此文档为《Java代码审计零基础入门到项目实战》配套教材，由【闪石星曜CyberSecurity】出品。

请勿对外泄露，一经发现严格处理！

课程学习中有任何疑问，可添加好友 Power_7089 寻求帮助，为你答疑解惑。

本节讲述JavaWeb代码审计之任意文件读取/下载漏洞。

在【第一阶段】Java代码审计之基础篇1.4小节学习了Java文件操作之文件读取与下载。并给出了多种读取/下载文件的示例代码。示例代码中代码几乎没有任何防护。

（备注：下面部分讲解会涉及到`【第一阶段】Java代码审计之基础篇 - 1.4小节学习了Java文件操作之文件读取与下载`中`webreadfile`项目工程。建议学完前置教程后再开启学习本篇教程。）

一、任意文件读取/下载漏洞

在先前的学习中了解到文件读取和下载的小区别。但对于我们代码审计和渗透测试目的来看这两者区别并不是很大，如果存在漏洞不论是读取还是下载，我们都是能获取到目标服务器敏感文件信息。

1、什么是任意文件读取/下载漏洞

任意文件读取/下载漏洞，常发生在查看文件/下载文件等地方，后端没有对用户查看或下载的内容做限制，导致可以查看或下载其他文件，甚至是服务器中敏感文件。

举个例子，比如系统中存在一处下载功能，抓包发现URL为`http://127.0.0.1/file?download=/img/touxiang1.img`。下面我们可以尝试配合使用`../`目录穿越漏洞读取系统内敏感文件，最终读取/下载任意文件URL为`http://127.0.0.1/file?download=../../../../../../../../../etc/passwd`。

当然了，上述仅是个在没有任何防护下的例子，为了便于理解任意文件读取/下载漏洞。

2、windows系统敏感文件

```
boot.ini #查看系统版本
c:/windows/php.ini #php配置信息
c:/windows/my.ini #MYSQL配置文件，记录管理员登陆过的MYSQL用户名和密码
c:/winnt/php.ini
c:/winnt/my.ini
c:\mysql\data\mysql\user.MYD #mysql.user表中的数据库连接密码
c:\Program Files\RhinoSoft.com\Serv-U\ServUDaemon.ini #存储了虚拟主机网站路径和密码
c:\Program Files\Serv-U\ServUDaemon.ini
c:\windows\system32\inetsrv\MetaBase.xml #查看IIS的虚拟主机配置
c:\windows\repair\sam #WINDOWS系统初次安装的密码
c:\Program Files\ Serv-U\ServUAdmin.exe #6.0版本以前的serv-u管理员密码
c:\Program Files\RhinoSoft.com\ServUDaemon.exe
C:\Documents and Settings\All Users\Application Data\Symantec\pcAnywhere\*.cif文件
#存储了pcAnywhere的登陆密码
c:\Program Files\Apache Group\Apache\conf\httpd.conf 或C:\apache\conf\httpd.conf
#查看WINDOWS系统apache文件
c:/Resin-3.0.14/conf/resin.conf #查看jsp开发的网站resin文件配置信息.
c:/Resin/conf/resin.conf /usr/local/resin/conf/resin.conf #查看linux系统配置的JSP虚
拟主机
```

```
d:\APACHE\Apache2\conf\httpd.conf
C:\Program Files\mysql\my.ini
C:\mysql\data\mysql\user.MYD #存在MYSQL系统中的用户密码
C:\Windows\System32\drivers\etc\hostswinserver配置Telnet信息
```

