Log4Shell Test Tool Analysis

Ashish Bastola (L)
Clemson University
Clemson, South Carolina, USA
abastol@g.clemson.edu

Siddheshwar Sanjay Munde
Clemson University
Clemson, South Carolina, USA
smunde@g.clemson.edu

Rajat Mittal
Clemson University
Clemson, South Carolina, USA
rajatm@g.clemson.edu

I. PROBLEM

Log4shell was a critical Remote Code Execution (RCE) vulnerability publicly disclosed on December 10th, 2021. It received a Common Vulnerability Scoring System (CVSS) rating of 10 [20], the highest and most dangerous rating possible on that scale, largely because it allowed RCE without user interaction. JNDI (Java Naming and Directory Interface) allows Java applications to use directory services like LDAP (Lightweight Directory access Protocol) or naming services like DNS. This particular lookup allows to trigger Remote Code Execution(RCE) Allow hackers to control java-based web servers and launch what are called 'remote code execution' (RCE) attacks The primary means of exploit was ensuring a specially-crafted string of characters be received by a system with outbound access to the Internet and then logged by that same system using an open-source Java library called log4j. Given its potential to be used in any Java application, organizations around the globe were scrambling to determine their exposure as well as identify methods to eliminate their exposure where possible and mitigate the risk elsewhere. This led to security teams needing tools to check for the vulnerability, assess fixes and mitigations, and demonstrate the vulnerability's impact in their environment. Both open-source and vendor communities were quick to deliver a wide variety of tools.

II. RELATED WORKS

A. Log4j

Log4j is a free and open-source library implementing a logging framework [3]. There are a number of reasons, security and otherwise, that a developer should implement logging. Cheng et al. [4] provided an overview of logging platforms, including log4j, as an alternative to Java's built-in logging system. The authors explained the benefits of using a single logging solution and reviewed the different capabilities a developer can use to track significant events generated by their application.

CHI'20, April 25–30, 2020, Honolulu, HI, USA

ISBN 978-1-4503-6708-0/20/04…$15.00
DOI: https://doi.org/10.1145/3313831.XXXXXXX

B. Log4shell

In 2013, a log4j user requested that a feature be added to log4j that allowed the use of Java Naming and Directory Interface (JNDI) lookups [5]. JNDI provides an interface to naming and directory services like Lightweight Directory Access Protocol (LDAP) or Remote Method Invocation (RMI). The requester wanted the feature so the logged application could use these services to look up items and produce better logs, rather than having to code each item individually. In December 2021 it was discovered that providing a specially crafted string to log4j would cause it to contact an external server and either run Java code specified by the server, provide data to the server, or deny service to the logging application [6]. Respectively, these vulnerabilities are generally classified as RCE, Information Disclosure, or Denial of Service (DoS).

C. Tools Description

The following tools were in scope for our research[21]:

- **CISA Log4j Scanner**[11]
  Cisa Log4j Scanner is a vulnerability scanning solution for log4j Remote Code Execution flaws (CVE-2021-44228 & CVE-2021-45046). The data and code in this repository are supplied "as is," having been compiled with the help of the open-source community and updated by CISA in collaboration with the larger cybersecurity community. False negatives may arise in some cases due to the jndi test cases presented used to trigger this attack being potentially insufficient specially when certain low level amendments or few temporary patches are designed with these payloads as reference; this is not intended to be a 100 percent true positive answer as there might be some other set of strings apart from the ones mentioned that can induce the RCE.
- **Burp Proxy Suite Log4Shell Everywhere extension**[17]
  This plugin only works on in-scope traffic and uses log4j exploits to inject headers into your proxy traffic. To eliminate false positives with pingbacks, such as DNS requests performed from host regex matching, and to potentially bypass filters, the $lower:x pattern is utilized in the DNS request to create confidence that some sort of jdni lookup is taking place - if this matches, the confidence is set to certain. Simply install it and go to the target website to use it. The results will be displayed in the 'Issues' tab. By changing /resources/injections, you may simply customize injected payloads.
- **Burp Proxy Suite Log4Shell Scanner extension**[22]

This plugin allows you to scan exclusively for Log4j (and not for other stuff like XSS or SQLi). If the scan configuration built as a result of following any of the instruction sets below is used, the scanner will only run Log4Shell checks on all insertion points. Because this plugin only provides payloads to the built-in active scanner, you'll need to configure your scan properly for best coverage vs. performance just like any other built-in or extension-provided scan.

