TLP:WHITE

Bundesamt
für Sicherheit in der
Informationstechnik

Kritische Schwachstelle in Log4j

CVE-2021-44228

Arbeitspapier Detektion und Reaktion

Version 1.2, Stand 16.12.2021

Änderungshistorie

Version	Datum	Name	Beschreibung
1.0	14.12.2021	Bundesamt für Sicherheit in der Informationstechnik BSI – CERT-Bund	Initiale Veröffentlichung
1.1	15.12.2021	Bundesamt für Sicherheit in der Informationstechnik BSI – CERT-Bund	CVE-2021-45046, Ransomware-Vorfälle, Hinweis zum Entfernen der Klassen, Redaktionelle Änderungen
1.2	16.12.2021	Bundesamt für Sicherheit in der Informationstechnik BSI – CERT-Bund	Aufnahme von 1.12.2 für Java 7

ⓘ Hinweis

Auf Grund der fortlaufenden Entwicklungen und Erkenntnisse des BSI wird dieses Dokument laufend aktualisiert und angepasst. Bitte achten Sie darauf, immer die aktuellste Version zu nutzen.

Bundesamt für Sicherheit in der Informationstechnik
Postfach 20 03 63
53133 Bonn
Internet: https://www.bsi.bund.de
Service-Center (Telefon): 0800 2741000
Service-Center (E-Mail): service-center@bsi.bund.de
Einen Vorfall melden: https://www.allianz-fuer-cybersicherheit.de/Webs/ACS/DE/IT-Sicherheitsvorfall/Unternehmen/Online-Meldung/online-meldung_node.html
Für die Zielgruppen und Partner des BSI gelten darüber hinaus die üblichen Meldewege.

Inhalt

1 Über dieses Dokument

Dieses Dokument soll die Cybersicherheitswarnung des BSI zu der „Log4Shell“ genannten Schwachstelle CVE-2021-44228 [1] in der weit verbreiteten Bibliothek Log4j um detailliertere Informationen zur Schwachstelle ergänzen und mögliche Mitigationsmaßnahmen und Detektionsmöglichkeiten konsolidieren und differenzieren. Hierbei sollen ebenfalls solche Informationen bereitgestellt werden, die im Gesamtkontext des Themenkomplexes Relevanz haben.

Es ist kurzfristig im Rahmen der akuten Lagebewältigung erstellt worden. Es fasst die aktuellen Erkenntnisse in einer sich dynamisch entwickelnden Lage zusammen. Sollten Sie fachliche Fehler identifizieren, melden Sie sich über die genannten Kanäle.

Die besondere Kritikalität des Themenkomplexes ergibt sich aus der generellen Ausnutzbarkeit, die auch für Folgeaktivitäten wie das Ausrollen von Ransomware oder die Kompromittierung von nachgelagerten Systemen und Produkten genutzt werden kann.

2 Beschreibung der Schwachstelle

Log4j ist eine beliebte Protokollierungsbibliothek für Java-Anwendungen. Sie dient der performanten Aggregation von Protokolldaten einer Anwendung. Die Bibliothek ist in vielen Softwareprodukten enthalten. Java-Software verwendet üblicherweise nicht die vom Betriebssystem bereitgestellten Systembibliotheken, sondern liefert die notwendigen Bibliotheken selbst mit, sodass verwundbare Versionen der Bibliothek auch auf dem System vorhanden sein können, ohne dass Log4j durch die Administrierenden selbst explizit installiert wurde.

2.1 Betroffene Versionen

In den Versionen 2.0-beta9 bis einschließlich 2.14.1 besteht eine Schwachstelle [2], die es ggf. auch nicht authentifizierten entfernten Angreifenden je nach konkretem Einsatzszenario ermöglichen kann, Daten zu exfiltrieren oder eigenen Code zur Ausführung zu bringen. Die Schwachstelle wurde daher mit dem höchstmöglichen CVSS-Score von 10.0 bewertet. Zusätzlich wurde zwischenzeitlich die Schwachstelle CVE-2021-45056 [3] mit dem CVSS-Score von 3,7 identifiziert, die einen Denial-of-Service-Angriff ermöglicht und in der Version 2.16.0 behoben wurde [2].

Hinweis: Die Log4j Version 2.12.2 ist nicht von der Schwachstelle betroffen. Diese Version dient der Kompatibilität mit Java Version 7. Dennoch befindet sie sich in der oben genannten Spanne der Versionsnummern.

