# Log4Shell: Sicherheitslücke in log4j für Java hält die IT-Welt in Atem

Jan-Frederik Timm 13.12.2021 16:04 Uhr

Bild: Björn Ruytenberg

Eine gravierende Sicherheitslücke in log4j Version 2.x hält seit dem Wochenende die IT-Welt in Atem und das nicht ohne Grund: log4j ist heute der De-facto-Standard zum Loggen von Anwendungsmeldungen (z.B. Fehlermeldungen) in Java, die Liste der betroffenen Apps ist entsprechend groß. Sicherheitsbehörden weltweit sind alarmiert.

## log4j ist omnipräsent

Betroffen sind Systeme mit log4j zwischen Version 2.0-beta9 und 2.14.1 in der Java Virtual Machine, die es dem Angreifer ermöglichen, die Exploit-Zeichenfolge zu übermitteln (z.B. via HTTP, TCP etc.), die daraufhin über das Logging-Framework protokolliert und über JNDI interpretiert wird.

Angreifer könnten über die Schwachstelle auf dem Zielsystem eigenen Programmcode ausführen und so den Server kompromittieren.

Auf GitHub existiert eine Liste betroffener Anwendungen, die inzwischen über 100 Einträge zählt. Ob ein System betroffen ist, lässt sich beispielsweise über ein auf GitHub veröffentlichtes Python-Script für Rechner mit HTTP-Server testen.

## Die Lücke ist leicht zugänglich

Neben der großen Verbreitung lassen zwei weitere Aspekte die Alarmglocken schrillen: Die Lücke wird bereits großflächig ohne Aufwand ausgenutzt und kurzfristig schließen lassen wird sie sich nicht. Prinzipiell wurde die Lücke bereits in Version 2.15.0 geschlossen, doch bis dieses Update in Anwendungen einfließt, kann viel Zeit vergehen. „*Das Patchmanagement von Java-*

*Anwendungen ist nicht trivial, sodass bis zu einer Update-Möglichkeit die kurzfristigen Mitigationen empfohlen werden*“, erklärt beispielsweise das Bundesamt für Sicherheit in der Informationstechnik (BSI).

Auch nicht direkt mit dem Internet verbundene Systeme sind nicht vor einer Attacke gefeit, weil schon eine Anfrage über einen kompromittierten Rechner ausreicht, um ein weiteres System zu infiltrieren – es bedarf keines Nachladens von Code über das Internet. Microsoft stellt als Fazit einer eigenen Analyse fest: Jedes System mit betroffenem log4j muss bereits heute als angegriffen gelten.

## Patches werden Zeit brauchen

Vorerst müssen ohne Software-Patches also andere Maßnahmen ergriffen werden, um sich gegen Angreifer zu schützen, doch das ist gar nicht so einfach. Das BSI stellt online (PDF) neben detaillierten Informationen zu den Hintergründen auch ständig aktualisierte Hinweise zur Einrichtung von Schutzmaßnahmen zur Verfügung. Wie kritisch die Lage ist, wird beim Blick auf die vorgeschlagenen Gegenmaßnahmen deutlich: Im Zweifel sollten nicht benötigte Systeme erst einmal heruntergefahren werden.

Auf Systemen mit Java-Anwendungen, die weiter laufen müssen, sollte die Option „log4j2.formatMsgNoLookups“ auf „true“ gesetzt werden, indem die Java Virtual Machine

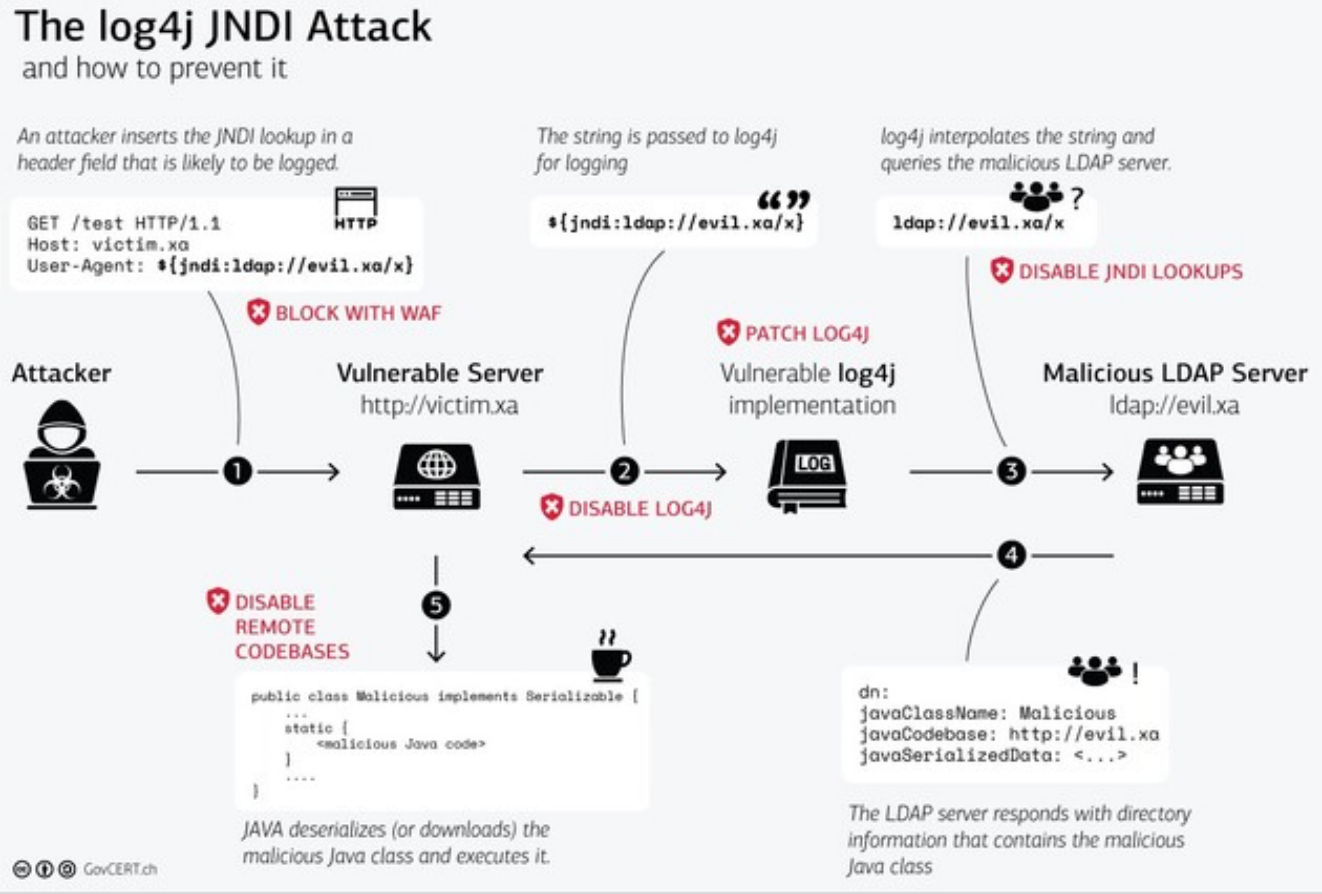
mit dem Argument „–Dlog4j2.formatMsgNoLookups=True" gestartet wird. Das funktioniert allerdings nur, wenn die Anwendung nicht davon Gebrauch macht.