- **Artic Wolf Log4Shell Deep Scan**[24]
  This program looks for Java applications on the system that contain the Log4J class JndiLookup.class, which is the cause of the Log4Shell vulnerabilities. If this class is identified in an application, the script searches for Log4J updates that show the program has been updated to utilize Log4J 2.16+ or Log4J 2.12.2+. The application is vulnerable if it contains JndiLookup.class but does not appear to have been updated. Each scan will have one of the following results:
  - **PASS**: all Java applications detected were scanned and no vulnerable applications were found.
  - **FAIL**: one or more vulnerable Java applications were found. See output and JSON for paths. If the script was unable to scan any applications, they will also be listed in the output and JSON.
  - **UNKNOWN**: no vulnerable Java applications were detected, but the script was not able to detect all scanned applications. See output and JSON for paths.
  - **ERROR**: the script encountered an error and was unable to complete. See output for details.
- **Bi.Zone Log4Shell Yara Rule**[10]
  We can use this YARA rule to detect the presence of Log4j and then determine whether it is vulnerable to Log4Shell (CVE-2021-44228) or not. If it is, then we can use mitigations listed below to handle this situation. In the Package folder we can find a collected package which includes YARA executable, the rule file (log4j.yar), and cmdscripts for running it on Windows and Linux systems.
- **Binary Defense Log4j Honeypot Flask**[9]
  Internal network honeypot for detecting if an attacker or insider threat scans your network for log4j CVE-2021-44228
  This can be installed on a workstation or server, either by running the Python app/app.py script directly (you'll need python3, Flask, and Requests) or as a Docker container.
  Important Note: This is a LOW-INTERACTION honeypot meant for internal active defense. It is not supposed to be vulnerable or let attackers get into anything.
  All it does is watch for suspicious string patterns in the requests (form fields and HTTP headers) and alert you if anything weird comes through by sending a message on Teams or Slack.
- **Crowdstrike Archive Scan Tool**[12]
  This tool is a fast scan that walks filesystems looking for vulnerable log4j versions.
  Currently, it examines a specified set of directories for JAR, WAR, ZIP, or EAR files, then searches for files that match a defined set of checksums.
- **Datto Log4Shell Enumeration, Mitigation, and Attack Detection Tool for Windows**[14]
  This is a PowerShell-based script that can be run on a Windows system (it has not been built for or tested on other platforms). It can be used to:
  - Set the LOG4J_FORMAT_MSG_NO_LOOKUPS environment variable to TRUE to protect the system from Log4Shell attacks with vulnerable Log4j versions.
    - Examine the system for any JAR files that include code that points to a vulnerable Log4j version.
      - This is not definitive and should only be used as a guide.
  - Using the YARA tool and Florian Roth's definitions[2], check all JAR, LOG and TXT files on the system for indicators of Log4Shell attacks The script was originally created as a component for the Datto RMM software[13] rmm; however, as part of Datto's continuous commitment to the MSP, it has been made available to the community for free.
- **Huntress Labs Log4shell Vulnerability Tester**[16]
  The program generates a random unique identifier that you may use to test input fields. If an input field or application is compromised, it will connect to this website using LDAP. Our LDAP server will quickly disconnect the connection and log it for a brief period of time. This utility will not execute any code on your systems.
- **Trend Micro Log4j Vulnerability Tester**[23]
  Proprietary tool provided by Trend Micro[18] which Quickly identify endpoints and server applications that may have Log4j.
- **Fox-IT Log4j finder**[1]
  A Python3 script that searches the disk for Log4j2 instances that are vulnerable to Log4Shell (CVE-2021-44228 & CVE-2021-45046 & CVE-2021-45105). It scans recursively on disk as well as within (nested) Java Archive files (JARs). See Figure **??** for reference.
  log4j-finder locates log4j2 libraries on your drive by comparing a list of known bad and known good MD5 hashes of certain files (currently only JndiManager.class) found in log4j2-core-* packages; the major package affected by log4shell. It looks for these files both within Java Archive files and on the disk. This file's MD5 hash is then used to determine the log4j2 version.
- **Google Log4jScanner**[5]
  A vulnerability filesystem scanner for log4j and a Go package for analyzing JAR files.
- **Fullhunt Log4j Scan**[3]
  A fully automated, precise, and comprehensive scanner for locating insecure log4j hosts.
  Features:
  - Support for lists of URLs.
  - Fuzzing for more than 60 HTTP request headers (rather than just 3-4 headers as in prior products).