Auch in den Versionen 1.x ist die Bibliothek grundsätzlich verwundbar. In diesen Fällen ist **diese** Verwundbarkeit Berichten zufolge jedoch nur über eine besondere Programmkonfiguration ausnutzbar [4], sodass eine Ausnutzung weit weniger wahrscheinlich erscheint. Für diese vom Entwickler nicht mehr unterstützte Version existieren allerdings weitere Schwachstellen, die diese Versionen betreffen [4], sodass hier auch unabhängig von dieser Schwachstelle ein Update empfohlen wird.

2.2 Betroffene Produkte

Durch die Vielzahl der Produkte, die Log4j einbinden und deren Konfigurationsoptionen ist eine vollständige Liste der betroffenen Produkte nicht erstellbar. Eine Liste von potenziell gefährdeten Produkten [5] wird gepflegt, erhebt aber keinen Anspruch auf Vollständigkeit. Die hohe Verbreitung von Java zum Beispiel auch im Privatanwender-, Netzwerkinfrastruktur- und Smartphonebereich legt nahe, dass neben Servern potenziell sehr viele weitere Systeme betroffen sein können, insbesondere auch wenn es sich um selbst entwickelte Anwendungen von Unternehmen handelt. Dies zeigt auch eine weitere Sammlung betroffener Produkte [6].

Aufgrund der derzeit nicht überschaubaren Vielfalt der betroffenen Anwendungen kann diese Schwachstelle sowohl z. B. Webseiten und automatisierte Schnittstellen als auch Client-Anwendungen betreffen. Darüber hinaus können auch Anwendungen betroffen sein, die in eingebetteten Systemen und Komponenten zur industriellen Steuerung (z.B. Siemens [7], Schneider Electric [8]) eingesetzt werden.

2.3 Gefahren

Die folgenden Gefahren bestehen, wenn Log4j verwendet wird, um eine vom Angreifenden kontrollierte Zeichenkette wie beispielsweise – im Falle einer Webanwendung – den HTTP User Agent zu protokollieren. Sollten die betroffenen IT-Systeme Verbindungen ins Internet aufbauen und nicht nur entgegennehmen können, kann über die Schwachstelle schädlicher Programmcode nachgeladen werden. Dafür können auch Protokolle verwendet werden, die üblicherweise nur innerhalb von IT-Netzwerken zum Einsatz kommen,

wie beispielsweise LDAP. Unabhängig von der Möglichkeit Schadcode nachzuladen, können durch DNS-Anfragen trotzdem Informationen wie beispielsweise Umgebungsvariablen zum Angreifenden gelangen. In diesen Umgebungsvariablen werden je nach Anwendungsfall auch Benutzerkennungen und Authentifikationsmerkmale gespeichert, sodass ein Abfluss der Daten mit der Kompromittierung dieser Informationen gleichzusetzen ist. Auch könnten Angreifende gegebenenfalls sensible Informationen in die Protokolldaten ausleiten. Bisher konnte nicht gesichert nachgewiesen werden, dass eine Ausführung von Schadcode auch mit untersagten Kommunikationsbeziehungen möglich ist, allerdings war dies bei ähnlich gelagerten Schwachstellen in der Vergangenheit unter bestimmten Umständen der Fall.

Durch den im Internet frei verfügbaren Quelltext einer beispielhaften Ausnutzung der Schwachstelle („Proof of Concept“) [9], können Angreifenden mit sehr geringem Aufwand verwundbare Systeme auffinden und die Schwachstelle anschließend ausnutzen.
Solche automatisierten Ausnutzungen konnten bereits durch das BSI beobachtet werden. Ebenfalls wurden bereits erfolgreiche Kompromittierungen mit z. B. Kryptominern und Ransomware [10] bestätigt; eine Rekrutierung in Botnetze scheint ebenfalls wahrscheinlich. Neben diesen Infektionen wurde auch schon über den Einsatz von so genannten Post-Exploitation-Frameworks wie Cobalt Strike berichtet. Solche Anwendungen werden von Angreifenden genutzt, um weitere Aktionen auf dem angegriffenen System auszuführen. Dies kann zum Beispiel das Verschleiern des Angriffs, das Nachladen eines Root-Kits oder von Ransomware beinhalten.

