Yash Agarwal

Apache Log4j Logging Framework and its Vulnerability

Metropolia University of Applied Sciences

Master of Engineering

Information Technology

Master's Thesis

1 March 2022

PREFACE

The major objective of this thesis is to share the knowledge about Apache Log4j and its vulnerability and how we can mitigate to secure our applications from various attacks from potential hackers.

Writing this thesis is not easy for me as the research topic is quite vast and unique for me as it's new. This wouldn’t be possible with the support.
of my friends.

I also would like to thank Ville Jääskeläinen, Head of Master’s program in Information Technology as a full-time job and study is never an easy undertaking.

Helsinki, April 9, 2022
Yash Agarwal

Abstract

Author:	Yash Agarwal
Title:	Apache Log4j Logging Framework and its Vulnerability
Number of Pages:	67 pages
Date:	1 March 2021
Degree:	Master of Engineering
Degree Programme:	Information Technology
Professional Major:	Networking and Services
Supervisors:	Sami Sainio
	Ville Jääskeläinen, Head of Master's program in IT

Apache Log4j (or Log4j) is a logging framework written in Java, developed by Apache to provide Logging activities for the web server. It has Java JNDI (Java Naming and Directory Interface API) lookup which provides Naming-lookup and Directory mapping feature. It uses LDAP (Lightweight Directory Access Protocol) query to identify different services and Java applications running on a platform together so that they can share resources and communicate with each other thereby avoiding the need of deploying same services for different applications.

The LDAP service has a vulnerability that allows an attacker to craft queries. These queries can be used to execute commands on the platform such as reading logs, querying services, and performing other, possibly malicious actions on it. The request forged in the LDAP query will be parsed by the JNDI API and will provide a way to remotely execute any command send by the attacker (Remote Execution Code). Various services, webservers developed by Microsoft, Oracle, Google running Java or Java applications were affected.

Aim of this thesis is to discuss Apache Log4j vulnerability currently present on the Log4j software (from version 2.0 to 2.14). A sample attack on a simple Minecraft server is also demonstrated to emulate the working of Log4j vulnerability in a real-life scenario.

It was important to shed some light to this vulnerability because of its harmful nature. The vulnerability caused havoc since it can be remotely executed (Remote Code Execution). CVSS (Common Vulnerability and Scoring System) analysis was also performed on this vulnerability to gain more insight on its working.

Some fixes and workarounds are also discussed since no permanent fix is available to this date. Log4j vulnerability scanning was also performed on author's device (Windows and Linux) to check for applications affected by this vulnerability.

Keywords: Security, Log4j, Logging, Java, JNDI, LDAP

Contents

List of Abbreviations

RCE	Remote Code Execution
JNDI	Java Naming and Directory Interface
LDAP	Lightweight Directory Access Protocol
JVM	Java Virtual Machine
KB	Kilo Byte
JAVA SE	Java Standard Edition
IP	Internet Protocol
API	Application Programming Interface
JRE	Java Runtime Environment
JDK	Java Development Kit
JNDI SPI	JNDI Service Provider Interface
CVE	Common Vulnerabilities and Exposures
MDC	Mapped Diagnostic Context
HTTP	Hypertext Transfer Protocol
POM	Project Object Model
XML	Extensible Markup Language
TomEE	Apache Tomcat Enterprise Edition
WAF	Web Application Firewall
STDOUT	Standard Output
CVSS	Common Vulnerability Scoring System
FIRST	Forum of Incident Response and Security Teams
NAIC	National Association of Insurance Commissioners
IBM SPSS	IBM Statistical Package for the Social Sciences

1 Introduction

The Apache Log4j or Log4j Vulnerability was discovered on November 24th, 2021, by Alibaba Security Group [1]. They manage the Security services for Alibaba company and related child organizations. Even though the vulnerability was reported in November, no strict actions were taken to address this vulnerability until December 10th, 2021, when it was made public by Apache themselves. The vulnerability was given ID CVE-2021-44228. Soon after multiple vulnerabilities related to Log4j like CVE-2021-44228, CVE-2021-44832, CVE-2021-44015 were also discovered.

This vulnerability allows an attacker to execute a RCE (Remote Code Execution) attack. This is performed by making a well-formatted JNDI (Java Naming and Directory Interface API) query which can be inserted into a LDAP (Lightweight Directory Access Protocol) server which can then lookup resources on local or remote machine. As a result, anything for which LDAP server is listening to and can find can be returned to the JNDI manager.

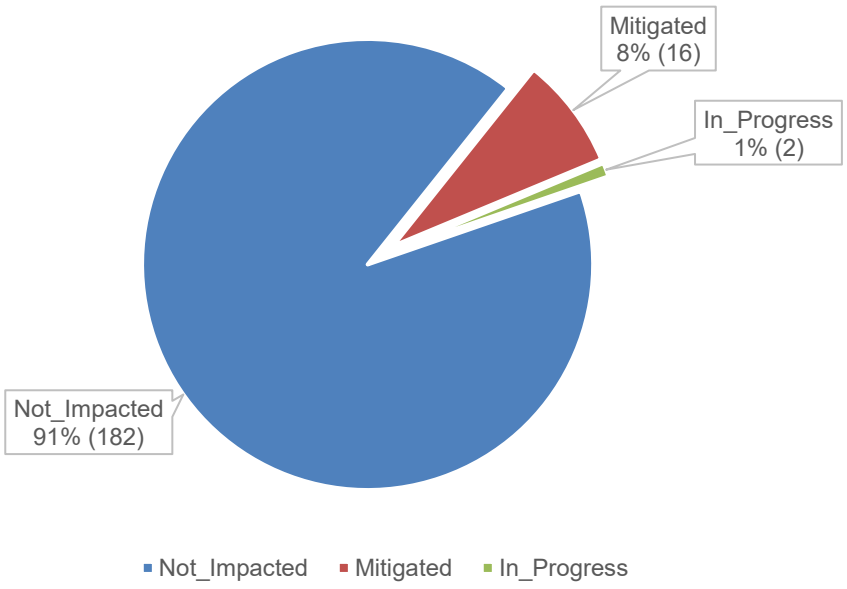


Figure 1 Google Cloud products and services affected by Log4j

Google also published an article on their security blog on December 17th, 2021, stating that at least 35,000 Java packages were affected [2] [3]. Io.netty, junit-jupiter-engine, springframework.boot among others were the most famous libraries hit by the Log4j Vulnerability. From the Maven-Repository total of top 8% widely used libraries were severely affected which relied on Log4j-core.

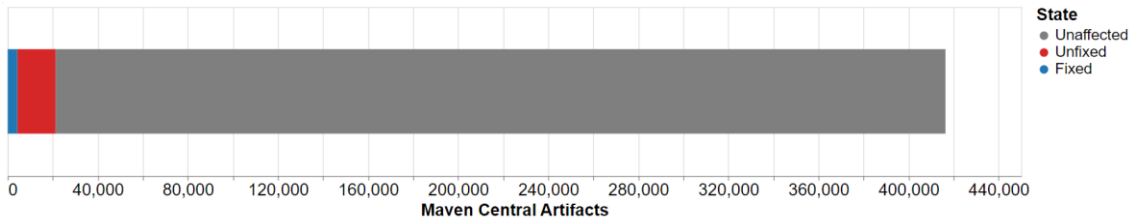


Figure 2 Maven Repositories affected by Log4j

Google security developers added that, many libraries from Maven repository depends indirectly on Log4j as well.

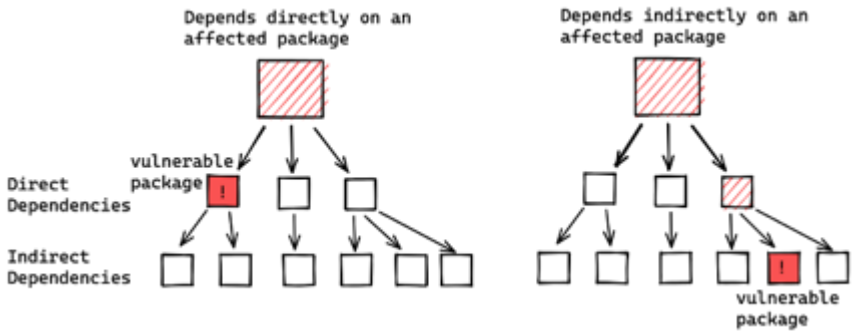


Figure 3 Direct & Indirect package dependency

The figure 3 clearly explains that if some packages are directly affected by Log4j Vulnerability, then other packages which direct dependency on those packages are affected as well. As a result, this chain can continue to deeper levels as well.

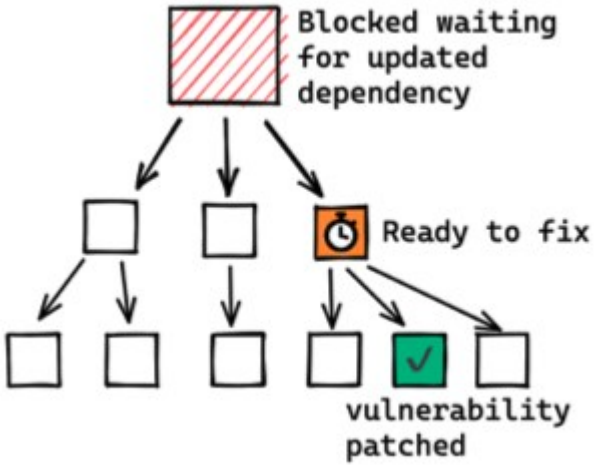


Figure 4 Recursive patching strategy

Packages on deeper levels needed to be identified first and then their parent packages will be fixed and finally bottom-up approach is needed to fix all the affected packages.

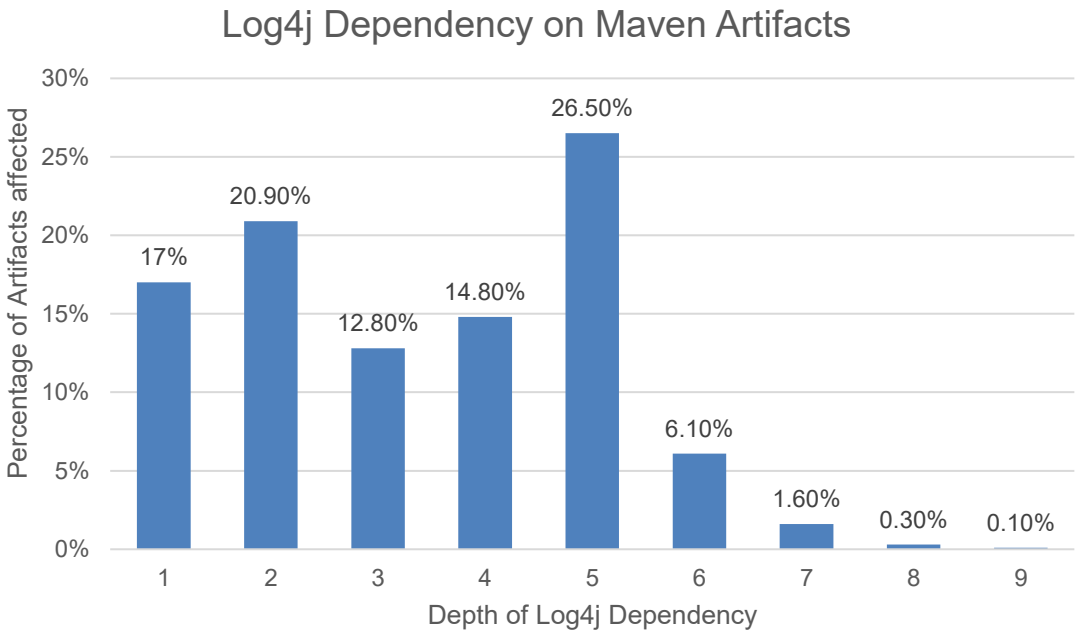
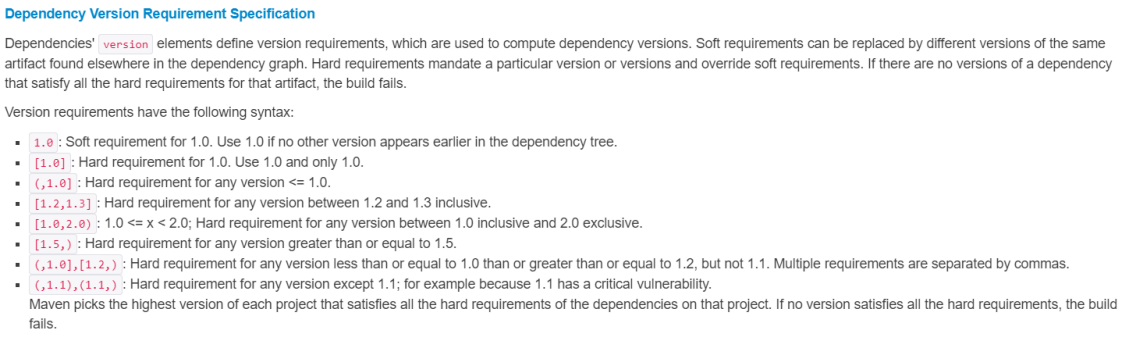


Figure 5 Log4j Dependency on Maven Artifacts

From the figure 5 it can be observed that more than 80% of the maven packages have indirect relation with Log4j library and while some have indirect relation as deep as 8 or 9 levels.

Another problem exists for the affected packages because Java uses “soft” as a naming convention to select the packages used to build an application. A “soft” version specifies a basic version in case the recent version cannot be obtained for that library.



Dependency Version Requirement Specification

Dependencies' `version` elements define version requirements, which are used to compute dependency versions. Soft requirements can be replaced by different versions of the same artifact found elsewhere in the dependency graph. Hard requirements mandate a particular version or versions and override soft requirements. If there are no versions of a dependency that satisfy all the hard requirements for that artifact, the build fails.

Version requirements have the following syntax:

- `1.0` : Soft requirement for 1.0. Use 1.0 if no other version appears earlier in the dependency tree.
- `[1.0]` : Hard requirement for 1.0. Use 1.0 and only 1.0.
- `(,1.0]` : Hard requirement for any version <= 1.0.
- `[1.2,1.3]` : Hard requirement for any version between 1.2 and 1.3 inclusive.
- `[1.0,2.0)` : 1.0 <= x < 2.0; Hard requirement for any version between 1.0 inclusive and 2.0 exclusive.
- `[1.5,)` : Hard requirement for any version greater than or equal to 1.5.
- `(,1.0],[1.2,)` : Hard requirement for any version less than or equal to 1.0 than or greater than or equal to 1.2, but not 1.1. Multiple requirements are separated by commas.
- `(,1.1),(1.1,)` : Hard requirement for any version except 1.1; for example because 1.1 has a critical vulnerability.

  Maven picks the highest version of each project that satisfies all the hard requirements of the dependencies on that project. If no version satisfies all the hard requirements, the build fails.

Figure 6 Maven "Soft" version naming conventions

In “soft” terms JAVA dictates that if no recent package is found from the repository, then the default version that is “1.0” will be used which can be decades old. “Hard” version on the other hand clearly specifies to use the exact version only [4]. This is a bad development approach because in-case if some recently updated and stable package is not available then its default version will be used which can be unstable and maybe be packed with older vulnerabilities.

2 Theoretical Background

2.1 What is Apache Log4j

Apache Log4j is a reliable, fast, and flexible logging framework which is written in Java. All the developers and security analysts use logging in the applications. This helps them to get various insights during the code execution. Developers use this for troubleshooting to find problems with the application, [5] and security analysts use this to find anomalies within the flow-traffic. Logging using Log4j can be configured at runtime to produce in-depth logging through external configuration files. The Logging process could be viewed in terms of levels. Different Logging levels like debug, info, warn, error, fatal and so on which can generate log at various levels. For example, the logging level starts from trace, debug, info, warn, error and ends at fatal.

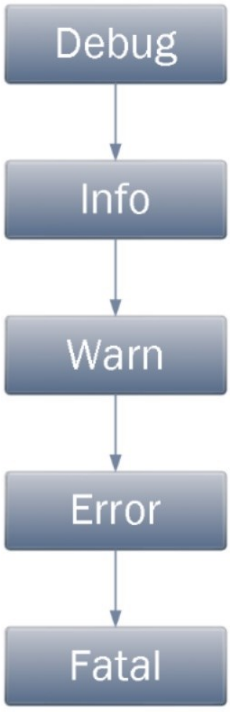


Figure 7 Logging levels

The Apache Log4j logging framework also offers mechanisms to direct logging information to different destinations like a database or file or console or uniSys log and others.

2.2 Features of Apache Log4j

The most important feature of Apache Log4j is that it is thread safe. The “logger-class” used is a “synchronised class”. As a result, whatever method(s) which will be used with the class will be all synchronised as well. It is also optimised for high speed because the tracing happens very fast in a logging framework, so it supports multiple output “appenders” per logger. It also supports internationalisation which means that it can also communicate with various programming languages. It is also not restricted to a predefined set of facilities which means it can be configured in any way the user wants hence it is flexible.

By these features, the any logging behaviour at the runtime can be logged to a configuration file. The configuration file can be used to read and write since it is generated in “**.xml**” format or “**.properties”** if different frameworks are used as well. The logging level can be set in a way that all that logging level before it will follow the level set automatically. For example, from the above figure 7, if we set the logging level as “warn” then “debug” and “info” before it will automatically be logged as well.

The format of the log output can be easily changed by extending the layout class. Hence the layout is useful in cases where a time stamp like date/time is needed with the logs. The target of the log output file as well as the writing strategy can also be altered by the implementations of the appender interface.

Logging is an important component of the software development. A well-written logging code offers quick debugging structure of an application. The application can be effectively debugged at runtime which allows easy bug-tracking and maintenance. But, Logging does have its drawbacks as well since it can slow

down an application. If the logging configuration is made too verbose it can cause **Scrolling blindness** [6] [7]. Scrolling Blindness means that multiple exceptions and errors are thrown to the program and program becomes unresponsive.

2.3 Apache Log4j Components

There are 3 major components of Apache Log4j are Loggers, Appenders and Layouts.

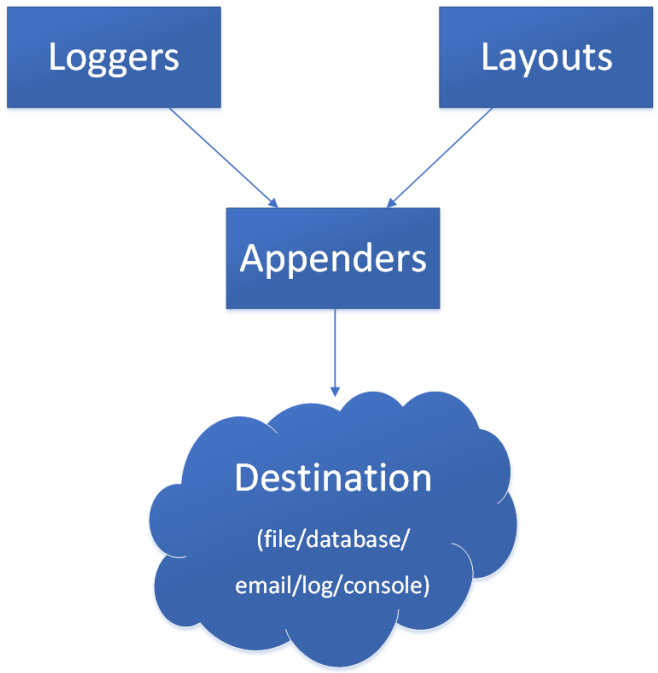


Figure 8 Log4j components

Java is highly Object-Oriented-Programming-language. All the code in Java is written with the help of classes. Further “class file” is generated from the compiled code so that the code can easily run on any platform running the “JVM”. The **Loggers** are used to capture the logging information required for the class-class communication. Log4j provides a method called as “logger.getLogger()” specifically for this purpose. The name of class is passed as an argument to this method and logging is obtained for that class directly onto the console. **Appenders** are used to write the information to any preferred destination

provided being it a file or database or email or log-file or console. **Layouts** is responsible for formatting logging information into different styles.

2.4 Apache Log4j Architecture

Apache Log4j allows a layered structure where each layer provides different objects to perform different tasks. This makes the design flexible and easy to extend. There are two types of objects available with Log4j framework "**Core objects**" and "**Support objects**".

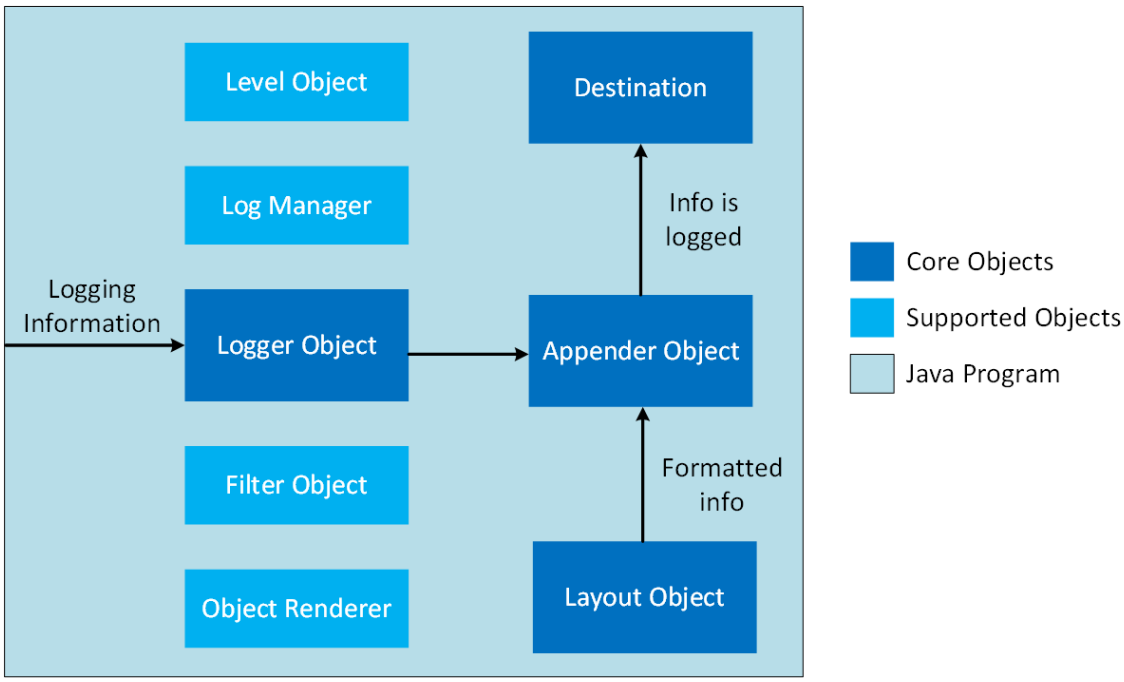


Figure 9 Log4j Architecture

- The **Core objects** are mandatory objects of the framework as they are required to use the framework itself. Support objects are optional objects, and they support core objects to perform additional but important tasks.
- The **Logger object** is a top layer in the Log4j Framework. It is responsible for capturing the all the "Logging" information and then the Logs are stored in a "Namespace hierarchy".