- Fuzzing for HTTP POST Data parameters.
- Fuzzing for JSON data parameters.
- Supports DNS callback for vulnerability discovery and validation.
- WAF Bypass payloads.

- **Mergebase Log4j Detector**[7]
  Scanner for vulnerable Log4J versions to assist teams in determining their vulnerability to CVE-2021-44228 (CRITICAL), CVE-2021-45046, CVE-2021-45105, and CVE-2021-44832. Can look for Log4J instances by thoroughly inspecting the entire file system, including all installed applications. It can locate Log4J instances that are several levels deep. It runs on Linux, Windows, and Mac, as well as anyplace else Java is supported!

- **Logpresso CVE 2021-4428 Scanner**[6]
  This log4j2-scan is a single binary command-line program for screening and patching CVE-2021-44228 vulnerabilities. It also allows you to scan and fix nested JAR files. It also detects CVE-2021-45046 (log4j 2.15.0), CVE-2021-45105 (log4j 2.16.0), CVE-2021-44832 (log4j 2.17.0), CVE-2021-4104, CVE-2019-17571, CVE-2017-5645, CVE-2020-9488, CVE-2022-23302, CVE-2022-23305, CVE-2022-23307 (log4j 1.x), and CVE-2021-42550 (logback 0.9-1.2.7) vulnerabilities.
  Each scan will have one of the following results:
  - **_VULNERABLE_**: You need to upgrade or remove this file.
  - **_OKAY_**: We report this for Log4J versions 2.3.1, 2.12.2, 2.12.3, 2.15.0, 2.16.0, and 2.17.0. We recommend upgrading to 2.17.1.
  - **_SAFE_**: We currently only report this for Log4J versions 2.3.2, 2.12.4, and 2.17.1 (and greater).
  - **_OLD_**: You are safe from CVE-2021-44228, but should plan to upgrade because Log4J 1.2.x has been EOL for 7 years and has several known-vulnerabilities.
  - **_POTENTIALLY_SAFE_**: The "JndiLookup.class" file is not present, either because your version of Log4J is very old (pre 2.0-beta9), or because someone already removed this file. Make sure it was someone in your team or company that removed "JndiLookup.class" if that's the case, because attackers have been known to remove this file themselves to prevent additional competing attackers from gaining access to compromised systems.

- **Palantir Log4j Sniffer**[8]
  TThis log4j-sniffer explores a directory looking for all instances of log4j on disk. This tool can be used to see whether there are any vulnerable log4j instances in a directory tree and either report or delete them, depending on the mode.
  Based on the suffix, log4j-sniffer will scan a disk for all files of the following types:
  - Zips: .zip
  - Java archives: .jar, .war, .ear
  - Tar: .tar, .tar.gz, .tgz, .tar.bz2, .tbz2

D.Patch history

Version	Date	CVE	Description
2.15.0	12/06	2021-44228	Lookups within message text disabled by default.
2.16.0, 2.12.2	12/13	2021-45046	JNDI disabled by default. Support removed for message lookups.
2.17.0, 2.12.3, 2.3.1	12/17	2021-45105	Recursion in string substi- tution fixed. Limit JNDI to the Java protocol.
2.17.1, 2.12.4, 2.3.2	12/27	2021-44832	Fixed possible RCE via JDBC Appender when attacker controls server configuration.

III. METHODOLOGY

We created a taxonomy of scanners and evaluated them against multiple products to see how effective they were at their primary purpose, whether that was detecting log4shell, emulating a vulnerable system, or any other purpose. In total we took 6 vulnerable applications along with their ZIP files (few tools only supports .zip traverse) and tested them on MacOS, Linux and Windows Platform. We ran test on an aggregated folder of all the vulnerable application and analysed how much time each tool is taken and what is the output.