Es können auch Systeme verwundbar sein, die nicht explizit aus dem Internet erreichbar sind, aber z. B. Daten von im Internet exponierten Diensten verarbeiten. Gibt beispielsweise eine Internetseite Suchanfragen an einen Java-basierten internen Suchindexdienst weiter, dann ist dieser ebenfalls gefährdet. Eine ähnliche Problematik könnten Spamfilter oder Virenscanner, die Java-basiert sind, aufzeigen. Das sind lediglich zwei von zahlreichen möglichen Konstellationen. Eine weitere Möglichkeit zur Infektion von nicht aus dem Internet erreichbaren Systemen bietet die grundsätzliche Wurmfähigkeit der Schwachstelle.

3 Mögliche Mitigationen

Die Schwachstelle kann nur durch ein Update auf die aktuellste Version von Log4j abschließend behoben werden. Das Update 2.15.0 setzt die unten angegebene Konfigurationsoption als neuen Standardwert, die Version 2.16.0 setzt weitere Härtungsmaßnahmen zum Schutz von Anwendungen um [11].

Dieses Update können i.d.R. nur die Softwarehersteller vornehmen, die die Bibliothek in ihre Programme eingebunden haben. Dafür müssen die Updates des jeweiligen Programms von den Administratoren installiert werden. Das alleinige Aktualisieren der Bibliothek über die Softwareverwaltung von Betriebssystemen, wie das zum Beispiel in vielen Linux-Distributionen möglich ist, reicht zum Schließen der Schwachstelle nicht aus. Ebenfalls ist – entgegen anderslautender Berichte – ein Update von Java selbst kein Mittel zur Mitigation [12].

Einige Hersteller haben bereits Workarounds oder Sicherheitsupdates für ihre Produkte zur Verfügung gestellt. Das BSI empfiehlt daher die Herstellerseiten und von Herstellern bereitgestellten Security Advisories zu verfolgen. Sobald weitere Sicherheitsupdates veröffentlicht werden, sollten diese unverzüglich installiert werden [13].

Achtung: Es ist möglich, dass die Bibliothek in Anwendungen auf eine Weise verwendet wird, die die folgende Konfigurationsanpassung unwirksam macht [14].

Anwender von betroffener Software können bis zu einer Update-Möglichkeit ab Version 2.10.0 die JVM-Command Line Option

-Dlog4j2.formatMsgNoLookups=True

oder die entsprechende Umgebungsvariable

LOG4J_FORMAT_MSG_NO_LOOKUPS=True

verwenden, um die problematische Funktionalität manuell zu deaktivieren. Da der Start von Java-Anwendungen nicht unbedingt immer trivial implementiert ist, ist es in jedem Fall zu empfehlen, die vorgenommenen Einstellungen für jeden Dienst einzeln auf Wirksamkeit zu überprüfen.

Achtung: Durch alle zurzeit bekannten Mitigationen, die die Nutzung der Bibliothek selbst betreffen, wird die problematische Funktionalität deaktiviert. Sollte eine Anwendung die Funktionalität für den Betrieb benötigen, könnte sie (ggf. teilweise) nicht mehr funktionsfähig sein.

Sollte dies nicht möglich sein, kann das Risiko kurzfristig auch durch Deaktivieren der Systeme reduziert werden, wenn diese Systeme verzichtbar sind. Einige Dienstleister empfehlen ihren Kunden stattdessen einen Hotpatch [15].

Außerdem besteht die Möglichkeit, die problematische Klasse manuell zu entfernen [2].

Achtung: Auch diese Maßnahme kann die Funktionsfähigkeit und somit auch die Verfügbarkeit der betreffenden Applikation einschränken, wenn die Funktionalität tatsächlich benötigt wird. Es wird empfohlen, Backups der entsprechenden Archive zu erstellen, bevor diese Maßnahme umgesetzt wird.

Hinweis: Diese Maßnahme geht nicht rekursiv vor. Die Klasse kann innerhalb der Applikationsarchive erneut innerhalb eines Applikationsarchives weiterhin enthalten sein.