- The **Layout layer** provides objects (**Layout objects**) which are used to format logging information in different styles or layouts. It produces the result to the Appender in a better human readable and reusable format. By using layout objects logging information can be decided on how to display the logging information like adding a timestamp.
- **Appenders** provide "**Appenders objects**" which are at the lower level of Apache Log4j architecture. They are responsible for publishing logging information to various preferred destinations like database or a file or a console.
- **Support objects** are the **Optional objects**. They work with Core Objects to provide additional help. As their name suggests they are optional to use. Support Objects are further divided into different Objects which are :-
  - *Level object* defines the granularity and priority of any logging information. There are seven levels of logging defined within this API which are debug, info, error, warn, fatal and all.
  - *Filter object* is used to analyse logging information and make further decisions on whether that information should be logged or not. The Appender objects can have several filter objects associated with them. If logging information is passed to a particular appender object all the filter objects associated with them need to approve the logging information before they can publish to the attach destination.
  - *Object render* specialises in providing a string representation of different objects passed to the logging framework. It is used by the layout objects to prepare the logging information.
  - *Log Manager* manages the logging framework is responsible for reading the initial configuration parameters from a system-wide configuration file or a configuration class.

2.5 Apache Log4j Configuration

Configuring Apache Log4j involves assigning the level defining appenders and specifying the layout objects in the configuration file. The **Log4j.properties** file is

a Log4j configuration file. This keeps properties in key value pairs. The configuration is written following the naming convention of Java. This helps Log4j to find the properties file easily and all the configurations can be easily embedded.

log4j.properties

```
# Root logger option
log4j.rootLogger=INFO, file, stdout

# Direct log messages to a log file
log4j.appender.file=org.apache.log4j.RollingFileAppender
log4j.appender.file.File=C:\\logging.log
log4j.appender.file.MaxFileSize=10MB
log4j.appender.file.MaxBackupIndex=10
log4j.appender.file.layout=org.apache.log4j.PatternLayout
log4j.appender.file.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %-5p %c{1}:%L - %m%n

# Direct log messages to stdout
log4j.appender.stdout=org.apache.log4j.ConsoleAppender
log4j.appender.stdout.Target=System.out
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
log4j.appender.stdout.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %-5p %c{1}:%L - %m%n
```

Figure 10 Log4j ".properties" file

By default, the Log Manager looks for the file name **Log4j.properties** in the class path hence it is important to keep this property file in the class path. The Log4j properties syntax is defined at the root logger along with the appenders in the properties file.

2.6 Apache Log4j Appenders

Apache Log4j provides “Appender Objects” which are primarily responsible for printing logging messages to different destination which can be files or sockets and the event log or other different destinations.

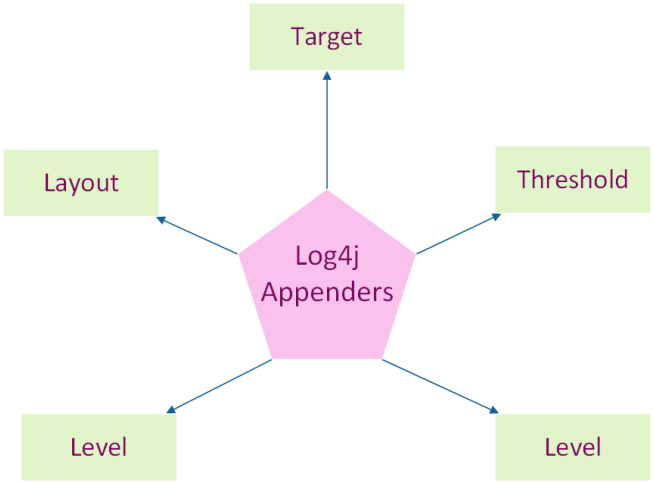


Figure 11 Apache Log4j Appenders

Each appender object has different properties which are Layout, Target, Level, Threshold.

- Appenders uses the Layout objects and the conversion pattern associated with them to format the logging information.
- Target objects define the path where the logging information needs to be sent, the destination can be such as console, a file or something else where logging information can be stored and read.
- Level Objects control the filtration of the log messages.
- Threshold Appender can set a threshold level for the logging information and that can be independent of the logger level.
- Filter objects can analyse logging information beyond level matching and decide whether logging requests should be handled by a particular appender.

Here are the several appenders listed below:

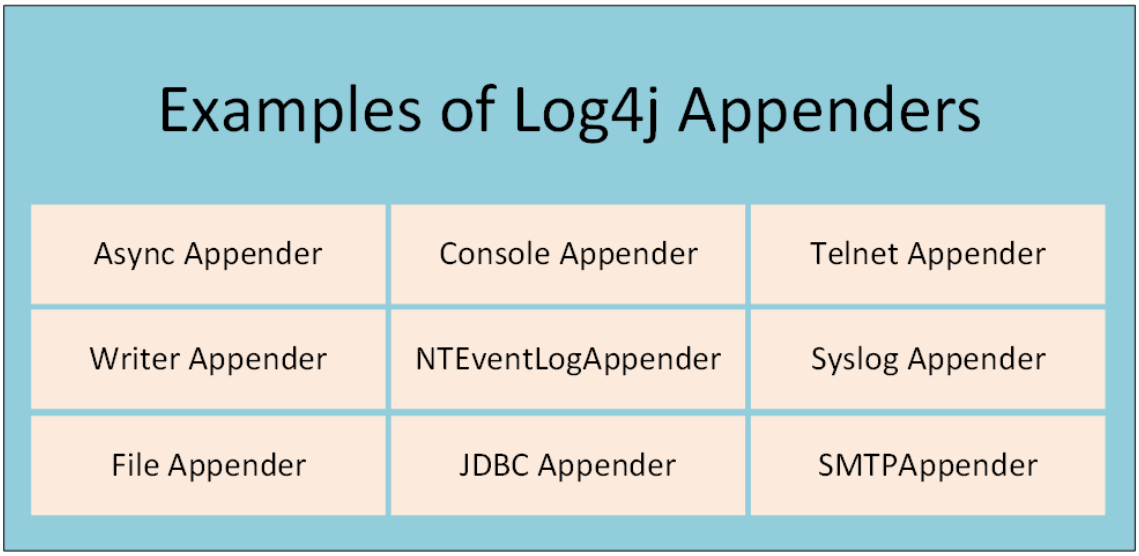


Figure 12 Log4j Appenders example

2.7 Apache Log4j File Appender Configuration

To write the logging information to a file `org.apache.Log4j.FileAppender` method is used. `FileAppender` method has various parameters like:

- `ImmediateFlush`,
- `Threshold`
- `Filename`
- `Append`
- `MaxFileSize`
- `MaxBackupIndex`

and many more. These parameters can be used to set value manually or if not then default values will be used.

log4j.FileAppender

```
log4j.rootLogger = DEBUG, FILE

log4j.appender.FILE=org.apache.log4j.FileAppender

log4j.appender.FILE.File=${log}/log.out

log4j.appender.FILE.ImmediateFlush=true

log4j.appender.FILE.Threshold=debug

log4j.appender.FILE.Append=true

log4j.appender.FILE.MaxFileSize=5MB

log4j.appender.FILE.MaxBackupIndex=2

log4j.appender.FILE.layout=org.apache.log4j.PatternLayout
log4j.appender.FILE.layout.conversionPattern=%m%n
```

Figure 13 Log4j.FileAppender properties

Immediate flush is a flag which is by default set to true governing the delivery timing of the logging information that will be flushed to the output file specified. In simple terms log messages are delivered after every method call.

The threshold value can be used to set the threshold level for each appender along with the level which is set at the root level. The File name as it says for itself is the name of the log file.

Append attribute dictates that if the file size of the logging file exceeds a maximum file size limit, then remaining logging information will be sent to new a file and this process continues until the logging stops. If buffered input output is enabled it indicates that the buffer size is 8 KB.

Logging can also be done in multiple files which is called rolling files. This happens whenever logging file reaches maximum file size limit so we create additional files and all the new data.

3 Practical Implementation of Apache Log4j

3.1 Introduction

In this topic, the implementation of the Log4j attack will be discussed [8]. Various tools and scripts are used to explain the attack. To successfully demonstrate the attack some of the following things are required: Minecraft clients running the official Minecraft game, Java SE Development Kit version "1.8.0_181" to be precise, A Minecraft server running on the Victim's machine, A Kali Linux attacker Machine running Http and Maven server to parse the requests.

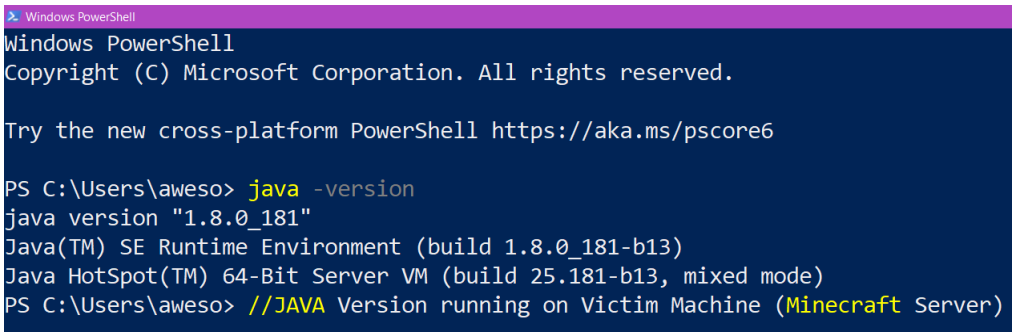


Figure 14 Java version on "Victim's machine"

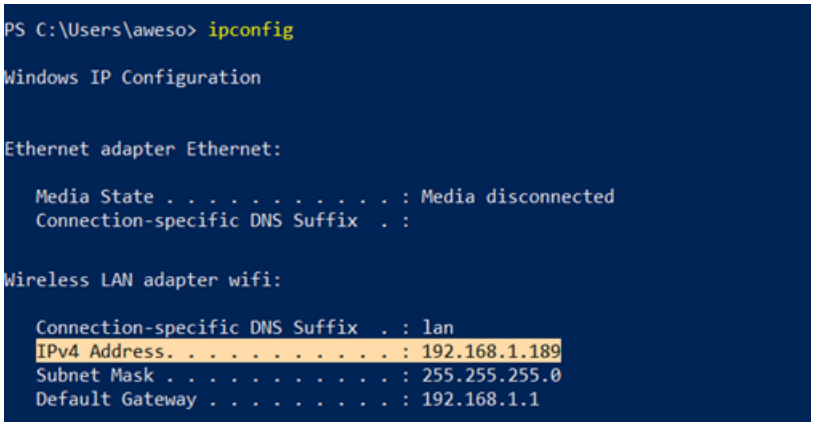


Figure 15 IP configuration of Victim's Machine

Figure 16 Java version of Attacker Machine 1 (Ubuntu)

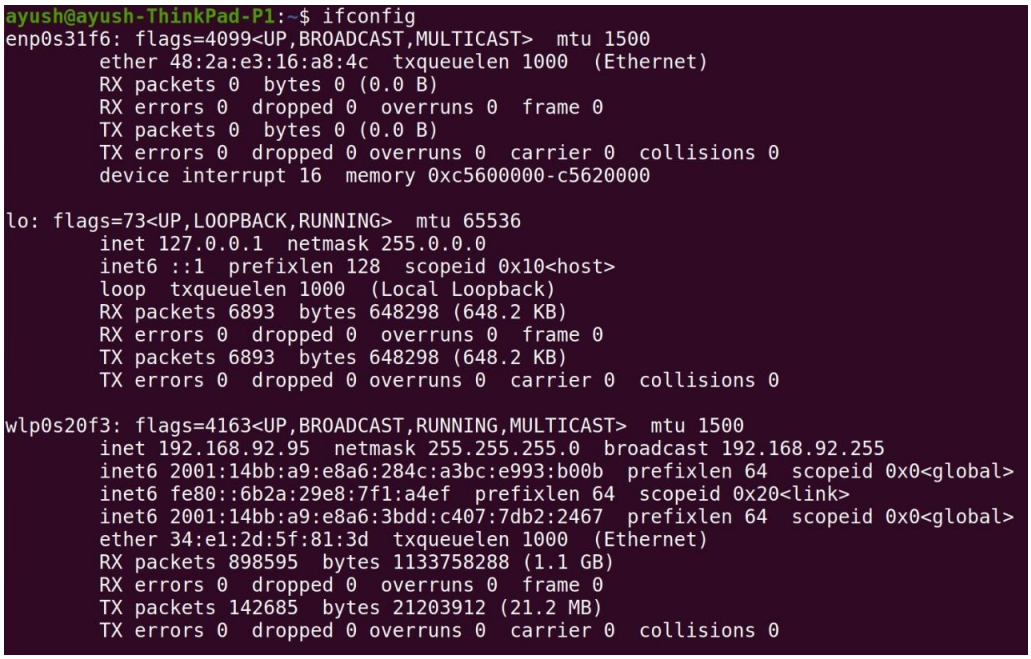


Figure 17 IP Configuration of Attacker Machine 1 (Ubuntu)

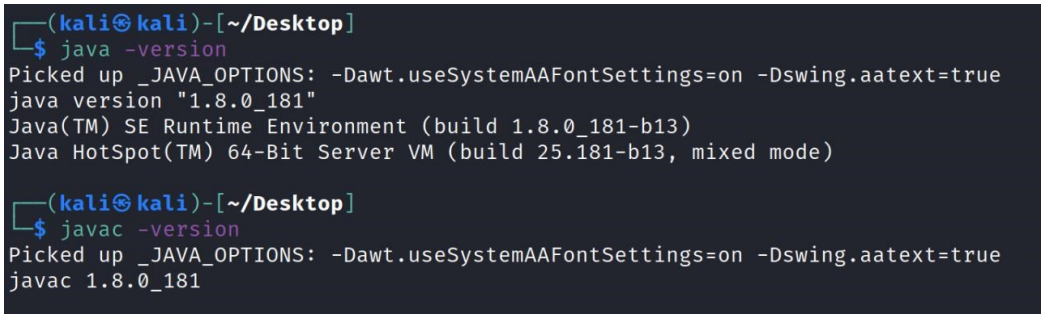


Figure 18 Java version of Attacker Machine 2 (Kali Linux)

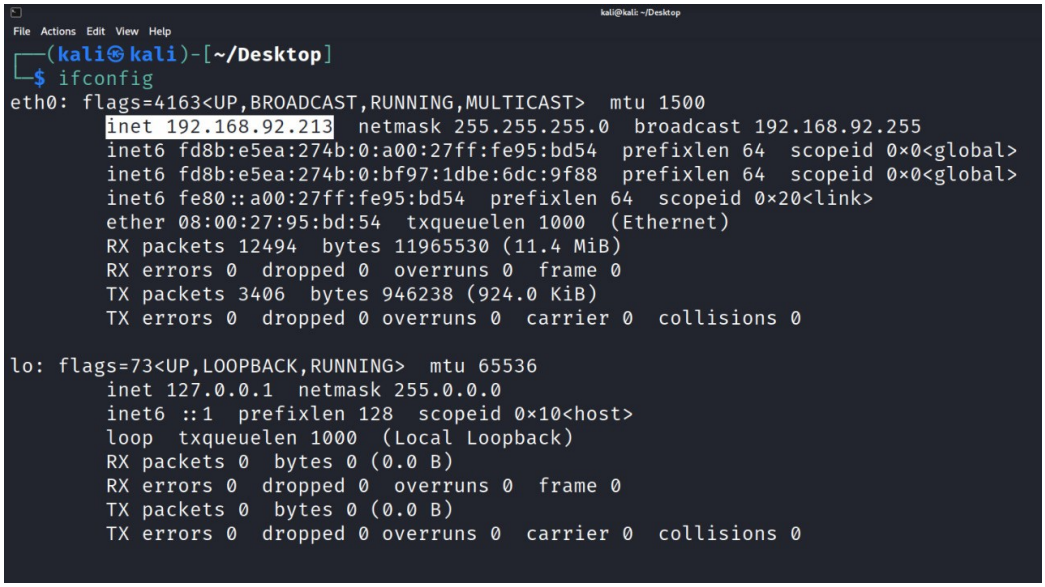


Figure 19 IP Configuration of Attacker Machine 2 (Kali Linux)

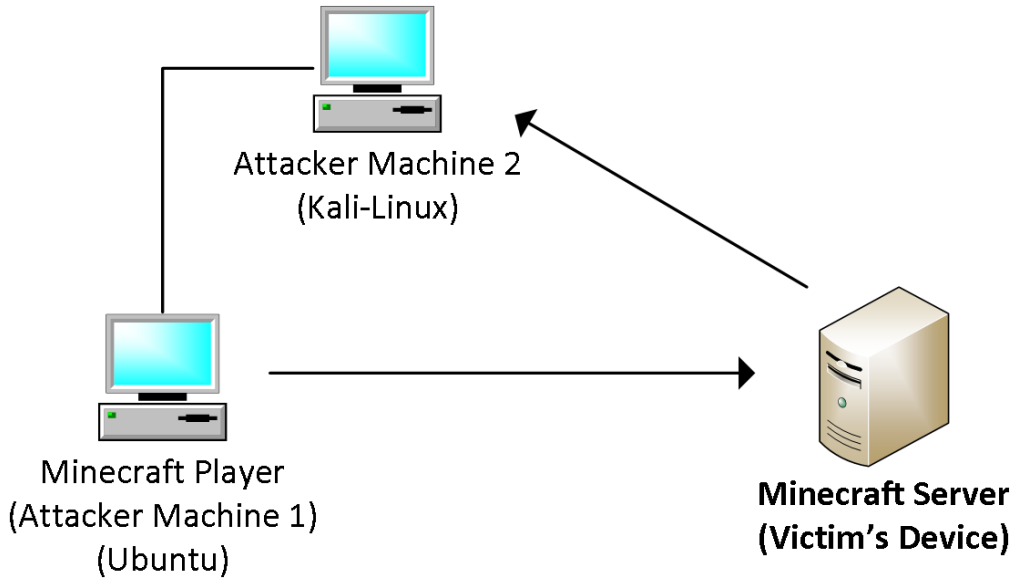


Figure 20 Attack Diagram

The Minecraft server is running on the Windows platform with a Java version of 1.8.0_181. On the Minecraft server, anyone can join and connect to it so that Minecraft can be played with other people on the server. The Attacker Machine 1 is running a Minecraft game and is connected to the server, it has also the previous version of Java installed i.e., 1.8.0_181 to match server requirements.

This machine will send the LDAP request to the Server. The Attacker Machine 2 is also connected to the server so that it is also in the same network as the server. This machine will parse the LDAP queries sent by the Attacker Machine 1 and will reply so that remote code execution can be executed. This is also running the same version of Minecraft and Java to match the server requirements.

3.1.1 Minecraft Server: “Spigot”

SpigotMC.org created “Spigot” in 2012 to provide open-source lightweight Minecraft servers [9]. Their main aim was to provide fast and ready-to-deploy Minecraft servers so that Minecraft players can play games easily with each other and showcase their work. They have also introduced different tools like BungeeCord and CraftBukkit. With the help of these tools, one can easily connect different Minecraft servers so that cross-platform and cross-server gaming can be enabled. There also exists an API for the “Spigot” client so that users can easily modify and add plugins to the “Spigot” project, providing Minecraft-Server connectivity to different platforms like Discord and Twitch that Server-Management can be made easier.

3.1.2 Java Runtime Environment (JRE)

Java was introduced as a programming language so that Applications can be developed on one platform and deployed on multiple platforms [10]. JRE 1.8.0_181 was specifically used in the demonstration because Log4j vulnerability exists in this version. Also, it was the widely used and most stable Java Development framework before the Log4j was discovered. The Minecraft Server uses version 1.8.0_181 so that vulnerability in consideration can be run.

3.1.3 Java Naming and Directory Interface (JNDI)

Java Naming Directory Interface as the name suggests is an API for Java Applications to provide naming and Directory functionality [11]. It provides a

Naming-lookup and Directory mapping feature so that various objects can be referred to in an environment by a Java application if more than one Java application is simultaneously running on a platform. So various services can be implemented so that if multiple applications exist simultaneously, they all can use the services thereby avoiding the need to deploy each service manually.

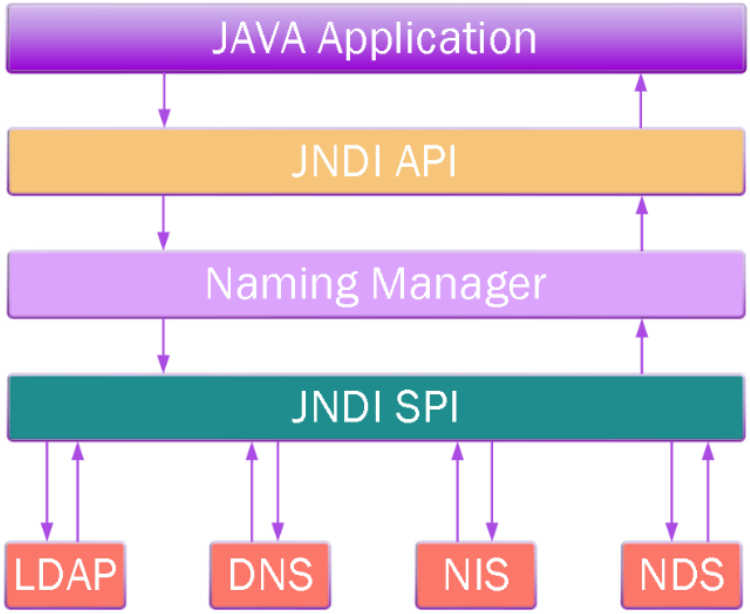


Figure 21 JNDI Architecture

The Java application(s) communicated with the JNDI API which then communicates with the Naming Directory so that names of available services can be provided which are present and running. Further, the Naming Directory communicates with the JNDI SPI (Service provider interface) to which the different Services are attached. Hence a particular process can call one or more services simultaneously.

3.1.4 Kali Linux

Kali Linux is one of the most famous open-source Linux distros that is used for Cyber-Security analysis. It comes with pre-installed tools and various scripts that

can be used by Security researchers to do Security analysis and run exploits and malicious scripts to record their behaviour.

3.1.5 HTTP server with Python

HTTP server implements a simple Client-Server communication so that it can handle HTTP requests [12]. It provides access to a resource which can be a database, web page, etc depending on the request query. In this context, an HTTP server with python was deployed. Python provides an abstract implementation of an HTTP server that can be run with just simple command or configured to be deployed manually using python's built-in classes in scripts to serve a specific purpose.
`python3 -m http.server` is a one-liner command to deploy a simple HTTP server. This command loads the http-class from python with default configurations and provides simple access to the server. As a result, an HTTP server will be set up on the localhost with 8080 being the default port.