Vulnerable applications we used :

Vulnerable Application	Size
Ghidra	933MB
New-Relic	19.5MB
Log4shell samples mater zip file	397MB
Log4shell vulnerable app-main	274kb
Palantir log4j samples	64MB
Log4j finder(Fox-IT) samples	1MB

Tools we are considering for experiment:

Tool Name	Strengths	Weakness	Tool Type
Fullhunt Log4j Scan	WAF bypass	Performs only JNDI lookup	Active
Log4shell vulnerability Test Tool	Performs LDAP & DNS lookup		Active
Huntress Labs Log4shell vulnerability Tester	Returns time stamp with external IP addresses	Only JNDI lookup	Active
Arctic Wolf Log4shell Deep Scan	Scans through jar/war/ear archive files. Very fast.	Privileged access required	Passive
Fox-IT Log4j Finder	Provides detailed analysis	Time consuming	Passive
Mergebase Log4j Detector	Scans through hidden deep layers	Does not provide summary	Passive
Palantir Log4j Sniffer	Scans Zips, Java archives & Tar	Requires **go** installation	Passive
Datto Log4shell enumeration, mitigation, and Attack Detection Tool for Windows and Linux	Detection & mitigation both can be done	Difficult to install	Passive

We categorized above tools into two categories.

- **Active** : Tools such as Huntress, Log4shell vulnerability test tools generates a JNDI syntax or payload on thier own web server. Log4j2 has a logging option named "Message Lookup Substitution" by default. This feature allows specific special strings to be substituted by other dynamically produced strings during logging. For example, logging the string Running "$java:runtime" will result in something like:

  $$\text{Running Java version 1.7.0\_67}$$

  It was revealed that one of the lookup methods, notably the JNDI lookup in conjunction with the LDAP protocol, will download a given Java class from a remote source and deserialize it, executing some of the class's code in the process.

  This implies that if a remote attacker can control any part of a recorded string, the remote attacker achieves remote code execution on the program that logged the text.

  The most typical replacement string that exploits this flaw will look something like this:

  $$\$\{jndi:ldap://somedomain.com\}$$

  Attack tools mentioned in 3 generate similar payload which is used to trigger attack. This payload can be either inserted in header of a request or any input field of website.

- **Passive** : Another type we categorized is passive, passive tools are used for scanning JAR TAR WAR files. These tools may take ample amount of time identify vulnerability, it's subjective to the size of the file.

V. EVALUATION

We are planning to depict performance and time-taken to scan of tools and analysing which one is better for which scenario. Our final details tools list consists of the parameters such as time-taken, OS Support, Programming language to easily choose tool to find and mitigate the vulnerability.

We also aim to benchmark these open source tools with various vulnerable inputs (i.e. various versions of vulnerable applications tweaked to simulate various test cases). Tools that properly identify the vulnerability are calculated as true positives. Tools that do not properly identified the vulnerability with the previous ones are physically affirmed and then only it is stated as false positives.

We have setup test-bed of three different operating systems as mentioned below and 8 vulnerable applications 3.

- MacOS M1 Chip 16 GB
- Linux Ubuntu 20.04 8 GB VM
- Windows 10 Intel Core i5 8 GB

We took 3 attack tools and ran them on GHIDRA application [4]. All 3 tools generate payload on their respective server. This payload is then used to trigger attack as mentioned in methodology.
Fullhunt tool [3] generates payload once you trigger the attack and this tool is used to scan URLs only.

Figure 1. FullHunt scan performed on a domain

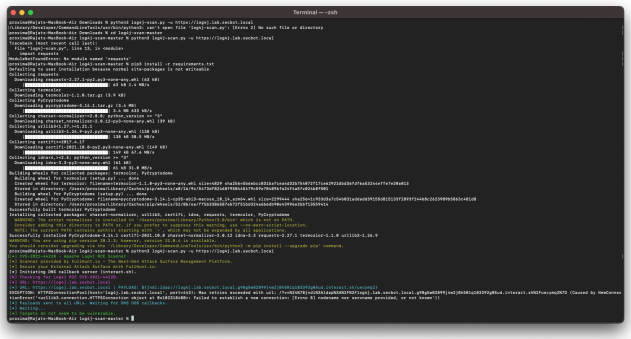


Figure 3 is the test performed of Fullhunt tool where we need to provide URL to test and then it return whether target is vulnerable or not.

