一、文件上传

为了让用户将文件上传到网站，就像是给危机服务器的恶意用户打开了另一扇门。即便如此，在今天的现代互联网的Web应用程序，它是一种常见的要求，因为它有助于提高业务效率。企业支持门户，给用户各企业员工有效地共享文件。允许用户上传图片，视频，头像和许多其他类型的文件。向用户提供的功能越多，Web应用受到攻击的风险和机会就越大，这种功能会被恶意用户利用，获得到一个特定网站的权限，或危及服务器的可能性是非常高的。上传文件本身没有错，问题与漏洞在于服务器怎么处理上传的文件。

1、文件上传代码

```
filename = filename.substring(filename.lastIndexOf("\\")+1);
//获取item中的上传文件的输入流
InputStream in = item.getInputStream();
//创建一个文件输出流
FileOutputStream out = new FileOutputStream(savePath + "\\" + filename);
//创建一个缓冲区
byte buffer[] = new byte[1024];
//判断输入流中的数据是否已经读完的标识
int len = 0;
//循环将输入流读入到缓冲区当中，(len=in.read(buffer))>0就表示in里面还有数据
while((len=in.read(buffer))>0){
    //使用FileOutputStream输出流将缓冲区的数据写入到指定的目录(savePath + "\\" + filename)当中
    out.write(buffer, 0, len);
}
//关闭输入流
in.close();
//关闭输出流
out.close();
//删除处理文件上传时生成的临时文件
item.delete();
message = "文件上传成功！";
```

2、文件上传漏洞必须满足的几个条件

2.1 文件上传功能能正常使用。

2.2 文件类型允许上传

2.3 上传路径可以确定

2.4 文件可以被访问、可以被执行或者被包含

2、修复建议

白名单限制：获取文件后缀，判断是否在白名单内

黑名单限制：获得文件后缀名，判断是否在黑名单内。

重命名后缀

代码：

```
//处理获取到的上传文件的文件名的路径部分，只保留文件名部分
filename = filename.substring(filename.lastIndexOf("\\")+1);
String lastfilname = filename.substring(filename.indexOf(".")+1,filename.length());
if("txt".equals(lastfilname))
```

```
//注意：不同的浏览器提交的文件名是不一样的，有些浏览器提交上来的文件名是带有路径的，如： c:\a\b\1.txt，而有些只是单纯的文件名，如：1.txt
//处理获取到的上传文件的文件名的路径部分，只保留文件名部分
filename = filename.substring(filename.lastIndexOf("\\")+1);
String lastfilname = filename.substring(0,filename.indexOf("."));
filename = lastfilname+".txt";
```

二、文件下载

下载漏洞原理：

任意文件下载漏洞，正常的利用手段是下载服务器文件，如脚本代码，服务器配置或者是系统配置等等。但是有的时候我们可能根本不知道网站所处的环境，以及网站的路径，这时候我们只能利用./../来逐层猜测路径，让漏洞利用变得繁琐。

代码：

```
16    public void doGet(HttpServletRequest request, HttpServletResponse response)
17            throws ServletException, IOException {
18        //得到要下载的文件名
19        String fileName = request.getParameter("filename");  //23239283-92489-阿凡达.avi
20        fileName = new String(fileName.getBytes("iso8859-1"),"UTF-8");
21        //上传的文件都是保存在/WEB-INF/upload目录下的子目录当中
22        String fileSaveRootPath=this.getServletContext().getRealPath("/WEB-INF/upload");
23        //通过文件名找出文件的所在目录
24        String path = findFileSavePathByFileName(fileName,fileSaveRootPath);
25        //得到要下载的文件
```

```
35        String realname = fileName.substring(fileName.indexOf("_")+1);
36        //设置响应头，控制浏览器下载该文件
37        response.setHeader("content-disposition", "attachment;filename=" + URLEncoder.encode(realname, "UTF-8")
38        //读取要下载的文件，保存到文件输入流
39        FileInputStream in = new FileInputStream(fileSaveRootPath + "\\" + fileName);
40        //创建输出流
41        OutputStream out = response.getOutputStream();
42        //创建缓冲区
43        byte buffer[] = new byte[1024];
44        int len = 0;
45        //循环将输入流中的内容读取到缓冲区当中
46        while((len=in.read(buffer))>0){
47            //输出缓冲区的内容到浏览器，实现文件下载
48            out.write(buffer, 0, len);
49        }
50        //关闭文件输入流
51        in.close();
52        //关闭输出流
53        out.close();
```

修复建议：

1、过滤文件名不允许./ ../

2、下载文件白名单

```
http://localhost:8080/FileUploadAndDownLoad/servlet/DownLoadServlet?
filename=../WEBINF/web.xml
```