Log4j – Aktueller Stand unserer Analyse

Wir überprüfen derzeit sowohl die von uns genutzten als auch die von uns entwickelten Tools umfassend auf eine mögliche Bedrohung durch die log4j Sicherheitslücke.

Unsere Serverapplikationen ilabServer, ilabWeb (inkl. App) sowie Mein Laborergebnis / Mein Testergebnis sind nicht betroffen. Auch in älteren Versionen wurde log4j nicht eingesetzt. Wir überprüfen dennoch alle installierten Pakete auf den von uns gehosteten Systemen und empfehlen dies auch für externe Installationen. Ebenfalls nicht betroffen ist unsere Neuentwicklung ilabOrder.

Die älteren iLab2000-Server sind grundsätzlich nicht betroffen.

Wir können aber nicht absolut sicher ausschließen, dass irgendwelche Software von Dritten auf diesen Systemen installiert wurde, die kompromittiert ist.

Die Systeme, die bei uns gehostet sind, haben wir bereits weitgehend überprüft.

Der ilabClient enthält ab Version 3.0 bis 3.2.x gefährdete Versionen der log4j Bibliothek.

Wir haben am Montag die Version 3.3.0 zum Download auf unserem Server bereitgestellt. Diese Version ist nicht von dieser Gefährdung betroffen. Sie beinhaltet die korrigierte Version 2.15.0 von log4j. Am Dienstag, 14.12.2021, haben wir diese Version durch die Version 3.3.1 ersetzt, die jetzt die Version 2.16.0 enthält.

Bereits am Wochenende haben wir mit der Gefährdungsanalyse begonnen und können sagen, dass es uns (trotz Insider-Wissen bzgl. unserer Software) nicht gelungen ist, die Schwachstelle in den ausgelieferten Versionen auszunutzen.

Allerdings können wir nicht absolut ausschließen, dass ein „Profi-Hacker“ mit einer ausgeklügelten „Man in the Middle“-Attacke mehr „Erfolg“ als wir hat.

Wir haben für die weitergehende Analyse eine modifizierte Version unserer Software gebaut, um die Schwachstelle noch tiefer untersuchen zu können. Auch bei diesen Tests haben wir keine Hinweise gefunden, wie die Schwachstelle direkt ausgenutzt werden könnte.

Grundsätzlich müsste ein Angreifer unserer Software ein Objekt so übergeben, dass dieses direkt mit Log4J geloggt würde. Das machen wir aber an keiner Stelle. Wir protokollieren stets Texte, die wir selbst innerhalb der Software erzeugen. Es gibt nur 1 Programmstelle, wo wir einen Text unverändert protokollieren. Hier ist ein Angriff aber nahezu ausgeschlossen.

Wir stufen die Gefährdung durch den ilabClient als „sehr gering“ ein, empfehlen aber aus grundsätzlichen Überlegungen das Update auf Version 3.3.1 zeitnah durchzuführen.

In den installierten Versionen kann die Schwachstelle auch kurzfristig entsprechend der Empfehlungen des log4j-Entwicklerteams (https://logging.apache.org/log4j/2.x/) deaktiviert werden:

- ilabClient Versionen bis einschließlich 3.1.3 enthalten log4j in einer Version < 2.10.0.
    - In diesen Versionen muss die Klasse JndiLookup.class aus der ilabclient.jar entfernt werden:
        - ilabClient schließen
        - llabclient.jar (i. d. R. C:\ilabClient\ilabclient.jar) z. B. mit 7zip öffnen
        - in das Unterverzeichnis \org\apache\logging\log4j\core\lookup\ navigieren
        - JndiLookup.class löschen
        - ilabClient neu starten

- ilabClient Versionen ab 3.1.4. enthalten log4j in einer Version >= 2.10.0 und < 2.15.0
    - Alternativ zu dem obigen Workaround, kann in diesen Versionen auch eine Umgebungsvariable verwendet werden:
        - ilabClient schließen
        - Systemumgebungsvariable mit Namen LOG4J_FORMAT_MSG_NO_LOOKUPS und Wert true hinzufügen
        - ilabClient neu starten

Bei dem von uns genutzten Zammad Ticketsystem haben wir eine potenzielle Gefährdung bereits entsprechend der Vorgaben der Entwickler umgangen. Sobald hier ein Update bereitsteht, werden wir dieses installieren. In anderen bei uns intern genutzten Tools konnten wir keine Probleme erkennen.
Sofern einzelne Anwender Prüfmodule der KBV installiert haben, um diese mit unserem LDT-Viewer zu nutzen, empfehlen wir diese umgehend zu löschen. Der LDT-Viewer funktioniert (abgesehen von der Prüfung durch das Prüfmodul) auch ohne diese Software.