While other two tools, Log4shell vulnerability test tool [15] and Huntress tool [16] generates their payload on an website which can be used to trigger attack by either passing as HEADER parameter of any request or triggering website's API with payload in request parameters as query or body and return IP address, Type, Source of application etc.

Figure 2. Scan performed using Log4shell vulnerability test tool

Time	Type	Source	Message
2022-04-28 03:03:50	recv_dns_query	prov1.cpe.cl.vy.clearcable.net.	DNS query received. It is likely that your log4j deployment supports doing lookups. This can lead to information leakage.
2022-04-28 03:03:51	recv_dns_query	prov1.cpe.cl.vy.clearcable.net.	DNS query received. It is likely that your log4j deployment supports doing lookups. This can lead to information leakage.
2022-04-28 03:03:52	recv_dns_query	67.202.185.122	DNS query received. It is likely that your log4j deployment supports doing lookups. This can lead to information leakage.

Figure 3. Scan performed using Huntress tool

IP Address	Date/Time	Extra Keys
67.21.186.151	2022-04-28T03:07:34.815Z	[]
67.21.186.151	2022-04-28T03:07:34.448Z	[]

For the analysis of passive tools, We took a aggregated folder of vulnerable applications and ran it through all the tools. We compared the time each tool is taking on different different platform and how is the user experience. Here are some findings:

Figure 4. Scan performed using Arctic wolf tool

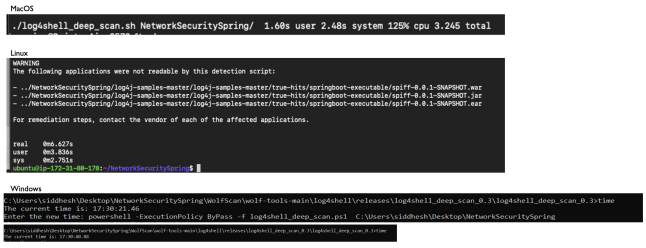


Figure 5. Arctic wolf scan ouput

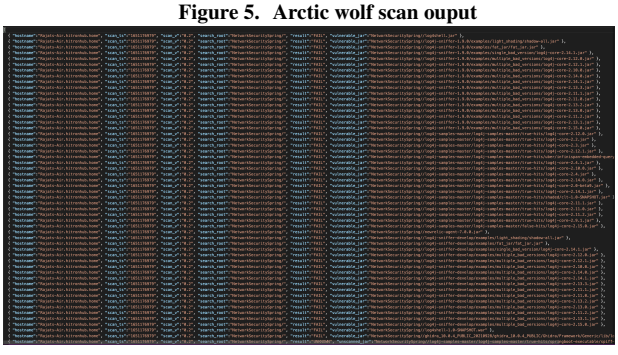
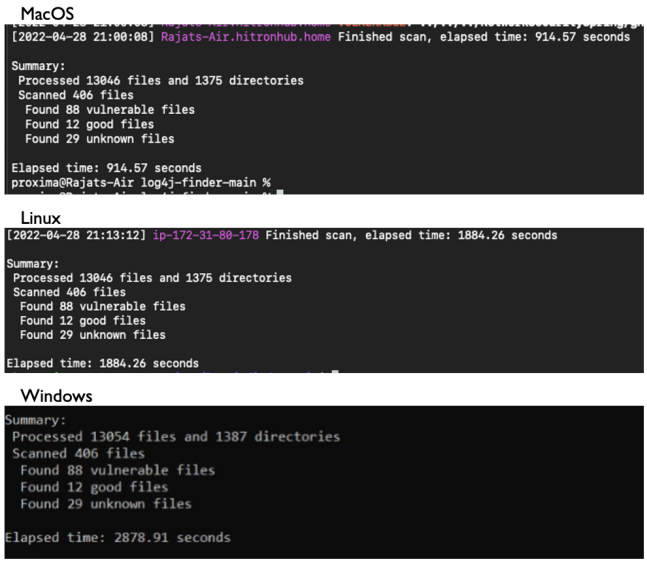


Figure 5 represents the output of Arctic wolf scan where it shows a detailed view of each file that was scanned and whether it is vulnerable or not.

