VRL使用和扩展介绍

2016.08.26

VRL命令：

show exploits|vulnerabilities|payloads|tools

use exp|vul|pay

set exp|vul

run exp|vul

info exp|vul

make | stop 等

tool name

gdb

attach

aslr

guide | reload等

show命令

VRL启动时自动检查所有合格的Vulnerability， Exploit， Payload， Tool

show就是简单的列出这些名称

```
Terminal
Welcome to VRL
VRL > show exploits
================================================================================
rop
code_injection
borrowed_code_chunks
================================================================================
VRL >
```

use命令

使用一个vul|exp|pay

```
VRL > use v stack_overflow
Vulnerability Loaded.
==========================Vulnerability information:==========================
A simple server with stack overflow vulnerability.
        In current version, ASLR should always OFF, script will ignore it.
        When allow_stack_exec = True, use code_injection exploit.
        When allow_stack_exec = False, use code-reuse exploit.

============================Supported Exploits:============================
code_injection
code_reuse
=============================================================================
VRL >
```

use命令

```
VRL > use exp borrowed_code_chunks
Exploit Loaded.
==============================Exploit information:==============================
Exploit for stack_overflow with allow_stack_exec = False.
===========================Supported Vulnerabilities:===========================
stack_overflow
----------------------------------------------------------------------------
>Vulnerability exist, auto sync options(exp->vul).
VRL >
```

```
VRL > use pay print_passwd
Payload Loaded.
>Payload requirements of the exploit:
No DEP
Allow NULL byte
>Payload info:
这段shellcode的作用是读出/etc/passwd的内容
Are you sure to use the payload?(y/n):(y)
New payload loaded.
VRL >
```

set命令

```
VRL > set port 12345
Vulnerability options updated.
Exploit options updated.
VRL >
```

```
VRL > setvul port 54321
Vulnerability options updated.
VRL >
```

```
VRL > show options
==============================Vulnerability options:==============================
aslr : False
port : 54321
allow_stack_exec : True
================================Exploit options:=================================
aslr : False
default_payload : print_passwd
dIP : 127.0.0.1
port : 12345
allow_stack_exec : True
=================================================================================
VRL >
```

run命令

```
VRL > run vul
[Warring]: Options of vulnerability and exploit do not match,
Continue? y/n:(n)
```

```
VRL > run vul
Vulnerability Running...
ASLR>>OFF, may need password.

[sudo] password for readm:
kernel.randomize_va_space = 0
Script Finished.
VRL >
```

Terminal

```
waiting for connect...(port 54321)
```

run命令

```
VRL > run e
Exploit Running...
ret_addr = 0x7fffffffdf30
Script Finished.
VRL >
```

readm@ubuntu: ~/VRL/vulnerabilities/stack_overflow

```
systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
syslog:x:104:108::/home/syslog:/bin/false
_apt:x:105:65534::/nonexistent:/bin/false
messagebus:x:106:110::/var/run/dbus:/bin/false
uuidd:x:107:111::/run/uuidd:/bin/false
lightdm:x:108:114:Light Display Manager:/var/lib/lightdm:/bin/false
whoopsie:x:109:116::/nonexistent:/bin/false
avahi-autoipd:x:110:119:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/bin/false
avahi:x:111:120:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
dnsmasq:x:112:65534:dnsmasq,,,:/var/lib/misc:/bin/false
colord:x:113:123:colord colour management daemon,,,:/var/lib/colord:/bin/false
speech-dispatcher:x:114:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bin/
false
hplip:x:115:7:HPLIP system user,,,:/var/run/hplip:/bin/false
kernoops:x:116:65534:Kernel Oops Tracking Daemon,,,:/:/bin/false
pulse:x:117:124:PulseAudio daemon,,,:/var/run/pulse:/bin/false
rtkit:x:118:126:RealtimeKit,,,:/proc:/bin/false
saned:x:119:127::/var/lib/saned:/bin/false
usbmux:x:120:46:usbmux daemon,,,:/var/lib/usbmux:/bin/false
readm:x:1000:1000:readm,,,:/home/readm:/bin/bash
readm@ubuntu:~/VRL/vulnerabilities/stack_overflow$
```