3.1.6 Maven server

Maven or Apache Maven is a tool that is used to manage projects in Java programming language [16]. It is based on the Project-Object Model. A POM file is written in XML. The POM file provides the project information along with its configuration details. This file is fed to the Maven so that it can read the configuration and build the system accordingly.

```
<project>
  <!--  model version is always 4.0.0 for Maven 2.x POMs  -->
  <modelVersion>4.0.0</modelVersion>
  <!--  project coordinates, i.e. a group of values which uniquely identify this project  -->
  <groupId>com.mycompany.app</groupId>
  <artifactId>my-app</artifactId>
  <version>1.0</version>
  <!--  library dependencies  -->
  <dependencies>
    <dependency>
      <!--  coordinates of the required library  -->
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <!--  this dependency is only used for running and compiling tests  -->
      <scope>test</scope>
    </dependency>
  </dependencies>
</project>
```

Figure 22 Sample POM File example for Maven

So, with the help of Maven everything related to the project can be specified in one XML file, and then the project can be built with ease. This will be provided as an LDAP refer server.

The Maven server in consideration uses the 1.8.0_181 Java version because it should have the same requirements as the Minecraft Server.

3.2 Attack Execution

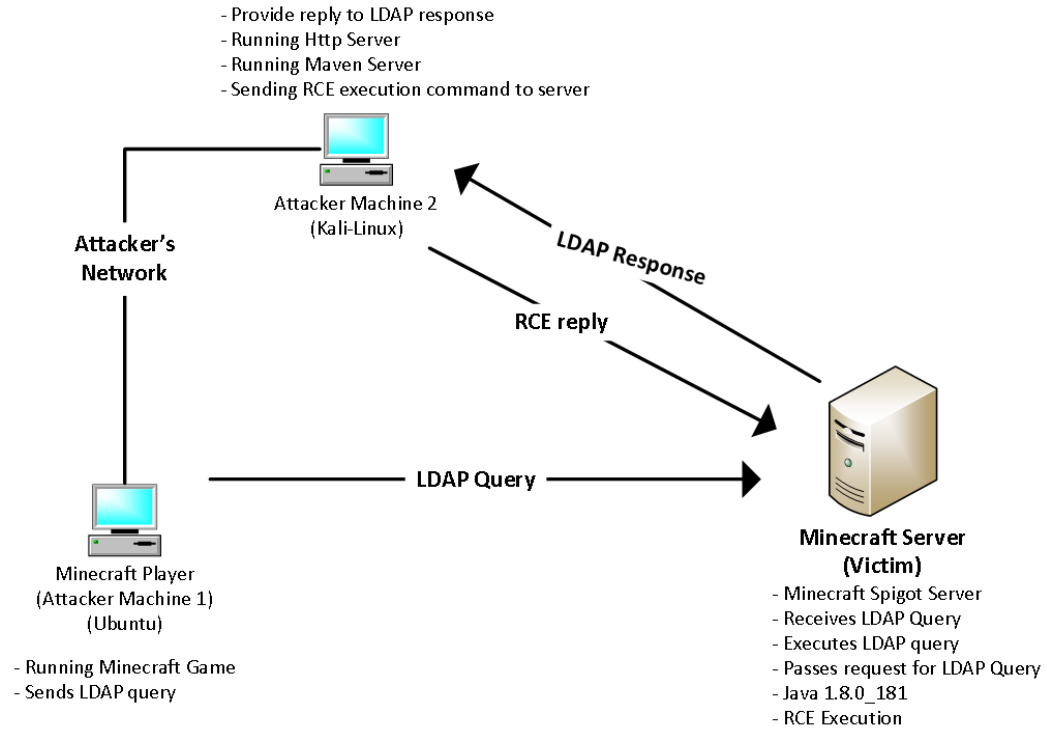


Figure 23 Detailed attack execution explanation

The attack execution is demonstrated in the following steps:-

Step 1: The Server Machine (victim) hoisting the Minecraft, is started using the command: "Java -Xms2G -Xmx2G -jar spigot-1.7.2-R0.4-SNAPSHOT-1339.jar". The spigot provides ready to run Minecraft server, hence it can be started in the Windows PowerShell or CMD using the proper parameters.

It should be noted that IP address of the Minecraft server is "192.168.1.189".

```
Windows PowerShell
PS F:\Log4j\pigot-mc-server> java -Xms2G -Xmx2G -jar spigot-1.7.2-R0.4-SNAPSHOT-1339.jar
Warning, your max perm gen size is not set or less than 128mb. It is recommended you rest
M
Please see http://www.spigotmc.org/wiki/changing-permgen-size/ for more details and more
Loading libraries, please wait...
[22:34:27 INFO]: Starting minecraft server version 1.7.2
[22:34:27 INFO]: Loading properties
[22:34:27 INFO]: Default game type: SURVIVAL
[22:34:27 INFO]: This server is running CraftBukkit version git-Spigot-1339 (MC: 1.7.2) (
[22:34:27 INFO]: Using 4 threads for Netty based IO
[22:34:27 INFO]: Server Ping Player Sample Count: 12
```

Figure 24 Running Minecraft server

Step 2: Minecraft-Launcher in Attackers 1 machine.

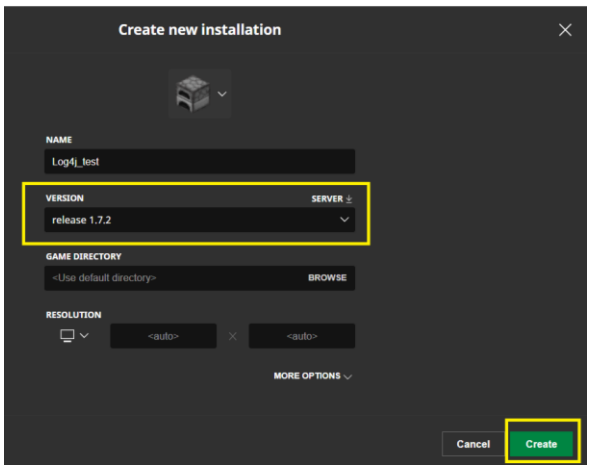


Figure 25 Launching Minecraft in Attacker Machine 1

Here, a new installation was created and Minecraft version: “release 1.7.2” was selected because Log4j-vulnerability exists in this version, hence it is easy to test in this version.

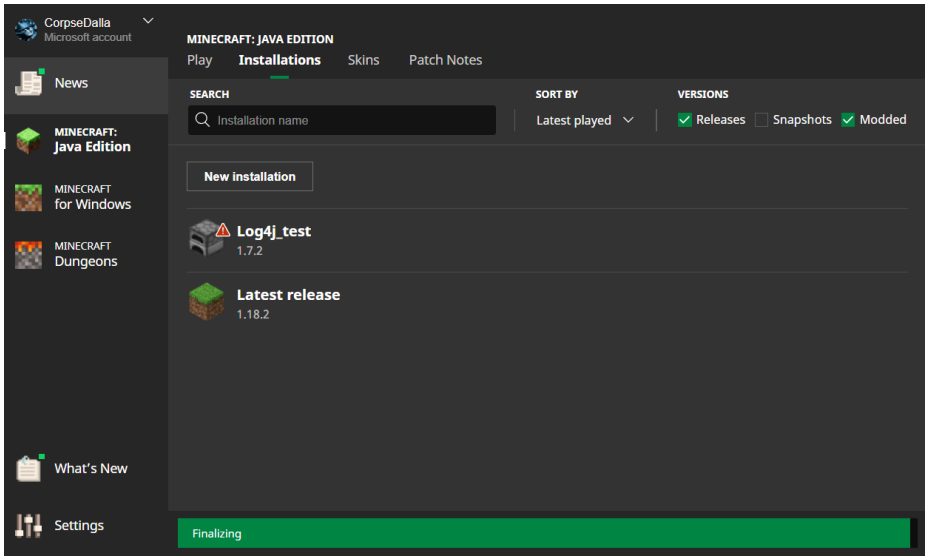


Figure 26 Creating new Minecraft installation on Attacker Machine 1

When the play button was pressed, the game will begin and a welcome screen will be displayed.

Figure 27 Starting Minecraft

Further, the multiplayer option was selected, and then connection to the server was made by adding the IP address of the server.

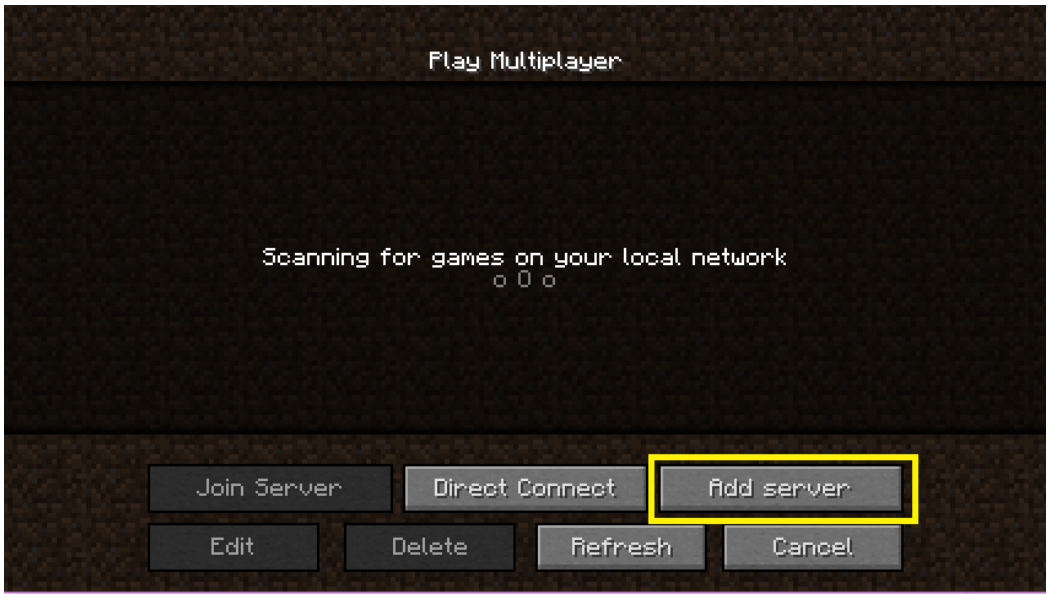


Figure 28 Adding server on local network

Figure 29 Adding server details (name and IP)

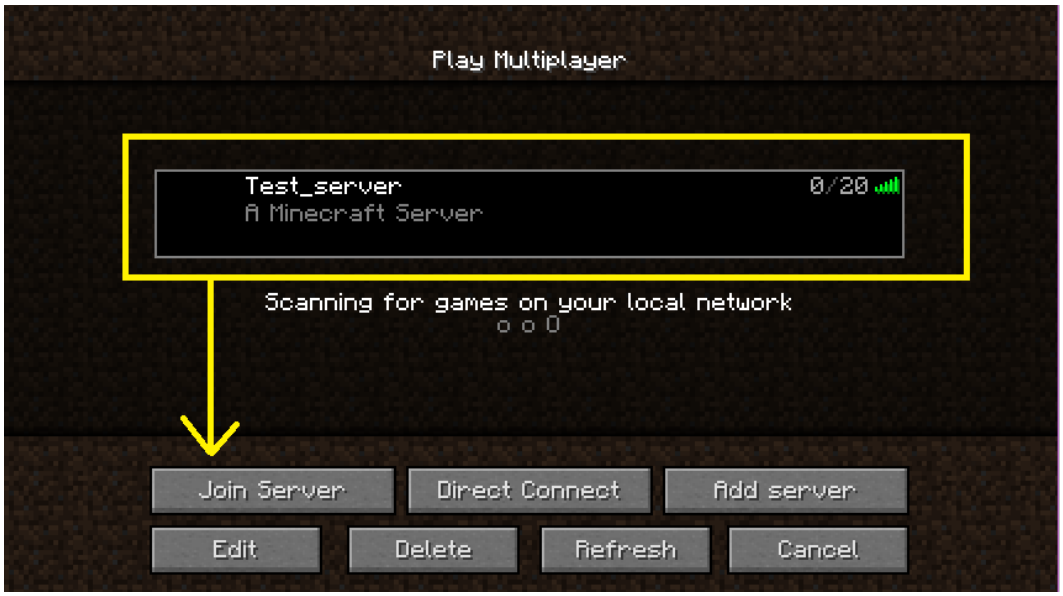


Figure 30 Connecting to server

After entering the details, it can be noted that the server added popped as active with ping in signal bars. Further “Join-Server” was selected and then Minecraft was launched on the Attacker Machine as a client on the server.

Figure 31 Minecraft session joined on local server

The Server also logs all the activities of all the clients who joined or left the game. It can be observed that a user “CorpseHhr” joined a game.

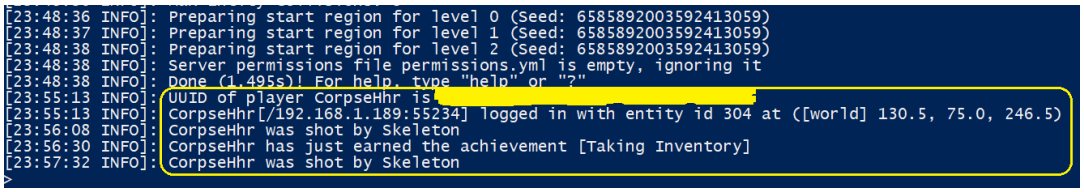


Figure 32 Minecraft server logs

Step 3: From the GitHub, “Marshelsec” repository is used on the Attacker Machine 2. This repository provides a ready-to-use Maven server that can be built and be directly used. This repository provides all the tools to execute a successful Log4j RCE attack on any server machine which is running previous or old versions of Java.

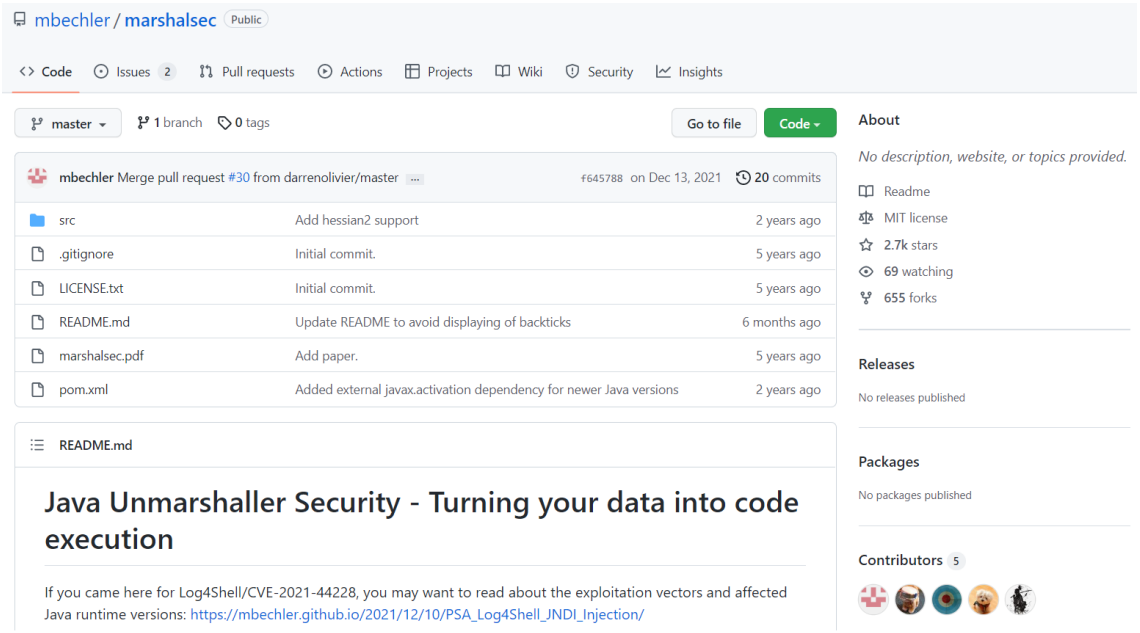


Figure 33 GitHub repository used for attacking

```
−<project xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>org.eenterphace.mbechler</groupId>
    <artifactId>marshalsec</artifactId>
    <version>0.0.3-SNAPSHOT</version>
  −<properties>
      <maven.compiler.source>1.8</maven.compiler.source>
      <maven.compiler.target>1.8</maven.compiler.target>
      <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    </properties>
  −<dependencies>
      <!-- Util -->
    −<dependency>
        <groupId>com.sun.activation</groupId>
        <artifactId>javax.activation</artifactId>
        <version>1.2.0</version>
      </dependency>
    −<dependency>
        <groupId>org.javassist</groupId>
        <artifactId>javassist</artifactId>
        <version>3.19.0-GA</version>
      </dependency>
    −<dependency>
        <groupId>org.reflections</groupId>
        <artifactId>reflections</artifactId>
        <version>0.9.9</version>
      </dependency>
    −<dependency>
        <groupId>org.slf4j</groupId>
        <artifactId>slf4j-nop</artifactId>
        <version>1.7.24</version>
      </dependency>
    −<dependency>
```

Figure 34 Maven configuration from cloned GitHub repository

Step 4: After cloning this GitHub repository and building the Maven-LDAP server, the server is started. The repository also includes a pre-made pom.xml file which can be used by the Maven to automatically build our files and required objects.

The Maven tells the LDAP referral server what to run, on which IP address (localhost in this case) and refers to which specific class name or Java name it should execute after sending the response. The command used is:-

“Java -cp target/marshalsec-0.0.3-SNAPSHOT-all.jar marshalsec.jndi.LDAPRefServer **"http://192.168.85.213:8000/#CALC""**

Here, the IP of the Attacking Machine 2 is supplied in the command because the request should be called to this machine. Also, **CALC** is the name of the Java class which should be called for the **LDAP** reference. Finally, it will run the LDAP-reference server on the IP provided and start to listen on the port **“1389”**.

Figure 35 Starting LDAP listener

Step 5: In some other directory (/marshalsec/poc in this consideration) a Java file is created. It consists of a payload syntax that will be executed on the Victim’s machine. The Java code described executes a “Notepad.exe” command on the Victim opening it 10 times as the payload syntax is inside the Loop. This “CALC.Java” file is compiled using the “Javac CALC.Java” command and a “CALC.class” file is generated.

Figure 36 Java file containing malicious code

Figure 37 Making a class file from previous Java file

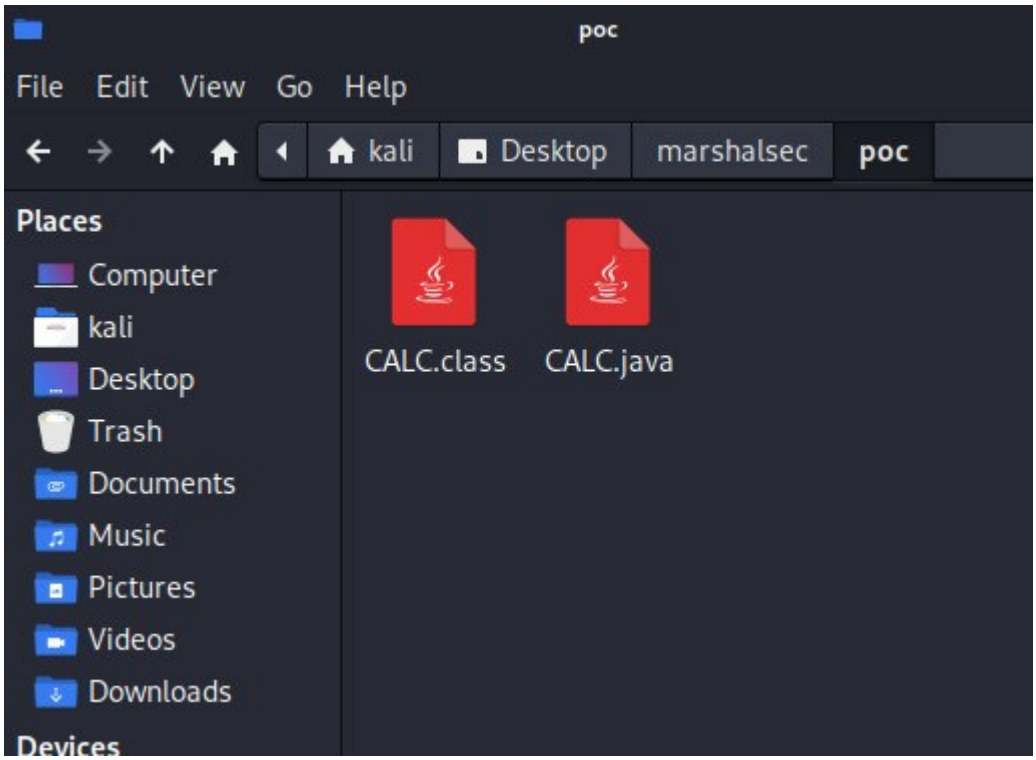


Figure 38 "CALC.class" Class file created

Step 6: After this, another “HTTP” server is started using python3 which will execute the Java code on the target (Victim Machine in this case).

Figure 39 Starting python-http server

Step 7: After this, the listener for the LDAP and referral server is waiting to provide the reference call-back to the **“8000”** port where the http-server is running.

Step 8: In the Minecraft client session, a well-formatted LDAP query is passed into the message box to the Server. The LDAP query in consideration is “$(jndi:ldap://IP-Of-Attacker-machine-2:PORT/Class-Name)”.
The final query looks like: **“${jndi:ldap://192.168.92.213:1389/CALC}”**.

Figure 40 Executing LDAP query in Minecraft session in command box

When the query is run without the port number, the command does nothing. But when the query is run with the port the payload is executed. As a result, 10 instances of Notepad open up.