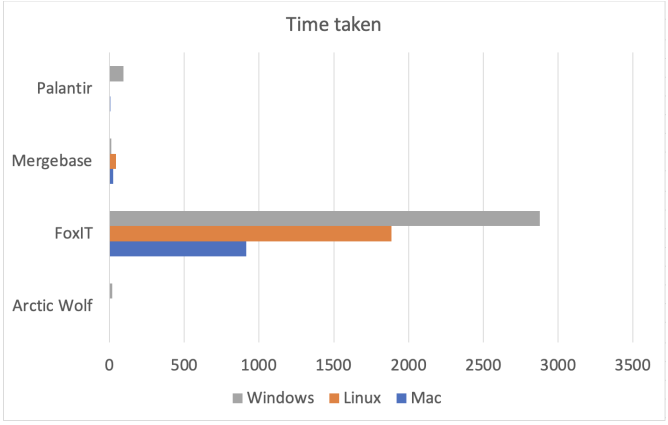
Figure 6. Scan performed using FoxIT tool



We performed time analysis on 4 tools and this was the output:

Time taken by tools in Seconds			
Tool Name	**MacOS**	**Linux**	**Windows**
Arctic Wolf	2.48	3.83	19
FoxIT	914	1884	2878
Mergebase	27	45	14
Palantir	7	N/A	95

Figure 7. Time analysis of tools



We came to the conclusion that Arctic Wolf was the best tool since it required very little time to execute on each platform Table 4 and provided detailed output Figure 5.

VI. CONTRIBUTIONS

Our team has divided tools and assigned each member to research and perform in-depth analysis of time, user experience, platform support, output etc. Figure 8 is list of the assigned tools. Remaining tools has been picked on voluntarily basis.

Figure 8. Tools assigned to each team member

Assignee	Tool Name	Programming Language	Applies Fix/Mitigation	Platform	Tool Type
Ashish	CISA Log4j Scanner	Java, Python	No fixing mechanisms	Any	Attack
Ashish	Google Log4jScanner	Golang	No fixing mechanisms	Any	Filescan
Rajat	Huntress Labs Log4shell Vulnerability Tester	Java	No fixing mechanisms	Any	Attack
Ashish	Artic Wolf Log4Shell Deep Scan	Java, Python	No fixing mechanisms	Any	Filescan
Siddheshwar	Crowdstrike Archive Scan Tool	Powershell Script	No	Windows	Filescan
Siddheshwar	Mergebase Log4j Detector	Java	No	Any	Filescan
Rajat	Datto Log4Shell Enumeration, Mitigation, and Attack Detection Tool for Windows and Linux	Yara rule, Shell Script	No	Any	Filescan

VII. FUTUREWORK

Some of the limitations with current analysis is the quantification of efficacy or performance. With the continuation of this study we aim to quantify the performance of these analysis tools based on metrics such as Accuracy, Recall, Precision, F-Score Specificity. We also aim to widen our scope with other commercial scanning solutions by tech giants such as Microsoft windows defender [19] which also supposedly tend to alert on false positives.

VIII. CONCLUSION

While analyzing tools, we encountered several modifications, such as MacOS rejecting a few tools because it couldn't validate them. Aside from that, there are tools with insufficient documentation, and executing them on Windows Powershell is a time-consuming operation in and of itself. Apart from that, attacking an application might result in a security breach for the business. For example, testing GHIDRA is dangerous because it is controlled by the National Security Agency. According to our taxonomy, tools rans on MacOS delivered results faster than windows, with windows being the slowest. Few tools are simple to test, and others are considerably more difficult to setup. Few passive tools, out of all those available, require privileged access to run. Last but not least, not all extensions are supported by all tools.