Unabhängig von der Anpassung von spezifischen Softwarekonfigurationen für diesen spezifischen Fall, sollten Systeme grundsätzlich nur solche Verbindungen (insbesondere in das Internet) aufbauen dürfen, die für den Einsatzzweck zwingend notwendig sind [16]. Andere Zugriffe sollten durch entsprechende Kontrollinstanzen wie Paketfilter und Application Layer Gateways unterbunden werden [16]. Zudem bietet das Unterbinden von DNS-Anfragen auf Domains, die nicht zwingend für den Regelbetrieb notwendig sind, einen Schutz vor der Exfiltration von Informationen über das DNS-Protokoll. Hierfür kann zum Beispiel ein interner DNS-Server verwendet werden, der solche Anfragen nicht weiterleitet.

4 Detektion

Vorbemerkung: Da bei festgestellten Auffälligkeiten tiefergehende forensische Analysen notwendig werden können, empfiehlt es sich, bereits vor der Überprüfung auf Indikatoren Datensicherungen (vorzugsweise durch das Erstellen von Snapshots) der zu untersuchenden Systeme durchzuführen.

Sollte eine Betroffenheit festgestellt werden, ist es zu empfehlen, die Systeme im Falle von virtuellen Systemen in den Standby zu versetzen, damit von den Prozessen im Arbeitsspeicher enthaltene Informationen für eine mögliche forensische Analyse erhalten bleiben.

Sollte ein Update oder eine Evaluierung der Betroffenheit zurzeit nicht möglich sein, empfiehlt es sich, ausgehende Verbindungsversuche aller Systeme zu protokollieren und auf Unregelmäßigkeiten hin zu untersuchen. Hierbei sind vor allem interessant:

- DNS-Anfragen für unübliche Domains, deren Subdomains Konfigurationsdetails einer Anwendung (wie z. B. Benutzernamen) enthalten.
- LDAP(S)-Verbindungsversuche ins Internet.
- Persistente Verbindungen ins Internet.

Ebenfalls können eingehende Anfragen bei Webservern durch einen Reverse-Proxy detailliert protokolliert und auf Indikatoren [17] hin untersucht werden. Es können auch weitere Detektionsmaßnahmen [18] der Community verwendet werden. Diese Maßnahme ist aufgrund der Vielzahl der möglichen Ausnutzungswege jedoch keinesfalls zur Mitigation geeignet, sondern ergänzt lediglich Detektionsmöglichkeiten.

Die folgende Tabelle stellt mögliche Maßnahmen zur Detektion gegenüber.

Maßnahme	Voraussetzungen	Nachteile
Einfache Anfragenauswertung mit Regelwerken	Web Server Logs und einfache Regelwerke	Erkennt nur Angriffe auf in den Server Logs erfasste Teile der Anfrage
Detaillierte Anfragenuntersuchung mit Regelwerken	Zugriff auf die vollständigen Anfragedaten, Regelwerke und Forensik-Kenntnisse	Ressourcenintensiv und möglicherweise unentdeckte Anfragen durch Umgehung des Regelwerks
Anomalieerkennung auf Netzwerkebene	Netzwerktraffic- und Firewall-Log-Daten	Ressourcenintensiv und nicht trivial einzurichten.
Einfache Prozesslistensuche	Zugriff auf die Prozessliste	Nur in trivialen, positiven Fällen der Kompromittierung konklusiv
Ressourcenverbrauchsanalyse	Monitoring	Nur in trivialen, positiven Fällen der Kompromittierung konklusiv
Forensische Speicherabbilduntersuchung	Speicherabbild	Personalintensiv und je nach Infektionsgrad nicht konklusiv

Weiterhin können betreffende Systeme im Monitoring durch erhöhten Ressourcenverbrauch auffällig werden.

5 Weiterführende Informationen

Weiterführende Informationen zu dem Themenkomplex können für die folgenden Bereiche aus den entsprechenden Quellen entnommen werden.

5.1 Schwachstellenbeschreibung

- Schwachstellenmitteilung des Herstellers [2]
- Erster Bericht über die Schwachstelle [19]
- Eintrag der Schwachstelle in der CVD [1]
- Ausführlicher Bericht eines Sicherheitsdienstleisters [20]

5.2 Ausnutzung/Payloads

- Twitter-Thread zu möglichen Payloads und Ausnutzungen [21]

5.3 Mitigation

- Sammlung weiterer Mitigationsmaßnahmen eines Sicherheitsdienstleisters [22]
- Weitere Sammlung des NCSC-NL [23]

5.4 Detektion

- Github-Repository eines lokalen Scanners für die Schwachstelle [24]
- Analyse von Microsoft [25]
- Sammlung von öffentlich beobachteten Samples [26]

6 Literaturverzeichnis

[1] Mitre, 2021. [Online]. Available: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-44228.