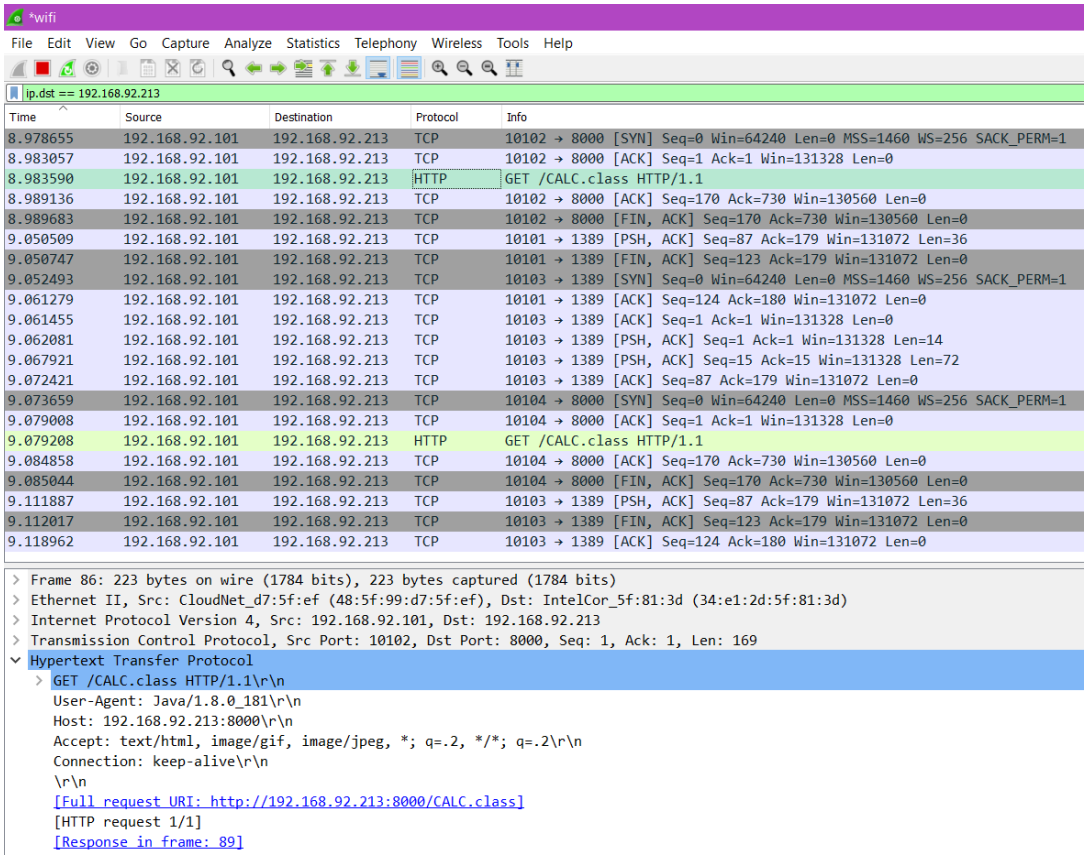


Figure 41 Network Capture of executed JNDI query (JNDI request and response)

From the network trace on the Victim machine on the Wireshark after applying the filter for the Destination IP of the Attacker Machine 2, it can be noticed that LDAP-Query is converted to the HTTP request from Victim to Attacker Machine 2. Also, the User-Agent is our Java hence Log4j Vulnerability is executed.

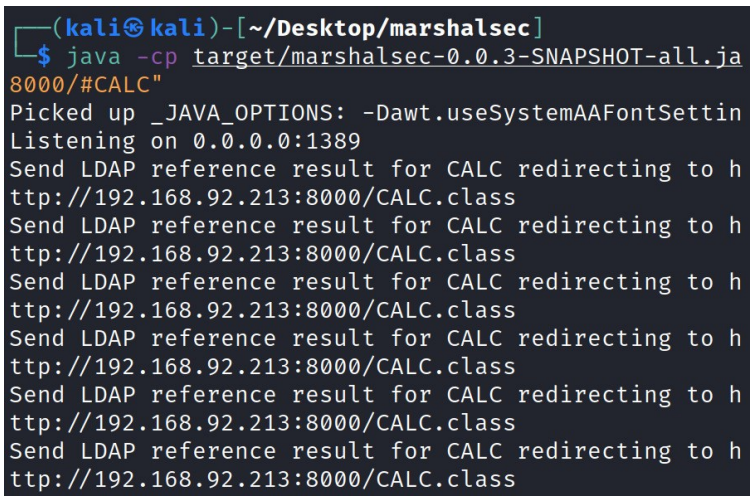


Figure 42 Executing JNDI query on Victim (Minecraft server in this case)

On the Attacker Machine 2, the maven and HTTP servers produce a log that shows that a connection request has been made from the Victim Machine. It can be observed that maven LDAP-Listener handles the request from the Victims' LDAP query and passes it to the HTTP server.

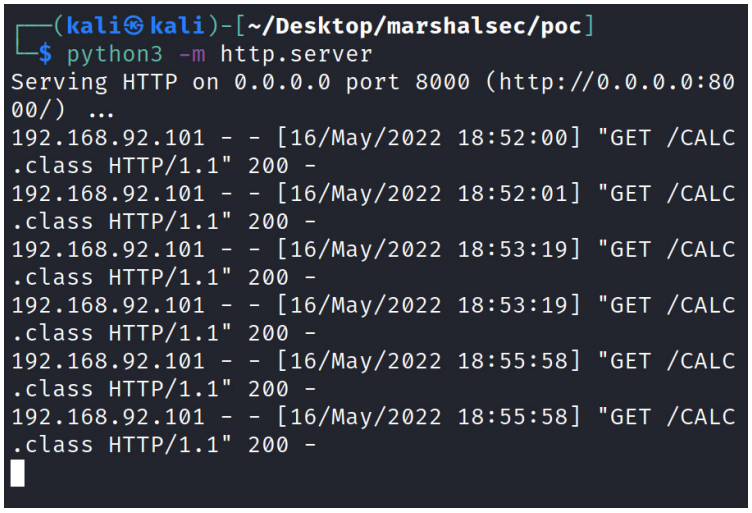


Figure 43 Handling LDAP query on LDAP listener

4 Log4j2 Vulnerability and possible Fixes

Log4j2 is the version 2 of Log4j which provides improvements over its previous version. Recently three versions of Log4jJava have been released, which are 2.15, 2.16, and 2.12.2 [17]. The hackers got into this vulnerability when they witnessed this as a bug in Microsoft Game. It was then further weaponized. In the Minecraft game project, a **JNDI** class file is used. As the name is self-explanatory it is a **Java naming directory interface**. It is a feature of Java where all the application objects are defined as a tree structure which will have one root node, stems, branches, and multiple leaf nodes. This structure is used for application references for all JNDI roles [18].

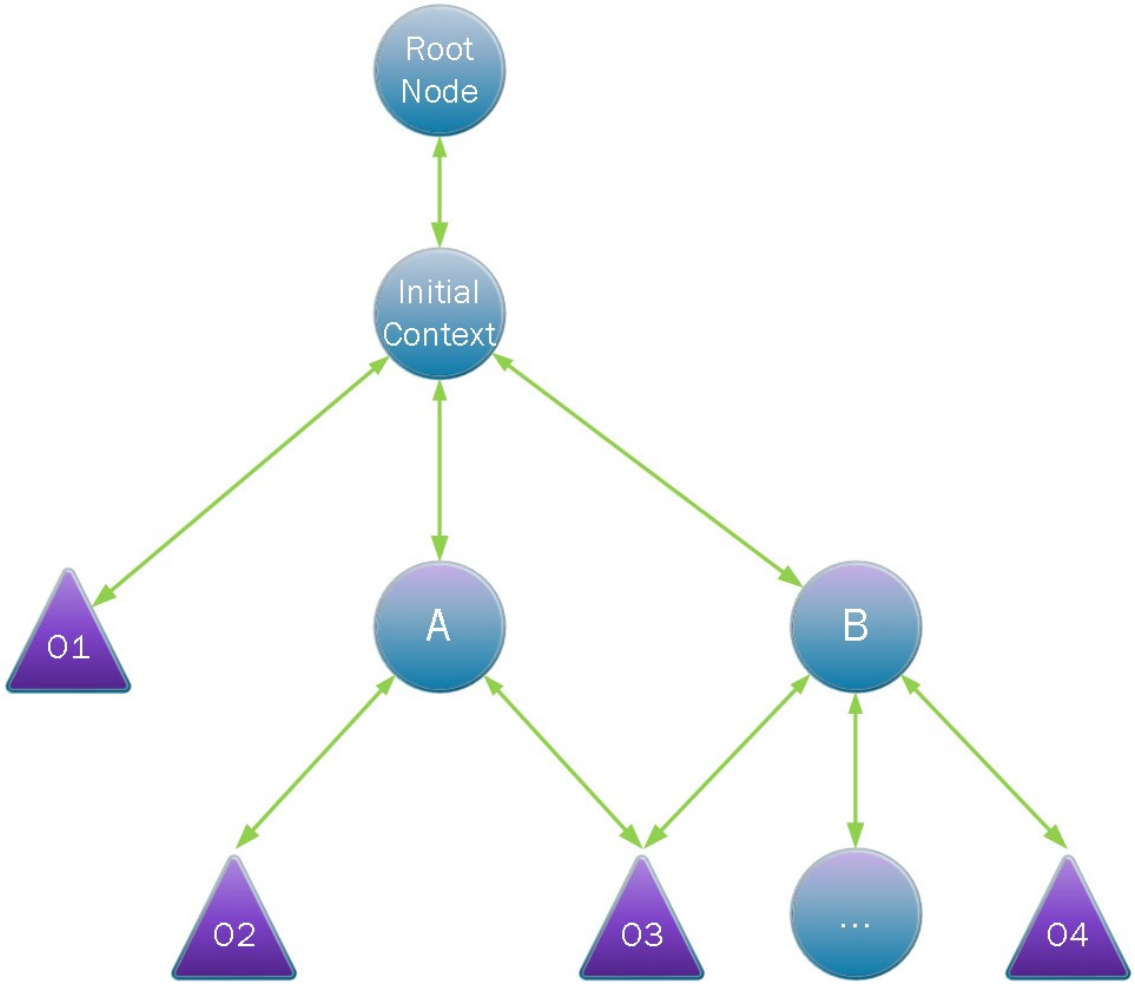


Figure 44 JNDI tree mapping for resource access

From the above figure, for example if O4 (Object 4) bound to interface B needs to communicate with O2 (object 2) bound to interface A, the calling will be:-

```
RootNode.InitialContext.B.O4(RootNode.InitialContext.A.O2)
```

In similar way if the application requires a database connectivity it will make a JNDI lookup query for it and then JNDI direct reference goes and looks for it. This is useful in cases if logging is needs to be done for a specific item in the tree hence avoiding the ned to log whole system. Hence if objects are added and present into the JNDI Tree, they can be called in this manner.

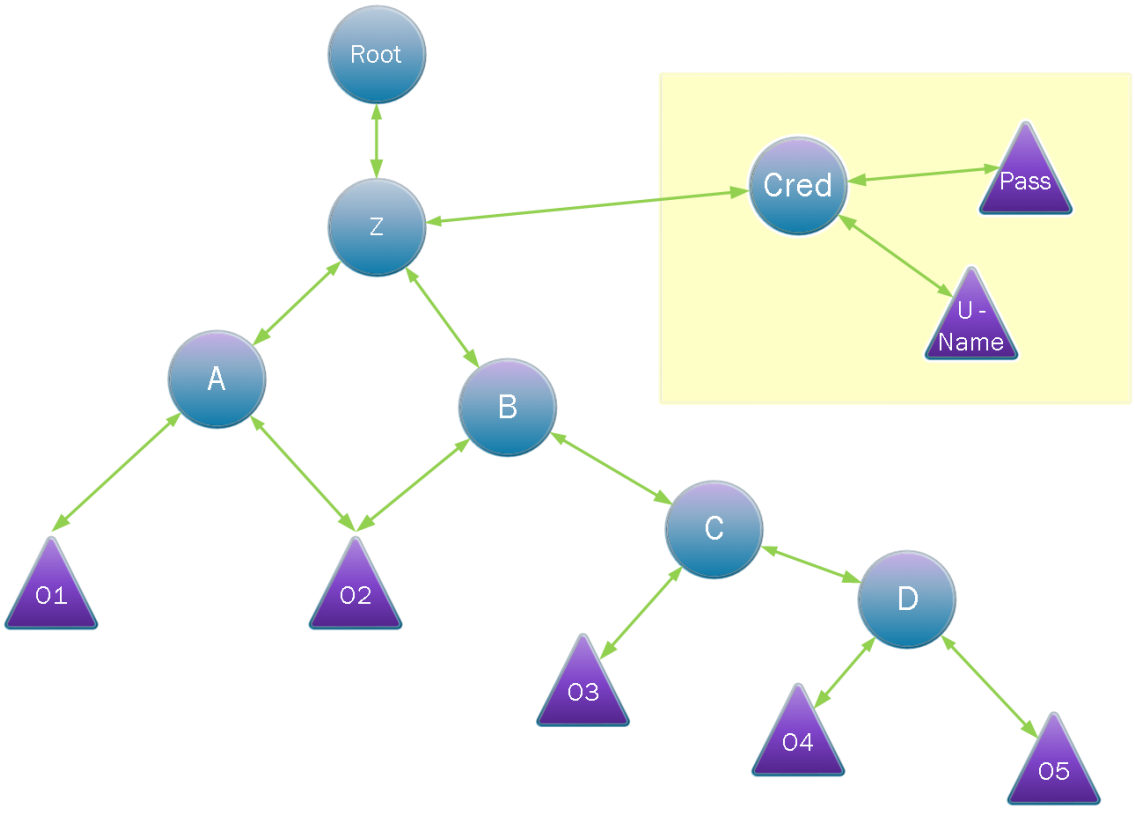


Figure 45 Accessing sensitive objects using JNDI

But this is extremely dangerous in some cases because it might be possible that an application has a security interface which will store the user's credentials.

Similarly in the above manner any interface can call this interface and the data from the interface can be logged easily. This creates a potential impact on the application.

```
// Getting User Name by different objects
Root.Z.B.O2(Root.Z.Cred.U-Name)
Root.Z.B.C.D.O4(Root.Z.Cred.U-Name)

// Getting Password
Root.Z.B.C.D.O5(Root.Z.Cred.Pass)
Root.Z.B.C.O3(Root.Z.Cred.Pass)
```

Log4j2 vulnerability was rated ten out of ten which is very rare and it is considered to be one of the worst vulnerabilities found in the last few years.

4.1 Should we be worried about this vulnerability?

Anyone who uses Java applications should be worried about this vulnerability because Apache Log4j2 is open source which means that anyone who uses Java applications use Apache Log4j one way or another (direct or indirect reference). Apache is very well known for their libraries for example like Tomcat, HTTP Servers, TomEE.

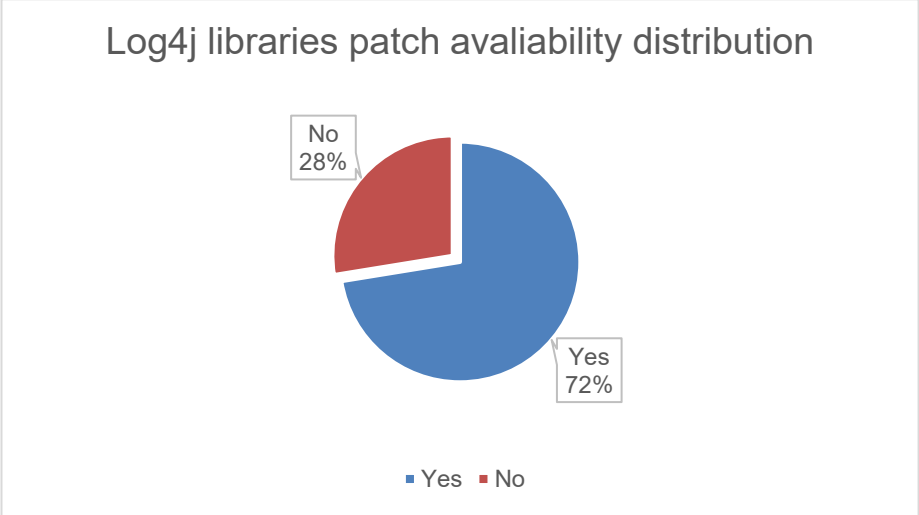


Figure 46 Patch Availability of affected Log4j Libraries till date

Appendix 1 demonstrates some of the affected applications made by some famous companies. Since these products are affected to a greater extent hence this vulnerability should be taken seriously.[13]

4.2 Scanning for affected Apache Log4j version

It is important to scan out which version of Apache Log4j is present in the operating system and whether it is affecting our system or not.

Security researchers have made many GitHub repositories and open-source programs/code/scripts/libraries so that Log4j vulnerable libraries can be detected in the system. One such tool is made by QUALYS company [18]. This tool is called “Log4jScanner”. This is only available for Windows Operating systems. It can determine the vulnerable Log4j versions like 1.x or 2.x. Further it can scan local files/folders/directories/network directories as well.

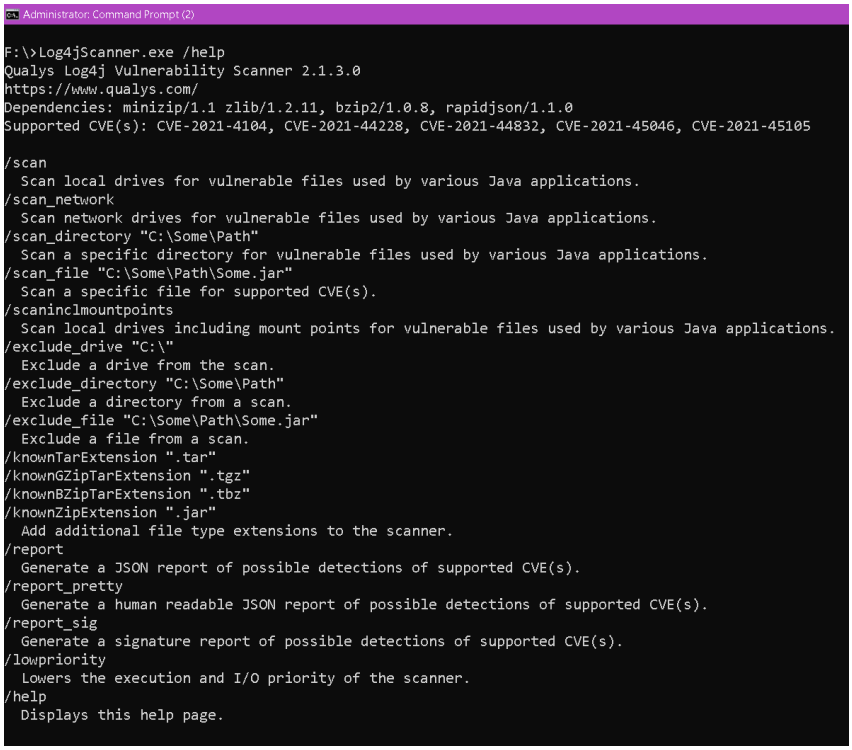


Figure 47 Log4j Scanner for windows

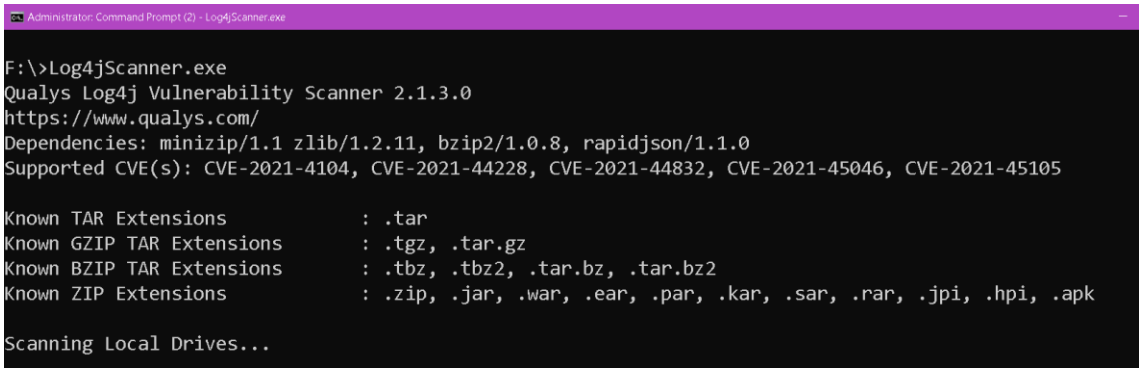


Figure 48 Scanning on windows for affected Log4j Libraries

The sample output of the scanner is listed in the Appendix 2. In the result generated by this utility, it can be seen that it reports some important details like “detectedLog4j”, “detectedLog4j1.x” ,” detectedLog4j2.x” etc. This gives proper insights as to which Log4j version is used, files affected and also what related CVE is present with that file.

4.3 Mitigation of Log4j vulnerability

Mitigation is a process which should be done immediately right after the Log4j vulnerability has been found. Basically, mitigation is not fixing the solution permanently, but it is a temporary fix by removing or replacing something which gives a problem [14].

Steps to mitigate the Log4j Vulnerability :-

Step 1. When the attacker begins the Attack, it inserts a JNDI lookup string in the header field that will be likely logged. A WAF or Web Application Firewall can be set which can filter out such queries.

Step 2. Disable Log4j and update it. The Log4j version should be identified at this stage and should be updated. Patch should be downloaded from apache.org website instead of other sources.

Step 3. Use tools and other software to identify the programs installed on the Operating System that are using Log4j to identify the version used. Update and patch them if necessary.

Step 4. Disable JNDI lookups. When the JNDI string is received by the JNDI Adapter, it passes that to the LDAP server. As a result, malicious LDAP server can be queried hence JNDI lookups should be disabled at this stage.

4.4 Affected versions of Apache Log4j

Apache Log4j 1.X and Log4j 2.x until 2.16.0 are vulnerable, every new version in Apache comes with a new CVE version number. So, **CVE-2021-44228** was the very first version in which Apache mentioned that Log4j2 2.0-beta9 version is vulnerable because of JNDI lookup features. To fix this problem Apache has disabled the “JNDI lookup” option in the latest patch version and as a result this functionality has been completely removed from this specific version.

The next affected version of Apache is Log4j 2.15.0 version and its CVE number is **CVE-2021-45046** in which certain non-default configuration fixes were incomplete and the attackers can take control over Context Map (MDC) input data when the logging configuration uses a non-default Pattern Layout. In order to fix this, Apache have introduced Log4j 2.16.0 for Java 8 version and Log4j 2.12.2 for Java 7 version.

The next version is Log4j2 version 2.0-alpha1 whose CVE number is **CVE-2021-45105.** This did not protect from uncontrolled recursion from self-referential lookups so this issue has been present in Apache version Log4j 2.17.0 and in Log4 2.12.3.

5 Common Vulnerability Scoring System

5.1 Introduction to CVSS

CVSS stands for Common Vulnerability Scoring System. As the name suggests it a Scoring System especially designed to numerically score the Vulnerabilities based on their characteristics and severity. It includes various elements and factors which can affect the scoring of a vulnerability.