REFERENCES

[1] GitHub - fox-it/log4j-finder: Find vulnerable Log4j2 versions on disk and also inside Java Archive Files (Log4Shell CVE-2021-44228, CVE-2021-45046, CVE-2021-45105). (????). https://github.com/fox-it/log4j-finder [Online; accessed 2022-03-20].

[2] 2021. GitHub - Florian Roth's definitions. (2021). https://github.com/Neo23x0/signature-base/blob/master/yara/expl_log4j_cve_2021_44228.yar

[3] 2021. GitHub - Fullhunt Log4j Scan. (2021). https://github.com/fullhunt/log4j-scan [Online; accessed 2022-03-28].

[4] 2021. GitHub - Ghidra: Ghidra is a software reverse engineering (SRE) framework created and maintained by the National Security Agency Research Directorate. (2021). https://github.com/NationalSecurityAgency/ghidra [Online; accessed 2022-03-28].

[5] 2021. GitHub - Google log4jscanner. (2021). https://github.com/google/log4jscanner [Online; accessed 2022-03-28].

[6] 2021. GitHub - LogPresso CVE 2021-4428 Scanner. (2021). https://github.com/logpresso/CVE-2021-44228-Scanner [Online; accessed 2022-03-28].

[7] 2021. GitHub - Mergebase Log4j Detector. (2021). https://github.com/mergebase/log4j-detector [Online; accessed 2022-03-28].

[8] 2021. GitHub - Palantir log4j-sniffer. (2021). https://github.com/palantir/log4j-sniffer [Online; accessed 2022-03-28].

[9] Binary Defense. 2022. log4j-honeypot-flask. (jan 2022). https://github.com/BinaryDefense/log4j-honeypot-flask original-date: 2021-12-14T18:08:45Z.

[10] Bi.Zone. 2022. Log4j Detection. (jan 2022). https://github.com/bi-zone/Log4j_Detector original-date: 2021-12-13T14:46:03Z.

[11] CISA. 2022. Log4j Scanner. (jan 2022). https://github.com/cisagov/log4j-scanner original-date: 2021-12-21T16:23:29Z.

[12] Crowdstrike. 2021. CrowdStrike Launches Free Targeted Log4j Search Tool | CrowdStrike. (dec 2021). https://www.crowdstrike.com/blog/free-targeted-log4j-search-tool/

[13] Datto. 2021. Datto RMM Software. (2021). https://www.datto.com/products/rmm/

[14] Datto. 2022. Log4Shell Enumeration, Mitigation and Attack Detection Tool. (jan 2022). https://github.com/datto/log4shell-tool original-date: 2021-12-13T17:09:38Z.

[15] Github. 2021. Github - Microsoft log4k Scanner. (2021). https://github.com/alexbakker/log4shell-tools

[16] Huntress. Huntress - Log4Shell Tester. (????). https://log4shell.huntress.com/

[17] Rob Jepson. 2022. PortSwigger/log4shell-everywhere. (jan 2022). https://github.com/PortSwigger/log4shell-everywhere original-date: 2021-12-16T09:43:53Z.

[18] Trend Micro. 2021. Trend Micro Apache Log4j vulnerability. (2021). https://www.trendmicro.com/en_us/apache-log4j-vulnerability.html

[19] Microsoft. 2021. Microsoft log4k Scanner. (2021). https://tinyurl.com/ynb7zve3

[20] National Institute of Standards AND Technology. 2021. NVD - CVE-2021-44228. (2021). https://nvd.nist.gov/vuln/detail/CVE-2021-44228

[21] Joe Panettieri and 2022. 2022. Log4j Vulnerability Scanners and Detection Tools: List for MSSPs and Threat Hunters. (jan 2022). https://tinyurl.com/4psm6twa

[22] Silent Signal. 2022. Log4Shell scanner for Burp Suite. (jan 2022). https://github.com/silentsignal/burp-log4shell original-date: 2021-12-12T14:52:49Z.

[23] Trend Micro. 2021. Log4j - Vulnerability Tester. (dec 2021). https://log4j-tester.trendmicro.com/

[24] Artic Wolf. wolf-tools/log4shell at main · rtkwlf/wolf-tools. (????). https://github.com/rtkwlf/wolf-tools