3、Linux系统敏感文件

```
/etc/httpd/conf/httpd.conf
/etc/rc.local 有时可以读出来apache的路径
/usr/local/apache/conf/httpd.conf
/var/www/html/apache/conf/httpd.conf
/home/httpd/conf/httpd.conf
/usr/local/apache2/conf/httpd.conf
/usr/local/httpd/conf/httpd.conf
/etc/apache/httpd.conf
/usr/local/lib/php.ini
/etc/hosts.deny 定义禁止访问本机的主机
/etc/bashrc bash shell 的系统全局配置
/etc/group 系统用户组的定义文件
/etc/httpd/httpd.conf
/etc/issue 显示Linux核心的发行版本信息（用于本地登陆用户）
/etc/issue/net 显示Linux核心和发行版本信息（用于远程登陆用户）----没成功
/etc/ssh/ssh_config ssh配置文件
/etc/termcap 终端定义和配置文件
/etc/xinetd.d
/etc/mtab 包含当前安装的文件系统列表 有时可以读取到当前网站的路径
redhat-release:包含识别当前Red Hat 版本号的字符串
shells:列出可用在系统上的shell命令行解释器(bash,sh,csh等).
/etc/vsftpd/vsftpd.conf
/etc/xinetd.conf xinetd 配置文件
/etc/protocols 列举当前可用的协议
/etc/logrotate.conf 维护 /var/log 目录中的日志文件
/etc/ld.so.conf “动态链接程序”（Dynamic Linker）的配置。
我在我的系统中安装了非常有用的 wget 实用程序。/etc/ 中有一个 /etc/wgetrc 文件
/etc/wgetrc
Linux操作系统用户配置文件
/etc/passwd
/etc/shadow
/etc/inputrc
DNS客户机配置文件，设置DNS服务器的IP地址及DNS域名
/etc/resolv.conf
内容为Default Router的ip地址
Redhat 5.x: /etc/sysconfig/network
/etc/sendmail.cf (Linux) Sendmail(EMAIL服务器)配置文件
/etc/sendmail.cw 本地主机名
```

二、任意文件读取/下载漏洞代码审计

代码审计流程大致分为下面几步，首先是确定功能是否存在文件读取/下载功能，其次是分析文件参数是否可控，再其次分析路径是否可控，如果存在路径限制则尝试绕过，最终经过一系列分析确定是否存在任意文件读取/下载漏洞。

任意文件读取/下载漏洞代码审计本身不难，确定了功能点后，如果后端直接接受前端传来的文件名，没有对路径做限制，那大概率存在任意文件读取/下载漏洞。当然具体情况还得具体分析。

如果存在路径限制，这部分属于目录穿越漏洞范畴了，下节进一步讲解。

1、确定功能点

确定目标系统是否存在读取或下载功能方式很多。可以通过阅读使用手册，官方文档，部署环境后前端定位功能，后端关键字查找。

下面是一些关键字。

```
org.apache.commons.io.FileUtils
org.springframework.stereotype.Controller
import java.nio.file.Files
import java.nio.file.Path
import java.nio.file.Paths
import java.util.Scanner
sun.nio.ch.FileChannelImpl
java.io.File.list/listFiles
java.io.FileInputStream
java.io.FileOutputStream
java.io.FileSystem/Win32FileSystem/WinNTFileSystem/UnixFileSystem
sun.nio.fs.UnixFileSystemProvider/WindowsFileSystemProvider
java.io.RandomAccessFile
sun.nio.fs.CopyFile
sun.nio.fs.UnixChannelFactory
sun.nio.fs.WindowsChannelFactory
java.nio.channels.AsynchronousFileChannel
FileUtil/IOUtil
BufferedReader
readAllBytes
scanner
```

上面是给出的文件操作类关键字，这些关键字不仅仅能定位到文件读取或下载操作，还会涉及到一些比如文件删除，文件移动，文件遍历等操作。

总之上面通过关键字定位到文件操作类功能时，大家都可以进一步审计，也许还会存在任意文件删除，任意文件遍历，任意文件移动等漏洞。换汤不换药，后面实战中遇到再进一步讲解吧。