info命令和其他

```
VRL > info
==============================Exploit information:==============================
Exploit for stack_overflow with allow_stack_exec = True.
============================Vulnerability information:============================
A simple server with stack overflow vulnerability.
        In current version, ASLR should always OFF, script will ignore it.
        When allow_stack_exec = True, use code_injection exploit.
        When allow_stack_exec = False, use code-reuse exploit.

VRL >
```

make，stop和run基本一致，只是调用，没有可以不写。

tool命令

```
VRL > tool link_payload

    This is an easy tool for payload linking.
    Use 'add payload' to add a new payload to your new payload.
    Use 'make' to link them all.
    'q' for quit!
Link_Payload >
```

```
Link_Payload > add
json_sample   print_passwd
Link_Payload > add print_passwd
Payload Loaded.
>Payload info:
这段shellcode的作用是读出/etc/passwd的内容
Are you sure to add the payload?(y/n):(y)y
New payload added.
Link_Payload > add json_sample
A test payload for json format.
>Payload info:
A test payload for json format.
Are you sure to add the payload?(y/n):(y)y
New payload added.
Link_Payload > make
Enter a new name:new
New payload saved.
VRL >
```

reload命令

```
Enter a new name:new
New payload saved.
VRL > show p
================================================================================
json_sample
print_passwd
================================================================================
VRL > reload
VRL > show p
================================================================================
new
json_sample
print_passwd
================================================================================
VRL >
```

attach和GDB

```
Terminal
waiting for connect...(port 34567)
```

2

```
Terminal
This GDB was configured as "x86_64-linu
Type "show configuration" for configura
For bug reporting instructions, please
<http://www.gnu.org/software/gdb/bugs/>
Find the GDB manual and other documenta
<http://www.gnu.org/software/gdb/docume
For help, type "help".
Type "apropos word" to search for comma
Reading symbols from code_injection...d
Attaching to program: /home/readm/VRL/v
ion, process 3161
Reading symbols from /lib/x86_64-linux-
r/lib/debug//lib/x86_64-linux-gnu/libdl
done.
Reading symbols from /lib/x86_64-linux-
/lib/debug//lib/x86_64-linux-gnu/libc-2
done.
Reading symbols from /lib64/ld-linux-x86-64.so.2...Reading symbols from /usr/lib
/debug//lib/x86_64-linux-gnu/ld-2.23.so...done.
done.
0x00007ffff79117e0 in __accept_nocancel ()
    at ../sysdeps/unix/syscall-template.S:84
84      ../sysdeps/unix/syscall-template.S: No such file or directory.
(gdb)
```

3

```
Terminal
Are you sure to add the payload?(y/n):(y)y
New payload added.
Link_Payload > make
Enter a new name:new
New payload saved.
VRL > show p
==================================================
json_sample
print_passwd
==================================================
VRL > reload
VRL > show p
==================================================
new
json_sample
print_passwd
==================================================
VRL > run v
Vulnerability Running...
ASLR is already OFF

Script Finished.
VRL > attach
VRL >
```

1

y using

aslr命令

```
VRL > aslr
check         conservative  off           on            status
VRL > aslr status
ASLR: OFF

VRL > aslr on
ASLR>>ON, may need password.

[sudo] password for readm:
kernel.randomize_va_space = 2
VRL > aslr check
ASLR: ON

VRL >
```

其他没什么用的

guide：显示一个简单的guide

!command：执行bash命令，!pwd

py command：执行python指令，py print ‘1’

