安全小课堂第102期【web漏洞挖掘之任意命令执行漏洞】

京东安全应急响应中心 7月16日

当应用需要调用一些外部程序去处理内容的情况下，就会用到一些执行系统命令的函数。当用户可以控制命令执行函数中的参数时，将可以注入恶意系统命令到正常命令中，造成命令执行攻击。

JSRC **安全小课堂第102期**，邀请到**Ximumu**作为讲师就**任意命令执行漏洞**为大家进行分享。感谢白帽子盆友的精彩提问与互动~

任意命令执行漏洞的原因及原理?

京安小妹

Ximumu：

举个例子来说： 后台的代码这么写<?php system($_GET['cmd']); ?>， ，就是通过get传入一个参数,这里的cmd参数是可以直接控制，那么我们可以通过发送请求 http://127.0.0.1:8080/?cmd=ls 来让ls命令运行

这里与任意代码执行漏洞稍微有点区别的是，代码执行漏洞是调用eval这样的函数执行php代码。

讲师

任意命令执行漏洞的利用（从代码层角度为例）

京安小妹

Ximumu：

总的来说就是，任意命令执行就是程序调用到了exec这样的函数来执行，那么我们要利用的话，，首先是，1 代码中存在调用系统命令的函数 2.函数中存在我们可控的点 3.可控点没有过滤，或过滤不严格。

如下php代码段：uri = $request['uri'];

$from = $request['from'];

$to = $request['to'];

$tmp = '/tmp/act_css_tmp_' . $uri;

system("/usr/bin/wget $from -O $tmp");

$request变量来自于用户URL的输入，最终进入到system函数里作为命令来执行，但是这段代码没有安全处理用户的输入，任意用户都可以通过如下URL来在机上执行自己的命令， php?

cmd=1190&func=sync_css&uri=hi&from=;cat /etc/passwd;&to=hi&1=2 最后导致系统的沦陷。

命令执行漏洞有如下的利用

1.存在回显示的话，可以直接读取各种配置文件，密码文件，数据库连接文件等等

2. 遇到不回显的情况，最可靠的方法使用时间延迟推断，类似与盲注的方法。通过一些命令的延时作用来判断漏洞的存在，例如ping命令

3. 不能在浏览器直接看到回显，可将命令重定向到当前目录下的文件中并查看。或者用TFTP上传工具到服务器，用telnet和netcat建立反向shell，用mail通过SMTP发送结果给自己的计算机

4. 查看自己的权限，可以提升自己权限，访问敏感数据或控制服务器。

讲师

任意命令执行漏洞的分类?

京安小妹

Ximumu：

对命令执行漏洞的分类方法挺多的，，这里我是按照漏洞的利用场景和利用方式对漏洞进行分类

通用的代码层命令执行：一些商业应用需要执行命令，商业应用的一些核心代码可能封装在二进制文件中，在web应用中通过system函数来调用：

system("/bin/program --arg $arg");

系统的漏洞造成命令执行：bash破壳漏洞（CVE-2014-6271），如果我们控制执行的bash的环境变量，就可以通过破壳漏洞来执行任意代码。

调用第三方组件存在代码执行漏洞：典型的就是WordPress中，可以选择使用ImageMagick这个常用的图片处理组件，对用户上传的图片进行处理（默认是ImageMagick库），造成命令执行。JAVA中的命令执行漏洞（structs2、ElasticsearchGroovy等

挖掘任意命令执行漏洞的奇技淫巧都有哪些呢?

京安小妹

Ximumu：

现实场景中很少会发现此类漏洞，这是因为大部分情况下代码业务主要是数据操作、文件操作、逻辑处理和api接口调用等，很少直接使用系统命令。需要通过手工测试的方式先定位到哪些业务功能有可能使用到了外部程序，然后进一步构造payload进行攻击。

一般挖掘漏洞都可以分为 黑盒，白盒 和灰盒挖掘。

白盒测试

可以代码审计的话,直接搜索含有常用的调用函数system、exec、shell_exec、passthru、pcntl_exec、popen、proc_open,以及反引号也可以执行命令。然后联系上下文看看有没有课控制的输入参数,可控制的点指的是,我们可以传入参数,如果在代码里写死了命令:<?php system("ipconfig");?>就是无法利用的,针对可控制的参数,在进一步绕过过滤限制。

黑盒测试

黑盒挖掘任意命令执行漏洞要点在于找到可能调用第三方命令的业务场景，很多时候要半蒙半猜的去想后台代码是什么样的。通常在图片处理、大文件压缩、文件格式转化、日志处理以及数据库导出等功能比较容易调用一些小脚本进行辅助处理。能够确定某个业务模块使用到了第三方工具，就可以进一步对命令注入语句进行分析，是否存在各种限制，最常见的用各种fuzz推测后端对输入进行了哪些限制，对其进行相应的绕过。构造出可以利用的payload。

此外，应该尽量多的收集目标使用的各种组件信息，查找以往是否有暴过任意命令执行漏洞，测下是否在目标上依然存在这些漏洞。

讲师

分享一些经典的任意命令执行漏洞的案例?

