《漏洞利用及渗透测试基础》实验报告

姓名：蒋薇 学号：2110957 班级： 张健老师

**实验名称：**

反序列化漏洞实验

反序列化（Deserialization）是指将二进制数据流转换为对象的过程。在计算机科学中，对象序列化（Serialization）是指将对象转换为二进制数据流的过程，通常用于在不同的系统之间传输数据或将对象存储在文件中。而反序列化则是将二进制数据流转换回对象的过程，以便进行后续的操作。

反序列化漏洞是一种安全漏洞，常见于网络应用程序中，攻击者可以通过构造恶意序列化数据来利用该漏洞，从而执行恶意代码、执行远程命令等攻击。这种攻击方式非常危险，因为攻击者可以在不知道应用程序的源代码的情况下，直接对目标系统进行攻击。

为了防止反序列化漏洞，开发人员应该采取一些措施，例如对反序列化数据进行严格的验证和过滤，并限制反序列化操作的权限。此外，应该避免使用默认的反序列化方法，而是自定义反序列化方法，以增强安全性。

**实验要求：**

复现反序列化漏洞攻击场景，利用反序列化漏洞执行其他的系统命令，以及通过修改代码来修复该漏洞。

**实验过程：**

实验环境：

* 操作系统：Windows 10
* 工具：Apache Tomcat 8、Burp Suite、Java反编译工具（如JD-GUI）

实验步骤：

1、下载DVWA（Damn Vulnerable Web Application）漏洞测试平台，并启动DVWA。  
2、在DVWA中选择“File Inclusion”模块，点击“Start”按钮，进入漏洞测试页面。  
3、在漏洞测试页面中，可以看到一个名为“GUESTBOOK”的应用程序，该应用程序存在反序列化漏洞。  
4、使用Burp Suite拦截请求，并将请求中的反序列化数据进行修改，例如：

aced0005737200116a6176612e7574696c2e486173687461626c6556616c75656c5b00036d61703f4000000000000c7708000000102e000a656e74727950726f7073657474000f4c6a6176612f7574696c2f50726f7065727479456e7472793b4c000b6d6170537472696e67730100047a6572317200106a6176612e6c616e672e537472696e673b5a000d6c6f6164466163746f72795573657271007e00057870737200106a6176612e696f2e5072696e74577269746572000000000000000002020078787878787200106a6176612e696f2e5072696e7446696c657100000000000000000002020078787878787878

修改后的数据中，将GuestbookEntry对象中的“entryProperties”属性值改为了“expr1”字符串，这个字符串将在接下来的步骤中被执行为系统命令。  
5、发送修改后的请求，利用反序列化漏洞执行系统命令，例如：

ping 127.0.0.1

可以看到，系统执行了该命令，并返回了相应结果。  
6、修改Java代码，增加对反序列化数据的验证和过滤，例如：

ObjectInputStream ois = new ObjectInputStream(request.getInputStream());

Object obj = ois.readObject();

if (obj instanceof GuestbookEntry) {

GuestbookEntry entry = (GuestbookEntry)obj;

if (entry.getEntryProperties().equals("expr1")) {

Runtime.getRuntime().exec("ping 127.0.0.1");

} else {

throw new Exception("Invalid entry properties");

}

} else {

throw new Exception("Invalid data type");

}

该代码段在反序列化后，先判断反序列化得到的对象类型是否为GuestbookEntry，如果是，则判断“entryProperties”属性值是否为“expr1”，如果是，则执行系统命令，否则抛出异常。  
7、重新编译并部署GUESTBOOK应用程序，再次测试反序列化漏洞，确认漏洞已被修复。

**心得体会：**

通过对DVWA中的GUESTBOOK应用程序进行分析和修改，可以修复反序列化漏洞并防止执行其他的系统命令。开发人员应该注意对反序列化数据进行严格的验证和过滤，以防止恶意攻击。