@和@@： 执行脚本

扩展

扩展向导.md

所有的扩展样例可以在sample中找到（工具没有）。

exploit需要在exploits文件夹下新建一个以exploit名字命名的文件夹，并新建run.py。

vulnerability同上。

payload和工具直接在payloads和misc中加入.json或者.py文件。

Vulnerability篇

```
class Exploit(exploit.VRL Exploit):
    def __init__(self):
        '''Add information of your exploit here'''
        self.name = 'stack_overflow'
        self.info = 'information'
        self.options={'dIP' : '127.0.0.1',
                      'dPort' : '12345'}
        self.vulnerability= 'stack_overflow'

    def run(self):
        '''Run your exploit here, if this script could success, th
        When the exploit run, follow the options.'''
        print 'run your attack here'

'''Bellowing is default, simply ignore it.'''
if __name__ == "__main__":
    if '__init__.py' not in os.listdir(os.curdir):
        os.mknod('__init__.py')
```

Vulnerability篇

info：简介

options：设置选项

exploit：支持的exploit

run()：

可选：

make()

stop()

Vulnerability篇

```
Terminal
waiting for connect...(port 34567)

readm@ubuntu: ~/VRL/vulnerabilities/stack_overflow
readm@ubuntu:~/VRL/vulnerabilities/stack_overflow$ python run.py
Vulnerability:  stack_overflow

Checking:

Check Finished

Running:

ASLR is already OFF

readm@ubuntu:~/VRL/vulnerabilities/stack_overflow$
```

Exploit篇

```
class Exploit(exploit.VRL_Exploit):
    def __init__(self):
        '''Add information of your exploit here'''
        self.name = 'stack_overflow'
        self.info = 'information'
        self.options={'dIP' : '127.0.0.1',
                      'dPort' : '12345'}
        self.vulnerability= 'stack_overflow'

    def run(self):
        '''Run your exploit here, if this script could success, th
        When the exploit run, follow the options.'''
        print 'run your attack here'

'''Bellowing is default, simply ignore it.'''
if __name__ == "__main__":
    if '__init__.py' not in os.listdir(os.curdir):
        os.mknod('__init__.py')
```

Exploit篇

其他同Vulnerability

payload：默认的payload，如果有default_payload，可以为空

payload_info：更换payload时显示的提示要求。

options[‘default_payload’]：默认的payload名称（在VRL内的）

```
self.payload = ''
self.payload_info = 'No DEP\nAllow NULL byte'
self.info = 'Exploit for stack_overflow with allo
self.property = {'offset':24}
self.options={'dIP' : '127.0.0.1',
              'port':'34567',
              'allow_stack_exec' : 'True',
              'aslr' : 'False',
              'default_payload':"print_passwd"}
self.vulnerability= 'stack_overflow'
```

注意事项：

不要让后台等待占据终端，例如：

```
(VRL)run vul
server start...
waiting for client...
                <--VRL命令行消失，因为这时vul的执行过程。
```

解决方法：使用subprocess或module.script_tools，在新的终端调用。

```
os.popen('./vul').readlines() 或
os.system('./vul')
```

更改为：

```
from modules.tools import *
os.popen(new_terminal('./vul')).readlines() 或
os.system(new_terminal('./vul'))
```

注意事项

- 为了分离vulnerability和exploit，在不同的脚本中运行。我们更希望数据的交互是通过options，但是如果运行时才能决定的值，可以通过script_tools中的share系列函数完成数据共享。
- aslr系列命令可以通过script_tools中对应的命令加入到你的脚本中，以自动查询和修改系统ASLR状态。
- script_tools中另外一些便利函数：
  - pidof，通过程序名找到pid
  - print_line，使你的提示占据一行更加明显（然并卵）

扩展payload

payload支持两种格式：

.py和.json（工具生成的为.json）

样例在sample中

目前只有两个属性：

info

data

扩展tools

在misc文件夹下增加一个.py文件。其中的run()函数会被调用。

你可以写成单次输入的命令行形式，

也可以写成一层新的命令行交互，

甚至不需要什么输入。

目前可用的样例

vulnerability：一个栈溢出样例

exploit：三种不同的攻击方法

payload：打印passwd

misc：payload连接工具