京安小妹

Ximumu：

任意命令执行漏洞的案例

目前网络上流传的任意命令执行漏洞案例还是挺多的，这里分享的是【CVE-2016-3714】ImageMagick远程代码执行漏洞ImageMagick是一个免费图片处理的软件。该漏洞产生于一个 mvg文件转化成jpg图片的功能。

原理：

ImageMagick在实现该功能的时候有一个功能叫做delegate，是通过调用外部的lib来处理文件。而调用外部lib的过程是使用系统的system命令来执行的，整个执行流程为：ConvertMain() -> MagickCommandGenesis()-> ConvertImageCommand() -> ReadImages() -> ReadImage() ->ReadMVGImage() -> DrawImage() ->ReadImage() -> InvokeDelegate() -> system()

其中，ConvertImageCommand() -> ReadImages()-> ReadImage()这个环节，主要做了读取判断文件名类型,根据文件类型调用相应的decoder,而调用decoder的方式使用的是delegate模式，在处理的过程中，对内容没有严格的处理，导致了任意命令执行利用方法：

a.将以下内容直接保存成exploit.mvg文件，放在imagemagic目录下，其中内容为：

```
----------
push graphic-context
viewbox 0 0 640 480
fill 'url(https://example.com "|ls -la")'
pop graphic-context
------------
```

b.执行convert命令： ./convert exploit.msg 1.jpg

生成了1.jpg文件，查看内容发现命令被执行了

我稍微搜了几个案例记着，大家有兴趣可以看看

1

D-Link Service.Cgi远程命令执行漏洞

http://www.freebuf.com/articles/terminal/164680.html

2

CVE-2018-7600 Drupal核心远程代码执行漏洞分析

https://research.checkpoint.com/uncovering-drupalgeddon-2/

https://www.anquanke.com/post/id/104697

3

3
Electron < v1.8.2-beta.4 远程命令执行漏洞
https://xz.aliyun.com/t/1990
4
Django的Secret Key泄漏导致的命令执行实践
http://www.polaris-lab.com/index.php/archives/426/
5
最新然之协同(包含专业版)及喧喧及时聊天系统远程命令执行漏洞详解
https://paper.seebug.org/534/
6
对华为HG532远程命令执行漏洞
https://xlab.tencent.com/cn/2018/01/05/a-new-way-to-exploit-cve-2017-17215/
7
D-Link 路由器信息泄露和远程命令执行漏洞分析及全球数据分析报告
https://paper.seebug.org/385/
8
[CVE-2017-11366]Codiad 漏洞挖掘笔记 (0x01) [环境搭建以及远程命令执行]
https://www.jianshu.com/p/41ac7ac2a7af
9
Supervisord远程命令执行漏洞（CVE-2017-11610）
https://www.leavesongs.com/PENETRATION/supervisord-RCE-CVE-2017-11610.html
10
Huawei HG532 系列路由器远程命令执行漏洞分析
https://paper.seebug.org/490/
11
Apache Tika 任意代码执行详细分析——【CVE-2016-6809】
https://mp.weixin.qq.com/s/kd9IlyHm_4m8iK6z9CWdtw
12
Android蓝牙远程命令执行漏洞利用实践
https://xz.aliyun.com/t/1521/
13
discuzx某远程命令执行漏洞分析
https://mp.weixin.qq.com/s?__biz=MzA5NzQxOTQ1MA==&mid=2247483680&idx=1&sn=fba748bb23b52bc1a692511793527a39&scene=1&srcid=
14
PRTG < 18.2.39 Command Injection Vulnerability

https://www.codewatch.org/blog/?p=453
15
Major Vulnerabilities in Foscam Cameras
https://blog.vdoo.com/2018/06/06/vdoo-has-found-major-vulnerabilities-in-foscam-cameras/
16
TP-Link TL-WA850RE - Remote Command Execution
https://www.exploit-db.com/exploits/44912/
17
SSD Advisory – QRadar Remote Command Execution
https://blogs.securiteam.com/index.php/archives/3689
18
Bitmain Antminer D3/L3+/S9 - Remote Command Execution
https://www.exploit-db.com/exploits/44779/
19
SSD Advisory –
TerraMaster TOS Unauthenticated Remote Command Execution
https://blogs.securiteam.com/index.php/archives/3602

讲师

任意命令执行漏洞防御方法?

京安小妹

Ximumu：

根据该漏洞的产生原理，，代码上的防御主要有三点

1.尽量少的调用执行系统命令的函数，通过黑名单的方式过滤敏感函数，如在PHP的配置文件php.ini中禁止一部分危险函数。

disable_functions=system,passthru,shell_exec,exec,popen，白名单的方式对特殊输入的类型/长度进行限制。

2.对开发者要执行特定系统命令，必须把命令转换成一个字符串，然后传给执行者（也就是 shell ），然后 shell 再解析，这个传递过程就可能会出现信息传递不对等的问题，就很容易造成实际执行命令和预期执行的产生差别。

3.如果非要使用到该功能，尽量使用 pcntl_exec 这类可以限制一次只执行一条命令并且参数为数组传入的函数而不是 system 这种直接调用 sh 去执行命令的函数，同事在执行系统命令的时候检查用户输入参数（即可控部分），在使用 pcntl_exec 之前也需要小心地检查被执行的命令是否存在执行子命令的可能。

企业角度的防御

对于企业来说，任何安全问题的防御需要从多个层面进行，从漏洞的预防-检测-防御。

1. 预防：在业务得研发初期就需要安全的接入，在软件的设计，开发，测试，运行环境等等环节进行管控，提前将可能出现的问题扼杀在初期。

2. 检测：在业务上线后，对生产环节进行周期性检测，做好各类资产管理，安全问题排查，做好风险评估工作，对系统提早发现并修复漏洞。

3. 防御: 建立完整的漏洞管理方案，漏洞应急响应流程，建立各种防御系统，在漏洞发生的时候有效的拦截。

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