The scoring of vulnerability is extremely important because this can help not only the security developers but stakeholders as well so that actions can be prioritized, and related action can be decided at earlier stage. Before CVSS many other Scoring Systems existed which sometimes were incompatible, and they generated confusion because they scored differently for a given vulnerability. As a result, CVSS was formed like a “unified scoring system” by FIRST (Forum of Incident Response and Security Teams). [15]

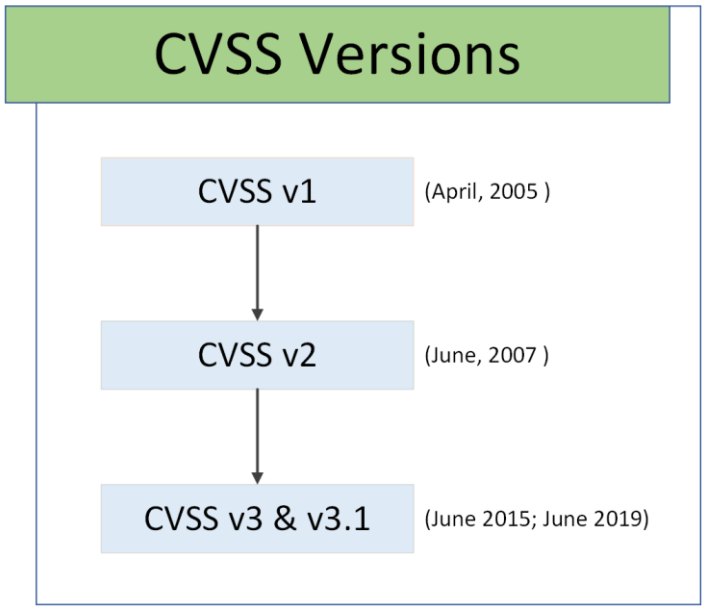


Figure 49 CVSS versions over the years

CVSS has also matured over the years from CVSS version 1 to latest CVSS v3.1.

5.2 CVSS Version 2

The first revision of CVSS i.e., CVS v1 was made public in April 2005. NAIC (National Association of Insurance Commissioners) choose FIRST to be the original developers for this system. The CVSS in its preliminary stage requested the global community to provide their feedback so that its use can be generalized and adjusted to address the inconsistencies left by previous scoring metrics.

After getting a proper evaluation from the global community, CVSS v2 was announced by FIRST and backed by Common Vulnerability Scoring System-Special Interest Group (CVSS-SIG). The version 2 was extensively revised by both the supporting organizations. It was tested and re-tested against hundreds of real-world vulnerabilities [22].

CVSS v2 calculates the score using “Metrices”. By assigning the values to these “Metrices” calculation workflow is made which gives the score based on some set of equations.

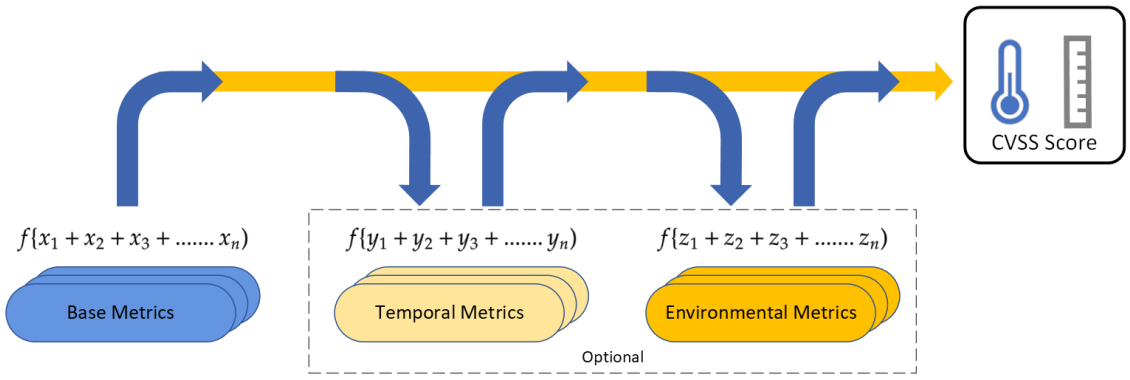


Figure 50 CVSS version 2 workflow

The CVSS v2 Score is calculated using three metrices: Base, Temporal and Environmental. All these three metrics uses logical equations with variables where the inputs are provided by output of the previous one and finally the CVSS score is calculated [21].

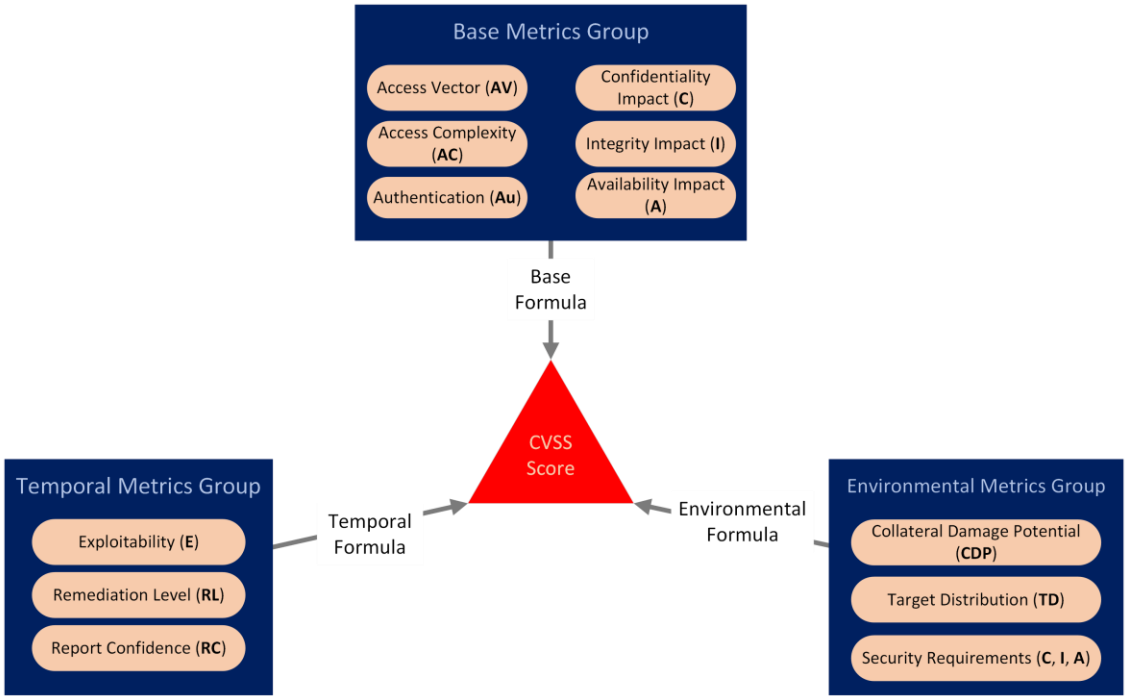


Figure 51 CVSS version 2 Metrices

5.2.1 Base Metric

The Base Metric defines the characteristics of the vulnerability which will be constant over the time and uniform with different user-environments. It is further divided into Access Vector (**AV**), Access Complexity (**AC**), Authentication (**Au**), Confidentiality Impact (**C**), Integrity Impact (**I**), Availability Impact (**A**). These are called as Base Metric Variable's and they decide how the vulnerability will be accessed and exploited, if extra conditions are required to execute the vulnerability or not. They also define if the IT asset in consideration will be directly or indirectly affected by the vulnerability.

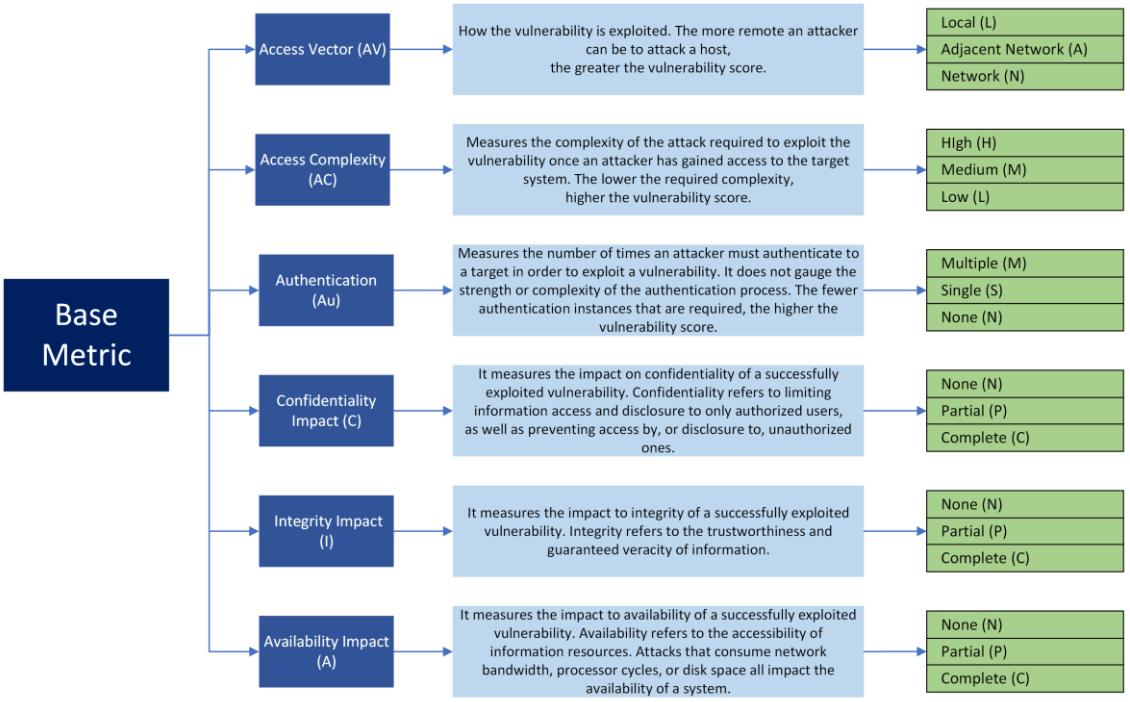


Figure 52 CVSS Version 2 Base Metric

5.2.2 Temporal Metric

The Temporal Metric defines attributes of threat which can change over time. This metric can be skipped from the final score calculation. It is further divided into Exploitability (**E**), Remediation Level (**RL**) and Report Confidence (**RC**). These are called Temporal Metric variables; they decide how the vulnerability will affect over given time frame. It is mainly concerned with three things which are: technicality of the vulnerability, can/cannot the vulnerability be fixed (if fix exist what fix is it?) and the confidence level in the existence of Vulnerability.

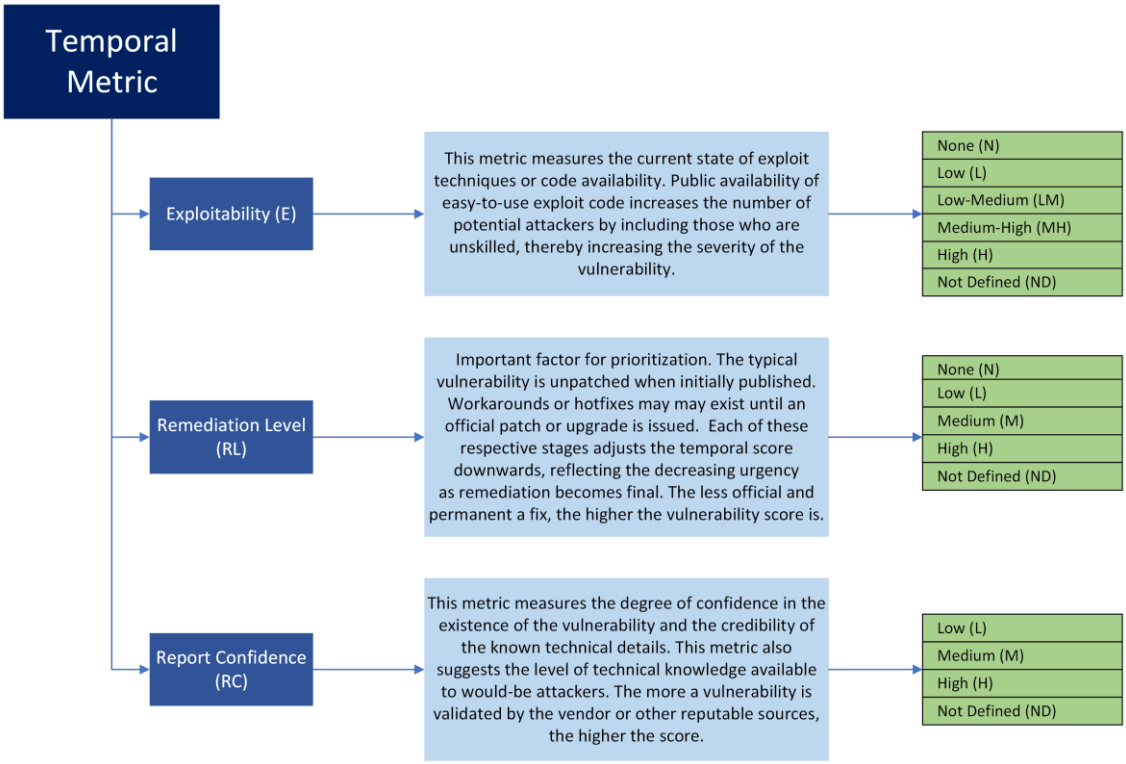


Figure 53 CVSS Version 2 Temporal Metric

5.2.3 Environmental Metric

The Environmental Metric defines the characteristics of vulnerability on the IT assets and IT environment for the user. It is also an optional metric which has no effect on the overall CVSS score. It is further divided into Collateral Damage Potential (**CDP**), Target Distribution (**TD**) and Security Requirements (**SR**).

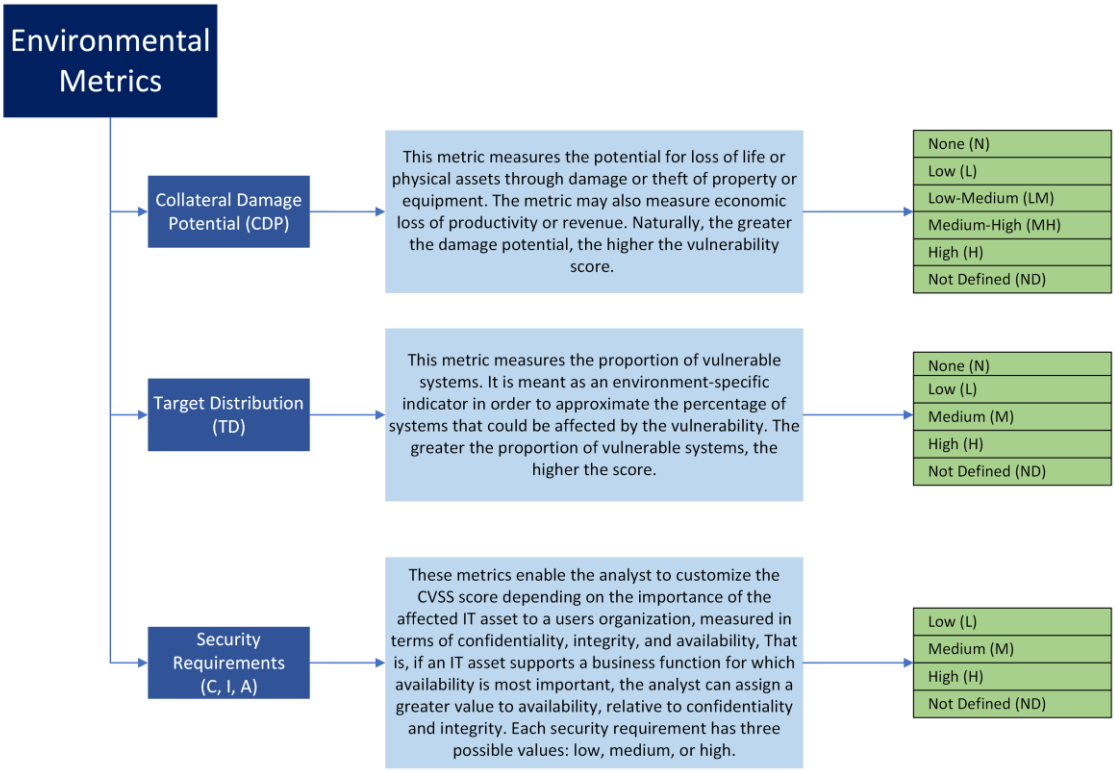


Figure 54 CVSS Version 2 Environmental Metric

5.2.4 CVSS v2 Metric Equations

CVSS v2 score is calculated by using set of Equations as set by the Metrices. The equations are defined for Base, Temporal and Environmental Metrics are called as Base Equation, Temporal Equation and Environmental Equation for respective metrices.

Base Equations is the first and primary part towards calculating the final CVSS v2 score. Base Equation uses Impact, Exploitability as input variables. Impact is calculated using Confidentiality, Integrity and Availability impact scores (calculated from these respective metrices). On the other hand, Exploitability variable is calculated using Access Vector, Access Complexity and Authentication scores.

```
BaseScore = round_to_1_decimal(((0.6*Impact)+(0.4*Exploitability)-1.5)*f(Impact))

Impact = 10.41*(1-(1-ConfImpact)*(1-IntegImpact)*(1-AvailImpact))

Exploitability = 20* AccessVector*AccessComplexity*Authentication

f(impact)= 0 if Impact=0, 1.176 otherwise

AccessVector     = case AccessVector of
                        requires local access: 0.395
                        adjacent network accessible: 0.646
                        network accessible: 1.0

AccessComplexity = case AccessComplexity of
                        high: 0.35
                        medium: 0.61
                        low: 0.71

Authentication   = case Authentication of
                        requires multiple instances of authentication: 0.45
                        requires single instance of authentication: 0.56
                        requires no authentication: 0.704

ConfImpact       = case ConfidentialityImpact of
                        none:             0.0
                        partial:          0.275
                        complete:         0.660

IntegImpact      = case IntegrityImpact of
                        none:             0.0
                        partial:          0.275
                        complete:         0.660

AvailImpact      = case AvailabilityImpact of
                        none:             0.0
                        partial:          0.275
                        complete:         0.660
```

Figure 55 CVSS Version 2 Base Metric Equation [21]

Temporal Equation will combine score from Base Metric and uses Exploitability, Remediation Level and Report Confidence metrices to calculate Temporal Score

```
TemporalScore = round_to_1_decimal(BaseScore*Exploitability
                *RemediationLevel*ReportConfidence)

Exploitability   = case Exploitability of
                       unproven:             0.85
                       proof-of-concept:     0.9
                       functional:           0.95
                       high:                 1.00
                       not defined:          1.00

RemediationLevel = case RemediationLevel of
                       official-fix:         0.87
                       temporary-fix:        0.90
                       workaround:           0.95
                       unavailable:          1.00
                       not defined:          1.00

ReportConfidence = case ReportConfidence of
                       unconfirmed:          0.90
                       uncorroborated:       0.95
                       confirmed:            1.00
                       not defined:          1.00
```

Figure 56 CVSS Version 2 Temporal Metric Equation [21]

Finally, Environmental Equation will use Temporal equation value and other metrices of Collateral Damage Protection, Target Distribution, and Security Requirements.

```
EnvironmentalScore = round_to_1_decimal((AdjustedTemporal+
(10-AdjustedTemporal)*CollateralDamagePotential)*TargetDistribution)

AdjustedTemporal = TemporalScore recomputed with the BaseScore's Impact sub-
equation replaced with the AdjustedImpact equation

AdjustedImpact = min(10,10.41*(1-(1-ConfImpact*ConfReq)*(1-IntegImpact*IntegReq)
                    *(1-AvailImpact*AvailReq)))

CollateralDamagePotential = case CollateralDamagePotential of
                                none:           0
                                low:            0.1
                                low-medium:     0.3
                                medium-high:    0.4
                                high:           0.5
                                not defined:    0

TargetDistribution        = case TargetDistribution of
                                none:           0
                                low:            0.25
                                medium:         0.75
                                high:           1.00
                                not defined:    1.00

ConfReq          = case ConfReq of
                        low:            0.5
                        medium:         1.0
                        high:           1.51
                        not defined:    1.0

IntegReq         = case IntegReq of
                        low:            0.5
                        medium:         1.0
                        high:           1.51
                        not defined:    1.0

AvailReq         = case AvailReq of
                        low:            0.5
                        medium:         1.0
                        high:           1.51
                        not defined:    1.0
```

Figure 57 CVSS Version 2 Environmental Metric Equation [21]

5.3 CVSS Version 3.1

CVSS 3 and 3.1 were launched in June 2015 and June 2019 respectively. CVSS 3.1 is the most recent and updated version of the CVSS. CVSS version 3 was developed to improve the shortcomings present in the CVSS version 2 which were identified at later stage. Version 2 assumed that user will have complete knowledge about the vulnerability in consideration which is incorrect. Several Metrices like Attack Vector, Authentication and so forth which did not fit properly for many new vulnerabilities discovered at later stages, as a result their CVSS

score can not be calculated. Hence CVSS 2 was misleading in many cases for newer vulnerabilities [22].

Improvements are made in CVSS 3 and 3.1 to address these shortcomings. In version 3.1 [24] three major metric groups remain unchanged, although changes were made inside these metrices: -

1. In the Base Metric, attributes of Confidentiality, Integrity and Availability were reassigned as None, Low and High [23]
2. Attack Vector inside Base Metric has a new attribute called as Physical (P)
3. In Base Metric, new metrics: User Interaction (UI) and Privilege Required (PR) and Scope (S) were added. Also Access Complexity (AC) was renamed to Attack Complexity (AC).
4. Inside Temporal Metric Group, Exploit Code Maturity (ECM) was added, and Exploitability (E) was removed.
5. Environmental Metric Group was completely changed. It now has Modified Base Metrices (MBM), Confidentiality Requirement (CR), Integrity Requirement (IR) and Availability Requirement (AR).

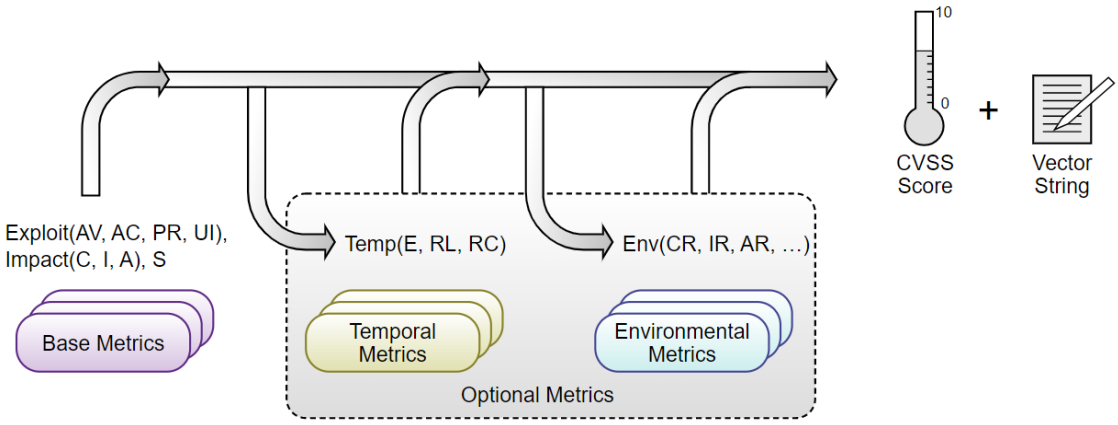


Figure 58 CVSS Version 3.1 workflow[24]

5.3.1 Base Metric Group

The Base metric group for CVSS 3.1 is like that of CVSS version 2 with some slight differences. Three new metrices are added which are Privilege Requirement (PR), User Interaction (UI), and Scope (S). These were added to

remove the inconsistencies from CVSS version 2 for some vulnerabilities that were discovered at later stage [23].

