登录

Burp Suite 反序列化插件 Java Deserialization Scanner

ly55521 / 2017-06-02 14:03:17 / 浏览数 5796 安全工具 工具 顶(0) 踩(0)

介绍

Java反序列化的漏洞的发现和披露，在一月前由Gabriel Lawrence和Chris Frohoff 发现了这些严重漏洞，java序列化序列化的对象（见Gabriel Lawrence和Chris Frohoff）中的潜在缺陷，受影响的可能会在各种框架和库的中。然而有许多方法可用于防止被利用。

原来披露的漏洞并未引起太多关注。直到2016年11月，Stephen Breen安全研究员 发现实际在生产 java服务器上可以被利用。

在这篇文章中，我们将不关注如何序列化漏洞的工作，以及如何修复它们，因为已经有很多关于这个主题的文章。相反，今天我们将专注于如何可靠地检测和利用这些漏洞。
将序列化对象做反序列化，java类试图把对象进行反序列化。故接收到的对象为反序列化操作，即使他们不是预期的类中实例化的；在这种情况下，在反序列化异常出现时，

因此，能够实现远程命令执行（RCE）就是要找到一个"链"的对象，一旦反序列化，允许攻击者执行任意的java代码。显然，所选择的对象的类必须在目标系统中被序列化。

一旦反序列化问题被发现了，ysoserial
工具可用于开发payload。此工具生成自定义开发载体，基于"脆弱"的目标系统中加载的库。在这篇文章中我们将分析如何发现和利用java序列化的漏洞利用Burp
Suite插件，我们开发了基于 ysoserial：java序列化扫描器：Java Deserialization Scanner.

安装

java序列化扫描插件有两种安装方法：

1.直接下载在 Extdener-》Bapp Store
，这是最简单的方法来获得插件，但下载的版本可能不是最新的。目前，例如，最新的版本（0.5预发行版）只能从GitHub（见下一个方法）。当发布的版本将发布，我们将

2.从GitHub下载最新发布和手动从Burp Suite扩展标签安装JAR Extdener-》Extdeners-》Add

检测

对反序列化的漏洞检测并不是一个简单的任务。通过产生的 ysoserial
将其发送到目标应用程序的payload，通常我们可以获得一个java堆栈跟踪（如果幸运的话我们可以发现问题，存在的只是一个脆弱的系统有针对性的）或没有详细的输出

因此，为了可靠地检测漏洞的存在，我们修改 ysoserial 生成java本地sleep
有效载荷代替RCE的有效载荷和我们说这些有效载荷的java反序列化扫描器。这个任务需要使用java本地sleep有效载荷，因为java调用是同步的；执行系统所产生的ysoseri
sleep 使用默认的RCE的有效载荷将是无用的，因为他们是异步的，我们将 sleep 命令结束前从服务器得到响应，无论是否存在这个问题。

在插件的最新版本中，我们增加了两个新的方法来进一步提高检测：一个基于DNS和一个CPU。

为了生成java执行本地DNS解析的有效载荷，我们再次修改ysoserial。通常情况下，DNS解析请求是最有可能绕过企业防火墙，因此是一个相当好的检测方法。在一般情况下
不稳定的系统或高度延迟的网络。感谢Burp Suite Collaborator，这是没有必要有一个DNS插件，一切都可以在Burp Suite工具完成。

CPU检测方法是基于Wouter Coekaerts"serialdos工作：没有任何脆弱的系统检测反序列化问题。有效载荷是基于一个系统对象（java
util。HashSet），采用多CPU周期为反序列化任务。SerialDOS是创建一个POC的拒绝服务（DOS）攻击，但通过降低CPU周期需要反序列化它也可以用来作为一种检测方
所有载荷）。

现在，让我们演示如何使用我们的插件进行检测。检测集成在Burp
Suite的主动和被动的扫描器。默认情况下，CPU和DNS检查添加到扫描器中，但他们可以禁用插件的配置面板，在部分"自动扫描配置"：

Deserialization Scanner.-》Automatic scanner configurations

为了减少扫描器执行的请求数量，只有在原始请求中存在序列化对象时，才由插件添加的检查执行。有效载荷是相同的编码在原始请求发现编码（例如，如果序列化对象的编

- Raw
- BASE64
- ASCII HEX
- GZIP
- BASE64 GZIP

在主动扫描检查时也就是默认不使用CPU检测方法，因为它必须谨慎使用：发送数量巨大的 serialdos
载荷可能仍然在旧的或高负载的系统造成的问题。为了执行检查自定义插入点或使用CPU负载，该插件提供了"Manual
Testing"（"手动测试"）选项卡，用户可以在其中选择插入点（目前在同一时间只有一个支持），选择查询类型（DNS，sleep，或CPU），选择优先编码和测试参数。通过

测试要求可以手动插入手动测试"或可以从其他Burp Suite标签发送使用打开的鼠标右键菜单：

手动测试工具的配置说明如下图片：

开发

“Exploiting”选项卡提供了一个舒适的界面开发反序列化的漏洞。这个标签使用ysoserial工具生成开发载体，包括产生一个HTTP请求负载。ysoserial作为论据脆弱的图书馆

现在，让我们演示如何使用我们的插件开发。首先，我们需要打开“Configuration”选项卡并插入路径，我们有一份的ysoserial工具（ysoserial仅需要开发；检测的有效载荷

然后，我们看到了手工测试，它可以插入要求手动或把它从其他Burp
Suite标签使用打开的鼠标右键菜单。然后用户可以选择插入点（目前在同一时间只有一个支持），将ysoserial命令（ysoserial手册
https://github.com/frohoff/ysoserial），点击“Attack”按钮，根据所需的编码。“Exploiting”工具的配置在下面的图片中解释：

插件提供的界面使开发过程更快更舒适。

这就是全部！该插件的最后一个版本（目前0.5预发行版）可以从GitHub发布页下载。如果您发现任何错误，或者如果你有想法改进请打开GitHub上的一个新问题。

https://github.com/federicodotta/Java-Deserialization-Scanner/releases

相关链接：

1. https://github.com/federicodotta/Java-Deserialization-Scanner/releases
2. https://www.slideshare.net/frohoff1/appseccali-2015-marshalling-pickles
3. https://foxglovesecurity.com/2015/11/06/what-do-weblogic-websphere-jboss-jenkins-opennms-and-your-application-have-in-common-this-vulnerability/
4. https://github.com/frohoff/ysoserial
5. https://portswigger.net/
6. https://gist.github.com/coekie/a27cc406fc9f3dc7a70d

referer ：https://techblog.mediaservice.net/2017/05/reliable-discovery-and-exploitation-of-java-deserialization-vulnerabilities/

上一篇：2017补天成都沙龙讲师PPT 下一篇：刷SRC不按常规思路刷会发生什么？

1. 0 条回复
   - 动动手指，沙发就是你的了！