprise applications, including Apache Struts, Flume, Kafka, Flink, Tomcat, Solr and VMware vCenter. Due to the prevalence of Log4j, defenders have been scrambling to identify which of their deployed applications are affected. Attempts at exploiting this vulnerability are particularly hard to detect because any string that might get logged by log4j could trigger the vulnerability, it could be anything from user-agent strings to email subject lines. The exploit could even be triggered down the line by, for instance, a vulnerable SIEM system that stores logs using log4j at some later point in time.

The IT surface of most modern enterprises is expansive and complex. It is impossible to stop or even detect all malicious activity at the perimeter, and some will inevitably seep through the cracks. Security teams responsible for large IT systems should assume breach when handling security. There will be malware in your system somewhere. The broadness and evasiveness of Log4Shell illustrates this point perfectly. All managers should ask themselves if they will be able to detect whether or not attackers are using the malware to penetrate their systems. A defense-in-depth strategy is the best way to respond.

Defense-in-depth strategy

Defense in depth is often analyzed through the lens of the Lockheed-Martin cyber kill chain or the Mitre ATT&CK framework, which in essence both say that any successful cyber attack will have to proceed through a series of conceptually well-defined steps. In the case of Log4Shell, attackers can use it to establish initial access and install malware like Cobalt Strike and most likely the initial access will be on an internet-facing machine like an Apache HTTP web server. The next steps after initial access are persistence and lateral movement. Using a defense-in-depth strategy, security leaders are equipped to ask how well they are covered to detect these types of activity. For instance, it might make sense to detect persistence with host agents that report back to SIEM and to detect lateral movement with a network detection system that reports to SIEM and establishes baselines for which machines communicate with which machines. In this way, security teams have the ability to uncover an Apache web server that acts as a client with respect to a domain-joined host behind it, which would be highly suspicious given the nature of Log4Shell. In other words, it pays to spend time and money on logging what you want instead of what you have.

With a defense-in-depth strategy and a well-thought-out logging regime, security leaders can extend security capabilities with additional software, such as user and entity behavior analytics (UEBA), which is ML-based software that can automatically generate baselines like the one in the aforementioned example. Teams can also use security orchestration, automation and response (SOAR) solutions to automatically initiate an action to reduce response time and help reduce the impact of an attack, such as automatically severing the connection between the Apache web server and the domain host. The SOAR can also notify an analyst after the response action, giving the analyst the relevant information to investigate the webserver and whether or not the connection should be re-established.

All LogPoint alerts are mapped to the MITRE ATT&CK framework, which helps security analysts understand the current security stance prioritize alerts.

What we know about Log4Shell

Chen Zhaojun of Alibaba Cloud Security Team reported CVE-2021-44228 (CVSS 10 out of 10) to the Log4j developers on November 24 leading to Apache releasing a patch on December 6. A PoC exploit was made public on December 10. Cloudflare's CEO has said that the earliest evidence of exploitation they have found so far is from December 1, suggesting the vulnerability was in the wild at least 9 days before being publicly disclosed.

Interestingly, Log4Shell uses the JNDI attack vector that was previously presented at BlackHat USA 2016. Exploitation of the vulnerability allows a remote attacker to execute code on the application if it logs the attacker-supplied string value with the attacker's JNDI LDAP server lookup. To trigger the vulnerability, an attacker needs to include a particular string in their requests, such as in user agents, that the application that uses the vulnerable Log4j library will log. The server then sends a request to the attacker's address via JNDI. The attacker's server, such as LDAP, responds by sending a path to a remote Java class file which gets injected into the vulnerable server process and executes the payload code. Additionally, administrators should be aware that threat actors can and have dumped secret data from environment variables, like AWS secret keys, to compromise the cloud.

Deutsche Telekom CERT has reported exploitation attempts originating from the Tor network. On December 10, Imperva disclosed that they had observed upwards of 1.4 million attacks targeting CVE-2021-44228 with peaks reaching roughly 280,000 attacks per hour. Cloudflare also reported to have blocked a peak of 20,000 exploit requests per minute with around 200-400 IPs appearing to be actively scanning at any given time.

Major vendors like Cisco, VMware, SonicWall, Okta and RedHat are investigating which of their products are affected. LogPoint advises administrators to regularly check the advisories as they are being updated by the respective vendors. Meanwhile, administrators can use Canary tokens to test for the presence of the Log4Shell vulnerability in their applications.

Microsoft has observed attackers dropping Cobalt Strike beacons by exploiting Log4Shell. Security researcher Markus Neis tweeted that apart from coin miners, he is seeing Muhstik and Mirai being dropped as payloads by Log4Shell exploitation. Administrators can look at the base64 payloads released by GreyNoise for reference to know what in-the-wild Log4Shell exploitation looks like.

Initial assessments show LogPoint products are not affected by the vulnerability. We will update the blog accordingly as we proceed through our assessments.

Detecting Log4Shell exploitation

Administrators can use Florian Roth's sigma rule to detect generic exploitation attempts from web server logs.

Searching for generic exploitation attempts in web server logs

Adversaries are mostly using the user agent field for exploiting Log4Shell. However, administrators should note that the patterns of the vulnerability can be triggered by these patterns being present in any string that gets logged by log4j. The query should be adjusted accordingly. Also, there are numerous permutations to bypass the signature, which administrators should keep in mind when using the detection.