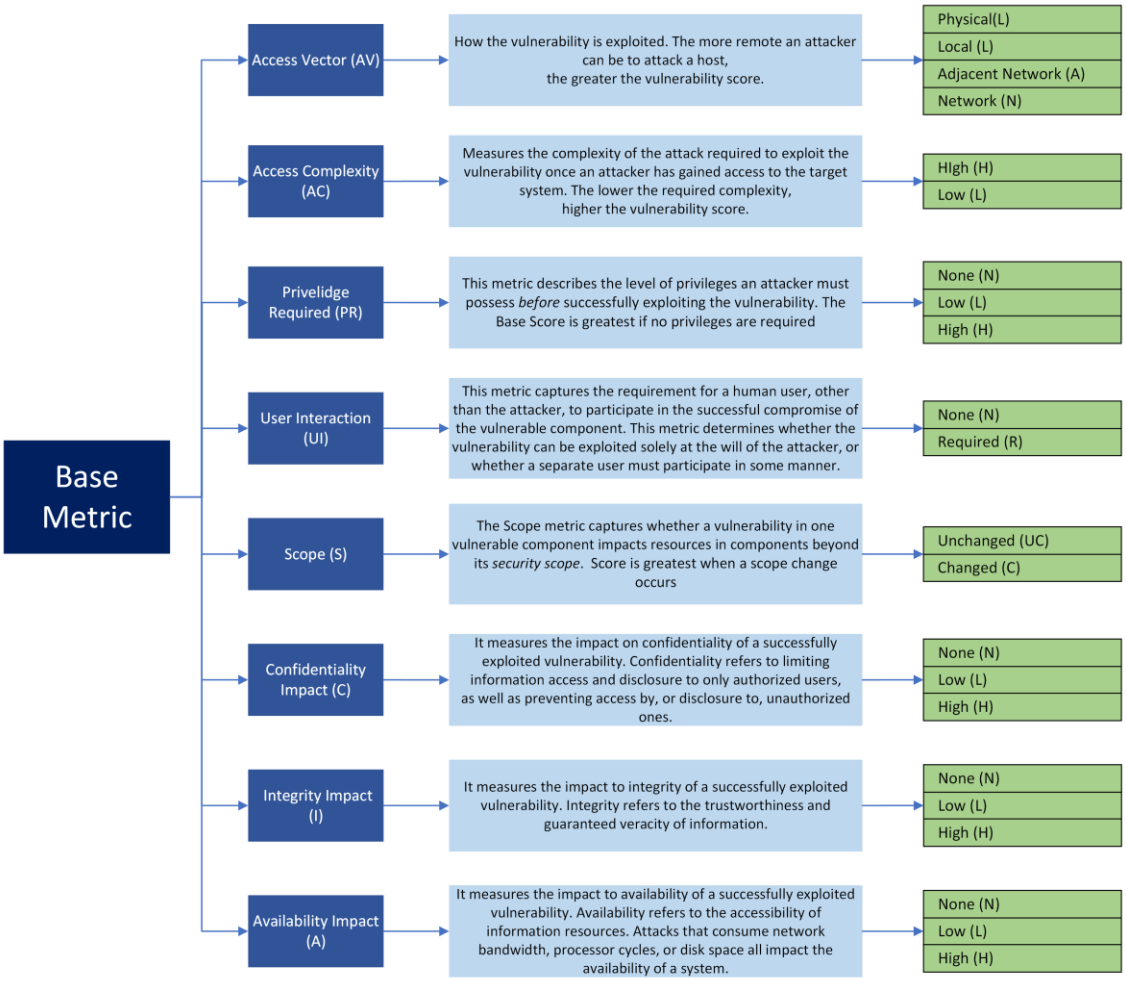


Figure 59 CVSS Version 3.1 Base Metric[24]

5.3.2 Temporal Metric Group

The Temporal Metric Group also has some changes in CVSS version 3 as compared to version 2. Exploitability (E) was removed, and Exploit Code Maturity (ECM) was added [24]. This was done because some attack-methods of some vulnerabilities may improve over time hence they can become more dangerous in the Future.

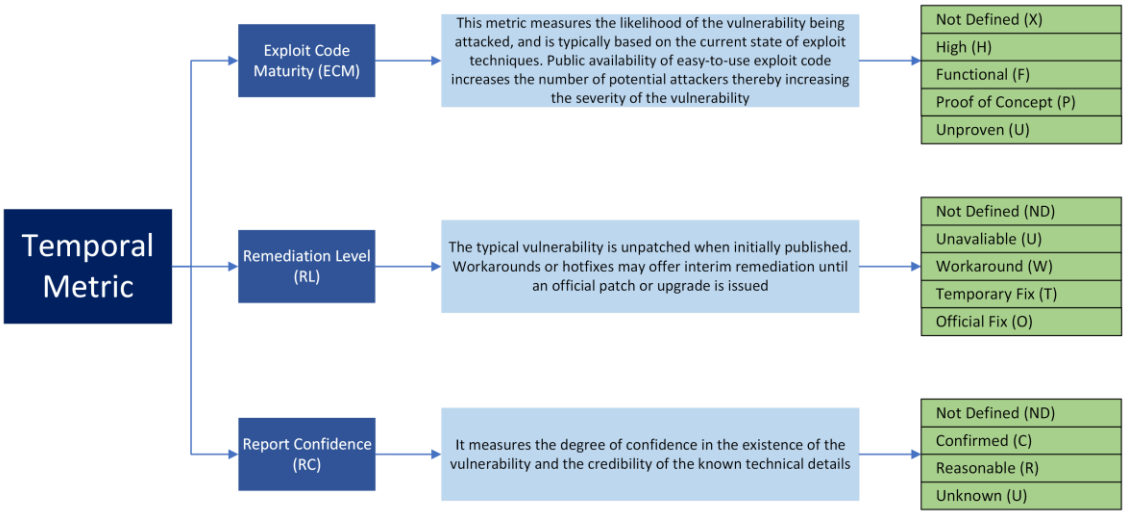


Figure 60 CVSS Version 3.1 Temporal Metric

5.3.3 Environmental Metric Group

The Environmental Metric Group has some significant changes from version 2. Modified Base Metrices are also included [24]. This was done because some/all metrices from the Base Metrices can affect the Environmental hence those metrices are also considered again in this metric group.

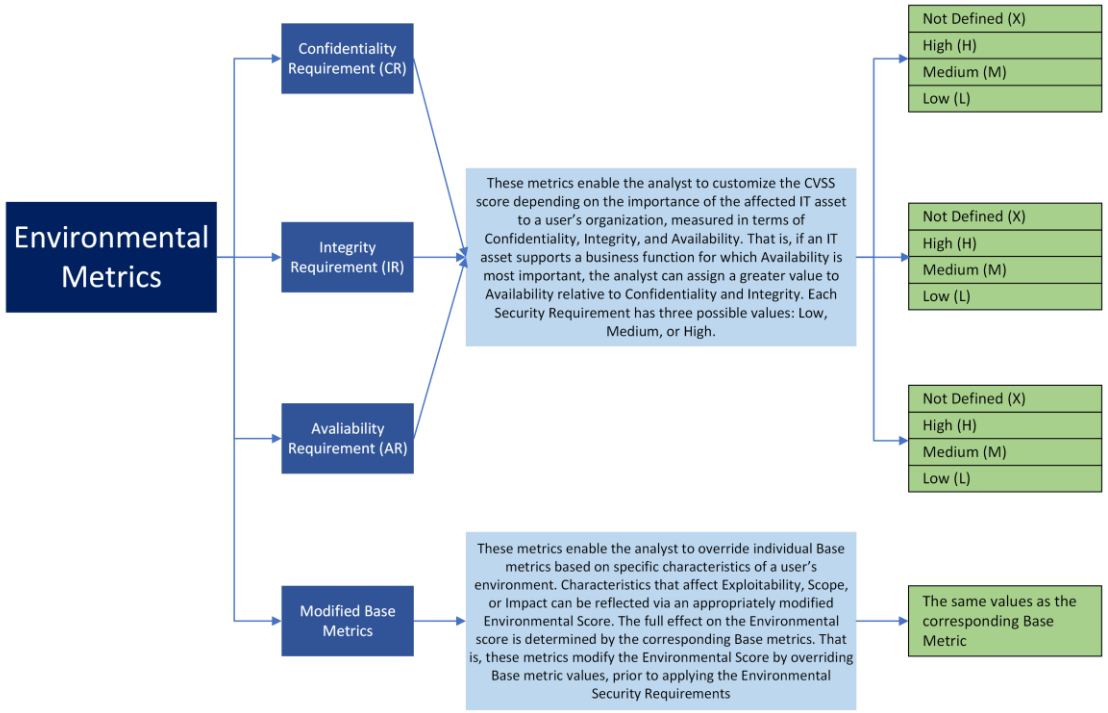


Figure 61 CVSS Version 3.1 Environmental Metric

5.3.4 CVSS v3.1 Metric Equations

Similarly, like version 2, CVSS version 3 employs more set of complex equations to calculate the Vulnerability Score. The feedback from the metric serves as an input for next step, hence the feedback is more tightly bound for the final CVSS value.

Unlike CVSS 2, The Base Metric sore calculated in First step serve as input again for Environmental if the vulnerability is expected to affect the metrics in this group.[25]

Base

The Base Score is a function of the Impact and Exploitability sub score equations. Where the Base score is defined as,

If (Impact sub score <= 0) *0 else,*
Scope Unchanged[4] $Roundup(Minimum[(Impact + Exploitability), 10])$
Scope Changed $Roundup(Minimum[1.08 \times (Impact + Exploitability), 10])$

and the Impact sub score (ISC) is defined as,

Scope Unchanged $6.42 \times ISC_{\text{Base}}$
Scope Changed $7.52 \times [ISC_{Base} - 0.029] - 3.25 \times [ISC_{Base} - 0.02]^{15}$

Where,

$ISC_{Base} = 1 - [(1 - Impact_{Conf}) \times (1 - Impact_{Integ}) \times (1 - Impact_{Avail})]$

And the Exploitability sub score is,

$8.22 \times AttackVector \times AttackComplexity \times PrivilegeRequired \times UserInteraction$

Temporal

The Temporal score is defined as,

$Roundup(BaseScore \times ExploitCodeMaturity \times RemediationLevel \times ReportConfidence)$

Environmental

The environmental score is defined as,

If (Modified Impact Sub score <= 0) 0 else,

If Modified Scope is Unchanged Round up(Round up (Minimum [(M.Impact + M.Exploitability) ,10]) × Exploit Code Maturity × Remediation Level × Report Confidence)

If Modified Scope is Changed Round up(Round up (Minimum [1.08 × (M.Impact + M.Exploitability) ,10]) × Exploit Code Maturity × Remediation Level × Report Confidence)

And the modified Impact sub score is defined as,

If Modified Scope is Unchanged $6.42 \times [ISC_{Modified}]$

If Modified Scope is Changed $7.52 \times [ISC_{Modified} - 0.029] - 3.25 \times [ISC_{Modified} \times 0.9731 - 0.02]^{13}$

Where,
$ISC_{Modified} = Minimum\ [[1 - (1 - M.IConf \times CR) \times (1 - M.IInteg \times IR) \times (1 - M.IAvail \times AR)], 0.915]$

The Modified Exploitability sub score is,

$8.22 \times M.AttackVector \times M.AttackComplexity \times M.PrivilegeRequired \times M.UserInteraction$

4 Where "Round up" is defined as the smallest number, specified to one decimal place, that is equal to or higher than its input. For example, Round up (4.02) is 4.1; and Round up (4.00) is 4.0.

Figure 62 CVSS Version 3.1 Equations [25]

Metric	Metric Value	Numerical Value
Attack Vector / Modified Attack Vector	Network	0.85
	Adjacent	0.62
	Local	0.55
	Physical	0.2
Attack Complexity / Modified Attack Complexity	Low	0.77
	High	0.44
Privileges Required / Modified Privileges Required	None	0.85
	Low	0.62 (or 0.68 if Scope / Modified Scope is Changed)
	High	0.27 (or 0.5 if Scope / Modified Scope is Changed)
User Interaction / Modified User Interaction	None	0.85
	Required	0.62
Confidentiality / Integrity / Availability / Modified Confidentiality / Modified Integrity / Modified Availability	High	0.56
	Low	0.22
	None	0
Exploit Code Maturity	Not Defined	1
	High	1
	Functional	0.97
	Proof of Concept	0.94
	Unproven	0.91
Remediation Level	Not Defined	1
	Unavailable	1
	Workaround	0.97
	Temporary Fix	0.96
	Official Fix	0.95
Report Confidence	Not Defined	1
	Confirmed	1
	Reasonable	0.96
	Unknown	0.92
Confidentiality Requirement / Integrity Requirement / Availability Requirement	Not Defined	1
	High	1.5
	Medium	1
	Low	0.5

Figure 63 CVSS Version 3.1 Metric Values

5.4 CVSS Score calculation of Log4j

Log4j was also evaluated using CVSS version 2 and CVSS version 3.1. The CVSS version 2 and version 3.1 scores were calculated using official NIST CVSS calculators available online.[26]

Results for CVSS version 2: -

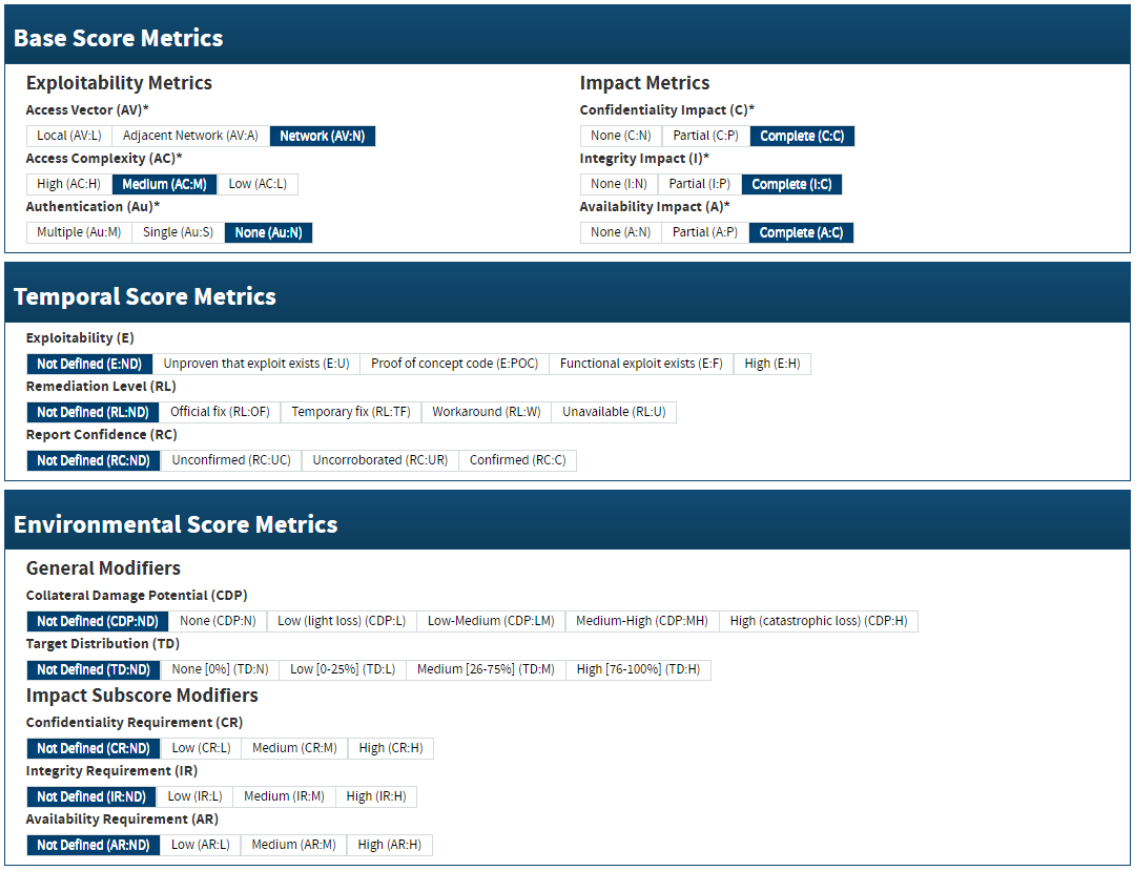


Figure 64 CVSS Version 2 attribute selection [27]

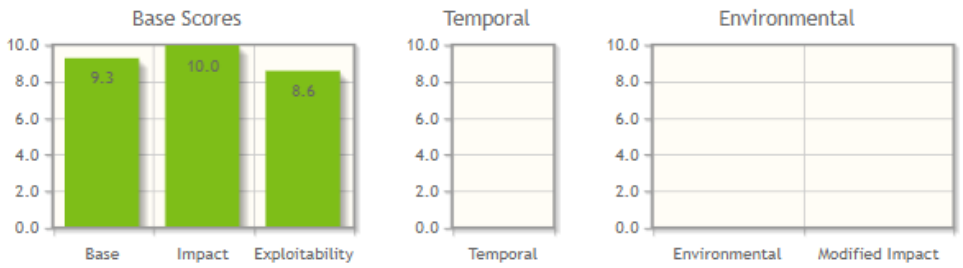


Figure 65 CVSS Version 2 metrices scores [27]

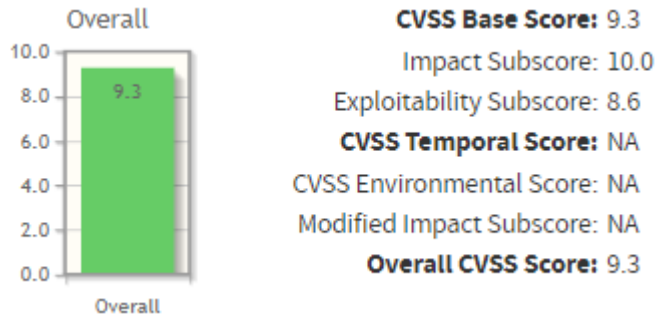


Figure 66 CVSS Version 2 final score

Results for CVSS version 3.1: -

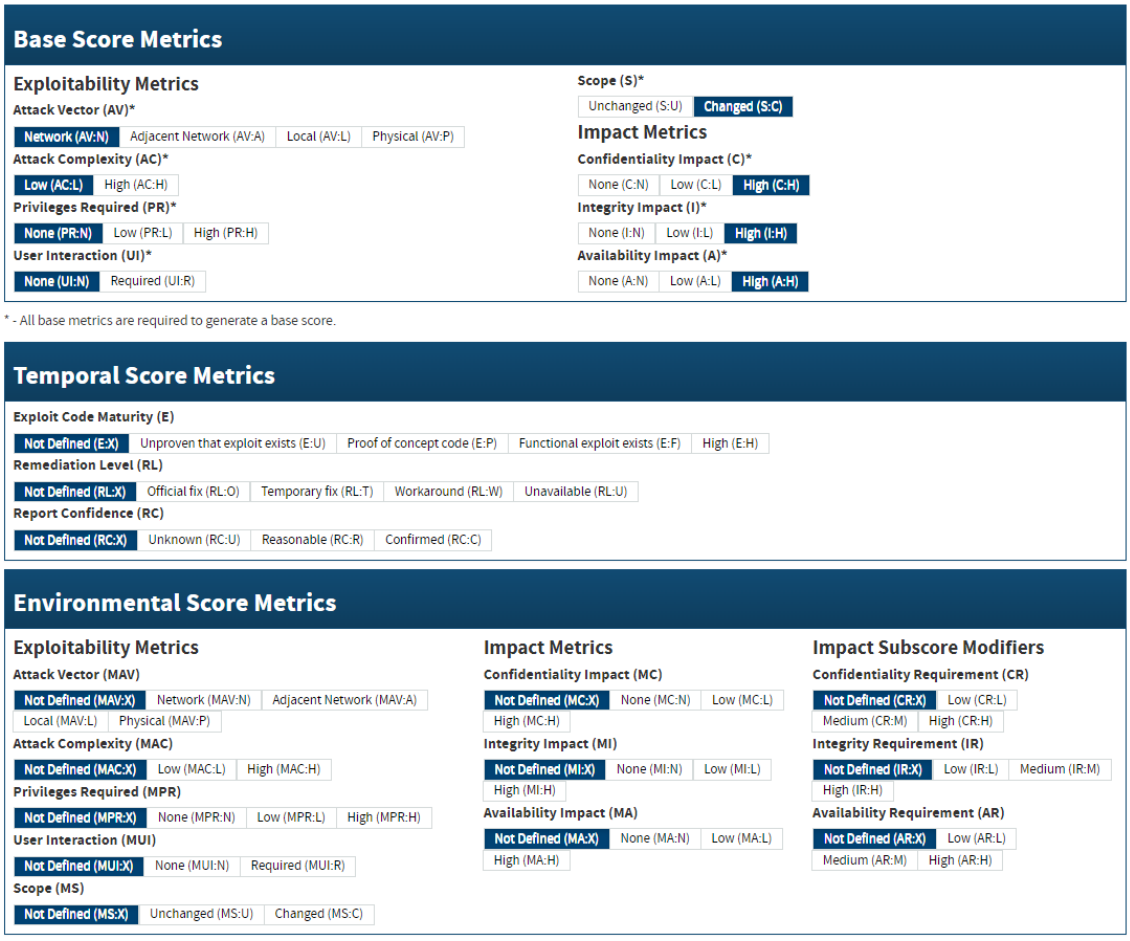


Figure 67 CVSS Version 3.1 attribute selection [28]

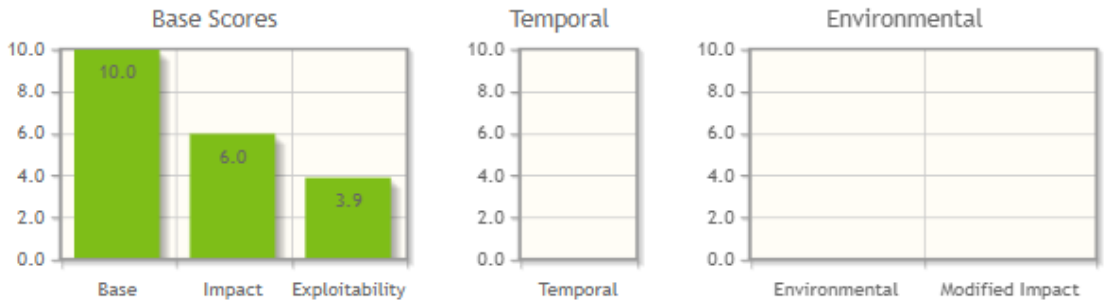


Figure 68 CVSS Version 3.1 metrices score [28]

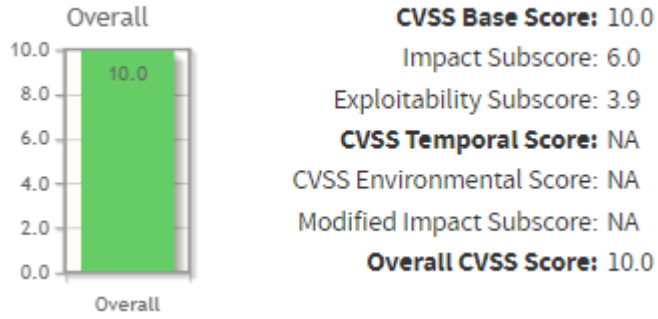


Figure 69 CVSS Version 3.1 final score

5.5 CVSS analysis of Log4j

The CVSS scores for Log4j using version 2 and version 3.1 were completely different. The CVSS version 2 score is 9.3 and VCSS version 3 score is 10 for the vulnerability in consideration [26].