Alternativ schlägt das BSI folgende Maßnahmen vor:

- In Web-Application-Firewalls (WAF), Intrusion Prevention Systemen (IPS) oder Reverse Proxies Verbindungen, die Angriffsmuster aufweisen, direkt ohne Weitergabe an die Fachapplikation abweisen oder nicht zwingend benötigte HTTP-Header auf statische Werte setzen.
- Blockieren aller nicht zwingend notwendigen, ausgehenden Verbindungen.
- Umfassendes Logging und die Protokollierung aller eingehender und ausgehender Verbindungen, um im Nachgang eine Kompromittierung leichter feststellen zu können.
- Anomaliedetektion auf dem Host betreiben.
- Prüfen, mit welchen Rechten der betroffene Dienst betrieben wird und diese auf dasm notwendige Minimum reduzieren.
- Verbindungen zu anderen Systemen sollten getrennt werden.
- Systeme, auf denen log4j bereits ab Version 2.15.0 vorliegt, sind ebenfalls nicht direkt als sicher zu klassifizieren. Für nach Bekanntwerden der Schwachstelle gepachte Systeme muss zusätzlich untersucht werden, ob diese bereits kompromittiert

wurden. Dies betrifft auch Systeme, die nicht direkt mit dem Internet verbunden sind, da diese über verbundene Systeme kompromittiert worden sein könnten.

Veröffentlicht wurde die Sicherheitslücke am vergangenen Donnerstag inklusive Proof of Concept (PoC) über GitHub. Kurz darauf wurde die Lücke unter der Kennung CVE-2021-44228 anerkannt. Über die Lücke lässt sich auf dem attackierten System beliebiger Code ausführen, erste Angriffe hätten die Installation von Crypto-Minern oder die Aufnahme der Rechner in Bot-Netze zum Zweck gehabt.




Log4Shell-Gegenmaßnahmen bis zum Patch (Bild: GovCERT.ch)

> *Neben erfolgreichen Kompromittierungen mit Kryptominern gibt es unter [3602021] erste Hinweise darauf, dass die Schwachstelle auch von Botnetzen*

> *ausgenutzt wird. Die Wahrscheinlichkeit ist groß, dass die mit dieser Schwachstelle in Verbindung stehenden Angreiferaktivitäten in den nächsten Tagen deutlich zunehmen werden.*
>
> *Das BSI zu Log4Shell*

## Schon länger ausgenutzt als öffentlich bekannt

CloudFlare CEO Matthew Prince zufolge gab es auf den eigenen Systemen allerdings schon neun Tage vor Bekanntwerden der Lücke erste Attacken über den log4j-Angriffsvektor.

## Auswirkungen auf ComputerBase

Auf den ComputerBase-Servern kommen Java und log4j ausschließlich für das Such-Backend Elasticsearch zum Einsatz, welches sowohl das CMS als auch die Forumsoftware XenForo nutzen. Die empfohlenen Maßnahmen gegen die log4j-Sicherheitslücke haben wir am frühen Samstagmorgen umgesetzt. Es gibt keine Anzeichen dafür, dass die Sicherheitslücke auf ComputerBase ausgenutzt wurde. Wie es scheint, ist unser Setup ohnehin nicht verwundbar, weil Nutzer und somit auch ein potenzieller Angreifer auf Elasticsearch nicht direkt zugreifen können, sondern dies nur indirekt über das CMS oder XenForo geschieht (Ankündigung von XenForo).