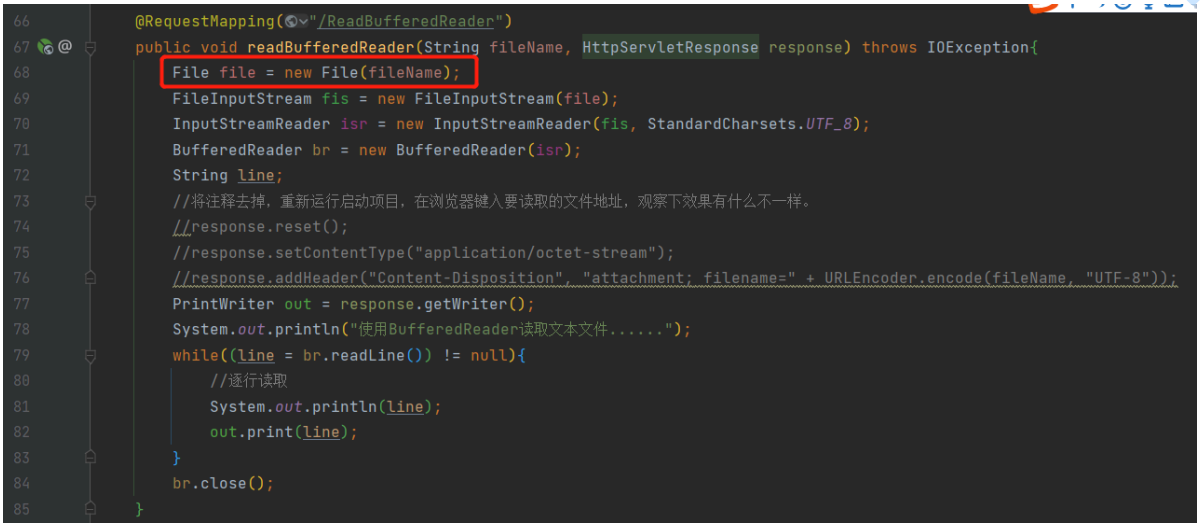
2、文件参数可控

打开`webreadfile`项目工程。以`ReadFilesController`中第`66到85行`代码为例。

在确定了项目中存在文件读取/下载功能后，我们进一步进行代码审计。

第一步查看文件名从何而来。

从第67，68行可以确定，后端接受前端传来的文件名，也就意味着文件名我们可控。如下图所示：



3、路径无限制

第二步，我们查看路径是否可控，是否有限制。

分析下代码。

- 第68行：File file = new File(fileName);

创建文件类，其中fileName是前端传来的。

- 第69行：FileInputStream fis = new FileInputStream(file);

创建文件字节输入流。

- 第70行：InputStreamReader isr = new InputStreamReader(fis, StandardCharsets.UTF_8);

读取字节流通过指定的字符集解码为字符流。

- 第71行：BufferedReader br = new BufferedReader(isr);

将字符流放到字符流缓冲区之中。

- 第72行~85行：读取/下载文本文件

逐行读取文件内容。

```
    @RequestMapping("/ReadBufferedReader")
    public void readBufferedReader(String fileName, HttpServletResponse response)
throws IOException{
        File file = new File(fileName);
        FileInputStream fis = new FileInputStream(file);
        InputStreamReader isr = new InputStreamReader(fis,
StandardCharsets.UTF_8);
        BufferedReader br = new BufferedReader(isr);
        String line;
        //将注释去掉，重新运行启动项目，在浏览器键入要读取的文件地址，观察下效果有什么不一样。
        //response.reset();
        //response.setContentType("application/octet-stream");
        //response.addHeader("Content-Disposition", "attachment; filename=" +
URLEncoder.encode(fileName, "UTF-8"));
        PrintWriter out = response.getWriter();
```

```
        System.out.println("使用BufferedReader读取文本文件.....");
        while((line = br.readLine()) != null){
            //逐行读取
            System.out.println(line);
            out.print(line);
        }
        br.close();
    }
```

整个操作中对路径并没有任何限制，可以确定存在任意文件读取漏洞。

在实际项目中，会更严格规范一些，比如会用if判断文件名是否为空，是否存在该文件等操作。

并且大多情况下，代码中会设置读取/下载文件目录，比如：

```
String path = "C:\\Users\\powerful\\Desktop\\";
String filePath = path + fileName;
```

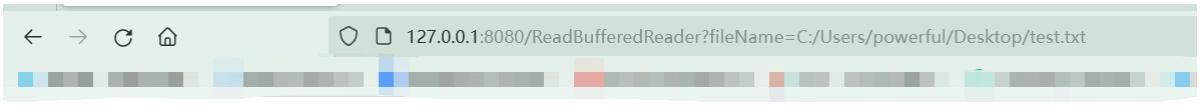
如果是上述情况，需要审计代码中是否过滤了`../`，如果没有则可以尝试配合目录穿越`../`来读取敏感文件。

这节暂且不谈。视角继续回到本节代码中发现此不没有任何限制。并且关于路径限制绕过，将放在下一节目录穿越漏洞中讲解。

4、任意文件读取/下载漏洞验证

启动webreadfile项目。我们以读取桌面中某个文件为例。

访问`http://127.0.0.1:8080/ReadBufferedReader?fileName=C:/Users/powerful/Desktop/test.txt`。



GG!

webreadfile项目还有其他基础文件读取/下载代码，请大家自行分析，研究，调试！