CVSS Version	Attack Vectors	Score
Version 2	AV:N/AC:M/Au:N/C:C/I:C/A:C	9.3
Version 3.1	AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H	10

Figure 70 CVSS score for Log4j

For CVSS Version 2 the following observations were made on which the score was calculated: -

- Access Vector was set to “Network” because the Log4j affects the whole Network, so it means that Log4j doesn’t require local network access to execute itself. It can be done remotely as well.
- Attack Complexity was set to “Medium” because the attacker needs some technical knowledge to execute the Log4j attack.
- Authentication was set to “None” because no authentication is required to exploit this vulnerability.
- Confidentiality Impact was set to “Complete” because this exploit will completely disclose the confidential information which was only meant to be seen/used by authorized users.
- Integrity Impact was set to “Complete” because whole system will be compromised by this vulnerability as the attacker will have full access to the system being attacked.
- Availability Impact was set to “Complete” because this exploit will result in complete availability of all the resources/documents and so forth.
- For Temporal Metric, Exploitability was set to “Not Defined” because no proper exploit techniques or codebase was available to sufficiently execute the exploit. Remediation Level was set to “Not Defined” because no fix or workaround was present when this vulnerability was discovered.
- Finally for the Environmental Metric, Collateral Damage Protection, Target Distribution and Security Requirements were all set to “Not Defined” because the Log4j vulnerability predicted no physical damage like loss of life or property or asset, not one single target system exist but multiple.

For CVSS version 3 the following conclusions were made when the score for version 3.1 was calculated: -

- Attack Network was set to “Network” with the similar ideology for previous version.

- Attack Complexity was set to “Low” because no special conditions exist to execute the attack, also the attack can be repeatedly made with full success on the same system.
- Privileges Required was set to “None” because no authorization or special privileges were needed to perform a successful attack.
- User Interaction was set to “None” because the attack can be carried out in the background and no user interaction is needed.
- Scope was set to “Changed” because the resources affected can be out-of-scope for the Security Authority in place. The Security Authority in force cannot apprehend the resources accessed while vulnerability is in action.
- Confidentiality Impact, Integrity Impact and Availability Impact were all set to “High” because the resources or system affected will loose confidential information, the trust from the affected system will reduce and the sensitive information will be fully available.
- Exploit Code Maturity, Remediating Level and Report Confidence were all set to “Not-Defined” because other values for these attributes cannot be used.
- Similarly for the Environmental Metrics, Modified Base Metric, Impact Metric were all set to “Not-defined” because no proper information was available.

6 Conclusion

In its current state Log4j is a harmful library due to vulnerabilities allowing malicious users access to the system using Log4j. As demonstrated from the example attack, it can be concluded that any simple internet user with basic knowledge about programming and IT skills can perform this attack and wreak havoc on the system. What makes this vulnerability more dangerous is that “NO” permanent fix exists till date, although several patches have been prepared, workarounds have been made and distributed, but they are not permanent fixes. Official authorities have made out statements clearly saying to either disable this logging feature or to completely remove this from the system. Java is widely used programming and almost 6 billion devices run on Java, so it’s important for everyone including organizations and end-users to identify if their applications are using Log4j or not. This vulnerability scored 9.3 on CVSS version which is “HIGH” and 10 on CVSS version 3.1 which is “CRITICAL” with several metrics set to “NONE” or “NOT-DEFINED” means some research is needed and hence it should be taken more seriously [29].

There are some open-source tools which exist on open-source forums like GitHub which provide Log4j scanners so that end-users can also scan their system for their affected applications [30]. Surprisingly, some applications were also found to be affected by this vulnerability on the author’s machine like: Fiji app (used to view 3d model/image of Human cells) and IBM SPSS statistics (used for mathematical modelling, statistics, and analysis).

Security seminars should be conducted, blog posts should be made and social media engagement about this vulnerability should be done. Further, a joint security research should be conducted about this vulnerability. Some alternatives to Log4j also exist like SLF4j, LogBACK and so forth, but they try to use Log4j-core in direct or indirect manner hence these also cannot be used now. A quick and permanent fix should be developed as soon as possible before any major cyber-incident happens.

References

1. Alibabacloud.com. 2022. Security Advisory on Apache Log4j 2 RCE Vulnerability (CVE-2021-44228). [online] Available at: <https://www.alibabacloud.com/notice/Log4j2>.

2. Understanding the Impact of Apache Log4j Vulnerability [Internet]. Google Online Security Blog. [cited 2022 Jan 6]. Available from: https://security.googleblog.com/2021/12/understanding-impact-of-apache-Log4j.html

3. Apache Log4j 2 Vulnerability Security Advisory [Internet]. Google Cloud. [cited 2022 Feb 8]. Available from: https://cloud.google.com/Log4j2-security-advisory

4. Maven – POM Reference [Internet]. maven.apache.org. [cited 2022 Feb 8]. Available from: https://maven.apache.org/pom.html

5. Tutorialspoint.com. n.d. Log4j Tutorial. [online] Available at: <https://www.tutorialspoint.com/Log4j/index.htm> [Accessed 17 March 2022].

6. Welcome to Log4j 2! [Internet]. Log4j – Overview - Apache Log4j 2. Apache Software Foundation; [cited 2022Mar18]. Available from: https://logging.apache.org/log4j/log4j-2.12.4/manual/index.html

7. Magaji J. Log4Net and Log4j [Internet]. SolarWinds Worldwide; [cited 2022Mar18]. Available from: https://www.papertrail.com/solution/guides/log4net-and-log4j/

8. CVE-2021-44228 - Log4j - MINECRAFT VULNERABLE! (and SO MUCH MORE) [Internet]. www.youtube.com. Available from: https://www.youtube.com/watch?v=7qoPDq41xhQ&ab_channel=JohnHammond

9. Microsoft. IMPORTANT MESSAGE: SECURITY VULNERABILITY IN JAVA EDITION [Internet]. Microsoft Minecraft Staff; 2021 [cited 2022Mar18]. Available from: https://www.minecraft.net/en-us/article/important-message--security-vulnerability-java-edition

10. About SpigotMC [Internet]. spigotmc.org. 2022 [cited 3 April 2022]. Available from: https://www.spigotmc.org/wiki/about-spigot/

11. Lesson: Overview of JNDI (The JavaTM Tutorials > Java Naming and Directory Interface) [Internet]. docs.oracle.com. Available from: https://docs.oracle.com/Javase/tutorial/jndi/overview/index.html

12. http — HTTP modules — Python 3.10.5 documentation [Internet]. docs.python.org. [cited 2022 June 4]. Available from: https://docs.python.org/3/library/http.html

13. Tekslate. JNDI Startup and Shutdown Classes [Internet]. Tekslate; 2018 [cited 2022Jun5]. Available from: https://tekslate.com/jndi-startup-shutdown-classes

14. Duraisamy R, Verma A, Ang MC, Surana N. Patch Now: Apache Log4j Vulnerability Called Log4Shell Actively Exploited [Internet]. Trend Micro; 2021 [cited 2022Jun5]. Available from: https://www.trendmicro.com/en_us/research/21/l/patch-now-apache-log4j-vulnerability-called-log4shell-being-acti.html

15. Gupta H, Chaudhary A, Kumar A. Identification and analysis of LOG4J vulnerability. 2022 11th International Conference on System Modeling & Advancement in Research Trends (SMART). 2022;

16. Maven – POM Reference [Internet]. maven.apache.org. Available from: https://maven.apache.org/pom.html

17. Key K. Critical Apache Log4j2 Exploit Demonstrated in Minecraft [Internet]. PCMag UK. 2021 [cited 2022 June 6]. Available from: https://uk.pcmag.com/security/137653/critical-apache-Log4j2-exploit-demonstrated-in-minecraft

18. Github Download Links [Internet]. GitHub. 2022 [cited 2022 June 19]. Available from: https://github.com/Qualys/Log4jscanwin

19. Constantin L. 4 ways to properly mitigate the Log4j vulnerabilities (and 4 to skip) [Internet]. CSO Online. 2021. Available from: https://www.csoonline.com/article/3645348/how-to-properly-mitigate-the-Log4j-vulnerabilities.html

20. Mell P, Scarfone K. CVSS V2 history [Internet]. FIRST. National Institute of Standards and Technology; 2013 [cited 2022Nov18]. Available from: https://www.first.org/cvss/v2/history

21. Mell P, Scarfone K, Romanosky S. CVSS V2 Complete Documentation [Internet]. FIRST. Forum of Incident Response and Security Teams; [cited 2022Nov18]. Available from: https://www.first.org/cvss/v2/guide

22. Team F. Understanding CVSSV2 and CVSSV3: How It Works and their shortcomings [Internet]. Flashpoint. flashpoint.io; 2022 [cited 2022Nov18]. Available from: https://flashpoint.io/blog/understanding-cvssv2-and-cvssv3/

23. CVSS v2 vs CVSS V3 [Internet]. Balbix. Balbix; 2022 [cited 2022Nov18]. Available from: https://www.balbix.com/insights/cvss-v2-vs-cvss-v3/

24. CVSS v3.1 Specification Document [Internet]. FIRST. Forum of Incident Response and Security Teams; [cited 2022Nov18]. Available from: https://www.first.org/cvss/v3.1/specification-document

25. CVSS v3 equations [Internet]. NVD. National Vulnerability Database (NVD) | NIST; [cited 2022Nov18]. Available from: https://nvd.nist.gov/vuln-metrics/cvss/v2-calculator/equations

26. CVE-2021-44228 Detail [Internet]. NVD. National Vulnerability Database (NVD) | NIST; [cited 2022Nov18]. Available from: https://nvd.nist.gov/vuln/detail/CVE-2021-44228

27. Common Vulnerability Scoring System Calculator CVE-2021-44228 [Internet]. NVD. National Vulnerability Database (NVD) | NIST; [cited 2022Nov18]. Available from: https://nvd.nist.gov/vuln-metrics/cvss/v2-calculator?name=CVE-2021-44228&vector=%28AV%3AN%2FAC%3AM%2FAu%3AN%2FC%3AC%2FI%3AC%2FA%3AC%29&version=2.0&source=NIST

28. Common Vulnerability Scoring System Calculator CVE-2021-44228 [Internet]. NVD. National Vulnerability Database (NVD) | NIST; [cited 2022Nov18]. Available from: https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?name=CVE-2021-44228&vector=AV%3AN%2FAC%3AL%2FPR%3AN%2FUI%3AN%2FS%3AC%2FC%3AH%2FI%3AH%2FA%3AH&version=3.1&source=NIST

29. Feng S, Lubis M. Defense-in-depth security strategy in LOG4J vulnerability analysis. 2022 International Conference Advancement in Data Science, E-learning and Information Systems (ICADEIS). 2022;

30. Sopariwala S, Fallon E, Asghar MN. Log4jPot: Effective Log4Shell Vulnerability Detection System. 2022 33rd Irish Signals and Systems Conference (ISSC). 2022;

Appendix 1: Different Software affected by Log4j

Vendor	Software	Patch_avaliable
Alertus	Alertus Console	Yes
Amazon Web Services	Amazon Linux AMI	Yes
Amazon Web Services	Amazon Linux AMI	Yes
Amazon Web Services	Amazon Linux AMI	Yes
Apache Foundation	Apache Log4j	Yes
Apache Foundation	Flink	Yes
Apache Foundation	Apache Solr	No
Apache Foundation	Apache Tika	Yes
Apache Foundation	Apache Archiva	Yes
Apache Foundation	Apache Calcite Avatica	Yes
Apache Foundation	Apache EventMesh	No
Apache Foundation	Apache Druid	Yes
Apache Foundation	Apache Fortress	Yes
Apache Foundation	Apache Geode	Yes
Apache Foundation	Apache Hive	No
Apache Foundation	Jena	Yes
Apache Foundation	Apache JMeter	Yes
Apache Foundation	OFBiz	Yes
Apache Foundation	Apache Ozone	No
Apache Foundation	SkyWalking	Yes
Apache Foundation	Apache Traffic Control	Yes
Apache Foundation	Apache Nifi	Yes
Apache Foundation	Apache Tapestry	No
Apache Foundation	Apache Spark	Yes
Apereo Foundation	Apereo CAS	Yes
Apereo Foundation	Opencast	Yes
Apple Inc.	Apple Xcode	Yes
arduino	Arduino IDE	Yes
Arista Networks	CloudVision Portal	No
Atlassian	Bitbucket Server	No
BENTLEY SYSTEMS	SYNCHRO 4D Pro	Yes
Brian Pangburn	SwingSet	Yes
Broadcom	Symantec Advanced Authentication	No
Broadcom	Symantec Endpoint Protection Manager	No
Canonical Ltd.	libLog4j2-Java (Ubuntu package)	Yes
Canonical Ltd.	libLog4j2-Java (Ubuntu package)	Yes

Canonical Ltd.	libLog4j2-Java (Ubuntu package)	Yes
Clavister	InCenter	No
Cloud Foundry Foundation	CF Deployment	Yes
cPanel, Inc	cPanel	No
Debian	apache-Log4j2 (Debian package)	Yes
Dell	EMC NetWorker Server	No
Dell	Dell NetWorker Virtual Edition	No
Dell	Dell EMC Unity Operating Environment (OE)	Yes
Dell	Storage Center - Dell Storage Manager	Yes
Dell	Connectrix MDS-DCNM	Yes
Dell	Nutanix AOS	Yes
Dell	Nutanix Objects	Yes
Dell	Dell EMC Unisphere Central	Yes
Evaluating Variations in Language Laboratory	The Java Graphical Authorship Attribution Program	Yes
F-Secure	F-Secure Policy Manager	Yes
F-Secure	F-Secure Policy Manager for Linux	Yes
F-Secure	F-Secure Policy Manager Proxy	Yes
F-Secure	F-Secure Policy Manager Proxy for Linux	Yes
F-Secure	F-Secure Endpoint Proxy	Yes
FileCap	FileCap Server	Yes
Fortinet, Inc	FortiSIEM	No
Fortinet, Inc	FortiCASB	No
Fortinet, Inc	FortiPortal	No
Fortinet, Inc	FortiNAC	No
Fortinet, Inc	FortiConverter	No
Fortinet, Inc	FortiAIOps	No
Fortinet, Inc	FortiSOAR	No
GitHub	GitHub Enterprise Server	Yes
Gradle	Gradle Enterprise	Yes
Gradle	Gradle Enterprise Test Distribution Agent	Yes

Gradle	Build Cache Node	Yes
Graylog	Graylog	Yes
HMS Networks	eCatcher	Yes
IBM Corporation	IBM WebSphere Application Server	Yes
IBM Corporation	IBM Spectrum Protect Plus	Yes
IBM Corporation	IBM Cloud Application Business Insights	Yes
IBM Corporation	IBM Spectrum Protect Snapshot for VMware	Yes
IBM Corporation	IBM Cognos Controller	Yes
IBM Corporation	IBM Cognos Analytics	Yes
IBM Corporation	IBM Spectrum Copy Data Management	Yes
IBM Corporation	IBM Business Automation Workflow	Yes
IBM Corporation	Netcool/OMNIbus	Yes
IBM Corporation	IBM Edge Application Manger	Yes
IBM Corporation	IBM Resilient SOAR	Yes
IBM Corporation	Jazz for Service Management	Yes
IBM Corporation	IBM DB2	Yes
IBM Corporation	IBM Watson Assistant for IBM Cloud Pak for Data	Yes
IBM Corporation	SPSS Statistics Subscription	Yes
IBM Corporation	IBM SPSS Statistics Server	Yes
IBM Corporation	IBM DCNM	Yes
IBM Corporation	IBM Spectrum Scale for IBM Elastic Storage Server	Yes
IBM Corporation	Netcool Operations Insight	Yes
IBM Corporation	IBM Watson Knowledge Catalog in Cloud Pak for Data	Yes
IBM Corporation	Industry Models – IBM Data Model For Energy and Utilities	Yes
IBM Corporation	Industry Models – IBM Unified Data Model for Healthcare	Yes

IBM Corporation	Industry Models – IBM Banking and Financial Markets Data Warehouse	Yes
IBM Corporation	Industry Models – IBM Insurance Information Warehouse	Yes
IBM Corporation	Crypto Hardware Initialization and Maintenance (CHIM)	Yes
IBM Corporation	IBM Sterling Connect:Direct for zOS	Yes
IBM Corporation	IBM Tivoli Monitoring	Yes
IBM Corporation	IBM Sterling Transformation Extender	Yes
IBM Corporation	IBM Sterling B2B Integrator	Yes
IBM Corporation	IBM Spectrum Symphony	Yes
IBM Corporation	IBM Spectrum Conductor	Yes
IBM Corporation	IBM Sterling File Gateway	Yes
IBM Corporation	Financial Transaction Manager for Digital Payments (DP)	Yes
IBM Corporation	Financial Transaction Manager for Corporate Payment Services (CPS)	Yes
IBM Corporation	Financial Transaction Manager for ACH Services and Check Services	Yes
IBM Corporation	IBM Spectrum Control	Yes
IBM Corporation	Operational Decision Manager	Yes
IBM Corporation	Rational Test Automation Server	Yes
IBM Corporation	Cloud Pak for Security (CP4S)	Yes
IBM Corporation	RSA RT	Yes
IBM Corporation	IBM Global High Availability Mailbox	Yes
IBM Corporation	Log Analysis	Yes
IBM Corporation	Log Analysis	Yes

IBM Corporation	IBM Informix Dynamic Server on Cloud Pak for Data	Yes
IBM Corporation	Informix Dynamic Server	Yes
IBM Corporation	IBM UrbanCode Release	Yes
IBM Corporation	IBM Cloud Pak System	Yes
IBM Corporation	IBM TRIRIGA	Yes
IBM Corporation	IBM PureData System for Operational Analytics	Yes
IBM Corporation	IBM DS8000 Hardware Management Console	Yes
IBM Corporation	IBM DB2	Yes
IBM Corporation	IBM Tivoli Netcool/OMNIbus Integration – Java Netcool Utility Library	Yes
Intel	Intel Audio Development Kit	No
Intel	Intel Datacenter Manager	No
Intel	Intel oneAPI sample browser plugin for Eclipse	No
Intel	Intel System Debugger	No
Intel	Intel Secure Device Onboard	No
Ivanti	Avalanche	Yes
Ivanti	Ivanti File Director	No
Ivanti	MobileIron Core	No
Ivanti	MobileIron Sentry	No
Ivanti	MobileIron Core Connector	No
Ivanti	Reporting Database (RDB)	No
JetBrains s.r.o.	YouTrack	Yes
JetBrains s.r.o.	Hub	Yes
Johnson Controls	exacqVision Enterprise System Manager	Yes
Johnson Controls	PowerManage	Yes
Karsten Hahn	PortEx	Yes
LiveAction	LiveNA	Yes
LiveAction	LiveNX	Yes
LucaNet	LucaNet	Yes
Metabase	Metabase	Yes

Microsoft	Kafka Connect for Azure Cosmos DB	Yes
Mojang Studios	Minecraft	Yes
MongoDB, Inc.	Atlas Search	Yes
N-able	Risk Intelligence	Yes
National Security Agency	Ghidra	Yes
Nelson	Nelson	No
Neo4j	Neo4j: Graphs for Everyone	Yes
NetCore j.s.a.	unimus	Yes
Netflix	Netflix Atlas	No
Netflix	dgs-framework	Yes
Netflix	Spectator	Yes
New Relic	Containerized private minion (CPM)	Yes
New Relic	Java agent	Yes
New Relic	Java agent	Yes
Okta	Okta RADIUS Server Agent	Yes
openHAB	openHAB Distribution	Yes
OpenMRS	OpenMRS Platform	Yes
OpenMRS	Reference Application	Yes
OpenSearch	OpenSearch	Yes
OWASP	ZAP	Yes
PaperCut Software	PaperCut MF	Yes
PaperCut Software	PaperCut NG	Yes
Red Hat Inc.	Red Hat OpenShift Container Platform	Yes
Red Hat Inc.	AMQ Streams	Yes
Red Hat Inc.	Red Hat OpenShift Container Platform	Yes
Red Hat Inc.	AMQ Streams	Yes
Red Hat Inc.	Red Hat Process Automation Manager (formerly JBoss BPM Suite)	Yes
Red Hat Inc.	Fuse	Yes
Red Hat Inc.	Red Hat Integration Camel-K	Yes
Red Hat Inc.	JBoss Enterprise Application Platform	Yes
Red Hat Inc.	eap7-yasson (Red Hat package)	Yes
Red Hat Inc.	eap7-yasson (Red Hat package)	Yes
Redis Labs	Jedis	Yes

Riverbed	Riverbed Portal	No
Riverbed	Riverbed NetIM	No
Riverbed	Riverbed UCExpert	No
Riverbed	Scon EX Director	No
Riverbed	Scon EX Analytics	No
Rundeck	Rundeck	Yes
Siemens	SPPA-T3000 Application Server	No
Silver Peak Systems, Inc.	Silver Peak Orchestrator	No
Sitecore	Sitecore XP	No
SmartBear	SoapUI	Yes
SolarWinds	Database Performance Analyzer	No
Stardog Union	Stardog	Yes
Stratodesk	NoTouch Center	Yes
Sumo Logic	Sumo Logic	Yes
SuSE	storm-supervisor	Yes
SuSE	storm-supervisor	Yes
SyncRO Soft	Oxygen Content Fusion	Yes
SyncRO Soft	Oxygen XML Web Author	Yes
SyncRO Soft	Oxygen Feedback	Yes
SyncRO Soft	Oxygen XML Publishing Engine	Yes
SyncRO Soft	Oxygen XML WebHelp	Yes
SyncRO Soft	Oxygen PDF Chemistry	Yes
SyncRO Soft	Oxygen License Server	Yes
SyncRO Soft	Oxygen XML Author	Yes
SyncRO Soft	Oxygen XML Developer	Yes
SyncRO Soft	Web Author PDF Plugin	Yes
SyncRO Soft	Oxygen Web Author Test Server Add-on	Yes
SyncRO Soft	XSD to JSON Schema Converter	Yes
SyncRO Soft	Git Client	Yes
SyncRO Soft	Batch Documents Converter	Yes
Syntevo	DeepGit	Yes
Trend Micro	Deep Discovery Director	Yes
Ubiquiti Networks	UniFi Network Application	Yes
VMware, Inc	vCenter Server	No
VMware, Inc	VMware Horizon	Yes
VMware, Inc	VMware HCX	Yes
VMware, Inc	NSX-T	No