[2] Apache, 2021. [Online]. Available: https://logging.apache.org/log4j/2.x/security.html.

[3] Mitre, 2021. [Online]. Available: https://nvd.nist.gov/vuln/detail/CVE-2021-45046.

[4] remkop, 2021. [Online]. Available: https://github.com/apache/logging-log4j2/pull/608#issuecomment-990494126.

[5] NCSC-NL, 2021. [Online]. Available: https://github.com/NCSC-NL/log4shell/tree/main/software.

[6] SwitHak, 2021. [Online]. Available: https://gist.github.com/SwitHak/b66db3a06c2955a9cb71a8718970c592.

[7] Siemens, 2021. [Online]. Available: https://cert-portal.siemens.com/productcert/pdf/ssa-661247.pdf.

[8] Schneider Electric, 2021. [Online]. Available: https://download.schneider-electric.com/files?p_Doc_Ref=SESB-2021-347-01.

[9] tangxiaofeng7, 2021. [Online]. Available: https://github.com/tangxiaofeng7/apache-log4j-poc.

[10] Cado Security, 2021. [Online]. Available: https://www.cadosecurity.com/analysis-of-novel-khonsari-ransomware-deployed-by-the-log4shell-vulnerability/.

[11] remkop, 2021. [Online]. Available: https://github.com/apache/logging-log4j2/pull/623#discussion_r767411342.

[12] G. Linares, 2021. [Online]. Available: https://twitter.com/Laughing_Mantis/status/1470412026119798786.

[13] BSI, 2021. [Online]. Available: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Grundschutz/Kompendium_Einzel_PDFs_2021/04_OPS_Betrieb/OPS_1_1_3_Patch_und_Aenderungsmanagement_Edition_2021.html.

[14] Elastico, 2021. [Online]. Available: https://discuss.elastic.co/t/apache-log4j2-remote-code-execution-rce-vulnerability-cve-2021-44228-esa-2021-31/291476.

[15] Amazon, 2021. [Online]. Available: https://aws.amazon.com/de/blogs/opensource/hotpatch-for-apache-log4j/.

[16] BSI, 2021. [Online]. Available: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Grundschutz/Kompendium_Einzel_PDFs_2021/09_NET_Netze_und_Kommunikation/NET_3_2_Firewall_Edition_2021.html.

[17] Curated Intel, 2021. [Online]. Available: https://github.com/curated-intel/Log4Shell-IOCs.

[18] Neo23x0, 2021. [Online]. Available: https://gist.github.com/Neo23x0/e4c8b03ff8cdf1fa63b7d15db6e3860b.

[19] Luna Sec, 2021. [Online]. Available: https://www.lunasec.io/docs/blog/log4j-zero-day/.

[20] Cloudstrike, 2021. [Online]. Available: https://www.crowdstrike.com/blog/log4j2-vulnerability-analysis-and-mitigation-recommendations/.

[21] G. Linares, 2021. [Online]. Available: https://twitter.com/Laughing_Mantis/status/1470378593347772420.

[22] Tech Solvency, 2021. [Online]. Available: https://www.techsolvency.com/story-so-far/cve-2021-44228-log4j-log4shell/#remediation.

[23] NCSC-NL, 2021. [Online]. Available: https://github.com/NCSC-NL/log4shell.

[24] hillu, 2021. [Online]. Available: https://github.com/hillu/local-log4j-vuln-scanner.

[25] Microsoft, 2021. [Online]. Available: https://www.microsoft.com/security/blog/2021/12/11/guidance-for-preventing-detecting-and-hunting-for-cve-2021-44228-log4j-2-exploitation/.

[26] Netlab 360, 2021. [Online]. Available: https://blog.netlab.360.com/ten-families-of-malicious-samples-are-spreading-using-the-log4j2-vulnerability-now/.

7 Abkürzungsverzeichnis

BSI	Bundesamt für Sicherheit in der Informationstechnik
CVSS	Common Vulnerability Scoring System
DNS	Domain Name System
IoC	Indicator of Compromise
JVM	Java Virtual Machine
LDAP	Lightweight Directory Access Protocol
PoC	Proof of Concept