ET labs have released signatures for Log4Shell which administrators can deploy on their IDS/IPS.

```
norm_id   IN ["Snort", "SuricataIDS"]   message="*CVE-2021-44228*"
```

Several vendors are releasing IoCs regarding Log4Shell that administrators can use to aid in their detection.

```
(source_address IN LOG4SHELL_IPS OR destination_address   IN LOG4SHELL_IPS)
```

Similarly, we can do the same using NetLab's IoCs for Mirai and Muhstik.

```
(domain   IN ["nazi.uy", "log.exposedbotnets.ru"] OR query IN   ["nazi.uy", "log.exposedbotnets.ru"]

OR   url IN ["http://62.210.130.250/lh.sh",   "http://62.210.130.250:80/web/admin/x86_64",
"http://62.210.130.250:80

/web/admin/x86",   "http://62.210.130.250:80/web/admin/x86_g",
"http://45.130.229.168:9999/Exploit.class",

"http://18.228.7.109/.log/log",   "http://18.228.7.109/.log/pty1;",   "http://18.228.7.109/.log/pty2;",
"http://18.

228.7.109/.log/pty3;",   "http://18.228.7.109/.log/pty4;",   "http://18.228.7.109/.log/pty5;", "http://210.

141.105.67:80/wp-content/themes/twentythirteen/m8",   "http://159.89.182.117/wp-
content/themes/twentyseventeen

/ldm"])
```

Importance of detecting post-compromise activity

In the current threat landscape, it is not enough for enterprise defenders to only be reactive and rely upon threat intelligence and detections. Defenders should be proactive by hunting for any suspicious activity in their environment. Even if defenders fail to detect the initial exploitation, they still have a fair chance to detect the attackers through their post-compromise activities. For starters, administrators can look out for any use of threat actors' common tools, such as Cobalt Strike, Tor and PsExec, and if detected, proceed to determine the entire kill chain.

LogPoint has several blogs that detail how to detect common threat actor tools:

- Microsoft disclosed that some attackers deploy Cobalt Strike payloads after exploiting Log4Shell, so administrators should check that their Cobalt Strike detections are up to date.
- Threat actors use Tor for anonymity and to hide their network traffic, so it’s important to check for Tor use in your enterprise.
- Threat actors deploy coin miners for monetary gains, like the one reported by NetLab, so enterprises need to hunt for cryptomining.

Attackers love to set up backdoor SSH access to the system by adding their public key to the authorized_keys file. Administrators can use auditd to monito r changes in the authorized_keys file of their servers.

```
norm_id=Unix "process"=audit event_type=SYSCALL   command=bash key="ssh_key_monitor"
```

The Muhstik botnet can set up backdoors, and as mentioned previously, remains one of the few botnets found to have exploited Log4Shell.

Finally, we can look for any anomalous server activities, like the execution of unusual processes, such as curl and wget. Administrators should note that allowlisting may be required depending upon the environment.

```
norm_id=Unix "process"=audit   event_type=PROCTITLE command IN ["*curl*", "*wget*",   "*chmod 
777*", "*chmod +x*"]
```

We cannot overemphasize the value of a properly implemented defense-in-depth approach, which can help detect threats that have penetrated the enterprise and where the initial detection had failed to trigger.

Defense-in-depth is key for enterprise security

It is important to note that the Log4Shell vulnerability is not as straightforward to exploit as it is to check for its presence as the successful exploitation depends upon several factors like JVM version and configuration used. Nevertheless, enterprises using vulnerable applications should assume they are breached and should readily scan the application logs for any compromise artifacts. Administrators should lookout for any suspicious outbound network traffic from their vulnerable applications.

Defense-in-depth remains the best possible strategy for detecting Log4Shell exploitation. Mass scanning activity has already commenced with coin miners and botnets already joining the party. It is only a matter of time for ransomware affiliates to join the bandwagon. Enterprise defenders should remain vigilant and should not rely on a single detection method for detecting exploitation of this critical vulnerability. Lastly, we stress to remind administrators to frequently check vendors' advisories for updates on mitigation and patch status of vulnerable products that are deployed in their enterprise.

***

Note to editors:
The blogpost, graphics and footage can be used free of charge by the media on the condition, that the byline is retained.

About Logpoint

Logpoint safeguards society in a digital world by helping customers and Managed Security Service Providers (MSSPs) detect cyberattacks. Combining reliable technology with a deep understanding of cybersecurity challenges, Logpoint makes security operations easier, giving organizations the freedom to progress. Logpoint's SIEMand NDRtechnologies improve visibility and give a multi-layered approach to cybersecurity that helps customers and MSSPs in Europe navigate the complex threat landscape. Headquartered in Copenhagen, Denmark, Logpoint has a European foundation and is the only European SIEM vendor with a Common Criteria EAL3+ certification. This demonstrates Logpoint's strong focus on data protection and cybersecurity regulations. For more information, visit http://logpoint.com.

Contacts

Maimouna Corr Fonsbøl
Press Contact
Head of PR
PR & Communications
mcf@logpoint.com
+45 25 66 82 98