VMware, Inc	VMWare Unified Access Gateway	No
VMware, Inc	VMware Workspace One Access	No
VMware, Inc	VMware Identity Manager	No
VMware, Inc	VMware vRealize Operations	No
VMware, Inc	vRealize Log Insight	No
VMware, Inc	vRealize Automation	No
VMware, Inc	vRealize Suite Lifecycle Manager	No
VMware, Inc	Telco Cloud Automation	No
VMware, Inc	Carbon Black Cloud Workload appliance	No
VMware, Inc	Carbon Black EDR Server	Yes
VMware, Inc	vSphere Replication	No
VMware, Inc	VMware Tanzu GemFire	Yes
VMware, Inc	Tanzu Greenplum	No
VMware, Inc	VMware Tanzu Operations Manager	Yes
VMware, Inc	VMware Tanzu Application Service for VMs	Yes
VMware, Inc	VMware Tanzu Kubernetes Grid Integrated Edition	No
VMware, Inc	VMware Tanzu Observability by Wavefront Nozzle	Yes
VMware, Inc	Healthwatch for Tanzu Application Service	Yes
VMware, Inc	Spring Cloud Services for VMware Tanzu	Yes
VMware, Inc	Spring Cloud Gateway for VMware Tanzu	Yes
VMware, Inc	Spring Cloud Gateway for Kubernetes	No
VMware, Inc	API Portal for VMware Tanzu	Yes
VMware, Inc	Single Sign-On for VMware Tanzu Application Service	Yes
VMware, Inc	App Metrics	Yes
VMware, Inc	VMware vCenter Cloud Gateway	No
VMware, Inc	vRealize Orchestrator	No

VMware, Inc	Cloud Foundation	No
VMware, Inc	Horizon DaaS	No
VMware, Inc	Horizon Cloud Connector	Yes
VMware, Inc	NSX Data Center for vSphere	Yes
VMware, Inc	AppDefense Appliance	No
VMware, Inc	Cloud Director Object Storage Extension	Yes
VMware, Inc	Telco Cloud Operations	No
VMware, Inc	Tanzu Scheduler	No
VMware, Inc	Smart Assurance NCM	No
VMware, Inc	Smart Assurance SAM	No
VMware, Inc	VMware Workspace One Access Connector	No
VMware, Inc	vRealize Business for Cloud	No
VMware, Inc	Integrated OpenStack	No
Wowza Media Systems	Wowza Streaming Engine	No
Yellowfin	Yellowfin	Yes
Zoho Corporation	Zoho ManageEngine EventLog Analyzer	Yes
ZyXEL Communications Corp.	NetAtlas Element Management System (EMS)	No

Appendix 2: Famous Libraries and Products affected by Log4j

Company	Affected Product Details
Adobe	Adobe Cold Fusion, Adobe Experience Manager
Amazon	Open Search
Atlassian	Bamboo, Confluence, Cruicible, Fisheye, Jira— Self-hosted if configured with Log4j.
Apache	Cassandra — via appender, Druid, Dubbo, Flink, Geode ,Hadoop, James, Kafka, Karaf — Depends on PAX logging which is affected, Solr, Spark, Storm — via Docker, Struts, Tapestry, Tika, Wicket
Cisco	Cisco Webex Meetings Server, Cisco CX Cloud Agent Software, Cisco Firepower Threat Defense, Cisco CloudCenter, Cisco Video Surveillance Operations Manager
F-Secure	F-Secure Elements Connector, F-Secure Endpoint Proxy, F-Secure Messaging Security Gateway, F-Secure Policy Manager — Note: Only the Policy Manager Server component is affected, F-Secure Policy Manager for Linux, F-Secure Policy Manager Proxy, F-Secure Policy Manager Proxy for Linux
Github	GitHub Enterprise Server
Microsoft	Azure DevOps Server — Versions 2020 & earlier affected, Team Foundation Server — Versions 2018.2+ affected.
Oracle	Enterprise Manager — Affected versions: 13.3.2, 13.4, & 13.5, Exadata — Affected versions: < 21.3.4.
SonicWall	Email Security — ES 10.0.11 and earlier versions are affected, NSM — Affected, WAF — Version 3.x with Cloud Management enabled is affected.
Team Viewer	TeamViewer Engage, TeamViewer Frontline, TeamViewer IoT — Affected, low.
Ubiquiti	UniFi Network Application, UniFi Network Controller
Vmware	API Portal for VMware Tanzu, App Metrics,Carbon Black Cloud Workload Appliance, Carbon Black EDR Servers, Cloud Foundation, HCX, Healthwatch for Tanzu Application Service, Horizon, Identity Manager, NSX-T Data Center, Single Sign-On for VMware Tanzu Application Service, Site Recovery Manager, Spring Cloud Gateway for Kubernetes, Spring Cloud Gateway for VMware Tanzu, Spring Cloud Services for VMware Tanzu, Tanzu Application Service for VMs, Tanzu GemFire, Tanzu Greenplum, Tanzu Kubernetes Grid Integrated Edition, Tanzu Observability by Wavefront Nozzle, Tanzu Operations Manager, Tanzu SQL with MySQL for VMs, Telco Cloud Automation, Unified Access Gateway, vCenter Cloud Gateway, vCenter Server, vRealize Automation, vRealize Lifecycle Manager, vRealize Log Insight, vRealize Operations, vRealize Operations Cloud Proxy, vRealize Orchestrator, WorkspaceOne Access

Appendix 3: Log4j Scanner Results

{
scanSummary : {
scanEngine : 2.1.3.0,
scanHostname : ***********,
scanDate : 2022-06-12T12:59:13+0300,
scanDurationSeconds : 550,
scanErrorCount : 15,
scanStatus : Partially Successful,
scannedFiles : 812095,
scannedDirectories : 212889,
scannedJARs : 2182,
scannedWARs : 0,
scannedEARs : 0,
scannedTARs : 1682,
scannedCompressed : 4216,
excludedDrives : [],
excludedDirectories : [],
excludedFiles : [ ],
knownTarExtensions : [.tar"],
knownGZipTarExtensions : [.tgz,.tar.gz],
knownBZipTarExtensions : [.tbz,.tbz2,.tar.bz,.tar.bz2],
knownZipExtensions :
[.zip,.jar,.war,.ear,.par,.kar,.sar,.rar,.jpi,.hpi,.apk],
vulnerabilitiesFound : 11},

scanDetails : [
{

file :
C:\\Users\\aweso\\AppData\\Roaming\\.minecraft\\libraries\\com\\mojang\\logging\\1.0.0\\logging-1.0.0.jar,
manifestVendor : Unknown,
manifestVersion : Unknown,
Log4j : true,
Log4j1.x : false,
Log4j2.x : true,
detectedJNDILookupClass : false,
Log4jManifest : false,
Log4jVendor : Unknown,
Log4jVersion : Unknown,
CVE-2021-4104 :false,
CVE-2021-44228 :false,
CVE-2021-44832 :false,
CVE-2021-4504 :false,
CVE-2021-45105 :false,
CVE_Status : N/.A
},

{

file :
C:\\Users\\aweso\\AppData\\Roaming\\.minecraft\\libraries\\com\\mojang\\netty\\1.8.8\\netty-1.8.8.jar,
manifestVendor : Unknown,
manifestVersion : Unknown,
Log4j : true,
Log4j1.x : false,
Log4j2.x : true,
detectedJNDILookupClass : true,
Log4jManifest : false,
Log4jVendor : Unknown,
Log4jVersion : Unknown,
CVE-2021-4104 :false,
CVE-2021-44228 :false,
CVE-2021-44832 :false,
CVE-2021-4504 :false,
CVE-2021-45105 :false,
CVE_Status : Unknown
},

{
file :
C:\\Users\\aweso\\AppData\\Roaming\\.minecraft\\libraries\\org\\apache\logging\\Log4j\\Log4j-api\\2.0-beta9\\Log4j-api-2.0-beta9.jar,
manifestVendor : org.apache,
manifestVersion : 2.0-beta9,
Log4j : true,
Log4j1.x : false,
Log4j2.x : true,
detectedJNDILookupClass : false,
Log4jManifest : false,
Log4jVendor : Log4j-api,
Log4jVersion : 2.0-beta9,
CVE-2021-4104 :false,
CVE-2021-44228 :true,
CVE-2021-44832 :true,
CVE-2021-4504 :true,
CVE-2021-45105 :true,
CVE_Status : Unknown
},

{
file :
C:\\Users\\aweso\\AppData\\Roaming\\.minecraft\\libraries\\org\\apache\\logging\\Log4j\\Log4j-api\\2.17.0\\Log4j-api-2.17.0.jar,
manifestVendor : Log4j,
manifestVersion : 2.17.0,
Log4j : true,
Log4j1.x : false,
Log4j2.x : true,
detectedJNDILookupClass : false,
Log4jManifest : true,
Log4jVendor : Log4j-api,
Log4jVersion : 2.17.0,
CVE-2021-4104 :false,
CVE-2021-44228 :true,

CVE-2021-44832 :true,
CVE-2021-4504 :true,
CVE-2021-45105 :true,
CVE_Status : Mitigated
},

{
file :
C:\\Users\\aweso\\AppData\\Roaming\\.minecraft\\libraries\\org\\apache\\logging\\Log4j\\Log4j-core\\2.0-beta9\\Log4j-core-2.0-beta9.jar,
manifestVendor : org.apache,
manifestVersion : 2.0-beta9,
Log4j : true,
Log4j1.x : false,
Log4j2.x : true,
detectedJNDILookupClass : true,
Log4jManifest : false,
Log4jVendor : Log4j-core,
Log4jVersion : 2.0-beta9,
CVE-2021-4104 :false,
CVE-2021-44228 :false,
CVE-2021-44832 :false,
CVE-2021-4504 :false,
CVE-2021-45105 :false,
CVE_Status : Potentially Vulnerable (CVE-2021-44228: Found CVE-2021-44832: Found CVE-2021-45046: Found CVE-2021-45105: Found)
},

{
file :
C:\\Users\\aweso\\AppData\\Roaming\\.minecraft\\libraries\\org\\apache\\logging\\Log4j\\Log4j-core\\2.17.0\\Log4j-core-2.17.0.jar,
manifestVendor : Log4j,
manifestVersion : 2.17.0,
Log4j : true,
Log4j1.x : false,
Log4j2.x : true,
detectedJNDILookupClass : true,
Log4jManifest : true,
Log4jVendor : Log4j-core,
Log4jVersion : 2.17.0,
CVE-2021-4104 :false,
CVE-2021-44228 :true,
CVE-2021-44832 :false,
CVE-2021-4504 :true,
CVE-2021-45105 :true,
CVE_Status : Potentially Vulnerable (CVE-2021-44228: NOT Found CVE-2021-44832: Found CVE-2021-45046: NOT Found CVE-2021-45105: NOT Found)
},

{
file :
C:\\Users\\aweso\\AppData\\Roaming\\.minecraft\\tlauncher_libraries\\Log4j\\Log4j\\1.2.17\\Log4j-1.2.17.jar,

manifestVendor : Apache Software Foundation,
manifestVersion : 1.2.17,
Log4j : true,
Log4j1.x : true,
Log4j2.x : false,
detectedJNDILookupClass : false,
Log4jManifest : true,
Log4jVendor : Log4j,
Log4jVersion : 1.2.17,
CVE-2021-4104 :false,
CVE-2021-44228 :true,
CVE-2021-44832 :true,
CVE-2021-4504 :true,
CVE-2021-45105 :true,
CVE_Status : Potentially Vulnerable (CVE-2021-4104: Found)
},

{
file : F:\\Log4j\\pigot-mc-server\\spigot-1.7.2-R0.4-SNAPSHOT-1339.jar,
manifestVendor : Bukkit Team,
manifestVersion : git-Spigot-1339,
Log4j : true,
Log4j1.x : false,
Log4j2.x : true,
detectedJNDILookupClass : true,
Log4jManifest : false,
Log4jVendor : Unknown,
Log4jVersion : Unknown,
CVE-2021-4104 :false,
CVE-2021-44228 :false,
CVE-2021-44832 :false,
CVE-2021-4504 :false,
CVE-2021-45105 :false,
CVE_Status : Unknown
},

{
file : F:\\Program Files\\Fiji.app\\jars\\Log4j-1.2.17.jar,
manifestVendor : Apache Software Foundation,
manifestVersion : 1.2.17,
Log4j : true,
Log4j1.x : true,
Log4j2.x : false,
detectedJNDILookupClass : false,
Log4jManifest : true,
Log4jVendor : Log4j,
Log4jVersion : 1.2.17,
CVE-2021-4104 :false,
CVE-2021-44228 :true,
CVE-2021-44832 :true,
CVE-2021-4504 :true,
CVE-2021-45105 :true,
CVE_Status : Potentially Vulnerable (CVE-2021-4104: Found)
},

{
file : F:\\Program Files\\IBM\\SPSS\\Statistics\\27\\as-3.2.3.0\\lib\\com.ibm.spss.http-3.2.3.0.jar,
manifestVendor : Unknown,
manifestVersion : Unknown,
Log4j : true,
Log4j1.x : false,
Log4j2.x : true,
detectedJNDILookupClass : false,
Log4jManifest : false,
Log4jVendor : Unknown,
Log4jVersion : Unknown,
CVE-2021-4104 :false,
CVE-2021-44228 :false,
CVE-2021-44832 :false,
CVE-2021-4504 :false,
CVE-2021-45105 :false,
CVE_Status : N/.A
},

{
file : F:\\Program Files\\IBM\\SPSS\\Statistics\\27\\as-3.2.3.0\\lib\\Log4j-api-2.13.3.jar,
manifestVendor : Log4j,
manifestVersion : 2.13.3,
Log4j : true,
Log4j1.x : false,
Log4j2.x : true,
detectedJNDILookupClass : false,
Log4jManifest : true,
Log4jVendor : Log4j-api,
Log4jVersion : 2.13.3,
CVE-2021-4104 :false,
CVE-2021-44228 :true,
CVE-2021-44832 :true,
CVE-2021-4504 :true,
CVE-2021-45105 :true,
CVE_Status : Mitigated
},

{
file : F:\\Program Files\\IBM\\SPSS\\Statistics\\27\\as-3.2.3.0\\lib\\Log4j-core-2.13.3.jar,
manifestVendor : Log4j,
manifestVersion : 2.13.3,
Log4j : true,
Log4j1.x : false,
Log4j2.x : true,
detectedJNDILookupClass : true,
Log4jManifest : true,
Log4jVendor : Log4j-core,
Log4jVersion : 2.13.3,
CVE-2021-4104 :false,
CVE-2021-44228 :false,
CVE-2021-44832 :false,
CVE-2021-4504 :false,

CVE-2021-45105 :false,
CVE_Status : Potentially Vulnerable (CVE-2021-44228: Found CVE-2021-44832: Found CVE-2021-45046: Found CVE-2021-45105: Found)
},

{
file : F:\\Program Files\\IBM\\SPSS\\Statistics\\27\\common\\ext\\bin\\spss.cognos.9\\Log4j-1.2.17.jar,
manifestVendor : Apache Software Foundation,
manifestVersion : 1.2.17,
Log4j : true,
Log4j1.x : true,
Log4j2.x : false,
detectedJNDILookupClass : false,
Log4jManifest : true,
Log4jVendor : Log4j,
Log4jVersion : 1.2.17,
CVE-2021-4104 :false,
CVE-2021-44228 :true,
CVE-2021-44832 :true,
CVE-2021-4504 :true,
CVE-2021-45105 :true,
CVE_Status : Potentially Vulnerable (CVE-2021-4104: Found)
},

{
file : F:\\Program Files\\IBM\\SPSS\\Statistics\\27\\Log4j-1.2-api-2.13.3.jar,
manifestVendor : Log4j,
manifestVersion : 2.13.3,
Log4j : true,
Log4j1.x : true,
Log4j2.x : true,
detectedJNDILookupClass : false,
Log4jManifest : true,
Log4jVendor : Log4j-1.2-api,
Log4jVersion : 2.13.3,
CVE-2021-4104 :false,
CVE-2021-44228 :true,
CVE-2021-44832 :true,
CVE-2021-4504 :true,
CVE-2021-45105 :true,
CVE_Status : Potentially Vulnerable (CVE-2021-4104: Found)
},

{
file : F:\\Program Files\\IBM\\SPSS\\Statistics\\27\\Log4j-api-2.13.3.jar,
manifestVendor : Log4j,
manifestVersion : 2.13.3,
Log4j : true,
Log4j1.x : false,
Log4j2.x : true,
detectedJNDILookupClass : false,
Log4jManifest : true,

Log4jVendor : Log4j-api,
Log4jVersion : 2.13.3,
CVE-2021-4104 :false,
CVE-2021-44228 :true,
CVE-2021-44832 :true,
CVE-2021-4504 :true,
CVE-2021-45105 :true,
CVE_Status : Mitigated
},

{
file : F:\\Program Files\\IBM\\SPSS\\Statistics\\27\\Log4j-core-2.13.3.jar,
manifestVendor : Log4j,
manifestVersion : 2.13.3,
Log4j : true,
Log4j1.x : false,
Log4j2.x : true,
detectedJNDILookupClass : true,
Log4jManifest : true,
Log4jVendor : Log4j-core,
Log4jVersion : 2.13.3,
CVE-2021-4104 :false,
CVE-2021-44228 :false,
CVE-2021-44832 :false,
CVE-2021-4504 :false,
CVE-2021-45105 :false,
CVE_Status : Potentially Vulnerable (CVE-2021-44228: Found CVE-2021-44832: Found CVE-2021-45046: Found CVE-2021-45105: Found)
},

{
file : F:\\Program Files (x86)\\Minecraft\\libraries\\com\\mojang\\patchy\\1.1\\patchy-1.1.jar,
manifestVendor : Unknown,
manifestVersion : Unknown,
Log4j : true,
Log4j1.x : false,
Log4j2.x : true,
detectedJNDILookupClass : false,
Log4jManifest : false,
Log4jVendor : Unknown,
Log4jVersion : Unknown,
CVE-2021-4104 :false,
CVE-2021-44228 :false,
CVE-2021-44832 :false,
CVE-2021-4504 :false,
CVE-2021-45105 :false,
CVE_Status : N/.A
},

{
file : F:\\Program FIles (x86)\\Minecraft\\libraries\\org\\apache\\logging\\Log4j\\Log4j-api\\2.8.1\\Log4j-api-2.8.1.jar,

manifestVendor : org.apache,
manifestVersion : 2.8.1,
Log4j : true,
Log4j1.x : false,
Log4j2.x : true,
detectedJNDILookupClass : false,
Log4jManifest : false,
Log4jVendor : Log4j-api,
Log4jVersion : 2.8.1,
CVE-2021-4104 :false,
CVE-2021-44228 :true,
CVE-2021-44832 :true,
CVE-2021-4504 :true,
CVE-2021-45105 :true,
CVE_Status : Unknown
},

{
file : F:\\Program FIles (x86)\\Minecraft\\libraries\\org\\apache\\logging\\Log4j\\Log4j-core\\2.8.1\\Log4j-core-2.8.1.jar,
manifestVendor : org.apache,
manifestVersion : 2.8.1,
Log4j : true,
Log4j1.x : false,
Log4j2.x : true,
detectedJNDILookupClass : true,
Log4jManifest : false,
Log4jVendor : Log4j-core,
Log4jVersion : 2.8.1,
CVE-2021-4104 :false,
CVE-2021-44228 :false,
CVE-2021-44832 :false,
CVE-2021-4504 :false,
CVE-2021-45105 :false,
CVE_Status : Potentially Vulnerable (CVE-2021-44228: Found CVE-2021-44832: Found CVE-2021-45046: Found CVE-2021-45105: Found)
},

{
file : F:\\Program FIles (x86)\\Minecraft\\libraries\\org\\tlauncher\\authlib\\2.0.28.1\\authlib-2.0.28.1.jar,
manifestVendor : Unknown,
manifestVersion : Unknown,
Log4j : true,
Log4j1.x : false,
Log4j2.x : true,
detectedJNDILookupClass : false,
Log4jManifest : false,
Log4jVendor : Unknown,
Log4jVersion : Unknown,
CVE-2021-4104 :false,
CVE-2021-44228 :false,
CVE-2021-44832 :false,
CVE-2021-4504 :false,

CVE-2021-45105 :false,
CVE_Status : N/.A
},

{
file : F:\\Program FIles
(x86)\\Minecraft\\libraries\\org\\tlauncher\\patchy\\1.1\\patchy-
1.1.jar,
manifestVendor : Unknown,
manifestVersion : Unknown,
Log4j : true,
Log4j1.x : false,
Log4j2.x : true,
detectedJNDILookupClass : false,
Log4jManifest : false,
Log4jVendor : Unknown,
Log4jVersion : Unknown,
CVE-2021-4104 :false,
CVE-2021-44228 :false,
CVE-2021-44832 :false,
CVE-2021-4504 :false,
CVE-2021-45105 :false,
CVE_Status : N/.A
},

{
file : F:\\Program FIles
(x86)\\Minecraft\\tlauncher_libraries\\Log4j\\Log4j\\1.2.17\\Log4j-
1.2.17.jar,
manifestVendor : Apache Software Foundation,
manifestVersion : 1.2.17,
Log4j : true,
Log4j1.x : true,
Log4j2.x : false,
detectedJNDILookupClass : false,
Log4jManifest : true,
Log4jVendor : Log4j,
Log4jVersion : 1.2.17,
CVE-2021-4104 :false,
CVE-2021-44228 :true,
CVE-2021-44832 :true,
CVE-2021-4504 :true,
CVE-2021-45105 :true,
CVE_Status : Potentially Vulnerable (CVE-2021-4104: Found)
},

{
file : F:\\Program FIles
(x86)\\Minecraft\\tlauncher_libraries\\org\\apache\\logging\\Log4j\
\Log4j-core\\2.14.1\\Log4j-core-2.14.1.jar,
manifestVendor : Log4j,
manifestVersion : 2.14.1,
Log4j : true,
Log4j1.x : false,
Log4j2.x : true,
detectedJNDILookupClass : true,

Log4jManifest : true,
Log4jVendor : Log4j-core,
Log4jVersion : 2.14.1,
CVE-2021-4104 :false,
CVE-2021-44228 :false,
CVE-2021-44832 :false,
CVE-2021-4504 :false,
CVE-2021-45105 :false,
CVE_Status : Potentially Vulnerable (CVE-2021-44228: Found CVE-2021-44832: Found CVE-2021-45046: Found CVE-2021-45105: Found) }]}